Modeling and Simulation of WAVE 1609.4-based Multi-channel Vehicular Ad Hoc Networks

Ali J. Ghandour
Department of Electrical and Computer Engineering
American University of Beirut
ajg04@aub.edu.lb

Marco Di Felice
Department of Computer Science
University of Bologna
difelice@cs.unibo.it

Hassan Artail
Department of Electrical and Computer Engineering
American University of Beirut
hartail@aub.edu.lb

Luciano Bononi
Department of Computer Science
University of Bologna
bononi@cs.unibo.it

ABSTRACT

Recently, the IEEE 1609.4 Standard for Wireless Access in Vehicular Environments (WAVE) has been proposed to enhance the performance of vehicular networks with multi-channel operations that allow for the coexistence of safety-related and non-safety related vehicular applications. However, while the benefits of the multi-channel approach are clear, the impact of the IEEE 1609.4 channel scheduler on the performance of delay-constrained vehicular applications remains to be well explored by researchers. At present, the evaluation of 1609.4-based Vehicular Ad Hoc Networks (VANETs) constitutes an open issue due to the lack of simulation tools that can provide a complete modeling of the IEEE WAVE 802.11p/1609.4 stack. In this paper, we provide three key contributions pertaining to multi-channel VANETs. First, we describe our implementation of the IEEE 1609.4 protocol in the ns2 simulator, and we detail its current integration with the existing ns2 implementation of the 802.11p MAC protocol. Second, using our simulation model we propose an evaluation study of 1609.4-based VANETs, and we show that the tight channel synchronization issues foreseen by the protocol might have a dramatic impact on the performance of safety-related applications with strict delivery ratio and delay requirements. Third, we propose two new enhancements for the WAVE protocol stack to favor the dissemination of safety messages in multi-channel VANETs. The suggested algorithms are shown to greatly improve packet delivery ratio and delay of safety applications in single and multi-hop topologies, while preserving the synchronization scheme of the IEEE 1609.4 protocol.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design - Wireless Communication.

Simutools 2012, March 19-23, Desenzano del Garda, Italy

DOI 10.4108/icst.simutools.2012.247743

General Terms

Performance, Design, Experimentation.

Keywords

Vehicular Ad Hoc Networks, WAVE 1609.4 Protocol, Modeling and Simulation, Performance Evaluation.

1. INTRODUCTION

Collision Warning systems [15], Local Trip Advisors [7], multimedia systems for driver comfort and entertainment [12] are just few examples of applications proposed for vehicular networks in these last years. While significant differences might exist between different classes of vehicular applications (e.g. safety-related vs infotainment applications) in terms of Quality-of-Service (QoS) requirements and relevance of the communication [20], it is likely that in a realistic deployment of vehicular networks multiple applications will operate on the same vehicle and contend for channel access over the same area. For these reasons, the IEEE P1609 Working Group in charge of the standardization of the Wireless Access in Vehicular Environment (WAVE) stack has released several protocols for vehicular communication that take into account the co-existence problem of vehicular applications with different classes and QoS requirements. Based on these protocols, the service class differentiation is implemented at the Physical (PHY) layer through the definition of a dedicated spectrum band for vehicle-to-vehicle and vehicle-to-infrastructure known as Dedicated Short Range Communication (DSRC), and the realization of seven channels reserved for different applications class/usage. On each particular channel, the channel differentiation among different Access Categories (AC) is achieved at the MAC layer through the EDCA parameters defined in the IEEE 802.11p MAC protocol [19]. Moreover, multi-channel operations are regulated by the IEEE 1609.4 WAVE protocol [14], that divides in both time and frequency the operations of safety-related and non-safety-related vehicular applications. Based on the 1609.4 scheme, each vehicle periodically switches and alternates between Control CHannel Interval (CCHI) and Service CHannel Interval (SCHI) of default length equal to 50 ms [14]. During the CCH interval, vehicles are tuned to the control channel (CCH) and can transmit routine `HELLO` messages or event-driven safety messages. Moreover, WAVE Service Advertisement (WSA) are exchanged during CCHI to publicize services to be offered on the coming SCHI. During SCH intervals, vehicles can access the remaining six services channels, and be engaged in various non-safety applications like multimedia exchange, peer-to-peer sharing, and many others.

While the benefits of the multi-channel approach in terms of capacity increasing are well known in the literature of wireless systems, there are some concerns on the impact of the strict channel synchronization imposed by the IEEE 1609.4 protocol on the QoS requirements (packet delivery ratio and delay) of safety and non-safety applications [16][18][8]. For instance, in [16], the authors investigate the performance of IEEE 802.11p networks through an analytical model, and demonstrate that the synchronization of backoff processes at the beginning of each CCH/SCH interval might easily produce an increase of the probability of collisions at the MAC layer. This probability is even higher for high-priority messages due to the small size of the backoff window, as shown in [8]. Current research on multi-channel vehicular networks has mainly focused on techniques to increase the available capacity through the utilization of dedicated radios for safety applications [11], or on determining the optimal trade-off between CCHI and SCHI, by considering the traffic conditions on each channel [16][18]. However, most of the existing work in the area of VANETs do not provide proper simulation studies, due to the lack of simulation tools that can provide a complete modeling of the IEEE 802.11p/1609.4 stack. To the best of our knowledge, [13] and [11] provide the only (actually the same) available simulation model of the IEEE 1609.4 protocol in network simulator 2 (ns2). The proposed implementation in these papers is quite limited and does not allow to realistically model multi-channel operations. Transmissions are only possible during CCHI and the presented model does not allow for simulations of SCH applications.
Nevertheless, we are aware of an available implementation for the DSRC/WAVE proposed by the release version 5 of the NCTUns tool [21]. However, given the popularity of ns2 among researchers in wireless networking, and in particular ITS, we opted to develop an implementation of the IEEE 1609.4 protocol for that environment. We believe that this will be especially welcomed by researchers who already have ongoing work in ns2 and therefore could use our implementation to further this work.

In this paper, we address the modeling and simulation of 1609.4-based Multi-channel Vehicular Ad Hoc Networks with three novel contributions. First, we describe with details our implementation of the IEEE 1609.4 protocol for the ns2. Our proposed implementation can be easily integrated with the existing implementation of the 802.11 MAC protocol suggested in [6]. We validate our implementation through an analytical model that provides an upper-bound on the packet delivery delay of safety broadcast messages in multi-channel environment. Second, we investigate the performance of 1609.4 protocol, and we show that the periodic switching between CCH/SCH intervals might have a negative impact on the QoS parameters of delivery and delay-constrained applications. Third, we propose two enhancements to the existing IEEE 1609.4 protocol in order to: (*i*) minimize the delivery delay of safety-related messages and (*ii*) mitigate the collisions caused by synchronous channel access at the beginning of each CCH/SCH interval.
Contrary to the previously mentioned solutions [18][8], our proposed enhancements do not require modifications to the existing IEEE 1609.4 protocol, and do not assume any specific collaboration/scheduling of messages at the application layer.

