Traffic Congestion Detection and Avoidance by finding optimal path in VANET

BY :

Upadhyaya Ajay Narendrabhai

(10MCES09)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD

MAY-2013

Traffic Congestion Detection and Avoidance by finding optimal path in VANET

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Upadhyaya Ajay Narendrabhai

(10MCES09)

Guided By

Prof. Manish Chaturvedi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

MAY-2013

DECLARATION

I, **Ajay Narendrabhai Upadhyaya**, **10MCES09**, give undertaking that the Major Project entitled **"Traffic Congestion Detection and Avoidance by finding optimal path in VANET"** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in Institute of Technology of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Date: May 15, 2013 **Upadhyaya Ajay Narendrabhai**

Place: Ahmedabad

Endorsed By:

Signature of Guide:

Certificate

This is to certify that the Major Project entitled "Traffic Congestion Detection and Avoidance by finding optimal path in VANET" submitted by Ajay Narendrabhai Upadhyaya (10MCES09), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Manish Chaturvedi
Guide, Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad

Prof. Vijay Ukani
Associate Professor and P.G. Co-ordinator (CSE)
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad

Dr K Kotecha
Director,
Institute of Technology,
Nirma University,
Ahmedabad

Abstract

A Vehicular Ad-hoc Networks are self-organizing networks established among vehicles equipped with communication facilities. Due to recent advancements in vehicular technologies vehicular communication has emerged. For a rich set of applications implementing Intelligent Highways, like application related to road safety, traffic monitoring and management, road disaster mitigation etc. The road side infrastructure plays a vital role for any VANET. This is the reason that efficient communication between the vehicles and the road side infrastructure is required. Meeting this requirement becomes very difficult as nodes in a VANET are highly mobile and thus the network topology is highly dynamic. In such a state of affairs it is required to optimize the vehicle to infrastructure communication to achieve better efficiency. The objective of work is to detect traffic congestion and avoid it by finding the optimal path and give the details of all available path to the user. In this thesis, approach for finding congestion of vehicle due to overloaded traffic on the road and also suggest optimal path for avoiding traffic congestion. The proposed algorithm can reduce the traffic congestion by giving routing suggestion to new arrival vehicle, by redirecting it on the road which contain less vehicle density. We develop Car Agent (CA) and Signal Agent(SA) for Traffic Congestion detection and avoidance by suggesting optimal path to vehicle. Simulation results shows that proposed approach reduce traffic congestion of vehicles on road and also give reduction in individual trip time.

Acknowledgements

A journey is easier when you travel together. Interdependence is certainly more valuable than independence. This report is the result of work whereby I have been accompanied and supported by many people.

First of all I express my sincere gratitude, regards and thanks to my guide **Prof. Manish Chaturvedi** for his excellent guidance and continuous encouragement at all the stages of my research work. I am also thankful to **Prof. Vijay Ukani** for his support. I would also like to offer thanks to **Dr. S. N. Pradhan** whose keen interest and excellent knowledge base helped me to finalize the topic of the dissertation work.

My sincere thanks and gratitude to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University and **Dr. Sanjay Garg**, Professor and Head, Computer Science Engineering Department, Institute of Technology, Nirma University, Ahmedabad for their continual kind words of encouragement and motivation.

I am thankful to Nirma University for providing all kind of required resources. I would like to thank The Almighty, my family, for supporting and encouraging me in all possible ways. I would also like to thank all my friends who have directly or indirectly helped in making this work successful.

- Ajay Narendrabhai Upadhyaya
10MCES09

Abbreviation Notation and Nomenclature

VANET .... Vehicular Ad-hoc NETwork

MANET .... Mobile Ad-hoc NETwork

OBU .... On-Board Unit

RSU .... Road Side Units

V2I .... Vehicle to Infrastructure

V2V .... Vehicle to Vehicle

WAVE .... Wireless Access in Vehicular Environment

DSRC .... Direct Short Range Communication

SA .... Signal Agent

CA .... Car Agent

RBID .... Road Block ID

AVG_{speed} .... Average speed of vehicle

RVC_i .... Road Vehicle Counter

$RVTh_i$ .... Road Vehicle Threshold value

CF_{dist} .... Congestion Factor based on distance

CF_{time} .... Congestion Factor based on time

$Dist_{ideal}$ .... Ideal Distance

$Dist_{actual}$ .... Actual Distance

$Time_{ideal}$ .... Ideal Time

$Time_{actual}$ .... Actual time

NCTUns .... National Chiao Tung University network simulator

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Basics of VANET

The past decade has witnessed the emergence of Vehicular Ad-hoc Networks VANETs), specializing from the well-known Mobile Ad Hoc Networks (MANETs) to Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) wireless communications[1].Vehicular Ad-hoc Networks are self-organizing networks established among vehicles equipped with communication facilities. Due to recent advancements in vehicular technologies vehicular communication has emerged. For a rich set of applications implementing Intelligent Highways, like application related to road safety, traffic monitoring and management, road disaster mitigation etc. the road side infrastructure plays a vital role for any VANET. This is the reason that efficient communication between the vehicles and the road side infrastructure is required. Vehicular networking is nowadays very hot topic within the research field of telecommunications. Lot of different approaches exists, although the standard approach called IEEE 802.11p is gaining the majority of interest. original motivation for VANETs was to promote traffic c safety, recently it has become increasingly obvious that VANETs open new vistas for Internet access, distributed gaming, and the fast-growing mobile entertainment industry. The evaluation of VANET protocols and applications could be made through real outdoor experiments, which are time-costly and claim for a large number of resources in order to obtain significant results. Instead, simulation is a much cheaper and easier to use method. Obviously, this leads network and application developers to use simulation in order to evaluate different simple or complicated and innovative solutions before implementing them .

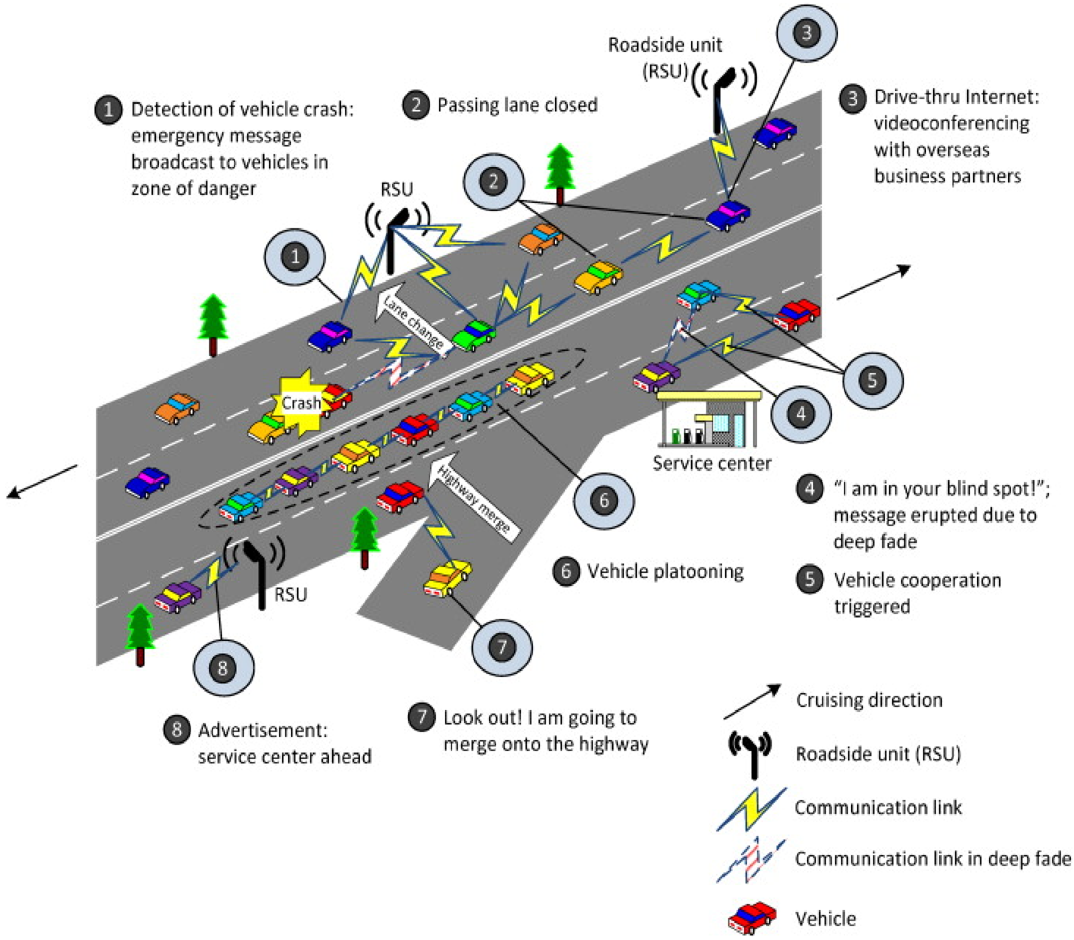


Figure 1.1: VANET

1.2 VANET Application

VANET would support life-critical safety applications, safety warning applications, electronic toll collections, Internet access, group communications, roadside service finder, etc. [2]

Life-Critical Safety Applications: e.g., Intersection Collision Warning/Avoidance, Cooperative Collision Warning, etc. In the MAC Layer, the Life-Critical Safety Applications can access the DSRC control channel and other channels with the highest priority. The messages can be broadcasted to all the nearby VANET nodes.

Safety warning Applications: e.g., Work Zone Warning, Transit Vehicle Signal Priority, etc. The differences between Life-Critical Safety Applications and Safety Warning Applications are the allowable latency requirements, while the Life-Critical Safety Applications usually require the messages to be delivered to the nearby nodes within 100 milliseconds, the Safety Warning Applications can afford up to 1000 milliseconds. In the

MAC Layer, the Safety Warning Applications can access the DSRC control channel and the other channels with the 2nd highest priority. The messages can be broadcasted to all the nearby VANET nodes.

Electronic Toll Collections (ETCs): Each vehicle can pay the toll electronically when it passes through a Toll Collection Point (a special RSU) without stopping. The Toll Collection Point will scan the Electrical License Plate at the OBU of the vehicle, and issue a receipt message to the vehicle, including the amount of the toll, the time and the location of the Toll Collection Point. In the MAC layer, the Electronic Toll Collections application should be able to access the DSRC service channels except the control channel, with the 3rd highest priority. It should be a direct one-hop wireless link between the Toll Collection Point and the vehicle.

Internet Access: Future vehicles will be equipped with the capability so that the passages on the vehicles can connect to the Internet. In the MAC layer, the Internet Access applications can use DSRC service channels except the control channel, with the lowest priority comparing with the previous applications. In the network layer, to support VANET Internet access, a straightforward method is to provide a unicast connection between the OBU of the vehicle and a RSU, which has the link toward the Internet.

Group Communications: Many drivers may share some common interests when they are on the same road to the same direction, so they can use the VANET Group Communications function. In the MAC layer, the Group Communications can use DSRC service channels except the control channel, with the lowest priority comparing with the safety related applications and ETCs. In the network layer, to support such application scenario, multicast is the key technology. In a VANET, however, since all nodes are located in a relatively local area, implementing such group communication becomes possible.

Roadside Services Finder: e.g., find restaurants, gas stations, etc., in the nearby area along the road. A Roadside Services Database will be installed in the local area that connected to the corresponding RSUs. In the MAC layer, the Roadside Services Finder application can use DSRC service channels except the control channel, with the lowest priority comparing with the safety related applications and ETCs. Each vehicle can issue a Service Finder Request message that can be routed to the nearest RSU; and a Service Finder Response message that can be routed back to the vehicle.

1.3 VANET Architecture

In comparison to other communication networks, Vehicular Ad-hoc Networks (VANETs) have unique requirements with respect to applications, types of communication, self-organization and other issues. In order to meet these requirements, the structuring of functionalities into protocols and their interaction must be re-thought for VANETs. The traditional approach of decomposition of functionality into protocol layers (layered approach) and a protocol design specifically tailored to the needs of VANETs (unlayered approach) lead to two extreme and opposed manifestations of a potential VANET protocol architecture. From these two extreme approaches we derive a stack-based VANET protocol architecture that combines the strengths of both sides. Here I discussed the three main architecture of VANET[3].

1.3.1 WAVE (Wireless Access in Vehicular Environment)

WAVE is standardised by IEEE[4]. It Supports DSRC (Dedicated sort range communication).DSRC spectrum divides into 6 service channel (SCH) and 1 Control channel (CCH).CCH used for emergency and control message and SCH is used for application packets. IF CCH active all other stop their communication. Each contains equal B/W of 10 MHZ and timeslot 50ms. In WAVE two modes of Communication: Safety communication(non IP), Non-Safety communication (IPV6)(Infotainment). It supports frequency band 5.9GHZ. IEEE introduced complete family of 1609 Protocol. This family has sub standards are: 1609.1, 1609.2, 1609.3, 1609.4, 1609.5, 1609.6. 1609.1 - Application (operational and management activities) 1609.2 - Security 1609.3 - Transport and Network Layer (Dedicated single Protocol-WSMP[Wave Short msg protocol]) 1609.4 - Channel Coordination (Multiple Channel) 1609.5 - Layer Management 1609.6 - Facilities (provide additional middle layer between transport and application layer)

Two available service set for network topology handling are : WBSS (WAVE Basic Service Set)(RSU-OBU), WIBSS(WAVE Independent Basic Service Set)(RSU-RSU).Routing is based on IP addressing. Single hope communication is done. It support Uni-cast but provide less flexibility. NCTUns simulator support provides full architecture support for WAVE.

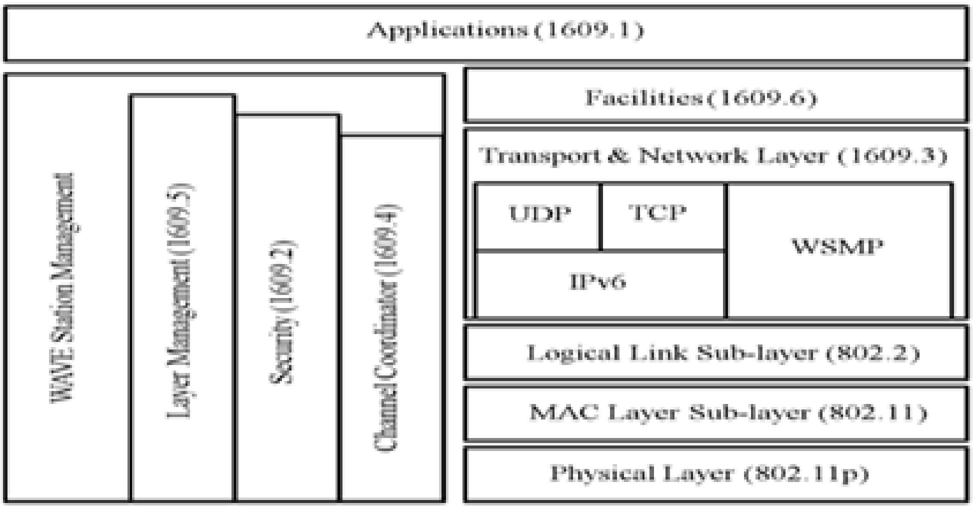


Figure 1.2: WAVE Architecture

1.3.2 CALM (Communications, Air-interface, Long and Medium creaks)

CALM is standardised by ISO[5]. It is based on CALM FAST protocol. It provides Heterogeneous communication between V2V, V2I and V2O. : COOPERS, SAFESPOT are based on CALM architecture. It uses all available interfaces which is opposite then WAVE. In backbone it uses satelight communication which is costly. Not any Simulator which support complete CALM. Architectural Components of CALM are CALM Interface Manager, CALM Network manager, CALM Application Manager. In CALM Routing is based on IP address and it Single hope communication. It support Uni-cast and Broadcast.

1.3.3 C2C-CC (Car to Car Consortium)

C2CNet is standardised by C2C Consortium[6]. It is based on C2CNet protocol. It supports internet access, Safety and Non-safety application. Routing is based on Geo-routing. It supports Single and Multiple hope communication. It supports Uni-cast, Broad-cast, Geo- Uni-cast, Geo-Broad-cast. This architecture is not support by any Open Source Simulator. It provide maximum flexibility which gives support for multiple Interface, MAC, N/W layer protocols. It Fast and Reliable due to geographical Routing. It has Sufficient Security mechanism and Good for Traffic management application (man-

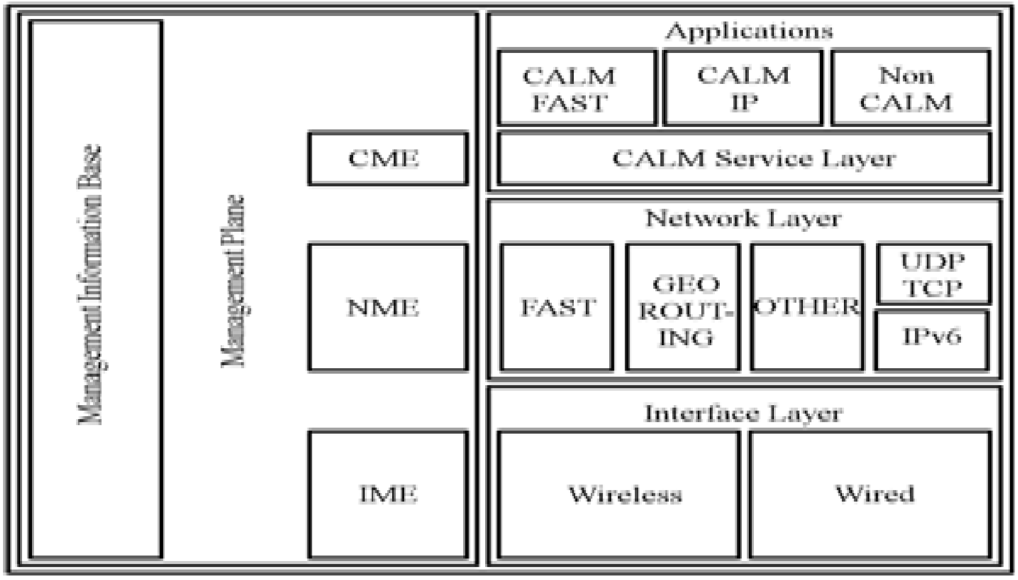


Figure 1.3: CALM Architecture

age traffic load, collision avoidance, Traffic Crossing, Red light violation, Local danger warning, Tall collection, emergency vehicle warning)

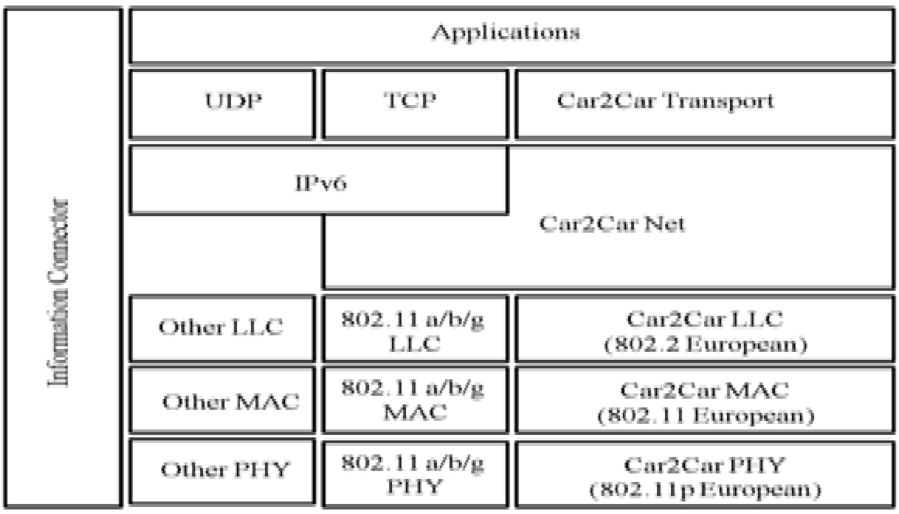


Figure 1.4: C2C Architecture

Chapter 2

Literature Survey and Important observations

2.1 Mobility Model

One of the challenges posed by the study of VANETs is the definition of a vehicular mobility model providing an accurate and realistic vehicular mobility description at both macroscopic and microscopic levels. Another challenge is to be able to dynamically alter this vehicular mobility as a consequence of the vehicular communication protocols. Many mobility models have been developed by the community in order to solve these two issues. Vehicular mobility models are usually classified as either macroscopic or microscopic. The macroscopic description models gross quantities of interest, such as vehicular density or mean velocity, treating vehicular traffic according to fluid dynamics, while the microscopic description considers each vehicle as a distinct entity, modeling its behavior in a more precise, but computationally more expensive way. The development of vehicular mobility models may be classified in four different classes: Synthetic Models wrapping all models based on mathematical models, Survey based Models extracting mobility patterns from surveys, Trace-based Models generating mobility patterns from real mobility traces, and finally Traffic Simulators-based Models, where the vehicular mobility traces are extracted from a detailed traffic simulator.[10][11][12]

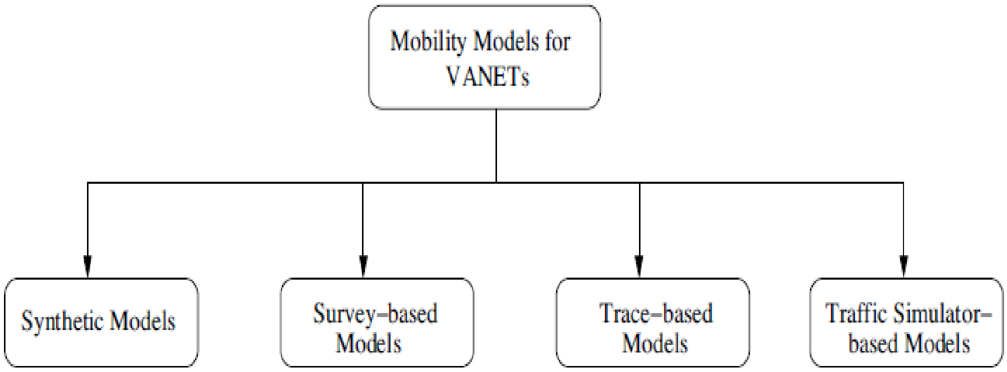


Figure 2.1: Type of Mobility Models

2.1.1 Synthetic Models

It is well known model. Major studies have been undertaken in order to develop mathematical models reflecting a realistic physical effect. According to Fiore's classification, Synthetic models may be separated into five classes: Stochastic models wrapping all models containing purely random motions, traffic stream models looking at vehicular mobility as hydrodynamic phenomenon, Car Following Models, where the behavior of each driver is modeled according to vehicles ahead, Queue Models which models roads as FIFO queues and cars as clients, and Behavioral Models where each movement is determined by behavioral rules such as social influences. the RWP ,Weighted Waypoint Model (WWM) ,Random Waypoint model (RWM) are categorized in this category.

