Emergency Braking:
A Study of Network and Application Performance

Michele Segata
University of Trento
Trento, Italy
msegata@disi.unitn.it

Renato Lo Cigno
University of Trento
Trento, Italy
locigno@disi.unitn.it

ABSTRACT

Safety applications are among the key drivers in VANET research. Their study is complex as it encompasses different disciplines, from wireless networking to car dynamics, to drivers' behavior, not to mention the economic and legal aspects. This work presents a simulative study of emergency braking applications tackled by embedding a mobility, cars' dynamic, and driver's behavior model into a detailed networking simulator (ns-3). The results, derived both at the network and at the application level, capture correctly the interactions of the communications and protocols with the car's adaptive cruise control system and the driver's behavior for cars that are not equipped with communication devices. The paper presents in detail the improvements we contribute in simulation techniques and model completeness. It introduces a novel and easy message aggregation technique to empower message re-propagation while controlling the network load during the peak due to the emergency braking. Finally it discusses the effectiveness of such applications as a function of the market penetration rate, showing that even cars that are not equipped with communication devices benefit from the smoother and earlier reaction of those cars that can communicate and whose adaptive cruise control implements a correct deceleration strategy.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - Wireless Communication; I.6.5 [**Simulation and Modeling**]: Model Development

General Terms

Design, Performance

Keywords

VANET, vehicular networks, emergency braking control, collaborative adaptive cruise control, ns-3 simulation, rebroadcast schemes

VANET'11, September 23, 2011, Las Vegas, Nevada, USA.

1. INTRODUCTION

Cooperative driving and safety applications are the leading forces for the development and deployment of VANETs. Among these applications, Emergency Electronic Brake Lights (EEBL), is probably the conceptually easiest and one of the most useful. A large fraction of car accidents and the correlated casualties are related to bad brake usage by drivers – or simply drivers who do not even attempt to brake[1].

In spite, or maybe because, of the application's simplicity, there are still quite a few issues which have not been completely explored. How do the application and the VANET network actually interact? What is the market penetration rate that makes EEBL benefits measurable? Do non-equipped vehicles benefit from a partial EEBL introduction? What is the devices' communication range that makes the application most effective? Are rebroadcast techniques needed to make EEBL more effective? If yes, how do they impact the network load? Can we find easy and efficient message aggregation techniques? Many more interesting questions arise spontaneously, but these are probably the most fundamental ones.

The focus of this paper is the joint analysis of EEBL and network protocols, an approach that to the best of our knowledge is utterly missing in the literature, where research either mainly focuses on network performance or analyzes an application. The contribution of this work are some answers (still partial) to some of the questions posed above: *i)* A tool for the joint simulation of application and network based on ns-3 is developed and made available to the community; *ii)* The influence of the application's dynamics on the network is analyzed, measuring the network load in space and time; *iii)* Advantages and disadvantages of message rebroadcast are investigated and a novel, joint rebroadcast and aggregate algorithm is proposed and evaluated. Its aim is maintaining a low network load while leveraging the benefits of rebroadcast; *iv)* The application's performance is analyzed in detail, both in terms of car crash reduction as a function of the market penetration rate, and in terms of detailed vehicle dynamics during the braking phase.

The paper methodology is simulative, for obvious reasons. It considers a simple highway scenario, coupled with a very detailed vehicle, driver and Adaptive Cruise Control (ACC) model, and a realistic packet-level propagation model below an 802.11p implementation, looking for the most appropriate level of modeling abstraction to tackle the complete problem via simulations.

The remaining part of the paper is organized as follows. Sect. 2

---

[1] This statement is supported by data published by the US National Highway Traffic Safety Administration `http://www-fars.nhtsa.dot.gov/`. By browsing the data published under the FARS section it is clear that accidents related to drivers' distraction or lack of driving capabilities, including drugs and alcohol, dominates the lot of multi-vehicle accidents and casualties

reviews the papers that most influenced our work, as well as those that compare more closely to it. Sect. 3 describes the simulation scenario and the models specifically developed for this work, including the rebroadcast and aggregate scheme. Sect. 4 discusses the network-level performance when all the cars are equipped, which is the most stressful scenario for the network. Sect. 5 presents the application level results as a function of the penetration rate. Sect. 6 discusses the impact of the traffic in the reverse direction, specially in case of non-rational drivers' behavior as they slow down to watch what is happening in the other carriage-way. Sect. 7 closes the paper with a discussion of the results and of the work still remaining ahead.

2. RELATED WORK

The literature proposes different methods for information dissemination in VANETs, both for unicast and multicast/broadcast transmissions. Problems related to broadcast communications, as for example, congestion issues derived from message repropagation, have been investigated since 1999 [15]. Several approaches have been analyzed, starting from probabilistic ones [9,20]. These proposals have been revisited in many ways in order to improve their performances: a common characteristic is the distance-based prioritization for rebroadcast decision [4,6,13,16].

To further reduce network utilization, more complex approaches have been described. Rebroadcast decisions can be taken upon different criteria, for example dividing the portion of the road in the transmission range of the sender into segments along the dissemination direction and then selecting the farthest node as rebroadcaster [12]. Other approaches behave differently as the traffic conditions change [19] or group vehicles into clusters [7]. Another interesting approach is based on a mixed clustering and aggregation [10] technique, i.e., grouping information about a cluster of vehicles in a single message.

The problem of all these protocols is that they are mostly, if not completely, network-oriented, in the sense that application requirements and/or vehicles dynamics are not considered. For example, in Ibrahim and Weigle [10], the protocol is described as "general purpose", meeting the requirements of safety and non-safety applications due to the update rate in the order of 300-400 ms. However, the U.S. D.O.T. indicates a rate of 10 Hz for some applications (e.g., EEBL) [1].

As a further remark, rebroadcast techniques are proposed without investigating whether a real need of repropagation exists; benefits are given for granted and not quantified.

On the other hand, purely application-oriented analysis exist, where advantages are shown but without considering the impact on the network [24].

Only recently, the importance of application-level requirements in the design of VANET network protocols has been highlighted [8, 14,18,25]. Surprisingly, as stated in Haas and Hu [8], the protocols proposed in the literature are analyzed with simulators using crash-free mobility models, so there is no way to evaluate their impact on safety.

These are the motivations behind our joint application/network analysis, using a network and a mobility simulator modified in order to account for accidents.

3. SIMULATION MODELS AND TOOLS

To perform our analysis we use the ns-3[2] (v 3.9) network simulator extended with the well-known IDM (Intelligent Driver Model) and MOBIL microscopic mobility models [21,22] embedded by Arbabi and Weigle [3]. Given the crash-free nature of IDM, we modify the original simulator in order to introduce realistic car accidents; indeed, as pointed out by Haas and Hu [8], previous works do not consider collisions, and we actually could not find open source simulators with this feature.

