

The Farther Relay and Oracle for VANET. Preliminary results

Alessandro Amoroso^{*},
Department of Computer
Science
University of Bologna
Mura Anteo Zamboni, 7
40127 Bologna, Italy
amoroso@cs.unibo.it

Marco Ciaschini
Department of Computer
Science
University of Bologna
Mura Anteo Zamboni, 7
40127 Bologna, Italy
miciaschi@cs.unibo.it

Marco Rocchetti
Department of Computer
Science
University of Bologna
Mura Anteo Zamboni, 7
40127 Bologna, Italy
rocchetti@cs.unibo.it

ABSTRACT

¹ We present a novel protocol for fast multi-hop message propagation in the scenario of ad hoc vehicular networks (VANET).

Our approach has been designed to gain optimal performance in scenarios that are very likely, but not common in literature. FROV faces asymmetric communications and varying transmission ranges. In this scenario it is able to broadcast any message with the minimal number of hops. Moreover, our proposal is scalable with respect to the number of participating vehicles, and tolerates vehicles that leave or join the platoon.

At the current state of development, our protocol is optimal in the case of unidimensional roads and we are studying its extension to a web of urban roads.

This paper presents the preliminary results of simulations carried out to verify the feasibility of our proposal.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless Communication

General Terms

Design, Performance, Algorithms

Keywords

VANET, ad-hoc networks, dissemination strategies, simulations

1. INTRODUCTION

Modern vehicles are equipped with several sensors, on board computers, and communication devices, that are increasingly integrated each other. It is easy to foresee that, in a

^{*}Correspondent author

¹Copyright 2008 ICST 978-963-9799-36-3

close future, the great majority of vehicles traveling on the roads will have such a configuration. Nowadays, taking advantage of those equipment to increase driving safety is an emerging trend, both in academia and industry.

A likely scenario is that all the roads will not be covered by a communication infrastructure, to connect the traveling vehicles themselves and with the Internet. It is highly probable that the vehicles will deliver the communication infrastructure by themselves.

One of the main issues of such vehicular networks (VANET) is the time to delivery a broadcast. This characteristic is strictly related to both the number of relays of the messages, *hops*, and to the congestion of the network [26, 17, 24]. The lower the number of hops the fastest the broadcast. The higher the congestion, the slower the broadcast. Depending on the number of relays, even a single message could lead to a network congestion, *e.g.* when the receivers attempt to relay the message at the same time. The choose of a single “good” relay, and its *timing*, is crucial both to avoid congestion, and to reduce the overall number of hops.

We consider that the next generation of vehicles will be commonly equipped with a wireless communication apparatus, according to the IEEE 802.11 standards [12]. Following the increasing integration between the equipment of a vehicle, our communication protocol takes advantage of the GPS receiver that equips the vehicles. Both of those apparatuses are becoming standard equipment in the car industries.

We call our protocol *Farther Relay and Oracle for VANET* (FROV), where the oracle monitors the platoon of vehicles, and farther relay is the broadcast scheme, as described below. FROV adopts a *multi-hop* approach to broadcast: any message is re-transmitted by several relays until it reaches the end of the platoon. The main idea of FROV is that the relay of a message is the receiver whose re-transmission spans further than any other. Based on this idea, FROV generates *run-time* a sequence of relays for any broadcast message. To accomplish this task, FROV estimates, both the position and the transmission range, in the direction of the broadcast, for all the receivers of a message.

We relaxed some of the most common assumptions made in several previous proposals. FROV does not require nei-

ther a constant, and known *a priori*, range of transmission, nor vehicles travelling at the same distance each other. The communications between vehicles in the platoon do not need to be *symmetrical*, *i.e.* if vehicle *A* receives a message from vehicle *B*, the opposite is not necessary. Furthermore, the system remains effective even if the transmission ranges of each vehicle shrink or expand, independently each other. Our system is also *adaptive* with respect to changes in vehicles distances, their relative positions, and speed. FROV is completely distributed, the information are collected and managed locally, therefore vehicles can *leave* or *join* the platoon without affecting the effectiveness of the communications. Moreover, the locality of the information allows for a good scalability with respect to the number of vehicles participating to FROV.

