

# An Evaluation of Inter-Vehicle Ad Hoc Networks Based on Realistic Vehicular Traces

Valery Naumov  
Computer Science  
Department  
ETH Zurich  
8092 Zurich Switzerland  
naumov@inf.ethz.ch

Rainer Baumann  
Computer Science  
Department  
ETH Zurich  
8092 Zurich Switzerland  
baumann@tik.ee.ethz.ch

Thomas Gross  
Computer Science  
Department  
ETH Zurich  
8092 Zurich Switzerland  
thomas.gross@inf.ethz.ch

## ABSTRACT

Vehicular ad hoc networks (VANETs) using WLAN technology have recently received considerable attention. The evaluation of VANET routing protocols often involves simulators since management and operation of a large number of real vehicular nodes is expensive. We study the behavior of routing protocols in VANETs by using mobility information obtained from a microscopic vehicular traffic simulator that is based on the on the real road maps of Switzerland. The performance of AODV and GPSR is significantly influenced by the choice of mobility model, and we observe a significantly reduced packet delivery ratio when employing the realistic traffic simulator to control mobility of nodes. To address the performance limitations of communication protocols in VANETs, we investigate two improvements that increase the packet delivery ratio and reduce the delay until the first packet arrives. The traces used in this study are available for public download.

**Categories and Subject Descriptors:** C.2.2 Network Protocols: Routing protocols

**General Terms:** Performance, Algorithms, Design

**Keywords:** Realistic mobility model, simulation, VANET, inter-vehicle communication, vehicular network, ad hoc network, routing, greedy forwarding, broadcasting.

---

This research was supported, in part, by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322, and by a gift from the Microprocessor Technology Lab (MTL) of Intel Corp.

R. Baumann's current address: Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*MobiHoc'06*, May 22–25, 2006, Florence, Italy.  
Copyright 2006 ACM 1-59593-368-9/06/0005 ...\$5.00.

## 1. INTRODUCTION

Ad hoc (or self-organizing) networks operate without a predefined fixed (managed) infrastructure. Vehicular ad hoc networks (VANET) using 802.11-based WLAN technology have recently received considerable attention. In such a network vehicles equipped with Wi-Fi hardware constitute the mobile nodes (hosts). A number of routing protocols for VANET have been proposed and evaluated. These evaluations often involve simulators since management and operation of a large number of real vehicular nodes is expensive.

The literature shows that the results of performance studies of ad hoc network depend heavily on the chosen mobility model [8, 4, 43]. Many of the ad hoc network studies use random way point, random directions, and other models where the nodes change their speed and direction randomly [3, 7, 8, 29]. Vehicular movements are often based on the same random models with just higher maximum node speed [25]. Sawant et al. [38] describe vehicular movement on a highway by simulating a set of nodes moving with a constant speed along the longer side of a rectangular area of  $1000 \times 100 m^2$ . A slightly more advance model is used in [10] where drivers' decisions are introduced.

A step forward is the use of the real city maps and/or some kind of vehicular traffic simulators to produce somewhat close to reality vehicular movement patterns. These patterns are then used as an input for a network simulator (e.g., ns-2 [14]). Saha and Johnson [37] convert the maps of USA available from TIGER (Topologically Integrated Geographic Encoding) into a set of  $x, y$  points corresponding to the starting/ending points of each road segment. The authors introduce mobile nodes into the network by choosing some random starting and destination points for each node and using Dijkstra's single source shortest path algorithm to calculate the route. However, TIGER maps do not provide speed limit information.

Lochert et al. [23] use the traffic flow simulator Videlio developed by Daimler Chrysler AG to create a movement pattern for a small part ( $6.25 km \times 3.45 km$ ) of the city of Berlin. This movement pattern is then used as an input for the ns-2 simulator. No information is given on how the density/distribution of vehicles were chosen. The authors use only 1 city area; 10 data sources that are active for only five seconds and the 914 MHz Lucent WaveLAN card and circular radio are simulated.

The only publicly available source to download realistic traces that we could find is [15]. The authors recorded the

movement traces of the buses of the public transportation system in Seattle, Washington. However, these traces describe the movement only of the buses; they represent a tiny fraction of the total number of road traffic participants.

Here we describe a new source of realistic mobility traces for simulation of inter-vehicle networks<sup>1</sup>. These traces are obtained from a multi-agent microscopic traffic simulator (MMTS) that was developed by K. Nagel (at ETH Zurich, now at the Technical University in Berlin, Germany). This simulator is capable to simulating public and private traffic over real regional road maps of Switzerland with a high level of realism [35, 34].

The traces generated by MMTS provide an interesting starting point to study the performance of routing protocols in VANETs. We focus on an example of a reactive (non-geographic) routing protocol (AODV) and on example of a geographic routing protocol (GPSR). Both protocols experience significant performance problems.

An analysis of the performance problems suggests a number of improvements. A Preferred Group Broadcasting (PGB) strategy with the ability to split and merge hops significantly improves the performance of table-driven broadcast based routing protocols.

The Advanced Greedy Forwarding (AGF) technique is more fault-tolerant than traditional greedy forwarding. When used with the GPSR protocol, AGF improves the performance of GPSR for VANETs. We evaluate the suggested schemes against the above described mobility model. To increase the credibility of this study we model irregular radio channel behavior.

This study investigates AODV [33] and GPSR [18]<sup>2</sup> for this study, as they represent two different large classes of ad hoc routing protocols: reactive non-geographic and geographic with greedy forwarding. AODV and GPSR are well documented, tested in many research studies and are known as good performers in their classes of routing protocols. However, the PGB and AGF enhancements are not bound to AODV and GPSR and can be used with other routing protocols that depend either on flooding or greedy forwarding. Proactive routing protocols are not considered since they typically create significant administrative overhead in scenarios with very high mobility of hosts such as VANETs.

This paper is organized as follows. We describe the new mobility model in Section 2. In Section 3 we show the benefits of using realistic vehicular traces and discuss the performance problems, which protocols experience in VANETs. In Section 4 we introduce the Preferred Group Broadcasting (PGB) with route auto-correction strategy. In Section 5 we present the Advanced Greedy Forwarding (AGF) algorithm, that improves the performance of GPSR. The common simulation setup is presented in Section 6. The results of a comparative study of AODV and GPSR together with the evaluation of PGB and AGF are given in Section 7. Section 8 concludes the paper.

## 2. NEW MOBILITY MODEL FOR VANET

Since real vehicular traces are not available, a traffic simulator can be used to generate the movement of vehicles. However, driver behavior on a road is very complex. Driving is interactive, drivers must react to changing road conditions. Road conditions (e.g., congestion) depend in turn on the drivers' plans and behaviors. Thus, the choice of the traffic simulator in the end influences the relevance and viability of the obtained results.

Vehicular traffic simulators can in general be classified into microscopic and macroscopic simulators. A macroscopic simulator considers such system parameters as traffic density (number of vehicles per km per lane) or traffic flow (number of vehicles per hour crossing some point, usually intersection) to compute road capacity and the distribution of the traffic in the road net. From the macroscopic perspective, vehicular traffic is viewed as a fluid compressible medium, and, therefore, is modeled as a special derivation of the Navier-Stokes equations. In contrast, microscopic simulators determine the movement of each vehicle that participates in the road traffic. Thus, a microscopic traffic simulator is potentially a better choice for our research.

The multi-agent traffic simulator developed at ETH Zurich is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism. MMTS models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment. E.g., one individual might take a car and go to work in the early morning, another one wakes up later and goes shopping using public transportation, etc. Travel plans are made based on road congestion; congestion in turn depends on the travel plans. To resolve this situation a standard relaxation method is used.

The street network that is used in MMTS was originally developed for the Swiss regional planning authority (Bundesamt für Raumentwicklung). The major attributes of each road segment are type, length, speed, and capacity. The street network is simulated on a Beowulf Pentium cluster of up to 30 CPUs. With the help of MMTS, the consequences of construction sites, road modifications, new roads, etc. can be simulated and potential economical influence (e.g., travel time and price changes for public and private transport) can be estimated.

