Optimizing Architecture of Vehicle to Infrastructure Communication.

Prepared By

POOJA SHAH

09MCES08

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

February-2012

Optimizing Architecture of Vehicle to Infrastructure Communication.

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Pooja Shah

(09MCES08)

Guide

[Dr. S. N. Pradhan]

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May-2012

Undertaking for Originality of the Work

I, Pooja P. Shah, Roll . No. 09MCES08, give undertaking that the Major Project entitled "Optimizing Architecture of Vehicle to Infrastructure Communication" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering (Split) 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.

Signature of Student
Date:
Place: Ahmedabad

CERTIFICATE

This is to certify that the Major Project entitled " Optimizing Architecture of Vehicle to Infrastructure Communication" submitted by Pooja P. Shah (09MCES08), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, 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.

Dr. S.N. Pradhan
Guide and Professor,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Prof. D. J. Patel
Professor and Head,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

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

Abstract

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. This project aims to achieve a better vehicle to infrastructure communication.

The application under consideration is providing a zero traffic lane (Z-Lane) for an ambulance. To achieve the goal first a suitable vehicular mobility model is identified. Krauss Mobility Model is used for simulation of road traffic scenario. The scenarios are implemented with the help of Simulation of Urban Mobility (SUMO) which is a road traffic simulator based on Krauss Mobility Model. For the normal working of the said application, Ambulances using the service are assumed to have their MAC address registered. Ambulance driver will be setting the destination coordinates (this information will be carried in the routing protocol header for further decision making) in the application unit and will trigger the infrastructure informing its presence. On reception of this trigger, road side infrastructure broadcasts alert messages informing the vehicle drivers to vacant the lane. By doing so the ambulance will get a zero traffic lane in advance.

The problem addressed in this project is the scenario where no infrastructure is present in the range of ambulance. In this case, the trigger from the ambulance must reach the nearest infrastructure as early as possible for efficient working of the application. For this purpose vehicle to vehicle communication is used. Various routing algorithms have been analyzed in terms of their end to end delay and packet delivery fraction. Ad-hoc On Demand Distance Vector (AODV) routing algorithm is chosen for the said communication. The AODV protocol has significant amount of end to end delay. The project aims to reduce the Route REQuest (RREQ) packet (this carries information regarding the destination

position also) generation in AODV. This is done by modifying the protocol in such a way that RREQ packets are sent only to the neighbor which gives highest advancement towards the destination. If no such neighbor is found the store carry forward approach is used for packet advancement. The simulation results of the implementation of modified AODV shows that the number of RREQ packets reduces drastically and in turn end-to-end delay also reduces. The network traffic simulation is done with the help of Network Simulator - 2 (NS-2). To generate the scenario in SUMO and converting it into NS-2 readable form MObiligy generator for VEhicular network (MOVE) is used.

Acknowledgements

With immense pleasure, I would like to present this report on the dissertation work related to "Optimizing Architecture for Vehicle to Infrastructure Communication". I am very thankful to all those who helped me for the successful completion of the first phase of the dissertation and for providing valuable guidance throughout the project work.

I would first of all like to offer thanks to **Dr. S. N. Pradhan** (Guide and Programme Coordinator M.Tech. CS&E), Institute of Technology, Nirma University, Ahmedabad whose keen interest and excellent knowledge base helped me to finalize the topic of the dissertation work. His constant support and interest in the subject equipped me with a great understanding of blacks and whites of the required architecture for the project work.

Along with him I cannot miss to thank Prof. Sharda Valiveti and Nikunj Doshi whose motivation and work are my driving forces. My sincere thanks and gratitude to **Prof. D. J. Patel**, Professor and Head, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his 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.

- Pooja Shah

09MCES08

Contents

List of Tables

List of Figures

Abbreviation Notation and Nomenclature

AODV — Ad-hoc On-demand Distance Vector routing
AOMDV — Ad-hoc On-demand Multipath Distance Vector routing
AU — Application Unit
CFM — Car Following Model
DSDV — Destination Sequenced Distance Vector
DSR — Dynamic Source Routing
DSRC — Direct Short Range Communication
ESRI — Environment Systems Research Institute
GDF — Geographic Data Files
GHR — Gazis Herman Rothery
IDM — Intelligent Driver Model
ITS — Intelligent Transport System
MANET — Mobile Ad-hoc NETwork
MOVE — MObility Generator for VEhicular network
NS-2 — Network Simulator - 2
OBU — On-Board Unit
PDF — Packet Delivery Fraction
RREP — Route REPly
RREQ — Route REQuest
RSU — Road Side Units
SUMO — Simulation of Urban MObility
TCL — Tool Command Language
TIGER — Topologically Integrated Geographic Encoding and Referencing system
V2I — Vehicle to Infrastructure
V2V — Vehicle to Vehicle
VANET — Vehicular Ad-hoc NETwork
VMM — Vehicular Mobility Model
WAVE — Wireless Access in Vehicular Environment
Z-Lane — Zero-traffic Lane

Chapter 1

Introduction

1.1 VANET and Application Areas

Recent advancements in short range wireless communication have enabled the user devices with a rich set of networking possibilities. In particular, ad-hoc network has emerged as one of the most researched areas in the networking avenue. Vehicle automation has further opened the gateways to the promising vehicular ad-hoc network (VANET). VANETs are self-organizing networks established among vehicles equipped with communication facilities. After the deployment of various vehicular technologies such as active road signs or toll collection, Vehicular communication (VC) systems have emerged. They comprise network nodes, that is, vehicles and roadside infrastructure units (RSU) equipped with on-board sensory processing, and wireless communication modules. Vehicle to infrastructure (V2I) and Vehicle to vehicle (V2C) communication can enable a wide range of applications to enhance transportation safety and efficiency, as well as infotainment.

1.2 Deviation from other MANETs

VANET being a type of MANET have some similarities with the mobile networks like it also uses multi-hop communication. But a one-size-fit-all solution is not possible in implementing various networking protocols for different variants of MANETs because each network has its own peculiarities.

Domains:	**Individual Components:**
In-vehicle	OBU (On-Board Unit))
Ad-hoc	AU (Application Unit)
Infrastructure	RSU (Road Side Unit)

Table 1.1: Categories of data dissemination between components

Peculiarities of VANETs: [1]

i Nodes move at very high speeds.

ii Number of nodes may be comparatively large.

iii Mobility patterns of the nodes are constrained by topology of roads, streets, speed limits etc.

iv Because of the onboard navigator, the nodes often know their own position and street maps of surrounding area.

v Highly dynamic topology.

vi Network is expected to be highly disconnected, as vehicles tend to travel around forming groups, but the distance between the groups can be much longer than the communication range.

No other family of mobile networks features hundreds or even thousands of nodes that travels at the speed of tens of kmph, that alternates high velocity intervals with full stop periods and whose freedom of movement is constrained by precise topologies like road topology and detailed rules like driving rules.

1.3 System architecture

The system architecture of VANET contains the following self-explanatory components.

Data dissemination approaches:

- In presence of road side units: push based and pull based
- In absence of road side units: flooding and relaying

Push based: data broadcasted to everyone.
Pull based: query-response based data dissemination.

Flooding: broadcast to all neighbors, suitable for delay sensitive application/sparsely connected/fragmented network, generates high message overhead.
Relaying: relay node is selected which is responsible to forward the packet further and so on, less contention.

1.4 V2I Communication Standard

Vehicular environments enforce a whole new set of requirements on today's wireless communication systems. The IEEE 802.11 standard body is currently working on a new amendment, IEEE 802.11p, to address these requirements. This amendment document is named Wireless Access in Vehicular Environment, aka WAVE. The draft document for IEEE 802.11p is progressing closer towards acceptance by the general IEEE 802.11 working group. IEEE 1609-family standards, are also developed to facilitate the provision of wireless access in vehicular environments.

WAVE protocol stack [7]
It can be observed that the WAVE physical (PHY) and medium access control (MAC) layers are based on IEEE 802.11p, which is an amendment to IEEE 802.11a for applying in the rapidly varying vehicular environment.

In the PHY layer, the main modifications for IEEE 802.11p are the number of channels, the size of channel bandwidths, and modulation and coding schemes. Seven channels with 10 MHz bandwidth for each are specified for overcoming the extreme multipath environments. With the utilization of the enhanced modulation and coding scheme, data transmission

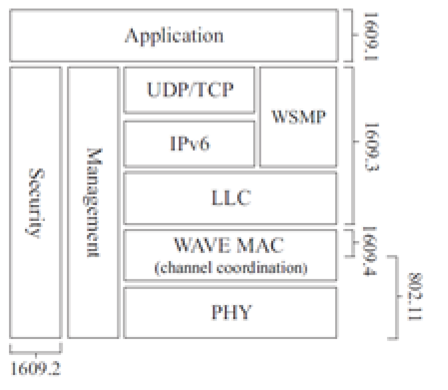

Figure 1.1: WAVE Protocol Stack

rates ranging from 3 to 27 Mbps are supported in high-speed mobile environments.

The functionalities of each channel and the enhanced priority control for traffic delivery are specified in the MAC part of the IEEE 802.11p. Furthermore, the authentication process for a device can be optionally ignored, and the transmission of data frames with different basic service set identifications (BSSIDs) is allowed among communicated devices.

As shown in Figure, the upper layers of the WAVE/DSRC architecture are the IEEE 1609-family standards, which extend the specification of the IEEE 802.11p standard to cover additional layers in the protocol suite. The documents include IEEE 1609.1 [3], IEEE 1609.2 [4], IEEE 1609.3 [5], and IEEE 1609.4 [6].

IEEE 1609.4 provides enhancements to the IEEE 802.11p MAC to support multi-channel operation. The specification of the functions associated with the logical link control (LLC), network, and transport layer of the OSI model are given in the IEEE 1609.3 standard. It provides addressing and data delivery services within a WAVE/DSRC system for V2V and V2I communications. IEEE 1609.2 covers the format of secure messages and their encryption/decryption processes. An application with methods for remote management between a resource manager and a remotely device is defined in IEEE 1609.1, which is categorized into the application layer of the OSI model.

1.5 Z-Lane

Goal of the project: Providing a zero traffic lane (Z-Lane) to an ambulance.

This section provides the details about the said goal.

1.5.1 Assumptions

Following are the assumptions for achieving the said goal:

a. The ambulance and other vehicles are intelligent vehicles.

b. The road under consideration is equipped with road side communication units.

c. The ambulance using this application has registered its MAC address in the system.

d. Emergency situation like accident or fire on the road is informed by the VANET in consideration by some forwarding technique.

1.5.2 Desired Working

The desired working of the said application is as under:
Ambulance departure and its destination will be informed by its driver to the VANET infrastructure or some other source. Based on this information the application must manage to transmit this information to the vehicles using the road such that the ambulance gets the lane clear till the destination. V2V communication must also be used to speed up the message forwarding. The infrastructure database must be cleared as soon as the ambulance passes past it so that the lane used by the ambulance can be accessed by other vehicles asap.

1.5.3 Design Issues

Under listed are the issues to be taken care of while achieving the goal:

- Different vehicles could have different transmission ranges.
- Transmission ranges of each vehicle could change while travelling.

- Communication is asymmetric.
- Global network topology knowledge is often unfeasible to obtain.
- Driver not following the instructions he gets from the infrastructure.
- Infrastructure not present for a significantly longer road span.

1.5.4 Objective

To achieve Z-Lane the technical objectives to be achieved are:

a. Identifying and simulating a Vehicular Mobility Model (VMM), Z-Lane Vehicular Mobility Model (Z-Lane VMM), considering the application.

b. Simulating various routing algorithms with the mobility trace generated as outcome of first step. This is to identify the best fit routing strategy for further optimization.

c. Proposing and simulating a topology control mechanism to optimize V2I communication.

Chapter 2

Literature Survey on Mobility Models

As mentioned in the section 1.5.4, first objective is to come up with a mobility model for Z-Lane.

Mobility modeling is gradually becoming as a critical step for simulation based evaluations of vehicular communication protocols to reach sufficient level of reliability and consistency with real world implementations. Connectivity dynamics play a major role in determining the performance of networking protocols. This is why VANETs' study cannot be separated from mobility analysis. To realize metropolitan-scale vehicular networks it is required to identify the level of mobility that such system would be required to support, both quantitatively and qualitatively.

2.1 Introduction

The mobility model is used to describe the movement pattern of mobile users, and how their location, velocity and acceleration change over a period of time. As mobility patterns may play a significant role in determining the performance of various protocols, it is desirable for mobility models to emulate the movement pattern of targeted real life applications in a reasonable way.

Various approaches can be adopted for mobility modeling in VANETs. All the mobility models undergo a trade-off between complexity and precision. A classification of mobility models based on the level of detail of motion representation is as under [1]:

1 Macroscopic: models gross quantities like vehicular density or mean velocity, treating vehicular traffic as fluid dynamics

2 Mesoscopic: individual mobile entities are modeled at an aggregate level. It exploits gas-kinetic and queuing theory results or macroscopic-scale metrics, such as velocity/density relationships to determine motion of vehicles.

