Self-Controlling Vehicle Using PID Controller And Estimation of traffic Congestion

**Progress Report**

**In fulfillment of the requirements for the**

**NU 302 R&D Project**

**At NIIT University**

****

**Submitted by**

**Sujoy Roy – U101114FCS191  
Sachin Reddy – U101114FEC210  
Rajat Singh – U101114FCS113  
Rajat Sinha – U101114FCS114  
Anshuman Singh – U101114FCS047**

**Area**

**NIIT University**

**Neemrana**

**Rajasthan**

***CERTIFICATE***

This is to certify that the present research work entitled " Self-Controlling Vehicle Using PID Controller and Estimation of Traffic Congestion” being submitted to NIIT University, Neemrana, Rajasthan, in the fulfillment of the requirements for the course at NIIT University, Neemrana, embodies authentic and faithful record of original research carried out by Sujoy Roy, Sachin Reddy, Rajat Singh, Rajat Sinha and Anshuman Singh, students of B Tech (CSE) at NIIT University, Neemrana,. She /He has worked under our supervision and that the matter embodied in this project work has not been submitted, in part or full, as a project report for any course of NIIT University, Neemrana or any other university.

JETENDRA JOSHI

List of Figures

Fig1:

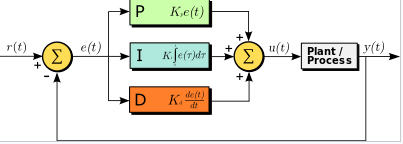


Fig2:

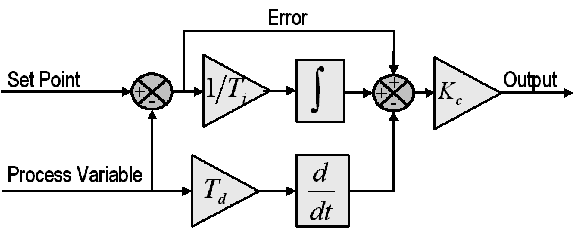


Fig3:

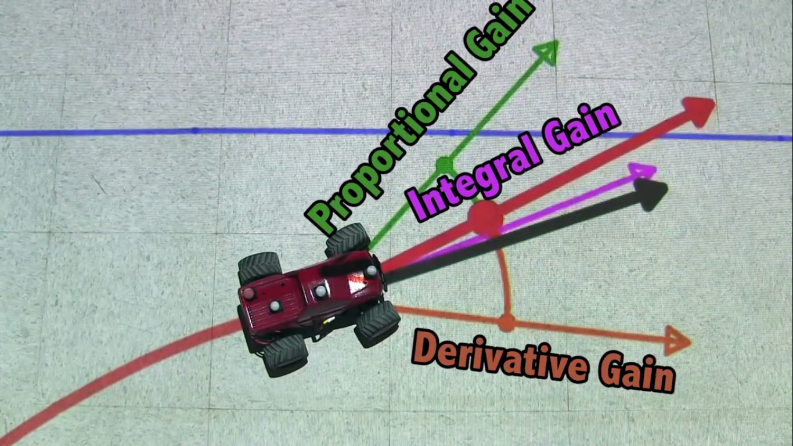


Fig4:

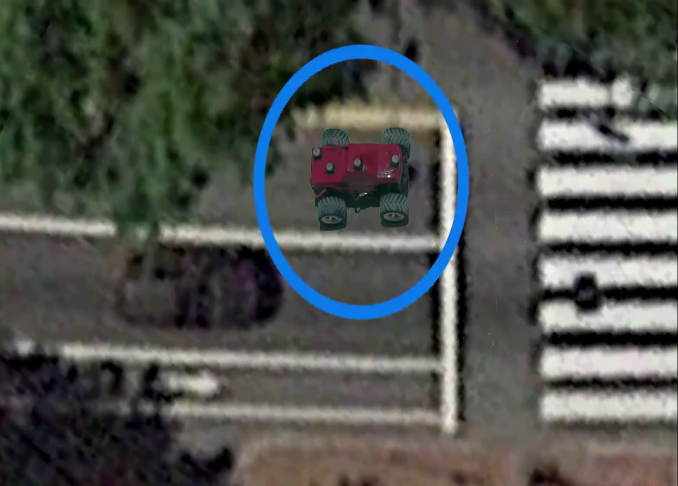


Fig5:

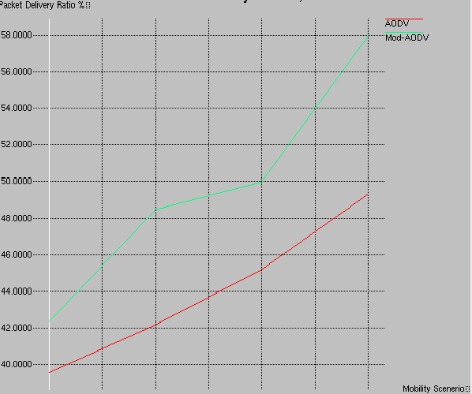
****

Fig6:

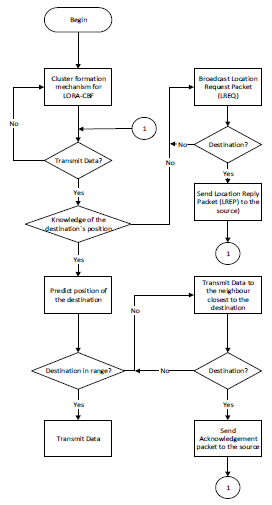


Fig7:

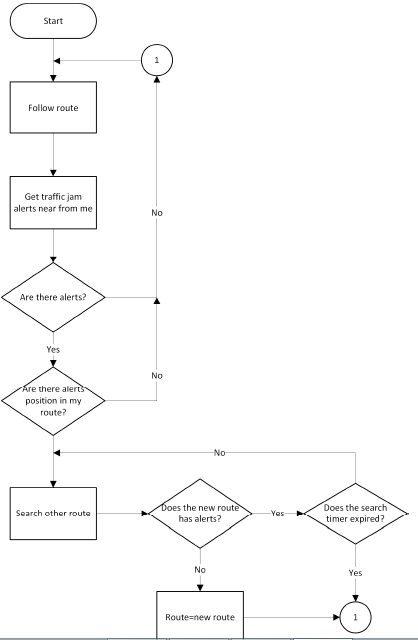


Fig8:

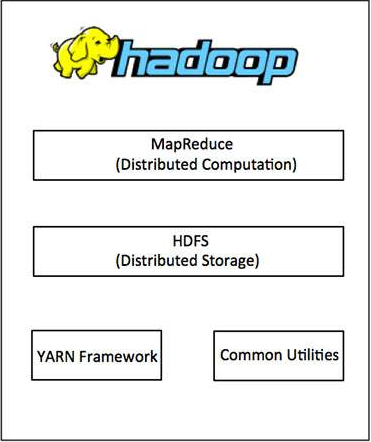


Fig9:

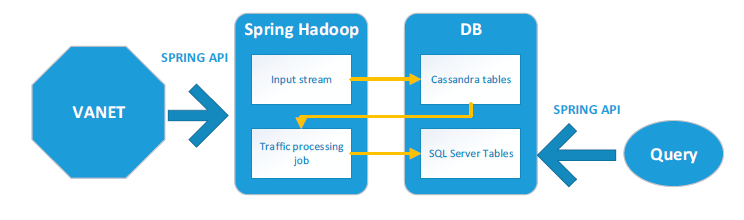


Fig10:

