An Efficient Scalable Trajectory Based Forwarding Scheme for VANETs

Houda Labiod
TELECOM ParisTech,
46, rue Barrault 75013 Paris, France
Email: labiod@telecom-paristech.fr

Nedal Ababneh
TELECOM ParisTech,
46, rue Barrault 75013 Paris, France
Email:ababneh@telecom-paristech.fr

Miguel García de la Fuente
Telecom Unit, Robotiker-Tecnalia,
Zamudio – Spain
Email: mgarcia@robotiker.es

Abstract— **The past few years have seen an increasing interest in the development of vehicular ad hoc networks resulting in many routing protocols proposals. Scalability issues in such networks are attracting increasing attention these days. To improve data delivery performance in large scale networks, we propose SIFT, a trajectory based routing scheme that requires sparse infrastructure and rely on low quality information. It merely uses the trajectory and the location of the last node that forwarded the packet to forward a data packet from a source to a destination. SIFT is evaluated against DREAM, a well-known routing protocol from the literature through realistic simulations using Omnet++. Simulation results demonstrate that under dense deployment, using a realistic mobility model scenario, the proposed scheme performs better in terms of delivery ratio, end-to-end delay and route length.**

Keywords-Vehicular ad hoc networks; Trajectory-based forwarding; Mobility model, Performance evaluation.

I. INTRODUCTION

Advances in wireless communication and the development of the Dedicated Short Range Communication System (DSRC) have made Vehicle to Vehicle communication (V2V) and Vehicle to Infrastructure (V2I) communication possible in mobile ad hoc networks (MANETs). This has given birth to a new type of MANET network known as the vehicular ad hoc networks (VANETs).

An important problem that has to be solved in VANETs is how to exchange traffic information among vehicles in a scalable fashion. To solve this problem we proposed a spatial-aware forwarding scheme, termed SIFT. Our proposal does not require an up-to-date list of active neighbors' positions at each node to be maintained, which allows it to operate without exchanging control messages among network nodes. This feature confers SIFT a very high performance in dynamic scenarios (e.g., VANETs). In contrast to the previously proposed trajectory-based forwarding (TBF) schemes, SIFT proposes a technique where forwarding decisions are shifted from the transmitter to the receiver and are not based on location information but based on timers, which allow network nodes to select themselves in an autonomous fashion as the most appropriate next forwarding node at each intermediate hop without exchanging any type of control messages.

In this paper, we have chosen DREAM [1] to be used for benchmarking purposes against our proposed scheme. Unlike DSR and LAR, DREAM is not an on demand routing protocol. Instead, each node in this proactive location-based protocol maintains a location table for all other nodes in the ad hoc network. To maintain the table, each node transmits location packets to nearby nodes in the ad hoc network at a given frequency and to faraway nodes in the ad ho network at another lower frequency. Since faraway nodes appear to move more slowly than nearby nodes, it is not necessary for a node to maintain up-to-date location information for faraway nodes. Thus, by differentiating between nearby and faraway nodes, DREAM attempts to limit the overhead of location packets. Furthermore, the advantage of exchanging location information is that it consumes significantly less bandwidth than exchanging complete link state or distance vector information, which means that it is more scalable.

The rest of this paper is organized as follows. Section II reviews related work. We present SIFT in section III, which describes design and implementation issues in depth. The simulation environment and system parameters are discussed in section IV. Finally, we present the simulation results in section V, and conclude the paper in section VI.

II. RELATED WORK

Our work has been informed and influenced by a variety of other research efforts proposed in the past few years. There are various proposals in the area of routing in VANETs, mostly validated either using theoretical analysis or simulation, and involve MAC or power control mechanism. These protocols deal with limitations and unique characteristics of VANETs, such as energy usage or network capacity. However, research on routing protocols for VANETs is still at its infancy.

In VANETs, Position-based forwarding [2] makes forwarding decision based on the location of the forwarding node, and the location of the source as well the destination area. The source sender defines the forwarding area between the source and destination areas. The nodes within the forwarding area should forward the message, while the nodes outside should drop it. The design of the forwarding area determines the delivery ratio and the packet communication overhead.