3 Microscopic: considers each vehicle as distinct entity, modeling its behavior in more precise but computationally expensive way. Used to model fine grained real world situations like lane changing, front to rear car interactions, flow merging at ramps.

The above mentioned traditional models become less significant when considering vehicular mobility modeling. Given the reduced spatial scale of short- and middle-range communication techniques envisioned for employment in vehicle-based networked systems, vehicular network simulations often require a high level of detail in terms of car motion representation. The necessity of precision of the order of meters in the definition of vehicles' absolute and relative positions, bounds the mobility descriptions to be used for network simulation to the microscopic or mesoscopic domain. Thus, a better fitting categorization as mentioned below is considered in VANET by differentiating on the nature of the diverse analytical representation of car motion encountered in the vehicular networking literature.

1 Stochastic models: Vehicle movement is regarded at a microscopic level and is (1) constrained on a graph representing the road topology, and (2) random, in the sense that mobile entities follow casual paths over the graph, traveling at randomly chosen speed. Stochastic models are the most trivial way to mimic car mobility.(Refer appendix A)

2 Traffic stream models: Vehicular mobility is treated as a continuous phenomenon. Traffic stream models determine cars' speeds, leveraging fundamental hydrodynamic physics relationships between the velocity, density, and outflow of a fluid, and thus fall into the macroscopic or mesoscopic categories defined before.(Refer appendix A)

3 Car-following models: The behaviour of each driver is computed on the basis of the state (position, speed, and acceleration) of the surrounding vehicles. Best to analytically describe microscopic-level mobility in vehicular traffic flow theory. (Refer section 2.2)

4 Flows-interaction models: Built upon car-interaction representations belonging to the stochastic and car-following classes above, and thus falling into the microscopic category as well, flows interaction characterizes the dynamics of vehicular flows merging, for example, at highway ramps or urban intersections. (Refer section 2.3)

2.1.1 Notations Used in Representing VMM

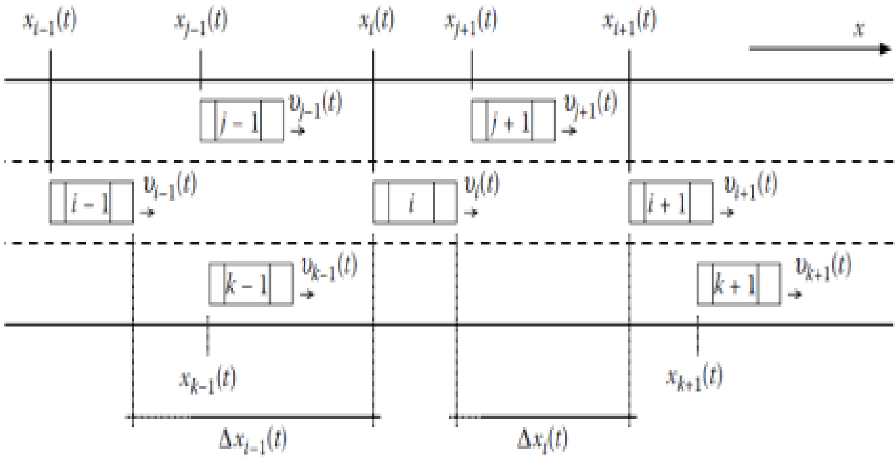

Figure 2.1: Vehicular mobility modeling notations

- The index i is used to refer the vehicle under investigation, while i 1 identifies the front (+) and back (-) vehicles on the current lane.
- For some vehicle i at time t, $x_i(t)$, and $v_i(t)$ represent its speed and position, i.e. its instantaneous acceleration can be expressed as $dv_i(t)/dt$.
- The distance between front bumper of i to back bumper i + 1 is identified as $x_i(t)$.

Parameter	**Symbol**
Acceleration	a
Declaration	b
Maximum allowed/Desired speed	v_{max}
Bumper to bumper safety distance	Δx_{safe}
Safety time headway	Δt_{safe}
Driver's reaction time	τ
Time step(discrete time model)	Δt
Space step(discrete space model)	Δx

Table 2.1: Notations used in vehicular mobility modeling

- The relative speed $v_{i+1}(t) - v_i(t)$ is denoted by $v_i(t)$.
- According to its definition, in the following a positive vi(t) will always mean that the distance of car i from its leading vehicle i + 1 is growing.
- The back and front cars on the left lane with respect to the some vehicle i is traveling on are denoted by j - 1 and j + 1, respectively.
- The back and front cars on the right lane with respect to the some vehicle i is traveling on are denoted by k - 1 and k + 1, respectively. Also, we indicate common model input parameters as summarized in following table.

2.2 Car Following Models - CFM

Car-following models describe the behavior of each driver in relation to its neighboring vehicles. CFMs fall into the category of microscopic-level descriptions as each car is regarded as an independent entity in CFM.

There are two basic categories of CFMs:

1 Follow The Leader Model

2 Cellular Automata (Refer appendix A)

2.2.1 Follow the Leader Model

Follow the leader model: Most car-following models determine the motion of a vehicle as a function of the state of a single neighboring car, typically the one in front. For this reason,

they are also referred to as follow-the-leader models. In such descriptions, the speed or acceleration depend on factors such as the distance from the front car and the absolute and relative speed or acceleration of both vehicles.
With reference to the notation introduced in Figure 2.1 and table 3.1, a general expression for a car-following model, in the form of a delayed differential equation formulation for a vehicle i, is shown in equation 2.1

$$\frac{d}{dt}v_i(t) = -f[v_i(t), v_{i+1}(t), \Delta x_i(t)] \tag{2.1}$$

The CFMs under consideration are GHR 2.2.1.1, Linear Model 2.2.1.2, Intelligent Driver Model 2.2.1.3 and Krauss Model 2.2.1.4. At the end of this section in 2.2.2, the said models are compared and the choice for the desired application is mentioned.

2.2.1.1 GHR Model

The model is named after its authors, Gazis, Herman, and Rothery. One of the best known expressions for the function f . It is based on seminal work. With respect to previous car-following prototypes, the GHR model led to fundamental advances on the calibration of vehicular models. It was the first analytical description to introduce different calibrations for non-congested and congested traffic conditions. The GHR model is defined as in equation 2.2

$$\frac{d}{dt}v_i(t) = k_1 v_i^{k_2}(t-\tau)\frac{\Delta v_i(t-\tau)}{\Delta x_i^{k2}(t-\tau)} \tag{2.2}$$

Where k1, k2, and k3 are constants that are to be calibrated to adapt the model to specific drivers' behaviors or traffic scenarios. Their values vary the maximum instantaneous acceleration (k1), the weight of vehicle i's absolute speed (k2), and that of the bumper-to-bumper distance from the vehicle ahead (k3).
The GHR formulation introduces a delay τ in the computation of the instantaneous acceleration, in order to account for the finite reaction time of drivers.

2.2.1.2 Linear Model

It describes the acceleration of a vehicle as a linear function f of the same factors employed by the GHR. It also introduces the concept of desired following distance $\Delta x_{des}(t)$, which represents the comfortable distance a driver would like to maintain with respect to the leading vehicle, considering its current speed and acceleration. The linear model computes the speed derivative for car i as from the equations 2.3

$$k2[\frac{dv_{i-1}(t+\Delta t)}{dt} - \mid \frac{dv_{i-1}(t+\Delta t)}{dt} \mid_R] \tag{2.3}$$

Where k1, k2, k3, k4, and k5 are again constants that must be calibrated according to the traffic scenario.
Advantages over GHR: a simpler formulation, a lower computational cost, a clearer physical meaning of the parameters' impact and a higher degree of agreement with real-world traffic data. It still retains GHR's calibration problems.

2.2.1.3 Intelligent Driver Model

IDM is one of the most common car-following descriptions used in vehicular networking research. It represents an evolved GHR/linear concept. IDM characterizes drivers' behavior by the instantaneous acceleration of vehicles, calculated based on the following equations:

$$\frac{dv_i(t)}{dt} = a[1 - (\frac{v_i(t)}{v_{max}})^4 - (\frac{\Delta x_{des}(t)}{\Delta x_i(t)})^2] \tag{2.4}$$

$$\Delta x_{des}(t) = \Delta x_{safe} + [v_i(t)\Delta t_{safe} - \frac{v_i(t)\Delta v_i(t)}{2\sqrt{ab}}] \tag{2.5}$$

$\Delta x_{des}(t)$ is the desired dynamic distance which is similar to the distance calculated using the linear model. minimum bumper-to-bumper distance
Δx_{safe}, the minimum safe time headway t_{safe}, the speed difference with respect to the front vehicle delta vi(t), and the maximum acceleration and deceleration a and b. By combining these formulae the instantaneous acceleration of the car can be achieved. The resultant is divided into a desired acceleration $[1 - \frac{v_i(t)}{/v_{max}}^4]$ on a free road and a braking deceleration

which is induced by the preceding vehicle $[\frac{\delta x_{des}(t)}{\delta x_i(t)}]^2$.

2.2.1.4 Krauss Model

Krauss model is a variation of the GHR description. Different from the GHR, linear, and IDM descriptions, Krauss proposes a discrete-time representation, modeling the vehicle's speed at each time step rather than its instantaneous acceleration. The Krauss model determines the speed of a vehicle i through the following formulation:

$$v_i^{safe}(t+\Delta t) = v_{i+1}(t) + \frac{\Delta x_i(t) - \tau v_{i+1}(t)}{[v_i(t) + v_{i+1}(t)]/2b + \tau} \tag{2.6}$$

$$v_i^{des}(t+\Delta t) = min\{v_{max}, v_i(t) + a\Delta t, v_i^{safe}(t+\Delta t)\} \tag{2.7}$$

$$v_i(t+\Delta t) = max\{0, v_i^{des}(t+\Delta t), k_1 a \Delta t \eta\} \tag{2.8}$$

- Equation 2.6 is used to compute the speed v_i^{safe}. The vehicle is required not to exceed this speed in order to maintain a safety distance from its leading vehicle.

- Equation 2.7 is used to determine the desired new speed v_i^{des} of a vehicle i is equal to the current speed added to the increment determined by a maximum uniform acceleration a Δ t. The upper bounds represented by the maximum allowed speed and by the maximum safe speed vi safe computed above.

- Equation 2.8 is used to determine the final value of the speed, introducing stochastic behavior into the model by means of a noise η is a random variable uniformly distributed in [0, 1]. Such randomness cannot exceed the measure of a maximum percentage k1 of the highest achievable speed increment a delta t.

2.2.2 Comparison of Various CFMs

Analysis of the CFMs discussed in previous section:

- GHR model calculates instantaneous acceleration i.e. $\frac{dv_i(t)}{dt}$ for a given vehicle taking into account maximum acceleration, absolute speed weight for vehicle and bumper to bumper distance. All these metrics are as calibration constants. Calibration may induce delay in computation. Also once calibration is done, driver reaction time τ is also induced.

- Linear Model which is similar to GHR model adds a new metric, desired following distance. Needs calibration of five constants for desired outcome which is time consuming than the other models.

- IDM accounts instantaneous acceleration and desired following distance. It also adds safety criterion in car following decisions by introducing a metric safe distance.

- Krauss provides the way to calculate safe velocity, desired velocity. From the bases of which instantaneous velocity is calculated. The model also accounts the velocity of the vehicle being followed and thus proves to be the most efficient.

For the application under consideration Krauss will be the most appropriate model. Its Lane changing counter part also helps in the said application wherein the crucial part will be lane changing. Thus the simulator called SUMO B is chosen for simulation of scenarios required in the research.

2.3 Flow Intersection Models

Models considered in previous mobility descriptions regulate the interaction of cars within the same flow, that is, cluster of cars moving along a common axis in a uni-dimensional motion. All such models are referred as car-interaction representations.

The description of vehicular mobility provided by car-following representations is often insufficient to simulate real-world scenarios. When considering highway traffic it would be desirable that characterizing phenomena like over takings between vehicles traveling on adjacent lanes or in-flow of cars at ramps be reproduced. Similarly, when simulating urban traffic the presence of intersections and roundabouts regulating the merging of flows from different roads cannot be neglected.

2.3.1 Lane Changing

The modeling of lane changes represents a hard challenge, due to the high number of variables that come into play. The behavior of a driver cannot be described as a function of the state of the leading car. In real situation, interacting vehicular flows are also presence. It must also take into account the speed and distance of back as well as front vehicles on adjacent lanes.

The standard approach to the problem is to perform lane-changing decisions half a time step before updating vehicles' movement through car-following rules.

That is, the motion at each time step is divided into two sub-phases:

a Interlane movement. Each vehicle decides if it has to move to a different lane. If so, the vehicle shifts to the new lane.

b Intralane movement. Each vehicle changes its acceleration and speed according to the single-lane car-interaction model.

The lane-changing decision of the first phase above is usually driven by two criteria:

a. *A safety criterion.* The movement of a vehicle to a new lane must occur in a way that the safety of the vehicle itself as well as that of its new neighboring cars are not threatened. This normally translates to the requirement that a sufficiently large gap be present in the new lane for a vehicle to perform the lane change.

b. *An incentive criterion.* There must be a precise reason for a vehicle to perform the interlane movement. Generally, the motivation comes from an advantage in terms of acceleration or speed; that is, by changing lanes the vehicle is able to proceed faster.

