

How Do You Quickly Choreograph Inter-Vehicular Communications? A Fast Vehicle-to-Vehicle Multi-Hop Broadcast Algorithm, Explained

Claudio E. Palazzi^(1,2), Stefano Ferretti⁽¹⁾, Marco Rocchetti⁽¹⁾, Giovanni Pau⁽²⁾, Mario Gerla⁽²⁾

¹Dipartimento di Scienze dell'Informazione,
Università di Bologna,
Bologna, Italy
{sferrett | roccetti}@cs.unibo.it

²Computer Science Department,
University of California, Los Angeles,
Los Angeles, CA, USA
{cpalazzi | gpau | gerla}@cs.ucla.edu

Abstract— As the technology available on cars is increasing, a wide range of applications, from safety to entertainment, are becoming factually accessible to passengers. Many of these applications involves a one-to-many transmission model where a single car broadcasts a message that has to be forwarded, even with multiple hops, in a very short time to all the other cars located within a range of few kilometers from the source. Since the high mobility and density of a car network scenario, specific solutions need to be devised to choreograph a fast-delivery multi-hop broadcast. To this aim, we developed a practical and efficient technique that allows cars to estimate their communication range with the help of a very limited message exchange and exploit this information to reduce the number of transmissions, as well as the hops to be traversed, and hence the time, required by a broadcasted message to reach all the cars following the sender within a certain distance.

Keywords- *Inter-Vehicular Communication, Car-to-Car Safety, Car-to-Car Entertainment, Fast-delivery Broadcast.*

I. INTRODUCTION

The computing technology that supported the moon landing almost 40 years ago was inferior to that normally available on today's cars. Indeed, sensors, on-board computers, and communication devices are now ready to be used to increase driving safety by avoiding accidents or, at least, minimizing their harmful effects. Moreover, cars are becoming computing platforms that can access any data at any time, thus evolving from a simple transportation means to an office on the move, as well as an information provider, and an entertainment center.

Yet, it is not guaranteed that all roads will be covered by a communication infrastructure able to connect cars among themselves and with the rest of the Internet. For this reason, a lot of research has been devoted to the study of inter-vehicular communication (IVC) and, in particular, to devise smart position-aware broadcast protocols to propagate urgent information.

Indeed, the ability of ensuring fast message broadcast is fundamental for many appealing applications. The most emergent example is represented by safe-driving where critical information such as hazards and alarms have to be propagated

with the lowest possible delay from one car to the following ones.

Even networked multimedia applications (e.g., multiparty chat, online games, digital billboards [1, 2]) could benefit by efficient broadcasting when they engage passengers of cars traveling in proximity of each other. For instance, many online games exist which require each message to be delivered within a very strict time threshold [3]. An efficient multi-hop broadcast scheme could hence quickly deliver game messages to a higher number of cars, also dispersed on a wider portion of road.

Actually, one can imagine an infinite number of applications; think, for instance, to the combination of the aforementioned two exemplars where entertainment could be used to stimulate safe-driving. In particular, it is well known in psychology that positive reinforcements are more effective than negative ones [11]. Therefore, instead of just relying on fines to discourage driving infractions, we could encourage safe-driving by implementing games among car drivers, where the one who is more scrupulously following road rules in a certain randomly selected moment will be rewarded with special points or other features of the online game played by her/his passengers. Desirably, the winner in this safe-driving contest could be rewarded with points that can be accumulated to win downloadable digital songs and videos, or special discounts at gas stations and lodging. Such applications would require an instantaneous image of driving characteristics be taken from a group of cars. This could be done by having a random car sending a broadcast message to all cars in a certain area so as to make them record relevant data for the safe-driving competition (i.e., speed, safe distance from other cars and obstacles, smooth driving). If this message reached all the cars within a small threshold of time, say 100ms, one could accept gathered data as simultaneously taken, and thus elect a winner.

