# Mobility model based on social network concept for VANETs

Ram S. Kale

*M-Tech, Computer Engineering, MIT Pune, Maharashtra, India*

**ABSTRACT**

*Vehicular Ad-Hoc Networks (VANET) are a particular type of wireless adhoc networks. They are formed when equipping vehicles on the roads with short range wireless communication devices. Validation of mobile ad hoc network protocols relies almost exclusively on simulation. The value of the validation is, therefore, highly dependent on how accurate the movement models used in the simulations are. However, most widely used models are currently very simplistic, their focus being the ease of implementation rather than soundness of foundation. Therefore, simulation results of protocols are often based on randomly generated movement patterns and, therefore, may differ considerably from those that can be obtained by deploying the system in real scenarios. In this we propose a new mobility model based on the social network theory. The mobility model creates movement pattern by taking into consideration the social relationship among the individuals, social relationship that might change depending on the simulation time. We also present the results obtained in validating our model using the realistic vehicular traces.*

**Keyword : -** *Mobility, Mobility Models, Social Network, Topologies*

---

## 1. INTRODUCTION

Vehicle-to-vehicle communication is a concept greatly studied during the past years. Vehicles equipped with devices capable of short-range wireless connectivity can form a particular mobile ad-hoc network, called a “Vehicular Adhoc NETwork” (or VANET). The users of a VANET, drivers or passengers, can be provided with useful information and with a wide range of interesting services.There are 3 types of applications developed in VANETs: route planning applications, safety-related applications and commercial applications.The first type of application consists of gathering real-time traffic information witch will then be shared to all those in traffic. For example we can include weather conditions publishing or adaptive signal control in intersections in this type of applications. The second type of application refers to disseminating urgent information, information that is not visible for the drivers. Such an application could, for example, be used to generate all kinds of warnings: ice on road, intersection violation or cars in front braking. Commercial and entertainment applications can be also considered successful examples of VANET instruments, as they consider the cases of electronic payments, reservations, advertisements or gaming and file transfer in the context of such systems.

VANETs are nowadays becoming more and more studied by the research community because of their appelance. However, due to the cost involved in testing in real-world a new designed solution for a particular VANET application, modeling and simulation is considered to be the de-facto approach to evaluating the performances of a wide-range of VANET technologies. In this context today many researchers are developing solutions to correctly test the performances of VANET systems using modeling. But, in order to design a complete and accurate modeling experiment, we need to be absolutely assured that the model used by a simulator closely resembles the behavior of the VANET components in realworld. This means that the VANET simulator must act very realistic. Any VANET simulator is composed of two main components: a network model and a vehicular traffic model. The network model must be able to realistically emulate the wireless network used by the entities of a VANET. The vehicular traffic model is responsible with simulating as realistic as possible the movement of vehicles in the simulated VANET system. The literature shows that the results of performance studies of ad hoc networks depend heavily on the chosen components. Also, recent studies discovered that, in order to obtain relevant results, the two described components above should be integrated and should cooperate in order to correctly assure all the functionalities of a realistic VANET system.

Integrating the two components is especially important for the situation when the vehicles' mobility is influenced by the messages the nodes receive. For instance, if some nodes are changing their route as a reaction to messages they receive, we cannot use previously generated vehicular traffic traces. Vehicular traffic simulators can in general be classified into microscopic and macroscopic simulators. When focusing on a macroscopic point of view, motion constraints such as roads, streets, crossroads, and traffic lights are considered. Also, the generation of vehicular traffic such as traffic density (number of vehicles per km per lane), traffic flows (number of vehicles per hour crossing some point, usually intersection), and initial vehicle distributions are defined. In contrast, the microscopic simulators determine the movement of each individual vehicle that participates in the road traffic and focuses on the vehicle behavior with respect to others. As the validation of mobile ad hoc network protocols relies almost exclusively on simulation, the value of the performance evaluation is highly dependent on how realistic the movement models used in the simulations are. Many of the ad hoc network studies use random way point, random directions, and other models where the nodes change their speed and direction randomly. It is obvious that such models cannot describe vehicular mobility in a realistic way. A realistic mobility model should consist firstly of a realistic topological map which reflect different densities of roads and different categories of streets with various speed limits and secondly should have a realistic movement model. In this paper we propose a new synthetic mobility model that in based on social network theory. The model allows collections of hosts to be grouped together in a way that is based on social relationships among the individuals. The model also allows to be defined different types of relationships during a certain period of time. We present results that show that the presented model simulates more realistically, compared with existing previous work in this area, the real-world behavior of the drivers that are part of a VANET system. This paper is organized as follows. Section 2 provides examples of other models that are related in some way to the presented mobility model. Section 3 describes advantages and disadvantages of existing mobility models. This section also provides information about realistic vehicular traces. In section 4 we present the VNSim simulator. This is the simulator on witch we worked on in order to prove the validity of the presented mobility model. Finally in section 5 we present the mobility traces that we imported in order to validate our model and then we provide implementation details about our mobility model. In section 6 we present simulation results and conclusions and in section 7 we will show the reference list for our paper.

## 2. RELATED WORK

A comprehensive review of the most popular mobility models used by the mobile ad hoc research community can be found in [7]. It is interesting to note that even the best solutions and approaches have only been tested using completely random models such as the Random Way-Point model, without grouping mechanisms. The work most directly related to ours can be found in [1]. In this paper the authors propose a synthetic mobility model that is based on social networks. However their simulator doesn't implement any model for lane changing and doesn't consider behavioral differences of the individuals involved in traffic. Another related example is presented in [4]. The model is based upon similar assumptions, but is considerably more limited in scope. In this model hosts are statically assigned to a particular group during the initial configuration process, whereas our model accounts for movement between groups. Moreover, the authors claim that mobile ad hoc networks are scale-free, but the typical properties of scale-free networks are not exploited in the design of the model presented by the authors. In [5], a technique for the creation of a mobility models that include the presence of obstacles is presented. The specification of obstacles is based on the use of Voronoi graphs in order to derive the possible pathways in the simulation space. Tuduce and Gross in [6] present a mobility model based on real data from the campus wireless LAN at ETH in Zurich. They use a simulation area divided into squares and derive the probability of transitions between adjacent squares from the data of the access points. Also in this case, the session duration data follow a power law distribution. Moreover, Tuduce and Gross' model represents the movements of the devices in an infrastructure-based network and not ad hoc settings. We argue that, unlike previous existing work, our model describe more accurately the real-world behavior of vehicles in term of mobility patterns. Our mobility model considers more parameters than any of the simulators presented above. Our social mobility model creates initial and final positions for all vehicles, positions that are calculated considering different attraction and rejection points, it maps the points on a geographical space and then creates movement patterns for all vehicles considering the social relationship between them.

