MC-Safe: Multi-channel Real-time V2V Communication for Enhancing Driving Safety

46

YUNHAO BAI and KUANGYU ZHENG, Department of Electrical and Computer Engineering, The Ohio State University, USA
ZEJIANG WANG, Department of Mechanical Engineering, University of Texas at Austin, USA
XIAORUI WANG, Department of Electrical and Computer Engineering, The Ohio State University, USA
JUNMIN WANG, Department of Mechanical Engineering, University of Texas at Austin, USA

In a Vehicular Cyber Physical System (VCPS), ensuring the real-time delivery of safety messages is an important research problem for Vehicle to Vehicle (V2V) communication. Unfortunately, existing work relies only on one or two pre-selected control channels for safety message communication, which can result in poor packet delivery and potential accident when the vehicle density is high. If all the available channels can be dynamically utilized when the control channel is having severe contention, then safety messages can have a much better chance to meet their real-time deadlines.

In this article, we propose MC-Safe, a multi-channel V2V communication framework that monitors all the available channels and dynamically selects the best one for safety message transmission. During normal driving, MC-Safe monitors periodic beacons sent by other vehicles and estimates the communication delay on all the channels. Upon the detection of a potential accident, MC-Safe leverages a novel channel negotiation scheme that allows all the involved vehicles to work collaboratively, in a distributed manner, for identifying a communication channel that meets the delay requirement. MC-safe also features a novel coordinator selection algorithm that minimizes the delay of channel negotiation. Once a channel is selected, all the involved vehicles switch to the same selected channel for real-time communication with the least amount of interference. Our evaluation results both in simulation and on a hardware testbed with scaled cars show that MC-Safe outperforms existing single-channel solutions and other well-designed multi-channel baselines by having a 23.4% lower packet delay on average compared with other well-designed channel selection baselines.

CCS Concepts: • **Networks** → **Cyber-physical networks**; **Mobile networks**;

Additional Key Words and Phrases: Cyber-physical system, V2V networks, real-time communication

ACM Reference format:
Yunhao Bai, Kuangyu Zheng, Zejiang Wang, Xiaorui Wang, and Junmin Wang. 2020. MC-Safe: Multi-channel Real-time V2V Communication for Enhancing Driving Safety. *ACM Trans. Cyber-Phys. Syst.* 4, 4, Article 46 (June 2020), 27 pages.
https://doi.org/10.1145/3394961

Authors' addresses: Y. Bai, K. Zheng, and X. Wang, Department of Electrical and Computer Engineering, The Ohio State University, 2015 Neil Ave, Columbus, Ohio, USA, 43210; emails: {bai.228, zheng.722}@osu.edu, xwang@ece.osu.edu; Z. Wang and J. Wang, Department of Mechanical Engineering, University of Texas at Austin, Austin, 204 E. Dean Keeton Street, Austin, Texas, USA, 78712; emails: wangzejiang@utexas.edu, jwang@austin.utexas.edu.

2378-962X/2020/06-ART46 $15.00
https://doi.org/10.1145/3394961

1 INTRODUCTION

Next-generation vehicles are expected to be equipped with a variety of sensors, actuators, and devices for Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications. As such, they constitute a new kind of Cyber Physical System (CPS), called Vehicular CPS (VCPS) [3, 29]. In VCPS, the Vehicle-to-Vehicle (V2V) network is a major cyber component, together with the physical parts such as autonomous driving and vehicle control applications, whose performance relies heavily on the timeliness of V2V communications. For V2V networks, enhancing driving safety is a major research objective [8], due to the fact that car accidents cost nearly 1.3 million lives every year [20]. By exchanging safety messages with critical vehicle information (e.g., car speed, location, direction), vehicles can be notified in real time if there is a potential accident. Such safety considerations have already been included in the current V2V standards, such as Wireless Access in Vehicular Environments (WAVE), a widely adopted vehicle communication protocol. The proposed WAVE protocol provides both safety and data services on the Dedicated Short-Range Communication (DSRC) band, using the IEEE 802.11p standard [1, 15].

To deliver safety messages in a timely manner, delay is one of the most important requirements for V2V communication in a VCPS. Generally, there are two types of safety messages: periodic safety messages (e.g., GPS location, speed) and event-driven safety messages (e.g., driving actions like braking and lane changing) [28]. Both types have some deadline requirements, but the event-driven messages commonly have more stringent deadlines. For example, on a crowded highway, missing the deadline of a safety message that alerts a sudden braking may cause a rear-end collision. Similarly, if a lane-changing message is not delivered in real time, then unsafe lane merging could cause severe accidents. The V2V communication deadline for transmitting event-driven safety message can be as short as 20 ms [28], based on the car speeds and their distance to each other. Even for periodic messages, some soft deadline requirements are also necessary, because these messages contain time-sensitive critical information about the vehicle's dynamics, such as velocity, and position, which may become outdated after a short period of time.

However, meeting the V2V communication deadline is challenging, particularly when the vehicle density is high. For example, the vehicle density of a downtown area is commonly more than 5,000 vehicles per square mile and can be 1.5 times during rush hours [21]. Such a high density can cause safety messages to miss their deadlines, due to significant wireless channel contentions on the control channel used by the current WAVE (802.11p) protocol, for two major reasons: First, when more vehicles compete for the limited bandwidth resources, the packet delay can become unbounded with the Carrier-Sense Multiple Access (CSMA) mechanism. Second, a high vehicle density can lead to a higher chance of having the well-known hidden terminal problem, which in turn can result in more packet dropping at the receiver vehicle. Such a channel contention problem is mainly due to the fact that WAVE is designed to transmit safety messages only on the control channel, despite that seven non-overlapping channels are actually available in the DSRC band for V2V communication. If other channels can also be utilized when the control channel is having contention, then safety messages can have a much better chance to meet their real-time deadlines.

Most existing work on improving the communication performance of safety messages focuses on adapting transmission rate and power [6, 28, 34], as well as message priority or period [9, 23, 31, 36]. Although those methods can be effective when the vehicle density is low, they still transmit all the safety messages on the single control channel where the total network capacity is limited. There are indeed some recent studies that investigate multiple channels for safety message communication [11, 30, 33]. However, instead of dynamically selecting the best channel for real-time communication, they mainly have two pre-selected channels for safety and non-safety messages, respectively. For example, some studies have proposed to adjust the time interval length of staying on the control channel to reduce the delay of vehicles' safety messages [30, 33]. Due to the limited channel choices, they can still have inferior performance when the vehicle density is high. To our best knowledge, existing work does not *dynamically* select the best channel from all the channel resources provided by DSRC to improve the *real-time performance* of safety messages.

We propose *MC-Safe*, a multi-channel V2V communication framework that monitors all the available channels and dynamically selects the best one for safety message transmission in an emergency scenario of VCPS. MC-Safe features a novel channel negotiation scheme that is activated whenever two or more cars are estimated to have a potential accident. To facilitate channel negotiation, a coordinator is selected, in a distributed manner, to be the one that suffers the least amount of interference among all the involved cars. With the coordinator, all the involved cars work collaboratively to identify a channel that can meet the specified delay and packet error ratio requirements for every car. Afterward, all the involved cars switch to the selected channel for real-time safety message communication, without suffering the interference from other cars in the vehicular network. MC-Safe is robust to varying channel conditions, because it can adapt its model to better estimate channel delays. Our evaluation both in simulation and on a hardware testbed with scaled cars show that MC-Safe outperforms existing single-channel solutions and other well-designed multi-channel baselines, by having a 12.3% lower deadline miss ratio and an 23.4% lower packet delay on average. Compared to WAVE, the state-of-the-practice solution, MC-Safe reduces the average delay of safety messages from 300 to 20 ms when the vehicle density is high.

Specifically, this article makes two major contributions:

- We observe that the existing work on V2V safety message communication focuses only on one or two pre-selected channels, and thus may have a low packet delivery ratio and a high deadline miss ratio when the car density is high. Accordingly, we propose to explore all the seven available non-overlapping channels for better real-time performance and a lower chance of having collisions.
- We propose MC-Safe, a multi-channel V2V communication framework that monitors all the available channels and dynamically selects the best one for safety message transmission. MC-Safe features novel coordinator selection and channel negotiation schemes that let all involved cars work collaboratively to find a channel that meets the desired delay and packet error rate requirements.

The rest of the article is organized as follows: Section 2 discusses the related work. Section 3 motivates our work by comparing message transmission on one single channel or multiple channels under different road situations. Section 4 introduces the design of MC-Safe. Section 5 presents the evaluation results. Section 6 concludes the article.

2 RELATED WORK

Due to the complex environment of V2V network, many studies conduct analysis of the general transmission performance in current V2V protocol [17, 19, 32, 34]. Campolo et al. have shown that the transmission delay can increase dramatically when the vehicle density becomes high [19]. On a typical road intersection with only 50 m of distance between vehicles, the control channel can be saturated [17]. These studies demonstrate that the current 802.11p standard is not sufficient for the transmission of delay-sensitive data, especially for the V2V real-time message communication.

Meanwhile, many recent studies are proposed to improve safety message transmission in the current V2V network based on WAVE standard. One major direction is alleviating the control channel workload by adjusting the message transmission rate or power [6, 28, 34]. Some studies also try to utilize other knobs, such as message priority, beaconing frequency, or duplicated packets [9, 12, 30, 31, 36]. For example, Xiang et al. [31] propose to add priorities to different messages to avoid collision on the control channel. Although the aforementioned methods improve the performance of safety message broadcasting, they still transmit on the single control channel where the total network capacity is limited.

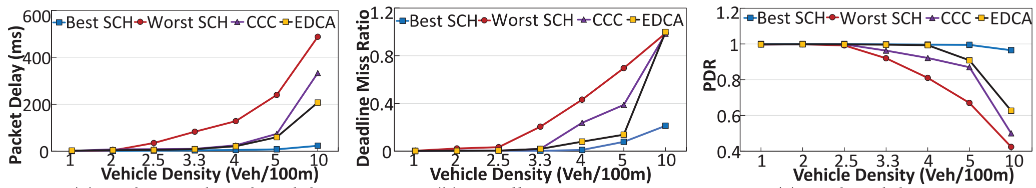
Recently, some studies begin to consider safety message broadcasting under the multi-channel scenarios [10, 11, 33]. For example, CRN-VANETs [10] aims to reduce data contention in the control channel, but it does not consider the stringent time requirements in a potential accident. Ghandour et al. propose to form a sub-network for each channel to deliver the event-driven message in time [11]. Yao et al. proposes to calculate the optimal bandwidth resource allocation for multi-channel V2V network. However, those studies mainly use one or two pre-selected channels for safety and non-safety message transmission without explicit delay consideration and do not consider real-time transmission requirements. In contrast, MC-Safe dynamically selects a channel that meets the real-time requirement through distributed channel negotiation among vehicles.

3 MOTIVATION

We now motivate the design of MC-Safe by investigating the real-time communication performance of WAVE, the state-of-the-practice, in a typical road scenario [1]. We also test Enhanced Distributed Channel Access (EDCA), which is the currently used for quality-of-service (QoS) support inheriting from 802.11e, to show that even a priority-based single-channel solution still cannot achieve the desired communication delays. EDCA can be used for transmitting emergency messages on the Common Control Channel (CCC) with four Access Categories (AC(0) to AC(3)). In EDCA, each AC has its own queue and corresponds to a different priority. Each AC is also assigned with a different carrier sensing window and a backoff counter to ensure the priority for that AC. Generally, AC(0) refers to the highest priority and AC(3) is the lowest. The details of EDCA can be found in Reference [9]. In this experiment, the emergency messages are given highest priority by being assigned in the AC(0) while the broadcast are assigned in AC(3) to see the maximum improvements of EDCA.