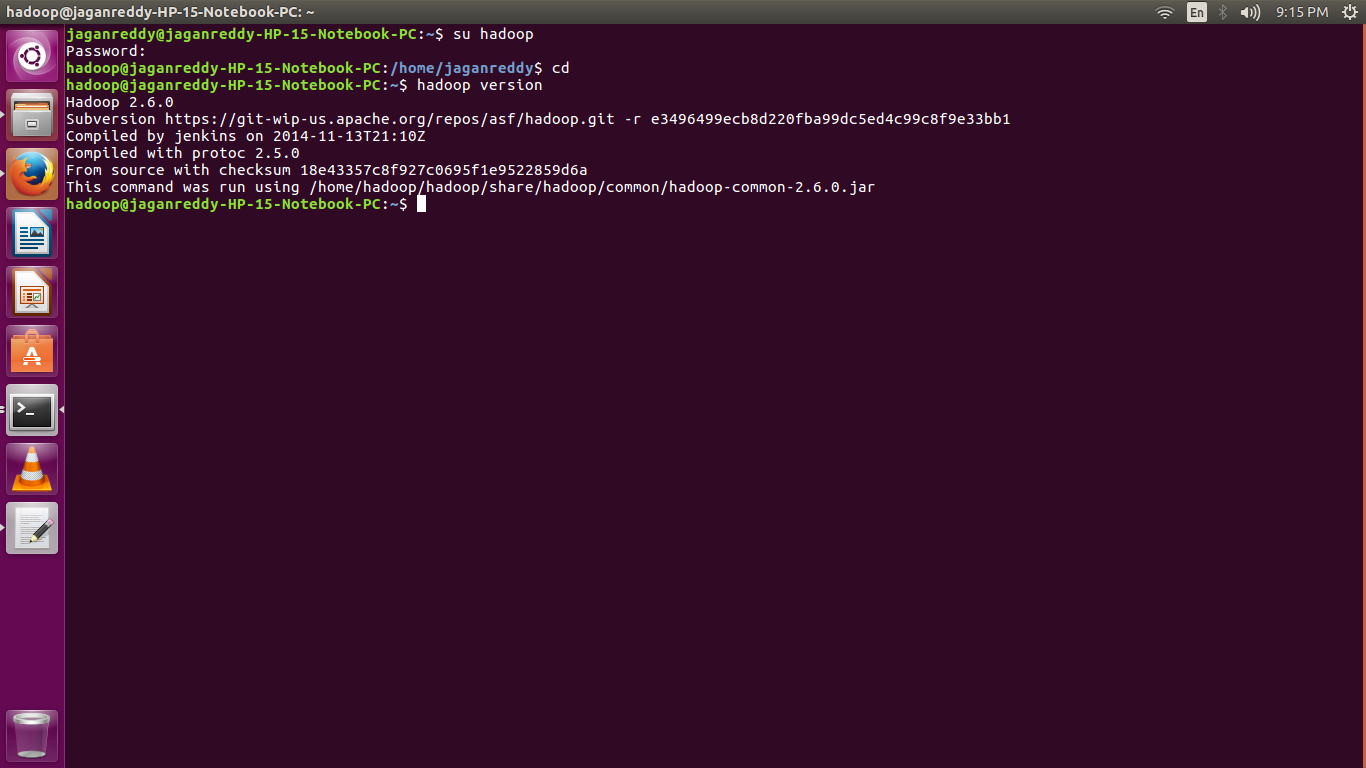


Fig11:

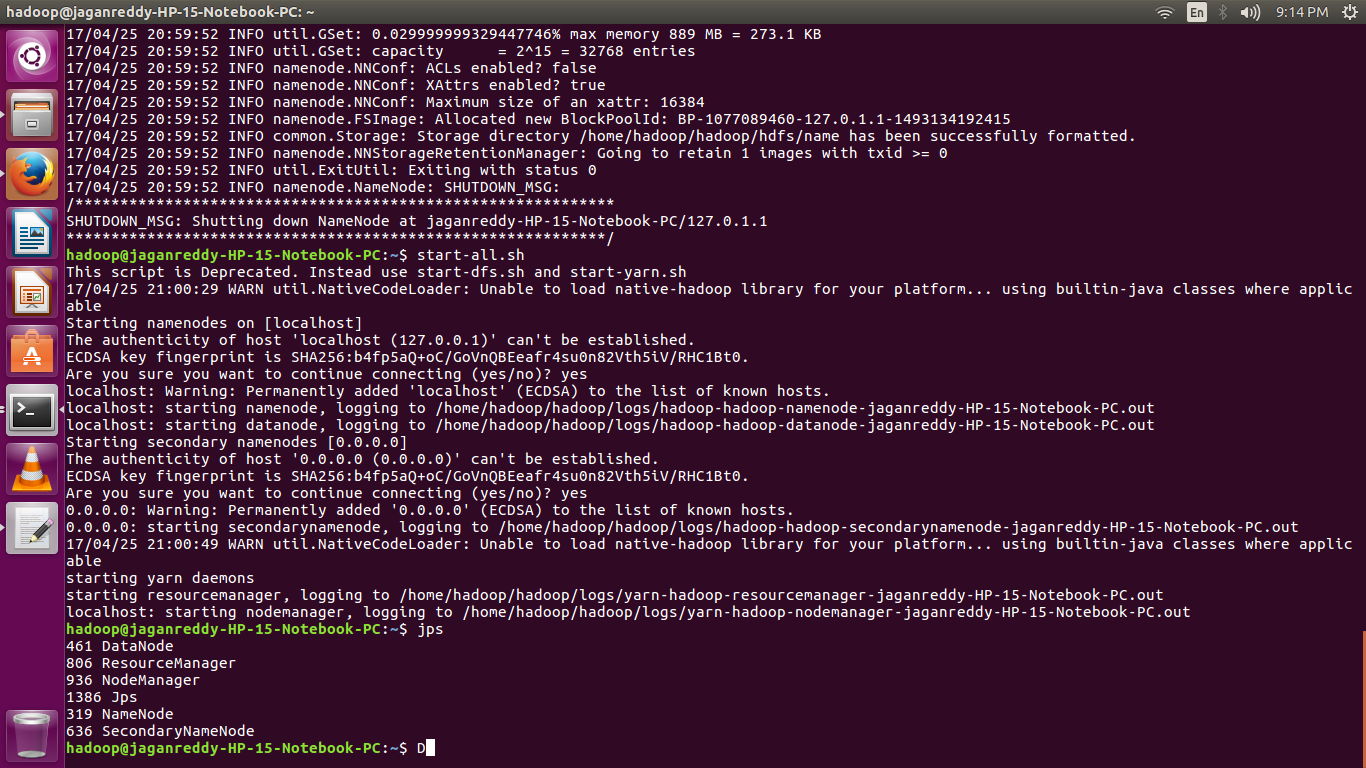


Fig12:

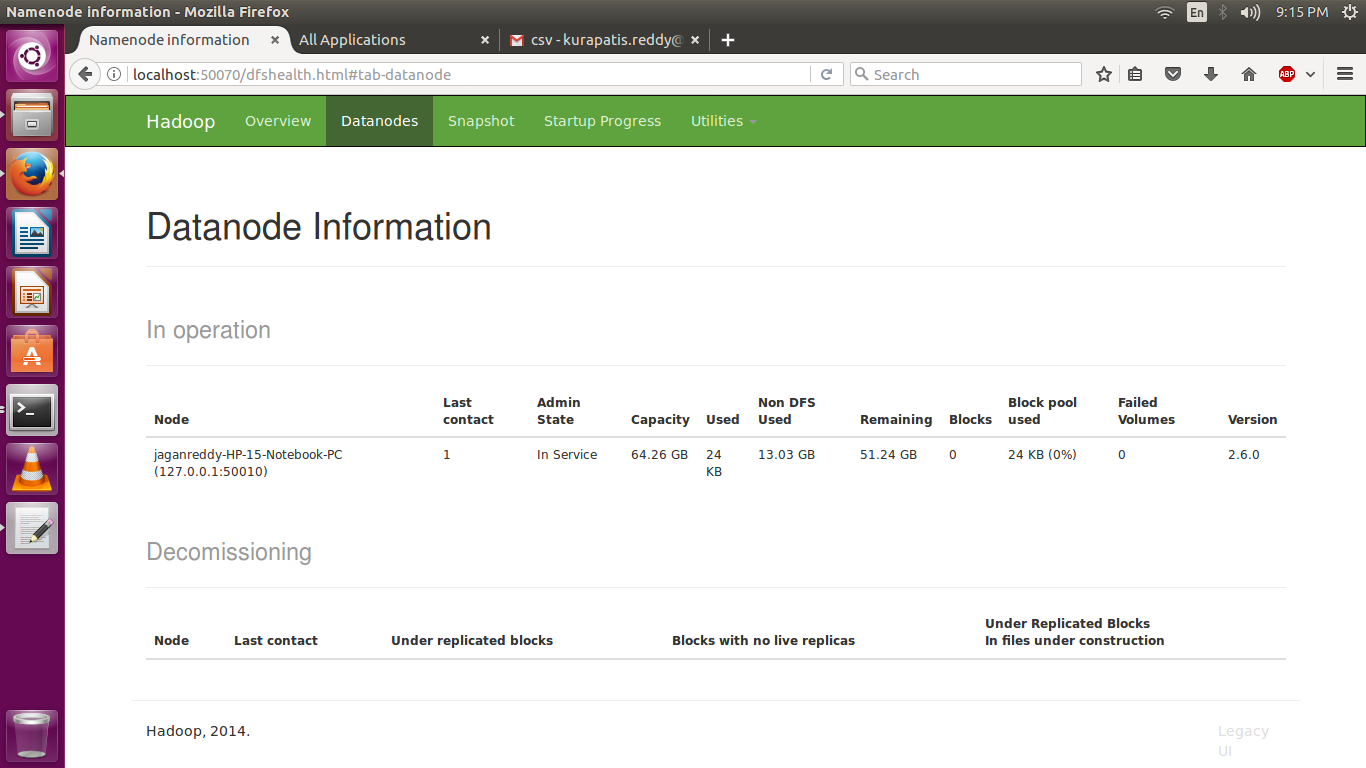


Fig13:

