SS-MAC: A Novel Time Slot-Sharing MAC for Safety Messages Broadcasting in VANETs

Feng Lyu, *Student Member, IEEE,* Hongzi Zhu, *Member, IEEE,* Haibo Zhou, *Member, IEEE,* Wenchao Xu,
Ning Zhang, *Member, IEEE,* Minglu Li, *Senior Member, IEEE,* and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Efficient and scalable media access control (MAC) protocol design is crucial to guarantee the reliable broadcast of safety messages in vehicular ad hoc networks (VANETs). To devise a MAC for safety message broadcasting with reliability and minimum delay, in this paper, we propose a novel time slot-sharing MAC, referred to as SS-MAC, which can support diverse periodical broadcasting rates. In specific, we first introduce a circular recording queue to online perceive time slot occupying status. We then design a distributed time slot sharing (DTSS) algorithm and random index first fit (RIFF) algorithm, to efficiently share the time slot and make the on-line vehicle-slot matching, respectively. We prove theoretically the efficacy of DTSS algorithm, and evaluate the efficiency of RIFF algorithm by using Matlab simulations. Finally, we conduct extensive simulations considering various driving scenarios and resource conditions to demonstrate the SS-MAC performance.**

***Index Terms*—Vehicular ad hoc networks; medium access control; time slot sharing; resource management.**

I. INTRODUCTION

DRIVING safety has been the number one priority in Intelligent Transportation Systems (ITS). Transmitting warnings about dangerous driving conditions among vehicles through vehicular ad hoc networks (VANETs) has emerged as a promising solution to enhance driving safety [1]. To enhance traffic safety and efficiency in VANETs, vehicles and road-side units (RSUs) need to periodically broadcast Basic Safety Messages (BSMs) [2] to all neighbors within one-hop. By being constantly aware of the events in their surrounding environment, high-priority safety applications such as pre-crash sensing, blind spot warning, emergency electronic brake lights and so on [3], can be supported. However, the communication channel may witness excessive network load generated by high-frequency periodical broadcasts, especially under high-density situations. Without efficient control, the aggregate of these broadcasts will congest the channel, impairing reception performance and safety benefit. In addition, safety messages are delay-sensitive and require ultra-low latency, however, it is hard to find the centralized infrastructure for the coordinations of safety messages distribution in VANETs. To devise an efficient media access control (MAC) protocol for safety applications with reliability and minimum delay under this strict distributed and high dynamic networks, is critical and challenging.

In the literature, various MAC protocols have been proposed for VANETs. In particular, the EEE 802.11p [4] standard as an amendment to the existing IEEE 802.11a-2007 or Wi-Fi [5] standard, has been dedicated by the Federal Communications Commission (FFC) as the MAC layer for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications in the United States. Even though the IEEE 802.11 is widely implemented, but it dose not provide an efficient broadcast service in VANETs due to the following reasons. First, the basic MAC method of IEEE 802.11p is the same as the distributed coordination function (DCF) of IEEE 802.11, which bases on the carrier sense multiple access/collision avoidance (CSMA/CA) mechanism. In the CSMA/CA, when a node wants to access the medium, it has to sense the channel first; if the channel is idle, the node can access the medium; otherwise the node has to perform random back-off. This kind of contention-based MAC may result in possible unbounded delays [4]–[7], which cannot satisfy the real-time requirement of safety applications in VANETs. Second, in broadcast mode of 802.11p protocol, request to send (RTS) /clear to send (CTS) /acknowledgement (ACK) packets are removed to facilitate real-time response, which leaves the hidden terminal problem unsolved [8]. Beyond that, the delay and reliability performance of 802.11p DCF-based beaconing have been analyzed in many studies, and the results demonstrate that the 802.11p MAC has serious issues with unbounded delay and channel congestion under high-density scenarios [9], [10]. Specifically, the authors in the work [11] discover that even though the number of collisions is reduced by dynamically adjusting the contention window in 802.11p protocol, the packet reception probability still never reaches 90% because of the randomness of CSMA-based schemes. In the observations of the work [12], the normal delay of beacons may last 200 ms and the value can be more than 500 ms and sometimes reaches 1 s in a dense scenario. As a result, it can be concluded that the contention-based MAC 802.11p is unsuitable for real-time communications due to its nondeterministic features.

Recently, time division multiple access (TDMA) based MACs have demonstrated their efficiency in VANETs [13]–

F. Lyu, H. Zhu (Corresponding author) and M. Li are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, 200240, (e-mail: fenglv@sjtu.edu.cn, {hongzi, liml}@cs.sjtu.edu.cn).

W. Xu and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1, (e-mail: {w74xu, sshen}@uwaterloo.ca).

H. Zhou is with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China, 210093, (e-mail: haibozhou@nju.edu.cn).

N. Zhang is with the Department of Computing Sciences, Texas A&M University at Corpus Christi, 6300 Ocean Dr., Corpus Christi, TX. 78412, (e-mail: ning.zhang@tamucc.edu).

[16]. In TDMA-based MACs, time is partitioned into frames consisting of a constant number of equal-length time slots and synchronized among vehicle nodes. Each vehicle is guaranteed to access the channel at least once in each frame; the slotted channel can guarantee the stringent time requirement of safety applications. Moreover, each vehicle not only transmits its application data but also reports the status of time slots used by its one-hop neighbors; by doing so, vehicles can perceive up-to-date knowledge of two-hop neighbors and acquire a distinct time slot with them, then vehicles can have the ability to detect collisions and avoid the hidden/exposed terminal problems without RTS/CTS/ACK schemes. However, in existing TDMA MACs, a high constant beaconing rate, normally 10 Hz, is configured for all nodes, incurring serious scalability problems for resource allocation. Specifically, when the node density is low, scarce channel resources [17], [18] are wasted due to the unnecessary broadcasting. According to the vehicle safety communications report of U.S. Department of Transportation [3], there are really distinct safety applications with the broadcasting rate ranging from 1 Hz to 10 Hz. Allocating excess resources to every vehicle for unnecessary broadcasting not only wastes channel resources but also increases the possibility to interfere others. On the other hand, due to the spatial reuse constraint of TDMA [1], when the vehicle density is high, such as at intersections, the *slot starving problem* may occur with a result exposing unfairness of medium access among vehicles. For instance, when some vehicles at the intersection have fully occupied time slots in every frame, subsequent entering vehicles may have no time slot to choose for transmission, then the time slot acquisition failures will last for a long time, resulting in a huge medium access delay. To make things worse, if vehicles with low-priority [2] safety requirements have successfully occupied time slots while vehicles with high-priority safety requirements have no chance for transmission, this extreme unfairness of medium access will badly impair the driving safety. Just like traffic management in real life, when there is a traffic jam, the vehicles with more importance, e.g, police cars or ambulances, have a higher priority of passing through; the medium access control should also support this kind of fairness when the channel is saturated. Moreover, to design dynamic beaconing approaches for safety messages, beacon rate control is the main beaconing category [19]–[21] in VANETs; all these application layer approaches need a scalable and flexible MAC protocol [22], [23] to support.

For the aforementioned considerations, we propose *SS-MAC*, a novel time slot-sharing MAC for safety messages broadcasting in VANETs, with reliability and minimum delay. In SS-MAC, broadcast requirements of safety applications are periodic with different rates, it is profitable to make multiple vehicles alternately broadcast on a same time slot through inerratic coordination. To capture the periodic characteristics of safety applications over time slots, we first introduce a *circular recording queue* to online perceive time slots occupying status. The circular recording queue records the time slot status (occupied or vacant) during the latest K successive frames; a suitable K recording queue can help perceive the seasonal occupied behaviors on each time slot. Based on these information, we design a distributed time slot sharing *(DTSS)* approach, to decide whether the time slot can support the sharing for a certain periodical broadcasting requirement, and how to share a time slot in an efficient way. Specifically, we present the precondition of a time slot sharing among vehicles; to satisfy the precondition, we propose normalizing cycles of vehicles for consolidated sharing; we then define *feasibility parameter* and *sharing potential parameter* as elaborated in Section III to motivate DTSS design with *perfect sharing* property. After that, we design a random index first fit *(RIFF)* algorithm based on the heuristic packing method, to make on-line vehicle-slot matching with maximizing the resource utilization of the network. We prove theoretically the efficacy of DTSS algorithm design and evaluate the efficiency of RIFF algorithm by using Matlab simulations. In addition, we conduct extensive simulations considering various driving scenarios and resource conditions to demonstrate the overall SS-MAC design in terms of *delay ratios* of the whole network and each vehicle. In particular, the main contributions of this paper are summarized as follows.