We have implemented the WAVE and EDCA algorithms and associated lower layer functions in the ns-2 network simulator. For the test, we consider a six-lane highway: Each lane's width is 4 m, and the total road length is 1,000 m. There are six Road-Side Units (RSUs) serving as receivers on the service channels (SCHs, i.e., channels other than the control channel), and the RSU is equipped with multiple radios so it can work on all the available service channels. We use Poisson distribution to model the distance of vehicles on the road, which is widely used for the road traffic analysis [4]. The vehicle density is set with parameter λ (the average inter-vehicle distance) of Poisson distribution; the average transmission interval is set to be 20 ms and the packet size is 300 bytes. Vehicles on the CCC are transmitting beacons with a frequency of 10 Hz, and the transmission interval of non-safety services is also 20 ms. We apply a realistic V2V network propagation model measured at 5.9-GHz band [7]. We define "service vehicle ratio" as the number of vehicles that are transmitting on channels other than the CCC divided by the total number of vehicles on the road. The vehicle using non-safety services is randomly assigned to one channel among SCH1 to SCH4. We adjust the vehicle density and service vehicle ratio to test different road traffic scenarios. In the motivation test, we choose the vehicles of interest as two neighboring cars in the middle of this road segment to avoid the inaccuracy of the two ends in the road segment.

For the real-time vehicle control system for accident prevention, a typical control period is usually 20 ms [28], and every packet needs to be received within its period to have the correct control action. As shown in Figure 1(a), the average package delay of using the CCC already becomes longer than 20 ms when the vehicle density is just 4 vehicles/100 m (about 1,600 per square mile), which is lower than the 2,000 vehicles/mile^2 vehicle density in most suburban areas [21]. In fact, the delay increases almost exponentially due to the backoff mechanism used in the 802.11 protocol. Note that a message delivered after the 20-ms delay requirement is outdated for real-time vehicle



(a) End-to-end packet delay (b) Deadline miss ratio (c) Packet delivery ratio

Fig. 1. Average packet transmission delay, packet delivery ratio, and deadline miss ratio (deadline = 20 ms) under different vehicle density with a fixed service vehicle ratio of 0.5.

control and could be misleading. Figure 1(b) shows the deadline miss ratio given the 20-ms deadline requirement. We can see that given a vehicle density higher than 5 vehicles/100 m, the deadline miss ratio increases rapidly, due to the severe contention on both the CCC and SCHs. Even with the best SCH, the deadline miss ratio is still unacceptable given a high vehicle density (10 vehicles/100 m). Meanwhile, Figure 1(c) shows that the Packet Delivery Ratio (PDR) of using the CCC drops significantly as the vehicle density increases. Though EDCA slightly improves the performance of CCC by assigning the emergency messages with the highest priority, its performance still shows long delays with a high vehicle density. To conclude, transmitting safety messages only on the single control channel can result in a long delay with a poor PDR. However, the delay and PDR of the best service channel (i.e., the service channel with the best delay or PDR result) remain small when the vehicle density increases. The reason is that the best service channel is far less crowded compared to the CCC. However, in the worst case, a service channel could perform even worse than the CCC, indicating that the channel selection upon an emergency scenario is not trivial: A bad choice of the service channel could harm the performance of the real-time vehicle control system. This evaluation provides a strong motivation to consider using other channels instead of the CCC to do real-time safety message transmission, especially given a high vehicle density. Moreover, The channel must be chosen carefully to avoid a even longer delay and poorer PDR.

4 DESIGN OF MC-SAFE

In this section, we first introduce the overview of MC-Safe, with its general workflow and major components. We then introduce the detailed design of each part.

4.1 Design Overview

MC-Safe aims to dynamically select the best channel for real-time safety message transmission in a pre-crashing scenario. Generally, when a possible collision for cars is predicted (determined by the vehicle trajectory predictor), distributed MC-Safe on the involved cars shall start negotiation to find the best channel between cars, establish the communication in a short time, and enable real-time and reliable safety message exchange for collision prevention control. We assume that the accident size is relatively small based on the study [2], and the traffic topology changes are relatively small and negligible during the negotiation phase with a micro-second level delay.

As examples, let us consider that there are two potential accidents: One is caused by lane changing (left one) while the other is caused by a hard braking (upper-right one). As shown in Figure 2(a), all the vehicles on the road listen on the CCC and receive the beacons from other vehicles, which contain their channel usages information. With this information, the vehicles can perform channel modeling and estimate the quality for each service channels periodically. Moreover, the received beacons are also used to calculate the Time To Collision (TTC) based on position information to predict a potential accident and form the accident avoidance group. Upon detecting a potential accident, the vehicles within a TTC threshold first form a temporary

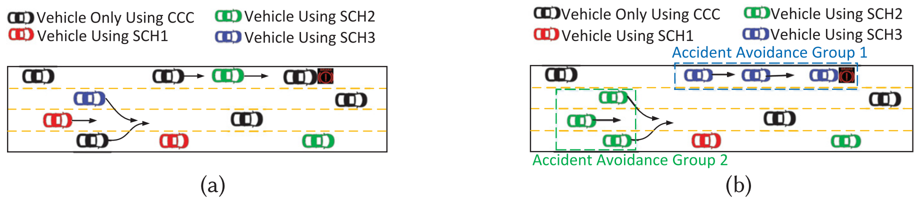


Fig. 2. An illustration of channel changes during the potential accidents. (a) Before predicting the potential accidents. (b) After predicting the potential accidents.

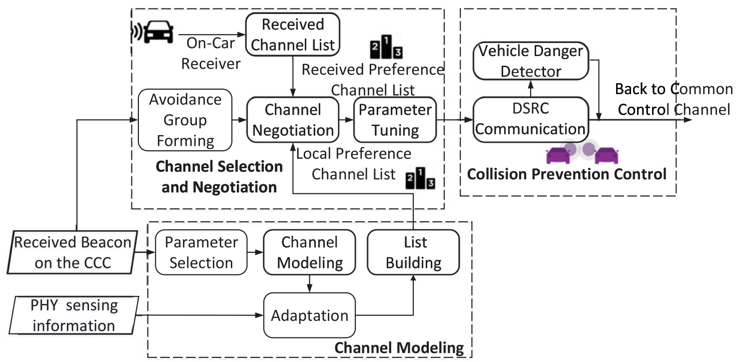


Fig. 3. Overview of the MC-Safe framework.

accident avoidance group based on the received beacons from nearby vehicles and change their channels to a selected channel that can meet the real-time communication requirement. Then, the involved vehicles send their vehicle dynamics through the V2V network, in a distributed manner, to all the involved cars in their respective groups on a channel other than the CCC, as shown in Figure 2(b). Thus, the accidents can be avoided. Note that different groups may decide to use different channels, and their communications can occur simultaneously without interfering with each other. After the dangerous condition is resolved (e.g., car distance becomes longer than the threshold), every involved car in the group will change its communication channel back onto the CCC. Figure 3 shows the overview of MC-Safe. Generally, MC-Safe realizes these functions with two major components: (1) Channel Modeling and (2) Channel Negotiation and Selection.

Channel Modeling. Channel modeling component on each vehicle is conducted periodically to estimate the packet delay and delivery ratio on all channels, with the delay and PDR requirements of the safety message. Based on such information, it evaluates the conditions of all the available channels using the theoretical model. After that we combine the results from the theoretical model and the PHY layer sensing information to adjust model to meet the current road situation. Then we constructs a local Channel Preference List (CPL), which is ordered by channel quality from the best to the worst, as the input to the channel negotiation and selection component.

Channel Negotiation and Selection. This component is invoked before any potential accident to find the best channel for the vehicles involved in the potential accident to perform real-time safety message communication. After a potential accident is predicted, based on their own CPLs and the CPLs received from other vehicles, the vehicles involved in the accident start the negotiation quickly to select the channel with the best quality for all the involved vehicles. If the selected channel cannot meet the requirements of safety message transmission, then MC-Safe will suppress the non-safety transmission of other vehicles on the selected channel to meet the requirements.

Table 1. Typical Signal Deadline Requirement in CAN

Message Type	Deadline	Jitter	Actuator
Steering Control	5 ms	0.2 ms	Motor Controller
Speed Control	20 ms	0.7 ms	Vehicle Controller
Emergency Brake	40 ms	0.5 ms	Brake
Shift in Progress	20 ms	1.4 ms	Motor Controller

The key parts in the design of MC-Safe are to (1) estimate the communication quality of each channel by channel modeling (Section 4.2) and (2) find the best channel among all involved vehicles by channel selection and negotiation (Section 4.3). The important notations used in the rest of the design of MC-Safe are listed as follows:

- p: Probability of transmission failure of one packet.
- T_i: Transmission interval of the safety packet.
- d_i: The delay of a single packet.
- Ω: The Maximum allowed interval between two consecutive packets.
- δ, Δ: Threshold probabilities for packet delivery ratio and delay requirement, respectively.
- p_b: The probability of backing off in the 802.11p backoff mechanism.
- CW_i: The maximum backoff counter of the ith retry.
- CW_{max}: The maximum backoff window defined in the 802.11p protocol.
- s_i: The average backoff delay for the ith retransmission in the 802.11p backoff mechanism.
- N_c: The number of vehicles in the transmission range.
- N_h: The number of vehicles serving as hidden terminals.

4.2 Channel Modeling

Channel modeling aims to decide whether one channel can meet the requirements of the collision prevention control system. We first introduce how to quantify the requirements of the control system, then we introduce how to estimate the performance for each channel in the real world.

4.2.1 Delay Requirement. Different emergency conditions can have different requirements of delay and PDR. For example, the adaptive cruise control system or hard brake reaction system require a 25-Hz sampling rate [22]; the lateral motion control proposed in Reference [26] requires a sampling rate of 50 Hz. To determine the requirement for each element in vehicle dynamics, we derive the deadlines from the Control Area Network (CAN) scheduling standard [27]. The requirement for each signal in the CAN system is listed in Table 1. With the consideration of variation, requirements are set as the deadline listed in Table 1 minus the jitter.