IDM results in crash-free scenarios, but only thanks to unrealistic decelerations. During early simulations, we observed decelerations larger than $10\,\mathrm{m/s^2}$, which are not reachable by common production cars [2,17]. To overcome this impairment, we introduce a maximum deceleration as a physical limit, using values obtained from braking tests on dry surface [17]. Equation:

$$a(t) = \max(-b_i^{\max}, a^{\text{IDM}}(t)) \tag{1}$$

where $b_i^{\max}$ is the maximum deceleration for vehicle i and $a^{\text{IDM}}(t)$ is the value of acceleration obtained from the IDM formula, defines the Limited-IDM (L-IDM), i.e., a more realistic model which accounts for fundamental physical limits. This modification results in crashes during emergency braking, as normally observed in highways. When two vehicles collide, the follower "pushes" its leader causing their speed to become the average of the two speeds. Moreover, if the follower has a stronger deceleration, they will continue with the same deceleration they had before the crash, otherwise they will remain bumper to bumper and continue to slow down with a deceleration equal to their average.

Each vehicle is equipped with an accelerometer, which basically measures the change of speed every 100 ms. This is more realistic than taking the perfect acceleration computed with the L-IDM formula and is used to send acceleration data in EEBL messages and to measure the peaks of acceleration caused by crashes.

Another improvement we implement is an independent per-car clock, used to avoid the occurrence of events at the same exact time, which is unrealistic and causes the simulation to generate synchronization phenomena. Moreover, ns-3 does not consider processing delays, so every send and receive event is delayed by a random time between 0 and $10\,\mu$s.

A fundamental component which is added to the simulator is the ACC, which processes the data received from other vehicles to keep safety distance and brake if the driver does not react in time to a dangerous situation. The ACC works first of all by considering the approaching rate: if it is negative (i.e., leading vehicle is traveling faster than its follower), then the vehicles are increasing their gap and the ACC remains disabled leaving the car control to L-IDM. If instead the approaching rate is positive, the ACC computes the safety gap to determine if a vehicle is too close to its leader. The safety gap depends on the speed of the vehicle and it is computed as

$$s_{\text{safe}} = T_{\text{ACC}} \cdot v_i + \varepsilon_{\text{ACC}}$$

where T_{ACC} is the time headway for the ACC, v_i is the current vehicle's speed and ε_{ACC} is a small quantity to account for errors. T_{ACC} is set to 1 s, which, as mentioned in Treiber et al. [23], is an average time measured on German freeways which is considered safe. ε_{ACC} is set to 1 m.

The actual gap between the car and leading vehicle can be greater or lower than the safety gap. If it is lower, then the follower must brake in order to move away from the leader. The applied deceleration can be computed from the leader's acceleration minus a small quantity. If, instead, the actual gap is greater than the safety gap, the ACC computes a deceleration using the formula:

$$a^{\text{ACC}} = \frac{v_{i+1}^2 - v_i^2}{2 \cdot (s_{\text{actual}} - s_{\text{safe}})}$$

---

[2] `http://www.nsnam.org`

which computes the acceleration needed to bring the speed of the following vehicle (v_i) to the speed of its leader (v_{i+1}) in a space equal to the difference between the actual gap (s_{actual}) and the safety gap (s_{safe}).

ACC is designed to work only with messages received directly from the vehicle in front, to guarantee that the deceleration correctly follows the goal of avoiding a direct crash. In presence of cars not equipped with communication devices, however, this can be a limitation, and will in any case hamper any benefit derived from propagating EEBL messages far from the originator. Indeed, EEBL messages coming from vehicles far ahead can be used as warnings, throttling the gas and resulting in a deceleration induced by the air resistance[3]. If no EEBL messages are received within two seconds, the vehicle returns to obey the L-IDM formula. The drag force is

$$F_{\text{drag}} = \frac{1}{2}\rho v^2 C_D A. \tag{2}$$

Eq. (2) is known as the *drag equation* [5] and gives the force which a body with a section A and a drag coefficient C_D is subject to if it runs at a speed v in a fluid of density ρ. The deceleration is a function of the speed and air density and vehicle mass M_v:

$$b_{\text{air}} = \frac{F_{\text{drag}}(v)}{M_v}$$

where $\rho = 1.20$ kg/m^3 (air at 20°C). The $C_D \cdot A$ product is randomly set for each vehicle using values for common production cars found on a site which collects this kind of measurements[4], while M_v is fixed at 1500 kg for the sake of simplicity.

This reaction to warnings is a simplification of a complex human and ACC behavior which could be enhanced by giving a weight to the warning according, for example, to the distance. This is left as a future work.

3.1 Network Protocols

Apart from the mobility modifications, we implement some network protocols in order to diffuse information among vehicles. The first we implement is the beaconing protocol, which broadcasts messages at a frequency of 1 Hz: messages contain information about the sending vehicle, such as speed, acceleration, position, etc. The same message structure is used for both beacons and EEBL messages, following the indications of the U.S. Department of Transportation [1]. Moreover, a header is defined to meet other application requirements, for example sender identification. Tab. 1 lists the fields of the message while the meaning of the header fields (101 bytes) is the following:

Type (1 byte) determines if the packet is a beacon, an EEBL message or an aggregated EEBL message;

Packet id (4 bytes) is used (for statistical purposes) as a unique identifier for the packet. In a real scenario it could be thought as a sequence number;

Originator id (4 bytes) is the id of the vehicle which has created the message, or better, the vehicle whose position, speed, etc., the packet describes;

TTL (1 byte) is the time-to-live of the packet in hops. Initially it is set to 0 for beacons and for EEBL messages which must not be rebroadcast, otherwise it is set as 5;

Sender id (4 bytes) is the id of the vehicle which has actually sent the message. In the case of a rebroadcast EEBL message, it is different from the originator's id;

Count (1 byte) tells, in the case of an aggregated EEBL frame, how many messages are included;

Certificate (58 bytes) is the digital certificate of the originator, for authentication purposes[5]. It is not used in the simulation, but it must be included in order to properly analyze the network load. The size of 58 bytes is taken from [10];

Digital Signature (28 bytes) is the signature of the originator. As for the certificate, it is not actually used in the simulation and its size is taken from [10].

Description	Size (bit)	Size (byte)
GPS coordinates	96	12
Time stamp	64	8
Vehicle speed	16	2
Vehicle acceleration/deceleration	16	2
Vehicle heading	16	2
Vehicle size (length, width, height)	48	6
GPS antenna offset (relative XYZ)	32	4
Total	288	36

Table 1: Required message data set for EEBL application, as suggested by U.S. D.O.T. [1]

In practice the EEBL protocol sends beaconing messages with a 10 Hz frequency as indicated by U.S. D.O.T. [1]. Switching between beaconing and EEBL protocol happens when the deceleration of the vehicle is greater than 1 m/s^2. Moreover, packets coming from vehicles behind or from vehicles driving in the opposite direction are ignored.