- We design a novel time slot-sharing MAC, referred to as SS-MAC, for safety messages broadcasting in VANETs. As SS-MAC supports the diverse periodical broadcasting rates, vehicles can access the medium according to their safety need. Hence, SS-MAC can work with a strong scalability in terms of channel resource management under all sorts of driving scenarios.
- To achieve the common agreement of the time slot sharing, we devise a distributed time slot sharing approach, called DTSS, to regulate the time slot sharing process among vehicles. It can maximize the sharing potential of each certain time slot.
- We develop the RIFF algorithm to assist the vehicle selecting a suitable time slot for sharing. It can not only satisfy the broadcasting requirement of every vehicle but also optimize the resource utilization of the network.

The remainder of this paper is organized as follows. We present the system model and TDMA MAC basics in Section II. We propose SS-MAC protocol in Section III. We evaluate the efficiency of the RIFF algorithm and conduct extensive simulations to evaluate the performance of SS-MAC in Section IV. Finally, we conclude this paper and discuss future work in Section V.

II. SYSTEM MODEL AND TDMA MAC BASICS

A. System Model

We consider the VANET scenario as shown in Figure 1, where nodes communicate with each other through wireless communication. In the scenario, communication nodes include vehicles and RSUs, both of which need to broadcast information periodically for driving safety. As the RSUs can access the channel via the same MAC protocol like the vehicles, for convenience, we call the node or the vehicle on behalf of the set of the RSU and the vehicle in this paper.

---

[1] For reliable transmission without collisions, vehicles in two-hop communication ranges should not use the same time slot.

[2] The priority is judged by the broadcast rate requirement in this paper.

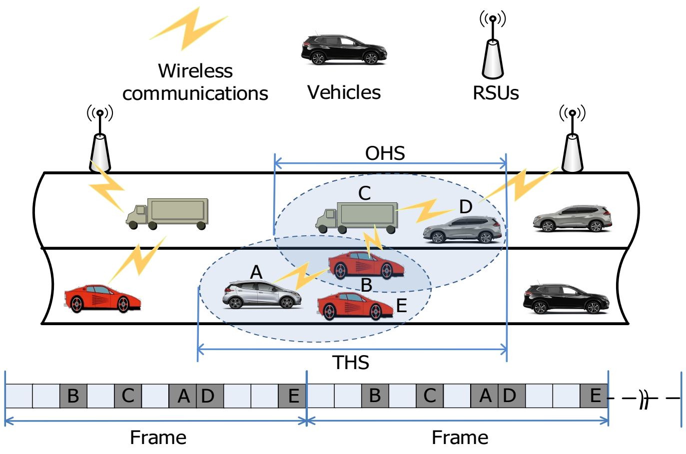


Fig. 1: Illustration of the system model.

Wireless communications. Dedicated Short Range Communications (DSRC) module is used by each node for wireless communication. The DSRC radio supports one Control Channel (CCH) and multiple Service Channels (SCHs) with two optional bandwidths of 10 MHz and 20 MHz [24], [25]. The CCH is used to transmit control information or high-priority short messages (such as periodic or event driven safety messages) while the SCHs are used to transmit non-safety user application data. In general, the radio switches the channel between the CCH and the SCH; when in CCH Intervals (CCHI), all nodes tune to the CCH to send/receive high priority safety messages or to negotiate the following usage of SCHs among nodes; when in SCH Intervals (SCHI), the nodes tune to the specific SCH for services according to the negotiation result [26]. If there are two antennas, the first one is tuned to the CCH and the second one is tuned to the SCH; safety applications and negotiation information of SCHs usage are transmitted over the CCH. In both cases, the transmission over the CCH is the foundation of safety applications and multi-channel operations. For this purpose, we focus on the design of an efficient medium access control on CCH for safety applications in this paper. To control the medium access on the CCH, the channel is set to a slotted/framed structure. Specifically, time is partitioned into frames and each frame contains N number of slots; each time slot is set to a equal-length duration for data transmission; before the data transmission, a node has to apply a vacant time slot first. As shown in Figure 1, the elements in light color denote vacant time slots while the elements in dark color denote occupied time slots by vehicles. In addition, the channel is thought to be symmetric which has been evaluated in the work [27]. Thus, if the node x is in the communication range of the node y iff the y is in the communication range of x. Each radio in the network has the identical communication capability and the same communication range R.

Vehicles. Each vehicle in the network is equipped with at least one DSRC radio and a global positioning system (GPS) receiver. The GPS not only provides the location information but also does the synchronization among vehicles. To synchronize vehicles, the pulse per second (1 PPS) signal provided by any GPS receiver can be utilized. Specifically, the rising edge of the 1 PPS signal is aligned with the start of every GPS second with an accuracy within 100 ns; this accurate 1 PPS signal can be used as a common time reference among all the nodes. As a result, at any instant, each node can determine the index of the current slot within a frame [15], [28]. In addition, the stability of the GPS receivers local oscillator at each node can still support synchronization with accuracy for a short time, coping with the temporary loss of GPS signal. For driving safety, each vehicle has to broadcast information periodically with a different frequency based on the safety application type and all the safety applications have a low tolerance of delay. According to the vehicle safety communications report of U.S. Department of Transportation, the frequency of safety applications can range from 1 Hz to 10 Hz [3]. Each vehicle is identified by a MAC address as well as a randomly generated short identifier (ID).

B. TDMA MAC Basics

Before transmission, each node needs to acquire a unique time slot and once a time slot is assigned, the node can use it in all subsequent frames. As shown in Figure 1, all the neighboring vehicles in the communication range of the vehicle A constitute the *one-hop set* (OHS) of the vehicle A. In addition, the *two-hop set* (THS) of the vehicle A refers to all the vehicles that can reach the vehicle A in two hops at most. Before introducing our design, in this subsection, we simply conclude some TDMA MAC basics which will also be adopted by SS-MAC.

Broadcasting additional frame information (FI). Due to the lack of infrastructure in VANETs, there is generally no centralized coordinator to do time slot allocation. To coordinate the time slot using in a distributed way, each vehicle needs to collect (passively hear) and broadcast addition information, called the frame information, to its OHS so that each vehicle can perceive the time slot occupying status. The frame information transmitted by a vehicle includes the vehicle IDs and the corresponding time slots in its OHS. In doing so, each vehicle can know all the occupied time slots by its THS. Adding such extra coordination data in broadcast packets is acceptable due to that, the total packet size of safety applications (normally 200-500 bytes [2]) is far less than the size of MAC layer protocol data unit.

For a vehicle x, the following sets are defined:

- $N_{cch}(x)$: the set of IDs of its one-hop neighbors, which are updated upon whether the node x has received packets directly on the channel in the previous N slots. In addition, the node x needs to broadcast this information with application data during each transmission.
- $N^2_{cch}(x)$: the set of IDs of its two-hop neighbors, indirectly obtained from the packets transmitted by its one-hop neighbors, i.e.,
$$N^2_{cch}(x) = N_{cch}(x) \bigcup \{N_{cch}(y), \forall y \in N_{cch}(x)\}.$$

Time slot acquisition. Obviously, vehicles in the same OHS should select different time slots to avoid transmission collisions. To eliminate the hidden terminal problem, vehicles in the same THS should also choose distinct time slots even

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2017.2780829, IEEE Transactions on Vehicular Technology

they may not collide with each other directly. Specifically, in Figure 1, the vehicle A and C are in the same THS with a common one-hop neighboring vehicle B; if vehicle A and C transmit messages simultaneously, collisions will happen at the vehicle B. By collecting frame information from its OHS neighbors, a vehicle can randomly choose a free time slot from the idle time slot set. Under this scheme, once a vehicle has the latest channel information of a frame time, the vehicle can make the slot acquisition decision. To maintain such information, for a running radio, it can update the channel information during each frame while a newly opened radio has to listen to the channel for an entire frame first.