To support the aforementioned and many more applications, we have designed a fast multi-hop broadcast protocol that reduces the time required by a message to propagate from the source to the farthest car in a certain (geographical) strip-shaped area-of-interest. Our broadcast

Partial financial support for this work is provided by: the Italian MIUR (under the ICTP/E-Grid, MOMA, DAMASCO, Unibo initiatives); the National Science Foundation (through grants CNS-0435515/ANI-0221528); and STMicroelectronics (under the UC-CoRe Grant MICRO 05-06).

protocol is based on a distributed mechanism for the estimation of the communication range of mobile nodes (i.e., vehicles). The estimated communication ranges are obtained by exchanging a $O(1)$ number of messages among the n cars in a certain area-of-interest, and then used to reduce the amount of hops needed to deliver the message.

While other research proposals exist which (unrealistically) assume that the transmission range is a known constant factor, our mechanism, instead, dynamically computes it. This allows our protocol to cope with a realistic scenario where cars' actual transmission ranges are subject to continuous changes due to physical obstacles, vehicle density, speed, etc. By virtue of this, we claim that our scheme represents a real advancement of the state of the art for inter-vehicular broadcast communications.

The remainder of the paper is organized as follows. Section II surveys broadcasting schemes for IVC. The considered system model is discussed in Section III. Section IV presents our scheme, while a practical example is reported in Section V. Section VI provides a simulative assessment, while Section VII discusses some preliminary results we obtained. Finally, Section VIII concludes this work.

II. RELATED WORK

The problem we address regards fast broadcasting of a message to all cars in a given strip-shaped area-of-interest. Indeed, for many vehicular applications, even few tens of milliseconds of delay represent a huge waste of time that may disrupt their performances. This represents a critical problem since the unreliable behavior of these applications may either endanger human lives, or large financial investments, or both [7, 9].

Experts report that principal reasons behind a slow broadcast delivery are due to a non-optimal number of hops experienced by a message to cover all the involved cars and, more in general, to an excessive number of vehicles that try to simultaneously forward the message [6, 8, 9, 10]. To tackle this problem a theoretically optimal broadcast algorithm has been recently proposed which propagates messages to cars making use of the notion of Minimum Connected Dominating Set [12]. This leads to great practical difficulties in the implementation of such algorithm as it would require a complete and continuously updated knowledge of the network topology. For instance, in an attempt to implement this algorithm with n cars, its authors have developed a scheme employing as many as $O(n \log n)$ control messages [13]. It goes without saying that this is not a scalable solution.

Addressing the fast-delivery broadcast problem from a more practical standpoint, various 802.11-based solutions have been proposed. For example, [9] proposes a backoff mechanism that reduces the frequency of message retransmissions when congestion is causing collisions. In [8], instead, as soon as a car receives a broadcast message from a following vehicle along a strip, it refrains from forwarding it as the reception of this message is a clear confirmation that subsequent cars have already received it. Unfortunately, both these two schemes do not consider a very important factor in determining the final propagation delay of a message: the

number of hops a broadcasted message traverses before covering its whole area-of-interest.

In [10], hops' minimization is achieved by individuating the farthest car within the source's backward transmission range, which has to forward the message. To this aim, jamming signals are emitted by each car with a duration that is directly proportional to the distance between the considered car and the message's source. The car with the longest jamming signal is clearly the farthest car from the source. Even if this guarantees a minimum number of hops to cover the whole area-of-interest, the time wasted to determine the next forwarder through jamming signals could make this scheme not suitable for a tight time delay scenario as the one we are considering.

A final scheme trying to statistically achieve a minimum number of hops when propagating a broadcasted message is discussed in [6]. In particular, different contention windows are here assigned to each car. The contention window represents the maximum number of time slots a car waits before taking upon itself the task of propagating the broadcasted message: each car randomly select a waiting time within its contention window. In [6], the authors propose that nodes set their respective contention windows with an inverse proportion of the distance from the sender. With this scheme, no control traffic is generated that causes useless overhead. Yet, it is assumed that there is a unique and constant transmission range for all cars in every moment. This is obviously not realistic, as discussed in Section I.