## 3. MOBILITY MODLES

One of the critical aspects when simulating VANET systems is the employment of mobility models that reflect as closely as possible the real behavior of vehicular traffic. Many of the ad hoc network studies use random way point, random directions, and other models where the nodes change their speed and direction randomly. Such models cannot describe vehicular mobility in a realistic way, since they ignore the peculiar aspects of vehicular traffic, such as cars acceleration and deceleration in presence of nearby vehicles, queuing at roads intersections, traffic bursts caused by traffic lights, and traffic congestion or traffic jams. A step forward is the use of the real city maps and/or some kind of vehicular traffic simulators to produce somewhat close to reality vehicular movement patterns. These patterns are then used as an input for a network simulator (e.g., ns-2).

### 3.1 Classification

Globally, the development of modern vehicular mobility models may be classified in four different classes: Synthetic Models wrapping all models based on mathematical models, Traffic Simulators-based Models, where the vehicular mobility models are extracted from a detailed traffic simulator, Survey-based Models extracting mobility patterns from surveys, and finally Trace-based Models, which generate mobility patterns from real mobility traces.

In the following section we present each mobility model showing their advantages and disadvantages.

#### 3.1.1 Synthetic Models

Synthetic models are represented by mathematical models reflecting a realistic physical effect. According to Fiore's classification [7], Synthetic models may be separated in five classes: Stochastic models wrapping all models containing purely random motions Traffic stream models looking at vehicular mobility as hydrodynamic phenomenon Car Following Models, where the behavior of each driver is modeled according to vehicles ahead Queue Models which models roads as FIFO queues and cars as clients Behavioral Models where each movement is determined by a behavioral rules imposed by social influences .

A major limitation of the synthetic models is the lack of realism towards human behavior. Accordingly, realistic mobility modeling must also consider behavioral theory, social networks for instance, which makes models far from being random. For example, in [1], Musolesi and Mascolo developed a synthetic mobility model based on social network theory that behaved very realistic. They managed to validate their model by using real traces.

#### 3.1.2 Traffic Simulator-based Models

More advanced mobility models are the ones used by the realistic traffic simulators. These models are based on the refining of the synthetic models and were developed through an intense validation process using real traces or behavior surveys. Developed for urban traffic engineering, fine-grained simulators are able to model urban microscopic traffic, energy consumption or even pollution or noise level monitoring. However, those simulators cannot be used straightaway for network simulators, as no interface has yet been developed and traces are mutually incompatible. By developing parsers between traffic simulator traces and network simulator input files, the end-user could however gain access to validated traffic patterns and obtain a level of detail never reached by any actual vehicular mobility model. The major drawback of this approach is the configuration complexity of those traffic simulators.

#### 3.1.3. Survey-based Models

The survey-based Model is based on extracting behaviors statistics like commuting time, lunch time, traveling distance, preferred lunch politics, etc., and then transforming those statistics into mobility patterns. By including this kind of statistics into a mobility model, one could be able to develop a generic mobility model able to reproduce the non random behavior observed in real daily life urban traffic. The movement of each node in this case is based on individual agenda, which includes all kind of activities on a specific day. Data from the US National Household Travel Survey has been used to obtain activity distribution, occupation distribution and dwell time distribution. The mobility simulator is based on surveys from a number of research areas. Unfortunately these kinds of statistics are very rare and cover small areas. The simulator based in this kind of mobility models can validate only particular applications.

**3.1.4. Trace-based Models**

Trace-based Models are used to directly extract generic mobility patterns from movement traces. The most difficult part in this approach is to extrapolate patterns not observed directly by the traces. By using complex mathematical models, it is possible to predict mobility patterns not reported in the traces to some extends. Yet, the limitation is often linked to the class of the measurement campaign. For instance, if motion traces have been gathered for bus systems, an extrapolated model cannot be applied to traffic of personal vehicles. Another limitation from the creation of trace-based mobility models is the few freely available vehicular traces.

**3.2 Maps**

When considering a mobility model we not only take into account the road topology, but also the road structure (unidirectional or bidirectional, single- or multi-lane), the road characteristics (speed limits, vehicle-class based restrictions) and the presence of traffic signs (stop signs, traffic lights, etc.). The selection of the road topology is a key factor to obtaining realistic results when simulating vehicular movements. The length of the streets, the frequency of intersections, the density of buildings, all this details affect mobility metrics such as the minimum, maximum and average speed of cars, or their density over the simulated map. There are a number of ways to define the road topology. In all the examples below the road topology is implemented as a graph over whose edges the movement of vehicles is constrained. This classification is based on the work conducted by Jerome Harri, Fethi Filali and Christian Bonnet in [7].

User-defined graph: the road topology is specified by listing the vertices of the graph and their interconnecting edges. GDF map: the road topology is imported from a Geographical Data File (GDF) [12]. Unfortunately, most GDF file libraries are not freely accessible. TIGER map: the road topology is extracted from a map obtained form the TIGER database [13]. The level of detail of the maps in the TIGER database is not as high as that provided by the GDF standard, but this database is open and contains digital descriptions of wide urban and rural areas of all districts of the United States. Clustered Voronoi graph: the road topology is randomly generated by creating a Voronoi tessellation on a set of non-uniformly distributed points. This approach is similar to that proposed in, but we also consider the presence of areas with different road densities which we refer to as clusters. Another way to define road topology is by vehicular traces. These files describe urban and rural areas and also contain dynamic information about vehicular movement. This kind of files can be found in nam or ns2 format and they can be used directly as an input for a network simulator.

## 4. THE VNSim SIMULATOR

VNSim is a VANET simulator developed at the University Politehnica of Bucharest. It is designed as a realistic simulator for evaluating the performances of a wide-range of VANET technologies, ranging from wireless networking protocols and dissemination strategies to applications being developed over VANETs. The VNSim simulator, implemented in Java, is composed of two main models: a vehicular mobility model that considers the behavior of the drivers, and a wireless networking model, responsible with the simulation of the networking components and the communication protocols envisioned by a VANET system.

