AutoCast: An Adaptive Data Dissemination Protocol for Traffic Information Systems

Axel Wegener,
Horst Hellbrück
and Stefan Fischer
Institute of Telematics, University of Lübeck
Email: {wegener, hellbrueck, fischer}@itm.uni-luebeck.de

Christiane Schmidt
and Sándor Fekete
Algorithms Group, Department of Computer Science,
Braunschweig University of Technology
Email: {c.schmidt, s.fekete}@tu-bs.de

Abstract*—Protocols and applications that rely on unicast and multicast communication are well accepted and still gain more and more popularity. However, these communication paradigms are not optimal for a class of wireless applications where communication partners neither establish specific relationships nor need roles like client and server between each other before data exchange. Applications we have in mind deal with up to several thousands of peers as autonomous wireless network nodes. Nodes communicate events like traffic accidents in a local region or information of common interest to a larger group of network nodes. Intermediate nodes forward or rather "gossip" information like in a social communication model, comparable to the news of the big fire of Rome in neronian times travelling through Europe and finally reaching villages in rural areas. The challenge of such a concept is to find efficient local rules, which balance communication with respect to bandwidth usage, latency of data, and data delivery ratio. We introduce the promising application AutoNomos – a decentralized traffic information system – which is well suited for the evaluation of such a data dissemination protocol. Next, we present our new approach called *AutoCast* that is well optimized and self-adaptable towards various dynamic topologies. We compare *AutoCast* against the theoretical optimum and existing data dissemination protocols. Finally, simulations will demonstrate the efficiency of the approach.

***Index Terms*—mobile ad-hoc networks, broadcast, data dissemination, Hovering Data Cloud, AutoNomos, AutoCast**

I. INTRODUCTION

Today's applications for distributed systems rely on unicast or multicast communication, where communication partners are identified by layer-3 addresses. Prominent and well known examples are Client-Server-Applications like webbrowsing. When mobile ad-hoc networks (MANETs) first came up, research concentrated on efficient routing to keep up the existing communication paradigm. By increasing the number of nodes and introducing dynamics in the network due to node mobility or switching nodes on and off in a large multi-hop network, routing based on addresses became a hard challenge. In the meantime systems grew larger consisting of thousands of cheap battery powered devices that organize themselves and open a new research field called sensor networks. Limitations of bandwidth, computing power and storage in sensor networks drive a new communication paradigm. While these systems need to be designed for frequent node failures, redundancy was introduced in the network decreasing the importance of single nodes and addresses. Also the stringent separation of the lower communication layers is open to discussion as it introduces computational and networking overhead especially when identification of nodes looses importance.

Fig. 1. A Hovering Data Cloud at a traffic jam's end warns incoming vehicles

The application that drives our work on data dissemination is a decentralized traffic information system called AutoNomos. Here cars cooperate to recognize various traffic situations like a traffic jam and disseminate data like position and length of a traffic jam to other nodes in the network as outlined in Figure 1. The dissemination as well as the generation of the traffic information is based solely on car to car communication without additional infrastructure. First results were published in [1], [2]. The AutoNomos system relies on single cars as nodes but needs to identify information about road conditions rather than identifying the nodes where this data was generated from and forwarded by.

In AutoNomos cars establish Hovering Data Clouds (HDCs) in reaction to certain road conditions, e.g. the end of the traffic jam depicted in Figure 1 as shaded cloud. HDCs are not bound to cars that have generated them but observe the road condition over time and hop from car to car when necessary. HDCs disseminate data units to inform oncoming traffic. The application controls the dissemination of these data units by simple parameters like spatial or temporal validity.

We foresee many applications with the requirement to disseminate news or other information of general interest through a multi-hop network. The advantage of our approach is that data is available automatically without the need for explicit data requests. This is comparable to a push mechanism, like video text, where information is pushed to the user. However, our approach goes beyond that simple broadcast by disseminating data adapted to the network characteristic.

The rest of the paper is organized as follows: Next section discusses related work and existing solutions for data dissemination mechanisms and illustrates the need for a more efficient solution. In Section III we introduce *AutoCast* in detail and

1-4244-0264-6/07/$25.00 ©2007 IEEE

compare it in Section IV to other basic approaches. We present simulation results in Section V and address future challenges after a short summary in Section VI.

II. RELATED WORK

Unicast is the most popular way of communication and can be seen as the standard for Client-Server based communication in the internet. Nowadays, using wireless multi-hop networks many applications have different requirements and various protocols and communication approaches have been developed in the past to match new applications' demands.

