Self-Configuring TDMA Protocols for Enhancing Vehicle Safety With DSRC Based Vehicle-to-Vehicle Communications

Fan Yu, *Student Member, IEEE*, and Subir Biswas, *Senior Member, IEEE*

***Abstract*— This paper presents a novel Medium Access Control protocol for inter-vehicular wireless networking using the emerging Dedicated Short Range Communication (DSRC) standards. The main contribution of the paper is the design of a self-configuring TDMA protocol capable of inter-vehicle message delivery with short and deterministic delay bounds. The proposed Vehicular Self-Organizing MAC (*VeSOMAC*) is designed to be vehicle location and movement aware so that the MAC slots in a vehicle platoon can be time ordered based on the vehicles' relative locations for minimizing the multi-hop delivery delay. A novel feature of *VeSOMAC* is its in-band control mechanism for exchanging TDMA slot information during distributed MAC scheduling. It is shown that by avoiding explicit timing information exchange, *VeSOMAC* can work without inter-vehicle time synchronization. The in-band control mechanism is also used for fast protocol convergence during initial network setup and topology changes due to vehicle movements. A simulation model has been developed for comparing *VeSOMAC's* performance with that of DSRC-recommended 802.11 MAC protocol for highway traffic safety applications.**

***Index Terms*— Inter-vehicle networks, MAC, self-configuration, intelligent transportation system.**

I. INTRODUCTION

A. Background and Motivation

THE INTELLIGENT Transportation System (ITS) [1] architecture is currently being developed for enhancing vehicle safety using vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) communications. Dedicated Short Range Communication (DSRC) [2], an emerging communication standard for ITS, was developed for an FCC allocated 75 Mhz spectrum at the 5.9 Ghz band. Although IEEE 802.11a is recommended as the Medium Access Control (MAC) for DSRC, it suffers from unbounded delivery latency [3] at higher loads because of the underlying random access. This implies that for a delay-sensitive ITS safety application such as Cooperative Collision Avoidance (CCA) [4], 802.11a may not be able to provide the required message delivery latency [5], [6]. Various solutions to this problem have been proposed in [7], [8]. It has been demonstrated that the stated latency issue is more severe in the presence of vehicle crowding and broadcast storms during road emergency events. In this paper we propose a self-configuring and distributed MAC protocol that can avoid those delay concerns of 802.11.

B. Related Work

The vehicular MAC protocols in the literature are in two broad categories: contention-based and schedule-based. The contention based approaches have the advantage of not being sensitive to underlying mobility and topology changes. As a result, unlike for the schedule-based protocols, vehicle movements do not impose any reconfiguration overhead due to the network topology changes. This is a significant advantage. The unbounded delay issue however, applies to all protocols in this category because of their underlying random access. A number of variations of CSMA/CA and 802.11 have been implemented in [9], [10]. Although somewhat mitigated, the fundamental issue of unbounded delay still remains within this category of protocols.

The protocols in [11], [12] and [13] propose schedule-based TDMA mechanisms, in which TDMA slots are self-selected by the nodes in a distributed manner. While providing bounded data plane latency, the contention based slot allocation process itself involves collisions, which are stochastically resolved. As a result, the slot reallocation due to topology change may often incur a large upfront delay before the application data can start or resume flowing. The protocol in [7] proposes to configure a token ring protocol so that the maximum delivery delay is always bounded by the round-trip token time. Although bounded, the delay can be very large for large rings due to high vehicle crowding. This can be addressed by having multiple rings, but then the ring management with frequently changing topology can cause huge upfront reconfiguration delays. MCS/CDMA [14] is another schedule based protocol that uses CDMA code scheduling. Each vehicle has to have parallel receiver matched filters corresponding to all the codes used. If the search for free code is done sequentially, it could take long time when the number of vehicles involved is large. This will add substantial reconfiguration latency in the event of a topology change. Also, when there are more vehicles than the number of codes, there could be a contention for free codes resulting in large delays in channel access. The protocol LCA [3] proposes a scheduled MAC, in which TDMA slots are allocated based on a vehicle's instantaneous geographical location, which is pre-allocated a TDMA slot. This mechanism offers bounded delay, and also there is no reconfiguration latency due to network topology changes. However, the system requires complete pre-mapping of geographical locations to TDMA slots, which may have practical limitations especially

Manuscript received February 1, 2007; revised June 2007. This work was partially supported by NSF grant (SCI-0438271).

The authors are with the the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: yufan@egr.msu.edu; sbiswas@egr.msu.edu).

Digital Object Identifier 10.1109/JSAC.2007.071004.

when the huge geographic coverage of a transportation system is considered. Also, dimensioning optimal cell size for varying application requirements and vehicle density is a non-trivial problem.

The following distributed TDMA protocols, which were proposed for wireless sensor networks, can also be considered in the context of vehicular networks. The protocols TDMA-W [15] and DRAND [16] both use an out-of-band handshake signaling mechanism for distributed TDMA slot allocation. Nodes send control packets during scheduled data slots, and that is how the neighbors' allocation information is disseminated. Based on its neighbors' allocation information, a node is able to select a collision-free transmission slot, which is used for subsequent data as well as control packet transmissions. Both these protocols have been designed for networks with limited node mobility, since frequent topology changes may require substantial reconfiguration latency caused by the two-way handshake signaling needed after every topology change event.

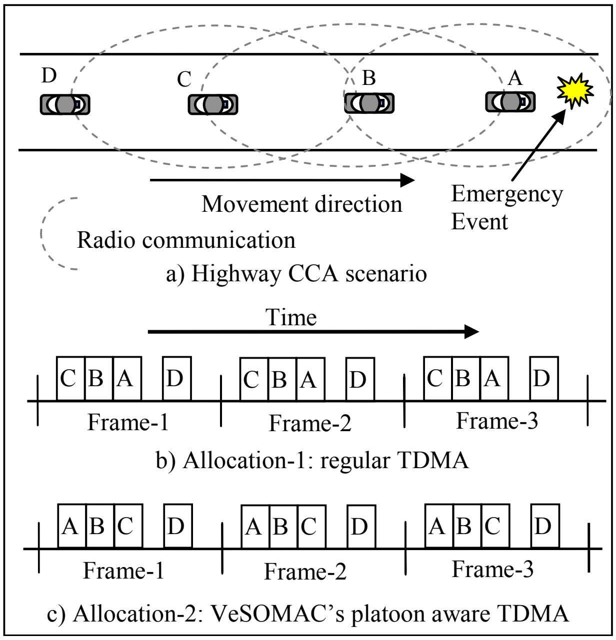
The protocol ZMAC [17] offers a hybrid solution, in which although each node is allocated a dedicated TDMA slot, the nodes are allowed to contend for bandwidth using CSMA with a goal to improve the bandwidth usage. This protocol too, due to its out-of-band allocation signaling, may suffer from high slot reallocation latency after a network topology change. The protocol D-MAC [18] allocates node and route aware TDMA slots with a goal of minimizing the end-to-end application delay. The primary difficulty of applying D-MAC to the targeted vehicular applications is that the MAC protocol assumes preset routes, which are not feasible in most safety related applications for their latency constraints. Additionally, the protocol does not provide syntax for completely avoiding hidden-collisions, which can prove detrimental for safety critical vehicular applications.

From the existing literature we conclude that schedule based protocols are desirable for their bounded delay, which is a critical requirement for ITS safety applications which require fast reconfiguration and low message delivery latency. However, the primary researchable question that still remains: how to cope with frequent topology changes by fast TDMA reconfiguration. This paper attempts to address this key question.

C. Proposed VeSOMAC Protocol

A distinctive feature of Vehicular Self-Organizing MAC (*VeSOMAC*) is its distributed design with fast schedule reconfiguration for coping with vehicular topology changes. Distributed design allows MAC allocation without having to depend on roadside infrastructure or virtual schedulers such as leader vehicles. This allocation autonomy, coupled with a novel bitmap based in-band signaling mechanism, allows *VeSOMAC* to perform fast slot reconfiguration after topology changes in dynamic scenarios such as highway platoon mergers, vehicle passing, and other urban traffic situations. Fast slot reconfiguration can be translated into low upfront latency (during a topology change) which was noted to be a serious issue for other deterministic protocols in [7] and [14].

VeSOMAC is designed to be vehicle location and movement aware for application specific delay provisioning. Consider the Cooperative Collision Avoidance (CCA) application as shown in Fig. 1(a). After an emergency event (e.g. an accident) in front of the platoon *A*-*B*-*C*-*D*, the platoon head *A* periodically broadcasts warning messages instructing other vehicles to slow down for avoiding collisions [4], [19]. Such warning messages are to be forwarded by all vehicles across the entire platoon with minimum possible delivery latency. With an example TDMA allocation [11] with arbitrarily slot placement (Fig. 1(b)), it will take three TDMA frames before the message generated by vehicle *A* will be delivered to all vehicles in the platoon. However, with a possible *VeSOMAC* allocation, in which slots are allocated based on the vehicles' relative locations (see Fig. 1(c)), the delivery delay can be significantly reduced. In this example, all messages can be delivered within a single frame. This improvement can be much more pronounced for larger platoons. This way *VeSOMAC* can effectively enhance highway safety by leveraging its ability to allocate slots based on location, speed, and other vehicular contexts.



a) Highway CCA scenario

b) Allocation-1: regular TDMA

c) Allocation-2: VeSOMAC's platoon aware TDMA

Fig. 1. Location aware MAC allocation by synchronous *VeSOMAC*.