The rest of this paper is organized as follows. In Section 2, we provide an overview of the IEEE 1609.4 protocol. In Section 3, we provide a detailed description of our implementation of the IEEE 1609.4 model for the ns2 simulator. In Section 4, we validate our implementation through comparison with an analytical model. Also, we investigate the performance and QoS parameters of multi-channel VANETs. In Section 5, we propose two protocol enhancements for the IEEE 1609.4. Finally we conclude the paper and suggest future work in Section 6.

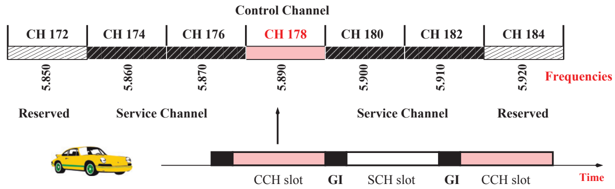


Figure 1: The DSRC spectrum frequencies and the IEEE 1609.4 WAVE protocol operations.

2. THE IEEE 1609.4 PROTOCOL

The IEEE 1609.4 standard [14] has been proposed in 2010 (as a revision of the previous 1609.4-2006 standard) to enhance the underlying 802.11p MAC protocol with multi-channel operations over a single-radio transceiver. The channels used by the IEEE 1609.4 standard are allocated in the DSRC band which has been reserved for vehicular communications in both US and Europe. Seven 10 MHz channels are available, divided into one control channel and six service channels. Figure 1 shows the current frequencies used in US, where the DSRC band is also known as Intelligent Transportation Systems (ITS) Radio. It is assumed that all vehicles maintain strict synchronization with the Coordinated Universal Time (UTC) that can be acquired from Global Positioning System (GPS) devices or from other vehicles. Based on these time information, vehicles synchronously switch between Control Channel (CCH) interval and Service Channel (SCH) interval, as shown in Figure 1. All devices are required to monitor the control channel at CCH intervals for safety-related messages, or advertisement messages about services that will be available during the next SCH interval. During SCH interval, vehicles can switch among the remaining six channels and transfer data of non-safety applications. The SYNChronization (SYNC) interval is simply the summation of the CCH interval and SCH interval. Sync Interval is 100 ms length and default values for control and service channel intervals are 50 ms [14]. Guard intervals are introduced at the start of each interval to minimize the effect of timing inaccuracies. Typical values for the Guard interval ranges from 4 to 6 ms [13]. During the Guard interval, transmissions are not allowed. At the beginning of each interval, previous MAC activities are suspended and the corresponding ones are started or resumed, thus ensuring that each packet is transmitted on the correct channel. For instance, a vehicle interested in an audio file transfer service will switch to the corresponding service channel to start receiving the file. If the transfer takes too long to complete, the vehicle must switch to the control channel to receive safety messages and then switch back to the service channel to resume the file transfer. Thus, using the multi-channel coordination, a vehicle can periodically monitor the control channel for safety messages while it continues to use available infotainments services in the network.
It is worthy to note here that the current IEEE 1609.4-2010 standard does not define the policy to adopt for scheduled messages that were not transmitted at the end of a given interval. Some sug-

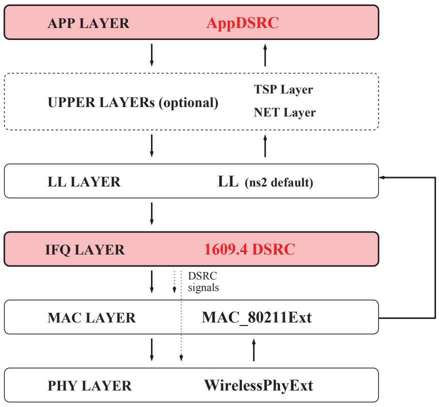


Figure 2: The revised Ns2.34 architecture.

gestions are provided in Annex C of the standard [14] for avoiding scheduled transmissions during guard interval. Moreover other details were also not defined in the current 1609.4 standard, or left open to user-implementations, such as the channel selector policy to adopt to decide on the channel to be served at each SCH interval when the vehicle is involved with multiple services running at different channels.

3. MODEL IMPLEMENTATION

Figure 2 shows the architecture of the revised ns2 (version 2.34) tool with our modifications to model the multi-channel WAVE protocol. We implemented the model of the IEEE 1609.4 protocol at the InterFace priority Queue (IFQ) layer (i.e. between the MAC and NET layers of ns2), as a subclass of the class `Queue`. Moreover, we integrated it with the current ns2 implementation of the IEEE 802.11 protocol at the MAC layer (i.e. the `Mac802_11Ext` class) [6] and the revised wireless propagation model at the PHY layer (i.e. the `WirelessPhyExt` class) [6]. We defined seven different packets queues, one for each channel in the DSRC band. Moreover, we added a field `channel_` in the common header of each packet. The channel information are inserted by the application layer, and are used at the IFQ layer to identify the correct queue to select. We implemented a Timer (called `DSRCTimer`) to account for the presence of different logical slots (slot and interval are used interchangeably in this paper). Once the Timer expires, one of three handlers is called based on the `next_slot_` to be scheduled. If the next slot is a CCH slot, then `handle_CCH_slot()` is invoked, and `DSRCTimer` is started for a duration equal to the `CCH_INTERVAL` (which is set to 46ms). Then, packets are dequeued from the CCH queue and sent to the MAC layer to be transmitted at the appropriate channel. Analogously, if the next slot is a SCH slot, then `handle_SCH_slot()` is invoked, and `DSRCTimer` is started for a duration equal to the `SCH_INTERVAL` (default value is 46ms). The channel used during the SCH slot is referenced by the `current_channel_` variable, that is decided by the `ChannelScheduler` class through the `get_next _channel()` method.

