This paper was presented as part of the Mobility Management in the Networks of the Future World (MobiWorld) Workshop at

VeMAC: A Novel Multichannel MAC Protocol for Vehicular Ad Hoc Networks

Hassan Aboubakr Omar and Weihua Zhuang
Department of Electrical and Computer Engineering
University of Waterloo, Canada

Li Li
Communications Research Center
Ottawa, Ontario, Canada, K2H 8S2

Abstract—**This paper introduces VeMAC, a novel multichannel TDMA MAC protocol designed specifically for a vehicular ad hoc network. The network has one control channel and multiple service channels. On the control channel nodes acquire time slots in a distributed way, while on the service channels nodes are assigned time slots in a centralized manner. VeMAC decreases the probability of transmission collisions caused by node mobility by assigning disjoint sets of time slots to vehicles moving in opposite directions and to road side units. Analysis and simulation results are presented to demonstrate the efficiency of VeMAC and compare it to ADHOC MAC, an existing MAC protocol based on TDMA. It is shown that, for the same number of contending nodes and available time slots, nodes can acquire time slots on the control channel much faster in VeMAC than in ADHOC MAC, when the number of available time slots is sufficiently larger than the number of contending nodes.**

I. INTRODUCTION

A vehicular ad hoc network (VANET) consists of a set of vehicles equipped with on board units for wireless communications and a set of stationary units along the road known as road side units (RSUs). The objective of a VANET is to provide reliable vehicle to vehicle (V2V) and vehicle to RSU (V2R) communications. Based on these two kinds of communications, a VANET can support a wide variety of applications in safety, entertainment, and vehicle traffic optimization [1]. Motivated by the importance of vehicular communications, the U.S Federal Communication Commission (FCC) has allocated 75MHz in the 5.9GHz band for Dedicated Short Range Communications (DSRC) to be exclusively used by V2V and V2R communications. The DSRC spectrum is divided into seven 10MHz channels: six service channels for safety and non-safety related applications, and one control channel for transmission of control information and high priority short safety applications.

The special characteristics of VANETs, such as the highly dynamic network topology and diverse quality of service (QoS) requirements of potential applications, result in significant challenges in the design of an efficient medium access control protocol (MAC) protocol. Various MAC protocols have been proposed for VANETs based either on IEEE 802.11 [2] or on channelization such as time division multiple access (TDMA) [3], space division multiple access [4], and code division multiple access [5]. The IEEE 802.11p is a recently proposed MAC standard for VANETs. The protocol is based on the legacy IEEE 802.11 standard which is widely implemented. However, the protocol suffers from problems such as Orphan frames [6] and the unreliable broadcast service which is subject to the hidden terminal problem. ADHOC MAC is based on TDMA and is proposed for inter-vehicle communication networks [3]. Unlike the IEEE 802.11p, ADHOC MAC can support reliable broadcast service without the hidden terminal problem, and 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 frame, which is suitable for non delay-tolerant applications. However, simulation results show that, due to node mobility, the throughput reduction can reach 30% for an average vehicle speed of 50km/h [7]. Another limitation of ADHOC MAC is that it is a single channel protocol, not suitable for the seven DSRC channels.

This paper proposes VeMAC, a multichannel MAC protocol based on TDMA and designed specifically for a VANET. The VeMAC assigns disjoint sets of time slots to vehicles moving in opposite directions and to RSUs, and hence can decrease the rate of *merging collision* [7] in ADHOC MAC caused by node mobility. In addition, for the same number of contending nodes and available time slots, nodes can acquire slots on the control channel much faster in VeMAC than in ADHOC MAC, when the number of available time slots is sufficiently larger than the number of contending nodes.

II. 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. Time is partitioned to frames. A frame consists of a fixed number S of constant-duration time slots. 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 nodes moving in left and right directions respectively. Every node (i.e. vehicle or RSU) is equipped with a GPS receiver. Each vehicle can determine its direction using GPS, and synchronization among nodes can be performed using the 1PPS signal provided by any GPS

This work was supported by a research grant from the Natural Science and Engineering Research Council (NSERC) of Canada.

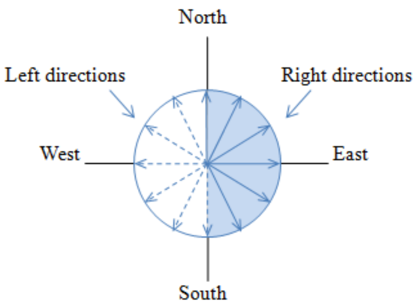


Fig. 1: Right and left directions.