Another type of position-based forwarding is referred as greedy forwarding [3]. Instead of defining forwarding area, the intermediate node forwards the message to the nodes nearest to the destination area or farthest from the immediate sender. Vehicles make the forwarding decision based on the distance to the immediate sender of the packet. Before rebroadcasting a packet, vehicles must backoff a time period, that is, wait a time period between the channel clearance and actual rebroadcast. This time period is inversely proportional to the distance from the destination node to the immediate sender, thus it is shorter for distant receiver to guarantee the maximum progress. Further, the rebroadcast is suppressed by overhearing the rebroadcast of the same message from other vehicles, which decreases the network load at the cost of lowering the redundancy [4]. In general, the backoff time constitutes a large part of delay in the greedy forwarding scheme.

Trajectory-based forwarding is a hybrid forwarding strategy of the source-based routing and greedy forwarding [5]. The approximate trajectory is defined by the source node, and each intermediate node makes geographical greedy forwarding along the trajectory. The source utilizes the digital map and GPS to define the trajectory of the message (i.e., choosing the proper road segments to constitute a shortest or fastest dissemination path). Compared to the position based forwarding, trajectory based forwarding has less data packet overhead at the cost of applying digital map. While the position based forwarding is more robust to the fragmentations of the vehicle groups. Even there is no vehicle on one path, message can be delivered by other paths, and the message delivery is not confined to a single trajectory. On the contrary to protocols such as GPSR or GEDIR, Trajectory-based forwarding offers a better performance, because in VANET it is possible to build a trajectory that can account for obstacles or voids that can produce a long detour of the data packet.

In [6], authors apply the digital map and Dijkstra shortest path algorithm to compute a sequence of junctions for the message to traverse. In [7], both the traffic density and the distance are taken into consideration to choose the road segments with the shortest propagation delay as part of the path. The traffic density of a road segment is measured by its number of lanes.

Based on the above analysis, we can find that existing approaches cannot meet the application requirements in the dynamic environment of VANETs, and we are motivated to study the problem of communication-efficiency in VANETs.

III. The SIFT Forwarding Scheme

In this section, we describe the design of the proposed scheme in depth.

A. Overview

Our proposal is based on the principle of trajectory-based forwarding mechanism. In contrast to the previously proposed trajectory-based forwarding schemes, SIFT is based on broadcast instead of point-to-point transmissions, and it does not require any neighbors' information to be maintained at each node since forwarding decisions are shifted from the transmitter to the receiver. Wireless transmission is broadcast in nature. It provides reachability to all one-hop neighbors. Upon receiving a data packet in SIFT, each node makes a decision on whether to forward it or not based, merely, on node's own position, the transmitter position and the trajectory.

The required information is encoded into the data packet header, hence, nodes do not exchange any control messages. This greatly reduces control overhead and energy consumption. This feature confers SIFT a very high performance in dynamic scenarios, like VANETs.

To support its functions, SIFT includes two phases:

- The trajectory computation phase: The first phase is implemented by the COMPUTE_TRAJECTORY procedure, and it is only executed by the source node before sending a new packet for the first time. This procedure cannot be invoked by intermediate nodes since SIFT is source-based and thus trajectories are established only by the source node. The intention behind this phase is to compute the trajectories and to send the messages for the first time triggering the multi hop forwarding process.
- The packet forwarding phase: The second phase is implemented by the FORWARD procedure, which is executed by each intermediate node when receiving a data packet that passes through the networks on its way towards its destination. The intention behind this phase is to decide at each intermediate node on whether to forward the packet or not.

These two phases are described in detail further, see Fig. 1.

B. Trajectory Computation Phase

SIFT is a reactive source-based routing scheme. Wherein, trajectory is calculated when needed and is just a digital map expressed in a different way, and the map is assumed to be pre-stored in nodes memory.

During the trajectory computation phase, trajectories are calculated. Before the trajectory computation phase, we assume that there is an entity (entity X as in Fig. 1) that creates and stores an appropriate digital map of the network topological environment in all the nodes of the environment.