The remainder of this paper is organized as follows. Next section briefly describes, and compares, some related works. Sections 3 and 4 discuss the system model and the main characteristics of FROV respectively. The subsequent section presents the simulation scenario and the first results of our implementation. Section 6 concludes this work.

2. RELATED WORK

The IEEE 802.11 DCF protocol is widely adopted as standard for VANETs. The protocol does not offer any kind of broadcast communication. The usual approach is to overcome this limitation is a *multi-hop flooding*. The receivers of a message relay it upon reception. An efficient broadcast should limit both the number of relay and the number of hops. In ideal circumstances, the optimal relays are the vehicles in the Minimum Connected Denominating Set (MCDS) [26]. A MCSD could be obtained recursively, under the unrealistic assumption that the vehicles in the platoon know, in real time, their topology and the transmission ranges of each one of them.

Several solutions in VANET circumvented the above requirement of global real time knowledge, by choosing as relay the farther vehicle from the sender [19, 20, 23]. This heuristic approach could be at least ineffective in the case of heterogenous, and varying, ranges of transmission. In the worst case it could lead to a disruption of the chain of relays. Other authors present solutions based on clustering and cross-layering [13, 3, 7].

The analysis of traffic in urban areas extends the problem of communication in a one dimensional road, such as highways [10, 26], to a web of streets flooded by vehicular traffic [8, 16, 5, 11, 14]. Some authors pointed out the importance of reliability in broadcast VANET [2].

In literature several scenarios for VANET communication have been studied. The most common application are drive safety and emergency, among the others [21, 22, 25, 9]. Another, completely different, scenario is the infotainment [15, 18, 1]. Both of those scenarios have the common requirement of fast broadcast.

Some proposals consider different range of transmission for each vehicle, in particular [8] considers two different antennas, one for *backward* and the other for *forward* communications with respect to the travelling direction of the vehicle.

Among the above mentioned solutions, we focussed on the Fast Multi-Broadcast Protocol (FMBP) [18, 19, 20], that inspired our work, and that is the one that closely resemble our proposal. We consider two main issues to compare the protocols: the *oracle* and the *relay chooser*. The *oracle* estimates both the actual transmission ranges and position of each vehicle, while the *relay chooser* represents the routing criterion. FMBP makes the implicit assumption that the communications are symmetrical; therefore, the oracle could underestimate the transmission ranges. FROV does not have this limitation. FMBP picks as relay the known receiver that is farther with respect to the sender. In case of heterogenous transmission ranges this could be the worst choice, and, generally, increases the number of hops. On the other end, FROV picks as relay the receiver whose retransmission spans farther than each other, thus reducing the number of hops.

3. SYSTEM MODEL AND ASSUMPTIONS

It is likely to suppose that, in a platoon of vehicles, the transmission range could be significantly different car by car. Those differences could depend on the physical characteristics of the radio equipment. In absence of collisions, any sent message is received by all the vehicles within the range of transmission. More generally, we suppose that when a message is received by a vehicle, then all the vehicles between the sender and that receiver have also received the message.

The transmission range of a car can vary while travelling, due to changes of the environmental conditions. Those changes include both atmospheric and topological surroundings. It is well known that humidity of the air, *e.g.* fog, rain, and snow, highly influences the range of transmission of radio waves. Moreover, topological conditions, such as tunnels, sharp curves, surrounding trees and buildings, could further influence the transmission ranges.

Due to the aforementioned motivations, the transmission ranges of a vehicle could be *asymmetrical* with respect to its *forward* and *backward* directions. As an example, consider a vehicle while entering or exiting a tunnel. Inside a tunnel the transmission range is reduced with respect to open air; therefore, the forward and backward transmission ranges differ each other. We measure the transmission ranges with respect to the distance between the sender and the farther receiver, because this is the most accurate measure that we can perform.