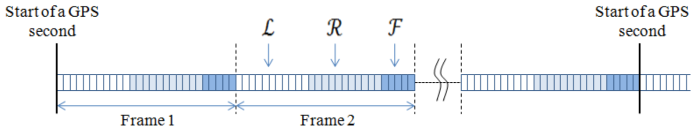


Fig. 2: Partitioning of each frame into $\mathcal{L}, \mathcal{R}$ and $\mathcal{F}$ sets.

receiver. The rising edge of this 1PPS is aligned with the start of every GPS second with accuracy within 100ns even for inexpensive GPS receivers. Consequently, this accurate 1PPS signal can be used as a common time reference among all nodes. Each second contains an integer number of frames as shown in Fig.2. Hence, at any instant, each node can determine whether the current time slot belongs to the $\mathcal{L}$, $\mathcal{R}$, or $\mathcal{F}$ set.

The VANET has one control channel (cch), and M service channels (schs), denoted by $sch_1, sch_2, \ldots, sch_M$. The cch is mainly 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 the cch and schs. The M schs are used for transmission of safety or non-safety related application messages. Each node has two transceivers: transceiver1 is always tuned to the cch, while transceiver2 can be tuned to any of the M schs. For a certain node x, the sch to which transceiver2 is currently tuned is denoted by $sch(x)$. It is assumed that the transmission power levels on the cch and schs 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. For a certain node x, the following two sets are defined:

- $N_{cch}(x)$: the set of one hop neighbours of node x on the cch, from/to which node x can receive/transmit packets on the cch;
- $N_m(x)$: the set of 'expected' one hop neighbours of node x on sch_m, $m = 1, \ldots, M$.

The set $N_m(x)$ is constructed by node x, $\forall m = 1, \ldots, M$, as follows. When node x receives a packet on the cch from another node y indicating that $sch(y) = m$, based on the position of node y which is included in the header of the packet, node x can estimate its distance to node y. Based on this estimated distance and on the fixed transmission power on sch_m which is known to node x, if node x decides that node y is in its communication range on sch_m, it adds node y to $N_m(x)$; otherwise, node x does not update $N_m(x)$.

III. VEMAC PROTOCOL

A. VeMAC Preliminaries

In the VeMAC protocol, each packet[1] transmitted on the cch 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. Information in the header, *AnS*, and *AcS* fields are necessary for a node to decide which time slots it should access on the cch and schs. On the cch, nodes acquire time slots in a distributed way and each node must acquire exactly one time slot in a frame. A provider is a node which announces on the cch for a service offered on a specific sch, while a user is a node which receives the announcement for a service and decides to make use of this service. It is the responsibility of the provider to assign time slots to all users and announce this slot assignment on the sch on a specific time slot called provider's main slot. For the purpose of time slot assignment on the cch and schs, in the header of each packet transmitted on the cch, the transmitting node x should include i) $sch(x)$ and the time slots used by node x on $sch(x)$, ii) $N_{cch}(x)$ and the time slot used by each node $y \in N_{cch}(x)$, iii) $N_m(x)$, where $m = sch(x)$, and the time slots used by each node $y \in N_m(x)$, and iv) the position and the current direction of node x (right, left, or RSU). Subsections B and C in the following explain in details how the nodes acquire time slots on the cch and schs respectively.

Header	AnS	AcS	High priority short applications

Fig. 3: Format of each packet transmitted on the cch.

Two types of collision on time slots can happen [7]: *access collision* among nodes trying to acquire time slots, and *merging collision* among nodes already acquiring time slots. *Access collision* happens when two or more nodes within two hops of each other attempt to access the same available time slot. On the other hand, *merging collision* happens when two or more nodes accessing the same time slot become members of the same two-hop set (THS)[2] due to node activation or node mobility. In VANETs, *merging collision* is likely to occur among vehicles moving in opposite directions or between a vehicle and a stationary RSU. For example, in Fig.4, if vehicle A moves to THS2, and if A is using the same time slot as D, then collision will occur at C. Upon detection of a *merging collision* on the cch, the colliding nodes should release their time slots and acquire new ones, which may generate more *access collisions*.

