Adaptive Topology Based Gossiping in VANETs Using Position Information*

Boto Bako, Igor Rikanovic, Frank Kargl, Elmar Schoch

Ulm University, Institute of Media Informatics
{*givenname.surname*}`@uni-ulm.de`

Abstract. Gossiping is a lightweight and simple technique for information dissemination in many application domains, be it in Wireless Sensor Networks (WSNs), Mobile Ad-hoc Networks (MANETs), or Vehicular Ad-hoc Networks (VANETs). Much research has been conducted in the past on probabilistic dissemination methods because of their efficiency compared with simple flooding and their simple application. However most work was focused on static gossiping, i.e., the gossiping probability cannot be adapted according to topology changes. Thus, topology characteristics have to be known in advance.
In this work the use of position information for building up a neighborhood relationship is proposed. Based on this information, a forwarding hierarchy is constructed and the protocol is capable to adjust the dissemination probability dynamically in a distributed manner. The protocol is evaluated in a highway scenario, where the network characteristic varies from sparse networks with highly mobile nodes to a traffic jam with very high node density and low node velocities. The applicability of the proposed protocol for such scenarios is shown by simulations.

Keywords: Information dissemination, gossiping, probabilistic broadcasting, vehicular ad hoc network (VANET)

1 Introduction

Controlled dissemination of information plays an important role in many network scenarios like wireless sensor or vehicular ad-hoc networks. A message is broadcasted to all nodes in a network to fulfill application specific services or to enable some basic mechanisms for routing. In sensor networks it is used e.g. to disseminate TAG-type queries [1], to broadcast control messages and so on. Flooding is the simplest form of broadcasting and is used also in many routing protocols, e.g. in DSR [2] and in AODV [3] to establish routes. Also protocols in the automotive domain use flooding, as CGGC [4] for example for area forwarding. However, flooding works well for a small number of nodes but leads to high performance problems in dense networks regarding channel congestion and packet collisions. This leads the so called “broadcast storm problem” [5]. As

---

* This work has been supported by a grant from the Ministry of Science, Research and the Arts of Baden-Wrttemberg (Az: 23-7532.24-14-19) to Boto Bako.

simple flooding wastes a lot of bandwidth, more efficient message dissemination schemes have been developed.

Gossiping is one widely used dissemination technique because of its robustness and scalability. The simplest variants use a global probabilistic value at all nodes and are thus called static gossiping (confer, [6], [7], [8]). All these approaches work only if the network topology is known in advance otherwise it will result in a low delivery ratio or a high number of redundant messages. To overcome these problems adaptive gossiping has been introduced.

Haas et al. [7] introduced the so called two-threshold scheme, an improvement for static gossiping based on neighbor count. A node forwards a message with probability $p1$ if it has more than n neighbors. If the number of neighbors of a node drops below this threshold n then messages are forwarded with a higher probability $p2$. The obvious advantage of this improvement is that in regions of the network with sparse connectivity messages are prevented to die out because the forwarding probability is higher than in dense regions.

[7] also describes a second improvement which tries to determine if a message is “dying out”. Assuming a node has n neighbors and the gossiping probability is p then this node should receive every message about $p \cdot n$ times from its neighbors. If this node receives a message significantly fewer, the node will forward the message unless it has not already done so.

In [5], Ni et al. introduced the Counter-Based Scheme. Whenever a node receives a new message, it sets a randomly chosen timeout. During the timeout period a counter is incremented for every duplicate message received. After the timeout has expired, the message is only forwarded if the counter is still below a certain threshold value.

Although all these adaptations improve the broadcast performance, they still face problems in random network topologies. For example, a node may have a huge number of neighbors, resulting in a small forwarding probability in all of these schemes. Despite this, there could e.g. still be a neighbor which can receive the message exclusively from this node. An example of such a situation is shown in Figure 1 (example taken from [9]).