2.3.2 Krauss Lane Changing Model(Overtaking)

The safety criterion here has similar meaning as MOBIL but consistent with Krauss car following formula. It is expressed in terms of speed instead of acceleration. The safety criterion for lane changing to the left in Krauss model is:

$$| v_{j-h}^{safe}(t + \Delta t) |_L \geqslant v_{j-h}(t) - b\Delta t, \forall h \geqslant 1 \tag{2.9}$$

The inequality in Equation 2.9 guarantees that, if the lane change is performed, each back vehicle in the new lane is still able to brake with a comfortable deceleration b and avoid a collision with the vehicle ahead.
The incentive criterion for a left -handed lane changing by vehicle is defined by the following conditions:

$$v_i^{safe}(t + \Delta t) < v_{max} \tag{2.10}$$

$$v_i^{safe}(t + \Delta t) \geq v_{jam} \tag{2.11}$$

$$| v_i^{safe}(t + \Delta t) |_L \geq v_{jam} \tag{2.12}$$

For the movement to the left to occur, the condition in Equation 2.10 must be verified; that is, vehicle i must not be able to reach the maximum desired speed on the current lane. Here the reason behind a lane change is the hope of approaching v_{max}. Also, the model prevents inter-lane movements when both lanes are experiencing traffic jams, identified by a speed of vehicles lower than a congestion threshold v_{jam}. Therefore, either Equation 2.11 or Equation 2.12 must hold; that is, at least one of the two lanes must not be congested.
When considering movements to the lane on the right, Krauss specified different criteria, to account for the asymmetry of traffic in most European countries. For a right-handed lane change, the safety criterion is similar to the one observed before:

$$| v_{k-h}^{safe}(t + \Delta t) |_R \geq v_{k-h}(t) - b\Delta t, \forall k \geq 1 \tag{2.13}$$

The incentive criterion is quite different, and it is described by the following set of inequalities:

$$v_i^{safe}(t + \Delta t) \geq v_{max} \tag{2.14}$$

$$\mid v_i^{safe}(t + \Delta t) \mid_R \geq v_{max} \tag{2.15}$$

According to above Equations, a lane change to the right occurs only when free-flow conditions are observed on both lanes; that is, the maximum speed can be reached on the right lane as well, and occupancy of the left lane is unnecessary.

The incentive criterion of the Krauss model is completed by the possibility of random lane changes, with probability k2. Therefore, a lane change to one of the two sides is performed if either the conditions described before are realized, or if the randomness inequality is verified $\eta < k2$

where η is a random variable uniformly distributed in [0, 1], and $0 <= k2 <= 1$, with k2 usually much smaller than 1.

a fourth speed update rule is added to avoid overtaking from right side

$$v_i(t + \Delta t) = \min v_i(t + \Delta t, \mid v_i^{safe}(t) \mid_L) \tag{2.16}$$

vehicle i upper bounds its speed not only on the safety speed evaluated with respect to the front vehicle on the lane it is currently traveling on, that is, i + 1, but also on that computed with respect to the front vehicle on its left lane, that is, j + 1. This way, the speed of vehicle i never exceeds the value it would have if i were traveling on the lane to its left , and no overtaking with respect to j + 1 can happen. In fact, Krauss applies such behavior only in the absence of traffic congestion, that is, when at least one of the inequalities in Equations 2.10 and 2.11 are verified. Instead, in the presence of traffic jams, passing on the right is allowed, because cars travel in large, slow clusters, and Equation e-10 is not enforced.

2.4 Intersection Management

For network simulation of vehicular traffic in urban and suburban environments study of intersection management is required. In such cases, a proper modeling of the interaction among nonparallel traffic flows must be provided (Refer appendix A

2.5 Z-Lane VMM

Minimum requirement of generating realistic mobility model are a car following model, a good intersection management and taking care of lane changing.

2.5.1 Parameters Considered for Z-Lane VMM Simulator

The following parameters are considered while choosing a the vehicular mobility model for the said application. More description about the parameters considered can be found in Appendix A.

a. Whether the model supports accurate and realistic topological maps like geographical maps obtained from TIGER (Topologically Integrated Geographic Encoding and Referencing system) database, GDF (Geographic Data Files), GPSTrack, Google Maps or ESRI(Environment Systems Research Institute) database.

b. Whether Obstacles on road are considered by the simulator. There are two types of obstacles possible:

   - Hurdles to cars mobility like Other vehicles,Intersection points and road topology constraint

   - Hurdles to communication like Communication range of the transceiver

c. Whether Trip motion simulation is possible? Simulation of vehicle traffic based on specified source and destination. In case of this project, the vehicle considered as ambulance will be defining trip motion, other vehicles will be random.

d. Human driving patterns must be based on Krauss car following model.

e. Intersection management (Refer Appendix A) should be there based on some stochastic model 2.1.

2.5.2 Survey Based Comparision

Based on the desired parameters mentioned in previous section and the available survey in [8] SUMO with MOVE will be used to generate mobility model for this project.

2.5.3 SUMO

The Simulation of Urban MObility (SUMO) [9] is an open source, highly portable, road traffic simulation package designed to handle large road networks. For the Traffic Generator part, route assignments may be imported from various sources. SUMO implements a car following model as the human driving patterns block. A stochastic traffic assignment modeled by a probabilistic route choice as path motion block. For the motion constraints, SUMO contains parsers for various formats, ranging from TIGER, GIS Arcview, or even to VISSIM. SUMO is also able to output traces directly usable by NS2. Refer Appendix B for details.

2.5.4 MOVE

In order to ease the configuration of SUMO, The Mobility Model Generator for Vehicular Networks (MOVE) [10]. It also enhances SUMO's complex configuration with an efficient GUI, and inherits all its features. Refer Appendix B for details.

Chapter 3

Literature Survey on Routing Algorithms

As mentioned in the section 1.5.4, there is a need to analyze the routing algorithms.

3.1 Introduction

Although the different networks listed below retain the same basic principles of multi-hop communications, they also have enough peculiarities to prevent a one-size-fits-all solution from being practical [15], [16].

3.1.1 Mobile ad hoc networks (MANET)

These consist of a collection of wireless nodes with arbitrary mobility patterns. Nodes are usually battery-operated, making energy efficiency one of the important design issues. Computation and memory resources can also be scarce, so routing protocols designed for MANETs should not be too complex. In addition, a particular mobility pattern cannot be assumed, nor the existence of additional valuable information such as the position or trajectory of the nodes. Routing protocols assume that the network is fully connected. That is, if the destination of a data packet lies on a different part of the network, such a packet is simply discarded.

3.1.2 Wireless mesh networks (WMN)

These are a particular case of ad hoc network in which nodes are like static base stations that are able to communicate using multihop routes. Client devices are mobile and switch among mesh nodes as they move around. Mesh nodes can be equipped with multiple radio interfaces for higher efficiency. In this case, energy, computation, and memory resources are not a concern. Mesh routers are, commonly, dedicated devices with a continuous power supply. Routing protocols are required to find the best possible routes for the aggregated user traffic.

3.1.3 Wireless sensor networks (WSN)

These consist of a set of generally tiny wireless devices with very limited energy, computation power, and memory. Therefore, energy efficiency and simple algorithms are the factors of paramount importance in these networks. They are mainly used to monitor the environment. So, a WSN may consist of hundreds or even thousands of devices. Sensors are usually assumed to have knowledge about their own positions and those of their neighborhood (commonly by including position information in periodic beacons). In many applications, the destination is a sink device that processes the data sensed by the nodes, so that its position is known a priori. When this is not the case, the destination's position is unknown and must be discovered. The process by which a node retrieves the current position of another one is commonly referred to as the "location service." Scalable location services are hard to develop.

3.1.4 Vehicular ad hoc networks (VANET)

These are a particular case of MANET in which nodes are vehicles able to move at very high speeds. In addition, they may consist of a very large number of nodes, and their mobility patterns are constrained by the topology of the roads, streets, speed limits, and so on. Cars do not usually have energy constraints, and can be equipped with high computing and communication capabilities. Thanks to the on-board navigator, they often know their own position and the street maps of the surrounding area. Nevertheless, those maps might not be fully accurate if they are out of date or if a special event is happening (e.g., a road is closed for repairs). Going on, it is expected that vehicles will issue periodic beacons to support collision avoidance applications (a control message about every 300 msec1). Position information, as well as velocity and trajectory, can be carried within these messages to learn the neighborhood topology and its evolution in time. The destination's position might be known in some cases, although this cannot be assumed in general. Finally, under many scenarios the network is expected to be highly disconnected, as vehicles tend to travel around forming groups, but the distance between groups can be much longer than the communication range.

3.1.5 Comparison of Various Networks

The table 3.1 shows the summarized comparison between various network types described in previous sub sections.

Property	**MANET**	**WMN**	**WSN**	**VANET**
Network Size	Medium	Moderate	Large	Large
Node's Mobility	Random	Static	Mostly Static	High, Non-random
Energy Limitations	High	Very Low	Very High	Very Low
Node's Computational Power	-	High	Very Low	High
Node's Memory Capacity	Low	Very Low	High	
Location Dependency	Low	Very Low	High	Very High

Table 3.1: Comparison of Properties of various networks

3.2 Overview of Routing

In general, MANET routing algorithms can be classified according to a number of criteria. The most used one classifies routing protocols as (1) proactive, (2) reactive, and (3) hybrid. Proactive protocols follow a very similar approach to those used in wired networks such as the Internet. Nodes taking part in proactive routing maintain a routing table, which is built by exchanging messages with other nodes of the network. Thus, routes are computed and maintained in advance, even if no data traffic is present in the network. A reactive routing protocol (also known as "on-demand") only searches for a route when it is needed, in other words, if the sending node does not know of a route to reach the destination. Unlike proactive routing protocols, route discovery only takes place when needed, but this usually increases to a small extent the end-to-end delay as the data source must wait some time before the routing path is established. Hybrid routing protocols are those that cannot be fully classified into the previous categories. Typical examples include those protocols that behave proactively for some destinations and reactively for others. In addition to those categories, we add for clarity an additional one: (4) geographic routing. Geographic routing generally works "on-demand." However, it is very different from traditional reactive routing protocols. Rather than routing based on the topology of the network, geographic routing protocols take routing decisions hop-by-hop in a per-packet basis. Each relay selects its next hop based on its position, its neighbors' positions, and the position of the destination.

Main technical limitations for MANET routing solutions in VANET scenarios as follows [20]:

Scalability. Most of the routing protocols designed for MANETs are only able to support a limited number of mobile nodes (about one or two hundred). The path computation mechanisms used by those protocols are very costly for very large networks such as VANETs. For instance, proactive protocols store routes to all other nodes in the network within their routing tables. In the case of a VANET, storing routing tables for all vehicles is really impractical.

Full connectivity. This assumption is not realistic in vehicular networks. Although the destination is not reachable at the moment of sending a packet, there could be a non-concurrent path between source and destination. This means that vehicles can move, carrying the packet until the destination is eventually reached. This paradigm is called delay (and disruption) tolerant networking (DTN), and is more appealing than MANET routing for delay-insensitive packets.

Mobility prediction. Most MANET routing protocols have not made any assumption about particular mobility patterns of mobile nodes. They assume arbitrary mobility patterns. Although that approach favours flexibility regarding the scenarios in which those protocols can be deployed, they are also inefficient in the cases where mobility of the nodes can be somehow predicted. This is the case for VANETs, where node movements are restricted by the topology of the streets, speed limits, traffic signals, and the like. Thus, traditional MANET routing solutions neglect the advantages that can be obtained by considering a constrained mobility pattern.

Anticipation of path breakages. MANET routing protocols deal with the mobility of the nodes (i.e., path breakages); either by periodic control messages or by periodic path creations. The timers used by MANET routing protocols are adjusted so that the protocol can react after a route breaks. However, in many VANET scenarios the knowledge about the mobility patterns of neighbouring nodes can help prevent path breakages before they happen.

Extensive use of flooding. Most MANET routing protocols are based on flooding .In reactive routing protocols the data source uses flooding to find a route to a destination. In the case of proactive protocols, every node sends periodic control messages to either its neighbourhood (one message issued by each node is as costly as a flooding initiated by one node) or the entire network. That kind of operation consumes a lot of bandwidth with control messages and limits very much the performance in large networks such as VANETs. Because in vehicular environments the number of nodes is very high (and unknown beforehand), any devised flooding mechanism must be scoped, that is, restricted to a limited area.

Non local operation. MANET routing protocols are distributed algorithms used to compute routing paths. However, the creation and maintenance of routing paths usually requires the effort of all nodes in the network. In proactive routing all nodes take part in building routing tables. In reactive protocols all nodes participate in the initial flooding required to find a route towards the destination. In VANETs with a potentially large number of nodes, localized routing solutions in which nodes only need information from their neighbourhood are more appealing in terms of scalability, control overhead, and adaptation to different network conditions. However, in order for this to work, the destination of the communication must be known or an efficient location service must be designed.