This phase actually starts when the application layer of the source node S decides to send a message *M* to a destination *D*. *D*, possibly, does not represent a single node, but a set of nodes potentially interested in receiving a certain type of information. The COMPUTE_TRAJECTORY procedure translates the destination *D* expressed as a set of geographical and topological coordinates into a mathematical equation, or a set of equations. In SIFT, we assume that this procedure posses the appropriate mechanisms that allows it to read the digital map and to extract the required information in order to calculate the equation(s) that will conform the trajectory. Once the trajectory is calculated, the COMPUTE_TRAJECTORY procedure creates a routing layer packet P and copies into its data field the entire data field of the application message *M*, see Fig. 2. A data packet *P* contains the following:

- PKTID: this field represents the ID of a Packet *P*. This parameter is filled by the source node before sending the packet for the first time. It must be unique within the entire network; to achieve so, PKTID is implemented as a sequential number obtained from source node's MAC address. This way, not only it can be differentiated the packets of each different node of the network, but also the different packets sent by each node.
- LFN-X and LFN-Y: current position coordinates of the last node that forwarded the packet. When the packet is sent for the first time, these parameters are filled with the current position of the source node. These two parameters are important since they will be used by the

intermediate nodes. Each time the packet is forwarded by an intermediate node, the node updates those fields with its own current position coordinates.

- TR: this field is used to represent the trajectory equation(s).

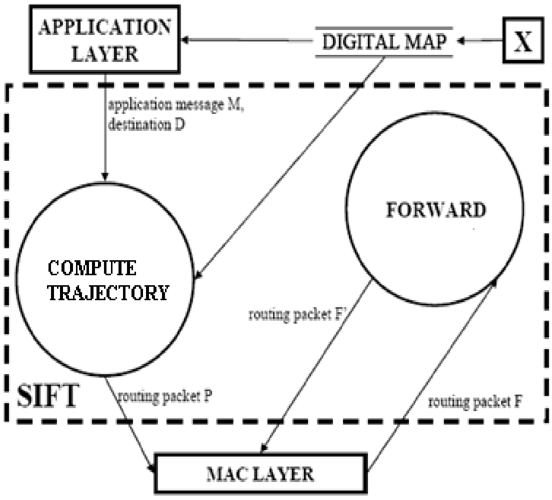

Figure 1. SIFT context data flow diagram.

The packet *P* will be passed to the MAC layer, and then it will be broadcasted to all one-hop neighbors.

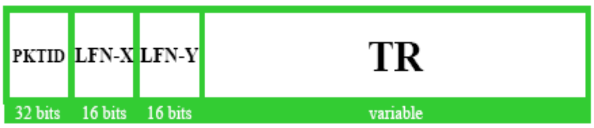

Figure 2. SIFT packet header format.

C. Forwarding Phase

Upon receiving a data packet, a node launches the FORWARD procedure, see Fig. 1. This procedure implements the forwarding scheme proposed by SIFT. Let us suppose that node N receives a packet P from last forwarder node LF, which might be a source or an intermediate node. In any case, packet *P* can be received not only by node *N*, but also by the rest of the one-hop neighbors of node LF. All, potential, forwarding nodes invoke the FORWARD procedure and one node will be elected to forward packet P. This process will be repeated hop by hop until packet P reaches its destination node(s).

Once a message is received by a node, it checks if a copy of this message is received before. If this is a duplicate, it drops the message. Otherwise, it checks its eligibility to forward the received packet P. A node N is eligible to forward the packet if it is located within the trajectory interval, see Fig. 3.