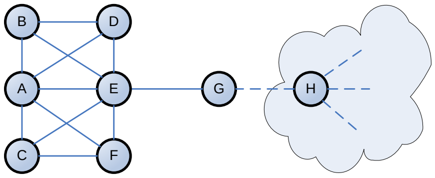

Fig. 1. Sample topology where static gossiping fails

When node A sends a message, all nodes in its neighborhood receive it. In this example scenario only node E should forward it with the probability of 1 since E is the only node that can propagate the message to node G. If the gossiping probability is only based on the neighbors count, node E will be assigned a low probability since it has many neighbors. So the broadcasted message will "die out" with a high probability and never reach G and all later nodes. If the part of the network connected only via G is very large, the overall delivery ratio will drop dramatically. Such situations can occur quite regularly in dynamic networks of a certain density.

Recently, the Smart Gossip protocol which addresses this problem was introduced by Kyasanur et al. [9]. The protocol assumes a static network with one single message source. Every node in the network uses neighborhood information from overheard messages to build a dependency graph. I.e., each node has a parent, sibling, and child set. Parents are nodes where this node receives new messages from, siblings receive messages from the same parents and the node delivers messages to child nodes.

Depending of the number of parents, every node calculates the probability by which its parents should forwards a message and informs its parents about this probability. A parent sets its forwarding probability to the maximum of all child probabilities. If a node has only one parent, the forwarding probability will be automatically set to 1, if a node has many parents, the probability will be comparatively small, but still large enough to ensure that the node will most likely receive the message at least once.

Smart Gossip solves the problem of adapting the forwarding probability dynamically, but in some cases there are still disadvantages: the described parent child relationship is dynamic, depending on which node sends or forwards a message. For ensuring to build up a stable directed graph, the authors make the assumption that there is only one message originator in the whole network. This assumption however cannot be fulfilled in our scenario. We want to apply a gossiping like message dissemination for vehicular ad hoc networks (VANETs) where the nodes are moving vehicles. VANETs have quite different properties compared e.g. with sensor networks (WSNs). In the upcoming section we give a brief overview over these characteristics and discuss their impacts to gossiping.

2 Scenario Description and Assumptions

VANETs represent highly dynamic networks. Due to the high mobility and speed of vehicles, the network topology changes rapidly. Two vehicles might be in mutual communication range only for a few seconds. Additionally, the properties like speed or movement patterns can vary significantly within VANETs.

On a free highway for example, vehicles can move with high velocity, whereas, on the same highway in a traffic jam the number of vehicles is extremely high and the velocity drops to nearly zero. In the first case there is a highly mobile and sparse network. Nodes are moving with high speeds, neighborhood changes constantly and the network is often partitioned. For this case because of the low density the message dissemination algorithms needs to fall back to flooding and forward the messages in every node.

On the other hand, the challenge in very dense networks like in traffic jams is to avoid channel congestion through an effective broadcasting algorithm. In such dense networks the overall node mobility is low, thus neighborhood changes are infrequent and the topology is almost static.

As an application example, consider a traffic information system where vehicles detect traffic density and report this information to other vehicles so they get a complete picture of the traffic jam. This information needs to be disseminated in a certain area and the message has to reach all vehicles within this area if possible. At the same time the number of retransmissions has to be minimized to avoid a broadcast storm.

In contrast to WSNs, we assume that energy consumption is not an issue. The vehicles will provide the necessary power and computing capacity. Another likely assumption is that all VANET-enabled vehicles will be equipped with a GPS receiver, thus they can determine their positions. This information is used in the proposed protocol to achieve effective message dissemination. There is another important difference from WSNs: there typically one dedicated node broadcasts e.g. control messages into the network. In the mentioned traffic jam scenario, it is quite obvious that every vehicle could be the originator of messages which inform other vehicles about its own status, detected road condition, traffic density, etc..

The Smart Gossip protocol is a promising approach for efficient message dissemination in WSNs but will fail in the presented scenario. These drawbacks are discussed in the next section.

3 Review of Smart Gossip with multiple originators

As mentioned before, one of the assumptions of the Smart Gossip protocol is that only one node in the network initiates broadcasting messages. This assumption is required to build up a correct parent-sibling-child relationship. The hierarchy is established by overhearing of normal gossip messages. The initial gossip probability is 1 and is adjusted step by step as more topology information becomes available.

Based on the sample topology in Figure 2 we will show where the original Smart Gossip algorithm fails, if there is more than one originator.