We implemented different policies for the channel scheduler as subclass of the `ChannelScheduler` class. The most intuitive policy is the `RoundRobinScheduler` class that switches among the active service channels in a round robin way. The `EDFScheduler` class always chooses the channel with the lowest average earliest deadline of the enqueued packets. Additional, fine-grained QoS-aware scheduling policies can be implemented in our model by extending the `ChannelScheduler` class. Thus, based on the value of `current_channel_`, a packet is dequeued from the appropriate queue and sent to the MAC layer to be transmitted. If the next slot is a Guard interval slot, then `handle_GI_slot()` is invoked, and `DSRCTimer` is started with a duration equal to the `GI_INTERVAL` (which is set to 4ms by default). The current IEEE 1609.4 protocol imposes that no transmission can occur during the Guard interval, but it does not specify the behavior to adopt for ongoing transmissions that were aborted at the start of the guard interval. In our model, we used signalling functions at both the MAC and PHY layer to notify the starting of a Guard interval. When the `MAC_notify_GI()` method is invoked, we check the status of the backoff procedure and we cancel any running back-off attempt. At the PHY layer, `PHY_notify_GI()` checks for any ongoing packet reception and then raises the error flag; thus the packet is dropped at the MAC layer.

Moreover, we implemented two different recovery strategies for packets present in the queue at the end of a given interval and the start of the guard interval. Obviously these packets did not get chance to access the medium during the corresponding interval and thus we suggest two strategies to deal with them: (*i*) a `PURGE` strategy, in which the packet is simply discarded at the MAC layer and (*ii*) a `REINSERT` strategy in which the packet is re-inserted at the head of the corresponding queue at the IFQ layer, and contended for transmission during the next CCH (if it was a safety-related message) or SCH (if it was a service message) slots. Researchers can suggest other strategies to be adopted. In Section 4, we evaluated the performance of both strategies in terms of delivery rate and delay of safety-related messages.
Moreover, in order to evaluate the performance of multi-channel DSRC-based VANETs, we implemented a model of a vehicular application that can be attached to the existing DSRC stack and generate data to be transmitted over the wireless channel. To this aim, we created a new `DRSCApp` class as a subclass of the ns2 `Agent` class. Each `DRSCApp` object has several parameters that can be set from the OTCL file, including the packet rate of the applications, the type of service (e.g. safety-related or non-safety), the application identifier, the channel to be used, in addition to other parameters. Once `DRSCApp` receives a message originated from another vehicle, it will log and discard the message, or re-broadcast it in case of multi-hop dissemination. To this aim, we implemented two possible message reception handlers: (*i*) *single-hop* dissemination, in which each packet received is logged and immediately discarded and (*ii*) *flooding* dissemination, in which each packet received is checked and, in case it has not been seen before by the current node, it is re-broadcasted till it has covered a maximum life distance (referred as Spatial Horizon in [2]) from the source node. The first strategy is suitable to model the behavior of periodic routine HELLO beacons and also WSA service advertisements. The latter strategy is suitable to model safety-related applications that need to disseminate alert messages over multi-hops (e.g. the scenarios discussed in [15], [2]). In Section 4, we evaluated the performance of the IEEE 1609.4 protocol under both single-hop and multi-hop dissemination. We highlight here that further user-defined dissemination algorithms can be implemented by overloading the `recv()` method of the `DRSCApp` class. The class diagram of the extended ns2.34 architecture is shown in Figure 3.

4. ANALYSIS AND VALIDATION

In this section, we first validate the correctness of our implementation described in Section 3 through a simplified analytical model of the IEEE 1609.4/802.11p standards. We then investigate

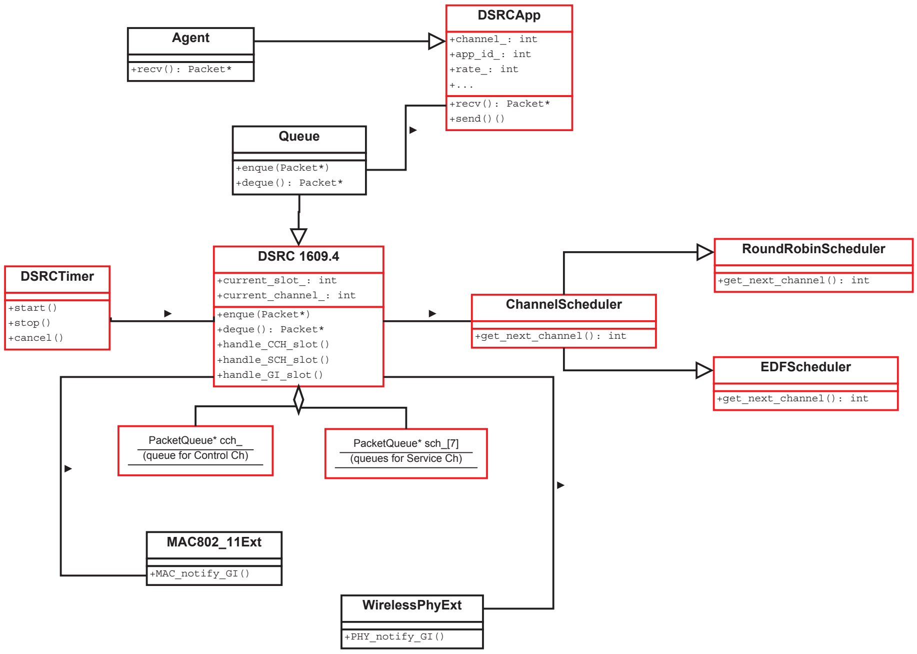


Figure 3: The class diagram of the IEEE 1609.4 implementation in ns2.

the performance of safety-application in multi-channel DSRC environment through the analysis of the delivery ratio and delay over a realistic highway multi-lanes scenario.

In this analysis, we attempt to derive an upper bound on the delivery delay of safety-related messages. Conversely to previous studies [5], we assume that each vehicle will generate exactly 1 safety-related message during the SCH interval, which constitutes the worst-case scenario from the delivery delay point of view. Moreover, we assume that: (*i*) un-transmitted messages are discarded at the end of the CCH slot (i.e. the `PURGE` policy discussed previously is used) (*ii*) messages are transmitted in broadcast mode without any acknowledgements (*iii*) all the safety messages belong to the same EDCA priority class (*iv*) all messages have the same payload size S and finally (*v*) and finally vehicles are in the same transmission area and no hidden node problem occurs.
The average delay required to transmit a safety message can be expressed as the sum of the average queuing delay $E[q]$, the average contention delay $E[c]$ on the control channel (since no transmission for safety messages happens on the SCH), the average transmission delay, and the average propagation delay. Transmission delay is assumed to be equal and constant for all packets based on assumption (*iv*) and thus omitted in this analysis. Moreover, propagation delay is safely assumed to be negligible. Thus, for the scope of this work, we focus on the average `Access Delay` which accounts for delays on the upper and lower MAC layer. Delay and Access Delay will be used interchangeably on this work. Hence, we have the following equation for average delay $E[d]$:

$$E[d] = E[c] + E[q] \tag{1}$$

Since we assume that each safety message is generated during the SCH slot and then transmitted during the CCH slot (worst-case analysis), then $E[q]$ can be computed as:

$$E[q] = \frac{SCH_d}{2} + GI_d \tag{2}$$

where SCH_d and GI_d are the length of the SCH and the Guard Intervals (in seconds) respectively. Let τ be the probability to transmit in a given slot. If we assume a uniform probability distribution to select a slot within the current Contention Window (CW), then τ can be derived as [4]:

$$\tau = \frac{1}{E[CW]+1} = \frac{2}{CW_{max}+1} \tag{3}$$

where CW_{max} is the maximum size of CW for broadcast messages. Let p_{idle} be the probability that a channel is idle in a given slot, and p_{busy} its converse. Similarly, let $p_{success}$ be the probability that a slot is occupied by a successful transmission, and p_{coll} the probability that a collision occurs during a slot. If we assume a scenario with M nodes with the above mentioned assumptions, it is easy to verify that p_{idle}, p_{busy}, $p_{success}$ and p_{coll} can be computed as:

$$p_{idle} = (1-\tau)^M \tag{4}$$

$$p_{busy} = 1 - p_{idle} \tag{5}$$

$$p_{success} = M \cdot \tau \cdot (1-\tau)^{M-1} \tag{6}$$

$$p_{coll} = 1 - p_{idle} - p_{success} \tag{7}$$

The average delay $E[c]$ can be expressed as a function of the average Contention Window size $E[CW]$ and the average duration of each logical slot T_{slot}:

$$E[c] = E[CW] \cdot T_{slot} = \frac{CW_{max}-1}{2} \cdot T_{slot} \tag{8}$$

The average duration of the logical slot T_{slot} can be derived as proposed in [1]:

$$T_{slot} = (1-p_{busy}) \cdot \sigma + T_{success} \cdot p_{success} + T_{coll} \cdot p_{coll} \tag{9}$$

where σ is the duration of an empty slot according to the MAC 802.11p [19]. $T_{success}$ is the time required for a successful transmission and T_{coll} is the average time of a collision event. Based on the message size (S), the transmission time of the preamble (T_{PRE}) and the data-rate used (R) on the control channel, the exact

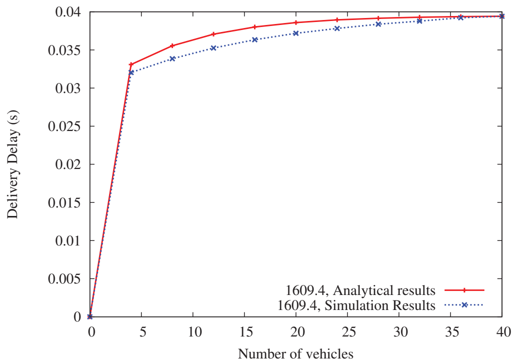


Figure 4: Analytical vs Simulation delivery delay.

Table 1: Simulation Parameters

Street Length	1 Km
Number of lanes	4
Number of channels (K)	7
Max Transmission range	250 mt
Channel Data Rate (R)	3 Mb/s
CCH slot duration	50ms
SCH slot duration	50ms
Guard Interval (GI) duration	4 ms
Simulation Time	20 seconds
Number of simulation runs	200
Confidence Interval accuracy	95%

value of $T_{success}$ and T_{coll} can be derived as:

$$T_{success} = DIFS + \sigma + \frac{S}{R} + T_{PRE} \quad (10)$$

$$T_{coll} = EIFS + \sigma + \frac{S}{R} + T_{PRE} \quad (11)$$

Figure 4 shows the delivery delay computed through our suggested simulation model and through Equation 1. From Figure 4, we can appreciate that the simulation results perfectly match the analytical results, thus validating the correctness of our implementation.

In Figures 5(a), 5(b) and 5(c), we investigate the performance of safety-related applications in multi-channel DSRC environment, and we analyze the current drawbacks of the IEEE 1609.4 protocol. Unless specified otherwise, we consider the configuration of network parameters shown in Table 1. We consider two different classes of vehicular applications in our simulation study:

- *Application I*) In Figures 5(a), 5(b), we analyze the performance of a vehicular applications that periodically broadcasts `HELLO` messages on the CCH slot, with a frequency of 10 Hz (i.e. one message for every CCH+SCH intervals). This might be the case of an application that periodically advertises the current vehicle's location and speed information to neighbour vehicles, in order to implement cooperative driving strategies (like the Cooperative Travelers Assistance (CTA) system proposed in [7]). However, it is worthy to mention that the content of the `HELLO` messages is application-dependant, and is out of the scope of our analysis.

- *Application II*) In Figure 5(c), we consider a safety application that broadcasts `SAFETY` messages on the CCH when detecting specific events along the path (e.g. car accident). The set of receivers is specified as the group of vehicles that are within a Risk-Zone (RZ) distance from the sender vehicle (in our case, we set this distance to 1km). Every vehicle receiving the `SAFETY` message will re-broadcast it, unless it is a duplicate message or the vehicle is outside the RZ. Again, we are not interested in the dissemination scheme to adopt (we assume a simple flooding technique), but rather we focus on the impact caused by the multi-channel environment on the multi-hop delivery delay of `SAFETY` messages.

Figure 5(a) shows the packet delivery ratio as a function of the number of vehicles in the current scenario, where *Application I* is considered. Every vehicle generates exactly 1 `HELLO` message every 100 ms, at a random instant of time. We evaluated three different DSRC stack configurations:

- *CCH/SCH Switch OFF*: We considered a scenario in which all the vehicles are always tuned on the CCH channel, and no channel switch occurs. This configuration reflects the current status of the stack used in ns2 by most researchers to simulate VANETs environment. We used this configuration as a term of comparison to analyze the impact of the strict synchronization and periodic channel switching imposed by the IEEE 1609.4 protocol.

- *CCH/SCH Switch ON - Purge*: We evaluated the normal behavior of the IEEE 1609.4 protocol, according to which all vehicles must perform periodic switching between CCH and SCH intervals. In case the CCH interval expires before the `HELLO` message has been transmitted, the message is discarded at the MAC Layer (PURGE policy).

- *CCH/SCH Switch ON - Reinsert*: This configuration is close to the previous one where all the vehicles must perform periodic switching between CCH and SCH slots. However, in case the CCH interval expires before the `HELLO` message has been transmitted, the message is enqueued and transmitted at the start of the next CCH slot (REINSERT policy).