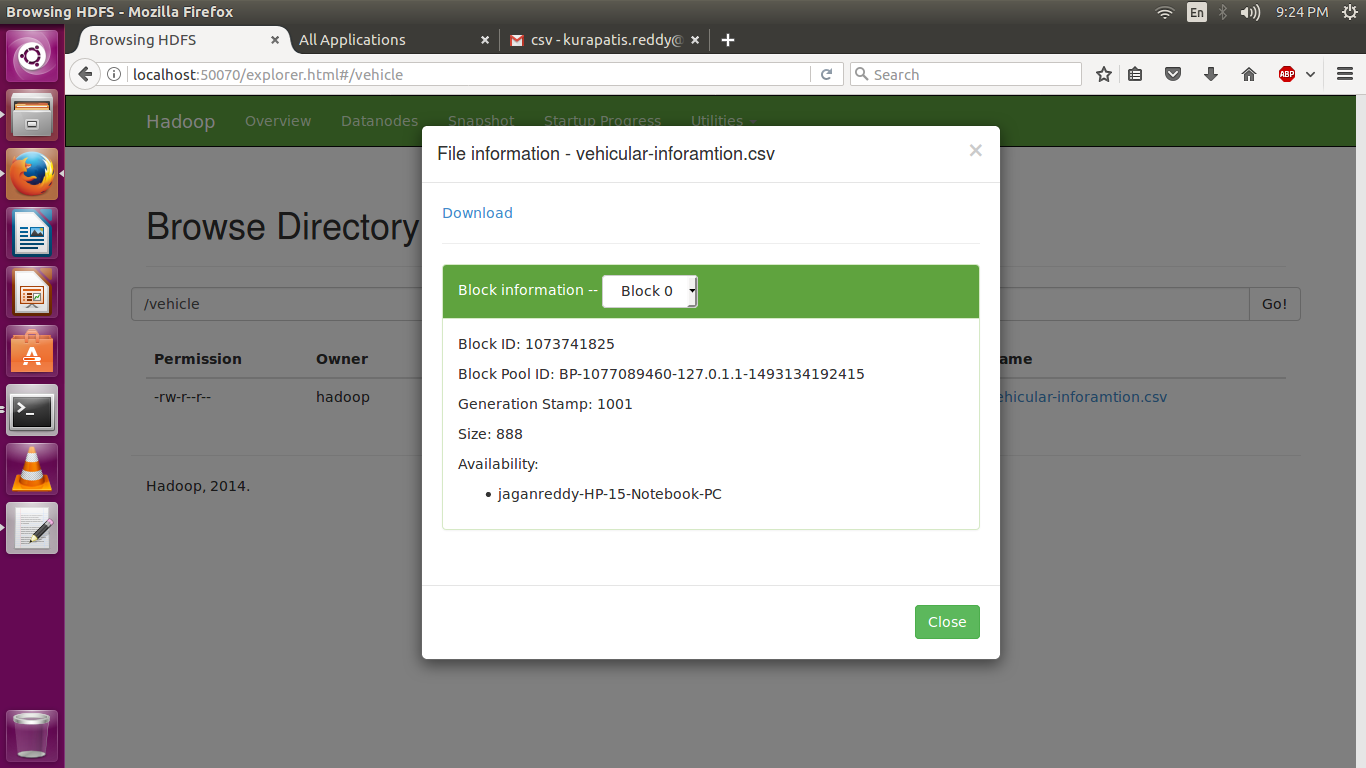


Fig14:

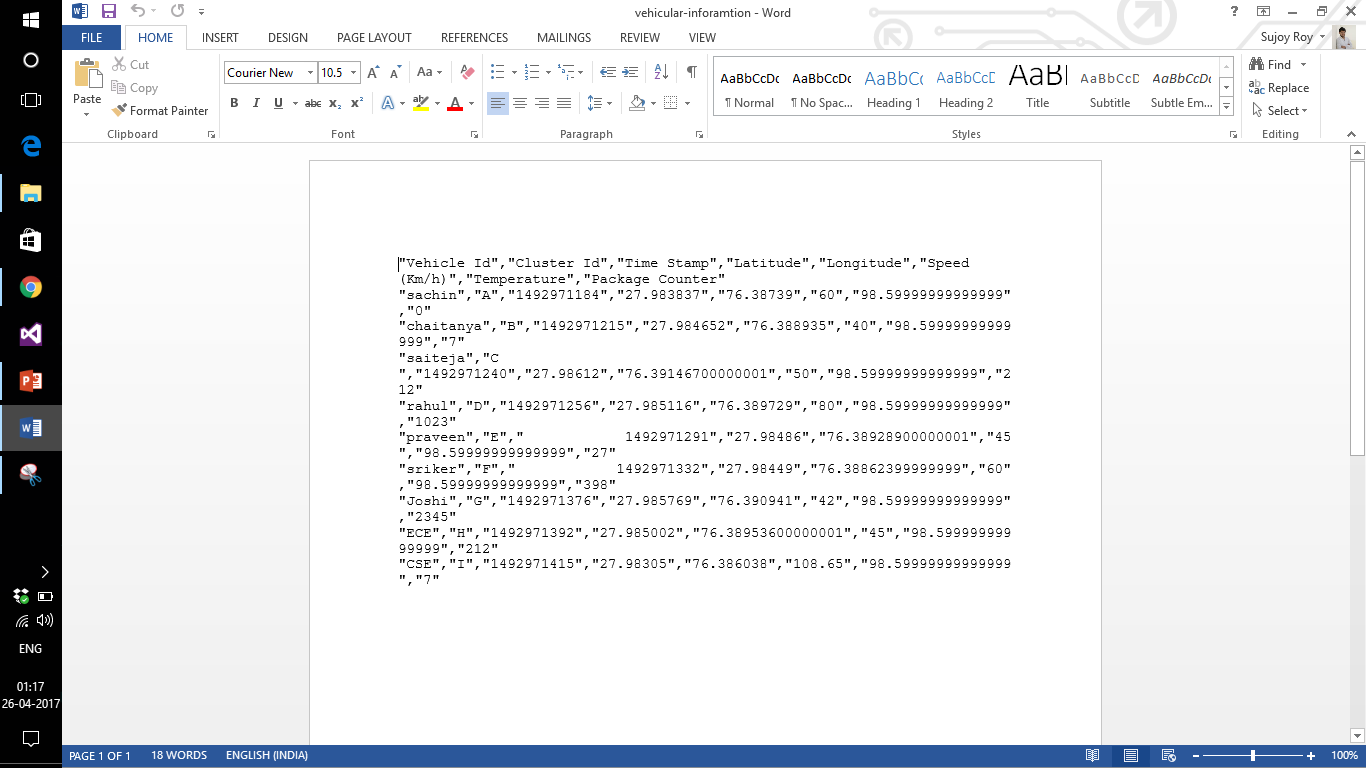


Fig15:

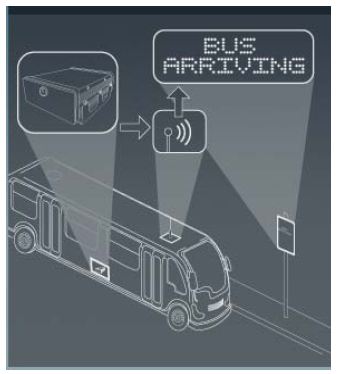


Fig16:

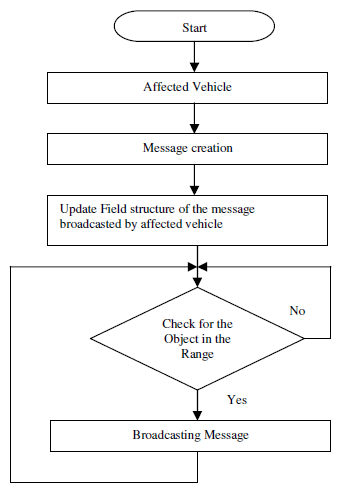


Fig17:

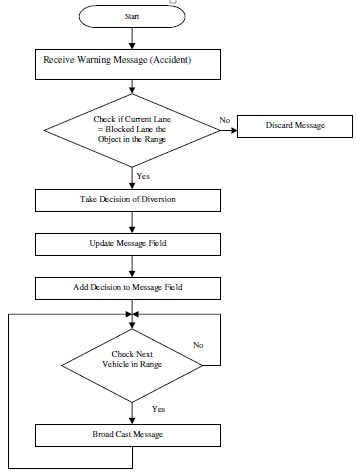


Fig18:

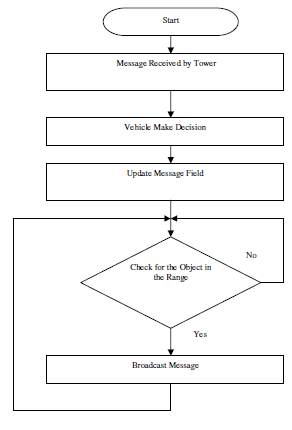


Fig19:

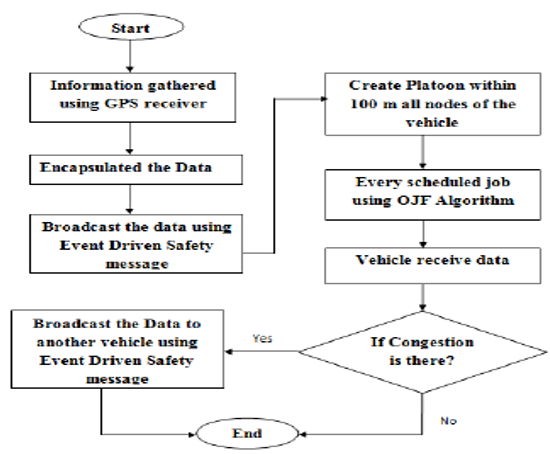


Fig20:

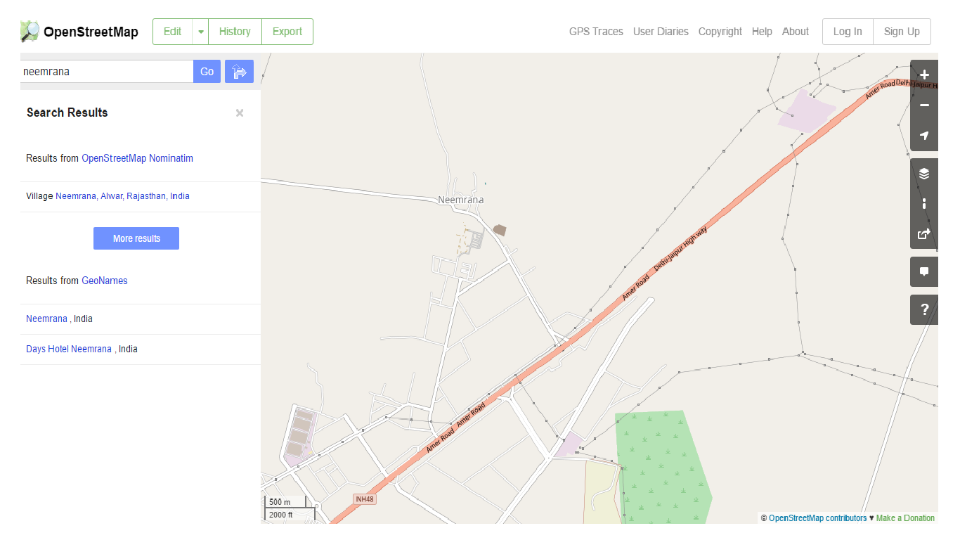


Fig21:

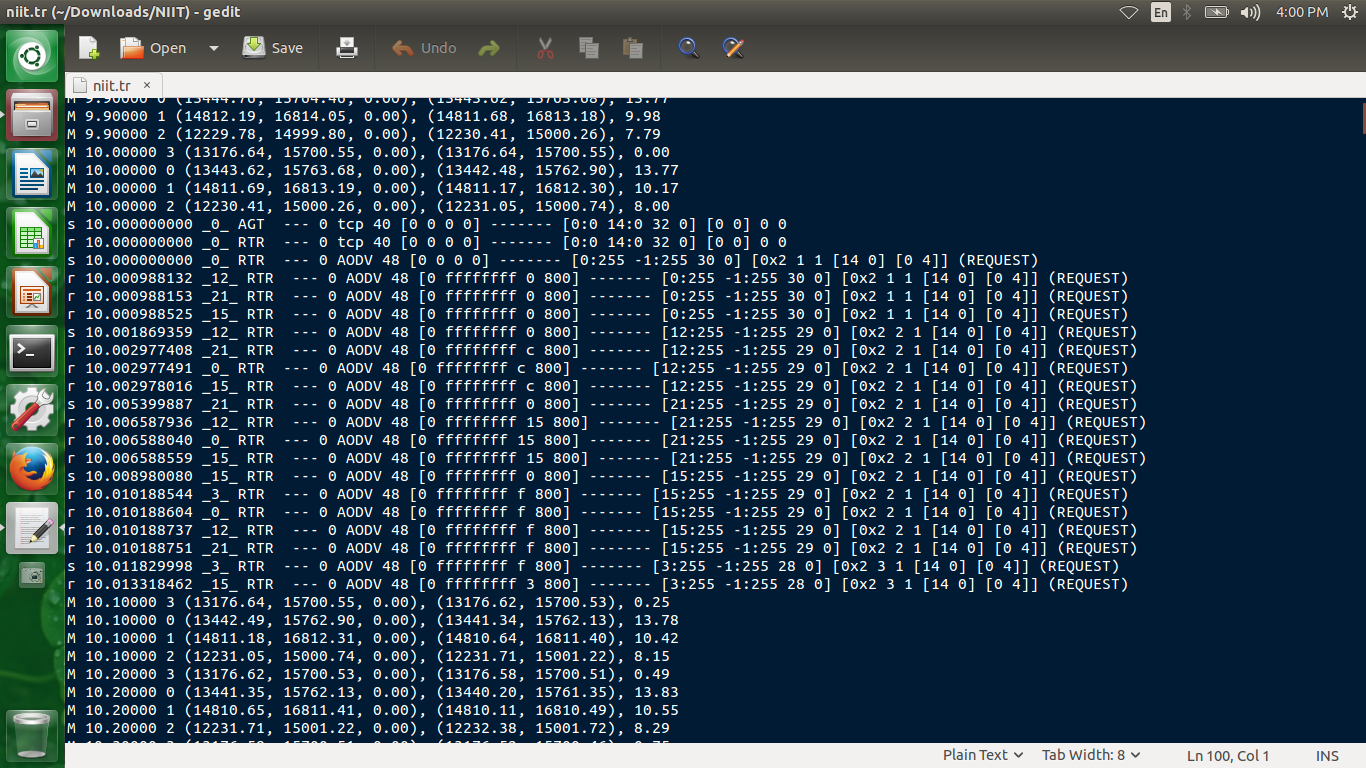


Fig22:

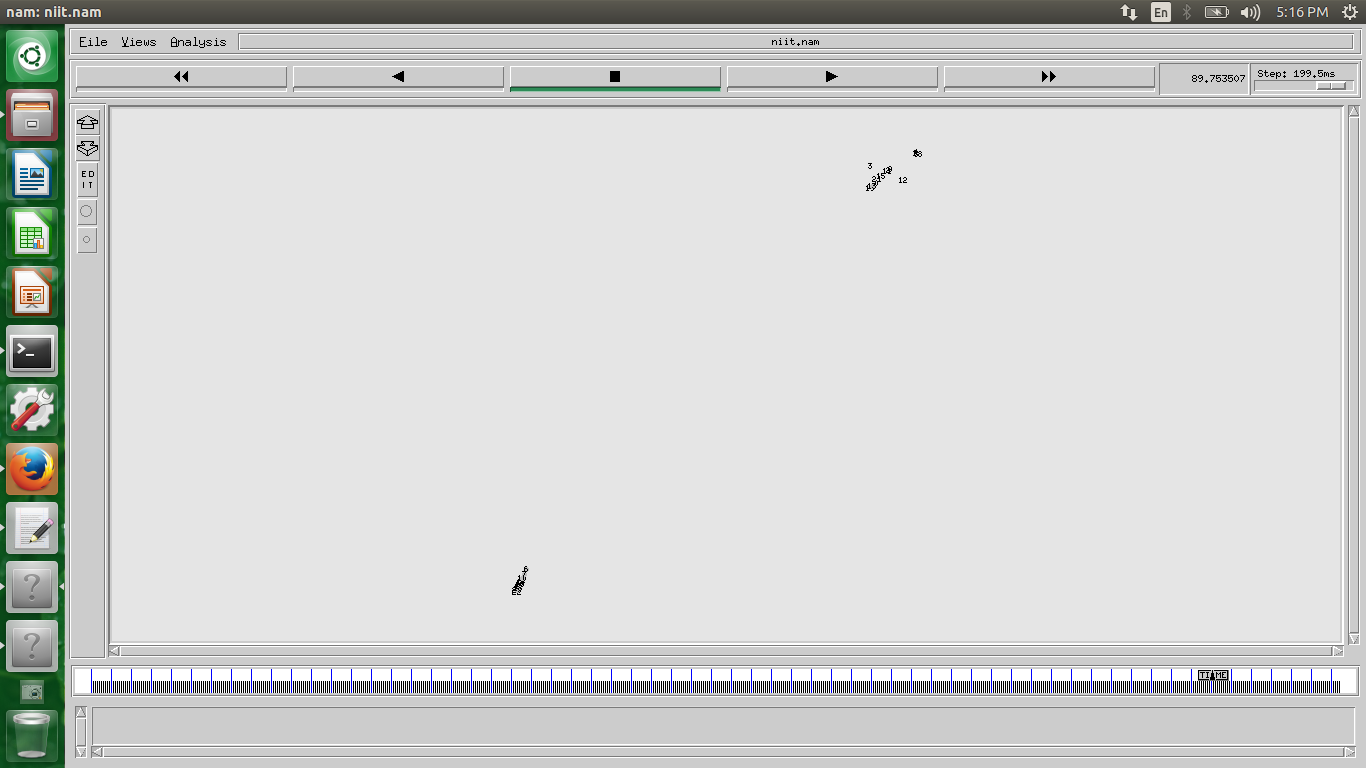


Fig23:

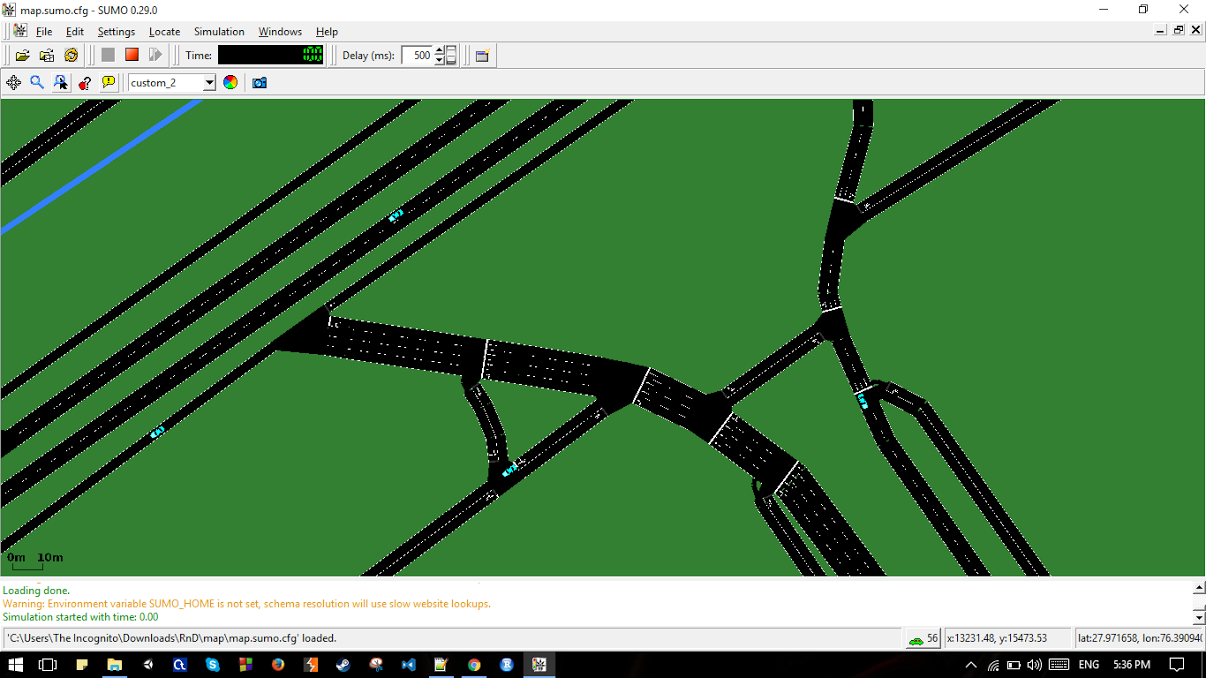


Fig24:

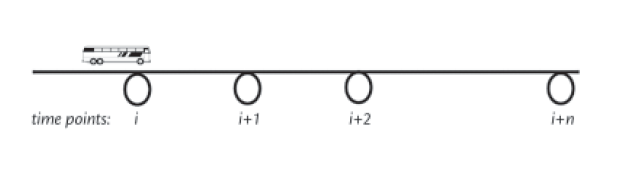
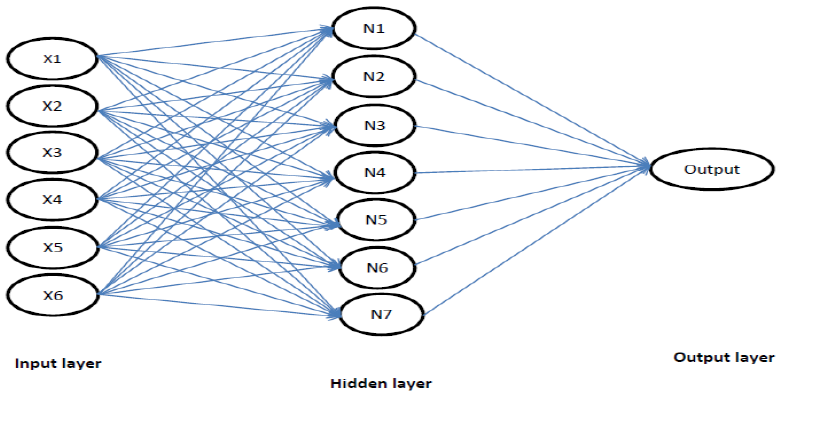


Fig25:



List of tables

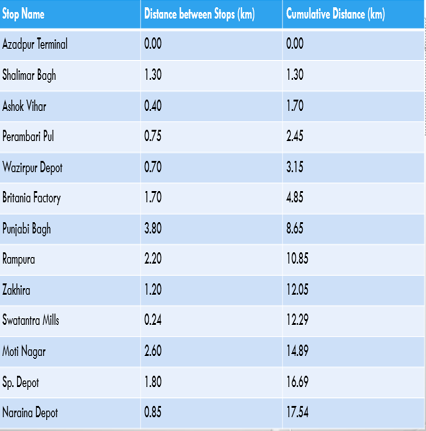
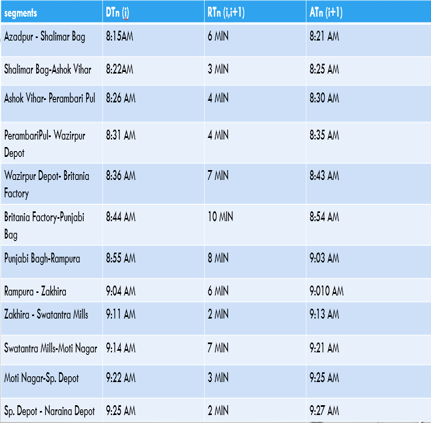
Table 1:   


Table 2:



CONTENTS

**Title Page no.**

Certificate III

List of Figures IV-XIV

List of Tables XIIV

**Rational 1**

**Literature Review 2**

**Objectives 3**

**Methodology 4**

**Results 5**

**Summary 6-24**

**Future Work 25**

**References XXVI**

CONTENTS

1. PID
2. SUMO
3. NS-2
4. Congestion and Detection Control Protocol
5. Algorithm Lora-CBF
6. Route Monitoring Algorithm
7. Hadoop
8. Field Implementation of Bus Arrival System and Prediction System.
9. Travel Recommendation from GPS Traces
10. V2V Communication
11. V2I Communication
12. Simulation
13. Prediction Methodology
14. Space Discretization Prediction Method
15. Time Discretization Prediction Method
16. Data Collection
17. Link Running Time Prediction
18. Artificial Neural Network Method
19. Kalman-Filtering Method
20. Performance Measure

Literature Review

1. Anas Mahmoud, “Non-Range Based Cooperative Localization for VANETs in Urban Environments”, Queen's University Kingston, Ontario, Canada, August 2015.
2. Dr. Lelitha Vanajakshi and Dr. Shankar C. Subramanian, “Development of a Real Time Bus Arrival Time Prediction System under Indian Traffic Conditions”, IIT Madras, June 2016.
3. Moussaoui Samira and Haouari Noureddine, “Efficient Local Density Estimation Strategy for VANETs”, Algiers, Algeria, 2013.
4. MARIAM ELAZAB, “Integrated cooperative localization in vanets for GPS denied environments”, Queen’s University Kingston, Ontario, Canada, October 2015.
5. Ang, K.H. and Chong, G.C.Y. and Li, Y. , “PID control system analysis, design, and technology”, IEEE Transactions on Control Systems Technology JULY 2005.

Objective

1. To study the implementation of PID controller and self driven automated cars.
2. To study the strategy for determining the traffic congestion of road, and its application scenario in estimation of bus travel time and parking slots allotment.
3. To study the Simulation in SUMO.
4. To study the Traffic Simulation using NS-2 and SUMO.
5. To Study the protocols of Traffic Simulation.
6. To implement the interconnection of Network Model and Traffic Model and Estimation of Traffic Congestion.
7. To store the Floating Data on Big Data Hadoop.

Methodology

1. To Study the architecture of VANET.
2. To Study the use of PID controller in Self Automated Car.
3. To Study and formulate strategy in determining the Traffic Detection.
4. To analyze the method of reducing the Traffic Congestion on roads.
5. To Calculate the Travel Time Estimation of Vehicles.
6. To Study the Traffic Simulation.
7. To Analyze the Simulation in NS-2.
8. To Analyze the Simulation in SUMO.
9. To Analyze the Simulation in NS-2 using SUMO.
10. To Study the Traffic Time Estimation of Vehicles
11. To Study the Prediction of Bus Arrival Time.
12. To Store and Retrieve the Floating Data on Hadoop.