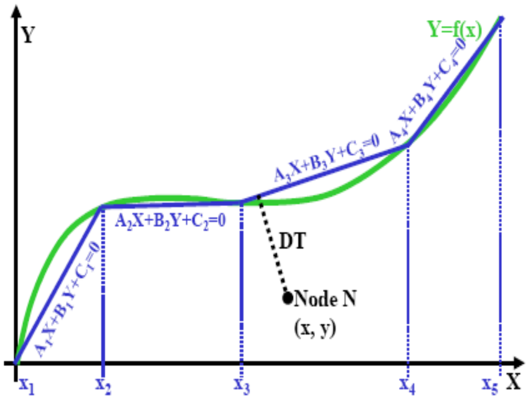

Figure 3. Distance from a node N to the trajectory example.

If the node is located within the admissibility strip margins (i.e., distance to trajectory ≤ admissibility strip threshold), then the distances from node N to the trajectory (DT) and from node N to LF (DL) are calculated as follows:

$$DT = \frac{|Ax+By+C|}{\sqrt{A^2+B^2}} \tag{1}$$

$$DL = \sqrt{(x_N - x_{LF})^2 + (y_N - y_{LF})^2} \tag{2}$$

Forwarding contention may occur when all one-hop neighbors of node LF try to forward the packet *P*. SIFT resolves this contention by enforcing each node to delay its broadcast of packet P for a timeout value *TMR*. Each node chooses a delay value based on its location with respect to the trajectory and to the LF. A node closer to the trajectory with longer distance to LF generates a smaller *TMR* value. At the end of the delay, the node broadcasts the packet to its one-hop neighbors. The TMR value is set as follows:

$$TMR = \alpha \,.\, \frac{DT}{DL} + PD \tag{3}$$

Where, α is an environment adaption constant, and *PD* is the packet propagation delay.

IV. SIMULATION ENVIRONMENT AND SYSTEM PARAMETERS

For the evaluation, we have used the Omnet++ [8] simulation platform and its mobility framework [9]. Network nodes are distributed within the simulation area according to the mobility model described below. The scenario that we have used in our simulation experiments consists of a 1000 m x 1000 m square area. The simulation map is a simple 10 x 10 non-uniform street grid. All nodes are equipped with an IEEE 802.11b communications interface, and the radio range fixed at 100 m. In all simulations, the source and the destination nodes are static. The source node sends one message per second to the destination node, and the simulation lasts for two hours. We compared the performance of our proposal with DREAM, one of the most widely cited position-based protocols. The summary of the simulation environment is shown in Table 1.

For all simulation results in this paper, each experiment is repeated ten times on different network topologies.

TABLE I. DETAILS OF THE SIMULATION.

Parameters	Values
Simulator	Omnet++
Data generation rate	1 packet/sec
Area, M x M	1000 m x 1000 m
Simulation Time	500 sec
Vehicle Radio Range	100 m

Due to the cost and difficulty associated with implementation of VANETs in real world, computer simulations remain as one of the primary technique to investigate networking Characteristics of VANETs [10]. In this regard, it is very important to adopt realistic vehicular mobility models, and design network protocols that are capable of delivering good end-to-end performance in such a highly mobile environment.

To ensure a realistic simulation, a simple mobility model for VANETs, based on Stop Sign Model (SSM) [11] is adopted, which addresses movement patterns of vehicles in an urban scenario. According to our model, initial nodes positions, as well as their destinations, are chosen randomly according to the streets layout within a city map, as shown in Fig. 4.

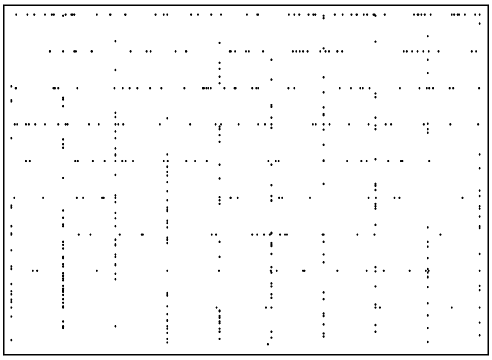

Figure 4. Example of nodes distribution (street layout).