The simulator is a discrete event-based simulator. The three types of events that can be put in the queue of events are: send, receive and GPS. At every moment of the simulation time, all the current events are pulled from the queue of events, and handled by the VANET simulated components. In the next section I will present the mobility model that is part of the simulator, this representing the starting point on which my solution was developed upon.

**4.1. Maps**

The VNSim simulator uses TIGER files. Thus, for every road, the TIGER files specify its end points, along with as many intermediary points as needed. If the road is straight, no intermediary points are contained, because they can easily be computed through interpolation. If the road has curves, a large number of intermediary points are represented, thus providing an accurate map. Furthermore, for each road a simple information is given, indicating its “class” (whether it is a small street, a local road, a State Route, an Interstate Highway and so on).

The TIGER files are available in two formats: Record Type 1 (RT1) and Record Type 2 (RT2). The two file types permit us to construct the road graph. The RT1 files contain all the road segments for a map region, with information like the type, name, direction, or starting and ending points. The RT2 files contain intermediate points of the road segments for the representation of curves.

Furthermore, some improvements are also added to the map: roads are split in even smaller road segments by adding more intermediary points and roads with the same name that are connected are merged if they don't form a loop.

**4.2. The traffic simulator**

The traffic simulator is based on the driver behavior model. The basic idea of this model is the assumption that a driver can be in one of four modes: free driving, approaching, following or braking. "Free driving" means there is no influence from preceding vehicles on the same lane. In this situation, the driver will seek to obtain and maintain a desired speed. The "approaching" mode means that there is a slower, preceding vehicle which influences the driver. In this situation, she/he will apply a deceleration in order to obtain the same speed as the preceding vehicle. The "following" mode means there is a preceding vehicle, but the speeds of the two vehicles are practically equal. In this situation, the driver will seek to keep the speed constant.

**4.3.1. Scenarios Generation**

All information about the roads is stored in an object called Map. The object is serialized in a file, so that all the required computations are performed only once. There are two steps to follow in order to create traffic scenarios. First, the user can load an existing Map object and add a list of entries and exits (Various routes are computed later between each entry-exit pair). All this information is then saved in an object that is then serialized in a .smf file. Second, the user can load the newly created object and add traffic information on the map using the provided GUI.

The user can also specify flows of vehicles and the routes to follow between entries and exits. All the information is stored in another Java object and serialized in a .fsc file. This file contains all the information required by the simulator in order to run a specific scenario.

The UML diagram of the scenario designer is represented in the following Figure:

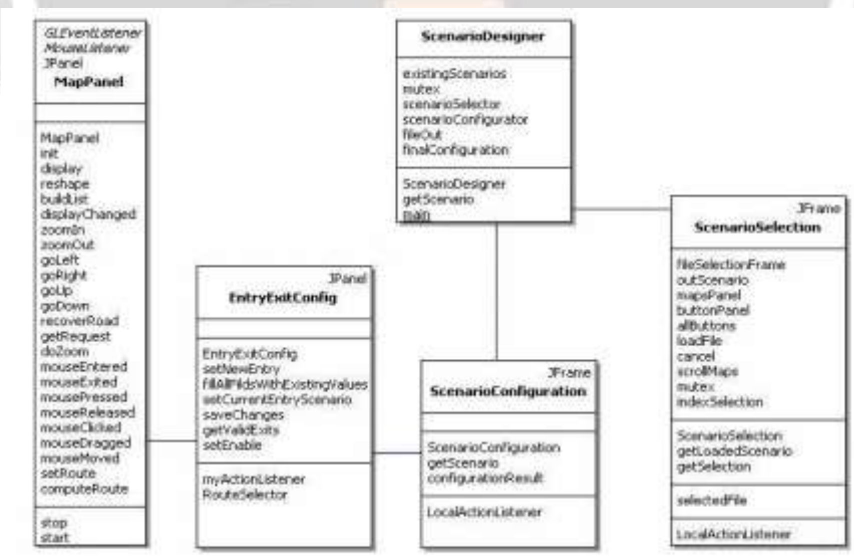



**Figure 1.** The UML diagram of the scenario designer

The "Scenario Configurator" is the container that manages all the configuration components. It contains a list of all existing entries, a set of buttons that control the map (zoom in, zoom out), the buttons for saving or canceling the current configuration, the buttons that set the configuration mode (adding a route or setting an entry's parameters) and an “Entry Exit Configurator” which is responsible with displaying all configuration parameters corresponding to the currently selected entry and with displaying the map.

For an entry there are a number of routes that lead to an valid exit point. For every entry the “Entry Exit Configurator” displays all the possible exits and all the parameters associated with the entry: vehicle flow, driver types and percent of flow for each exit. For each pair entry-exit, a list of routes is presented. For each route the percent of traffic that follows that route out of the total traffic between that entry and that exit has to be specified.

The “Entry Exit Configurator” is also responsible with displaying the map. On this map, the current entry is shown as a white square and the current exit is displayed as a black square. Each route between the two can be shown by a red line when the corresponding button is pressed. The map is also in charge with collecting the points provided by the user as coordinates for a new route.

The class diagram for the structures used by the scenario configurator.is presented in the following Figure:

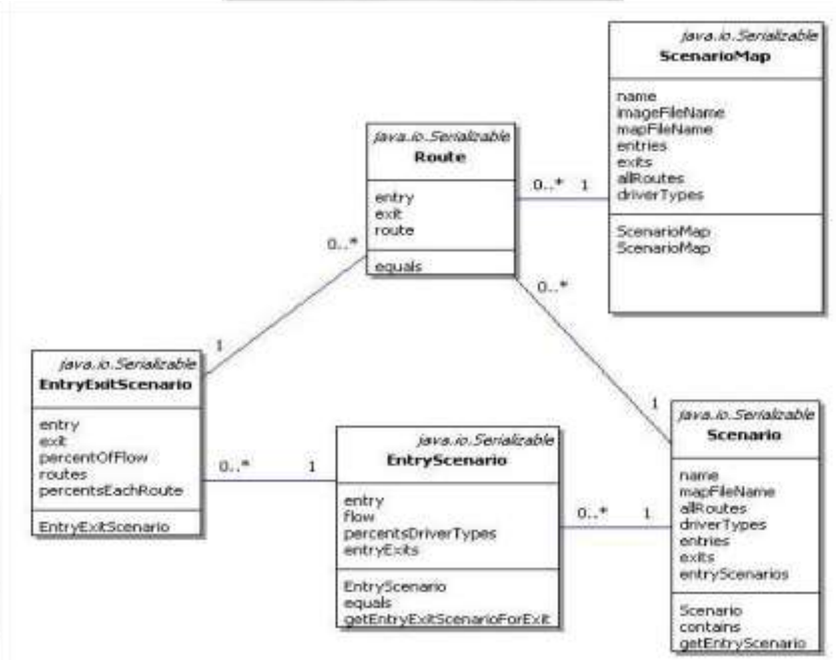



**Figure 2.** The UML diagram for the scenario configurator

A blank scenario is described by the “Scenario Map” class. Each scenario is build using the following information: the scenario’s name, the name of a picture that presents the map, the name of the map used, a list of driver types, a list of entries, a list of exits and a list of routes. Each route is described by an entry, an exit and a list of segments that form the route. The designer produces as an output a “Scenario” object. The initial purpose of the “Scenario” object was to keep only information about the configured routes and to keep it in a manner easily accessible for the simulator. This was done by keeping a list of “Entry Scenarios” for all entries that were configured. Due to the fact that many times it is useful to take a previously completed scenario and just tune it, it proved necessary to keep also the list of all entries, exits and routes from the “Scenario Map”. An “Entry Scenario” stores all the parameters corresponding to an entry: the flow of vehicles entering the map through that point, the proportions of driver personalities, and, for each configured exit, an “Entry Exit Configuration”.

The “Entry Exit Configuration” defines the traffic between an entry and an exit. It keeps all the routes used by the vehicles traveling between the two points, the percent of traffic distributed to each route and the percent of traffic coming towards that exit from the total traffic entering the current entry.

## 5. MOBILITY MODEL BASED ON SOCIAL NETWORKS

One of the most important parameters in simulating ad-hoc networks is the node mobility. It is important to use a realistic mobility model so that results from the simulation correctly reflect the real-world performance of a VANET. A realistic mobility model should consist of a realistic topological map which reflects different densities of roads and different categories of streets with various speed limits. Another important parameter that should be modeled is represented by the obstacles encountered on the roads. In addition, each vehicle needs to decide a turning directions at the intersection (turn left, turn right or go straight). Such a turning model could have an effect on the congestion of the road as well as the clustering of the vehicles.The VNSim simulator features "cars following" models, drivers' characterization, intersection management, and lane changing model. The vehicular traffic simulator is a synthetic model integrating basic microscopic motions where drivers may be in one of the following four modes: free driving, approaching, following, braking. A basic macroscopic model handles multi-lane and intersection management. The positions of Initial and Destination Position may be either random, random restricted on a graph or based on a set of attraction or repulsion points.

Attraction or Repulsion points are particular source or destination points that have a potentially attractive or repulsive feature. For instance, for a weekly morning, residential areas are repulsion points and office builds are attraction points, as a large majority of vehicles are moving from the former and to the latter. The VNSim simulator implements a random chosen initial and destination point and a random trip between them. If we want our simulator to act more realistic we need to consider that initial and destination points or the trip between them are not random. In this paper we propose two things. First, we propose adding a mobility model founded on social network theory. The model allows collections of hosts to be grouped together in a way that is based on social relationships among the individuals. This grouping is then mapped to a topographical space, with movements influenced by the strength of social ties that may also change in time. Secondly we will validate our mobility model comparing the results of the social network mobility model with the real vehicular traces from ETH Zurich.

### 5.1. Realistic Vehicular Traces

In order to validate our newly created mobility module we compared the simulation results with the one's obtaind from realistic traces. In order to do that we used the NAM simulator developed by [11] to extract the information we need from the trace file.

To validate our mobility module we use vehicular trace files developed at ETH Zurich.

#### 5.1.1. The ETH Zurich traces

Realistic Vehicular Traces are a new source of realistic mobility traces for simulation of inter-vehicle networks. These traces are obtained from a multiagent microscopic traffic simulator (MMTS) that was developed by K. Nagel (at ETH Zurich, now at the Technical University in Berlin, Germany). The multi-agent traffic simulator developed at ETH Zurich is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism. MMTS models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment. E.g., one individual might take a car and go to work in the early morning, another one wakes up later and goes shopping using public transportation, etc. Travel plans are made based on road congestion; congestion in turn depends on the travel plans. To resolve this situation a standard relaxation method is used. The major attributes of each road segment are type, length, speed, and capacity. The street network is simulated on a Beowulf Pentium cluster of up to 30 CPUs. With the help of MMTS, the consequences of construction sites, road modifications, new roads, etc. can be simulated and potential economical influence (e.g., travel time and price changes for public and private transport) can be estimated.

### 5.2. Social networks

#### 5.2.1. The social mobility model.

The definition of realistic mobility models is one of the most critical and, at the same time, difficult aspects of the simulation of applications and systems designed for mobile environments. Movement is strongly affected by the needs of humans to socialize or cooperate, in one form or another. The model that we want to add allows collections of hosts to be grouped together in a way that is based on social relationships among the individuals. This grouping is only then mapped to a topographical space. The movements of the hosts are driven by the social relationships among them. The model also allows for the definition of different types of relationships during a certain period of time (i.e., a day or a week). For instance, it might be important to be able to describe that in the morning and in the afternoon of weekdays, relationships at the workplace are more important than friendships and family one, whereas the opposite is true during the evenings and weekends.

We evaluated our model using real mobility traces provided by ETH Zurich and showed that the model provides a good approximation of real movements in terms of some fundamental parameters, such as the distribution of the contacts duration and inter-contacts time.

#### 5.2.3. Input information.

In order to collect information about input for our model I designed a GUI made in OpenGL. Figure 6 presents an example of the output of the developed interface. The first input represents the number of calm, aggressive and regular vehicles. This input sets the number of hosts and the procent of each type of vehicles. The connection threshold (named ct in this document) represents the lowest value that considers a interaction true. All values under the connection threshold are considered 0 in the adjacency matrix and 1 otherwise.