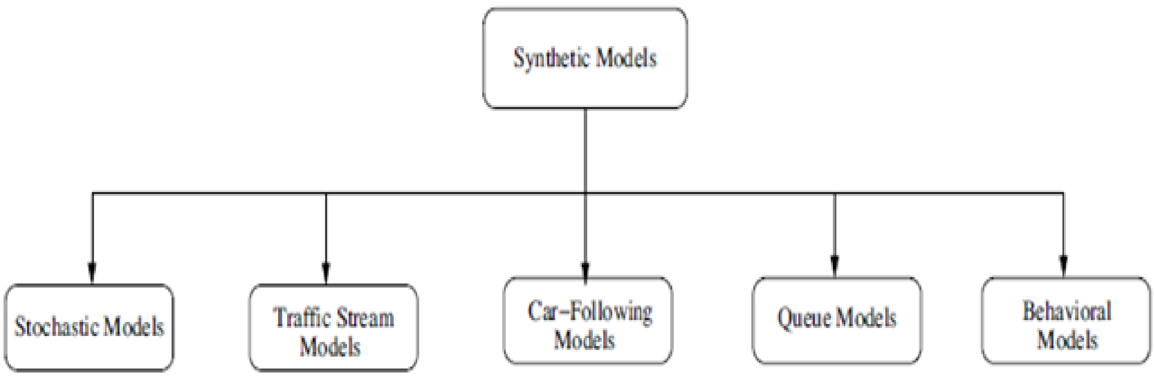


Figure 2.2: Types of Synthetic Model

2.1.2 Survey based Models

Surveys are an important source of macroscopic mobility information. The UDel Mobility Model [11] typically falls into this category. Based on data like the mobility simulator models arrival times at work, lunch time, breaks/errands, pedestrian and vehicular dynamics (e.g. realistic speed-distance relationship and passing dynamics), and workday time-use such as meeting size, frequency, and duration. Vehicle traffic is derived from vehicle traffic statistics collected by state and local governments such that it is able to model vehicle dynamics and diurnal street usage. The limitation of the survey-based approach, as survey or statistical data is only able to provide coarse grain mobility.

2.1.3 Trace-based Models

Due to the complexity of modeling vehicular mobility, only few very complex synthetic models are able to come close to a realistic modeling of motion patterns. A different approach could also follow. Instead of developing complex models and then calibrating them using mobility traces or surveys, a crucial time could be saved by directly extracting generic mobility patterns from movement traces. Limitation for the creation of trace-based vehicular mobility models is the limited availability of vehicular traces. Fleetnet and the Network on Wheels projects are based on different trace based models.

2.1.4 Traffic Simulators-based Models

By refining synthetic models and going through an intense validation process based on real traces or behavior surveys, some companies or research teams gave birth to realistic traffic simulators. Developed for urban traffic engineering, fine grain simulators such as PARAMICS, CORSIM, VISSIM, TRANSIMS and SUMO, are able to model urban microscopic traffic, energy consumption, or even pollution or noise level monitoring.

2.2 Study of Simulator

Researchers and developers have built VANET simulation software to allow the study and evaluation of various media access, routing, and emergency warning protocols. VANET simulation is fundamentally different from MANETs (mobile ad hoc networks) simulation because in VANETs, vehicular environment imposes new issues and requirements, such as constrained road topology, multi-path fading and roadside obstacles, traffic flow models, trip models, varying vehicular speed and mobility, traffic lights, traffic congestion, drivers' behaviour, etc. Currently, there are VANET mobility generators, network simulators, and VANET simulators.[13][14][15][16][17] I am comparing here different simulators based on their software characteristics, graphical user interface (GUI), popularity, ease of use, input requirements, output visualization capability, accuracy of simulation, etc. VANET simulations typically require a networking component and a mobility component. In most cases, these components are provided by two separate simulators. Researchers build their topology and produce a trace of vehicle movements using a mobility simulator. This movement trace is then fed to the networking simulator representing network node movements. There has been recent work on developing integrated simulators that contain both networking and vehicular mobility components and allow feedback between the two components, but so far, most research has been performed using two separate simulators. Detail specification and comparison is given in Annexure-1.

2.2.1 Network Simulator

Network simulator is mainly designed for evaluating network performance, such as throughput, delay, jitters, etc. I introduce here some popular Network simulators, such as NS-2,NS-3, GloMoSim, Qualnet, JiST/SWANS, SNS, OpeNET, J-Sim. Network simulators are particularly useful to test new networking protocols or to propose modifications to existing ones in a controlled and reproducible manner.

Conclusion of Network Simulator

As shown in [27][28], ns-2 is less suitable for simulating large networks but it is popular and easy to use, unlike SNS or JiST/SWAN. In fact, JiST/SWAN is the most difficult to install. All the studied simulators provide open source code and are available freely on the Internet. Users can modify and enhance the code further, creating new versions. The major shortcoming is the lack of considerations for VANETs. For example, vehicular traffic flow models are not considered and 802.11pMACis not included into the simulators (except for ns-2.33). Also, physical layer issues, obstacles, and road topologies present in a vehicular environment are often neglected.

2.2.2 Traffic Simulation

One of the main challenges posed by VANETs simulations is the faithful characterization of vehicular mobility at both macroscopic and microscopic levels, leading to realistic non-uniform distributions of cars and velocity, and unique connectivity dynamics. I analyse here some popular Traffic simulators, such as VanetMobiSim/ CanuMobiSim, SUMO, FreeSim, and CityMob ,MOVE, STRAW,TSIS-CORSIM, VISSIM, PARAMICS,SHIFT/SmartAHS, Microscopic Traffic Applet.

Conclusion of Traffic Simulator

Conclusion of Traffic Simulator SUMO is mostly used with NS2 which support all kinds of functionality. PARAMICS, SHIFT/SmartAHS, Microscopic Traffic Applet are developed for commercial purpose. VanetMobiSim and MOVE also contain similar functionality as SUMO but not popular as SUMO. support. However, only VanetMobiSim provides excellent trace support. CityMob is good in software features and traffic model support. FreeSim exhibits good software characteristics but is limited in other features.

2.2.3 VANET Simulator

In these types of applications, the traffic simulator feeds the network simulator with position information, speed, acceleration, direction, etc. There are only a few integrated frameworks available. Currently, the mobility and network models in integrated frameworks are implemented in two separated simulation tools. Therefore, there is a clear need for an integrated mobility and network simulator in order to evaluate effective performance. I introduce here some popular Trace simulators, such as GrooveNet, TraNS, MobiREAL, NCTUns.

Conclusion of VANET Simulator

All simulators provide both alphanumeric and config file input and console message output but the user interface for TraNS appears more sophisticated than others. All simulators provide street-level topology view. So far, only TraNS can support visualization using Google Earth. vehicles. A survey of recently published papers shows that GrooveNet and NCTUns are more frequently used for VANET simulations than others.

2.2.4 Previous Work

The main aims behind the development of Congestion detection algorithms are detect areas of high traffic density and low speeds. Each vehicle captures and disseminates information such as location and speed and process the information received from other vehicles in the network. Multiple approaches have been proposed to implement congestion detection in VANET. Collision avoidance systems are designed to detect a traffic incident in real-time and rapidly relay this information to nearby vehicles to prevent a collision. These systems are very different from traffic congestion systems, in the former; information needs to be relayed very fast over short distances and needs to be extremely reliable as it has a direct effect on life-and-death situations. Most current navigation systems are static and do not provide traffic information. Route selection is based only on static map data which leads to the system that fails to give the driver the most efficient route to his/her destination. Traffic information services are evolving fast towards a combination of historical, real-time, and predictive information allowing optimized (re-)routing and reliable Estimated Time of Arrival calculations. Major navigation brands such as Nokia and Google leverage probe data from their customer base to offer free traffic updates. Smaller vendors are shifting to lifetime traffic offers bundled with the navigation device. Some of the systems are developed which only receive the traffic road map, congestion details, accident details, whether details. The main characteristic of current systems is that they are one-way: vehicles only receive information, usually through FM radio or satellite. Different traffic vendors which provide the traffic details are: INRIX, ITIS 20,NAVTEQ. With the recent progress on vehicular networks, it has become clear that the study of vehicular traffic management and its requirements is currently the hotspot in vehicular ad-hoc networks. In this context, many research groups exploring this area of research so that they can improve Quality of Driving Experience (QDE) by influencing vehicles routes[45]. A number of Traffic Information Systems have been developed i.e. systems which gather traffic data and disseminate traffic information to users , so they can make better informed decisions regarding their route. While these systems do keep drivers better informed about traffic conditions, there is no such information how the driver will interpret the information given[46]. In[47], Abderrahmane Lekas and Moumena Chaqfeh present a vehicular communication system for road congestion detection and avoidance by disseminating and exploiting road information. In Japan, as one of traffic information dissemination systems, VICS (Vehicle Information and Communication System) are widely used . VICS provides latest traffic information to cars via FM broadcast and various types of beacons. T. Shinkawa and et al present a system for discovering and disseminating traffic congestion information is proposed where vehicles build their own local traffic maps of speeds experienced on visited roads, and share them with other vehicles. In this context TraffCon[48], a novel Traffic Management System for WAVE is introduced. In this TMS Vehicles act as client nodes and communicate with the server in two asynchronous modes i.e. information gathering and traffic management. Traffcon employs the Best Route Selection Algorithm in its route decision making process. Decision making process starts when a vehicle begins a journey by sending its origin and desired destination to the server.

Chapter 3

Problem Statement

Traffic congestions are formed by many factors. We can divide those factors in two main categories: one is predictable like road construction, rush hour and small roads and second is unpredictable like accidents, weather and human behavior[49,50]. Drivers, unaware of congestion ahead eventually join it and increase the severity of it. The ability for a driver to know the traffic conditions on the road ahead will enable him/her to seek alternate routes saving time and fuel. When many drivers have this ability, traffic congestions, specifically those related to localized incidents such as accidents or temporary disruptions will be less harsh and only the vehicle in the immediate surrounding area of the incident, at the time of the incident, will be affected. This would lead to a much more efficient use of our road infrastructure. In order to provide drivers with useful information about traffic ahead a system must identify the congestion, its location, severity and boundaries.

3.1 Assumption

- All vehicles are enabled with OBU.
- RSU are available with each cross road.
- All RSU are interconnected.

3.2 Congestion Detection system constraints

Following parameters we have to consider in Congestion detection system.

3.2.1 Bandwidth Utilization

Because of limited bandwidth, there is need to make smart use of available bandwidth. For better utilization of bandwidth one way is filtering of data. Try to send as lesser as possible data in network by different data processing and filtering.

3.2.2 Security

Security is a mandatory for real-world systems. There is need to detect malicious node, before they harm the system. Here we have to protect user from false congestion detection messages. So as a part of security we have to register vehicle node and need to authenticate before permitting it to transmit any kind of data.

3.2.3 Privacy of vehicle

There is need to keep privacy of vehicle by hiding the vehicle id to other user. it depends on user to declare his vehicle id to others or not. Here the question of privacy is generated from others and also we have to keep some vehicles id hide like VIP persons from other user for security reasons.

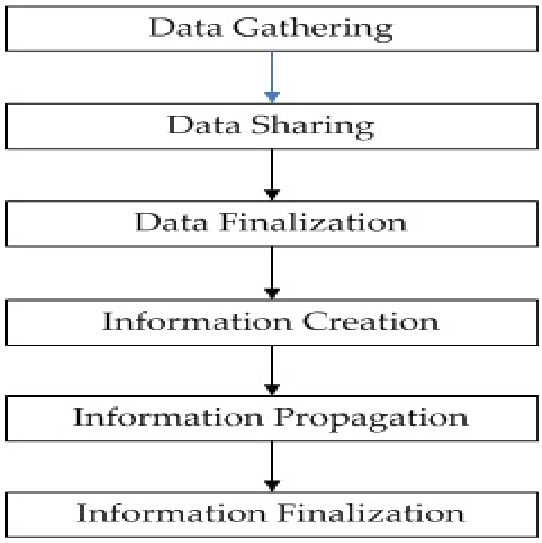
3.2.4 Participation of vehicle

All vehicles must be equipped with transceiver, which can transmit and receive VANET data. It is depends on a node whether it wants to participate in network or not. If amount of data is less then ok, but if amount of data to send is high then there question will arise why should I transfer?. So here need to do equal distribution of data lode on node.

Chapter 4

Proposed Solution

Congestion detection algorithms are designed to find areas of high traffic density and low speeds. Each vehicle disseminates the information it has obtained from its own hardware and from other sources and process the information received from other nodes in the network. Congestion detection is only one of many applications of VANETs and it is not designed to be used as means for automated driving but rather as a tool to deliver information to the driver that will help him/her make decisions to avoid heavy traffic. One important characteristic of congestions is that they move relatively slow and the vehicle density is high. We will use this to our advantage when designing our system for congestion detection and propagation. Congestion information travels by hopping from vehicle to vehicle and is also carried by vehicles going in the opposite direction on the highway via delayed retransmission. Because congestions are intrinsically slow the small lag caused by this delayed retransmissions does not significantly degrade the accuracy of the system. A vehicle that is in a real congestion receives multiple confirmations from vehicles nearby going also slower than normal allowing the vehicle to expand and validate its congestion area. Only after a suspected congestion area is validated by a reasonable number of vehicles it is then broadcasted. This reduces unnecessary network usage and false congestion from affecting the efficiency of the system. When vehicles are, or believe to be, in a congestion they collect location information from nearby vehicles in order to form the picture of the congestion. When developing a congestion detection system, we can either rely on some central network and data gathering infrastructure or we can use hardware in the vehicles themselves to collect, analyse and disseminate this information. A distributed approach to traffic congestion detection is the more efficient



because it provides a greater degree of reliability and does not require a major investment in infrastructure.

Basic steps to follow in Congestion Detection System

4.1 Basic steps to Detect Congestion

4.1.1 Data Gathering

The most basic component of a congestion detection algorithm is the gathering of data from the environment. Each vehicle must be equipped with one or more devices that are capable of gathering data such as current location and speed. This data constitutes the building blocks of the system and changes very rapidly.

4.1.2 Data Sharing

Gathered data will useful only when it shared to others. Vehicles must be equipped with a device that allows them to transmit and receive information wirelessly to and from other vehicles in their vicinity by creating a vehicular ad-hoc network.

4.1.3 Data Finalization

Data will be gathered from nearby vehicles as well as data relayed from far-away vehicles is received, it needs to be consolidated in order to eliminate inconsistencies and to keep a lid on the amount of information that is to be processed.

4.1.4 Information Creation

Information is a processed form of data. As the data is consolidated it is converted into information in the form of possible traffic congestion zones. This is the final product of the congestion detection algorithm and it is to be relayed to the vehicle's operator through some visual interface that shows the relevant congestion areas on a digital map.

4.1.5 Information Propagation

This information is propagated so that vehicles away from the congestion zone, unable to make this determination with the information received from vehicles around them, can benefit from it. Information is much more compact that raw data and it is also more meaningful. Once a determination is made that a traffic congestion exists at a certain location, this information can be propagated to nodes far away because this information is more likely to be useful than raw data.

4.1.6 Information Finalization

Information received from other nodes is combined and re-broadcasted so that changes in the current traffic situation can be reflected in real-time. The congestion detection algorithm must provide reliable and current information in a timely fashion.

4.2 Proposed Approach

4.2.1 Signal Agent (SA) as Traffic Congestion Controller

On each cross road Signal Agent(SA) will work as a Road Controller. Signal Agent (SA) will get the details of each vehicle enter in its range. We are assuming here each Signal Agent are connected with each other.

Car Agent (CA) can communicate with Signal Agent(SA), whenever it enter in the range of Signal Agent(SA).CA can send the details of Car ID(CID), Current Position(CP), Current Speed(CS), Max Speed(MS), No. of Neighbor CA(NCA), Road Block ID(RBID).

SA will maintain the Road Block wise counter . Whenever CA enter in the range of SA then counter will increment by one and vehicle goes out of the range from SA then counter will decrement by one. By doing this we can get the details of Vehicle

Density(VD) on each and every road block segment.

For each road one Threshold value will predefined based on type of road, No. of Lanes, Length of Road. At Regular interval of time SA will send the Road Block wise details like No. of Vehicle, Average speed of vehicle (AVG_{speed}).

Step 1: For each Road Block Assign Road Vehicle Counter(RVC_i), which indicates the no. of vehicle on road block.
RVC_i=0; Where i as Road Block ID.

Step 2: For each Road Block identify Road Vehicle permitting it to transmit Threshold value($RVTh_i$),which indicates the threshold value of no. of vehicle on road block.
$RVTh_i$:= X;
Assign appropriate Value based on Type of Road, No. of Lanes, Length of Road.

Step 3: If CA enter in the range of SA Then,
$RVC_i := RVC_i + 1$;
CA (CID,CP,CS,MS,NCA,RBID) $\rightarrow SA$

Step 4: If Car Agent goes out of the range Then ,
RVC$_i$:= RVC_i - 1;

Step 5: If (RVC_i >$RVTh_i$) Then
Declare "Congestion"
SA_j (SAID, RVC$_i$,AVG$_{speed}$, Time) $\rightarrow$SA$_{other}$

Step 6: Distribute Self details to other SA.
SA$_j$($SAID, RVC_i, AVG_{speed}, Time$) $\rightarrow$SA$_{other}$
SA_{other} (SAID, RVC$_i$, $AVG_{speed}, Time$) $\rightarrow$SA$_j$

Step 7: Suggest Optimal Path to user based on vehicle density and average speed of Road Block.