Detecting collisions. Once a vehicle successfully acquires a time slot, it can use it in all subsequent frames. However, due to the lack of centralized controllers and moving characteristics of VANETs, transmission collisions encounter with a relatively high probability. There are two kinds of collisions can be identified, i.e., *access collision* and *merging collision* [15]. The access collision happens when two vehicles in the same THS try to acquire a same time slot then the collision encounters at their common neighboring vehicles; while the merging collision happens when two vehicles which are in two different THS originally, use the same time slot and move into a same THS finally due to the diverse mobility. To detect the collisions, each vehicle checks the frame information received from its OHS during the previous $N-1$ time slots. Specifically, if packets received from all $y \in N_{cch}(x)$ indicate that $x \in N_{cch}(y)$, it means that there is no other node in the two-hop ranges of x accessing the same time slot with the node x; otherwise, the node x may collide with other during the last transmission. Once a collision is detected, the node should release its original time slot and try to apply a new time slot.

III. SS-MAC PROTOCOL DESIGN

A. Design Overview

To design a scalable slot-sharing TDMA MAC for diverse periodical broadcasting rates under strict distributed and high dynamic vehicular networks, a high-precision coordinating is in need. In the following subsections, we first introduce a circular recording queue to perceive time slots occupying status. We then design a distributed time slot sharing approach, called DTSS, to decide whether the time slot can support the sharing for a certain periodical broadcasting requirement and how to share a time slot in an efficient way. After that, we design a RIFF algorithm based on the heuristic packing method, to make on-line vehicle-slot matching with maximizing the resource utilization of the network.

B. Perceiving Time Slots Occupying Status

Prior to share a time slot, vehicles need to perceive the time slot using status. As safety applications periodically broadcast messages under different rates, one possible solution is for each vehicle to adopt a *circular recording queue* to record the most recently using status (one means *occupied* while zero means *free*) of every slot in each frame. Specifically, for each time slot i, $i \in [1, N]$, the vehicle initialises a circular recording queue, denoted as $\mathcal{Q}_i = [q_{K-1}, q_{K-2}, ..., q_0]$ where q_j for $j \in [0, K-1]$ is the $(j+1)$th element in the queue counted from right to left which means the using status of slot i in the previous jth frame, and q_0 means the using status of the current frame. For each vehicle, at the end of each frame, based on the updated time slots using status $U(x)$ of the current frame, if the slot i is used by its THS neighbors in $N(x)$, elements in $\mathcal{Q}_i$ move one step towards left direction which means the value q_j is replaced by q_{j-1} and the q_0 is set to the value one; otherwise, elements in $\mathcal{Q}_i$ move one step towards left direction and the q_0 is set to the value zero. With the circular recording queue keeping the most recently K frame time slots occupying information, the sharing status of every time slot can be perceived online and can be the coordination guideline for vehicles to share time slots.

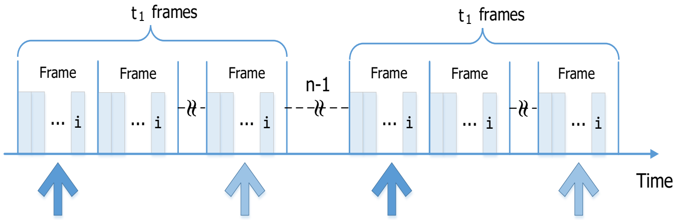


Fig. 2: Precondition of time slot sharing for two periodical safety applications.

To design an appropriate queue size K, the following tradeoff should be considered. On one hand, if the size of the queue is too small, the K frame time cannot cover a complete broadcasting cycle of safety applications with a low frequency, which means that the circular recording queue has no capability of collecting all the needed coordination information. On the other hand, if K is too large, the past K frame time used to obtain the slot occupying information can last long; however, during this long duration, the network topology and the time slot sharing status may change due to the solid high dynamic characteristics of VANETs. As a result, the outdated records may incur inaccurate decision of time slot sharing.

C. Distributed Time Slot Sharing Approach

In this subsection, we first show the precondition of a time slot sharing among vehicles. To satisfy the precondition, we then propose normalizing cycles of vehicles for consolidated sharing. We finally elaborate the DTSS design, in which the feasibility parameter and sharing potential parameter are defined to maximize the sharing potential of a time slot. Under this design, we prove that DTSS can work with the perfect sharing property.

Precondition of time slot sharing. As broadcast requirements of safety applications are periodic with different cycles, it is profitable to make multiple vehicles alternately broadcast on a same time slot for efficient utilization of channel resources. For convenience, we name the safety applications as *t-cycle* applications in this work when the safety application has a periodic broadcast requirement every t frame time (each frame usually last 100ms).

Lemma 1: For t_1-cycle and t_2-cycle applications, $1 < t_1 \leq t_2 \leq T$[3], T is the maximum cycle value, if $t_2 = n * t_1$, $\forall n = 1, 2, 3, ...$, then these two applications can share a same time slot.

Proof: As shown in Figure 2, the t_1-cycle application broadcast once at the slot i, $i \in [1, N]$, during every t_1 frames; the deep color arrow is an integer pointer p_1 pointing to the frame that the t_1-cycle application will use the slot i in this frame, $p_1 \in [0, t_1 - 1]$. For the t_2-cycle application, it also broadcast on the slot i and the light color arrow is an integer pointer p_2 pointing to the frame that the t_2-cycle application will use the slot i in this frame, $p_2 \in [0, t_1 - 1]$ and $p_2 \neq p_1$ at initial stage [4]. As $t_2 = n * t_1$, after $n*$ t_1 frames, the t_2-cycle application will broadcast a new message at the slot i and the value of p_1 and p_2 will not change. Without collisions under this periodic activities, then the lemma is proved. ■

However, if $t_2 = n * t_1 + k$, $k \in [1, t_1 - 1]$, then p_2 could be $(p_2 + k * m)\% t_1$, where $\%$ is the modulus operator. As a result, p_1 has the possibility to collide with p_2 or not, depending on the value of t_1, t_2, p_1 and p_2. We do not consider this sharing situations due to the lack of regularity.

Normalizing cycles for consolidated sharing. Lemma 1 shows a precondition of slot sharing which also limits the sharing scopes and brings complexity for coordination, especially in distributed systems. Specifically, for periodical applications with various cycles, few of them can have multiple relationships in terms of cycles; moreover, for vehicles, it is hard to detect all broadcast cycles under a wide range. One reasonable solution is to normalize the cycle of applications to a close value, which not only satisfies the time QoS of safety applications, but also would be convenient for sharing a time slot with other vehicles. To do it, we define a list of normalizing targets, denoted as $\mathcal{N} = [n_1, n_2, ..., n_Z]$, $1 < n_1 < n_2 < ... < n_Z \leq T$, where n_z for $z \in [1, Z]$ is the zth target in the list and $n_{z+1} = X * n_z$, X is an integer and $X > 1$.

Rule 1: For a t-cycle safety application, $\forall t \in [1, T]$, the cycle of the application is normalized as follows

$$t = \begin{cases} 1 & t < n_1; \\ n_z & n_z \leq t < n_{z+1}; \\ n_Z & n_Z \leq t \leq T. \end{cases} \tag{1}$$

After normalization, periodical applications can share a time slot with each other once the time slot has enough capacity. To collect needed coordination information, the size K of circular recording queue can be set to the value n_Z. Figure 3 is an example of time slot sharing, where $Z = 3$ and $n_1 = 2$, $n_2 = 4$, $n_3 = 8$. In the figure, a 2-cycle application, a 4-cycle application and a 8-cycle application are sharing a time slot i, $i \in [1, N]$. By checking the circular recording queue of slot i, vehicles can perceive the sharing status of the slot and find that it can still support sharing for another 8-cycle application.

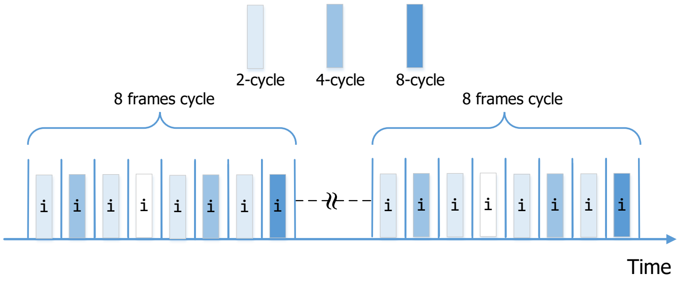


Fig. 3: An example of time slot sharing, where $Z = 3$ and $n_1 = 2$, $n_2 = 4$, $n_3 = 8$.