Instead, we propose in this paper a position-aware broadcasting scheme that is able to reduce the number of forwarding hops based on the transmission range estimation. With our scheme, broadcast messages are forwarded after a delay that depends on the node distance from the source and, peculiar of our algorithm, on a continuously estimated transmission range.

III. SYSTEM MODEL

We consider a general IVC application, as those discussed in Section I, that requires the support of a broadcasting scheme able to propagate a message generated by a car to following ones with very small delays. The strip-shaped area-of-interest of this message is limited in geographical extension and allows us to consider a group of cars that move on a highway following a linear track. Elements belonging to this group have an internal mobility with respect to each other.

Car-to-car communications can happen either directly or through multiple hops. To this aim, each car is endowed with an antenna for DSRC/802.11p, which has been declared able to guarantee a maximum range of 1000m under optimal conditions, or a smaller range at very high speeds (around 300m for a car traveling at 200K/h) [4].

On each car is installed a GPS that provides accurate information about time and position while power and computational resources are supposed largely adequate for our application's requirements.

IV. FAST BROADCAST

Our scheme, that we named *Fast Broadcast*, is composed by two phases. The first one, named *estimation phase*, is continuously active and is meant to provide each car with an up-to-date estimation of its backward transmission range. Instead, the second one is performed only when a message has to be broadcasted to all cars in the sender's area-of-interest and we name it *broadcast phase*.

A. Estimation Phase

During the estimation phase, each car tries to estimate its transmission range (frontward and backward) by the means of *hello messages*. The aim is that of having all the cars aware of their transmission range and exploiting it in the broadcast phase.

To continuously refresh cars' estimation of their transmission range, time is divided into *turns* and information collected during a certain turn are kept also for the whole duration of the next one, and then discarded. Information for the current turn is represented by *Current-turn Maximum Front Range* (CMFR) and *Current-turn Maximum Back Range* (CMBR). The former embodies the estimation of the maximum frontward distance from which another car along the strip-shaped area-of-interest can be heard by the considered one. Instead, the latter estimates the maximum backward distance at which the considered car can be heard.

These two variables are continuously updated based on the received hello messages until the current turn expires and, at that point, their value is stored in the *Latest-turn Maximum Front Range* (LMFR) and the *Latest-turn Maximum Back Range* (LMBR), respectively. We maintain both a last-turn and a current-turn value because the former guarantees to have a value computed with an adequately large number of hello messages, while the latter considers fresher information.

```

0 for each turn
1   sending_time := random(turn_size);
2   wait(sending_time);
3   if ¬(heard_hello_msg() ∨ heard_collision())
4     hello_msg.declared_max_range := max(LMFR, CMFR);
5     transmit(hello_msg);
6   endif
7 endfor

```

Figure 1. Hello message sending procedure.

With the help of Fig. 1 and Fig. 2 we explain our scheme's behavior during the procedures for sending and receiving hello messages, respectively. Focusing on the hello message sending procedure (Fig. 1), in every turn, each car determines a random waiting time (lines 1 and 2) after which, if neither other transmission is heard nor collision happened (line 3), proceeds with transmitting a hello packet that includes the estimated maximum frontward transmission range (lines 4 and 5).

As to the overhead caused by this procedure, it is worth mentioning that the total number of generated messages is in the order of $O(I)$. In fact, no more than one hello message is sent every turn by cars in the range of each other, regardless of the number of cars n .