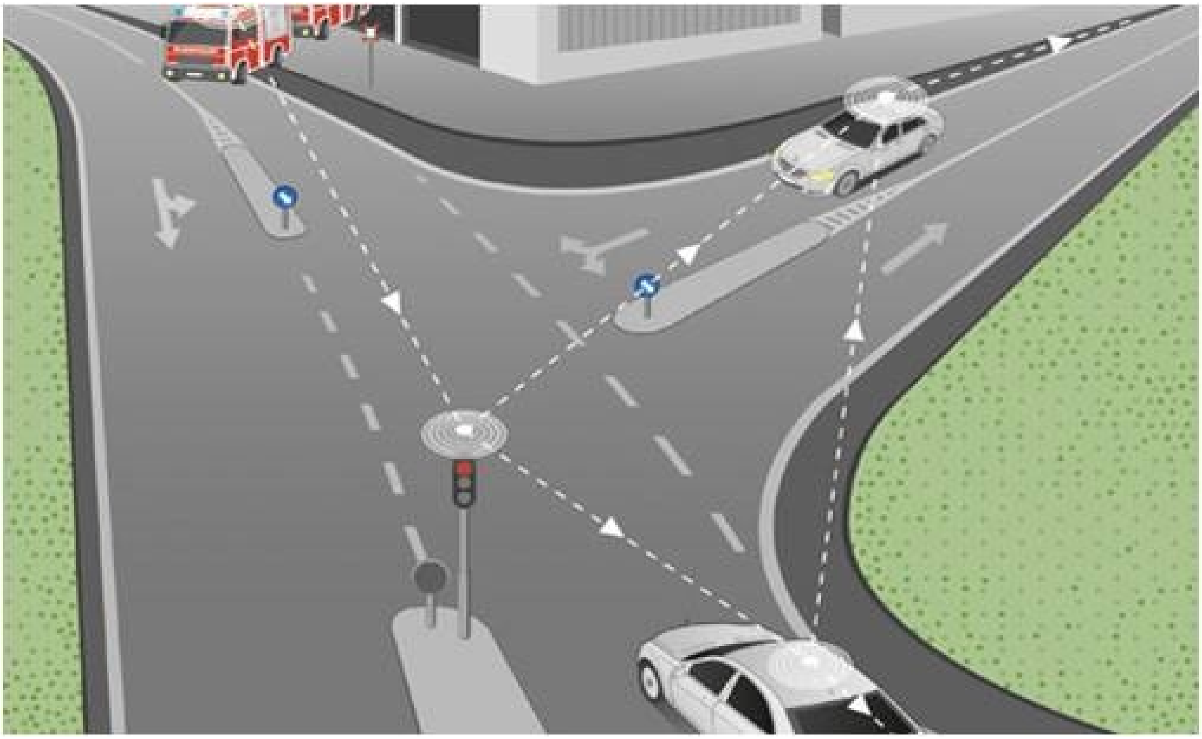


Figure 4.1: Signal Agent (SA) as Traffic Congestion Controller

SA(Optimal Path) $\rightarrow CA$

4.2.2 Car Agent (CA) as Traffic Congestion Controller

Optimal Path (OP) will be given as per the user choice. For finding Optimal Path Congestion Factor based on distance (CF_{dist}) and Congestion Factor based on time (CF_{time}) is used.
D and T are the weighted factor for Distance and Time.

OP=(D* CF_{dist} + T* CF_{time});
D =1-T;

Route/ trip wise congestion Factor based on distance (CF_{dist}) can be calculated. $Dist_{ideal}$ is an ideal distance for each route/trip which is predefined based on ideal condition and $Dist_{actual}$ is an actual distance covered by previous vehicle to reach to the destination through this route.

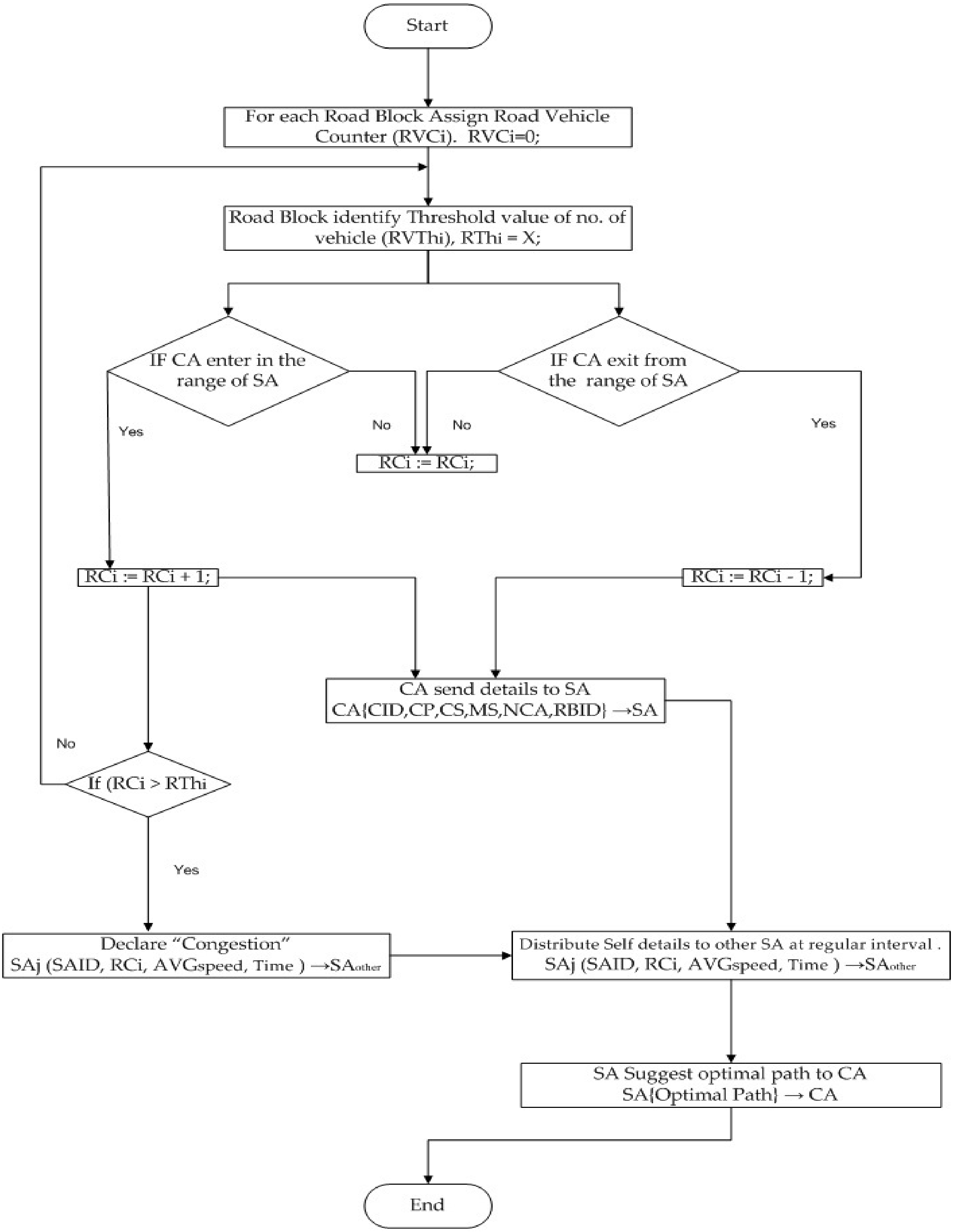


Figure 4.2: Flow chart of Signal Agent (SA) as Traffic Congestion Controller

$CF_{dist} \rightarrow$ (Dist_{actual} - $Dist_{ideal}$) / $Dist_{ideal}$;

Route/ trip wise congestion Factor based on time (CF_{time}) can be calculated. $Time_{ideal}$ is an ideal time for each route/trip which is predefined based on ideal condition and $Time_{actual}$ is an actual time taken by previous vehicle to cover that route.

$CF_{time} \rightarrow$ (Time_{actual} - $Time_{ideal}$) / $Time_{ideal}$;

Weighting factor D and T are used to finalize Optimal Path (OP) based on user choice which values can be varies between 0 to 1. If user wants to give more priority to distance then give more weightage to D otherwise give more weightage to T.

Step1: Get the details of destination from user.

Step 2: Get the Current position/Source Position from vehicle Car Agent.

Step 3: Generate number of routes based on source and destination.

Step 4: Car Agent (CA) will get the details of current traffic situation for each route from nearest Signal Agent (SA).
SA(Available Different Path)$\rightarrow CA$

Step 5: Get the details of ideal distance(Dist_{ideal}) and ideal time ($Time_{ideal}$) for given trip.
SA($Dist_{ideal}$, $Time_{ideal}$)) $\rightarrow CA$

Step 6: Get the details of actual distanceDist_{actual} and actual time $Time_{actual}$ from the Signal Agent (SA) who maintain the details of previous vehicles which cover that route.
SA($\text{Dist}_{actual}, Time_{actual}$) $\rightarrow CA$

Step 7:$Calculate$$\text{CF}_{dist}$ and CF_{time}.
$CF_{dist} \rightarrow (Dist_{actual} - Dist_{ideal})/Dist_{ideal}$;

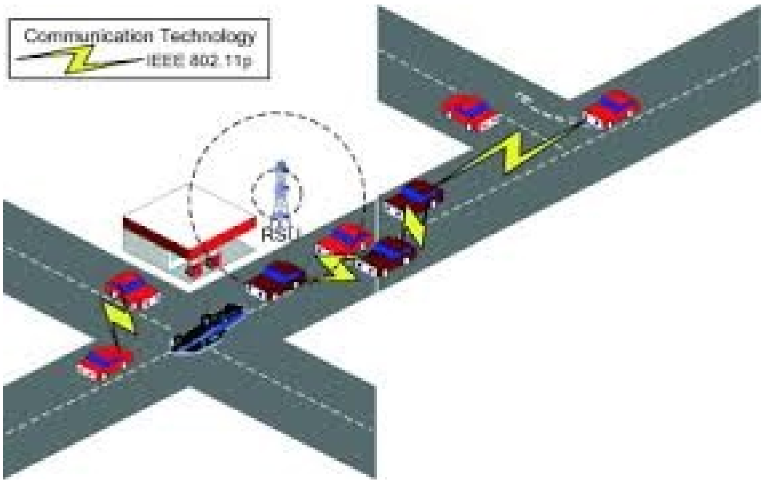


Figure 4.3: Car Agent (CA) as Traffic Congestion Controller

$CF_{time} \rightarrow (Time_{actual} - Time_{ideal})/Time_{ideal}$;

Step 8: Accept the weightage factor T and D from user for giving priority to distance or time.
$D = 1 - T$;

Step 9: Calculate the optimal path.
$OP = (D * CF_{dist} + T * CF_{time})$;

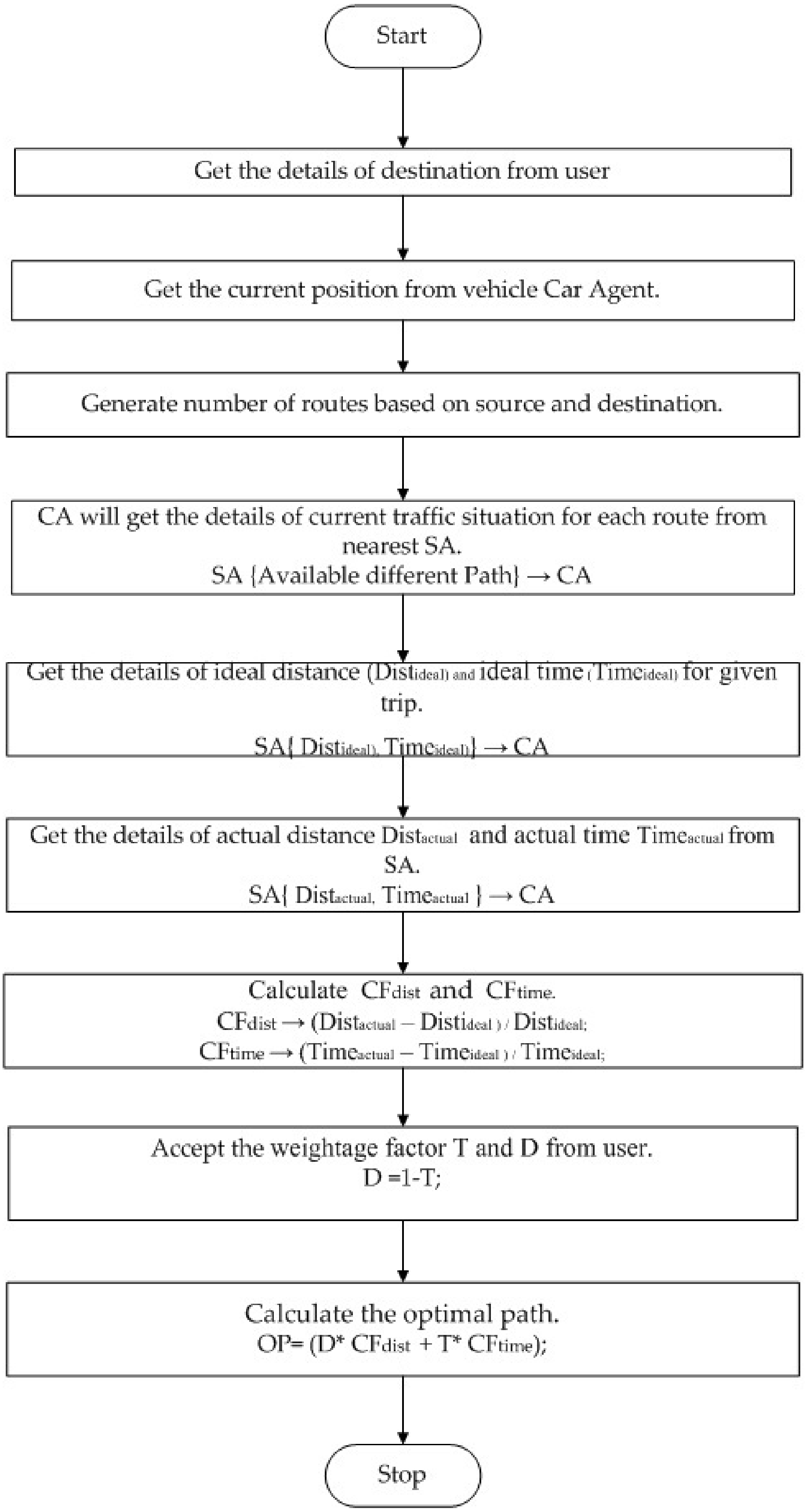


Figure 4.4: Flow chart of Car Agent (CA) as Traffic Congestion Controller

Chapter 5

Simulation Result

In this project report we are representing our work with two different types of road scenario. One is structured approach and other is real road map. For different Scenario specification of parameters are as given in Table 5.1:

Structured Approach As shown in Figure-5.1, we design structured approach for city area of total 25 km^2. Four roads are connected with cross signals. Each Traffic signal are work as a Road side Unit(RSU).Entire road is divided into no. of Road Blocks. Length of each Road Block is 1 km and each has two no. of lanes.

We deployed 2500 no. of vehicle in entire structured scenario. Each vehicle supports 802.11P. Each vehicle are consider as On Board Unit(OBU). We used NCTUns 6.0 as a simulator tool.

5.1 Result Analysis

5.1.1 Based on no. of vehicles getting the Status of Congestion

As discussed above we design simulation for structured approach of total 25 km^2(5000 m * 5000 m) with 2500 no. of vehicles. Here we run this simulation scenario in normal /regular condition with out using any intelligence. Then we run this simulation in the presence of Car Agent and Signal agent. Car Agent can provide intelligence to Car and communicate with traffic Signal Agent (SA). Whenever vehicle enters in the range of Signal agent(SA) at cross road then communication can be possible between CA and SA. Result are represented using two graph. Figure 5.3 representing Graph of Time wise No. of vehicles in the scenario and Figure 5.4 present the Graph of Time wise details of No

Table 5.1: Simulation Parameters

Parameters	Scenario -I	Scenario -II
Area	5000 5000 meter	8000 6000 meter
Description	Structured Road Block	Real City Road map
No. Of Vehicle	2500	4200
Simulation Time	2200 Sec.	3000 Sec.
Type of Vehicle	Car	Car
Traffic Light Support	Yes	Yes
Type of Packet Send	UDP	UDP
Max. Speed of Vehicle	10/20/30 m/s	10/20/30 m/s
Length of Vehicle	6 meter	6 meter
Safe Distance	Front and Rear -2 m	Front and Rear -2 m
Allow Overtaking	Yes	Yes
No. of LAN of Road	2	2
Width of LAN	6m	6m
Transmission of OBU	100 m	100 m
Transmission of RSU	250 m	250 m

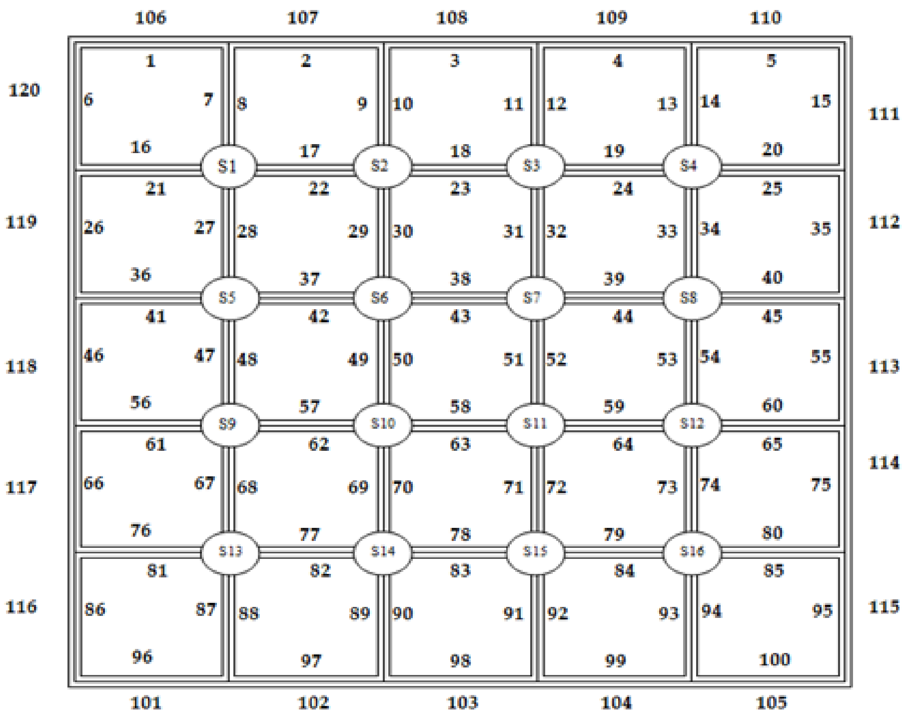


Figure 5.1: Structured Approach

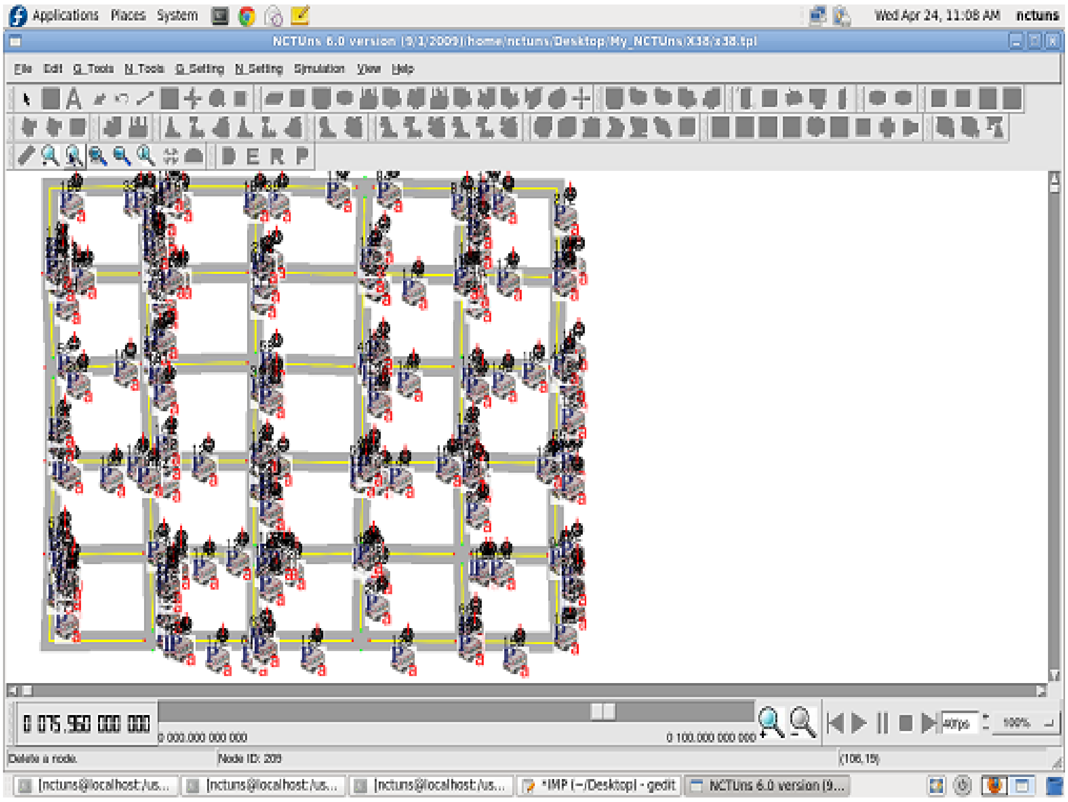


Figure 5.2: Scenario designed in NCTUns 6.0

of vehicle on particular road block (RB41-45).