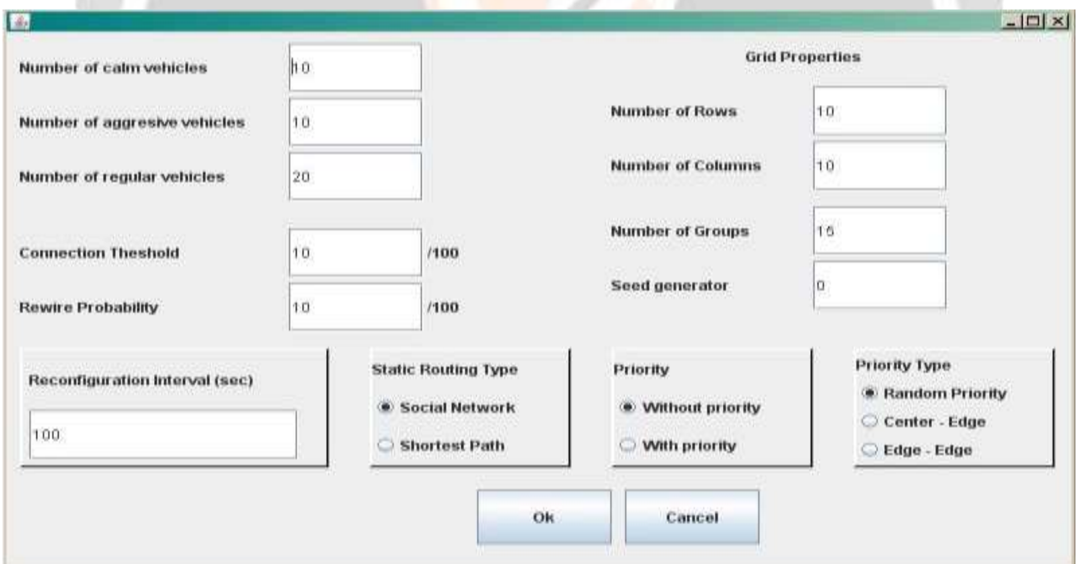


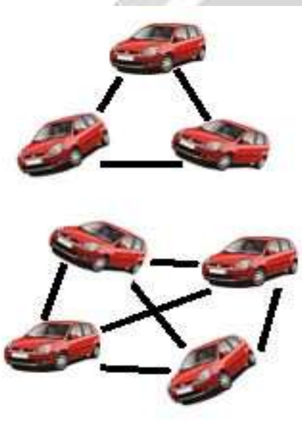

**Figure 3.** The GUI interface.

Rewire probability is used in the generation of the interaction matrix process. We will also use this variable to create the groups after creating the adjacency matrix. Number of rows and columns input sets the number of rows and columns for the grid. Number of groups sets the initial number of groups for the Caveman model and for the Girvan-Newman algorithm. Seed generator represents the seed for the generator of random numbers. Reconfiguration interval represents the number of seconds between two consecutive configurations. Static routing type sets the way that the routing will be conducted. The options are social network and shortest path. When calculating the route social network takes in consideration the entry, exit and the relation between the vehicles. The shortest path option takes in consideration only the entry and exit. The rout is calculated as the shortest path between the two points. The “with priority” and “without priority” buttons set the way in which groups are assigned to the grid. If the “with priority” button is set the groups are assigned according to a input information about points of attraction and rejection. The priority type button sets the positions of attraction and rejection points. The button can take three values.

Random Priority assigns random squares on the grid for both attraction and rejection points but the algorithm will never choose the same square for both types of points. Center-Edge button assigns for the attraction points squares in the center of the grid with higher priority. For the rejection points the algorithm assigns squares in the edge of the grid.

**5.2.4. Configuration algorithm.**

The social network is built starting from K fully connected graphs (K is an input for our model and it represents the initial number of groups). According to this model, every edge of the initial network is re-wired to point to a node of another cave with a certain probability p. The re-wiring process is used to represent random interconnections between the communities. For example Figure 4 shows an initial network configuration composed by 2 disconnected communities (caves) composed by 3 and 4 individuals; a possible social network after random rewiring is represented in Figure 5. Individuals of one group are closely connected, whereas populations belonging to different groups are sparsely connected. Therefore, the social networks generated using this model are characterized by a high clustering coefficient and low average path length. It has been proved that this model is able to reproduce social structures very close to real ones.



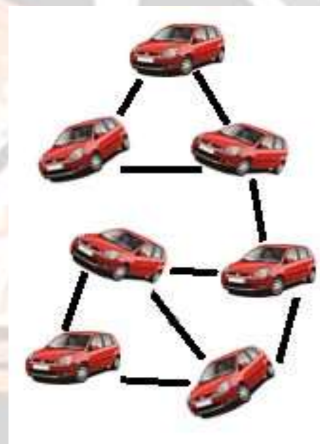


**Figure 4.** Initial configuration with 2. **Figure 5.**Generated social network after disconnected groups. rewiring.

After creating the interaction matrix using the described algorithm we generate the adjacency matrix. The adjacency matrix is formed of 0 or 1 values. If the interaction matrix has a value lower than a threshold then the adjacency matrix will have the value 0 on that position and the value 1 otherwise. From this point we will use two algorithms in order to create the communities of vehicles. We will use the Girvan-Newman algorithm in order ti create group for the entry positions of the vehicles and the Caveman Model proposed by Watts in order to create groups for the exit positions.

**5.2.4.1. The Girvan-Newman algorithm**

The Girvan-Newman algorithm focuses on the edges that are least central, the edges that are most "between" communities. The communities are detected by progressively removing edges from the original graph, rather than by adding the strongest edges to an initially empty network.

Vertex betweenness has been studied in the past as a measure of the centrality and influence of nodes in networks. For any node i, vertex betweenness is defined as the number of shortest paths between pairs of nodes that run through it. It is a measure of the influence of a node over the flow of information between other nodes, especially in cases where information flow over a network primarily follows the shortest available path. The Girvan-Newman algorithm extends this definition to the case of edges, defining the "edge betweenness" of an edge as the number of shortest paths between pairs of nodes that run along it. If there is more than one shortest path between a pair of nodes, each path is assigned equal weight such that the total weight of all of the paths is equal to unity. If a network contains communities or groups that are only loosely connected by a few intergroup edges, then all shortest paths between different communities must go along one of these few edges. Thus, the edges connecting communities will have high edge betweenness (at least one of them). By removing these edges, the groups are separated from one another and so the underlying community structure of the network is revealed.

The algorithm's steps for community detection are summarized below:

The betweenness of all existing edges in the network is calculated first.

The edges with the highest betweenness are removed.