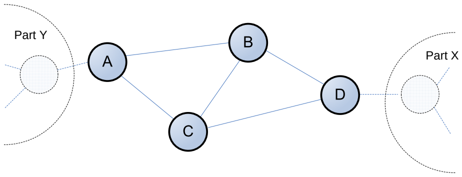

Fig. 2. Sample topology where Smart Gossip fails with multiple originators

We assume that nodes A and D are originators of gossip messages. If node A and D begin to send messages, the nodes B and C receive these messages and include A and D in their parent sets. According to the Smart Gossip terminology B and C are children of nodes A and D, A and D are parents of B and C, and B and C are siblings.

Because B and C have no children themselves, they need not forward a received gossip message. In the Smart Gossip protocol they nevertheless forward such messages with a low probability which achives a higher robustness. In this situation, messages which are created in network part X will likely be dropped by B and C and will most likely never reach part Y and vice versa. Hence, the average reception percentage in such topologies drops significantly. Although Figure 2 shows an artificially created topology, such node distributions can occur frequently in sparse areas of dynamic networks.

As we have shown, for vehicular networks we need a dissemination protocol which can cope with multiple message originators and therefore considers the direction of gossip messages while creating the hierarchy. I.e., messages that node B and C receive from node A have to be treated differently as messages from D. For our approach we keep the basic idea of the Smart Gossip protocol to establish a parent-sibling-child relationship and derive the gossip probability from this knowledge, but we use a different approach to build up the hierarchy. For being able to differentiate the direction of diffused messages, our approach uses position information.

Additionally, the proposed approach has to deal with varying degrees of mobility. This challenge is not discussed in [9], the authors only consider packet loss due to bad wireless links. This is a second reason why the original Smart Gossip protocol can't be applied in mobile ad hoc networks: the hierarchy is established in a static way and no renewal of neighborhood information is considered.

The main contribution of this work is the design and analysis of a new mechanism for hierarchy creation which considers message directions through position information and treatment of mobility. For this purpose the neighborhood information is not passively collected by overhearing normal gossip messages, but by active exchange of beacon messages. This enables the nodes to receive up to date information of their vicinity in mobile environments. Details of these process are discussed in the next section.

4 Protocol Description

In VANETs the message propagation – as well as node movement – is restricted to streets. Leaving intersections aside, messages can be propagated in two directions: in and against the driving direction. Therefore, e.g. in highway scenarios it is still possible to build a dependency graph even with multiple senders. The protocol has to distinguish between these two possible dissemination directions and build up the hierarchy accordingly. For solving the problem of message propagation direction, information about node positions is used in this work.

The idea is to build the parent-sibling-child relationship in a directed way. I.e. a node can only have parents from one direction. If we resume Figure 2 again, this means, that only node A or D could be a parent of nodes B and C and not both together. Thus, depending on the direction, node A is declared as parent and node D as child or vice versa. Based on the propagation direction of messages this means, that the hierarchy can be built in two ways: in driving direction and against driving direction. If we consider a traffic jam scenario, where vehicles which are approaching the traffic jam have to be informed, it is necessary to build the hierarchy against driving direction. In this case D would be defined as child and A as parent of B and C. In this work the hierarchy is built against the driving direction, since it satisfies our requirements of the highway scenario.

Of course, for some scenarios it would be desirable to send messages in both directions. This is simply achievable if at each node two different neighbor tables are held containing the parent-sibling-child relationships differentiated into the two possible directions.

Listing 1.1. Pseudo code of the proposed protocol

```
Receive_Beacon(fromNode j)
{
  RemoveFromNeighborSets(j);
  if (parent(j) not in NeighborSets &&
                position(j) in driving direction){
    AddToParentSet(j);
  }
  else if (parent(j) not in NeighborSets &&
                position(j) not in driving){
    AddToChildSet(j);
  }
  else if (parent(j) in SiblingSet){
    AddToChildSet(j);
  }
  else if (parent(j) in ParentSet){
    AddToSiblingSet(j);
  }
}
```

As mentioned in the last section another difference to the Smart Gossip protocol is the use of beacons for building the hierarchy. Every vehicle generates beacons and sends them to their 1-hop neighbors. Unlike gossip messages, beacons are not forwarded by the nodes, they are only used to exchange neighborhood information for building up the parent-sibling-child relationship. Additionally, mobility is an important factor. Due to mobility, the neighborship of a node changes frequently. Therefore building the hierarchy must be a continuous process, which is carried out in regular intervals to adapt to topology changes quickly.