For the evaluation of inter-vehicle routing schemes, we use a 24 hour detailed car traffic trace file generated by MMTS. The file contains detailed simulation of the area in the canton of Zurich, this region includes the part where the main country highways connect to the city of Zurich, the largest city in Switzerland. Around 260'000 vehicles are involved in the simulation with more than 25'000'000 recorded vehicles direction/speed changes in an area of around 250 km x 260 km (Figure 1).

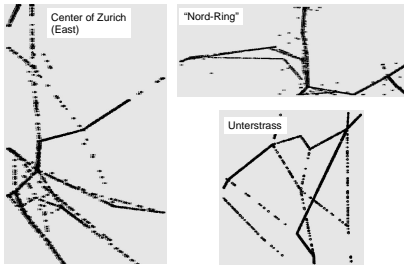
The car traffic simulator file describes in XML format the step by step movement of each vehicle within the 24 hour time period. To use that data we convert the XML input from the car traffic simulator into an ns-2 simulator movement pattern in Tcl format [14, 16]. Each run of the simula-

<sup>1</sup>These traces are publicly available from <http://www.lst.inf.ethz.ch/research>

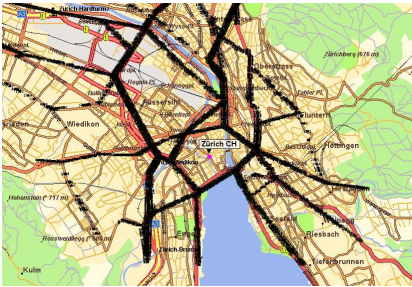
<sup>2</sup>We recently learned that some of the ideas of GPSR appear also in the Greedy-Face-Greedy (GFG) routing method [6]. Since we have access to an implementation of GPSR we therefore use this protocol as point of reference.



**Figure 1: Road network and car traffic in NE Switzerland.**



**Figure 2: Nam visualizer output: examples of different inner city regions.**



**Figure 3: Superposition of ns-2 movement pattern on a map of the center of Zurich.**

tor accepts as input a scenario file that describes the exact motion of each node together with the sequence of packets generated by each node as time progresses. However, the file resulting from car traffic simulator movement pattern contains too many nodes (vehicles) to be processed by the ns-2 simulator<sup>3</sup>.

Thus to allow simulation on a (communication) network

<sup>3</sup>To the best of our knowledge, none of existing event-driven ad hoc network simulators is capable of processing more than tens of thousands of nodes. We have 260'000 vehicles.

simulator, we select smaller subregions that are suitable for ns-2 simulations. The selected region contain 24 hour movement pattern of all the vehicles that travel inside or through a region. As a next step we capture different levels of activity of vehicles in the region (the number of events per time slot) and select three time periods that correspond to high density rush hour (more than 50 vehicles per km of road), medium density (30-40 vehicles/km), and low vehicle density (less than 15 vehicles/km). Each period has a length of 0.5 hour. And finally from each of these periods we create ten 300 s long scenario files, each shifted by 150 s from the previous one.

As we want to make sure that only vehicles that at a given moment in time are participating in the road traffic in the area take part in routing, we add a simple extension to the ns-2 simulator. When a node enters the area, a special Tcl command is issued that activates the wireless interface of a node. Nodes that end their trip inside the selected area (or leave the area) switch off their wireless interface.

The resulting ten ns-2 movement files per region and per density of vehicles are then used as the input for ns-2 simulation. The results of simulation are two files: a trace file and a nam (Network AniMator) file. The trace file contains a detailed description of each packet (Application/Routing/MAC packets) processed by each individual node as time progresses. The nam file can be used to visualize the simulation results with the help of the ns-2 network animator.

Figure 2 shows some nam-visualized example of different areas that we use in our research. In Figure 3 the nam screen-shot of mobile nodes (vehicles) moving about the center of Zurich-city is superimposed above a map of the region.

### 3. AODV AND GPSR IN VANET

We now discuss the performance of AODV and GPSR on the simulation scenarios described above.

#### 3.1 Influence of mobility model

The literature shows that the results of performance studies of ad hoc network depend heavily on the chosen mobility model [8, 4]. To allow our study to be compared with prior work, we start with an investigation of AODV and GPSR based on the random waypoint mobility model [7] and compare the results with our realistic vehicular traces.

For the random waypoint model we simulate 3 km x 2 km large area populated by 550 nodes that move about the area. Nodes' speeds vary between 15 km/h and 55 km/h; pause time 300 s, 200 s, and 0 s (constant motion).

From the vehicular traces we select the same area size representing the city region. We identify the 300 s time interval when the selected area is populated by at least 500 nodes and at most by 600 nodes. Pause time is modeled like follows: for 300 s pause time (static picture) we cancel all nodes mobility; for 200 s pause time we keep nodes static for 200 s and then allow movement; for 0 s pause time no restrictions are applied.

Table 1 shows the packet delivery ratio of AODV and GPSR when 10 CBR sources are simulated for 300 s with the random waypoint model and with our realistic vehicular traces.

The performance of both protocols shows noticeable dependence on the chosen mobility model. With the random waypoint model, GPSR always outperforms AODV; with the vehicular traces the situation changes to the opposite.

**Table 1: Packet delivery ratio of AODV and GPSR**

Pause time	Random waypoint model		City vehicular trace	
	AODV	GPSR	AODV	GPSR
300 s	99 %	99 %	34 %	13 %
200 s	90 %	93 %	29 %	12 %
0 s	83 %	88 %	27 %	12 %

The packet delivery ratio of both protocols in the city scenarios turns to be much lower than in the corresponding random waypoint model scenarios, and the average data path length (in number of hops) is around 1.7-2 times longer.

In the following two subsections we discuss how these performance problems occur.

### 3.2 AODV performance problems in VANET

Uncontrolled (also known as *blind*) flooding generates many redundant transmissions, which may cause the so called *broadcast storm* problem [27]. As the size of a network (number of hosts) grows, various performance metrics start suffering from the increasing administrative load [26, 27, 21] that prevents successful operation of the network. Moreover, in blind flooding a lot of collisions occur, and they sometimes lead to lost transmissions and prevent the dissemination of information to all the nodes [26].

In vehicular scenarios, data paths follow roads. Thus, the same road segment may be shared by many data routes and broadcasts. In the above mentioned city scenarios, from 70 % to 95 % of the network traffic is dedicated to broadcasting of route requests (RREQ). The metric of route discovery effectiveness is the number of RREQs received to the number of RREQs sent: the higher this ratio, the better. Logically, a broadcast ideally should be received by all of a node's neighbors. If many nodes cannot properly receive it, then this deficiency is a sign of a high level of congestion and of an ineffective route discovery. From the experiments in the city scenarios we see that half of the RREQ are dropped by nodes, due to collisions.

At the same time, the level of redundancy during RREQ broadcasts is extremely high in vehicular scenarios. If the nodes are lined up along the roads, then any rebroadcasts between the previously broadcasted node and its farthest neighbor that rebroadcasted the same RREQ are redundant.

These facts lead to the poor performance demonstrated by AODV in the above presented city scenarios, when data transmissions are jammed by broadcasts of RREQ, mostly redundant. Given the limited nature of wireless resources, minimizing the broadcasting load is a high priority in protocol design.

In Section 4 we describe a scheme that considers the quality of a connection to improve this connection's communication and furthermore to eliminate redundant broadcast retransmissions.

### 3.3 GPSR performance problems in VANET

When evaluating GPSR in VANETs, we observe that inconsistency of neighbor tables information leads to significant problems and dramatically low throughput (lower than original AODV). The impact of location inconsistencies on geographic routing protocols in the case of GPSR is analyzed in [19]. However, the authors just summarize the influence of different factors on the protocols performance and do not propose any method to combat the problem.