4.2.2 Channel Modeling. First, we use two well-known constraints adopted by many networked control systems (e.g., autonomous vehicles, industrial automation, and robots [16, 24]) to evaluate the performance and stability of each channel: (a) Maximum Allowable Transfer Interval (MATI) and (b) Maximum Allowable Delay (MAD) requirement. However, due to specific V2V conditions in our case (i.e., sophisticated backoff mechanism in 802.11p), we further revised the formulation, with details presented later in Equations (5) and (7). The original constraints are defined as follows:

$$\begin{aligned} \text{MATI}: &\quad 1 - p^{\lceil \frac{\Omega}{T_i} \rceil} \geq \delta \\ \text{MAD}: &\quad d_i \leq \Delta, \end{aligned} \tag{1}$$

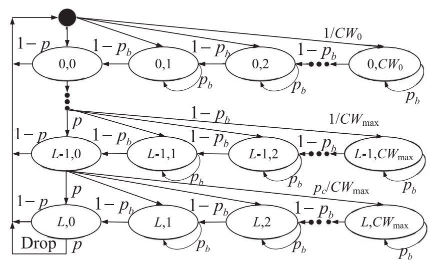


Fig. 4. Markov chain model of 802.11 backoff mechanism.

where the thresholds (Ω, δ, and Δ) are determined by the control system. Specifically, Δ is determined from the CAN standard for automobiles, and Ω is determined by the recommended value in vehicular safety applications [16]. Generally speaking, MAD requires that a packet should not have a delay larger than the threshold Δ, and MATI requires that the probability is larger than δ to receive at least one packet successfully in the T_i period of time. To derive the transmission failure probability p and packet delay d_i, we model the backoff procedure in 802.11p as a Markov process (shown in Figure 4) [5], which is proved to be sufficiently accurate compared to the performance measured in real world [5, 35, 37]. Based on the analysis proposed by Yao et al. [32], the backoff probability p_b and failure probability p shown in Figure 4 can be calculated as follows:

$$p_b = 1 - e^{-N_c \tau}, \tag{2}$$

$$p = 1 - e^{-N_c \tau} e^{-N_h \tau T_v / \sigma}, \tag{3}$$

where N_c is the number of vehicles in the carrier sensing area of the currently considered channel; N_h is the number of hidden terminals. These two parameters on the road can be known from the beacon signal sent by others, as the position and the channel usage information for a vehicle can be added into the periodic beacons and sent by other vehicles. Using the positions of other vehicles, the vehicle itself can determine whether another vehicle is a direct neighbor (N_c), a hidden interference (N_h) or not an interference node (due to weak signal strength) on a service channel based on the propagation model of 5.9 Ghz wireless signal [7]. T_v is the hidden-terminal vulnerable period, which is related to the packet length; τ is the transmission probability in one slot, whose value can be found in Reference [5] with known N_c only. σ is the length of a time slot. Since we are modeling the delay on the service channel (SCH), we assume that there are always packets in the MAC queue waiting for transmission [30]: This assumption can give us the upper bound of these parameters.

Based on the above models, we propose to calculate the total packet delay d_i in the following way, which is the summation of two parts: The service delay $\frac{1}{\mu}$ and the queuing delay $E[W]$. Using the value of p_b and p, we can calculate $\frac{1}{\mu}$ as the summation for the backoff delay for each retransmission multiplied by its corresponding probability. Mathematically, $\frac{1}{\mu}$ can be calculated as:

$$\frac{1}{\mu} = \sum_{i=0}^{L} (1-p)p^i \frac{\min(2^i CW_0, CW_{max})}{2} E[X], \tag{4}$$

where $E[X]$ is the expected time for the backoff counter decreasing by one, and it can be calculated based on the probability of a time slot being being busy p_b, having a collision p_c, and respective Interval Frame Space (IFS) time values defined in 802.11p, i.e., $E[X] = \text{DIFS}/p_b + \text{EIFS}/p_c$. $(1-p)p^i$ represents the probability of one packet going through i times of retransmission, and the other terms in the summation represent the expected delay for the i^{th} retransmission, i.e., s_i. The queuing delay $\mathrm{E}[W]$ can be directly calculated with M/G/1 queue model [32].

In Equation (1), T_i is not a constant value in 802.11p and cannot be used directly. Even though in the 802.11 backoff mechanism, it has a default upper bound retransmission times L in Figure 4, there are specific related delay requirements for the V2V real-time communication. Therefore, we propose to derive a new form of MATI requirement compatible with 802.11p. We define a new parameter, the maximum allowable times of retransmission L_m, which is calculated as:

$$\arg\max_{L_m} \sum_{i=0}^{L_m} s_{\mathrm{mod}(i,L)} <= \Omega. \tag{5}$$

The expected delay s_i for i^{th} retransmission is

$$s_i = \frac{\min(2^i CW_0, CW_{max})}{2} \mathrm{E}[X], \tag{6}$$

and then the MATI requirement in Equation (1) can be re-written as follows:

$$1 - p^{L_m} \geq \delta. \tag{7}$$

L_m is easy to get and has solid physical meaning in 802.11p compared to T_i. By using the above proposed constraint in Equation (7), MC-Safe on every vehicle can efficiently conduct the above channel analysis distributively by checking the number of vehicles within one-hop distance N_c, and the number of hidden terminal from the beacons of one-hop neighbors N_h on each channel, calculate L_m and s_i with Equations (5) and (6), and check whether the given channel can meet the MATI or not by Equation (7) (where p is calculated in Equation (3)).

4.2.3 Model Adaptation. The above channel estimation utilizes the N_c and N_h information in the beacon messages from other vehicles directly. However, the road environment can be complex, and the beacon messages can be interfered with errors. Therefore, we further design a dynamic adaptation scheme in MC-Safe to improve the channel estimation accuracy. To estimate the packet drop rate and delay, besides using the beacon messages, MC-Safe also measures the network status by leveraging the PHY layer sensing information to adjust idle probability p_i, collision probability p_c and busy probability p_b. In each period, it monitors each channel and collects the counts of the idle slot, the busy slot and the collision slot, then calculates the empirical probabilities for p_i', p_c', and p_b'. Then, to get the final results for adaptation $\mathbf{p^f} = (p_i^f, p_c^f, p_b^f)$ in the coming period, MC-Safe uses a weighted summation of the theoretical values and empirical values in the previous period with a dynamic factor α, i.e., p_i^f is updated as follows:

$$p_i^f = \alpha p_i + (1-\alpha) p_i'. \tag{8}$$

To determine the value of α, MC-Safe periodically uses the monitored empirical probabilities (p_i^f, p_c^f, p_b^f) in the current period as the ground truth and calculates the value of α that provides the best estimation accuracy. Then, the updated α is used in the estimation for the next coming period.

After calculating $\mathbf{p^f}$, we re-estimate the delay d_i and packet error rate p. First, we re-estimate N_c and N_h with the updated probabilities p_i^f, p_i^f, and p_b^f. This step is done by finding the values of N_c and N_h that yield the closed result of p_i^f, p_i^c, and p_b^f using the Markov model. Then, the packet error rate p and packet delay d_i can be adapted to the real-world environment. This adaptation process is applied periodically, so that the estimation can be updated in a timely manner to reflect the current road situation. The adaptation period can be set based on the channel conditions and road situations. In our simulation, we set the adaptation period as 2 s, because the topology of the vehicular network does not change much given a typical speed, and each vehicle can have sufficient sensing information to make the empirical probabilities $\mathbf{p^f}$ stable.

ALGORITHM 1: Model Adaptation

Input: Empirical probabilities of the current period $\mathbf{p'} = (p'_i, p'_c, p'_b)$, empirical probabilities of the previous period $\mathbf{p^p}$=(p^p_i, p^p_c, p^p_b), theoretical probabilities $\mathbf{p} = (p_i, p_c, p_b)$, weight parameter α
Output: packet delay d_i, packet error rate p
1: Update $\alpha = \arg\min_\alpha(\|\mathbf{p'} - \mathbf{p^p}\|)$ for the current adaptation period
2: Calculate $\mathbf{p^f} = (p^f_i, p^f_c, p^f_b)$ by weighted average:
3: $\quad p^f_i = \alpha p_i + (1-\alpha)p'_i$
4: $\quad p^f_c = \alpha p_c + (1-\alpha)p'_c$
5: $\quad p^f_b = \alpha p_b + (1-\alpha)p'_b$
6: **for** N_h =0,1,2, … **do**
7:     **for** N_c =0,1,2, … **do**
8:         Calculate $\mathbf{p^n} = (p^n_i, p^n_c, p^n_b)$ for (N_h, N_c) pair with Markov model.
9:     **end for**
10: **end for**
11: Update (N_h, N_c) by finding $\mathbf{p^n}$= $\arg\min(\|\mathbf{p^f} - \mathbf{p^n}\|)$
12: Update packet error rate p as Equation (3)
13: Update packet delay d_i as Equation (4) and M/G/1 model
14: prepare data for the next adaptation period: $\mathbf{p^p} = \mathbf{p^f}$
15: **return** (d_i, p)

The pseudo code for the model adaptation algorithm is shown in Algorithm 1. In the current adaptation period, the weight parameter α is updated first as the value that gives the least error between the empirical probabilities $\mathbf{p'}$ and the final probabilities of the previous adaptation period (line 1). After that, we calculate the final probabilities $\mathbf{p^f}$ using Equation (8) (lines 3–5). Then we re-estimate N_c and N_h. First, we go through each possible N_c and N_h pair and calculate the slot idle, busy, and collision probabilities as $\mathbf{p^n}$ (lines 6–10). Then, after the Euclidean distance between the final probabilities $\mathbf{p^f}$ and the probabilities for each N_c and N_h pair, i.e., $\mathbf{p^n}$, is calculated, we choose N_c and N_h that yield the least distance to $\mathbf{p^f}$ (line 11). Finally, the delay d_i and the packet error rate p are re-estimated with the Markov model using Equations (2) and (4) and M/G/1 model (lines 12 and 13). Then the final probabilities $\mathbf{p^f}$ for the current adaptation period become obsolete and will be used as $\mathbf{p^p}$ in the next adaptation period (line 14).

With the above scheme, MC-Safe can then select the channel meeting the MATI and MAD requirements by combining the theoretical model and empirical statistics. To further reduce the computation overhead, we build an offline lookup table to store the theoretical values related to N_c and N_h. Each element in the table stores p_i, p_c, and p_b for one specific N_c and N_h pair. With the look-up table, the modeling process can be finished within 13 μs, which is smaller than the required time for generating Acknowledgment (ACK) frame.

In summary, the channel modeling component works as follows: based on the received beacons from other vehicles, each vehicle decides whether the sender of this beacon is a direct neighbor or hidden node on the service channel indicated in the beacon, and keeps monitoring every service channel continuously. After a certain period of time, based on the monitored network status and the theoretical value in the lookup table, the estimated p_b and the transmission failure probability p can be updated based on Algorithm (1) and Equations (2) and (3). After all probabilities (i.e., p_b and p) are known, the MAD requirement is checked by calculating the end-to-end delay with

Equation (4), and the MATI requirement is checked with Equations (5) and (7). Then the CPL is constructed, which indicates the quality of each channel for a vehicle on the road.

4.2.4 Channel Preference List. After we get both the estimated delay and the error rate for each channel, we can categorize it into one of three types:

- Type 1: This channel meets both the MATI and MAD requirements.
- Type 2: This channel does not meet the MATI and MAD requirements at the same time. However, by suppressing the transmission of other nodes within the one-hop distance, this channel may become a Type-1 channel.
- Type 3: This channel can meet the MATI or MAD requirements unless the two-hop neighbors are suppressed because of hidden terminals.

Note that building up CPL may incur considerable computation overheads. Thus, instead of building CPL when channel negotiation is needed, we maintain CPL periodically based on the beacons from other vehicles and the adaptation algorithm to incrementally adjust the CPL.

4.3 Channel Negotiation and Selection

Here we introduce our proposed approach for real-time channel negotiation and selection. We use the Constant Yawing Rate and Acceleration (CYRA) model [25] to calculate Time-To-Collision (TTC) to detect potential accidents, which is widely used to predict the vehicle's short-time trajectory. TTC is updated periodically with the received beacons from other vehicles, because it contains the position and speed information for the sender, and are updated at a minimum frequency of 10 Hz. When the TCC is below a certain threshold, MC-Safe on each vehicle is notified for a potential accident and start to form a temporary emergency avoidance group. Specifically, every vehicle keeps calculating the TTCs for other vehicles within a certain distance based on the vehicle speed, and builds a network topology for its nearby vehicles. Each nearby vehicle's location, speed, and TTCs to other nearby vehicles are recorded in this network, and updated with the received beacons from other vehicles. Once a vehicle detects a potential accident, it starts to form the group by initially including all the neighbor vehicles that have a TTC smaller than the threshold, broadcasting its group information in its beacon, and expanding the group iteratively by adding any missing vehicles that are in its neighbors' groups based on the beacons received from its neighbor vehicles. Hence, though it is indeed possible that different vehicles involved in the potential accident might initially form different groups, all the involved vehicles will rapidly converge to have the same group based on beacon exchanging. With a small size emergency group (usually less than four according to Reference [2]), the amount of time for the vehicles to converge can be considered negligible. Even in the rare case of continuous beacon losses, the vehicle can still join an existing group if it snoops negotiation packets sent by other vehicles that include this vehicle itself. When one vehicle snoops such information, it can join group and start the channel negotiation process with other vehicles in the same group.

