VeMAC: A TDMA-Based MAC Protocol for Reliable Broadcast in VANETs

Hassan Aboubakr Omar, *Student Member*, *IEEE*,
Weihua Zhuang, *Fellow*, *IEEE*, and Li Li, *Member*, *IEEE*

Abstract—The need of a medium access control (MAC) protocol for an efficient broadcast service is of great importance to support the high-priority safety applications in vehicular ad hoc networks (VANETs). This paper introduces VeMAC, a novel multichannel TDMA MAC protocol proposed specifically for a VANET scenario. The VeMAC supports efficient one-hop and multihop broadcast services on the control channel by using implicit acknowledgments and eliminating the hidden terminal problem. The protocol reduces transmission collisions due to node mobility on the control channel by assigning disjoint sets of time slots to vehicles moving in opposite directions and to road side units. Analysis and simulation results in highway and city scenarios are presented to evaluate the performance of VeMAC and compare it with ADHOC MAC, an existing TDMA MAC protocol for VANETs. It is shown that, due to its ability to decrease the rate of transmission collisions, the VeMAC protocol can provide significantly higher throughput on the control channel than ADHOC MAC.

Index Terms—TDMA, medium access control, reliable broadcast, and vehicular ad hoc networks

1 Introduction

An ad hoc network is defined as a collection of nodes dynamically forming a network without any existing infrastructure or centralized administration. One special type of mobile ad hoc networks is the network among moving vehicles, which is known as vehicular ad hoc network (VANET). A VANET consists of a set of vehicles equipped with a communication device, called on-board unit (OBU), and a set of stationary units along the roads, called road side units (RSUs), which can be connected together and/or to the Internet via wireless or wireline links. Each OBU has a radio interface to connect to other OBUs and RSUs, as well as a wireless or wired interface to which an application unit can be attached. The main objectives of VANETs are to provide efficient vehicle-to-vehicle (V2V) and vehicle-to-RSU (V2R) communications. Based on these two kinds of communications, VANETs can support many applications in safety, entertainment, and vehicle traffic optimization [2], [3]. Motivated by the importance of vehicular communications, the United States Federal Communication Commission (FCC) has allocated 75-MHz radio spectrum in the 5.9-GHz band for Dedicated Short Range Communications (DSRC) to be exclusively used by V2V and V2R communications. The DSRC spectrum is divided into seven 10-MHz channels: six service channels for safety and nonsafety related applications, and one control channel for transmission of control information and high-priority short safety messages.

Most (if not all) of the high-priority safety applications proposed for VANETs are based on one-hop broadcast of information. For instance, for V2V communication-based applications such as the precrash sensing, blind spot warning, emergency electronic brake light, and cooperative forward collision avoidance, each vehicle periodically broadcasts information about its position, speed, heading, acceleration, turn signal status, and so on, to all the vehicles within its one-hop neighborhood [2]. Similarly, for V2R communication-based applications, such as the curve speed warning and traffic signal violation warning, an RSU periodically broadcasts to all the approaching vehicles information related to the traffic signal status and timing, road surface type, weather conditions, and so on [2]. As the precision of the safety applications is directly related to the safety of people on road, the need of a medium access control (MAC) protocol which provides an efficient broadcast[1] service is crucial for VANETs.

Various MAC protocols have been proposed for VANETs, based either on IEEE 802.11 or on channelization such as time division multiple access (TDMA), space division multiple access (SDMA), and code division multiple access (CDMA). In SDMA schemes, each vehicle decides whether or not it is allowed to access the channel based on its location on the road [4], [5]. An SDMA scheme consists of three main parts: A *discretization procedure* which divides the road into small areas called cells, a *mapping function* which assigns to each of the cells a unique time slot, and an *assignment rule* which specifies which time slots a vehicle is allowed to access based on the cell where it is currently located. Similarly, CDMA is proposed for MAC in VANETs due to its robustness against interference and noise [6], [7]. The main problem which arises with CDMA in VANETs is how to allocate the pseudonoise (PN) codes to different vehicles. Due to a large number of vehicles, if every vehicle is assigned a unique PN code, the length of

- *H.A. Omar and W. Zhuang are with the Center for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. E-mail: {h3omar, wzhuang}@uwaterloo.ca.*
- *L. Li is with the Communications Research Center, Ottawa, ON K2H 8S2, Canada. E-mail: li.li@crc.ca.*

Manuscript received 18 Sept. 2011; revised 29 Jan. 2012; accepted 18 May 2012; published online 12 June 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-09-0516.
Digital Object Identifier no. 10.1109/TMC.2012.142.

[1] When "broadcast" is mentioned solely, it refers to one-hop broadcast.

Published by the IEEE CS, CASS, ComSoc, IES, & SPS

these codes will become extremely long, and the required bit rates for VANET applications may not be attained. Consequently, it is mandatory that the PN codes be shared among different vehicles in a dynamic and fully distributed way [7]. On the other hand, the IEEE 802.11p is a recently proposed MAC standard for VANETs [8]. The protocol is based on the legacy IEEE 802.11 standard [9], which is widely implemented, but does not provide an efficient broadcast service. The reason is that, for broadcast frames, no RTS/CTS exchange is used and no acknowledgment is transmitted from any of the recipient of the frame [9]. This lack of RTS/CTS exchange results in a hidden terminal problem which reduces the frame delivery ratio of the broadcast service, especially with the absence of acknowledgment frames [10]. Another limitation is that, in a VANET scenario, by employing the enhanced distributed channel access (EDCA) scheme defined in the IEEE 802.11 standard, the high-priority safety messages will be assigned to the high-priority access categories (ACs) which contend for the wireless channel using a small contention window size [9]. Although this small contention window size allows the high-priority safety frames to be transmitted with small delays, it increases the probability of transmission collisions when multiple nodes within the same communication range are simultaneously trying to broadcast their safety messages [11]. Moreover, unlike the unicast case, the size of the contention window is not doubled when a collision happens among the broadcasted safety messages because there is no collision detection for the broadcast service without CTS and acknowledgment frames [9]. Different from the contention-based IEEE 802.11p standard, the ADHOC MAC protocol is based on TDMA and is proposed for intervehicle communication networks [12]. The ADHOC MAC protocol operates in a time-slotted structure, where time slots are grouped into virtual frames, i.e., no frame alignment is needed. By letting each node report the status of all the time slots in the previous (sliding) virtual frame, the ADHOC MAC can support a reliable[2] broadcast service without the hidden terminal problem [12]. As well, the ADHOC MAC provides a multihop broadcast service, which can cover the whole network using a significantly smaller number of relaying nodes than that using a flooding procedure. Moreover, in ADHOC MAC, each node is guaranteed to access the channel at least once in each virtual frame, which is suitable for non delay-tolerant applications. However, simulation results show that, due to node mobility, the throughput reduction can reach 30 percent for an average vehicle speed of 50 km/h [13]. Another major limitation of ADHOC MAC is that it is a single channel protocol, not suitable for the seven DSRC channels.

This paper presents VeMAC, a novel multichannel TDMA protocol developed based on ADHOC MAC [12] and designed specifically for VANETs. On the control channel, the protocol provides a reliable one-hop broadcast service without the hidden terminal problem as well as an efficient multihop broadcast service to disseminate information all over the network. The VeMAC assigns disjoint sets of time slots to vehicles moving in opposite directions and to RSUs, and hence can decrease the rate of transmission collision on the control channel caused by node mobility. As well, the VeMAC employs new techniques for the nodes to access the available time slots and to detect transmission collisions. These techniques are different from the ones used by ADHOC MAC, which have some limitations as to be discussed in details. It is shown that the proposed VeMAC protocol provides significantly higher throughput on the control channel than that of ADHOC MAC and ADHOC enhanced (an enhanced version of ADHOC MAC introduced in this paper).

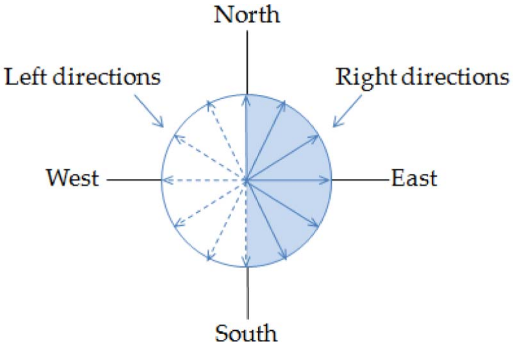


Fig. 1. Right and left directions of vehicle movement.

The rest of this paper is organized as follows: Section 2 describes the system model and Section 3 presents the VeMAC protocol. The performance of the VeMAC protocol on the control channel is analyzed in Section 4. Section 5 presents the simulation results and explains some limitations of the ADHOC MAC protocol. A discussion on the main features of the VeMAC protocol in comparison with ADHOC MAC and some possible VeMAC extensions are presented in Section 6. Finally, Section 7 concludes this research and suggests some future works.

2 System Model

The VANET under consideration consists of a set of RSUs and a set of vehicles moving in opposite directions on two-way vehicle traffic roads. A vehicle is said to be moving in a left (right) direction if it is currently heading to any direction from north/south to west (east), as shown in Fig. 1. Based on this definition, if two vehicles are moving in opposite directions on a two-way road, it is guaranteed that one vehicle is moving in a left direction while the other vehicle is moving in a right one.

The VANET has one control channel, denoted by c_0, and M service channels, denoted by $c_1, c_2, \dots, c_M$. Channel c_0 is used for transmission of two kinds of information: high-priority short applications (such as periodic or event driven safety messages), and control information required for the nodes to determine which time slots they should access on channel $c_i, i = 0, \dots, M$. The M service channels are used for transmission of safety or nonsafety related application messages. A *provider* is a node which announces on channel c_0 for a service offered on a specific service channel, while a *user* is a node which receives the announcement for a service and decides to make use of this service. Each node has two transceivers: Transceiver1 is always tuned to channel c_0, while transceiver2 can be tuned to any service channel $c_i, i = 1, \dots, M$. It is assumed that the transmission

[2] In this paper, the term "reliable" means "acknowledged."