Figure 1 shows an example of asymmetric communication. The range of transmission of each vehicle is represented by a dotted line, that ends aligned with the last vehicle that receives the transmission. Note that those transmission ranges are asymmetrical. When the vehicle *E* broadcasts a message, the vehicles *C* and *D* receive that message. Suppose that, as a response to that message, vehicle *C* broadcasts a second message that does not reaches vehicle *E*. Vehicle *E* could receive the response message from *C* only if vehicle *D* relays it.

In this paper, we made the further assumption that the vehicles are travelling on a road that is both *one-way* and *unidimensional*. We made this simplifying assumption, quite common in literature, at this stage of development of our

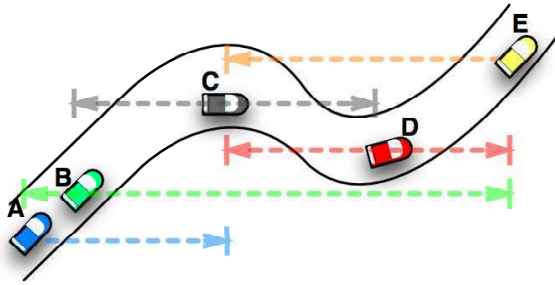


Figure 1: Complex scenario

protocol, to test its effectiveness in a such controlled scenario.

4. THE FARTHER RELAY AND ORACLE FOR VANET

Consider the scenario depicted in Figure 1, when vehicle *A* broadcasts a message, which one between vehicles *B* and *C* should relay the message? According to FMBP, vehicle *C* is the farther one with respect to the sender, therefore it will relay the message. To reach vehicle *E*, the message needs a further relay by vehicle *D*. Instead, according to FROV, the best vehicle to relay the message from *A* is the one whose relay spans farther with respect to all the receivers. Therefore, in the depicted scenario, the best relay will be vehicle *B*; resulting in less hops with respect to the previous solution.

It is worthwhile to point out that all the information managed by FROV are *local*. Each vehicle constructs a view of its current neighborhood, and assesses its actual transmission ranges, exclusively by means of the messages that it received lately. The relay of a message is based on those local information, and does not require any kind of global knowledge. Each vehicle does not know anything about the platoon, except for the vehicles that directly exchanged messages with it.

Our proposal is made by two functional blocks: the *oracle* and the *relay chooser*. The oracle assesses both the car positions and their transmission ranges. The chosen relay is a “good” relay among the receivers of a message. The following sections discuss their design and the main characteristic of FROV.

4.1 The oracle in FROV

The oracle runs parallel to the communications. Each vehicle participating to FROV asynchronously broadcasts an oracle message, at a random time in a predefined period, and receives the oracle messages sent by its neighbor. Based on those oracle messages, any vehicle maintains the following local lists. Each list stores ID, position and announced transmission range for each one of its entry:

- *Listened*: the senders of the oracle messages received

by the vehicle. This list is included into the oracle messages sent by the vehicle;

- *Reached*: known receivers of the oracle messages sent by the vehicle;
- *Aware*: known further receivers of the oracle messages sent by vehicles that are also in the *Listened* list.

Those lists are empty at the beginning of the protocol, are populated during the startup phase, and are maintained at run-time. For optimization reasons, all the above lists are doubled, considering the *forward* and *backward* transmissions. For the sake of simplification, we do not discuss this aspect in detail; instead, we consider only a single direction, while it is symmetrical with respect to the other direction.

When a participant vehicle *R* receives an oracle message from vehicle *S*, it updates its *Listened* list, and searches for its own ID in the list contained by the message. A successful search means that *S* has previously received an oracle message from *R*, therefore *R* updates its *Reached* list. In this case, *R* might also update its current transmission range in the direction of *S*, by comparing its position with respect to *S*.

