CHAPTER 3: SIMULATION TOOLS FOR VANETS

3.1 DSRC AND STANDARDS OF VANETS

Some of the research issues and active challenges like standard protocol design, data dissemination methods, issues related to privacy and security etc., are mentioned in related work. The problems relating the current research issues of VANETs are seriously considered by researchers throughout the world and various standards are developed to make the VANETs more versatile, full proof and comfortable. We will discuss the standards and simulation tools to understand the concept of VANETs and ITS in this chapter.

3.1.1 Dedicated Short-Range Communication (DSRC)

Dedicated Short-Range Communication (DSRC) is a standard developed to provide facility to vehicular networks in North America. As we found in the earlier studies the safety-driven applications of vehicular communication have strict reliability and delay requirements, which are not satisfied by current wireless standards.

To enable communication in the vehicular environment, the DSRC standard was developed. DSRC comprises a set of communication protocols and standards specific for automotive use. Traffic related issues and accidents have been a long standing problem in most of the countries in the world. In 1991, the US Congress passed the Intermodal Surface Transportation Efficiency Act that resulted in the creation of the first generation Intelligent Transportation System [6]. The ultimate goal of the ITS program is to incorporate technology into the transportation infrastructure to improve safety.

The first generation of the DSRC system operates at 915 MHz and has a transmission rate of 0.5 Mbps. The project had limited success and was used primarily by commercial vehicles and mostly for toll collection. The second generation of DSRC standard started in 1997 when USA requested the Federal Communications Commission (FCC) for allocation of an additional 75 MHz of bandwidth. Originally, DSRC was proposed to work in the 915 MHz band, and US-FCC later allocated 75 MHz of spectrum at 5.9 GHz for DSRC in 1999. Similar activities also undergo in Japan and Europe, where 5.8 GHz band is used for DSRC instead. Since the current Wi-Fi standards do not satisfy the DSRC requirements, a new draft amendment to the IEEE 802.11 standard was suggested, named as IEEE 802.11p. Furthermore, It is expected that DSRC must have a low cost and be very scalable. Also, DSRC should not require any license fee from the users to access the network.

3.2 WAVE PROTOCOL STACK

Recently, both industry and academia have been extensively working on standardization of DSRC. IEEE 1609 and IEEE 802.11p are developed respectively by IEEE working group 1609 and IEEE task group p. These standards together are known as Wireless Access in Vehicular Environments (WAVE). The basic goal of the WAVE is to promote wireless communication in vehicular networks. IEEE 802.11p, being part of the DSRC system, operates in the licensed 5.9 GHz band. The changes at the IEEE 802.11 layer require providing new security and channel selection services at a higher layer and specifically in the control channel and multiple service channels. These are enabled, in addition to other network services, in the IEEE 1609 standards for WAVE.

1. IEEE 1609.1: Describes resource manager specification
2. IEEE 1609.2: Defines the formats and processing of secure messages
3. IEEE 1609.3: Covers network and transport layer services
4. IEEE 1609.4: Specifies improvement to the IEEE 802.11p MAC to support multichannel operation.

These standards facilitate a system in which vehicles can connect instantaneously on a common channel in the DSRC band for safety applications, while other channels are available for less critical communications. Fig. 3.1 shows the communication protocol stack for WAVE. Actually, it is a suite of standards that are focused on MAC and upper layers. It is a multichannel wireless standard based on IEEE802.11a PHY standard and IEEE802.11 MAC standard (i. e. WAVE uses CSMA/CA as basic medium access mechanism). WAVE uses one control channel to set up transmissions and data channels to send the data. WAVE allows high data rate (≤ 27 Mbps) in short distances (≤ 1 Km) communication.