DTSS algorithm design. After normalizing the cycle of applications, the other issue is how to coordinate among vehicles to share a time slot not only without collisions, but also fully utilizing the time slot. As the size K of circular recording queue is set to the biggest cycle n_Z, the occupying status of K frames for a slot will repeat in the following subsequent K frames and vehicles can coordinate to share a time slot just based on the information of the circular recording queue. To regulate the usage of a time slot, we design the DTSS algorithm.

Definition 1: (Item size) For a t-cycle safety application, the *item size* α of the application is $\alpha = \frac{1}{t}$, $\alpha \in (0, 1]$.

Definition 2: (Slot capacity) For a specific time slot, the *capacity* C of the time slot is calculated by the number of idle elements in the circular recording queue to the total size K and the capacity of a free time slot is $C = 1$.

Remark 1: For a safety application with a size of α, if the application has a chance to share the time slot, iff the remain capacity of the time slot is equal or bigger than the size α, i.e.,

$$C \geq \alpha. \tag{2}$$

For a t-cycle application, after normalizing the cycle to a value n_z, we define a list of serial numbers for the recording queue under this cycle by modulus operator, i.e, for an element q_j in the queue, $j \in [0, K-1]$, the serial number $s_j = j\% n_z$, $s_j \in [0, n_z - 1]$, is defined.

Definition 3: (Feasibility parameter) We define the *feasibility parameter* f_j^z for each element in the circular recording queue under different cycle values, $j \in [0, K-1]$ and $z \in [1, Z]$. For a given z and j, its serial number satisfies $s_j = j\% n_z$, for $\forall \{x | x\% n_z = s_j, x \in [0, K-1]\}$, if all $q_x = 0$, then f_j^z is set to the value one, otherwise it is set to the value zero, i.e.,

$$f_j^z = \begin{cases} 1, \forall \{x | x\% n_z = s_j, x \in [0, K-1]\}, q_x = 0; \\ 0, elsewhere. \end{cases} \tag{3}$$

Remark 2: If a n_z-cycle application can share a time slot, iff there is a $j \in [0, n_z - 1]$ and $f_j^z = 1$, i.e.,

$$\exists j \in [0, n_z - 1], s.t., f_j^z = 1. \tag{4}$$

If the application chooses the jth element in the recording queue to share the time slot, then all the elements $\forall \{x | x\% n_z = s_j, x \in [0, K-1]\}$ will by occupied by the application to demand its requirement. Following Remark 1 and Remark 2, vehicles can share a time slot without collisions. However, choosing an appropriate value j in Remark 2 determines the sharing efficient of the time slot. For instance, Figure 4 shows

---

[3] As 1-cycle applications need to broadcast during each frame, they can not share a time slot with other applications.

[4] When two periodical broadcast applications share a time slot, they should use the time slot at different frame; otherwise collisions will happen.

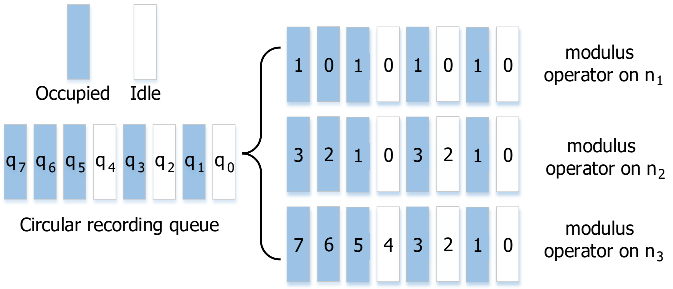


Fig. 4: An example of a circular recording queue and the corresponding modulus operators on different cycle values, where $K = 8$ and $n_1 = 2$, $n_2 = 4$, $n_3 = 8$.

an example of the circular recording queue of a specific time slot and the corresponding modulus operators on different cycle values, where the size of recording queue K is set to be 8 and $n_1 = 2$, $n_2 = 4$, $n_3 = 8$ respectively. In addition, the dark block denotes the occupied status while the white block denotes the idle status and the remain capacity of the time slot is calculated by the number of idle blocks to the total size K, i.e, $\frac{3}{8}$. According to the capacity restriction of Remark 1, only the 4-cycle or 8-cycle applications can be supported by this time slot. For a 8-cycle application, the element in the recording queue with a serial number 0, 2, 4 can satisfy its demand; if the application chooses to occupy the serial number 2, then the remain capacity of the time slot can still support a 4-cycle application, otherwise the time slot can only support 8-cycle applications if the number 0 or 4 is chosen.

Remark 3: For an jth element in the recording queue, if $f_j^z = 0$, then $f_j^{z-1} = 0$, $z \in [2, Z]$.

Remark 3 indicates that, if the jth element can not support sharing for the n_z-cycle applications, then it has no potential for supporting applications with a higher frequency. Considering this, we define the *sharing potential parameter* p_j for each element in the recording queue as follows.

Definition 4: (Sharing potential parameter) For an jth element in the recording queue, the sharing potential parameter p_j is the maximum item size of the n_z-cycle applications that the corresponding value of feasibility parameter satisfies $f_j^z = 1$, i.e.,

$$p_j = \frac{1}{n_z}, z = \min\{\forall z \in [1, Z], s.t., f_j^z = 1\}. \tag{5}$$

In addition, the sharing potential of the time slot p is the maximum p_j, for $j \in [0, K-1]$, i.e.,

$$p = \max\{p_j, j \in [0, K-1]\}. \tag{6}$$

Remark 4: If a n_z-cycle application can share a time slot, iff the sharing potential of the time slot p satisfies $p \geq \frac{1}{n_z}$.

Rule 2: For a n_z-cycle application, if exists a list J of elements in the recording queue($j \in [0, n_z - 1]$) can satisfy its sharing demand according to the limit of Remark 1 and Remark 2, DTSS algorithm chooses the element with the minimum value of sharing potential parameter from the list J.

As shown in Figure 4, for a 8-cycle application, the 0th, 2th and 4th element can satisfy its requirement with the sharing potential parameter $p_0 = \frac{1}{4}$, $p_2 = \frac{1}{8}$, $p_4 = \frac{1}{4}$ respectively. According to Rule 2, the 2th element will be chosen and the application can access the channel at this time slot after waiting for $(n_z - 1 - j)$ frames with a repeating cycle of 8 frames. DTSS algorithm for distributed efficient time slot sharing is shown in Algorithm 1.

Algorithm 1 DTSS algorithm for a time slot sharing.

```
Input: $\mathcal{Q}_i$ and cycle $t$
Output: waiting frames $w$
Initialize: $w = -1$, $p_{min} = 1$
Normalize: $t$, $t \leftarrow n_z$
if $C_i \geq \alpha_z$ then
    for $j \in [0, n_z)$ do
        if $f_j^z == 1$ then
            if $p_j \leq p_{min}$ then
                $p_{min} = p_j$
                $w = n_z - 1 - j$
return $w$
```

Definition 5: (Perfect sharing) We define the *perfect sharing* property for a time slot. It has the following feature; for a time slot with the capacity C and a application with the normalized cycle $n_z(z \in [1, Z])$, if $C \geq \frac{1}{n_z}$, then the time slot can support sharing for the application.

Lemma 2: DTSS algorithm can guarantee each time slot with the perfect sharing property [5].