The hello message receiving procedure is depicted in Fig. 2. In particular, a car receiving a hello message determines its own position (line 1), extracts from the hello message the sender's position and the included estimation of the maximum transmission range (lines 2 and 3), and determines the distance between itself and the sender (line 4). If the hello message is received from ahead the value of CMFR is updated (lines 5 and 6), otherwise CMBR is updated (lines 7 and 8). In both cases, the new value is obtained as the maximum among the old one, the distance between the considered car and the hello message sender, and the sender's transmission range estimation provided by the hello message.

```

0 // a hello message is received
1 mp := my_position();
2 sp := hello_msg.sender_position;
3 drm := hello_msg.declared_max_range;
4 d := distance(mp, sp);
5 if (received_from_front(hello_msg))
6   CMFR := max(CMFR, d, drm);
7 else // received from back
8   CMBR := max(CMBR, d, drm);
9 endif

```

Figure 2. Hello message receiving procedure.

B. Broadcast Phase

The rationale behind the estimation phase was that of making cars aware of their transmission range through hello message exchange. Instead, during the broadcast phase, a car that needs to broadcast a message puts to good use its estimated transmission range to reduce transmission redundancy and achieve a fast delivery of the message. Coherently with the best practical strategy discussed in Section II, each car exploits this transmission range information to assign itself a priority in forwarding the broadcasted message based on its distance from the message sender: the higher the relative distance, the higher the priority.

The broadcast phase is triggered by the running application and generates a *broadcast message* that has to be propagated over the whole strip-shaped area-of-interest of the sender. This broadcast message contains information related to the supported applications but also some data utilized by the broadcast algorithm. Therefore, it includes, among the others, also the following fields: *Sender_Position*, *MaxRange*.

MaxRange is a peculiar parameter of our algorithm and represents how far that transmission is expected to go backward before the signal becomes too weak to be intelligible. This value will be used by following cars to determine which one among them will forward the broadcasted message on the next hop. In particular, to minimize the number of hops, and hence the propagation delay, we want the farthest car from the sending (or forwarding) one to perform this task. Therefore, the higher the relative distance of the considered car from the sender with respect to the estimated transmission range, the higher the priority of the considered car will be in forwarding the broadcasted message.

More in detail, when a car has to send or forward a broadcast message it computes the MaxRange value in the broadcast message as $\max(\text{LMBR}, \text{CMBR})$.

Focusing on our algorithm, cars' priorities to forward the broadcast message are determined by assigning different waiting times from their reception of the message to the time at which they try to forward it. This waiting time is randomly computed based on a *contention window* value. If, while waiting, some of the following cars already transmitted the message, preceding ones abort their forwarding countdowns as the message has already been propagated. Obviously, the larger the contention window, the more likely somebody else will be faster in forwarding the broadcast message.

The contention window of each car is varied between a minimum value (CWMin) and a maximum one (CWMax), depending on the distance from the sending/forwarding car (Distance) and on the estimated transmission range (MaxRange) declared in the broadcasted message. This is summarized by (1).

$$\left\lceil \left(\frac{\text{MaxRange} - \text{Dis tan ce}}{\text{MaxRange}} \times (\text{CWMax} - \text{CWMin}) \right) + \text{CWMin} \right\rceil \quad (1)$$

The utilization of (1) in our scheme to determine which car will propagate the broadcast message on the next hop is explained in Fig. 3. Upon receiving a broadcast message from the front, a car utilizes (1) to determine its contention window (line 1) and then computes a random waiting time based on it (lines 2 and 3). If, while waiting, the same message has been heard again coming from behind (line 4), the message has already propagated over the considered car that can hence stop trying to forward it: somebody else already did it (line 5). Conversely, if the same broadcast message is heard from frontward (line 6), this means that a preceding car has already forwarded it. The procedure has to be restarted with the new parameters included in the newly heard broadcast message (line 7).

If the waiting time expires without having heard any other car forwarding the same message then the considered car broadcasts it (line 10) including the estimated transmission range (line 9). Obviously, if the broadcast fail, a backoff mechanism is utilized to compute the next transmission time.

```

0 // a broadcast message received from front
1 contwnd := compute_contwnd_with_equation1();
2 rcw := random(contwnd);
3 wait(rcw);
4 if (same_broadcast_heard_from_back())
5   exit();
6 elseif (same_broadcast_heard_from_front())
7   restart_broadcast_procedure();
8 else
9   maxrange := max(LMBR, CMBR);
10  transmit(broadcast_msg);
11 endif

```

Figure 3. Broadcast message forwarding procedure.