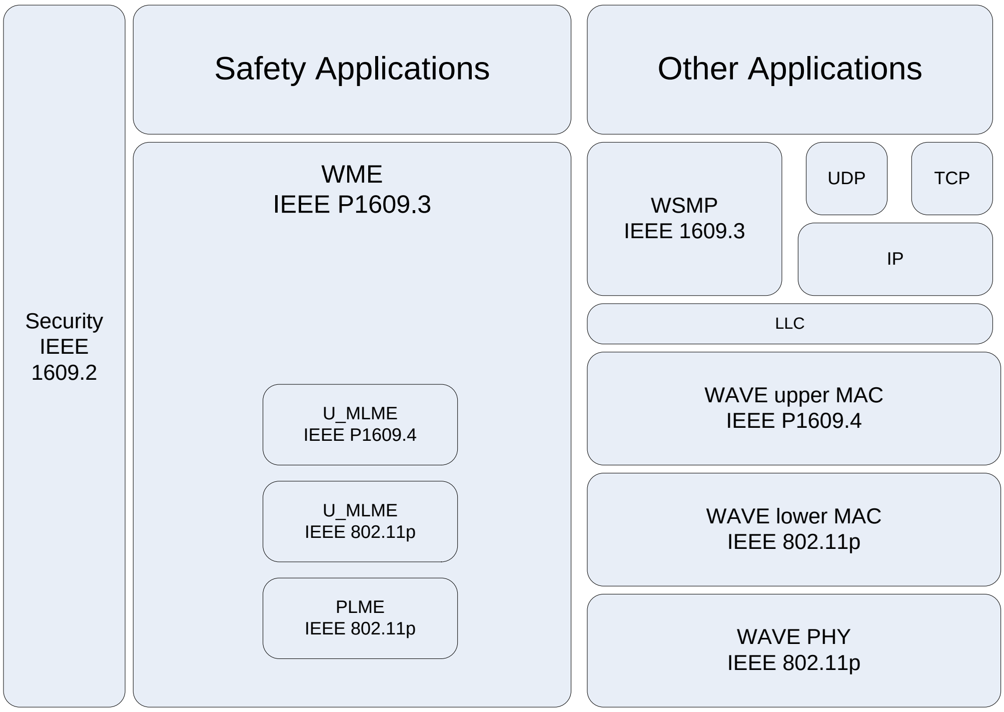

Fig. 3.1: WAVE protocol stack

3.2.1 Physical and MAC layer

IEEE 1609 is the family of standards for WAVE. It provides services and interfaces for V2V and V2I wireless communications. The physical layer and MAC layer of WAVE are functioning similar to OSI standard. The physical layer of IEEE 802.11p consists of seven channels at 5.9 GHz band, which is similar to IEEE 802.11a design, but the main difference is that the IEEE 802.11p uses 10 MHz bandwidth for each channel instead of 20MHz bandwidth in IEEE 802.11a. One of the specifications of IEEE 802.11p is that the management functions are connected to the physical and MAC layers called Physical Layer Management Entity (PLME) and MAC Layer Management Entity (MLME), respectively. The IEEE 802.11p uses CSMA/CA to reduce collisions and provide fair access to the channel.

3.2.2 Multi-channel Operation

IEEE 1609.4 being a part of the IEEE 1609 protocol family, manages channel coordination and supports MAC service data unit delivery. It describes 7 different channels with different features and usages (6 service channels and 1 control channel). In addition to this, the channels use different frequencies and transmission powers. It is mentioned that each station continuously alternates between the control channel and one of the service channels; however, the different channels cannot be used at the same time [51]. The IEEE 802.11p MAC layer is based on multichannel operation of WAVE standard and 802.11e, EDCA (Enhanced Distributed Channel Access).

3.2.3 Logical Link Layer

Logical Link Control (LLC) is another element of WAVE structure, which is similar to upper sub-layer of OSI layer 2. LLC provides the communication between upper layers and the lower layer.