The main contributions of the proposed *VeSOMAC* are as follows. First, MAC self-organization is achieved using a novel in-band signaling technique. Since there is no contention based operation in the protocol, it is collision free at steady state. Also, unlike the existing distributed TDMA mechanisms, by avoiding out-of-band explicit signaling *VeSOMAC* provides faster reallocation, which is particularly suitable for vehicular applications with frequently changing network topologies. Finally, an option of sequential TDMA slot allocation with respect to vehicles' physical positions allows *VeSOMAC* to reduce message delivery delay for a number of highway safety applications such as Cooperative Collision Avoidance.

II. VESOMAC PROTOCOL DETAILS

A. Frame and Slot Structures

As shown in Fig. 2(b), the transmission slots (and packets) in *VeSOMAC* are of constant duration τ, which includes neces-

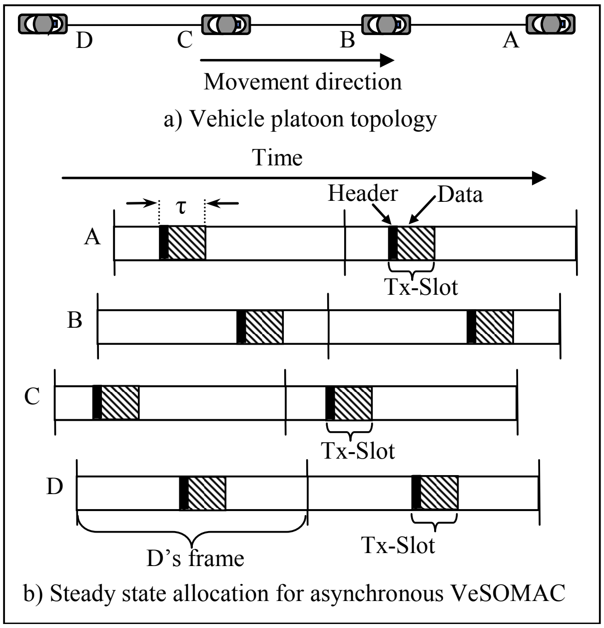


Fig. 2. Slot structure and steady state allocation in asynchronous *VeSOMAC*.

sary guard times for modem preambles and Tx-Rx switchover latencies. A frame is of duration T_{frame} sec., which defines the periodicity of transmission from any vehicle. Therefore, the allocated rate to a vehicle is $\lambda_{alloc} = 1/T_{frame}$ packets per sec. In *VeSOMAC* since a bitmap in the packet header is used for exchanging slot timing information, it is mandatory for each vehicle to send a packet every frame, even if no application data is available.

B. Synchronous and Asynchronous Operation

VeSOMAC can operate in both synchronous and asynchronous modes. In the synchronous mode (see Fig. 1(c)) all vehicles are assumed to be time synchronized, and therefore, they share the same frame and slot boundaries. In the asynchronous mode (see Fig. 2(b)), vehicles maintain their own frame boundaries and therefore the slots are also asynchronous.

C. Protocol Logic

1) *Slot Allocation*

Slot allocation in *VeSOMAC* needs to satisfy the following constraint.

Timing Constraint: *No two one-hop or two-hop neighbors' slots can overlap. Overlaps between one-hop and two-hop neighbors cause direct and hidden collisions respectively.*

A valid asynchronous *VeSOMAC* schedule for vehicles *A-D* is shown in Fig. 2(b). Except the pair (*A,D*) all other vehicles are within up to two-hop distance of each other. That is why all vehicles are allocated completely non-overlapping Tx-slots except those of vehicles *A* and *D*. Spatial reuse in *VeSOMAC* is accomplished by allowing vehicles which are more than two hops away to share TDMA slots.

2) *In-Band Header Bitmap*

Information about allocated slots is exchanged among the vehicles using a Bitmap Vector in each packet header. The concept is explained in Fig. 3. The top segment illustrates a vehicle P's allocated Tx-slot within its own TDMA frame. The middle row depicts the Tx-slots occupied by all of P's one-hop neighbors. Although these neighbors' slots are shown with respect to P's frame, each neighbor maintains its own asynchronous frame. The bottom row in Fig. 3 shows the bitmap vector that vehicle P inserts in each of its transmitted data packet headers. Middle of the bitmap represents P's own slot time. In this example, the bitmap vector is 4-bit long and each bit represents the occupancy status of two slots around P's own Tx-slot. The reason why we use one bit to represent two slots is that the neighbor's Tx-slot can partially occupy two slots in the asynchronous mode. For example, the '1' in "+1" location indicates that at least one of the two slots immediately following P's slot are already fully or partially occupied. Similarly, a '0' in the "-1" location indicates that vehicle P perceives both the slots before its own slot to be free. The bitmap vector length is a design parameter whose maximum value is the frame slot count. In Fig. 3, the frame size is 12, whereas the bitmap length is 4, which can convey the occupancy information about only 8 slots. With a bitmap size *4*, P is unable to represent the occupancy information about one of its neighbors' slots-the one in extreme left.

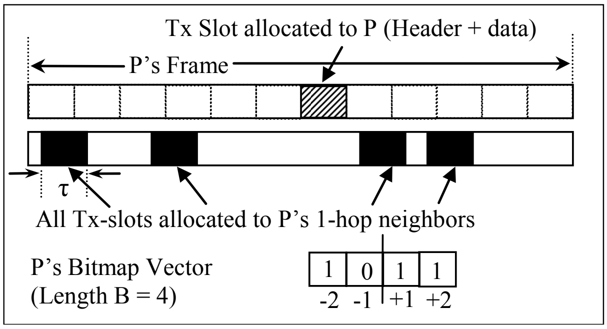


Fig. 3. In-band header bitmap for the asynchronous operation.

Using this header bitmap, a vehicle continuously informs its 1-hop neighbors about the slots occupied by its 1-hop neighbors. By listening to the bitmaps in all received packets, a vehicle can detect the slot locations of its 1-hop and 2-hop neighbors. This information can then be used for choosing a Tx-slot which is non-overlapping with the one and two hop vehicles' slots.

Since all exchanged timing information is relative, this bitmap based approach allows *VeSOMAC* to be implemented with or without network time synchronization. Although in the asynchronous version the bitmap efficiency is somewhat reduced by the overhead that one bit here represents the occupancy of two slots instead of one (see Fig. 3), in the synchronous case this overhead does not exist. In the asynchronous *VeSOMAC*, the capacity efficiency is traded for more distributed deployments without the need for time synchronization across the network.

VeSOMAC allocation needs to satisfy the following Constraint.

Bitmap Constraint: *For 1-hop neighbors i and j, i's chosen slot should be able to be represented within the bitmap vector of j. The same is applicable for vehicle j's slot. In the asynchronous case, since each bit corresponds to two slots,* *this constraint means that the slots of vehicles i and j can not be more than B slots apart, where B is the bitmap length.*

From a vehicle's perspective, the bitmap constraint is satisfied when it is able to find a '1' corresponding to its own time slot in the bitmaps from all its 1-hop neighbors. In Fig. 3, the separation between P's Tx-slot and one of its neighbors' Tx-slots (extreme left) is more than B (which is 4) slots. Therefore the bitmap constraint is not satisfied and the allocation is not stable. It is explained later in Section II-C-5, that once such an unstable situation arises due to topology changes, an iterative slot movement algorithm is needed for reaching a stable allocation convergence. Larger bitmap vectors can attain faster convergence. Note that for the maximum B value (frame size), the bitmap constraint is completely removed from the *VeSOMAC* logic. Even with the largest required bitmap vector, which is the frame duration in number of slots, the number of bits required is usually much smaller (around 5%) compared to the entire data packet, and therefore its capacity overhead can be considered negligible.

In *VeSOMAC* it is mandatory for a vehicle to send a dummy packet (with the proper bitmap information and zero information content) in each of its allocated slots even when there is no information to be transmitted. Since these dummy packets from a node do not ever compete with other nodes' slots, they do not contribute to any form of channel congestion. The advantage of this however is more responsive slot reorganization compared to the out-of-band mechanisms that rely on explicit reallocation signaling.

3) *Location Aware Slot Ordering for Delay Reduction*

As shown in the *Allocation-2* in Fig. 1(c), one way to achieve location aware delay reduction is to temporally order the slots in the same sequence as the vehicles appear in a platoon. A packet from the platoon-front A can now be delivered to the platoon-tail D within a single frame duration. This is because when it is a vehicle's turn to relay a packet, the packet is already available because it was transmitted by this vehicle's upstream neighbor during an earlier slot within the same frame. For example, since B's slot is preceded by A's slot, B is guaranteed to have received the packet from A before its own slot starts. The same applies to the slot for C. As a result, the maximum end-to-end delay in this case is bounded by the frame duration.