The simulation field map represents a simple grid-shaped urban scenario. Streets are placed creating a non uniform grid since streets are not equidistant. We assume that streets have two lanes, one for each direction, and vehicles are not allowed to overtake each other. Thus, vehicles movement depends on the preceding vehicle movement. Some crossroads has a stop sign; when a vehicle approaches a junction with a stop sign, it must stop at the signal for a waiting period which is proportional to the number of vehicles that are moving near that junction. This way, the more crowded a zone is, the longer it takes to pass through it. Each node follows the shortest path lying on the street layout towards its destination and, once it reaches the destination, nodes start again the procedure choosing randomly another destination along the shortest path. Since nodes speed depends on traffic conditions, only maximum speed limit can be set for each vehicle.

V. SIMULATION RESULTS AND ANALYSIS

In this section, we present our main findings. We discuss results from an extensive performance evaluation of our implementation of our protocol and the benchmark protocol. Both protocols are simulated using the same set-up as described in the above section, varying network size and nodes' speeds.

In the following experiments, we have adopted a number of metrics to evaluate the performance of the protocols. These performance metrics are defined as follows:

- Packet delivery ratio: the percentage of received data packets at the destination.
- End-to-End delay: the average delay a packet takes to travel from the source to the destination.
- Route length: the average number of hops a packet takes to travel from the source to the destination node.
- Control overhead: the total number of control messages exchanged.

A. Network Size

Table 2 shows only the unique parameters for these set of experiments. We fixed the maximum speed at 8 m/sec, and varied the network size from 100 to 1000 nodes. Other settings are the same as in Table 1.

TABLE II. ENVIRONMENT FOR NETWORK SIZE.

Parameters	Values
Max speed	8 m/sec
Network size	100 -1000

To see the impact of network size on the network performance, the source and the destination nodes were placed at points $P_s(0,300)$ and $P_d(200,500)$, respectively, according to the Cartesian coordinate system of the simulation area. For the rest of the nodes the maximum speed limit was set at 8 m/s. The performance of DREAM and SIFT as a function of number of nodes is shown in Figs. 5-8.

As expected, Fig. 5 depicts an increase in the delivery ratio of both schemes for increasing node density. However, DREAM performance decreases when the number of nodes in the network exceeds 500. If network density is too high, DREAM does not perform well because nodes generate a high volume of control overhead that gets the channel overloaded and, thus, many transmission errors occur, resulting in decreasing delivery ratio. SIFT is able to deliver more packets than DREAM in any case under a realistic scenario because SIFT is a spatial-aware protocol while DREAM uses a simple greedy routing technique. Nodes distribution is a key factor of network performance; so that, knowing that nodes are distributed according to streets layout, a routing protocol aware of that layout will be more efficient. On the other hand, DREAM is not aware of road layout and it happens often that there is a low-connectivity zone between source and destination nodes. That is why DREAM achieves low delivery ratio as shown in Fig. 5.

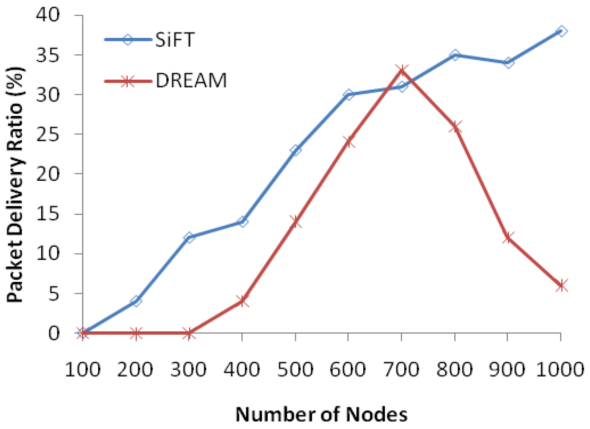

Figure 5. Delivery ratio as a function of number of nodes.