3.2.4 Network and Transport Layers

The IEEE 1609.3 defines the operation of services at network and transport layers. Moreover, it provides wireless connectivity between vehicles, and vehicles to roadside devices. The functions of the WAVE network services can be separated into two sets:

i) Data-plane services: They transmit network traffic and supports IPV6 and WSMP protocols. WAVE short-message Protocol network (WSMP) provides this capability that applications can send short message to increase the probability of receiving the messages in time.

ii) Management-plane services: Their functions are to configure and maintain system, for instance: IPV6 configuration, channel usage monitoring, and application registration. This service is known as WAVE management entity (WME).

3.2.5 Resource Manager

IEEE 1609.1 standard defines a WAVE application known as Resource Manager (RM) which allows communication between applications running on Roadside units (RSU) and On-board units (OBU). The Resource Manager resides on either OBUs or RSUs. It defines command message formats and data storage formats. It specifies the types of devices that may be supported by OBU.

3.2.6 Security Services

The IEEE 1609.2 standard defines security services for the WAVE architecture and the applications which run through this architecture. This standard defines the format and the processing of secure messages.

3.3 IEEE 802.11P PROTOCOL

IEEE ratified 802.11p in 2010 as WAVE Amendment defining the PHY and MAC layer for VANETs operating in 5.9 GHz Band [52]. It is the proposed standard of DSRC in the vehicular environment by the IEEE organization.

The design requirements of the standard are the following:

- The transmission range of operation should be large: approximately 1000m.
- The relative speed between nodes should be high: up to 500km/h. (Doppler effect should also be considered).
- Extreme multi-path environment.
- The standard should work in overlapping ad-hoc networks.
- Quality of service (QoS) should be high.
- It should support applications such as safety message broadcasting in vehicular environment.

The implementation of IEEE 802.11p is much comprehensible considering the other amendments of the IEEE 802.11 protocol.

3.3.1 802.11p Physical layer

The physical layer of the IEEE802.11p protocol is based on that of IEEE 802.11a: similar orthogonal frequency-division multiplexing (OFDM) based modulation is used and the frequency allocation is at the 5GHz band.

The FCC (Federal Communication Commission) allocated 75 MHz of the radio spectrum for DSRC. The 5.9 GHz DSRC spectrum is composed of six service channels which are each 10MHz. Also, one control channel is provided by the DSRC standard, which is also 10MHz. Fig. 3.2 provides the channel layout for DSRC. The data rates possible for a 10 MHz channels are 6, 9, 12, 18, 24, and 27 Mbps with a preamble of 3 Mbps. The modulation scheme used by DSRC is OFDM and the subcarrier frequency spacing of 802.11a is double that of DSRC. In addition, DSRC doubles the guard period in comparison to 802.11a.