After that, each vehicle in the group shall select one node in the group as the coordinator, and tries to send its CPL for channel negotiation distributively. The coordinator node then decides the channel to use (after negotiating with all the vehicles) and sends out the decision. When each vehicle receives this message, it switches its channel to the selected service channel indicated by the coordinator and terminates the channel negotiation process. Then, if the selected channel cannot meet the MAD/MATI requirements, the suppression scheme is applied to the selected channel to enforce these real-time requirement during the following DSRC communication phase. During the negotiation, we assume that the vehicle position does not change significantly, because the position changes for the vehicles during the negotiation process are relatively small given a short

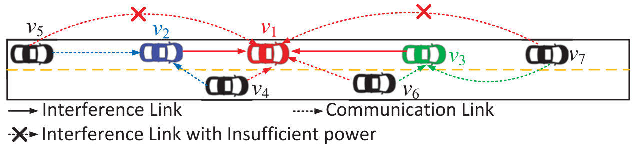


Fig. 5. An example of categorizing interference nodes among the cars v_1, v_2, and v_3.

negotiation duration (less than 1m for an average negotiation delay of 22 ms given a vehicle density of 8 vehicles/100 m at a speed of 40 mph). Moreover, vehicles in the group can know other vehicles' locations in the same group, either using the received beacon (calculating TTC), or the negotiation requests from other vehicles (which contain the group information and the position of another vehicle) to join an existing accident avoidance group. Generally, the channel selection consists of the following steps: (1) Coordinator selection, (2) channel negotiation, and (3) interference suppression.

4.3.1 Coordinator Selection. When the vehicles identify a potential danger, to know each vehicle's CPL and determine the channel quickly, MC-Safe selects an involved vehicle, in a distributed manner, as the coordinator, which is responsible for collecting CPLs from all the involved vehicles and making the final channel decision. To select the coordinator among all the vehicles in the potential accident, we propose an algorithm called *least interference algorithm*, which is designed to select the vehicle with the least interference as the coordinator. Generally, there are three major steps: (1) Calculating the interference score for each potential communication pair, (2) Finalizing the interference score, and (3) Deciding the coordinator. Among those three steps, step 1 is done periodically while step 2 and step 3 are done when a potential accident happens.

As shown in Figure 5, suppose that there is a potential accident among v_1, v_2, and v_3. From the view of v_1, for each pair of a transmitter (either v_2 or v_3) and receiver (v_1), the interference (other vehicles) can be categorized into two classes:

- Class 1: This node is a direct neighbor both to v_1 and the transmitter (either v_2 or v_3). When this node transmits a packet simultaneously when a negotiation packet is sent to v_1, the negotiation packet will be lost and must be retransmitted. Note that, due to the backoff mechanism, the packet loss will occur only when the negotiation packet and the interference packet are sent exactly at the same time. Otherwise, one of them will back off and try again. As shown in Figure 5, v_4 is a direct neighbor to both v_1 and v_2 within a certain range, so v_1 will list v_4 as a Class 1 interference to the v_1–v_2 pair; with the same logic, v_6 is a Class 1 interference to v_1–v_3 pair. However, v_5 is not a Class 1 neighbor to the v_1–v_2 pair, because it is too far away from v_1: Its packets do not have sufficient power to interfere with the negotiation packets even though the packet collision happens.
- Class 2: Hidden Terminal. This node is out of the communication range of the transmitter but within a certain range of the receiver and affects only the receiver. Thus, these nodes are within the two-hop distance from the sender, and one-hop distance from the receiver. However, if a node is more than two-hop away from the sender (i.e., one-hop away from the receiver), then the packet it sends shall not have sufficient power to directly interfere with the packet sent by the sender. Though Class 2 interference does not increase the backoff delay, neither shall it back off its transmission when the negotiation packet is being transmitted, because it cannot sense the transmission from the transmitter. Thus, this type of interference node can transmit the packet at any time and cause packet loss, while Class 1 nodes can only cause the packet loss when the interference node and the transmitter both sense the wireless media free and begin to transmit packets at the same time. Thus, Class 2

interference nodes cause more severe packet loss than Class 1 nodes if they exist. Due to the retransmission mechanism of 802.11p, the impact of more retransmission is larger than one more backoff waiting period. As a result, Class 2 interference nodes cause longer negotiation delay than Class 1 interference nodes. In Figure 5, v_4 is a Class 2 interference to v_1–v_3 pair as it can be sensed by the receiver (v_1) only; similarly, v_6 is a Class 2 interference to v_1–v_2 pair but a Class 1 interference to v_1–v_3 pair, because it can be sensed by v_3 but cannot be sensed by v_2.

We mainly consider the interference that can affect the receiver, because the coordinator in MC-Safe mainly serves as the receiver in the negotiation process. Moreover, the interference node that does not affect the receiver only increases the backoff delay, which is small compared to the delay caused by the packet retransmission (mainly due to the interference from the receiver side).

For each communication pair that involves v_1, v_1 performs the above categorization using periodic beacons from others, the propagation model for 5.9-GHz communication, and the Signal to Interference and Noise Ratio (SINR) requirement for the modulation scheme. To combine the two classes of interference, an interference score S for each communication pair is calculated as:

$$S = aI_1 + bI_2, \tag{9}$$

where I_1 and I_2 represent the number of Class 1 and Class 2 interference nodes, respectively, and a and b are weights for Class 1 and Class 2 interference nodes, respectively. a is chosen as the probability of more than one node transmitting together at the same time, which can be derived by transmission probability τ and N_c (derived from the channel modeling component), i.e., $a = 1 - (1 - \tau)^{N_c - 1}$. b should be much larger than a, because the impact of hidden terminals is much stronger [32].

Since the coordinator needs all the CPLs from other involved vehicles to determine the final channel to use, the negotiation delay should be determined by the largest end-to-end delay among all the transmission pairs of the involved vehicles, which includes queuing delay, CSMA backoff delay and packet transmission delay. Thus, after v_1 acquires interference score S for all of its communication pairs, v_1 uses the larger S score between the v_1–v_2 pair and the v_1–v_3 pair to represent the interference level for v_1 upon the potential accident, as this S will be the worst case in receiving the negotiation packet from the other transmitters, if v_1 itself is chosen as the coordinator.

After knowing the S score for itself, v_1 still needs to know the S scores for v_2 and v_3 to finalize its coordinator selection. To avoid calculating the S score for every other involved vehicle in the potential accident (v_2 and v_3), every vehicle on the road includes S scores for all of its potential communication pairs within beacon messages and broadcasts them periodically, no matter whether a potential accident happens or not. Note that in an emergency scenario, the distances between the vehicles are usually small compared with the direct communication range (250 m), i.e., one hop communication distance of the 802.11p protocol with typical transmission power (20 dbm). As a result, we can assume that the vehicles in the same emergency avoidance group can receive others' beacon signals, because they are all involved in the same potential accident within a short range (one-hop distance). Hence, in Figure 5, the beacon from v_1 does not only contain S scores for the involved communication pairs (i.e., v_1–v_2 pair, v_1–v_3 pair) but also contains the other pairs (e.g., v_1–v_4 pair). Once the potential accident happens, v_1 can directly know the S scores of communication pairs for other involved vehicles (v_2 and v_3) and decides the coordinator choice for itself quickly with beacon exchanging. The coordinator selection is done individually by every vehicle in the group, so v_2 and v_3 also select the coordinator, which could be a different choice than v_1. However, in most cases, the correct S scores for these vehicles can be received and known by all the vehicles in the group with beacon exchange on the CCC. With a minimum transmitting frequency of 10 Hz, the S score can be updated in time, even when some beacons are lost occasionally. Thus, all the vehicles in the same group can unanimously select the right coordinator most of the time.

However, due to the heavy contention and strong interference, in the case of continuous beacon losses, there could exist multiple coordinators in one potential accident, due to the lack of some S scores from others. There are two methods to solve this problem: First, every possible coordinator snoops the negotiation requests from other vehicles: When one coordinator candidate realizes that the destinations of any negotiation packets are not itself (i.e., another potential coordinator) and the MAC address for the destination is smaller than that of itself, it stops being the coordinator and starts to send its own negotiation request packet to the coordinator indicated in that snooped negotiation request packet. Moreover, this candidate also sends all the received CPLs (from the vehicles that consider it as the coordinator) to the coordinator candidates with a smaller MAC address (indicated in the snooped negotiation request packet from other vehicles). Second, each self-selected coordinator (e.g., the vehicle decides itself as the coordinator) can broadcast this information in the beacon, and let other coordinators decide whether to stop being the coordinator based on the MAC address. Thus, even if there is no negotiation packet to be snooped or lost, the coordinator is still unique in the emergency avoidance group. In this way, though all the vehicles can have different choices of the coordinators, all the required CPLs can be received by the coordinator with the minimum MAC address at the end of the negotiation process, because other coordinators will stop being the coordinator when they are notified. Even though the final coordinator may not be the one with the minimum S in this situation, we still need to avoid having multiple coordinators in the same group. Otherwise, they would be waiting for the CPLs for each other and form a deadlock situation in the following channel negotiation.

4.3.2 Channel Negotiation. After the coordinator is selected, each vehicle in the group starts to send the channel negotiation message to the selected coordinator. The coordinator waits for all the CPLs in the group for a certain period, and tries to find a SCH to use for vehicle dynamic control. The channel negotiation process is similar to the typical rendezvous problem in cognitive radio system [18], and works as follows: First, the vehicles that are not selected as the coordinator start the negotiation process by sending the negotiation request and its CPL to the coordinator. To reduce the queuing delay of the CCC, we give the negotiation packet the highest priority by placing it in the head of the MAC queue. To reduce the service delay, we limit the CW_{max} to be equal to CW_0: The backoff window remains a constant in the backoff procedure. Further decreasing CW_0 yields more retransmissions, making the packet delay longer than using CW_0 directly. Moreover, to make sure the negotiation packet can be received correctly, we also cancel the limit on maximum times allowed of re-transmission for the negotiation packet. The vehicle shall try sending the negotiation packet to the coordinator unless it receives the final channel decision due to a timeout. Note that these manipulation are conducted only for negotiation packets, because they are sent out *only once*. This cannot be periodically used for safety messages to meet MATI/MAD requirements, because it would cause severe congestion to other cars that are not involved in the accident.

Besides the CPL, each vehicle uses the CYRA model to calculate the remaining time T_{ava} for every Type-1 channel to meet the MATI/MAD requirements in spite of vehicle mobility. For example, if one channel is occupied by a cluster of vehicles near the involved vehicles, T_{ava} for that channel is smaller than those of other channels, because those nearby vehicles cause more interference. we only calculate T_{ava} for these channels that can meet the MATI/MAD requirements and set T_{ava} for others to 0. T_{ava} is transmitted along with the CPL as a reference for channel selection.