V. A PRACTICAL EXAMPLE

We provided details to help a reader in actually implementing Fast Broadcast. At the same time, our algorithm is composed by several, even if simple, components. We hence deem that stepping through a simple example can be necessary

for clear comprehension. To this aim, we use Fig. 4 as a storyboard for some rounds of Fast Broadcast.

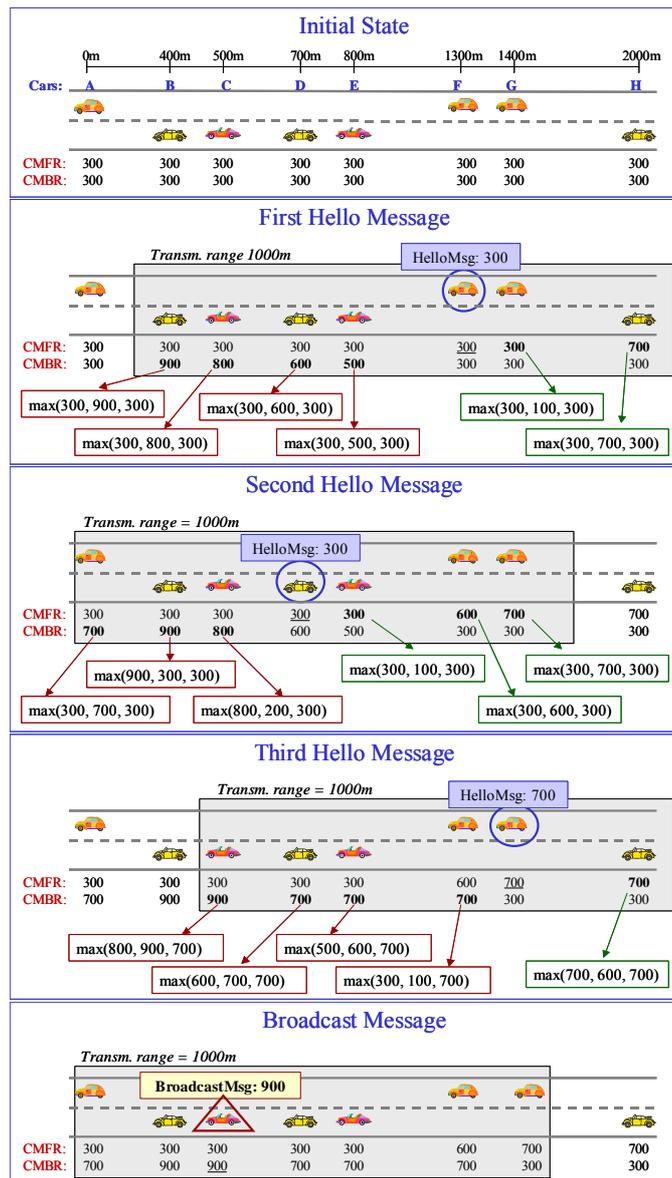


Figure 4. Example of our scheme.

We first suppose to have an initial state where all cars have CMFR = CMBR = 300m (we ignore for simplicity LMFR and LMBR in this example). We consider a 2km portion of road, with cars positioned as indicated in the top part of the figure. The actual transmission range is 1000m. We use simple parameters here as our only purpose in this section is that of showing the behavior of our algorithm.

Without loss of generality, we suppose that car F sends the first hello message stating its CMFR value (i.e., 300m). All the cars hearing this message update their CMFR values if they are in front of car F, or their CMBR values if they follow car F. These variables are updated with the maximum among: i) the old value, ii) the distance from car F, and iii) the Declared_MaxRange in hello message (see lines 5-8 of Fig. 2).

Analogous operations are performed when car D and car G send the second and the third hello message, respectively.