From the Figure 5.3 we can analyse that in graph No. of car are added gradually in the road network so traffic is increasing and when cars reached to their destination at that time road traffic is decreasing in both the With Car Agent and With Out Car Agent mode. In Car agent (CA) mode vehicle getting feedback from Signal Agent (SA). Based on given feedback CA will chose optimal path to reach at destination, so congestion is reduced compared to without CA mode. It will not follow the path which already congested by other vehicles.

As we can see from the Figure 5.3 in both the mode car increasing rate remains same but due to congestion aware routing we can see much reduction in terms of No. of vehicle in With Car Agent (CA) mode. After refereing the entire road block we want to do analysis of particular Road area. We selected Set of Road Block which has high traffic compare to others. Set of Road block(RB:41-45) is taken for better understanding the result. Length of this set of road block is 5 km.

In Figure 5.4 traffic of particular set of road block is represented, whose total length is 5km. Here we can consider if total no. of vehicle exceed then some specific threshold

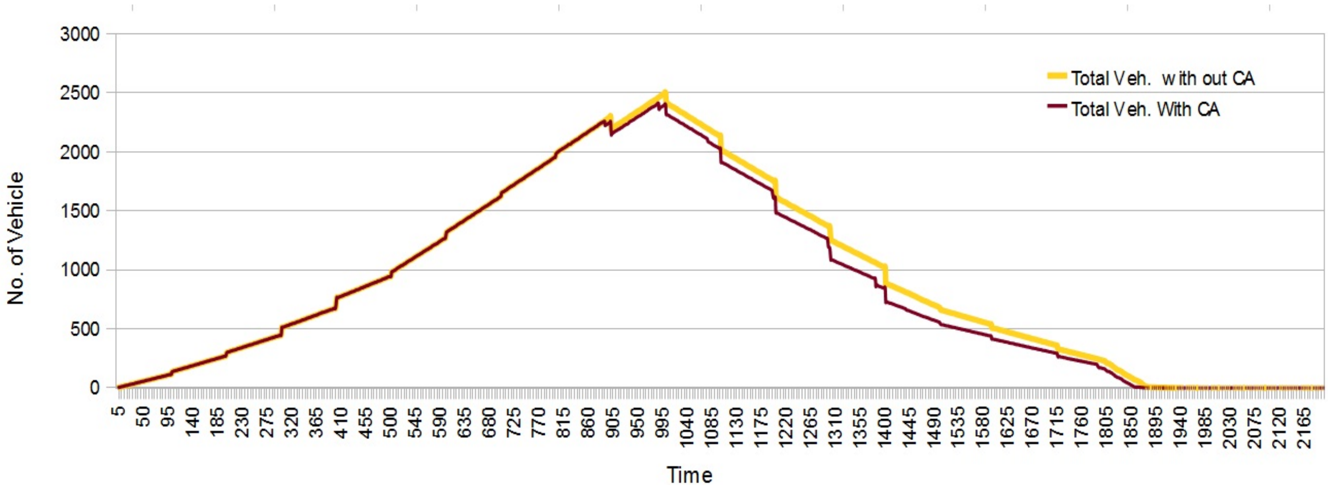


Figure 5.3: Time vs Number of vehicles

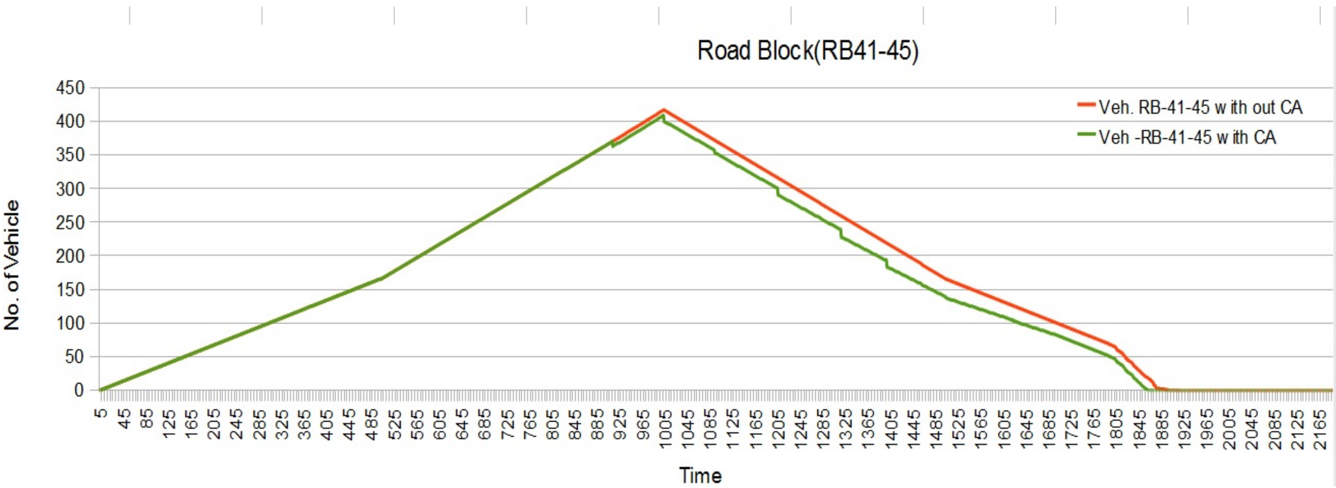


Figure 5.4: Time vs Number of vehicles for RB41-45

Table 5.2: Vehicle Trip Time

Veh.No	Type	Time With Out CA	Time with CA	Difference(%)
8	CAR	846	760	10.17
125	CAR	1057	963	8.86
436	CAR	1016	904	11.02
736	CAR	980	870	11.22
836	CAR	1296	1160	10.49
1023	CAR	1143	1028	10.06
1324	CAR	540	500	7.41
1688	CAR	992	880	11.29
1734	CAR	940	830	11.7
1863	CAR	1110	1018	8.29
			Sum (Σ) Average	100.55 10.05%

value then we can declare congestion for that area. Here we consider that threshold value is 300. So total no. of vehicles are higher then 300 declared it as a congestion. From the graph congestion time is 768 sec to 1202. So for total 434 second Road Blocks have a congestion, which is a 19.72% of total simulation time. All the vehicles in both the mode reach to their destination at simulation time "1900", so after that we getting no. of vehicles zero for remaining time.

5.1.2 Trip wise reduction in Time.(comparison Between with and without CA)

After the analysis of no. of vehicles on different road block we analysis the different trip of different vehicle and based on that find the total average saving time in scenario. Table 5.2 gives the details of vehicle id, type, time taken by vehicle to complete the trip with out help of Car Agent(CA) and with the help of Car Agent(CA).

Table:5.3 gives the vehicles details in more depth. It additionally give the details of source and destination , trip path of vehicle, maximum speed of each vehicle.

By analyzing the trip time of each vehicle in the presence of Car Agent(CA) and with out Car Agent (CA), we derived that vehicle required less time in presence of Car Agent(CA) to complete the trip. We find the percentage difference between each and from it derived average time difference. So through our approach we get the advantage in average speed up of 10.05%.

Table 5.3: Vehicle Detailed Trip

Veh. No	Type	Source	Destination	Path	Max .Speed (m/s)	Start time	End time
8	CAR	(2010,5005)	(4005,5)	(98 99 71 51 31 11 109)	10	0.00	760.00
125	CAR	(5006,5008)	(2000,10)	(105 104 91 71 51 31 11 3)	10	0.00	963.00
436	CAR	(2010,5005)	(5005,2004)	(96 97 89 69 49 38 39 40)	15	0.00	904.00
736	CAR	(10,1006)	(4000,5007)	(16 17 18 19 34 54 74 94)	15	0.00	870.00
836	CAR	(5004,5012)	(9,8)	(105 93 84 71 63 49 42 27 7 1)	10	0.00	1160.00
1023	CAR	(4996,998)	(10,4998)	(25 24 30 50 70 90 102 101)	10	0.00	1028.00
1324	CAR	(4995,10)	(10,10)	(5 4 3 2 1)	10	0.00	500.00
1688	CAR	(5014, 20)	(15,5012)	(5 14 34 54 74 84 83 82 81 86)	20	0.00	880.00
1734	CAR	(10,4000)	(4095,2000)	(76 77 78 79 73 53 40)	10	0.00	830.00
1863	CAR	(15,5000)	(4990,12)	(96 87 77 69 58 51 39 33 20 15)	20	0.00	1018.00

Table 5.4: Reduction in Traffic Congestion

Time	With Out CA	With CA	Difference	Reduction(%)
1000	2502	2402	100	4
1050	2274	2183	91	4
1100	2129	2023	94	5
1150	1881	1787	94	5
1200	1744	1604	140	8.03
1250	1489	1370	119	7.99
1300	1362	1185	177	13
1350	1137	989	148	13.02
1400	1020	842	178	17.45
1450	787	642	145	18.42
1500	681	556	125	18.36
1550	596	485	111	18.62
1600	502	409	93	18.53
1650	441	358	83	18.82
1700	376	305	71	18.88
1750	287	231	56	19.51
1800	228	160	68	29.82
1850	82	20	62	75.61
Sum (Σ)	19518 Total Reduction	17551 10.08 %		

5.1.3 Time wise reduction in Traffic Congestion

For Entire Simulation

After analyzing No. of vehicles per road block, Total Trip time per vehicle, we analyzed Reduction in number of vehicles based on time. We taken the reading of timing after the reduction in vehicles is started. After that we taken the readings for each 50 second and calculated no. of vehicles. Here we taken the readings for two mode one is for with out Car Agent(CA) and another is with Car Agent(CA). Difference between two mode in percentage is presented through Figure 5.5. We taken the total sum of vehicles with out Car Agent (CA) mode and with Car Agent (CA) mode. So we get the total reduction of 10.08%.

For Road Blocks RB41-45

In table 5.5 traffic details of road block RB41-45 is given. Here reading is taken after the reduction is started in no. of vehicles. For both the mode with and without Car Agent(CA), No. of vehicles are calculated for each 50 second. Difference between two mode in percentage is represented through Figure 5.6 for road block RB41-45. We taken the total sum of vehicles with out CA mode and with CA mode and getting total reduction of 9.82 %.

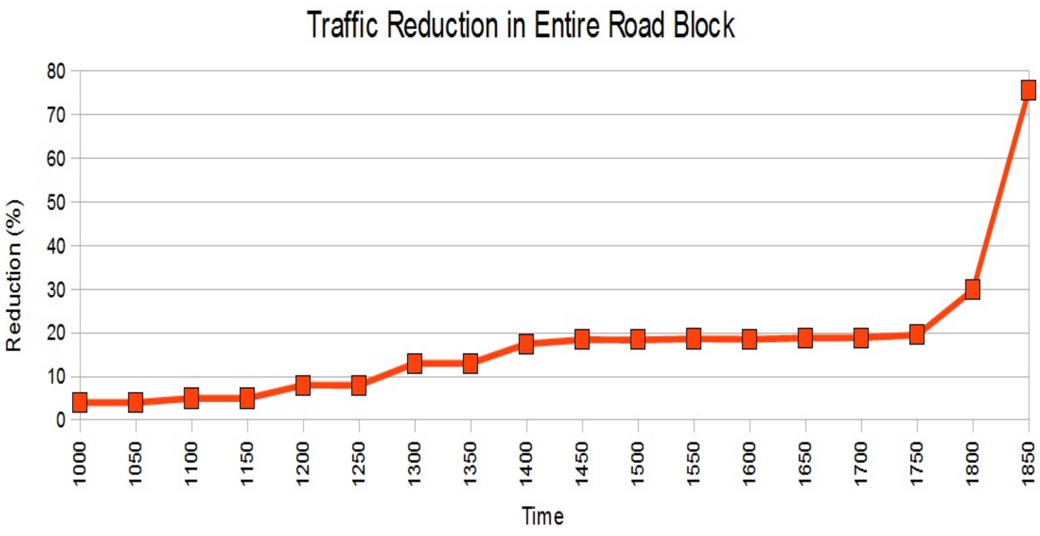


Figure 5.5: Time vs Reduction in Traffic

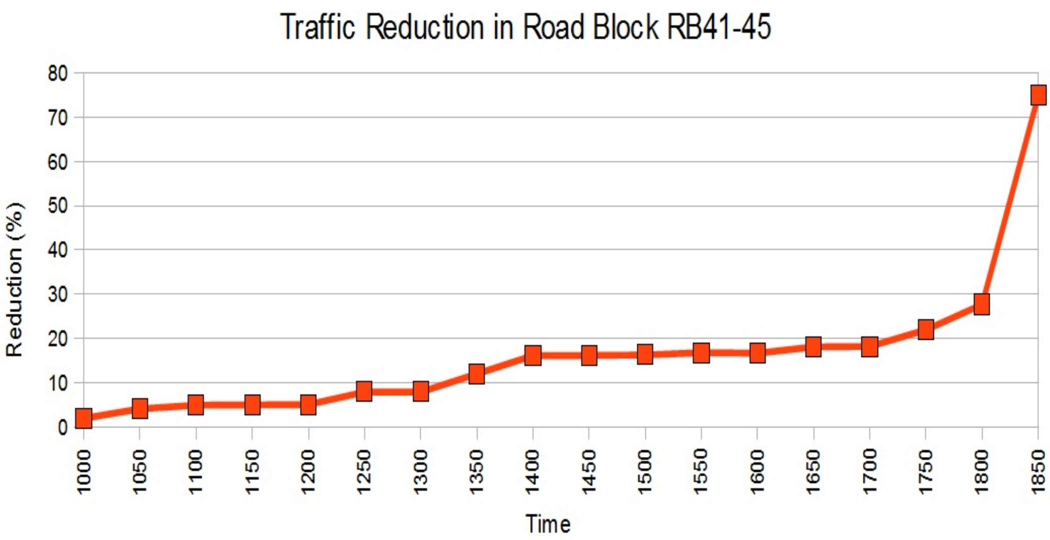


Figure 5.6: Time vs Reduction in Traffic for RB41-45

Table 5.5: Reduction in vehicle traffic congestion for RB41-45

Time	With Out CA	With CA	Difference	Reduction(%)
1000	417	409	8	1.92
1050	392	376	16	4.08
1100	367	349	18	4.9
1150	342	325	17	4.97
1200	317	301	16	5.05
1250	292	269	23	7.88
1300	267	246	21	7.87
1350	242	213	29	11.98
1400	217	182	35	16.13
1450	192	161	31	16.15
1500	166	139	27	16.27
1550	149	124	25	16.78
1600	132	110	22	16.67
1650	116	95	21	18.1
1700	99	81	18	18.18
1750	82	64	18	21.95
1800	65	47	18	27.69
1850	24	6	18	75
Sum (Σ)	3838 Total Reduction	3497 9.82 %		

Table 5.6: Trip Time for Alternate Path

Path	Speed (km/hr)	Distance (km)	Trip Time (Minute)
Path -1	40	8	12
Path -2	36	8	13.33
Path -3	52	8	9.23
Path -4	72	10	8.33

Based on average speed of vehicle suggest fastest path to vehicle

For finding best path source and destination will be accepted from user and based on it different route is generated. For each route Current average speed of vehicle is calculated from Signal Agent (SA).

As shown in Figure 5.8, For given source and destination four different route is generated. For each route current average traffic speed and distance is calculated and based on that as per Table:5.5 best trip time can be generated and given to the user. Here we are maintaining the result of average speed of each road block. For each 250 second result is taken for average speed of different road block. we are maintaining average speed with and without Car Agent (CA). Here we will discuss three different case. Normal scenario for Road Block RB-1 and RB-45. After that we analyse the Road Block in Accident case.

