Design and Analysis of Realistic Mobility Models for Wireless Mesh Networks

Master Thesis

Philipp Sommer

mail@philipp-sommer.net

Advisors:

Dr. Rainer Baumann
Dr. Franck Legendre

Supervisor:

Prof. Dr. Bernhard Plattner

Communication Systems Group
Computer Engineering and Networks Laboratory (TIK)
Department of Information Technology and Electrical Engineering

September 2007

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Acknowledgments

I would like to thank Prof. Dr. Bernhard Plattner for giving me the opportunity to write my master thesis at the Communication Systems Group and for the supervision of my work.

I would like to thank my two advisors Dr. Rainer Baumann and Dr. Franck Legendre for their continuous support and valuable discussions during progression of this work. Their suggestions and constructive feedback guided me through the work on this thesis. It was a pleasure collaborating with them.

Thanks also go to my friends and colleagues for their emotional support during these six months. Special thanks go to Christoph Studer and Patrick Sommer for proof-reading of my thesis.

Finally, I want to express my special gratitude to my parents for their unconditional support and particularly for the possibility to obtain this excellent education.

Abstract

Vehicular ad-hoc networks are a prospective technology which contribute to safer and more efficient roads. Inter-vehicle communication provides drivers and concerned authorities with information about road conditions, supports the coordination in case of emergency and improves traffic management. Furthermore, vehicular ad-hoc networks offer information and entertainment services to mobile users. Since large real-world testbeds are not feasible, research on vehicular ad-hoc networks depends mainly on simulations. Therefore, it is crucial that these simulations employ realistic mobility models.

In this thesis, we design a generic mobility simulation framework (GMSF). The implementation of this framework greatly simplifies the generation of mobility traces and the evaluation of different mobility models. The modular design enables researchers to extend the framework with additional functionality.

Furthermore, we propose a new vehicular mobility model representative of real-world characteristics. The GIS mobility model is based on highly detailed road maps from a geographic information system (GIS) and realistic microscopic behaviors (car-following and traffic lights).

We perform an extensive comparison of our new GIS-based mobility model with popular mobility models (Random Waypoint, Manhattan) and realistic vehicular traces from a traffic simulator (MMTS). In order to compare different mobility models by statistical means, we develop a set of mobility-related and graph-related metrics. Eventually, we investigate the influence of different mobility models on the performance of popular ad-hoc routing protocols (AODV and OLSR) in a network simulator.

Contents

List of Figures

List of Tables

1 Introduction

A Mobile Ad-Hoc Network (MANET) is a collection of mobile nodes which form a temporary network using wireless devices without requiring any existing infrastructure. Mobile nodes can move, leave or enter the network arbitrarily resulting in a fast changing network topology. The transmission range of a network node is limited by the environment (e.g. terrain, buildings). Only nodes within the transmission range are reachable over a direct wireless link. For all other nodes, data packets have to be forwarded from node to node until they reach the destination. Different routing protocols have been developed to cope with the constantly changing network topology in MANETs.

1.1 Vehicular Ad-Hoc Networks

A Vehicular Ad-Hoc Network (VANET) is a special form of a mobile ad-hoc network which is based on inter-vehicle communication. Using vehicles as network nodes reduces the number of base stations required to cover a densely populated area and alleviates the load on the fixed infrastructure network. Car-to-car communication will improve the traffic safety with timely provided information about car accidents, traffic jam or road conditions. Additionally, ad-hoc networks can provide mobile users with access to information and entertainment services.

1.2 Motivation

Currently, research in vehicular ad-hoc networks depends mainly on simulations since real-world testbeds consisting of a large number of nodes are infeasible. Therefore, it is crucial that the mobility models employed for these simulations are able to mimic the behavior of mobile nodes in a realistic way. Many different mobility models have been proposed by the research community for this purpose. We believe that many of the models used for simulations of vehicular ad-hoc networks cannot mimic a realistic behavior of vehicles since these models do not restrict the movement of vehicles to a realistic road topology. Furthermore, only very few mobility models take into account how the movement of a vehicle is affected by the presence of other vehicles or traffic lights. The development of vehicular mobility models representative of the real-world characteristics remains an issue.

1.3 Goals

The goals of this master thesis can be summarized as follows:

- Design of a generic mobility simulation framework.
- Design and implementation of a realistic mobility model which is based on a topology extracted from geographical data. This mobility model should mimic the movement of vehicles on roads as realistically as possible.
- Comparison of the new mobility model with common mobility models and to realistic vehicular traces from a traffic simulator.
- Development of a set of metrics for the analysis of mobility models by statistical means and by simulations in a network simulator.

1.4 Contributions

The contributions of this thesis include the following main issues:

- A generic framework for the simulation and analysis of vehicular mobility.
- A new mobility model which is based on realistic geographical data and takes into account interactions between vehicles and the influence of traffic lights.
- Different metrics to analyze the influence of different mobility models on mobility behavior and network performance.
- Evaluation results of simulations with different mobility models in realistic application scenarios for vehicular ad-hoc networks.

1.5 Structure

The remainder of this thesis is organized as follows. The second chapter deals with related work. Chapter 3 presents metrics for the evaluation of node mobility, network graphs and routing protocol performance. In Chapter 4, a new mobility model based on geographical data is proposed. Additionally, we introduce realistic vehicular traces obtained from a traffic simulator. Chapter 5 outlines the design and implementation of our generic mobility simulation framework. In Chapter 6, we present evaluation results for different mobility models using realistic scenarios. Finally, Chapter 7 concludes this thesis and gives an outlook.

2 Related Work

This chapter presents a selection of work on mobile ad-hoc networks related to this thesis. We first review common mobility models and then present different radio propagation models and routing protocols for MANETs. Finally, we conclude this chapter by an overview of network simulator software.

2.1 Mobility Models

Research in ad-hoc networks relies mainly on simulations since real-world testbeds are often not available. Simulations are fast, cheap, repeatable and make it possible to investigate the influence of single parameter variations. A large number of network nodes can be simulated which is not feasible in real-world experiments. The simulation of a mobile ad-hoc network requires the exact position information of all nodes throughout the whole simulation time. It is important that the movements of nodes are modeled in a way which matches the intended real-world deployment. In the ideal case, position traces of real users are available from a measurement campaign and can be used directly in the simulation. Unfortunately, real traces are rarely available in advance for the designated application scenario. Therefore, researchers have to rely on mobility models which define the node trajectories. A mobility model defines the exact position of a mobile node at any time. Different mobility models have been proposed to model the behavior of mobile nodes for different application scenarios. Pedestrians on a university campus move in a different manner than vehicles on a highway or a group of vehicles during a military operation. Various mobility models have been proposed and investigated by the research community. Härri et al. [1] classify vehicular mobility models in four different classes (see Figure 2.1):

- Synthetic models which are based on mathematical models.
- Models based on traffic simulators.
- Survey-based models where mobility patterns are extracted from surveys.
- Trace-based models which generate mobility patterns based on real traces acquired through measurements.

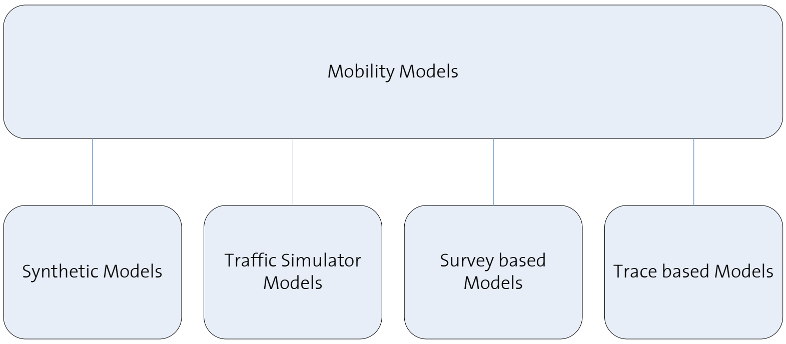


Figure 2.1: Categories of mobility models.

2.1.1 Synthetic Models

Synthetic models are used to mimic the behavior of mobile nodes in different application scenarios. In entity-based mobility models, each node acts independently of other nodes. Group-based models organize nodes in different groups which move together towards a common destination (e.g. a group of soldiers assigned to a common task). Another category are vehicular mobility models which take into account interactions with other vehicles in the proximity. In the simplest case, a vehicle maintains a constant distance to the vehicle in front. Advanced models extend the simulation of road traffic by road signs or traffic lights.

Entity Mobility Models

Entity models simulate mobile nodes which are acting independently of other nodes. The two most commonly used entity-based mobility models are Random Walk and Random Waypoint.

Random Walk The Random Walk Mobility Model was first described mathematically by Einstein when investigating Brownian Movement [2]. It mimics the erratic movements of particles observed in nature. Starting from its initial position, a node randomly chooses a direction (uniformly distributed in $[0, 2\Pi]$) and a travel speed (uniformly distributed in $[speed_{min}, speed_{max}]$). It moves in this direction either for a constant time interval t or until it has travelled a constant distance d. Each time a node reaches a destination point, a new direction and speed is selected randomly leading to a new movement. The random walk model is memoryless because speed and direction of subsequent movements are completely independent. This results in abrupt changes of the node's direction and speed.

Random Waypoint The Random Waypoint Mobility Model [3] is an extension of the Random Walk model. It introduces a pause after the node has reached its destination point. The pause time is uniformly distributed between a minimum and a maximum value. After the pause, the node starts moving towards the next randomly chosen destination point and the process is repeated again. Figure 2.2 shows the movement pattern of a node according to the Random Waypoint mobility model.

The distribution of the nodes in the simulation area and the distribution of the node speeds varies over the simulation time. Even though the speed of a node is chosen from a uniform distribution, the distribution of the node speed sampled after a long enough time will differ from the uniform distribution where the speed for the next trip is chosen from. A lower speed is more likely to be

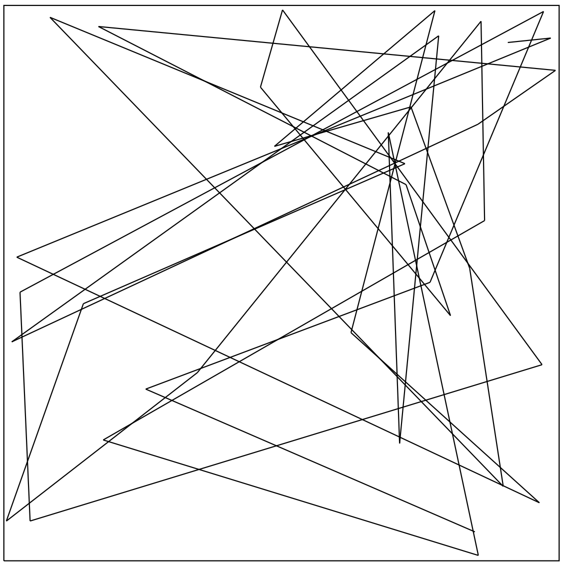


Figure 2.2: Movement pattern of a node using the Random Waypoint Mobility Model.

observed at an arbitrary point in time since a node will take more time to complete a trip with a lower speed and so, on average, nodes spend more time on trips with lower speed. The same effect can be observed for the node position density. Although the start and end point of a trip are chosen independently and uniformly over the whole simulation area, the probability that a node can be observed at a random point is not equal everywhere. In the steady-state, it is more likely to find a node in the center of the simulation area. Camp et Al. [4] analyzed the steady-state distribution and proposed different solutions to avoid this problem when performing simulations based on the Random Waypoint Mobility Model:

- Discarding the first seconds of the simulation (e.g. 1,000 seconds) to ensure that the simulation is in the steady-state phase.
- Saving the position of nodes in the steady-state to a file and start from the saved positions.
- Start sampling the initial node position and speed from the steady-state distribution.

Navidi and Camp [5] derived the stationary distributions for location, speed and pause time for the Random Waypoint Model. Le Boudec et al. [6] showed how the steady-state distribution can be calculated using Palm calculus and presented an implementation for sampling directly from the stationary distribution (“perfect sampling”) for random trip models.

Group Mobility Models

Group mobility models simulate the behavior of mobile nodes which move together towards a common destination.

Reference Point Group Mobility Model In some application scenarios nodes are organized in teams (e.g. military vehicles in a battlefield) which move in close distance towards a common target. The Reference Point Group Mobility model [7] assigns a logical center to each group and a group motion vector $\overrightarrow{GM}$ (see Figure 2.3). The trajectory of the group center ($GC(t)$) is generated using the Random Waypoint model. Each node has a reference point $RP(t)$ within a certain range from the group center which is moved together with the movement of the group center.

$$RP(\tau + 1) = RP(\tau) + \overrightarrow{GM} \tag{2.1}$$

The position of a node is calculated by adding a random motion vector $\overrightarrow{RM}$ to the reference point.

$$x(t) = RP(t) + \overrightarrow{RM(t)} \tag{2.2}$$

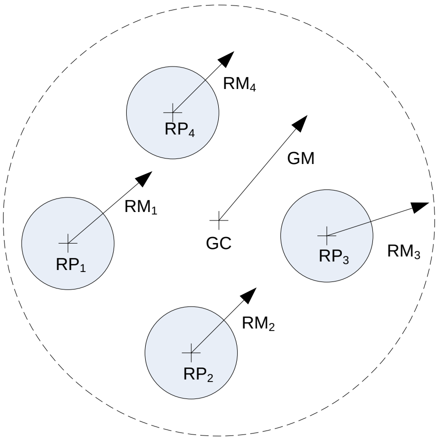


Figure 2.3: Reference Point Group Mobility Model.

Vehicular Mobility Models

Vehicular mobility models can be classified based on the modeling approach used: Macroscopic models consider aspects like the road topology, speed limits and collective vehicle behavior like traffic flows and traffic density. Microscopic models, instead, focus on the behavior of each individual vehicle and the influence of other vehicles on the driver's behavior.

Freeway Model The Freeway mobility model proposed by Bai et al. [8] models the behavior of vehicles traveling on a freeway. The movement of a node is restricted to a lane of a freeway (see Figure 2.4) and is temporally dependent on the previous speed and other vehicles travelling in front on the same lane.
Relationship between the speeds at subsequent time slots:

$$v_i(t + 1) = v_i(t) + random() * a_i(t) \tag{2.3}$$

where a_i is the acceleration constant of node i.

Relationship between the speed of a vehicle (i) and the speed of the vehicle in front (j):

$$d(i, j) \leq d_{safety} \Rightarrow v_i(t) \leq v_j(t) \tag{2.4}$$

where d_{safety} is the safety distance to the front vehicle.

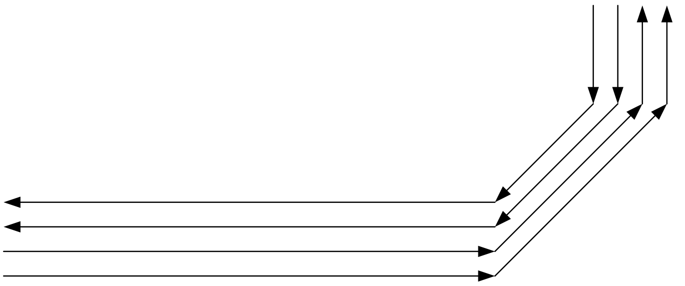


Figure 2.4: Sample map for the Freeway mobility model.

Manhattan Model The Manhattan mobility model was introduced in [8] and models the behavior of vehicles in a city area. Movements of vehicles are restricted to a map containing roads in horizontal and vertical direction (see Figure 2.5 for an example). If a node reaches an intersection, it randomly selects the street it will follow. The probability of moving straight is 0.5 and the node turns left or right with probability of 0.25 each. As in the Freeway model, the node's speed depends on its speed in the previous time slot and on the speed of the node in front on the same lane.

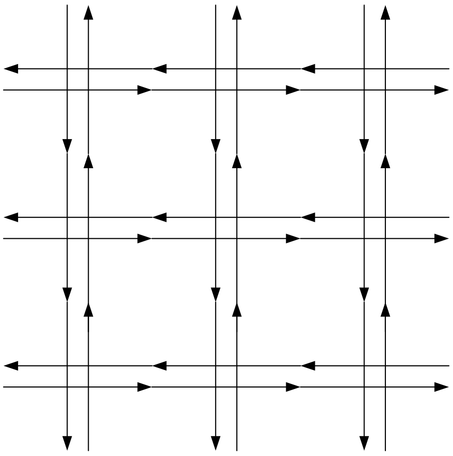


Figure 2.5: Sample map for the Manhattan mobility model.

Street Map-based Models More advanced vehicular mobility models make use of real street maps. Saha and Johnson [9] use detailed street maps from the U.S. Census Bureau. The bureau's TIGER (Topologically Integrated Geographic Encoding and Referencing) database contains information about start and endpoint of a road along with the road type. Nodes move between randomly selected points on the road graph. The exact path of a node is calculated using Dijkstra's shortest path algorithm. If a node reaches its destination point, a new destination point is randomly selected and a new path calculated. The speed of a node is randomly chosen in the range of 5 miles/hour above and below the speed limit of the corresponding road type. Baumann et al. [10] proposed a realistic mobility model using detailed vectorized street maps of Swiss cities. Node movements are generated according to the steady-state random trip model.

Stop Sign/Traffic Sign Model An important characteristic of vehicular mobility is the behavior of mobile nodes at traffic intersections (stop signs, traffic light, roundabouts). Potnis and

Mahajan [11] proposed two mobility models which simulate the influence of stop signs and traffic lights on the mobility of nodes. In the Stop Sign Model (SSM) each vehicle stops at every intersection and waits for a fixed time period. In the Traffic Sign Model (TSM) vehicles stop with the probability of 0.5 for a random time in front of a traffic light. It is possible in both models that vehicles will queue at an intersection before proceeding their journey. This leads to a clustering of nodes which has an influence on the routing protocol performance.

STreet RAndom Waypoint (STRAW) Choffnes et al. proposed STRAW [12] which uses roads defined by real street maps to model vehicular mobility. A car following model is implemented to model inter-segment mobility. Vehicles adapt their speed to the vehicle in front on the same lane. Simplified traffic control mechanism models are employed to describe the behavior of vehicles at intersections.

2.1.2 Traffic Simulator-based Mobility Models

This category of mobility models is based on traffic simulators. Such simulators are used for the understanding of traffic phenomenons like congestion and to study large-sized road networks. In order to provide accurate simulations of the behavior of road traffic, different models have been proposed by traffic researchers [13]. Since such models provide detailed data about position and speed of the vehicles involved in the simulation, vehicular simulation traces can be used in network simulators.

Simulation of Urban MObility Simulation of Urban MObility (SUMO) [14] is an open source traffic simulation package. Main purpose of SUMO is to simulate a city-sized road network. The simulation consists of individual agents which can walk, take public transport or drive a car. The position of a vehicle is updated every time step based on the street topology (intersections) and other vehicles on the same road. The underlying traffic model is the car following model described in [15].

Multi-Agent Microscopic Traffic Simulator The Multi-Agent Microscopic Traffic Simulator (MMTS) [16] was developed at ETH Zurich for the simulation of public and private transports in Switzerland. MMTS models people living in a certain area as agents which decide on a type of transportation according to their day's schedule. The number of people living in an area is based on statistical data gathered by the Swiss government. Each agent chooses a start time and a means of transportation dependent on the environment (availability of public transport, time consumption) and needs (start of workday, holiday, shopping, leisure). Vehicular traces from MMTS were used in the simulation of realistic vehicular networks [10].

2.1.3 Survey-based Mobility Models

Statistics about the social behavior of people were gathered through a large number of surveys. The University of Delaware mobility model [17] is based on surveys by the US Bureau of Labor about time-use of workers and vehicle mobility surveys by planning authorities. These data are used to model realistic behavior of people in an urban setting. Activities of individuals in the model are based on statistical data of daily routines (arrival time at work, lunch break duration) and movements of people.

2.1.4 Trace-based Mobility Models

Since complex mobility models are required to produce realistic motion patterns, researchers extract mobility patterns from observed movement traces. Kotz et al. [18] collected and analyzed user traces from the campus wide wireless local area network (WLAN) at Dartmouth College over a time period of eleven weeks. Tuduce and Gross [19] proposed and evaluated a mobility model based on WLAN user traces collected at the ETH Zurich. Parameter sets for the mobility model are extracted from the collected WLAN user traces of campus wide deployed access points. The area in the proximity of an access point is modeled as a cell and mobile users move between cells. Zhang et al. [20] used mobility traces of 40 buses of the UMass DieselNet to build a mobility model for transport-based disruption tolerant networks. McNett et al. [21] collected mobility patterns of wireless handheld users in a campus wide wireless network to develop a trace-based campus waypoint mobility model.

2.1.5 Mobility Traces Generator Software

Many different applications or frameworks are available to produce traces of mobile nodes according to a chosen mobility model. The Random Waypoint model is integrated into most network simulators. CanuMobiSim [22] is a Java-based framework for user mobility modeling featuring different mobility models. Traces can be exported to the file format of various network simulators. VanetMobiSim [23] extends CanuMobiSim with vehicular mobility and is able to import maps from the US Census Bureau TIGER database and other sources. STRAW [12] generates vehicular traces using real road maps and takes into account car-following and traffic lights. The IMPORTANT framework [8] contains metrics to capture mobility characteristics and evaluate the impact of the performance of routing protocols. TraNS [24] combines vehicular traces from the SUMO traffic simulator with the capabilities of the ns-2 network simulator.

2.2 Radio Propagation Models

In contrast to wired networks where devices communicate over cables or optical fibers, communications in ad-hoc networks require a wireless communication channel between the transmitter and the receiver. Radio waves are exposed to reflection, diffraction or scattering based on the environmental conditions leading to multipath propagation. The multiple signal paths are added up at the receiver leading to constructive or destructive interference which causes the received power level to vary.

A radio signal can be successfully received when the signal to noise ratio (SNR) is above the receiver's sensitivity. The area around the transmitter where the signal can be received correctly (if no interference is present) is called communication range. In the simplified case, the communication range lies within a circle around the transmitting station (see Figure 2.6). Although the signal power is too low for a successful reception outside the communication range, the signal can still interfere with a signal from an unrelated station. Nodes inside the interference range are said to be hidden nodes [25]. Xu et al. [26] showed that the interference range of a node is around 1.78 times the communication range.

2.2.1 Model of a Wireless Communication System

A radio transmission system consists of multiple interconnected components which each influences the overall performance of the system. A link budget is the sum of the contributions of all gains and losses in a radio communication system between the transmitter and the receiver [27].

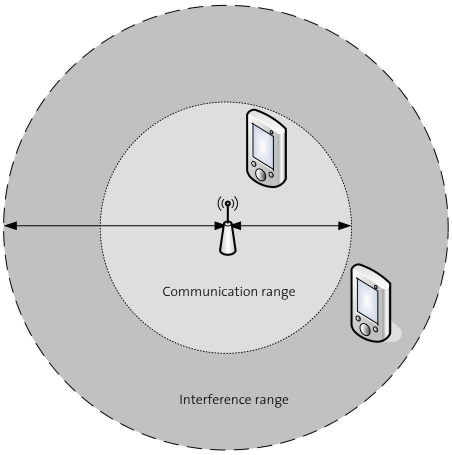


Figure 2.6: Communication and interference range of a wireless node.

Figure 2.7 shows a simple model of a radio transmission system. On the wireless channel the radio signal is affected by path loss as a function of distance. Antennas and hardware components such as connectors contribute additional gains and losses to the link budget of the system.

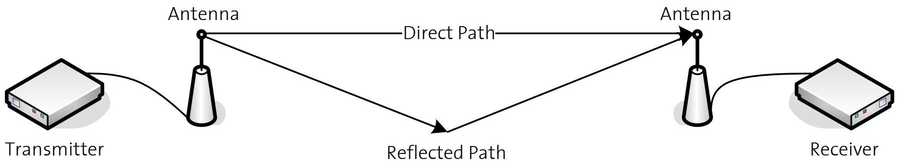


Figure 2.7: Schematic of a radio transmission system.

The received power can be calculated as:

$$P_r = P_t L_p G_t G_r \tag{2.5}$$

where:
P_r = received power
P_t = transmitter output power
G_t = total transmitter gain
L_p = path loss
G_r = total receiver gain

2.2.2 Path Loss Models

The output power of the transmitter and system gains are usually constant values depending on hardware components. Path loss is the ratio of the transmitted signal power to the signal power at the receiver. It is dependent on the path length, the used radio frequency, and on the terrain. Realistic path loss models have to take into account propagation effects due to the specific characteristics of the terrain which involves complex computations. The UDelModel for radio propagation [28] uses a 3-dimensional ray-tracing approach to determine the channel gain matrix by calculating signal paths. Because of the high computational complexity of such realistic models, simple radio propagation models such as the Free-Space model or the Two-Ray model are used to estimate the path loss.

Free-Space Model The Free-Space model [29] is the simplest radio propagation model. It assumes ideal propagation with a line-of-sight path between the transmitter and the receiver. The received signal power in free-space depending on the distance d between the transmitter and the receiver is given by the Friis law:

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{2.6}$$

A transmitted signal can be successfully received at the distance of d if the received signal power is above the receiver's sensitivity. Positions where the signal power is equal to the receiver's sensitivity are located on a circle around the transmitter. All stations inside this circle are in the communication range of the transmitter.

Two-Ray Model The Two-Ray model [29] considers, additionally to the direct path, a ground reflection path from the transmitter to the receiver. It uses the heights of the transmitter antenna (h_t) and receiver antenna (h_r) to compute the ground reflection effect. The Two-Ray model predicts the received power at a distance of d from the receiver with the following equation:

$$P_r(d) = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4} \tag{2.7}$$

The received power in the Two-Ray model decreases as the fourth power of distance while the received power in the Free-Space model decreases as the square power of distance.

Log-Distance Path Loss Model The Log-Distance Path Loss Model [30] predicts the signal power relative to the mean received signal power at the reference distance d_0. Formally,

$$\frac{P_r(d_0)}{P_r(d)} = \left(\frac{d}{d_0}\right)^\beta \tag{2.8}$$

The reference distance d_0 should always be chosen in the far field region of the antenna (e.g. 100 m away from the transmitter). The received signal power at the reference distance $P_r(d_0)$ is calculated using the Free-Space path loss model.
The path loss exponent β can be determined empirically by measurements. Table 2.1 gives some typical values [31].

Environment		β
Outdoor	Free space	2
	Shadowed urban area	2.7 to 5
In building	Line-of-sight	1.6 to 1.8
	Obstructed	4 to 6

Table 2.1: Some typical values of the path loss exponent β.

2.2.3 Fading

Both, the Free-Space model and the Two-Ray model, are based on deterministic equations to predict the received signal power as a function of distance. In reality, the situation is much more complex and the received signal power changes in the course of time due to the multiple signal paths which are added up at the receiver leading to constructive and destructive interference.

We call this effect shadowing fading when the signal variations are caused by buildings or large objects which block the direct transmission path. The popular network simulator ns-2 (see Section 2.4) features a radio propagation model called “Shadowing Model” consisting of a path loss part and a shadowing part. In this model, signal power variations are modeled using a Gaussian random variable X_{dB} with zero mean and a variance σ_{dB}, called the shadowing deviation.
Adding both parts leads to the following equation for the received signal power with the Shadowing Model:

$$\left[\frac{P_r(d)}{P_r(d0)}\right]_{dB} = -10\beta log_{10}\left(\frac{d}{d0}\right) + X_{dB} \tag{2.9}$$

Typical values of the shadowing deviation σ_{dB} obtained by measurements (see [31]) are listed in Table 2.2.

Environment	σ_{dB} (dB)
Outdoor	4 to 12
Office, hard partition	7
Office, soft partition	9.6
Factory, line-of-sight	3 to 6
Factory, obstructed	6.8

Table 2.2: Some typical values of shadowing deviation σ_{dB}.

Figure 2.8 compares the received signal power for the three radio propagation models presented in this section.