The *Aware* list takes into account the necessity of relaying oracle data. Consider, again, the vehicles *C*, *D*, and *E* in the Figure 1. Suppose that vehicle *E* broadcasts an oracle message. Both vehicle *C* and *D* receive that message, and both update their *Listened* list. The next oracle message sent by vehicle *C* will contain, among the others, the information that *C* listened messages from *E*. Receiving this message from *C*, the vehicle *D* become aware that even *C* received the oracle message from *E*. While *C* is farther than *D* with respect to *E*, vehicle *D* stores *C*'s data into its *Aware* list. The next oracle message sent by vehicle *D* will contain a merge between its *Listened* and *aware* lists. Receiving this oracle message from *D*, vehicle *E* knows that its transmission range spans until the position of *C*, even if it does not receive direct communications from *C*.

Summarizing, the oracle messages contain data related to the sender: ID, position, known forward and backward transmission range, and the two *Listened* and *Aware* lists.

4.1.1 Frequency of the oracle messages

The positions and transmission ranges of the vehicles vary continuously. The *accuracy* of the oracle depends on the rapidity of the changes with respect to the frequency of the oracle messages. In general, the higher the frequency, the better the accuracy. On the other end, an high frequency of oracle messages leads to an high overhead in terms of bandwidth occupation. Since the accuracy of the oracle is its adherence to the reality, the frequency of the oracle messages is also related to the rates of modifications in the observed platoon of vehicle. The lesser the changeability, the lesser could be the frequency to preserve the same accuracy level.

The lists maintained by the vehicles contain data that are not contemporaneous each other, and retaining “old” data leads to less accuracy. To minimize the risk of using outdated data, FROV associates an integer value, called *TTL*, at the entries in the *Listened* and *Reached* lists.

The value of *TTL* is set to a predefined value when an entry is inserted into the lists. Those *TTL* values are changed with respect to the sending and receiving of oracle messages. Any send of an oracle messages causes the decrement by one of all the *TTL* values, while any receive of an oracle messages reset the *TTL* of the corresponding entry to its initial value. When a *TTL* becomes zero, the entry is deleted from the list. Since any vehicle sends its oracle messages randomly during a predefined period, the entries of the *Listened* and *Reached* lists are deleted when they are old enough. The *Aware* list is populated at any receive, and it is emptied at each send of oracle messages; therefore it does not have any *TTL*.

4.2 The relay chooser in FROV

The communications between the vehicles participating to FROV depend on the running application. FROV provides the underlying routing mechanism to any application running into the VANET. Application messages are exchanged in parallel with respect to the oracle messages and their frequency and size are under the exclusive control of the running application.

For the sake of simplicity, we consider the most general scheme of communication for the running application: any message is a broadcast to all the vehicles. This scheme could be very likely for applications such as drive security or intra-vehicular gaming.

Before a send of an application message, a vehicle sorts its *Reached* list, with respect to each receivers distance added to its transmission range, and picks the firsts of the list as *designated relays*. Needless to say, the sender sorts in both forward and backward direction. When sending the application message, the vehicle adds to it the list of designated relays.

The sorted list of designated relays faces the possibility that the first of them, *i.e.* the *best* relay, could not receive the application message, resulting in a broken broadcast. This situation could happen mainly for two possible causes. The first cause could be that the *Reached* list of the sender is outdated, therefore the best relay does not belong any longer to the set of receivers. The second cause could be that the application message had a collision, in the area of the best relay, with some other message, either an oracle or an application message.

Designating few possible relays, instead of all the receivers, avoids possible “broadcast storm problem”. In addition, it avoids troubles in case the relay does not reaches some of the receivers of a message. Those receivers, possibly the closer to the sender, could then start a pointless retransmission of the message, wasting both bandwidth and computational resources.