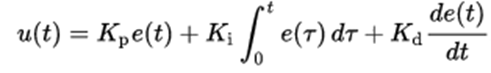
Result’s

Explained in Summary itself with the mentioned figure number.

Summary

**PID**Proportional-Integral-Derivative (PID) control is the most common control algorithm used in industry and has been universally accepted in industrial control. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity, which allows engineers to operate them in a simple, straightforward manner.

**Working**Proportional-Integral-Derivative (PID) control is the most common control algorithm used in industry and has been universally accepted in industrial control. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity, which allows engineers to operate them in a simple, straightforward manner.  
PID algorithm consists of three basic coefficients; proportional, integral and derivative which are varied to get optimal response, as shown in figure: 1.   
The output of PID controllers will change in response to change in measurement or set-point.



Where,

* Kp=Proportional gain
* Ki=integral gain
* Kd=derivative gain
* e(t)=error=Set point-process variable

**Proportional Term**

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant *K*p, called the proportional gain constant.  
The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the Error term. The proportional gain *(Kc)* determines the ratio of output response to the error signal. For instance, if the error term has a magnitude of 10, a proportional gain of 5 would produce a proportional response of 50. In general, increasing the proportional gain will increase the speed of the control system response. However, if the proportional gain is too large, the process variable will begin to oscillate. If Kc is increased further, the oscillations will become larger and the system will become unstable and may even oscillate out of control, refer figure: 2 and 3.  
Pout=Kp\*e(t)

**Integral Term**

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. Steady-State error is the final difference between the process variable and set point. A phenomenon called integral windup results when integral action saturates a controller without the controller driving the error signal toward zero. The accumulated error is then multiplied by the integral gain (*K*i) and added to the controller output, refer figure: 3.  


**Derivative Term**

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain *K*d. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, *K*d.  


Some applications may require using only one or two terms to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller is called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target value.  
It mainly causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Increasing the derivative time *(Td)* parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response. Most practical control systems use very small derivative time (Td), because the Derivative Response is highly sensitive to noise in the process variable signal. If the sensor feedback signal is noisy or if the control loop rate is too slow, the derivative response can make the control system unstable, refer figure: 3.

**Tuning Methods**Before the working of PID controller takes place, it must be tuned to suit with dynamics of the process to be controlled. Designers give the default values for P, I and D terms and these values couldn’t give the desired performance and sometimes leads to instability and slow control performances. Different types of tuning methods are developed to tune the PID controllers and require much attention from the operator to select best values of proportional, integral and derivative gains. Some of these are given below.

* **Trial and Error Method:** It is a simple method of PID controller tuning. While system or controller is working, we can tune the controller. In this method, first we have to set Ki and Kd values to zero and increase proportional term (Kp) until system reaches to oscillating behavior. Once it is oscillating, adjust Ki (Integral term) so that oscillations stops and finally adjust D to get fast response.
* **Process reaction curve technique:** It is an open loop tuning technique. It produces response when a step input is applied to the system. Initially, we have to apply some control output to the system manually and have to record response curve.
* **Zeigler-Nichols method:** Zeigler-Nichols proposed closed loop methods for tuning the PID controller. Those are continuous cycling method and damped oscillation method. Procedures for both methods are same but oscillation behavior is different. In this, first we have to set the p-controller constant, Kp to a particular value while Ki and Kd values are zero. Proportional gain is increased till system oscillates at constant amplitude.

**SUMO**SUMO - is meant to be used to simulate networks of a city's size, but can be also used for the smaller networks and larger, too, if the system power is large enough.

Continuous Road Traffic Simulation models are both stochastic (with random components) and dynamic (time is a variable). Single server queues can be modeled very well using discrete event simulation, as servers are usually at a single location and so are discrete.  
Continuous time simulation can solve the shortcoming of discrete event simulation where the model is required to have input, state and output trajectories within a time interval.

**Features**

* Consists of all applications required for a traffic simulation.
* Network Imports like import from VISSIM, OSM, XML- Descriptions, etc.
* Routing
* High Portability

**Why Only Sumo??**

* Fast and portable
* Slow visualization
* The dynamical user assignment is made within the simulation itself.
* Faster data structures

Two major design approached by SUMO are:-   
The software shall be fast and it shall be portable and because of this the very first versions of SUMO was developed to run only on the command line - no graphical interface was supplied at first and all parameter had to be inserted by hand. This should increase the execution speed by leaving off slow visualization. Also, due to these goals, the software was split into several parts. Each of them has a certain purpose and must be run individually. This is something that makes SUMO different to other simulation packages where, for instance, the dynamical user assignment is made within the simulation itself, not via an external application like here. This split allows an easier extension of each of the applications within the package because each is smaller than a monolithic application that does everything. Also, it allows the usage of faster data structures, each adjusted to the current purpose, instead of using complicated and ballast-loaded ones. Still, this makes the usage of SUMO a little bit uncomfortable in comparison to other simulation packages. As there are still other things to do, we are not thinking of a redesign towards an integrated approach by now.

**NS-2 (Network Simulator , version-2)**

1. Network Simulator version 2
2. Creates Network Topologies
3. Analyze events to understand the Network Behavior.
4. Dependable and Realistic.
5. More practical.

**TCL**

1. Tool Command Language.
2. Wireless Networking.
3. Configured Using Nodes.
4. Commands are commonly veridic.

**QoS**

1. Quality of service.
2. Self-organized network.
3. Reduces throughput and energy consumption.
4. Specific sensing and communication range, refer figure: 4.

**Throughput**

1. Amount of work done .
2. Number of successful received packets in a unit time.
3. Runs many programs simultaneously.
4. Average successful deliveries over channel.
5. Presented in BPS.

**Packet delivery ratio**

1. Ratio between the received packets by the destination and the generated packets by the source.
2. Calculated on the received and generated packets as recorded in the trace file, refer figure: 5.

**Congestion And Detection Control Protocol**

1. The affected vehicle of an accident event occur only broadcast the message (information) to nearby vehicle.
2. For the vehicle in short range, the information is transmitted through v2v communication, and for the long range it is done through v2i and rsus.
3. Each vehicle in the communication have a unique identification number (id).
4. Parameters of a vehicle : speed of the vehicle and the status of the lane to show that particular lane is blocked.
5. After receiving the message the nearby vehicle will take proper decision for diversion and then updates the message field and forwards it to other vehicles.
6. Next incoming vehicle gets decision about the blocked lane and divert form it.
7. The roadside units are also getting warning message that accident happened in particular lane, according they starts broadcasting message which assists in congestion control. This is long range communication so well in advance vehicles are getting message of slow down the speed, they have sufficient time to take proper decision and helps in avoiding the congestion.
8. Due to short range and long range communication the message are received by all the vehicles coming in range.
9. The lane changing vehicle according to the decision immediately stops the broadcasting the message thus limiting the number of messages getting broadcasted and also limiting the overloading of the system.

**Algorithm LORA-CBF**

The LORA-CBF communication algorithm works hierarchically. It is formed by a cluster head, zero or more members in each group and one of more link portals to communicate with the other cluster heads. Each cluster head contains a “Cluster Table”. A “Cluster Table” is defined as a table that contains the address and geographical location of the members and Gateway nodes. The

cluster-forming mechanism is the first to be executed and is maintained at all times as shown in the fig 9 . When a source tries to send data to a destination, it first checks the routing table to determine if the destination’s location is known. If it is, it sends the packet to the neighbor closest to the destination. Otherwise, the sources stores the data in its intermediate memory, starts a timer and Location Request (LREQ) packet transmissions. Only Gateway nodes and cluster heads can transmit a LREQ packet. Gateway nodes only transmit gateway packets to one another in order to minimize unnecessary transmissions and only if the gateways belongs to different cluster groups. By receiving a request of location, each cluster confirms that the destination is a member of its cluster. If successful, it generates a Location Reply (LREP),refer figure: 6.

**Road-Side Unit**

The RSU device is responsible for communicating vehicles with the fixed infrastructure and forwards the information to a Gateway before resending the data compiled from the vehicular network

to the big data Cluster. The RSU can function as a gateway when it connects to another infrastructure. It is comprised of a System on Module (SoM) that contains a Quad Core Cortex A9 processor at 1 GHz and a 1 GB DDR3 64 bit RAM memory. Additionally, the RSU has an 802.11 b radio that supports ad-hoc communication with 12 dBm transmission power and an omnidirectional antenna

with 9 dBi gain for V2I and I2I communication. When the RSU functions as a Gateway, it will connect via either via Ethernet to the same network segment a.s the big data Cluster, or directly to the internet to

communicate to the big data Cluster.

**On-Board Unit**

A9 quad core processor at 1 GHz with 1 GB DDR3 of RAM memory on a 64 bit architecture. Also, the OBU possesses a touch-screen and an 802.11 b radio which supports ad-hoc communication with a transmission capacity of 12 dBm and an omnidirectional antenna with a 5 dBi gain. The OBU performs the LORA-CBF communication protocol and a monitoring algorithm to propose alternate routes.