In sensor networks the communication paradigm shifts away from the node-centric way where data is delivered between nodes identified by addresses to a data-centric way of communication. The basic idea of the data-centric communication is that nodes subscribe to a type of data identified by a unique name and receive data associated with this name as shown in [3]–[5]. Since data is often sent to one or only few sinks in sensor networks, approaches like [6] deal with moving sinks while the rest of the network stays immobile. In any case, routes between the originator of data and its subscribers are needed to transport data through the multi-hop network. All these approaches fail for the fast changing network topologies in vehicular ad-hoc networks (VANETs). Projects like [7] address the new challenges of VANETs but often use data dissemination approaches limited to emergency data. Like other examples as [8], [9] emergency notifications are assumed to occur only rarely, statically and will be short-lived. By contrast, our approach is able to handle numerous data units in parallel, even when they are disseminated at the same time to arbitrary directions and created at arbitrary positions in the network.

Approaches that concentrate on disseminating traffic conditions, like [10], [11], focus on the adaptation of broadcast interval, e.g. according to the vehicle's speed. The proposed techniques are closely bound to specific applications with fixed sized road segments and distinguish only between regular communication and emergency data. In particular, they are not designed for dynamic appearance and heterogeneity of data units with individual life time and suddenly occurring long-lived data that describes e.g. a traffic jam.

A further optimization to save bandwidth while ensuring that every node gets as much data as possible is described in [12]. Each data unit is represented by a hash value. In a unicast approach, new data is sent after a three way handshake comprising advertisement, request, and delivery of data units.

Each approach has its individual optimum working condition and is mainly created on the fly to solve a particular problem. In the next chapter we will derive step by step a generic optimized protocol for data dissemination inspired by our AutoNomos application.

III. PROTOCOL

The proposed protocol *AutoCast* is targeted at applications that need to communicate in a many-to-many manner, without a need to set up an association or connections between network nodes. In a traffic information system, each car contributes to the knowledge of road conditions that may be important for nearby cars.

In general, dissemination of data in mobile ad-hoc networks can be achieved in two ways. This may either be by the movement of network nodes (see [13]) or by multi-hop ad-hoc communication between nodes. Because communication is much faster than carrying data piggybacked on moving nodes, it is preferred in most scenarios. However, node movement can support communication in partitioned networks, e.g., by using opposite-lane traffic for bridging gaps between cars.

Because ad-hoc networks already use a broadcast medium, unicast communication is an artificial constraint. Sending broadcast messages is more efficient than unicast messages, gaining even more with increasing network density. Even if a particular data unit is not useful for a node, it can assist in further dissemination of the data unit.

The most intuitive technique is pure *flooding*, where each node receiving a data unit rebroadcasts it exactly once and as soon as possible. Flooding can be fast for fully connected networks. However, the single-rebroadcast property causes network partitions to stop data forever. As a consequence, flooding cannot bridge communication gaps; in addition, it jams the wireless channel in dense networks with a broadcast storm [14].

A well-established method for disseminating data slowly and more reliably, even when network partitions occur frequently, is a periodic rebroadcast of received data with a short delay. As described in [15] the protocol *MILE* is designed for the exchange of location information. We enhance *MILE* to work with generic data units instead of location information. By randomly choosing several data units from all locally known data units when broadcasting, data dissemination reaches an acceptable speed. The main drawback is that this technique does not scale with increasing network density and increasing number of data units in the network.

It can easily be seen from the detailed results in Section V that both basic approaches have advantages and disadvantages.

In order to measure the best possible performance, we introduce a *theoretical* protocol as benchmark. This protocol assumes unlimited transmission rate, propagation speed of light, and a perfect intuition of the sender as to which data units need to be sent to which nodes, just in the moment when they are able to receive them correctly. This happens magically, especially when network partitions merge again without any delay and additional communication overhead.

As a first improvement, we optimize *MILE* by reducing the amount of data that needs to be transmitted periodically. The idea is to use simple and well-known hash values. Nodes create short hash values from data units, so-called IDs – sometimes also called metadata – and send these instead of complete data units. This complete list of IDs is broadcast periodically by each node, together with a subset of data units. If a node gets an incomplete list of IDs from a neighbor, it will add the missing data units in its next periodic update packet. By this simple extension we avoid an explicit request of missing data by individual nodes and thereby achieve an additional reduction of bandwidth usage. The drawback is an increased delay, as nodes add the content of the data units only when other nodes within the transmission radius are found that do not know about a particular data unit. We call this extension *MILE on-demand*.