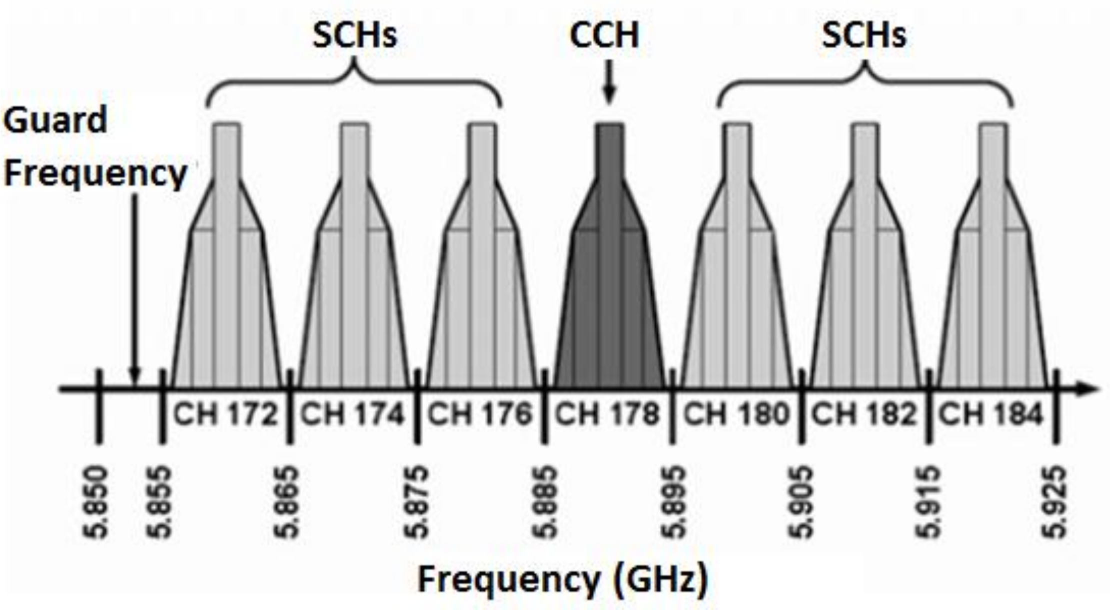

Fig. 3.2: Channel layout

Control channel is the most important channel of DSRC, and the efficient usage of this channel is critical. Each OBU monitors the control channel for both broadcast safety messages and service channel announcements. The control is monitored by each vehicle and RSU. Since there is a limited amount of bandwidth available, communication on the control channel is brief. The vehicles must periodically switch to the control channel to receive safety messages. A requirement of DSRC is that all vehicles must switch to the control channel every 100 ms and remain on the channel for a minimum period of time. The purpose of vehicle switching to the control channel every 100 ms is to allow the reception of the safety broadcast from the surrounding vehicles. To guarantee that safety messages are not sent before the vehicles switch to the control channel, the time that the vehicles switch to the channel must be synchronized.

3.3.2 802.11p MAC layer

Its latest version was approved by IEEE on 17 June, 2010. The IEEE 802.11p MAC layer has the same core mechanism of EDCA, as specified in IEEE 802.11e, which is an enhanced version of the basic Distributed Coordination Function (DCF) from 802.11. EDCA uses the

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme. EDCA provides differentiated and distributed channel access by assigning to traffic belonging to each of four Access Categories (ACs) a set of distinct channel access parameters, including AIFS (Arbitration Inter-Frame Space) and contention window size (CWmin and CWmax) as shown in Fig. 3.3.

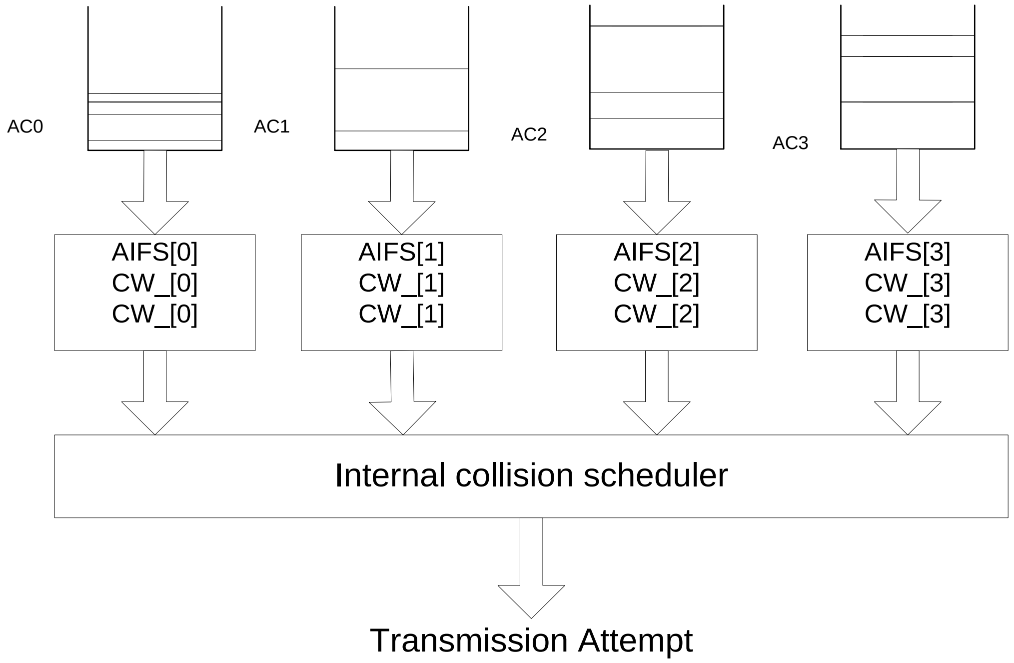

Fig. 3.3: Four access categories (ACs) [53,54]

