The DYMO Routing Protocol in VANET Scenarios

Christoph Sommer and Falko Dressler
Computer Networks and Communication Systems
Dept. of Computer Sciences, University of Erlangen-Nuremberg, Germany
{christoph.sommer,dressler}@informatik.uni-erlangen.de

***Abstract*—Coupling Vehicular Ad Hoc Networks (VANETs) with wired networks such as the Internet via access points creates a difficult mix of highly mobile nodes and a static infrastructure. In order to evaluate the performance of typical ad hoc routing protocols—in particular, we used Dynamic MANET On Demand (DYMO)—in such VANET scenarios, we combined microsimulation of road traffic and event-driven network simulation. Thus, we were able to analyze protocols of the Internet protocol suite in VANET scenarios with highly accurate mobility models. Varying parameters of DYMO for a multitude of traffic and communication scenarios helped point out approaches for improving the overall performance and revealed problems with the deployment. It could be shown that in realistic scenarios, even for medium densities of active nodes and low network load, overload behavior leads to a drastic decrease of the perceived network quality. Cross-layer optimization of transport and routing protocols therefore seems highly advisable.**

I. Introduction

Recent research in the area of Vehicular Ad Hoc Networks (VANETs) was primarily focused on the development and the evaluation of highly specialized protocols, e.g. for the exchange of position information or hazard warnings between cars [1]. Significantly less work dealt with evaluating the use of existing Internet protocols, along with standard hard- and software, to create and maintain VANETs and couple these networks with the Internet.

The Mobile Ad Hoc Network (MANET) working group of the Internet Engineering Task Force (IETF) develops standards for routing in dynamic networks of both mobile and static nodes. One protocol currently in the working group's focus is Dynamic MANET On Demand (DYMO) [2]. It was conceived as successor to the popular Ad Hoc on Demand Distance Vector (AODV) routing protocol [3], [4]. Its use in the context of VANETs has already been extensively investigated [5]. The DYMO protocol draft expressly provides for the coupling of a MANET with the Internet, which makes an evaluation of communication connections between mobile nodes and static infrastructure especially attractive.

A car taking part in a MANET scenario could already establish such connections in reach of one of an ever growing number of public hotspots while driving in the city, and a deployment of access points along highways in the near future seems feasible. Apparently, this coupling of MANET and Internet is especially attractive for road users if it allows the utilization of virtually all existing resources of the Internet without relying on expensive dedicated channels provided by a cellular network.

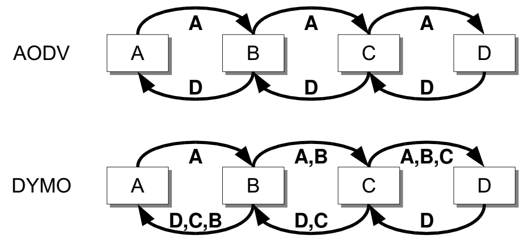

Fig. 1. Routing information dissemination in AODV and DYMO

In this work, the feasibility, the performance, and the limits of ad hoc communication using DYMO were evaluated and potentials for optimizing the deployed transport and routing protocols were investigated. Special care was taken to provide realistic scenarios of both road traffic and network usage. This was accomplished by simulating a variety of such scenarios with the help of two coupled simulation tools [6]. A microsimulation environment for road traffic supplied vehicle movement information, which was then fed into an event-driven network simulation that configured and managed a MANET model based on this mobility data. The protocols of the transport, network, data link, and physical layers were provided by well-tested implementations for the network simulation tool, while MANET routing was performed by our own implementation of DYMO.

II. Dynamic MANET On Demand (DYMO)

DYMO [2] is a new reactive (on demand) routing protocol, which is currently developed in the scope of the IETF's MANET working group. DYMO builds upon experience with previous approaches to reactive routing, especially with the routing protocol AODV. It aims at a somewhat simpler design, helping to reduce the system requirements of participating nodes, and simplifying the protocol implementation.