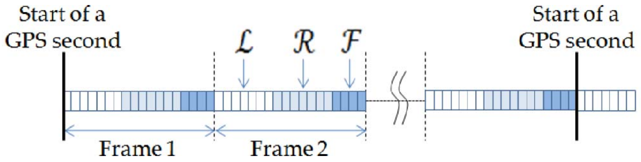


Fig. 2. Partitioning of each frame on channel c_0 into $\mathcal{L}, \mathcal{R}$, and $\mathcal{F}$ sets.

power levels on all channels are fixed and known to all nodes. All channels are symmetric, in the sense that node x is in the communication range of node y if and only if node y is in the communication range of node x. Each node is identified by a MAC address as well as a short identifier (ID). The ID is chosen by each node at random, included in the header of each packet transmitted on channel c_0, and changed if the node detects that its ID is already in use by another node [12].

Time is partitioned to frames consisting of a constant number of fixed duration time slots. The number of time slots per frame on channel c_m is denoted by s_m, $m = 0, \ldots, M$, and a time slot on channel c_m is identified by the index of this time slot within a frame on channel c_m.[3] On channel c_0, each frame is partitioned into three sets of time slots: $\mathcal{L}$, $\mathcal{R}$, and $\mathcal{F}$, as shown in Fig. 2. The $\mathcal{F}$ set is associated with RSUs, while the $\mathcal{L}$ and $\mathcal{R}$ sets are associated with vehicles moving in left and right directions, respectively. Every node (i.e., vehicle or RSU) is equipped with a global positioning system (GPS) receiver and can accurately determine its position and moving direction using GPS. The current position of each node is included in the header of each packet transmitted on channel c_0, and synchronization among nodes is performed using the 1PPS signal provided by any GPS receiver. The rising edge of this 1PPS is aligned with the start of every GPS second with accuracy within 100 ns even for inexpensive GPS receivers. Consequently, this accurate 1PPS signal can be used as a common time reference among all the nodes. All the channels are slot synchronized and, on each channel, each second contains an integer number of frames as shown in Fig. 2 for channel c_0. Hence, at any instant, each node can determine the index of the current slot within a frame on any channel c_m, $m = 0, \ldots, M$, and whether it belongs to the $\mathcal{L}$, $\mathcal{R}$, or $\mathcal{F}$ set on channel c_0. In case of a temporary loss of GPS signal, the synchronization among different nodes can still be maintained within a certain accuracy for a time duration, which depends on the stability of the GPS receiver's local oscillator at each node [14]. If the GPS signal is lost in a certain area for a long duration (longer than a specified threshold), a distributed synchronization scheme, such as the one presented in [14], should be employed until the GPS signal is recovered. Details of such a back up synchronization scheme are out of scope of this paper.

For a certain node x, the following sets are defined:

- $N(x)$: The set of IDs of the one-hop neighbors of node x on channel c_0, from which node x has received packets on channel c_0 in the previous s_0 slots;
- $T_m(x)$: The set of time slots that node x must not use on channel c_m in the next s_m time slots, $m = 0, \ldots, M$.

Header	AnS	AcS	High priority short applications

Fig. 3. Format of each packet transmitted on channel c_0.

Set $T_m(x)$ is used by node x to determine which time slots it can access on channel c_m without causing any hidden terminal problem. Constructing and updating set $T_0(x)$ is different from sets $T_m(x)$, $m = 1, \ldots, M$, as described in Section 3.

3 VeMAC Protocol

3.1 VeMAC Preliminaries

In the VeMAC protocol, each node must acquire exactly one time slot in a frame on channel c_0. Once a node acquires a time slot, it keeps accessing the same slot in all subsequent frames on channel c_0 unless a transmission collision is detected (as to be explained in details in Section 3.2). Each packet[4] transmitted on channel c_0 is divided into four main fields: header, announcement of services *(AnS)*, acceptance of services *(AcS)*, and high-priority short applications, as shown in Fig. 3.

Each node must transmit a packet during its time slot even if the node has no data to include in the high-priority short applications field. The reason is that information in the header, *AnS* and *AcS* fields, is necessary for other nodes to decide which time slots they can access on the control channel and service channels as to be described in Sections 3.2 and 3.3.

Two types of transmission collision on time slots can happen on channel c_0 [13]: access collision and merging collision. An access collision happens when two or more nodes within two hops of each other attempt to acquire the same available time slot. On the other hand, a merging collision happens when two or more nodes acquiring the same time slot become members of the same two-hop set[5] (THS) due to node activation or node mobility. The difference between the two types of collision is that access collisions occur among nodes which are trying to acquire a time slot, while merging collisions occur among nodes which have successfully acquired a time slot. In VANETs, although merging collisions can happen among vehicles moving in the same direction due to acceleration or deceleration, it is more likely to occur among vehicles moving in opposite directions (approaching each other) or between a vehicle and a stationary RSU because they approach each other with a much higher relative velocity as compared to vehicles moving in the same direction. For example, in Fig. 4, if vehicle x moves to THS2 and if x is using the same time slot as z, then collision will occur at y. Upon detection of a merging collision on channel c_0, each colliding node should release its time slot and acquire a new one, which may generate more access collisions.

3.2 Accessing Slots on the Control Channel

For the purpose of time slot assignment on channel c_0, in the header of each packet transmitted on channel c_0, the

[3] Note that, the same time slot can have different indices on channels c_i and c_j, $i \neq j$, because s_i is not necessarily equal to s_j.

[4] In the rest of the paper, the term "packet" is used instead of "frame" to refer to MAC layer Protocol Data Unit (MPDU), i.e., the unit of data exchanged between two peer MAC entities. It is to avoid confusion with the "frame" which is a collection of time slots.

[5] A THS is a set of nodes in which each node can reach any other node in two hops at most.

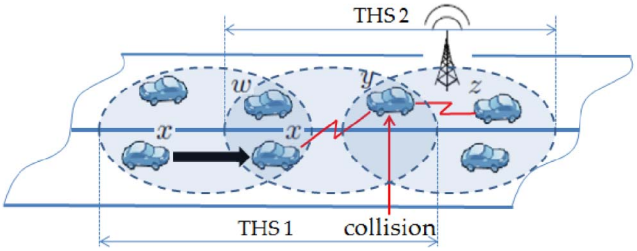
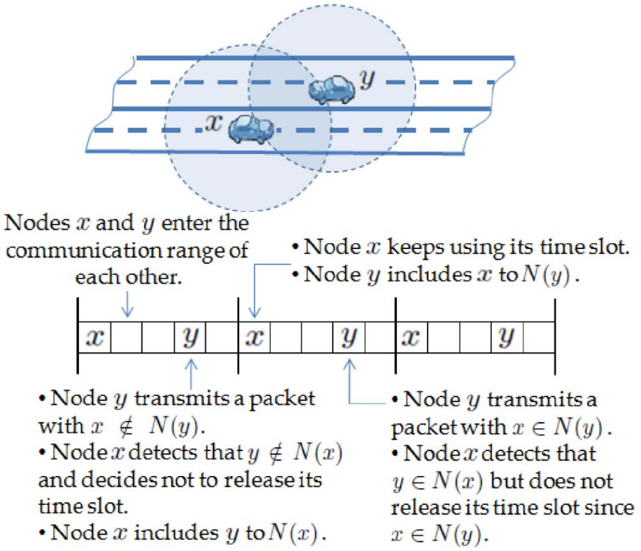


Fig. 4. Merging collision caused by node mobility.

transmitting node y should include set $N(y)$ and the time slot used by each node $z \in N(y)$. The short IDs in set $N(y)$ serve to decrease the overhead as compared to including the MAC address of each one-hop neighbor in the header of each transmitted packet. Suppose node x is just powered on and needs to acquire a time slot. It starts listening to channel c_0 for s_0 successive time slots (not necessarily in the same frame). At the end of the s_0 slots, node x can determine $N(x)$ and the time slot used by each node $i \in N(x)$. In addition, because each node $i \in N(x)$ announces $N(i)$ and the time slot used by each node $j \in N(i)$, node x can determine the time slot used by each of its two-hop neighbors, $j \in N(i), j \notin N(x), \forall i \in N(x)$. Accordingly, node x sets $T_0(x)$ to the set of time slots used by all nodes within its two-hop neighborhood. Then, sets $N(x)$ and $T_0(x)$ are updated by node x at the end of each time slot (always based on information received in the previous s_0 slots).

Given $T_0(x)$, node x determines the set of accessible time slots $A(x)$ (to be discussed) and then attempts to acquire a time slot by randomly accessing any time slot in $A(x)$, say time slot k. If no other node in the two-hop neighborhood of node x simultaneously attempts to acquire time slot k, then no access collision happens. In this case, the attempt of node x is successful, and each one-hop neighbor i of node x adds node x to the set $N(i)$ and records that node x is using time slot k. On the other hand, if at least one node within the two-hop neighborhood of node x accesses time slot k, then all the transmissions in the slot fail and time slot k is not acquired by any of the contending nodes. Node x will determine whether its attempt was successful or not by observing the $s_0 - 1$ time slots following k. The attempt of node x is considered successful iff the packets received from all $i \in N(x)$ indicate that $x \in N(i)$. Otherwise, node x reaccesses one of the time slots in $A(x)$ until it successfully acquires a time slot. Once node x acquires a time slot, it keeps using the same slot in all subsequent frames unless a merging collision happens. Similar to an access collision, a merging collision is detected by node x as soon as it receives a packet from a node $i \in N(x)$ indicating that $x \notin N(i)$.