The coordinator needs all the CPLs from other vehicles in the potential accidents to decide the channel to switch to. Although there is no limit on maximum allowable times of re-transmissions for the negotiation packet, the coordinator cannot wait tool long in the emergency scenario. Thus, after the coordinator receives the first negotiation packet, it sets up a timer and waits for other negotiation packets in the same group. If all the CPLs are received in time, then it starts the channel selection process. Otherwise, when the timer fires, it will also start the channel selection with only the received CPLs. Specifically, the channel selection process contains several list matching procedures. The final channel to use is chosen according to the rules below:

- If available Type-1 channels exist, then the channel with the largest T_{ava} is chosen; if these channels have the same T_{ava}, then the one with the least N_c is selected;
- If there is no common Type-1 channel, then a Type-2 channel that requires the least suppression effort is selected.
- If there is no common Type-2 channel either, then a Type-3 channel with the least hidden terminals is chosen.

To quickly settle the channel to use and avoid unnecessary negotiation delay, for MC-Safe we propose to adopt the *max-min* policy to compare T_{ava}, which means that T_{ava} is determined by the smallest value in the CPLs received by coordinator. For example, T_{ava} is 3.2 for Channel-1 and 3.0 for Channel-2 in the CPL of vehicle C. Then, vehicle C (the coordinator) receives CPL from vehicle A indicating T_{ava} is 4.0 for Channel-1 and 2.4 for Channel-2. Vehicle C also receives vehicle B's CPL indicating T_{ava} is 1.2 for Channel-1 and 3.1 for Channel-2, vehicle C will set T_{ava} for Channel-1 to be 1.2 and Channel-2 to be 2.4. If Channel-1 and Channel-2 are both Type-1 channels, then Channel-2 is selected for vehicles A, B, and C to communicate on. The coordinator will make the decision and transmit it in the ACK frame to all other vehicles involved in this potential accident, and change its own channel. After the ACK is received by the other nodes, they will also change to the chosen channel indicated the ACK frame accordingly. Thus, all n involved vehicles select the same channel. Though multiple rounds of negotiation can make the channel selection result more robust, the extra negotiation delay can be intolerable given a critical condition on the road. Thus, we use the above proposed one-time handshake format to settle the negotiation. Once the coordinator selection and channel negotiation are finished, the involved vehicle shall change to the desired channel for DSRC communication and begin to transmit emergency messages for vehicle control applications. Thus, the delay estimated in the channel modeling process does not include the delay introduced by negotiation and channel selection.

MC-Safe aims to establish communication under an emergency situation, so the negotiation delay should be as small as possible. The negotiation delay can be calculated as follows: Because the negotiation packet is put in the head of the MAC queue and the backoff counter does not increase due to retransmission, the delay for transmitting the negotiation packet (for one time) consists of only the service delay s_0. After the negotiation packet is received by the other side, the ACK will be ready after the period of time defined as Short InterFrame Space (SIFS) [15] and not go through the backoff procedure again. After the ACK frame is received by the transmitter, the negotiation process is finished, and the involved vehicles switch their channel and start to communicate with each other. According to our proposed model, the negotiation delay d_n can be calculated as:

$$d_n = \frac{s_0}{1-p} + \text{SIFS} + T_{ACK} + T_s, \tag{10}$$

where SIFS is the waiting time to generate ACK frame, T_{ACK} is the transmission time for the ACK frame, and T_s is the channel switch overhead. The average T_s is only 2.87 ms, which is negligibly short compared with the negotiation delay, according to our evaluation using the Intel 3945 Network Interface Card [13]. For a medium vehicle density (8 vehicles/100 m) and a typical vehicle speed (40 mph) on the road, the average negotiation delay is 22 ms and the traveling distance of a vehicle during the negotiation is less than 1 m. For heavy congestion cases, the negotiation can

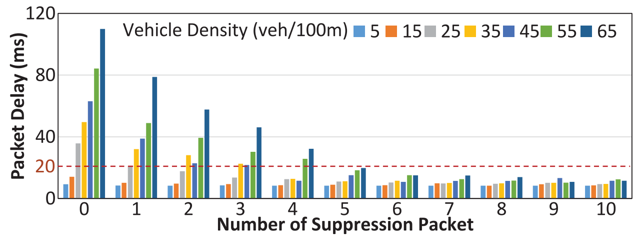


Fig. 6. Delay during DSRC communication phase with different numbers of suppression packets and direct neighbors. Deadline (20 ms) has been marked as the red line.

be longer than 100 ms, but car speeds normally are reduced with a higher vehicle density, so the traveling distance during negotiation is still limited. Thus, the negotiation delay for MC-Safe is sufficiently small. Therefore, though MC-Safe incurs extra delay in the channel negotiation and selection compared with using the CCC only, the sum of end-to-end packet delay and the negotiation overhead is usually smaller than the end-to-end delay on CCC. In our evaluation part, we test the emergency braking scenario, where the following vehicle starts to decelerate as long as it receives the first packet from the leading vehicle. Our results show that even with extra negotiation overhead, MC-Safe still outperforms using the CCC only by 38.2% (see Section 5.8). Moreover, we also test the total delay for channel negotiation involving more than two vehicles. The total delay for three, four, and five vehicles can be 38 ms, 46 ms and 54 ms with a vehicle density of eight vehicles/100 m, respectively. The traveling distance for each vehicle is still smaller than 1 m.

4.3.3 Interference Suppression. If there is no common Type-1 channel available in the channel selection phase, then the suppression mechanism is applied to meet the MATI and MAD requirements after the vehicle switches to a Type-2 channel. Specifically, when a Type-2 channel is selected for suppression, several packets during the DSRC communication will be heard by other vehicles to suppress those vehicles that are not involved in the potential accident and sending non-safety messages on the same channel. The suppression scheme is implemented by adopting a p-persist mechanism: For each transmitting packet, it has a probability of p_t to be transmitted; otherwise, it goes back to the backoff procedure. There are two important parameters to be selected: The number of suppression packets and the transmission probability p_t.

The number of suppression packets must be chosen carefully, because too few suppression packets may not be able to accomplish the desired suppression. However, sending too many suppression packets unnecessarily causes delay to real-time V2V safety message exchange and collision prevention; the value of p_t is directly related to meeting the MATI/MAD requirements during the DSRC phase. For the number of suppression packets, we conduct an experiment to show the suppression performance with different vehicle densities and suppression packets (with the deadline requirement being 20 ms) to determine the optimal number of suppression packets. Figure 6 shows that for most cases (13 vehicles/100 m is a high vehicle density considering a car's length), five suppression packets are sufficient to suppress other nodes under the required deadline.

For the transmission probability p_t, we transform the original problem (i.e., finding p_t) into another problem. As the packet only has the probability of p_t to be transmitted, it is equivalent to increasing the original contention window size CW_0 by $1/p_t$ times. Thus the new contention window size $CW_0^{'}$ can be calculated as:

$$CW_0^{'} = \frac{CW_0}{p_t}. \tag{11}$$

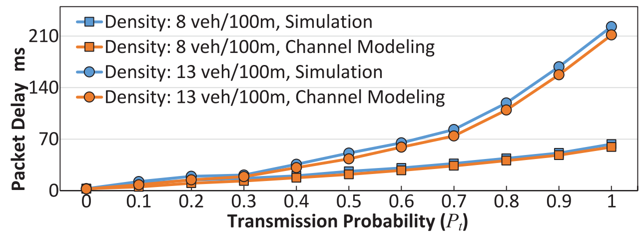


Fig. 7. The packet delay of theoretical values and simulation results with transmission probability p_t.

Equation (11) indicates that we can derive p_t by calculating $CW_0^{'}$, as $CW_0^{'}$ is directly used as a parameter for the Markov model: Given the packet delay d_i and packet error rate p requirements specified by the vehicle control application, $CW_0^{'}$ can be derived by the equation set composed of Equations (1), (4), and (6). Then p_t can be calculated with Equation (11). To verify our hypothesis, we run several rounds of simulations, with the same setup as the motivation test. Figure 7 shows the simulation results and theoretical values with transmission probability p_t. The simulation results correspond well to the theoretical values, which proves the correctness of our analysis.

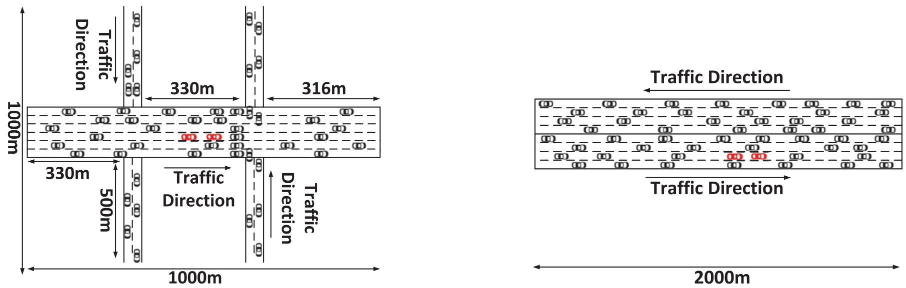
If there are no common Type-1 nor Type-2 channels, then a Type-3 channel with the least number of hidden terminal is chosen. In this case, the channel cannot satisfy MATI/MAD by only suppressing the nodes within one-hop distance. Thus, hidden terminals need to be suppressed. However, the worst situation is rare, and usually there is at least one common Type-1 or Type-2 channel.

4.4 Discussion

4.4.1 Overhead Analysis. Here we analyze the time and transmission overheads of MC-Safe. The time overheads mainly come from two components: (1) Channel Modeling and (2) Negotiation. For channel modeling, as the theoretical parameters (p_b, p_c, p_i) are estimated offline with the lookup table, the time overhead of channel modeling comes from online monitoring, which includes estimating p and d_i for each channel and checking MATI/MAD requirements. The overhead for those two parts is 8 and 5 μs based on the measurements on our hardware testbed, respectively. For negotiation, the complexity of the algorithm is $O(n\log m)$ (m is the number of involved vehicles and n is the number of available service channels). Given a limited number of service channels (six in WAVE) and involved vehicles (e.g., three vehicles), the measured computation overhead is small (6.43 μs) compared with other terms in Equation (10) and so negligible.

For transmission delay, the extra bits for MC-Safe in the beacon include the channel usage information for channel modeling (i.e., which SCH the vehicle is using) and the S scores for the direct links used in the coordinator selection step. The channel usage information can be represented with the channel number defined in WAVE from 172 to 184 with one byte [1]. Each S score for the direct link could take up to six bytes (4 for the MAC address and 2 for the S score), which could take much space in the beacon frame if all the S scores are included in the beacon and transmitted on the air. In MC-Safe, to significantly reduce space, only S scores of the vehicle pairs within a certain distance shall be selected for transmission in the beacon. For example, for a vehicle density of 10 vehicles/100 m and a distance threshold of 100 m, only 10 S scores shall be transmitted in the beacons and take up 120 bytes. Even with the lowest transmitting rate (3 Mb/s), the extra bytes incur 0.16 ms more delay and thus can be negligible.

4.4.2 Reliability of Negotiation Messages. The message reliability of MC-Safe can be ensured by its MAC protocol, 802.11p: The negotiation request will be retransmitted if the sender does not



(a) Intersection (Int) (b) Highway

Fig. 8. The road topology of our two test scenarios. The vehicles of interest are marked in red.

receive an ACK from the coordinator within a specified time interval until it reaches the retry limit. In the rare cases when there are indeed packet losses (after retry limit is violated), MC-Safe could select a channel that is not the best. However, MC-Safe just becomes Random in this case, whose performance is still much better than using CCC directly (see Section 5 for results). To ensure that all the vehicles involved select the same channel for negotiation, the coordinator must receive ACK from every other vehicle (regardless of the retry limit) after it sends out the channel decision.