**Route Monitoring Algorithm**

The route monitoring algorithm is responsible for observing the path where the car should be free from traffic at one-minute intervals. If there is a traffic congestion alert near the car or if any alert is on the current vehicle route, the algorithm seeks and assigns a new path which contains no traffic congestion alerts. If the process of finding a new alert-free route does not succeed, or the search time expires, the same route is maintained. The route monitoring algorithm, shown in Figure: 7.

**HADOOP**

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores

Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.

Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.

Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

It is also an open source implementation of MapReduce framework.

**Components Of Hadoop**

There are mainly 3 components of Hadoop:

1. Hadoop Common Package (files needed to start Hadoop)
2. Hadoop Distributed File System: HDFS
3. MapReduce Engine

**Hadoop Distributed File System: HDFS**

Requires data to be broken into blocks. Each block is stored on 2 or more data nodes on different racks.

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware. HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing

**FEATURES OF HDFS**

* It is suitable for the distributed storage and processing.
* Hadoop provides a command interface to interact with HDFS.
* Streaming access to file system data.
* HDFS provides file permissions and authentication.

**Hadoop Architecture and Data Transmission Architecture**

Shown in Figure 8 and 9.

**TOOLS USED**

1. We Used Hadoop version 2.6.0.
2. Ubuntu 14.04.
3. Java 1.7.0\_75.
4. Cassandra 2.0.2(Hadoop Tool). (Shown in figure 10 and 11).

**DATA NODE INFORMATION**

1. This gives us the information about the cloud storage of the Hadoop hosted through our local server.
2. Generates the information about the number of nodes present and also the remaining storage data.
3. These are formed by the file systems in which our memory is used for storing the data, refer figure 12.

**FILE STORAGE**

* The data send to the Hadoop stores in the blocks of the Hadoop with the different pool ID.
* Hence it is easy for us to retrieve back the information.
* These whole blocks of data with the respective pool ID’s will be stored in the large memory block. Shown in figure 13.

**DATA USED**

The data used for the storage is Vehicular information.

This includes the information about the vehicle.

1. Latitude of the Vehicle.
2. Longitude of the Vehicle.
3. Vehicle ID.
4. Timestamp. Speed.

Shown in figure 14.

**COMMANDS USED**

1. To Start Hadoop: Start-all.sh.
2. To Stop Hadoop: Stop-all.sh.
3. To send the data to Hadoop: Copyfromlocal.
4. To retrieve data from Hadoop: Copytolocal.

**FIELD IMPLEMENTATIONS OF BUS ARRIVAL PREDICTION SYSTEMS**

1. Bus Time – With the arrival prediction system, we can calculate the time that a bus takes to go from one stop to another. This time can be calculated for every bus on a particular route.
2. Estimated time of arrival predictions – If we know the time a bus takes to go from one stop to another, we can predict the arrival time of buses.
3. Next Bus – With the knowledge of time taken by buses, we will get to know when the next bus will come.
4. Parking Lot – Imagine that one slot of parking lot is empty and with the help of arrival prediction system, we get to know that the allotted car/bus to occupy that slot is stuck in traffic or will take more time than usual, then we can allot that parking slot to some other car/bus so that efficiency of the parking lot increases.

**Travel Recommendations from GPS Traces**

There are two types of recommendations that a user can get with the help of GPS traces.

1. First is a Generic recommendation. This type recommends a user with top interesting locations/places and travel sequences in a geo-spatial region based on general data obtained from other users.
2. Second is a personalized recommendation. The interests of the user are recorded based on previous outings. Through this data, the user is recommended with locations/places and travel sequences matching his/her travel preferences.

**V2V Communication**Shown in figure 16.  
For the affected Vehicles, Shown in figure 17.

**V2I Communication**There are mainly 2 methods for V2I communication and these are:-

1. When the vehicle is in the long range then the message is received from the road side unit. I.e. tower is broadcast the message which are received by affected vehicle. The vehicles in long range received message from tower and take the suitable decision. Shown in figure 18.
2. Second Meathod:-  
   1. Data Packets Are Generated And Broadcasted By Affected Vehicle Itself   
    Which Contains Decision Message.  
   2. Data Packets Are Generated In Case Of Event Occurrence Only.  
   3. Based On Decision Vehicles Adapt The Driving Behavior And Helps In   
    Controlling Congestion.  
   4. Roadside Infrastructure Monitors The Traffic And If Traffic Is Above   
    Threshold Value It Broadcasts The Messages.  
   5. Data Are Gathered And Encapsulated In Data Packets That Are Broadcast   
    Over The Wireless Medium. This Is What Calls The Data Dissemination   
    Phase.  
   6. Data Dissemination Process Done Using “Event Driven Safety Message”.  
   7. Event Driven Safety Messages May Be Generated As A Result Of A   
    Dangerous Situation Or When Abnormal Condition Is Detected Such As   
    Road Accident. This Message Usually Has Strong Reliability And Need   
    To Be Delivered To Each Neighbor With Almost No Delays  
   8. The Jobs Are Subdivided Into Equal Size. Within 100m All The Node Of   
    The Vehicle Are Called Platoon. Nodes Are Subdivided Into Equal Size   
    Platoon. Each Job is scheduled Under Oldest Job First Algorithm Basis, Shown in Figure: 19.

**Simulation**

1. Simulation of real-time traffic using dummy data in SUMO and NS2.
2. Obtaining the floating data of each vehicle: longitude, latitude and speed.
3. Predicting traffic congestion with the help of a trace file obtained in between the process.
4. Disseminating the relevant information (using V2V and V2I) to other vehicles.
5. Open Street Map, shown in figure 20.
6. Trace File, shown in figure 21.
7. Ns-2 simulator working and it’s map, shown in figure 22 and 23.

**PSEUDO CODE FOR TRAFFIC CONGESTION**

X ----- Will be a variable, for the incoming vehicle id.

While (true) -------------------- the loop will run continuously

{

If ((xlang < lang1 && xlang > lang2) && (xlong < long1 && xlong > long2))

Count ++;

}

If (count > threshold\_value)

{

// updated message will be broadcast about traffic congestion

}

**PREDICTION METHODLOGY**

The establishment of precise bus arrival time information is basic knowledge that encourage more user to use public transport. Creating a prediction method for bus travel times can provide such information. Prediction Method can be dependent on data or may use a mathematical model. In recent times, traffic congestion has been increasing drastically in Indian due to rapid changes in urbanization, economy levels, vehicle ownership and population growth that has led to several negative impacts such as delays, pollution, etc. To solve these kind of problems, there is a need to provide more facilities by such as Bus Rapid Transit Systems (BRTS) Ahmedabad, and Metro Rail Systems (Delhi, Hyderabad, Chennai, Bangalore).There is a need to explore better traffic operations and management systems. One of the major challenges of traffic management in most of the developing countries such as India is due to discrete traffic, which comprises both fast and slow vehicles with diverse vehicular characteristics. The fast vehicles include passenger cars, trucks, buses, motor cycles, autorickshaw, whereas the slow vehicles include bicycles etc. The use of Intelligent Transportation Systems (ITS) for operation and management of traffic is a better option that is gaining interest in recent years. Attracting more travelers towards public transportation system is one way to reduce congestion, which comes under Advanced Public Transportation Systems (APTS). APTS is one of the serviceable areas of ITS that can help to attract more people towards public transport. Predicting accurate bus travel times and providing accurate information to passengers is a popular APTS application. However, the information provided to passengers should be accurate; otherwise customers may reject the system due to lack of reliability. The reliability of such information being provided depends on the prediction method and the input data used for the same.

There are many studies on prediction of travel time using various techniques such as historical and real-time approaches, statistical techniques, machine learning techniques and model-based techniques. However, there are only limited studies under heterogeneous traffic conditions such as the one existing in India. Machine learning techniques such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM) are commonly used to predict travel time because of their ability to solve complex non-linear relationships. ANN has proved to be one of the most effective tools for pattern recognition across different sets of problems. Considering anomalies in datasets, which are true for travel time across a particular stretch of road, ANN seems to be a suitable candidate for prediction. This study attempts the short term Bus Travel Time Prediction (BTTP) by developing a neural network model taking most correlated previous trips of same day and same week in to account. In this study, a particular section of the road is divided into different subsections and the network is trained separately for each subsection. The disadvantage is that these types of techniques need a large amount of data to train the system.

Model-based techniques, on the other hand, will use models that capture the dynamics of the system by establishing mathematical relationship between variables. In this study, equations that can characterize the evolution of travel time over space are used. Many model-based studies use the Kalman Filtering Technique (KFT) for estimation. Advantages of this approach are its limited data requirement and suitability for real time implementation. In this study, a model-based approach that uses data from just two previous buses will be compared with the ANN technique.

The algorithms used for travel time prediction in this study are based on dynamic traffic models.