DYMO retains proven mechanisms of previously explored routing protocols like the use of sequence numbers to enforce loop freedom. At the same time, DYMO provides enhanced features, such as covering possible MANET–Internet gateway scenarios and implementing path accumulation as depicted in Figure 1.

Besides route information about a requested target, a node will also receive information about all intermediate nodes of a newly discovered path. Therein lies a major difference between DYMO and AODV, the latter of which only generates route table entries for the destination node and the next hop, while DYMO stores routes for each intermediate hop. This is illustrated in Figure 1. When using AODV, node A knows only the routes to nodes B and D after its route request is satisfied. In DYMO, the node additionally learned a route to node C.

To efficiently deal with highly dynamic scenarios, links on known routes may be actively monitored, e.g. by using the MANET Neighborhood Discovery Protocol [7] or by examining feedback obtained from the data link layer. Detected link failures are made known to the MANET by sending a route error message (RERR) to all nodes in range, informing them of all routes that now became unavailable. Should this RERR in turn invalidate any routes known to these nodes, they will again inform all their neighbors by multicasting a RERR containing the routes concerned, thus effectively flooding information about a link breakage through the MANET.

DYMO was also designed with possible future enhancements in mind. It uses a generic MANET packet and message format [8] and offers ways of dealing with unsupported elements in a sensible way.

III. Simulation Tools

For the selection of a suitable traffic simulation tool, two aspects had to be weighed against each other. Clearly, the underlying traffic model was to be as simple and comprehensible as possible, so that reproducible results could be obtained. On the other hand, the simulation model needed to be complex enough to produce realistic patterns, which—as has been shown in related work—greatly influence the quality of results obtained from overlaid network simulations [9].

Microsimulation of road traffic was performed by an adaptation of *TrafficApplet*[1], an open source traffic microsimulation tool that provides an accurate model of microscopic driver behavior, as opposed to the still common simplistic [10] or proprietary behavior models [11]. It implements the microsimulation models IDM [12] and MOBIL [13] to calculate longitudinal and lateral movement, respectively. The behavior of simulated vehicles can be configured with simple parameters like "desired velocity" or "comfortable acceleration", which were used to model two different types of road users. Nodes of type *Truck* traveled at a maximum speed of $22.2\,\mathrm{m/s}$ (approx. $80\,\mathrm{km/h}$, $50\,\mathrm{mph}$) and made up $20\,\%$ of the vehicles simulated. The remaining $80\,\%$ of vehicles were of type *Car* and traveled at speeds of up to $33.0\,\mathrm{m/s}$ (approx. $120\,\mathrm{km/h}$, $75\,\mathrm{mph}$). All simulations were performed at a density of 4.2 vehicles per kilometer and lane, representing nightly traffic, as well as at a density of 28.0 vehicles per kilometer and lane, which modeled rush-hour traffic [14], [15].

Sample speed traces recorded in both scenarios are shown in Figure 2. Obviously, using a smaller number of simulated vehicles allowed the cars to move nearly unimpaired by trucks or other cars and to travel at or near top speed. The scenario thus maximized speed differences between nodes, so links between cars of different lanes, between cars and trucks, as well as between vehicles and roadside infrastructure were highly unstable. A larger number of simulated vehicles forced cars and trucks into a stop-and-go motion, reducing the cars' top speed to that of trucks. This stabilized links between vehicles and reduced speed differences between vehicles and roadside infrastructure, but caused large oscillations of local node densities.