5 EVALUATION

In this section, we conduct the evaluation of MC-Safe. We first introduce the experiment setup. We then test MC-Safe and other baselines in different scenarios in simulation (Sections 5.2 to 5.9). At last, we test the performance of MC-Safe on a hardware testbed (Section 5.10).

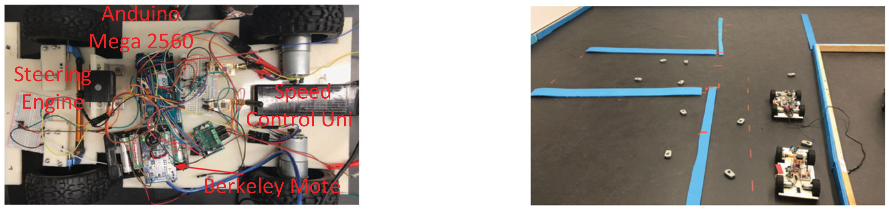
5.1 Experiment Setup

5.1.1 Road and Vehicle Setup. We test MC-Safe in two major road scenarios. The first one is a typical traffic scenario in the urban area with one main road and two branches: The main road is one-way with six lanes; the branches are one-way two-lane roads. The width of each lane is 4 m. The second one is a bi-directional highway scenario of eight lanes with a total length of 2,000 m. Upon arriving at the end of one direction, vehicles re-enter the scenario at the start point of the other direction, and thus the total number in the simulation remains constant. Figure 8 shows the two traffic scenarios and the cars of interest: The heading car suddenly stops and the following car starts the channel selection process when the following car receives the hard-brake message.

We formulate the inter-vehicle distance in Poisson distribution as a typical traffic modeling method [4]. To minimize the variance brought by random factors from the traffic distribution, we run 20 independent and identically tests to get the average results.

5.1.2 DSRC Setup. The DSRC radio on each car broadcasts safety information on the CCC periodically and the frequency is 10 Hz as recommended. The size of safety message is set to be 300 bytes [28]. The transmission power is set to a recommended value (20 dbm) [15]. Among all available data rates defined in 802.11p protocol (*3* to *27 Mbps*) [15], we choose the lowest 3 Mbps to test the worst-case scenario in terms of delay performance. The traffic on SCHs follow the Poisson distribution with an arrival rate ranging from 2 to 30 ms to emulate different kinds of applications.

5.1.3 Hardware Testbed Setup. We implement a prototype of MC-Safe using small-scaled cars. As shown in Figure 9(a), the control algorithm is implemented with the Arduino Mega 2560 broad, which connects to the steering engine and speed control actuator to realize the driving control



(a) The scaled car (b) Road Setup

Fig. 9. Hardware testbed of MC-Safe.

using a typical PID controller. Although MC-Safe should be implemented on real DSRC devices for V2V communication, as a proof of concept system, we use Tmote Sky motes as the communication device and implement MC-Safe using TinyOS. The initial speed of the two scaled cars are set to 150 cm/s (60 mph for a real car). Figure 9(b) shows the road setup for our experiment. We create one two-lane road and deploy other Tmote Sky motes randomly as interference nodes.

5.1.4 *Baselines for Comparison with MC-Safe.*

- **Common Control Channel (CCC):** Similarly to the state-of-the-practice solution WAVE, CCC relies only on the control channel to transmit safety messages.
- **EDCA:** EDCA is the currently used in 802.11p to enhance the QoS of different applications by assigning different priorities to them [9]. With EDCA, packets from a high-priority application has a higher chance of being sent than those from a low-priority application, due to a shorter carrier sensing window and a smaller backoff counter. The priority setting for the emergency packets and the beacons are the same as show in the motivation test. Although better than CCC, EDCA is still a single channel solution that could fail given a high vehicle density.
- **Random:** When a pre-crashing condition is detected, each vehicle randomly selects one service channels other than the CCC, and send its choice to the coordinator. the coordinator finalizes the channel decision as the majority choice in the negotiation process afterward.
- **The Least Congested Channel First (LCCF):** It chooses the channel that has the least vehicles within the one-hop distance. Thus, it may have degraded performance when there are more hidden terminals on the same channel.
- **Ideal:** One channel is reserved and always available *only* for the vehicles involved in this particular potential accident. However, in a real V2V network, other vehicles may also use this channel when they are likely to have collision. Hence, this method is unrealistic, but just serving as a baseline with the performance upper bound.

5.2 Analysis of a Typical Scenario

Here, we investigate the whole channel selection process from the time point when the danger is identified by MC-Safe to the time point when the danger is resolved. We set the vehicle density to be seven vehicles/100 m. Vehicles of interest are selected in the middle of the road segment to avoid the inaccuracy of the two ends in the road segment (marked in red in Figure 8).

Figure 10 shows the full traces of message transmission delay of the four methods (Random, MC-Safe, LCCF, and EDCA) during the whole process. Except EDCA, each trace contains three phases: The channel negotiation phase, the DSRC communication phase, and the back to CCC phase. The packet delay at first is large, because the vehicle is on CCC whose condition has large variations. When the following vehicle receives the negotiation request from the leading one due

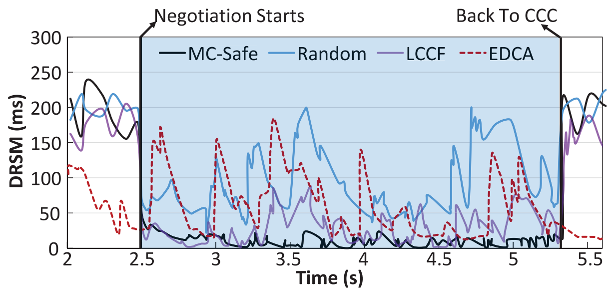


Fig. 10. The Delay of Received Safety Message (DRSM) during the whole channel selection process, where the channel switch happens at 2.5 s from the CCC to the selected channel, and back to the CCC at 5.4 s.

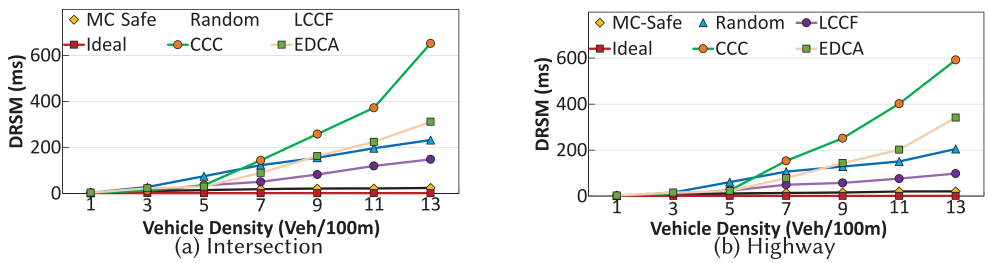


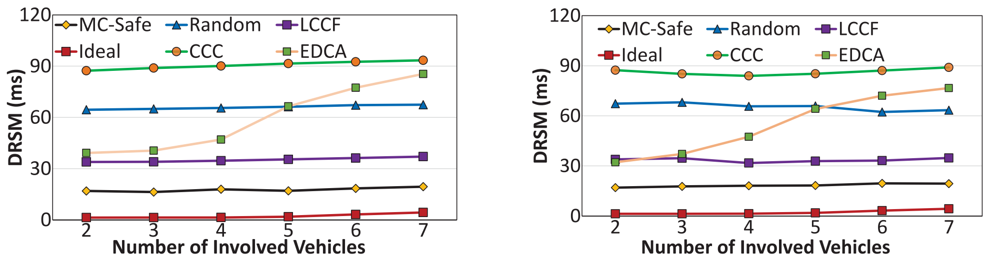
Fig. 11. DRSM of different methods with different vehicle densities.

to a smaller TTC than the threshold, it starts to negotiate for a common channel to use (at around 2.5 s). When the negotiation starts, the delay drops from 200 to 50–75 ms, because the negotiation packet is placed at the head of the MAC queue and its backoff counter does not double for retransmissions. After the ACK frame is received, the two vehicles can communicate with each other with a negligible delay (10–15 ms). After the danger is resolved, the two vehicles switch back to CCC and start to send beacon on CCC again. Figure 10 shows that MC-Safe can achieve the best performance compared to the baselines. Random has an average delay of 120 ms during DSRC communication, which is much larger than the 20-ms requirement. The delay performance is also not stable: The largest delay can be around 200 ms, which will result in a large blackout period and degrade the performance of control algorithm. LCCF has an average delay of 40ms, which is much smaller than that of Random but it still cannot meet the typical control requirement (20 ms). LCCF shows some spikes in the DSRC communication phase, indicating that LCCF has large variations and suffers from unexpected high delay at some time points, because LCCF does not consider the hidden terminal issue and suffers a lot of retransmissions in the MAC layer. EDCA is a single channel solution so it does not has the three phases, but its packet delay is still longer than that of LCCF and also varies largely due to high vehicle density in the experiment.

5.3 Different Vehicle Densities

In this set of experiments, we compare MC-Safe with the baselines (Random, LCCF, Ideal, CCC, and EDCA) under different vehicle densities. We test two typical scenarios (shown in Figure 8) and assign different speeds to the vehicles. Figure 11 shows PDR and the Delay of Received Safety Message (DRSM) for the intersection and highway, respectively.

MC-Safe outperforms LCCF 23.4% in terms of DRSM. CCC is unusable when the vehicle density reaches to seven vehicles per 100 m, which is common in the urban area, especially during rush hour. The delay for a single packet can be as long as 160 ms, which is 8 times the CAN deadline.



(a) Intersection (b) Highway

Fig. 12. Comparison of different numbers of involved vehicles.

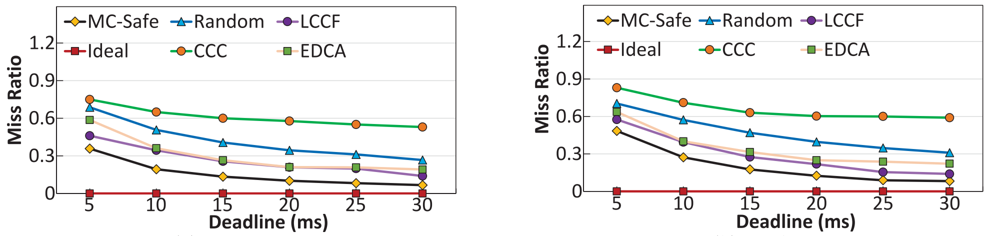
The delay of Random increases rapidly (from 15.91 ms to 61.32 ms) as the vehicle density increases from three to five vehicles per 100 m. Even with EDCA, a single channel solution still performs worse than Random given a high vehicle density (nine vehicles/100 m). LCCF has a similar delay to MC-Safe, but it only considers the one-hop neighbors. Therefore, as the vehicle density increases, its DRSM begins to increase due to the increase of its two-hop hidden terminal neighbors, which is not considered in its design. MC-Safe maintains a small DRSM for all the vehicle densities, with the closest performance to that of Ideal. For a higher vehicle density where there is no suitable channel for real-time communication (nine vehicles/100 m and above), because the packet delay cannot meet the MAD requirement, MC-Safe can still achieve an average delay below 20 ms while LCCF has an average delay of more than 50 ms.