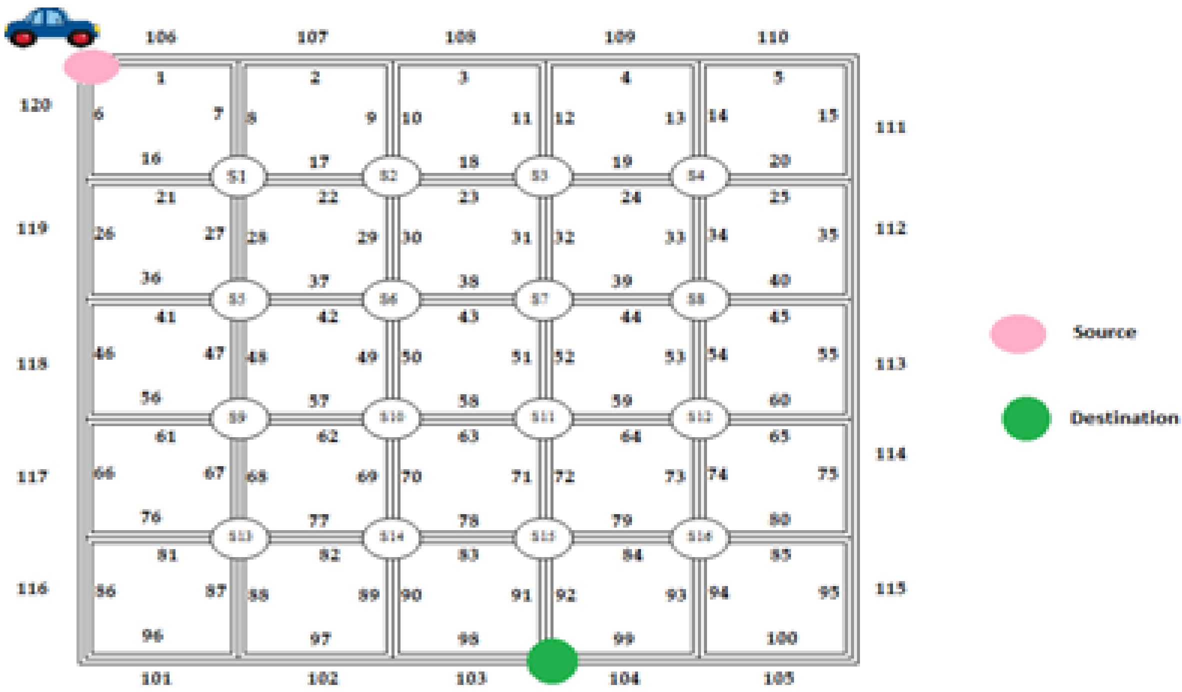


Figure 5.7: Car in Structured approach with Source and Destination

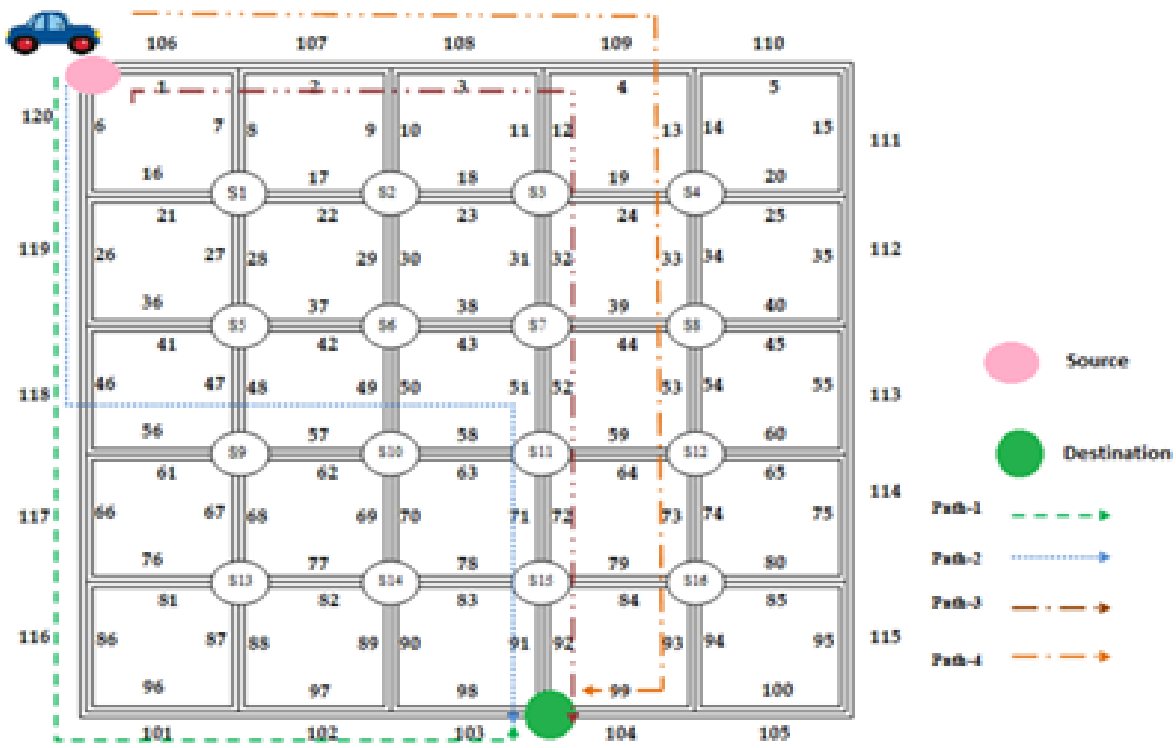


Figure 5.8: Available Alternate path between Source to Destination

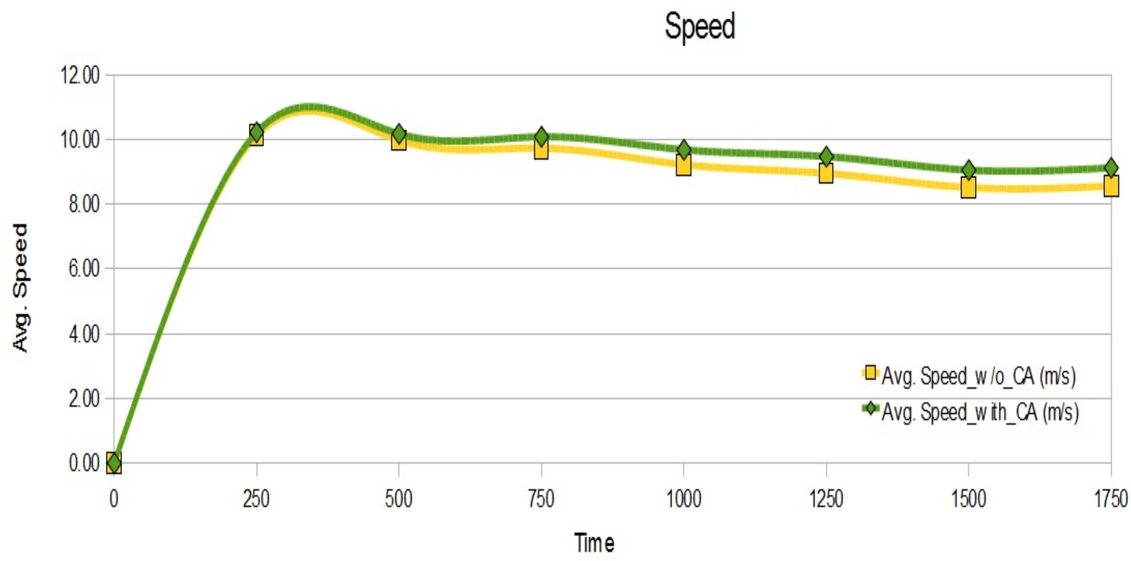


Figure 5.9: Time vs Average Speed RB-1

Normal Scenario for Road Block -1

Figure 5.9 represents time wise Average Speed of Road Block. Here we are getting difference between the readings of Average speed of vehicle with and without Car Agent(CA). Maximum average speed of vehicles is 10.23m/s with agent and 10.13 m/s with out agent at 250 second. Minimum average speed of vehicles is 9.06m/s with agent and 8.52m/s with out agent at 1500 second. We are getting lower average speed for with out Car agent Mode. But the difference is very less for low traffic areas.

Normal Scenario for RB-45

Here we are discussing the result of Road Block 45 which contain higher traffic compare to road block 1. Figure 5.10 represents Time wise Average Speed of Road Block -45. Maximum average speed of vehicles is 8.98 m/s with agent and 8.78 m/s with out agent at 500 second. Minimum average speed of vehicles is 8.43m/s with agent and 7.88 m/s with out agent at 1250 second. We are getting lower average speed for with out Car agent Mode. But the difference is higher compare to Road Block -1 due to high density of vehicle.

Vehicle Accidents

Here we created accident in Road Block-43. We want to analyse the effect of accident on average speed of Road Bock. Also we analyze here No. of Vehicle passed on road block in the presence of Car Agent(CA).

Figure 5.11 present the time wise average speed of vehicle with and with out Car agent(CA).

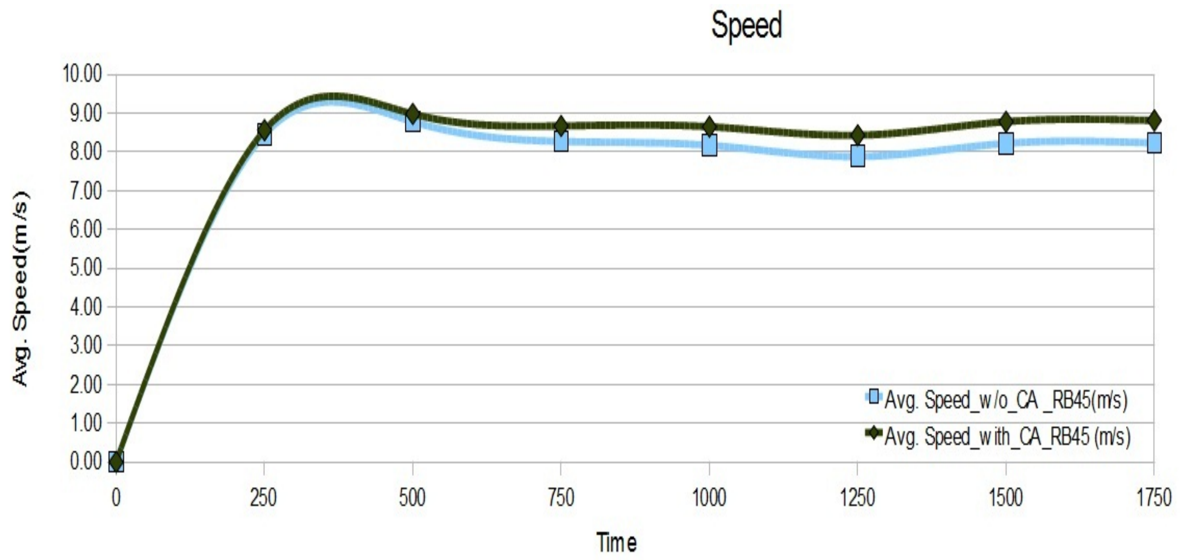


Figure 5.10: Time vs Average Speed Road Block -45

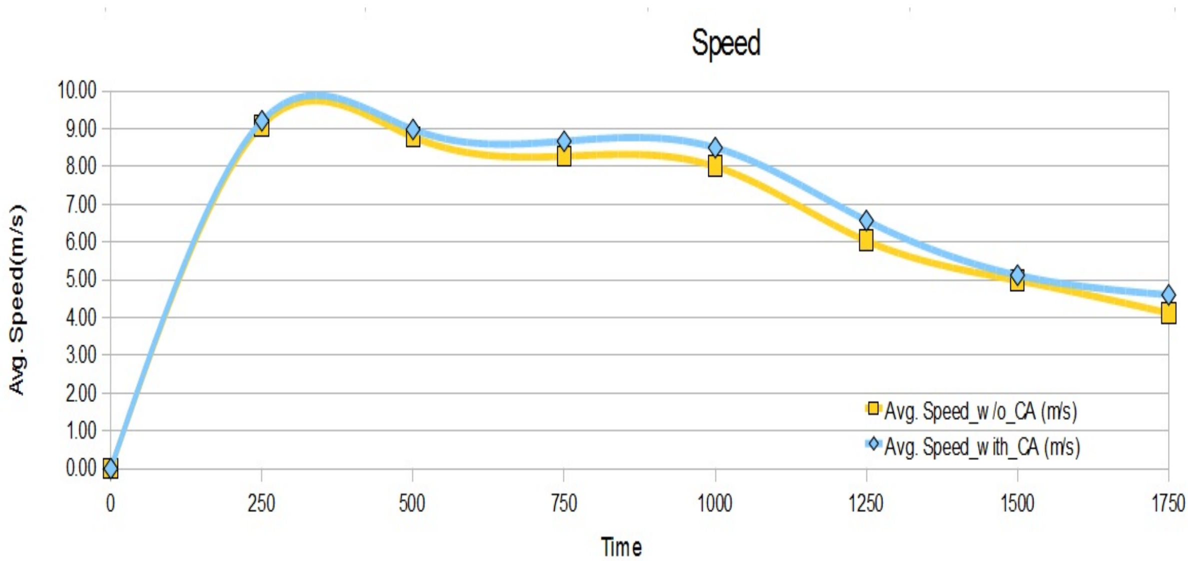


Figure 5.11: Time vs Average Speed for Accident Scenario

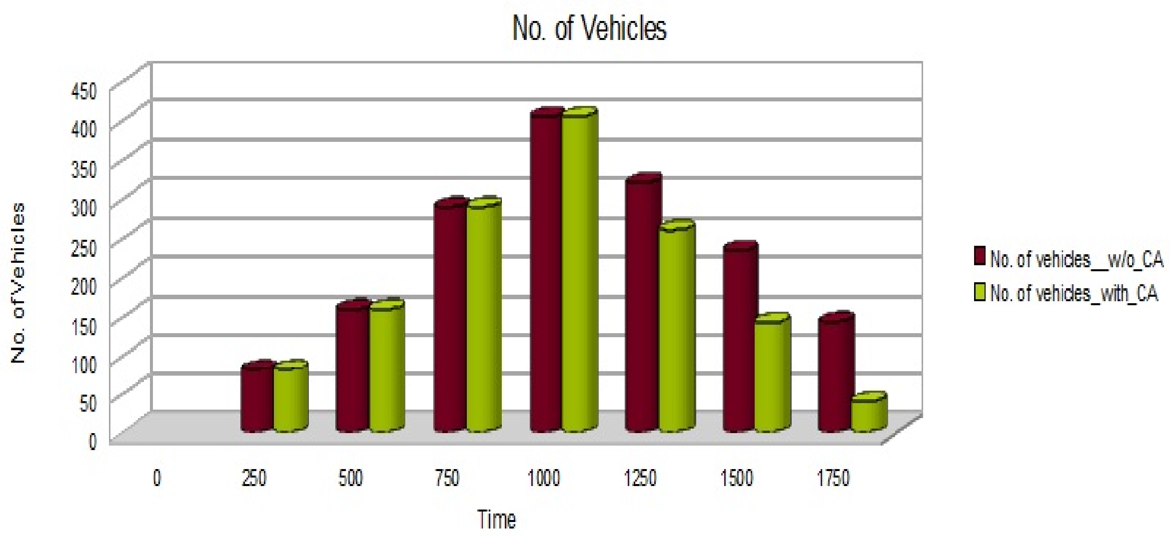


Figure 5.12: Time vs No. of vehicles for Accident Scenario

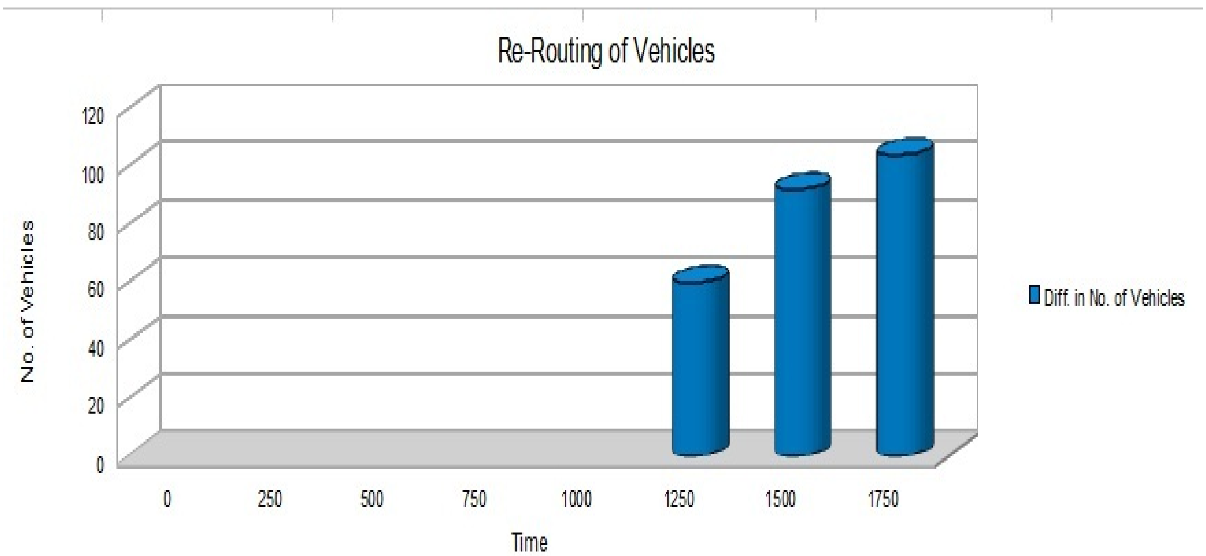


Figure 5.13: Re-routing of vehicles for Accident Scenario

Here at 1000 second accident is occur and due to it we get reduction in average speed. In both the mode average speed is reduced.

Figure 5.12 represents the no. of vehicle passed through particular road block. Here we can see in with Car agent mode vehicle are avoiding to choose path of road Block-43. Reduction in no. of Vehicle we can see after the time of 1000 second.

Figure 5.13 represent the no. of vehicles are re-routed due to the occurrence of accident. Here In Car Agent mode each vehicle communicate with Signal Agent (SA) and according to it choose optimal path.

5.1.4 City Scenario

After analysing the structured approach we want to analyse working of our approach in city scenario, which is less structured. As shown in Figure 5.14 we taken the city map of Taiwan. In our city scenario we taken the area of (8000m*5000m).We run the simulation for 3000 second.

Figure 5.15 presenting the graph of Time vs No. of Vehicle. Here the No. of vehicle presented using two approach, with and without Car Agent(CA). Due to congestion avoidance approach of Car Agent(CA) we get the less no. of vehicle in with Car Agent(CA) mode compare to without Car Agent(CA) mode after the 1500 second.

After analysing simple case we want to analyse accident case in city scenario. Here accident is occur at time of 1000 second. Figure 5.16 presenting graph of Time wise Average Speed of road block. Due to accident case in both the approach, with and without Car Agent(CA) we are getting reduction in average speed of vehicle. From the graph we can say that in both the approach there is negligible difference.

So, for checking the effectiveness of our approach we analyse no. of vehicle passed through that accident case in with or without presence of Car Agent(CA).

Figure 5.17 presenting the Time wise No. of Vehicle with both approach, with and without Car Agent(CA). In With out Car Agent(CA) mode vehicle entering in the road block are unaware about accident, so they enter in congestion area and increasing the congestion. When in with Car Agent (CA) mode Signal Agent (SA) will redirect the entering vehicle with CA on another path, due to this congestion will decreasing.
Figure 5.18 presenting the Time wise No. of vehicle, taken the re-routing decision. Gradually amount of rerouted vehicles are increasing.