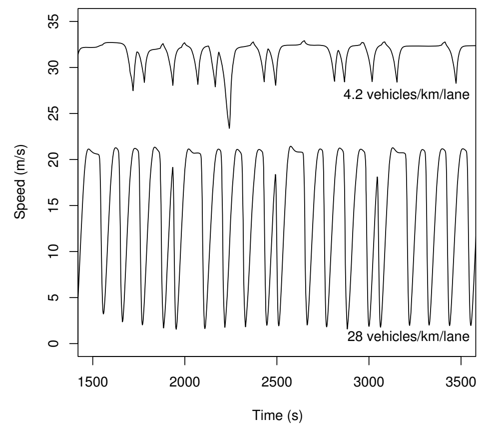

Fig. 2. Speed samples of simulated cars at different traffic densities

Realistic communication patterns of MANET nodes were modeled using OMNeT++ 3.2p1 [16], a simulation environment free for non-commercial use, and its *INET Framework* 20060330, a set of simulation modules released under the GPL. OMNeT++ runs discrete, event-based simulations of communicating nodes on a wide variety of platforms and is getting increasingly popular in the communications community. Scenarios in OMNeT++ are represented by a hierarchy of reusable modules written in C++. Their relationships and communication links are stored as *Network Description* (NED) files and can be modeled graphically. Simulations are either run interactively in a graphical environment or executed as command-line applications. The *INET Framework* provides a set of OMNeT++ modules that represent various layers of the Internet protocol suite, e.g. the TCP, UDP, IPv4 and ARP protocols. It also provides modules that allow the modeling of spatial relations of mobile nodes and IEEE 802.11 transmissions between them.

IV. Simulation Model

The DYMO routing protocol was implemented as an application-layer module of the *INET Framework* module set. Following the specification [2], it employs UDP to communicate with other instances of DYMO. Additionally, it uses two helper modules to support DYMO operation on the network layer. The complete protocol stack is shown in Figure 3. The first helper module is able to queue outbound packets before routing in the network layer occurs, so that a route can be set up by DYMO. The queue can then be signaled to release buffered packets for a given destination—either in order to have them routed to the first hop or to have them discarded by the network layer because no route could be found. The second helper module is installed as a hooking function in

[1]http://www.vwi.tu-dresden.de/~treiber/MicroApplet

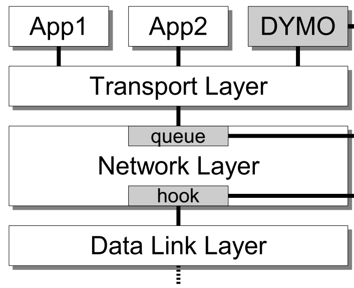

Fig. 3. DYMO and support modules in the protocol stack

the inbound packet path. It notifies DYMO of the arrival of packets. This way, routing table entries can be refreshed and route errors can be sent, respectively. DYMO and its helper modules are assembled together with various components of the *INET Framework* to form simulated MANET nodes.

Mobile nodes are represented by modules of type *Car*, which perform DYMO along with TCP or UDP applications that generate application specific traffic. Communication with other nodes takes place via an IEEE 802.11 module. The road-side infrastructure is provided by modules of type *AccessPoint*, which execute DYMO only to route between the wireless and the wired network, i.e. the Internet. Internet connectivity is modeled by a node of type *CSTMGateway* that is also running DYMO. It sends back delayed response messages to requests via TCP or UDP, i.e. it simulates the application servers that are used by the clients (the *Cars*).

For all communications, the complete network stack, including ARP, was used and wireless modules were configured to closely resemble IEEE 802.11b network cards transmitting at $11\,\mathrm{Mbit/s}$ with RTS/CTS disabled. The TCP protocol implementation follows the TCP Reno specification. Thus, results can be readily compared with existing Linux implementations of DYMO, e.g. *NIST DYMO* or *DYMOUM*. For the simulation of radio wave propagation, a plain free-space model was employed and the transmission ranges of all nodes adjusted to a fixed value of $180\,\mathrm{m}$, a trade-off between varying real-world measurements described in related work [17], [18]. All simulation parameters used to parameterize the modules of the *INET Framework* are summarized in Table I.