The receivers assign a timeout to any entry of the designated relays list. A relay compute its timeout by multiplying its index in the list by the length of a transmission slot. The family of 802.11 protocols usually set that slot to 20 or 50 milliseconds, and allows for $CWMin = 0$. We consider that the first designated relay has index zero. After the reception of an application message, each designed relay starts a timer

and waits until its timeout. At the end of its waiting period the designated relay sends the message. If a designated relay receives a duplicate of the message while waiting, it aborts the procedure, because someone else took care of the relay. If the first of the list is the actual relay, it immediately sends the message without any waiting.

The actual relay applies the same technique, shown for the sender: it sorts the list of next designated relays. In addition, taking care of the direction of propagation of the message. In the best case, a broadcast reaches all the vehicles by means of the minimal number of hops, and without additional waiting at each hop.

Any application message has a unique identifier. When a vehicle receives a message, it checks if had previously received the same message. At the first reception of a message, a vehicle stores the unique identifier of the message in its local list. When it receives a message more than once, a vehicle ignores the message. This behavior leads to a further restriction to the broadcast storm problem mentioned above.

A possible weakness of FROV broadcasting comes from outdated oracle data: the best relay could not be optimal anymore. However, if the best relay receives the message, it does the relay; possibly gaining a sub-optimal hop. If the best relay does not receive the message, one of the following in the list of designed relays will retransmit the message. In this scenario FROV does not suffer broadcast disruption, but the routing of this specific message becomes sub-optimal.

4.3 Main characteristics of FROV

In this section we briefly discuss some of the main characteristics of FROV: optimal number of hops, scalability, and adaptivity to changes in the vehicle platoon.

4.3.1 Optimal number of hops

Considering an almost linear road, we can assume that any message sent by a vehicle *S* and received by vehicle *R* is also received by all the vehicles between *S* and *R*. Under this assumption, FROV requires the minimum number of hops to broadcast a message to all the participating vehicles.

Doubtless, this is true for the first hop, that spans as far as possible by definition. Among the vehicles that received the messages, FROV selects the relay whose messages span farther. Hence, each step maximizes the distance from the original sender and the farther receiver. Therefore, the overall number of hops is minimal.

The quality of FROV broadcast is strictly dependent on the accuracy of the oracle, *i.e.* closer to the reality is the oracle, the closer to optimal are the number of hops.

4.3.2 Scalability

FROV does not require any global knowledge, it solely manages local info; therefore, increasing the number of vehicles does not affect neither the performance nor the required resources.

The limiting factor for both performance and resources is the *density* of proximity vehicles. That is the number of

vehicles that could generate both transmissions and entries in the local lists. An increasing number of proximal vehicles causes more and longer oracle messages, then increases the bandwidth occupation and the probability of collisions between messages. Moreover, the required memory to store the local lists is directly proportional to the number of proximal vehicles.

Nevertheless, the physical dimensions of vehicles, and the minimum distance between them, sets up a strict upper limit to the number of vehicle per length unit.

4.3.3 Adaptivity

The mechanism to manage messages obsolescence, shown in section 4.1.1, could be considered as a kind of *fault detector*. More generally, the oracle provides a mechanism to manage the group membership of the vehicles participant to FROV, *i.e.* vehicles could *join* or *leave* a running FROV group.

Any vehicle participating to FROV broadcasts an oracle messages in a predefined period τ , *i.e.* it broadcast its data randomly picking a moment inside τ . This timing implies that at any participant announces its position to the neighborhood at most 2τ later than the last announcement. On average, any vehicle announces its position every τ .

A vehicle that leaves FROV does not announce its position anymore, and its absence starts to be recorded at most after 2τ of its leaving. In few more τ , depending on the degree of obsolescence of the oracle informations, the vehicle will disappear from FROV. On the contrary, a new vehicle joining FROV announces its presence by means of its first broadcast. Therefore, a joining vehicle is immediately recorded by its neighborhood.