Figure 5(a) shows that both "Switch ON" configurations experience much higher message losses than the currently used "Switch OFF" configuration. In fact, Packet Delivery Radio (PDR) degrades dramatically for both Switch ON configurations with the increase in number of vehicles. This fact can be explained with the following observations: (*i*) broadcast messages are transmitted using the minimum Contention Window size (CW) at the MAC layer and are not acknowledged and (*ii*) for all messages generated during the SCH interval, the corresponding vehicles will contend for the channel at the beginning of the next CCH interval, thus incurring the risk of synchronous channel access due to the reduced size of the CW. Moreover, Figure 5(a) shows that no appreciable difference can be seen between the PURGE and RETRANSMIT policies. This result needs to be explored in future work.

Figure 5(b) confirms the results of the previous analysis, by showing the delivery delay (on the x-$y1$ axis) and PDR (on the x-$y2$ axis) for the Switch ON - PURGE configuration. Here, we considered two sub-configurations of *Application I*:

- *Sub-Configuration 1*: All vehicles generate the `HELLO` message at a random interval during the SCH interval.

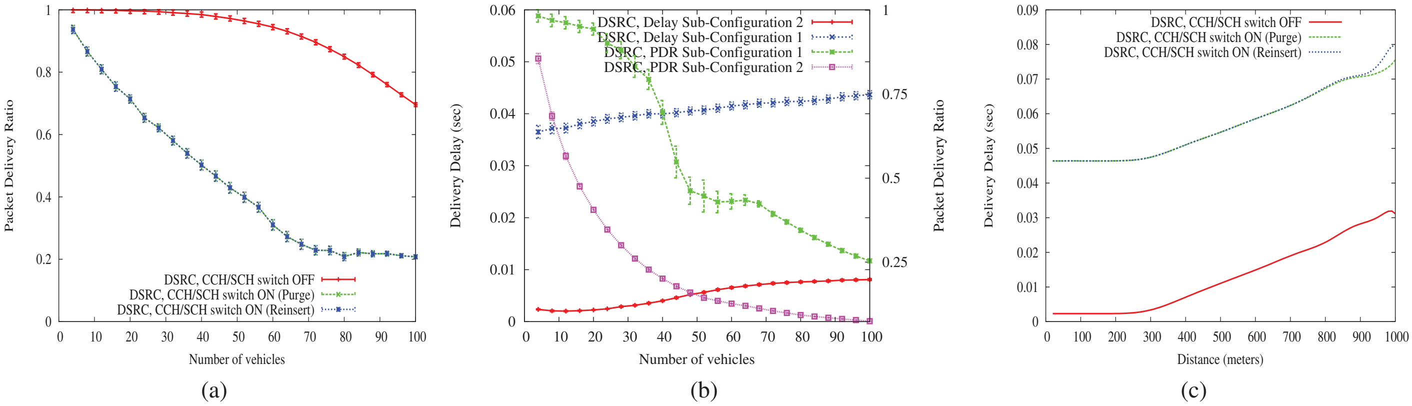


Figure 5: The delivery rate for Application I is shown in Figure 5(a). The delivery rate and delay for two different sub-configurations of Application I is shown in Figure 5(b). The delivery delay as a function of the distance for the Application II is shown in Figure 5(c).

- *Sub-Configuration 2*: All vehicles generate the HELLO message at a random interval during the CCH interval.

In *Sub-Configuration 1*, vehicles are unable to immediately broadcast the message since they are tuned on a service channel, and must wait till the start of next the CCH interval before performing the transmission attempt. This will cause additional delivery delay (caused by the queueing delay expressed by Equation 2) and reduced delivery ratio, caused by the synchronous channel access problem described above. Thus, these results confirm that the message generation time at the application layer might have a direct impact on the performance of the application itself. To this aim, the authors of [13] propose to synchronize the generation of messages at the application layer with the channel scheduling operations performed by the IEEE 1609.4 protocol. Figure 5(b) shows that this solution might produce a significant performance gain both in terms of delivery delay and ratio. However, this requires to modify the implementation of vehicular applications, and make them aware of lower layers status which might be a non-feasible solution for large class of vehicular applications suggested so far. In Section 5, we propose an alternative solution that is found to increase the delivery ratio without assuming any specific message scheduling from the application layer.

Figure 5(c) shows the delivery delay for SAFETY message generated by *Application II*, as a function of the distance traveled by each message. Thus, Figure 5(c) provides an indicator of how fast the SAFETY message is propagated over multi-hops VANET. Again, we compared the performance of the three different DSRC stack configurations described above. As expected, both Switch ON configurations suffer of higher delivery delay compared to the Switch OFF configuration. Figure 5(c) shows again that no appreciable difference can be seen between the PURGE and RETRANSMIT policies. Also, Figure 5(c) demonstrates that the problem of periodic CCH/SCH slot switching is exacerbated in a multi-hop environment, where a SAFETY message might experience additional delay at each hop/retransmission because of the expiration of the CCH interval. The additional delay might constitute a severe problem for safety applications with strict delay bound requirements, as the one described in [15]. For this reason again, in Section 5 we propose a solution that is demonstrated to greatly reduce the delivery delay of SAFETY messages transmitted on the control channel, and also without requiring any changes to the IEEE 1609.4 multi-channel operations.

5. PROTOCOL ENHANCEMENTS

We propose here two enhancements to the IEEE 1609.4/802.11p stack in order to reduce the delivery delay of safety messages in multi-channel VANETs and solve the problem of synchronous transmissions at the start of the CCH/SCH slots. Both these solutions are compliant with the IEEE 1609 operations, and do not require significant changes to the IEEE 802.11 MAC protocol.

Enhancement1 (EN1). It is shown in [10] that the increase in data traffic will render the 50ms CCH slot capacity not enough to transmit all the packets generated during this interval. At the end of the interval, some packets will be available in the WAVE queue of the nodes. We will refer to these packets as the un-transmitted packets. The un-transmitted packets should wait till the next control channel interval and thus will witness an additional delay of 50 ms. We simulate a simple scenario with bitrate equals 3Mbps, payload size is 800 bytes and number of cars is 100. Each car broadcasts one HELLO message during the 50ms control channel interval. In fact, it is suggested in [17] that the message size would reach 800 bytes after adding security features to it. The percentage of un-transmitted packets in the scenario is 44.18%, which means that almost half of the packets did not make it to the physical medium. We increased the bitrate to 6Mbps in the simulation scenario. As a result, the percentage of un-transmitted packets falls to 18.78%, but remains very large to tolerate in a safety-related system. Thus, un-transmitted packets will witness an additional delay specific for multi-channel WAVE environment known as $T_{untrans}$ in [10]. Another possible scenario where packets will encounter an additional $T_{untrans}$ delay is when the packet is generated during service channel interval. The probability of a safety situation to emerge at the road during SCH slot is $\frac{1}{2}$ according to the time synchronization mechanism described in Section 2. In this case, the car will need to wait until the next CCH interval to be to attempt to transmit the packet. Other cars might also have safety packets waiting for transmission in the queue. This will result in a high probability of synchronized collisions at the start of the control channel interval [13]. The problem of synchronized collision will be discussed in details in *Enhancement2*.