The absence of location-aware slot ordering introduces end-to-end routing delay in the following manner. According to the allocation in Fig. 1(b), vehicle A will forward a packet to B during A's slot in Frame-1. By the time B receives this packet, it is past B's transmission slot in Frame-1, and therefore B has to wait till the next frame for transmission. After B sends the packet to C in Frame-2, the same misordering issue is faced by C. In the end, a packet from A to D will require three complete TDMA frames, as opposed to just one frame in the case of *Allocation-1*. In the worst case, when all N vehicles in a platoon have the complete reverse ordering, the end-to-end delay will be $N-1$ frame durations. Therefore, in order to minimize end-to-end routing delay, the *VeSOMAC* allocation needs the following constraint to be satisfied.

Ordering Constraint: *If two vehicles i and j are geographical neighbors and i's location is ahead of j in the platoon, then i's chosen slot should be earlier than j's slot in the time domain.*

The *ordering constraint* is optional, and it is useful when the wireless messages flowing from the front to the tail of a platoon are more delay critical than the messages flowing in the reverse direction. While this is generally true for most ITS safety applications, a reverse requirement can be accommodated by adjusting the definition of the constraint itself. A disadvantage of the *ordering constraint* is that it can delay the self-configuration process of *VeSOMAC* by slowing down its convergence. More about convergence will be discussed later in Section II-C-5.

4) *Transmission Slot Feasibility*

A feasible transmission slot for a vehicle is one that satisfies the *timing*, *bitmap*, and the *ordering* constraints as defined in Sections II-C-1, II-C-2, and II-C-3.

Feasible Slots: Based on the timing and the bitmap constraints, an *admissible time region* for a vehicle is defined by the region that is represented by shared '0's in the bitmaps transmitted by all its neighbor vehicles. A slot chosen from this admissible region is guaranteed to satisfy the timing and the bitmap constraints. Within the admissible region, a *feasible time region* for a vehicle is defined by the duration which is sooner (on the left on time axis) than the slots of all its rear neighbors, and later (on the right on the time axis) than the slots of all its front neighbors. Any slot chosen within the feasible region will satisfy all the protocol constraints.

Consider the *VeSOMAC* example in a highway platoon scenario in Fig. 4. A new vehicle R joins in between two unconnected vehicles P and Q. Bitmaps (with length 4) from P and Q, as received by the new vehicle R, are shown in Fig. 4(a). The shared '0's in the bitmaps of P and Q indicate an admissible time region for vehicle R. The feasible region is indicated in Fig. 4(a).

Proof of Feasibility: Since a shared '0' indicates that the corresponding admissible region is not used by any of P's and Q's 1-hop neighbors, a slot chosen in that region is guaranteed to be hidden collision free from all R's 2-hop neighbors. And, since the chosen slot within the admissible region is within the bitmap of all R's 1-hop neighbors (P and Q in this case), it is guaranteed not to be used by any of those 1-hop neighbors. Also, since the admissible region is both sooner than R's rear neighbors' (Q and its neighbors) slots and later than R's front neighbors' (P and its neighbors) slots, the *ordering constraint* is also satisfied. Therefore, in Fig. 4(a), the entire admissible region is also a feasible region.

However, in Fig. 4(b), since there are no shared '0's, no admissible time region is there for R. If a slot is chosen by R from the time region indicated by a '0' in P's bitmap, then it would collide with a 1-hop neighbor of Q. Therefore, because of the violated timing constraint a hidden collision may not be avoided. Another situation is shown in Fig. 4(c), in which the bitmap constraint is violated. As a result, although collision-free slots indicated by '0' in P's and Q's bitmaps exist, no feasible slots will be available for R. Finally, in Fig. 4(d) although an admissible region is found, a feasible region is not there because of a violation of the ordering constraint. This is because although the admissible region for R is later than vehicle P's slot, a slot cannot be chosen from the admissible region so that it lies sooner than Q's slot.

5) *Protocol Overview*

If a newly joined vehicle cannot immediately identify a feasible time region, it first chooses a non-feasible slot that lies on the right side of the slot of the vehicle immediately ahead in the platoon. At this stage, the *ordering constraint* for the rear neighbor is not satisfied and the bitmap constraint may or may not be satisfied. For the asynchronous *VeSOMAC*, upon entering the network, a node arbitrarily chooses its frame starting point. For the synchronous version, it is chosen based on the frames of the existing vehicles. Upon choosing a slot, the vehicle starts transmitting data periodically once per frame. This action may first cause its 1-hop neighbors to be forced out of their stable allocation state which satisfy all three constraints. But then an iterative slot movement is used for all the vehicles in the neighborhood to incrementally attain stable allocation states including the perturbing new vehicle. During these iterations, each vehicle attempts to place its slot behind the slot of its immediate front neighbor.

Both the initial slot selection and the iterative slot movement process require a vehicle to know its neighbors' location information. This is accomplished by including a vehicle's location in the header of each of its transmitted packets. This is a reasonable assumption for most of the ITS application models [7], [20] with access to onboard GPS systems.

Consider the topology in Fig. 5, in which the allocation step-1 depicts Tx-slots chosen by vehicles A-E before vehicle C enters the network. Here we assume the maximum bitmap length (frame length), thus eliminating the *bitmap constraint*. Therefore, we only consider the *timing* and the *ordering* constraints. With the allocation in step-1, vehicles A, B, D, and E each has a collision free slot that satisfies the *timing* constraint. Also vehicle pairs A-B and D-E individually satisfy the *ordering* constraint. This is because A's slot is ahead of B's, and D's is ahead of E's. As a result, at step-1 all vehicles are at a steady allocation state.

Upon entering the network, C learns about the slot locations of its 1-hop neighbors B and D from their periodic transmissions, and the 2-hop neighbors (A and E) through the bitmaps in B's and D's packet headers. Since the temporal gap between B's and D's slots is less than a single slot, C is not able to find a slot satisfying the *ordering constraints* with respect to both B and D. As shown in step-2, unable to find a feasible slot, C chooses a collision free but non *ordering-constraint* compliant slot right after E's slot. With this, the *ordering constraint* with respect to B is satisfied but not with respect to D. Since at the end of step-2 the slot ordering of C-D is reverse to their physical ordering, now vehicles D and C cannot satisfy the *ordering* constraint, and therefore their allocations become unstable. Vehicles A, B, and E however still remain stable.

Responding to its own instability, in step-3 D moves its slot after its front neighbor C's slot for satisfying the *ordering* constraint. Similarly in step-4, E moves its slot after D's slot. After these moves, the allocation slot ordering of the platoon becomes consistent with the vehicles' physical ordering. Finally, the allocation pattern after step-4 becomes compliant with the *timing*, *bitmap* and the *ordering* constraints, and therefore it is considered to be stable.

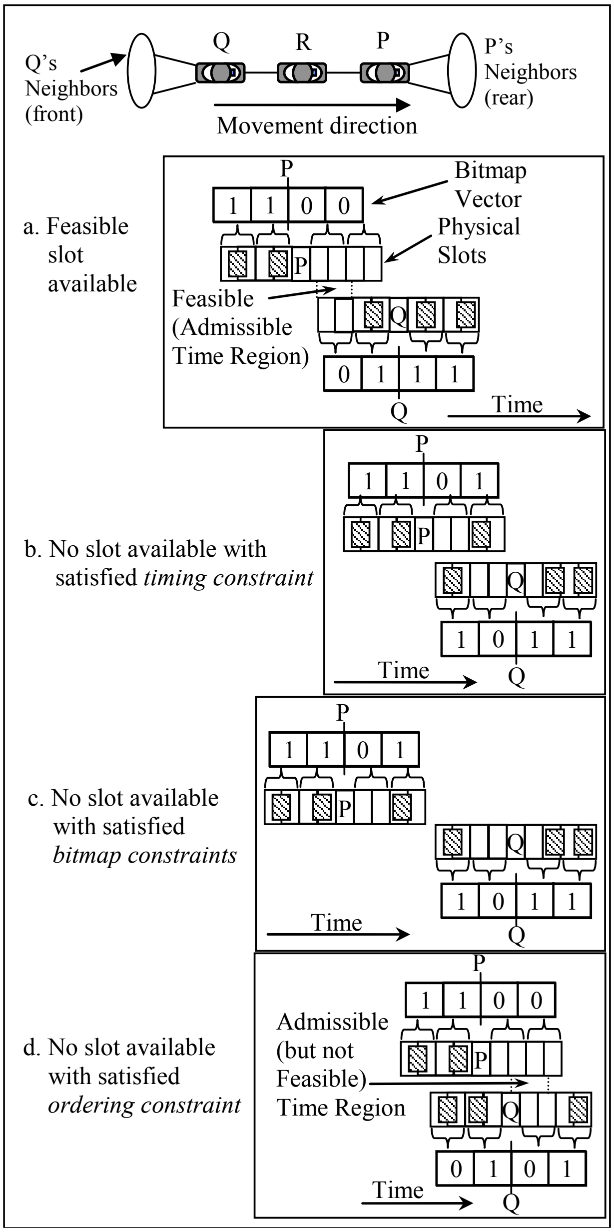


Fig. 4. Slot feasibility scenarios in example asynchronous *VeSOMAC*.