The leaving of a vehicle does not affects the performance of FROV, unless the vehicle is the first designated relay of a message currently broadcasted. In this case the second relay of that message has to wait its turn to send.

5. EXPERIMENTAL RESULTS

We aimed our preliminary simulations to a comparison between FROV and FMBP; we show the first results in this section. Since FROV belongs to the same family of FMBP, we searched experimental evidence that our ideas are effective.

To obtain meaning results, we used the same simulator and the original code of FMBP, *ns 2*. We simulated a platoon of 100 vehicles traveling on a road $2Km$ long. The initial distance between those vehicles is random, ranging from 10 to 50 meters. The vehicles travel at constant speed. Both forward and backward ranges for each car are randomly picked into the interval $[75, 300]meters$, independently each other. Those are quite common parameters, *e.g.* [4, 3].

We performed several experiments varying the number of vehicles generating application messages; they were: 1, 10, 20, 30, 40, and 50. Any vehicle generates a new application message after a random interval ranging from 0.1 to 0.6 seconds. We choose this massive rate of generation for the application messages, both to stress the system and to represent the case of very interactive applications. We be-

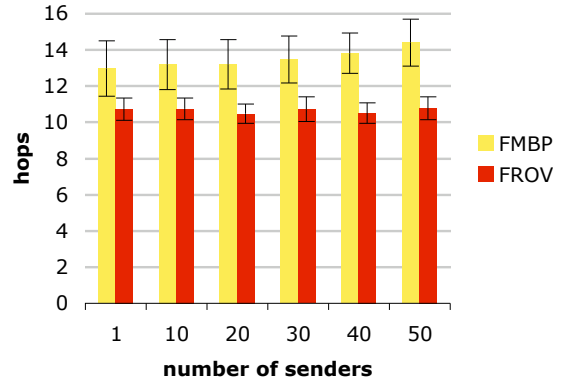


Figure 2: Number of hops

lieve that this could be an extreme case, but still likely, for applications such as interactive games.

In our measurement we considered two fixed vehicles, the first one, *A*, close to the beginning of the road, and the second vehicle, *B*, approaching the end of the road. A message exchanged between *A* and *B* should travel almost all the road to reach its destination.

Figure 2 shows the average number of hops of any message exchanged between the vehicles *A* and *B*. The *x* axis reports the number of vehicles generating application messages, and length of the bars represents the number of hops. We performed several experiments, differentiated each other by the random seed. The segments at the top of the bars represent the interval of confidence of 95%. As expected, the number of hops is almost constant regardless of the number of vehicles generating application messages. The FROV requires, approximately, a good 20% less hops than FMBP. This result shows the effectiveness of the relaying policy adopted by FROV.

Neither FROV nor FMBP are *reliable*, *i.e.* they might drop some message. This means that some sent messages never reach the other vehicle under measure. This phenomenon could be explained by the random presence of highly congested areas, that covers all the possible relays of a dropped message. The high bandwidth contention could depend on both oracle and application messages. This is a scenario under our investigation, that depends on both the high frequency of application messages, and the high density of vehicles.

Figure 3 shows the average number of dropped messages exchanged between the vehicles *A* and *B*. Note that, with respect to this measure, the vehicles generating application messages are competing with *A* and *B* for the bandwidth; therefore they generate a kind of “noise” messages. As expected, the number of lost messages increases with respect to the number of vehicle generating messages, *i.e.* with re-