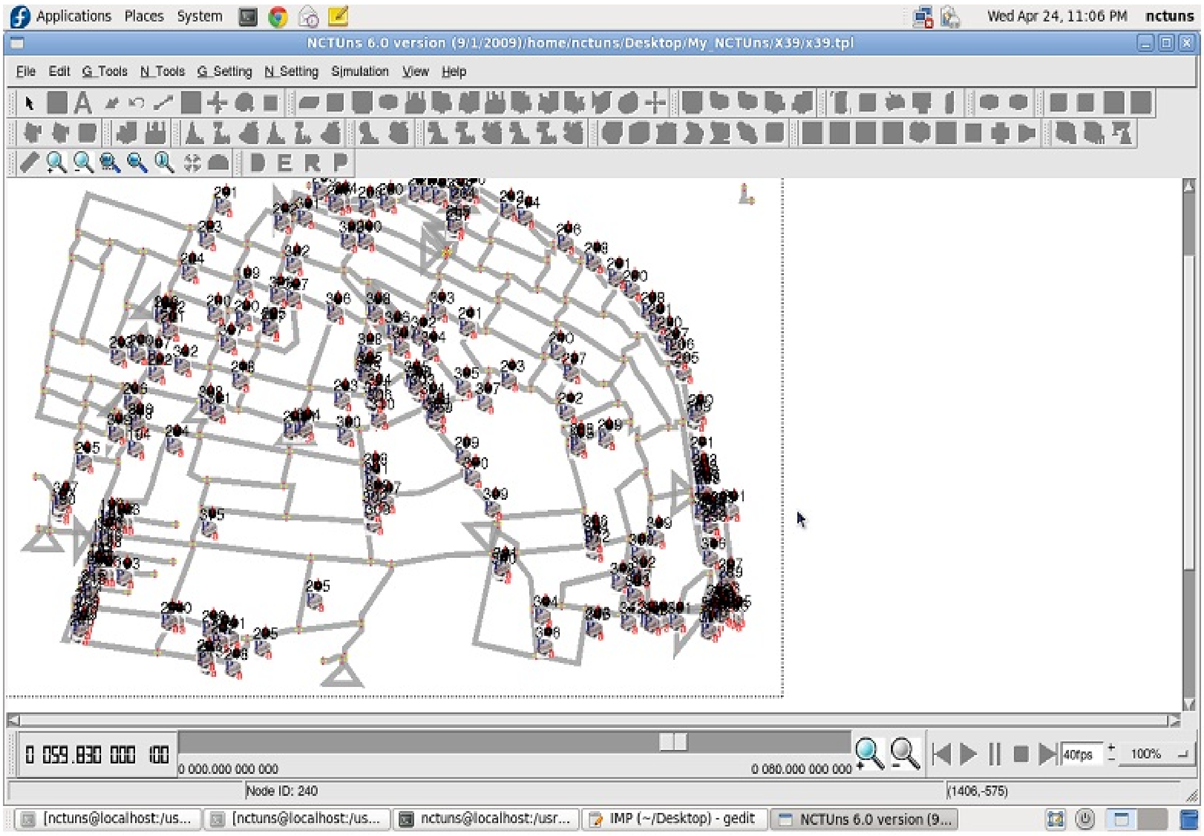


Figure 5.14: City Road Map of Taiwan

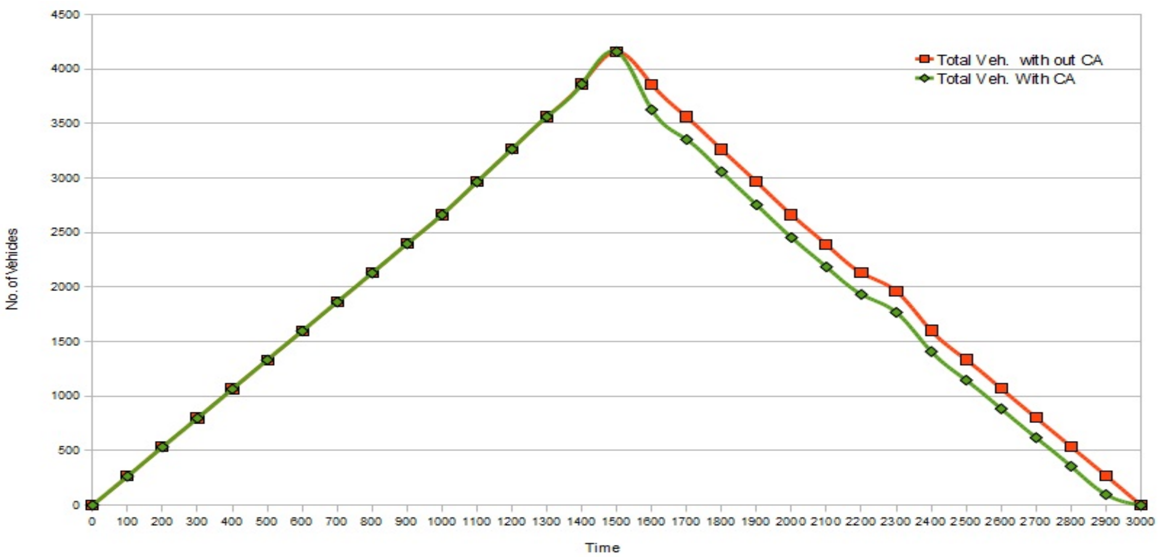


Figure 5.15: Time Vs No of Vehicles for accident case in city scenario

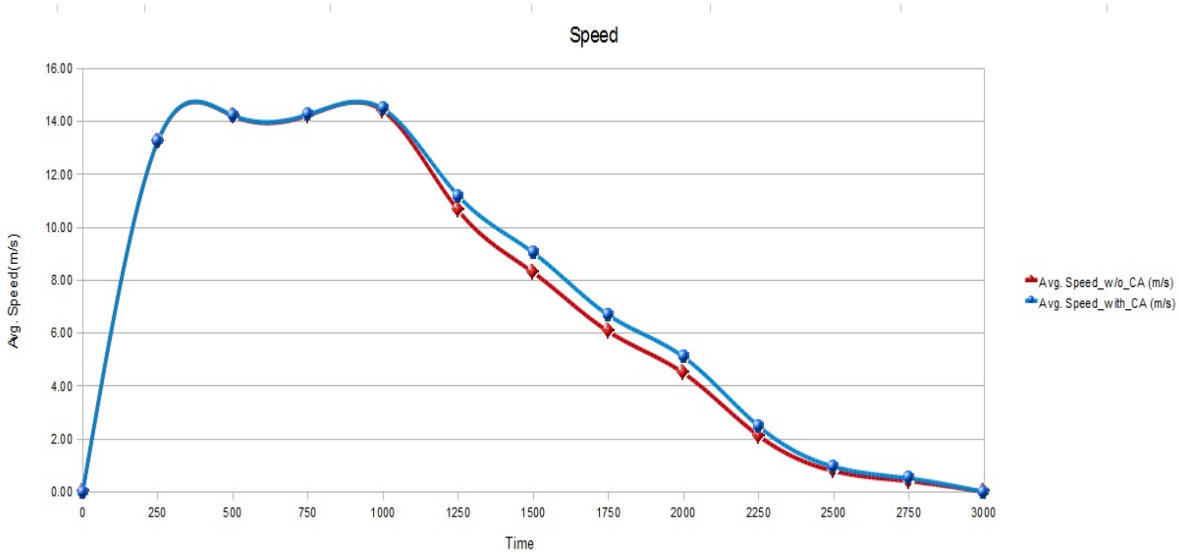


Figure 5.16: Time wise Average Speed of Road Block for accident case in city scenario

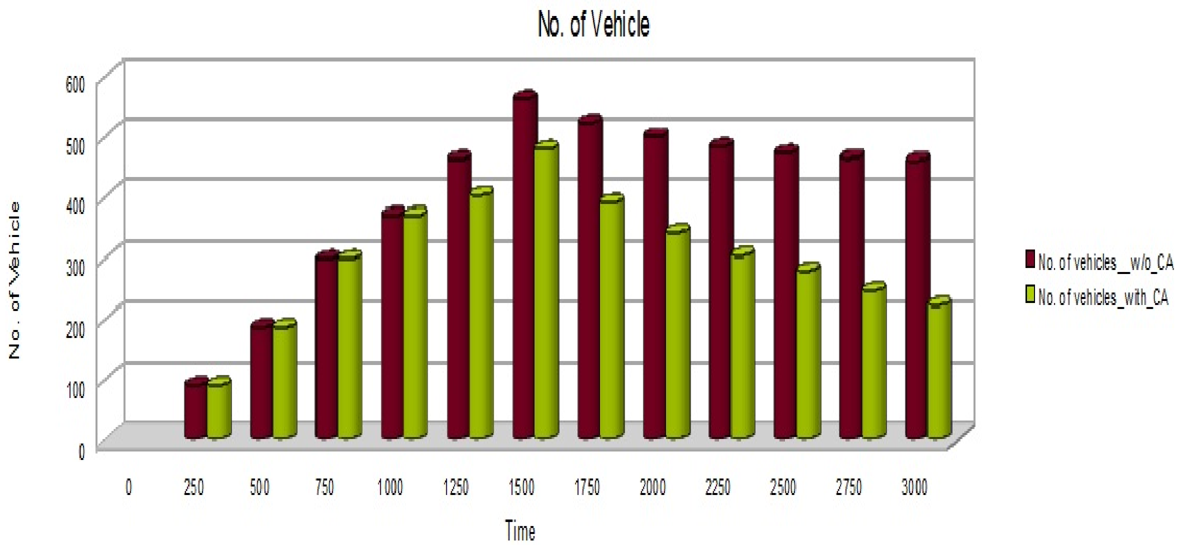


Figure 5.17: Time wise No. of Vehicles for accident case in city scenario

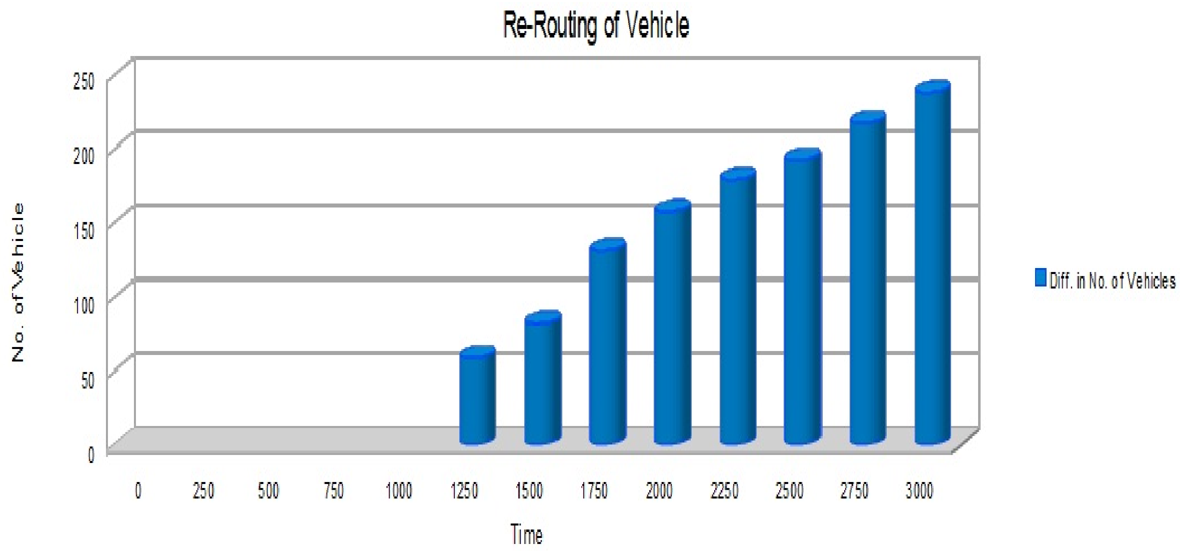


Figure 5.18: Re-routing of vehicle for accident case in city scenario

5.1.5 Estimation Error

Error Estimation of Entire Road Block(5 *5 km)

Here we are comparing the results generated from simulation and result retrieved from proposed new protocol based approach and find the estimation error for given scenario. For finding estimation error first the difference(D) is taken between simulation result and proposed approach. Then D*D calculated for each. Then from the available square difference Sum is calculated. Here we are getting sum ($\Sigma(D * D)$) 8508.From it we derived Mean Deviation (Δ) 472.67 and Standard Deviation (σ) 21.74. As shown in Table: 5.7 reading of no. of vehicles are taken for each 120 second from the standard deviation we can say that estimation error is very less 21.74 for 2500 no .of vehicles. So total percentile error is 0.8696.

Estimation Error in Road Block of RB41-45

Same as above in table: 5.8 reading is taken for road block RB41-45. In table second and third column no. of vehicles given by simulation and proposed approach result. After the calculation, here we are getting sum ($\Sigma(D * D)$) 989.From it we derived Mean Deviation (Δ) 54.94 and Standard Deviation (σ)7.41. As shown in Table: 5.7 reading of no. of vehicles are taken for each 120 second from the standard deviation.Here percentile error is 1.7555%.

Table 5.7: Estimation Error of Entire Scenario

Time	Total No. of Veh.[Simulator] (A)	Total No. of Veh [Protocol](B)	Difference (D)	D*D
0	0	0	0	0
120	164	159	5	25
240	360	357	3	9
360	612	607	5	25
480	912	896	16	256
600	1274	1253	21	441
720	1711	1663	48	2304
840	2107	2060	47	2209
960	2385	2346	39	1521
1080	2078	2072	6	36
1200	1604	1610	-6	36
1320	1051	1035	16	256
1440	659	637	22	484
1560	475	449	26	676
1680	327	312	15	225
1800	160	158	2	4
1845	33	32	1	1
1920	0	0	0	0
			$\Sigma(D*D)$ Mean Deviation (Δ) Standard Deviation (σ) Standard Deviation (σ)%	8508 472.67 21.74 0.8696

Table 5.8: Error Estimation of RB41-45

Time	Total No. of Veh.[Simulator] (A)	Total No. of Veh [Protocol](B)	Difference (D)	D*D
0	0	0	0	0
120	40	43	-3	9
240	80	85	-5	25
360	120	121	-1	1
480	160	157	3	9
600	216	210	6	36
720	276	259	17	289
840	335	329	6	36
960	406	386	20	400
1080	362	360	2	4
1200	301	304	-3	9
1320	326	237	-11	121
1440	165	165	0	0
1560	121	115	6	36
1680	87	84	3	9
1800	47	48	-1	1
1845	10	12	-2	4
1920	0	0	0	0
			$\Sigma(D*D)$ Mean Deviation (Δ) Standard Deviation (σ) Standard Deviation (σ)%	989 54.94 7.41 1.7855

Table 5.9: Error Estimation of Trip

Veh. No	Type	Time with CA [Simulator]	Time with CA[Protocol]	Difference (D)	D*D
8	CAR	760	765	-5	25
125	CAR	963	970	-7	49
436	CAR	904	910	-6	36
736	CAR	870	876	-6	36
836	CAR	1160	1168	-8	64
1023	CAR	1028	1035	-7	49
1324	CAR	500	504	-4	16
1688	CAR	880	886	-6	36
1734	CAR	830	836	-6	36
1863	CAR	1018	1025	-7	49
				$\Sigma(D * D)$ Mean Deviation (Δ) Standard Deviation (σ) Standard Deviation (σ)%	396 39.6 6.29 0.2859

Estimation Error in Trip time

Here we want to find estimation error between result given from simulation and time given by proposed approach. In Table:5.9 Trip time for difference vehicle is given. First the difference between both the result is calculated. After that we calculate the difference between time in D. After D^2, we derived the sum ($\Sigma(D * D)$) 396.From it we derived Mean Deviation (Δ) 39.6 and Standard Deviation (σ)6.29. Here percentile error is 0.5241%.

Chapter 6

Conclusion

We implemented Car Agent (CA) and Signal Agent (SA), which can do the communication with each other. SA gathers traffic information from vehicles traversing through the given road segment and can suggest optimal path to other CAs. SA computes vehicle density and average speed periodically (every 15 seconds) of particular road block.

Summary of major project finding of our work are as follows:

- With proposed approach we get the reduction of 10.05% in trip time. Reduction in no. of vehicles 10.08% in entire traffic road scenario and 9.82% in road block which have heavy traffic (RB41-RB45 in simulation scenario).

- We generated accident on a given road segment to see effectiveness of proposed approach in rerouting vehicles. Total 72.63% vehicles changed their path due to awareness of congestion on the road block having accident.

- Estimation error in average vehicle count on different road blocks is 7.41 vehicles (1.7941%). Estimation error in total vehicle count for entire road network is observed to be 21.75 vehicles (0.9886%). Estimation error in average trip time of all the vehicles is 6.29 seconds which is negligible (less than 1%) compared to overall trip time.

6.1 Future Work

In this thesis work we considered homogeneous vehicles in simulations and assumed that all the vehicles are VANET enabled. We considered lane-based traffic for vehicle mobility. In future we plan to analyse our approach for heterogeneous vehicles and lane-less traffic movement which is more realistic for representing traffic in developing countries. Also, we plan to analyse effect of penetration of VANET technology on error in estimating traffic information.

Bibliography

[1] *Stephan Olariu and Michele Weigle, "Vehicular networks from theory to practice", CRC press*

[2] *Yi qian, and nader moayeri : "Design secure and application-oriented vanet" , National institute of standards and technology*

[3] *Sherali zeadally,ray hunt ,yuh-shyan chen, and aamir hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges", springer science+business media, llc 2010*

[4] Y. L. Morgan, "Notes on DSRC and WAVE standards suite: its architecture, design and characteristics", IEEE communication Surveys and tutorials

[5] Sajjid akbar mohammad, asim rashheed, amir qayyum, "Vanet architecture and protocol stack: a survey"

[6] Car to car communication consortium manifesto, v1.1, august 2007

[7] Nidhi and d.k. Lobiyal, "Performance evaluation of realistic vanet using traffic light scenario", international journal of wireless and mobile networks (ijwmn) vol. 4, no. 1, february 2012

[8] Trupti gajbhiye1, akhilesh a. Waoo2 and p.s pathija3, "Traffic management through intercommunication between cars using vanet system", international journal on advanced computer engineering and communication technology vol-1 issue:1

[9] Bjrn schnemann, jan w. Wedel and ilja radusch, "V2X-based traffic congestion recognition and avoidance, tamkang journal of science and engineering, vol. 13, no. 1, pp. 63-70 (2010)

[10] Jerome harri, fethi filali, and christian bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy", ieee communications surveys tutorials, vol. 11, no. 4, fourth quarter 2009

[11] Vaishali d. Khairnar ; Dr. S.n.pradhan , "Mobility models for vehicular ad-hoc network simulation" , international journal of computer applications (0975 - 8887) Volume 11- No.4, December 2010

[12] J. Harri, M. Fiore, F. Filali, and C. Bonnet, "A Realistic Mobility Simulator for Vehicular Ad Hoc Networks", Technical Report RR- 05-150, Institut Eurecom, January 2007.

[13] Mitul K. Patel, "Comparative Study of Vehicular Ad-hoc Network Mobility Models and Simulators", Gujarat Technological University, Shree Swami Atmanand Saraswati Institute of Technology, International Journal of Computer Applications (0975 - 888) Volume 47- No.6, June 2012

[14] Aamir Hassan, "VANET Simulation", Technical Report, IDE0948, School of Information Science, Computer and Electrical Engineering Halmstad University, May 2009.

[15] Vaishali d. Khairnar ; Dr. S.n.pradhan , "Comparative Study of Simulation for Vehicular Ad-hoc Network", International Journal of Computer Applications (0975 - 8887)Volume 4- No.10, August 2010

[16] Vaishali d. Khairnar ; Dr. S.n.pradhan , "Simulation Based: Study and Analysis of Routing Protocol in Vehicular Ad-hoc Network Environment", International Journal of Computer Applications Vol 3 (2), 617-625

[17] Francisco J. Martinez, Chai Keong Toh, Juan-Carlos Cano, Carlos T. Calafate and Pietro Manzoni , "A survey and comparative study of simulators for vehicular ad hoc networks (vanets)", WIRELESS COMMUNICATIONS AND MOBILE COMPUTING Wirel. Commun. Mob. Comput. (2009), Wiley interscience

[18] NS2 Available at : "http://www.isi.edu/nsnam/ns/index.html"

[19] NS3 Available at : "http://www.nsnam.org/"

[20] Glomosim Available at : "http://pcl.cs.ucla.edu/projects/glomosim/"

[21] Qualnet Available at : "http://www.scalable-networks.com/content/"

[22] Jist/SWANS Available at : "http://vanet.info/node/98"

[23] SNS Available at : "http://www.cs.cornell.edu/people/egs/sns/"

[24] Openet Available at : "http://www.opnet.com/solutions/networkrd/modeler.html"

[25] J-Sim Available at : "http://www.j-sim.zcu.cz/"

[26] Vanetmobisim Available at : "http://vanet.eurecom.fr/"

[27] SUMO Available at : "http://sumo.sourceforge.net/"

[28] Michael Behrisch, Laura Bieker, Jakob Erdmann and Daniel Krajzewicz, "SUMO - Simulation of Urban Mobility", SIMUL 2011 : The Third International Conference on Advances in System Simulation

[29] Freesim An Overview: "http://www.freewaysimulator.com/"

[30] Citymob Available at : "http://www.grc.upv.es/Software/citymob.html"

[31] MOVE Available at : "http://lens.csie.ncku.edu.tw/Joomla-version/index.php/research-projects/past/18-rapid-vanet"

[32] STRAW Available at : "http://www.aqualab.cs.northwestern.edu/projects/STRAW/index. .php "

[33] CORSIM Available at: "http://mctrans.ce.ufl.edu/featured/tsis/"

[34] VISSIM Available at : "http://www.ptv-vision.com/en-uk/products/vision-traffic-suite/ptv-vissim/overview/"

[35] PARAMICS Available at : "http://www.paramics.com/"

[36] Smartahs Available at : "http://www.path.berkeley.edu/smart-ahs/Default.htm"

[37] Microscopic Traffic Applet Available at : "http://mit.edu/its/mitsimlab.html"

[38] Groovenet Available at: "http://www.seas.upenn.edu/ rahulm/Research/groovenet/"

[39] Trans Available at: "http://wiki.epfl.ch/trans"

[40] Micha Pi orkowski, Maxim Raya, Ada Lezama Lugo, Panos Papadimitratos, Matthias Grossglauser, and jeanpierre Hubaux, National Competence Center in Research on Mobile Information and Communication Systems

[41] Mobireal Available at : "http://www.mobireal.net/"

[42] Nctuns Available at: "http://sertel.upc.es/ maguilar/simulators.html"

[43] U.S. Census Bureau TIGER system database, "http://www.census.gov/geo/www/tiger"

[44] J. Tian, J. Haehner, C. Becker, I. Stepanov, K. Rothermel "Graph- Based Mobility Model for Mobile Ad-Hoc Network Simulation", Annual Simulation Symposium 2002.

[45] Dikaiakos, M.; Florides, A.; Nadeem, T. and Iftode, L. Location-aware services over vehicular ad-hoc networks using car-to-car communication Selected Areas in Communications, IEEE Journal on, IEEE, 2007, 25, 1590-1602

[46] Dornbush, S. Joshi, A. StreetSmart traffic: Discovering and disseminating automobile congestion using VANET's Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th, 2007, 11-15

[47] Abderrahmane Lakas and Moumena Chaqfeh: A Novel Method for Reducing Road Traffic Congestion Using Vehicular Communication, IEEE Journal on, IEEE,2011, 978-1-4244-8329-7/11/

[48] Tamura, K. Hirayama, M. Toward realization of VICS-Vehicle Information and Communication System Vehicle Navigation and Information Systems Conference, 1993., Proceedings of the IEEE-IEE, 1993, 72-77

[49] Collins, K. Muntean, G. Route-Based Vehicular Traffic Management for Wireless Access in Vehicular Environments Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th, 2008, 1-5

[50] A. Lakas, M. Cheqfah, "Detection and Dissipation of Road Traffic Congestion Using Vehicular Communication", Proc. of IEEE Mediterrannean Microwave Symposium, Tangiers, November 2009.