B. Accessing Slots on the Control Channel

Suppose node x is just powered on and needs to acquire a time slot on the cch. Node x starts listening to the cch for one complete frame. At the end of this frame, node x

---

[1] The term 'packet' is used instead of 'frame' to refer to Layer 2 Protocol Data Unit, in order to avoid confusion with 'frame' which is a collection of time slots.

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

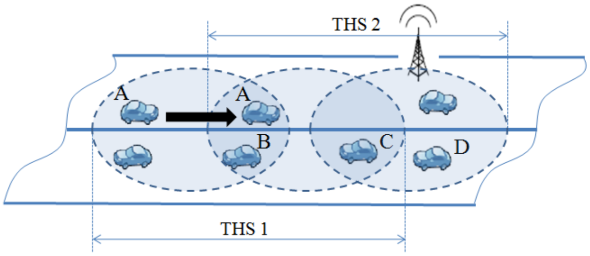


Fig. 4: *merging collision* caused by node mobility.

can determine $N_{cch}(x)$ and the time slot used by each node $i \in N_{cch}(x)$. In addition, since each $i \in N_{cch}(x)$ announces $N_{cch}(i)$ and the time slot used by each $j \in N_{cch}(i)$, node x can determine the time slot used by each of its two hop neighbours, $j \in N_{cch}(i), j \notin N_{cch}(x)$, $\forall i \in N_{cch}(x)$. Hence, by listening to one complete frame, node x can determine the set of time slots used by all nodes within its two-hop neighbourhood, denoted by $U_{cch}(x)$. This set represents the time slots that node x cannot use on the cch, in order to avoid any hidden terminal problem.

Given $U_{cch}(x)$, node x determines the set of accessible time slots $V_{cch}(x)$ (to be discussed) and then attempts to acquire a time slot by randomly accessing any time slot in $V_{cch}(x)$, say time slot k. If no other node in the two-hop neighbourhood of node x attempts to acquire time slot k, then no *access collision* happens. In this case, the attempt of node x is successful and all nodes $i \in N_{cch}(x)$ add node x to the sets $N_{cch}(i)$ and record that node x is using time slot k. On the other hand, if at least one node within the two-hop neighbourhood of node x accesses time slot k, then all the transmissions fail and time slot k is not acquired by any of the contending nodes. In this case, node x will discover that its attempt was unsuccessful as soon as it receives a packet from any node $i \in N_{cch}(x)$ indicating that node $x \notin N_{cch}(i)$. Node x then re-accesses one of the time slots in $V_{cch}(x)$, and so on until all nodes $i \in N_{cch}(x)$ indicate that node $x \in N_{cch}(i)$ and announce the time slot accessed by node x. Note that node x needs at most $S-1$ time slots to discover whether or not its attempt to acquire a time slot was successful.

Consider that node x is moving in one of the right directions. Initially, node x limits the set $V_{cch}(x)$ to the available time slots associated with the right directions, i.e. $V_{cch}(x) = \overline{U_{cch}(x)} \cap \mathcal{R}$. If after a certain number of frames, say τ frames, node x cannot acquire a time slot, then node x augments $V_{cch}(x)$ by adding the time slots associated with the opposite direction, i.e. $V_{cch}(x) = \overline{U_{cch}(x)} \cap (\mathcal{R} \cup \mathcal{L})$. 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. $V_{cch}(x) = \overline{U_{cch}(x)}$. 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 $V_{cch}(x) = \overline{U_{cch}(x)} \cap \mathcal{F}$, and then $V_{cch}(x) = \overline{U_{cch}(x)}$. The parameter τ is referred to as split up parameter, and the choice of the τ value is critical since it directly affects the rates of *access collision* and *merging collision*. For example, when $\tau = 0$, the rate of *merging collision* is maximized since vehicles moving in opposite directions and RSUs are accessing the same set of time slots. However, when *merging collision* happens, the probability of an *access collision* is minimized since each colliding node x can choose to access any time slot in $\overline{U_{cch}(x)}$. On the other extreme, when $\tau = \infty$, the rate of *merging collision* is minimized since vehicles moving in opposite directions and RSUs are accessing disjoint sets of time slots, and hence *merging collision* only happens among vehicles moving in the same direction or when a vehicle changes direction. However, when *merging collision* happens, for example among vehicles moving in a right direction, the probability of *access collision* is maximized since the choice of each colliding vehicle x is limited to time slots in $\overline{U_{cch}(x)} \cap \mathcal{R}$. How to determine a suitable value for τ to balance between the rates of *access* and *merging collisions* should be further studied.