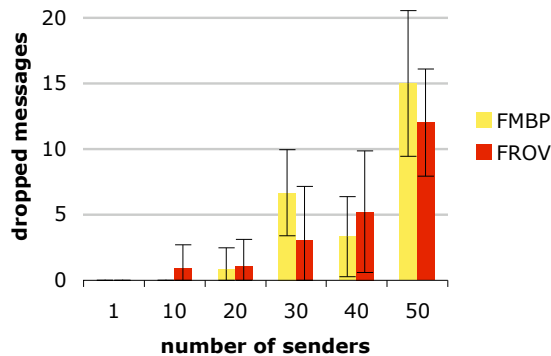


Figure 3: Percentage of dropped messages

spect to the number of sent messages. Note that the intervals of confidence are very broad with respect to their values, *i.e.* there is a large variance in the results of different experiments. Nevertheless, it seems that FROV drops significantly less messages than FMBP in almost all the circumstances. In both cases, the “density” of exchanged application messages directly affects the number of dropped messages. Moreover, since the choice of the relay made by FMBP is based of a random waiting inside a time slot, it could happen that this leads to a greater number of application messages collision with respect to FROV.

The Figure 4 shows the average propagation time that any message, exchanged between vehicles *A* and *B*, takes to reach the other vehicle with respect to the sender. In this case, FROV takes advantage of both the reduced number of hops and the deterministic choice of a relay, without random waits. The number of vehicles generating application messages does not effects the latency of FROV, while it strongly influences the latency of FMBP. As shown in [6], the propagation time shown by FROV is well suited even in the case of first person interactive games.

6. CONCLUSION AND FUTURE WORKS

Our protocol has been designed to obtain optimal performances in scenarios that are very likely, but not common in literature. The current implementation of FROV confirmed in our preliminary experiments its ability to deal with both asymmetric communications between vehicles, and varying transmission ranges. Even in this complex scenario it is able to pick the optimal number of relays. Moreover, our proposal is scalable with respect to the number of vehicles, and can sustain high density of application messages. FROV can, almost effortlessly, accommodate for vehicles joining or leaving the platoon.

We show the results of preliminary experiments that verified the efficacy of our protocol. The assumption of a one-way unidimensional road was related to a “proof of concept”. We are planning further and more intense simulations, involving

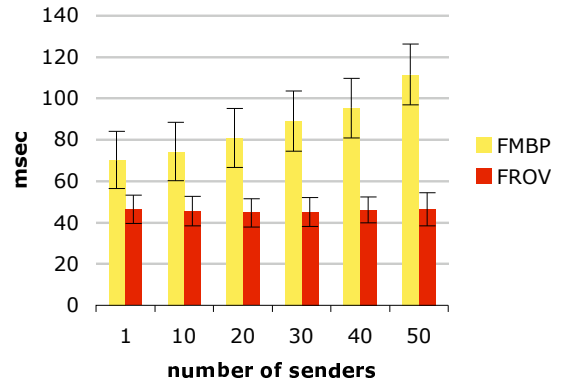


Figure 4: Propagation time

vehicle speed variations, and measures of the accuracy of the oracle. We are also developing a new implementation, tailored for another simulator, in order to double-check our preliminary and comforting results.

We are investigating the possibility to extend our design to a network of roads, such as an urban scenario. We would also investigate the use of synchronous oracle messages to reduce as much as possible the probability of collisions among them.

7. REFERENCES

- [1] F. Bai, T. Elbatt, G. Hollan, H. Krishnan, and V. Sadekar. Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective. In *In Proceedings of IEEE Workshop on Automotive Networking and Applications (AutoNet)*, 2006.
- [2] N. Balon and J. Guo. Increasing broadcast reliability in vehicular ad hoc networks. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 104–105, New York, NY, USA, 2006. ACM.
- [3] L. Bononi and M. DiFelice. A cross layered mac and clustering scheme for efficient broadcast in vanets. In *IEEE intl. conf. on Mobile Adhoc and Sensor Systems*, 2007.
- [4] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 85–97, 1998.
- [5] D. R. Choffnes and F. E. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 69–78, New York, NY, USA, 2005. ACM.
- [6] M. Claypool and K. Claypool. Latency and player actions in online games. *Commun. ACM*, 49(11):40–45, November 2006.