We combine the underlying ideas of *flooding* and *MILE on-demand* for further reducing the communication overhead and increasing the speed of data dissemination, as well as the data delivery ratio. We call the new protocol *AutoCast*; it works as follows, making use of two basic mechanisms.

Newly generated data units are flooded through the network in the beginning, but only a portion of the nodes participate actively in the flooding. Instead of using the magic numbers of 60 % to 80 % as a forwarding probability (as suggested in [16]), we adapt to the dynamics and irregularity of the network. Nodes derive the forwarding probability from their number of neighbors. To avoid broadcast storms, on average only two nodes of those receiving a new data unit rebroadcast it. It has been shown in [17] that on the average only about 40 % of the neighboring nodes receive the data unit for the first time as 60 % of the nodes have received the previous broadcast already. Consequently, a node with 10 neighbors forwards the data unit with a probability of $2/(10 \cdot 0.4) = 0.5$, which is according to the results of [16] for this scenario. However, the forwarding probability for single nodes will decrease further when network density increases, thus ensuring scalability. In a traffic jam, the number of neighbors can reach 100 cars easily where with our approach an individual node forwards data unit with a probability of 5 %.

The second mechanism was introduced by *MILE on-demand*: periodically rebroadcast IDs of elder data units, because due to bad luck, flooding might stop sometimes when several nodes do not forward data. Periodic rebroadcasts are also important to reach locally consistent states in the network, especially when new nodes join the network or network partitions merge. Like the forwarding probability, the rebroadcast interval also depends on the number of neighbors, and in addition on the network dynamics. In a static network in which nodes neither move nor appear, periodic rebroadcasting does not help at all, as flooding already delivered the data units to all reachable nodes. On the other hand, increasing speed of nodes combined with frequent network partitioning will force the update interval to reach zero, which means all nodes broadcast permanently. Thus, the key issue is to determine the optimal update interval; furthermore, how can individual nodes calculate it on their own?

Assume that we know n as the size of a node's one-hop neighborhood. The waiting time until the next rebroadcast is calculated as n/p_{ref}, where p_{ref} is a constant that describes the desired number of broadcasts per second. We will explain our choice of p_{ref} more detailed in Section V. A positive side-effect is the following: cars driving near a network partition boundary send twice the number of packets as cars driving inside that partition, as border nodes have only half the expected number of neighbors.

IV. SIMULATION SETUP

After having discussed five different approaches (including the *AutoCast* protocol), we set up a simulation environment to evaluate and compare them.

We have chosen a rather dynamic highway scenario with varying network density and the influence of opposite-lane traffic onto our protocol's performance. Cars drive on a highway section of ten kilometers, with two lanes in each direction and an average speed of 100 km/h. In order to reach realistic node movements that will appear in VANETs due to individual cars' behavior, we decided to use the traffic simulator SUMO (see [18]), which is based on the microscopic car following model as described by Krauß in [19]. The mean distance between two consecutive cars on one lane is around 110 m. Taking all lanes into account leads to a mean car density of 36 cars/km. Because road density is hard to compare to other simulation setups, Table I shows neighborhood sizes in our setup that result from different fractions of cars equipped with AutoNomos devices, so-called penetration rates.

The nominal duration of our traffic simulation is 26 min, with an initial startup time of 10 min to spread the cars all over the road. The last 16 min of the generated cars' mobility are stored into ns2-trace files, each with a different penetration rate.

ns-2 [20] is used as network simulator for performance evaluation of the different data dissemination protocols. All simulations use standard IEEE 802.11 MAC-layer, with a radio range of 250 m and a bandwidth of 1 Mbps in combination with the Two-Ray Ground propagation model. Periodically, the car driving closest to km 5 at the appointed time generates a data unit. It is published all over the simulated road (5 km in each direction). The lifetime of such a data unit is set to 50 s.