Figure 6 shows the influence of changing the network density on packets delivery delay. DREAM is not capable of delivering any data packet when the number of nodes is lower than 400. In this case, the delay is infinite since no packet is delivered. SIFT experiences a decrease in the delay when density increases; the higher density, the better positioned is

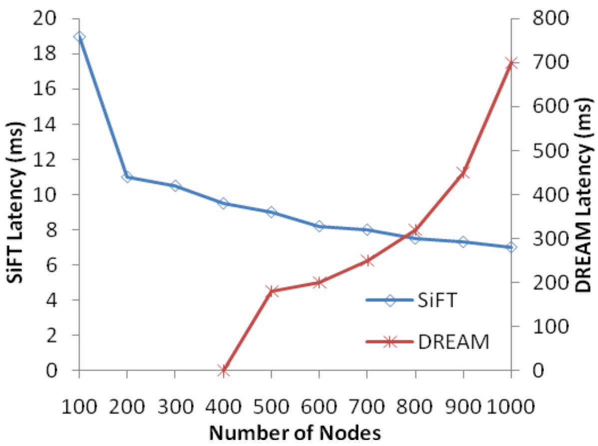

Figure 6. End-to-End delay as a function of number of nodes.

the next hop node with respect to the given routing trajectory, and therefore, the shorter its timeout lasts. With a high network density, DREAM delivery delay is very high due to the location-information dissemination procedure. This process generates a very high volume of control traffic, which overloads the communication channel. In an overloaded network, collisions occur, and packets must be retransmitted incurring a high delay. In general, the average delay in SIFT is always lower than in DREAM because of the aforementioned reason. However, if density is not very high, that is, around 400 nodes, DREAM does not overloads the channel and its routing procedure is faster than SIFT forwarding mechanism, which is always timer-based. So, in low density network with adequate connectivity, DREAM performs better than SIFT regarding delivery delay. Figure 7 shows the average number of hops as a function of number of nodes in the network. In this graph, the number of hops is represented by an infinite value when packets do not reach the destination. In our scenario it occurs with DREAM when the number of nodes in the network is lower than 400. The routing mechanism implemented by SIFT chooses the furthest node from the last forwarder to carry the burden of forwarding. Thus, in SIFT, the number of hops depends on transmission radio range, but not on density. So that, in this protocol, delivery delay decreases when density increases because the more nodes in the network, the better positioned they are regarding the given trajectory, and the shorter the intermediate timeouts are. However, the number of hops in DREAM depends on the density since all the nodes that are within the routing cone forward the given packet; hence, the number of intermediate hops is proportional to the number of total nodes in the network.

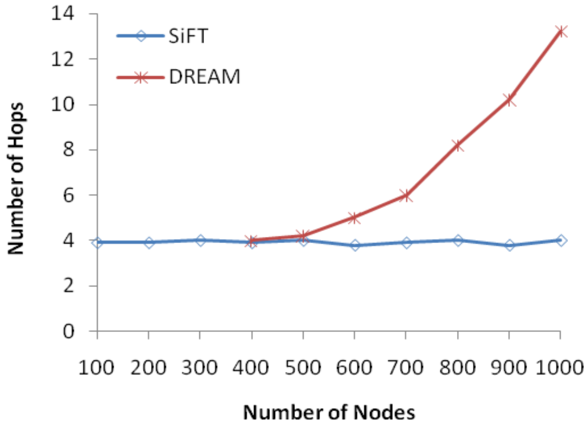

Figure 7. Route length as a function of number of nodes.

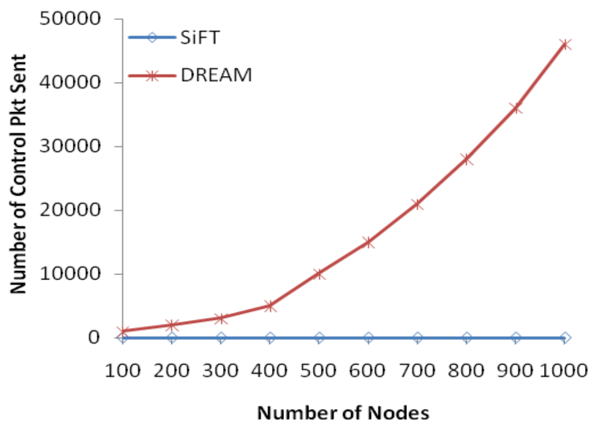

Figure 8. Control overhead as a function of density.