Outdated information in the neighbor tables can be healed with the more frequent beaconing, as the GPSR authors suggest [18]. However, this would certainly increase the congestion and the potential for collisions.

To improve the accuracy of neighbor tables we may add information about the node's speed and/or its direction of travel to the beacon. However, such information would not help to discover new neighbors. Here we try to identify the main source of the problem and propose a solution. We experimented with the 4 km long straight part of the highway with nodes moving up to 30 m/s in both directions.

In GPSR, when packets are forwarded in greedy mode, the selection of the next-hop node is based on the neighbor table of the forwarding node. But the neighbor table is not always up-to-date, so the selected neighbor may be not optimal or even may not be a neighbor any more.

According to the experiments with GPSR in VANETs, when a node accesses its neighbor table, on average 10-20 % of the neighbors listed in the table are not anymore within the communication range. When the node picks up a neighbor to forward the packet, only in 20-35 % of the cases the chosen node is really the closest one to the destination.

To estimate the influence of picking-up non optimal neighbor from the table we perform an experiment that is only possible in a simulator: we modify the GPSR code so that a node (before forwarding a packet) directly picks up the node that is currently its closest neighbor toward the destination, instead of selecting a next hop node from the neighbor table. Although this change improves the packet delivery ratio of GPSR from about 10 % to about 16 % in the highway scenarios, the problem of poor performance remain.

When a packet is sent from a source node, the location of the destination node is written into the packet header. This information is never updated while the packet is traveling. In a highly mobile environment such as a highway, the destination node may travel a substantial distance in a short period of time. If a packet is in greedy mode, the forwarding node tries to find a node that is as close as possible to the location stored in the packet header. It does not check if the destination node is in its neighborhood. So if the destination node moves away and another node moves near the former location of the destination node, this node is selected as the next hop. And now because no other node can be found that is closer to the destination location, the packet enters into perimeter mode and eventually is dropped.

To see whether and how this fact influences the performance, we force the updates of the destination coordinates on every hop, using the simulator's global knowledge. This change significantly improved the performance of GPSR from around 10 % up to 56 % of delivered data packets in a straight road scenario. When the position of the destination is updated only at the last hop, the improvement still is noticeable: 50 % of the data packets are delivered.

Thus, based on the above observation, in Section 5 we suggest a greedy forwarding technique that significantly improves GPSR performance in VANETs.

## 4. PREFERRED GROUP BROADCASTING

### 4.1 Preferred Group Broadcasting Algorithm

Preferred Group Broadcasting (PGB) is a broadcasting mechanisms that aims (1) to reduce control messages overhead, by eliminating redundant transmissions and (2) to ob-

tain stable routes with the ability to auto-correct. While a minimization of a routing load is very desirable in any of the possible ad hoc scenarios, the stability of a chosen route becomes especially important in an environment where fast moving vehicles are used as wireless hosts.

In a broadcasting-based route discovery process, no special criteria are used when choosing the intermediate nodes. Thus it can easily be the case that two nodes in a path are very close to each other, or vice versa are separated by a distance close to the maximum communication range. Neither of the cases is desirable. Short distances between hops imply a high number of hops in the path. On the other hand, when a hop length is close to a maximum coverage range, the connection can be easily lost if one of the nodes moves out of the range, or slightly changed interference jams the weak signal.

Moreover, poor connection quality leads to throughput degradation, due to an increased number of errors. If an adaptive data-rate is used, then a high bit error rate forces the data-rate to decrease, and this step may lead to a bottleneck problem in the current node.

Another factor that seriously limits the performance of ad hoc networks is the “hidden terminal” problem. Using RTS/CTS handshake, node *A* “secures” the transmission of a data packet to node *B*. However, only the nodes that are within the receiving range from node *A* and *B* adjust their NAV (Network Allocation Vector) not to interfere with the data transmission from *A* to *B*. Other nodes just perform the carrier sense step, and whenever the medium is free may start a transmission. If then the signal to noise ratio reaches some threshold (e.g., 10 dB) in node *B*, then *A*’s transmission fails. The weaker the signal from *A* in node *B*, the easier it is that the communication *A* – *B* can be interrupted by a hidden terminal.

PGB addresses all of the above cases by allowing only certain nodes to rebroadcast a route request. This restriction significantly limits the set of possible intermediate nodes and in turn automatically reduces the routing load. PGB works in both cases: when the information about nodes coordinates is available and when not.

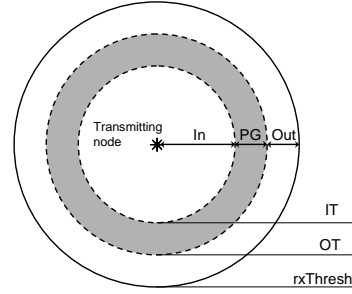
PGB classifies each node that receives a broadcast packet (e.g., route request) into one of the three groups, based on the sensed signal level (Figure 4):

- Preferred group (PG) – the preferred set of nodes;
- IN group – nodes with a signal stronger than in PG;
- OUT group – nodes with a signal weaker than in PG.

For simplicity we depict the communication range of a node and the power levels for the IN and OUT groups as circles in Figure 4 and 6, but our algorithm by no means assumes that the range is indeed circular.

To classify a node, the power of a received signal is compared to two values: Inner Threshold (*IT*) and Outer Threshold (*OT*). We assume that signal power is measured in **dBm**<sup>4</sup>. Both thresholds have the following relation to the

<sup>4</sup>The IEEE 802.11 standard defines a mechanism by which RF energy is to be measured by the circuitry on a wireless NIC. This numeric value is an integer with an allowable range of 0-255 (a 1-byte value) called the Receive Signal Strength Indicator (RSSI). Different hardware vendors use different strategies to convert the sensed signal power into RSSI. However, in many cases these RSSI values have close relation to the signal power measured in dBm [2].



**Figure 4: Node groups in Preferred Group Broadcasting.**

receiving threshold (*rxTh*) – the power of a signal that corresponds to the maximum transmission range:

$$IT = rxTh + f_{IT}$$

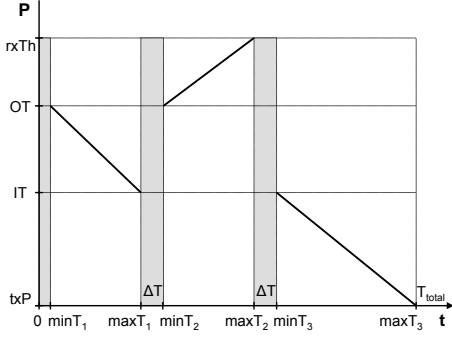
$$OT = rxTh + f_{OT}$$

Here  $f_{IT}, f_{OT}$  are positive integer values (measured in **dB**), managed by the system to adapt to changing neighborhood conditions, and/or different types of traffic (as will be discussed in Section 4.3). If the sensed signal power is more than *IT* the node classifies itself as belonging to the IN group. If the signal power is less than *OT*, the node falls into the OUT group. Otherwise, if the signal power is more than *OT* and less than *IT*, the node belongs to the preferred group (PG). Each group has an *id*. Nodes from the PG have *id* = 1 and the highest priority to be chosen as relays, then nodes from the OUT group (*id* = 2), and finally from IN (*id* = 3).

The groups can also be constructed if information about nodes’ coordinates is available. In this case a sending node adds its coordinates in a broadcast packet. The coordinates together with the sensed signal power determine the classification of a receiving node.