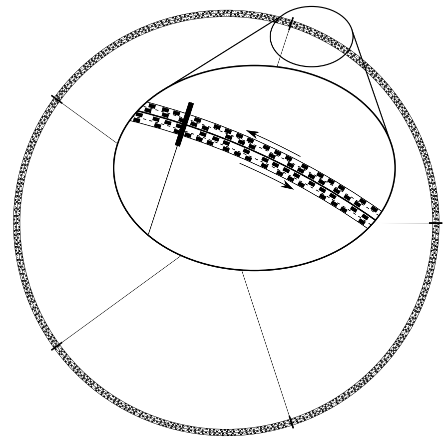

Fig. 4. Simulated MANET scenario

TABLE I
INET FRAMEWORK MODULE PARAMETERS

Parameter	**Value**
TCP.mss	$1024\,\mathrm{Byte}$
TCP.advertisedWindow	$14\,336\,\mathrm{Byte}$
TCP.tcpAlgorithmClass	TCPReno
ARP.retryTimeout	$1\,\mathrm{s}$
ARP.retryCount	3
ARP.cacheTimeout	$100\,\mathrm{s}$
mac.address	auto
mac.bitrate	$11\,\mathrm{Mbit/s}$
mac.broadcastBackoff	$31\,\mathrm{slots}$
mac.maxQueueSize	$14\,\mathrm{Pckts}$
mac.rtsCts	false
decider.bitrate	$11\,\mathrm{Mbit/s}$
decider.snirThreshold	$4\,\mathrm{dB}$
snrEval.bitrate	$11\,\mathrm{Mbit/s}$
snrEval.headerLength	$192\,\mathrm{bit}$
snrEval.snrThresholdLevel	$3\,\mathrm{dB}$
snrEval.thermalNoise	$-110\,\mathrm{dB}$
snrEval.sensitivity	$-85\,\mathrm{dB}$
snrEval.pathLossAlpha	1.9
snrEval.carrierFrequency	$2.4\,\mathrm{GHz}$
snrEval.transmitterPower	$2\,\mathrm{mW}$
channelcontrol.carrierFrequency	$2.4\,\mathrm{GHz}$
channelcontrol.pMax	$2\,\mathrm{mW}$
channelcontrol.sat	$-80\,\mathrm{dBm}$
channelcontrol.alpha	1.9

In order to ensure realistic application layer traffic, the following three different communication scenarios were modeled:

1) Vehicles polled traffic information from an Internet host. At 5 minute intervals, starting at a random point in time no more than 5 minutes from the start of a simulation, a vehicle tried to send a $256\,\mathrm{Byte}$ UDP packet to the gateway, which, upon reception of the packet, answered with a $1024\,\mathrm{Byte}$ response packet.
2) Mobile nodes checked a POP3 mailbox (using TCP) for new messages, configured with a maximum segment size of $1024\,\mathrm{Byte}$ and an advertised window size of $14\,336\,\mathrm{Byte}$, to send eight $16\,\mathrm{Byte}$ commands, each triggering a $32\,\mathrm{Byte}$ response. As in the first case, the mailbox check was repeated 5 minutes after sending the first command and the maximum session length limited accordingly.
3) Vehicles requested RSS feeds from a web server (also using TCP). This was represented by changing the second case's parameters, so that nodes would only send a single, $256\,\mathrm{Byte}$ request message and receive a single, $65\,536\,\mathrm{Byte}$ response message, with fragmentation and reassembly taking place in lower layers.

The modeled nodes were then further combined to create the MANET scenario shown in Figure 4, a simulated highway with two lanes in each direction forming a $10\,\mathrm{km}$ long closed ring with evenly spaced access points at distances of $2\,\mathrm{km}$, $5\,\mathrm{km}$ or $10\,\mathrm{km}$, depending on the scenario.