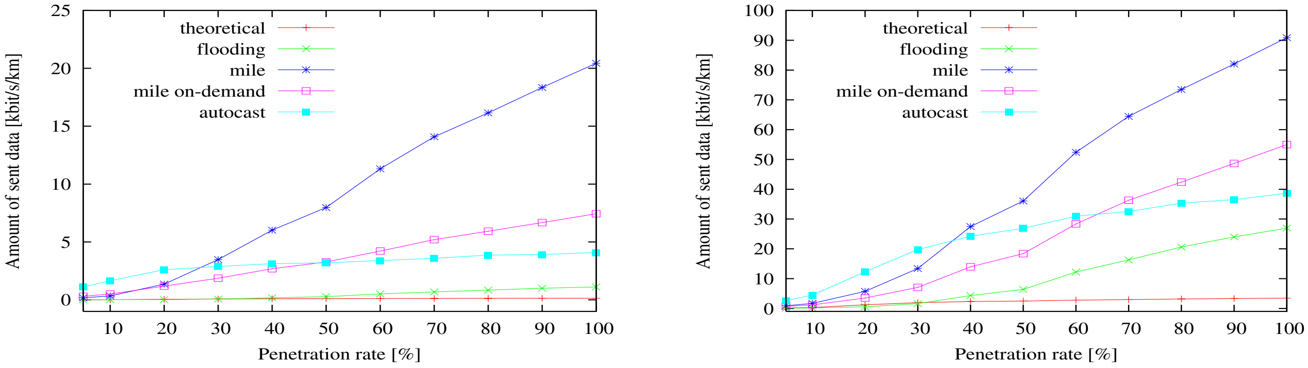
Each protocol is simulated with different penetration rates, as shown in Table I, and between two and 50 data units that need to be disseminated concurrently.

V. RESULTS

Figure 2 shows the results of the simulation, with each line in the graph showing a protocol. In all plots the x-axis shows the penetration rate of cars that participate in the VANET. The left column shows the protocols' behavior, if only two data units are disseminated. Figures on the right show the results for 50 concurrent data units. In order to leave enough network capacity for other applications and protocols, data dissemination should be optimized for low bandwidth; Figures 2(a) and 2(b) show the transmitted data per km, as concurrent communication is possible if sending nodes have a distance of more than four times the transmission radius. Figures 2(c) and 2(d) show the achieved speed of data dissemination. A fast speed is preferable, because data units may comprise time-critical data like emergency messages. To evaluate the success

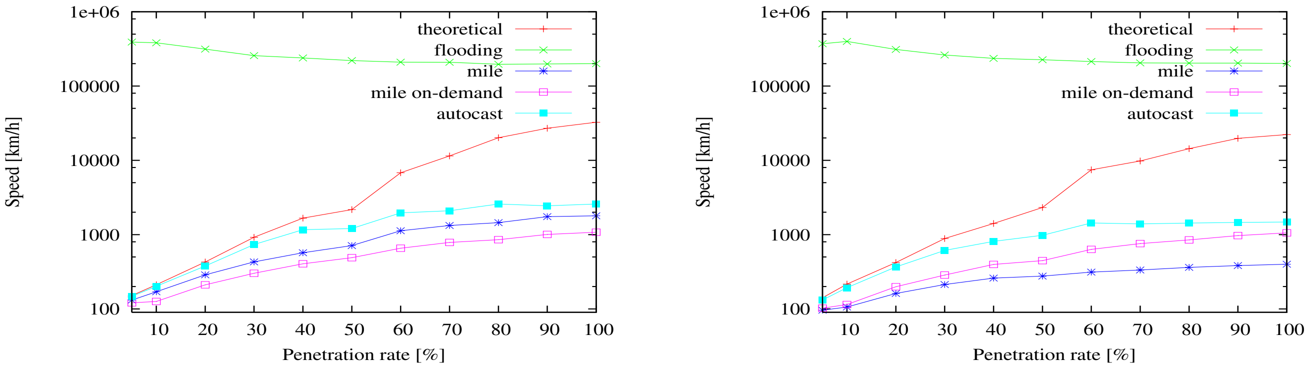
TABLE I
AVERAGE NEIGHBORHOOD SIZES FOR DIFFERENT PENETRATION RATES.

penetration rate [%]	5	10	20	30	40	50	60	70	80	90	100
neighborhood size	0.9	1.8	3.6	5.4	7.2	9	10.8	12.6	14.4	16.2	18