To deal with fast changes we added a timestamp based check if entries in the neighborhood table are up-to-date. If a node does not receive a beacon from a neighbor within a certain period it removes the neighbor from its neighbor table. It is obvious, that in this case the hierarchy has to be adapted to the changing neighborship. Therefore, at each reception of a beacon the originator is assigned a role (parent, sibling or child) according to the newest neighborhood information.

The code listing 1.1 shows how the neighbor relationship is established step by step when a node receives a beacon message. At each reception of a packet the method *Received_Beacon* is called. First the sender is removed from the neighbor tables, which are the parent, sibling and child sets. If the parent of the sender (node j in this case) is not in the neighbor tables (thus it is unknown at the receiver node) and the sender is in front of the receiving node, the sender is inserted into the parent set. If the parent of the sender is unknown, but the sender is behind of the receiver, the sender is inserted into the child set. When the parent of the sender is known, then depending if it is in the sibling or parent set, the sender is added to the child or sibling set.

In the following an illustrative example for building this parent-sibling-child relationship is given. Figure 3 shows the initial situation. There are 4 vehicles

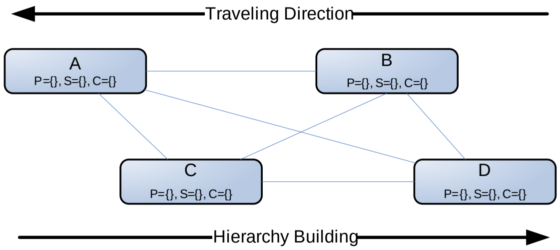

Fig. 3. Example for building the neighbor relationships: initial situation

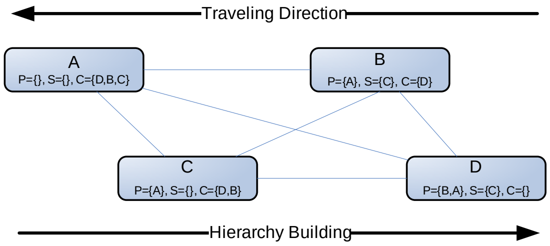

Fig. 4. Example for building the neighbor relationships: situation after all nodes sent 1 beacon

(Nodes A, B, C and D) and they have no information about each other so far. This means, the neighbor tables (parent, sibling and a child set) of those nodes are empty. Since every vehicle sends periodically beacons (including position information and a list with its parents), the hierarchy can be built up step by step. Assume in this example that node D sends first such a beacon which is received from the other three nodes. The nodes A, B and C have to determine their relation to node D. First, the direction to D is checked. Since D is behind the other nodes, it cannot be a parent node. Because the received beacon has an empty list of parents of the sender, D can not be a sibling of the other nodes. Thus, D is put into the child set of vehicles A, B and C.

Continuing the example, assume node B sends the next beacon. The nodes A and C put B into their child list for the same reason as before. Node D determines the direction to node B. Since B is before D, it can not be a child. And because B has no parents determined yet, they can not be siblings. Thus, node D adds B into its parent set. Next, let node A send a beacon. Nodes B, C and D receive the beacon and insert A in their parent set because node A has not the same parents as the other nodes (actually A has no parents at all) and the position of A is in front of B, C and D. In the next step node C sends a beacon. A inserts C in its child set because the parent of C is A itself. Nodes B and D have the same parent as C (node A) and therefore they put C in their sibling set. Figure 4 shows the updated neighbor tables after this step.

In the following a second round is considered where the nodes send in the same order as before. This will show how the topology neighborhood relationship is adapted if newer or more precise information are available. Node D sends a beacon again. For node A there is no change because in the beacon message A is

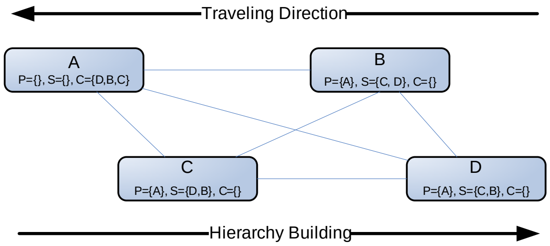

Fig. 5. Example for building the neighbor relationships: the final neighborhood relationships

specified as a parent of D and D is already a child of A. But B and C put node D in their sibling set because they have the same parent as D. Next, node B sends a beacon. Again the sets of A are unchanged. Node C and D insert B into their sibling sets cause they have the same parent as B. With further beacons the neighbor tables does not change anymore, since the relationship is already built up correctly. Figure 5 shows the final neighbor tables of these nodes.