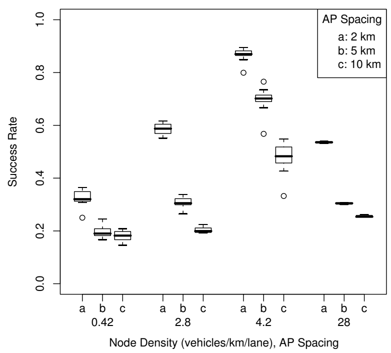

Fig. 5. Success rate at different distances and node densities; Scenario: UDP

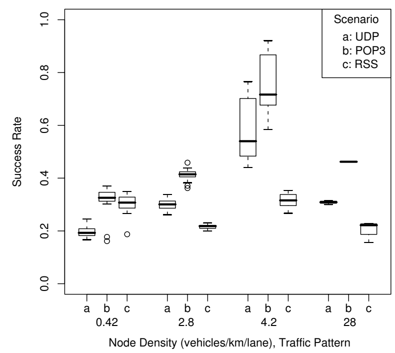

Fig. 6. Success rate in different scenarios at different node densities; AP distance: $5\,\mathrm{km}$

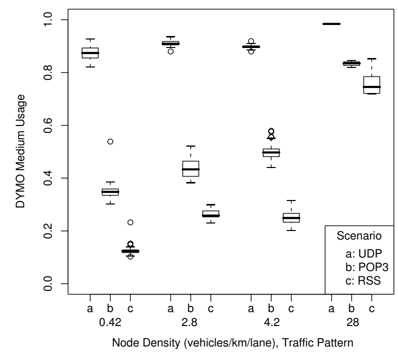

Fig. 7. DYMO overhead in different scenarios at different node densities; AP distance: $5\,\mathrm{km}$

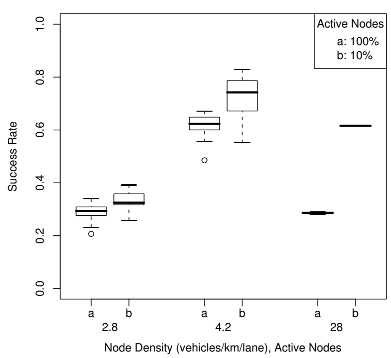

Fig. 8. Success rate improvement with passive nodes; AP distance: $5\,\mathrm{km}$, Scenario: UDP

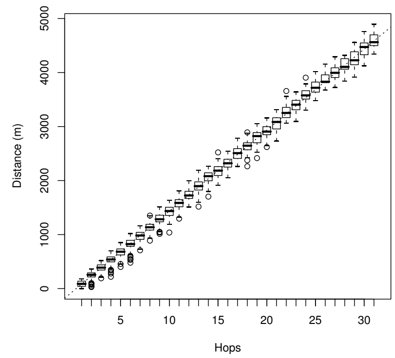

Fig. 9. Hop counts of established bidirectional links vs. distance bridged

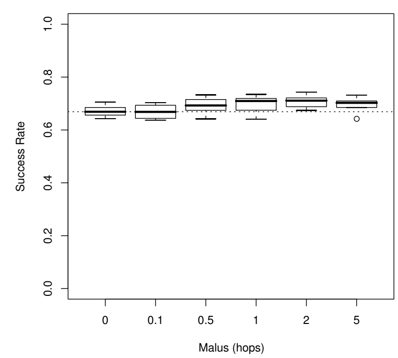

Fig. 10. Improvement of success rate when applying a hop count malus

V. Performance Analysis

Perceived performance of the VANET was estimated by recording the overall success rate, i.e. the probability of successful reception of a UDP information packet, the last POP3 response, or a complete RSS feed, at the requesting vehicle depending on the traffic pattern in use. Performance was measured at four different node densities of 0.42, 2.8, 4.2, and 28 vehicles per kilometer and lane, corresponding to the two chosen traffic densities and fractions of $10\,\%$ and $100\,\%$ DYMO-equipped vehicles, respectively. Three different minimum route lifetimes of $1\,\mathrm{s}$, $3\,\mathrm{s}$, and $10\,\mathrm{s}$ were tried for each of the simulated scenarios. A value of $1\,\mathrm{s}$ proved to balance the amount of route request and route error messages in the network best. Also, setting the DYMO *Network Size* parameter to 50 hops instead of to the default 10 hops proved to be beneficial when access points were spaced more than $2\,\mathrm{km}$ apart and node densities did not exceed 28 vehicles per kilometer and lane.