To summarize, the core idea of *VeSOMAC* is to let each vehicle iteratively move its slot following its immediate front neighbor's slot until a combined allocation pattern which is stable for all vehicles in the neighborhood emerges.

If the *ordering constraint* is removed, the iterations above become much simpler. In Fig. 5, the allocation reconfiguration will now stop at step 2 after C chooses its slot. This indicates a much faster convergence (2 steps) compared to the one with *ordering constraint* scenario (4 steps) as explained above. Generally speaking, by removing the *ordering constraint* in *VeSOMAC*, it is possible to trade data plane delay for faster protocol convergence during a topology change.

6) *Generalized Slot Selection Logic*

After joining a network, if a vehicle finds no feasible slot as defined in Section II-C-4, it picks a collision-free slot

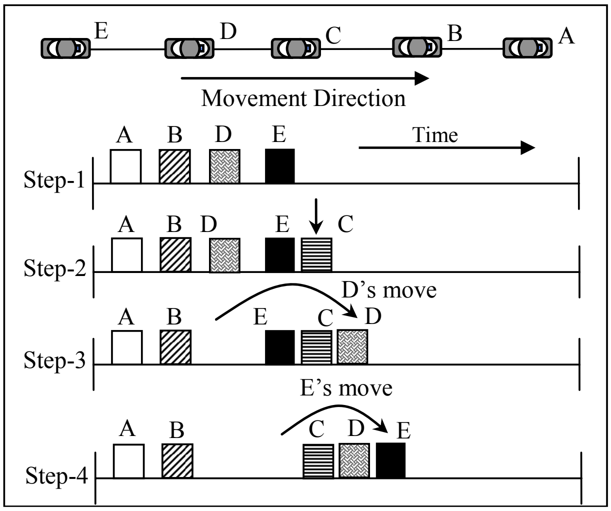


Fig. 5. Iterative slot movements for allocation convergence.

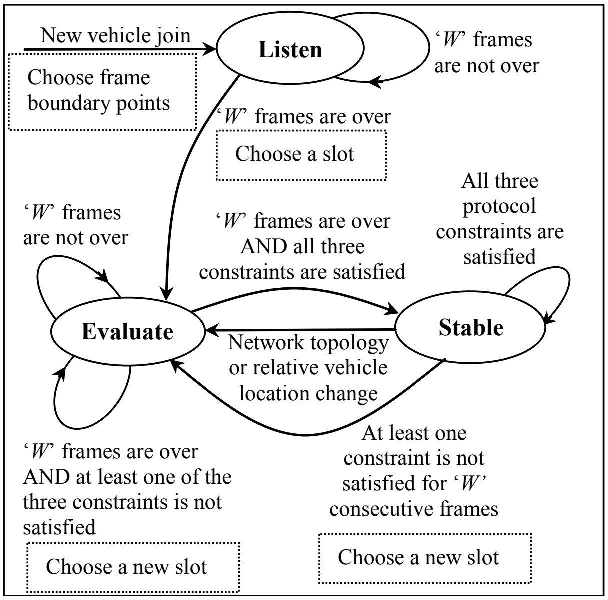


Fig. 6. State machine for the *VeSOMAC* protocol logic with all constraints.

(satisfying *bitmap* and *timing*, but not the *ordering* constraint) that is chosen randomly (uniformly distributed) within $B.\tau$ duration after the slot of its immediately front vehicle. If no such collision-free slot is found, which means a *bitmap constraint* violation, the $B.\tau$ search range is expanded using a binary exponential strategy, by doubling the search range in each successive iteration, until a collision-free slot is available. This randomness, coupled with the binary exponential range extension, reduces the allocation collision probability by introducing stochasticity when two or more unstable vehicles end up computing the same point for choosing their new slots. If an allocation collision still happens, the mechanism for its resolution is presented in Section II-C-7.

Although in Fig. 5 vehicle-by-vehicle sequential slot movements are shown for explanation purpose, no such temporal sequence for slot movement is enforced in *VeSOMAC*. Slots are moved autonomously and asynchronously, and therefore multiple vehicles' slots can end up colliding because of their slot movement to overlapping time regions.

7) *Collision Detection and Resolution*

Packet collisions in *VeSOMAC* are detected using implicit acknowledgements through the bitmaps. From the bitmaps transmitted by all its neighbors, a vehicle can infer if all those neighbors have successfully received its own transmission. If not, the vehicle concludes that its transmission got either corrupted, or lost due to a collision. If the situation persists for a preset number of frames, a collision is declared. In Fig. 4, if vehicles P and Q choose overlapping slots, a hidden collision will take place at vehicle R. Since R is not able to listen to P's and Q's transmissions due to the collision, it will simply indicate those two overlapping slots to be empty ('0') in its own bitmap. Upon receiving R's bitmap, P looks for its own slot location in that bitmap to see if P's transmission was successfully heard by R. A '0' corresponding to P's Tx-slot will indicate that there was a collision or packet corruption. If the situation persists for a preset number of frames then as shown in Fig. 5, P will move its Tx-slot iteratively for resolving the collision. Q will also behave similarly. To summarize, a vehicle is required to be able to see a '1' for its own slot in the bitmaps of all its 1-hop neighbors. Otherwise, a collision is detected. This way, the bitmaps in *VeSOMAC* are used as implicit acknowledgements for detecting collisions.

Note that the *VeSOMAC* logic can handle collisions in the presence of interference range as used in 802.11. Consider a situation with two vehicles A and C, which are more than two hops away and sharing a transmission slot. Another vehicle B is only A's 1-hop neighbor but within C's interference range. In this situation, B can not receive a message from A, since it will be corrupted due to the interference from vehicle C. According to *VeSOMAC* logic, vehicle B in this situation will not acknowledge to A's transmission in its bitmap, therefore vehicle A will eventually move its time slot thus resolving the collision with C. This means that the timing constraint in *VeSOMAC* for 2-hop communication range turns into an equivalent constraint for 2-hop interference range, when the latter is considered."

8) *Protocol State Machine*

VeSOMAC protocol state machine with all three constraints is presented in Fig. 6. The *Stable* state for a vehicle indicates the allocation steady state, and *Listen* and *Evaluate* are transient states. After a vehicle chooses a slot through the *Listen* state, it spends a preset (W) number of slots in the *Evaluate* state before getting into the *Stable* state. Any subsequent perturbations will force the vehicle to switch from the *Stable* state to the *Evaluate* state. Whenever a new slot is chosen by a vehicle, either as its first choice, or as a choice during the iterative allocation convergence, the vehicle enters in the *Evaluate* state. Subsequently, the slot is evaluated for W frames to make sure that the vehicle monitors its neighborhood activities to decide if its own allocation became stable. In our implementation, typical value of W was chosen to be 3. When the state machines for all vehicles in a neighborhood reach the *Stable* state, the protocol is said to have converged.

In *VeSOMAC*, a special type of collision can happen in extremely rare situations when all vehicles in the network happened to have entered the network exactly at the same time and they have chosen the same slot. In such a situation, the implicit acknowledgement described in Section II-C-2 fails, thus none of the vehicles are able to detect the collision. For exiting from this situation, each vehicle is programmed to randomly skip one of its slots once in a large number of frames, so that during the skipped slot it can listen if any of its neighbors is transmitting during the same Tx-slot. If such a situation is detected, the vehicle simply chooses a new feasible slot, and transits to the *Evaluate* state. Although extremely rare, this type of collision is slightly more frequent for the synchronous *VeSOMAC* because of its common frame timing across the nodes.

Since the asynchronous *VeSOMAC* relies on local clock for slot and frame timing, the relative clock drift between vehicles can move a vehicle's transmission slot with respect to those of its neighbors. This can perturb allocation stability of *VeSOMAC* by progressively violating the mapping between the relative slot locations and the information coded in the corresponding bitmap vectors. As a result, nodes can transition from *Stable* to *Evaluate* state, thus affecting the stability of the protocol. With periodic time synchronization using onboard GPS, the relative drift is periodically reset, and therefore it is less of an issue for the synchronous *VeSOMAC*.

Due to the capacity scarcity created by inappropriately dimensioned frame size (see Section II-D) when collisions are unavoidable in *VeSOMAC*, the optimality of allocation can be defined as how fairly the collisions are cycled among the participating vehicles. The objective is to prevent only a specific set of vehicles from suffering repeated slot collisions while the others enjoying the available bandwidth. According to the collision resolution logic in Section II-C-7, the implicit acknowledgement mechanism in *VeSOMAC* forces the vehicles with colliding slots to move their slots so that the slot collisions are uniformly and randomly shifted around the vehicle in the neighborhood of limited capacity.