When the broadcast message is then issued by car C, cars' values for CMFR and CMBR have been substantially changed by the algorithm from the initial setting. The broadcast message issued by car C also includes in the MaxRange field the CMBR value. In particular, the maximum transmission range estimated by car C after only three rounds of hello messages is 900m over an actual one of 1000m. Cars following C and hearing the broadcasted message can then utilize (1) to compute their contention window.

VI. SIMULATION ASSESSMENT

To analyze Fast Broadcast we simulated a scenario with a strip-shaped road and considered an area-of-interest of 8km. Several scenarios have been tested and for each of them 20 simulations have been run and their outcomes averaged to produce charts presented in Section VII. Cars' number was set to 500, 700, and 1000, to test different car densities, while their speeds were uniformly distributed in the range 72-144Km/h. Taking inspiration from the real IEEE 802.11 protocol, we have set CWMin and CWMax equal to 32 and 1024 slots, respectively [5]. Finally, we considered two possible cases for the actual transmission range: 300m and 1000m.

We measure in slots all the time variables in the system such as MAC layer contention windows, time intervals, and other random waiting times. These time slots can be very small (e.g., the IEEE 802.11g's slot amounts to $9\mu s$ [5]) thus allowing a fine grained, even if discrete, representation of the time. Having a discrete progression of the time is not a necessary condition for our scheme to work but allows us to provide a clearer explanation of the algorithm.

Inspired by [12], we have compared our scheme with a similar one that does not make use of dynamically estimated transmission range (we name this scheme *Static*). The Static scheme tries to assign different forwarding priorities through random backoff delays that depend on the node distance from the source. However, this is done by utilizing a fixed MaxRange value, regardless of the actual transmission range, instead of dynamically computing it as Fast Broadcast does. In particular, two instances of the Static scheme were contrasted with our algorithm: i) the Static scheme utilizing a fixed MaxRange of 300m (*Static300*) and ii) the Static scheme utilizing a fixed MaxRange of 1000m (*Static1000*).

We compared the various schemes in two scenarios having actual transmission ranges of 300m and 1000m, respectively. Therefore, in each of the two scenarios, one of the Static schemes will be optimally set, while the other one will make use of a non properly set MaxRange value. We will show how Fast Broadcast's performances are always equal to those achieved by the optimally set Static algorithm, thus ensuring fast broadcast delivery. Prominent advantage of our approach, this result is achieved without requiring perfect knowledge of the network topology and eliminating the performance dependency on the difference between the assumed transmission range and the actual one. In fact, prominent contribution of this work is the design of a novel transmission

range estimator that dynamically adapts to actual channel conditions with a very limited traffic overload.

To compare the various schemes from a propagation delay viewpoint, we focused on parameters that have a direct delay impact. Specifically, we consider the following metrics: the total number of hops required to propagate the broadcast message over the whole area-of-interest, the total number of MAC layer slots waited before actually transmitting the broadcasted message over the various hops, and the total number of MAC layer collisions (and hence time wasting retransmissions) experienced by the broadcasted message.

VII. PRELIMINARY RESULTS

In Fig. 5 and Fig. 6, we show the number of hops that a broadcast message experiences before covering the whole considered portion of road. In particular, the actual transmission range was 300m in Fig. 5 and 1000m in Fig. 6, respectively. As expected, the Static scheme employing a non-optimal MaxRange value always requires a higher number of hops to propagate the message with respect to the one utilizing the right one. Noticeably, our scheme is able to adapt itself to real channel conditions of both scenarios, finding out the actual transmission range, and achieving best performance.

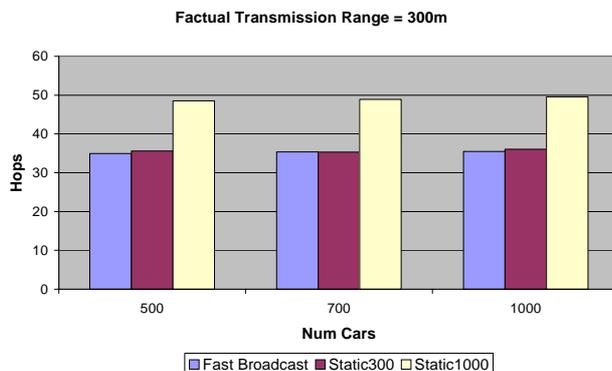


Figure 5. Number of hops required to propagate the broadcast message with an actual transmission range of 300m.