To analyze the application's benefits and the impact of message repropagation on network load, we implement a rebroadcast protocol taken from the literature, as depicted by the listing in Fig. 1. We call it EEBL with Rebroadcast (EEBLR) and it is based on the *weighted p-persistence* broadcast suppression mechanism [20]. The rebroadcast decision is taken with a probability p which depends on the distance from the sender. If a message is seen for the first time (identified with the 'Packet id' field), it is passed to the application for processing and, if needed, rebroadcast; otherwise it is ignored. If TTL$=0$, it is not rebroadcast even if it is seen for the first time. Parameters are the same as in [20].

The rebroadcast of single messages is highly inefficient due to the 802.11p MAC protocol, whose overhead for channel contention is very large. For this reason, we propose an *aggregation and rebroadcast* protocol, which blends the advantage of efficient message dissemination with a lower usage of network resources. The protocol is described by the listing in Fig. 2 and builds on the same idea of stochastic rebroadcast of EEBLR, but provides for application level message aggregation; for this reason we call it EEBL with Aggregation (EEBLA). The rebroadcast decision is taken with the same criterion of EEBLR, but, instead of sending a message immediately, it is inserted into a queue to be sent in the same 802.11p

---

[3] We disregard for the time being other phenomena as the motor braking effect, since at high speed the air drag is dominant.

[4] `http://rc.opelgt.org/indexcw.php`

[5] In the case of rebroadcast a single certificate (with a single signature) is not enough because the receiver is interested in the "full chain", from the originator to the last rebroadcaster. However, security in VANETs is a problem in its own and the development of a secure protocol goes beyond the scope of this work and it is left as future work

```
list $KnownPackets$

on InitProgram()
$KnownPackets \leftarrow \emptyset$

on ReceiveEEBLPacket($eebl$):
if $KnownPackets$.Contains($eebl$) then
    return
else
    $KnownPackets$.Insert($eebl$)
    ProcessPacket($eebl$)
    if $eebl.ttl \neq 0$ then
        $p \leftarrow$ ComputeRebroadcastProbability($eebl$)
        if Random() $< p$ then
            $eebl.ttl \leftarrow eebl.ttl - 1$
            Broadcast($eebl$)
        end if
    end if
end if
```

Figure 1: EEBLR protocol

frame as the (potential) local EEBL message, generated with the 10 Hz frequency. Since the queue is managed by the application, if another copy of the message is received it means that the ‘Packet id’ message has already been rebroadcast by someone else close by, and it is removed from this queue, reducing the number of useless rebroadcasts. This is not possible with EEBLR due to the fact that, once a packet has been scheduled for rebroadcast, it cannot be removed from the MAC queue even if it has not been transmitted yet. Also local EEBL messages are inserted in the same queue. Every 100 ms (10 Hz), the queue is emptied and a single frame (if the maximum frame size allows it) containing all messages is sent.

3.2 Simulated Scenario

The scenario we concentrate on is basic but fundamental: a highway stretch where, due to an accident or whatever other impairment, some cars brake until they arrive to a complete stop, forcing all the following cars (the platoon) to come to a complete stop too. The leader (or the leaders, in the multi-lane scenario) brakes with a constant deceleration of 4 m/s^2. We randomly choose driving styles and vehicles characteristics. For example, we model the aggressiveness of the driver using different values for desired speed, time headway (i.e., the T parameter of the IDM formula) and politeness (i.e., the p parameter of MOBIL), while we characterize vehicles through different maximum deceleration and different drag area (see Eq. (2)). These parameters are described in Sect. 3.3.

The tests always refer to the four models we define: **L-IDM**, where VANET technologies are not employed in the car; **EEBL**, with beaconing and plain EEBL protocol without rebroadcast; **EEBLR**, with beaconing and EEBLR protocol; **EEBLA**, with beaconing and EEBLA protocol. Beacons are never rebroadcast and L-IDM can be freely mixed with any other model to test situations where VANET technologies have a limited penetration.

We repeat all tests 20 times with different parameters and (when appropriate) 95% confidence intervals are computed. We investigate scenarios with 1, 2, 3, 4, and 5 lanes with average desired speeds of 50, 70, 90, 110, 130, and 150 km/h, but we report here only the most interesting results for lack of space. The results encompass the fraction of cars involved in collisions, the average maximum decelerations, and the deceleration profiles of the different models, the network load and the fraction of lost messages. Again, not all these metrics are reported here for the lack of space. In particular, deceleration profiles and distributions are missing, though they confirm all other results and show the additional benefits of the introduction of EEBL in terms of driving comfort. The collision rate, a “traditional” metric of MAC protocols performance is instead missing. The reason is that, albeit often considered one of the most important metrics, it is hard to define and also misleading, since in real scenarios per-packet channel capture phenomena, where some stations receive a frame and some others do not or even receive another one, are dominating over the naive view where all frames are lost during a collision. Indeed, in a situation as complex as 250 moving cars (in the five-lane scenario), even the definition of the communication channel is entirely blurred, let alone the computation of the fraction of frames that would have been received if they did not collide on the channel. We measure instead, as mentioned above, the fraction of frames that are not received by any other station, which is clearly defined and identifies information which is really lost for the entire system.

```
list $KnownPackets$
list $SendQueue$

on InitProgram():
$KnownPackets \leftarrow \emptyset$
$SendQueue \leftarrow \emptyset$
ScheduleEvent(SendPackets, $100ms$)

on ReceiveAggregatedEEBLPacket($aggregatedeebl$):
for all $eebl$ in $aggregatedeebl$ do
    ReceiveEEBLPacket($eebl$)
end for

ReceiveEEBLPacket($eebl$):
if $KnownPackets$.Contains($eebl$) then
    if $SendQueue$.Contains($eebl$) then
        $SendQueue$.Remove($eebl$)
    end if
    return
else
    $KnownPackets$.Insert($eebl$)
    ProcessPacket($eebl$)
    if $eebl.ttl \neq 0$ then
        $p \leftarrow$ ComputeRebroadcastProbability($eebl$)
        if Random() $< p$ then
            $eebl.ttl \leftarrow eebl.ttl - 1$
            $SendQueue$.Insert($eebl$)
        end if
    end if
end if

on SendPackets():
if $SendQueue$.Size() $\neq 0$ then
    if $SendQueue$.Size() $= 1$ then
        $packet \leftarrow SendQueue$.Get(0)
    else
        $packet \leftarrow$ CreateAggregatedPacket($SendQueue$)
    end if
    Broadcast($packet$)
    $SendQueue$.Empty()
end if
ScheduleEvent(SendPackets, $100ms$)
```

Figure 2: EEBLA protocol

First, the system is evaluated with homogeneous scenarios, where all cars are equipped with VANET technologies. Then, a market penetration rate (MPR) analysis is conducted. MPR indicates the fraction of vehicles that are equipped with VANET technologies and ACC. The aim is to determine if it is possible to gain benefits when not all vehicles are equipped. These tests are repeated 30 times to have better estimates, because the fraction of cars involved in an accident has a strong dependency on the random distribution of equipped cars.