**Space Discretization Prediction Method**

To start with, the route under consideration is divided into *N* subsections and it is assumed that the evolution of travel time of two consecutive subsections is related as :-

where,  
 x(k)is the travel time for covering the kthsubsection,   
 a(k)is a parameter which relates the travel time taken in the (k+1)thsubsection to the travel time taken in the kth subsection and   
 w(k)is the process disturbance associated with the kth subsection.

The measurement process is assumes to be given by*:-*

where,   
 z(k)is the measured travel time of the kth subsection and v(k)is the   
 measurement noise.

**Time Discretized Prediction method**

The evolution of travel time between various time intervals in a given subsection was assumed to be:-

Where,  
 a(t)is a model parameter that relates the bus travel time taken in the tth trip   
 (t+1)th trip in a particular subsection,  
 x(t)is the travel time taken for covering the given subsection in the tth trip   
 w(t)is the associated process disturbance.

The measurement process was assumed to be governed by:-

Where,

z(t)is the measured travel time in a given subsection for a trip t

v(t)is the measurement noise.

**Data collection**  
The route chosen for this study was Delhi Transport Corporation (DTC) bus route 78 in New Delhi, India. This is one of the important routes in the city, with a length of approximately 18 km and a travel time of 60 to 70 minutes. The route comprises 13 bus stops connecting Azadpur bus terminal to Naraina depot via Wazirpur, Punjabi Bagh, and Naraina, all of which are important and relatively crowded bus stops. To predict the bus running time along a particular link at instant k+1, the first algorithm, *Link Running Time Prediction Algorithm*, makes use of the last three-day historical data of the bus link running time for the instant of prediction k+1, as well as the bus link running time for the previous bus on the current day at the instant k. The study used data for the previous three days only as this was deemed practical, given the limited historical data available for the study. Obviously, in real-world applications, the algorithm can make use of longer ranges of historical data. The second algorithm, Passenger Arrival Rate Prediction Algorithm, employs similar historical data of passenger arrival rate. To predict the dwell time at a particular stop, the predicted arrival rate is simply multiplied by the predicted headway (i.e., the actual arrival time of the last bus minus the predicted arrival time of the next bus) and by the passenger boarding time.

**Link Running time Prediction**

Shown in figure 24.  
ATn (i+1) = DTn (i) +RT n (i,i+1)  
Where,

ATn (i+1) is the predicted arrival time of bus n at stop i+1,

RTn (i,i+1) is the predicted running time between i and i+1,

DTn (i) is the actual departure time of bus n from stop i

This predicted arrival time ATn (i+1) is used to predict the dwell time for bus n at stop i+1 based on the passenger arrival rate and the average passenger boarding time at stop i+1.

DWTn (i+1) = 8 (i+1) \* [ATn (i+1) - ATn-1 (i+1)] \* Davg (i+1)

Where,

DWTn (i+1) is the predicted dwell time for bus n at stop i+1,

(i+1) represents the predicted passenger arrival rate at stop i+1 from Kalman   
 filter prediction algorithm,

ATn-1 (i+1) is the actual arrival time of the previous bus n-1 at stop i+1,

ATn (i+1) - ATn-1 (i+1)] is the predicted headway for bus n at stop i+1

Davg (i+1) represents average passenger boarding time at stop i+1, assume DTO   
 be 2.5 seconds/passenger.

Having the arrival time and dwell time for bus n at stop i+1 predicted, it is now easy to calculate the predicted departure time for bus n from stop i+1 by adding the predicted arrival time to the predicted dwell time at stop i+1.

DTn (i+1) = ATn (i+1) + DWTn (i+1)

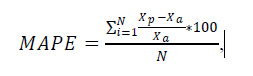
Where,

DTn (i+1) is the predicted departure time for bus n from stop i+1.  
Refer table 1 and 2, for the data.

**Artificial Neural Network (ANN):**

A neural network is a massively parallel distributed processor made up of simple processing 12 units that have a natural propensity for storing experimental knowledge and making it available 13 for later use *(16)*. This model basically tries to replicate how our brain works, its learning process 14 and response to new sets of events. There are three layers in an ANN namely the input layer, the 15 hidden layer and the output layer as shown in Figure 2. A basic unit of the connection is called a 16 neuron, which is connected by other neurons through synaptic weights. Based on the desired 17 target value, the network is trained, i.e., the weight matrix is updated after every iteration of the 18 algorithm. Thus, as the number of iterations increases, the predicted output value matrix shifts 19 closer to that of the target value, refer figure 25.

The ANN model requires a good dataset of the desired output with corresponding inputs, making up the training set. After training, the model is simulated with a new input data setto check its efficacy. The predicted value obtained after simulation and the actual target value are compared and error percentage is found out. The total error is quantified usingMean Absolute Percentage Error (MAPE), which is



Where𝑥a is the actual value and 𝑥𝑥𝑝𝑝is the predicted value. It is decided to use the six most 9 correlated trips’ travel time on that particular stretch for each day as the inputs for the neural 10 network. Thus, the numbers of nodes in the input and output.

The training rule for the ANN used in the study is



Where *J* is the Jacobian matrix that contains first derivatives of the network errors with respect to 19 the weights and biases, and *e* is a vector of network errors. The Jacobian matrix can be computed 20 through a standard back propagation technique.

**Kalman-Filter Prediction Algorithms**

As mentioned above, the prediction modeling system consists of two Kalman filter algorithms. In general, the Kalman filter is a linear recursive predictive update algorithm used to estimate the parameters of a process model. Starting with initial estimates, the Kalman filter allows the parameters of the model to be predicted and adjusted with each new measurement. Its ability to combine the effects of noise of both the process and measurements, in addition to its easy computational algorithms, has made it very popular in many research fields and applications, particularly in the area of autonomous and assisted navigation. The main assumption used in developing the Kalman filters is that the patterns of the link running time and passenger arrival rate at stops are cyclic for a specific period of day. In other words, knowledge of the link travel time and number of passengers waiting for a specific bus in a certain period of time will allow one to predict these variables for the next bus during the same period, so long as conditions remain steady. When conditions change (e.g., demand surge at a stop and/ or an incident occurred at a link), the model can update the effect of the new conditions on the predictions, so long as the new conditions persist for a sufficient length of time. The Kalman filter algorithm works conceptually as follows. The last three-day historical data of actual running times along a particular link at the instant k+1 plus the last running time observation at the instant k on the current day are used to predict the bus running time at the instant k+1. Similarly, passenger arrival rates of the previous three days at the instant k+1 plus the passenger arrival rate at the instant k on the current day are used to predict the passenger arrival rate at the instant k+1. The historical passenger arrival rate is obtained from the APC data as in this fashion: The number of on-passengers at a bus stop is divided by the most recent headway.

Generally, a Kalman filter algorithm for bus link running time has the following structure:-  
The input variance VAR [data in] is calculated at each instant k + 1 using the actual running time values for the last three days: art1 (k + 1), art2 (k + 1) and art3 (k + 1):-

VAR [data in] = VAR[art1(k + 1),art2(k + 1),art3(k + 1)]

Where:

art1 (k+1) is the actual running time of the bus at instant (k+1) on the  
 previous day.

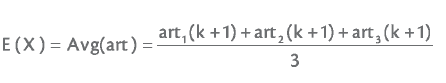
art2 (k+1) is the actual running time of the bus at instant (k+1) two days ago

art3 (k+1) is the actual running time of the bus at instant (k+1) three days   
 ago.

The definition of the variance for a random variable is:

VAR[X] = E[(X – E[X])2]

Where,



Now the variance can be calculated as shown in equation:-

D1 = [art1(k+1) – avg(art)]2

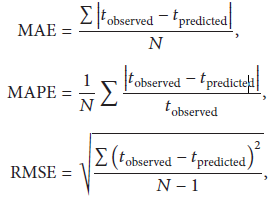
D2 = [art2(k+1) – avg(art)]2

D3 = [art3(k+1) – avg(art)]2

VAR[Data]= (D1+ D2+ D3)/3.

**Performance Measure**

After training, the model is simulated with a new input data set to check its efficacy. The predicted value obtained after simulation and the actual target value are compared and error percentage is found out. The total error is quantified using Mean Absolute Percentage Error (MAPE), which is



Where 𝑡observed is the observed bus travel time; 𝑡predicted is the predicted bus travel time; and𝑁 is the number of the bus trips observed.

Future Work

1. To compare the performance measure of various prediction models and choose the efficient.
2. A large number of OBU’s should be implanted on different vehicles.
3. Forming Multiple Cluster in Hadoop.
4. Using the NS3 version for simulation.

References

1. <http://www.ni.com/white-paper/3782/en/>
2. <https://www.elprocus.com/the-working-of-a-pid-controller/>
3. <http://ospid.com/blog/what-is-pid-control/>
4. <https://emergingtechnology.wordpress.com/2007/10/03/vanet-the-vehicular-ad-hoc-network/>
5. <http://searchcloudcomputing.techtarget.com/definition/Hadoop>