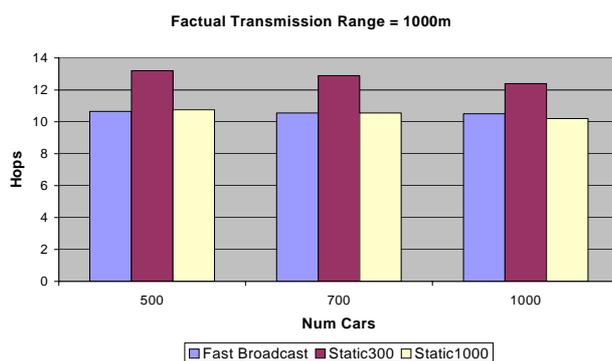


Figure 6. Number of hops required to propagate the broadcast message with an actual transmission range of 1000m.

The consequences on propagation delays caused by utilizing a wrong transmission range parameter (i.e., MaxRange) are even more evident in Fig. 7 and Fig. 8. In

particular, we expect that when the Static scheme considers a larger transmission range than the actually available one, the average waiting time will be higher. This is caused by the fact that the farthest car hearing the broadcasted message will make use of a contention window $CW_{Farthest} > CW_{Min}$. The consequences of this problem are evident in Fig. 7 that shows how the total number of MAC layer slots waited before propagating the broadcasted message is hugely higher when the Static1000 is utilized in the scenario with 300m of actual transmission range.

On the other hand, even an excessively conservative estimation of the transmission range leads to disruptive consequences in terms of experienced propagation delays. In this case, the proportional decrease of the contention window size stops at the estimated transmission range, while all the cars between this distance and the actual transmission range will make use of CW_{Min} as their contention window. This means that potentially many cars attempt to forward the broadcast message after only few time slots, thus incurring in collisions and time consuming retransmissions. As a confirmation, Fig. 8 shows how sensibly higher is the number of collisions when the Static300 is used in the scenario with 1000m of actual transmission range.

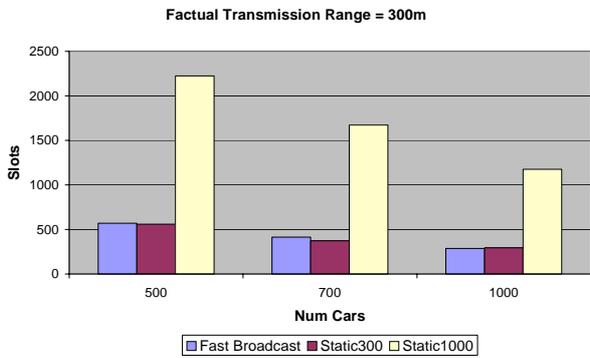


Figure 7. Total number of slots required to propagate the broadcast message with an actual transmission range of 300m.

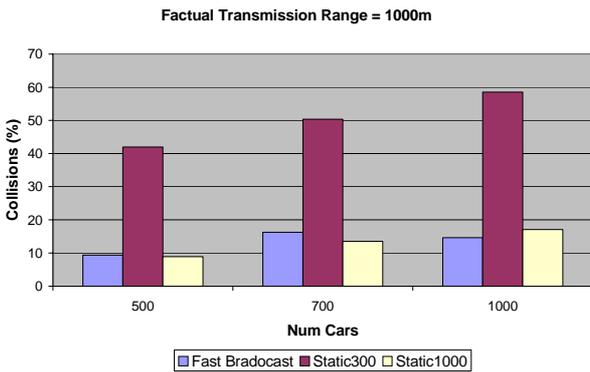


Figure 8. Percentage of collisions/retransmissions experienced by a broadcasted message over the total number of transmissions required to propagate it; actual transmission range = 1000m.