Exploitation of existing knowledge. VANET nodes can be assumed to be equipped with on-board units providing relevant information about expected trajectories, current speed and direction, topological map of streets or roads, and so on. All the information is extremely valuable for enhancing the performance of routing protocols. Unfortunately, MANET routing solutions, by trying to be effective regardless of the mobility of the nodes, usually just neglect all that information.

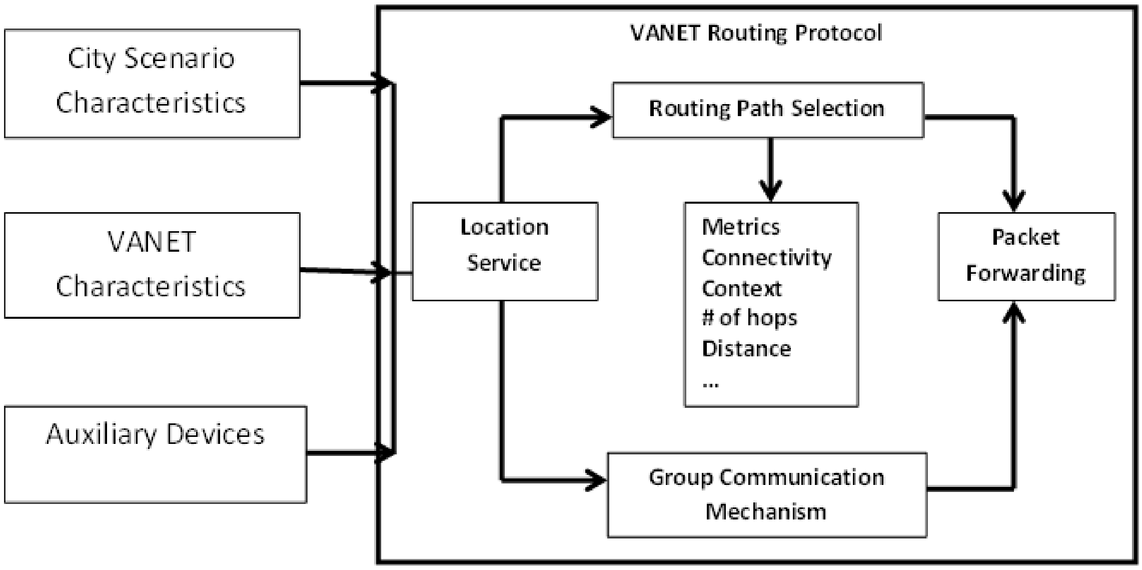

Figure 3.1: Elements of VANET Routing

3.2.1 Routing in VANET

Figure 3.1 shows the elements involved in VANET routing [18]:

3.2.2 Survey of existing VANET routing algorithm

Methods used in existing VANET Routing Protocols:
VANET protocols based on geographic routing are prone to message losses because they usually assume ideal transmission ranges, which is not applicable in real vehicular setups.Trajectory based routing must carefully choose the criterion to forward a data message, since it could get stuck or move away from the final destination, reducing the chances of successful delivery. To ensure the scalability routing decisions must be taken on the bases of information available in a node's local vicinity. Therefore, exchanging information by beaconing is the fundamental part of routing in VANET.
By using greedy heuristics and available node positions, some protocols choose as next hop the neighbor that may provide greater advancement toward the destination node position eg GPCR and CAR. A digital map can be used to identify a list of junctions that a packet must travel eg GSR and A-STAR. Previous routing algorithms assume that a path exists between

source and destination. This is not a realistic assumption. Store-carry-forward paradigm can be employed to solve the problem with disconnected network. In this case, a node stores a message when there is no neighbor which provides advance towards destination. When a suitable neighbor is found, the node forwards it. Eg. Spatial Aware Routing(SAR). Another technique is using a planned trajectory of vehicle. Data is forwarded to the node whose trajectory is more useful than the current node's. Eg. Opportunistic Geographical Routing (GeOpps).

Issues with existing Routing Protocols:
Transmission range assumption:

- The probability of reception decreases as the distance from transmitter to receiver increases.
- The geographic routing protocols use greedy approach and thus tend to transmit data to the farthest node which has less probability of reception. This problem is more severe as traffic density increases, as it is more likely to have a neighbour near the theoretic limit of range. Solution: (1) No transmission range is assumed. Send the data message without pre-selecting next hop. Among the neighbours that receive the message the farthest one is selected. (2) Make forwarding decisions based on link status with the neighbouring vehicles. Transmitter checks link status with the neighbours and selects the on having highest probability of reception and advance towards destination. Use of stale information:
  - Beacon Losses: The neighbour information is updated periodically via beacon messages. When beacons get lost due to transmission errors, some nodes become unaware of existing neighbour and the position information they have may get out-dated because of the node mobility.
  - Stale Position: Messages may be dropped due to stale neighbour positions. Outdated information may cause loops when Store carry forward paradigm is used.

Eg. A and B in opposite directions. To eliminate the loops, vehicles can piggy back their current velocity vectors within the beacon.

The surveys say that, no existing VANET routing protocol is efficient enough for taking care of VANET characteristics while routing. Thus, for the problem addressed in this project, the focus is on routing protocols which are most used in MANET scenarios successfully. MANET routing protocols viz. AODV, DSDV, AOMDV and DSR are analyzed in terms packet delivery fraction and average end to end delay [19]. For the safety critical applications in VANET it is needed that routing is done in such a way that average end to end delay is less and the packet delivery ratio(received packets/sent packets X 100) is high.

3.2.3 Experiments with Existing Routing Protocols

The experiment for all four protocols' analysis are done using Network Simulator - 2.34 on Fedora 14. The experiment parameters used are as mentioned in the table 3.2:

Parameter	**Value**
Number of Nodes	varies from 10 to 50
Area	500 X 500 meters road segment
Maximum Node Speed	10, 20 and 30 mps
Simulation time	100 seconds
Connections	5
Rate	2 packets/second
Pause time	0.0
MAC layer	IEEE 802.11p
Measurement Metrics	Packet Delivery Ratio Average End to End Delay

Table 3.2: Experiment Parameters

The pause time is taken 0.0 as per the assumption that all the vehicles are constantly moving. The experiments for different speed values are taken 10 times each. The results shown in the next section is average taken from these experiment outcomes.

3.2.4 Experiment Results

As per the parameters specified in 3.2 the results in terms of PDF and end to end delay for AODV,DSDV, DSR and AOMDV are presented in this section. The speed of nodes in the scenarios are variable from 10 to 30 mps to reach to a generic conclusion.

3.2.4.1 Maximum Node Speed 10 mps

Tables 3.3 and 3.4 shows the Packet delivery fraction and average end to end delay readings for maximum node speed parameter set to 10 mps.

No. of Nodes	AODV	DSDV	AOMDV	DSR
10	99.4197	63.7879	92.0849	84.2403
20	98.6538	46.2017	88.0859	73.487
30	99.6101	47.7387	92.5781	94.0631
40	99.0458	36.842	91.2525	92.7536
50	99.044	30.6854	90.3101	86.1953

Table 3.3: Average Packet Delivery Fraction; Max. Speed = 10mps

From table 3.3 it can be seen that the PDF of AODV does is least effected by the number of nodes considered in the scenario.

No. of Nodes	AODV	DSDV	AOMDV	DSR
10	2.51405	1.92415	2.2068	2.69914
20	4.1001	3.4474	3.88266	5.30411
30	2.5048	2.13965	2.24982	2.52368
40	2.70699	2.311	2.4995	3.3008
50	2.72494	2.2117	2.4398	3.1145

Table 3.4: Average End to End Delay; Max. Speed = 10mps

It can be observed that AODV has a good packet delivery ratio as compared to other protocols. It can also be seen that DSDV introduces the minimum end to end delay.

3.2.4.2 Maximum Node Speed 20 mps

Tables 3.5 and 3.6 shows the Packet delivery fraction and average end to end delay readings for maximum node speed parameter set to 20 mps.

No. of Nodes	AODV	DSDV	AOMDV	DSR
10	99.0329	52.6799	90.5512	57.6667
20	97.669	35.1784	78.0583	70.1803
30	98.0583	35.141	82.9173	72.1519
40	98.2625	39.0741	93.0556	82.4959
50	96.2921	29.2506	85.4932	72.5071

Table 3.5: Avg. Packet Delivery Fraction; Max. Speed = 20mps

The experiments with growing speeds and varying number of nodes suggest that the PDF of AODV is the highest and the least effected by the node density.

No. of Nodes	AODV	DSDV	AOMDV	DSR
10	15.7279	2.07272	2.51544	3.90172
20	4.4012	2.99001	3.13838	6.81308
30	3.40298	2.26758	2.55562	9.80222
40	2.62247	2.36745	2.40471	2.52449
50	3.23221	2.06427	2.37905	6.78488

Table 3.6: Average End to End Delay; Max. Speed = 20mps

It can be observed that AODV has a good pacaket delivery ratio as compared to other protocols but the end to end delay of AODV is comparatively high.

3.2.4.3 Maximum Node Speed 30 mps

Tables 3.7 and 3.8 shows the Packet delivery fraction and average end to end delay readings for maximum node speed parameter set to 30 mps.

From table 3.7 we can say that as the amount of speed increases the PDF of AODV gets some what effected by the variation in number of nodes. But still the

No. of Nodes	AODV	DSDV	AOMDV	DSR
10	97.2868	56.4062	88.14	69.1689
20	98.646	51.9448	90.8203	84.0266
30	96.8992	30.777	77.0751	60.6061
40	98.0769	39.3379	90.079	83.0159
50	99.0177	31.187	100	70.4225

Table 3.7: Avg. Packet Delivery Fraction; Max. Speed = 30mps

PDF of AODV is appreciable as compared to other protocols under consideration.

No. of Nodes	AODV	DSDV	AOMDV	DSR
10	3.3657	2.2536	2.66415	4.0733
20	2.4654	2.2292	2.15063	8.3199
30	3.73979	2.38899	3.03488	6.02244
40	2.75092	2.40022	2.5699	3.2482
50	2.95856	1.80902	2.67279	3.8061

Table 3.8: Average End to End Delay; Max. Speed = 30mps

Again the results prove AODV is better in terms of packet delivery fraction as compared to all the protocols considered here. But it can again be observed that the end to end delay of AODV is higher as compared to DSDV.

3.2.5 Conclusion

From the experimental results displayed in previous section it can be seen that, AODV performs better in all the three speed values and the average end to end delay is less when DSDV is used. As the DSDV's PDF is very less AODV is chosen for turning it to be VANET adaptable. The working of AODV protocol is as explained in section 3.3.

3.3 Working of AODV

AODV routing consists of three phases: route discovery, data transmission and route maintenance.

Phase I - Route discovery

This phase starts when a node wants to transmit data and has no route to destination. AODV call route discovery process. Source node broadcasts a Route Request Packet (RREQ) to its neighbor. Nodes that receive RREQ packets divide into three categories: the receiver node is the destination of route, the node that has a route to destination or none of both. In the two first situations, receiver unicast a Route Reply (RREP) packet to the route that received Route Request (RREQ) packet from it. The route that RREP packet traverses, selected as one of the main routes for source that has been sent RREQ packet. In the last situation receiver generate another RREQ packet and broadcast it to its neighbors. Last situation repeats until one of the first two situations occur.

Phase II - Data transmission phase

Data packets are transmitted across selected route. In this phase, it is possible that a link is broken and results in route expiration. In this situation, the Maintenance phase comes into play. Repairs broken route or find a new route to destination. Node that its link was broken, unicast a Route Error (RERR) packet to the source node. The Source node after receiving RERR packet, searches in its routing table if find another route to old destination select that route as new main route for data transmission, else rebroadcasts new RREQ packet and seeks new route to continue data transmission. If source node cannot find new route to destination, data transfer stops and failure happen. One advantage of AODV is that for any pair of source and destination finds more than one route. more often this advantage acts as disadvantage. Finding several route need to exchange more control packet. This leads to increase routing overhead. In addition, it increases bandwidth consumption. Obviously, some of scenarios use all of discovered route and others only use part of discovered route and rest of routes are wasted. All of unused routes are routing overhead and all control packets for this route are wasted.

Chapter 4

Proposed Algorithm

The desired work flow for the said application is decided as under taking into account research work mentioned in [13] and [14]:
Ambulance node:

* Activate the application by selecting destination coordinates.
* Forward the request packet.

Other nodes:
Nodes other than ambulance work as per the following flow specified in figure 4.1

4.1 Changes in AODV

In order to implement the said algorithm flow described in figure 4.1 following changes are needed in original AODV.

* The AODV header will be modified to carry the flag indicating the application activation and also the coordinate information of destination.

* The information regarding the geographical co-ordinates of the neighbor is obtained before sending RREQs.