Proof: For a time slot, it can support sharing for n_1 numbers of n_1-cycle applications and a group elements for a n_z-cycle application can be divided into X groups elements for X n_{z+1}-cycle applications sharing, $n_{z+1} = X * n_z$. Considering a time slot is being shared by the n_z-cycle application with the number of x_z, $x_z \geq 0$, $z \in [1, Z]$ and

$$\sum_{z=1}^{Z} x_z \frac{1}{n_z} \leq 1. \tag{7}$$

According to Rule 2 choosing the minimum sharing potential element, it can guarantee that if the group elements for a n_z-cycle application still have the space for a n_{z+1} application, the n_{z+1} application will choose elements from the remaining and will not occupy elements which have potentials for other n_z-cycle applications. Based on this rule, the elements occupied by X numbers of n_{z+1}-cycle applications finally can be combined to a group elements for a n_z-cycle application. Then the occupying status of a time slot can be represented by the n_i-cycle application with the number of y_i, $0 \leq y_i < X(y_1 < n_1)$, $i \in [1, Z]$. Under this considering, the capacity of the time slot $C = \sum_{i=1}^{Z} l_i \frac{1}{n_i}$ and the l_i satisfies

$$l_i = \begin{cases} n_1 - 1 - y_1 & i \doteq 1; \\ X - 1 - y_i & 1 < i < Z; \\ X - y_i & i = Z. \end{cases} \tag{8}$$

To support the requirement of a n_j-cycle application, if $\exists i \in [1, j], s.t., l_i > 0$, apparently the time slot can support sharing for the application; if $\forall i \in [1, j], l_i = 0$, according to the

---

[5] The Lemma 2 demonstrates the efficacy of the DTSS algorithm design in theory.

expression of the capacity C and the limit of Eq. (8), thus the $C < \frac{1}{n_j}$ which violates the condition in Definition 5. Then the lemma is proved. ∎

D. On-Line Vehicle-Slot Matching Approach

Given the recording queue information of a time slot, DTSS algorithm can determine whether the time slot can satisfy the time QoS of applications and how to share a time slot greedily to maximize future sharing potential. However, in practical, the medium is set to be numbers of time slots and how to select a satisfied time slot for nodes has an impact on the resource utilization of the network. According to Remark 4, each time slot has a sharing potential value, and only the item size of the application is smaller than the potential value then the application can use the slot. Inspired by this, the on-line vehicle-slot matching problem can be modeled as a on-line bin packing problem, where each application has a item size and each free time slot has a full potential 1. The on-line bin packing problem is a well-known NP-hard problem; some classical on-line algorithms are proposed such as WF (Worst-Fit), BF (Best-Fit), FF (First-Fit) and so on; these classical algorithms can be generalized to the *Any-Fit* and the *Almost Any-Fit* classes [29].

Any-Fit constraint: If $B_1, B_2, ...B_j$ are the current nonempty bins, the current item will be packed into B_{j+1} iff it does not fit in any of the bins $B_1, B_2, ...B_j$.

Almost Any-Fit constraint: If $B_1, B_2, ...B_j$ are the current nonempty bins and the $B_k (k \leq j)$ is the unique bin with the smallest content, the current item will be packed into B_k iff it does not fit in any of the bins to the left of B_k.

The class of on-line heuristics that satisfies the Any-Fit constraint will be denoted by $\mathcal{AF}$ and the class of on-line algorithms satisfying both constraints above will be denoted by $\mathcal{AAF}$.

Theorem 1 (Johnson [30]): For every algorithm $A \in \mathcal{AF}$,

$$R_{FF}^{\infty} = R_{BF}^{\infty} \leq R_{A}^{\infty} \leq R_{WF}^{\infty}{}^{6}. \tag{9}$$

Theorem 2 (Johnson [30]): For every algorithm $A \in \mathcal{AAF}$,

$$R_{A}^{\infty} = R_{FF}^{\infty}. \tag{10}$$

Theorem 1 and 2 demonstrate that the FF and BF algorithm can achieve the lowest worst-case ratio compared with algorithms in $\mathcal{AF}$ and $\mathcal{AAF}$ class. In FF algorithm, the current item will be packed into the first nonempty bin which it fits, if no such nonempty exists, the algorithm will open a new bin to pack the item. While the BF algorithm packs the current item into an open bin of largest content in which it fits; if no such nonempty exists, the algorithm will open a new bin. However, during time slot acquisitions, there are many nodes may acquire time slots simultaneously; the fitting results for different nodes of FF or BF have a high possibility to be the same value, which may incur collisions when more than two nodes are assigned to a same time slot. To solve this, we design a new heuristic algorithm called *RIFF* based on FF. If $\mathcal{U} = \{U_1, U_2, ..., U_j\}$ is the current used time slot list and $U_i (i \in [1, j])$ is the ith element in the list $\mathcal{U}$, RIFF algorithm firstly generates a random variable $k (k \in [1, j])$ to index the elements in the $\mathcal{U}$ as follows

$$I = \begin{cases} i + j - k + 1 & i < k; \\ i - k + 1 & i \geq k, \end{cases} \tag{11}$$

where the value I is the index of the element. RIFF algorithm then allocates the lowest indexed time slot which can satisfy the time requirement of the application to the current node; if no such used time slot exists, RIFF will randomly choose a free time slot for the node. Apparently, RIFF algorithm satisfies the Any-Fit constraint and meets the demand of the Almost Any-Fit constraint at the most of time. RIFF algorithm for on-line vehicle-slot matching is shown in Algorithm 2.

Algorithm 2 RIFF algorithm for on-line vehicle-slot matching.

```
Input: $\mathcal{U}$, $\mathcal{S}$ and cycle $t$
Output: time slot $s$ and waiting frames $w$
Initialize: $s = 0$, $w = -1$
if $t == 1$ then
    $s \leftarrow random(\mathcal{S} - \mathcal{U})$
    $w = 0$
else
    $k = random(1, len(\mathcal{U}))$
    for $i \in (1, len(\mathcal{U}))$ do
        $index = i$
        if $index < k$ then
            $index = index + len(\mathcal{U}) - k + 1$
        else
            $index = index - k + 1$
        $w = DTSS(\mathcal{Q}_{\mathcal{U}_{index}}, t)$
        if $w \neq -1$ then
            $s = \mathcal{U}_{index}$
            break
    if $s == 0$ then
        $s \leftarrow random(\mathcal{S} - \mathcal{U})$
        $w = 0$
return $s$,$w$
```

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of RIFF algorithm by using Matlab and conduct simulations to evaluate the efficiency of SS-MAC design.

A. Evaluating RIFF Algorithm

As the efficiency of DTSS algorithm has been theoretically proofed in the subsection III-C, we only evaluate the proposed on-line vehicle-slot matching approach. We compare RIFF algorithm with the following two alternative approaches.

- **Random Fit Approach.** In this approach, each node randomly chooses a time slot which can meet its demand.
- **FF Approach.** FF approach allocates the current node with the lowest nonempty indexed time slot which can satisfy its sharing requirement, assuming there is such

---

[6] R_{A}^{∞} is the asymptotic *worst-case ratio* (or asymptotic performance ratio, APR) and the number is the value that packings produced by algorithm A compare to optimal packings in the worst case .

TABLE I: Parameters used to evaluate the RIFF algorithm.

Parameters	**Value**
Number of time slots	$[1, 100]$
Cycles of applications (in $frames$)	$[1, 10]$
Number of vehicles	$[1, 1000]$
Matching rounds	$[1, 500]$
Simulation times	50

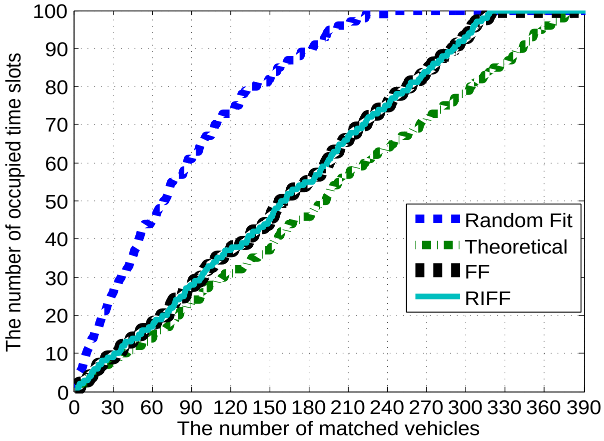


Fig. 5: The comparison of online matching results by different algorithms.

a time slot; if no such a time slot exists, FF approach assigns a free time slot to the current node.

Random Fit approach is easily implemented while FF approach has been proofed that it can achieve the best performance in terms of worst-case properties among all on-line matching heuristics. After the matching process, all nodes adopt DTSS algorithm to share a specific time slot in all matching approaches. The detailed simulation parameters are shown in Table I. Each frame is set with 100 time slots in the environment; the application cycle t of each vehicle is randomly generated ranging from 1 to 10, where the unit is a frame duration time; matching algorithms try to assign the time slot for a vehicle at each round. We consider the following two metrics to evaluate the proposed matching approaches:

1) **Number of occupied time slots.** It means the needed time slot resources to guarantee the QoS of matched vehicles.
2) **Sharing efficiency.** It refers to the ratio of the sharing capacity to the total capacity of a time slot.