The values of  $f_{IT}, f_{OT}$  are adjusted by each node to a certain number (a configurable parameter) of nodes within the closeup power range of  $\Delta P = IT - OT$ . Consider nodes *A* and *B*. Whenever *A* transmits, *B* senses a power level  $rxP_A$  of *A*’s signal. If the information of the transmitting power  $txP_A$  is available (either all nodes use the same power, or this information is added in each packet), then node *B* can estimate the signal loss equal to  $|txP_A - rxP_A|$ <sup>5</sup>. If this value is less than  $\Delta P$ , then when some other node *C* sends a broadcast to forward, nodes *A* and *B* may potentially classify themselves belonging to the same PG. Thus, each node tries to keep the given number of neighbors in its closeup power range by adjusting the value of  $f_{IT}$ . The value of  $f_{OT}$  is set once and changed only in case of the information dissemination traffic (Section 4.3). In a sparse network the  $\Delta P$  of each node increases to maintain PG-based connectivity, if possible (if there are neighbors). If no information

<sup>5</sup>The information about the neighbors and received signal powers from them can be also obtained from the MAC-level beacons, as according to 802.11 standard wireless hosts in both infrastructure mode and ad hoc mode use beacons to indicate their presence and/or readiness for data transmissions [1].



**Figure 5: The correspondence of the node's hold off time and the sensed signal power.**

about neighbors and signal powers from them is available; the values of  $f_{IT}$  and  $f_{OT}$  are equal to the initial default user settings.

The PGB algorithm makes receiver-based decisions to rebroadcast, thus PGB can operate with different kinds of mobile hosts, which may have different hardware installed, resulting in different coverage ranges and  $rxTh$ . If PGB is used for obtaining routes (forwarding RREQs), two other fields should be added to a broadcast packet – the transmission power ( $txP$ ) and the  $rxTh$  of the sending node. With the help of these fields, a receiving node may estimate the signal loss and whether its own hardware (maximum transmission power, antenna gain, etc.) is capable of transmitting effectively back to the sending node.

After classifying itself a node waits for a *hold off* period before taking a decision whether to rebroadcast a packet or not. During this time the node listens to other retransmission of the same packet. The hold off time and the strategy for making a rebroadcasting decision depend on the group the node belongs to.

The hold off time  $hoTime$  for each node in group  $i$  is calculated using the following formula:

$$hoTime = T_i + jitterT_i \quad (1)$$

$$\Delta P_i = |rxTh - txP| - f_i \quad (2)$$

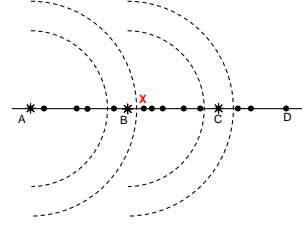
$$T_i = \Delta P_i \cdot dt_i + minT_i$$

$$jitterT_i \in [0, dt_i)$$

where  $f_i$  is equal to either  $f_{IT}$  or  $f_{OT}$  depending on the group ( $f_{1,2} = f_{OT}$ ,  $f_3 = f_{IT}$ ),  $dt_i$  is the maximum value for the jitter time in group  $i$ . Jitter is used to prevent concurrent rebroadcasts from the nodes that have the same value of  $txP$ .

The dependence of the hold off time on the sensed signal power is schematically shown on Figure 5. The values  $minT_i$  and  $maxT_i$  are fixed configurable parameters. As hardware may influence the timing of nodes, a time interval  $\Delta T$  separates the values of  $maxT_{i-1}$  and  $minT_i$  for  $i = 2, 3$ , with  $minT_1 = \Delta T/3$ . From these parameters and  $f_{IT}$  and  $f_{OT}$ , the values of  $dt_i$  are computed as follows:

$$\begin{aligned} \max(\Delta P_i) \cdot dt_i &= maxT_i - minT_i \\ dt_i &= \frac{maxT_i - minT_i}{\max(\Delta P_i)} \end{aligned} \quad (3)$$



**Figure 6: Preferred Group Broadcasting example on a straight road.**

where  $\Delta P_i$  is computed by Equation 2. The exact computation of  $dt_i$  values depends on the group and is discussed later in this section.

To understand the basic principles of PGB, consider a simplified situation when the nodes are distributed along a straight road (Figure 6). Node A initiates a route discovery and sends a RREQ broadcast packet to a currently unknown destination D. Ideally only one node from each preferred group should forward the request (we assume for now that the corresponding groups are not empty). If a PG node receives the same RREQ twice from two different nodes, it can drop it, since the node assumes that some other nodes have already rebroadcasted the request. However, if two nodes from the same PG group occasionally rebroadcast the request almost at the same time, then this RREQ will never propagate further. No other node will rebroadcast it, since every node will receive two consequent rebroadcasts of the same RREQ, which will be understood as the RREQ was successfully forwarded by two PG nodes from two different groups.

To avoid the situation described above, every node rebroadcasting a RREQ adds in the packet the id of the predecessor and, if available, the predecessor's coordinates together with the node's own coordinates (again, if available). For node B in Figure 6 the predecessor from which B has received the first copy of the RREQ is node A. Each node receiving the same copy of a RREQ counts the number of unique predecessors of the corresponding nodes. Based on this number accumulated within the hold off time a node makes its forwarding decision.

The decision-taking process can be optimized if nodes have information about their coordinates from GPS or another positioning system. In this case, a node listening to other broadcasts during the hold off time may reconstruct positions of each sending node and each sending node's predecessor. Consider the situation when nodes C and B receive the first copy of RREQ from node A and are now waiting for some hold off time. Assume B has a shorter hold off time, so it rebroadcasts first. Node C extracts from the RREQ received from B the information about the position of B and B's predecessor (node A). Now C analyzes the triangle ABC. If the triangle height from node B is less than a certain value, specified as a parameter, C drops the packet.

If not only coordinates are available, but also feedback from a navigation system, together with the triangulation method, nodes may use coordinates of crossings in making the rebroadcasting decision, to increase the chances of

getting the broadcast propagated to all the roads joining a crossing.

The shortest hold off time (the highest priority to rebroadcast) have nodes from the PG, then the nodes from OUT group, and then from IN. Inside each group nodes that sensed the received signal power closer to  $OT$  threshold have higher priority.

**PG nodes** ( $OT \leq rxP < IT$ ,  $f_i = f_{OT}$ ) – A PG node belongs to the preferable set of nodes that should serve as relays. That is why the hold off time for this group is the shortest one (Equation 1). The  $dt_1$  value computed from the Equation 3 is:

$$dt_1 = \frac{maxT_1 - minT_1}{f_{IT} - f_{OT}}$$

The nodes from the PG that have  $rxP$  closer to the  $OT$  choose shorter delays than those closer to  $IT$ .

If the PG node receives the same broadcast from two other nodes with different predecessors in the hold off time, the node drops the packet. If the geographical information is available, the node analyzes the triangle height for each received copy of the same RREQ (as described above). The packet is dropped if the triangle height is less than a certain threshold (configurable parameter).

**OUT node** ( $rxP < OT$ ,  $f_i = f_{OT}$ ) – If the preferred group is empty, or due to whatever reason none of PG nodes rebroadcasts the packet, then nodes from the OUT group should forward the broadcast. The hold off time is again computed by Equation 1. The value of  $dt$  (from Equation 3):

$$dt_2 = \frac{maxT_2 - minT_2}{f_{OT}}$$

The behavior of an OUT node is the most complex among three groups. Ideally, a node from the OUT group should hear at least as many retransmissions of a broadcast packet as a node from the PG. Nodes from the OUT group check that the transmission from PG was successful and was rebroadcasted further. If, however, the transmission from the PG did not reach any further nodes, a node from the OUT group should rebroadcast the packet. The idea behind is simple – better to have a potentially less stable connection than no connection at all. Thus PGB tries to keep a network connected if that is physically possible.

In the ideal situation depicted in Figure 6 an OUT node hears the same broadcast three times. The first time the node (e.g., node  $X$ ) hears the request when it classifies itself as an OUT node. The node should hear the broadcast a second time when some PG node (in the example node  $B$ ) rebroadcasts the request. And finally the node should hear the broadcast a third time when the request is forwarded further by a node  $C$  that received the transmission from node  $B$ . Thus, when an OUT node hears the same broadcast for the second time, it randomly chooses a new hold off time from the interval

$$\begin{aligned} hoTime &\in [t_{min}, t_{max}] \\ t_{min} &= maxT_3 \\ t_{max} &= t_{min} + 2 \cdot dt_3 \end{aligned}$$

If after this additional delay (big enough for a next-hop IN node to rebroadcast a packet after waiting its  $hoTime$  <sup>6</sup>)

<sup>6</sup>IN nodes have highest delay among all groups.

the node still did not detect the third rebroadcast, the node rebroadcasts the packet itself.