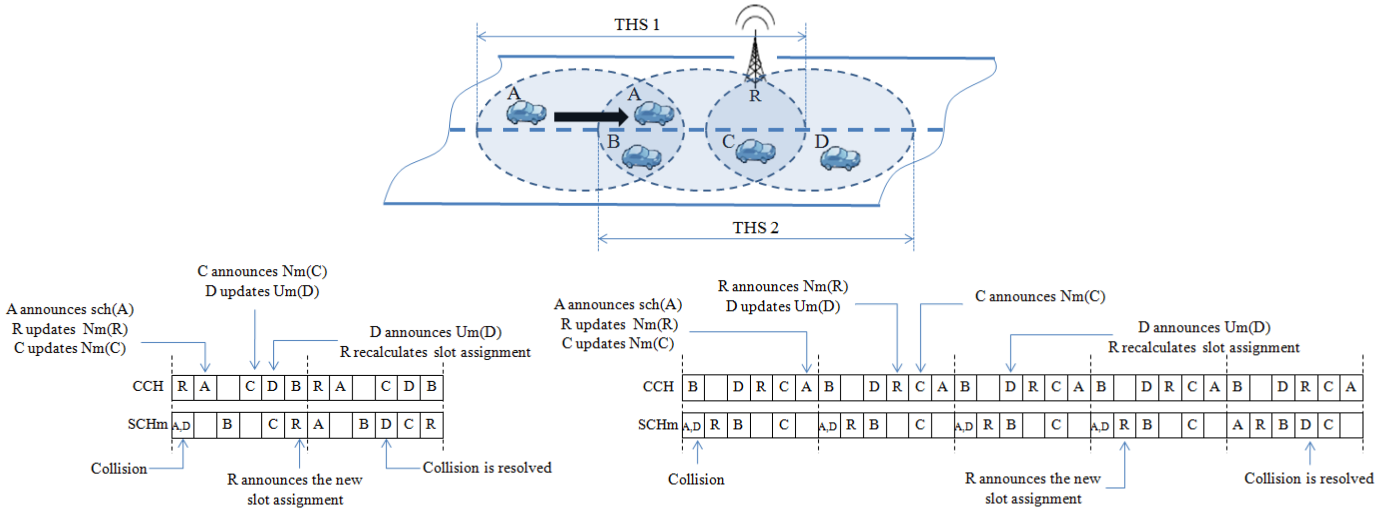
C. Accessing Slots on the Service Channels

As mentioned, the assignment of time slots to nodes on the schs is performed by the providers in a centralized way. For the slot assignment without a hidden terminal problem, each node x should determine $U_m(x)$ defined as the set of time slots used on sch_m by all nodes which are expected to be within the two-hop neighbourhood of node x on sch_m. This set represents the time slots that node x cannot use on sch_m, and will be used by the provider to assign time slots to nodes without causing any hidden terminal problem. Each node x constructs $U_m(x), \forall m = 1, \ldots, M$, as follows. When node x receives a packet on the cch from another node y indicating that $sch(y) = m$, if $y \in N_m(x)$, node x adds to $U_m(x)$ the time slots used by each node $j \in N_m(y)$; otherwise, node x does not update $U_m(x)$.

When a provider (R) has a service to offer on an sch, it announces the following information in the *AnS* field of the next packet transmitted on the cch: priority of the service, address(es) of the intended user(s), provider's main slot, and the sch on which the service will be offered. Based on the information announced by provider R on the cch, each node $x \in N_{cch}(R)$ determines whether or not to make use of the announced service. If node x decides to use the service by provider R on sch_m, it transmits the following information in the *AcS* field of the next packet transmitted on the cch: $U_m(x)$, address of provider R, and the number of time slots that node x needs. Once node x indicates its acceptance of the service, it tunes transceiver2 to sch_m and waits for the time slot assignment transmitted on the provider's main slot.

After the provider announces the service, it listens to the cch for the duration of one complete frame to determine the set $\mathcal{I}$ of users which are interested in the service. For each node $x \in \mathcal{I}$, the provider has $U_m(x)$, the movement direction of node x, and the number of time slots required by node x. Accordingly, the provider assigns time slots for each $x \in \mathcal{I}$ and announces this slot assignment in the main slot.

Because of node mobility, $U_m(x)$ may change for some node(s) $x \in \mathcal{I}$. Hence, the slot assignment needs to be recalculated based on the new $U_m(x)$. For this reason, each



(a) Collision remains for one frame.