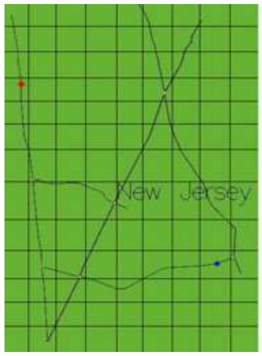
The betweenness of all edges affected by the removal is recalculated.

Steps 2 and 3 are repeated until no edges remain.

The end result of the Girvan-Newman algorithm is a dendrogram. As the Girvan-Newman algorithm runs, the dendrogram is produced from the top down (ie. the network splits up into different communities with the successive removal of links). The leaves of the dendrogram are individual nodes. Our algorithm is based on the calculation of the betweenness of edges. This provides a measure of the centrality of nodes. For example, considering two communities connected by few inter-community edges, all the paths through the nodes in one community to nodes in the other must traverse one of these edges, that, therefore, will be characterized by a high betweenness. Intuitively then, one of the possible estimation of the centrality of an edge is given by the number of shortest (geodesic) paths between all pairs of vertices that run along it. In other words, the average distance between the vertices of the network has the maximum increase when the nodes with the highest betweenness are removed. Therefore, in order to extract the communities from the network, nodes characterized by high values of centrality are progressively detected in subsequent rounds. At each round, one of the edges of the host with the highest centrality is removed. The final result is a network composed of groups of hosts. The complexity of this algorithm is O(mn2), considering a graph with m edges and n vertices. The calculation of the shortest path between a particular pair of vertices can be performed using a breadthfirst search in time O(m) and there are $O(n^2)$ vertices. However, in [9], Newman and Girvan proposed a faster algorithm with a complexity equal to O(mn). As we said, the algorithm can be run a number of times on the graph, severing more and more links and generating a number of distinguishable communities. However we also need a mechanism to stop the algorithm when further cuts would decrease the quality of the results: this would mean that we have reached a state when we have meaningful communities already. We adopted a solution based on the calculation of an indicator defined as modularity Q [9]. This quantity measures the proportion of the edges in the network that connect vertices within the same community minus the expected value of the same quantity in a network with the same community division but random connections between the vertices. If the number of edges within the same community is no better than random, the value of Q is equal to 0. The maximum value of Q is 1; such a value indicates very strong community structure. In real social networks, the value of Q is usually in the range [0.3, 0.7]. The analytical definition of the modularity of a network division can be found in [9]. At each run the algorithm severs one edge and measures the value of Q. The algorithm terminates when the obtained value of Q is less than the one obtained in the previous edge removal round. This is motivated by the fact that Q presents one or, at maximum, but much more rarely, two local peaks: therefore, we can stop when the first local peak is reached. This is clearly an approximation since the value of the other possible local peak (if exists) may be higher, but it has been observed that the quality of the division that we obtain is, in the vast majority of the cases, very good [9]. Also, by adopting this technique, we considerably simplify the computational complexity of the algorithm.

**5.2.5. The routing algorithm** At each moment we need to calculate the rout we have the entry and exit points for every vehicle. There are two ways to calculate the route. The choice is given by the user as an input for my model.

The first way is to form routes as the shortest path from the entry to the exit.



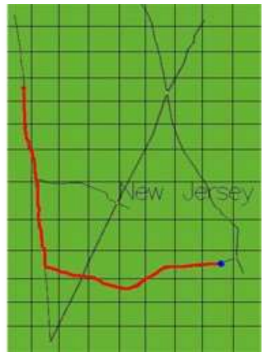




**Figure 6** shows a route computed using this method. **Figure 7**. The shortest route between entry (red point) and exit (blue point).

The second way is to form routes that take into consideration the social relationship between the vehicles. We will take a closer look at this algorithm in the following paragraph. For each node we have the initial point of entry and after the final point of exit. Knowing this information and having the adjacency matrix we are able to calculate the vehicle's Sociability Factor for every square in the grid. The Sociability Factor for a given vehicle Id and for a given square SqId represents the sum of interactions (higher than a given threshold) between Id and the vehicles placed in the square SqId divided by the number of cars in the square sqId. The formula is the one given below.

$$SFi = \frac{1}{z} \sum_{\substack{j \neq i \\ m > ct}}^{j=1..n} m_{i,j}$$

where j is a vehicle inside the square in question and z are all the vehicles that are in that square of the grid. For each vehicle after computing the Sociability Factor matrix we choose a square between the entry square and the exit square that has the highest SF value. The founded square will be added in the list of intermediary points. We repeat the algorithm until the square with the highest SF value is the entry or the exit square. In this moment we have a list of intermediary point that will be used to compute the route. The vehicle will than move according to its new route. Following this algorithm hosts will move towards their destination gradually. The nodes start moving towards the geographical region where other nodes that have strong interactions with them will converge.

For a better understanding we will consider the following example. Vehicles are placed on the map, vehicle1 has the entry point in the grid on the position (4,1) – the red dot from figure 13a. After a reconfiguration vehicle1 is in a group with other 5 vehicles. The group has the exit point in the square from the position (10,8) – the blue dot from figure 13a.

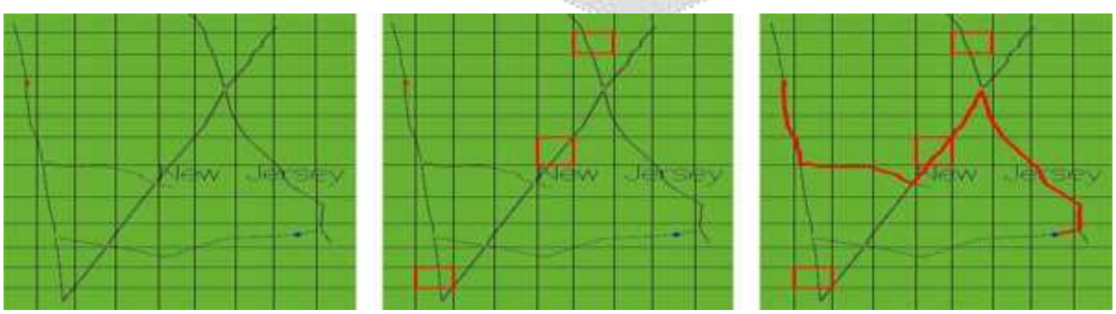



Figure 8. The social network route between entry(red point) and exit (blue point) the position of the entry and exit points. the squares on the map with the SF value greater than 0. the route that considers all intermediary points. We

will calculate for vehicle1 the Sociability Factor for each square from the grid. The only positions where the Sociability Factor is higher than 0 is for the squares (7,5), (12,2) and (2,6) – the outlined squares from figure 8b. We will choose the square between the entry and exit with the highest Sociability Factor. In our example the square is (7,5). This square is not the exit position so we now need to calculate the root between (7,5) and exit. The highest square is again (7,5) so we won't have any more intermediary points. The returned route will be the shortest path that goes thru all intermediary points. In our example the rout is the one shown in figure 8c. The described algorithm is at the basis of the dynamics of the nodes and it is based on the strength of social relationships. The reason for choosing this algorithm is that research shows that individuals with strong relationships move towards (or within) the same geographical area.