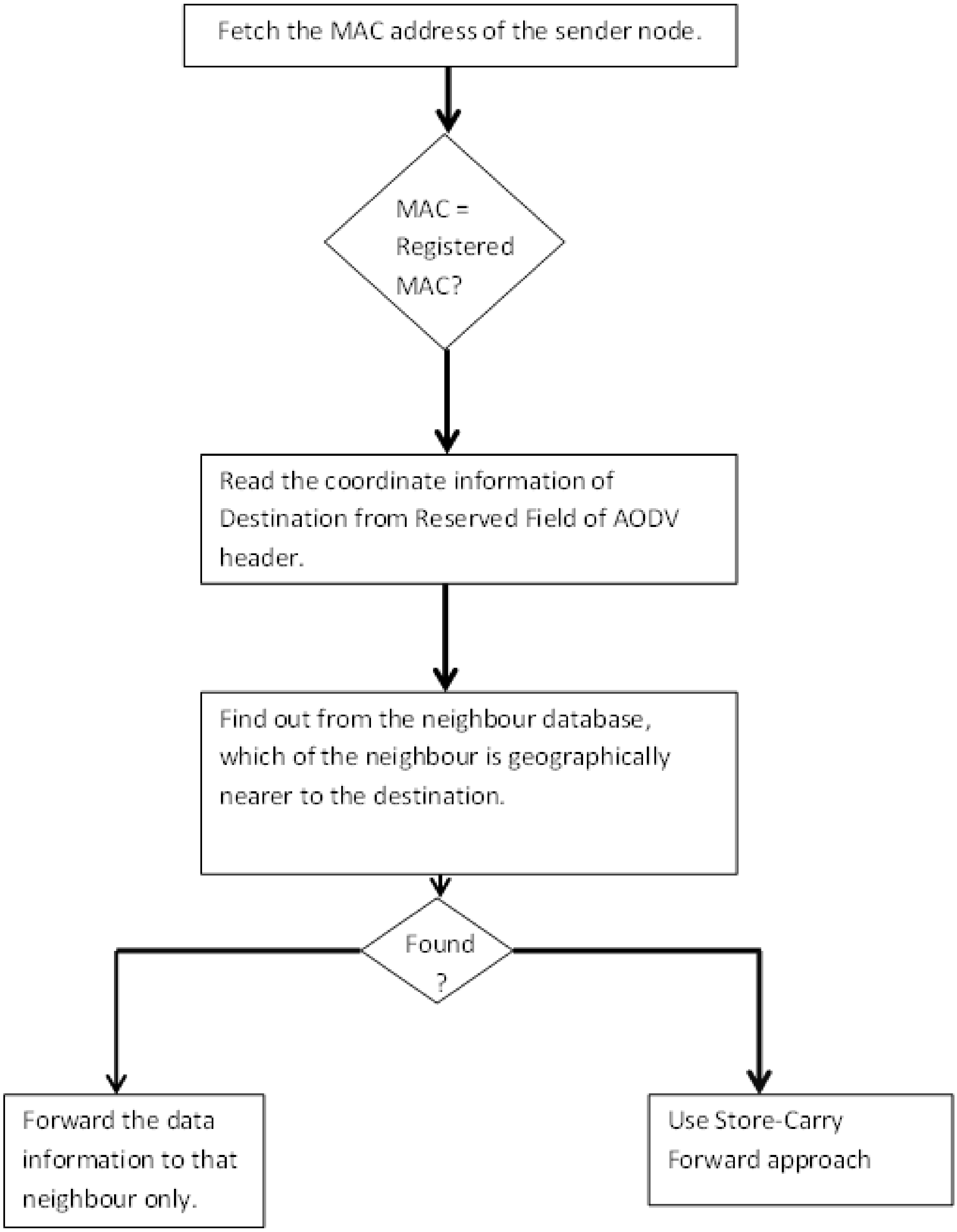

Figure 4.1: Proposed Working of AODV

* The forwarding decisions are based on the calculation of advancement that may be provided by the neighbor towards the destination.
* In case no neighbors are found providing advancement the store-carry-forward approach of the Delay Tolerant Networks should be used if it is a vehicle node.

4.2 Elements of Simulation Environment

The simulation environment for implementing the modified AODV will consist of:

a. Fedora - 14
b. MOVE [12]
c. java sdk 1.6
d. SUMO version 0.13.1
e. NS-2 Version 2.34
f. XML parser
g. FOX toolkit (GUI toolkit)
h. PROJ -(Cartographic Projection Library)
i. GDAL - (Geospatial Data Abstraction Library)

The working, installation and simulation scenarios of MOVE, SUMO and NS-2 can be referred from Appendix B

4.3 Results

Figures 4.2, 4.3 and 4.4 shows the comparison graphs of RREQ overhead v/s number of nodes for original AODV and modified AODV for node speeds 20 mps, 30 mps and 40 mps respectively. These simulations were executed for 100 seconds.

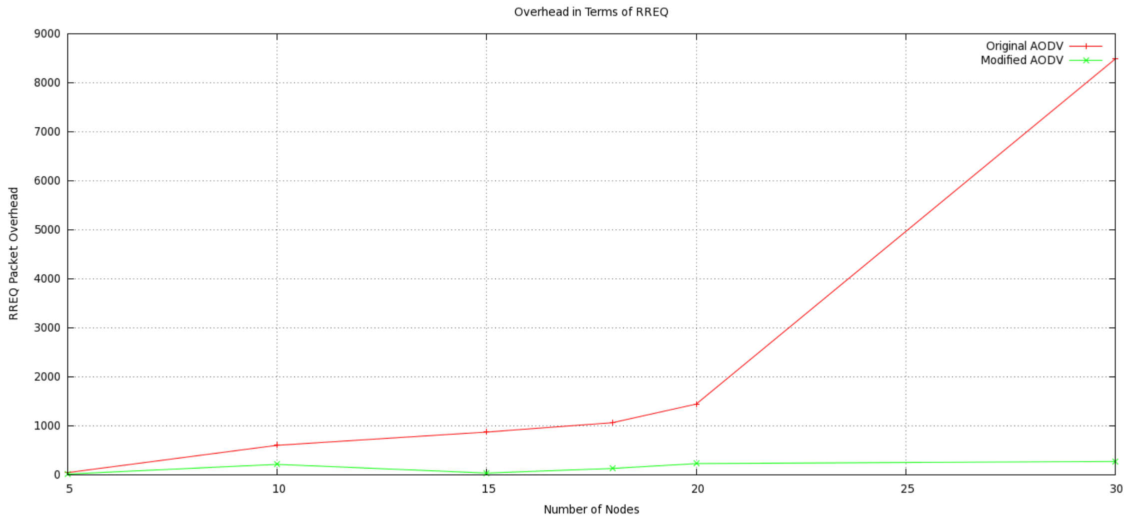

Figure 4.2: Speed 20 mps and simulation time 100 sec

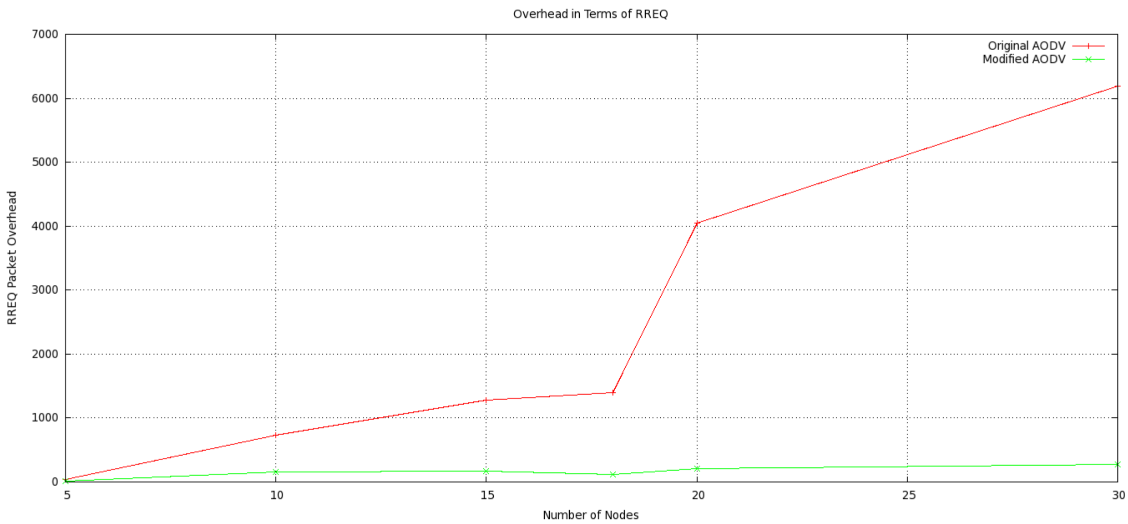

Figure 4.3: Speed 30 mps and simulation time 100 sec

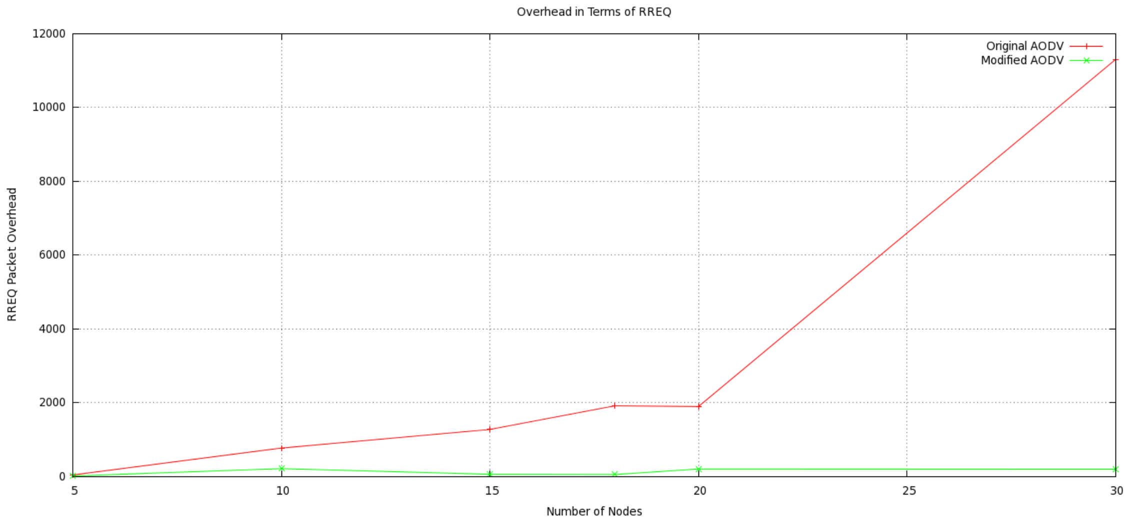

Figure 4.4: Speed 40 mps and simulation time 100 sec

From the results it can be seen that the RREQ overhead decreases significantly when modified AODV is implemented. Figures 4.5, 4.6 and **??** shows the comparison graphs of RREQ overhead v/s number of nodes for original AODV and modified AODV for node speeds 20 mps, 30 mps and 40 mps respectively. These simulations were for 1000 sec.

From the results of experiments with simulation time 1000 sec it is proved that the RREQ overhead decreases significantly when modified AODV is used.