Finally, we also investigate the impact of the traffic in the reverse direction. We modify the L-IDM model in the opposite direction to model drivers that behave as "curious": some randomly chosen vehicles, when approaching the braking platoon, decelerate to simulate the curiosity of their driver.

3.3 Simulation Parameters

Tab. 2 lists the values for IDM parameters we use in the simulations. We randomly set cars' maximum deceleration in $[5.9, 8.4]$ [17]. We also randomly set the desired speed as the average speed of the simulation ± 15%. We set the L-IDM desired deceleration to -4 m/s^2, quite large but suitable for dangerous situations. T ranges randomly between 0.1 s (very aggressive driver) and 1.1 s (safe driver). Finally, we take s_o and δ from the original IDM paper [22].

Parameter	Value	Unit
b_{max}	$[5.9, 8.4]$	m/s^2
v^{des}	$\bar{v} \cdot [0.85, 1.15]$ $(\pm 15\%)$	m/s
a	1.7	m/s^2
b	-4	m/s^2
T	$[0.1, 1.1]$	s
s_0	2	m
δ	4	

Table 2: IDM parameters used in simulations

The second set of parameters (shown in Tab. 3) is for MOBIL. The politeness factor p ranges randomly between 0 (totally impolite) and 0.5 (very polite). We set the b_{safe} parameter so that 7 m/s^2 is the maximum deceleration a driver can cause to incoming vehicles by changing lane. We set the minimum gap s_{min} to 2 m, δ_{thr} to 0.3 m/s^2 and δ_{bias} to 0.2 m/s^2, so $\delta_{R\rightarrow L}$ and $\delta_{L\rightarrow R}$ are 0.5 and 0.1 m/s^2 respectively.

Parameter	Value	Unit
p	$[0, 0.5]$	#
b_{safe}	7	m/s^2
s_{min}	2	m
δ_{thr}	0.3	m/s^2
δ_{bias}	0.2	m/s^2

Table 3: MOBIL parameters used in simulations

Tab. 4 lists the wireless network (802.11p) parameters. We use two different ACs for beacons and EEBL messages: AC_BK and AC_VO respectively. The data rate is 6 Mbps and the transmission power is 20 dBm. To model the propagation loss, we use the "three log distance" of ns-3 with default parameters. The loss for this model is

$$L = \begin{cases} 0 & d < d_0 \\ L_0 + \lg\left(\frac{d}{d_0}\right)^{N_0} & d_0 \leq d < d_1 \\ L_0 + \lg\left(\frac{d_1}{d_0}\right)^{N_0} + \lg\left(\frac{d}{d_1}\right)^{N_1} & d_1 \leq d < d_2 \\ L_0 + \lg\left(\frac{d_1}{d_0}\right)^{N_0} + \lg\left(\frac{d_2}{d_1}\right)^{N_1} + \lg\left(\frac{d}{d_2}\right)^{N_2} & d_2 \leq d \end{cases}$$

where L_0 is the path-loss at reference distance (also accounting for antenna gains), d_0, d_1 and d_2 are the three distance fields, N_i is computed as $10 \cdot n_i$ for the three path-loss exponents n_0, n_1 and n_2. For the time being we deliberately do not consider more complex propagation models, as the Nakagami, for the sake of easy results' interpretation. The study of the impact of propagation conditions is one of our next research goals.

Parameter	Value	Unit
IEEE standard	802.11p CCH	
AC (beacons)	AC_BK	
AC (EEBL)	AC_VO	
Data rate	6	Mbps
Bandwidth	10	MHz
Tx power	20	dBm
Propagation loss	Three log distance	
d_0, d_1, d_2	1, 200, 500	m
n_0, n_1, n_2	1.9, 3.8, 3.8	#
L_0	46.67	dB

Table 4: Network parameters

4. NETWORK LEVEL RESULTS

Beacons and EEBL messages are all transmitted in broadcast, as 802.11p safety-related messages, using the 6 Mbit/s PHY transmission speed as suggested in Jiang et al. [11].

The goal of this section is the evaluation of the impact the EEBL application has on the network as a function of the protocol used: plain EEBL, EEBLR or EEBLA. Trivial computations indicate that for low car densities or a small number of lanes the network load is marginal. This is confirmed by the results, so we concentrate on scenarios where all cars are equipped with EEBL and there are multiple lanes (up to five) on the highway.

Albeit it may seem strange, the simple definition of *network performance* is not trivial in these scenarios. The "communication channel" is distributed in space and time, and it evolves as the platoon of cars compacts due to braking and slowing down. The network load is difficult to define due to the complexity of the communication channel. The collision rate is hard to measure both in reality and with a realistic simulator like ns-3, which, on a per-station basis, computes the correct decoding probability, so that per-packet channel captures are taken into account.

We decide to measure the network performance based on two metrics. The first one is the *channel load at each station* $\rho_i(t)$, defined as the fraction of time, for each second t, that a station s_i observes the channel busy, obviously including the time s_i itself uses to transmit. The measure of this metric would be straightforward, which is one of the reasons why we chose it, but its correct computation in ns-3 simulations is not, and Fig. 3 explains why: events in the simulator record when the frames are eligible for transmission (time instants Tx_i in the figure) and when they are correctly received (time instants Rx_i in the figure), so that in presence of contention, when some stations defer the channel access, proper corrections must be implemented to avoid counting transmission times twice (or more). Moreover, transmissions are not aligned to the second, so that proper compensation must be implemented.

The second metric is the percentage of frames l_{uf} that are *not received by any station*. l_{uf} is again time-varying and typically zero during normal operation. However, we are interested in l_{uf} during the "most stressful" period for network load. In the absence of a formal definition of this period, we heuristically choose to let it start when leaders begin to brake and end when the deceleration of all vehicles is less than 1 m/s^2 and their speed is lower than 30 km/h. Notice that it is not necessary that $\rho_i(t)$=1 for l_{uf} to be larger than zero. Together $\rho_i(t)$ and l_{uf} define reasonably well the status

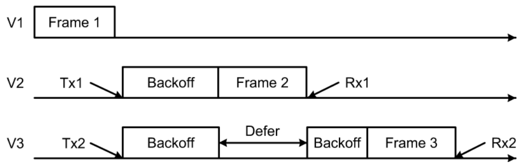


Figure 3: Computation of the channel load $\rho_i(t)$ in the simulator

of the communication channel and the network "stress" due to the emergency braking event.