[51] Vansimfm will be available at "http://www.uc3m.es/portal/page/portal/research/researchgroups/seti/itse SAVE20project/Results1"

[52] AxelWegener, Micha Piorkowski, Maxim Raya, Horst Hellbruck, Stefan Fischer and Jean-Pierre Hubaux: "TraCI: An Interface for Coupling Road Traffic and Network Simulators", CNS '08,155-163.

[53] Evaluations of Intelligent Traffic Signal Control Algorithms under Realistic Landmark-based Traffic Patterns over the NCTUns Network Simulator By Shie-Yuan Wang and Yu-Wei Li, 15th International IEEE Annual Conference on Intelligent Transportation Systems

[54] S.Y. Wang and C.C. Lin, "NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches," 2nd IEEE International Symposium on Wireless Vehicular Communications, September 21-22, 2008, Calgary, Canada.

[55] S.Y. Wang and C.L. Chou, "NCTUns Tool for Wireless Vehicular Communication Network Researches," Simulation Modelling Practice and Theory, Vol. 17, No. 7, pp. 1211-1226, August 2009

[56] S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, and C.C. Lin, "The Design and Implementation of the NCTUns 1.0 Network Simulator," Computer Networks, Vol. 42, Issue 2, June 2003,pp. 175-197.

[57] "NCTUns 6.0 Network Simulator and Emulator," web site at http://NSL.csie.nctu.edu.tw/nctuns.html.

[58] Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights By Carolina Tripp Barba, Miguel A ngel Mateos, Pablo Reganas Soto, Ahmad Mohamad Mezher, Monica Aguilar Igartua , Universitat Polit'ecnica de Catalunya (UPC). Barcelona, Spain

[59] Simulating VANET Scenario in Nctuns-6.0 By Shivani Singh1, Dr B.V.R Reddy2 National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing (RTMC) 2011 Proceedings published in International Journal of Computer Applications (IJCA)

[60] A Comparative Study of Data Dissemination Models for VANETs By Tamer Nadeem, ITR ANI-0121416 and CNS-0520123

[61] T. Nadeem, S. Dashtinezhad, C. Liao, L. Iftode, "TrafficView: A Scalable Traffic Monitoring System," Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM), Berkeley, California, USA, January 19-22, 2004.

[62] A Review of Various VANET Data Dissemination Protocols by Rakesh Kumar1 and Mayank Dave2 International Journal of u- and e-

Service, Science and Technology Vol. 5, No. 3, September, 2012

Appendix A

Study of Simulators

A.1 Network Simulator

Network simulator is mainly designed for evaluating network performance, such as throughput, delay, jitters, etc. I introduce here some popular Network simulators, such as NS-2, NS-3, GloMoSim, Qualnet, JiST/SWANS, SNS, OpeNET, J-Sim. Network simulators are particularly useful to test new networking protocols or to propose modifications to existing ones in a controlled and reproducible manner.

A.1.1 NS-2

[18] NS-2 is an open-source discrete event network simulator that supports both wired and wireless networks, including most MANET routing protocols and an implementation of the IEEE 802.11 MAC layer. As mentioned earlier, NS-2 has an implementation of the random trip mobility model. NS-2 simulates the physical layer and the important parameters that influence its behaviour (e.g., channel fading). NS-2 is the most widely-used simulator for networking research.

A.1.2 NS-3

[19] NS-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use. The ns-3 project is committed to building a solid simulation core that is well documented, easy to use and debug, and that caters to the needs of the entire simulation workflow, from simulation configuration to trace collection and analysis.

A.1.3 GloMoSim

[20] GloMoSim is the open-source version of QualNet, developed in 1999 at UCLA. Both QualNet and GloMoSim were designed initially to support MANET simulations, therefore they include a large set of wireless and MANET routing protocols, as well as physical layer implementations. GloMoSim 's mobility is Random waypoint, Random drunken, and Trace based. The manual for GloMoSim is at.

A.1.4 Qualnet

[21] QualNet is the commercial simulator, developed in 1999 at UCLA. As a commercial product, QualNet includes a sophisticated GUI for setting up, modifying source codes and running simulations. The mobility includes Random waypoint, Random drunken, and Trace based.

A.1.5 JiST/SWANS

[22] SWANS (Scalable Wireless Ad hoc Network Simulator) was developed to be a scalable alternative to NS-2 for simulating wireless networks. Based on comparisons of SWANS, GloMoSim, and NS-2, SWANS was the most scalable, the most efficient in memory usage, and consumed the least runtime. Recently, the network model in SWANS was validated against NS-2. It was shown that along with better performance, SWANS delivered similar results as NS-2, at least for the network components that were implemented in both.

A.1.6 SNS (Staged Network Simulator)

[23] SNS is a staged simulator which is based on ns2. Staged simulation proposes to eliminate redundant computations through function caching and reuse. The central idea behind staging is to cache the results of expensive operations and reuse them whenever possible. On a commonly used ad-hoc network simulation setup with 1500 nodes, SNS executes 50 times faster than regular ns2; approximately 30x of this improvement is due to staging, the rest to engineering.

A.1.7 OpeNET

[24] OPNET is a sophisticated network simulator that includes implementations of many wireless technologies such as MANETs, IEEE 802.11 wireless LANs, WiMAX, Bluetooth, and satellite. OPNET is GUI-based and includes a graphing package for presenting results of network measurements. The mobility model in OPNET includes Random Drunken Model (randomly selects from four directions or stationary at each interval), Random Waypoint Model (Stop-Think-GO), Trace (load the location replacement), Path loss-matrix (mobility is determined by a matrix).

A.1.8 J-Sim

[25] J-Sim is an open-source simulation environment, developed entirely in Java. J-Sim provides two mobility models: trajectory-based and random waypoint. J-Sim is presented as an alternative to NS-2 because it is designed to be easier to use, but J-Sim has not been updated since 2006. In J-Sim, a process() method handles incoming events.

A.2 Traffic Simulator

One of the main challenges posed by VANETs simulations is the faithful characterization of vehicular mobility at both macroscopic and microscopic levels, leading to realistic non-uniform distributions of cars and velocity, and unique connectivity dynamics. I introduce here some popular Traffic simulators, such as VanetMobiSim/ CanuMobiSim, SUMO, FreeSim, and City-Mob ,MOVE, STRAW,TSIS-CORSIM, VISSIM, PARAMICS, SHIFT/SmartAHS, Microscopic Traffic Applet.

A.2.1 VanetMobiSim/ CanuMobiSim

[26] VanetMobiSim is an extension of the CanuMobiSim project. Canu MobiSim is designed for mobility simulation by CANU (Communication in Ad Hoc Networks for Ubiquitous Computing) research group at University of Stuttgart. The VanetMobiSim extension focuses on vehicular mobility and supporting more maps. VanetMobiSim implements new mobility models. Vehicles regulate their speed depending on nearby cars, overtake each other and act according to traffic signs in presence of intersections. VanetMobiSim mobility patterns have been validated against TSIS-CORSIM (or CORSIM)- a well known and validated traffic generator - proving the high level of realism reached by VanetMobiSim.

A.2.2 SUMO

[27][28] SUMO (Simulation of Urban Mobility) is an open-source mobility simulator that uses Random Waypoint path movement and the Krau car-following model. SUMO supports maps from TIGER/Line and ESRI. MOVE is an extension to SUMO that allows its vehicle mobility traces to be imported into NS-2 or QualNet.

A.2.3 FreeSim

[29] FreeSim is a fully-customizable macroscopic and microscopic free-flow traffic simulator. FreeSim allows for multiple freeway systems to be easily represented and loaded into the simulator as a graph data structure with edge weights determined by the current speeds. Traffic and

Table A.1: Comparison of Network Simulator

	NS-2	NS-3	GloMoSim	Qualnet	JiST/SWANS	SNS	OpeNET	J-Sim
Open Source	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Free available	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Portability	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
GUI support	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Mobility	Random trip	RandomWalk	RWP	RWP	RWP	RWP	RWP	RWP
Console support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Software support	C++	C++, Python	Parsec (C based)	C	Java	*	*	Java, Python
Platform	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux
Scalability	Yes	Yes	Yes	Yes	Yes	Yes	*	Yes
Scalability Level	Moderate	Moderate	High	High	High(most)	High	High	High
Larger Network	No	No	Yes	Yes	Yes(Max)	Yes	Yes	Yes
Speed	Medium	Medium	Medium	Medium	Fast	Faster	Medium	Medium
802.11P Support	Only for ns-2.33	No	No	No	No	No	No	No
Phy. Layer Consider	No	No	No	No	No	No	No	No
Obstacles Support	No	No	No	No	No	No	No	No
Road topo. Support	No	No	No	No	No	No	No	No
Ease of setup	Simple	Simple	Moderate	Moderate	Hard	Simple	Moderate	Simple
Ease of use	Hard	Hard	Hard	Hard	Hard	Hard	Moderate	Simple

graph algorithms can be created and executed for the entire network or for individual vehicles or nodes, and the traffic data used by the simulator can be user-generated or be converted from real-time data gathered by a transportation organization. The vehicles in FreeSim can communicate with the system monitoring the traffic on the freeways, which makes FreeSim ideal for Intelligent Transportation System simulation.

A.2.4 CityMob

[30] CityMob is a mobility pattern generator for VANETs, designed to be used with the ns-2 simulator. CityMob can generate traces for VANETs scenarios using three different mobility models: Simple, Manhattan, and Downtown.

A.2.5 MOVE

[31] A tool MOVE (MObility model generator for VEhicular networks) is used to facilitate users to rapidly generate realistic mobility models for VANET simulations. Our tool MOVE is built on top of an open source micro-traffic simulator SUMO The output of MOVE is a mobility trace file that contains information of realistic vehicle movements which can be immediately used by popular simulation tools such as ns-2 or qualnet. In addition, MOVE provides a set of Graphical User Interfaces that allows the user to quickly generate realistic simulation scenarios without the hassle of writing simulation scripts as well as learning about the internal details of the simulator.

A.2.6 STRAW

[32] STRAW (STreet RAndom Waypoint) provides accurate simulation results by using a vehicular mobility model on real US cities, based on the operation of real vehicular traffic. STRAW's current implementation is written for the JiST/SWANS discrete-event simulator, and its mobility traces cannot be directly used by other network simulators, such as ns-2. STRAW is part of the C3 (Car-to-Car Cooperation) project. A more realistic mobility model with the appropriate level of detail for vehicular networks is critical for accurate network simulation. The STRAW mobility model constrains node movement to streets defined by map data for real US cities and limits their mobility according to vehicular congestion and simplified traffic control mechanisms.

A.2.7 TSIS-CORSIM

[33] CORSIM (Corridor Simulation), developed by by the U.S. Department of Transportation and Federal High Way Administration (FHWA), is a powerful commercial microscopic traffic simulator. This simulator is specially designed for highways and surface streets, including traffic

signals and stop signs. CORSIM is a featured tool by the US Federal Highway Administration and has been widely used in the transportation research community. CORSIM is developed from two separate programs: NETSIM (an arterial simulator) and FRESIM (an expressway and interstate freeway simulator). CORSIM combines these two simulators to provide a complete simulator.

A.2.8 VISSIM

[34] VISSIM is produced by the German company PTV. Like CORSIM, it is a powerful commercial microscopic traffic simulator. VISSIM can be used to simulate various types of roadways from simple highways to urban streets with intersections and traffic signals. The mobile object on roadways can be cars, trucks, buses, heavy rail, trams, LRT, bicyclists and pedestrians. VISSIM includes a 3D animation feature to allow users to visualize the simulation. VISSIM runs on Windows platforms only.

A.2.9 PARAMICS

[35] The PARAMICS suite, developed by Quadstone Ltd, is a set of high-performance micro simulation software tools. PARAMICS can simulate intersections, highways, urban areas, work zones, etc. It also includes a 3D visualization tool. PARAMICS runs on Microsoft Windows platforms only. PARAMICS suite includes eight modules: Modeler, Processor, Analyzer, Monitor, Designer, Estimator, Converter, and Programmer.

A.2.10 SHIFT/SmartAHS

[36] SmartAHS is an Automated Highway Systems (AHS) simulator developed as part of the California PATH project at UC-Berkeley. It was originally built to simulate automated vehicles, but a human driver model [DS01], based on the cognitive driver model COSMODRIVE [Bel98], and was later added. SHIFT/SmartAHS is still available for free download from PATH. SHIFT is a programming language. SmartAHS is based on SHIFT.

A.3 VANET Simulator

In these types of applications, the traffic simulator feeds the network simulator with position information, speed, acceleration, direction, etc. There are only a few integrated frameworks available. Currently, the mobility and network models in integrated frameworks are implemented in two separated simulation tools. Therefore, there is a clear need for an integrated mobility and network simulator in order to evaluate effective performance. I introduce here some popular Traffic simulators, such as GrooveNet, TraNS, MobiREAL, NCTUns.

Table A.2: Comparison of Traffic Simulator-1

	VanetMobiSim	SUMO	FreeSim	CityMob	MOVE	STRAW	TSIS-CORSIM
Open Source	Yes	Yes	Yes	Yes	Yes	Yes	No
Free available	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Portability	Yes	Yes	Yes	Yes	Yes	Yes	Yes
GUI support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Graphical Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Animation Support	No	No	No	No	No	No	No
Console support	No	Yes	No	Yes	Yes	*	*
Map Support	RWP, UD, RM	RWP, UD, RM	RM	Manhattan, RWP	RWP,UD, Real map	RWP	RWP
Language support	Java	C++,Java	Java	C	Java	Java	C/C++ and FORTRAN
Platform	any	Linux	Linux	Linux	Linux	Linux	Windows
Microscopic	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Macroscopic	No	No	Yes	No	No	No	No
Multiple LAN	Yes	Yes	*	Yes	Yes	Yes	Yes
LAN Changing	Yes	Yes	*	Yes	Yes	Yes	Yes
Speed Limitation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Support Diff. Vehicle	No	Yes	No	Yes	Yes	Yes	Yes
Manage Veh. Density	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Traffic Light Support	Yes	Yes	*	Yes	Yes	Yes	Yes
Manage Trans. Range	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ease of setup	Moderate	Moderate	Simple	Simple	Simple	Moderate	Simple
Ease of use	Moderate	Tough	Simple	Simple	Moderate	Moderate	Simple

Table A.3: Comparison of Traffic Simulator-2

	VISSIM	PARAMICS	SHIFT /SmartAHS	MTA
Open Source	No	No	No	Yes
Free available	Viewer is Free	Demo	Yes	Yes
Portability	Yes	Yes	Yes	Yes
GUI support	Yes	Yes	Yes	*
Graphical Support	Yes	Yes	Yes	Yes
Animation Support	Yes	Yes	No	No
Console support	No	No	No	No
Map Support	RWP	*	*	*
Language support	C++	C	Shift	Java
Platform	Windows	Windows	*	*
Microscopic	Yes	Yes	Yes	*
Macroscopic	*	*	*	*
Multiple LAN	Yes	Yes	Yes	Yes
LAN Changing	Yes	Yes	Yes	Yes
Speed Limitation	Yes	Yes	Yes	Yes
Support Diff. Vehicle	Yes	Yes	Yes	Yes
Manage Veh. Density	Yes	Yes	Yes	Yes
Traffic Light Support	Yes	Yes	Yes	Yes
Manage Trans. Range	Yes	Yes	Yes	Yes
Ease of setup	Simple	Simple	Simple	Simple
Ease of use	Simple	Simple	Simple	Simple

A.3.1 GrooveNet

[38] GrooveNet is a hybrid simulator which enables communication between simulated vehicles and real vehicles. By modeling Inter Vehicular Communication within a real street map based topography, it eases protocol design and in-vehicle deployment. GrooveNet's modular architecture incorporates mobility, trip and message broadcast models over a variety of link and physical layer communication models. GrooveNet supports simulations of thousands of vehicles in any US city as well as the addition of new models for networking, security, applications, and vehicular interactions. It provides multiple network interfaces, and allows GPS and event-triggered (from the vehicles' onboard computer) simulations.

A.3.2 TraNS

[39][40] TraNS (Traffic and Network Simulation Environment) is a simulation environment that integrates both a mobility generator and a network simulator and it provides a tool to build realistic VANET simulations. TraNS provides a feedback between the vehicle behaviour and the mobility model. For example, when a vehicle broadcasts information reporting an accident, some of the neighbouring vehicles may slow down. TraNS is an open open-source project providing an application-centric evaluation framework for VANETs. TraNS is written in Java and C++ and works under Linux and Windows (trace-generation mode). The current implementation of TraNS uses the SUMO traffic simulator and the ns-2 network simulator.

A.3.3 MobiREAL

[41] MobiREAL provides a new methodology to model and simulate realistic mobility of nodes and evaluate MANET applications. It is a network simulator that can simulate realistic mobility of humans and vehicles, and allow the changing of their behaviour depending on a given application context. MobiREAL can easily describe mobility of nodes using C++. It adopts a probabilistic rule based model to describe the behaviour of mobile nodes, which is often used in cognitive modelling of human behaviour. The proposed model allows one to describe how mobile nodes can change their destinations, routes and speeds/directions based on their positions, surroundings (obstacles and neighbouring nodes), and information obtained from applications.

Table A.4: Comparison of VANET Simulators

	GrooveNet	TraNS	MobiREAL	NCTUns
Use	New	(SUMO + NS2)	GTNetS	New
Open Source	Yes	Yes	No	Yes
Free available	Yes	Yes	Yes	Yes
Portability	Yes	Yes	Yes	Yes
GUI support	Yes	Yes(Best)	Yes	Yes
Graphical and Animation inter phase	Yes	Yes	Yes	Yes
Console support	Yes	Yes	Yes	Yes
Two-way communication	Yes	Yes	Yes	Yes
Time[Continuous/Discrete]	Discrete	Discrete	Discrete	Discrete
Software support	C++	Java, C++	C++	C
Platform	Linux	Linux, Windows	Windows	Linux(Fedora 9)
Microscopic	Yes	Yes	Yes	Yes
Generate real time GPS trace	Yes	No	No	No
Support Multiple LAN	Yes	Yes	Yes	Yes
Support LAN Changing	Yes	Yes	Yes	Yes
Speed	street speed	street speed	street speed	Random
Individual Vehicle Speed support	Yes	Yes	Yes	Yes(autopilot mode)
Support Overtaking	Yes	Yes	Yes	Yes
Support Different Vehicle	Yes	Yes	Yes	Yes
Traffic Light Support	Yes	Yes	Yes	Yes
Traffic lights	Manually defined	Manually defined	Manually defined	User defined
Trip	Dijkstra based	Random, manually defined	Manually defined	Manually defined
802.11P Support	Yes	Yes	No	Yes
Map Support	TIGER	TIGER,Shape file maps	Yes	OSM Shape File
Street-level topology view	Yes	Yes	Yes	Yes
Google Earth Visualization	No	Yes	No	No
Safety Messages	Yes (Vehicle warning)	Yes (Road danger warning)	No	No
Handle emergency message with priority	Yes	No	No	No
Dynamic reroute	No	Yes	No	No
Adaptive rebroadcast	Yes	No	No	No
Real time vehicle interaction Support	Yes	No	No	No
Obstacles Support	No	No	No	Yes
Usage Frequency	High	average	average	High
zoom ability	Yes	Yes	Yes	Yes
Ease of setup	Moderate	Moderate	Simple	Tough
Ease of use	Tough	Moderate	Tough	Tough
User Friendly	Better	Best	Good	Good

Appendix B

NCTUns

B.1 VANET Simulator

NCTUns(National Chiao Tung university network simulator) is used as VANET Simulator.NCTUns is both a microscopic traffic simulator and a network simulator. NCTUns is a software tool that integrates user-level processes, operating system kernel, and the user-level simulation engine into a cooperative network simulation system. The NCTUns network simulator is a high-fidelity and extensible network simulator capable of simulating various devices and protocols used in both wired and wireless networks. NCTUns provides many unique advantages that cannot be easily achieved by traditional network simulator such as OPNET Modeler and ns-2. NCTUns is written in C++ with a powerful GUI support.[54]

B.1.1 Different components on NCTUns

[55][56]

1) Dispatcher: NCTUns provides a flexibility by which the GUI program and the simulation engine program can be run on different machines. It sends the inquiry message to know which simulation server is currently available. The Dispatcher program is responsible for monitoring the status of the simulation servers to serve the simulation request issued from the GUI program.
2) Coordinator: The Coordinator program has the following four tasks:

- Processing the commands sent from Dispatcher
- Forking (creating) a simulation engine process to perform a simulation
- Reporting the status of the created simulation engine process to the Dispatcher program
- Collecting the simulation results produced by its created simulation engine process and

sending them to the GUI program

3) NCTUns Client: NCTUns provides a front-end GUI program (called "nctunsclient" in its package), which provides useful facilities for users to efficiently create simulation scenario. GUI has been further classified into four components.

a) The "Draw Topology" mode: In this mode, one can insert network nodes, create network links, and specify the locations and moving paths of mobile nodes. In addition, the GUI program provides a complete tool kit for users to construct road networks.

b) The "Edit Property" mode: In this mode, one can double-click the icon of a network node to configure its properties (e.g., the network protocol stack used in this node, the applications to be run on this node during simulation, and other parameters).

c) The "Run Simulation" mode: In this mode, the GUI program provides users with a complete set of commands to start/pause/stop a simulation. One can easily control the progress of a simulation by simply pressing a button on the GUI control panel.

d) The "Play Back" mode: After a simulation is finished, the GUI program will automatically switch itself into the "Play Back" mode and read the packet trace file generated during the simulation. In this mode, one can use the GUI program to replay a node's packet transmission/reception operations in an animated manner.