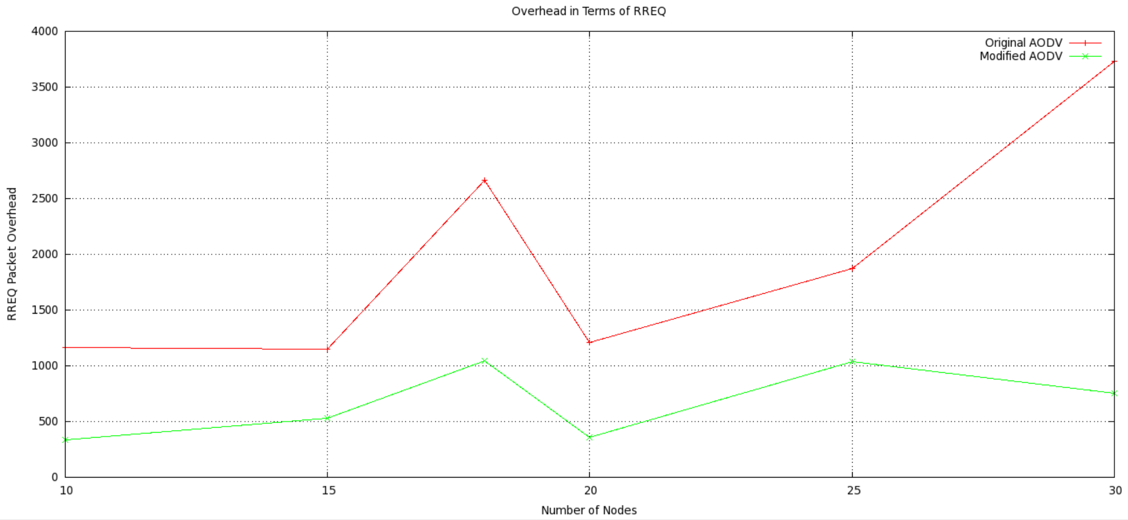

Figure 4.5: Speed 20 mps and simulation time 1000 sec

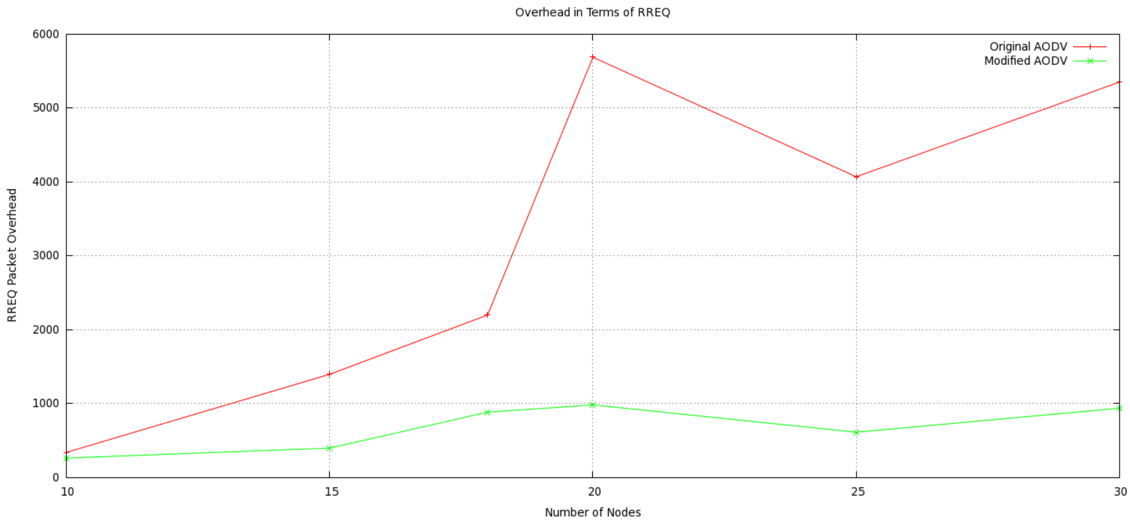

Figure 4.6: Speed 30 mps and simulation time 1000 sec

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Vehicle to Infrastructure communication helps in deployment of numerous VANET applications. For the application under consideration, i.e. Z-Lane, to function well the RSUs are responsible for sending the alert messages to the vehicles using the road. On getting the alert, these vehicles can in turn vacant the lane before the ambulance arrives. But in certain scenarios the RSU may not be there in the range of on going message transmission. In absence of RSUs vehicle to vehicle communication can be used to deliver the message to the nearest RSU. Experiments were performed for V2V communication using AODV, DSDV, DSR and AOMDV for varying number of nodes and varying node speeds.For the Z-lane to function well, as per the conclusion stated in section 3.2.5 AODV protocol is chosen for the V2V communication.

As per the proposed modification in AODV mentioned in figure 4.1 the purpose of delivering the needed information to the nearest infrastructure node as early as possible should be served. From the results mentioned in section 4.3 it can be concluded that the proposed modification in AODV helps in reducing the RREQ overhead significantly. This in turn reduces the number of packets to be processed by each node and thus reduces the end to end delay. Thus the architecture of V2I communication has been optimized. As the experimental outcomes are scenario dependent, the conclusions stated here are for the scenarios considered and not generic.

5.2 Future Work

The work done has a scope of extension in terms of increasing the efficiency by partitioning the network and then following the scheme in case road segment has no RSU for a much larger distance. In the implementation done in this project only one single neighbor is selected for forwarding the message. This can eventually create a single point of failure for the application. Instead of choosing a single neighbor the algorithm can be further modified for selecting a small set of neighbors.

Appendix A

Vehicular Mobility Modeling

This Appendix describes the concepts which are not emphasized in the said research work but should be known to better understand the domain under consideration.

A.1 Building Blocks for VMM

For generating realistic mobility model the following building blocks are to be considered:

(a) Accurate and realistic topological maps: street topologies should manage different densities of intersections, contain multiple lanes, different categories of streets and their associated speed limitations.

(b) Obstacles: obstacles should be understood in a wide sense, as both constraints to cars mobility and hurdles to wireless communications.

(c) Attraction/repulsion points: initial and final destinations of road trips are not random. Most of the time, drivers are moving to similar final destinations, called attraction points (e.g. office), or from similar initial locations, called repulsion points (e.g. home), a feature that creates bottlenecks.

(d) Vehicles characteristics: each category of vehicle has its own characteristics, which has an impact on a set of traffic parameters. For example, macroscopically speaking, some urban streets and highways are forbidden to trucks

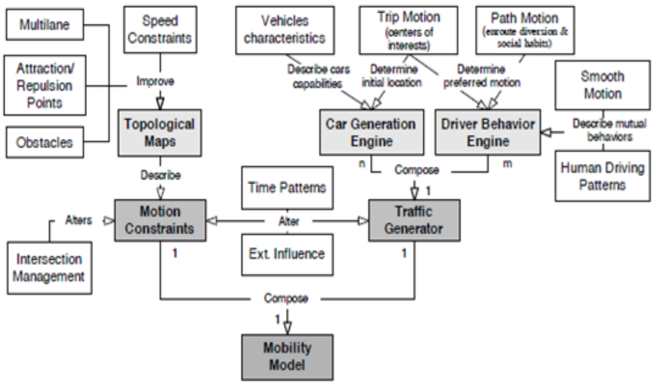

Figure A.1: Building Blocks of VMM [8]

depending on the time of the day. Microscopically speaking, acceleration, deceleration and speed capabilities of a car or a truck are different. Accounting for these characteristics alters the traffic generator engine when modeling realistic vehicular motions.

(e) Trip motion: a trip is macroscopically seen as a set of source and destination points in the urban area. Different drivers may have diverse interests, which affect its trip selection.

(f) Path motion: a path is macroscopically seen as the set of road segments taken by a car on its trip between an initial and a destination point. As it may also be observed in real life, drivers do not randomly choose the next heading when reaching an intersection, as it is currently the case in most vehicular networking traffic simulations. Instead, they choose their paths according to a set of constraints such as speed limitations, time of the day, road congestion, distance, and even drivers personal habits.

(g) Smooth deceleration and acceleration: vehicles do not abruptly break and accelerate. Models for decelerations and accelerations should consequently be considered.

(h) Human driving patterns: drivers interact with their environments with respect to static as well as dynamic obstacles such as neighboring cars and pedestrians. Accordingly, the mobility model should control the mutual interactions between vehicles, such as traffic jam, overtaking and preferred paths.

(i) Intersection Management: It corresponds to the process of controlling an intersection, and may either be modeled as a static obstacle (stop signs), a conditional obstacle (yield sign), or a time-dependent obstacle (traffic lights). It is a key part in this framework that however only influences the Motion Constraint block, as the Traffic Generator block cannot not see the difference between a stop sign or a high density traffic. Both are interpreted as a motion constraint.

(j) Time patterns: traffic density is not identical during the day. A heterogeneous traffic density is always observed at peak times, such as rush hours or during special events. This block influences the Motion Constrains and the Traffic Generator blocks, as it may alter the trip or path computation, and also the attraction/repulsion points.

(k) External Influence: some motion patterns cannot be proactively configured by vehicular mobility models as they are externally influenced. This category models the impact of accidents, temporary road works, or real-time knowledge of the traffic status on the motion constraints and the traffic generator blocks. Communication systems are the primary source of information about external influence.

A.2 Stochastic Models

It consist of all those mobility descriptions that constrain random moments of vehicles on a graph. They represent the most intuitive way to describe a vehicular mobility when no particular requirements in terms of realism are to be met.

A.2.1 City Section model

It constrains cars movement on a grid-shaped road topology, in which all edges are considered to be bidirectional, single-lane roads. Vehicles randomly select one of the intersections of the grid as their destination and move towards it

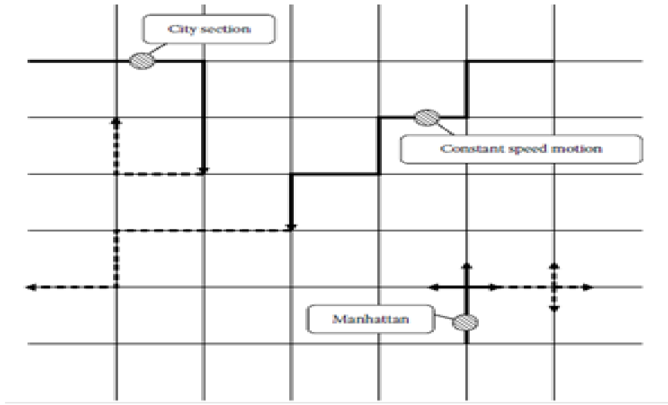

Figure A.2: City Section Model [1]

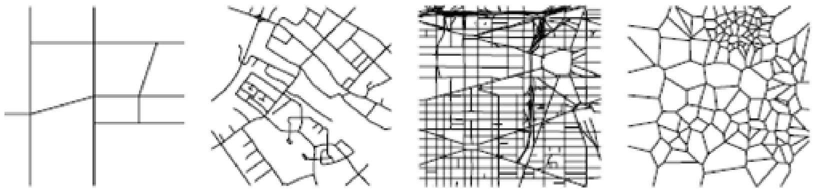

Figure A.3: City Section Model [1]

at constant speed, with (at most) one horizontal and one vertical movement, as depicted in Figure. The speed depends on the road on which the vehicle is moving. High-speed and low-speed are the two allowed road classes. According to road class, each vehicle sets its speed to a high or low value.

A.2.2 Constant Speed Motion model

It describes a random vehicular movement on a graph, representing the road topology. No particular constraint is forced on the graph nature, so that it can embody different levels of realism. Examples of graphs that can be employed with the Constant Speed Motion model are shown in Figure.

A car's motion is structured in trips, that is, movements between vertices of the graph, referred to as destinations and randomly selected.

At the beginning of each trip vehicle i chooses its next destination. Then com-

putes the route to it using shortest path algorithm on the graph, with link costs biased by parameters such as the road length, traffic congestion, speed limits etc as in Figure .
It then sets its speed to

$$v_i = v_{min} + \eta(v_{max} - v_{min}) \tag{A.1}$$

where η is a uniformly distributed random variable in [0, 1].
Such a speed value v_i is selected once at the beginning of each trip and kept constant until the destination is reached.

A.2.3 Saha Model

Represents vehicular traffic as a random mobility of cars over real road topologies extracted from the maps of the U.S. Census Bureau TIGER database. Vehicles select one point over the graph representing the map as their destination and compute the shortest path to get there. The sequence of edges is obtained by weighting the cost of traveling on each road on its speed limit and on the number of vehicles already moving on it. This helps to reproduce the real-world drivers' behavior to avoid congested paths.
A mobile entity's speed is set to a constant value in the range $[v_{max} - \varepsilon, v_{max} + \varepsilon]$, where v_{max} is the speed limit of the road on which the car is moving. All roads are considered bidirectional and single lane. No car-to-car interaction is modeled.
This model can be regarded as a typical case of the Constant Speed Motion model.

A.2.4 Freeway Model

Designed for road topology graphs representing non-communicating, bidirectional, multilane freeways traversing the entire simulated area. The movement of each vehicle is restricted to the lane it is moving on.
The following speed management rules apply to vehicle i:

a Speed update. The speed is varied by a random acceleration of maximum magnitude a. If we define as η a random variable uniformly distributed in [-1, 1], then this rule can be expressed as $vi(t + \delta t) = vi(t) + \eta a \delta t$

b Speed bounding. At any time, the speed of a vehicle cannot be lower than a minimum value v_{min} and cannot exceed a maximum value v_{max}. This constraint is enforced as $v_i(t + \delta t) = minmax[v_i(t + \delta t), v_{min}], v_{max}$

c Speed reduction. In order to avoid overlapping, that is, a collision situation, with the front vehicle, a minimum safety distance must be maintained. Formally

$$v_i(t + \Delta t) = v_{i+1}(t) - \frac{a}{2} if x_i(t) < \Delta x_{safe} \tag{A.2}$$

$$= v_i(t + \Delta t) otherwise \tag{A.3}$$

Each vehicle starts its movement at one end of a lane, with a speed that is at first selected as uniformly distributed in an interval $[v_{min}, v_{max}]$, and ends it when it reaches the other extremity of the same lane. Then a new movement, on a randomly selected lane, is started over.

A.2.5 Manhattan mobility model

It employs the same speed management rules as the Freeway model. It extends the free way model to an urban scenario. At each intersection, a vehicle chooses to keep traveling in the same direction with probability 1/2. A vehicle may turn left or right with probability 1/4 in each case. This approach thus abandons the concept of trip, in favor of an intersection by intersection decision on the route of a vehicle.

A.3 Traffic Stream Models

This model takes vehicular mobility as a hydrodynamic phenomenon and try to relate the three fundamental variables of density p (x, t) (measurable in vehicles/km), velocity v(x, t) (measurable in km/h) and flow q(x, t) (measurable in vehicles/h). All of these are functions of space x and time t, averaged over sufficiently large regions. They fall into the category of macroscopic model.

The basic equation for traffic stream models comes from the idea that, given a road section, the number of vehicles on the section can only vary due to cars entering or leaving the section. This leads to the following continuity equation (capable of modeling kinematic waves)

$$\frac{\partial p}{\partial q} = -\frac{\partial q}{\partial x} = \frac{\partial \rho v}{\partial x} \tag{A.4}$$

Given their macroscopic nature, traffic stream models can handle large quantities of vehicles, at the cost of precision. This makes them appropriate for analytical studies of traffic behavior. However, traditional traffic stream models cannot reproduce the independent motion of each vehicle, a fundamental aspect to account for in vehicular networking research, where it dramatically affects key communication factors such as network connectivity and link duration.