We propose in *EN1* a modification to the existing WAVE protocol stack to enable transmission of high priority safety messages during SCH interval. The suggested system enhances safety road conditions by alerting drivers about critical safety situations with shorter delay which results in faster drivers' reaction. The only change we made to the standard is to introduce the following rule: whenever a car decides to switch from the control channel to any of the service channels, it should inform its neighbors. Since vehicles are periodically exchanging HELLO messages with a rate of 10 or 20 Hz, we added an additional entry in each HELLO message. This entry is set to the number of the service channel the vehicle will move to. This simple regulation will permit each vehicle to have a list of the channels where all its neighbors are available during service channel slot. It is worthy to mention that this modification does not compromise the drivers' privacy since many services will be offered at each specific SCH and vehicles will be switching there IDs frequently according to suggested privacy schemes in VANETs [9].

Thus in our modified system, each vehicle will maintain a list of the channels where each neighbor can be found during the next service channel. Each car can be available at one of the seven DSRC channels during SCH interval. Our goal is to enable vehicles to broadcast immediately high priority safety messages that might emerge during service channel interval and high priority safety messages that might remain in the queue at the end of the control channel. As we stated, the neighbors might be tuned to seven different channels and thus a vehicle that needs to transmit a high priority safety message cannot ensure that neighbors cars will receive the message. However, the transmitting vehicle can leverage the channels' table collected from the HELLO messages to decide how to disseminate its data. In graph theory terms, the channels form seven different cliques. Data can be broadcasted easily between nodes belonging to the same clique. However, to communicate with nodes at a different clique, a car needs to tune its DSRC radio to the frequency of the target clique. We need to make the critical safety information available at all the cliques in the shorter possible time with minimal overhead.
It is worthy to note here that the EN1 scheme will consume bandwidth resources during SCH interval, but will definitely increase the transmission opportunities for safety-related messages, and thus improve drivers safety through added event-driven message delivery reliability and timeliness. This is a typical tradeoff problem between SCH resource usage and safety message delivery.

We consider a scenario in which vehicle η has a SAFETY message to transmit to its $(M-1)$ neighbors. If the message is generated during the CCH slot, then the vehicle will contend for the control channel to transmit the message following the normal behavior of the IEEE 1609.4 protocol. If the high priority message was not able to access the medium at the end of the CCH interval, or otherwise if the message was generated during the SCH interval, then our Algorithm 1 is executed. First, vehicle η broadcasts the SAFETY message at its current channel ($\eta\_current\_channel$) if one or more neighbors are available at this channel. Broadcasting at the current channel does not incur any switching delay overhead and will permit to transfer the message quickly to neighbors already available at the same channel. Then, vehicle η broadcasts multiple copies of the SAFETY message on different channels, based on the number of neighbors on each channel and on the relevance of the message.

Let n_i be the number of neighbors of η tuned to channel c_i where $c_i \in \{c_1..c_K\}$ and $c_i \neq \eta\_current\_channel$ and where K is the total number of channels used in the system (K equals 7 in DSRC). At the start of the SCH interval, vehicle η creates a list of channels c_i ordered by the number of neighbors n_i. Let us denote with L the list $[c_1, .., c_k]$ s.t. $\forall i, j$ if $c_i < c_j$ then $n_i < n_j$.

Then, vehicle η performs the repeat-until cycle described in Algorithm 1. At each iteration, vehicle η removes the first element of L (denoted by c_j), tunes its radio to channel c_j and then broadcasts a copy of the SAFETY message. The iterative cycle ends when: (*i*) the SCH interval expires or (*ii*) L is empty (i.e. a copy of the message was sent on the all the c_k channels) or (*iii*) the maximum number of message retransmission MAX_{copies} is reached. We introduced the MAX_{copies} threshold to avoid the problem of starvation of non-safety applications on the SCH slots. The value of $1 \leq MAX_{copies} \leq K-1$ should be decided based on the relevance of the SAFETY message. In our experiments, we set $Max_{copies} = K-1$, i.e. the vehicle η will attempt to transmit a copy of the SAFETY message on all the available K channels. Each vehicle β receiving the SAFETY message from vehicle η will also rebroadcast it by using Algorithm 1 if flooding dissemination is enabled. As a result, vehicles of the VANET can receive the SAFETY message during the SCH interval even if they are tuned on different service channels. However, there will still be some possibilities that a vehicle tuned on a specific channel c_i will not be able to receive the message.[1] For this reason, we also assume that vehicle η will transmit a copy of the SAFETY message at the start of the next CCH interval.

Algorithm 1: Enhancement 1

1. In case a SAFETY message is generated during CCHI.
Broadcast a copy of the SAFETY message on the CCH.

2. In case a SAFETY message was un-transmitted at the end of CCHI or it is generated during SCHI.
if $n_{\eta\_current\_channel} > 0$ **then**
  Broadcast a copy of the SAFETY message on $\eta\_current\_channel$
end if
Create the list L of channels c_i ordered by neighbors count n_i
set number of copies $num_{copies} = 0$
repeat
  Get c_j = head(L)
  Switch to channel c_j
  Send a copy of the SAFETY message on channel c_j
  $num_{copies} = num_{copies} + 1$
until $L = \emptyset$ or $num_{copies} > MAX_{copies}$ or SCH expired
Broadcast a copy of the SAFETY message at the start of the next CCH interval.