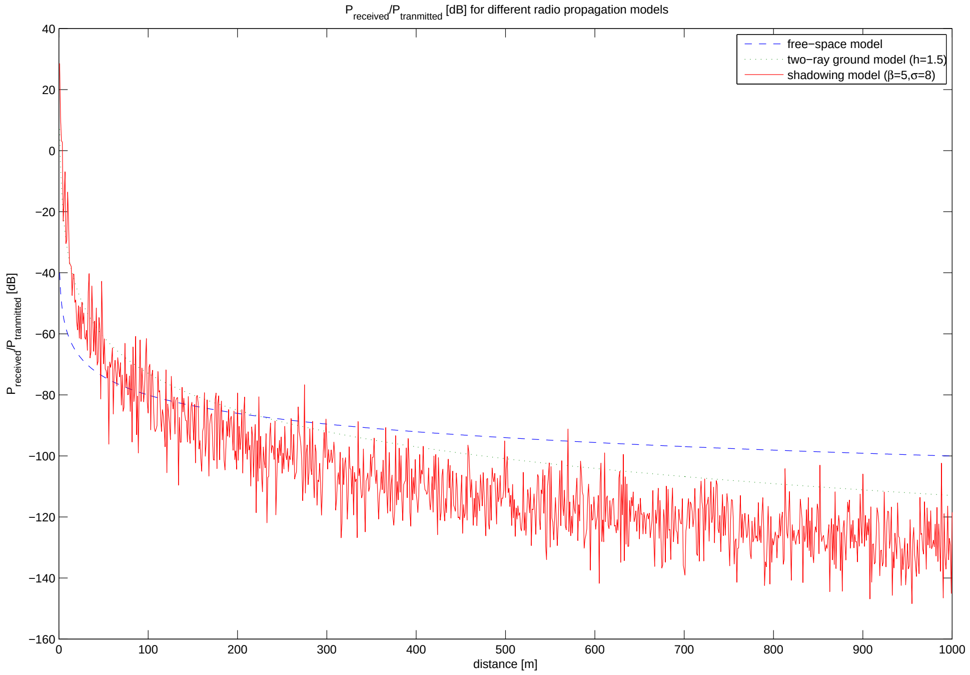


Figure 2.8: Comparison of the received power for the three different radio propagation models.

2.3 Routing Protocols for Ad-Hoc Networks

In this section, we present two commonly used routing protocols for ad hoc networks:

- Optimized Link State Routing (OLSR)
- Ad-Hoc On-demand Distance Vector (AODV)

Wireless ad-hoc networks consist of mobile nodes connected by wireless links. A network is usually modeled as graph $G =< V, E >$ containing the set of network nodes $V = \{1 \ldots n\}$ and the set of edges $E \subseteq V \times V$ between the nodes (see Figure 2.9). The network graph of an ad-hoc network is dynamic due to the mobility of nodes and radio propagation effects.

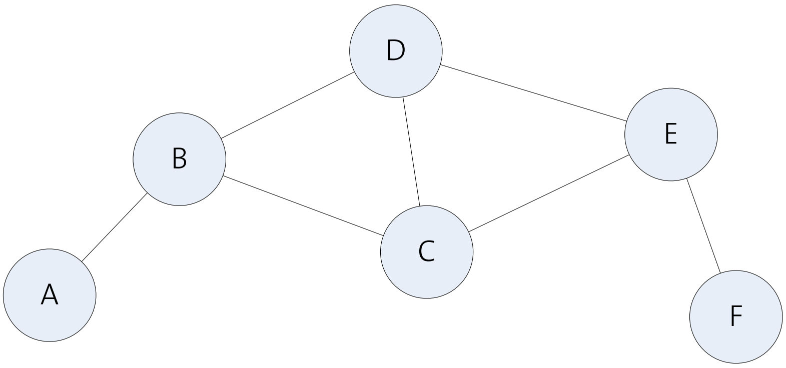


Figure 2.9: Network represented by a graph.

Due to the limited communication range of wireless devices, nodes can only communicate with direct neighbors. If source node A wants to send a packet to destination node F which is not in the communication range of A, the packet may have to be forwarded by intermediate nodes. A routing protocol is employed to select a path (route) in a network and forward data packets along this path. Routing involves two tasks:

- Computation of a path between source and destination by a distributed algorithm
- Forwarding of data packets along the route

Routing in MANETs is challenging due to the constant changes in the network topology and the constrained transmission bandwidth. Therefore, routing protocols for ad-hoc networks differ from traditional routing protocols used in wired infrastructure networks like the Internet.
Routing protocols can be categorized by the time when routes are calculated:

- Pro-active routing protocols try to maintain an up-to-date view of the network at each node. Routes are calculated in advance, before they are actually used.
- Re-active protocols, in contrast, calculate routes only on demand. This is useful when bandwidth is limited or energy consumption should be minimized.

From an other point of view, routing protocols can be categorized into two classes based on the type of routing information exchanged between nodes:

- Distance Vector (DV) protocols: exchange of whole routing table with neighbors
- Link State (LS) protocols: state of the link to neighbors is flooded to other nodes

2.3.1 Ad-Hoc On Demand Distance Vector (AODV)

The Ad-Hoc On Demand Distance Vector protocol was first described in [32]. AODV is a re-active routing protocol. Routes are calculated on demand when a node wants to send a data packet. The route discovery process is started when a source node S wants to send a data packet to a destination node D for which no route is available in the routing table of S. Node S floods a route request packet (`RREQ`) into the network. A route request packet contains: source identifier, source sequence number, destination identifier, destination sequence number, broadcast identifier and a time to live (TTL). An intermediate node replies with a route reply packet (`RREP`) if it knows a valid route to the destination node, otherwise, the route request is forwarded to its neighbors. When forwarding a route request packet, a node sets up a reverse path to node S which uses the neighbor of S from which the request packet has first been received.

A route request flooded into the network will cause a large amount of routing packets generated throughout the network even if the destination node is only a few hops away. Expanding ring search [33] is introduced to make AODV scale better in large networks. The source node successively increases the search area into which `RREQ` packets are flooded. This is done by adapting the initial time to to live (TTL) value in the `RREQ` packets.

Nodes periodically send `HELLO` messages to detect link failures. If a link failure is detected, a node sends a route error packet (`RERR`) towards the source node. Routing table entries that are not actively used expire after a pre-defined interval and are removed from the routing tables.

2.3.2 Optimized Link State Routing (OLSR)

The Optimized Link State Routing protocol [34] is a pro-active routing protocol based on link state. Network topology information is continuously distributed over the network and stored locally at each node. Continuous flooding generates a lot of redundancy, increases the possibility of collisions and wastes a lot of bandwidth. OLSR optimizes the distribution of topology control (TC) messages over the network by an optimized forwarding mechanism called multipoint relaying. Link state updates are only sent over a subset of all links. Every node selects a set of multipoint relays (MPR) from its one-hop neighbors which forward the link state packets generated by the node (see Figure 2.10). Neighbors which are not a MPR for that node do not forward the link state packets. The set of MPRs of a node n have the following property: Every node in the two-hop neighborhood of n has a link to a MPR of node n. Nodes send periodic HELLO messages to discover its neighbors and to announce its neighbors to other nodes. The Multipoint Relay Selector Set of a node n contains the neighbors that have selected node n as an MPR. Each node periodically sends topology control messages to announce that it has reachability to the nodes of its MPR selector set.

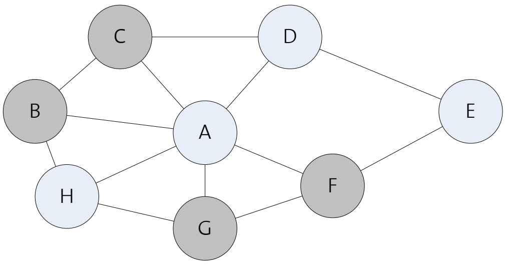


Figure 2.10: OLSR: Multipoint relays (gray) of node A.

2.4 Network Simulators

Network simulators are important tools for research in wireless networks. A network simulator models the different layers of a communication stack. Although developers of network simulators aim to model the network as realistically as possible, a simulation can not always predict the reality. Testing a newly developed protocol in a real-world testbed is an indispensable step before deployment. However, simulations have some advantages compared to real-world tests. Simulations are fast, cheap, reproducible and allow parameter isolation. Therefore, research organizations, universities and commercial companies have made a large effort to develop sophisticated network simulator software. This section presents some of the commonly used network simulators.

2.4.1 The Network Simulator

The Network Simulator (often called ns-2) [35] is a popular event-based network simulator for academic research. Development of ns began in 1989 as a variant of the REAL network simulator and has since then gained support by research agencies, universities and commercial companies. Ns-2 is released as open source software and is available for different platforms. The simulator software is written in C++. Simulation parameters are scripted in OTcl, an object oriented extension of the programming language Tcl. For each object in C++ there exists a corresponding Tcl-object which makes it possible to access and modify simulation properties during the simulation. Ns-2 contains a wide variety of physical layer models and network protocols. The Network Animator (NAM) tool [36] distributed as a separate application enables to visualize mobility and packet traces produced during simulations in ns-2.

2.4.2 Global Mobile Information Systems Simulation Library (GloMoSim)

GloMoSim [37] is a library for sequential and parallel simulation of wireless networks developed by the Parallel Computing Laboratory at the University of California in Los Angeles. It is based on PARSEC, a compiler for a C-based parallel simulation language. An extensible set of library modules models specific communication protocols in the network protocol stack.

2.4.3 Qualnet

Qualnet [38] is a commercial network simulator developed by Scalable Network Technologies as the successor of GloMoSim. It is available for various operating systems (Microsoft Windows, Linux, Mac OS X, Sun Solaris) and supports parallel computing environments. Qualnet offers highly detailed models and can be extended with user-defined modules. The Qualnet product family offers graphical tools to design, run and analyze simulation scenarios.

2.4.4 OPNET

OPNET [39] is a commercial network simulator which models communication devices and protocols and offers a large variety of built-in analysis tools. It is mainly for the simulation of network designs, to evaluate the influences of changes in the network and the configuration of the network.

3 Metrics for Mobility Evaluation

Mobility models are used to generate movements of mobile nodes in an ad-hoc network. Position, speed and moving direction of nodes are defined by the mobility model during the whole simulation. Since mobility causes constant changes in the topology of the ad-hoc network, it has a large impact on the performance of the network. In this chapter, we present different metrics to evaluate mobility, network topology and routing protocol performance.

3.1 Mobility Metrics

In order to compare statistical properties of node mobility generated by different mobility models, a set of well-defined metrics is required. We first give a definition of the terminology used to describe the metrics.

3.1.1 Definition of Terms

- Node: Uniquely identified entity in the simulation which represents a pedestrian, vehicle or base station equipped with wireless devices. Nodes are either moving with a certain speed and direction or staying at the same position. The characteristics of the mobility of nodes depend on the underlying mobility model.

- Population: Set of nodes currently inside the specified simulation area. The population is defined as the set of currently participating nodes:

  $$P = \{1, 2, ..., N\} \tag{3.1}$$

  Depending on the mobility model, the population size remains constant or is changing due to nodes entering or leaving the simulation area. Furthermore, network nodes can be turned off or on during the simulation.

- Neighbor: Two nodes which are within their mutual communication range are called neighbors. The set of neighbors of node n is called the neighborhood of node n and consists of all nodes within the communication range of node n. The communication range of a node can be modeled as a circle with radius R in the ideal case (Free-space model), see Figure 3.1.

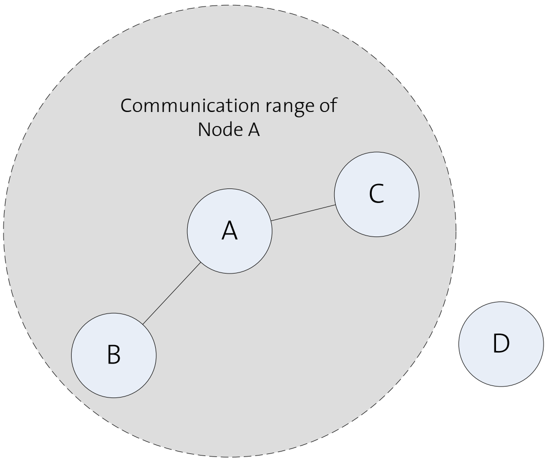


Figure 3.1: Neighborhood graph of node A.

- Contact: A contact defines the time period during which two nodes are neighbors. A contact starts from the moment when the two nodes are in communication range until the link between the two nodes breaks.

  A contact between nodes a and b is defined by the following properties:

  $$c_{ab} = \{a, b, t, \Delta t\} \tag{3.2}$$

  whereas a and b are the node identifiers, t the start time of the contact and Δt the contact duration.

  The set of contacts of node n during the whole simulation time is denoted by C_n:

  $$C_n = \{c_{nm}\} \tag{3.3}$$

  whereas nodes m and n are neighbors during the interval $t \in [c_{nm}.t, (c_{nm}.t + c_{nm}.\Delta t)[$.

Description of Input Data The movements of nodes are given as a collection of discrete events which describe continuous movements or pauses. Sampling the state of the simulation at pre-defined sampling times yields to the time-discrete representation of the simulation. The following information is available for each node n at the sampling time $t = t_k$:

- node position $(x_n[k], y_n[k])$
- current speed $v_n[k]$
- current direction $direction_n[k] \in [0, 2\pi]$ (only defined if node is moving)
- set of neighbors $N_n[k] = \{m \in P : n$ and m are neighbors at $t = t_k\}$

3.1.2 Definition of Mobility Metrics

- Node density:
  The node density is defined as the number of nodes within a certain area. Formally,

  $$\text{node density} = \frac{\text{nodes}}{\text{area}} \tag{3.4}$$

- Node distance:
  The distance between two nodes a and b is defined as the Euclidian distance between the two nodes. Formally,

  $$distance(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \tag{3.5}$$

- Neighbor distance:
  The neighbor distance is defined in the same way as the node distance, but we calculate it only for nodes which are neighbors.

- Node speed:
  The current speed of node n is denoted by v_n.

- Speed ratio between neighbors:
  The speed ratio between a node n and its neighbor k is calculated as follows:

  $$v_{ratio}(n, k) = \frac{min(v_n, v_k)}{max(v_n, v_k)} \tag{3.6}$$

- Spatial dependence between neighbors:
  The spatial dependence [8] between two nodes a and b is a measure of the similarity of the velocity of the two nodes. Formally,

  $$\text{spatial dependence} = cos(direction_a - direction_b) * v_{ratio}(a, b) \tag{3.7}$$

  The spatial dependence is only defined if both nodes are moving. A high value of spatial dependence results if both nodes are moving in the same direction with equal speed.

- Neighbor contact duration:
  The contact duration $t_{contact}$ is defined as the time interval from the moment when the two nodes n and m are in communication range until they leave their communication range:

  $$t_{contact}(n, m) = c_{nm}.\Delta t \tag{3.8}$$

- Neighbor inter-contact duration:
  The paired inter-contact duration of two nodes n and m is defined by the time interval between the end of a contact $c_{nm}[i]$ and the beginning of the subsequent contact $c_{nm}[i+1]$.

  $$t_{inter-contact}(n, m) = c_{nm}[i + 1].t - (c_{nm}[i].t + c_{nm}[i].\Delta t) \tag{3.9}$$

3.2 Network Graph Metrics

A wireless ad-hoc network can be modeled as graph $G =< V, E >$ containing the set of network nodes $V = \{1 \ldots n\}$ and the set of edges $E \subseteq V \times V$ between the nodes. An edge in the network graph indicates a wireless link between the corresponding nodes. Two nodes which are not direct neighbors can reach each other only if there exists one or more paths between the two nodes. Figure 3.2 shows a simple example of a network graph, a more complex graph of a network with 200 nodes is shown in Figure 3.3.

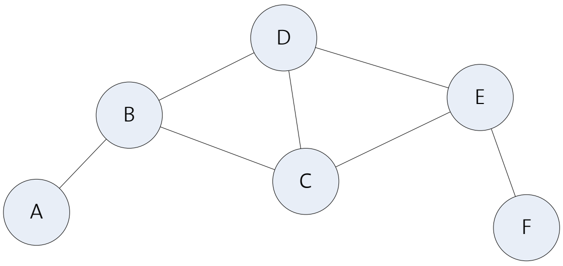


Figure 3.2: Network topology graph.

3.2.1 Definition of Network Graph Metrics

- Number of neighbors:
  The number of direct neighbors of network node n is given by the degree of node n in the network graph.

- Graph topology changes:
  The network topology (and also the network graph) is constantly changing due to the mobility of nodes. Two type of changes are possible for each node between subsequent sampling points:

  - A node has a new wireless link (edge) since the last sample time.
  - The link to another node has been lost since the last sample time.

  The topology change metric is defined as the sum of the number of gained and lost neighbors during the last sampling interval.

- Number of reachable node pairs:
  A pair of nodes can exchange data packets only if there exists a path in the network graph between the two nodes. This metric calculates the shortest paths for all possible sender-receiver combinations and counts the number of reachable pairs (normalized regarding to the total node pairs). In a simulation environment with a population of N nodes there exist exactly $\frac{N \cdot (N-1)}{2}$ node pairs.

$$\text{Reachable node pairs} = \frac{\sum_{i \in P} \sum_{k \in P, k \neq i} path(i,k)}{N \cdot (N-1)} \tag{3.10}$$

  whereas $path(i, k)$ is defined as follows:

$$path(i,k) = \begin{cases} 1 & \text{if there exists a path between node } i \text{ and node } k \\ 0 & \text{else} \end{cases} \tag{3.11}$$

- Average path length:
  Calculates the average number of hops on the shortest path between all reachable node pairs in the network graph.

3.3 Routing Protocol Performance Metrics

In this section, we present metrics for the performance analysis of routing protocols. Routing protocols are used to establish a network path from a source to a destination node and then forward packets on this path towards the destination node. The performance of a routing protocol can be analyzed using two different types of metrics. The first type of metrics evaluates the efficiency of the routing protocol (e.g. CPU cycles, memory usage, bandwidth or energy consumption). The second type of metrics evaluates the performance of the routing service offered to higher layer protocols. In this thesis, the focus lies on this second category of metrics for the evaluation of routing protocol performance.

3.3.1 Definition of Routing Protocol Metrics

- Packet Delivery Ratio:
  The packet delivery ratio (PDR) is defined as the number of data packets delivered relative to the number of packets generated. Formally,

$$\text{PDR} = \frac{\text{packets generated}}{\text{packets delivered}} \tag{3.12}$$

- Throughput:
  The throughput of a connection between two nodes is measured as the number of bytes delivered per time unit. Formally,

$$\text{Throughput} = \frac{\text{Total bytes received}}{\text{Total time}} \tag{3.13}$$

  The available bandwidth in the network and the protocol overhead influence the overall throughput in the network.

- End-to-End Packet Delay:
  The end-to-end packet delay is calculated as the time interval when the packet is generated and ready for the transmission until it is delivered to the receiving application at the destination node.

- Routing Protocol Overhead:
  The routing protocol generates a certain amount of routing packets in order to calculate paths and update the network topology information. Depending on the routing protocol used and the number of changes in the network topology due to mobility, routing packets can be accountable for a large part of the network traffic. The overhead introduced by the routing protocol can be calculated as the ratio between the number of routing packets sent and the total number of packets sent. Formally,

$$\text{Routing Protocol Overhead} = \frac{\text{Number of routing packets sent}}{\text{Number of packets sent}} \tag{3.14}$$

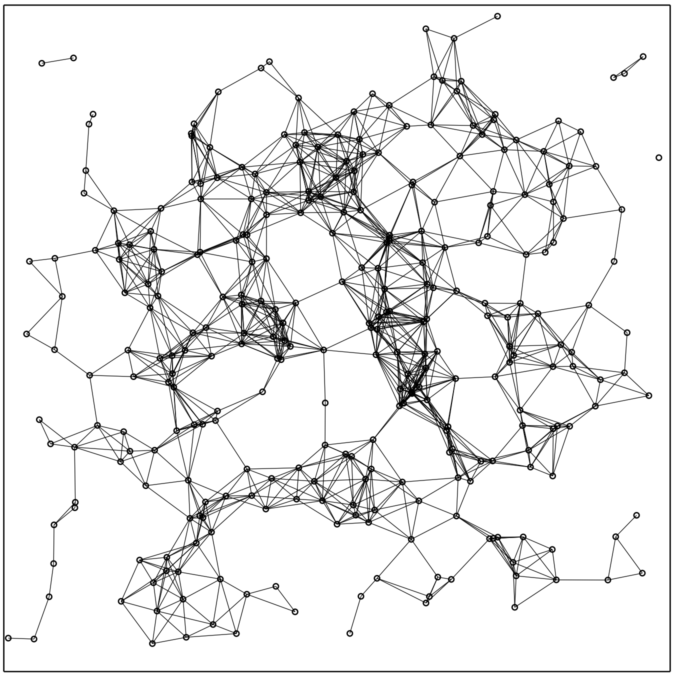


Figure 3.3: Network topology graph of a network with 200 network nodes.

4
Generic Mobility Simulation Framework

This chapter describes the design and implementation of our generic framework dedicated for the simulation of mobility in wireless ad-hoc networks. In the first section, we present the design of our framework. Furthermore, we discuss some aspects of the actual implementation to give a starting point for users who want to extend the simulator with their own mobility models or modules.

4.1 Design

The Generic Mobility Simulation Framework (GMSF) is a tool to simulate and analyze node mobility in wireless networks. It is designed to work together with commonly used network simulators. Figure 4.1 shows the basic components of the framework.

The simulation of a wireless network requires the exact position of mobile nodes during a simulation period. For this purpose, GMSF allows generating mobility traces using any of its implemented mobility models. These traces can then be exported to various trace formats. GMSF also allows defining data communication patters which can be exported. In order to achieve realistic simulations, GMSF provides several radio propagation models. Eventually, GMSF allows to analyze mobility traces and the network topology using a set of provided metrics. GMSF is designed to ease the further development of additional mobility models or modules extending the framework with additional functionalities.

4.1.1 Framework Architecture

GMSF relies at its core on a *Simulation Runtime Controller* and a *Data Storage Manager*. Besides, it is based on modules which are responsible for specific tasks (e.g. definition of node positions, output formatting, visualization).

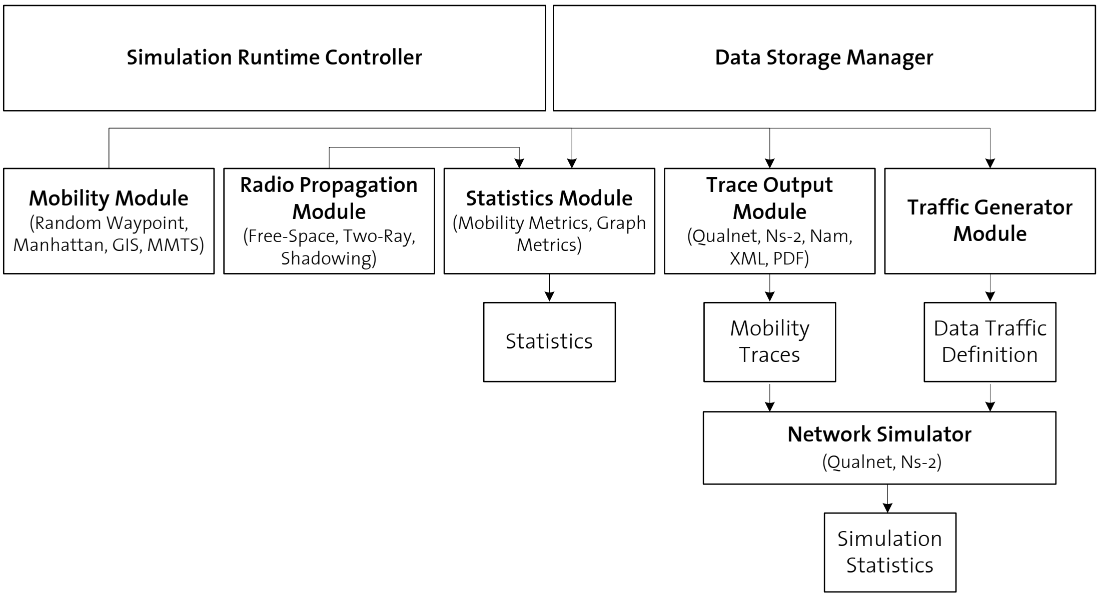


Figure 4.1: Overview of GMSF.

4.1.2 Simulation Runtime Controller

The operation of GMSF is controlled by the *Simulation Runtime Controller*. It is responsible for the initialization of all enabled modules and the scheduling of tasks provided by these modules.

Realistic behavior of nodes in road traffic depends on the interaction with other nodes. Therefore, microscopic mobility models have to take into account the current position, speed and moving direction of other nodes to calculate the movements of nodes during the current sampling interval. This makes it necessary to split the simulation into small time-steps and let the mobility module update the node position at each of these time points[1]. Therefore, the runtime controller splits the simulation period into sampling intervals of a specified duration Δt (usually 1 second) (see Figure 4.2).

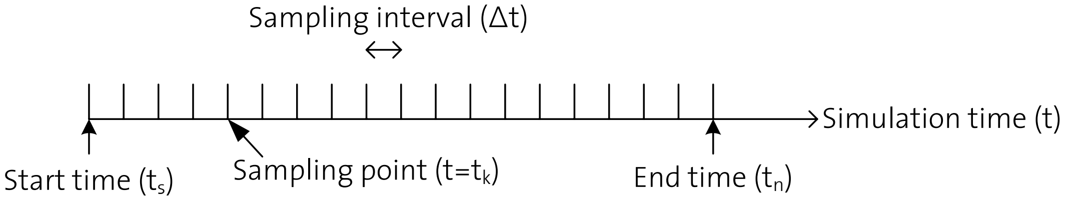


Figure 4.2: Sampling points during the simulation period.

Then, the simulation is performed in the following way:

At the beginning of a simulation run, the runtime controller initializes the mobility model and additional modules. The mobility model determines the initial position of all participating nodes. For each sampling time t_k, the simulation time t is updated and the runtime controller performs

[1] This approach would not be necessary for entity-based mobility models since node movement is completely independent.

the following operations:

1. Requesting the employed mobility model to update the positions of nodes according to the current simulation time
2. Invoking additional modules to let them perform their tasks

At the end time $t = t_n$, the mobility model and all modules are requested to perform final tasks and clean-up afterwards.

Algorithm 1 shows the pseudocode representation of the runtime controller's main loop.

```
Initialize simulation time
time = startTime ;
Initialize mobility module
mobilityModule.init();
Initialize all other modules
foreach module do
    module.init();
end
while time $\leq$ endTime do
    Update modules to the current simulation time
    mobilityModule.next();
    foreach module do
        module.next();
    end
    time = time + timeStep;
end
Finish simulation and clean-up
mobilityModule.finish();
foreach module do
    module.finish();
end
```

Algorithm 1: Pseudocode of the main loop in the simulator container.

4.1.3 Data Storage Manager

The *Data Storage Manager* of GMSF provides modules with access to simulation relevant data (e.g. configuration parameters). It keeps an up-to-date list of all nodes currently in the simulation area. Furthermore, it maintains a log of all mobility events generated throughout the simulation period.