At the end of each time slot, the collision detection by a certain node x should be done *before* updating the set $N(x)$. Upon receiving a packet from a node y indicating that $x \notin N(y)$, we stress on that, node x should approve this collision detection and release its time slot iff the transmitting node $y \in N(x)$. This condition is referred to as the slot release prevention (SRP) condition, and its main objective is to prevent node x from unnecessarily releasing its time slot when it just enters the communication range of another node y. To illustrate that, consider the time slot assignment shown in Fig. 5 for the two nodes x and y.



Nodes x and y enter the communication range of each other.

- Node x keeps using its time slot.
- Node y includes x to $N(y)$.

- Node y transmits a packet with $x \notin N(y)$.
- Node x detects that $y \notin N(x)$ and decides not to release its time slot.
- Node x includes y to $N(x)$.

- Node y transmits a packet with $x \in N(y)$.
- Node x detects that $y \in N(x)$ but does not release its time slot since $x \in N(y)$.

Fig. 5. The SRP condition preventing node x from unnecessarily releasing its time slot.

When node x enters the communication range of node y, even if no collision happens, the first packet received by node x from node y will indicate that $x \notin N(y)$. The reason is that, by the time node y transmits its packet, node y has not yet received any packet from node x to include it in $N(y)$. By applying the SRP condition, when node x receives the first packet from node y, node x determines that node $y \notin N(x)$ and does not release its time slot (remember that collision detection by node x is done before updating $N(x)$). After node x's transmission, the subsequent packets transmitted by node y will indicate that $x \in N(y)$ and, hence, the unnecessary release of node x's time slot is prevented. Note that, without the SRP condition, when two nodes enter the communication range of each other, one of them will eventually release its time slot even if no merging collision happens. This behavior can significantly decrease the performance of a TDMA protocol as discussed in Section 5.3.

Consider that node x is moving in one of the right directions. Initially, node x limits the set $A(x)$ to the available time slots associated with the right directions, i.e., $A(x) = \overline{T_0(x)} \cap \mathcal{R}$. If after a certain number of frames, say τ frames, node x cannot acquire a time slot, then node x augments $A(x)$ by adding the time slots associated with the opposite direction, i.e., $A(x) = \overline{T_0(x)} \cap (\mathcal{R} \cup \mathcal{L})$. If, after τ more frames, node x still cannot acquire a time slot, node x will start to access any available time slot, i.e., $A(x) = \overline{T_0(x)}$. The same procedure applies for a vehicle moving in a left direction by replacing $\mathcal{R}$ with $\mathcal{L}$. Similarly, if node x is an RSU, for the first τ frames $A(x) = \overline{T_0(x)} \cap \mathcal{F}$, and then $A(x) = \overline{T_0(x)}$. The parameter τ is referred to as split up parameter, and the choice of the τ value can affect the rates of access collision and merging collision. For example, when $\tau = 0$, all the vehicles and RSUs are accessing the same set of time slots. Hence, a merging collision is possible between any two nodes. However, when a merging collision happens, each colliding node x is free to access any time slot in $\overline{T_0(x)}$, which can decrease the probability of an access collision. On the other extreme, when $\tau = \infty$, the vehicles moving in opposite directions and the RSUs are accessing disjoint sets of time slots. However, when a merging collision happens, for example, among vehicles moving in a right direction, there is a higher probability of an access collision (compared with the $\tau = 0$ case) because the choice of each colliding node x is limited to time slots in $\overline{T_0(x)} \cap \mathcal{R}$. A performance comparison between theses two extreme cases is provided in Section 5.

Using the proposed scheme, a reliable broadcast service can be provided on channel c_0. That is, if node x transmits a broadcast packet on time slot k, by listening to the $s_0 - 1$ time slots following k, node x can determine the set ξ of one-hop neighbors which have successfully received the packet, where $\xi = \{i \in N(x) : x \in N(i)\}$. In other words, when node i indicates that $x \in N(i)$, it is considered as an implicit acknowledgment by node i of receiving the packet broadcasted by node x.

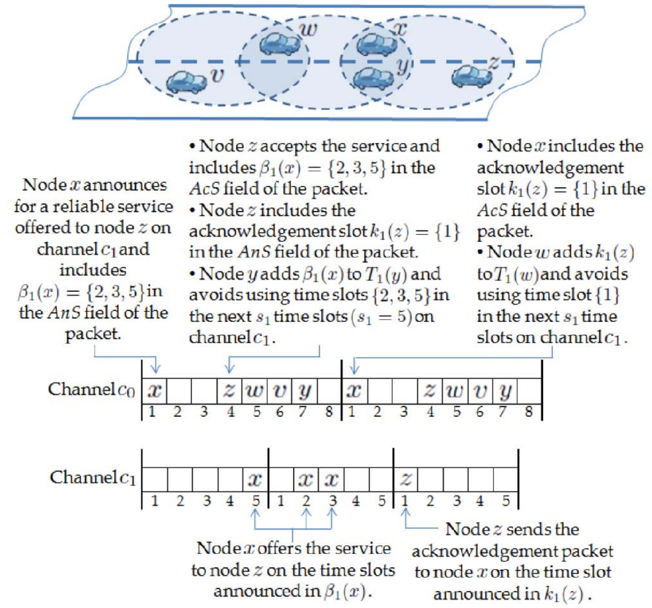
3.3 Accessing Slots on the Service Channels

Consider that a node x has a MAC layer service data unit (MSDU)[6] to be delivered to a certain destination (assuming unicast) on service channel c_m. In what follows, the term "service" refers to the delivery of an MSDU on a certain service channel. By using $T_m(x)$ (how node x constructs $T_m(x)$ will be explained), node x determines the set of time slots that it will access on channel c_m to offer the service, denoted by $\beta_m(x)$, such that $\beta_m(x) \cap T_m(x) = \phi$. Accordingly, node x announces the following information in the *AnS* field of its next packet transmitted on channel c_0:

1. priority of the service,
2. reliability of the service (i.e., acknowledged or not),
3. MAC address of the intended destination y,
4. the number m of the service channel, and
5. $\beta_m(x)$.

Once the provider x announces for the service, no further action is needed unless the destination accepts the service as described below.

Based on the information announced by provider x on channel c_0, the destination y determines whether or not to make use of the announced service. If node y decides to use the service by provider x on channel c_m, it accepts the service by including $\beta_m(x)$ in the *AcS* field of its next packet transmitted on channel c_0. The announcement of $\beta_m(x)$ by the user y is for the surrounding nodes to update their T_m sets as to be discussed. Also, for a reliable service, node y should include in the *AnS* field the time slot that will be used by node y to transmit the acknowledgment packet, denoted by $k_m(y)$. Node y determines $k_m(y)$ such that $k_m(y) \notin T_m(y)$. When provider x receives the acceptance of the service, it tunes its transceiver2 to channel c_m and starts offering the service on the time slots announced in $\beta_m(x)$. As well, if the service is reliable, node x should include $k_m(y)$ in the *AcS* field of its next packet transmitted on channel c_0. Again, the announcement of $k_m(y)$ by provider x is to avoid the collision of the acknowledgment packet by properly updating the T_m sets of the surrounding nodes.



Node x announces for a reliable service offered to node z on channel c_1 and includes $\beta_1(x) = \{2, 3, 5\}$ in the *AnS* field of the packet.

- Node z accepts the service and includes $\beta_1(x) = \{2, 3, 5\}$ in the *AcS* field of the packet.
- Node z includes the acknowledgement slot $k_1(z) = \{1\}$ in the *AnS* field of the packet.
- Node y adds $\beta_1(x)$ to $T_1(y)$ and avoids using time slots $\{2, 3, 5\}$ in the next s_1 time slots ($s_1 = 5$) on channel c_1.

- Node x includes the acknowledgement slot $k_1(z) = \{1\}$ in the *AcS* field of the packet.
- Node w adds $k_1(z)$ to $T_1(w)$ and avoids using time slot $\{1\}$ in the next s_1 time slots on channel c_1.

Node x offers the service to node z on the time slots announced in $\beta_1(x)$.

Node z sends the acknowledgement packet to node x on the time slot announced in $k_1(z)$.

Fig. 6. Node x offering a service to node z on channel c_1.

Node y should transmit the acknowledgment only after node x announces $k_m(y)$ on channel c_0.

Each node updates sets T_m, $m = 1, \ldots, M$, as follows: When node x receives a packet on channel c_0 from another node y, based on the position of node y which is included in the header of the packet, and the position of node x obtained from the GPS receiver, node x can estimate its distance to node y. Based on this estimated distance and on the fixed transmission power on all channels, which is known to node x, node x can determine whether or not node y is in its communication range on channel c_m, $m = 1, \ldots, M$.[7] If node x decides that it can reach node y on a certain channel c_j, node x adds to set $T_j(x)$ the time slots indicated by each β_j set and k_j slot included in the *AcS* field of the packet transmitted by y. The reason is that, each β_j represents a set of time slots over which node y will receive a packet on channel c_j from a certain provider in the next s_j slots. Similarly, each k_j indicates a time slot over which node y will receive an acknowledgment packet on channel c_j from a certain user in the next s_j slots. Consequently, by updating T_j in the way described, collision at node y can be prevented because all the one-hop neighborhood of node y will avoid using the time slots over which node y will receive packets. At the end of time slot i_m on channel c_m, if $i_m \in T_m$, i_m is removed from T_m, $m = 1, \ldots, M$ and $i_m = 1, \ldots, s_m$. Note that updating the T_m, $m = 1 \ldots, M$, sets is based on information in the *AcS* (not in the *AnS*) field, which eliminates any exposed terminal problem. The following example illustrates how the nodes access the service channels.