Of course, due to mobility the neighbor tables can change again. Nodes can leave the neighborhood of other nodes, or new nodes can arrive but also a shift of the relationship based on the relative positions is possible. But through the use of beacons and timestamps for establishing the relationship, accurate neighbor relationships can be maintained even for highly mobile nodes. An evaluation of the performance for static scenarios as well as for highly mobile nodes is given in the next section.

5 Analysis

To evaluate the performance of the proposed dissemination protocol and to compare it with the Smart Gossip approach we have conducted extensive simulations. For this study we used the Java based network simulator JiST/SWANS [10]. The simulation setups are divided into three parts: in the first setup we use similar simulation parameters as in [9] and make a comparison between our protocol and Smart Gossip with multiple message originators. In the second setup we compare the performance of the two protocols in the highway scenario described in Section 2. We evaluate the performance – reliability and efficiency – of these protocols for varying node densities. In the last setup we investigate the impact of mobility and show the simulation results for different node speeds and densities.

For the first simulation setup, 50 nodes are randomly placed on a field with a size of 1000 per 1000 meters. The wireless transmission range is set to 280 m. Mobility is not considered in this setup, thus nodes are static. The only difference to the parameters used in [9] is the number of message originators in the network. In this setup multiple nodes can initiate broadcast messages. As it can be seen in Figure 6, the delivery ratio of the Smart Gossip protocol drops notably. The fluctuation of the achieved delivery ratio is high and in many cases it drops below 90%. According to the authors from [9] the delivery ratio should be about 99% with one message originator. Thus, multiple message originators have a

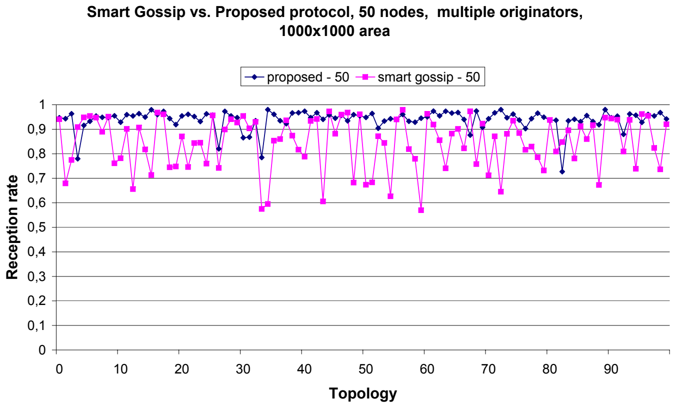

Fig. 6. Performance evaluation of Smart Gossip and proposed protocol with multiple originators

high impact on the Smart Gossip protocol, in a scenario with low node density as in this case. On the other hand, our proposed protocol achieves much better results and a lower deviation at the same time. For a better comparability we included pure flooding into our simulations. The average delivery ratio for the three broadcasting mechanisms in 100 simulation runs is shown in Table 1. As we can see the best delivery ratio is achieved with flooding. This is obvious, since all nodes retransmit the broadcasted message, achieving a high reliability at the cost of communication complexity. This is the main problem of flooding, especially in dense networks the high number of redundant messages causes channel congestion, resulting in a drastically dropping of the delivery ratio in that case. On the other hand, the delivery ratio between our proposed protocol and flooding differs only by 2%, while the original Smart Gossip ratio is in average almost 10% lower than the proposed protocol.

Flooding	Proposed protocol	Smart Gossip
95.8%	93.9%	84.4%

Table 1. Delivery ratio Flooding, Proposed protocol and Smart Gossip

So far a comparison of the selected broadcast protocols was given in a network topology as used in [9]. Now the focus lies on the highway scenario. Therefore a field of 1000 per 10 meters is used, which should represent a road segment. The other simulation parameters are the same as in the last simulations.