4.1.4 Modules Overview

GMSF is designed as a generic and extensible framework for the simulation of node mobility. A set of configuration parameters allows users to define which modules should be activated and to set the module parameters. The mobility module and the radio propagation module provide node mobility and radio propagation and are hence mandatory modules of GMSF. Additional

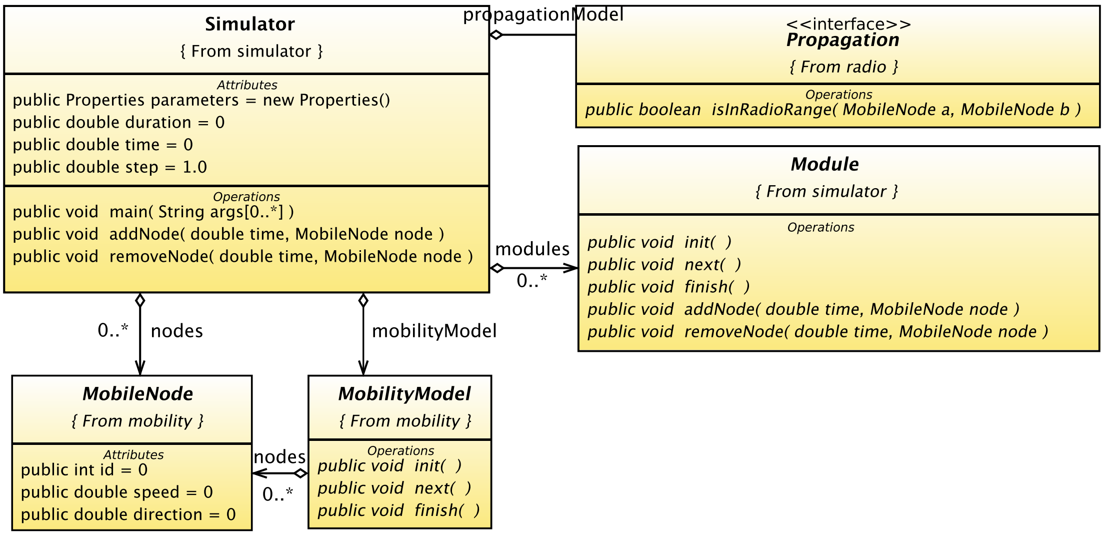


Figure 4.3: UML class diagram of the simulator framework (simplified for the sake of presentation).

functionalities can be provided by adding pluggable modules to the simulator container. Figure 4.3 shows the (simplified) UML class diagram of the simulator container and additional modules. In this section, we present the following modules part of GMSF:

- Mobility module
- Radio propagation module
- Statistics module
- Output module for mobility traces
- Output module for data traffic definition file

Mobility Module The Mobility Module is a core module of GMSF. It is responsible for the dynamic instantiation of mobile nodes and modeling of realistic mobility behavior. Node mobility is modeled by mobility events which are bounded to a specific simulation time and have assigned a certain duration. Entity-based mobility models (e.g. the Random Waypoint model) employ events which start and end at arbitrary time during the simulation. Microscopic mobility models (e.g. GIS Model with car-following), in contrast, generate only events which define node mobility within the current sampling interval, since the next movement of a node also depends on the position of other nodes at the next sampling time.

Each node has its own queue of mobility events. Mobility events are generated by the mobility model and inserted in the event queue of the corresponding node. The node processes all its pending mobility events to update its current position in the simulation area[2].

[2] In fact, only the position of a node at the sampling points is relevant since this position is used by microscopic mobility models and when calculating mobility-related and graph-related metrics.

Currently, four different event types are used to describe node mobility: `JOIN`, `LEAVE`, `MOVE` and `PAUSE`.

A `JOIN` event is issued when a node joins the simulation. The following attributes are associated to this type of event:

- Time when the node joins the simulation
- Initial node position in the simulation area

A `LEAVE` event is issued when a node leaves the simulation area. This event type has the following attributes:

- Time when the node leaves the simulation
- Last node position in the simulation area

When issuing `JOIN` or `LEAVE` events, the mobility model informs the simulator container that a node has joined or left the simulation. The container updates its list of participating nodes and passes this information to all active modules which can then perform updates to their internal data structures (if necessary).

A `MOVE` event is issued by the mobility model to update a moving node's position. A movement is the displacement of a node performed during a certain time interval (e.g. between a simulation sampling interval) and defines a translation with constant speed on the direct line between from the start point to the destination point (see Figure 4.4). The following attributes describes a `MOVE` event:

- Time $t1$ when node starts to move
- Start position $(x1, y1)$ of the movement
- End position $(x2, y2)$ of the movement
- Time $t2$ when node reaches the end point

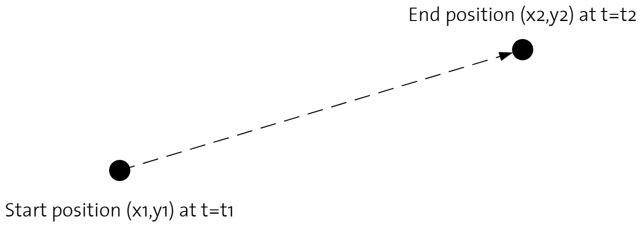


Figure 4.4: Model of a movement as a transition between two points.

A `PAUSE` event is issued when a node stays for a longer time at the same position (e.g. waiting in the queue in front of a traffic light). The following attributes are associated with a `PAUSE` event:

- Time when the pause starts
- Position of the node during the pause

- Duration of the pause

Currently, GMSF contains the implementation of the following mobility models: Random Waypoint model, Manhattan model, GIS model and MMTS model.

Radio Propagation Module The Radio Propagation module is a core module of the simulator. It is used to determine the communication range of mobile nodes. Two nodes are in communication range if one node can transmit a radio signal and the power of the received signal at the other node is above the receiver sensitivity threshold. This calculation is performed by employing a radio propagation model (see Section 2.2) to estimate the path loss and fading variations as a function of the distance between the two nodes. It is assumed that the communication channel is symmetric: i.e. if node A can receive node B's signal, the same is also true for signals in the opposite direction.
The current implementation of GMSF contains the Free-Space model, the Two-Ray model and the Shadowing model.

Statistics Module The Statistics module is used to analyze the mobility behavior of nodes using the metrics presented in Chapter 3. This module has access to all attributes of mobile nodes currently in the simulation area. Two type of metrics are included: Mobility metrics which operate only on the current position, speed and moving direction of a node. Network related metrics operate on the network graph which is built by employing the radio propagation module to determine communication ranges of nodes. The Statistics module collects and outputs statistical data for different metrics.

Mobility Traces Output Module The Mobility Traces Output module exports the position information of mobile nodes to a specific trace format. GMSF can export mobility traces for ns-2, nam and Qualnet. Additionally, traces can be exported to a generic XML format and to a PDF document.

Traffic Generator Module The Traffic Generator module generates a file defining the data traffic which should be transmitted between network nodes in the simulation. Data traffic defined by this module depends on the designated application scenario and is usually generated according to a specific traffic pattern, e.g. packets are only sent between a fixed number of source-destination pairs with a constant data rate.

4.1.5 User Interface

The simulator container accepts configuration parameters as arguments on the command line. Since no graphical user interface is required to specify simulation parameters, a simulation can be started directly from scripts or external applications. This makes it possible to link GMSF to a publicly accessible web interface.

A user willing to use the framework to generate mobility traces may visit our service website. This website is dynamically generated by a PHP [40] script and offers a form where the user can specify his simulation parameters. After the user has submitted the form, the script on the web server adds a new job entry to the database. This entry contains the parameter set to run the simulation in the framework (e.g. the mobility model, the radio propagation model, and their associated parameters). A job scheduler, which is completely decoupled from the web server, periodically checks the database for unprocessed jobs. If a new job is found, the scheduler removes the entry from the database and starts a new instance of the simulator using the specified parameters. Upon completion of the simulation, the job scheduler informs the user by an e-mail that the job has been completed and specifies the download location of the resulting files. Figure 4.5 shows the interconnection between the simulator framework and the web-based interface.

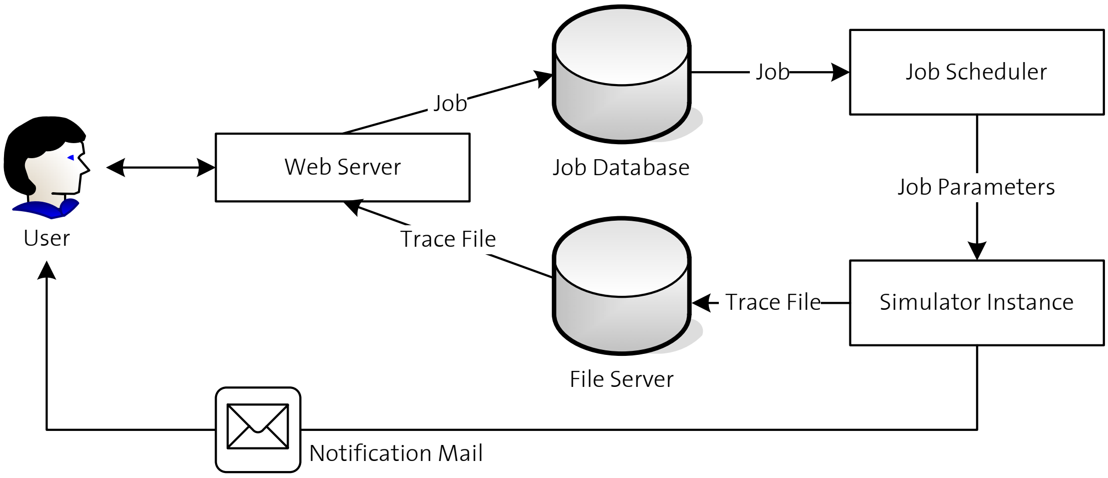


Figure 4.5: Operation mode of the web interface to the simulation framework.

4.2 Implementation

In this section, we highlight implementation details of the framework. A complete overview of the implementation details is provided in the documentation delivered with GMSF.

4.2.1 Supported Platforms

The simulation framework is implemented in the Java programming language [41]. Java has been chosen because it is available on different platforms (currently on Windows, Mac OS, Solaris and Linux) which makes the simulator portable. Java version 6 is required to compile and run the software.

4.2.2 Extending the Framework

Additional modules can be added to the simulator framework to extend its functionality. The simulator requires at least the presence of a mobility module and a radio propagation module, all other modules are optional. A module has to follow a certain structure to be able to interact with other modules. Module implementations have to extend the abstract base class `simulator.Module` (see Listing 4.1). A module is activated in the framework by adding it to the list of modules in the simulator container (`simulator.Simulator`):

```
// adds new module to the simulator
modules.add(new ExampleModule());
```

Mobility Modules Mobility modules are handled in a special way since they are mandatory core modules of the framework. Operating the framework without a module controlling the position of nodes makes not much sense. Mobility modules have to implement the abstract class `mobility.MobilityModel` (see Listing 4.2). Mobile nodes managed by a certain mobility model have to extend the abstract base class `mobility.MobileNode`.

Currently, the following mobility models are implemented as modules in the simulation framework:

- Random Waypoint model (`mobility.rwp.RandomWaypointModel`)
- Manhattan model (`mobility.mn.ManhattanModel`)
- GIS model (`mobility.gis.GISModel`)
- MMTS model (`mobility.mmts.MMTSModel`)
- Fixed node model (`mobility.fixed.FixedModel`)
- Test model (for testing) (`mobility.test.TestModel`)

Radio Propagation Modules All radio propagation modules have to implement the `radio.Propagation` interface (see Listing 4.3). The `isInRadioRange()` method is called by other modules to determine if two mobile nodes are in each others communication range. References of the nodes are passed as the method's input parameter. Normally, the node position is used to estimate the path loss based on the Euclidian distance between the two nodes. Alternatively, the node positions can be used to calculate the path loss based on the environment (e.g. taking buildings into account).
Currently, the following model for radio propagation are implemented in the simulation framework:

- Free-space model (`radio.FreeSpacePropagation`)
- Two-ray model (`radio.TwoRayGroundPropagation`)
- Shadowing model (`radio.ShadowingModelPropagation`)
- Fixed communication range model[3] (`radio.FixedRangePropagation`)

Traces Output Modules Output modules for mobility traces have to extend the abstract class `output.TraceFormatter`. A trace output module has access to all mobility events generated by the mobility model describing the exact movements of mobile nodes. This information has to be written to a file according to the definition of the trace format (see Appendix C for a description of common mobility trace formats).
Currently, output modules for the following trace formats are implemented:

- ns-2 mobility trace format (`output.NS2Formatter`)
- nam mobility trace format (`output.NAMFormatter`)
- Qualnet mobility trace format (`output.QualnetFormatter`)
- XML mobility trace format (`output.XMLFormatter`)
- PDF containing mobility traces as lines (`output.PDFFormatter`)

[3] This model can be used when one wants to model a fixed communication range independent on radio parameters, path loss or fading effects.

Data Traffic Generator Modules A traffic module defines communication patterns between mobile nodes participating in the simulation. The current implementation contains the `traffic.CBRTrafficGenerator` class which generates data traffic with constant data rate (CBR) between a specific number of source-destination pairs.

Visualization Module A graphical interface to the simulation framework is implemented in the `gui.GUI` module which visualizes both the current position of nodes in the simulation area and the network graph. Snapshots of the current network topology can be exported to PDF documents. Figure 4.6 shows a screenshot of the visualization module.

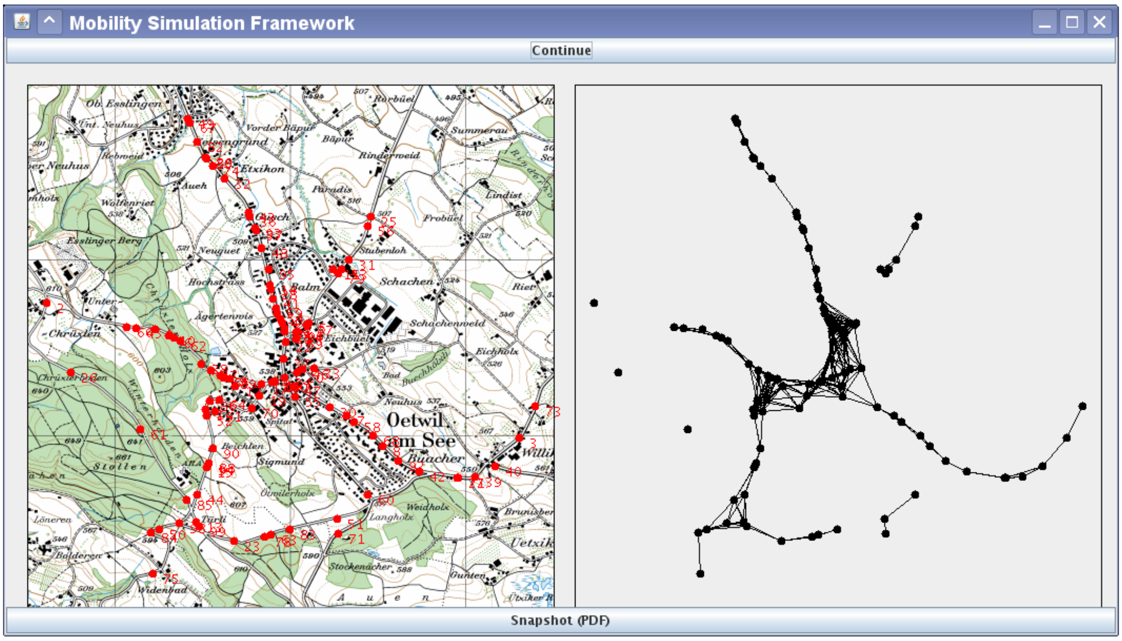


Figure 4.6: Screenshot visualizing node positions (left side) and network topology (right side).

```
 public abstract class Module {

  /** Initializes the module */
  public abstract void init();

  /** Lets the module execute the next time step in the
     simulation.*/
  public abstract void next();

  /** Finishes the module after a completed simulation run. */
  public abstract void finish();

  /** Notifies the module that a new node has joined the
     simulation
   *  @param time time when node joins the simulation
   *  @param node Node which joins the simulation */
  public abstract void addNode(double time, MobileNode node);

  /** Notifies the module that a node has left the simulation
   *  @param time time when node leaves the simulation
   *  @param node Node which leaves the simulation */
  public abstract void removeNode(double time, MobileNode node)
     ;
}
```

Listing 4.1: Source code for the abstract class Module (extract).

```
public abstract class MobilityModel {

  /** list of mobile nodes */
  public List<MobileNode> nodes;

  /** initialization method */
  public abstract void init();
  /** update node position for the next sample point */
  public abstract void next();
  /** finish and clean-up */
  public abstract void finish();
}
```

Listing 4.2: Source code for the abstract class MobilityModel (extract).

```
public interface Propagation {
        /** Returns true if the two specified mobile nodes are
           in communication range */
        boolean isInRadioRange(MobileNode a, MobileNode b);
}
```

Listing 4.3: Source code for the interface Propagation (extract).

5 Realistic Mobility Models

In this chapter, we present one of the major contributions to this thesis: a realistic mobility model for vehicular ad-hoc networks. The aim of this new mobility model is the simulation of realistic behavior of vehicles. It uses detailed street maps from the Swiss geographic information system (GIS). In addition, we make use of realistic vehicular traces from a microscopic traffic simulator [16]. This mobility model is taken as a reference to evaluate the mobility behavior of the GIS-based model.

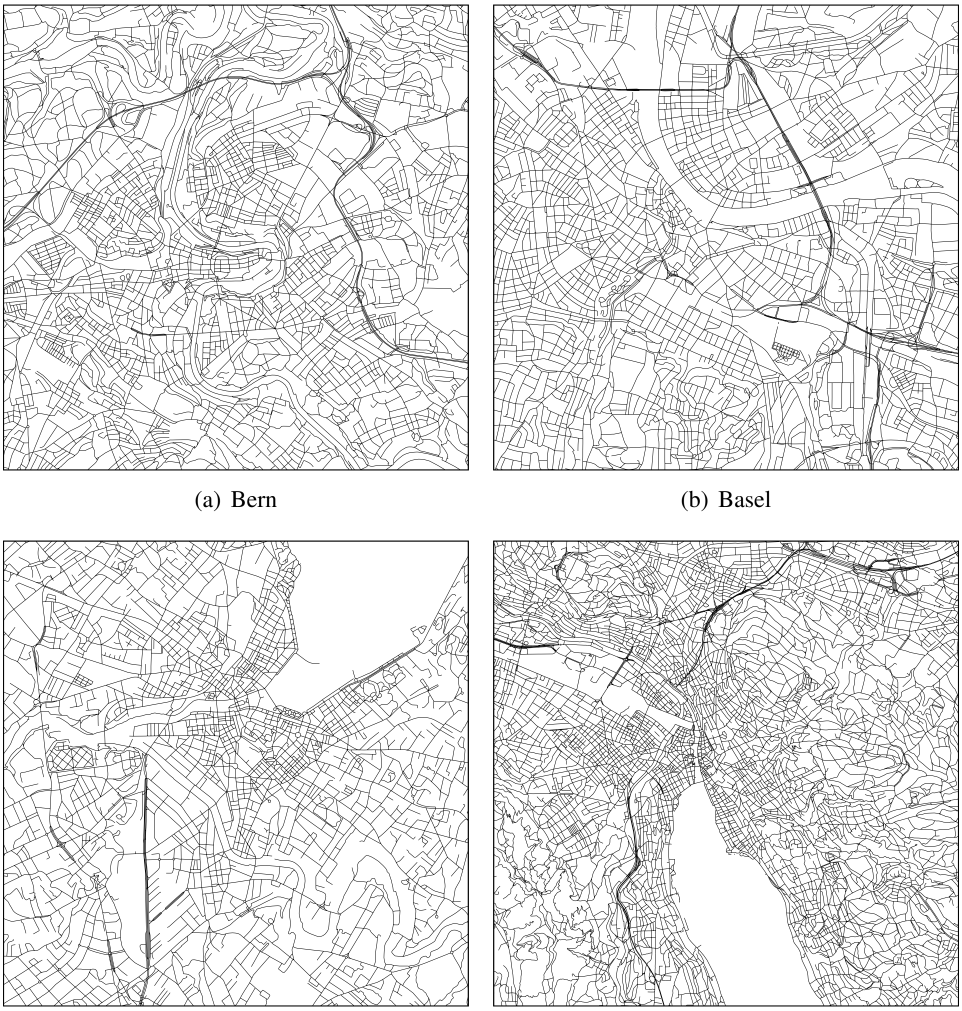
5.1 GIS-based Mobility Model

This section presents the details of the GIS-based mobility model. First, we introduce the landscape model of Switzerland which defines the road topology. Then, we present different features composing the mobility behaviors of vehicles both at the macroscopic and microscopic levels. On the macroscopic level, our mobility model considers aspects like the road topology, speed limits and collective vehicle behavior like traffic density. The influence of other vehicles or traffic lights on the driver behavior is modeled on the microscopic level.

5.1.1 Landscape Model of Switzerland

Geographic information systems are computer systems capable of storing, editing, analyzing and displaying geographically-referenced information. This technology is widely used, e.g. in research, resource management, cartography or route planning. During the last 30 years, a number of commercial GIS applications have evolved and are nowadays also used by the public (e.g. web mapping services). The Swiss Federal Office of Topography [42] offers a detailed landscape model of Switzerland (VECTOR25). The model consists of approximately 7 million different objects grouped into 9 different layers (e.g. roads, primary surfaces, buildings, facilities). Data source are the official Swiss national map products (“Landeskarte 1:25’000”).

Road Network Layer The “Road network” layer contains all roads which are pictured in the Swiss national maps (1:25’000). Figure 5.1 shows the road networks of four major Swiss cities. Roads are modeled by connected line segments (edges). Different roads crossing at the same height have a common node in the GIS data model (see Appendix E) which allows to run route searching algorithms on top of the road network graph. Each road is attributed with an unique road identifier and the road type.



(a) Bern

(b) Basel

(c) Geneva

(d) Zurich

Figure 5.1: Street maps of four Swiss cities.

Buildings Layer The "Buildings" layer in the GIS data contains the outline of buildings together with an attribute describing the building type. This layer contains only buildings which can also be found in the Swiss national maps.

Primary Surfaces Layer The layer "Primary surfaces" describes the soil coverage. The surface type can be used to determine if a given position is inside a settlement area or is an uninhabited zone. This information is useful for consideration in a radio propagation model to estimate path-loss or fading parameters (see Section 2.2).

5.1.2 Macro-Mobility Behavior

The proposed mobility model consists of modules operating at different layers to define the macroscopic behavior of vehicles (see Figure 5.2). The most important macro-mobility feature of our model is the underlying highly detailed road topology employed in the *Road Network Layer*. On top of this road topology, the *Trip Selection Layer* chooses a start and end point for trips from a set of points of interest. Based on a start and end point, the *Route Selection Layer* calculates the exact route for this trip. The different macro-mobility layers of the GIS model are discussed in the remainder of this section.

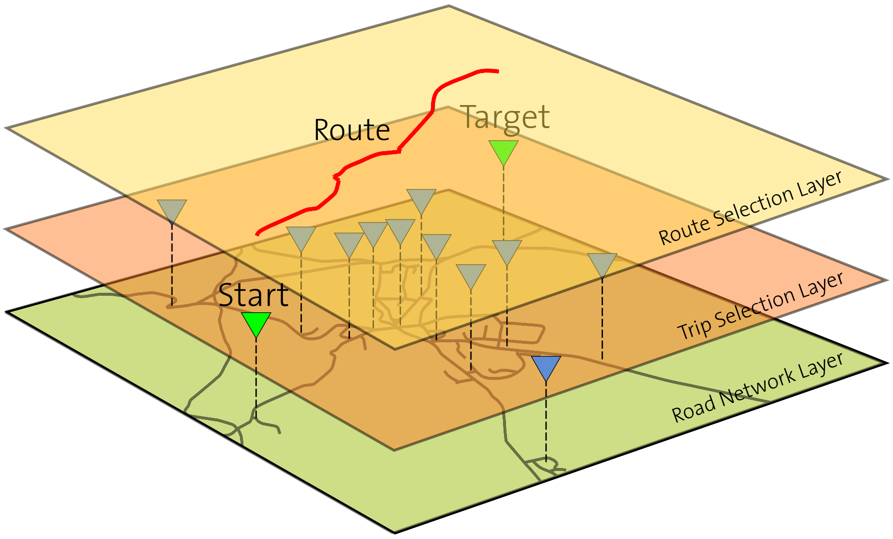


Figure 5.2: Different macro-mobility layers in the GIS model.

Road Network Layer The topology of the road network is based on the highly detailed landscape model presented in Section 5.1.1. A road is modeled as a collection of line segments following the exact course of the road rather than just connecting the start and end point of a road with a direct line. Figure 5.3 shows an example of the detail level of the employed road data. Furthermore, a speed limit is assigned to each road based on the road category.

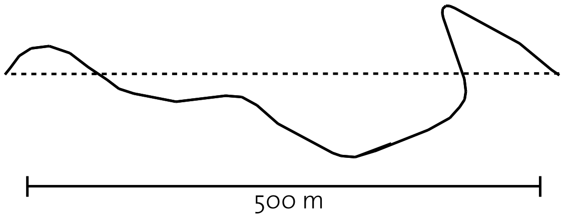


Figure 5.3: Example showing the detail level of the road topology.

The road topology for a specified simulation area is extracted from the GIS landscape model using a script (see Appendix B) and converted to a text file. Roads are defined in a simple text format (see Listing 5.1), which makes it also easy to import user-defined or third-party topologies.

```
<Road>
# roadid roadtype x1 y1 x2 y2
# first road segment
00012345 05 1515.200000 1247.000000 1519.000000 1256.400000
# next road segment
00012345 05 1519.000000 1256.400000 1525.500000 1263.900000
# next road segment
00012345 05 1525.500000 1263.900000 1532.300000 1268.400000
# next road segment
00012345 05 1532.300000 1268.400000 1539.000000 1269.900000
</Road>
```

Listing 5.1: Road data format (sample).

Trip Selection Layer People do not use their car to travel between random points on a map, instead they have an exact plan what they want to do (e.g. travel from home to work, go shopping or do leisure activities). The trip selection layer operates on a set of points of interest to generate movement patterns which are mapped to exact routes by the route selection module. Points of interest can be taken from one of the following sources:

- Imported from a text file defining the exact coordinates of point of interests relative to the simulation area.
- Road intersections from the road topology as point of interests.
- Buildings from the GIS data as points of interest (not yet implemented).

The *Trip Selection Layer* randomly chooses a source and destination point among the points of interest and invokes the *Route Selection Module* to calculate a path for the movement of vehicles. Upon arrival at the destination, a next destination is selected and the *Route Selection Module* invoked, again.

Route Selection Layer The route selection layer is responsible to calculate a path in the road topology graph based on the source and destination point obtained from the *Trip Generation Layer*. The routing task can be performed using different algorithms for the path calculation:

- *Shortest path algorithm*: Set the edge weight (costs) to the length of the road and execute Dijkstra's shortest path algorithm.
- *Fastest path algorithm*: Set the edge weight (costs) to the time it takes to pass through the road with the maximal allowed speed.
- *Real-time fastest path algorithm*: Same as the fastest path algorithm, but take the average speed of vehicles currently on this road to calculate the time it takes to pass through the road instead of applying the maximal allowed speed. This approach takes into account the current traffic load on each road for the path calculation. This algorithm is currently not implemented in the mobility model.

5.1.3 Micro-Mobility Behavior

The micro-mobility module defines the exact speed and acceleration of vehicles in the simulation. Depending on the desired level of detail, one of the following micro-mobility models is employed to simulate the microscopic behavior of vehicles:

- Entity model
- Car-following model
- Car-following with traffic lights

Entity Model In the entity-based micro-mobility model, vehicle speed is imposed only by the speed limit of the current road ignoring any other vehicles in the proximity. If the current travel speed is below the speed limit, the vehicle smoothly accelerates until it reaches the maximal allowed speed. If it enters a road with a lower speed limit than its current speed, the vehicle smoothly decelerates. Such a model is easy to implement, but fails to reproduce realistic traffic effects like traffic congestion or queuing in front of traffic lights.

Car-Following Model The Intelligent-Driver Model [43] belongs to the category of car-following models. The speed of a vehicle in the next simulation step depends on the current speed v, the desired speed v_0 and on the distance to the front vehicle.

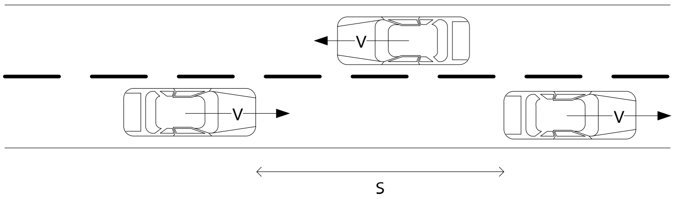


Figure 5.4: Car-following in the Intelligent-Driver Model.

The following equation defines the speed change for a vehicle between subsequent simulation steps:

$$\frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*}{s}\right)^2\right] \tag{5.1}$$

A desired dynamical safety distance s^* to the front vehicle is maintained which depends on the current speed, the speed difference to the front vehicle Δv, a driver's reaction time T and the minimal acceptable distance between two vehicles s_0. Formally,

$$s^* = s_0 + \left(vT + \frac{v\Delta v}{2\sqrt{ab}}\right) \tag{5.2}$$

The acceleration in Equation 5.1 is divided into two parts: a "Free-road" term and a "Breaking" term. The Free-road term is used to accelerate the vehicle until the speed limit is reached. The Breaking term restricts the speed to maintain a safety distance to the front vehicle.
Table 5.1 shows the parameters of the Intelligent-Driver Model.