In MAC layer, EDCA adds quality of service to IEEE 802.11 networks. Messages are categorized into four different ACs (AC0, AC1, AC2 and AC3), where AC0 has the lowest priority and AC3 has the highest priority. In order to ensure that highly relevant safety messages can be exchanged timely and reliably, even when operating in a dense scenario, the 802.11p MAC protocol accounts for the priority of the messages using different Access Categories (ACs). There are four available data traffic categories with different priorities: Background traffic (BK or AC0), Best Effort traffic (BE or AC1), Video traffic (VI or AC2) and Voice traffic (VO or AC3) as given in Table 3.1.

Table 3.1 Access Categories

Access Categories	Description
AC_BK (Background)	Background (uninvited) traffic will come under AC_BK
AC_BE (Best Effort)	Best effort (E-mail) and video belong to AC_BE
AC_VI (Video)	Video and video/controlled load belong to AC_VI
AC_VO (voice)	Voice traffic and network control belong to AC_VO

A basic EDCA scheme is illustrated in Fig. 3.4. If packets from different queues in the same station contend for the access, a virtual resolution function will solve the conflict and assign the transmission opportunity to the highest priority queue, while the lowest priority packet will be retransmitted (or discarded if a maximal number of retries is reached) if the first attempt would have failed. In 802.11p the access mechanism is properly modified to work in the multi-channel WAVE environment, by implementing two separate EDCA functions, one for CCH and one for SCH, which handle different sets of queues for packets destined to be transmitted on different channel intervals.

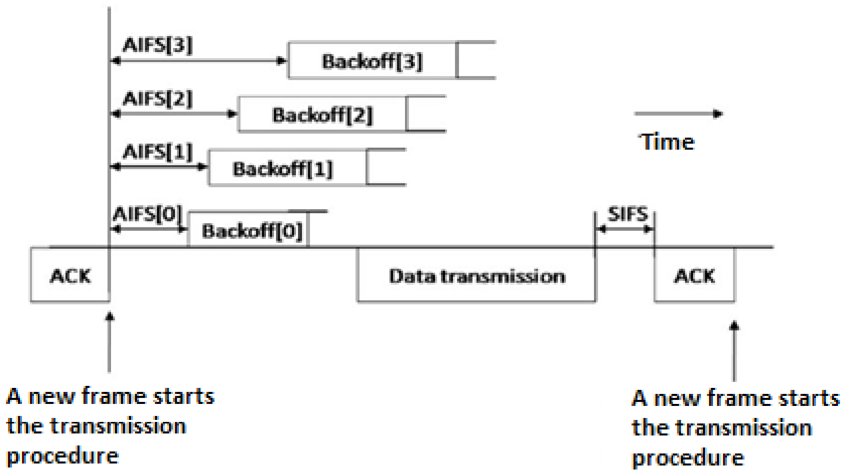

Fig. 3.4: EDCA scheme [55]

EDCA uses CSMA/CA, meaning that a node willing to transmit will sense the medium first, and if it is free for an AIFS (Arbitration Inter-frame Space), the node shall defer the transmission by selecting a random backoff time. The backoff procedure in 802.11 functions as follows:

(i) The node selects a backoff time uniformly at random from the interval [0,CW +1], where the initial CW (Contention Window) value equals CWmin;

(ii) The interval size will increase (double), if the subsequent transmission attempt fails, until the CW value equals CWmax;

(iii) The backoff value will be decreased only when the channel is free;

(iv) When reaching a backoff value of 0, it will send immediately.