(b) Collision remains for four frames

Fig. 5: Collision resolving on sch_m

user x keeps on transmitting the updated $U_m(x)$ in the *AcS* field of each packet transmitted on the cch. The provider is therefore aware of the updates in $U_m(x)$, $\forall x \in \mathcal{I}$, and can recalculate the slot assignment either periodically or when there is a change in $U_m(x)$ for any node $x \in \mathcal{I}$. However, when a collision happens, some delay may exist until the collision is resolved and new time slots are assigned by the provider to the colliding node(s). This delay depends mainly on the time slot assignment on the cch and sch_m. For example, consider two extreme cases for the THS configuration shown in Fig.5. Initially, nodes B, C, and D use the service by provider R on sch_m, while node $A \notin N_{cch}(R)$ uses the same time slot as node D on sch_m. When node A moves to THS2, transmissions from nodes A and D on sch_m collides at node C and provider R. Assuming no collision happens on the cch, for the slot assignment in Fig.5a, collision is resolved in the next frame; while for the slot assignment in Fig.5b, collision remains for four frames until node D is assigned a new slot by the provider. The effect of this delay on VeMAC performance needs further investigation.

D. Broadcast Service

This subsection shows that the efficient broadcast service presented in [3] for ADHOC MAC can be directly supported by VeMAC on the cch and schs, and hence no layer 3 broadcast protocol is required. Suppose node x transmits a broadcast packet on the cch. For each node $i \in N_{cch}(x)$, define Z_i as the set of one-hop neighbours 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:

- $Z_i = \phi$;
- $\exists j \in N_{cch}(i) \backslash Z_i$ such that $Z_i \subseteq N_{cch}(j)$ and $|N_{cch}(j)| > |N_{cch}(i)|$;
- $\exists j \in N_{cch}(i) \backslash Z_i$ such that $Z_i \subseteq N_{cch}(j)$, $|N_{cch}(j)| = |N_{cch}(i)|$, and $ID(j) > ID(i)$; where $ID(i)$ denotes the address of node i.

When node i receives a broadcast packet from node x, it listens to the cch for S successive time slots. At the end of this duration, node i can determine the sets $N_{cch}(j)$, $\forall j \in N_{cch}(i)$, and $Z_i = \{j \in N_{cch}(i) : x \notin N_{cch}(j)\}$. Accordingly, node i relays the packet if none of the previous three conditions is satisfied. By using this broadcasting procedure, it is shown in [3] and [8] that, in most cases, the minimum set of relaying nodes needed to cover the whole network is selected. The previous broadcasting procedure can be applied to sch_m by replacing $N_{cch}(\bullet)$ with $N_m(\bullet)$. However, on sch_m, the broadcast service is less reliable since $N_m(\bullet)$ is the set of 'expected' (not the actual) one-hop neighbors on sch_m.

IV. PERFORMANCE EVALUATION

Consider K contending nodes, each of which needs to acquire a time slot on the cch. We want to determine the average number of nodes which acquire time slots 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: a) all the contending nodes belong to the same set of THSs, with the same U_{cch} and V_{cch}, e.g. nodes C and R in Fig.5; b) the set of THSs to which the contending nodes belong does not change; c) the set V_{cch} is not augmented when a node fails to acquire a time slot after τ frames, i.e. $\tau = 0$; d) at the end of each frame, each node is aware of all acquired time slots during the frame, and updates U_{cch} and V_{cch} accordingly, i.e. all nodes are within the communication range of each other; e) 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 V_{cch}, and attempts to access this slot during the coming frame.

Let N be the number of initially available time slots, and X_n be the total number of nodes which acquired time slots within

n frames. Under the assumptions, X_n is a stationary discrete-time Markov chain with the following transition probabilities. If $K \leq N$,

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

where $W(l, v, s)$ is the number of ways by which l nodes can acquire a time slot given that there are v contending nodes each randomly choosing a time slot among s 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.6.

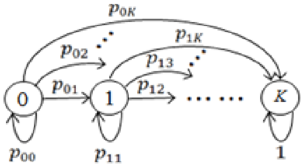


Fig. 6: Markov chain for X_n when $K \leq N$.

If $K > N$,

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

The Markov chain is illustrated in Fig.7. To calculate $W(l, v, s)$, considering v different balls randomly distributed in s different boxes with equal probabilities, $W(l, v, s)$ 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 [9].