We also investigated the schemes effect on control overhead. Figure 8 shows the traffic overhead with respect to the network size. Nodes in SIFT do not exchange control messages at all. Therefore, SIFT is more adapted to large scale network with a high mobility degree. In DREAM, control traffic overhead is proportional to density, the more nodes deployed in the network, the more control packets are exchanged.

B. Speed

The final experiment set involves the investigation of the impact of nodes speed on both schemes with the average speed is 2-12 m/s and number of nodes fixed at 500 distributed in the simulation area according to the SSN mobility model described above. The source node was placed at point $P_s(0,300)$, and the destination node was fixed at point $P_d(200,500)$. The maximum speed limit of nodes was properly tuned in order to get a speed average from 2 to 12m/s, according to typical speed values for vehicles in an urban scenario.

TABLE III. ENVIRONMENT FOR SPEED.

Parameters	Values
Number of nodes	500
Average speed	2 – 12 m/s

As shown in Figs. 9-12, DREAM experiences a poor performance affected by nodes speed, since its location information dissemination procedure is completely depends on nodes speed. The faster nodes move, the more frequent nodes new position must be disseminated through the network. When speed exceeds a certain value, the channel gets overloaded and transmission errors occur. When nodes speed increases delivery ratio decreases and delivery delay increases, this is because the control overhead increased. However, route length remains constant as this parameter only depends on network density and distances between sources and destination nodes, which are constant in this set of experiments. In general, nodes speed is a parameter that has a negative impact on location table-driven routing protocols for mobile ad hoc networks because the information kept in the location routing table is completely linked to nodes speed.

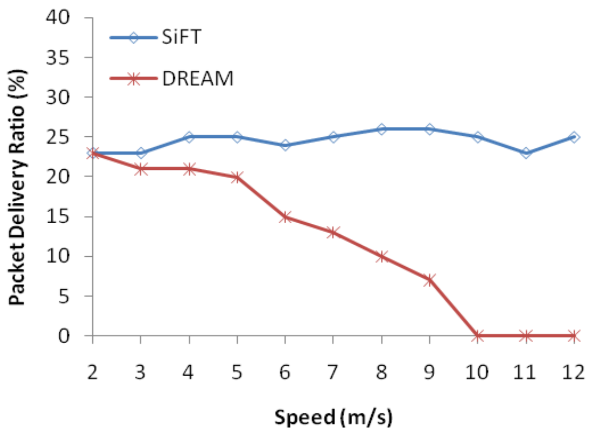

Figure 9. Delivery ratio as a function of speed.

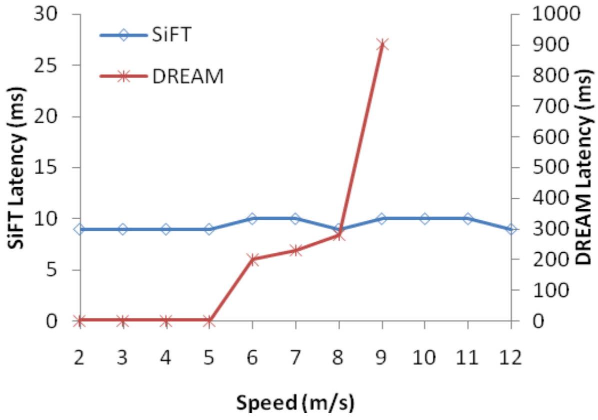

Figure 10. End-to-end delay as a function of speed.

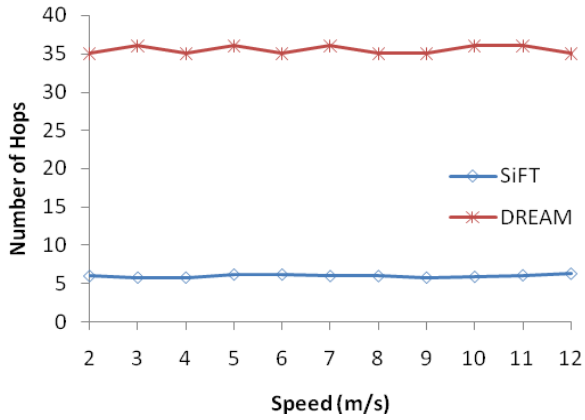

Figure 11. Route length as a function of speed.