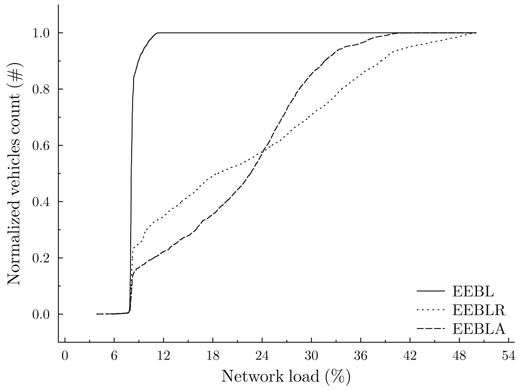


Figure 4: Maximum channel load $\rho_i^{\max}$ observed by each vehicle during the simulation for the five-lane scenario for plain EEBL, EEBLR and EEBLA

Fig. 4 reports the CDF of the maximum load $\rho_i^{\max} = \max_t[\rho_i(t)]$ observed by each vehicle during the simulations (all runs). The plain EEBL protocol does not load the network significantly, but, as we will see in Sect. 5, it also results in poor performance of accidents prevention. The EEBLR and EEBLA rebroadcasting protocols instead lead to fairly heavy network loads peaking over 50% for EEBLR. The network load as defined by $\rho_i(t)$ does not however tell all the story about network performance, because it is averaged over 1 s intervals and because it is well known that CSMA/CA protocols can lead to high data loss rates even for moderate network loads. Fig. 5 reports l_{uf} as a function of the number of lanes. This is probably the main measure to evaluate the possibility that the application will not react correctly due to lack of information. Albeit neither the plain EEBL nor EEBLA have an information loss rate strictly equal to zero, l_{uf} remains extremely low. EEBLR on the contrary leads to a very high fraction of frames that are not received by any station, so that they are completely lost for the application.

The maximum load $\rho_i^{\max}$ and the message loss rate l_{uf} give only a snapshot of the network dynamics, which is not enough to understand the "big picture" of how the VANET evolves as the car platoon brakes. Fig. 6 reports a color plot of the load $\rho_i(t)$ of the network as a function of time (vertical axis) and space (horizontal axis) measured from the first car of the platoon, so as to capture the dynamics of the network as the platoon brakes and finally stops. We regret to have to resort to colors, which may hamper clarity in printed versions, but it has proven impossible to plot the results satisfactorily with a 3-D plot. The color in the plots represent the channel load $\rho_i(t)$ mediated over the cars present in road sectors of 50 m.

Black areas identify the portions of the highway without cars due to the simulation scenario. The void area on the lower right corner

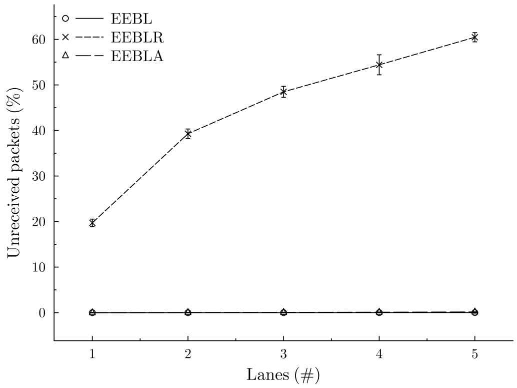
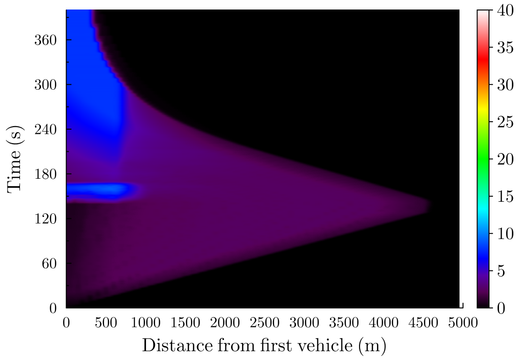
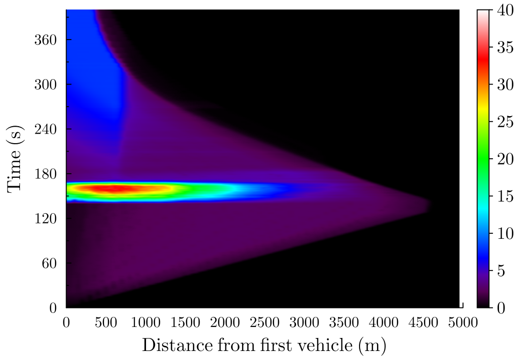


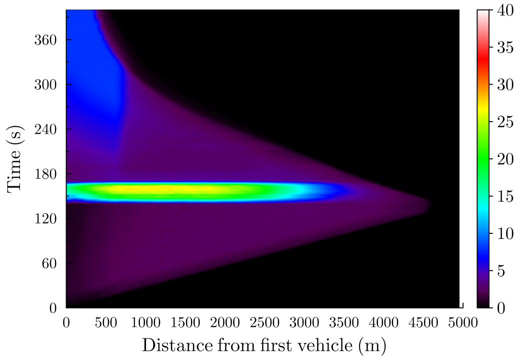
Figure 5: l_{uf} as a function of the number of lanes for plain EEBL, EEBLR and EEBLA



a) plain EEBL



b) EEBLR



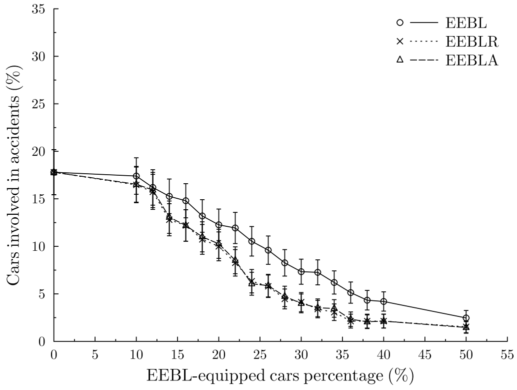
c) EEBLA

Figure 6: Average load as a function of the time and the distance from the platoon head for the five-lane scenario, average speed 130 km/h

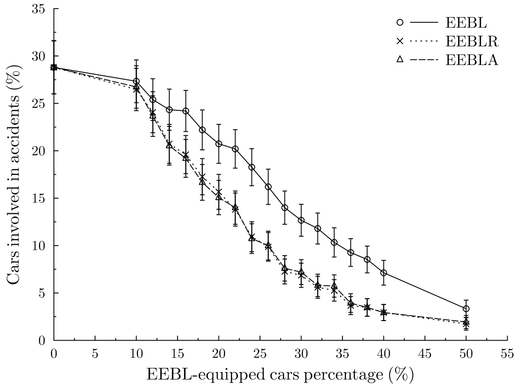
of the plots is due to the fact that cars enter the simulated stretch of highway following the cars that will simulate the emergency braking. Around the time instant 130 s all cars are on the 5 km stretch of highway and the emergency braking begins. Black area on the upper right corner of the plots is due to the platoon compacting in space as the cars come to a complete stop. Beaconing gives rise to dark blue colors equivalent to 3–5% of channel occupation when cars are moving and a light blue color equivalent to about 10% of channel occupation when they are still. With plain EEBL, the increment in load due to the emergency braking is visible but very low. EEBLR instead leads to channel congestion (more than 35% load) for more than 30 s and a stretch of highway more than 1 km; EEBLA maintains the network load much lower and the messages spread more uniformly in space, indicating that the communication pattern is more efficient.