Figure 7 shows the result for varying node densities. As it can be seen, the proposed protocol outperforms Smart Gossip for almost all evaluated node densities in terms of reliability and communication complexity. In the case of 50 nodes the Smart Gossip protocol has a lower average forwarding rate than the proposed. But it should be considered that also the average reception ratio for the Smart Gossip protocol at this node density is approximately 10% lower.

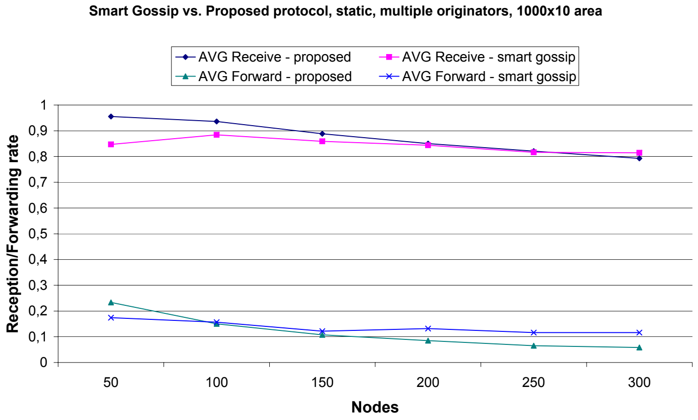

Fig. 7. Performance evaluation of Smart Gossip and proposed protocol with multiple originators on a 1000 per 10 meter field

Therefore, a new metric is needed that combines these two measured values – reception rate and forwarding rate – and enables thus a better comparison between the protocols. This combined metric can be specified in the following way:

$$Efficiency\ Rate = \frac{Reception\ Rate}{Forwarding\ Rate}$$

	Smart Gossip			Proposed Protocol		
Nodes	*Reception*	*Forwarding*	*Efficiency*	*Reception*	*Forwarding*	*Efficiency*
50	84.7%	17.4%	4.86	95.5%	23.3%	4.09
100	88.4%	15.6%	5.66	93.6%	15.0%	6.24
150	85.9%	12.2%	7.04	88.8%	10.8%	8.22
200	84.3%	13.2%	6.38	85.0%	8.4%	10.11
250	81.6%	11.6%	7.03	82.1%	6.5%	12.63
300	81.4%	11.6%	7.01	79.3%	5.8%	13.67

Table 2. Performance comparison between Smart Gossip and proposed protocol

The higher the efficiency rate is the better is the performance of a protocol. Table 2 gives an overview of the reception rate, forwarding rate and efficiency rate for both protocols. As this values show, the efficiency rate of Smart Gossip is only at 50 nodes better. But the delivery ratio at this density is approx. by 10% lower. If an application needs a high reception rate for this node density, our protocol is the better choice. All other efficiency ratios show that our protocol outperforms with an increasing node density more and more the Smart Gossip protocol. Thus, this dissemination mechanism is better suited then Smart Gossip in a highway scenario for a wide range of road traffic: it performs well in low densities as well as in traffic jams.

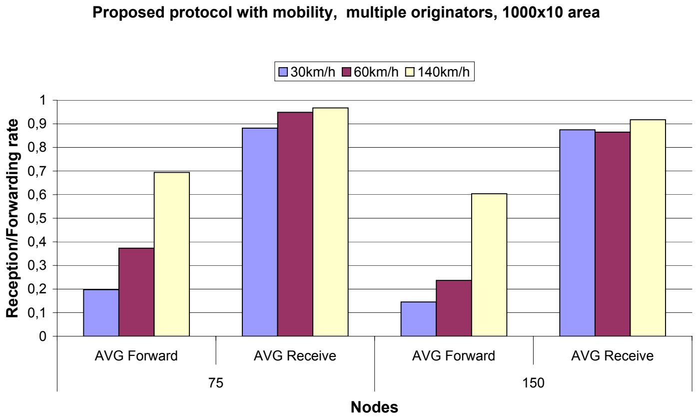

Fig. 8. Impact of mobility on the proposed protocol

In the last simulation setup the performance of the proposed protocol with mobile nodes is investigated. These simulations have been carried out only with our proposed protocol, since the Smart Gossip protocol is not designed to deal with mobility. The neighbor relationship is built up in a static way and no mechanisms were considered to hold such a hierarch up-to-date as it is needed in mobile environments.