All results are shown as boxplots. For each data set, a box is drawn from the first quartile to the third quartile, and the median is marked with a thick line. Additional whiskers extend from the edges of the box towards the minimum and maximum of the data set, but no further than 1.5 times the interquartile range. Data points outside the range of box and whiskers are considered outliers and drawn separately.

Figure 5 shows the overall probability of a simulated UDP session being successfully completed for different node densities and access point distances. As can be seen, even low node densities of 4.2 nodes per kilometer and lane, as well as sparse access point deployment of one node per $5\,\mathrm{km}$ highway, sufficed to permit the exchange of UDP packets in approx. $50\,\%$ of all tries. Results for other communication scenarios are shown in Figure 6, which plots the overall probability of a session being successfully completed for a fixed access point distance of $5\,\mathrm{km}$. Due to the retry mechanisms offered by the TCP protocol, POP3 sessions always had a significantly higher chance of being completed than plain UDP sessions, even though completion of a POP3 session required the exchange of more packets. Also visible is a rapidly decreasing probability of sessions being successfully completed when node densities increased to above 4.2 nodes per kilometer and lane or when larger messages were to be delivered. While at 0.42 nodes per kilometer and lane, the probability of RSS sessions completing was almost at par with that of POP3 sessions, at 2.8 nodes per kilometer and lane already only half as many RSS sessions finished successfully—approx. $20\,\%$ compared to approx. $40\,\%$ POP3 sessions. Figures 7 and 8 illustrate a reason for this decrease. With a rising number of communicating nodes, network traffic on the shared medium was increasingly dedicated to DYMO packets until, at 28.0

vehicles per kilometer and lane, the MANET was almost exclusively busy exchanging routing messages. Reducing the number of actively participating nodes to $10\,\%$ significantly improved figures—even for node densities as low as 2.8 vehicles per kilometer and lane.

To estimate the impact of overload effects on the quality of routes established by DYMO in the VANET, the relation between the length of a route in number of hops and the total distance bridged between vehicle and access point was examined. As can be seen in Figure 9, the bridged distance is closely related to the number of hops and it is increasing linearly by approx. $150\,\mathrm{m}$ per hop—not much less than the nodes' communication range of $180\,\mathrm{m}$.

In order to reduce the stress imposed on the network due to constant link breakages and subsequent flooding of route error and new route request messages, a promising mechanism was implemented for estimating the potential route stability by taking movement directions into account. When comparing two routes to find the shortest path, DYMO now added a malus of 0.1–$5.0\,\mathrm{hops}$ for each time a packet was sent to a vehicle traveling in the opposite direction. Information about a vehicle's relative travel direction was assumed to be estimable by the physical layer. Figure 10 shows the results of this adaptation. Success rates of sessions could indeed be significantly improved by adding such a malus, but the adaptation failed to produce the huge effects observed by other groups [5] when completely ignoring oncoming traffic for route selection.

VI. Conclusion and Future Work

Evaluation of the feasibility and the expected quality of VANETs operated with the routing protocol DYMO showed that for small amounts of payload data to be transported, ad hoc networks of vehicles and static highway infrastructure can be successfully setup, maintained, and used with well-known protocols from the Internet protocol suite alone. Even low node densities and sparse access point deployment sufficed to support routine polling of information via an Internet gateway, e.g. the checking of a POP3 mailbox. Larger amounts of network traffic to be transported over the ad hoc network, however, induced overload effects that noticeably destabilized the VANET. Particularly at higher node densities, which commonly occurred in micro-jams, the routing and transport protocol behavior led to a drastic increase in network load. When the network became congested and new connections could not be established, simple retry mechanisms only furthered congestion.

Simulation results therefore seem to encourage an adaptation of the protocols in use, so problems perceived by lower layers are reacted to in a sensible way and application requirements are taken into account when the network becomes overloaded. Cross-layer optimization might keep nodes from using potentially unstable routes for low-priority messages in favor of a reduction of network load. Also, the results of the conducted simulations make a simple flooding of messages through the VANET and the selection of routes without taking node position and mobility into account, as proposed in the current draft of DYMO, appear wasteful. An experimental modification of DYMO, which penalized routes across the lanes when assessing the quality of potential routes, proved beneficial, but failed to produce the predicted increase in overall network quality that was claimed in related work.