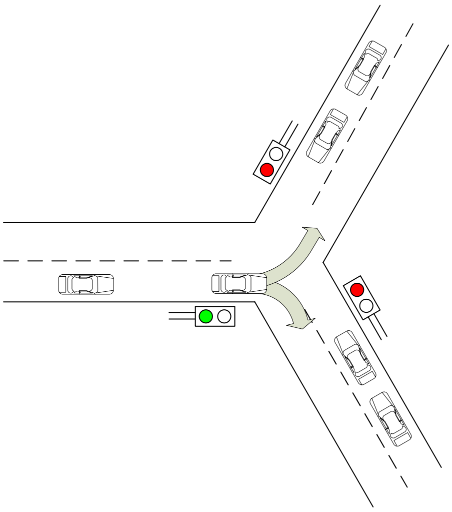
Parameter	Description	Value
v_0	Desired speed	speed limit
a	Acceleration constant	0.6
b	Deceleration constant	0.9
T	Reaction time	0.5 s
s_0	Minimal gap between vehicles	1 m

Table 5.1: Parameters for the Intelligent-Driver Model.

Car-Following Model with Traffic Lights The Intelligent-Driver Model is extended to simulate the behavior of vehicles at intersections. The foremost vehicle on each road checks if the intersection at the end of the road is free to pass. If the intersection is free, no additional action is performed and the vehicle acceleration is calculated according to Equation 5.1. In the case the traffic light regulating access to the intersection is on red, the foremost vehicle decelerates and stops in front of the traffic light. A red traffic light is modeled as a vehicle with zero speed at the position of the traffic light. Formally,

$$\begin{aligned} s &= d \\ \Delta v &= v \end{aligned} \tag{5.3}$$

where d is the distance between the vehicle and the traffic light. Other vehicles queue up behind the first vehicle until the traffic light switches to green and all vehicles start again to accelerate.



Currently, no information is available from the GIS data to determine which traffic rules apply or if there is a traffic light. Therefore, admission control to intersections is based on a set of simple rules:

- Less important intersections (small roads) are served by a first-come first-serve principle: The vehicle which has the smallest distance to the intersection is allowed to pass, other vehicles have to wait until the intersection is free.

- Important intersections are controlled by traffic lights: One road at a time has a green traffic light, all the others are red. Scheduling of the green phase is done using a round-robin algorithm where the duration of the green phase is set proportional to the road category.

5.2 Traffic Simulator-based Mobility Model

Different traffic models have been developed mainly to understand the mechanisms behind traffic. The traffic volume continues to increase which causes people to spend a large amount of time in traffic jams. Traffic models are used by the transportation community in order to study how to optimize road infrastructures. Different approaches exist to model vehicular traffic. Helbling [13] reviewed different traffic models. The most important modeling approaches for vehicular traffic are:

- Microscopic (particle-based) models
- Mesoscopic (gas-kinetic) models
- Macroscopic (fluid-dynamic) models

5.2.1 The Microscopic Multi-Agent Traffic Simulator (MMTS)

The Microscopic Multi-Agent Traffic Simulator [16] was developed at the ETH Zurich to model public and private traffic in Switzerland. MMTS simulates the behavior of a large number of agents which are modeled as intelligent individuals. Each agent performs trips according to its daily schedule (e.g. go to work early in the morning, go shopping). The simulation of a typical workday in Switzerland generates around 5 million trips. Figure 5.5 shows a simulation of vehicular traffic in a part of Switzerland.

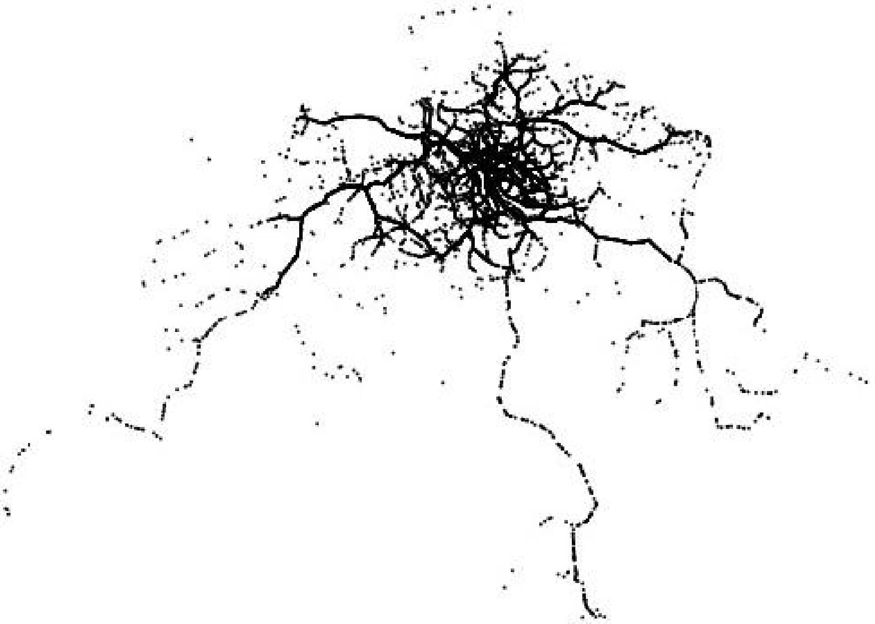


Figure 5.5: MMTS: Simulated vehicular traffic in the north-east of Switzerland.

MMTS is based on the road network of the Swiss regional planning authority which includes only major roads that are attributed with type, length, speed and capacity. The detail level of the road topology is not that high compared to the GIS data[1]. Figure 5.6 shows the different road network models for the same area in the MMTS and the GIS model.

[1] The road network used in MMTS contains only major roads.

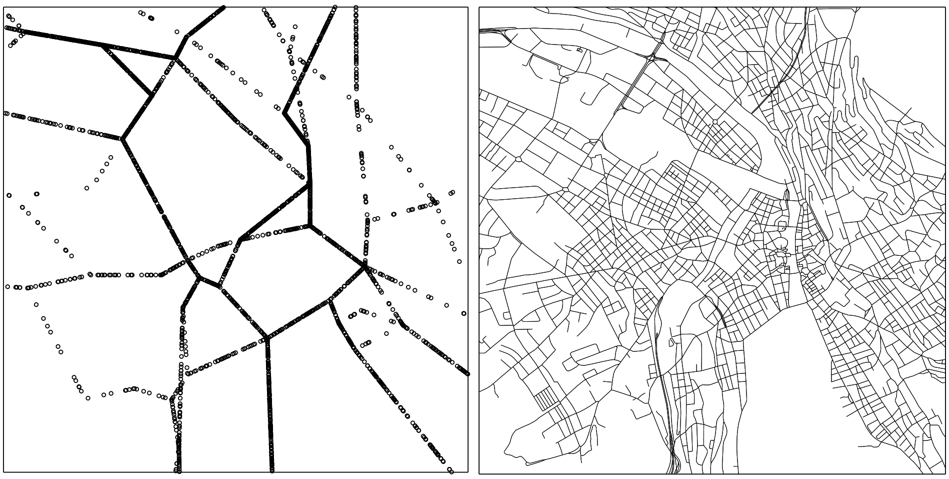


Figure 5.6: Vehicles in the MMTS model (left) and road network from GIS data (right).

5.2.2 MMTS Mobility Model

Vehicular mobility traces from MMTS are publicly available in the ns-2 mobility trace format [44]. The files contain all vehicular mobility traces in the region around Zurich during the 24-hours simulation period. In a first step, vehicular traces are converted from the ns-2 specific format to a more generic trace format. This new trace file is filtered to output traffic occurring in a defined area (i.e. simulation area) and at a specific time period.

The MMTS mobility model generates the movements of mobile nodes based on the vehicular traces from MMTS. Each mobile node in the simulation corresponds to a vehicle in the MMTS simulation.

Network Simulator Performance Workaround Since vehicles are constantly entering and leaving the simulation area, the number of unique nodes in the simulation is much larger than the average number of nodes in the simulation area. This imposes strict constraints on simulators since their performance degrades with the number of simulated nodes[2]. To avoid performance problems when simulating the MMTS mobility model with a large number of nodes, mobile nodes in the network simulator are reused[3] during the simulation phase to simulate the movement of another vehicle. When a node leaves the simulation area, the network interface is disabled which causes all packets remaining in the queue to drop. Entries in the routing table expire after a certain timeout period, since contact to neighbors is lost due to the inactive network interface. Then, the node is inserted into a pool of idle nodes and can be used to simulate the movements of another vehicle. The network interface is reactivated when the node re-enters the simulation again.

[2] Note that no simulator supports the dynamic instantiation of nodes during an on-going simulation.

[3] Nodes are only reused in the network simulator. This has no impact on the evaluation results of node mobility in the Statistics module in GMSF.

6 Evaluation

In this chapter, we compare different mobility models by statistical means using mobility-related and graph-related metrics. In addition, we investigate the influence of different mobility models on the performance of ad-hoc routing protocols (AODV and OLSR) by simulations in Qualnet. Finally, we summarize the major findings of our evaluation.

6.1 Methodology

In this section, we present the details of the evaluation process. The evaluation is performed with GMSF, our framework which was designed for this exact purpose (see Figure 6.1). In the first step of the evaluation process, we generate mobility traces for a specific simulation scenarios by employing the different mobility models. Using GMSF, we analyze these mobility traces and the corresponding network graph using the metrics presented in Chapter 3. In the following step, the ad-hoc network defined by the mobility traces is simulated in Qualnet. Finally, we evaluate the statistics obtained from the simulation.

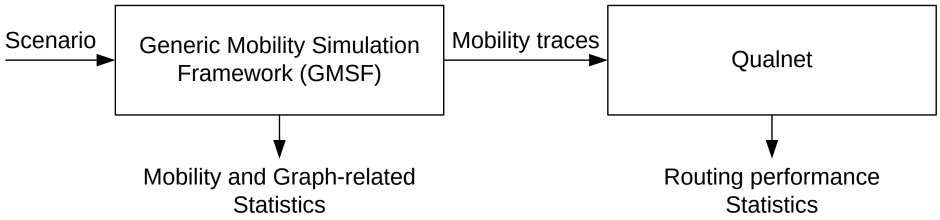

Scenario
Generic Mobility Simulation Framework (GMSF)
Mobility traces
Qualnet
Mobility and Graph-related Statistics
Routing performance Statistics

Figure 6.1: Overview of the evaluation process using GMSF and Qualnet.

When comparing mobility models, the question of their representativeness of reality is always an issue. In our case, MMTS is considered as one of the most realistic traffic simulators available. Due to the nature of MMTS, only movement of vehicles on major roads are included in the vehicular traces of the simulation scenarios.

6.1.1 Geographic Region Scenarios

In order to obtain a fair comparison of mobility models, we define three distinct scenarios for the simulation of vehicular mobility. Each scenario corresponds to a geographical area of 3,000 m × 3,000 m, see Figure 6.3. The three scenarios differ in the structure of the road network and in the population density:

- "City Scenario": This scenario is located in the downtown area of Zurich, Switzerland's largest city with a population of around 370,000 people.
- "Urban Scenario": This scenario is located at the city limit of Zurich.
- "Rural Scenario": This scenario is located in a rural area in the canton of Zurich.

Each mobility model is parameterized to reproduce the specifics of motion in the geographic area related to each scenario. We use the MMTS model as a reference for the number of nodes involved in road traffic (see Figure 6.2). Due to the nature of the MMTS model, nodes continuously leave and enter the simulation area. For each scenario, we identify a time period of 2,000 seconds where the corresponding geographical area is populated by a relatively constant number of nodes (see Table 6.1) with a high vehicle density. We are not interested in the simulation of the scenarios during time periods with a low vehicle density because then a vehicular ad-hoc network would be infeasible.

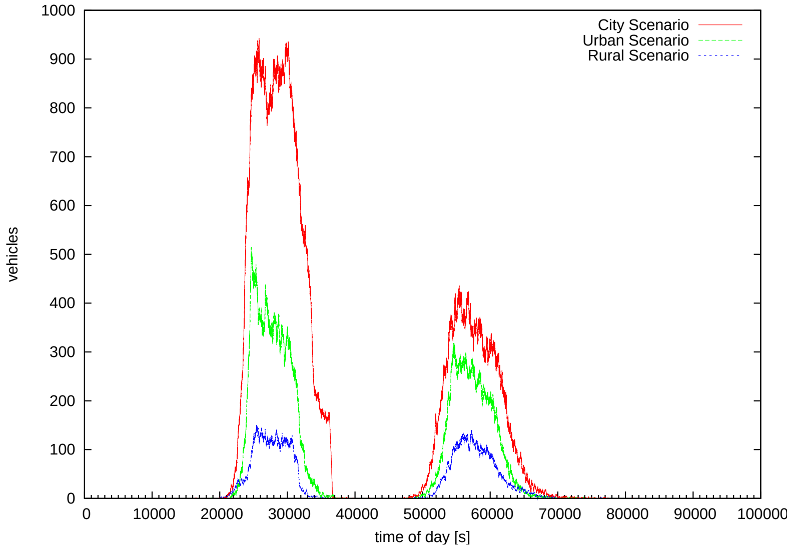
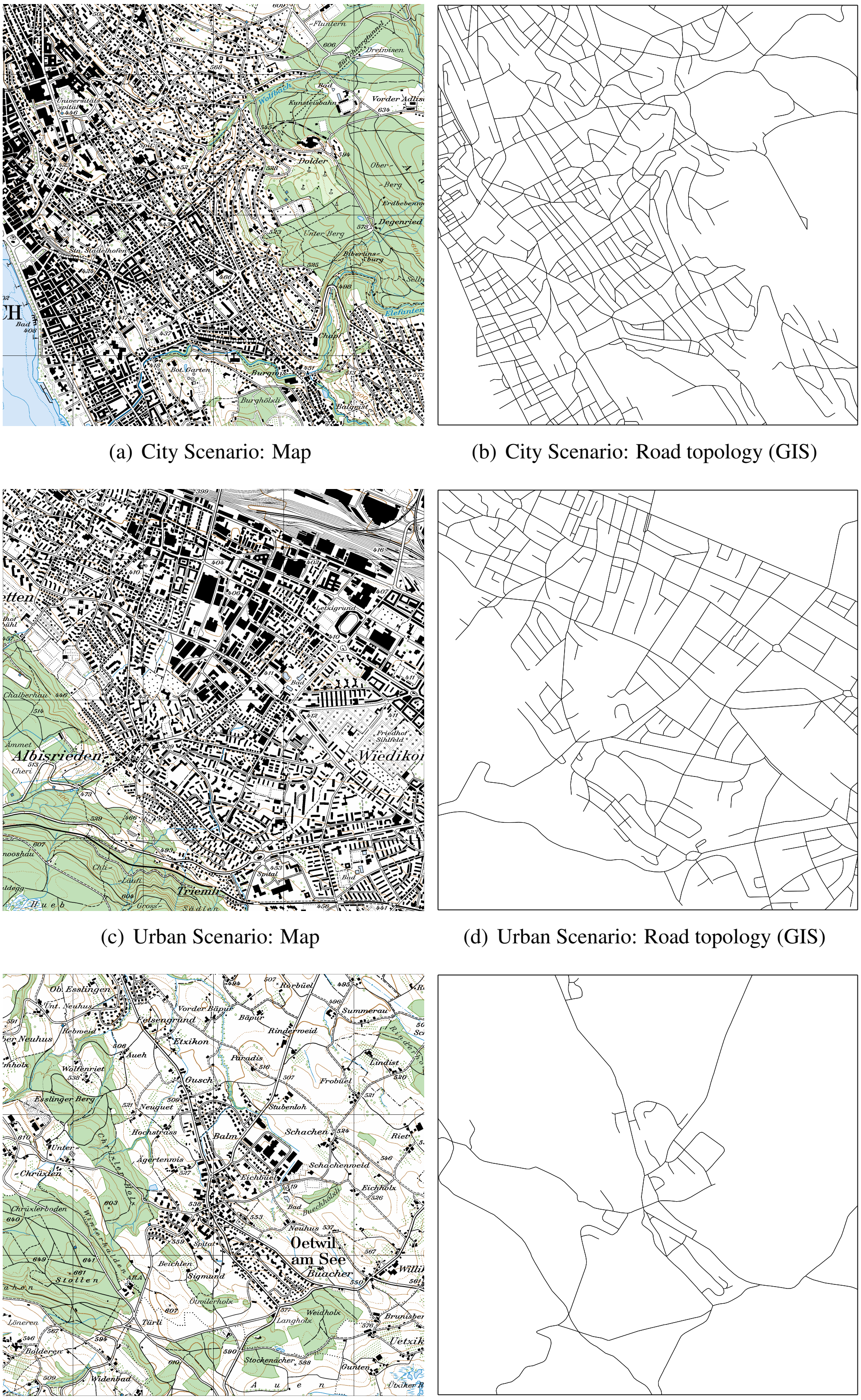


Figure 6.2: Number of vehicles in the MMTS model for the different scenarios.

The average number of nodes present in the MMTS model for the different scenarios (see Table 6.1) is used to parameterize the number of nodes simulated by the other mobility models.

Scenario	Time Period (s)	Average Vehicles	Unique Vehicles
Rural	27,000-29,000	116.84	1142
Urban	28,000-30,000	420.26	4473
City	28,000-30,000	884.46	8543

Table 6.1: Number of vehicles in the MMTS model for the different scenarios.



(a) City Scenario: Map

(b) City Scenario: Road topology (GIS)

(c) Urban Scenario: Map

(d) Urban Scenario: Road topology (GIS)

(e) Rural Scenario: Map

(f) Rural Scenario: Road topology (GIS)

Figure 6.3: Simulation scenarios selected for the evaluation. Maps are reproduced with authorization by swisstopo (BA071556).

6.1.2 Mobility Model Set-Up

In this section, we present the different mobility models employed for the evaluation. The set of parameters for each mobility model is chosen to reproduce the specific characteristics of movements in each scenario.

Using GMSF, we generate traces for all three scenarios with the following mobility models:

- Random Waypoint model
- Manhattan model
- GIS model
- MMTS model

Random Waypoint Model (RWP) The node speed in the Random Waypoint model is uniformly distributed between 9 and 16 m/s covering the speed limits of the different road categories present in the simulation scenarios. The pause time between subsequent trips is uniformly distributed between 0 and 10 seconds to simulate a short stop at the destination. The initial values for a node's position and speed are set according to the steady-state initialization method proposed by Navidi et al. [5].

Manhattan Model (MN) The road topology used for the Manhattan Model is a grid consisting of horizontal and vertical roads. The distance between two road intersections is set to 200 meters. Nodes travel at speeds between 12 and 14 m/s. The upper speed bound corresponds to the speed limit of roads within settlement areas in Switzerland. The acceleration is uniformly distributed between -0.1 m/s^2 and 0.1 m/s^2. If the distance to the front vehicle is less than 25 meters and the current speed is higher than the speed of the front vehicle, the speed is limited to the front vehicle's speed. Since there exists no analytical method to start in the steady-state, the initial 5,000 seconds of the simulation are discarded.

GIS Model (GIS, GIS-F and GIS-F-T) The road network for the three scenarios are extracted from geographical data (GIS) corresponding to the simulation area. Only roads which are accessible by vehicles are imported into the road topology of the GIS mobility model. The speed limit for each road is set according to the road category[1] (see Table 6.2).

Road Category	Speed Limit
1, 2, 3, 4	33.33 m/s
5, 6, 7	16.67 m/s
8, 9	13.89 m/s
others	8.34 m/s

Table 6.2: Speed limits for the different road categories in the GIS model.

The simulation of the GIS model is performed separately with the following three micro-mobility settings:

- GIS: Nodes perform random trips restricted to the topology defined by the road network.

[1] Road categories correspond to the values of the road type attribute in the GIS data model (see Appendix E).

- GIS-F: Nodes perform random trips on the road network graph but adapt their speed to maintain a safety distance to the front vehicle (car-following model).
- GIS-F-T: Same behavior as in GIS-F, additionally, nodes take traffic lights into account.

Since there exists no analytical method to start in the steady-state, we discard the initial 5,000 seconds of the simulation.

MMTS Model (MMTS) Vehicular traces from MMTS are available in a single trace file [44] containing all nodes in the greater Zurich area. For the three selected scenarios, we extract only the traces of vehicles that are inside or traverse the simulation area during the specified time period. The movement of nodes in the MMTS model is generated according to the extracted vehicular traces.

6.1.3 Network Simulator Set-Up

The simulations of the vehicular ad-hoc networks are performed using the Qualnet 4.0 network simulator [38] (see Figure 6.4). We discuss the important simulation parameters for Qualnet. The complete configuration parameters used in the Qualnet simulations are listed in Appendix D.

Physical Layer Settings Mobile nodes in the simulation are equipped with wireless devices following the IEEE 802.11b standard [45]. Table 6.3 shows the parameters of the physical layer model in Qualnet. All devices operate on the same wireless channel.

Parameter	Value
Standard	IEEE 802.11b
Frequency	2.4 GHz
Data rate	11 Mbps
Transmission power (11 Mbps)	15 dBm
Receiver sensitivity	-83.0 dBm
Antenna type	omnidirectional

Table 6.3: Qualnet physical layer settings.

Radio Propagation Model In order to have a realistic model for signal propagation, we use a combination of the Two-Ray path loss model with an additional shadowing loss of 6 dB. The parameter settings for the physical layer and the employed propagation model result in a communication range of about 250 meters.

Media Access Control Layer We use Qualnet's implementation of the distributed coordination function (DCF) which employs Request-to-Send (RTS) and Clear-to-Send (CTS) packets for channel reservation.

Routing Protocols We investigate the performance of two popular routing protocols for wireless ad-hoc networks: AODV and OLSR. Simulations were performed with Qualnet's default settings for both routing protocols.

Communication Scenario The main goal of the ad-hoc network simulation in Qualnet is to investigate the influence of node mobility on the routing protocol performance. We assume that only a few randomly selected source-destination pairs communicate over a limited time period with a low data rate. This application scenario corresponds to a transmission of short information messages or a small data file where no high throughput is required.

Each source node generates 200 data packets of size 512 bytes during a period of 100 seconds. The time interval between the generation of two successive packets is set to 0.5 seconds. The transmission of the first data packet is started 50 seconds after the beginning of the simulation. Subsequent source-destination pairs start the transmission of packets shifted by a short time interval to avoid concurrent initiation of a large number of route requests when using AODV. The maximal number of concurrently active source-destination pairs remains constant (see Table 6.4). The simulation is carried on for additional 50 seconds after the last data packet is generated. Data traffic between pairs is generated using Qualnet's constant bit-rate (CBR) traffic generator which employs the UDP transport protocol for the data packets.

Scenario	Nodes	Pairs
Rural	117	10
Urban	420	15
City	884	20

Table 6.4: Number of concurrently active source-destination pairs.

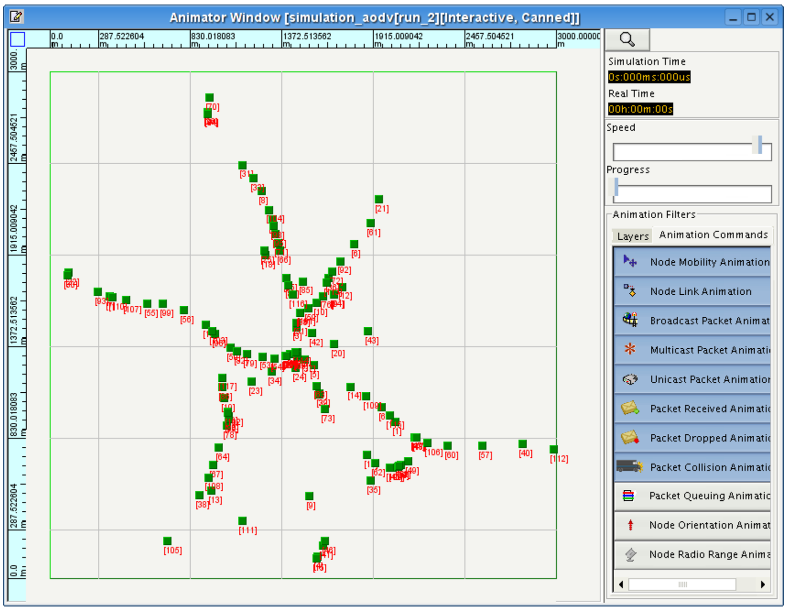


Figure 6.4: Screenshot of the Qualnet 4.0 network simulator.

6.2 Evaluation of Mobility-Related Metrics

In this section, we present simulation results for the mobility-related metrics presented in Section 3.1. This evaluation focuses on a selection of important simulation results. The complete simulation results are contained in Appendix A.

6.2.1 Node Density

The distribution of nodes in the simulation area highly depends on the topology restrictions imposed by the mobility model. To measure the average node density, we split the simulation area in 50 m×50 m unit squares. Then, we measure the average number of nodes inside each square throughout the simulation period. The measurement results of the node densities for the Urban Scenario are shown in Figure 6.5 as two-dimensional histograms.

Random Waypoint Model No regional restrictions are imposed on the node mobility since the movement of nodes is completely random and not limited to a road topology. Figure 6.5(a) shows the node density for the Random Waypoint model. Nodes can be observed in almost every unit square (99.8% of squares covered) during the simulation period. Remarkable is the higher node density in the center of the simulation area. This result is consistent with the observations measured by Navidi et al.[5] and corresponds to the steady-state distribution of nodes in the Random Waypoint model.

Manhattan Model The results show that the movement of nodes is restricted to the grid-like road network. Nodes can be observed with equal propability within all road segments. The node density is twice as high at positions of road intersections (see Figure 6.5(b)).

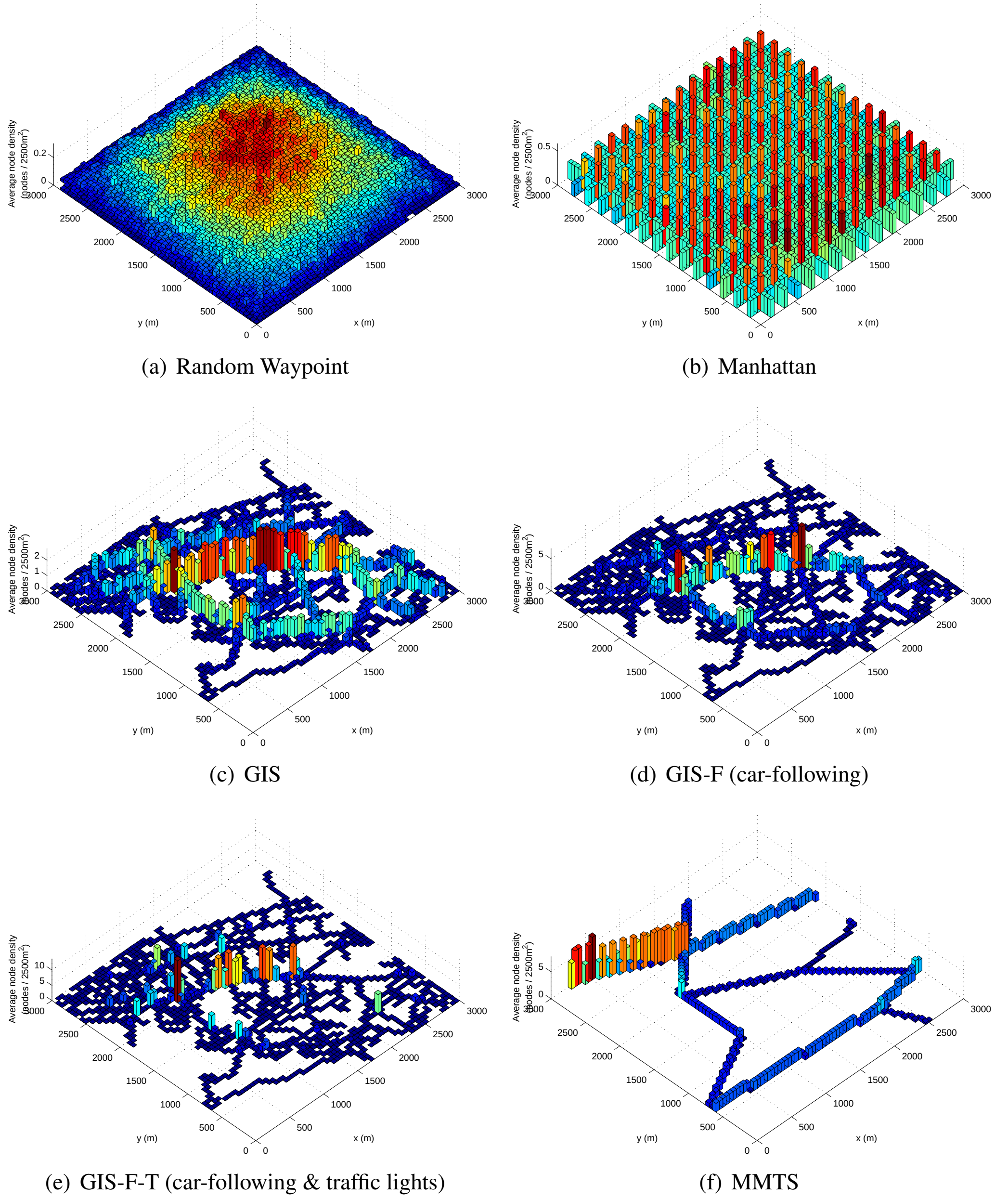
GIS Model The GIS model restricts node movements along the exact course of roads. Therefore, it is only possible to observe a node in about 41% of the unit squares. The car-following model (GIS-F) and the traffic light model (GIS-F-T) do not influence the area covered by node movements but introduce hotspot regions with higher node densities in the center and in the proximity of traffic lights.