Frame with priority value is mapped into an AC, when it is arriving at the MAC. The Table 3.2 specifies the values of EDCA parameters, which belong to different access categories of IEEE 802.11e.

Table 3.2: Default EDCA parameter set[53,54]

AC	**CWmin**	**CWmax**	**AIFSN**
AC_BK	aCWmin	aCWmax	9
AC_BE	aCWmin	aCWmin	6
AC_VI	(aCWmin+1)/2 -1	aCWmin	3
AC_VO	(aCWmin+1)/4 -1	(aCWmin+1)/2 -1	2

Different AIFSN (Arbitration Inter-Frame Space Number) and CW values are chosen for different ACs, as illustrated in Table 3.2. To achieve differentiation, instead of using fixed DIFS as in the DCF, an AIFS is applied, where the AIFS for a given AC is determined by the following equation 3.1:

AIFS=SIFS + AIFSN * aSlotTime 3.1

Where AIFSN is AIFS Number and is determined by the AC & physical settings, and

aSlotTime is the duration of a time slot.

The highest priority will be given to the AC with the smallest AIFS. The CWmin value is 15 and the CWmax value is 1023.

Table 3.3: Parameter settings for different application categories in IEEE802.11p [53,54].

AC	**CWmin**	**CWmax**	**AIFSN**
AC_BK	15	1023	9
AC_BE	15	1023	6
AC_VI	7	15	3
AC_VO	3	7	2

3.4 MOBILITY/TRAFFIC SIMULATOR

The choice of traffic simulator for our research has been reduced to two candidates:

SUMO (Simulation of Urban MObility) and VanetMobiSim. For the simulations required for this research it is not needed to perform complex mobility simulations, and both systems have a set of features which exceed our requirements. Both frameworks provide a wide range of configurations, including both real and artificial layouts, multi-lane roads, influence of traffic signs, etc. and can export their mobility traces as NS2 format. Hence, the choice of traffic simulator is not critical between our two candidates and it will become a choice of the researcher, who may even choose one simulator or the other according to each particular simulation requirements [56].

3.4.1 SUMO

SUMO [24] was started to be implemented in 2001, with a first open source release in 2002. It is an open source, highly portable microscopic simulator mainly developed by the employees of the Institute of Transportation System at the German Aerospace Center. There are two reasons for making the work available as open source. The first is the wish to support the traffic simulation community with a free tool into which own algorithms can be implemented. While there are some open source traffic simulations available, most of them have been implemented within a student thesis and got unsupported afterwards. A major drawback - besides reinventing the wheel - is the almost non-existing comparability of the implemented models or algorithms. A common simulation platform should be of benefit here. The second reason for making the simulation open source is the wish to gain support from other institutions.

SUMO is not only a traffic simulation, but rather a suite of applications, which helps to prepare and to perform the simulation of traffic. As the traffic simulation SUMO requires the representation of road networks and traffic demand to simulate in an own format, both have to be imported or generated using different sources. SUMO road networks can be generated either by an application named "*netgen*" or by importing a digital road map. The road network importer "*netconvert*" allows to read networks from other traffic simulators as VISUM, VISSIM, or MATsim. It also reads other common formats, as shape files or Open Street Map. Due to the lack of applications, the support for TIGER [57] networks was dropped, but TIGER networks are also available as shape files and were included in the OSM data base. Besides these formats, "*netconvert*" is also capable to read less known formats, as RoboCup network format, or openDRIVE.

The simulation is time-discrete with a default simulation step length of 1s. It is space-continuous and internally, each vehicle's position is described by the lane the vehicle is on and the distance from the beginning of this lane. When moving through the network, each vehicle's speed is computed using a so-called car-following model. SUMO uses an extension of the car-following model developed by Stefan Krauss[58]. Changing the lane is done using a model developed during the implementation of SUMO. Two versions of the traffic simulation exist. The first is a pure command line application for efficient batch simulation. The second version is a graphical application, which renders the performed simulation using openGL. SUMO allows to generate various outputs for each simulation run. These range from simulated induction loops to single vehicle positions written in each time steps for all vehicles and up to complex values as information about each vehicle's trip or aggregated measures along a street or lane. Besides conventional traffic measures, SUMO is also extended by a noise emission and a pollutant emission / fuel consumption model.