When a node's coordinates are available the above described triangulation method is applied before increasing the broadcast counter.

**IN node** ( $IT \leq rxP$ ,  $f_i = f_{IT}$ ) – If there are no nodes in the PG and OUT groups, nodes from the IN group should rebroadcast the packet. We want to have a connected network with stable routes, at the same time the hop length should not be too short to avoid long paths (with high hop counts). Thus, the nodes from the IN group have the longest hold off time among all three groups. The  $dt_3$  value in Equation 3 is:

$$dt_3 = \frac{maxT_3 - minT_3}{(txP - rxTh) - f_{IT}}$$

An IN node drops a previously received broadcast packet if it hears during the hold off time the same broadcast from another node with a different predecessor.

An IN node also drops the packet if the height of some other sensed rebroadcast, calculated with the triangulation method is less than a set threshold.

To allow propagation in multiple directions from a source node at the start, the number of heard rebroadcasts (dropping threshold) is doubled for all nodes that hear directly the source. (In this case the initiator of a broadcast and the predecessor are the same nodes.)

## 4.2 Splitting/merging of hops

During our studies with VANETs we observed that two particular cases occur quite often:

1. in a data path, the distance between two subsequent nodes  $A$  and  $B$  increases; eventually the route is broken;
2. for some part of a data path  $A - B - C$ , the distance between nodes  $A$  and  $C$  decreases; eventually there is a redundant retransmission for node  $B$ .

To tackle the above described cases we integrated into the PGB algorithm an automatic route optimization by allowing splitting and merging of hops. This optimization requires interaction with MAC layer or promiscuous listening mode.

**Hop splitting** – In situation (1), whenever node  $A$  detects that the connection quality to node  $B$  becomes poor, it sets a special splitting bit in all RTS packets transmitted to node  $B$ . Node  $B$  does the same in all CTS packets to  $A$  (note that node  $B$  can also be the initiator of this process). Every intermediate node that eventually overhears these transmissions compares the power level  $P_A$  and  $P_B$  of the received signals from nodes  $A$  and  $B$ . It checks if these powers are high enough so that the current node can effectively communicate with  $A$  and  $B$ . If it the case, the node backs off for time  $t$ .  $t$  is calculated as following  $(1 - P_A/P_B) * dt$ , where  $dt$  is small delay from 1 to 5 ms (configurable). The node that has the shortest back-off time broadcasts a RREP packet to node  $A$  with the new path to node  $B$  set over the current node. Node  $A$  corrects the path and resets the splitting bit in subsequent transmissions. Other nodes, hearing this broadcast, do not transmit any other RREPs.

This works in all the cases independent of the movement directions of nodes  $A$ ,  $B$  and  $C$ , i.e. movement in the same direction, opposite directions, etc.



**Hop merging** – In situation (2), whenever node  $C$  hears  $A$  trying to send to  $B$  a packet that node  $B$  then sends to  $C$ ,  $C$  sends RREP to  $A$ , provided the quality of the signal received from  $A$  was sufficient. After receiving the RREP, node  $A$  updates its routing table and directly sends the next data packet from the same flow to node  $C$ .

Splitting and merging allows effective auto-correction of a route on the fly without employing a more expensive broadcasting-based route-repairs.

### 4.3 PGB for information dissemination

The PGB algorithm can work with the mixed traffic when RREQ broadcasts and information dissemination broadcasts coexist<sup>7</sup>. In case of information dissemination there is no need to worry about the stability of a connection. Thus, when a PGB node receives an information-dissemination broadcast, it sets the values of  $f_{OT} = rxTh$  and  $f_{IT} = txP$  forcing the PG's computation of the hold off time to be used.

A node that has the shortest delay broadcasts first and starts listening that the packet is rebroadcasted further. If no further rebroadcasts happen (no next hop neighbors are currently present), the same node repeats the broadcast after waiting  $10 \cdot \max T_1$ . The process is repeated until a next-hop neighbor appears and rebroadcasts the packet further.

Nodes that heard the same broadcast from two other nodes with different predecessors drop the packet.

### 4.4 Related work

To find a multi-hop route MANETs commonly employ routing algorithms that use *flooding*. Many ad hoc routing protocols, multicast schemes, or service discovery schemes depend on flooding [33, 17, 11, 28, 18]. Flooding can also be used for information dissemination. In a survey of inter-vehicle communication Luo and Hubaux also stress the importance of broadcasting for data dissemination in VANET [24].

A number of research groups have proposed optimized broadcasting techniques that aim to minimize the administrative load created by a routing protocol while attempting to guarantee reliability. Each of these techniques has its advantages and disadvantages. Generally the existing broadcasting techniques can be classified as: *simple flooding*; *probability based methods*; *area based methods*; *neighbor knowledge methods*.

The disadvantages of simple flooding have already been discussed at the beginning of this section. In probability based methods [27], a node makes the rebroadcasting decision only if the probability to cover a substantial additional number of nodes is high. Either some percentage of randomly chosen nodes rebroadcast packets, or the nodes that have not heard enough rebroadcasts of the same RREQ from other nodes perform a broadcast. Probabilistic schemes tend to be ineffective in networks with a non-uniform distribution of nodes (sparse spots).

In area based methods a node evaluates the additional coverage area based on all received redundant retransmissions [27]. A node either possesses geographical knowledge (GPS), or determines its own relative position to other nodes by analyzing the power of received signals. A node rebroadcasts the packet only if the additional area is more than a threshold value. Area based methods only consider the

coverage area of a transmission, not whether nodes actually exist within the area. Thus, they may fail to find a route in a sparse ad hoc network.

A node that runs a neighbor knowledge method for broadcasting control obtains the list of 1-hop or 2-hops neighbors by periodic exchange of hello-messages with other nodes [22, 30, 31, 32, 40]. In methods that exploit 2-hop set of neighbors [22, 31, 32, 40] nodes add also the list of direct nodes in each hello-packet. In some neighbor knowledge methods, a node rebroadcasts if it would reach any additional nodes; the node refrains from rebroadcasting otherwise. In some other methods, the special algorithm determines the relay nodes that rebroadcast route requests. Due to periodic hello-messaging neighbor knowledge methods impose additional load on a network. In a highly mobile environment this load may be very high. On the other hand, infrequent exchange leads to stale information in the neighbor tables, and as a result to ineffective routing.

The developers of GERA [45] suggest to modify the RTS/CTS handshake so that each node may determine its own priority as a relay for forwarding a data packet. The priority is based on the relative location of the node compared to the distance between the transmitter and the intended final destination. However, broadcasts in ad hoc networks normally do not use RTS/CTS, and for data transmission it is optional. Also the effectiveness of RTS/CTS handshake is questionable, e.g., many theoretical studies, simulations, and real world experiments show that when no RTS/CTS handshake is used, a network's throughput improves [9, 36].

## 5. ADVANCED GREEDY FORWARDING

Based on the observation discussed in Section 3.3 we design the Advanced Greedy Forwarding (AGF) algorithm to significantly improve GPSR performance in VANETs.

Both the sources and the destinations nodes inform each other with the help of the location discovery service (e.g., reactive location service [23]) about their moving directions and speeds – *velocity vectors*<sup>8</sup>. Velocity vector information is also added into HELLO beacons of all nodes. Velocity vector requires additional two bytes to store the information about nodes speed and direction. The first byte encodes the direction in the range of 0-127 (the MSB indicates whether velocity vector information is available or not). The second byte stores the speed in km/h (enough for representing the allowed maximum speed in most countries).