5. APPLICATION PERFORMANCE

In Sect. 4 we evaluate the network performance of EEBL, EEBLR, and EEBLA protocols when all cars are equipped with the system, which is the most stressful situation for the network. In this case no accidents at all occur for any protocol. More interesting from the emergency braking point of view is the performance as a function of the different protocols when only a fraction of cars is equipped with EEBL capabilities.

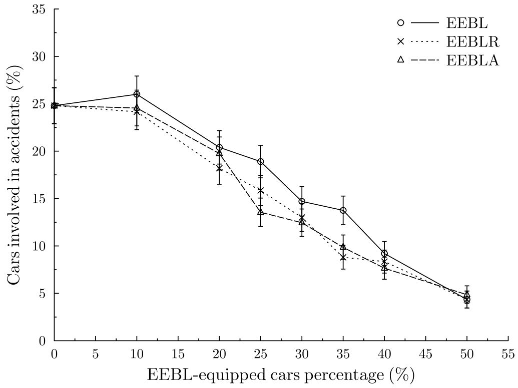


a) 130 km/h initial speed

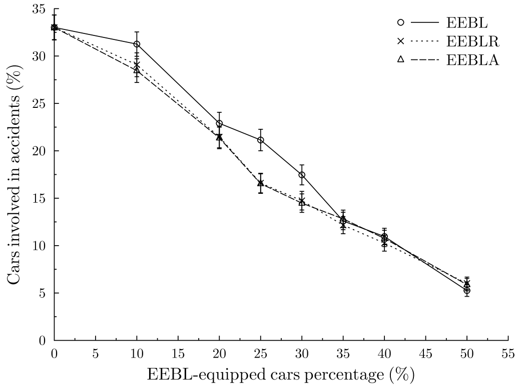


b) 150 km/h initial speed

Figure 7: Percentage of cars involved in accidents vs. MPR for single-lane tests for different protocols and different average speeds



a) two-lane scenario



b) five-lane scenario

Figure 8: Percentage of cars involved in accidents vs. MPR for multi-lane tests, average speed 130 km/h, two and five-lane scenarios

The scenario envisaged is the progressive introduction of the EEBL system on cars, so that only a percentage of them is able to communicate and react to communications: the others behave as realistic intelligent drivers, i.e., they follow the L-IDM model as defined in Sect. 3. Fig. 7 reports the fraction of cars involved in crashes as a function of the Market Penetration Rate (MPR) of EEBL systems from 0% to 50% in the basic scenario of a single lane. The upper plot refers to an average initial speed of 130 km/h, while the lower plot refers to 150 km/h. Error bars report the 95% confidence interval. Simulations for slower initial speeds yields qualitatively similar results, with an obvious reduction in accidents. Even small penetration rates (10–20%) give measurable and statistically meaningful accident reduction. The advantage of message rebroadcasting is evident: the percentage of cars involved in accidents with plain EEBL is roughly double compared to EEBLR and EEBLA protocols: only for very high MPRs the performance of plain EEBL converge to the others. The adoption of intelligent rebroadcasting seems important specially during the initial commercialization of these systems. The result can be explained observing that with a low MPR the chance of having an equipped car right ahead is marginal, so that the accident reduction is due to "early warning" received from cars not directly in front, which leads to throttling the gas, thus slowly reducing the speed and enabling more efficient braking by the L-IDM model when the car in front will eventually brake itself. This effect is amplified by the fact that drivers of unequipped cars will react to this slow deceleration generating a "wave" of early speed reduction instead of the abrupt braking typical of emergency situations.

Results for the single-lane scenario are very promising, yet the performance when there are more lanes and the network is more loaded are those of major interest. Fig. 8 analyzes the results for

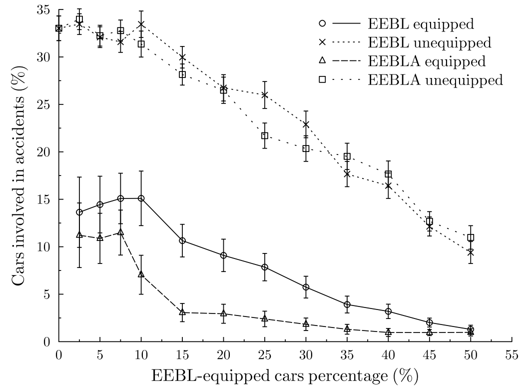


Figure 9: Split-down of equipped an unequipped cars involved in accidents vs. MPR; tests in a five lane-scenario for EEBL and EEBLA, average speed 130 km/h

2 (upper plot) and 5 (lower plot) lanes respectively. As a general comment, we observe that the system performance remains extremely positive even with low MPRs. We observe that the L-IDM model forecasts an increment of accidents as the number of lanes increases (results confirmed by the 3 and 4 lanes experiments, not reported for lack of space). At the same time EEBL systems give an increased benefit for low MPRs as the number of lanes increases, because the chance that cars communicate is increased by the higher density. EEBLR and EEBLA protocols still give an advantage at the application level, though the gain is smaller, once more due to the higher density of cars. The slight increase in average the collision rate for EEBL in the two-lane scenario is within the intrinsic randomness of different simulation runs, as proven by the highly overlapped confidence intervals.

A question that often arises is the impact of the presence of EEBL-equipped vehicles on the other lot. Fig. 9 attempts to provide an answer: also non-equipped cars benefit from the introduction of EEBL enabled vehicles, even at medium-low MPRs. The figure refers to a five-lane scenario, but results for smaller number of lanes are similar. The impact of the rebroadcast protocol (EEBLA compared to plain EEBL) on non-equipped cars is marginal, but the reduction in accident probability is unquestionable for MPR as low as 15–20%. The explanation is that the earlier and smoother deceleration of EEBL-equipped vehicles allow other drivers (the L-IDM model indeed) to react earlier with still a larger safety margin in front. Equipped vehicles have a much larger benefit as expected, and, for this category, message rebroadcast with EEBLA has a huge impact, halving the collision rate up to a 40–45% MPR. We were not able to distinguish (in simulation) between cars causing the accidents and cars involved without guilt, but we conjecture that equipped cars are (almost) never the cause of accidents. Verifying this conjecture is part of our future work.