MMTS Model Not so surprisingly, vehicles in the MMTS model cover only around 9% of the area since MMTS restricts movement to the major roads.

Conclusion As a first result of this evaluation, we observe that the realistic GIS mobility model restricts the movement of nodes to less than half of the simulation area (see Table 6.5). Furthermore, we observe that enabling the car-following model (GIS-F) has not a large influence on the number of nodes per unit square. Yet, traffic lights in the GIS model increase the clustering of nodes and lead to a similar node density as in the MMTS model.

6.2.2 Distance between Nodes

Figure 6.6 shows the measurement results of the distance between nodes in the simulation area. The distribution of the distance between nodes in the Random Waypoint and in the GIS models follows a heavy-tailed normal distribution. Remarkable are the peaks in the distance distribution in the Manhattan model. The peaks occur at multiples of the grid width (200 meters).



(a) Random Waypoint

(b) Manhattan

(c) GIS

(d) GIS-F (car-following)

(e) GIS-F-T (car-following & traffic lights)

(f) MMTS

Figure 6.5: Average node density for the different mobility models in the Urban Scenario.

Mobility Model	Nodes	Nodes per Unit Square	Covered Squares (Ratio)
RWP	420	1.080	0.998
MN	420	1.159	0.412
GIS	420	1.421	0.421
GIS-F	420	1.523	0.422
GIS-F-T	420	2.278	0.412
MMTS	420	2.236	0.093

Table 6.5: Simulation results for the Urban Scenario.

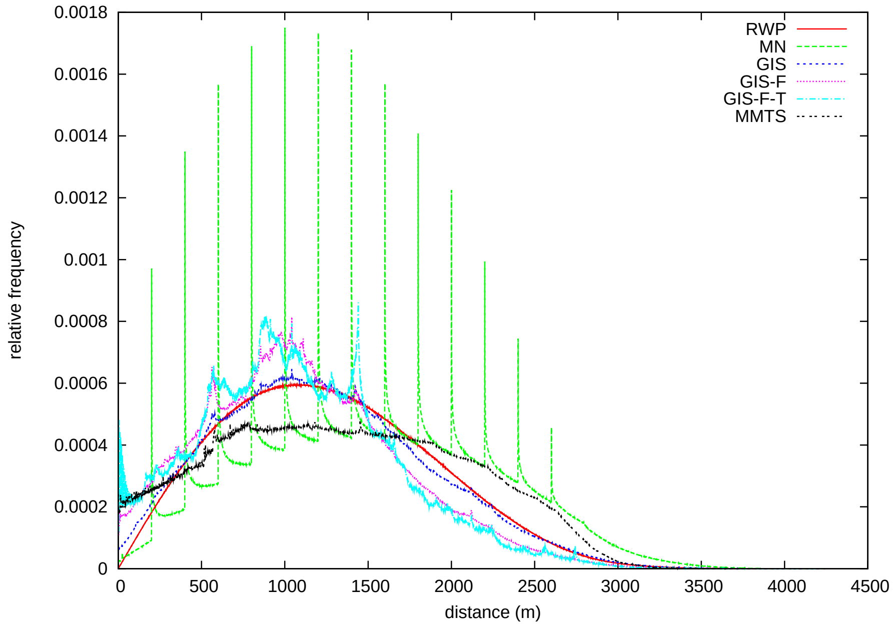


Figure 6.6: Distance between nodes in the City Scenario.

6.2.3 Distance between Neighbors

Figure 6.7 shows the measured distance between nodes which are neighbors. All neighbors are within the communication range of about 250 meters. Remarkable is the sharp peak at a distance of around 200 meters in the Manhattan model which corresponds to the distance range between nodes traveling on parallel roads. The measurement results of the neighbor distance in the GIS model with traffic lights (GIS-F-T) show peaks at intervals of 5 meters which correspond to the distances between nodes queued in front of traffic lights.

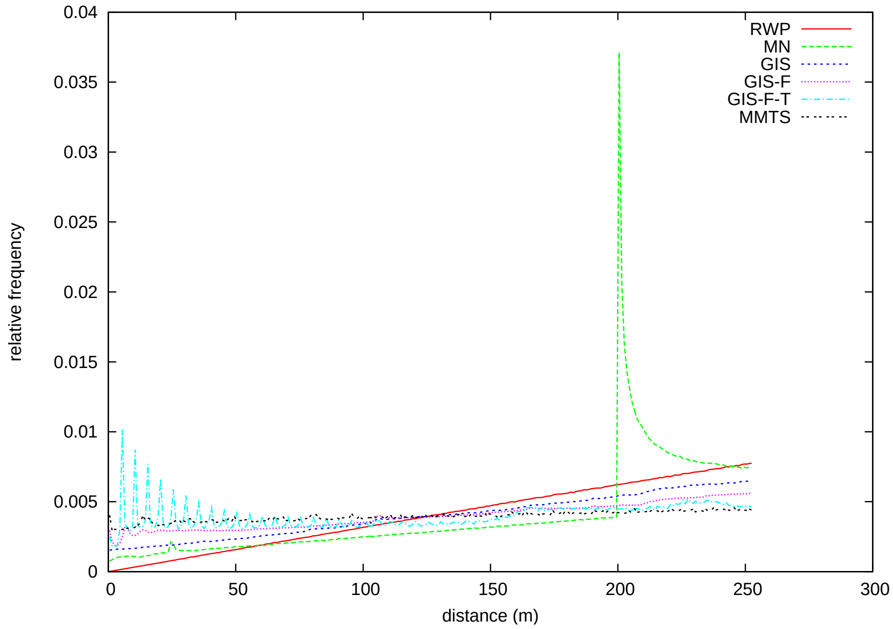


Figure 6.7: Distance between neighbors in the City Scenario.

6.2.4 Node Speed

Measurement results of node speed are shown in Figure 6.8. The measured average speed values for the different mobility models are listed in Table 6.6.

Random Waypoint Model The node speed in the Random Waypoint model is distributed within the specified speed range. Since it takes more time to complete a trip with a low speed, nodes can be observed with a higher probability at lower speeds. This measurement results are in conformance with the observations made by Navidi et al. [5] when investigating the steady-state behavior of Random Waypoint.

Manhattan Model The speed of nodes in the Manhattan model approaches the lower speed bound since nodes have to adapt their speed to the front vehicle to avoid collisions.

GIS Model The two peaks in the GIS model account for the speed limit of the two predominant road categories present in the scenario. In addition, node speed in the GIS-F and GIS-F-T models is distributed over the whole range since nodes are continually accelerating or decelerating due to the car-following model or traffic lights.

MMTS Model A large part of the vehicles in the MMTS model travel at speeds between 12 and 17 m/s. In addition, we measure a peak at around 3 m/s which is presumably arisen due to high traffic load on certain road segments.

Mobility Model	Average Node Speed (m/s)
RWP	12.156
MN	12.601
GIS	14.216
GIS-F	11.208
GIS-F-T	7.337
MMTS	10.495

Table 6.6: Measured average speed for different mobility models in the Urban Scenario.

Conclusion Figure 6.9 shows the measured average speed values in the three scenarios for the different mobility models. The average speed in the GIS model with car-following (GIS-F, GIS-F-T) and in the MMTS model decreases when the number of nodes in the simulation area increases. The GIS-F model shows nearly the same behavior as the MMTS model. The inclusion of traffic lights in the GIS model (GIS-F-T) causes the average speed to decline. The speed in the Manhattan model decreases only slightly when the node number is increased. The average speed in the entity-based Random Waypoint model is not affected by an increased number of nodes.

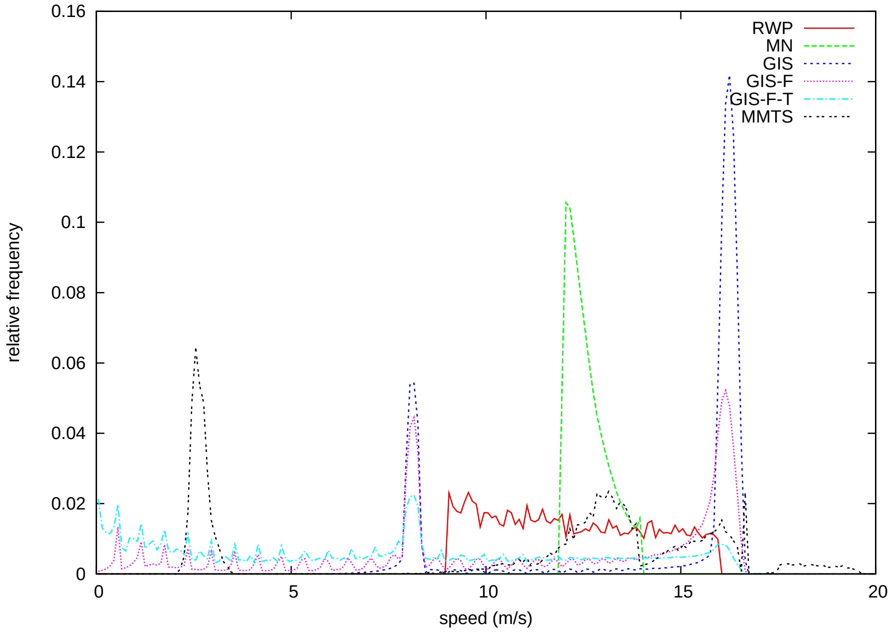


Figure 6.8: Node speed in the Urban Scenario.

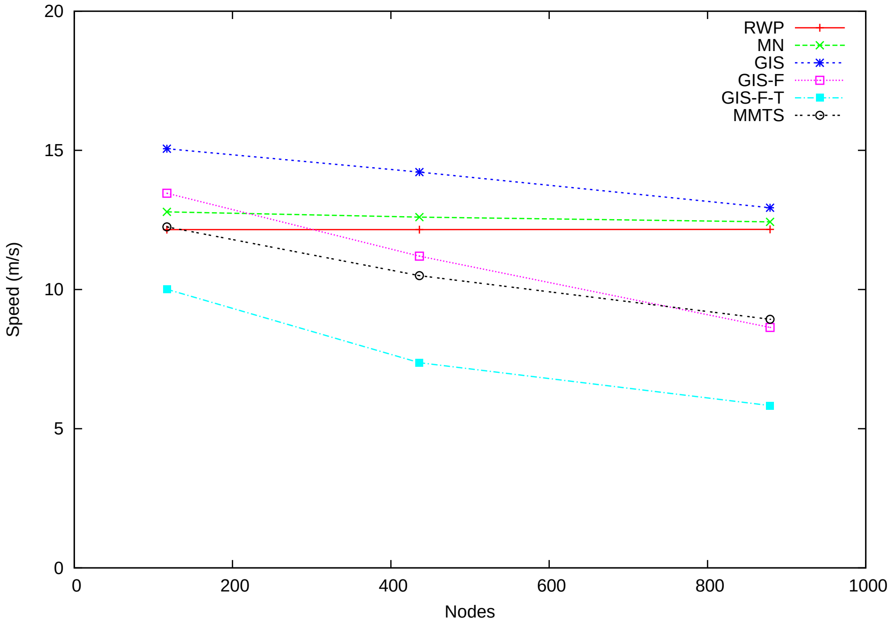


Figure 6.9: Node speed for different mobility models and scenarios (Rural, Urban, City).

6.2.5 Speed Ratio between Neighbors

The speed ratio compares the node speed to the speed of a neighbor. A high speed ratio corresponds to a small absolute speed difference. We conclude from the results of our measurements (see Figure 6.10) that the speed ratio between neighbors in the Manhattan, GIS and MMTS models is noticeable higher than in the Random Waypoint model.

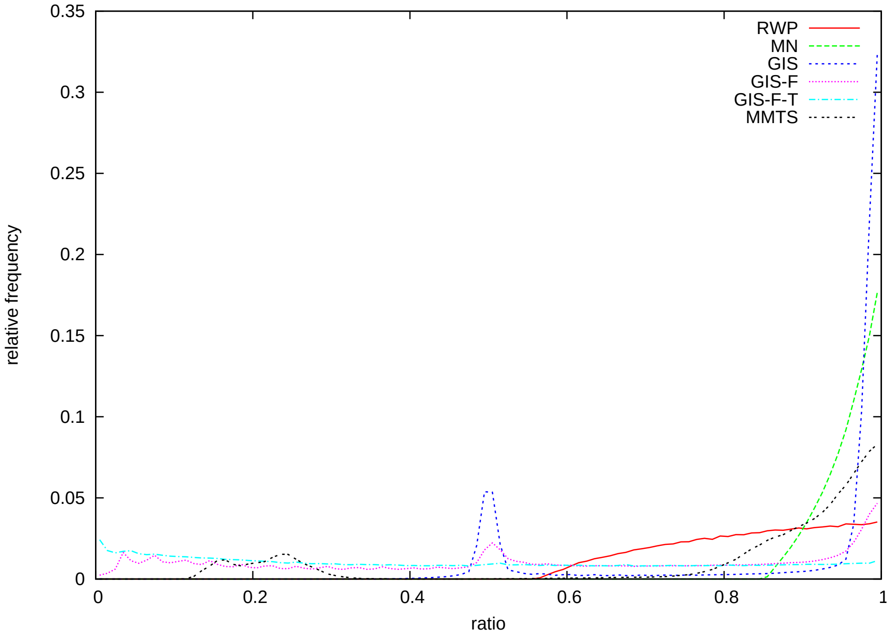


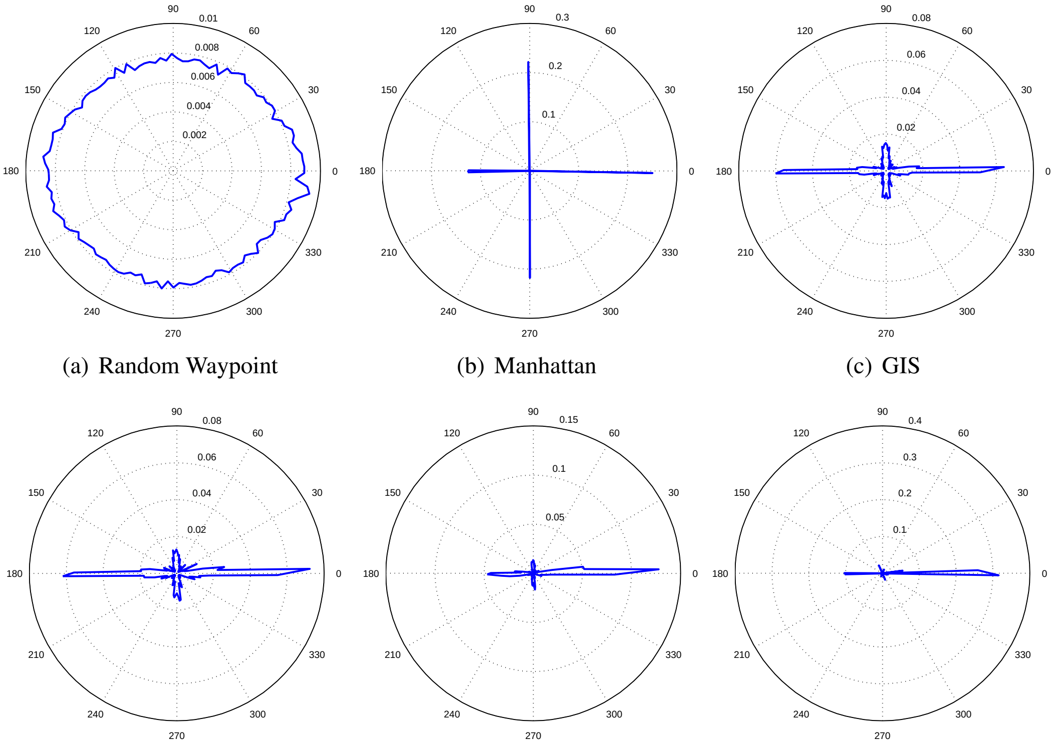
Figure 6.10: Speed ratio in the Urban Scenario.

6.2.6 Relative Movement Direction between Neighbors

We measure the relative moving direction of neighbors. Results are plotted in polar coordinates in Figure 6.11. The movement directions of neighbors are uncorrelated in the Random Waypoint model. The plots of all other mobility models show peaks at specific movement directions. The grid structure of the road network in the Manhattan model can be clearly seen in the perpendicular movement directions of neighbors in the plot. Neighbors in the GIS models move mainly in the same or in the opposite directions. Furthermore, neighbors in the GIS models often move in perpendicular directions which is a result of streets crossing at a right angle. The plot for the MMTS model is highly unbalanced since traffic flows mainly in one direction during rush hours in the vehicular traces from MMTS.

6.2.7 Spatial Dependence between Neighbors

Spatial dependence takes into account the speed difference and the relative movement direction of neighbors. It is a measure of the extent of similarity between the movements of two neighbors. Nodes traveling in the same direction with equal speed have a high spatial dependence. The same is true for nodes which are currently not moving (e.g. they are waiting in the same traffic light queue). Nodes which travel in opposite directions have a low spatial dependence. Hence, this metric can be used to describe the stability of radio links between nodes. Figure 6.12 shows the measurement results of the spatial dependence in the Urban Scenario. Remarkable is



(a) Random Waypoint (b) Manhattan (c) GIS

(d) GIS-F (e) GIS-F-T (f) MMTS

Figure 6.11: Relative movement direction of neighbors in the Urban Scenario.

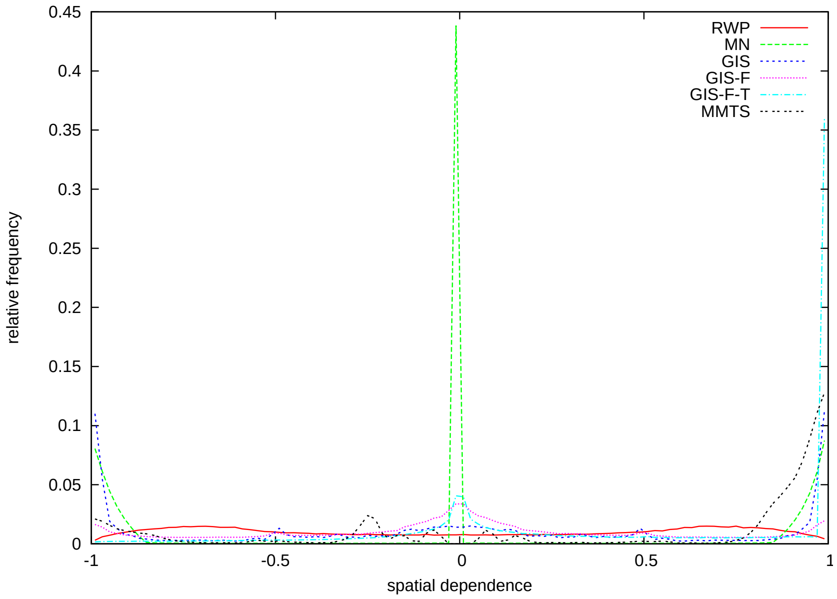


Figure 6.12: Degree of spatial dependence between neighbors in the Urban Scenario.

the peak around zero for the Manhattan model which is caused by neighbors traveling on perpendicular roads. The Manhattan, GIS and MMTS models show a high spatial dependence between neighbors.

6.2.8 Contact Duration

The contact duration measures the time period during which two nodes are neighbors (see Figure 6.13). The contact is lost when the distance between the two nodes exceeds the communication range.

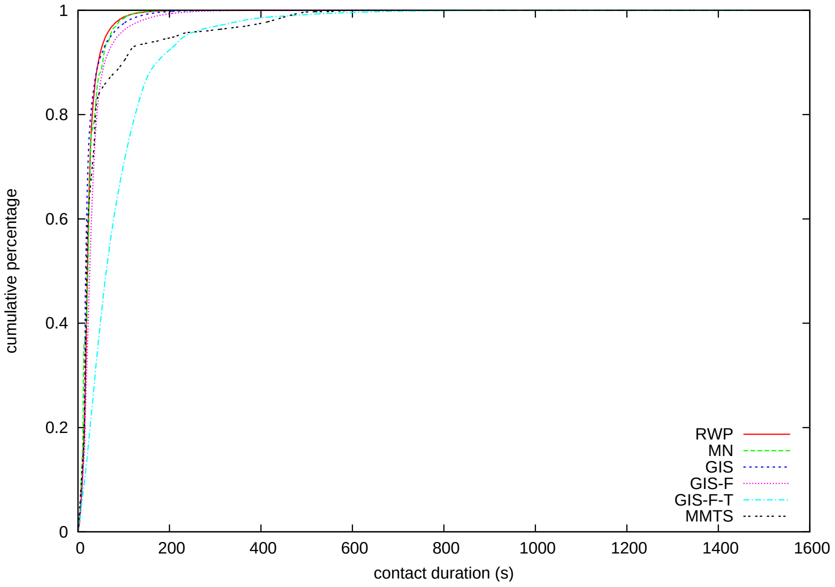


Figure 6.13: Contact duration for nodes in the Urban Scenario.

The cumulative distribution of the contact duration are similar for Random Waypoint, Manhattan, GIS and GIS-F. The traffic lights in the GIS-F-T model introduces queueing and, therefore, nodes have the same neighbors for a long time period due to the stop-and-go traffic phenomenon. Nodes move for a long time in close distance to each other due to the structure of the road network in the MMTS model.

6.2.9 Paired Inter-Contact Duration

The paired inter-contact duration measures the time period between the time when a contact is lost until the same two nodes are neighbors again. The paired inter-contact duration (see Figure 6.14) is noticeable smaller in the MMTS model compared to the other mobility models. Around 80% of the successive contacts occur within the following 60 seconds. It is rarely possible that nodes meet again after a long time, since nodes in the MMTS model remain on average for only 212.27 seconds in the simulation area. In contrast to this behavior of the MMTS model, nodes in the other mobility models do not leave the area during the simulation period. Therefore, it is more likely that two nodes will meet again (e.g. about 88% of the nodes in the Random Waypoint model have a paired inter-contact duration of less than 1,000 seconds).

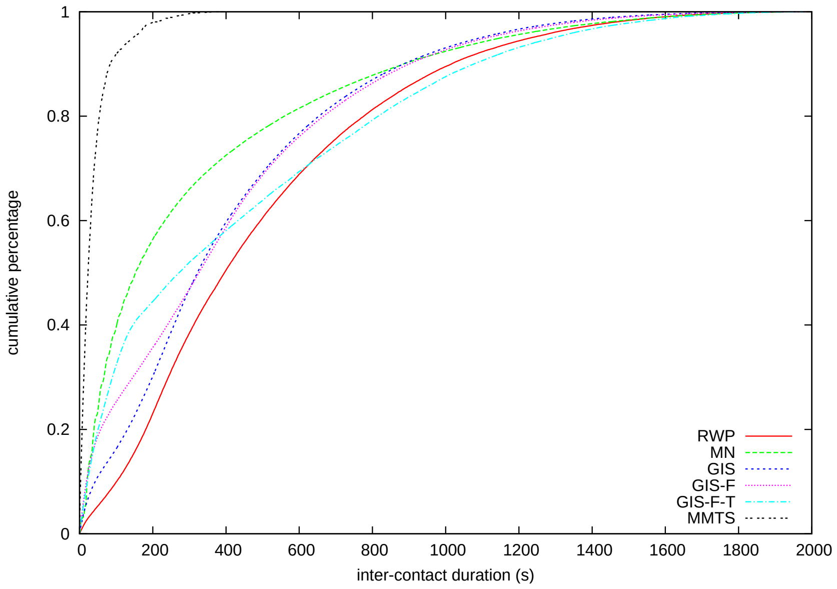


Figure 6.14: Inter-contact duration of nodes in the Urban Scenario.

Mobility Model	Contact Duration	Inter-Contact Duration
RWP	25.720 s	488.849 s
MN	26.303 s	315.852 s
GIS	25.016 s	409.098 s
GIS-F	33.611 s	398.070 s
GIS-F-T	86.252 s	429.812 s
MMTS	46.582 s	38.775 s

Table 6.7: Average values of contact and inter-contact duration for the different mobility models in the Urban Scenario.

6.2.10 Conclusion

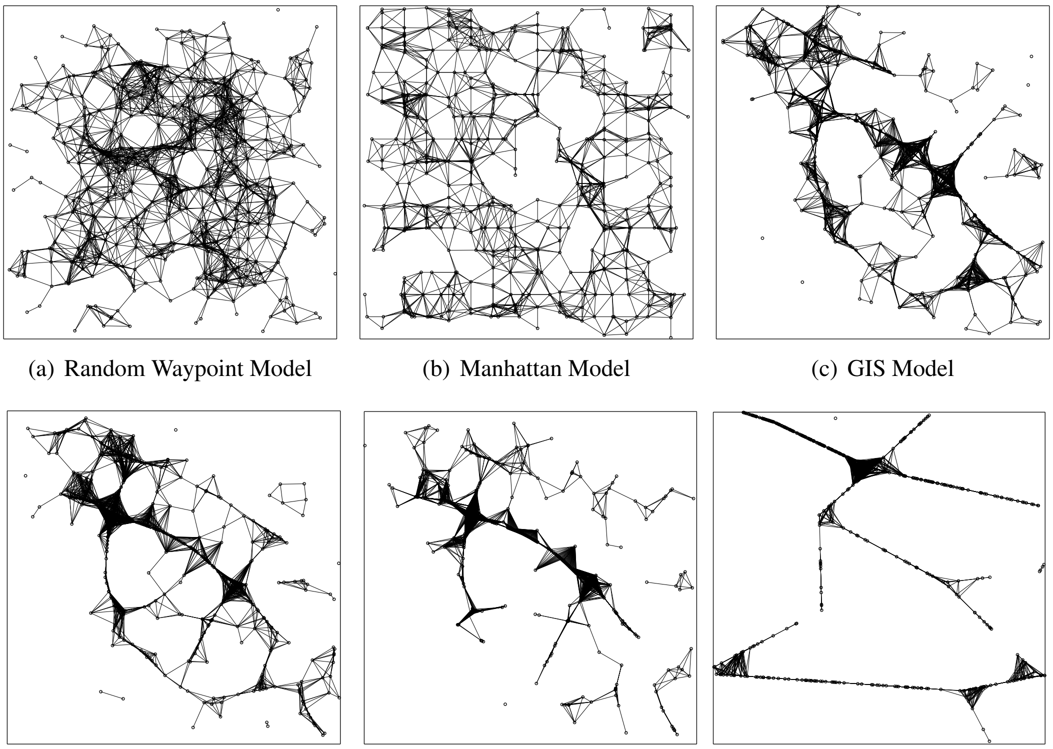
The measurement results of different mobility-related metrics presented in this section show a clear influence of the mobility model on the statistical properties of the generated mobility traces. An entity-based mobility model such as the Random Waypoint model clearly fails to mimic realistic mobility behavior. The Manhattan model takes into account interactions between nodes but restricts movement to an artificial road grid which differs from the real road topology present in the simulation scenarios. The GIS models and the MMTS model show some similarities (node density, speed, moving direction, spatial dependence) revealing their similar nature. The GIS-F, GIS-F-T and MMTS models exhibit a decrease of the average speed when the total number of nodes in the simulation area increases. This effect can be observed also in reality (e.g. speed decreases due to road congestion).