Fluid Traffic Motion (FTM) model: It's an exception. It applies a traffic stream approach to individual mobile entities, exploiting macroscopic metrics on a microscopic scale, and thus gene-rating a mesoscopic description. It computes the speed of each car as a monotonically decreasing function of vehicular density, forcing a lower bound on the velocity when the traffic congestion reaches a critical state:

$$v_i(t + \Delta t) = \max[v_{min}, v_max(1 - \frac{\rho(x,t)}{\rho_{jam}})] \tag{A.5}$$

where $\rho(x,t)$ is the current vehicular density on the road car i is traveling on, ρ_{jam} is the vehicular density for which a traffic jam is detected. The model computes $\rho(x,t)$ as n/l. Here l is the length of the road segment and n is the number of cars on the same road of i.

According to this formulation, cars traveling on very crowded streets are forced to slow down, possibly to the minimum speed, while the speed of cars is increased

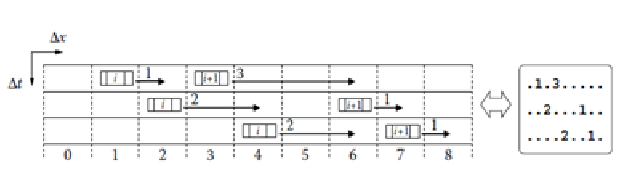

Figure A.4: CFM-Cellular Autometa [1]

towards the maximum value when less congested roads are encountered.

A.4 Car Following Model based on Cellular Automata

Cellular automata models discrete time and space, which is fragmented into cells, each of which can host a single mobile entity at a time. Moreover, the possible states of each vehicle, that is, their instantaneous speed, must be finite, leading to a discrimination of velocity as well. The movement of cars is thus described as a shift of finite states along a one-dimensional lattice of subsequent cells, as in Figure below:

A.5 Intersection Management

For network simulation of vehicular traffic in urban and suburban environments study of intersection management is required. In such cases, a proper modeling of the interaction among nonparallel traffic flows must be provided.

A.5.1 Pause Model

In this model all vehicles reaching a road junction stop for certain period of time, which can be stochastic or deterministic. This minimum-cost solution provides very approximate results, as (1) it is independent from the inflow rate of vehicles, and thus does not generate delays proportional to the vehicular density, as encountered in unsignalized intersections, and (2) it lacks the temporization typical of signalized intersections. This solution can be employed within any graph-constrained mobility model, no matter whether it belongs to the stochastic, traffic stream, or car-following categories.

A.5.2 Constant Speed Motion

It includes the possibility of forcing pauses at intersections, that is, graph vertices, encountered within a trip. Pauses of different cars at the same intersection are not independent, but correlated. In fact, considering a vehicle arriving at an intersection, two cases are possible: (1) if no other car is waiting at the intersection, it picks a random pause time uniformly distributed in a given range [0, Tp]. (2) If one or more other vehicles are already paused at the intersection, it forces its own pause time to match the residual pause time of the first car that arrived at the intersection under examination.
According to such rules, cars paused at a same road junction leave the intersection in the same instant, mimicking the clustering of vehicles leaving a crossroad, a typical effect in real-world traffic, due to the presence of semaphores. However, it should be noted that this solution induces improper behaviors with increasing vehicular densities. Although delays are expected to increase, the stated rules tend to reduce the average waiting time at an intersection in the presence of higher densities. As the probability of finding already stopped vehicles at crossroads grows.

A.5.3 PTSM (Probabilistic Traffic Sign Model)

Adopts a stochastic approach. It distinguishes between vehicles arriving at a clear intersection and vehicles reaching a road junction where other cars are already waiting. Thus, when a vehicle reaches an intersection: (1) If no other car is already stopped there, it directly crosses the intersection with probability p, while with probability 1 - p it picks a random pause time in [0, Tp] and waits for such a time to expire, before traversing the intersection. (2) If other cars are waiting at the intersection, it stops as well, for a time equal to the residual pause time of the previous car that reached the same intersection, plus 1 sec. PSTM also adds to the pause time 1 sec of delay for each car already stopped.
With respect to the Constant Speed Motion model, PTSM increases the level of realism of the simulation, as it models the delay as a linear function of the

density. However, it remains a stochastic solution that cannot reproduce the deterministic aspects of signalized intersection management.

A.5.4 IDM-IM (Intelligent Driver Model - Intersection Management

This model can be used to model road junctions ruled by stop signs or traffic lights, and, in both cases, IDM-IM only acts on the first vehicle approaching the intersection, as the car-following description in the IDM model automatically adapts the behavior of cars following the leading one. The basic principle is to force the leading vehicle to believe that an obstacle (i.e., a still car) is present right before the intersection ahead if the vehicle should stop. In that case, the IDM car-following model induces a deceleration in the leading vehicle, in order to avoid a collision with the imaginary obstacle, and makes it stop right before the intersection. All following cars will queue up behind the first one, without need of any further intervention.

By removing the obstacle, the possibility of crossing the intersection is granted to the leading car. As an example, this occurs if the vehicle has right of way at a stopsign- regulated intersection, or if the corresponding traffic light has turned green.

As this scheme is based on simple car-to-car interaction rules, its employment is not limited to the IDM description, but it can be adapted to any other car-following model. In the IDM-IM model, the imaginary obstacle is implemented by tweaking, in the acceleration update equations of the first vehicle on lane i, the distance and speed difference with respect to its front car. As this front car does not actually exist, the standard values observed by i would be :

$$\Delta x_i(t) = \infty \tag{A.6}$$

$$\Delta y_i(t) = 0 \tag{A.7}$$

substituting these values in equation 2.4 the deceleration term is canceled and thus gives the free flow condition. IDM-IM imposes following settings to i:

$$\Delta x_i(t) = x_{stop} - x_i(t) \tag{A.8}$$

$$\Delta v_i(t) = -v_i(t) \tag{A.9}$$

x_{stop} - position of location vehicle i has to stop at.
the settings mimic that i would observe if a still $[v_{i+1}(t) = 0]$ front vehicle were located at a distance $x_{stop} - x_i(t)$. It is also desirable that x_{stop} be placed some distance from the actual vertex representing the intersection, so that i halts before reaching the center of the intersection.
Using the formulation of the car-following model, this setting of $\Delta v_i(t)$ and $\Delta x_i(t)$ allows vehicles to freely accelerate when far from the next intersection. It also allows to smoothly decelerate as they approach the road junction. The resulting deceleration profile is not completely accurate, as braking dynamics induced by intersections are different from those due to car-to-car interactions.

Appendix B

Simulation Environment Setup

Deploying and testing VANETs involves high cost and time. Simulations of VANETs often involve large and heterogeneous scenarios. Mobility behavior of node in VANET significantly affects simulation results. In a vehicular network, nodes (vehicles) can only move along streets, prompting the need for a road model. Nodes in VANETs do not move independently of each other. They move according to well established vehicular traffic models.

VANET mobility generators are used to generate traces of the vehicle's motion that can be usually saved and subsequently imported into a network simulator in order to study the performances of the protocol/application. It is important to generate realistic movement traces in order to thoroughly evaluate VANET protocols because in general performances depend on the vehicles movement traces. The inputs of the mobility generator include the road model, scenario parameters like maximum vehicular speed, vehicle arrivals and departure rate etc.. The output of the trace contains the location of each vehicle at every time instant for the entire simulation time and their mobility profiles. Examples are SUMO, MOVE, FreeSim and VanetMobiSim.

Network simulators allow researchers to study how the network would behave under different circumstances. Users can then customize the simulator to fulfill their analysis needs. Network simulators are relatively fast and inexpensive compared to cost and time involved in setting up an entire experiment containing multiple networked computers, routers and data links. They allow researchers to test scenarios that might be difficult or costly to emulate with real hardware, especially in VANETs. Network simulators perform detailed packet level simulation of source, destinations, data traffic transmission, reception, route, links, and channels. Network simulators are particularly useful to test new networking protocols or to propose modifications on it. Examples are NS-2, GloMoSim and JiST/SWANS. Most existing network simulators are developed for MANETs and hence require VANET extensions before they can be used to simulate vehicular networks.

Simulation is therefore, the most common approach to developing or testing new protocol for a VANET. Choosing a right simulation tools has been a key step to get accurate prediction of real world environment. This chapter covers the details and installation steps for MOVE, SUMO and NS-2.

B.1 Mobility Simulators

Mobility simulators classification spreads from sub-microscopic to macroscopic depending on the level of detail of the simulation. This is reflected on the smallest entity considered by the simulator. Macroscopic simulators consider the whole traffic flow as the basic entity.

On the other hand, microscopic simulation considers the vehicle the smallest simulation unit. There are simulators which are in-between macroscopic and microscopic, referred as mesoscopic. The latter consider individual vehicles moving between queues, which are the main simulated entity.

There are also sub-microscopic simulators which consider not only each vehicle, but also the components of them, as the engine or the gear-box, and their parameters. The different granularities are represented in Fig. B.1. From left to right: macroscopic, microscopic, submicroscopic (within the circle: mesoscopic)

For VANET simulations, where every individual vehicle will be considered a node and the simulation of the vehicle components and their status are not relevant, the most adequate approach to mobility simulation is microscopic. It provides

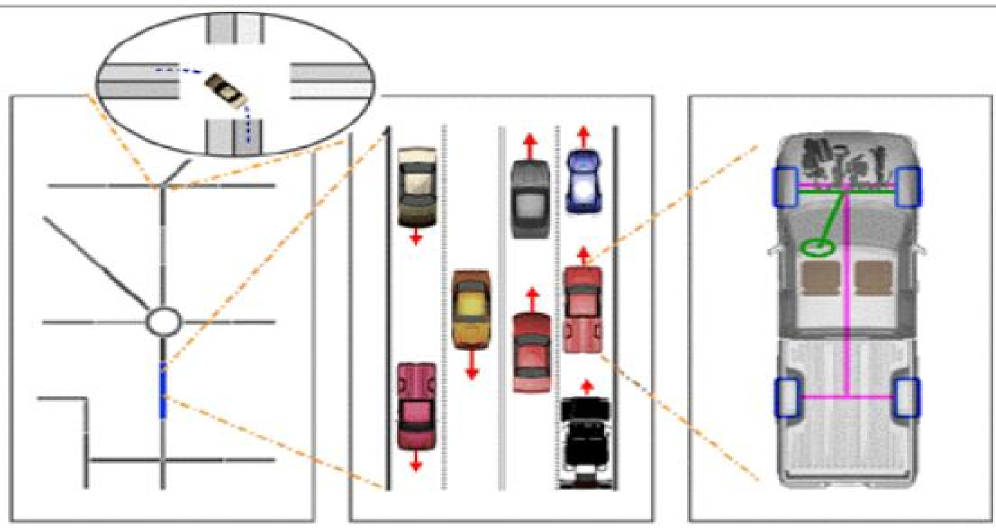

Figure B.1: Mobility simulator categories [8]

enough resolution of the system as to provide realistic traces, but without the overload of simulating sub-microscopic details which would not provide relevant information for this research.

B.1.1 Simulation of Urban MObility

"Simulation of Urban MObility (SUMO)", is an open source, microscopic, multi-modal traffic simulation. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic i.e., each vehicle is modeled explicitly, has an own route, and moves individually through the network.

It incorporates realistic traffic simulation algorithms, with the possibility to have different types of vehicles, different networks -from generated grid, spider-web or random artificial roads to imported VISUM, Tiger, OSM, etc. models, and has a high speed performance.

B.1.2 MObility model generator for VEhicular networks(MOVE)

MOVE is built upon an open source micro-traffic simulator SUMO [15]. Simulation of Urban MObility (SUMO) is a microscopic, space continuous and time discrete traffic simulator written in C++ capable to provide accurate and realistic mobility patterns. The project started as an open source project in 2001 with the goal to support the traffic research community with a common platform to test and compare models of vehicle behaviour, traffic light optimization, routing etc.

MOVE [17] is an extension to SUMO that adds a GUI for describing maps, defining vehicle movement and allows the user to import real world map databases such as TIGER [16] and Google Earth. 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, by providing a set of graphical user interfaces that automate the simulation script generation, MOVE 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.

Figure B.2 shows MOVE modules and defines flow how we can create mobility and then simulate it with netwok simulators. Users input information of Map Editor and Vehicle Movement Editor is then fed into SUMO to generate a mobility trace which can be immediately used by a simulation tool such as ns-2 or qualnet to simulate realistic vehicle movements. Users can also visualize the generated mobility trace with SUMO. MOVE consists of two main components: Map Editor and Vehicle Movement Editor.