6. IMPACT OF REVERSE TRAFFIC

So far the evaluation is limited to a single mono-directional stretch of highway, but we are well aware that the reverse direction may influence results, specially if drivers start to brake to watch what happens, or similar deprecated, but common behaviors. To this purpose we implement a modified version of the L-IDM driver behavior, whereby some drivers brake and slow down. In particular, one third of the drivers, when approaching the braking platoon (i.e., in the 100 m of road preceding the critical point), decelerate uniformly, down to a speed no lower than 70% of the average speed of the simulation (e.g., 90 km/h for the 130 km/h scenario). The deceleration is uniformly distributed between 0.5 and 2 m/s^2. Since the goal is measuring the network level performance rather than verifying if such irrational behaviors lead to collisions in the reverse direction also, we limit the study to the case of full market penetration where cars are all equipped and the network load is maximum.

Fig. 10 shows the same color plots described in Sect. 4, this time considering also negative distances because of the presence of the reverse platoon, which is depicted by the violet zone which was not previously present. From these color plots it is possible to notice that the inverse direction does not cause a big increment in the network load. In fact it seems to reduce it, but this is caused by the averaging procedure, which, due to the variety of situations generated by different braking dynamics, leads to the computation of partially misleading results. This hypothesis is strengthened by the spread in time of the high-load zone, which lasts more than observed in Fig. 6. A further confirmation comes from observing a single simulation run as in Fig. 11. Without the mediation effect, the network load for EEBLR saturates over 40%, in particular reaching a maximum peak of roughly 55% as shown in Fig. 12, which plots the CDF of maximum load $\rho_i^{\max}$.

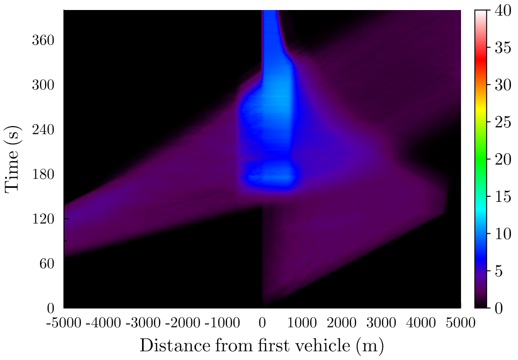
Fig. 11 needs some further explanations regarding the horizontal stripes that are visible in the upper part of the plot: these are most probably caused by the curious drivers which, when decelerating, send EEBL messages for some seconds resulting in a slight increase of the network load.

Finally we compute the l_{uf} of this scenario (Tab. 5). This time we calculate this value by considering first of all frames sent by vehicles on both directions and then only by vehicles in the direction of the braking platoon. This distinction is needed in order to compare the values with results of Fig. 5: indeed, if we consider also the reverse direction, we take into account vehicles which are distant from the "stressed" zone and not influenced by network congestion. As shown in Tab. 5, EEBLR results (considering both directions) in a l_{uf} of roughly 46%, which is less than in Fig. 5. If we take into account only forward-moving platoon, l_{uf} reaches roughly 60%, which is consistent with the results in Sect. 4. However, this result could be underestimated, since we are ignoring also vehicles in the reverse direction which are inside the "stressed" zone. The table emphasizes the fact that the reverse direction does not jeopardize the EEBL function. In general the network load is affected, but not dramatically, and that EEBLA mantains the l_{uf} to a negligible value.

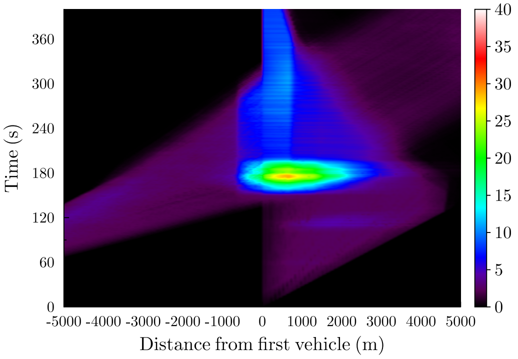
We conclude this analysis with two observations. First, we assume a 6 Mbit/s broadcast PHY speed, which, in general, is not guaranteed: if the 3 Mbit/s speed is used, the network load would be much higher, and probably the EEBL application can suffer. Similarly, highways can have more than 5 lanes per direction, though this should not be really a limitation. The car density we consider, 14 car per km at 130 km/h, is conservative. Second, the 10 Hz message frequency seems excessive for the physical dynamics and inertia of cars, so that dynamic adaptive protocols can be studied, as well as dynamic strategies for EEBLA rebroadcasting, for instance reducing the TTL if further studies will highlight that this setting is not necessary.

Direction	EEBL	EEBLR	EEBLA
Both	0.04%	46.08%	0.07%
Forward only	0.03%	59.97%	0.11%

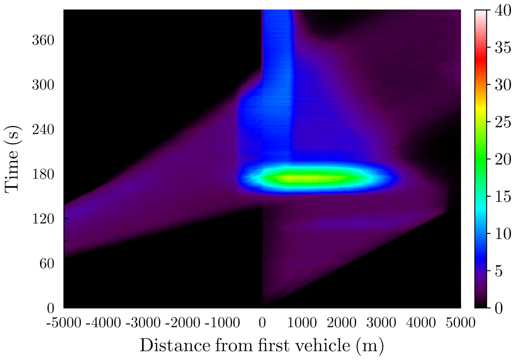
Table 5: l_{uf} for plain EEBL, EEBLR and EEBLA for the ten-lane scenario, average speed 130 km/h



a) plain EEBL

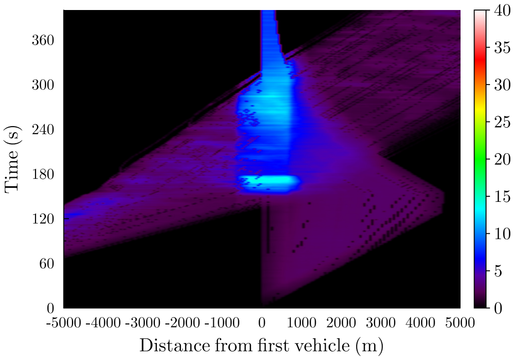


b) EEBLR

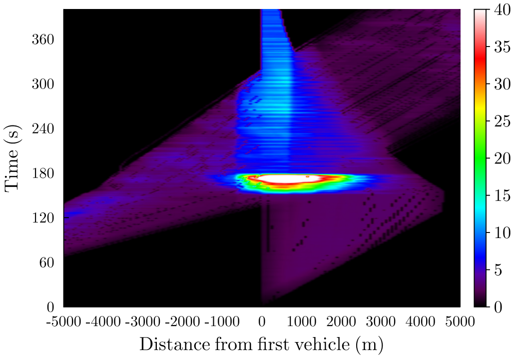


c) EEBLA

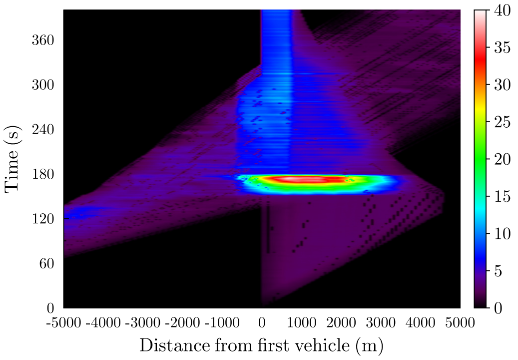
Figure 10: Average loads as a function of the distance from the platoon head for the ten-lane scenario, average speed 130 km/h



a) plain EEBL



b) EEBLR



c) EEBLA

Figure 11: Network loads as a function of the distance from the platoon head for a single ten-lane scenario, average speed 130 km/h

7. CONCLUSIONS AND FUTURE WORK

In this work we present a simulation study of emergency braking applications developing a sophisticated mobility, cruise control and driver behavior model fully embedded into the ns-3 package. To the best of our knowledge this is the first time that a safety application is studied with this level of realism and detail, albeit still via simulations.