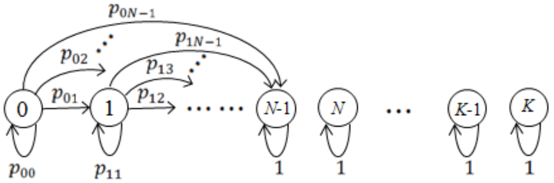


Fig. 7: Markov chain for X_n when $K > N$.

If $v \leq s$,

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

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

If $v > s$,

$$W(l, v, s) = \begin{cases} C_l^v A_l^s \Big((s-l)^{v-l} - \\ \sum_{i=1}^{s-l} W(i, v-l, s-l) \Big), & 0 \leq l < s \\ 0, & l \geq s. \end{cases}$$

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

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

where E is the event that node x acquires a time slot within n frames and $p(E|X_n = i) = \frac{C_{i-1}^{K-1}}{C_i^K} = \frac{i}{K}$ since all nodes have equal chances of acquiring a time slot.

Fig.8 shows F_n^{all} for different values of N and K. It is observed that, with a probability around 0.9, all nodes acquire a time slot within two frames for the cases $(N = 15, K = 5)$ and $(N = 20, K = 7)$, four frames for the cases $(N = 15, K = 10)$ and $(N = 20, K = 12)$, and eight frames for the cases$(N = 15, K = 15)$ and $(N = 20, K = 20)$.

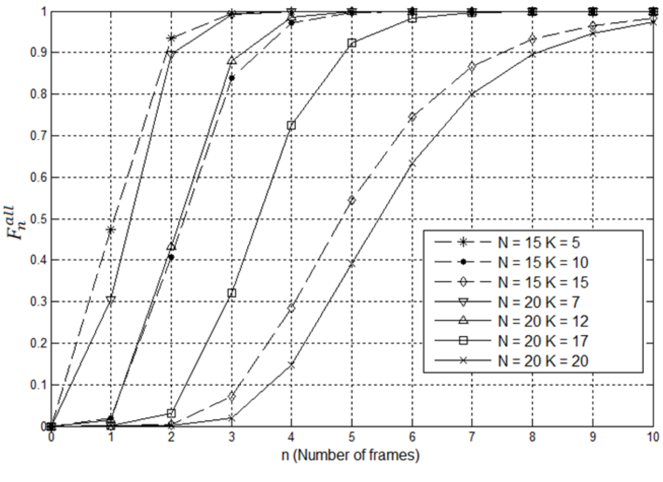


Fig. 8: Probability that all nodes acquire a time slot within n frames.

Simulations have been conducted using MATLAB to verify the analysis. In the simulations, assumption e) is removed. This assumption is not very realistic since in VeMAC, node x detects collision as soon as it receives a packet from any node $i \in N_{cch}(x)$ indicating that node $x \notin N_{cch}(i)$. Consequently, it is not realistic that all nodes detect collision together at the end of each frame. Some nodes may detect collision earlier

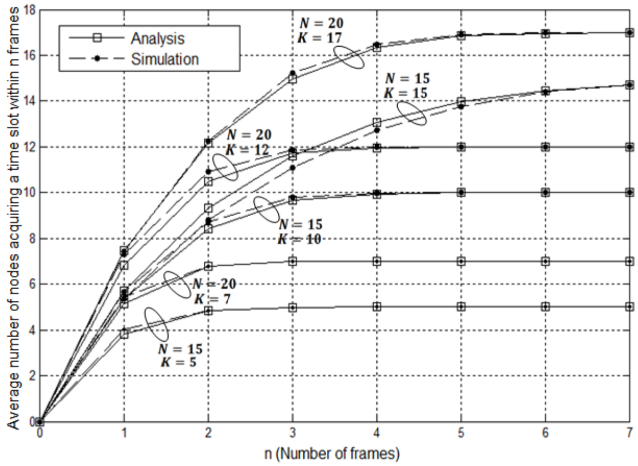


Fig. 9: Average number of nodes acquiring a time slot within n frames.

or later based on which slots they attempted to access during this frame. The average number of nodes which acquire a time slot within n frames is calculated for different K and N, and denoted by μ_n^{sim}. The 98% confidence interval of μ_n^{sim} is less than 0.33 node for all n, K, and N. As shown in Fig.9, the results of μ_n^{sim} obtained from simulations without assumption e) are very close to μ_n obtained from analysis for different K and N.