Also the information about the packet travel time is added in a data packet header. Every node forwarding a data packet adds its own processing time into packet header. A next hop node is chosen based on the velocity vectors information stored in the neighbor tables.

A node receiving a data packet checks if the destination is listed in its neighbor table and the entry is still valid, taking into account the packet travel time and the node's and the destination's velocity vectors. If that is the case, the node sends the packet directly to the destination. If the destination is in the neighbor table, but the new position estimation tells that the destination is most likely already out of the range, then the node closest to the new position of the destination is chosen as a next hop.

<sup>7</sup>We assume, that a routing protocol may distinguish between two types of packets: RREQ and information dissemination.

<sup>8</sup>In this study we do not use any location service, but rather obtain the information from the simulator. The reason for that is described further in Section 6.



```

NT - neighbor table; TTL - Time To Live in number of hops
proc AGF(PKT, DST)
  if TTL > 0 then TTL--; else return FAILURE; endif;
  Le = estimated current DST location
  if Le should be in my range then
    if DST is in my NT then
      send PKT to DST
      if succeeded then return SUCCESS
      else remove DST from NT; endif
    endif
    broadcast non-propagating RREQ searching for DST
    wait for any answer within X ms
    if RREQ is answered by DST or DST-in-NT-node then
      add answered node to NT
      send PKT to answered node
      return SUCCESS
    else
      TTL=2; /* limit the number of retries left */
    endif
  endif
  while there are neighbors in NT closer to Le than me
    N = next closest to Le neighbor
    if N should be in my range then send PKT to N; endif
    if succeeded then return SUCCESS; endif
    remove N from NT
  endwhile
  return FAILURE

```

Figure 7: Advanced Greedy Forwarding

If no destination is found in the neighbor table the node consults the packet travel time and estimates whether it may potentially reach the position of the destination recorded in the packet header within one hop, taking into account a distance potentially traveled by the destination node within the packet travel time. If yes, the non-propagating broadcast is sent around, with the search for the destination. If no answer is received (either from the destination, or the node that has the destination in its table and is closer to the destination, than the current node), then the next closest to the destination node is chosen, and the process repeats.

Figure 7 shows the pseudo-code for the AGF.

## 6. SIMULATION SETUP

In our experiments we use the 2.28 release of ns-2 simulator. The ns-2 simulator was validated [16] and verified in a number of later publications, e.g., [21]. Its interface with Tcl/Tk allows changing the behavior of nodes dynamically during the simulation execution. We exploit this feature in the extension for switching the wireless interface of nodes on and off.

We use the default setups for the GPSR and AODV routing protocols and enabled the optimizations designed for better scalability. The beaconing interval of GPSR is left at the default value – 3 s. In case of GPSR we want to test only the ability of this protocol to successfully route data packets to their destinations in VANETs by employing greedy forwarding and perimeter mode routing<sup>9</sup>. Thus, we do not use any location service, but rather provide coordinates of destination from the simulator’s global knowledge.

**Radio Propagation Model** – ns-2 implements three different radio propagation models: Free Space Model, Two-ray Ground Reflection Model, and probabilistic Shadowing Model [14]. Whenever a node receives a packet, the signal power is determined by applying the chosen model. If the

Table 2: Common values for deviation  $\sigma_{dB}$

Environment	$\sigma_{dB}$ (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 3: Common values for the path loss exponent  $\beta$

Environment		$\beta$
outdoor	free space	2
	shadowed urban area	2.7 to 5
in building	line-of-sight	1.6 to 5
	obstructed	4 to 6

signal power is below the receiving threshold, the packet is marked as error and is dropped by the MAC layer.

The Free Space and Two-ray Ground Reflection models are often used in simulation studies of routing protocols [7, 12, 18, 42]. These radio models are based on distance across flat terrain; radio communications are received perfectly within some circular “range” and not at all outside of that range. Kotz et al. [20] analyzed a set of publications on routing protocols from 1995 till 2003 and summarized the common assumptions that are used in the evaluation of those protocols: *a radio’s transmission area is circular; all radios have equal range; if I can hear you, you can hear me (symmetry); if I can hear you at all, I can hear you perfectly; signal strength is a simple function of distance.*

However, according to many researchers these assumptions do not hold and the basic simple propagation models alone do not reflect the complexity of real radio systems [44, 41, 20]. Real radios demonstrate a strikingly non-uniform non-circular behavior [20, 44]. With real radios, spatial or temporal signal fluctuations may cause rapid changes in network connectivity. As a consequence many of the suggested routing algorithms and protocols may perform much more poorly under such dynamic conditions; some may even fail to converge and thus fail to work in real world environments.

Thus, in this research we use the probabilistic Shadowing model. Unlike the Free Space and the Two-ray Ground Reflection models, the Shadowing model does not predict the receiving power as a deterministic function of  $d$ . Instead, the Shadowing model uses a statistical approach to calculate the receiving power. It takes into account multi-path propagation effects. The model is described by Equation 4.

$$\left[ \frac{P_r(d)}{P_r(d_0)} \right]_{dB} = -10\beta \log \left( \frac{d}{d_0} \right) + X_{dB} \quad (4)$$

$X_{dB}$  is a gaussian distributed random variable with zero mean and standard deviation  $\sigma_{dB}$ .  $\beta$  is the path loss exponent. Some typical values (determined empirically) for  $\sigma_{dB}$  and  $\beta$  are shown in Tables 2 and 3.  $P_r(d_0)$  is the reference received power for a close-in distance  $d_0$ .

In [39, 13] real world experiments with inter-vehicle communication are presented. [39] investigates the performance achievable by 802.11b between a pair of communicating vehicles in different environments. The communication range varies, depending on the environment between 450 m and 600 m (sometimes up to 1 km). Results from [13] are similar – vehicles could communicate up to a distance of 550 m.

<sup>9</sup>The code for GPSR is taken from [18].

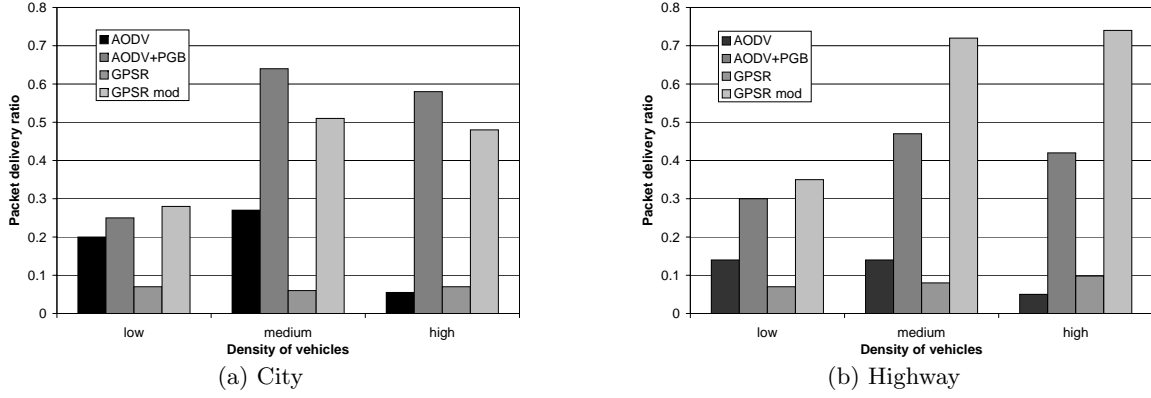


Figure 8: Packet delivery ratio for city and highway scenarios.