Figure 5 shows the time slot occupied results under different matching algorithms; a theoretical value [7] is added for a better comparison. We can have the following three observations. First, RIFF algorithm can achieve the similar performance with FF algorithm as their two curves are tightly closed. Second, when resources are sufficient, RIFF algorithm can outperform the Random Fit algorithm; for instance, when there are 100 vehicles in the environment, Random Fit algorithm needs about 66 time slots for data transmission while the number of needed time slots can be reduced to only 30 by adopting RIFF algorithm and the value is very close to the theoretical value 26. More than half of resources are saved which greatly improves the utilization of resources. Third, when meets the resources shortage condition, RIFF algorithm can serve much more vehicles comparing with Random Fit algorithm; in specific, RIFF algorithm can support about 320 vehicles with 100 time slots while Random Fit algorithm can only support about 240 vehicles under the same resource condition.

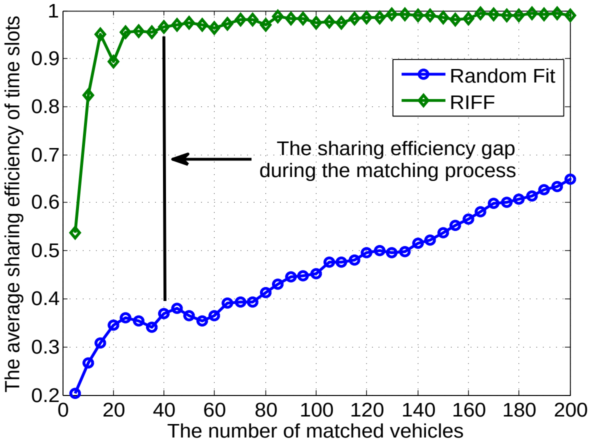


Fig. 6: The sharing efficiency comparison between the RIFF and the Random Fit algorithm.

Figure 6 shows the average sharing efficiency of time slots during the matching process. As the curve of FF algorithm is tightly close to RIFF, we just show the curve of RIFF for clear observations. We can have the following two main observations. First, the average sharing efficiency is always higher under RIFF algorithm compared with the values under Random Fit algorithm. Second, during the matching process, most sharing efficiency values achieved by RIFF algorithm are above 98% while most values achieved by Random Fit algorithm are under the 50%; this is also the main reason that RIFF can greatly improve the utilization of resources comparing with Random Fit algorithm.

B. Evaluating the SS-MAC Design

In this subsection, we conduct extensive simulations to evaluate the efficiency of SS-MAC, considering various road scenarios and resource conditions.

Simulation Setup. We use the Simulation of Urban Mobility (SUMO) [31] to conduct simulations evaluating the performance of SS-MAC. We consider two typical VANET road scenarios, i.e., highways and urban surface roads. In the highway scenario, a bidirectional 8-lane highway with 10 km long is set; each direction contains four lanes with a speed limit of 60 km/h, 80 km/h, 100 km/h and 120 km/h, respectively. In the urban scenario, four bidirectional 6-lane roads with 4 km long converge at an intersection; each of the three lanes in one direction is given a speed limit of 50 km/h, 60 km/h and 70 km/h respectively; traffic lights are set at each inbound road segment at the intersection with the duration of green light being 20 s.

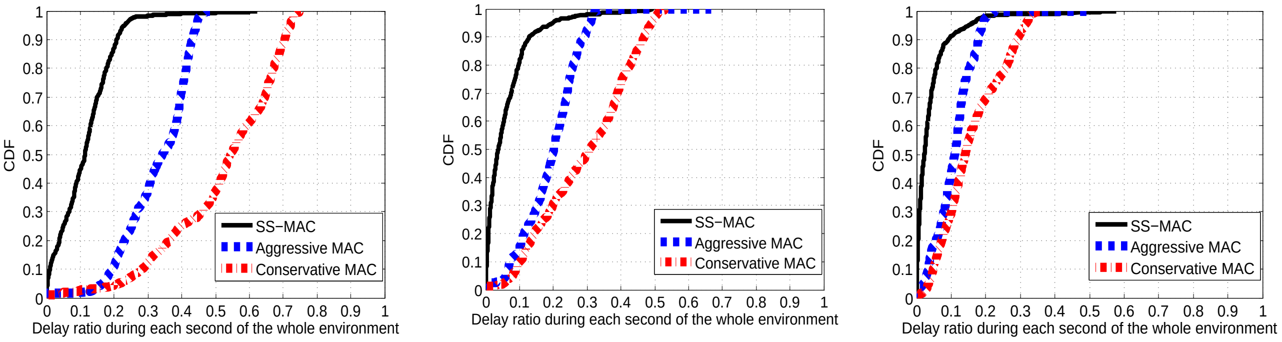
In both scenarios, vehicles have different performance parameters in terms of maximum velocity, acceleration ability and deceleration ability. Ten different sets of vehicle

---

[7] The theoretical value is just the sum size of matched vehicles. Apparently, this value can not be achieved in practical and is just a comparing reference.

TABLE II: Attributes of vehicles in simulations.

Attributes	Values	Description
maxSpeed	$[80, 240]$	The vehicle's maximum velocity (in km/h).
accel	$[1.0, 5.0]$	The acceleration ability of vehicles (in m/s^2).
decel	$[3.0, 10.0]$	The deceleration ability of vehicles (in m/s^2).
length	$[4.0, 7.0]$	The vehicle's length (in m).
minGap	$[3.0, 10.0]$	The minimum offset to the leading vehicle when standing in a jam(in m).
car-following model	$Krauss$	The model used for car following.
lane-changing model	$LC2013$	The model used for changing lanes.
sigma	$[0.5, 1.0]$	The car-following model parameter defining the driver imperfection (between 0 and 1).
impatience	$[0.5, 1.0]$	Willingness of drivers to impede vehicles with higher priority (between 0 and 1).



(a) With $N = 40$ (b) With $N = 50$ (c) With $N = 60$

Fig. 7: CDFs of delay ratio during each second of the whole environment under different resource conditions in the highway scenario.

TABLE III: Simulation parameters.

Parameters	Highway	Urban
Road length	$10\ km$	$4\ km$
Number of road segments	1	4
Number of intersections	0	1
Number of lanes on each road	8	6
Speed limit in lanes (in km/h)	$[60, 120]$	$[50, 70]$
Transmission range	$300\ m$	$300\ m$
Frame duration	$100\ ms$	$100\ ms$
Cycles (in $frames$)	$[1, 10]$	$[1, 10]$
Slot duration	$1\ ms$	$1\ ms$
Number of slots	$(40, 50, 60)$	$(80, 90, 100)$
Loaded vehicles	1630	1360
Running vehicles	$400 - 500$	$500 - 600$
Simulation time	$1000\ s$	$1000\ s$

parameters are configured according to the main types of vehicles on the market. Vehicles are driven under the Krauss car-following model and the LC2013 lane-changing model. In addition, driver imperfection and impatience parameters are also set to introduce the human factor into simulations. Table II summarizes the attributes of vehicles in simulations.

Vehicles are generated at the entrance of each road [8] with different rates to mimic normal traffic conditions, i.e., highway (10 vehicles/lane/minute), urban (6 vehicles/lane/minute). Each vehicle randomly chooses a performance parameter configuration and the destination road; when vehicles arrive at the destination, they will leave the simulation system. In all simulations, the transmission range R is set to be 300 m according to the observation that 802.11p-compatible onboard units can support reliable data transmission within 300 m [25]. Following the vehicle safety communications report of U.S. Department of Transportation [3], the time duration of a frame is set to be 100 ms satisfying the highest frequency requirement of safety applications and the application cycle t (in $frames$) is set to be [1,10]. Table III summarizes the simulation parameters [9].

Methodology. We compare the proposed slot sharing SS-MAC with the following two reasonable candidate MAC schemes.

- **Aggressive MAC scheme.** In this scheme, once a vehicle has acquired a time slot successfully, it will broadcast its information during each frame no matter what its application cycle is. By doing so, vehicles within the same THS can know its time slot acquisition and will not try to occupy the same time slot.
- **Conservative MAC scheme.** On the contrary, in this scheme, a vehicle just transmits its information when in need according to its application cycle. The advantage of this scheme is that only necessary resources are used.