Although the *VeSOMAC* protocol is described only for the linear vehicular topologies in this paper, the core logic of in-band and self-configurable slot allocation is equally applicable for more general 2-dimensional topologies as presented in [21]. Ongoing work on this topic includes applications for handling multi-lane and urban road network scenarios that require 2-dimensional extensions of *VeSOMAC*.

D. Model for Frame Size Dimensioning

For a given upper bound of the vehicle density, the VeSOMAC frame size can be designed in a platoon length independent manner as described below. For packet duration of τ seconds, the channel capacity is $1/\tau$ packets per second per vehicle (ppsv). If M is the maximum number of combined 1-hop and 2-hop neighbors, then the wireless bandwidth in a neighborhood is shared by $(M+1)$ vehicles. Note that for a vehicle in a highway platoon scenario, the number of 2-hop neighbors is typically twice the number of 1-hop neighbors. Therefore, the maximum data rate that can be allocated to each vehicle is given by: $\lambda_{\max}-1/\tau\,(M+1)\ldots.$ (1). Let the actual allocated data rate be $\lambda_{alloc}\text{pps}(\max[\lambda_{alloc}]=\lambda_{\max})$ and the corresponding frame duration be T_{frame} seconds. With one slot per vehicle per frame allocation, $T_{frame}=1/\lambda_{alloc}$ and since $T_{frame}=F\times\tau$, one can write: $F=1/\left(\tau\times\lambda_{alloc}\right)$ $\ldots.$ (2). Considering $\lambda_{alloc}\leq\lambda_{\max}$, from Eqns. 1 and 2: $F\geq M+1\ \ldots.$ (3). This equation represents the bound imposed by the timing *constraint* of the *VeSOMAC* logic.

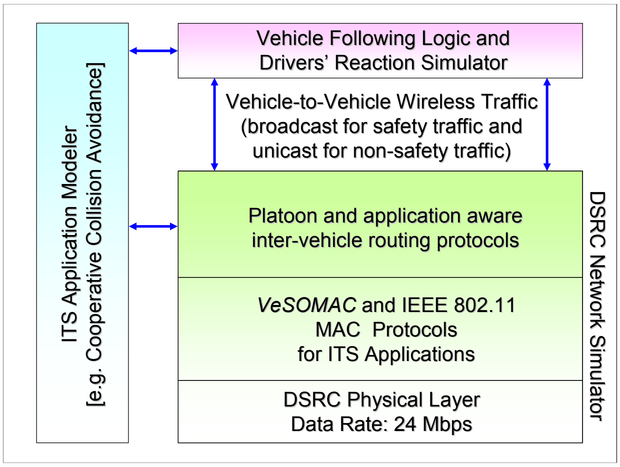


Fig. 7. InVeNTSim network with *VeSOMAC* implementation.

For the asynchronous *VeSOMAC*, according to the *bitmap constraint*, the bitmap from a vehicle is required to represent the slots of all its N 1-hop neighbors, and since each neighbor's slot can occupy at most two bits in the bitmap, $B\geq 2N$. Also, since each bit may correspond to at most two slot locations, one can write $F\geq 2B$. Combing these two, it can be written: $F\geq 4N\ \ldots.$ (4). From Eqns. 3 and 4, the lower bound of frame size for asynchronous *VeSOMAC is*: $F\geq\max\left(M+1,4N\right)\ \ldots.$ (5). For the synchronous *VeSOMAC*, while the timing constraint poses the same lower-bound for F described by Eqn. 3, the bitmap constraint requires the conditions $B\geq N$ and $F\geq B$ to be satisfied. Therefore the lower bound is: $F\geq\max\left(M+1,4N\right)\ldots.$ (6).

To dimension the frame size F, it should be chosen between the lower bound, computed through equations 5 or 6, which is decided by the maximum number of neighbors a vehicle has, and an upper bound decided by the tolerable MAC delay which is $F\tau/2$ in the average case and $F\tau$ in the worst case.

III. PERFORMANCE EVALUATION

We have developed a hybrid simulator for joint evaluation of wireless protocols and ITS applications. This Inter Vehicle NeTwork Simulator (InVeNTSim) (Fig. 7) was built within ns-2 [22] by adding a vehicle traffic module that can interact with ITS applications and driver behavior logic. InVeNTSim currently implements several ITS applications including Cooperative Collision Avoidance (CCA) and Cooperative Cruise Control (CCC).

The experiments in this paper were carried out for the CCA application on a one-lane highway platoon scenario [19] with DSRC radio communications. V2V communication was leveraged for reducing chain vehicle crashes caused by emergency events in front of moving highway platoon. First, an emergency event is simulated in front of a moving platoon. Upon detecting the event, the platoon-front vehicle rapidly decelerates (8 m/s^2) and starts broadcasting periodic Wireless

TABLE I
BASELINE EXPERIMENTAL PARAMETERS

Vehicle Related	
Platoon Size	50 vehicles
Vehicle Speed	68 mph (30 m/sec)
Inter-vehicle Spacing	25m to 45m ≡ [0.8 sec to 1.5 sec]
Vehicle Length	4 m
Emergency Deceleration	$8\ m/s^2$
Regular Deceleration	$4\ m/s^2$
Drivers' Reaction Time	0.75 sec to 1.5 sec
Network Related	
Channel	DSRC 5.9 GHz band, 24Mbps
Radio Model	Two ray ground
Radio Range	300m
MAC Protocols	IEEE 802.11 and Worst case *VeSOMAC-Synchronous*
WCW Packet Size	300 bytes (0.1 ms)
WCW Message Period	100 ms
VeSOMAC Frame Size	100 packets (10 ms)
VeSOMAC Bitmap Size	96 Bits (very weak *bitmap constraint*)
VeSOMAC Evaluation Time	*W = 3* frames

Collision Warning (WCW) packets with the format (*event-id*, *sequence-number*). While the *event-id* remains constant for an event, the *sequence-number* is incremented within subsequent WCW packets, transmitted once in every 100ms [4]. Upon receiving a WCW packet for an event for the first time, each vehicle in the platoon starts decelerating (4 m/s^2) [2] in order to avoid any impending collision due to the emergency event. Also, it rebroadcasts the packet only when it receives it for the first time. Fast and reliable WCW message delivery across the platoon is expected to reduce the number of vehicles involved in a chain collision [4], [19]. Multiple simultaneous emergency events (although rare) can be handled by adding a unique event-id (appending the generating vehicle-id and the generation time) to each WCW packet.

The CCA application was simulated in the presence of background traffic generated by non-safety ITS applications. Because of their non-deterministic message recipients, all CCA traffic is forwarded using MAC layer broadcasts coupled with multi-hop broadcast forwarding [4]. The non-safety traffic is unicast forwarded both at MAC and routing layers. Unless specified otherwise, we have always used the worst case *VeSOMAC* allocation in which the TDMA slot ordering is completely reverse to the vehicle ordering in a platoon. The purpose is to evaluate *VeSOMAC*'s worst case application impacts compared to the DSRC-recommended standards 802.11. The baseline vehicle, network, and *VeSOMAC* parameters are summarized in Table I. Each presented data point corresponds to the average from 500 independent simulation runs.

A. Vehicle Crash Performance

Effects of Vehicle Spacing: The number of vehicles crashed as a percentage of a 50-vehicle platoon is plotted in Fig. 8. With the CCA system turned off, if the vehicles decelerate based only on the tail brake light of the front cars, then for this entire range of vehicle spacing, all cars in the platoon were found to crash in a chain collision. However, as shown in Fig. 8, by turning the CCA system on, with *VeSOMAC* as the MAC layer protocol, it was possible to bring the platoon collision down to 8%, when the vehicle spacing is nearly one second (i.e. 30 m). With 802.11, however, the vehicle crash probabilities are observed to be significantly higher, especially for closely following vehicles. As expected, fewer vehicles crash with increasing vehicle spacing. This is because with larger inter-vehicle space a vehicle gets a longer time cushion for safely decelerating to a stop before crashing into the vehicle in front.

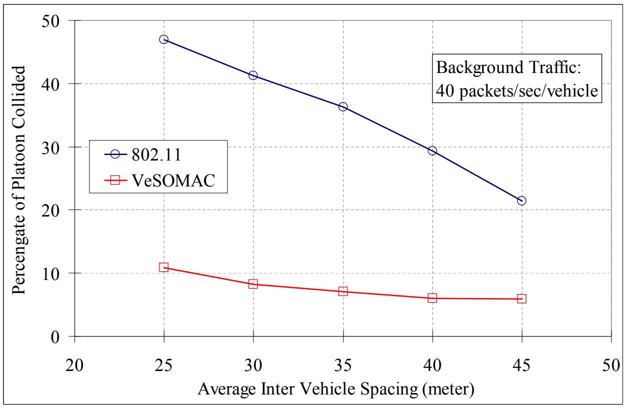


Fig. 8. Efficiency of *VeSOMAC* for reducing vehicle crashes for CCA.