For this simulation, also the highway scenario is used, thus nodes are placed into a field with a size of 1000 per 10 meters. For this evaluation the random waypoint mobility model was used, with different node velocities. This mobility model doesn't fit the realistic movements of vehicles on a highway. Nevertheless, this represents a worst-case scenario since nodes are moving in arbitrary directions. A directed movement of nodes into the same direction would better fit to the nature of our protocol where the hierarchy is built depending of road directions. Therefore, if the performance of the protocol is sufficient for this use case, then it should be by far better with a realistic mobility model. As Figure 8 shows, mobility has only a very small impact on the delivery ratio. On the other hand, with higher node velocity the forwarding ratio grows. This is due to the fact, that with higher node speeds the entries in the neighborhood tables are going to be outdated very often. Thus the neighborhood of a node changes often and nodes send with probability 1 if their vicinity is unknown.

It should be noted that in all simulations in this section the delivery ratio was measured by the percentage of reception of a broadcast message at all nodes. This means, a broadcast message is delivered in both directions: into driving direction and against. It is obvious that this situation is not well suited for the proposed protocol. In our approach a directed dependency between neighbors is built, thus a message should be forwarded only against the driving direction. For such a directed forwarding the proposed protocol should achieve a much better performance. We used the general case (actually the worst case) for being able to compare our approach with Smart Gossip in a scenario with multiple message originators.

6 Summary and Outlook

In this work a new probabilistic broadcasting approach derived from the Smart Gossip protocol is introduced. It uses position information to build up neighborhood relationship and – in contrast to Smart Gossip – supports multiple data sources and node mobility. Based on information from beacon messages, appropriate forwarding probabilities can be calculated. This way our approach can adapt well to different network topologies and node densities. The forwarding hierarchy is built up in a directed way. Therefore, the protocol can be applied for networks where nodes have a directed movement, like VANETs for example.

As shown by simulations, the protocol performs well in a wide range of VANET scenarios: it delivers good results in low density networks as well as in networks with high node density. Moreover, node mobility is also considered in this work.

Future work has to show the applicability of the protocol in specific VANET applications: for example a traffic information system which informs upcoming vehicles about traffic density down the road. In such cases, the performance of our protocol should be even better than in the presented simulations, as the messages are forwarded only in one direction and not into the whole network. Other aspects to look at in future work are the investigation of link losses and their impact on the protocol, the evaluation of mechanisms that ensure the robustness of the protocol in such cases, and solutions to deal with road intersections.

References

1. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TAG: a Tiny AGgregation service for ad-hoc sensor networks. Volume 36., New York, NY, USA, ACM Press (2002) 131–146
2. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski, T., Korth, H., eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996) 153–181
3. Perkins, C.E., Royer, E.M.: Ad-Hoc On-Demand Distance Vector Routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computer Systems and Applications. (1999) 90–100
4. Maihöfer, C., Eberhardt, R., Schoch, E.: CGGC: Cached Greedy Geocast. In: WWIC. (2004) 13–25
5. Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The Broadcast Storm Problem in a Mobile ad hoc Network. In: MOBICOM. (1999) 151–162
6. Chandra, R., Ramasubramanian, V., Birman, K.: Anonymous Gossip: Improving Multicast Reliability in Mobile Ad-Hoc Networks. Technical report, Ithaca, NY, USA (2001)
7. Haas, Z.J., Halpern, J.Y., Li, L.: Gossip-based ad hoc routing. IEEE/ACM Trans. Netw. **14** (2006) 479–491
8. Miller, M.J., Sengul, C., Gupta, I.: Exploring the Energy-Latency Trade-Off for Broadcasts in Energy-Saving Sensor Networks. icdcs (2005) 17–26
9. Kyasanur, P., Choudhury, R.R., Gupta, I.: Smart Gossip: An Adaptive Gossip-based Broadcasting Service for Sensor Networks. In: Mobile Adhoc and Sensor Sysetems (MASS), IEEE International Conference. (2006) 91–100
10. Barr, R., Haas, Z.J., van Renesse, R.: JiST: an efficient approach to simulation using virtual machines: Research Articles. Softw. Pract. Exper. **35** (2005) 539–576