Figure 6 shows the performance gain introduced by the EN1 scheme. We considered a scenario in which a vehicle generates a SAFETY message that is re-broadcasted by other vehicles in a multi-hop way till a maximum life distance is covered (i.e. Application II of Section 4). We considered a configuration with 100 vehicles on an highway of 1km length. Moreover, each vehicle produces the following traffic in background: (*i*) 1 HELLO message during each CCH interval, and (*ii*) 1 SERVICE message during the SCH interval. We assume that each vehicle is randomly associated to a service channel at the start of the simulation. Figure 6 shows the delivery delay of SAFETY message as a function of the distance covered, using both the current and the enhanced version of the IEEE 1609.4. In our simulation, we specifically consider the scenario where the SAFETY message got generated at the sender vehicle during a SCH interval. When using the IEEE 1609.4 scheme, the SAFETY message experiences additional delay at the first hop. The sender is unable to complete the transmission during the SCH interval, and thus must wait for additional $T_{untrans}$ delay before being able to access the control channel. The same problem might also happens when the message got re-broadcasted

[1] This might happen for instance when the SCH interval expires before vehicle η or any other rebroadcasting vehicle β is able to broadcast the message at a certain channel c_i.

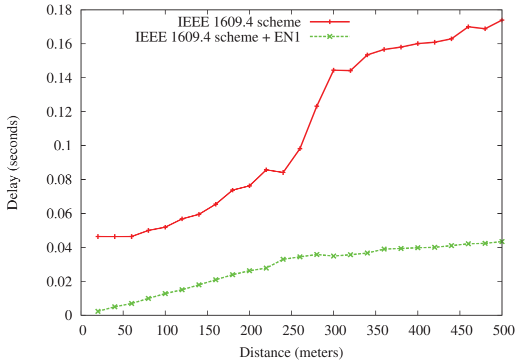


Figure 6: The improvement produced by the `EN1` scheme in terms of delivery delay.

beyond the first-hop (at a distance of around 250 meters from the source). The high contention of the control channel might prevent the forwarding vehicle from accessing the channel during the CCH interval. However when `EN1` is used, this additional $T_{untrans}$ delay is greatly mitigated, since high priority `SAFETY` messages are forwarded during SCH intervals using service channels. As a result, the `EN1` scheme provides a significant reduction of the delivery delay over multi-hop VANETs, and this improvement increases with the distance from the source vehicle, as shown by Figure 6.

Enhancement2 (EN2). We propose to apply an extended version of the Distributed Contention Control (DCC) mechanism [3] at the lower MAC layer (802.11p) to enhance the utilization of the control channel, by mitigating the problem of synchronous collisions at the start of the CCH interval. To this aim, we introduce the metric of *Slot Utilization (SU)* that measures the level of contention on the control channel for each vehicle η. During the back-off mechanism, each vehicle η counts the number of transmission attempts it observes on the channel (Num_Busy_Slots) and then divides this number by the total number of slots available for transmission observed on the channel ($Num\_Available\_Slots$). The result is a lower bound on the contention level of CCH, and is given by:

$$SU = \frac{Num\_Busy\_Slots}{Num\_Available\_Slots} \tag{12}$$

It is easy to see that SU provides a value between 0 and 1, where a value of 1 indicates a full saturated channel. In *EN2*, once the back-off counter value reaches zero, instead of transmitting the message as foreseen by the traditional 802.11 MAC protocol, the vehicle η computes the Probability of Transmission (P_T) defined as follows:

$$P_T = (1 - SU^{\phi}) \tag{13}$$

Vehicle η transmits the safety message with probability P_T, while with probability $1 - P_T$ it defers its scheduled transmission attempt and increments by 1 the number of transmission attempts (ϕ), which is originally set to one. Basically, if the transmission was deferred, the vehicle behaves like a virtual collision occurred during its attempt. The parameter ϕ is introduced to avoid fluctuations between high/low channel utilization periods, and provides different levels of priorities in accessing the channel. ϕ is made equal to the number of transmission attempts to privilege old transmissions attempts, so that the probability to access the medium increases with the users' waiting time. Figure 7(a) shows the value of P_T as a function of SU, for different number of transmission attempts, ϕ. The analytical model proposed in [3] demonstrates that the DCC mechanism described by Equations 13 is able to converge to an equilibrium point, in which the utilization of the channel is maximized while the risk of network collapse is avoided. In our approach, each vehicle might have a message to transmit at each CCH slot, and thus it might collect multiple observations of the SU of the control channel. For this reason, vehicle η accounts for the channel history in the computation of the SU as follows:

$$SU = SU_{new} \cdot \alpha + SU_{old} \cdot (1 - \alpha) \tag{14}$$

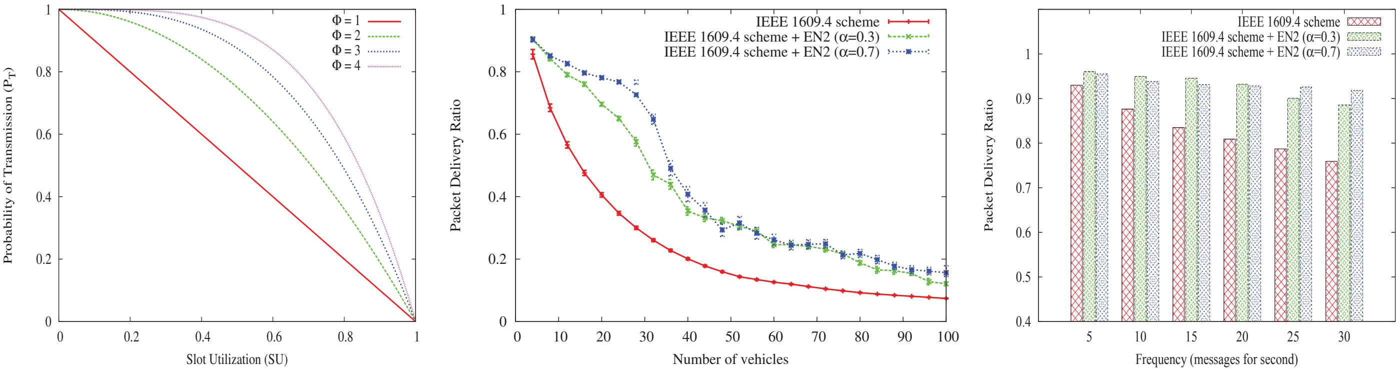
Here, SU_{new} is the value computed through Equation 12 during the current transmission attempt, SU_{old} is the previous stored value of the SU metric, and α is a parameter that decides the relevance of history in the current decisions. Algorithm 2 shows the operations of our scheme. Figure 7(b) shows the delivery rate as a function of the number of vehicles, when all the safety messages are generated during the SCH slot (worst-case analysis). We considered the original IEEE 802.11p MAC protocol, and two different configurations of the `EN2` scheme with different values of the α parameter (Equation 14). Figure 7(b) confirms that our solution is able to mitigate the problem of synchronous collisions at the start of the CCH slot, thanks to the contention control mechanism introduced by Equation 13. Moreover, the performance gain produced by the `EN2` scheme increases with the contention load on the control channel, as demonstrated by Figure 7(c). Here, we show the delivery rate as a function of the number of broadcast messages generated per second by each vehicle (we considered a scenario with only 12 active vehicles). The three schemes perform similarly under low network load conditions. However, the performance gains of the `EN2` scheme are evident under moderate and high network loads. In these cases, the increased load on the control channel will produce an increased SU level perceived by each vehicle. As a result, some of them will defer their transmissions attempts (Equation 13), thus reducing the risk of collisions. However, the utilization of ϕ parameter assures that a prioritized set of vehicles will still be able to access the channel, avoiding the risk of fluctuations between high/low channel utilization periods.