4) Simulation Engine The simulation engine program is composed of a set of various protocol modules and an event scheduler. The former is responsible for simulating protocol behaviors while the latter is responsible for scheduling events. The simulation engine can be thought of as a small operating system kernel. It performs basic tasks such as event processing, timer management, packet manipulation, etc. Its API plays the same role as the system call interface provided by an UNIX operating system kernel.

5) Application Program Application programs are responsible for generating network traffic in a simulated network.

6) Kernel Patches NCTUns uses the real-life Linux network protocol stack to "simulate" transport-layer and network-layer protocols, such as TCP, UDP, IP, and ICMP.

B.1.2 Required tools for Installing NCTUns 6.0

- Fedora 12 (torrent.fedoraproject.org/torrents/Fedora-12-i686-Live.torrent)

- NCTUns 6.0 (http://nsl10.csie.nctu.edu.tw/products/nctuns/download/download.php)

- Virtual Box (If want to work with Window)(www.virtualbox.org/wiki/Downloads)

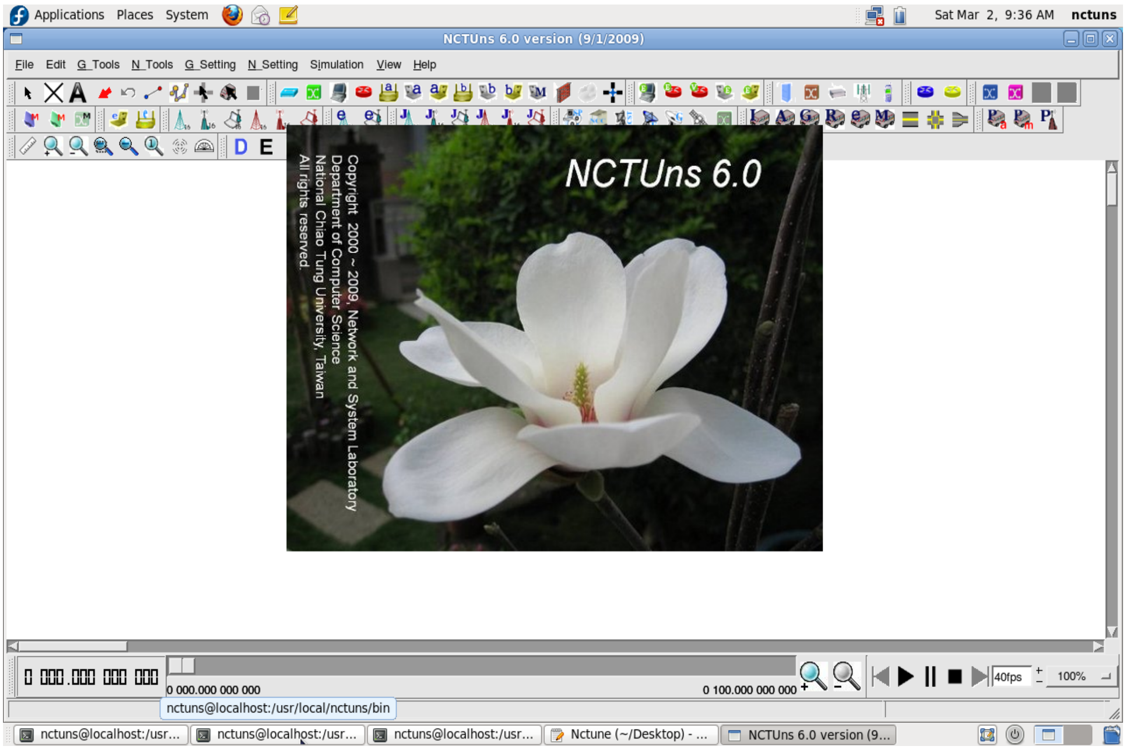


Figure B.1: Startup Screen of NCTUns 6.0

B.1.3 Installation steps of NCTUns 6.0

Step 1: If use want to work with window then install Virtual Box and then load fedora 12 iso image in it.

Step 2 : After the successfully installation user can start with NCTUns installation.

Step 3 : Extract NCTUns 6.0.rar

Step 4: Open Terminal and browse the NCTUns 6.0 folder

Step 5: Enable terminal with SU(Super User)

Step 6: Install Library yum groupinstall "Development Libraries" "Development Tools" "Legacy SoftwareDevelopment"

Step 7: Install mkinitrd rpmfile

yum install mkinitrd -y

Step 8: Install the xterm rpmfile

yum install xterm

Step 9: Give command ./install.sh

Step 10: Configure shell preferences by including the NCTUns environment variables.

Add this into "/home/user/.bashrc" file.

export NCTUNSHOME=/usr/local/nctuns

export NCTUNS_TOOLS=$NCTUNSHOME/tools$

$exportNCTUNS\_BIN$ =NCTUNSHOME/bin

export PATH=$NCTUNS_BIN$:PATH

Step 11: Restart Fedora

Step 12: After successful installation verify the new kernel of NCTUns6.0 in grub loader.

Step 13: Login with User name "nctuns" and Password "nctuns"

Step 14: Start NCTUns Dispatcher through following step.

Start new terminal and open directory "/usr/local/nctuns/etc/"

Enable terminal with SU(Super User)

Write command "./dispatcher"

Step 15: Start NCTUns Coordinator.

Start new terminal and open directory "/usr/local/nctuns/etc/"

Enable terminal with SU(Super User)

Write command "./coordinator"

Step 16: Start NCTUns Client.

Start new terminal and open directory "/usr/local/nctuns/etc/"

Enable terminal with SU(Super User)

Write command "./nctunsclient"

B.1.4 Scenario Generation -Path II(Using NCTUns)

Step 1: Start NCTUns Dispatcher

Step 2: Start NCTUns Coordinator

Step 3: Start NCTUns Client.

Step 4: Draw Topology: Design or import road structure and load the VANET nodes on the road.

Step 5: Edit Property: Configure the node and change parameters if required.

Step 6: Run Simulation: Set the Simulation Time and other real time and dynamic Settings.

Step 7: Play Back: After Successfully Run state Play back for Study[57].

B.1.5 Functionality of NCTUns 6.0

- In simulator,user can design any road structure using available road network component.
- In simulator, user can select different types of Car-OBU(On board Unit) and RSU (road Side Unit). Also manage the Speed and Route of vehicles.
- In simulator graphical representation of V2V and V2I is possible.

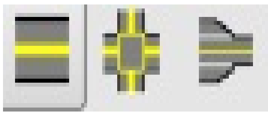


Figure B.2: Road Structure

Figure B.3: Different types of Car-OBU

Figure B.4: Different types of -RSU

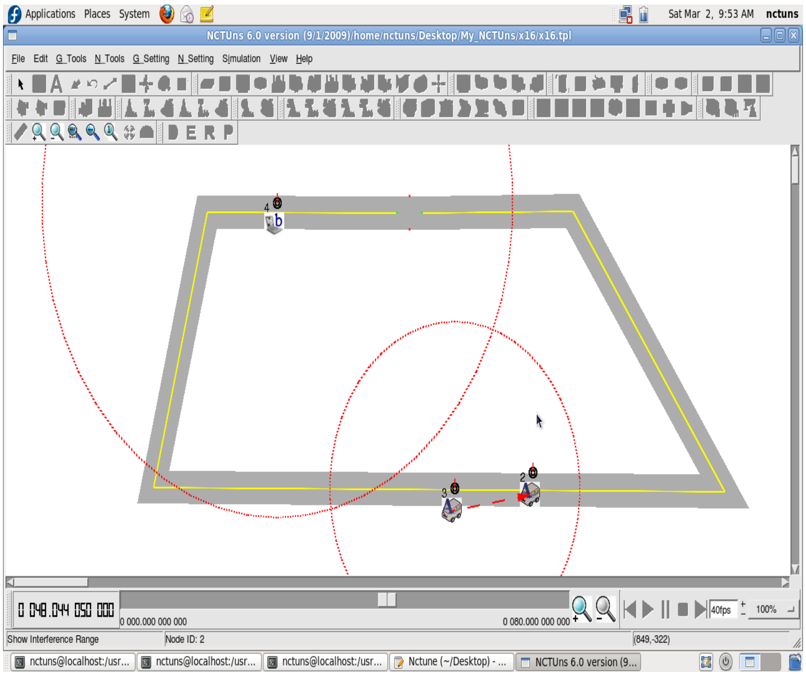


Figure B.5: V2V Communication

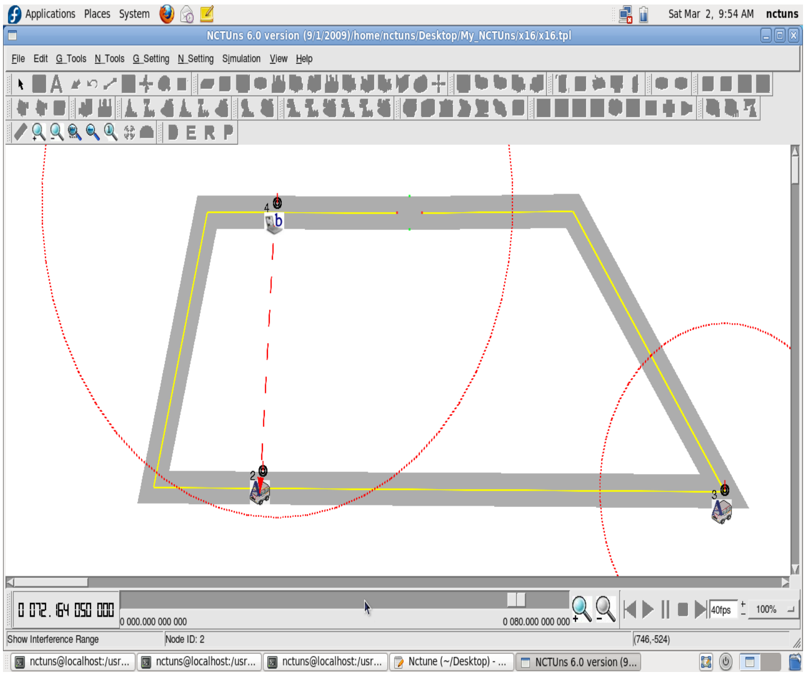


Figure B.6: V2I Communication

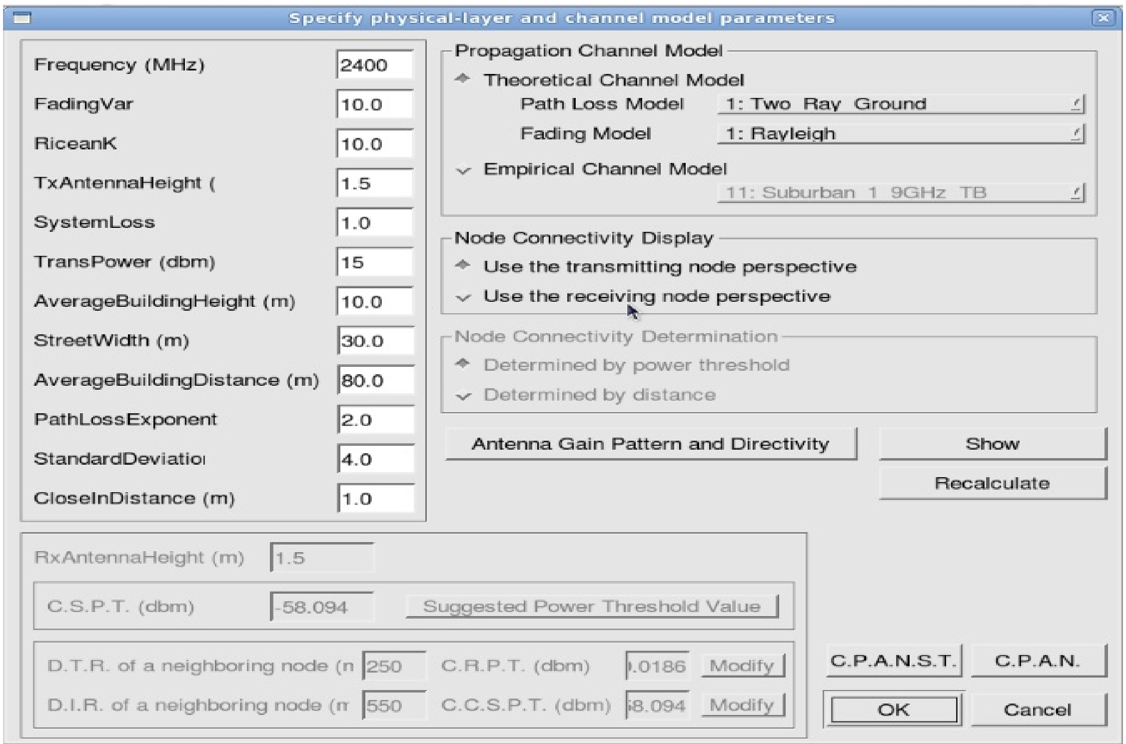


Figure B.7: Physical parameters

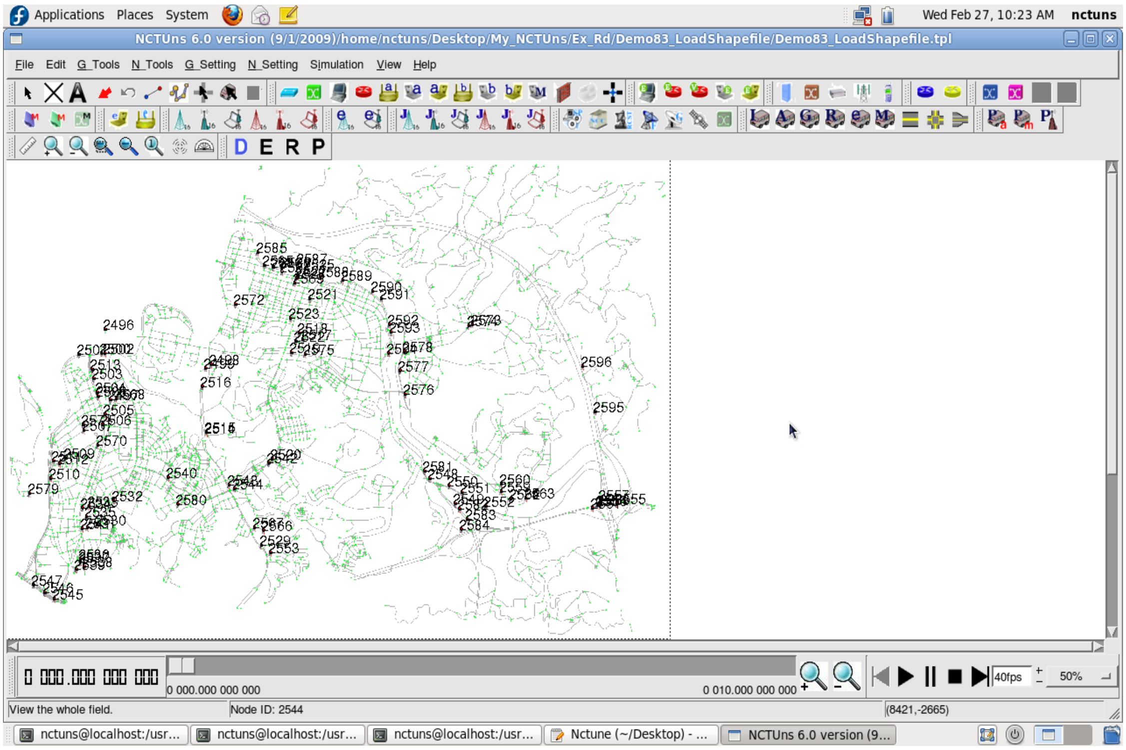


Figure B.8: Road map of city

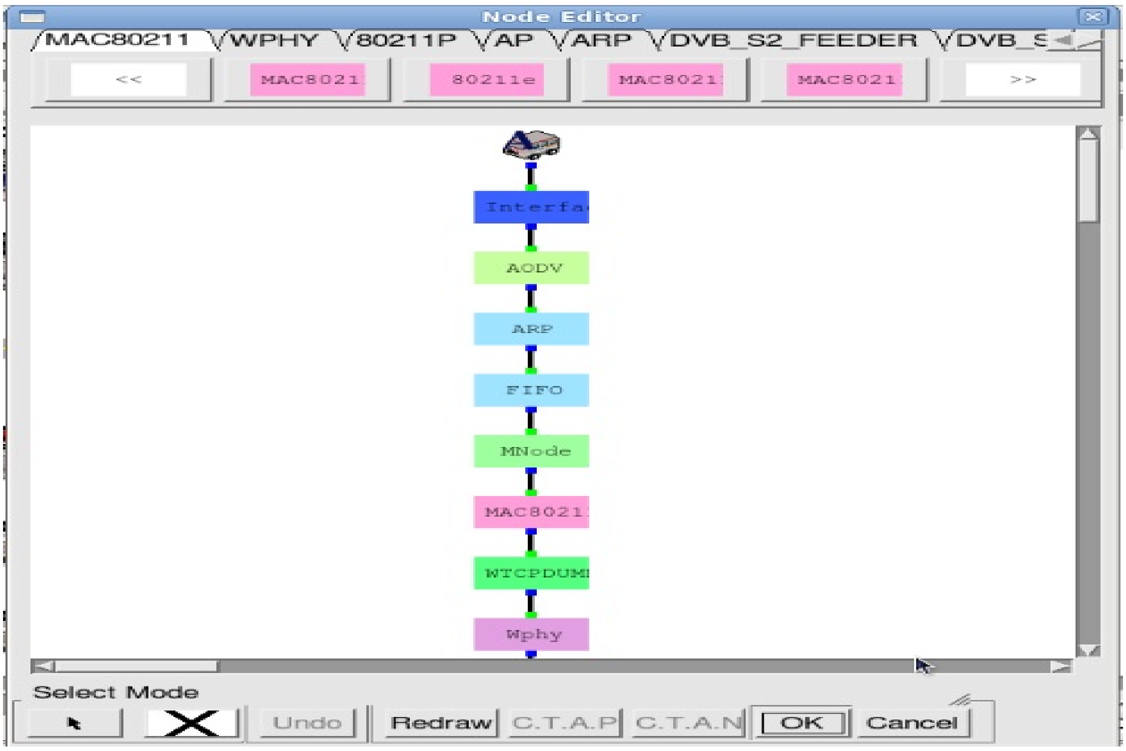


Figure B.9: Node editor

- User can manage the transmission range and other physical parameters.
- Simulator give support of shape file to load road map of city.
- Simulator give support of Node editor , which allow modification in any layer and give support of future protocols for researchers[58].

Index