## 6. EVALUATION RESULTS

In this we presented a new mobility model founded on social network theory. The model allows collections of hosts to be grouped together in a way that is based on social relationships among the individuals. This grouping is then mapped to a topographical space. The movement of the vehicles will then be influenced by the strength of social ties that may also change in time. The results presented in this chapter show that the model provides a good

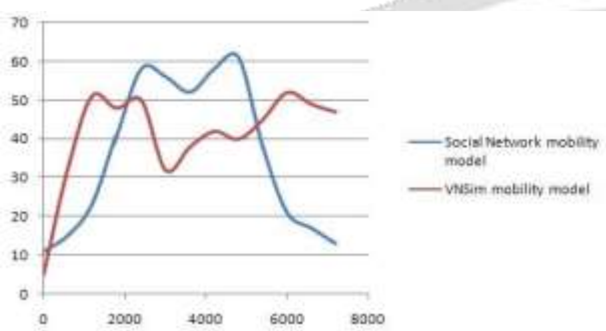
**TABLE 6.1**. Degree of connectivity

| Time (s) | 10 | 1200 | 2400 | 3600 | 4800 | 6000 | 7200 |
|---|---|---|---|---|---|---|---|
| Degree of connectivity | | | | | | | |
| Social network 225 squares | 11 | 23 | 58 | 52 | 61 | 21 | 13 |
| VNSim mobility model | 5 | 51 | 50 | 38 | 40 | 52 | 47 |

approximation of real movements in terms of some fundamental parameters, such as the

distribution of the contacts duration and inter-contacts time. We tested our mobility model using several runs generating different mobile scenarios and we compared the results with the real movement patterns provided by ETH Zurich. We evaluated the mobility model using several runs generating different mobile scenarios, first for two kinds of intersections and then for a large topological space (3,8 km x 2,4 km). The tests were performed considering a scenario composed of 100 hosts in different simulation areas divided into a grid composed of different number of squares. For each simulation we choose the corresponding priority type. We are interested in the characteristics of connection opportunities, i.e. how many and when do they occur, how often and how long. We choose to characterize these opportunities in term of contacts. To do so, we define three parameters: degree of connectivity, contact duration and inter-contact time. The contact duration is the time interval for which two network devices can communicate when they come into range. The number of such contacts and the distribution of contact durations is an important factor in determining the capacity of opportunistic networks. It gives insight on how much data can be transferred at each opportunity. The inter-contact time is the time interval between two contacts. This parameter strongly affects the feasibility of opportunistic networking, and has rarely been studied in the literature. The nature of the distribution will affect the choice of suitable forwarding algorithms to be used to maximize the successful transmission of messages in a bounded delay. The degree of connectivity is the number of contacts between vehicles in a particular moment. This parameter affects is important because it relies the maximum number of cars that can be connected in a certain moment of time. This parameter affects the choice of bandwidth and the choice of priority algorithms. We compared the results that our model obtains with the result for ETH realistic vehicular traces and we also considered the implications of the work of [10]. For the first 4 tests we considered a scenario composed of 200 vehicles in a simulation area of 3,8 km x 2,4 km divided into a grid composed of different number of squares. We choose a relatively large scenario with a low population density, in order to better see the differences in the results obtained with the two different models.

These values have not been chosen to reproduce the movements described by the traces provided by ETH Zurich, rather, we were more interested in observing if similar patterns could be detected in synthetic and real traces. In other words, our goal has mainly been to verify whether the movement patterns observed in ETH Zurich traces were reproduced by our mobility model. The duration of the simulation is two hours and the reconfiguration interval 1 hour. Each vehicle is equipped with a antenna with a transmission range of 250m. We simulated a number of scenarios involving different numbers of squares. We started with 225 (the number of rows and columns is set to 15) and in the end reached 100 (the number of rows and column is 10). A part of the results from this simulation are presented in the following table. Figure 15 shows the distribution of the degree of connectivity for the simulation made on my mobility model. As expected the degree of connectivity shows a increase in the beginning. A first local pick for the degree is somewhere between 2800-3000 s and then until 3600 s we will find a downhill. This probably happens because in the moment 0 all vehicles leave from their start point and start moving towards their destination . This is why is normal to see a increase in the majority's degree of connectivity as the vehicle approaches the center of the city. In the point where most of the routes pass we will see the highest value for the degree. After all the vehicles pass this point we can se decrees in the value of the degree.



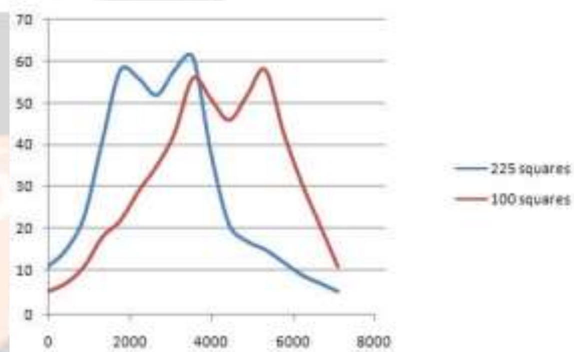




**Figure 9.** Distribution of the degree of connectivity. **Figure 10**. Distribution of the degree of connectivity for different densities of squares.

Figure 10 shows the way that the density of squares influences the distribution of the degree of connectivity. It is visible that the peaks shift to the right as the density of the squares increases. This happens because, as the number of squares increases the density of vehicles per squares decrees, so the probability that vehicles have farther starting points is greater. If the distance between the majority of the starting points is greater is normal that the distribution of degrees of connectivity will start increasing slower in the beginning. Even if the peak shifts to the right the overall form of the graph remains the same and is very similar with the curve described by the traces imported from ETH. The VNSim mobility model instead shows an increase in the beginning and then an approximate constant behavior. This is due to the fact that the VNSim mobility model has the same entry an exit points for vehicles in any moment of time.