Consider the THS configuration shown in Fig. 6, node x has a reliable service to offer to node z on time slots numbered 2, 3, and 5 on channel c_1. Fig. 6 shows the sequence of actions taken by provider x, user z, and the surrounding nodes y and w. First, node x announces for

[6] An MSDU is a unit of data arriving to the MAC layer entity from the layer above.

[7] It is assumed that each node has a path loss model for each service channel $c_m, m = 1, \ldots, M$.

the service and includes $\beta_1(x) = \{2, 3, 5\}$ in the AnS field of its packet transmitted on channel c_0. Following this announcement, no action is taken by both surrounding nodes w and y. Once node z accepts the service and announces $\beta_1(x)$, node x starts offering the service on channel c_1 on time slots $\{2, 3, 5\}$ as announced in $\beta_1(x)$. When node y receives the packet transmitted by node z on channel c_0, it adds $\beta_1(x)$ to $T_1(y)$ to avoid using the upcoming time slots $\{2, 3, 5\}$ over which node z will receive packets from node x (assume that node y can reach node z on channel c_1). Note that, node w is free to use the time slots in $\beta_1(x) = \{2, 3, 5\}$ because it did not receive the acceptance of service transmitted by node z on channel c_0; hence, simultaneous transmissions from node w to v and from node x to z are allowed on channel c_1, i.e., no exposed terminal problem. However, in the absence of the exposed terminal problem, it is possible that node w announces a service to node y on time slots $\{2, 3, 5\}$ after node x did the same announcement to node z (note that simultaneous transmissions from node w to y and from node x to z result in a collision at node y). In this case, if node y accepts the service and includes $\beta_1(w) = \{2, 3, 5\}$ in the AcS field of its packet transmitted on channel c_0 (on time slot $\{7\}$), node x will receive this packet transmitted by node y, includes $\beta_1(w)$ to $T_1(x)$, and avoids using the upcoming time slots $\{2, 3, 5\}$ on channel c_1 to prevent collision at node y (recall the definition of $T_1(x)$), although node x was supposed to transmit a packet to node z on the time slot $\{3\}$ following node y's acceptance of service. This missing packet, together with the other packets incorrectly received by node z, are (re)transmitted by node x after it receives the acknowledgment packet from node z. The acknowledgment packet is transmitted using the same procedure as illustrated in Fig. 6.

3.4 Multihop Broadcast Service

This section shows that the efficient multihop broadcast service presented in [12] for ADHOC MAC can be directly supported by VeMAC on channel c_0. Suppose node x transmits a broadcast packet on channel c_0, and consider that this packet needs to propagate throughout the whole network. For each node i which receives the broadcast packet, define $Z(i)$ as the set of one-hop neighbors of node i which did not receive the packet broadcast by node x. Node i does not relay the packet if one of the following holds:

- $Z(i) = \phi$;
- $\exists j \in N(i) \backslash Z(i)$ such that $Z(i) \subseteq N(j)$ and $|N(j)| > |N(i)|$, where $|\cdot|$ denotes the cardinality of a set;
- $\exists j \in N(i) \backslash Z(i)$ such that $Z(i) \subseteq N(j)$, $|N(j)| = |N(i)|$, and $\mathrm{ID}(j) > \mathrm{ID}(i)$.

When node i receives the broadcast packet from node x, it listens to channel c_0 for $s_0 - 1$ successive time slots. At the end of this duration, node i can determine the sets $N(j)$, $\forall j \in N(i)$, and $Z(i) = \{j \in N(i) : x \notin N(j)\}$. Accordingly, node i relays the packet if none of the previous three conditions is satisfied. By using this relaying procedure, it is shown in [12] and [15] that, in most cases, the minimum set of relaying nodes needed to cover the whole network is selected.

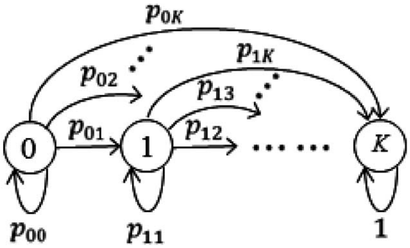


Fig. 7. Markov chain for X_n when $K \leq N$.

4 Performance Analysis

4.1 Time Slot Acquisition

The objective of the analysis in this section is to investigate how fast the contending nodes can acquire a time slot on channel c_0 by using the VeMAC protocol. Let K denote the number of contending nodes, each of which needs to acquire a time slot on channel c_0. We want to determine the average number of nodes which acquire a time slot within n frames, the probability that a specific node acquires a time slot within n frames, and the probability that all the nodes acquire a time slot within n frames. To simplify the analysis, the following assumptions are made:

1. all the contending nodes belong to the same set of THSs, with the same T_0 and A sets, e.g., node w and node x in its final position in Fig. 4;
2. the set of THSs to which the contending nodes belong does not change;
3. the set A is not augmented when a node fails to acquire a time slot after τ frames, i.e., $\tau = 0$;
4. at the end of each frame, each node is aware of all acquired time slots during the frame, and updates the sets T_0 and A accordingly, i.e., all nodes are within the communication range of each other;
5. at the end of each frame, all contending nodes are informed whether or not their attempts to access a time slot during this frame were successful. Based on this information, each colliding node randomly chooses an available time slot from the updated A set, and attempts to access this slot during the coming frame.

Let N be the number of initially available time slots in a frame, and X_n be the total number of nodes which acquired a time slot within n frames. Under the assumptions, X_n is a stationary discrete-time Markov chain with the following transition probabilities:

If $K \leq N$,

$$p_{ij} = \begin{cases} \dfrac{W(j-i, K-i, N-i)}{(N-i)^{K-i}}, & 0 \leq i \leq K-1, \\ & i \leq j \leq K \\ 1, & i = j = K \\ 0, & \text{elsewhere,} \end{cases}$$

where $W(l, u, v)$ is the number of ways by which l nodes can acquire a time slot given that there are u contending nodes each randomly choosing a time slot among v available time slots. A node acquires a time slot if no other nodes choose to access the same slot. The Markov chain is illustrated in Fig. 7.

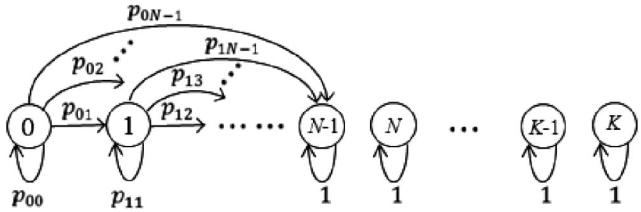


Fig. 8. Markov chain for X_n when $K > N$.

If $K > N$,

$$p_{ij} = \begin{cases} \dfrac{W(j-i, K-i, N-i)}{(N-i)^{K-i}}, & 0 \le i \le N-1, \\ & i \le j \le N-1 \\ 1, & i = j, N \le i \le K \\ 0, & \text{elsewhere.} \end{cases}$$

The Markov chain is illustrated in Fig. 8. To calculate $W(l, u, v)$, considering u different balls randomly distributed in v different boxes with equal probabilities, $W(l, u, v)$ is the number of ways of having l boxes each containing exactly one ball. This special occupancy problem is solved in a recursive way as follows [16]:

If $u \le v$,

$$W(l,u,v) = \begin{cases} C_l^u A_l^v \Big((v-l)^{u-l} - \\ \displaystyle\sum_{i=1}^{u-l} W(i, u-l, v-l) \Big), & 0 \le l < u \\ A_l^v, & l = u \\ 0, & l > u, \end{cases}$$

where $A_l^v = \frac{v!}{(v-l)!}$ and $C_l^u = \frac{A_l^u}{l!}$.

If $u > v$,

$$W(l,u,v) = \begin{cases} C_l^u A_l^v \Big((v-l)^{u-l} - \\ \displaystyle\sum_{i=1}^{v-l} W(i, u-l, v-l) \Big), & 0 \le l < v \\ 0, & l \ge v. \end{cases}$$

Let P be the one-step transition probability matrix, and P^n the n-step transition probability matrix. Given that initially all nodes are contending for time slots, i.e., $X_0 = 0$ with probability 1, the unconditional probability distribution of X_n is represented by the first row of P^n. That is,

$$p(X_n = i) = P^n_{1,i+1}, i = 0, \ldots, K.$$

The probability that all nodes acquire a time slot within n frames is

$$F_n^{all} = p(X_n = K) = P^n_{1,K+1}.$$

The average number of nodes which acquire a time slot within n frames is

$$\mu_n = \sum_{i=0}^{K} i P^n_{1,i+1}.$$

The probability that a specific node, say node x, acquires a time slot within n frames is

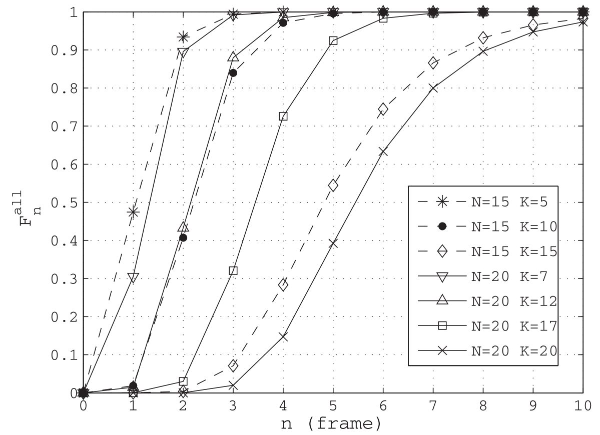


Fig. 9. Probability that all nodes acquire a time slot within n frames.

$$\begin{aligned} F_n &= \sum_{i=0}^{K} p(E|X_n = i) p(X_n = i) \\ &= \sum_{i=1}^{K} \frac{C_{i-1}^{K-1}}{C_i^K} P^n_{1,i+1} = \frac{\mu_n}{K}, \end{aligned}$$

where E is the event that node x acquires a time slot within n frames and $p(E|X_n = i) = \frac{C_{i-1}^{K-1}}{C_i^K} = \frac{i}{K}$ because all nodes have equal chances of acquiring a time slot. Note that, because the VeMAC assumes a fixed number of constant duration time slots in a frame on channel c_0, the choice of the s_0 value should always ensure that $K \le N$. However, the analysis of the protocol for the case $K > N$ can be useful in the future to determine an optimal value for s_0. This analysis gives an indication of how the protocol will behave if the number of nodes in a THS becomes larger than s_0.