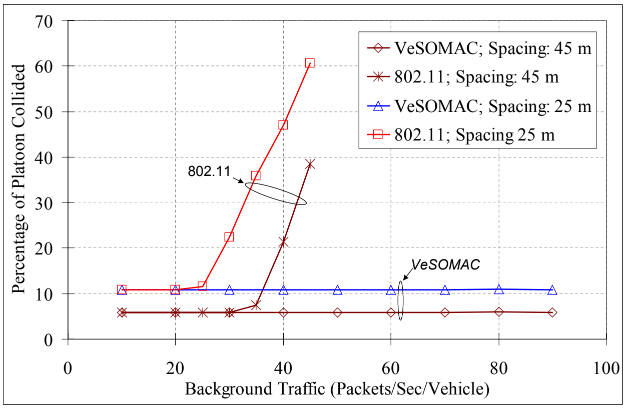


Fig. 9. The effects of non-safety traffic on avoiding vehicle crashes.

Effects of Background Traffic: As shown in Fig. 9, the performance advantage of *VeSOMAC* over 802.11 is maintained for a large range of background traffic load from the non-safety applications. Unlike 802.11, for which the MAC layer delivery delay goes up with the background traffic due to increased access contentions, the TDMA based *VeSOMAC*'s delay is virtually insensitive to the amount of background traffic. This is because of zero packet collisions, which explain why the vehicle crash count remains flat for *VeSOMAC* with increasing background traffic. For 802.11, on the other hand, the crash count increases linearly beyond background traffics of 25 ppsv and 35 ppsv for vehicle spacing of 25m and 45m respectively.

The cross-platoon message delivery latency for an example run of CCA with *VeSOMAC* is presented in the top graph of Fig. 10. Latency is defined by the duration between when the emergency event occurs at the platoon front and when a corresponding WCW message is delivered to a vehicle. Relative stop distances between consecutive vehicles are reported in the middle graph. Since the vehicle length is assumed to be 4m, any relative stop distance of 4m or less corresponds to a

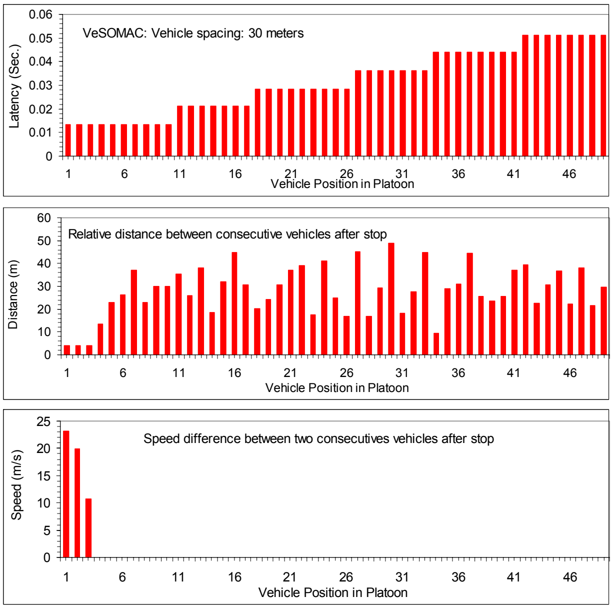


Fig. 10. Latency and crash statistics for CCA with *VeSOMAC*.

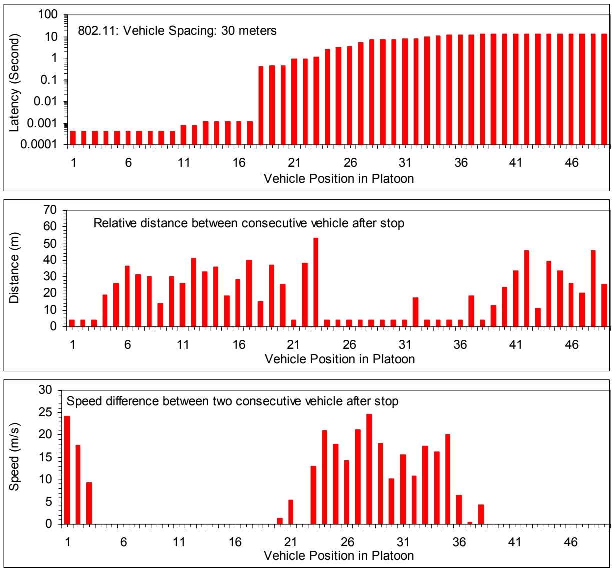


Fig. 11. Latency and crash statistics for CCA with 802.11.

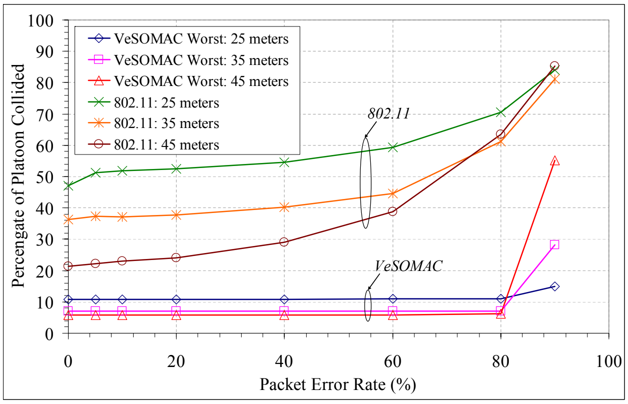


Fig. 12. Effects of channel error on cooperative collision avoidance.

crash. For vehicles avoiding a crash, the relative distance thus indicates the margin of safety provided by CCA. The bottom graph further reports the severity of vehicle crashes in terms of the relative speed between two crashing vehicles. Any relative speed greater than zero indicates a crash, and its magnitude indicates the crash severity. Results for an example CCA run with 802.11 MAC are reported in Fig. 11.

For *VeSOMAC*, since there are no packet collisions and the cross-platoon latencies are very small (up to only 51 ms), the only crashes are in the platoon front. These crashes are due to the lack of enough distance cushions and cannot be avoided with CCA even with zero message delivery latency. For 802.11, due to packet collisions and delay unpredictability, the WCW message latency increases significantly towards the rear of the platoon, where the broadcast traffic load increases progressively. This abrupt increase in latency from few milliseconds to few seconds causes a cluster of vehicles to crash due to insufficient reaction time for their drivers to brake. This explains the chain crashes at the middle of the platoon. For the vehicles towards the rear, although the absolute latencies are very high, the relative latency is small. Therefore, all vehicles get sufficient time to react, thus avoiding crashes. This explains why there are no crashes towards the platoon end. Crash performance from these example CCA runs are consistent with the average crash results from 500 different experiments presented in Fig. 8 and 9.

<u>Effects of Channel Errors</u>: Crash performance with independent bit errors (no fading and burst errors were modeled) are reported in Fig. 12. With increasing packet errors up to 80%, *VeSOMAC*'s *crash* performance is insensitive to the errors, whereas for 802.11, the crashes steadily increase. For 802.11, with increasing channel error, latency for successful reception of a WCW message progressively increases towards the platoon end. This delay is compounded by 802.11's own contention related packet drops. As shown in Fig. 11, higher latency translates into more vehicle crashes. With *VeSOMAC*, the latency was found to be low and steady for up to the 80% error mark. This low latency is due to the fact that in the absence of MAC collisions, a vehicle receives a specific WCW message several times - once from each of its neighbors. Therefore, in spite of certain number of packet losses due to channel errors, the vehicle is still able to receive at least one copy of the message from an emergency event. *VeSOMAC*'s low latency and the subsequent insensitivity to channel errors are because this reception redundancy. For error rates beyond 80%, however, the delay and the vehicle crash count shoot up. Packet drops beyond 80% is not practical, and they are shown only for understanding purposes.

Within the practical range of channel errors, the crash performance expectedly worsens with smaller inter-vehicle spacing for both the protocols. With very large packet rates though, there is an interesting trend reversal in which closely spaced vehicles actually suffer from less crashes. The reason for this turns out to be higher reception redundancy at lower inter-vehicle spacing.

Based on these results we conclude that the proposed *VeSOMAC* protocol offers significantly better CCA vehicle crash performance compared to the DSRC proposed 802.11 protocols. In fact, within an acceptable packet errors of less than 10%, *VeSOMAC* shows no sensitivity to errors.

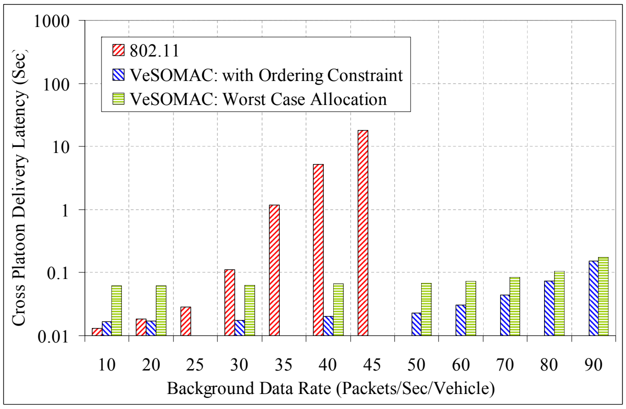


Fig. 13. WCW message delivery latency across the entire platoon.