We consider the **delay ratio** as the metric to evaluate the performance of SS-MAC. It refers to the number of unsuccessful transmissions [10] to the total number of needed transmissions according to the cycle of applications.

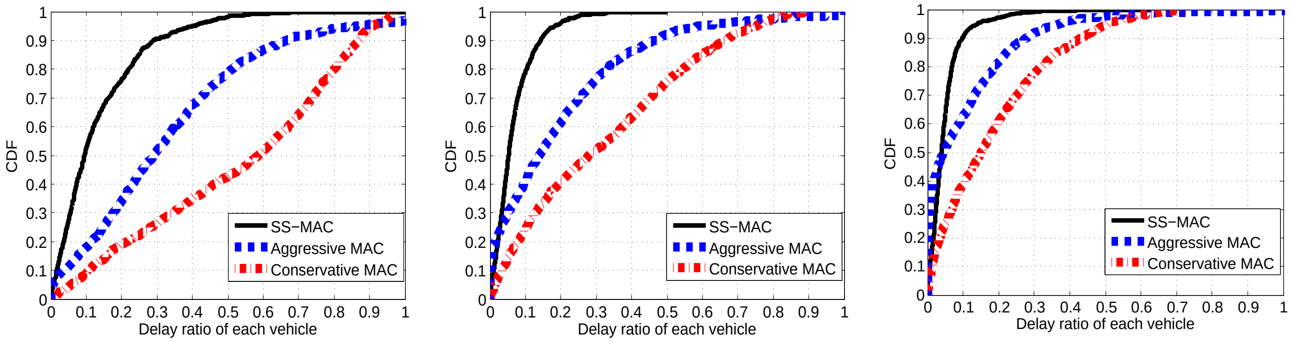
Performance comparison. In this subsection, we compare SS-MAC with other alternative MAC schemes under highway and urban scenarios. We focus on the delay ratio of the whole network and the delay ratio of each vehicle to consider the performance of SS-MAC.

---

[8] The highway can be with 8 entrance lanes while the urban is $3*4$ entrance lanes.

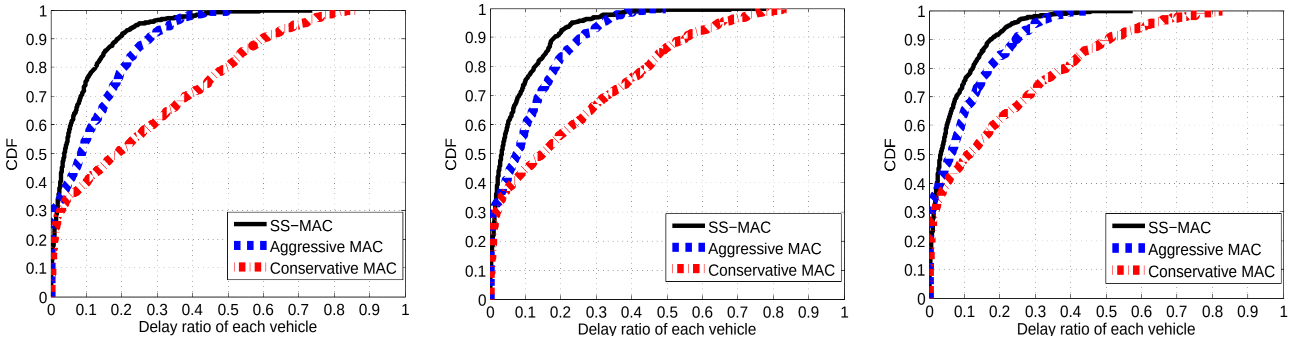
[9] Note that, the numbers of running vehicles are recorded at saturated traffic conditions and the number is larger in urban scenario due to the traffic lights.

[10] An unsuccessful transmission happens due to two reasons; one is the medium access collision during the transmission, the other one is the case without medium to access due to the unfairness of the resource allocation.



(a) With $N = 40$ (b) With $N = 50$ (c) With $N = 60$

Fig. 8: CDFs of delay ratio of each vehicle under different resource conditions in the highway scenario.



(a) With $N = 80$ (b) With $N = 90$ (c) With $N = 100$

Fig. 9: CDFs of delay ratio of each vehicle under different resource conditions in the urban scenario.

Figure 7 shows the cumulative distribution functions (CDFs) of delay ratio during each second of the whole network under different resource conditions in the highway scenario. We have the following three main observations. First, it can be seen that SS-MAC outperforms other schemes under all resource conditions; for instance, when $N = 40$, more than 80% delay ratios are smaller than the value of 0.2 in SS-MAC, while the value can reach about 0.43 in Aggressive MAC scheme and 0.7 in Conservative MAC scheme. Second, with more resources, although all MAC schemes can ease the delay ratio, the bending nature of curves do not change; specifically, the curve of SS-MAC is *convex* where the fast growth rate of y locates at the smaller x parts while the curves of the other two MACs are *concave* where the fast growth rate of y locates at the larger x parts. For instance, within the value of 0.1, the proportion can increase to about 40%, 80% and 90% respectively under different resource conditions in SS-MAC, while the proportion just increase to 2%, 15% and 40% respectively in Aggressive MAC scheme and 3%, 10% and 30% respectively in Conservative MAC scheme. The nature of concavity and convexity demonstrates the *inherent* efficiency of SS-MAC. Third, the CDF gaps between SS-MAC with the other two MACs decrease with the improving of the resource conditions; it demonstrates the stability of the SS-MAC in face of tense resource conditions.

Figure 8 shows CDFs of delay ratio of each vehicle under different resource conditions in the highway scenario. We have the following two main observations. First, SS-MAC can achieve the lowest delay ratio under all resource conditions; for example, when $N = 40$, more than 90% vehicles can achieve the delay ratio smaller than the value of 0.3 in SS-MAC, while the value can be about 0.66 and 0.88 in Aggressive MAC scheme and Conservative MAC scheme respectively. Second, the curves of SS-MAC are steep while the curves of the other two MACs are flat; specifically, to reach the proportion nearly 100%, the delay ratios can range from the value 0 to the value about 0.68, 0.5 and 0.41 respectively under different resource conditions in SS-MAC, while the values reach to 1, 1 and 0.74 respectively in Aggressive MAC scheme and 0.97, 0.9 and 0.72 respectively in Conservative MAC scheme; the curve of the flat shows the unfairness the medium access.

Similar results can also be observed in urban scenarios. we just show CDFs of delay ratio of each vehicle in Figure 9 due to the space limitations. From the figure, we can easily observe that SS-MAC can achieve supreme performance under all resource conditions.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel time slot-sharing MAC, named SS-MAC, to support diverse beacon rates for safety applications in VANETs. In specific, we have first introduced a circular recording queue to perceive occupancy states of time slots online and then designed a distributed time slot sharing approach called DTSS to efficiently share a specific time slot. In addition, we have developed the random index first fit algorithm, i.e., RIFF, to assist vehicle selecting a suitable time slot for sharing with maximizing the resource utilization of the network. We have theoretically proved the efficacy of DTSS design, and evaluated the efficiency of RIFF algorithm by using Matlab simulations. Finally, we have conducted extensive simulations considering various driving scenarios and resource conditions to demonstrate SS-MAC design; particularly, delay ratios of the overall system and each vehicle can be greatly reduced in highway and urban scenarios under all kinds of resource conditions. For our future work, we will investigate the dynamical resource assignment based on safety-awareness for safety applications and non-safety applications.

ACKNOWLEDGMENT

This research was supported in part by National Natural Science Foundation of China (Grants No. 61472255, 61420106010, 61373157, 61471236, 91638204, 61772340) and Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected Vehicles: Solutions and Challenges," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[2] SAE International, "Dedicated Short Range Communications (DSRC) Message Set Dictionary," Nov. 2009.

[3] CAMP Vehicle Safety Communications Consortium and others, "Vehicle safety communications project: Task 3 final report: identify intelligent vehicle safety applications enabled by DSRC," *National Highway Traffic Safety Administration, US Department of Transportation, Washington DC*, Mar. 2005.