**6.5. Vehicle density in intersections**

Another performed simulation involved the use of an intersection scenario. We simulated 2 hours of traffic with a reconfiguration interval of 1 hour. We compared the simulation results with measurement made in the Iancului intersection 30 minutes at 2 different hours in a Monday (8:30 – 9:00 and 16:30 – 17:00). The simulation is made with a edge-edge priority.

**Figure 13**. The vehicles density in two different moments of time.

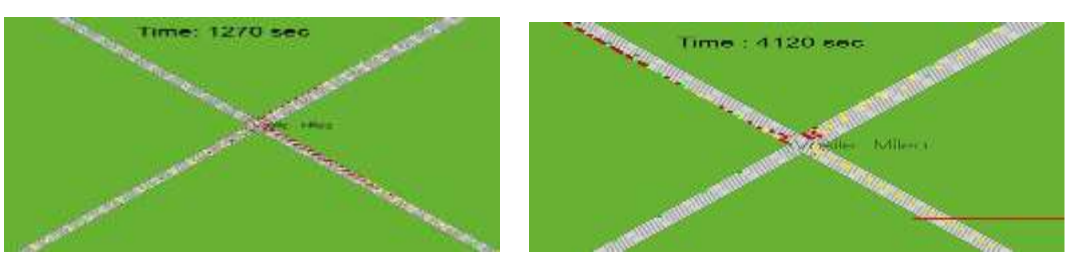
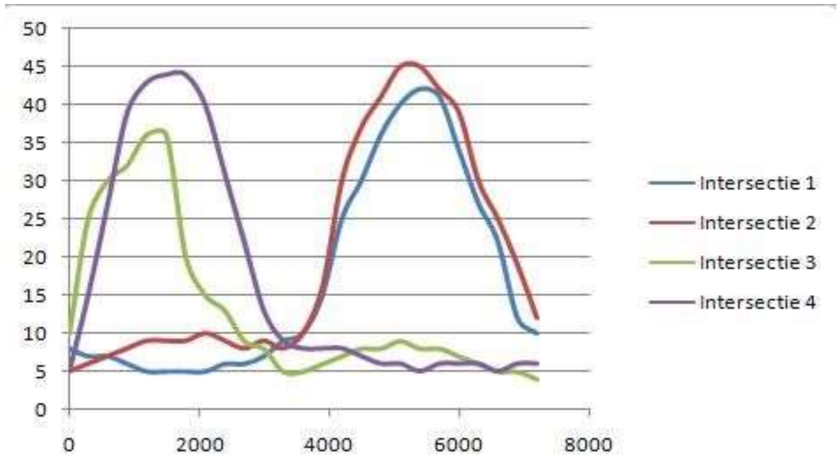


**Figure 14** shows the result of the simulation for the social network mobility model and the values obtained thru the measurements described above.




As expected, in the beginning the density of cars increases (this corresponds with the vehicles leaving home and moving towards the intersection) and than we can see a decrease (the moment vehicles pass the intersection and move towards the place of work). After a reconfiguration (moment 3600 s) the density of cars start increasing again (vehicles start moving from the work area and head home) and then decreasing again (vehicles pass the intersection). It is visible that the values obtained in the simulation are very similar with reality. In the peak hour in the morning two streets have a greater density and in the evening the other two have a greater density. This is usual for most of the intersections in real life.

## 7. CONCLUSIONS AND FUTURE WORK

The paper has described the generation of the mobility model, its implementation and an evaluation based on the comparison between our approach, existing mobility model in the VNSim simulator and real movement traces. We have shown that our mobility model generates traces that present characteristics similar to real ones, in terms of degree of connectivity, intercontacts time and contacts duration. We presented results demonstrating that the proposed mobility model reflects more accurately the real-world behavior of drivers participating in a VANET system. The presented results were performed using modeling and simulation, a technique largely used today to evaluate the performance of a widerange of VANET technologies and preferred due to the lower costs involved. Our research may be continued in a number of directions. First we could add a module that allows users to select the attraction and rejection points for entry and exit. Our mobility model allows users to select different priorities for the center and the edge of the map. It would be useful if this selection could be done in more detail. This would allow users to select the exact scenario to simulate. The module could help users to personalize a scenario (a intersection for example) so that the simulation will give more relevant information,

Another improvement could be done to the interaction matrix generation module. In the social mobility model the matrix is symmetric since, to a first approximation, interactions can be viewed as being symmetric. It is probable that by performing psychological tests, the importance of a relationship, will be valued differently by the different individuals involved; in our model, this would lead to an asymmetric matrix. We plan to investigate this issue further in the future.Finally we could add a module in the simulator that will monitor the traffic and will improve the routing algorithm using the gathered information.

## 8. REFERENCES

Mirco Musolesi and Cecilia Mascolo, ”A Community Based Mobility Model for Ad Hoc Network Research” , Florence, Italy, May 2006.

J. Scott. Social Networks Analysis: A Handbook. Sage Publications, London, United Kingdom, second edition, 2000.

D. J. Watts. Small Worlds The Dynamics of Networks between Order and Randomness. Princeton Studies on Complexity. Princeton University Press, 1999.

K. Hermann. Modeling the sociological aspect of mobility in ad hoc networks, September 2003.

A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Real world Environment Models for Mobile Ad hoc Networks. IEEE Journal on Special Areas in Communications - March 2005.

C. Tuduce and T. Gross. A Mobility Model Based on WLAN Traces and its Validation, March 2005.

Jerome Harri, Fethi Filali, Christian Bonnet - Mobility Models for Vehicular Ad

Hoc Networks: A Surveyand Taxonomy, Martie 2007

A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott - Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding, February 2005

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks, February 2004

Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, James Scott. Pocket Switched Networks:Real-world mobility and its consequences for opportunistic forwarding, February 2005

NAM Simualtor, http://www.isi.edu/nsnam/nam/

Geogrpahic Data Files (GDF) - http://www.ertico.com/en/links/links/gdfgeographicdatafiles.htm

Tiger Map Service - http://www.census.gov/geo/www/tiger/tigermap.html