5.4 Different Number of Involved Vehicles

Here, we evaluate the performance of MC-Safe under different numbers of involved vehicles. Figure 12 shows the DRSM of MC-Safe and the baselines with a fixed vehicle density (seven vehicles/100 m). We can see that the DRSM of MC-Safe stays under 20 ms for up to seven involved vehicles, which is 50.3% and 78.1% less than that of LCCF and Random, respectively. Compared with the single channel solution EDCA, MC-Safe outperforms it by 66.5% in terms of DRSM. Unlike other schemes, the DRSM of EDCA has a much larger increase with the increase of involved vehicles. The lower improvement of EDCA in this case is mainly due to the fact that not only the number of transmitters increases, but also the contention in the same priority increases. Moreover, the DRSM for MC-Safe increases only 6.43 ms from the scenario that involves only two vehicles to that of up to seven vehicles, while the DRSM of EDCA increases from 82 to 133 ms when the involved vehicle increases to seven. This is because MC-Safe can either select a channel that meets the MATI/MAD requirements or uses suppression to make a channel meet those requirements even if there exists no channel suitable for the collision avoidance control algorithm. However, MC-Safe has slightly more increase in the communication delay compared with the baselines. This is because the interference is already strong on the CCC and the channel selected by Random, even for two vehicles, thus the space for interference increase is small. Nevertheless, MC-Safe still keeps the average DRSM lower than the real-time requirements (20 ms).

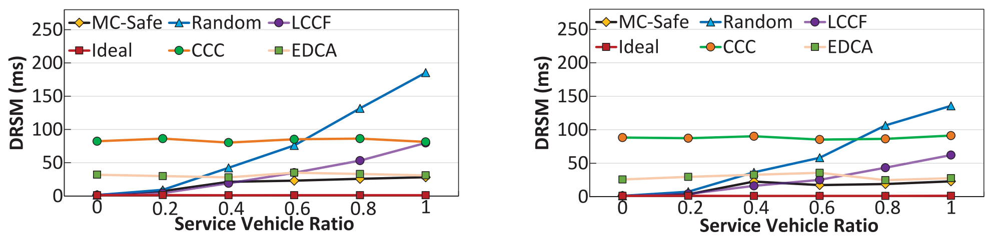
5.5 Different Deadlines, and Service Vehicle Ratios

Here, we examine the impacts of two key design factors: deadline and service vehicle ratio. First, we consider the impact of different deadline values. Figure 13 shows the deadline miss ratio when the deadline changes from 5 to 30 ms. Even with EDCA, MC-Safe still outperforms the single channel solution by 37.1% on average. For multi-channel solutions, MC-Safe is 41.3% lower than that of Random on average and is 58.7% better under the tightest deadline (5 ms). We can also



(a) Intersection (b) Highway

Fig. 13. Deadline miss ratio under different deadline values.



(a) Intersection (b) Highway

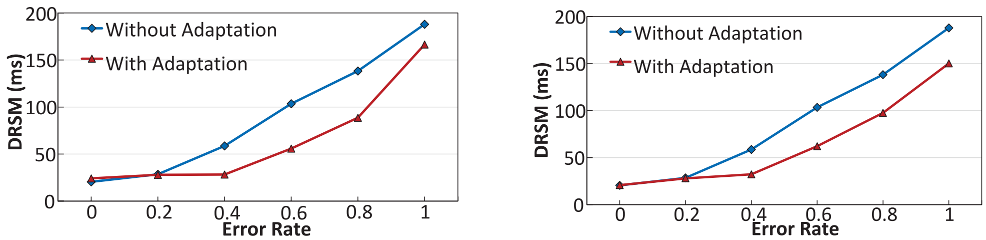
Fig. 14. DRSM under different service vehicle ratios.

see that MC-Safe outperforms LCCF by 12.3% on average, 10.5% for the relatively loose ones, and 21.7% under the tight ones. The main reasons are twofold: First, LCCF does not consider the hidden terminal. Thus, for one packet it has to re-transmit many times, causing a high variance on the packet delay. Second, it does not consider the control requirements, thus lacking of a well-designed suppression mechanisms as well.

We then test different service vehicle ratios. A higher service vehicle ratio means more vehicles are transmitting on SCHs other than the CCC (i.e., service channels), thus having a larger chance to have contention after the vehicles switch to their selected channel. Figure 14 shows that the average packet delay of MC-Safe degrades slightly when the service vehicle ratio increases from 0 to 1, for both the intersection and highway scenarios. In contrast, the delay of Random increases from 1ms to more than 134 ms. LCCF's the delay becomes larger than 20 ms when the service vehicle ratio reaches 0.4 as the number of hidden terminals increases. The difference between MC-Safe and Random/LCCF becomes larger when the service vehicle ratio increases, mainly because MC-Safe is designed to consider the hidden terminals and has a suppression mechanism. Generally, MC-Safe outperforms Random and LCCF by 35.2% and 15.8% on average, respectively.

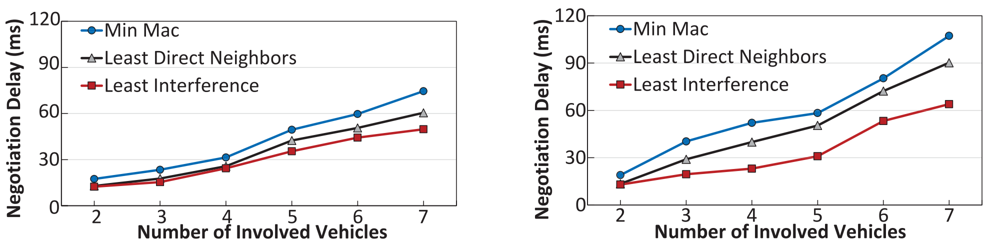
5.6 Effectiveness of Model Adaptation

Here we evaluate our model parameter adaptation algorithm. To test the adaptation performance, we deliberately alter the beacon content of the currently used channel of some vehicles, thus inserting error to the model parameters (i.e., N_c and N_h). The reported number of one-hop neighbors (N_c) for each service channel will be inaccurate. As shown in Figure 15, the error rate ranges from 0 (totally accurate) to 1 (totally wrong) when vehicles send the beacon packet. These errors will have two consequences: First, due to the inserted error, MC-Safe will not select the right channel to use, leading to lower PDR and longer delay, and, second, the error will also affect the suppression algorithm, because the probability of virtual collision depends on the reported N_c. For MC-Safe, it uses the adaption algorithm to solve these issues by adaptively monitoring the channels, and



(a) Intersection (b) Highway

Fig. 15. Comparison of DRSM with and without adaptation algorithm.



(a) Intersection (b) Highway

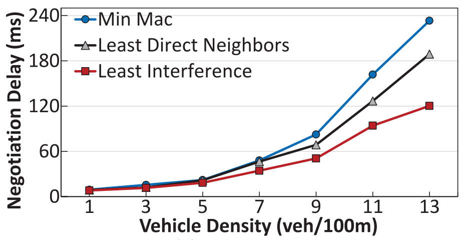
Fig. 16. Negotiation delay of different number of involved vehicles with fixed vehicle density (eight vehicles/100 m).

re-estimates the number of nodes on the channel in the 1-s time window. Figure 15 shows the improvements due to the adaptation algorithm: The average delay decreases from 58 to 36 ms when the error is 40%, and it is also improved from 103 to 62 ms under 60% error. However, when the error is above 80%, the improvement gets smaller as the error is too large to be corrected.

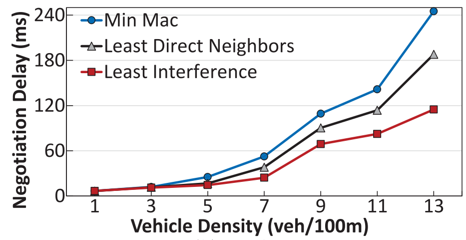
5.7 Different Coordinator Selection Algorithms

Here we evaluate the performance of least interference algorithm proposed in Section 4.3.1 by comparing it with two baselines: (1) MinMAC, which is designed to select the vehicle with the smallest MAC address as the coordinator; (2) Least Direct Neighbor, which selects the vehicle with the least number of direct neighbor (N_c). Then we test the performance of the coordinator selection algorithm with different vehicle densities and the different number of involved vehicles.

Figure 16 shows that the negotiation delay for the least interference algorithm is the smallest among all the schemes: 25.3% and 18.1% lower than that of MinMAC and least direct neighbors, respectively. Though MinMAC is the easiest scheme to implement and it can avoid having multiple coordinators, its negotiation delay increases rapidly when the number of involved vehicles is larger than five. If there exists a potential chain accident on the highway, then MinMAC would incur long negotiation delay and fail to prevent the accident from happening. However, least direct neighbor algorithm fails to consider the interference strength of each direct neighbor. Thus, its negotiation delay is longer than that of the least interference algorithm. Figure 17 shows the negotiation delay of each coordinator selection scheme for different vehicle densities, with a fixed number of involved vehicles. MinMAC can incur a delay of more than 200 ms when the vehicle density is high, which could result in a collision during the negotiation as the distance between vehicles is small. The negotiation delay for least interference algorithm is about 120 ms, which is 50.6% shorter than that of MinMAC and 32.1% lower than that of least direct neighbor. With



(a) Intersection



(b) Highway

Fig. 17. Negotiation delay with different vehicle densities.

Table 2. Comparison of Collision Probability

Inter-vehicle Distance	MC-Safe	LCCF	Random	EDCA	CCC
close (25 m)	18%	42%	60%	45%	78%
Relatively close (30 m)	11%	25%	38%	35%	50%
Relatively safe (40 m)	0%	0%	6%	1%	15%

MC-Safe achieves the lowest collision probability.

this improvement, the collision probability can be further reduced, because the distance traveled during the negotiation is shorten by about three meters for a typical vehicle speed (40 mph).

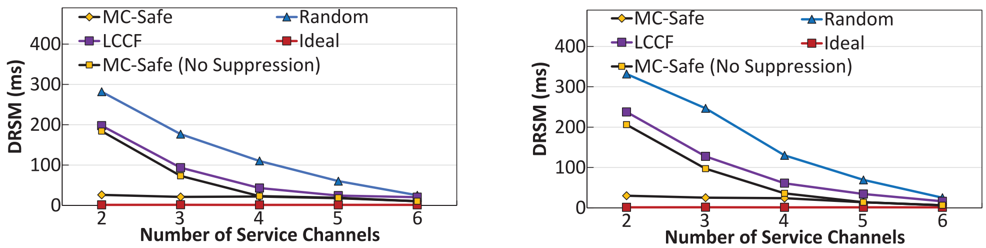
5.8 Comparison of Accident Probability

In this section, we compare the accident probability of MC-Safe with other baselines in a real-life rear-end collision scenario, to see how MC-Safe can help in a pre-crashing scenario. We set the vehicle density as seven vehicles/100 m and their speeds as 40 mph. When the heading vehicle takes the hard braking action, the following vehicle will change the channel to communicate with the heading one. After the first safety message is received by the following vehicle, it will brake at the largest deceleration allowed. Here we use the typical three-phase braking model [14] to calculate the braking distance based on how soon the safety message is received (i.e., message delay). With different vehicle distances, we list the calculated accident probabilities in Table 2. Even though there is no negotiation delay in CCC, it still performs the worst compared with other solutions, because the contention of the CCC is so large that the first packet cannot be received in a timely manner, even with only communication delay. Results show that in the real case, MC-Safe can achieve the lowest collision probability due to more timely (i.e., shorter delay) and reliable (i.e., higher PDR) safety message communication.