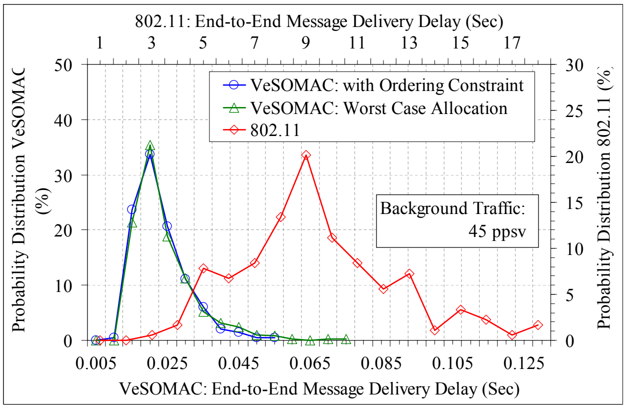


Fig. 14. Distribution of delivery latency for 802.11 and *VeSOMAC*.

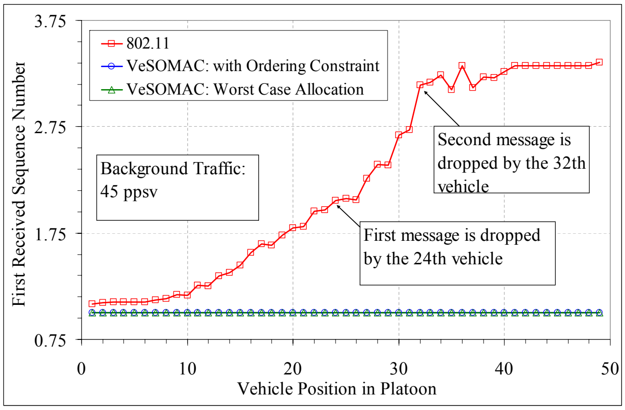


Fig. 15. Message drop at different platoon locations.

B. *Data Plane Network Performance*

1) *Cross-Platoon Message Delivery*

Delivery Delay: Experiments were carried out with both *VeSOMAC* and 802.11 protocols with varying levels of background unicast load generated using a Poisson traffic model. For a 50-node platoon, the multi-hop WCW message delivery delay to vehicle number 49 is depicted in Fig. 13. Note that the platoon-front vehicle that encounters the emergency event is number 0 and the vehicle at the platoon end is number 49. In addition to the worst case allocation (see Section III), a version of *VeSOMAC* with the ordering constraint turned on has also been experimented with. With increasing background traffic, the cross-platoon latency goes up for all the protocols due to increased MAC and queuing delays. However, compared to *VeSOMAC*, the delay for 802.11 is much more sensitive to the background data rate. With 802.11, for traffic higher than approximately 33 ppsv, the latency goes beyond seconds. With both versions of *VeSOMAC* however, the delays are constrained within only tens of milliseconds even for traffic rates as high as 90 ppsv. This large comparative latency of 802.11 further explains its poor CCA crash performance as shown in Figs. 9 and 10.

At lower rates (i.e. between 10 to 20 ppsv), there is no queuing latency for both the protocols, and the MAC delay for *VeSOMAC* is actually larger than that of 802.11. This is because unlike in 802.11, for TDMA protocols a half-frame average MAC latency is there irrespective of the data rates. As the data rate increases, the 802.11 delay increases at a much faster rate due to MAC layer collisions, retransmissions and carrier-sensing delays. For *VeSOMAC* however, there are no such delays at the MAC layer at steady state. As a result, the queuing effects are also less severe. These account for the lower rate sensitivity for the *VeSOMAC* delay.

Delay Distribution: Distributions from 500 experiments involving message deliveries to the platoon-end vehicle are plotted in Fig. 14. For 802.11, the distribution has a much wider spread (tens of seconds) compared to both the versions of *VeSOMAC* (only tens of milliseconds). Unlike 802.11, since *VeSOMAC* does not have stochastic delay due to carrier sensing, collisions, and retransmissions, its latency is deterministic. This explains the low delay jitter for *VeSOMAC*, which not only contributes to its better CCA crash performance, but it can be also beneficial for inter-vehicle audio and video streaming applications [2].

Packet Drops: Drop statistics for the WCW safety messages across the platoon is presented in Fig. 15. Due to collisions, 802.11 is susceptible to frequent packet drops. For instance, with a background data rate of 45 ppsv, on an average an 802.11 based CCA application would loose the first WCW message by the time it reaches the 24th vehicle in the platoon. Meaning, if the platoon-front was not periodically sending the WCW messages, the vehicles beyond the 24th vehicle would not have received the message, thus suffering from the possibility of chain crashes. Similarly, the 2nd WCW message gets lost by the time it reaches the 32nd vehicle. However, because of zero collisions, *VeSOMAC* can deliver the very first WCW message to all vehicles in the platoon. The results from Fig. 15 reinforce the CCA crash performance findings in Fig. 8 and 9.

2) *Benefits of Location-Aware Slot Ordering*

Cross-platoon latency performance for the worst, the average and the best case *VeSOMAC* are presented in Fig. 16. The best and the average cases represent *VeSOMAC* with *ordering constraint* turned on and off respectively. And the worst case represents a forced allocation in which the TDMA slot ordering is completely reverse to the vehicle ordering in a platoon. For a low background data rate of 10 packets/sec/vehicle (ppsv), *VeSOMAC* with *ordering constraint* is able to deliver a WCW message to the platoon-end vehicle approximately 21.5ms sooner than the average case (without *ordering constraint*), and approximately 45ms sooner than

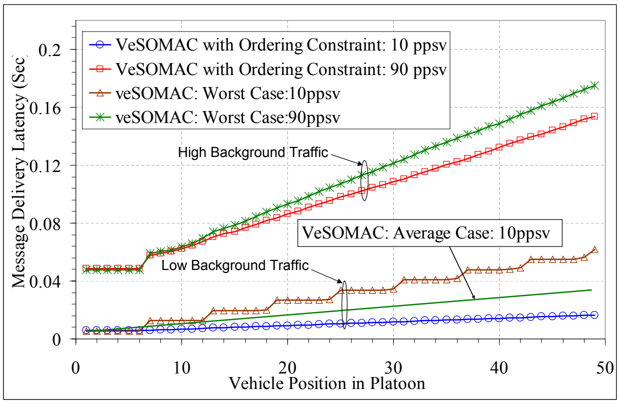


Fig. 16. Effects of the ordering constraint on delay for *VeSOMAC*.

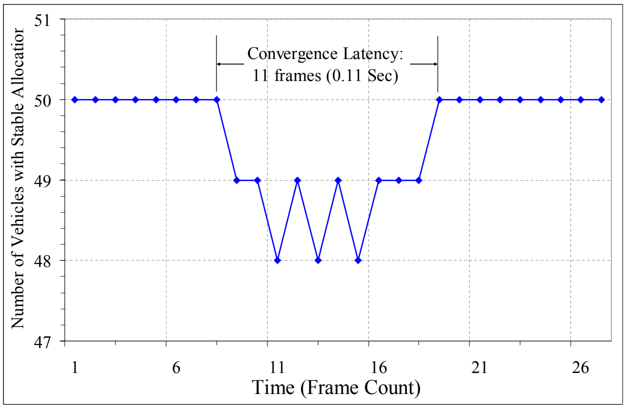


Fig. 17. *VeSOMAC* convergence dynamics after a topology change.

the worst case allocation. This indicates that the *ordering constraint* in the *VeSOMAC* logic (Section II-C-8) can indeed achieve reduced multi-hop packet delivery latency through a location aware slot allocation.

However, for higher background data rates (e.g. 90 ppsv) the delay differential between with and without *ordering constraint* scenarios are less prominent. This is because packet queuing at higher rates introduces delay, which makes the cut-though forwarding with ordered TDMA as explained in Section II-C-3 not possible. With the present experimental setup, the benefits of *ordering constraint* were found to be preserved for rates up to approximately 46 ppsv.

C. Control Plane Convergence

After a network topology change, the convergence latency for *VeSOMAC* is defined as the time interval from when at least one vehicle becomes unstable to when all the vehicles become *Stable*, as defined in the state machine in Fig. 6. The scenario shown in Fig. 17 corresponds to an instability triggered by a vehicle passing 40 vehicles in a platoon. From the $VeSOMAC$ standpoint, the platoon is stable till the 8th frame, when the switching takes place. The vehicle passing causes temporary instability by forcing at least two vehicles (not necessarily always the passing and the last-passed vehicles) out of their *VeSOMAC stable* states. But eventually, the network converges after 11 frames, which is 0.11 second for 10ms long frames.

1) *Platoon Merger*

Table II depicts convergence performance when two platoons, with individually stable *VeSOMAC* allocation, merge together. A bitmap size of 96 bits, which is closed to the frame size (100 slots), has been used for eliminating the bitmap constraint (see Section II-C-2). Convergence during a platoon merger is needed because certain vehicles within the merging platoons may have had slots that can violate the timing and/or the ordering constraints after the merger. Therefore, post-merger slot movements are needed to resolve such situations. Observe that the convergence latency is always smaller without the ordering constraints, because convergence in this case is needed for fewer nodes which violate only the timing constraint. Also, the convergence is faster when two platoons of 10 and 40 vehicles merge, compared to the 20 and 30 vehicles case. The results in Table II demonstrate that during mergers of practical size highway platoons, the allocation convergence latency of *VeSOMAC* remains within only few hundreds of milliseconds, which are deemed acceptable for ITS related applications. It is also evident that the ordering constraint in *VeSOMAC* logic can be used for trading reduced data plane latency for slower protocol convergence.