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

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

[6] S. Subramanian, M. Werner, S. Liu, J. Jose, R. Lupoaie, and X. Wu, "Congestion Control for Vehicular Safety: Synchronous and Asynchronous MAC Algorithms," in *Proceedings of the Ninth ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*. New York, NY, USA: ACM, Jun. 2012, pp. 63–72.

[7] Q. Ye, W. Zhuang, L. Li, and P. Vigneron, "Traffic-Load-Adaptive Medium Access Control for Fully Connected Mobile Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9358–9371, Nov. 2016.

[8] Q. Ye and W. Zhuang, "Token-Based Adaptive MAC for a Two-Hop Internet-of-Things Enabled MANET," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1739–1753, Oct. 2017.

[9] Q. Chen, D. Jiang, and L. Delgrossi, "IEEE 1609.4 DSRC multi-channel operations and its implications on vehicle safety communications," in *2009 IEEE Vehicular Networking Conference (VNC)*, Oct. 2009, pp. 1–8.

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

[11] Y. Mertens, M. Wellens, and P. Mahonen, "Simulation-Based Performance Evaluation of Enhanced Broadcast Schemes for IEEE 802.11-Based Vehicular Networks," in *VTC Spring 2008 - IEEE Vehicular Technology Conference*, May. 2008, pp. 3042–3046.

[12] R. Reinders, M. van Eenennaam, G. Karagiannis, and G. Heijenk, "Contention window analysis for beaconing in vanets," in *2011 7th International Wireless Communications and Mobile Computing Conference*, Jul. 2011, pp. 1481–1487.

[13] M. Hadded, P. Muhlethaler, A. Laouiti, R. Zagrouba, and L. A. Saidane, "TDMA-Based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis, and Open Research Issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2461–2492, Jun. 2015.

[14] 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, Jul. 2004.

[15] H. A. Omar and W. Zhuang and L. Li, "VeMAC: A TDMA-Based MAC Protocol for Reliable Broadcast in VANETs," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1724–1736, Jun. 2013.

[16] X. Jiang and D. H. C. Du, "PTMAC: A Prediction-Based TDMA MAC Protocol for Reducing Packet Collisions in VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9209–9223, Nov. 2016.

[17] H. Zhou, N. Cheng, Q. Yu, X. Shen, D. Shan, and F. Bai, "Toward Multi-Radio Vehicular Data Piping for Dynamic DSRC/TVWS Spectrum Sharing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2575–2588, Oct. 2016.

[18] H. Zhou, B. Liu, T. H. Luan, F. Hou, L. Gui, Y. Li, Q. Yu, and X. Shen, "ChainCluster: Engineering a Cooperative Content Distribution Framework for Highway Vehicular Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2644–2657, Dec. 2014.

[19] S. A. A. Shah, E. Ahmed, F. Xia, A. Karim, M. Shiraz, and R. M. Noor, "Adaptive Beaconing Approaches for Vehicular Ad Hoc Networks: A Survey," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–15, Jul. 2016.

[20] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 84–94, Dec. 2007.

[21] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wireless Networks.*, vol. 8, no. 2/3, pp. 153–167, Mar. 2002.

[22] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in uav-aided networks," *IEEE Network*, vol. 30, no. 1, pp. 100–105, Jan. 2016.

[23] Z. M. Fadlullah, Y. Kawamoto, H. Nishiyama, N. Kato, N. Egashira, K. Yano, and T. Kumagai, "Multi-hop wireless transmission in multi-band wlan systems: Proposal and future perspective," *IEEE Wireless Communications*, vol. PP, no. 99, pp. 1–6, Oct. 2017.

[24] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9457–9470, Dec. 2016.

[25] F. Lv, H. Zhu, H. Xue, Y. Zhu, S. Chang, M. Dong, and M. Li, "An Empirical Study on Urban IEEE 802.11p Vehicle-to-Vehicle Communication," *13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, Jun. 2016.

[26] "IEEE Draft Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation," *IEEE Unapproved Draft Std P1609.4/D6.0*, Mar. 2010.

[27] F. Bai, D. D. Stancil, and H. Krishnan, "Toward Understanding Characteristics of Dedicated Short Range Communications (DSRC) from a Perspective of Vehicular Network Engineers," *Proceedings of ACM MobiCom*, Sep. 2010.

[28] R. Ben-El-Kezadri and G. Pau, "Timeremap: stable and accurate time in vehicular networks," *IEEE Communications Magazine*, vol. 48, no. 12, pp. 52–57, Dec. 2010.

[29] E. G. Coffman Jr, J. Csirik, G. Galambos, S. Martello, and D. Vigo, "Bin packing approximation algorithms: survey and classification," pp. 455–531, Jun. 2013.

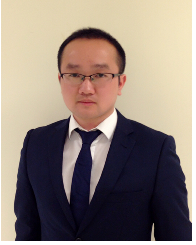
[30] D. S. Johnson, "Fast algorithms for bin packing," *Journal of Computer and System Sciences*, vol. 8, no. 3, pp. 272–314, Jun. 1974.

[31] DLR Institute of Transportation Systems, "Sumo: Simulation of urban mobility," http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-1213/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2017.2780829, IEEE Transactions on Vehicular Technology

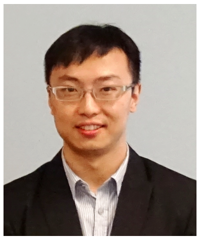
Feng Lyu received the BS degree in software from Central South University in 2013. He is pursuing his Ph.D degree in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. From October 2016 to October 2017, he was a visiting Ph.D. student at the Broadband Communications Research (BBCR) group in the Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada. His research interests include vehicular ad hoc networks and cloud computing.

Hongzi Zhu received his Ph.D. degree from SJTU in 2009, and M.Eng. and B.S. from Jilin University in 2001 and 2004, respectively. He was a Post Doctoral Fellow at University of Waterloo and the Hong Kong University of Science and Technology in 2009 and 2010, respectively. He is now an associate professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU). His research interests include wireless networks, mobile sensing and mobile computing, and vehicular ad hoc networks. He received the Best Paper Award from IEEE Globecom 2016. He is a member of IEEE Computer Society and IEEE Communications Society. For more information, please visit http://lion.sjtu.edu.cn.



Haibo Zhou (M'14) received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014. From 2014 to 2017, he worked as a Post-Doctoral Fellow with the Broadband Communications Research Group, ECE Department,University of Waterloo. Currently, he is an Associate Professor with the School of Electronic Science and Engineering, Nanjing University. His research interests include resource management and protocol design in cognitive radio networks and vehicular networks.

Wenchao Xu received the B.E. and M.E. degrees from Zhejiang University, Hangzhou, China, in 2008 and 2011, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. In 2011, he joined Alcatel Lucent Shanghai Bell Co. Ltd., where he was a Software Engineer for telecom virtualization. His interests include wireless communications with emphasis on resource allocation, network modeling, and mobile data offloading.



Ning Zhang [M'15] (ning.zhang@tamucc.edu) received his Ph.D degree from the University of Waterloo in 2015. He is now an assistant professor in the Department of Computing Science at Texas A&M University-Corpus Christi. Before that, he was a postdoctoral research fellow at the University of Waterloo and at the University of Toronto. He is the recipient of the Best Paper Award at IEEE Globecom 2014 and IEEE WCSP 2015. He is a lead guest editor of Wireless Communications and Mobile Computing and International Journal of Distributed Sensor Networks, and a guest editor of Mobile Information System. His current research interests include next generation wireless networks, software defined networking, vehicular networks, and physical layer security.

Minglu Li received the PhD degree in computer software from Shanghai Jiao Tong University in 1996. He is a full professor and the vice dean of the School of Electronics Information and Electrical Engineering, the director of Network Computing Center at Shanghai Jiao Tong University. His research interests include grid computing, services computing, and sensor networks. He is a member of the IEEE.

Xuemin (Sherman) Shen (IEEE M'97-SM'02-F09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a University Professor and the Associate Chair for Graduate Studies, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Dr. Shen's research focuses on resource management, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, Infocom'14, IEEE VTC'10 Fall, and Globecom'07, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the General Co-Chair for ACM Mobihoc'15, Chinacom'07 and the Chair for IEEE Communications Society Technical Committee on Wireless Communications. He also serves/served as the Editor-in-Chief for IEEE Internet of Things Journal, IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, and IEEE Communications Magazine, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.