6.3 Evaluation of Graph-Related Metrics

In this section, we present an evaluation of the simulation results for graph-related metrics. We use GMSF to generate traces for the different mobility models and scenarios. The Two-Ray propagation model (with an additional constant shadowing loss) is employed to identify radio links between nodes and build the corresponding network graph. Then, we use the graph-related metrics presented in Chapter 3 to analyze the network topology by statistical means.

6.3.1 Network Graph

Figure 6.15 shows snapshots of the network graph for the different mobility models in the Urban scenario. We observe a huge influence of the underlying mobility model on the topology of the wireless network. Nodes are distributed over the whole simulation area in the Random Waypoint or Manhattan model. The GIS models lead to clustering of nodes on the major roads. Nodes in the MMTS model are concentrated on the major roads modeled in the MMTS road network.



(a) Random Waypoint Model (b) Manhattan Model (c) GIS Model

(d) GIS Model with car-following (e) GIS Model with car-following and traffic lights (f) MMTS Model

Figure 6.15: Snapshots of the network topology for different mobility models in the Urban Scenario.

6.3.2 Number of Neighbors

We measure the number of neighbors for each node which corresponds to the node degree in the network graph. Figure 6.16 shows the cumulative distribution of neighbors for different mobility models in the Urban Scenario.

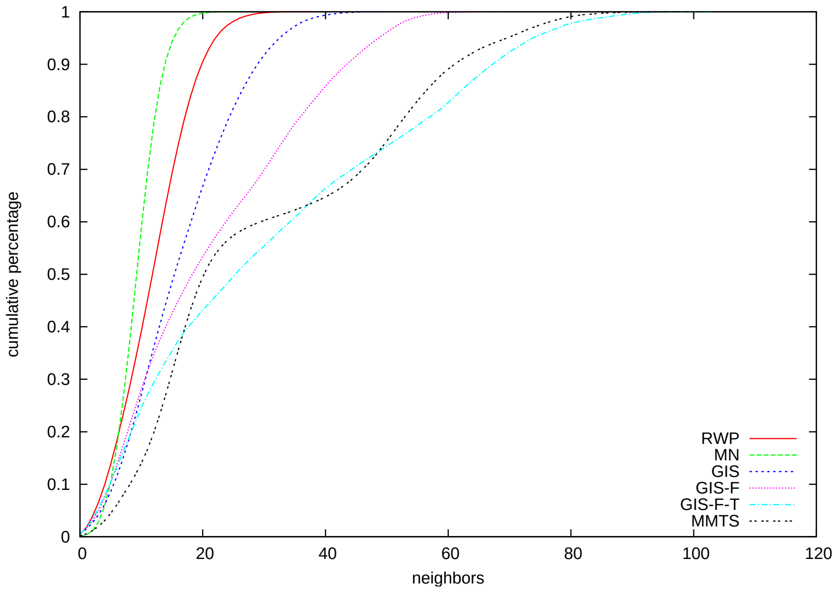


Figure 6.16: Number of neighbors in the Urban Scenario.

Around 20 percent of the nodes have less than 15 neighbors regardless of the employed mobility model. While 80 percent of the nodes in the Random Waypoint model have less than 20 neighbors, 80 percent of the nodes in the MMTS model have less than 60 neighbors. We observe that the number of neighbors increases when car-following is enabled in the GIS model (GIS-F). Queuing in front of traffic lights (GIS-F-T model) leads to clustering of nodes which increases the number of neighbors even more. As a result of this evaluation, we observe that the distribution of the number of neighbors in the GIS-F-T model approaches the realistic vehicular traces from MMTS.

6.3.3 Link Changes

The network graph is dynamically changing due to node mobility and radio propagation effects. We measure how many of the existing links have disappeared and how many new links have been established between subsequent sampling points. The number of new links established (gained) during a sample interval of one second is shown in Figure 6.17. During more than 55 percent of the sampling intervals no new links have been established in our measurements.

6.3.4 Network Connectivity Analysis

We employ Dijkstra's shortest path algorithm on the network graph to find paths between all nodes in the network. All edges in the network graph have weight 1. We measure the ratio of all possible node pairs that can be reached over a path in the current network graph. It is common practice to show the measurement results as the ratio of unreachable pairs (see Figure 6.19).
While paths to almost all other nodes exist in the Random Waypoint, Manhattan, GIS and GIS-F models, partitioning of the network graph is more likely in the GIS-F-T and MMTS models. The snapshot of the network graph for the GIS-F-T and MMTS model (see Figure 6.15) shows a situation where a partitioning of the network graph occurs.

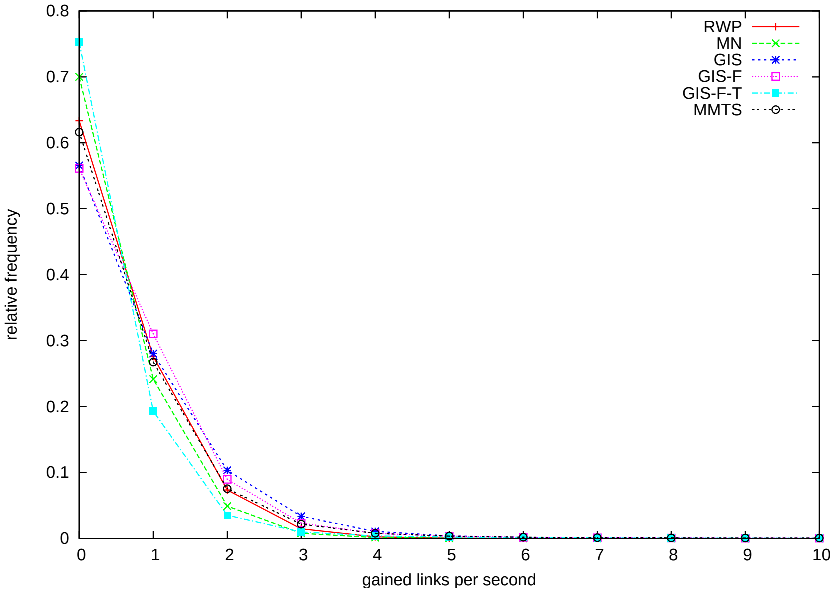


Figure 6.17: Number of links gained during a sampling interval in the Urban Scenario.

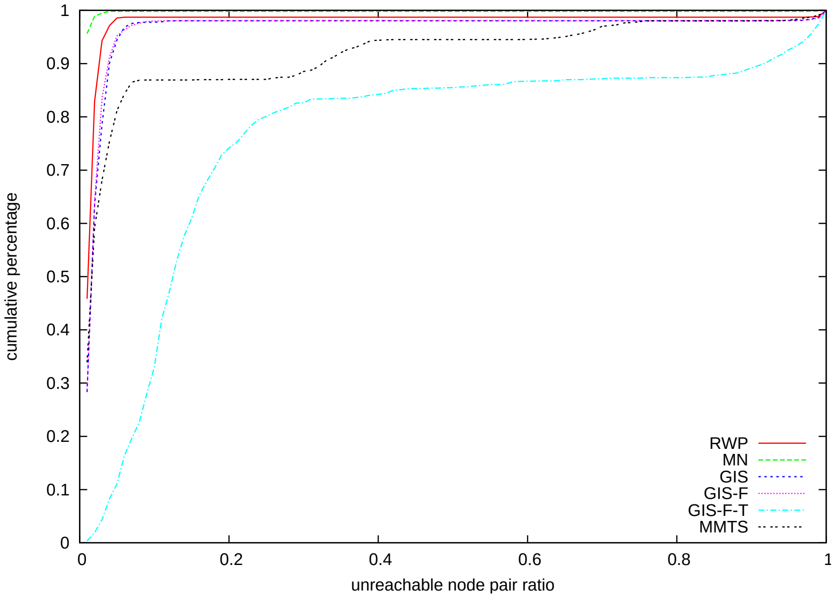


Figure 6.18: Unreachable node pairs, Urban Scenario.

6.3.5 Path Length

For all the shortest paths calculated in the network graph, we measure the number of nodes (hops) traversed from the source to the destination node (see Figure 6.18). Remarkable is the fact that a path between two random nodes in the network graph contains, on average, more hops in the MMTS model. We interpret this as an effect of the road topology of MMTS which includes only major roads. Since there are no intermediate nodes on the small roads (this can be clearly seen in the snapshot of the network topology in Figure 6.15), the path has to make a detour along the course of the major roads.

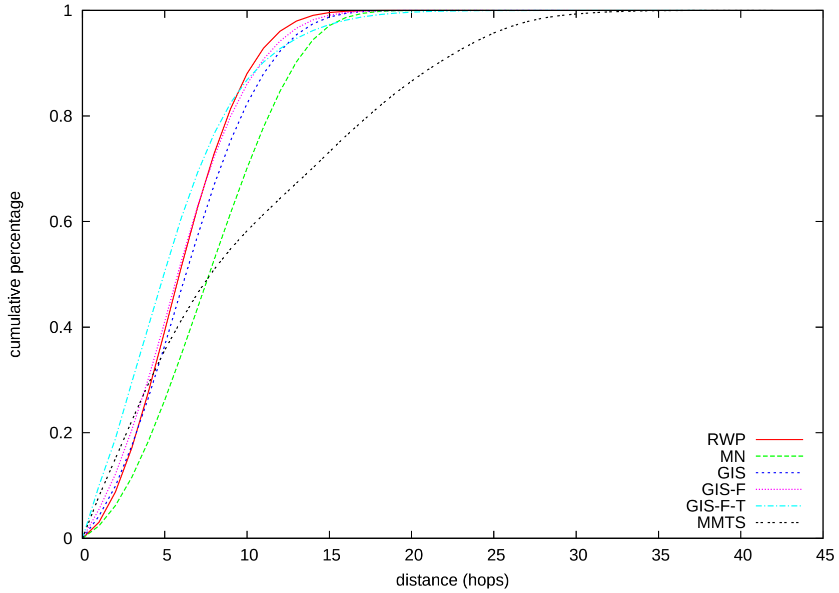


Figure 6.19: Number of hops on the shortest path between nodes in the network graph (Urban Scenario).

6.3.6 Conclusion

Our evaluation using graph metrics reveals that the graph topology is heavily dependent on the employed mobility model. Furthermore, the network graph is subject to constant changes due to mobility and radio propagation effects. The evaluation of the network graph shows that there exist paths between a large percentage of node pairs. However, it is important to remind that the network graph employed in our evaluation is only a snapshot of the current network situation. The network graph is built using a simplified model for radio propagation (Two-Ray plus additional constant shadowing loss). Therefore, it is possible that two nodes connected by an edge in the network graph are not able to communicate in reality due to interferences caused by other nodes.

6.4 Evaluation of Routing Protocol Performance Metrics

In this section, we present an evaluation of the network simulations performed in the Qualnet network simulator using the AODV and OLSR[2] routing protocols. The simulation of the ad-hoc network is based on the mobility traces generated in GMSF for the different mobility models and scenarios. We restrict our evaluation of the routing protocol performance to two metrics: the packet delivery ratio and the routing protocol overhead (see Section 3.3).

6.4.1 Packet Delivery Ratio

We measure the packet delivery ratio for the constant bit-rate (CBR) connections between the randomly selected source-destination pairs. The packet delivery ratio is defined as the ratio between packets generated at the source node and the number of packets received at the destination node. The results obtained by the simulations in Qualnet are shown in Figure 6.20. The 95 % confidence intervals for the mean of the packet delivery ratio are indicated by the error bars in the plots.

AODV We observe significant differences in the packet delivery ratio of AODV between the different mobility models in the Rural scenario. The performance with the Random Waypoint or Manhattan model is significant lower compared to the GIS models and the MMTS model. The difference between the different mobility models is smaller in the Urban Scenario and in the City Scenario. The highest packet delivery is achieved with the GIS model with inclusion of car-following and traffic lights (GIS-F-T) in the Rural scenario. It is remarkable, that a doubling of the nodes in the simulation area from the Urban to the City Scenario does not lead to a significant higher packet delivery ratio neither for the Random Waypoint model nor for the Manhattan model.

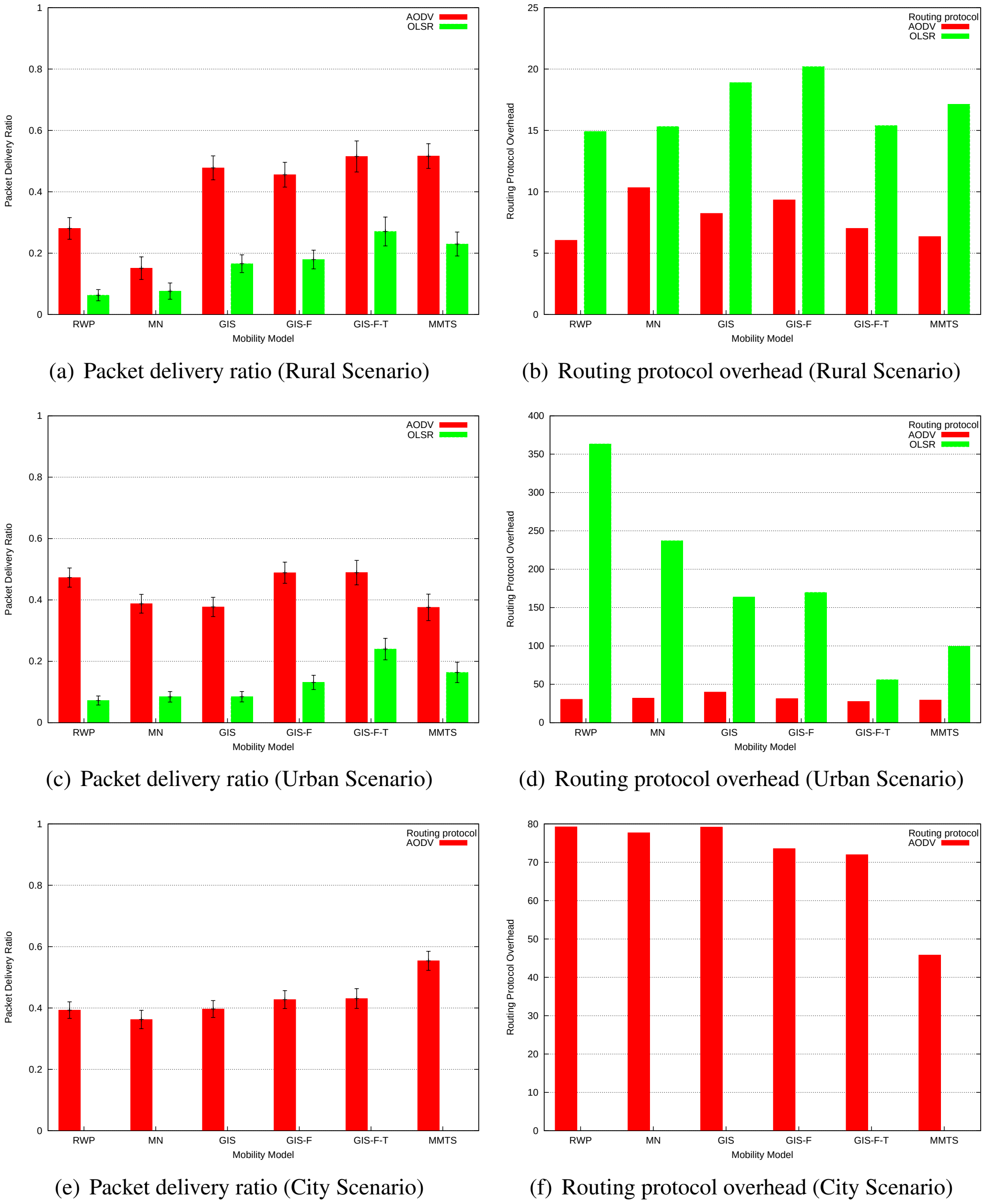
OLSR OLSR achieves a significant lower packet delivery ratio compared to AODV. We measured the highest delivery ratio for OLSR when the GIS-F-T or MMTS model is used. The evaluation of the mobility behavior of these two mobility models revealed that they have the highest node contact duration amongst all mobility models. It seems that the default interval for `HELLO` messages in Qualnet's OLSR implementation is too large to cope with the high node mobility in our scenarios.

6.4.2 Routing Protocol Overhead

To compare the efficiency of routing protocols, we measure the ratio between the routing packets and data packets sent during the simulation. The measurement results for the different mobility models and scenarios are shown in Figure 6.20.

The pro-active calculation of routes in OLSR requires a large amount of routing packets for the continuous distribution of topology information. Therefore, the routing overhead in OLSR is much higher compared to the re-active AODV protocol. Furthermore, the routing overhead in both OLSR and AODV increases with an increasing number of nodes in the network.

[2] Simulation of the OLSR protocol in the City Scenario turned out to be infeasible in the remaining time of this thesis. Therefore, we present the simulation results of OLSR for the Rural Scenario and the Urban Scenario.



(a) Packet delivery ratio (Rural Scenario)

(b) Routing protocol overhead (Rural Scenario)

(c) Packet delivery ratio (Urban Scenario)

(d) Routing protocol overhead (Urban Scenario)

(e) Packet delivery ratio (City Scenario)

(f) Routing protocol overhead (City Scenario)

Figure 6.20: Packet delivery ratio and routing protocol overhead for different mobility models and scenarios.

6.4.3 Conclusion

We simulate the different mobility models and scenarios in Qualnet to investigate the influence on the performance of AODV and OLSR. For this purpose, we measure the packet delivery ratio and the routing protocol overhead. The re-active AODV routing protocol achieves a significant higher delivery ratio than the pro-active OLSR protocol for all simulated mobility models and scenarios. In the best case, AODV was able to deliver only around half of the data packets to their destination which is clearly not sufficient for most application scenarios. In addition, the routing protocols themselves are responsible for a high percentage of the packets generated in the network. We conclude from our results that the performance of a routing protocol depends highly on the underlying mobility model.

6.5 Summary of Major Findings

In this section, we present a summary of the most important evaluation results. Based on our evaluation of different mobility models by statistical means and by simulations in the Qualnet network simulator, we draw the following conclusions:

- The underlying mobility model has a high influence on the statistical behavior of nodes. The average node speed varies by a factor of two between different mobility models. We observe a decrease of the average speed in the GIS-F, GIS-F-T and MMTS model with an increasing number of nodes in the simulation area.

- Realistic mobility models exhibit a high extent of similarity between the movements (speed, direction) of two nodes which are in close distance to each other. Neighbors in the Manhattan, GIS, GIS-F, GIS-F-T and MMTS models move mainly in the same or opposite directions and with similar speed. This behavior cannot be observed in the entity-based Random Waypoint model.

- The restriction of node mobility to the realistic road topology in the GIS model has a significant influence on the node density and introduces clusters at hotspots (traffic lights, major roads). On average, we observe in the GIS-F-T model twice as many nodes per square unit as in the Random Waypoint model. The contact duration between nodes is more than a factor of three higher in the GIS-F-T model than in the Random Waypoint model.

- We observe a huge influence of the underlying mobility model on the topology of the wireless network. The GIS-F-T and the MMTS model have a significant higher number of neighbors compared to the Random Waypoint and Manhattan models. Clustering of nodes in the GIS-F-T and MMTS model can lead to a partitioning of the network graph.

- The re-active AODV routing protocol achieves a higher packet delivery ratio than the pro-active OLSR protocol in our simulations. Furthermore, OLSR generates a much larger number of routing packets than AODV. The constant flooding of topology information through the network in OLSR accounts for the significant higher routing protocol overhead of OLSR compared to AODV.

7 Conclusion and Outlook

In this chapter, we present the conclusion of this thesis. In addition, we summarize the contributions and major findings. Finally, we give an outlook on possible future work.

7.1 Conclusion

In this thesis, we design a generic mobility simulation framework (GMSF). The fully functional implementation of this framework greatly simplifies the generation of mobility traces and the evaluation of different mobility models. As a result of its modular design, the framework can be easily extended with additional functionality.

Furthermore, we propose and implement a GIS-based mobility model which uses highly precise road maps, realistic speed limits, and incorporates the influence of other vehicles and traffic lights.

Due to the lack of real vehicular traces from measurement campaigns, it is difficult to benchmark how close our mobility model comes to reality. Therefore, we restrict our evaluation to a comparison of our new GIS mobility model with the Random Waypoint model, the Manhattan model and with vehicular traces from a traffic simulator (MMTS) that are considered highly realistic. This comparison is done both by statistical means and simulations in a network simulator. For this purpose, we select three different scenarios (City, Urban and Rural Scenario) which correspond to real areas in Switzerland.

7.2 Contributions

The contributions of this master thesis to the design and analysis of realist mobility models can be summarized in five main issues:

- Generic Mobility Simulation Framework (GMSF). The fully functional implementation in Java contains the following features:

  - Four mobility models (Random Waypoint, Manhattan, GIS, MMTS).
  - Three radio propagation models (Free-Space, Two-Ray and Shadowing model).
  - A module for the statistical analysis of mobility traces using mobility- and graph-related metrics.
  - Traffic generator module.
  - Output module for mobility traces in 5 different formats (Qualnet/Glomosim, ns-2, nam, XML and PDF).
  - A definition of a new XML-based file format for mobility traces.
  - Graphical user interface to visualize the network graph.

- Design and implementation of the GIS-based mobility model which includes the following features:

  - Realistic road topology from the Swiss geographic information system (GIS).
  - Realistic speed limits based on the road category in the GIS data.
  - Random trips between a set of points of interest.
  - Car-following model (Intelligent-Driver model) to incorporate the influence of close-by vehicles.
  - Traffic light model which controls admission to roads at the major intersections.

- Definition of a set of metrics for the analysis of mobility models by statistical means and simulations. This set of metrics includes:

  - Mobility-related metrics (e.g. node density, speed, contact duration)
  - Graph-related metrics (e.g. neighbors, graph connectivity)
  - Routing protocol performance metrics (e.g. packet delivery ratio, routing protocol overhead)

- Comparison of different mobility models by statistical means and routing performance evaluation of two routing protocols using different mobility models.

7.3 Major Findings

Based on our evaluation of different mobility models by statistical means and simulations, we draw the following conclusions:

- The underlying mobility model has a high influence on the statistical behavior of nodes and on the routing protocol performance in simulations. Therefore, it is crucial to employ a highly realistic mobility model for simulations of VANETs.

- Realistic mobility models exhibit a high extent of similarity between the movements (speed, direction) of two nodes which are in close distance to each other. The Random Waypoint model clearly fails to mimic this realistic behavior of vehicles in road traffic scenarios.

- The restriction of node mobility to a highly realistic road topology in our GIS-based mobility model has a significant influence on the node density and on clustering of nodes at hotspots (traffic lights, major roads). This effect can be observed to a similar extent in the realistic vehicular traces from the traffic simulator (MMTS).

- Measurements of the routing protocol performance in our specific scenarios reveal that the re-active AODV routing protocol achieves a higher packet delivery ratio than the pro-active OLSR.

7.4 Outlook and Future Work

The current implementation of our GIS-based mobility model could be enhanced with additional features to further improve the representativeness of vehicular movement. At the microscopic level, we propose the following two possible refinements of our model. Firstly, the current road topology should be extended with roads containing multiple lanes. In addition, the car-following model should be extended to allow overtaking of vehicles. At the macroscopic level, the start and destination points of trips could be derived from real data of population density for the simulation area.

The landscape model from the Swiss geographic information system (GIS) offers additional layers containing valuable data which could be exploited for further improvements of mobility and radio propagation models. Consideration of buildings in the radio propagation model could improve the estimation of the received signal power. Furthermore, our GIS-based mobility model contains only vehicles currently participating in road traffic. The vehicular ad-hoc network could be extended with currently parked vehicles and additional base stations integrated into buildings.

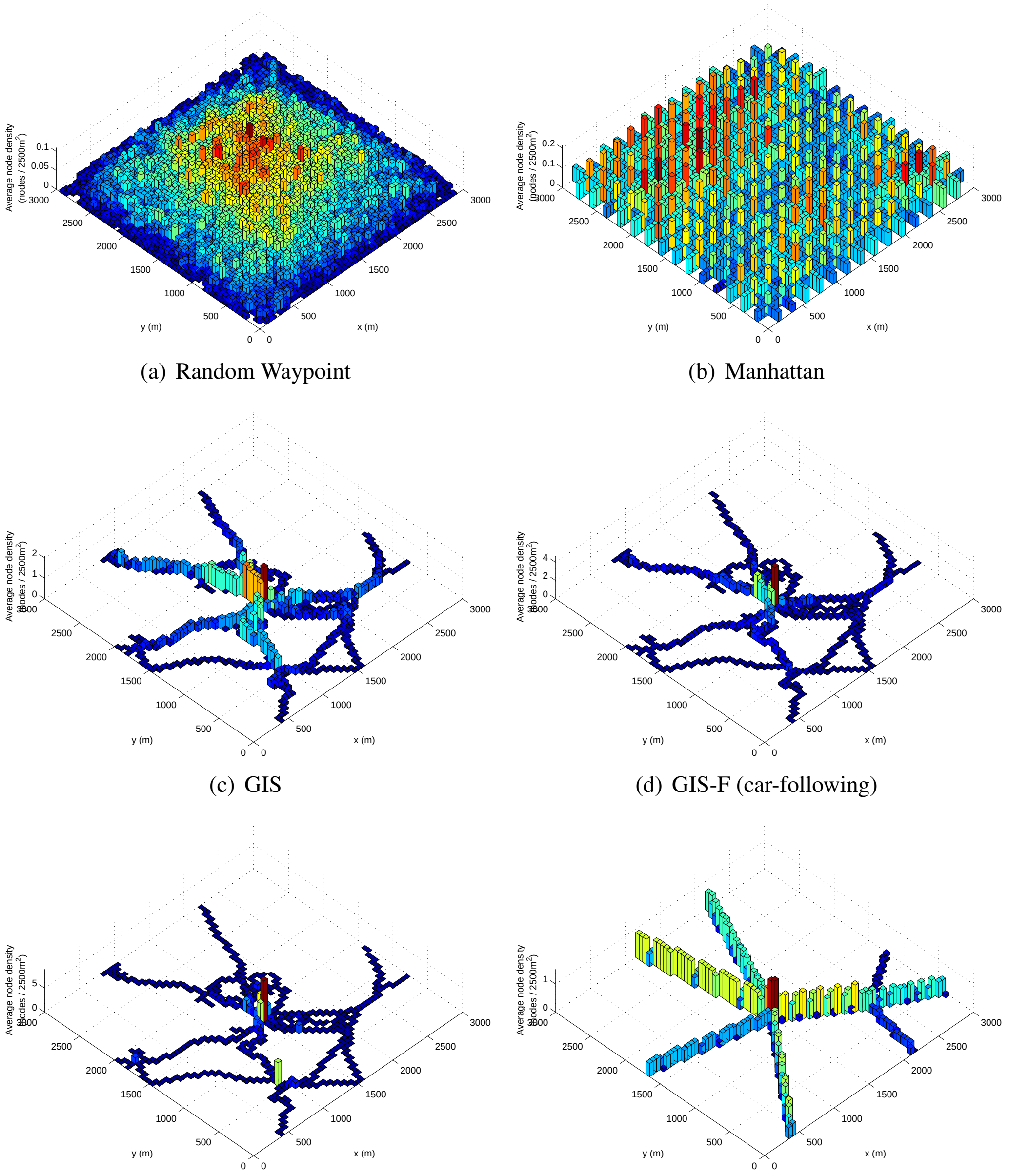
The comparison of the different mobility models revealed the large influence of the mobility model on the network graph. It would be interesting to further deepen the analysis of the network graph.

Furthermore, our evaluation of different mobility models is limited to three selected real-world scenarios. Since the selection of the scenario directly influences the road topology in the GIS and MMTS model, further evaluations should also consider additional scenarios in other areas of Switzerland.