Finally, VeMAC is compared with ADHOC MAC in terms of μ_n^{sim} via simulations. For VeMAC assumptions a) to d) hold, while for ADHOC MAC all nodes are assumed to be within the communication range of each other (assumption d)). Simulation results obtained for ADHOC MAC matches the results originally presented in [3] for all K and N, with the only difference that we neglect the first frame in which each node listens to the channel to determine which slot the node is allowed to access. This is necessary for the comparison to be fair, as in VeMAC all nodes are initially aware of the sets U_{cch} and V_{cch}. Fig.10 shows μ_n^{sim} for ADHOC MAC and VeMAC with different values of K and N. It is clear that, when $N > K$, nodes acquire a time slot much faster in VeMAC than in ADHOC MAC. For example, for $(N = 100, K = 50)$ and $(N = 200, K = 100)$, all nodes acquire a time slot within three frames in VeMAC, which increases to six frames in ADHOC MAC. This result indicates that, for the same N and K, when $N > K$ VeMAC can decrease the rate of *access collision*, as compared to ADHOC MAC. When K increases, the gap between the performances of the two protocols decreases, and ADHOC MAC performs slightly better when $K = N$.

V. CONCLUSIONS AND FUTURE WORK

This paper proposes VeMAC, a multichannel MAC protocol based on TDMA for VANETs. 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 ways that nodes acquire time slots on the control channel and service channels are designed to avoid any hidden terminal problem. Each node is provided with full knowledge of the channel access of its one and two hop neighbours, which is useful information for some layer 3 protocols. Compared to ADHOC MAC, VeMAC can make use of the seven DSRC channels, support the same broadcast service on the control channel and service channels, and decrease the rates of *merging* and/or *access collision* based on the value of the split up parameter τ. In the future, we plan to determine the value of τ in order to balance between *merging* and *access collisions*, to determine how the provider should assign time slots to users on a service channel, and to investigate the performance of VeMAC via simulations with realistic mobility models.

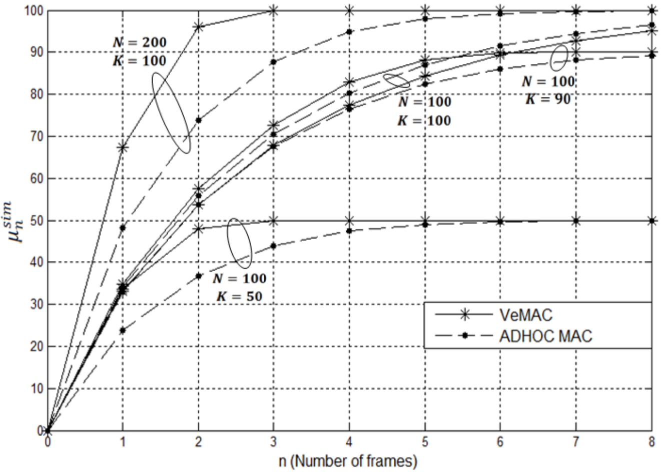


Fig. 10: Average number of nodes acquiring a time slot within n frames.

REFERENCES

[1] Manifesto of the Car-to-Car Communication Consortium, www.car-to-car.org, 2007.

[2] 802.11p-2010 IEEE 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, 2010.

[3] 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, no. 4, pp. 359-366, 2004.

[4] J. B. Jeremy and E. Azim, "A reliable link-layer protocol for robust and scalable intervehicle communications,"*IEEE Trans. Intelligent Transportation Systems*, vol. 8, no. 1, pp. 4-13, 2007.

[5] 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*, vol. 2, pp. 1527-1532, 2003.

[6] C. Nakjung, C. Sungjoon, S. Yongho, K. Taekyoung, and C. Yanghee, "A solicitation-based IEEE 802.11p MAC protocol for roadside to vehicular networks,"*Proc. Mobile Networking for Vehicular Environments*, pp. 91-96, 2007.

[7] 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 VTC*, vol. 4, pp. 2310-2314, 2005.

[8] F. Borgonovo, L. Campelli, M. Cesana, and L. Coletti, "MAC for ad-hoc inter-vehicle network: services and performance,"*Proc. IEEE VTC*, vol. 5, pp. 2789-2793, 2003.

[9] J- H. Wen and J- W. Wang, "A recursive solution to an occupancy problem resulting from TDM radio communication application,"*Applied Mathematics and Computation*, vol. 101, no. 1, pp. 1-3, 1999.