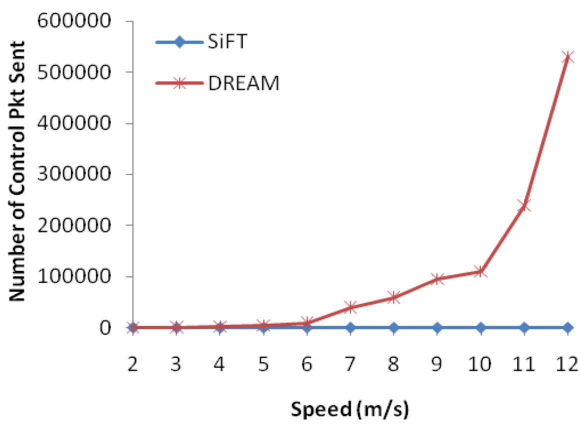

Figure 12. Control overhead as a function of speed.

From this remark, the most important assertion that can be obtained is that SIFT performance remains completely constant in terms of speed, as shown in Figs. 9-12. This is because of the special routing technique implemented by SIFT, which is not location table-driven.

VI. Conclusion

In this paper, we consider SIFT, a trajectory based routing protocol for VANET networks. It is based on broadcast instead of point-to-point transmissions, and it does not require any neighbor information a priori, since forwarding decisions are shifted from the transmitter to the receiver. Upon receiving a data packet, each node makes a decision on whether to forward it or not based merely on node's own position, the transmitter position and the trajectory. All these low-quality information can be piggybacked in the data packet itself.

The performance of the proposed scheme is evaluated through simulation. It is benchmarked against a well-known and comparable position-based routing protocol, namely DREAM. Encouraging results are obtained. Indeed, the proposed scheme reduces the control overhead to zero and achieves a better delivery ratio and shorter route length.

References

[1] Basagni, S., I. Chlamtac, V. Syrotiuk, and B. Woodward. "A Distance Routing Effect Algorithm for Mobility (DREAM)," In Proceedings of ACM/IEEE MOBICOM, 1998.

[2] Ko, Y. and N. Vaidya. "Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms," In Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, 1999.

[3] Briesemeister, L. and G. Hommel. "Integrating simple yet robust protocol layers for wireless ad hoc intervehicle communications," In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS), 2002.

[4] Briesemeister, L., Group membership and communication in highly mobile ad hoc networks, in PhD thesis, School of Electrical Engineering and Computer Science. 2001, Technical University of Berlin.

[5] Niculescu, D. and B. Nath. "Trajectory based forwarding and its applications," In Proceedings of the 9th annual international conference on Mobile computing and networking, 2003.

[6] Lochert, C., H. Hartenstein, H.F.l. J. Tian, D. Herrmann, and M. Mauve. "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments," In Proceedings of IEEE Intelligent Vehicles Symposium (IV2003), 2003.

[7] Wu, H., R. Fujimoto, R. Guensler, and M. Hunter. "MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks," In Proceedings of 1st ACM workshop on vehicular ad hoc networks, 2004.

[8] "OMNET++ Community Site - http://www.omnetpp.org/."

[9] Li, J., J. Jannotti, D.d. Couto, D.R. Karger, and R. Morris. "A scalable location service for geographic ad hoc routing," In Proceedings of ACM Mobicom, 2000.

[10] Wang, W., F. Xie, and M. Chatterjee. "TOPO: Routing in Large Scale Vehicular Networks," In Proceedings of IEEE 66th Vehicular Technology Conference, VTC-2007 Fall, 2007.

[11] Mahajan, A., N. Potnis, K. Gopalan, and A. Wang, "Evaluation of mobility models for vehicular ad-hoc network simulations," In Technical Report TR-051220, Florida State University, 2005.