Fig. 9 illustrates F_n^{all} for different values of N and K. As shown in Fig. 9, in a dense scenario such as ($N = 15$, $K = 15$), there is a probability greater than 0.9 that all the contending nodes acquire a time slot within eight frames. Hence, given a frame duration of 35 ms (as discussed in Section 4.2), the simplifying assumption of invariant THSs (assumption 2) is acceptable, because it is reasonable to assume that the THS configuration remains constant for a sufficiently large time after all the contending nodes acquire a time slot. The analysis presented in this section is verified in [1] via Matlab simulations.

4.2 Protocol Overhead and Packet Delay

As discussed in Section 3.1, a VeMAC packet transmitted on channel c_0 consists of header, *Ans*, *Acs*, and high-priority short applications fields. The size of each field in a packet transmitted by a node, x, is estimated as follows: The main part of the header consists of announcing the set $N(x)$ and the time slot used by each node in $N(x)$ on channel c_0. On the other hand, the main parts of the *Ans* and *Acs* fields consist of the sets $\beta_m(x)$ and $\beta_n(x)$ of time slots over which node x is offering service or will receive packets on service channels c_m and c_n, respectively. If the maximum number of nodes which can exist in a THS is N_{max}, at least $\lceil \log_2 N_{max} \rceil$ bits are required to represent a node ID, where $\lceil \cdot \rceil$ denotes the ceiling function. Similarly, $\lceil \log_2 s_i \rceil$ bits are sufficient to identify a time slot in a frame on channel $c_i, i = 0, \ldots, M$. Therefore, the total VeMAC packet size S (in bits) is

TABLE 1
Simulation Parameters

Parameter	Highway	City
Highway length	$1km$	–
# horizontal streets	–	3
# vertical streets	–	3
City street length	–	$430m$
# city blocks	–	4
City block edge length	–	$200m$
# lanes/direction	2	1
Lane width	$5m$	$5m$
Speed mean value	$100km/h$	$50km/h$
Speed standard deviation	$20km/h$	$10km/h$
Transmission range	$150m$	$150m$
# slots/frame	100	100
# slots for left directions	50	50
# slots for right directions	50	50
# slots for RSUs	0	0
Slot duration	$1ms$	$1ms$
Simulation time*	2 min.	2 min.
# vehicles	80 to 280 (step = 20)	150 to 600 (step = 50)
THSO	0.24 to 0.96 (step = 0.06)	0.17 to 0.70 (step = 0.06)

**Each minute is simulated with a unique initial distribution of the vehicles.*

$$S = |N(x)|\big(b_{ID} + \lceil \log_2 s_0 \rceil\big) + |\beta_m(x)|\lceil \log_2 s_m \rceil + |\beta_n(x)|\lceil \log_2 s_n \rceil + b_{app} + b_{extra},$$

where b_{ID} is the number of bits to represent a node ID, b_{app} is the number of bits for the high-priority application field, and b_{extra} is the number of bits for all other information in the packet such as MAC addresses, node x position, priority fields, error correcting codes, and so on. Assuming $N_{max} = 100$ (as assumed in [12]), $b_{ID} = 1$ byte, $s_i = 100, i = 0, \ldots, M$, $b_{app} = 200$ bytes, $b_{extra} = 30$ bytes, $|\beta_m(x)| = |\beta_n(x)| = 10$, and $|N(x)| = 100$, the estimated VeMAC packet size in this case is $S = 3{,}480$ bits ≈ 435 byte. While the information in the header, *AnS* and *AcS* fields, represents an overhead of the VeMAC protocol on channel c_0, the communications over the service channels $c_i, i = 1, \ldots, M$, are overhead free.

The delay that a high-priority safety packet experiences on channel c_0 depends on the value of s_0 as well as the duration of a time slot. Considering a maximum VeMAC packet size of 450 byte and a transmission rate of 12 Mbps,[8] the packet requires a transmission time of 0.3 ms. By adding guard periods and taking account of the physical layer overhead, such as the preamble and the physical layer header, a 0.35-ms slot duration can be assumed. In terms of synchronization, this slot duration is suitable as it is much larger than the jitter of the 1PPS of GPS receivers, which is usually in the order of nanoseconds. Assuming that $s_0 = 100$ slots, the duration of one complete frame on channel c_0 is 35 ms. Hence, when a node is successfully acquiring a time slot, it can transmit its high-priority safety messages once every 35 ms, which is compliant with the 100 ms maximum delay requirements for most of the safety applications in [2]. However, due to packet queuing, the total delay that a safety packet encounters on channel c_0 can be larger than the duration of one complete frame. For this reason, a detailed packet delay analysis on channel c_0 will be considered in the future.

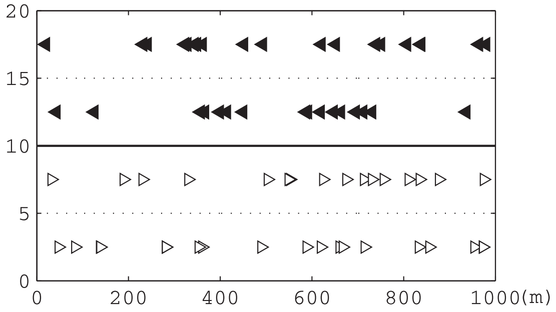


Fig. 10. A snap shot of the simulated highway segment.

5 Simulations

This section presents Matlab simulation results to evaluate the performance of VeMAC as compared with ADHOC MAC in accessing channel c_0 in highway and city scenarios.

5.1 Simulation Scenarios and Performance Metrics

The first scenario under consideration is a segment of a two-way vehicle traffic highway. A vehicle can communicate with all the vehicles within its communication range, i.e., no obstacles. Each vehicle moves with a constant speed drawn from a normal distribution, and the number of vehicles on the highway segment remains constant during the simulation time. When a vehicle reaches one end of the highway segment, it reenters the segment from the other end. For this reason, to prevent the unrealistic merging collisions caused by vehicles which jump from one end to the other end, if a vehicle is located at a distance $d \leq R$ (R is the communication range) from one end of the highway segment, it can communicate with vehicles located within a distance $R - d$ from the other end of the segment. In this way, for each traffic direction, the vehicles at the end of the segment act as if they are following the vehicles at the start of the segment.

The second scenario is a city grid layout consisting of three horizontal and three vertical two-way vehicle traffic streets. All the streets have the same dimensions, and the horizontal and vertical streets are evenly spaced resulting in four identical square city blocks.[9] The area of intersection of a horizontal street with a vertical one is referred to as a junction area. Each vehicle moves with a constant speed drawn from a normal distribution. When a vehicle reaches a junction area, it chooses one of all possible moving directions with equal probability (vehicles are not allowed to leave the simulation area during the simulation time). A vehicle located at a junction area can communicate with vehicles within its communication range located on both streets intersecting at the junction area. On the other hand, a vehicle located at a street but not at a junction area cannot communicate with vehicles located on other streets due to the existence of city blocks which obstruct the wireless signal.

For both scenarios under consideration, all the transmitted packets are broadcast packets, the wireless channel is ideal, and the only source of packet errors is the transmission collision. Table 1 summarizes the simulation parameters and Figs. 10 and 11 show snap shots of the simulated highway and city scenarios, respectively, where

[8] The 12 Mbps is one of the mandatory supported bit rates by the IEEE 802.11p OFDM physical layer for the 5-GHz band.

[9] A city block is the smallest area that is surrounded by streets.

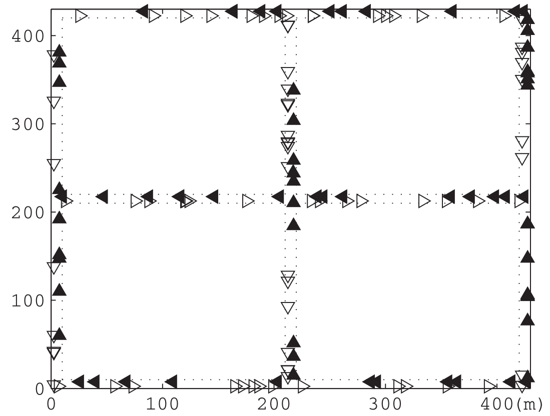


Fig. 11. A snap shot of the simulated area of the city.

the black and white triangles represent vehicles moving in opposite directions.

We define a parameter, called the THS occupancy (THSO), equal to $N_v \times \frac{2R}{L_h} \times \frac{1}{s_0}$ or $\frac{N_v}{N_s} \times \frac{2R}{L_s} \times \frac{1}{s_0}$ in the highway and city scenarios, respectively, where N_v is the total number of vehicles, N_s is the total number of streets in the city, R is the communication range, L_h is the length of the highway segment, L_s is the length of a city street, and s_0 is the number of slots per frame on channel c_0. Note that, the ratio $\frac{N_v}{N_s}$ approximately equals the number of vehicles on a city street, the number s_0 represents the maximum number of time slots available for a THS, and the length $2R$ is the maximum length that a THS can occupy on the highway segment or on a city street. Consequently, the THSO indicates the ratio of the number of time slots required by a THS to the total number of time slots available for a THS. However, the THSO is not guaranteed for each THS in the simulations. The reason is that, if there are N_v moving vehicles, this does not mean that at each instant, each THS on the highway consists of $N_v \times \frac{2R}{L_h}$ vehicles or each THS in the city consists of $\frac{N_v}{N_s} \times \frac{2R}{L_s}$ vehicles. Also, in the city scenario under consideration, a THS located near a junction area can occupy a length on the streets up to $4R$ ($2R$ on each of the horizontal and vertical street intersecting at the junction area).