5.9 Number of Service Channels

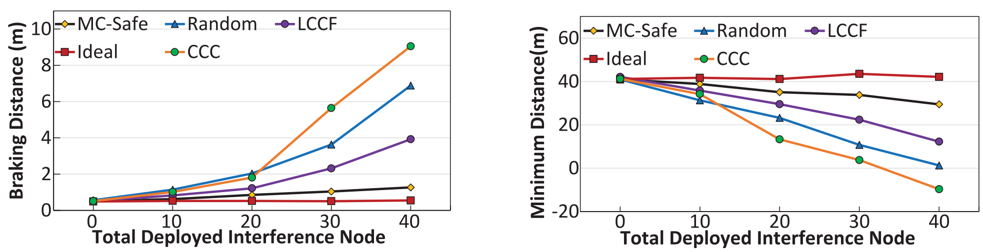
Here we test MC-Safe with different numbers of available SCHs. We choose the number of SCHs from two (at least two channels to select from) to six (which is the maximum allowable channels other than the CCC defined in WAVE 1609.4), and fix the vehicle density as seven vehicles/100 m. We only compare MC-Safe to the multi-channel baselines (i.e., Random, LCCF), as the performance of the single channel solutions shall not be affected (i.e., CCC, EDCA).

Figure 18 shows the DSRM of MC-Safe compared with the baselines and MC-Safe without suppression. We can see that with the increase of available SCHs, the delays of all the schemes decrease accordingly. This is because with more available SCHs, the contention (i.e., the number of other vehicles using the SCH) on each channel decreases with a fixed vehicle density. When the number of SCH is less than four (the number of non-safety channels specified in the WAVE 1609.4 standard), though MC-Safe cannot find a channel that can meet the deadline requirement, it can still



(a) Intersection (b) Highway

Fig. 18. Comparison of MC-Safe with other baselines when the number of Service Channels (SCH) is changed.



(a) (b)

Fig. 19. Comparison of control performance with MC-Safe. (a) Stopping distance for different baselines in the hard-braking case. (b) The minimum distance between the two scaled cars in the lane changing case.

achieve the required packet delay by suppressing other interference nodes on the road. Moreover, even without suppression, MC-Safe still has a 23.1% lower packet delay than LCCF, because of its theoretical model of the channel status and adaptation algorithm. When the number of available SCHs becomes greater than four, MC-Safe is able to find a suitable channel for the emergency communication, and achieves a 15.2 ms (60.1%) lower packet delay than LCCF on average.

5.10 Hardware Testbed Experiment

To evaluate the effectiveness of MC-Safe in the real-world scenario, two emergency scenarios are investigated with our hardware testbed: (1) Rear-end collision, which is similar to previous simulation; (2) Lane changing, the car tries to avoid the collision when it detects the abrupt lane change action of another car through DSRC. For the first scenario, we investigate the *stopping distance* of MC-Safe and the baselines when different number of interference nodes are deployed. The stopping distance is defined as how long the following car travels between the moment when the heading car sends out the braking message and the moment when the following car is fully stopped. For the second scenario, we investigate the *minimum distance* during the whole lane change process, which is defined as the shortest distance between the two scaled cars from the moment when one car starts to change its lane to the moment it merges into the new lane completely. We use this metric because if the lane-changing message can be received with a shorter delay, the other car can act earlier to stay farther away from the lane-changing car. Hence, the greater the distance, the better the real-time performance. Since the performance of EDCA has been tested in the previous experiments here, we compare MC-Safe mainly with other multi-channel baselines.

Figure 19(a) shows the result for the first emergency scenario: MC-Safe has the best performance compared with other baselines. With 30 interference nodes, the stopping distance of MC-Safe is 1.32 m shorter than that of LCCF, and 2.53 m shorter than that of Random. Note that 2.53 m for scaled cars is approximately 20.24 m for real cars. Such a short stopping distance (due to better real-time communication) would significantly lower the probability of having a rear-end collision. Figure 19(b) shows that MC-Safe provides the largest minimum distance for the lane changing scenario. The minimum distance between the two vehicles can be decreased by 18.24 cm on average in the 40-interference-node case compared with LCCF. This distance is about the width of one vehicle in the real case (1.47 m). Note that a distance lower than zero means the accident avoidance system fails and a collision has occurred (e.g., CCC with 40 interference nodes). With MC-Safe, a collision can be avoided even for a high vehicle density.

6 CONCLUSION

In this article, we have proposed *MC-Safe*, a multi-channel V2V communication framework that monitors all the available channels and dynamically selects the best one for safety message transmission in a VCPS. MC-Safe features a novel channel negotiation scheme that is activated whenever two or more cars are determined to have a potential collision. All the involved cars work collaboratively in a distributed manner to select a coordinator and then identify a channel that can meet the delay and packet error rate requirements for every car. Afterward, all the involved cars switch to the selected channel for real-time safety message communication. Our evaluation results show that MC-Safe outperforms existing single-channel solutions and other well-designed multi-channel baselines, by having a 23.4% lower delay on average. Compared to WAVE, MC-Safe reduces the average delay of safety message transmission from 300 to 20 ms for a high vehicle density.

REFERENCES

[1] 2006. IEEE 1609.4. IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE).
[2] 2008. *Sampling Design Used in the National Motor Vehicle Crash Causation Survey.* Technical Report.
[3] Radhakisan Baheti and Helen Gill. 2011. Cyber-physical systems. *The Impact of Control Technology* (2011).
[4] Fan Bai et al. 2009. Spatio-temporal variations of vehicle traffic in VANETs: Facts and implications. In *Proceedings of the Annual Conference on Vehicular Ad-hoc Networks (VANET'09).*
[5] Giuseppe Bianchi. 2000. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Select. Areas Commun.* 18, 3 (2000), 535–547.
[6] Xi Chen et al. 2016. How cars talk louder, clearer and fairer: Optimizing the communication performance of connected vehicles via online synchronous control. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'16).*
[7] Lin Cheng et al. 2007. Mobile vehicle-to-vehicle channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band. *IEEE J. Select. Areas Commun.* 25, 8 (2007), 1501–1516.
[8] Elias C. Eze et al. 2014. Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward. In *2014 20th International Conference on Automation and Computing (ICAC'14).*
[9] Jose R. Gallardo et al. 2009. Performance analysis of the EDCA medium access mechanism over the control channel of an IEEE 802.11 p WAVE vehicular network. In *Proceedings of the 2009 IEEE International Conference on Communications (ICC'09).*
[10] Ali J. Ghandour et al. 2013. Improving vehicular safety message delivery through the implementation of a cognitive vehicular network. *Ad Hoc Netw.* 11, 8 (2013), 2408–2422.
[11] Ali J. Ghandour et al. 2014. Dissemination of safety messages in IEEE 802.11 p/WAVE vehicular network: Analytical study and protocol enhancements. *Perv. Mobile Comput.* 11, 1 (2014), 3–18.
[12] Ge Guo et al. 2014. Control over medium-constrained vehicular networks with fading channels and random access protocol: A networked systems approach. *IEEE Trans. Vehic. Technol.* 64, 8 (2014), 3347–3358.
[13] Michael Harris and Sarah Harvey. 2009. Channel Switching Overhead for 802.11 b. *Technical Report* (2009).
[14] Karl Hedrick. 1997. *Brake System Modeling, Control and Integrated Brake/Throttle Switching Phase I.* Technical Report. UCB-ITS-PRR-97-21.
[15] Daniel Jiang et al. 2008. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In *Proceedings of the Annual Vehicular Technology Conference (VTC'08).*
[16] Georgios Karagiannis et al. 2011. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Commun. Surv. Tutor.* 13, 4 (2011), 548–616.

[17] Long Le et al. 2009. Vehicular wireless short-range communication for improving intersection safety. *IEEE Commun. Mag.* 47, 11 (2009), 104–110.
[18] Zhiyong Lin et al. 2011. Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'11)*.
[19] Francesca Martelli et al. 2012. A Measurement-based Study of Beaconing Performance in IEEE 802.11p Vehicular Networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'12)*.
[20] Richard Martin. [n.d.]. Annual Global Road Crash Statistics. Retrieved from http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics.
[21] Damien Newton. 2010. Car Density and What It Means for the Future of LA. Retrieved from http://la.streetsblog.org/2010/12/13/density-car-ownership-and-what-it-means-for-the-future-of-los-angeles/.
[22] Sinan Oncu et al. 2014. Cooperative adaptive cruise control: Network-aware analysis of string stability. *IEEE Trans. Intell. Transport. Syst.* 15, 4 (2014), 1527–1537.
[23] Yongtae Park et al. 2012. Application-Level Frequency Control of Periodic Safety Messages in the IEEE WAVE. *IEEE Trans. Vehic. Technol.* 61, 4 (2012), 1854–1862.
[24] Yalcin Sadi et al. 2014. Minimum energy data transmission for wireless networked control systems. *IEEE Trans. Wireless Commun.* 13, 4 (2014), 2163–2175.
[25] Robin Schubert et al. 2008. Comparison and evaluation of advanced motion models for vehicle tracking. In *Proceedings of 11th International Conference on Information Fusion (ICIF'08)*.
[26] Zhibin Shuai et al. 2014. Lateral motion control for four-wheel-independent-drive electric vehicles using optimal torque allocation and dynamic message priority scheduling. *Contr. Eng. Pract.* 24 (2014), 55–66.
[27] Ken Tindell and Alan Burns. 1994. Guaranteeing message latencies on control area network (CAN). In *Proceedings of the 1st International Conference on Controller Area Network (CAN'94)*.
[28] Marc Torrent-Moreno et al. 2009. Vehicle-to-vehicle communication: Fair transmit power control for safety-critical information. *IEEE Trans. Vehic. Technol.* 58, 7 (2009), 3684–3703.
[29] Jiafu Wan et al. 2014. Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Commun. Mag.* 52, 8 (2014), 106–113.
[30] Qing Wang et al. 2012. An IEEE 802.11 p-based Multichannel MAC Scheme with Channel Coordination for Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transport. Syst.* 13, 2 (2012), 449–458.
[31] Qiao Xiang et al. 2015. Data preference matters: A new perspective of safety data dissemination in vehicular ad hoc networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'15)*.
[32] Yuan Yao et al. 2013. Performance and reliability analysis of IEEE 802.11 p safety communication in a highway environment. *IEEE Trans. Vehic. Technol.* 62, 9 (2013), 4198–4212.
[33] Yuan Yao et al. 2017. A Flexible Multi-Channel Coordination MAC Protocol for Vehicular Ad Hoc Networks. *IEEE Commun. Lett.* 21, 6 (2017), 1305–1308.
[34] Yuan Yao et al. 2017. LORA: Loss Differentiation Rate Adaptation Scheme for Vehicle-to-Vehicle Safety Communications. *IEEE Trans. Vehic. Technol.* 66, 3 (2017), 2499–2512.
[35] Hongqiang Zhai et al. 2004. Performance Analysis of IEEE 802.11 MAC Protocols in Wireless LANs. *Wireless Commun. Comput.* 4, 8 (2004), 917–931.
[36] Le Zhang et al. 2016. Congestion Control for Vehicular Networks With Safety-Awareness. *IEEE/ACM Trans. Netw.* 24, 6 (2016), 3290–3299.
[37] Yu Zheng et al. 2006. Performance Analysis of IEEE 802.11 DCF in Imperfect Channels. *IEEE Trans. Vehic. Technol.* 55, 5 (2006), 1648–1656.

Received June 2019; revised February 2020; accepted April 2020