Algorithm 2: Enhancement 2

```
if Backoff Counter == 0 then
    Compute SU_new through Equation 12
    Compute SU = SU_new · α + SU_old · (1 − α)
    Compute P_T = f(SU, φ) through Equation 13
    if (Random[0,1[ > P_T) then
        Transmit message on the control channel
    else
        φ = φ + 1
    end if
end if
```

6. CONCLUSIONS

In this paper, we have investigated the performance of multi-channel VANETs through three novel contributions. First, we have proposed an implementation of the IEEE 1609.4 WAVE protocol for the ns2 simulator, and we have integrated it with the existing implementation of the IEEE 802.11 MAC protocol. Second, we have proposed a performance analysis of 1609.4-based multi-channel VANETs, and we have discussed the current inadequacy of the protocol in guaranteeing strict QoS requirements of safety-related applications in terms of delivery rate and delay. Third, we have proposed two enhancements to the existing DSRC 1609.4/802.11p stack, and we have shown that our suggested solutions can considerably increase the delivery rate and reduce the delay of safety messages, without requiring significant changes to the existing stack.



(a) (b) (c)

Figure 7: The P_T values as a function of the SU level are shown in Figure 7(a). The improvement produced by the `EN2` scheme in terms of delivery ratio is shown in Figures 7(b) and Figure 7(c).

Future works will include the implementation of the EDCA policy at the MAC layer for the ns2 model, and a graph-theory based formulation of the `EN2` scheme for the dissemination of broadcast messages on the multi-channel VANETs. Finally, we plan also to study the behavior of IEEE 1609.4 when both `EN1` and `EN2` are implemented together.

7. REFERENCES

[1] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3), 535-547, 2000.

[2] L. Bononi and M. Di Felice and S. Pizzi. DBA-MAC: Dynamic Backbone Assisted Medium Access Control for Efficient Broadcast in VANETs. *Journal of Interconnection Networks (JOIN)*, 10(4), pp. 1793-1813, 2009.

[3] L.Bononi, M.Conti and L.Donatiello. Design and Performance Evaluation of a Distributed Contention Control (DCC) Mechanism for IEEE 802.11 Wireless Local Area Networks. *Journal of Parallel and Distributed Computing*, 60(4), pp. 407-430, 2000.

[4] F. Cali', M. Conti and E. Gregori. Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. *IEEE/ACM Transactions on Networking*, 8(6), pp. 785-799, 2000.

[5] C. Campolo, A. Vinel, A. Molinaro and Y. Koucheryavy. Modeling Broadcasting in IEEE 802.11p/WAVE Vehicular Networks. *IEEE Communication Letters*, 15(2), pp. 199-201, 2011.

[6] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi and H. Hartenstein. Overhaul of IEEE 802.11 Modeling and Simulation in NS2. In *Proc. of ACM MSWIM*, Crete Island, pp. 159-168, 2007.

[7] CVIS EU project. http://cvisproject.org. FP7.

[8] S. Eichler. Performance Evaluation of the IEEE 802.11p WAVE Communication Standard. In *Proc. of IEEE VTC-Fall*, Baltimore, pp. 2199 - 2203, 2007.

[9] M. Gerlach and F. Guttler. Privacy in VANETs using Changing Pseudonyms - Ideal and Real. In *Proc. of IEEE VTC-Spring*, Dublin, pp. 2521-2525, 2007.

[10] A. J. Ghandour, K. Fawaz and H. Artail. Data Delivery Guarantees in congested Vehicular Ad Hoc Networks using cognitive networks. In *Proc. of IEEE IWCMC*, Istanbul, p.871-876, 2011.

[11] K. Hong, J. B. Kennedy, V. Rai and K. P. Laberteaux. Evaluation of Multi-Channel Schemes for Vehicular Safety Communications. In *Proc. of IEEE VTC-Spring*, Taipei, pp. 1-5, 2010.

[12] T. Hossain, Y. Cui and Y. Xue. Vanets: Case Study of a Peer-to-Peer Video Conferencing System. In *Proc. of IEEE CCNC*, Las Vegas, pp. 1-2, 2009.

[13] D. Jiang and L. Delgrossi. IEEE 1609.4 DSRC Multi-Channel Operations and Its Implications on Vehicle Safety Communications. In *Proc. of IEEE VNC*, Tokyio, pp. 1-8, 2009.

[14] IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation. *IEEE Std 1609.4-2010*, 2010.

[15] A. Amoroso, G. Marfia and M. Roccetti. Going Realistic and Optimal: A Distributed Multi-Hop Broadcast Algorithm for Vehicular Safety. *Computer Networks*, 55(10), pp. 2504-2519, 2011.

[16] J. Misic, G. Badawy and V. Misic. Performance Characterization for IEEE 802.11p Network with Single Channel Devices. *IEEE Transactions on Vehicular Technology*, 68(4), pp. 1775-1789, 2011.

[17] M. Raya and J.-P. Hubaux. The Security of Vehicular Ad Hoc Networks. In *Proc. of ACM SANS*, Alexandria, pp. 11-21, 2005.

[18] Q. Wang, S. Leng, Y. Zhang and H. Fu. A QoS Supported Multi-channel MAC for Vehicular Ad Hoc Networks. In *Proc. of IEEE VTC-Spring*, Budapest, pp. 1-5, 2011.

[19] IEEE Std 802.11p, Part 11, Amendment 6: Wireless Access in Vehicular Environments (WAVE), 2010.

[20] T. L. Willke, P. Tientrakool and N. F. Maxemchuk. A Survey of Inter-Vehicle Communication Protocols and Their Applications. *IEEE Communications on Surveys and Tutorials*, 11(2), pp. 3-20, 2009.

[21] S. Wang and C.-C. Lin. NCTUns 5.0: a network simulator for IEEE 802.11(p) and 1609 wireless vehicular network researches. In *Proc. of IEEE VTC-Fall*, Calgary, 11(2), pp. 3-20, 2008.