Traffic Light Algorithms

The evaluation of developed traffic light programs or algorithms for making traffic lights adaptable to the current traffic is one of the main applications for microscopic traffic flow simulations. As SUMO's network model is relatively coarse compared to commercial applications as VISSIM, SUMO is usually not used by traffic engineers for evaluating real-life intersections. Still, SUMO's fast execution time and its open TraCI-API for interaction with external applications make it a good candidate for evaluating new traffic control algorithms, both for controlling a single intersection [59] and for net-wide investigations. By distinguishing different vehicle types, SUMO also allows the simulation of public transport or emergency vehicle prioritization at intersections [60].

3.4.2 **MOVE** (MObility model generator for VEhicular networks)

MOVE [61] is a special application based on Java. It is built on SUMO (Simulation of Urban Mobility) [24] with a facility of Graphical User Interface. It is available with a very good visualization tool and focuses mainly on traffic level. It solves the problem of SUMO composite construction with just few mouse clicks without perturbing about the internal details of the simulator. It can facilitate simulation by generating mobility traces from the TIGER database or Google earth. In addition to that, it also supports custom graphs defined by user and random generated graphs. With random generated graphs, MOVE restricts the node movement to grid i.e. the node should only move on the grid. MOVE uses parser to extract topological maps from above mentioned tools.

MOVE is composed of:

(i) The Map editor:

It creates topological maps for network scenario.

(ii) The vehicular movement editor:

It generates movement patterns automatically or can also be defined by the users in the editor.

For manual generation, a trip must be defined on the basis of either attraction or repulsion point or randomly. First of all configure the start and end positions, then after random or activity based trip can be generated by MOVE. Path is nothing but a suitable way to the destination. MOVE calculates path by the means of Random Waypoint Mobility model[62] or using Dijkstra shortest path first algorithm [63].

Node velocity in MOVE is either smooth or road dependent. MOVE does not contain any network simulation capabilities but instead parses the traces to be further processed by the network simulator. MOVE generates topological maps using parser provided with the map editor and the node parameters that are defined with the help of the vehicular movement editor. This data is then passed to the network simulator. This way they both benefit from interpreters and are able to perform network & traffic tuning.

MOVE can also generate its own mobility model but the results obtained are not satisfactory as compared to that of standard mobility models. The problem accompanied with this mobility model is the lack of support for large networks i.e. its packet delivery ratio drops as the number of nodes increase, moreover multiple radio interfaces are not supported by larger networks [64].

While generating mobility traces, MOVE takes micro-mobility into consideration. The micro-mobility feature does not include any Lane-changing or Obstacle mobility models. The intersection management follows simplistic stochastic model [65] and therefore random movement of a node in the topology is not considered. The car behaviour and interaction with human behaviour follows only the car following model.

3.4.3 VanetMobiSim

The Vehicular Ad Hoc Networks Mobility Simulator (VanetMobiSim) [24] is an extension to CanuMobiSim, a framework for user mobility modelling used by the CANU (Communication in Ad hoc Networks for Ubiquitous Computing) Research Group [66], University of Stuttgart. VanetMobiSim includes a number of mobility models, as well as parsers for geographic data sources in various formats, and a visualization module. The framework is easily extensible. The software is specially written for vehicular communications and it therefore provides all the features that might be necessary for VANET simulations [67]. It is based on the concept of pluggable modules.

The set of extensions provided by VanetMobiSim consists mainly of:

i) a vehicular spatial model using GDF-compliant data structures, and;

ii) a set of vehicular-oriented mobility models.