References

[1] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS - a self-organizing traffic information system," in *Proceedings of the 57th IEEE Vehicular Technology Conference (VTC 03 Spring)*, 2003.

[2] I. Chakeres and C. Perkins, "Dynamic MANET On-Demand (DYMO) Routing," Internet-Draft, draft-ietf-manet-dymo-06.txt, October 2006. [Online]. Available: http://tools.ietf.org/wg/manet/draft-ietf-manet-dymo/draft-ietf-manet-dymo-06.txt

[3] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, July 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[4] C. Perkins and E. Royer, "Ad hoc On-Demand Distance Vector Routing," in *2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90–100.

[5] R. Baumann, "Vehicular ad hoc networks (VANET) - engineering and simulation of mobile ad hoc routing protocols for VANET on highways and in cities," Master's thesis, ETH Zürich, 2004.

[6] C. Sommer, I. Dietrich, and F. Dressler, "Realistic Simulation of Network Protocols in VANET Scenarios," in *26th Annual IEEE Conference on Computer Communications (IEEE INFOCOM 2007): Mobile Networking for Vehicular Environments (MOVE 2007), Poster Session*. Anchorage, Alaska, USA: IEEE, May 2007.

[7] T. Clausen, C. Dearlove, J. Dean, the OLSRv2 Design Team, and the MANET Working Group, "MANET Neighborhood Discovery Protocol (NHDP)," Internet-Draft, draft-ietf-manet-nhdp-00.txt, June 2006. [Online]. Available: http://tools.ietf.org/pdf/draft-ietf-manet-nhdp-00.pdf

[8] T. H. Clausen, C. M. Dearlove, J. W. Dean, and C. Adjih, "Generalized MANET Packet/Message Format," Internet-Draft, draft-ietf-manet-packetbb-02.txt, July 2006. [Online]. Available: http://tools.ietf.org/pdf/draft-ietf-manet-packetbb-02.pdf

[9] A. Mahajan, "Urban mobility models for vehicular ad hoc networks," Master's thesis, Department of Computer Science, Florida State University, 2006.

[10] A. K. Saha and D. B. Johnson, "Modeling mobility for vehicular ad-hoc networks," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, 2004.

[11] C. Lochert, A. Barthels, A. Cervantes, M. Mauve, and M. Caliskan, "Multiple simulator interlinking environment for IVC," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, 2005.

[12] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, p. 1805, 2000.

[13] M. Treiber and D. Helbing, "Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell," in *ASIM 2002, Tagungsband 16. Symposium Simulationstechnik*, 2002.

[14] O. Kaumann, K. Froese, R. Chrobok, J. Wahle, L. Neubert, and M. Schreckenberg, "On-line simulation of the freeway network of north rhine-westphalia," in *Traffic and Granular Flow '99*, D. Helbing, H. Herrmann, M. Schreckenberg, and D. Wolf, Eds. Heidelberg: Springer, 2000, pp. 351–356.

[15] B. Tilch and D. Helbing, "Evaluation of single vehicle data in dependence of the vehicle-type, lane, and site," in *Traffic and Granular Flow '99*, D. Helbing, H. Herrmann, M. Schreckenberg, and D. Wolf, Eds. Heidelberg: Springer, 2000.

[16] A. Varga, "The OMNeT++ discrete event simulation system," in *Proceedings of the European Simulation Multiconference (ESM2001)*, 2001.

[17] F. Hui and P. Mohapatra, "Experimental characterization of multi-hop communications in vehicular ad hoc network," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, 2005.

[18] H. Wu, M. Palekar, R. Fujimoto, J. Lee, J. Ko, R. Guensler, and M. Hunter, "Vehicular networks in urban transportation systems," in *Proceedings of the 2005 national conference on Digital government research*, 2005.