The following performance metrics are considered:

1. Rate of merging collisions: the average number of merging collisions per frame per THS;
2. Rate of access collisions: the average number of access collisions per slot per THS;
3. Tx throughput: the average number of successful transmissions per slot per THS. A transmission by a vehicle x in a certain time slot is considered successful iff no other vehicles in the two-hop neighborhood of x transmits in the same slot;
4. Rx throughput: the average number of successfully received packets per slot per THS. As mentioned, packet errors only happen due to transmission collision.

Each of the metrics is calculated first for the whole simulation area, and then multiplied by $\frac{2R}{L_h}$ or $\frac{1}{N_s} \times \frac{2R}{L_s}$ for the highway and city scenarios, respectively. Note that, unlike the other three metrics, the rate of merging collisions is calculated per frame not per slot. The reason is that, merging collisions happen due to the movement of the vehicles, which is negligible in the duration of one time slot. The metrics are obtained for each of the MAC protocols mentioned in Section 5.2. At the beginning of the simulations, the vehicles are randomly (uniformly) placed on the highway segment and on all streets of the city. The vehicles remain stationary and try to acquire a time slot by using the MAC protocol under consideration. Once no more vehicle can acquire a time slot, the vehicles begin moving and the simulation timer starts. The objective of this process is to quickly bring the system to a steady state where most of the vehicles have acquired a time slot.

5.2 Simulated Protocols

Two versions of the VeMAC protocol are considered: VeMAC with $\tau = 0$ and $\tau = \infty$. As will be shown in Section 5.3, both versions of the VeMAC protocol significantly outperform the ADHOC MAC protocol in [12]. The poor performance of ADHOC MAC is caused by the following two main reasons. First, due to the lack of a condition similar to the SRP condition in VeMAC, when two vehicles having acquired a time slot enter the communication range of each other, one of them releases its time slot even if no merging collision happens. Second, as mentioned in [12], a node which needs to acquire a time slot should attempt transmission in the next available time slot with probability p. For a certain time slot, the optimal probability $p_{opt} = 1/N_c$, where N_c is the number of contending nodes attempting to acquire this time slot [12]. However, because N_c is not known to any of the contending nodes, each contending node x sets $N_c = N_{max} - N_{succ}(x)$, where N_{max} is the maximum number of nodes which can exist in a THS and $N_{succ}(x)$ is the number of nodes in the two-hop neighborhood of node x which have successfully acquired a time slot as derived from the framing information received by node x [12]. This estimation of N_c is far from accurate. The reason is that, if a node x detects that $N_{succ}(x)$ nodes have successfully acquired a time slot, this does not mean at all that there are $N_{max} - N_{succ}(x)$ nodes which need to acquire a time slot in the two-hop neighborhood of node x. Also, even if there are exactly $N_{max} - N_{succ}(x)$ contending nodes, they do not necessarily contend for the same time slots because each of the nodes may belong to a different set of THSs. Additionally, N_{max} is not constant because it depends on parameters such as the intervehicle distance and the number of lanes, which considerably vary based on the scenario (i.e., highway, city, urban, suburban, or rural areas).

Based on the two limitations of the typical ADHOC MAC protocol [12], two more versions of ADHOC MAC are considered in the simulations: the ADHOC-enhanced and the ADHOC-optimal. The ADHOC-enhanced eliminates the first limitation of ADHOC MAC by using a condition similar to the SRP condition of VeMAC. More precisely, a node x does not release its time slot based on a packet received from a node y unless node x has previously received a packet from node y, i.e., unless node y is included in the framing information [12] constructed by node x. For both ADHOC MAC and ADHOC-enhanced, the probability of accessing an available time slot by a contending node x is $p = \frac{1}{s_0 - N_{succ}(x)}$. Note that, N_{max} is replaced by s_0 (i.e., the

TABLE 2
The Simulated Protocols

Protocol	Abbreviation
VeMAC with $\tau = \infty$	V-inf
VeMAC with $\tau = 0$	V0
ADHOC MAC as in [12]	ADHOC
ADHOC-enhanced	AE
ADHOC-optimal	A-opt

maximum number of slots available for a THS) as it is not mentioned in [12] how to determine N_{max}. To evaluate the second limitation of ADHOC MAC, the ADHOC-optimal protocol is implemented. The ADHOC-optimal is similar to the ADHOC-enhanced protocol with the difference that, for each time slot, each contending node is aware of the number of contending nodes N_c within its two-hop neighborhood and sets $p = p_{opt} = \frac{1}{N_c}$. Note that this awareness of N_c is provided by the simulator and cannot be achieved in reality. Hence, the ADHOC-optimal is not a realistic protocol, it just represents an upper bound on the performance of ADHOC MAC. The five MAC protocols under consideration are summarized in Table 2.

To demonstrate the difference among the three ADHOC MAC versions, Fig. 12 shows the number of vehicles successfully acquiring a time slot, denoted by N_q, in the first 5 seconds of the simulation in the highway scenario. For the ADHOC protocol, due to the lack of the SRP condition, N_q drops from 60 to 20 vehicle/THS in the first second of the simulation. Also, each vehicle which releases its time slot in the first second cannot quickly acquire a new one due to the inexact probability of accessing an available time slot. For this reason, N_q remains below 20 vehicle/THS at the end of the five seconds. Unlike ADHOC MAC, in the AE protocol, the sudden decrease in N_q is eliminated thanks to the SRP condition. For this protocol, N_q decreases gradually and reaches 54 vehicle/THS at the end of the 5 seconds. On the other hand, the A-opt protocol does not show any decrease in N_q at the end of the 5 seconds because it can control the access collisions by using the optimal probability p_{opt} for accessing the available time slots. Similar behaviors of the three ADHOC MAC versions were seen in the city scenario (results are omitted).

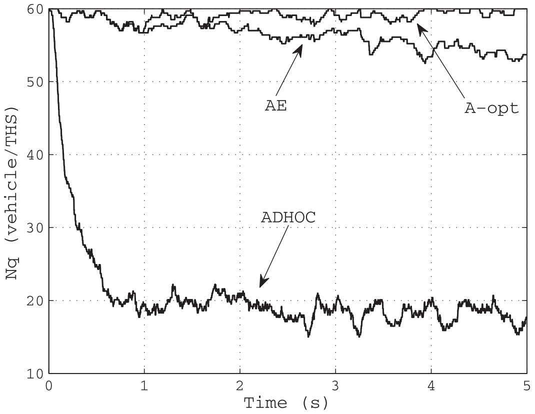


Fig. 12. The number of vehicles acquiring a time slot for the three ADHOC MAC versions in the highway scenario, at $\mathrm{THSO} = 0.6$ (i.e., 60 vehicle/THS).

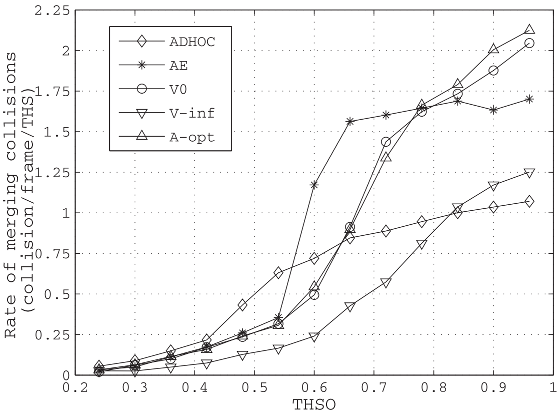


Fig. 13. The rate of merging collisions in highway.

5.3 Simulation Results

5.3.1 Highway Scenario

Fig. 13 shows the rate of merging collisions for all the MAC protocols under consideration. The V-inf protocol achieves a low rate of merging collisions because it assigns disjoint sets of time slots to vehicles moving in opposite directions. The V0 and A-opt protocols have almost the same rate of merging collisions for different THSO values. Note that for a high THSO, the ADHOC protocol provides a low rate of merging collision, even less than the V-inf protocol, due to a small number of nodes which successfully acquire a time slot as compared to other protocols (recall that, by definition, a merging collision happens only among the nodes that are successfully acquiring a time slot).

The rate of access collisions is shown in Fig. 14 for all the protocols. As expected, the A-opt protocol shows a considerably smaller rate of access collisions than both ADHOC and AE protocols, which verifies the inefficiency of both protocols in determining the probability of accessing an available slot. Due to the ability of the V-inf protocol to decrease the rate of merging collisions, as shown in Fig. 13, it also achieves a less rate of access collisions than that of the V0 protocol. The reason is that, each merging collision generates access collisions, especially for a high THSO, until each node which released its time slot reacquires a new one. Both VeMAC protocols (V-inf and V0) provide a rate of access collisions which is slightly higher than that of the A-opt protocol but

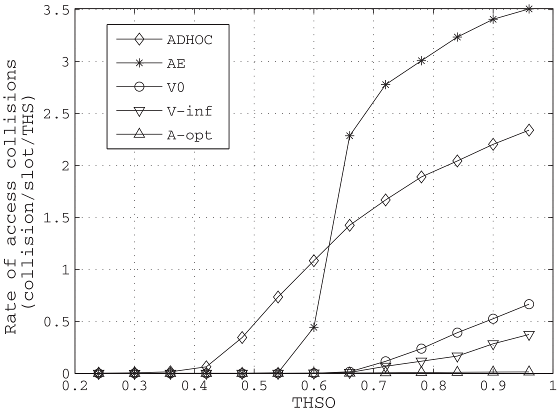


Fig. 14. The rate of access collisions in highway.