TABLE II

VeSOMAC CONVERGENCE LATENCY FOR PLATOON MERGERS.

VeSOMAC Convergence latency (sec)	Size of merged platoons	
	10 and 40 vehicles	20 and 30 vehicles
w/ ordering constraint	0.989	0.984
w/o ordering constraint	0.698	0.802

2) *Intra-Platoon Vehicle Passing*

Fig. 18 reports *VeSOMAC*'s convergence performance when a vehicle passes few vehicles in front to get ahead in the platoon. Convergence after a passing is needed because a passing vehicle's slot can violate the *timing* and/or *ordering constraints* of the vehicles that it is passing and the ones in the neighborhood of its new location in the platoon. As expected, the convergence latency increases with longer passing events because more vehicles' slots are prone to be violated in these cases. Also, the absence of the ordering constraint does expedite the convergence process. For all the experimented scenarios within a 50-vehicle platoon, the post-passing allocations have always converged within 100 ms, when the *ordering constraint* was not used.

VeSOMAC has a speed-aware component which prevents a passing vehicle from perturbing the allocation stability of the vehicles that it passes with a higher speed. The relative speed is inferred from the vehicle-location and timing information in the packet headers. A vehicle's *VeSOMAC* state machine does not react to the transmissions from neighbors with relative velocity larger than a threshold. Thus, the effects of passing shows up only after the passing vehicle settle down at a platoon location.

We have observed the convergence behavior with different size of the bitmap vectors. As expected, since with larger bitmap size the *bitmap constraint* becomes weaker, the conver-

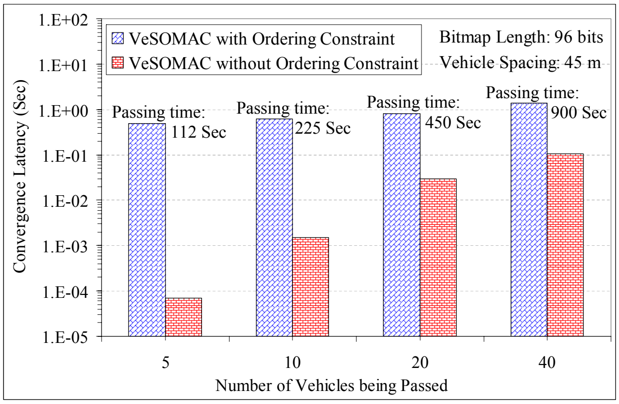


Fig. 18. *VeSOMAC* convergence latency for intra-platoon vehicle passing.

gence becomes faster. For example, in an experiment in which a vehicle passes five other vehicles, the convergence latency was registered to be 0.54 sec. and 0.49 sec. for bitmap size of 72 and 96 respectively. With *ordering constraint* turned on, the latencies have dropped but they showed similar bitmap dependency. This indicates that the largest possible bitmap size should be used for the fastest protocol convergence.

IV. Summary and Ongoing Work

We have proposed a Vehicular Self-Organizing MAC (*VeSOMAC*) protocol for distributed TDMA allocation in vehicle-to-vehicle wireless networks. A bitmap vector is used in packet headers for exchanging relative slot timing information across the 1-hop and 2-hop neighbor vehicles. It is shown that by avoiding explicit timing information exchange, *VeSOMAC* can work without network time synchronization. *VeSOMAC* is designed to be vehicle location and movement aware so that the MAC slots in a highway platoon are time ordered based on the vehicles' locations, thus minimizing the multi-hop delivery delay of ITS safety messages. Simulation results demonstrate that unlike the 802.11 style contention based protocols, *VeSOMAC* can offer better vehicle safety through smaller and bounded packet latency. It has been also shown that the protocol convergence during topology changes is fast under highway scenarios including platoon mergers and vehicle passing.

Ongoing work includes extensions of the *VeSOMAC* framework for broader ITS scenarios involving vehicle-to-roadside communication, bi-directional and multiple highway lanes, urban intersection applications, and the presence of multiple simultaneous emergency events. We also intend to evaluate it in the presence of more detailed physical layer models including fading and multi-paths. Finally, variable traffic rates will be introduced in *VeSOMAC* by allocating multiple slots to a vehicle in each *TDMA* frame.

References

[1] "Intelligent transportation systems," United States Department of Transportation [Online]. Available: www.its.dot.gov/index.htm

[2] L. Armstrong, "Dedicated short range communications (DSRC)," [Online]. Available: www.leearmstrong.com/DSRC/DSRCHomeset.htm

[3] Y. Wang and B. Bensaou, "Achieving fairness in IEEE 802.11 DFWMAC with variable packet size," in *Proc. IEEE Global Telecommun. Conf.*, 2001.

[4] H. Krishnan and C. Kellum, "Use of communication in vehicle safety application," Internal Report of General Motors Company, 2002.

[5] Y. Liu, F. Dion, and S. Biswas, "Dedicated short-range wireless communications for intelligent transportation system applications: State of the art," ITS Veh.-Highway Aut., no. 1910, 2005.

[6] S. Katragadda, G. Murthy, R. Rao, S. Kumar, and R. Sachin, "A decentralized location-based channel access protocol for inter-vehicle communication," in *Proc. Veh. Technol. Conf.-Spring*, 2003.

[7] M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, "WTRP-wireless token ring protocol," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, 2004.

[8] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *IEEE Commun. Mag.*, Jan. 2006.

[9] N. H. Vaidya, "Mobile ad-hoc networks: Routing, MAC and transport issues," [Online]. Available: http://www.crhc.uiuc.edu/~nhv

[10] R. M. Yadumurthy, A. Chimalakonda, M. Sandashivaiah, and R. Makanaboyina, "Reliable MAC broadcast in directional and omni-directional transmissions for vehicular ad hoc networks," in *Proc. 2nd ACM Int. Workshop Veh. Ad Hoc Networks*, 2005.

[11] T. Liu, J. A. Sylvester, and A. Polydoros, "Performance evaluation of R-ALOHA in distributed packet radio networks with hard real-time communication," in *Proc. IEEE Veh. Technol. Conf.*, 1995.

[12] R. Verdone, "Multihop R-ALOHA for intervehicle communications at millimeter waves," *IEEE Trans. Veh. Technol.*, no. 4, 1997.

[13] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free MAC for sensor networks," in *Proc. Int. Conf. Embedded Sensor Syst.*, Nov. 2003.

[14] T. Nagaosa and T. Hasegawa, "Code assignment in an inter-vehicle CDMA communications network," *IEICE Trans.*, vol. E81-A, no. 11, 1998.

[15] Z. Chen and A. Khokhar, "Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks," *IEEE SECON*, Oct. 2004, pp. 335-341.

[16] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks," in *Proc. 7th ACM Int. Symposium Mobile Ad Hoc Networking Computing*, May 2006.

[17] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: A hybrid MAC for wireless sensor networks," in *Proc. 3rd Int. Conf. Embedded Networked Sensor Systems*, 2005, pp. 90-101.

[18] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proc. 18th Int. Parallel Distributed Processing Symposium*, Apr. 2004, pp. 224, 26-30.

[19] X. Yang, J. Liu, F. Zhao, and N. Vaidya, "A vehicle-to-vehicle communication protocol for cooperative collision warning," in *Proc. 1st Int. Conf. Mobile Ubiquitous Syst.: Networking Services*, 2004.

[20] P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Trans. Automat. Contr.*, vol. 38, 1993.

[21] S. Biswas and F. Yu, "An in-band self-organized MAC protocol for sensor networks," in *Proc. SPIE Wireless Sensing Processing*, May 2006, vol. 6248.

[22] The network simulator [Online]. Available: NS-2; www.isi.edu/nsnam/ns

Fan Yu received his B.S. degree in Automation Control and M.S. degree in Pattern Recognition and Intelligent Control from Huazhong University of Science and Technology, Wuhan, Hubei, China in 2001 and 2004 respectively. He is currently a Ph.D. student in electrical and computer engineering at Michigan State University. His research interests include wireless sensor networks, mobile ad-hoc networks, and vehicular networks.

Subir Biswas is an Associate Professor and the Director of the Networked Embedded and Wireless Systems Laboratory at Michigan State University. Subir received his Ph.D. from the University of Cambridge and has held various research positions at the NEC Research Institute, Princeton, AT&T Laboratories, Cambridge, and Tellium Optical Systems, NJ. He has published about 70 peer-reviewed articles in the area of network protocols, and is co-inventor of 4 U.S. patents. His current research interests include the broad area of wireless data networking, low-power network protocols, and application-specific sensor networks. He is a senior member of IEEE and a fellow of the Cambridge Philosophical Society.