A
Complete Results

A.1 Results of Simulation using GMSF

A.1.1 Rural Scenario



(a) Random Waypoint

(b) Manhattan

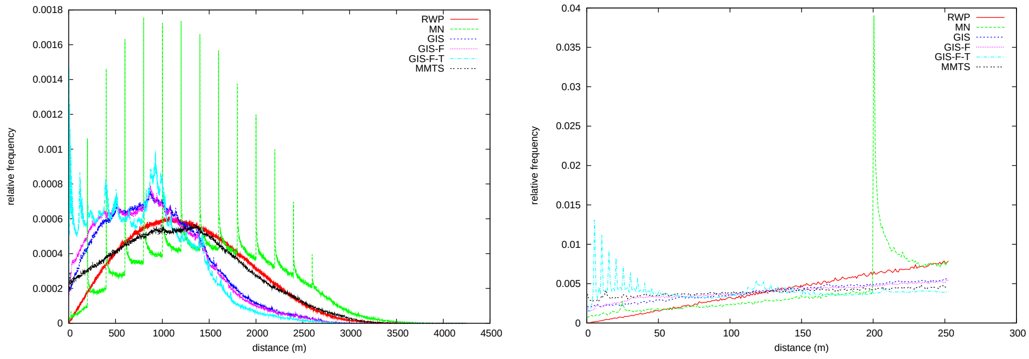
(c) GIS

(d) GIS-F (car-following)

(e) GIS-F-T (car-following & traffic lights)

(f) MMTS

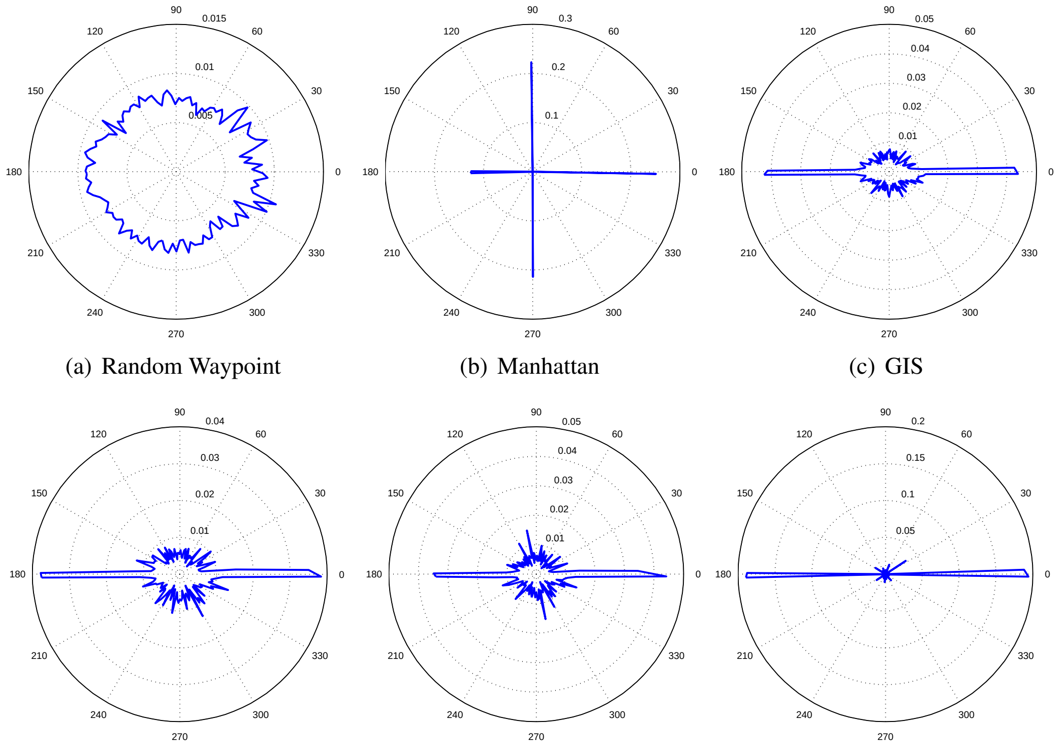
Figure A.1: Average node density for the different mobility models in the Rural Scenario.



(a) Distance between nodes

(b) Distance between neighbors

Figure A.2: Distance between nodes and distance between neighbors in the Rural Scenario.



(a) Random Waypoint

(b) Manhattan

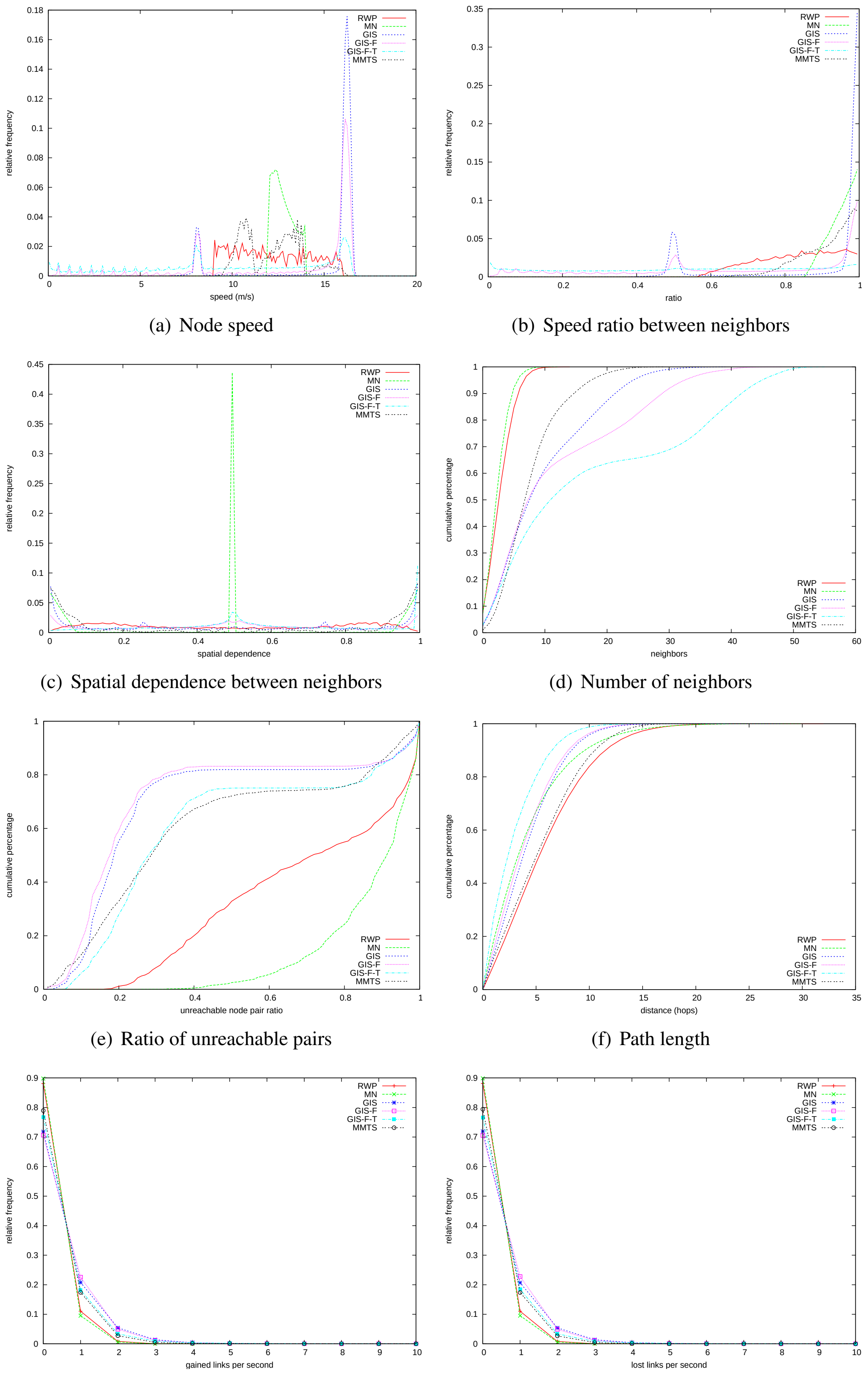
(c) GIS

(d) GIS-F (car-following)

(e) GIS-F-T (car-following and traffic lights)

(f) MMTS

Figure A.3: Relative movement direction of neighbors in the Rural Scenario.



(a) Node speed

(b) Speed ratio between neighbors

(c) Spatial dependence between neighbors

(d) Number of neighbors

(e) Ratio of unreachable pairs

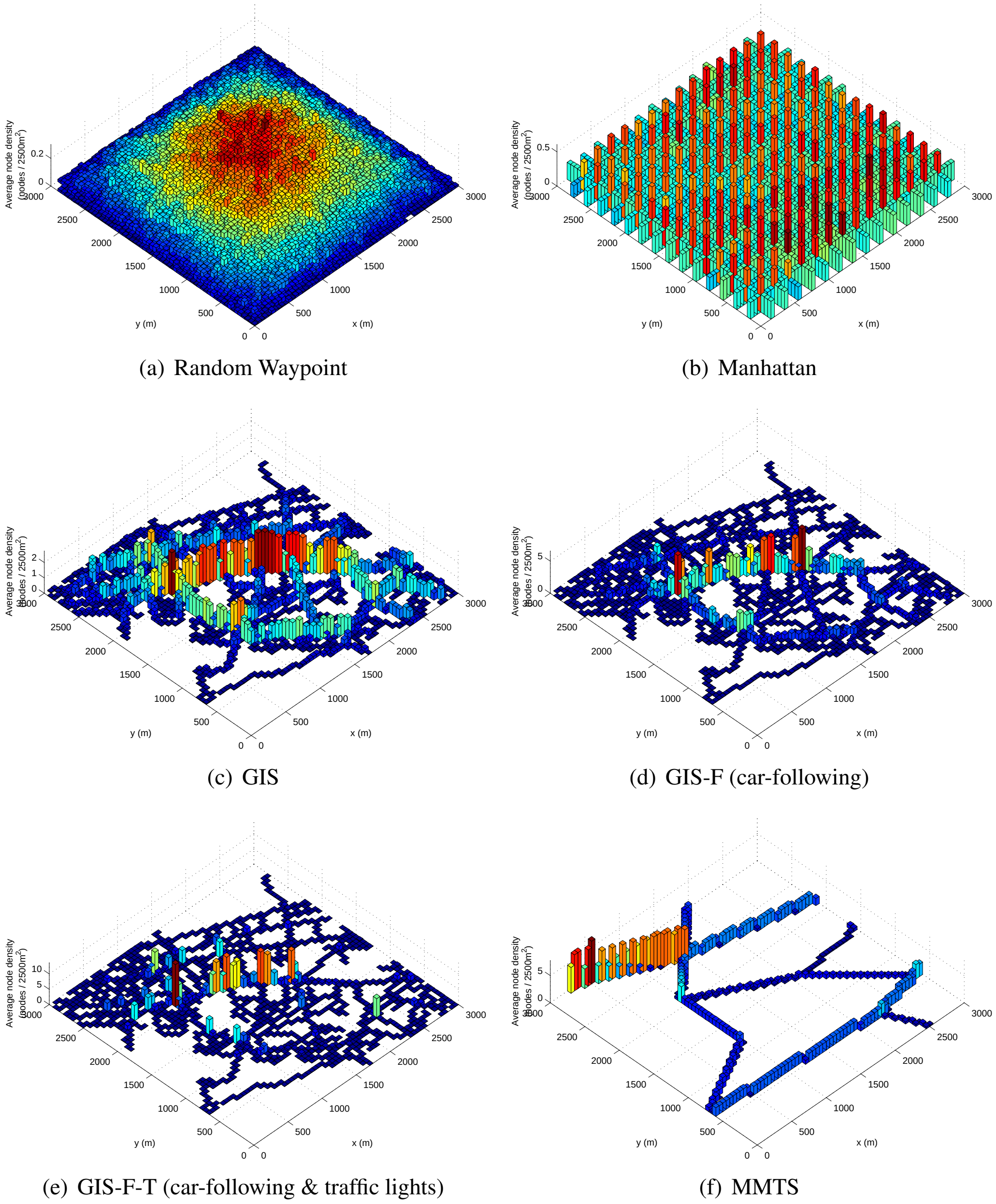
(f) Path length

(g) Links gained during sampling interval

(h) Links lost during sampling interval

Figure A.4: Plots for different metrics in the Rural Scenario.

A.1.2 Urban Scenario



(a) Random Waypoint

(b) Manhattan

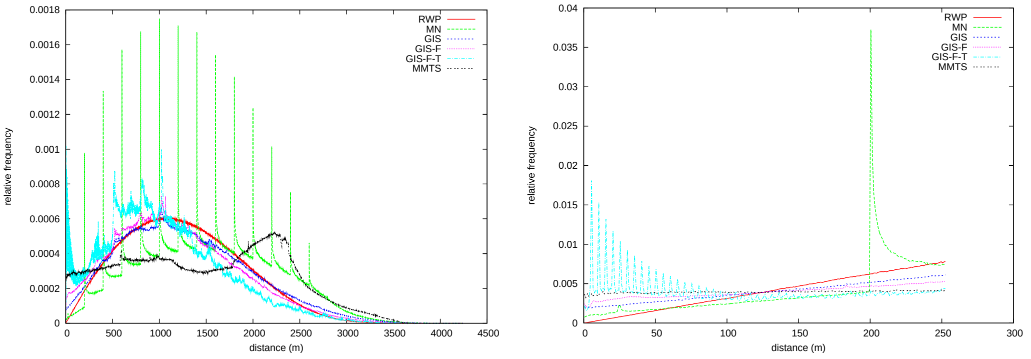
(c) GIS

(d) GIS-F (car-following)

(e) GIS-F-T (car-following & traffic lights)

(f) MMTS

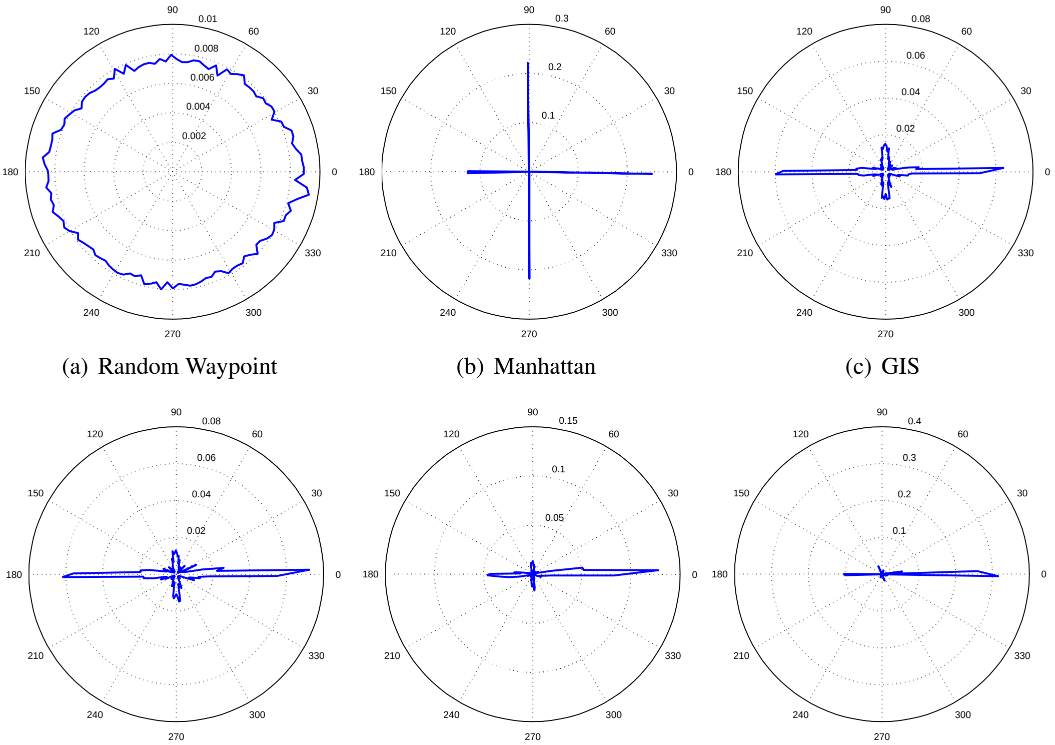
Figure A.5: Average node density for the different mobility models in the Urban Scenario.



(a) Distance between nodes

(b) Distance between neighbors

Figure A.6: Distance between nodes and distance between neighbors in the Urban Scenario.



(a) Random Waypoint

(b) Manhattan

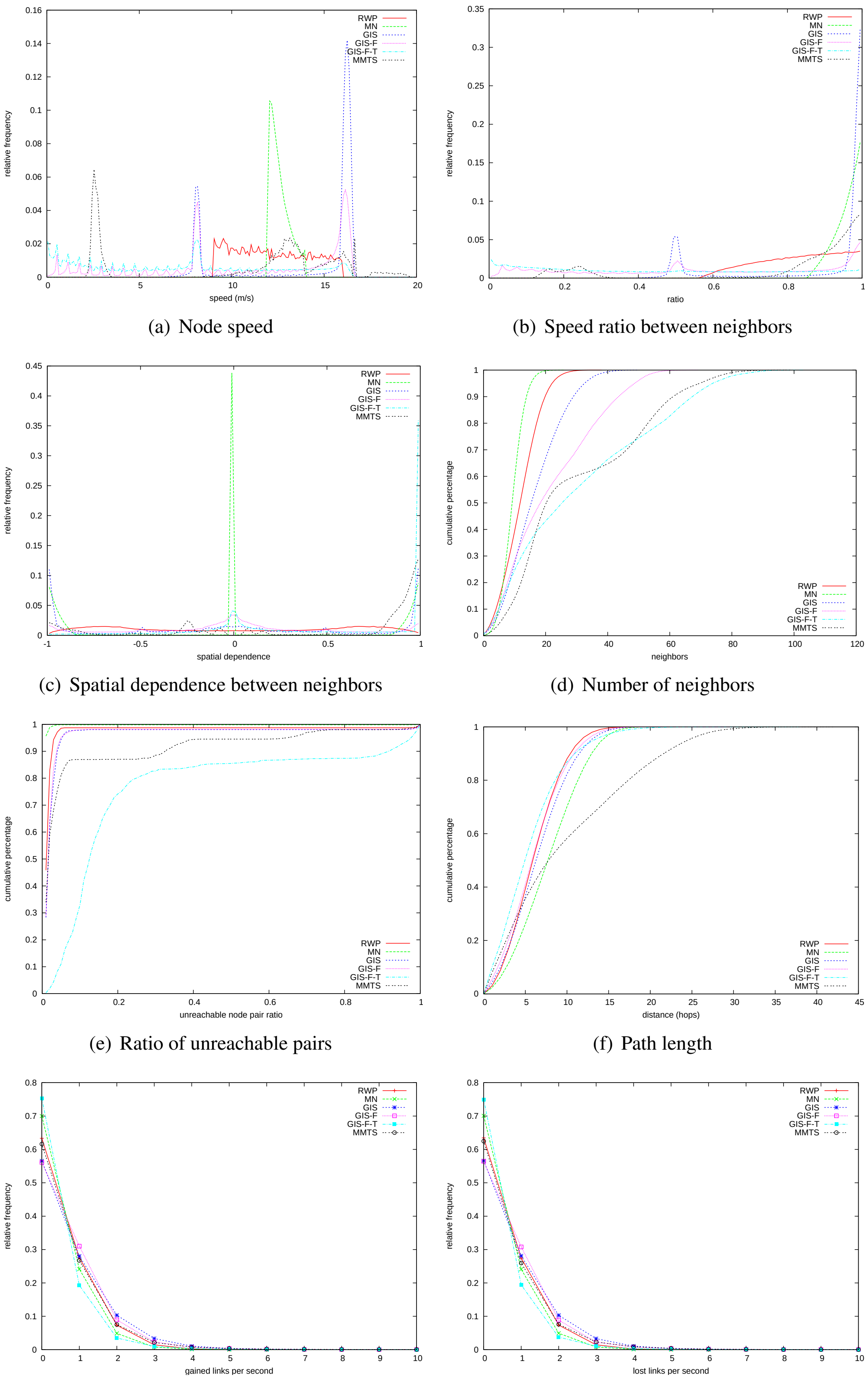
(c) GIS

(d) GIS-F (car-following)

(e) GIS-F-T (car-following and traffic lights)

(f) MMTS

Figure A.7: Relative movement direction of neighbors in the Urban Scenario.



(a) Node speed

(b) Speed ratio between neighbors

(c) Spatial dependence between neighbors

(d) Number of neighbors

(e) Ratio of unreachable pairs

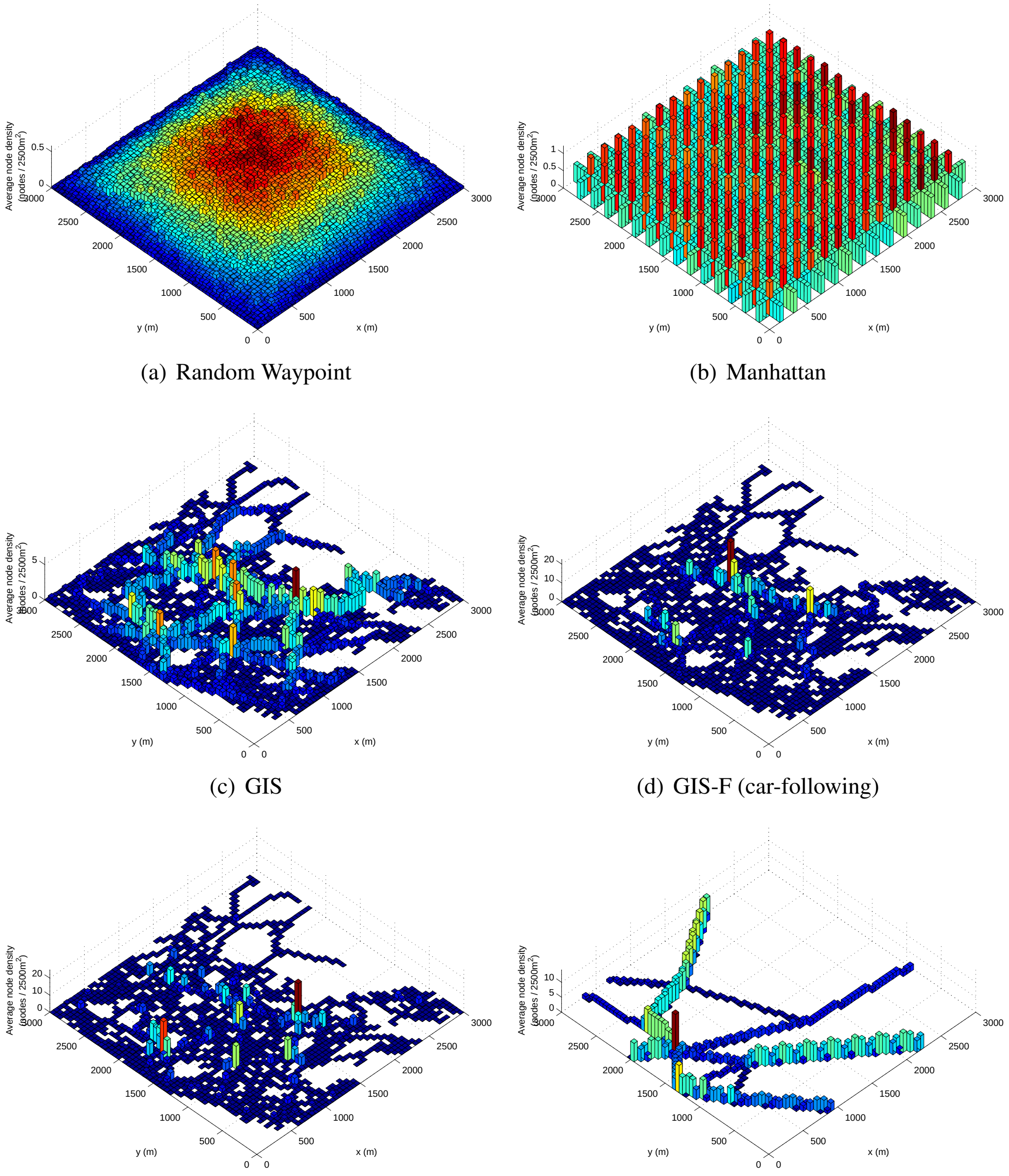
(f) Path length

(g) Links gained during sampling interval

(h) Links lost during sampling interval

Figure A.8: Plots for different metrics in the Urban Scenario.

A.1.3 City Scenario



(a) Random Waypoint

(b) Manhattan

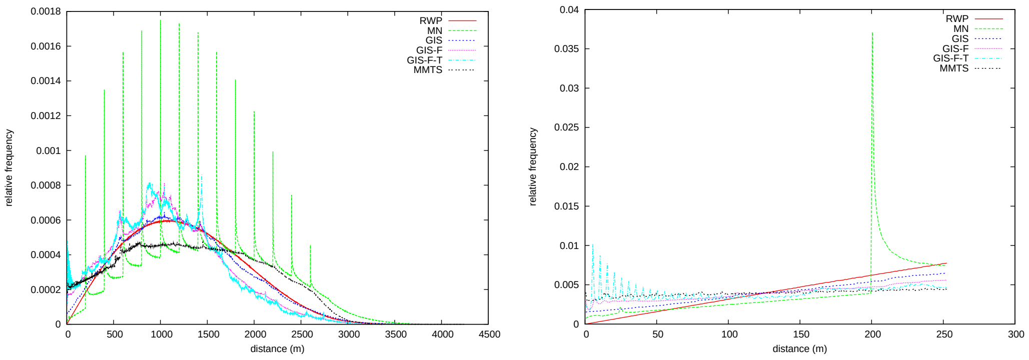
(c) GIS

(d) GIS-F (car-following)

(e) GIS-F-T (car-following & traffic lights)

(f) MMTS

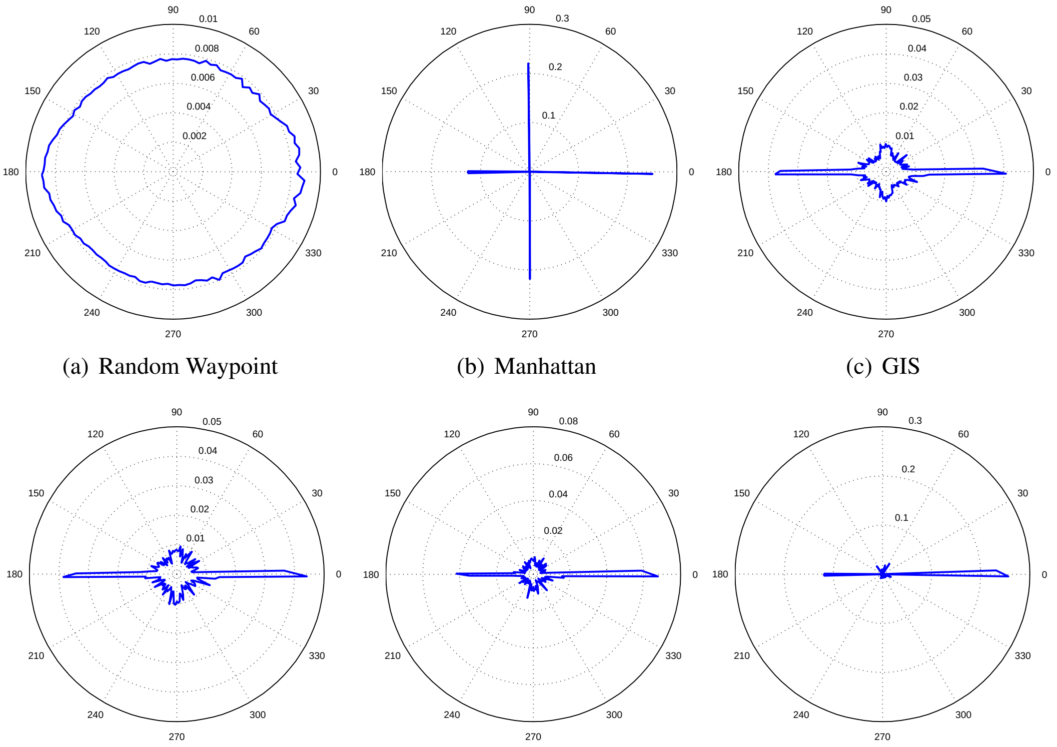
Figure A.9: Average node density for the different mobility models in the City Scenario.



(a) Distance between nodes

(b) Distance between neighbors

Figure A.10: Distance between nodes and distance between neighbors in the City Scenario.