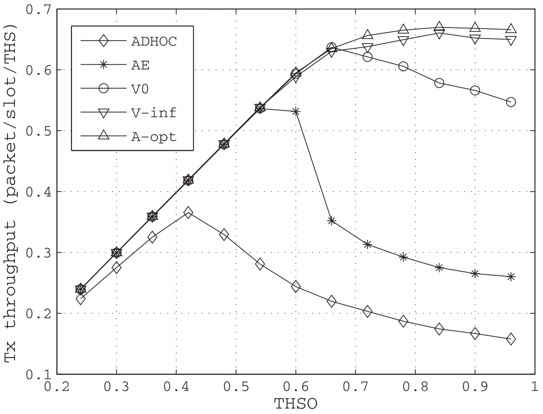


Fig. 15. The Tx throughput in highway.

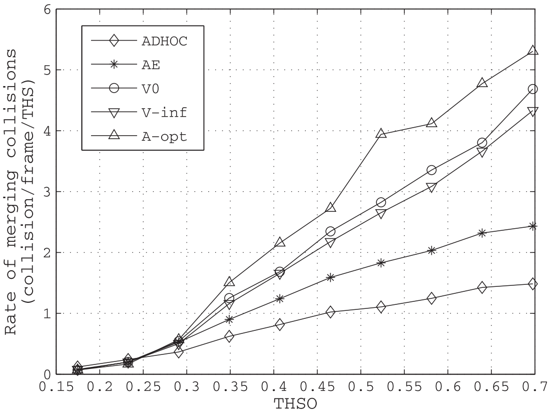


Fig. 17. The rate of merging collisions in city.

significantly lower than the rates provided by the ADHOC and AE protocols especially for a high THSO.

Fig. 15 shows the Tx throughput for all the protocols. Because of the limitations discussed in Section 5.2, the performance of the ADHOC protocol is the lowest among all the MAC protocols for all the THSO values. The AE protocol has better performance than the ADHOC protocol, but its Tx throughput decreases for a high THSO due to its inability to handle the access collisions. For a $\mathrm{THSO} < 0.7$, the V-inf and V0 protocols have almost the same Tx throughput, while for a $\mathrm{THSO} > 0.7$, the V-inf protocol starts to perform better than the V0 protocol. Both protocols outperform the AE and ADHOC protocols for all the THSO values, and the Tx throughput of the V-inf is slightly less than the unrealistic A-opt protocol for a $\mathrm{THSO} > 0.7$.

The Rx throughput is shown in Fig. 16. It is clear that, the V-inf and V0 protocols achieve a higher Rx throughput than both of the AE and ADHOC protocols for all the THSO values. For instance, at $\mathrm{THSO} = 0.78$, the V-inf protocol provides an Rx throughput of 51 packet/slot/THS as opposed to only 21 packet/slot/THS in the case of the ADHOC protocol (i.e., a 143 percent increase in the Rx throughput). Note that, for a high THSO, even if the Tx throughput remains constant or slightly decreases, the Rx throughput continues increasing. The reason is that, for the same Tx throughput, when the number of vehicles on the highway segment increases (i.e., when the THSO increases), more vehicles can receive packets because all the packets transmitted are of broadcast type. Similar to the Tx throughput, the V-inf protocol provides a slightly less Rx throughput than the A-opt protocol. For the range of THSO considered in the highway, the maximum relative difference[10] between the Rx throughput of the V-inf and A-opt protocols is approximately 3.9 percent (achieved at $\mathrm{THSO} = 0.72$).

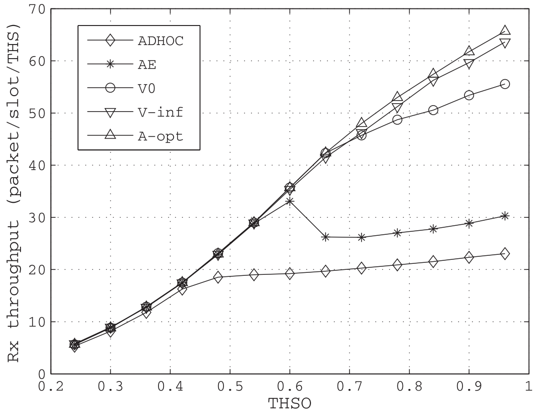


Fig. 16. The Rx throughput in highway.

5.3.2 City Scenario

The rate of merging collision in the city scenario is shown in Fig. 17 for all the protocols. It is noted that the relative difference between the rate of merging collision provided by the V-inf protocol and that provided by the V0 protocol is reduced as compared to the highway scenario. For instance, at a $\mathrm{THSO} = 0.7$ in the highway scenario, the V0 protocol shows approximately 150 percent higher rate of merging collision than the V-inf protocol, as opposed to only an 8 percent increase in the city scenario at the same THSO. The reason is that, in the city scenario, the V-inf protocol suffers from the merging collisions near the junction areas due to vehicles which change their moving direction. This kind of merging collision does not exist with the V-inf protocol when employed in the highway scenario (the merging collisions only happens among vehicles moving in the same direction). The close rate of merging collisions of both V-inf and V0 protocols also results in a close rate of access collisions, as shown in Fig. 18. Similar to the highway scenario, both V-inf and V0 protocols provide a rate of access collision, which is higher than that of the A-opt protocol but lower than those provided by the AE and ADHOC protocols.

The Tx throughput and Rx throughput are shown in Figs. 19 and 20, respectively. The V-inf and V0 protocols have the same performance for a $\mathrm{THSO} < 0.5$, while the V-inf protocol performs slightly better for a $\mathrm{THSO} > 0.5$. Unlike the highway scenario, where the A-opt and V-inf protocols have very close Tx and Rx throughputs, in the city scenario the A-opt outperforms the V-inf protocol. This outperforming is a result of the excess merging collisions that the V-inf protocol experiences in the city scenario due to vehicles which change their moving directions. However, similar to the highway scenario, both V-inf and V0 protocols provide higher Tx and Rx throughputs than the AE and ADHOC protocols.

[10] The relative difference between two values x and y is defined as $\frac{|x-y|}{\min(x,y)}$.

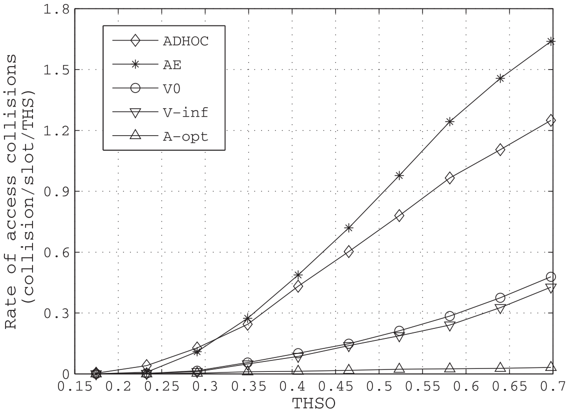


Fig. 18. The rate of access collisions in city.

6 Discussion

In terms of communication over the control channel, the main similarities and differences between the VeMAC and ADHOC MAC protocols can be summarized as follows: Both protocols are based on TDMA, work over the physical layer of different standards (such as the IEEE 802.11), and achieve an efficient multihop broadcast service as well as a reliable one-hop broadcast service without the hidden terminal problem. Also, they both require each node to periodically announce the time slots used by all nodes within its one-hop neighborhood. However, the VeMAC protocol significantly outperforms the ADHOC MAC protocol, thanks to the following three main features: the reduction of the access collision rate by using fixed time frames (versus sliding frames in ADHOC MAC) and a new method for the nodes to access the available time slots, the reduction of the merging collision rate by assigning disjoint sets of time slots to vehicles moving in opposite direction and to RSUs, and the SRP condition which prevents the nodes from unnecessarily releasing their time slots when they just enter the communication range of each other. These advantages of VeMAC come in addition to being a multichannel protocol more suitable for the DSRC spectrum as compared to the single channel ADHOC MAC protocol.

Since the VeMAC protocol assumes constant frame and slot durations on channel c_0, the value of s_0 should be large enough to accommodate all the nodes which can exist in a THS and prevent any node starvation. In a low-density scenario, when the number of nodes in a THS is much less than s_0, or in case of an event-driven high-priority safety message, it is possible that a node accesses additional time slots in a frame on channel c_0 by virtually behaving as multiple nodes. That is, the node employs the same procedure as described in Section 3.2 by using a different node ID to access each additional time slot. However, the conditions based on which a node is allowed to access more than one time slot per frame need more investigation to prevent any unfair scenario, where some nodes are accessing more than one time slot, while other nodes cannot successfully acquire a time slot.

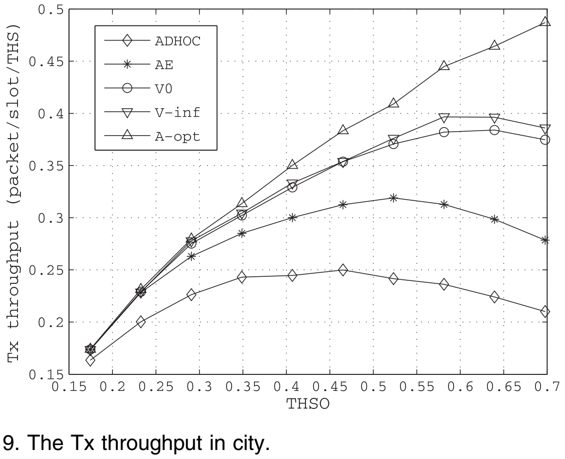


Fig. 19. The Tx throughput in city.