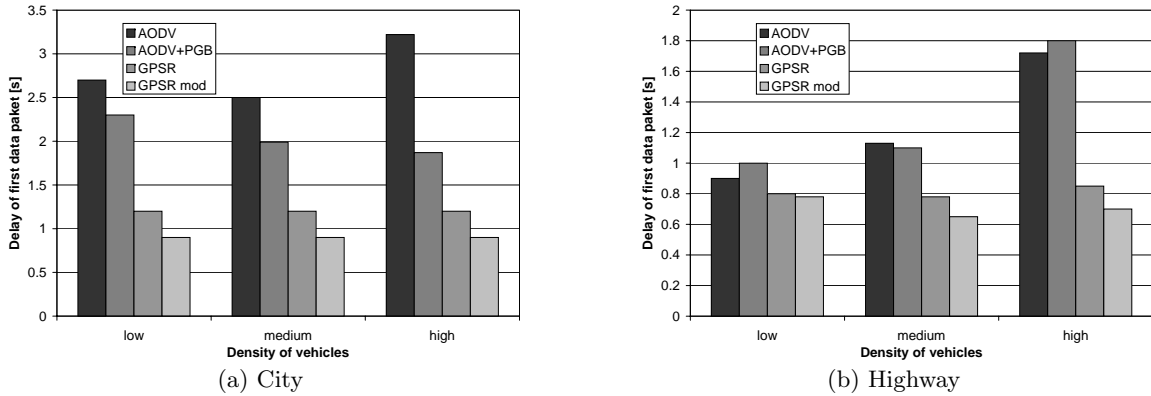


Figure 9: Delay of a first data packet for city and highway scenarios.

In response to these results, we adjust in our experiments the Shadowing model so that the maximum possible communication range (with 75 % probability of a successful transmission) stays 400 meters for the city scenarios and 500 meters for the highway. We choose  $\beta = 2.5$ ,  $\sigma_{dB} = 4$  for the highway scenarios and  $\beta = 3.5$ ,  $\sigma_{dB} = 6$  for the city ones.

In our research we model the 2.4 GHz band with 2 Mbps data rate. We chose a fixed data rate since we do not aim to investigate the maximum achievable throughput in VANETs; we want to investigate the ability of protocols to successfully find and maintain routes in a vehicular ad hoc network.

In this paper we present results for three different densities of nodes (*low*, *medium*, *high*) in the following movement scenarios: **highway** (averaged over 3 different highway areas, each 10 sub-scenarios for every density of vehicles) and **city** (averaged over 3 different city areas, each 10 sub-scenarios for every density of vehicles). These aggregate names (*highway* and *city*) are used further in the paper to refer to the whole subset of corresponding scenarios. 20 CBR traffic sources with a sending rate of 4 packets/s are considered. Sources start generating data packets within the first 50 s of the simulation time and stop generating data packets 50 s before the simulation ends to avoid data packet drops due to the end of the simulation. All source/sink nodes stay inside the simulated area for the duration of their communications.

The values of the configurable parameters in the PGB algorithm are:  $\Delta T = 10ms$ ;  $maxT_1 = 15ms$ ;  $maxT_2 = 40ms$ ;  $maxT_3 = 100ms$ ;  $f_{IT} = 12dB$ ,  $f_{OT} = 6dB$ .

## 7. SIMULATION RESULTS

We present the following metrics for comparing the performance of original AODV and GPSR with our modification of these protocols:

- *Packet delivery ratio (PDR)* – the fraction of the data packets originated by an application that each routing protocol delivers.
- *Average delay of a first data packet* – the average difference between the time the first data packet is originated by an application and the time this packet is received at its destination. The time expanded to obtain a route (if there is none in the sender's routing table) is included in this figure.
- *Normalized routing load (NRL)* – the ratio of administrative routing packet transmissions to data packets delivered. A data packet is counted as delivered when it is received by the destination node. When counting transmissions, each hop is counted separately.

Figure 8 shows the packet delivery ratio for the city and highway scenarios with different densities of vehicles. For all

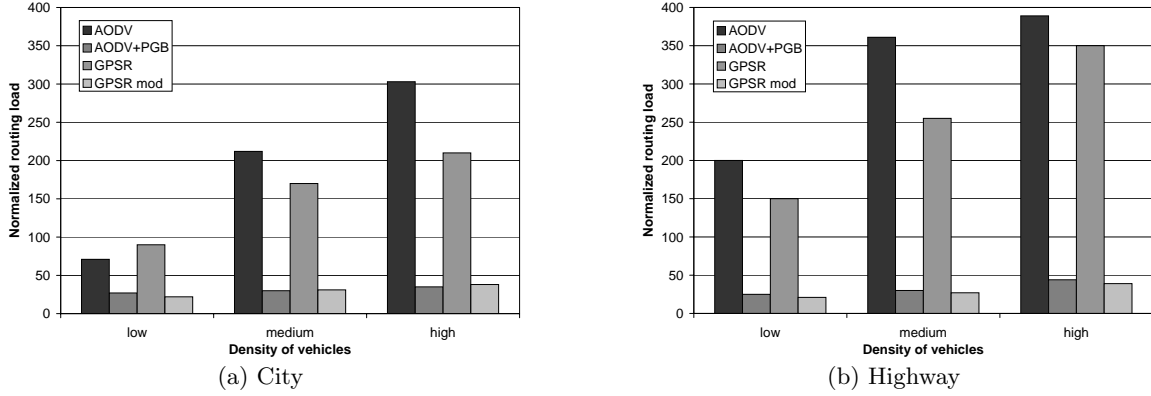


Figure 10: Normalized routing load for city and highway scenarios.

densities AODV+PGB shows better or similar results with other protocols. Surprisingly, GPSR performed poorly in the city scenario, only 5-7 % of the data packets are delivered. Also the modification of this protocol shows lower PDR than AODV+PGB, although the result is noticeably higher than the one obtained from standard GPSR.

The performance problems of AODV occur mainly due to the high administrative load (see also [26]). One problem of GPSR is that it chooses from the set of neighbors the longest possible hop, which in turn sometimes leads to a poor reception quality and as a consequence to drops of data packets. Another problem of GPSR is that in a complex city environment greedy forwarding often fails, and consequently packets enter into perimeter mode. Perimeter mode turned out to be ineffective in city scenarios and often fails to terminate. Therefore for routing in VANET GPSR may benefit from the approach when a signal power is considered for choosing a next hop node (e.g., like in PGB) and from a more suitable algorithm to resolve the local maximum problem.

The highway scenario is geographically less sophisticated than the city setup, thus the modified GPSR shows noticeably better PDR in the highway area.

Another interesting metric is the delays of the first data packet. The results for this metric are shown in Figure 9; the original and the modified GPSR stay always better than AODV. For AODV, the route discovery process precedes every first data transmission to an unknown destination, this adds to the delay of the first data packet. Although in PGB we introduce small additional delays, AODV+PGB demonstrates similar (sometimes even lower) delays than the original AODV. PGB induces a lower routing load than AODV, as a result less data transmissions are delayed waiting for broadcasts to finish. Thus, PGB's approach with delayed retransmissions of broadcast packets seems to pay off also when this QoS metric is considered.

Figure 10 shows the NRL. AODV+PGB demonstrates in all the cases a significantly lower routing load than AODV, thus the primary goal of PGB to reduce the administrative load is achieved. The NRL of GPSR stays high and is compatible with the one of AODV, since GPSR uses beaconing. However, the load of GPSR is more evenly distributed over the time than for AODV, thus it is less "disturbing" to the network performance. Modified GPSR has the same beaconing load, but better PDR than the original version of this protocol, which in turn lead to the better values of the NRL.

In the scenarios with low density of nodes the network often becomes disconnected, leading to the low PDR of all tested protocols. Thus, a strategy that forces the nodes that detect a next-hop disconnection to wait for some timeout period before dropping a packet should be considered as an option to tackle the low PDR problem.

## 8. CONCLUDING REMARKS

Few traces of actual node movement in realistic scenarios for ad hoc networks are available, so a car traffic simulator used by city planners and civil engineers is the next best choice to obtain mobility information. In this paper we report on an investigation of the effectiveness of AODV and GPSR in an inner city environment and on a highway segment. This evaluation is based on traces obtained from a microscopic vehicle traffic simulation on the real road maps of Switzerland. To increase the credibility of our study, we model the irregular radio channel behavior by the probabilistic Shadowing signal propagation model.