The vehicular spatial model is composed of spatial elements, their attributes and the relationships linking of these spatial elements in order to describe vehicular areas. The spatial model is created from topological data obtained in four different ways:

- *User-defined* :- The user defines a set of vertices and edges composing the backbone of the vehicular spatial model.
- *Random* :- The backbone is randomly generated using the Voronoi tessellations.
- *Geographic Data Files (GDF)* :- Backbone data is obtained from GDF files.
- *TIGER/line Files* :- Similar to the previous one, but based on the TIGER/line files from the US Census Bureau.

Any one of those methods must be loaded after the spatial model, as it controls all data describing the topology. Then, it adds vehicular specific spatial elements such as multi-lane and multi-flow roads, stop signs and traffic lights.

The main component of the vehicular oriented mobility model is the support of a microscopic level mobility model named “Intelligent Driving Model with Intersection Management (IDM_IM)” describing perfectly car-to-car and intersection managements. In the Intelligent Driving Model with Lane Changing (IDM_LC), we also included an overtaking model called as MOBIL [68], which interacts with IDM_IM to manage lane changes and vehicle accelerations and decelerations. A *Spatial Environment* must be loaded for *user-oriented* and *vehicular oriented* mobility models to work. A spatial environment may be loaded for all other mobility models specified in CanuMobiSim. VanetMobiSim is based on JAVA and can generate movement traces in different formats, supporting different simulation or emulation tools for mobile networks including NS2. SUMO, MOVE and VanetMobiSim, all have good software features and traffic model support. However, only VanetMobiSim provides excellent trace support [69].

The following Table 3.4 shows the qualitative comparison of mobility generators VanetMobiSim and SUMO/MOVE/TraNS.

Table 3.4: Comparison of SUMO and VanetMobiSim

Attribute	**SUMO/MOVE/TraNs**	**VanetMobiSim**
Custom Graphs	Supports	Supports
Random Graphs	Grid Based	Voronoi Graphs
Graphs from Maps	TIGER database	GDF
Multilane Graphs	Support	Support
Start/End position	AP, Random	AP, Random
Trip	Random Start-End	Random Start-End
Path	RandomWalk, Dijkstra	RandomWalk, Dijkstra
Velocity	Road Dependent, Smooth	Road Dependent, Smooth
Human Patterns	Car Following Models	Intelligent driver model, Intelligent driver model with intersection management, Intelligent driver model with Lane changes
Inter section Management	Stoch turns	Traffic lights and signs
Lane changing	No Support	MOBIL
Radio Obstacles	No Support	Supports
Supports GUI	Yes	Yes
Output	NS2,GlomoSim, QualNet	NS2,GlomoSim,QualNet
Comments	Federated/Integrated	Separate

3.5 NETWORK SIMULATOR

3.5.1 Network Simulator 2

In 1989, NS2 appeared as a network simulator that provides significant simulation of transport, routing, and multicast over wired and wireless networks. NS2 code is written in C++ & OTCL and is kept in a separate file that is executed by OTCL interpreter, thus generating an output file for NAM (Network animator). It then plots the nodes in a position defined by the code script and exhibits the output of the nodes communicating with each other. It is packaged with a bundle of rich libraries for simulating wireless networks. All the mobile nodes in NS2 quickly assume that they are the part of ad-hoc network and the simulation mobile nodes connected with infrastructure networks are not really possible. For simulating a wireless node the physical layer, the link layer and MAC (medium access control) protocol are all included at the same time. In [70], authors developed MIRACLE (Multi InteRfAce Cross Layer Extension) framework extension to NS2 to simulate full space propagation model, cross layer and multi-technology in NS2. Although there are several different network simulators available today, NS2 is one of the most common. NS2 differs from most of the others by being open source software, supplying the source code for free to anyone that wants it. We have identified NS2 as our network simulator because it is the most widely accepted network simulation tools in the scientific community. Its software architecture is well prepared for extensions and enables attaching software modules for data exchange with other programs. NS2 features a comprehensive model for simulating multi-hop wireless networks and includes an implementation of the IEEE 802.11 MAC-protocol.