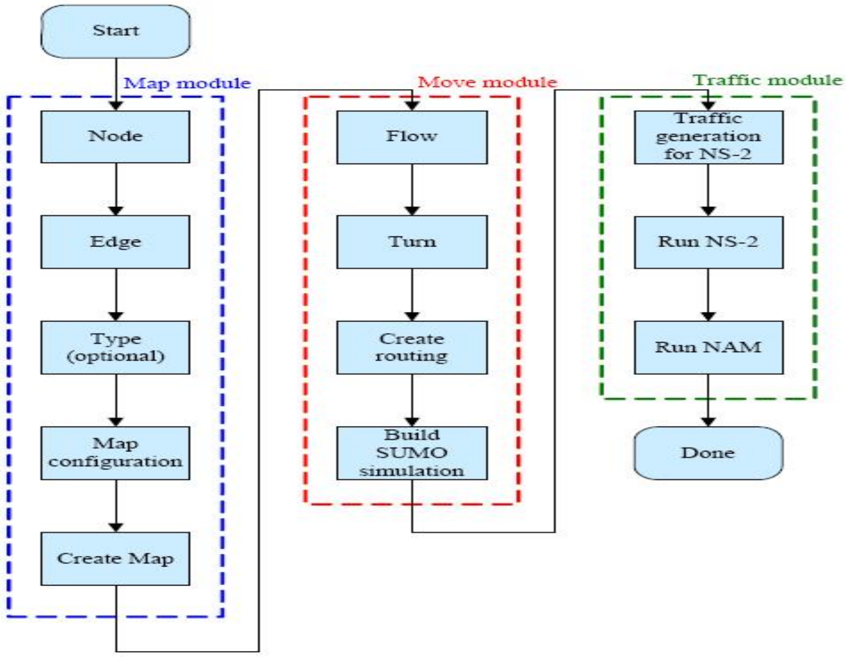

Figure B.2: Modules of MOVE [17]

B.2 Network Simulators

B.2.1 NS-2

NS-2 in its different versions is one of the most popular simulation environments for research. It has a hybrid programming simulations with both an object-oriented version of Tcl scripting called OTcl and C++. This duality can lead to confusion when not familiar with the system, but it proves to be very convenient once the user becomes acquainted with it. The modules are developed using C++, in order to provide higher simulation speeds by the use of compiled code.

C++ modules are configured and executed via OTcl scripts, which provide the description of the simulation environment and the configuration parameters for each module involved. This OTcl scripts are not compiled but interpreted by the ns software. This makes the set-up of simulations very easy and convenient to batch, as there is no compilation needed to run the scripts, and these contain all the required configuration parameters for the C++ modules.

This duality becomes critical when it comes to develop or modify modules. The modules have two parts: one programmed using C++ and other OTcl. This is required to provide the usability features previously mentioned.

There is an All-in-One package available for most of the releases. These versions include the network simulator, network animator –NAM– and xGraph in the latest version available at the moment of the creation of the package. The installation is not quite straightforward if you are not using one of the systems supported out-of-the-box for that version, but is easy to find community-developed scripts to compile and install the software properly.The installation of extra modules may require additions and modifications in the configuration files in order to work, being usually simple and well documented.

There is an extensive documentation for the network simulator [18] and its modules, in addition to *.tcl example files provided in the distribution in order to both validate the installation of the simulator and learn how to script for the different areas of application of the simulator.

The disadvantage of ns-2 is the limited scalability of number of nodes being simulated, which is not a fixed limit, but it depends on the simulation parameters. This fact is related with the lack of memory management of ns-2: it may require multiple times the amount of memory than some of its alternatives for similar simulations [19]. This is in part a consequence of the use of interpreted software (OTcl), which in 1989 when the ns project was born was a very convenient method to improve the simulation work-flow. However, at present, when the compilation process is not time-consuming, it is considered an unnecessary legacy burden when conducting large simulations.

Another important disadvantage has already been introduced when stated that ns-2 is in its different versions the most used software: not all modules are updated and valid for all the versions. Outdated versions lack general improvements and patches on different parts of the software[20] which may influence the simulation results and their validity.

B.3 Installation steps for MOVE

To install MOVE using the latest version of SUMO, following additional softwares need to be installed:

* Linux (Fedora 14)
* Java SDK 1.6 - http://java.sun.com
* SUMO version: 0.13.1 - http://sumo.sourceforge.net
  - Xerces (XML-parser) - http://xerces.apache.org/xerces-c/index.html
  - FOX-Toolkit (GUI Toolkit) - http://www.fox-toolkit.org/
  - PROJ (Cartographic Projections Library): http://www.remotesensing.org/proj/
  - GDAL (Geospatial Data Abstraction Library): http://www.remotesensing.org/gda
* NS2 version: 2.34 (all-in-one) or later - http://www.isi.edu/nsnam/ns/

B.3.1 Installation steps for SUMO

Prerequisites for SUMO:

* libtiff-devel
* libpng-devel
* libjpeg-devel
* libXft-devel
* zlib-devel
* bzip2-devel
* mesa-libGLU-devel
* mesa-libGL-devel
* glibc-devel

Prerequisites for gdal and proj4:

* gcc-c++
* gcc
* libpng
* libtiff

SUMO installation succeeds with the help of RPM files that are available for Fedora 14.

SUMO installation process:

1 Download all the RPM files: Here is the following link for downloading all the rpm files related to SUMO:
`http://download.opensuse.org/repositories/home:/behrisch/Fedora_14/i386`

2 Install all the RPM files downloaded in following sequences using command rpm -ivh packagename :
   (1) `libfox1_6-1.6.43-22.1.i386.rpm`
   (2) `fox16-1.6.43-22.1.i386.rpm`
   (3) `fox16-devel-1.6.43-22.1.i386.rpm`
   (4) `fox16-devel-static-1.6.43-22.1.i386.rpm`
   (5) `fox16-example-apps-1.6.43-22.1.i386.rpm`
   (6) `libt4k_common0-0.1.1-10.1.i386.rpm`
   (7) `t4k_common-0.1.1-10.1.i386.rpm`
   (8) `t4k_common-devel-0.1.1-10.1.i386.rpm`
   (9) `tuxmath-2.0.3-21.3.i386.rpm`
   (10) `sumo-0.13.0-7.4.i386.rpm`
   (11) `sumo-svn-836.1.i386.rpm`

3 Install all the missing files using "yum install" which are needed while running rpms.

Final package versions used in simulation environment:

* FOX Toolkit: 1.6.43
* GDAL: 1.7.3-9
* PROJ: 4.7.0-3
* XERCES: 3.0.1-20
* SUMO: 0.13.1

Steps for building and installing source of SUMO:

```
./configure --with-fox=$HOME --with-proj-gdal=$HOME
        --with-xerces=$HOME --prefix=$HOME
make
make install
```

As a result the source of SUMO was builded and installed successfully.

B.3.2 Installation steps for NS-2

a. su , password

b. `yum install libX11-devel libXext-devel libXau-devel libXmu-devel`

c. exit

d. tar xvf ns-allinone-2.34.tar.gz

e. cd ns-allinone-2.34

f. ./install

g. cd

h. cd root

i. `export PATH=/NS2/ns-allinone-2.34/bin: /home/nikunj/NS2/ns-allinone-2.34/tcl8.4.18/unix: /home/nikunj/NS2/ns-allinone-2.34/tk8.4.18/unix`

j. `export LD_LIBRARY_PATH= /NS2/ns-allinone-2.34/otcl-1.13, /home/nikunj/NS2/ns-allinone-2.34/lib`

k. `export TCL_LIBRARY=/NS2/ns-allinone-2.34 /tcl8.4.18/library`

l. `cd ns-2.34; ./validate`

m. `Put following code in /root/.bash_profile`

```
#LD_LIBRARY_PATH
OTCL_LIB=/NS2/ns-allinone-2.34/otcl-1.13
```

```
NS2_LIB=/NS2/ns-allinone-2.34/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB

#TCL_LIBRARY
TCL_LIB=/NS2/ns-allinone-2.34/tcl8.4.18/library
export TCL_LIBRARY=$TCL_LIB

#PATH
XGRAPH=/NS2/ns-allinone-2.34/bin:/home/nikunj/NS2/ns-allinone-2.34
        /tcl8.4.18/unix:/home/nikunj/NS2/ns-allinone-2.34/tk8.4.18/unix
NS=/NS2/ns-allinone-2.34/ns-2.34/
NAM=/NS2/ns-allinone-2.34/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

B.3.3 Simulation Process

SUMO simulation can be easily generated by MOVE GUI. MOVE is a .jar source/application available from

http://lens.csie.ncku.edu.tw/Joomla_version/index.php/research-projects/pas

The files used in SUMO simulation for generating road traffic mobility pattern are:

a. <Filename>.nod.xml - Defines nodes(road junctions, terminations etc)
b. <Filename>.edge.xml - Defines edges (roads)
c. <Filename>.netc.cfg
d. <Filename>.net.xml - Generated using net-convert command from the above two files
e. <Filename>.flow.xml - Defines traffic flow definition
f. <Filename>.rou.xml - Defines vehicle routes
g. <Filename>.sumo.cfg - Simulation configuration

Running the .sumo.cfg file in the SUMO GUI one can visualize the road network and traffic simulation.

The road traffic mobility model generated by SUMO is converted to network traffic model that can be used with NS2 for further analysis and research. This is easily done using MOVE. MOVE helps in converting the .sumo.cfg file to NS2 compatible .tr (node trace file), .tcl(network traffic simulation) and .nam (for visualization) files.

Steps for Simulation Setup:

1 Define the nodes and edges in the desired road topology in the .nod.xml and .edge.xml files respectively.This can be done using MOVE GUI or by simply creating and editing a file.

2 Generate the road map i.e. .net.xml using the net-convert command or MOVE.

3 Define the vehicle flow in .flow.xml file and generate sumo.cfg file using MOVE.

4 Visualization of the road traffic configuration can be done using sumo-gui.

5 .tr, .tcl and .nam files are generated using MOVE form the road traffic simulation.

6 Necessary changes are made in the tcl file.

7 Change the AODV and application agent as specified in the proposal.

References

[1] Book - ”Vehicular Networks from Theory to Practice”, Edited By Stephan Olariu And Michele C. Weigle, CRC Press.

[2] Daniel Jiang, Luca Delgrossi, ”IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments”, 978-1-4244-1645-5/08/2008 IEEE

[3] Draft Standard for Wireless Access in Vehicular Environments (WAVE) - Remote Management Service, IEEE P802.1609.1TM/D1.2, Dec. 2009.

[4] Draft Standard for Wireless Access in Vehicular Environments (WAVE)- Security Services for Applications and Management Messages, IEEE P1609.2TM/D3.0, Jan. 2010.

[5] Draft Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services, IEEE P1609.3TM/D4.0, Feb. 2010.

[6] Draft Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-channel Operation, IEEE P1609.4TM/D4.0, Feb. 2010.

[7] Kai-Yun Ho, Po-Chun Kang, Ching-Hai Lin and Chung-Hsien Hsu,”Implementation of WAVE/DSRC Devices for Vehicular Communications”, International Symposium on Computer, Communication, Control and Automation 978-1-4244-5567-6/10/2010 IEEE.

[8] Jerome Harri, Fethi Filali, and Christian Bonnet, ”Mobility Models For Vehicular Ad Hoc Networks: A Survey And Taxonom”, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 11, NO. 4, FOURTH QUARTER 2009

[9] SUMO - Simulation of Urban Mobility, http://sumo.sourceforge.net

[10] F. Karnadi, Z. Mo, and K.-C. Lan, "Rapid Generation of Realistic Mobility Models for VANET", in Proc. IEEE Wireless Communication and Networking Conference (WCNC'07), March 2007.

[11] Vaishali Khairner and Dr. S.N. Pradhan, "Mobility Models for Vehicular Adhoc Network Simulation, International Journal of Computer Applications", No. 4, Article 2, 2010.

[12] MOVE http://www.cs.unsw.edu.au/klan/move/

[13] Omid Abedi, Mahmood Fathy, Jamshid Taghiloo, "Enhancing AODV Routing Protocol Using Mobility Parameters in VANET", 978-1-4244-1968-5/08/2008 IEEE.

[14] Ben Ding, Zehua Chen, Yan Wang, Hui Yu, "An Improved AODV Routing Protocol for VANETs",978-1-4577-1010-0/11 2011 IEEE.

[15] Liu Jiancai Chen Feng Xu Jiakai, "The Study of Routing Strategies in Vehicular Ad-hoc Networks", 978-1-4244-7555-1/10 2010 IEEE.

[16] Nur Diana Mohd. Nuri, Halabi Hasabullah, "Stretagy of Efficient Routing in VANET", 978-1-4244-6716-7/10 2010 IEEE.

[17] Pooja Rani, Nitin Sharma, Pariniyojit Kumar Singh, "Performance Comparison of VANET Routing Protocols", 978-1-4244-6252-0/11 2011 IEEE.

[18] Shikhul Islam Chawdhury, Won-II Lee, Youn-Sang Choi, Guen-Young Kee, Jae Young Pyun, "Performance Evaluation of Reactive Routing Protocols in VANET", 17th Asia Pacific Conference on Communication, 2011.

[19] Saishree Bharadwaj, Rashmi, Shylaja B.S., "Perfomance Evaluation of MANET Based Routing Protocols for VANETs in Urban Scenarios", International Conference on Network and Electronics Engineering, IACSIT Singapore, 2011.

[20] Victor Cabrera, Francisco J. Ros, Pedro M. Ruiz, "Simulation-based Study of Common Issues in VANET Routing Protocols", 978-1-4244-2517-4/09 IEEE 2009.

Index