- [7] M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross layering in mobile ad hoc network design. *IEEE Computer*, 37:48–51, 2004.
- [8] L. Da, H. Huang, X. Li, and F. Tang. A distance-based directional broadcast protocol for urban vehicular ad hoc network. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 1520–1523, Piscataway, 2007. IEEE.
- [9] S. Das, A. Nandan, and G. Pau. Spawn: A swarming protocol for vehicular ad-hoc wireless networks. In *In Proceedings of VANET*, pages 93–94, 2004.
- [10] H. Füßler, M. Torrent-Moreno, R. Krüger, M. Transier, H. Hartenstein, and W. Effelsberg. Studying vehicle movements on highways and their impact on ad-hoc connectivity. Technical report, ACM SIGMOBILE Mobile Computing and Communications Review, 2005.
- [11] C. Gorgorin, V. Gradinescu, R. Diaconescu, V. Cristea, and L. Iftode. An integrated vehicular and networking simulator for vehicular ad-hoc networks. In *in ESM'06: Proceedings of the 20th European Simulation and Modelling Conference*, 2006.
- [12] IEEE. *802.11 WG, IEEE Std 802.11, 1999 Ed, Part II: Wireless LAN MAC and PHY layer specs*.
- [13] T. D. C. Little and A. Agarwal. An information propagation scheme for vanets. In *Intelligent Transportation Systems*, pages 155–160, September 2006.
- [14] R. Mangharam, D. S. Weller, D. D. Stancil, R. Rajkumar, and J. S. Parikh. Groovesim: A topology-accurate simulator for geographic routing in vehicular networks. In *in Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 59–68. ACM, 2005.
- [15] M. Mauve, S. Fischer, and J. Widmer. A generic proxy system for networked computer games. In *1st workshop on Network and system support for games*, pages 25–28. ACM Press, 2002.
- [16] V. Naumov, R. Baumann, and T. Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 108–119, New York, NY, USA, 2006. ACM.
- [17] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM.
- [18] C. E. Palazzi, S. Ferretti, S. Cacciaguerra, and M. Rocchetti. Interactivity-loss avoidance in event delivery synchronization for mirrored game architectures. *IEEE Transactions on Multimedia*, 8(4):874–879, August 2004.
- [19] C. E. Palazzi, S. Ferretti, M. Rocchetti, G. Pau, and M. Gerla. How do you quickly choreograph inter-vehicular communications? a fast vehicle-to-vehicle multi-hop broadcast algorithm, explained. In *3rd IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME'07)*. IEEE, Jan. 2007.
- [20] C. E. Palazzi, M. Rocchetti, S. Ferretti, G. Pau, and M. Gerla. Online games on wheels: Fast game event delivery in vehicular ad-hoc networks. In *3rd International Workshop on Vehicle-to-Vehicle Communications (V2VCOM 2007)*. IEEE, June 2007.
- [21] G. Resta, P. Santi, and J. Simon. Analysis of multi-hop emergency message propagation in vehicular ad hoc networks. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 140–149, New York, NY, USA, 2007. ACM.
- [22] C. Robinson, L. Caminiti, D. Caveney, and K. Laberteaux. Efficient coordination and transmission of data for cooperative vehicular safety applications. In *in the Proc. of the 3rd international*, pages 10–19, 2006.
- [23] M. Torrent-Moreno, D. Jiang, and H. Hartenstein. Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks. In *VANET 2004*, pages 10–18, 2004.
- [24] X. Yang, J. Liu, F. Zhao, and N. Vaidya. A vehicle-to-vehicle communication protocol for cooperative collision warning. In *MobiQuitous'04*, 2004.
- [25] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty. Performance evaluation of safety applications over dsrc vehicular ad hoc networks. In *Proceedings of the 1st ACM international*, pages 1–9, 2004.
- [26] A. Zanella, G. Pierobon, and S. Merlin. On the limiting performance of broadcast algorithms over unidimensional ad-hoc radio networks. In *WMPC04*, sept. 2004.