(a) Random Waypoint

(b) Manhattan

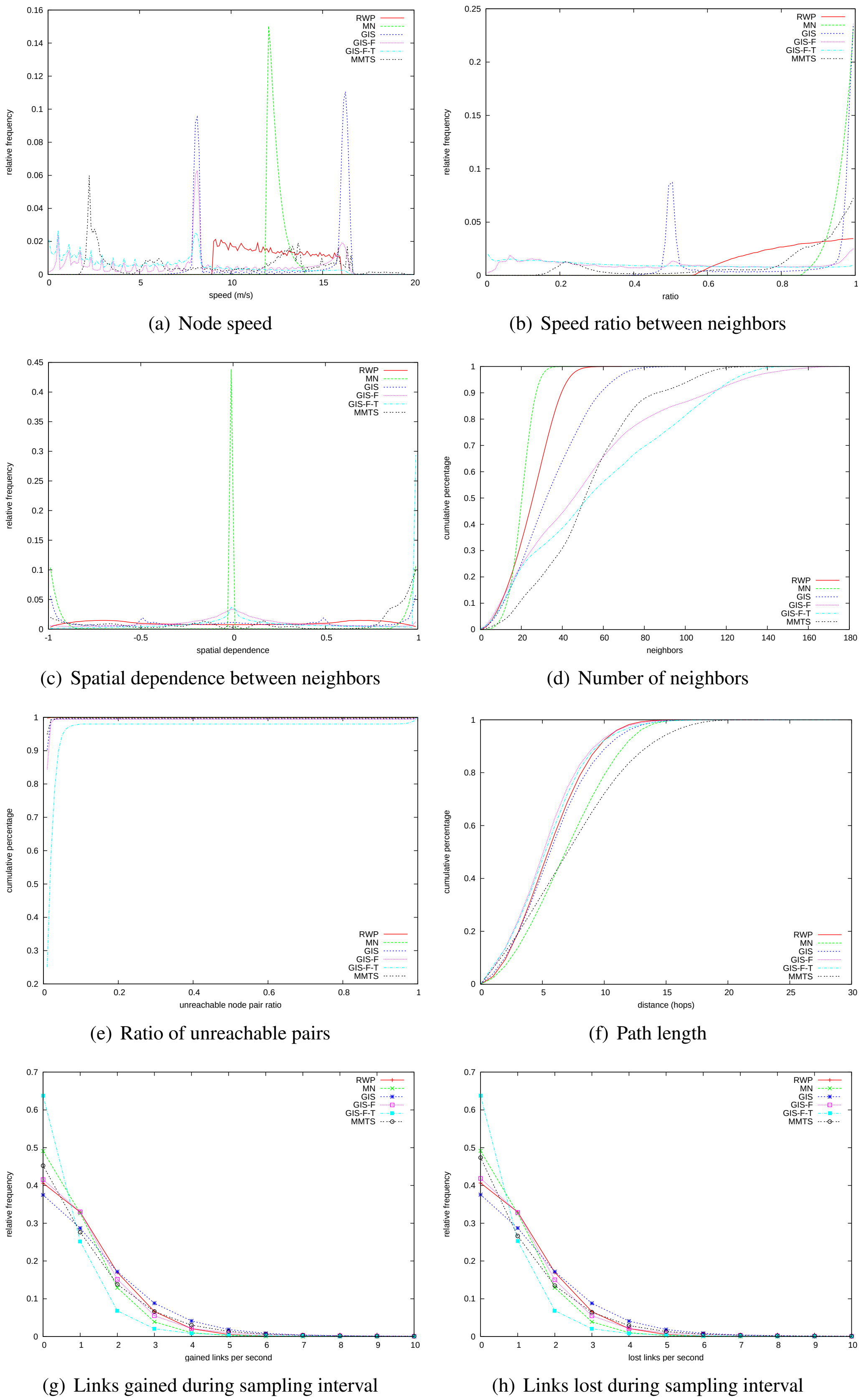
(c) GIS

(d) GIS-F (car-following)

(e) GIS-F-T (car-following and traffic lights)

(f) MMTS

Figure A.11: Relative movement direction of neighbors in the City Scenario.



(a) Node speed

(b) Speed ratio between neighbors

(c) Spatial dependence between neighbors

(d) Number of neighbors

(e) Ratio of unreachable pairs

(f) Path length

(g) Links gained during sampling interval

(h) Links lost during sampling interval

Figure A.12: Plots for different metrics in the City Scenario.

B
User Guide

B.1 Generic Mobility Simulation Framework (GMSF)

B.1.1 Building GMSF

A recent version of Java [41] and Ant [46] is recommended to build GMSF. The build process is started with the following command in the root directory of GMSF:

```
$ ant-1.6.5 compile
```

B.1.2 Running GMSF

Simulations in GMSF are started through the `gmsf.sh` script. The simulation parameters are passed as the first argument on the command line:

```
 $ gmsf.sh PARAM1=VALUE1,PARAM2=VALUE2
```

Simulation parameters have to be specified as comma-separated `key=value` pairs. The most important simulation parameters are:

- `SIMULATION_SIZE`: Size of the simulation area in meters.
- `TIME`: Simulation duration.
- `MODEL`: Specifies the mobility model (`MODEL=RWP` for Random Waypoint model, `MODEL=MN` for Manhattan model, `MODEL=GIS` for GIS model, `MODEL=MMTS` for MMTS model).
- `NODES`: Number of nodes.
- `PROPAGATION`: Radio propagation model (`PROPAGATION=FREESPACE` for Free-Space model, `PROPAGATION=TWORAY` for Two-Ray ground model, `PROPAGATION=SHADOWING` for Shadowing model).
- `OUTPUT`: Output format for mobility traces (`OUTPUT=QUALNET` for Qualnet format, `OUTPUT=NS-2` for ns-2 format, `OUTPUT=NAM` for nam format, `OUTPUT=XML` for XML format, `OUTPUT=PDF` for PDF format).

B.1.3 Importing Road Information from the GIS Data

Road information are extracted from the GIS data by a Gawk [47] script (`roads.awk`) which selects all roads contained in the specified area. The coordinates of the area (Swiss coordinates, CH1903+, [48]) have to be specified as the input arguments of the script:

```
$ gawk -f roads.awk -v x_min=687000 -v x_max=690000 \\
-v y_min=250000 -v y_max=253000 str.itf > roads.dat
```

The specified area's road segments are stored in the following format:

```
<Road>
# Segment 1
IDENTIFIER ROAD_TYPE START_X START_Y END_X END_X
# Segment 2
IDENTIFIER ROAD_TYPE START_X START_Y END_X END_X
# Segment 3
IDENTIFIER ROAD_TYPE START_X START_Y END_X END_X
</Road>

# next road
<Road>
# Segment 1
IDENTIFIER ROAD_TYPE START_X START_Y END_X END_X
# Segment 2
IDENTIFIER ROAD_TYPE START_X START_Y END_X END_X
</Road>
```

B.1.4 Importing Vehicular Traces from MMTS

NAM to DAT converter Vehicular traces from MMTS are available in the NS-2 and NAM trace formats. The scripts developed to prepare the traces for the use in the simulation framework are using the NAM traces as input file. In a first step the traces file is converted from the NAM format to an intermediate format (.dat). This conversion is done using an Gawk script and the output is written to the file `traces.dat`.

```
$ gawk -f nam2dat.awk traces.nam > traces.dat
```

Area and Time Cutter An additional Gawk script (`area_time_cutter.awk`) is provided to filter the vehicular traces by area and time period:

```
$ gawk -f area_time_cutter.awk traces.dat > scenario_unsorted.dat
```

Additionally, GMSF expects the traces to be ordered by increasing start time of the movements. This can be achieved with the `sort` command:

```
$ sort scenario_unsorted.dat > scenario_sorted.dat
```

Next, the node identifiers from MMTS are replaced by GMSF's unique node identifiers. This is done by the following command:

```
$ gawk -f rewrite_id.awk scenario_sorted.dat > scenario.dat
```

C Mobility Trace Files

C.1 Mobility Trace Support in GMSF

GMSF supports the generation of mobility traces in the following formats:

C.1.1 Qualnet Mobility Trace Format

Qualnet can use mobility traces in a separate file which can be specified with the parameter `MOBILITY FILE` in the configuration file. The file holding the node positions can be specified with the `NODE-POSITION-FILE` parameter (see Listing C.1).

```
MOBILITY FILE
NODE-POSITION-FILE nodes.mobility
```

Listing C.1: Configuration of file-based mobility in Qualnet.

Each line of the trace file describes the position of a node at a certain simulation time. Waypoints for a node have to specified in time increasing order (see Listing C.2).

```
# node-address simclock destination (x y z)
1 0.00S (2722.64, 2799.75, 0.00)
1 23.81S (2954.52, 2823.75, 0.00)
1 31.98S (2954.52, 2823.75, 0.00)
1 353.40S (228.62, 1119.03, 0.00)
1 362.53S (228.62, 1119.03, 0.00)
```

Listing C.2: Specification of node positions in the trace file (example).

Switching nodes on/off. A node can be switched off by disabling the network interface for a certain time. Failures of nodes or interfaces have to be defined in a separate configuration file

(FAULT-CONFIG-FILE parameter) prior to the start of the simulation. A failure can be either static or dynamic (random). Static failures are defined by a start and end time (see Listing C.3).

```
# interface 192.0.0.2 failed from 0S till 10S:
INTERFACE-FAULT 192.0.0.2 0S 10S
```

Listing C.3: Specification of node positions in the trace file (example).

C.1.2 Ns-2 Mobility Trace Format

Node mobility in ns-2 is normally defined by commands in a separate movement file (see Listing C.4).

```
$node_(0) set X_ 0.0
$node_(0) set Y_ 0.0
$node_(0) set Z_ 0.0
$ns_ at 0.0 "$node_(0) on"
$ns_ at 0.0 "$node_(0) setdest 984.8 941.2 12.2"
$ns_ at 100.0 "$node_(0) off"
```

Listing C.4: Example of a trace file in the ns-2 format.

Switching nodes on/off. Ns-2 offers no common solution to dynamically switch on or off a mobile node during the simulation. In contrast, the ns-2 energy model (see Listing C.5) offers two commands to switch on and off a mobile node dynamically (see Listing C.6). The energy model assigns an initial level of energy to each node at the beginning of the simulation. This energy level is decreased whenever a node transmits or receives a packet. The node is not able to receive or send packets if all the initial energy is consumed.

```
# Initial configuration of a node using the energy model
$ns_ node-config -energyModel $energymodel \
                 -rxPower $p_rx \
                 -txPower $p_tx \
                 -initialEnergy $initialenergy
```

Listing C.5: Initialization of the energy model for a node in ns-2.

```
$ns at 10.00 "$node off"
$ns at 30.00 "$node on"
```

Listing C.6: Switching on/off nodes using the ns-2 energy model (example).

C.1.3 Nam Mobility Trace Format

Nam (Network Animator) is a graphical tool provided by ns-2 for the visualization of mobility and packet traces. Trace files for nam consist of three parts: node initialization, network configuration and the actual node movements (see Listing C.7).

```
# Initialization of nodes
n -t * -s 0 -x 0.000000 -y 0.000000 -Z 0 -z 20  -v circle -c
   black

# Network configuration
V -t * -v 1.0a5 -a 0
W -t * -x 2201 -y 2401
A -t * -n 1 -p 0 -o 0xffffffff -c 31 -a 1
A -t * -h 1 -m 2147483647 -s 0

# Node movements
n -t 1.0 -s 1 -x 1000.0 -y 1000.0 -U 10.0 -V 10.0 -T 2.0
```

Listing C.7: Example for the nam trace format.

C.1.4 PDF Mobility Trace Format

GMSF allows to visualize mobility traces in the PDF format (see Figure C.1 for an example).

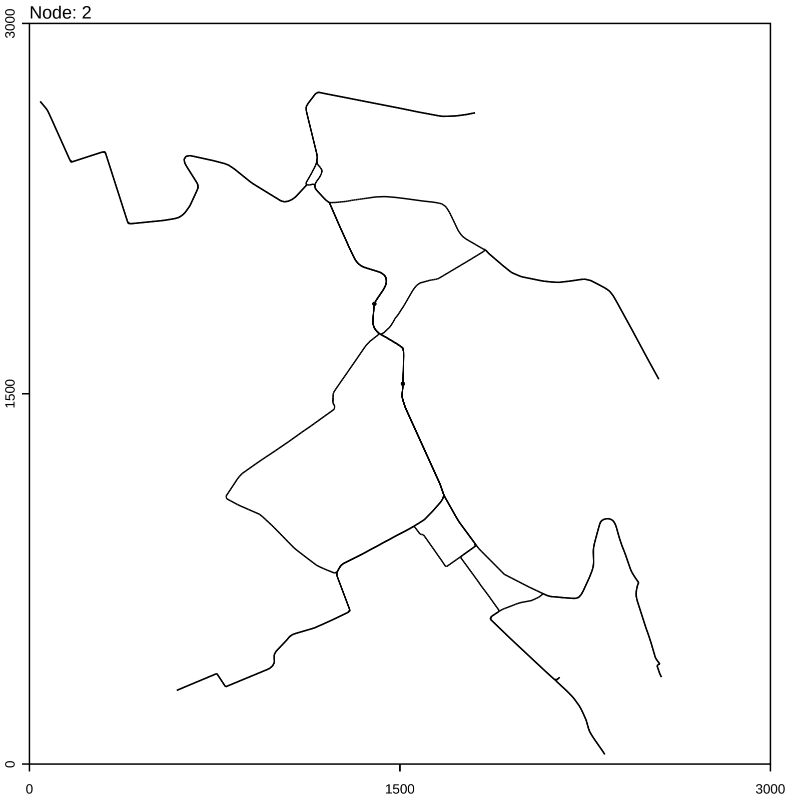


Figure C.1: Visualization of mobility traces for a single node.

C.1.5 XML Mobility Trace Format

The XML based mobility trace format is not bound to any specific network simulator. The XML schema describing the structure of the XML trace format is given in Listing C.8.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="traces">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="node" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="events">
                <xsd:complexType>
                  <xsd:choice maxOccurs="unbounded">
                    <xsd:element name="join" type="Waypoint"/>
                    <xsd:element name="leave" type="Waypoint"/>
                    <xsd:element name="move" type="Move"/>
                    <xsd:element name="pause" type="Pause"/>
                  </xsd:choice>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
            <xsd:attribute name="id" type="xsd:integer" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="Waypoint">
    <xsd:sequence>
      <xsd:element name="time" type="xsd:double"/>
      <xsd:element name="x" type="xsd:double"/>
      <xsd:element name="y" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Pause">
    <xsd:complexContent>
      <xsd:extension base="Waypoint">
          <xsd:sequence>
            <xsd:element name="duration" type="xsd:double"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="Move">
    <xsd:sequence>
      <xsd:element name="start" type="Waypoint"/>
      <xsd:element name="stop" type="Waypoint"/>
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

Listing C.8: Schema for the XML mobility trace format.

D
Qualnet Simulation Settings

The following listing shows a selection of relevant parameter settings used for the simulation in Qualnet.

```
# COORDINATE SYSTEM SETTINGS
COORDINATE-SYSTEM    CARTESIAN
TERRAIN-DIMENSIONS   (3000, 3000)

# MOBILITY SETTINGS
NODE-PLACEMENT   FILE
MOBILITY   FILE
NODE-POSITION-FILE trace.mobility
MOBILITY-POSITION-GRANULARITY   10
MOBILITY-GROUND-NODE NO

# RADIO PROPAGATION SETTINGS
PROPAGATION-CHANNEL-FREQUENCY 2.4e9
PROPAGATION-LIMIT   -111.0
PROPAGATION-PATHLOSS-MODEL TWO-RAY
PROPAGATION-SHADOWING-MODEL CONSTANT
PROPAGATION-SHADOWING-MEAN 6.0
PROPAGATION-FADING-MODEL NONE
PROPAGATION-MAX-DISTANCE 600
PROPAGATION-COMMUNICATION-PROXIMITY 100

# PHYSICAL LAYER SETTINGS
PHY-MODEL                   PHY802.11b
PHY-RX-MODEL                PHY802.11b
PHY-LISTENABLE-CHANNEL-MASK 1
PHY-LISTENING-CHANNEL-MASK  1
PHY-TEMPERATURE             290
PHY-NOISE-FACTOR            7.0
PHY802.11-AUTO-RATE-FALLBACK NO
PHY802.11-DATA-RATE                11000000
PHY802.11-DATA-RATE-FOR-BROADCAST  11000000
```

```
PHY802.11b-TX-POWER--1MBPS  15.0
PHY802.11b-TX-POWER--2MBPS  15.0
PHY802.11b-TX-POWER--6MBPS  15.0
PHY802.11b-TX-POWER-11MBPS  15.0
PHY802.11b-RX-SENSITIVITY--1MBPS  -93.0
PHY802.11b-RX-SENSITIVITY--2MBPS  -89.0
PHY802.11b-RX-SENSITIVITY--6MBPS  -87.0
PHY802.11b-RX-SENSITIVITY-11MBPS  -83.0

# ANTEANN SETTINGS
ANTENNA-GAIN              0.0
ANTENNA-EFFICIENCY        0.8
ANTENNA-MISMATCH-LOSS     0.3
ANTENNA-CABLE-LOSS        0.0
ANTENNA-CONNECTION-LOSS   0.2
ANTENNA-HEIGHT  1.5
ANTENNA-MODEL   OMNIDIRECTIONAL

# MAC LAYER SETTINGS
MAC-PROTOCOL MAC802.11
MAC-DOT11-ASSOCIATION    NONE
PROMISCUOUS-MODE   NO

# NETWORK LAYER SETTINGS
NETWORK-PROTOCOL   IP
SUBNET N16-0 { 1 thru 117}
IP-QUEUE-NUM-PRIORITIES   3
IP-QUEUE-PRIORITY-QUEUE-SIZE   50000
IP-QUEUE-TYPE   FIFO
IP-QUEUE-SCHEDULER   STRICT-PRIORITY
IP-FORWARDING YES
```

E
GIS Data Model

In this chapter, we give a very brief introduction to the structure of the geographic data used in this thesis.

E.1 Interlis Tranfer Format

Interlis [49] is a data model for geographical information systems (GIS). The Interlis transfer format is used to exchange data between different systems. The landscape model of Switzerland is described in the Interlis format.

E.1.1 Structure of the Road Network Layer

The following listing in the Interlis format defines the road network layer in the GIS landscape model.

```
TOPIC Strassennetz =

  !! Die Koordinate der Knoten sind durch den Anfang/Ende
  !! der Linie gegeben.
  TABLE str_node =
    ObjectId: INTEGER4;                 !! Eindeutiger Schluessel
    ObjectOrigin: objectOriginType;
    ObjectVal: OPTIONAL TEXT*20;
    YearOfChange: Jahr;
    Height: [0 .. 9999];
  IDENT ObjectId;
  END str_node;

  TABLE str =
    ObjectId: INTEGER4;                 !! Eindeutiger Schluessel
    ObjectOrigin:  objectOriginType;
    ObjectVal: strObject;
    YearOfChange: Jahr;
    BridgeType: OPTIONAL strBridgeType;
    TunnelType: OPTIONAL tunnelType;
    StradaId: OPTIONAL TEXT*24;   !! BaseID der Strada-Axe mit der
                                  !! hintenangehaengten durch '_'
  !! getrennten dreiziffrigen
  !! Segmentnummer
  !! z.B. ASB.RP..000000000145_005
  !! BaseID:  0-3 Besitzer der DB
  !!          4-7 Name der DB
  !!          8-19 Sequenznummer der DB
  !! Segmentnummer: 21-23
    FromNode: -> str_node;
    ToNode: -> str_node;
    Geometrie: POLYLINE WITH (STRAIGHTS) VERTEX LKoord;
  IDENT ObjectId;
  END str;

END Strassennetz.
```

E.1.2 Road Categories

The `ObjectVal` column in the table `str` holds the value for the road category. The following listing (extract for the sake of presentation) defines the different available road categories.

```
strObject = (
  Autobahn,
  Autob_Ri,       !! Autobahn richtungsgetrennt
  Autostr,        !! Autostrasse
  Ein_Ausf,       !! Autobahn Ein-/Ausfahrt
  A_Zufahrt,      !! Autobahnzufahrt
  Klass_1,        !! 1.Klass Strasse
  Klass_2,        !! 2.Klass Strasse
  Klass_3,        !! 3.Klass Strasse
  Klass_4,        !! 4.Klass Strasse
  Klass_5,        !! 5.Klass Strasse
  Klass_6,        !! 6.Klass Strasse
  Q_Klass,        !! Quartierstrasse
  HistWeg,        !! historischer Weg/Strasse
  PzPiste,        !! Panzerpiste
  Parkweg,
  BrueckLe,       !! alleinstehende Bruecke
  GedBruLe,       !! alleinstehende Bruecke gedeckt
  StegLe,         !! alleinstehender Steg
  Klass_5_Spur,   !! 5.Klass Strasse als Spur
  Klass_6_Spur  !! 6.Klass Strasse als Spur
);
```

Bibliography

[1] Jérôme Haerri, Fethi Filali, and Christian Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. Technical Report EURECOM+1951, Institut Eurecom, France, Mar 2006.

[2] Albert Einstein. Investigations on the theory of the brownian movement, 1905.

[3] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[4] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[5] William Navidi and Tracy Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.

[6] Jean-Yves Le Boudec and Milan Vojnovic. The Random Trip Model: Stability, Stationary Regime, and Perfect Simulation The Random Trip Model: Stability, Stationary Regime, and Perfect Simulation. *IEEE/ACM Transactions on Networking*, 14(6):1153–1166, 2006.

[7] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc wireless networks. In *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60, New York, NY, USA, 1999. ACM Press.

[8] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *INFOCOM*, 2003.

[9] Amit Kumar Saha and David B. Johnson. Modeling mobility for vehicular ad-hoc networks. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 91–92, New York, NY, USA, 2004. ACM Press.

[10] Rainer Baumann, Simon Heimlicher, and Martin May. Towards realistic mobility models for vehicular ad-hoc networks. In *26th Annual IEEE Conference on Computer Communications IEEE INFOCOM 2007, MObile Networks for Vehicular Environments (Infocom MOVE'07)*, 2007.

[11] Niranjan Potnis and Atulya Mahajan. Mobility models for vehicular ad hoc network simulations. In *ACM-SE 44: Proceedings of the 44th annual Southeast regional conference*, pages 746–747, New York, NY, USA, 2006. ACM Press.

[12] David R. Choffnes and Fabián E. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *VANET '05: Proceedings of the 2nd ACM international*

workshop on Vehicular ad hoc networks, pages 69–78, New York, NY, USA, 2005. ACM Press.

[13] Dirk Helbing. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics*, 73:1067, 2001.

[14] Daniel Krajzewicz, Georg Hertkorn, C. Rössel, and Peter Wagner. SUMO (Simulation of Urban MObility) - an open-source traffic simulation. In A. Al-Akaidi, editor, *4th Middle East Symposium on Simulation and Modelling*, pages 183 – 187, 2002.

[15] Stefan Krauß. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. PhD thesis, Hauptabteilung Mobilität und Systemtechnik des DLR Köln, 1998.

[16] Bryan Raney, Andreas Voellmy, Nurhan Cetin, Milenko Vrtic, and Kai Nagel. Towards a microscopic traffic simulation of all of switzerland. In *International Conference on Computational Science (1)*, pages 371–380, 2002.

[17] Jonghyun Kim and Stephan Bohacek. A survey-based mobility model of people for simulation of urban mesh networks. 2005. Available: `http://www.eecis.udel.edu/bohacek` (Accessed Sept 16, 2007).

[18] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 107–118, New York, NY, USA, 2002. ACM Press.

[19] Christian Tuduce and Thomas Gross. A mobility model based on wlan traces and its validation. In *Proceedings of INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 664–674 vol. 1, 2005.

[20] Xiaolan Zhang, Jim Kurose, Brian Neil Levine, Don Towsley, and Honggang Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. In *Proc. ACM Intl. Conf. on Mobile Computing and Networking (Mobicom)*, September 2007.

[21] Marvin McNett and Geoffrey M. Voelker. Access and mobility of wireless pda users. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(2):40–55, 2005.

[22] CANU Mobility Simulation Environment (CanuMobiSim), University of Stuttgart. Website. Available: `http://canu.informatik.uni-stuttgart.de/mobisim/` (Accessed Sept 16, 2007).

[23] VanetMobiSim, Insitut Eurécom, Sopia Antipolis, France. Website. Available: `http://vanet.eurecom.fr/` (Accessed on Sept 16, 2007).

[24] Ecole Polytechnique Fédérale de Lausanne. TraNS - traffic and network simulation environment. Website. Available: `http://trans.epfl.ch/` (Accessed Sept 16, 2007).

[25] Andrew Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 2002.

[26] Kaixin Xu, Mario Gerla, and Sang Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, pages 72–76, 2002.

[27] John B. Anderson. *Digital Transmission Engineering (IEEE Series on Digital & Mobile Communication)*. Wiley-IEEE Press, 2005.

[28] Vinay Sridhara and Stephan Bohacek. Realistic propagation simulation of urban mesh networks. *Comput. Networks*, 51(12):3392–3412, 2007.

[29] C. Siva Ram Murthy and B.S. Manoj. *Ad Hoc wireless networks: architectures and protocols*. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice Hall, May 2004.

[30] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[31] Nicolas Eude, Bertrand Ducourthial, and Mohamed Shawky. Enhancing ns-2 simulator for high mobility ad hoc networks in car-to-car communication context. In *7th Internation Conference on Mobile and Wireless Communication Networks (MWCN'05)*, 2005.

[32] Charles E. Perkins. Ad-hoc on-demand distance vector routing, 1997.

[33] Sung-Ju Lee, Elizabeth M. Belding-Royer, and Charles E. Perkins. Scalability study of the ad hoc on-demand distance vector routing protocol. *Int. J. Netw. Manag.*, 13(2):97–114, 2003.

[34] Thomas Clausen and Philippe Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003. available at: http://www.ietf.org/rfc/rfc3626.txt.

[35] The Network Simulator - ns-2, Information Sciences Institute - The University of Southern California. Website. Available: `http://www.isi.edu/nsnam/ns/` (Accessed Sept 16, 2007).

[36] Nam: Network Animator, Information Sciences Institute - The University of Southern California. Website. Available: `http://www.isi.edu/nsnam/nam/` (Accessed on Sept 16, 2007).

[37] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. *SIGSIM Simul. Dig.*, 28(1):154–161, 1998.

[38] Qualnet 4.0 network simulator, Scalable Network Technologies, Inc. Website. Available: `http://www.scalable-networks.com/` (Accessed Sept 16, 2007).

[39] OPNET, OPNET Technologies, Inc. Website. Available: `http://www.opnet.com/` (Accessed Sept 16, 2007).

[40] PHP - Hypertext Preprocessor. Website. Available: `http://www.php.com/` (Accessed Sept 16, 2007).

[41] Java Technology, Sun Microsystems, Inc. Website. Available: `http://www.java.sun.com/` (Accessed Sept 16, 2007).

[42] Swisstopo Geodata, Swiss Federal Office of Topography. Website. Available: `http://www.swisstopogeodata.ch/` (Accessed Sept 16, 2007).

[43] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review*, 62:1805, 2000.

[44] ETH Zurich Laboratory for Software Technology. Realistic vehicular traces (VANET traces). Website. Available: `http://lst.inf.ethz.ch/ad-hoc/car-traces/` (Accessed Sept 16, 2007).

[45] IEEE Standards Association. Website. Available: `http://standards.ieee.org/` (Accessed Sept 16, 2007).

[46] Apache Ant. Website. Available: `http://www.ant.apache.org/` (Accessed Sept 16, 2007).

[47] GNU Awk. Website. Available: `http://www.gnu.org/software/gawk/` (Accessed Sept 16, 2007).

[48] Swiss Federal Office of Topography. Formulas and constants for the calculation of the swiss conformal cylindrical projection and for the transformation between coordinate systems, May 06.

[49] INTERLIS - Ein Datenaustausch-Mechanismus für Land-Informationssysteme. Website. Available: `http://interlis.ch/` (Accessed Sept 16, 2007).