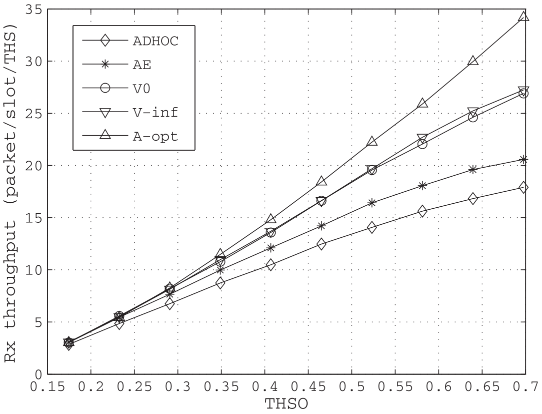


Fig. 20. The Rx throughput in city.

7 Conclusions and Future Work

In this paper, a novel multichannel TDMA MAC protocol, called VeMAC, is proposed for VANETs based on the ADHOC MAC protocol. Each node is ensured to access the control channel once per frame, and hence, nodes have equal opportunities to announce for services provided on the service channels and to transmit their high-priority application messages. The nodes access the time slots on the control channel and service channels in distributed ways, which are designed to avoid any hidden terminal problem. On the control channel, the VeMAC provides a reliable one-hop broadcast service, which is crucial for high-priority safety applications supported on this channel. As well, the efficient multihop broadcast service of ADHOC MAC can be directly supported by VeMAC on the control channel. Simulation results in highway and city scenarios show that, compared with the ADHOC MAC and ADHOC-enhanced protocols, the VeMAC provides a smaller rate of transmission collisions (access collisions and merging collisions), which results in a significantly higher throughput on the control channel.

In the future, the effect of the existence of RSUs on the performance of the VeMAC protocol will be investigated in both highway and city scenarios using realistic mobility models. As well, the performance of VeMAC using different values of the split up parameter τ, other than $\tau = 0$ and $\tau = \infty$, will be considered. Also, the mechanisms and the conditions that allow each node to access more than one time slot per frame on the control channel will be investigated. A detailed packet delay analysis will be conducted to calculate the total delay that a safety message experiences on the control channel. Another issue which will be examined is the effect of asymmetric wireless channels among the nodes as well as the effect of packet errors caused by the wireless channel impairments such as noise, fading, and shadowing. Since each node interprets any packet error as a transmission collision, the packet errors due to a poor wireless channel may result in nodes unnecessarily releasing their time slots on the control channel. Concerning the service channels, we plan to evaluate the proposed scheme for unicast via analysis and simulations, and extend this scheme to support a reliable broadcast on the service channels. Finally, the performance of VeMAC on the control channel and service channels will be compared with that of the IEEE 802.11p operating under the IEEE 1609.4 standard [17] for a multichannel operation.

Acknowledgments

The authors would like to thank the reviewers for their comments, which really helped to improve the quality of this paper. This work was presented in part at the 2011 IEEE INFOCOM Workshop [1]. This work was supported by a research grant from the Natural Science and Engineering Research Council (NSERC) of Canada.

References

[1] H.A. Omar, W. Zhuang, and L. Li, "VeMAC: A Novel Multichannel MAC Protocol for Vehicular Ad Hoc Networks," *Proc. IEEE INFOCOM,* pp. 413-418, Apr. 2011.

[2] "Vehicle Safety Communications Project Task 3 Final Report," Technical Report DOT HS 809 859, The CAMP Vehicle Safety Communications Consortium, Mar. 2005.

[3] R. Baldessari et al., "Car-2-Car Communication Consortium Manifesto," Technical Report Version 1.1, Aug. 2007.

[4] J.J. Blum and A. Eskandarian, "A Reliable Link-Layer Protocol for Robust and Scalable Intervehicle Communications," *IEEE Trans. Intelligent Transportation Systems,* vol. 8, no. 1, pp. 4-13, Mar. 2007.

[5] R. Mangharam, R. Rajkumar, M. Hamilton, P. Mudalige, and F. Bai, "Bounded-Latency Alerts in Vehicular Networks," *Proc. Mobile Networking for Vehicular Environments,* pp. 55-60, May 2007.

[6] F. Watanabe, M. Fujii, M. Itami, and K. Itoh, "An Analysis of Incident Information Transmission Performance Using MCS/CDMA Scheme," *Proc. IEEE Intelligent Vehicles Symp. (IV '05),* pp. 249-254, June 2005.

[7] H. Nakata, T. Inoue, M. Itami, and K. Itoh, "A Study of Inter Vehicle Communication Scheme Allocating PN Codes to the Location on the Road," *Proc. IEEE Intelligent Transportation Systems Conf. (ITSC '03),* vol. 2, pp. 1527-1532, Oct. 2003.

[8] *IEEE Std 802.11p-2010, Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments,* IEEE, pp. 1-51, July 2010.

[9] *IEEE Std 802.11-2007 (Revision of IEEE Std. 802.11-1999), Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE, pp. 1-1184, June 2007.

[10] M. Hassan, H. Vu, and T. Sakurai, "Performance Analysis of the IEEE 802.11 MAC Protocol for DSRC Safety Applications," *IEEE Trans. Vehicular Technology,* vol. 60, no. 8, pp. 3882-3896, Oct. 2011.

[11] S. Eichler, "Performance Evaluation of the IEEE 802.11p WAVE Communication Standard," *Proc. IEEE 66th Vehicular Technology Conf. (VTC '07-Fall),* pp. 2199-2203, Oct. 2007.

[12] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta, "ADHOC MAC: New MAC Architecture for Ad Hoc Networks Providing Efficient and Reliable Point-to-Point and Broadcast Services," *Wireless Networks,* vol. 10, pp. 359-366, July 2004.

[13] F. Borgonovo, L. Campelli, M. Cesana, and L. Fratta, "Impact of User Mobility on the Broadcast Service Efficiency of the ADHOC MAC Protocol," *Proc. IEEE 61st Vehicular Technology Conf. (VTC '05-Spring),* vol. 4, pp. 2310-2314, June 2005.

[14] W. Franz, H. Hartenstein, and M. Mauve, *Inter-Vehicle Communications Based on Ad Hoc Networking Principles: the FleetNet Project.* Universitätsverlag Karlsruhe, 2005.

[15] F. Borgonovo, L. Campelli, M. Cesana, and L. Coletti, "MAC for Ad-Hoc Inter-Vehicle Network: Services and Performance," *Proc. IEEE 58th Vehicular Technology Conf. (VTC '03-Fall),* vol. 5, pp. 2789-2793, Oct. 2003.

[16] "A Recursive Solution to an Occupancy Problem Resulting from TDM Radio Communication Application," *Applied Math. and Computation,* vol. 101, no. 1, pp. 1-3, June 1999.

[17] *IEEE Std 1609.4-2010 (Revision of IEEE Std 1609.4-2006), Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation,* IEEE, pp. 1-89, Feb. 2011.

Hassan Aboubakr Omar received the BSc degree in electronics and communications engineering in 2005 and the MSc degree in engineering mathematics in 2009, both from Cairo University, Egypt. Since 2010, he has been working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. From 2005 to 2008, he was a research assistant in the Science and Technology Research Center at the American University in Cairo. His current research interest includes medium access control, routing, and road side unit placement in vehicular ad hoc networks. He is a student member of the IEEE and the IEEE Computer Society.

Weihua Zhuang has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, since 1993, where she is a professor and a Tier I Canada research chair in wireless communication networks. Her current research focuses on resource allocation and QoS provisioning in wireless networks. She received Best Paper Awards from the IEEE Multimedia Communications Technical Committee in 2011, IEEE Vehicular Technology Conference Fall 2010, IEEE Wireless Communications and Networking Conference 2007 and 2010, IEEE International Conference on Communications 2007, and the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness 2007 and 2008. She received the Outstanding Performance Award four times since 2005 from the University of Waterloo, and the Premier's Research Excellence Award in 2001 from the Ontario Government. She is the editor-in-chief of *IEEE Transactions on Vehicular Technology*, and the Technical Program Symposia Chair of IEEE GlobeCom 2011. She was an IEEE Communications Society Distinguished Lecturer (2008-2011). She is a fellow of the IEEE, the Canadian Academy of Engineering, and the Engineering Institute of Canada, and an elected member on the Board of Governors of the IEEE Vehicular Technology Society.

Li Li received the MSc degree in electrical engineering from Southeast University, Nanjing, China, in 1990, and the PhD degree in electrical engineering from the University of Ottawa, Ontario, Canada, in 1993. From 1993 to 1999, she was with Nortel Networks Ltd. as a system architect and then product manager. From 1999 to 2003, she was the chief architecture at SS8 Networks, Inc. Since 2003, she has been with Communications Research Centre, Ottawa, Ontario, Canada, where she is a research scientist. She has contributed previously to the ITU-T and IETF standard working groups, coauthored IETF RFC, and has been awarded with several US patents. Her current research focuses on mobile tactical radio networking and adaptive networks, with particular interest in mobile network optimization, networking protocols and algorithms, and performance modeling of mobile networks. She has served as an associated editor for Springer's *Journal on Peer-to-Peer Networking and Applications*. She served as cochair for the IEEE PERCOM MP2P 2006-2008 workshops, track cochair for IEEE VTC 2010-Fall, session cochair for IEEE MILCOM 2008-2011, and track cochair for IEEE MILCOM 2012. She is an associate editor for the *IEEE Transactions on Vehicular Technology.* She is a member of the IEEE.