VANETs pose unique challenges to a routing protocol. The use of the random waypoint model to control node mobility may lead to an optimistic assessment. AODV and GPSR exhibit serious performance problems in the VANET scenarios investigated here. The packet delivery ratio of both protocols stays very low and varies from 5-20 % depending on the simulated scenario.

One way to address the performance problems is to use a preferred group broadcasting strategy (PGB) to split and merge hops. This strategy significantly reduces broadcasting load and noticeably improves performance of AODV in VANETs. PGB can be integrated to any broadcasting-based routing protocol and can also be used for effective data dissemination. In the scenarios studied here, the packet delivery ratio of AODV increases by up to 8 times when the PGB strategy is used.

The Advanced Greedy Forwarding (AGF) technique improves the effectiveness of greedy forwarding and, as a result, the performance of GPSR in VANETs. Packet delivery ratio of GPSR enhanced with AGF is up to 10 times better than of the standard version of this protocol. Although our investigation focusses on GPSR, AGF can also be used for other routing protocols that include greedy forwarding of data packets.

VANETs provide an interesting and challenging environment for ad hoc networks. To assist other researchers, we make our collection of traces available and look forward to additional studies and new ideas for protocol refinements as VANETs move from research to real-life implementations.

## Acknowledgments

We thank Kai Nagel, Bryan Raney, and Hinnerk Spindler for providing us with the car traffic simulator output and for their support of our work.

## 9. REFERENCES

- [1] LAN MAN standards committee of the IEEE computer society. Wireless lan medium access control (MAC) and physical layer (PHY) specifications. IEEE standard 802.11. Technical report, 1999.
- [2] Joe Bardwell. Converting signal strength percentage to dBm values. WildPackets' White Paper, November 2002.
- [3] C. Bettstetter. Smooth is better than sharp: A random mobility model for simulation of wireless networks. In *MSWiM'01*, pages 19–27, Rome, Italy, 2000.
- [4] C. Bettstetter. Mobility modeling in wireless networks: Categorization, smooth movement, and border effects. In *ACM SIGMOBILE'01*, volume 5, pages 55–67, 2001.
- [5] V. Bhandari and N. Vaidya. On reliable broadcast in a radio network. In *PODC'05*, pages 138–147, New York, NY, USA, 2005.
- [6] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *ACM DIAL M'99*, pages 48–55, August 1999.
- [7] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MOBICOM'98*, pages 85–97, 1998.
- [8] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *MOBICOM'02*, pages 483–502, August 2002.
- [9] C. Chaudet, D. Dhoutaut, and I. Lassou. Experiments of some performance issues with IEEE 802.11b in ad hoc networks. In *WONS'05*, pages 158–163, 2005.
- [10] Z. Chen, H.T. Kung, and D. Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *MobiHoc'01*, pages 247–250, 2001.
- [11] T. Clausen, P. Jacquet, A. Laouiti, and P. Muhlethaler et al. Optimized link state routing protocol. In *IEEE INMIC'01*, pages 151–162, 2001.
- [12] S. Das, Ch. Perkins, and E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *INFOCOM'00*, pages 3–12, 2000.
- [13] N. Eude, B. Ducourthial, and M. Shawky. Enhancing ns-2 simulator for high mobility ad hoc networks in car-to-car communication context. In *MWCN'05*, September 2005.
- [14] K. Fall. Ns notes and documentation. *The VINT Project*, 2000.
- [15] J. Jetcheva, Y. Hu, S. PalChaudhuri, A. Saha, and D. Johnson. Design and evaluation of a metropolitan area multiter wireless ad hoc network architecture. In *WMCSA'03*, pages 32–43, Monterey, CA, October 2003.
- [16] D. Johnson. Validation of wireless and mobile network models and simulation. In *DARPA/NIST Network Simulation Validation Workshop, Fairfax, Virginia, USA*, May 1999.
- [17] D. Johnson, D. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multihop wireless ad hoc networks. *Ad Hoc Networking*, pages 139–172, 2001.
- [18] B. Karp and H.T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MOBICOM'00*, pages 243–254, Boston, Massachusetts, United States, 2000.
- [19] Y. Kim, J. Lee, and A. Helmy. Impact of location inconsistencies on geographic routing in wireless networks. In *MSWiM'03*, pages 124–127, 2003.
- [20] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM'04*, pages 78–82, 2004.
- [21] J. Li, C. Blake, D. De Couto, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *MOBICOM'01*, pages 61–69, 2001.
- [22] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *MSWiM'00*, pages 61–68, Boston, Massachusetts, United States, 2000.
- [23] C. Lochert et al. A routing strategy for vehicular ad hoc networks in city environments. In *IVS'03*, pages 156–161, June 2003.
- [24] J. Luo and J.-P. Hubaux. A survey of inter-vehicle communication. Technical Report IC/2004/24, EPFL, Lausanne, Switzerland, 2004.
- [25] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. De Couto. Carnet: a scalable ad hoc wireless network system. In *EW 9*, pages 61–65, 2000.
- [26] V. Naumov and T. Gross. Scalability of routing methods in ad hoc networks. *Performance Evaluation*, 62(1–4):193–209, October 2005.
- [27] S. Y. Ni et al. The broadcast storm problem in a mobile ad hoc network. In *MOBICOM'99*, pages 151–162, 1999.
- [28] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF). <http://www.ietf.org/rfc/rfc3684.txt>, Internet Draft, IETF, February 2004. (Work in progress).
- [29] G. Pei, M. Gerla, X. Hong, and C. Chiang. A wireless hierarchical routing protocol with group mobility. In *IEEE WCNC'99*, pages 1536–1540, New Orleans, LA, September 1999.
- [30] W. Peng and X. Lu. Efficient broadcast in mobile ad hoc networks using connected dominating set. *Journal of Software*, 1999.
- [31] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *MOBIHOC'00*, 2000.
- [32] W. Peng and X. Lu. AHB: An efficient broadcast protocol for mobile ad hoc networks. *Journal of Science and Technology*, 2002.
- [33] Ch. Perkins, E. Royer, and S. Das. Ad hoc on demand distance vector (AODV) routing. draft-ietf-manet-aodv-10.txt, Internet Draft, IETF, March 2002. (Work in progress).
- [34] B. Raney, A. Voellmy, M. Vrtic, K. Axhausen, and K. Nagel. An agent-based microsimulation model of Swiss travel. *Networks and Spatial Economics*, 3:23–41, 2003.
- [35] B. Raney, A. Voellmy, M. Vrtic, and K. Nagel. Towards a microscopic traffic simulation of all of Switzerland. In *ICCS'02*, 2002.
- [36] S. Ray, J. Carruthers, and D. Starobinski. RTS/CTS-induced congestion in ad-hoc wireless lans. In *IEEE WCNC'03*, pages 1516–1521, March 2003.
- [37] A. Saha and D. Johnson. Modeling mobility for vehicular ad-hoc networks. In *VANET'04*, pages 91–92, 2004.
- [38] H. Sawant, J. Tan, and Q. Yang. Study of an inter-vehicle communication protocol for vehicle-infrastructure integration (VII). In *Transportation Research Board 84th Annual Meeting*, Washington, DC, 2005.
- [39] J. Singh et al. Wireless lan performance under varied stress conditions in vehicular traffic scenario. In *VTC'02*, volume 2, pages 743–747. IEEE, 2002.
- [40] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. Technical Report 248, CAIP, Rutgers University, September 2000.
- [41] M. Takai, J. Martin, and R. Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *MobiHoc'01*, pages 87–94, 2001.
- [42] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *MobiHoc'02*, pages 194–205, 2002.
- [43] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM'03*, pages 1312–1321, San Fransisco, California, USA, April 2003.
- [44] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys'04*, pages 125–138, 2004.
- [45] M. Zorzi and R. Rao. Geographic random forwarding (GERAF) for ad hoc and sensor networks: multihop performance. In *IEEE Trans. Mobile Computing*, 2(4), October 2003.