Remarkably, for all the considered scenarios, our scheme is able to dynamically adapt its estimated transmission range,

thus avoiding potential propagation delay sources (i.e., excessive number of hops, of waited slots, and of collisions).

VIII. CONCLUSION

We designed a multi-hop broadcast protocol (Fast Broadcast) and provided an algorithm to practically implement it in a realistic IVC scenario. Interferences caused by environmental conditions and cars' mobility are taken into account by dynamically computing cars' transmission ranges, whose estimated actual values are exploited to minimize the number of hops to be traversed, as well as the number of message retransmission, during the broadcast activity. Preliminary results confirm that broadcasted messages reach the end of their area-of-interest with as few transmissions as possible, thus reducing the required delivery time.

REFERENCES

- [1] A. Nandan, S. Das, B. Zhou, G. Pau, M. Gerla, "AdTorrent: Digital Billboards for Vehicular Networks", submitted for publication IEEE/ACM International Workshop on Vehicle-to-Vehicle Communications (V2VCOM), San Diego, USA, Jul 2005.
- [2] A. Nandan, S. Das, G. Pau M.Y. Sanadidi and M. Gerla ---Cooperative Downloading in Vehicular Ad Hoc Wireless Networks , in Proc. of IEEE/IFIP International Conference on Wireless On demand Network Systems and Services , St. Moritz, Switzerland, Jan 2005, pp:32-41.
- [3] C. E. Palazzi, S. Ferretti, S. Cacciaguerra, M. Rocchetti, "Interactivity-Loss Avoidance in Event Delivery Synchronization for Mirrored Game Architectures", IEEE Transactions on Multimedia, IEEE Signal Processing Society, vol. 8, no. 4, Aug 2006, pp. 874-879.
- [4] ASTM E2213-03, "Standard Specification for Telecommunications and Information Exchange Between Road-side and Vehicle Systems – 5GHz Band Dedicated Short-Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ASTM International, Jul 2003.
- [5] IEEE 802.11g, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 4: Further Higher Data Rate Extension in the 2.4GHz Band.
- [6] E. Fasolo, R. Furiato, A. Zanella, "Smart Broadcast Algorithm fo Inter-vehicular Communication", in Proc. of Wireless Personal Multimedia Communication (WPMC'05), IWS 2005, Aalborg, DK, Sep 2005.
- [7] L. Pantel, L. C. Wolf, "On the Impact of Delay on Real-Time Multiplayer Games", in Proc of NOSSDAV'02, Miami, FL, USA, 2002, pp. 23-29.
- [8] S. Biswas, R. Tatchikou, F. Dion, "Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety", IEEE Communication Magazine, vol. 44, no. 1, Jan 2006, pp. 74-82.
- [9] X. Yang, J. Liu, F. Zhao, N. Vaidya, "A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning", in Proc. of 1st Annual International Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), Boston, MA, USA, Aug 2004, pp. 114-123.
- [10] G. Korkmax, E. Ekici, F. Ozguner, U. Ozguner, "Urban Multi-hop Broadcast Protocol for Inter-vehicle Communication Systems", in Proc. of the 1st ACM Workshop on Vehicular Ad-hoc Networks (VANET 2004), Oct 2004, pp. 76-85.
- [11] K. O'Leary, S. O'Leary, Classroom Management: the Successful Use of Behavior Modification, Pergamon, New York, 1997.
- [12] A. Zanella, G. Pierobon, S. Merlin, "On the Limiting Performance of Broadcast Algorithms over Unidimensional Ad-hoc Radio Networks", in Proc. of WPMC04, Abano Terme, Italy, Sep 2004.
- [13] P. J. Wan, K. Alzoubi, O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad hoc Networks", in Proc. of IEEE INFOCOM 2002, Jun 2002.