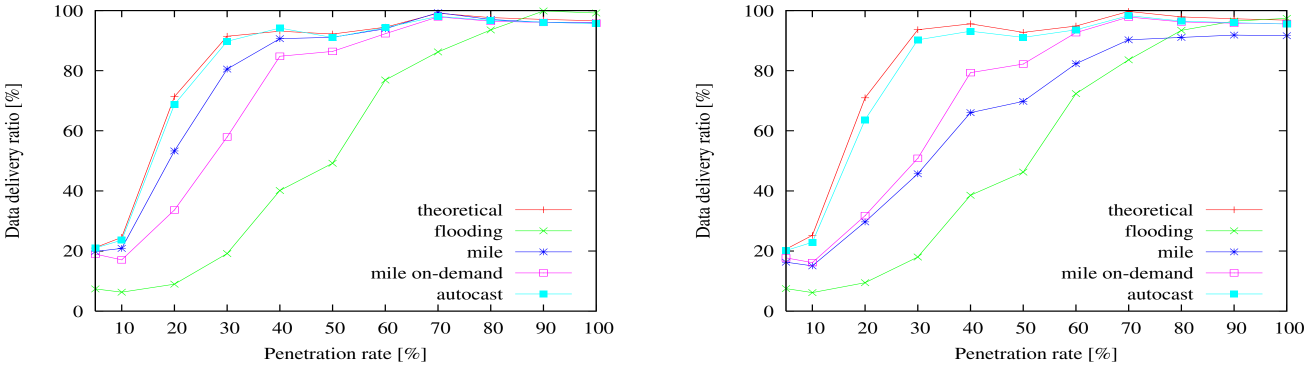
(a) Radio channel usage per km, 2 simultaneous data units

(b) Radio channel usage per km, 50 simultaneous data units



(c) Speed of data traffic, 2 simultaneous data units

(d) Speed of data traffic, 50 simultaneous data units



(e) Data delivery ratio, 2 simultaneous data units

(f) Data delivery ratio, 50 simultaneous data units

Fig. 2. Simulation results comparing the different data dissemination algorithms for few data units (left column) and 50 data units (right column).

of data dissemination, Figures 2(e) and 2(f) present the amount of successfully delivered data units.

The *theoretical* protocol sends a broadcast only if it will successfully inform a car, so less than 4 kbit/s/km of bandwidth are consumed in any case. With low penetration rate, the bounding factor for data speed is almost completely the cars' driving speed. As expected, it rises with increasing penetration rate, up to more than 20000 km/h. This speed cannot be achieved by any other protocol, as in reality there is a trade-off between data speed, data delivery ratio and rebroadcasting interval. The data delivery ratio shows that a reasonable usefulness can be achieved with a minimal penetration rate of 30 % standing for 10.8 equipped cars per km. With a further increase in the number of cars, the ratio of delivered data units grows only marginally.

At first sight the *flooding* protocol performs surprisingly well. It consumes few bandwidth and achieves a speed of above 100,000 km/h. However, the poor data delivery ratio puts that result in the right perspective as with pure flooding a data unit will stay in its network partition. Consequently, flooded data is delivered either very fast or never.

With regard to enhancing the data delivery ratio, in particular in the case of low penetration rates, the *MILE* protocol achieves a remarkable improvement, approaching the theoretical results. If more data units need to be disseminated than fit into one broadcast packet, the achieved data speed decreases from nearly 1800 km/h to under 400 km/h, even in case of a 100 % penetration rate. Moreover, the data delivery ratio drops as well.

Due to the exchange of data unit IDs, the protocol *MILE on-demand* can suppress the rebroadcasting of full data units that are already known by cars in the direct vicinity. The drawback of this method is a slight decrease in data speed, because the sender needs to know about missing data units before delivering them. Due to this effect *MILE on-demand* performs worse than pure *MILE* in case of only few data units. Nevertheless, the protocol's performance remains stable if more data units need to be handled. So far all protocols use fixed broadcast intervals of 2 s. This results in a linear increase of bandwidth usage when more cars participate in the VANET.

As mentioned in Section III, *AutoCast* produces a constant number of broadcast packets per second (p_{ref}), no matter how many nodes generate them. In order to calculate p_{ref} for our scenario, we analyze the *MILE on-demand* curve and find a minimum of 60 % penetration rate for a data delivery ratio above 90 %. With 10.8 neighbors, i.e., $p_{\mathrm{ref}} = 10.8$ cars in the neighborhood per 2 s, about 5 packets/s are transmitted. The value of p_{ref} is a good choice for our scenario, but is definitely not the optimum for all ad-hoc networks. This parameter needs to be analyzed in more detail; we will address this problem in future work. Nevertheless, bandwidth consumption remains stable, independent of the network density, and depending only on the number of concurrent data units. The data speed reaches about 2000 km/h, enough to cross Germany in less than 30 min. The data delivery ratio gets close to the *theoretical* protocol, so even the primary goal of reaching as many cars as possible is achieved.

AutoCast clearly outperforms the other protocols and gets close to the theoretical maximum with respect to data dissemination speed and data delivery ratio. Due to a limited network overhead, it leaves enough room for additional applications and protocols in the ad-hoc network.