The time-space analysis of the channel load is novel, as it gives insight in the dynamics of the joint network and application evolution.

The results in this work are extremely promising. They stress the need for proper application-level message aggregation strategies to avoid clogging the network in case of high vehicular density. The aggregation technique we present is interesting for its extreme simplicity and effectiveness, reaching almost the ideal performance of only doubling the channel load at each receiver station.

Furthermore, the market penetration rate analysis indicates that benefits are obtained for penetration rates as low as 5–10%, and, most notably, that also cars not equipped with cruise control and communication devices benefit from the presence of cars whose reaction is smoother and anticipated with respect to the standard human reaction.

Further work includes the refinement of the channel model, the further development of the vehicular dynamics models, and, most of all, the exploration of scenarios more complex than a stretch of highway, albeit the large scale emergency braking support we analyze is mostly needed in these cases, specially with low visibility conditions. Moreover, an efficient ACC automatic braking is a mandatory component for cooperative driving, leading to a more efficient use of vehicles and roads.

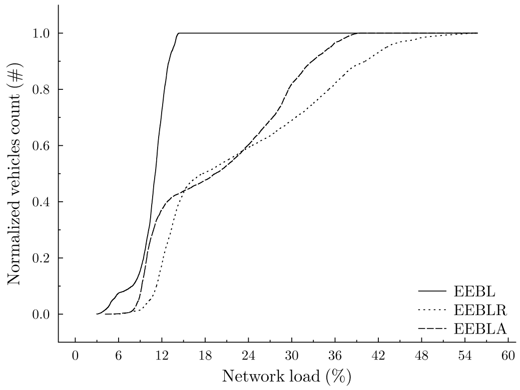


Figure 12: Maximum channel load $\rho_i^{\max}$ observed by each vehicle during the simulation for the ten-lane scenario for plain EEBL, EEBLR and EEBLA

Acknowledgment

The Autonomous Province of Trento (PAT) partially funded the research that led to this paper.

8. REFERENCES

[1] Vehicle safety communications project task 3 final report. Technical report, U.S. Department of Transportation, National Highway Traffic Safety Administration, Mar. 2005.

[2] 2010 Aftermarket Brake Pad Evaluation. Technical report, Michigan State Police Precision Driving Unit, Nov. 2010.

[3] H. Arbabi and M. C. Weigle. Highway Mobility and Vehicular Ad-Hoc Networks in ns-3. In *Winter Simulation Conference (WSC)*, Dec. 2010.

[4] A. Bachir and A. Benslimane. A multicast protocol in ad hoc networks inter-vehicle geocast. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, volume 4, pages 2456 – 2460, Apr. 2003.

[5] G. Batchelor. *An introduction to fluid dynamics*. Cambridge mathematical library. Cambridge University Press, 2000.

[6] L. Briesemeister, L. Schafers, and G. Hommel. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 522 –527, 2000.

[7] W.-R. Chang, H.-T. Lin, and B.-X. Chen. TrafficGather: An Efficient and Scalable Data Collection Protocol for Vehicular Ad Hoc Networks. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 365 –369, Jan. 2008.

[8] J. J. Haas and Y.-C. Hu. Communication requirements for crash avoidance. In *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*, VANET '10, pages 1–10, New York, NY, USA, 2010. ACM.

[9] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1707 – 1716, 2002.

[10] K. Ibrahim and M. Weigle. CASCADE: Cluster-Based Accurate Syntactic Compression of Aggregated Data in VANETs. In *GLOBECOM Workshops, 2008 IEEE*, pages 1 –10, Dec. 2008.

[11] D. Jiang, Q. Chen, and L. Delgrossi. Optimal data rate selection for vehicle safety communications. In *VANET '08: Fifth ACM international workshop on VehiculAr Inter-NETworking*, pages 30–38, 2008.

[12] G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *VANET '04: 1st ACM international workshop on Vehicular ad hoc networks*, pages 76–85, 2004.

[13] M. Li and W. Lou. Opportunistic broadcast of emergency messages in vehicular ad hoc networks with unreliable links. In *QShine '08: 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 1–7, 2008.

[14] M. Nekoui and H. Pishro-nik. Fundamental tradeoffs in vehicular ad hoc networks. In *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*, VANET '10, pages 91–96, New York, NY, USA, 2010. ACM.

[15] 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: 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, 1999.

[16] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse. MISTRAL: efficient flooding in mobile ad-hoc networks. In *MobiHoc '06: 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 1–12, 2006.

[17] G. Schultz and M. Babinchak. Final Report For The Methodology Study Of The Consumer Braking Information Initiative. Technical report, U.S. Department of Transportation, Mar. 1999.

[18] M. Sepulcre and J. Gozalvez. On the Importance of Application Requirements in Cooperative Vehicular Communications. In *Eighth International Conference on Wireless On-Demand Network Systems and Services*, 2011.

[19] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar. Broadcasting in VANET. In *2007 Mobile Networking for Vehicular Environments*, pages 7 –12, May. 2007.

[20] O. Tonguz, N. Wisitpongphan, J. Parikh, F. Bai, P. Mudalige, and V. Sadekar. On the Broadcast Storm Problem in Ad hoc Wireless Networks. In *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pages 1–11, Oct. 2006.

[21] M. Treiber and D. Helbing. Realistische Mikrosimulation von Strassenverkehr mit einem einfachen Modell. *16th Symposium Simulationstechnik ASIM*, 2002.

[22] M. Treiber, A. Hennecke, and D. Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *PHYSICAL REVIEW E*, 62:1805, 2000.

[23] M. Treiber, A. Kesting, and D. Helbing. Delays, Inaccuracies and Anticipation in Microscopic Traffic Models. *PHYSICA A*, 360:71, 2006.

[24] B. van Arem, C. van Driel, and R. Visser. The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):429 –436, Dec. 2006.

[25] Y. Zang, L. Stibor, H.-J. Reumerman, and H. Chen. Wireless local danger warning using inter-vehicle communications in highway scenarios. In *Wireless Conference, 2008. EW 2008. 14th European*, pages 1 –7, Jun. 2008.