VI. CONCLUSION AND FUTURE WORK

In this paper we illustrated the need for a new data dissemination mechanism serving a class of applications where information is more important than originator and forwarders as nodes do not know each other. This problem was not addressed fully in the past so that we derive a solution for this problem suitable for wireless networks as a trade-off between data dissemination speed, communication overhead and delivery ratio. We demonstrated the efficiency and the potential of this general purpose approach in comparison to a theoretical optimal reference. Assuming a 802.11 WLAN with 1 Mbps and high network dynamics as well as density, even an amount of 50 concurrent data units consume less than 5 % of the bandwidth, leaving enough room for many additional protocols and applications.

We are currently working on the challenges that were not focussed in this paper. Those comprise security and restricting amount of data by aggregation to avoid jam in the data network. An interesting problem still to solve is the optimal choice of p_{ref} as the number of every second rebroadcasts in the neighborhood, based just on local decisions. Finally, we need to implement the overall functionality for the AutoNomos project including global strategies for traffic regulation.

ACKNOWLEDGMENTS

The authors would like to thank the DFG for the funding of the project AutoNomos.

REFERENCES

[1] A. Wegener, E. M. Schiller, H. Hellbrück, S. P. Fekete, and S. Fischer, “Hovering data clouds: A decentralized and self-organizing information system,” in *Proceedings of International Workshop on Self-Organizing Systems*, sep 2006, pp. 243–247.

[2] S. P. Fekete, C. Schmidt, A. Wegener, and S. Fischer, “Hovering data clouds for recognizing traffic jams,” in *Proceedings of 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*, 2006, pp. 213–218.

[3] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *ACM/IEEE Trans. Netw.*, vol. 11, no. 1, pp. 2–16, feb 2002.

[4] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, “A scalable content-addressable network,” in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2001, pp. 161–172.

[5] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, “Data-centric storage in sensornets,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 137–142, 2003.

[6] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, “A two-tier data dissemination model for large-scale wireless sensor networks,” in *Proceedings of the eighth annual international conference on Mobile computing and networking*. ACM Press, 2002, pp. 148–159.

[7] W. Franz, H. Hartenstein, and M. Mauve, *Inter-Vehicle-Communications Based on Ad Hoc Networking Principles - The FleetNet Project*. Universittsverlag Karlsruhe, 2005.

[8] Q. Xu, T. Mak, J. Ko, and R. Sengupta, “Vehicle-to-vehicle safety messaging in DSRC,” in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. ACM Press, 2004, pp. 19–28.

[9] S. Oh, J. Kang, and M. Gruteser, “Location-based flooding techniques for vehicular emergency messaging,” in *Proceedings of 2nd International Workshop on Vehicle-to-Vehicle Communications*, jul 2006, pp. 1–9.

[10] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, “Adaptive broadcast for travel and traffic information distribution based on inter-vehicle communication,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, jun 2003, pp. 6–11.

[11] H. Xu and M. Barth, “An adaptive dissemination mechanism for inter-vehicle communication-based decentralized traffic information systems,” in *Proc. IEEE Intelligent Transportation Systems*, 2006, pp. 1207–1213.

[12] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in *Proceedings of 5th ACM/IEEE MobiCom Conference*. ACM Press, aug 1999, pp. 174–185.

[13] M. Grossglauser and D. N. C. Tse, “Mobility increases the capacity of ad-hoc wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, aug 2002.

[14] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, mar 2002.

[15] H. Hellbrück and S. Fischer, “MINE and MILE: improving connectivity in mobile ad-hoc networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 4, pp. 19–36, oct 2004.

[16] Z. J. Haas, J. Y. Halpern, and L. Li, “Gossip-based ad hoc routing,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, jun 2006.

[17] H. Hellbrück and S. Fischer, “Towards analysis and simulation of ad-hoc networks,” in *Proceedings of the 2002 International Conference on Wireless Networks*, jun 2002, pp. 69–75.

[18] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, “Sumo (simulation of urban mobility); an open-source traffic simulation,” in *Proceedings of the 4th Middle East Symposium on Simulation and Modelling*. SCS European Publishing House, sep 2002, pp. 183–187.

[19] S. Krauß, “Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics,” Ph.D. dissertation, Universität Köln, 1998.

[20] K. Fall and K. Varadhan, *The ns Manual*, The VINT Project, 1989–2001. [Online]. Available: www.isi.edu/nsnam/ns/doc/