Speed-Aware Flow Management with Packet Classification to Mitigate Congestion in VANETs

You-Chiun Wang[1,2] and Cheng-En Ho[1]
[1]Department of Computer Science and Engineering
[2]Information Security Research Center
National Sun Yat-sen University, Kaohsiung, Taiwan
Email: ycwang@cse.nsysu.edu.tw; hechengen1998@gmail.com

***Abstract*—Due to service diversity, there may be many packet flows with different urgency and importance in a *vehicular ad hoc network (VANET)*. In a traffic jam, numerous vehicles vie to send packets, which leads to network congestion. Moreover, when accidents occur, the vehicles detecting them will broadcast safety-critical messages to warn others, which could worsen congestion and thus cause the loss of many messages. To this end, the paper proposes a *speed-aware flow management with packet classification (SFM-PC)* scheme. In SFM-PC, packets will be classified by their services. According to its speed, each vehicle gauges the traffic conditions locally and adjusts the transmission rate of packets by itself. On receiving the packets of safety-critical messages, vehicles use packet consolidation to reduce the dissemination of duplicate messages. Through simulations, we demonstrate that the SFM-PC scheme can efficiently mitigate congestion in a VANET, where it has a much lower packet loss rate than other methods.**

***Keywords*—congestion, flow management, packet classification, speed, vehicular ad hoc network (VANET).**

I. Introduction

A *vehicular ad hoc network (VANET)* is composed of a set of vehicles equipped with on-board units for communications. Sometimes, a VANET may also contain the road infrastructure (called roadside units). VANETs can boost driving efficiency, reduce car accidents, and improve passenger comfort. This is achieved by exchanging traffic and entertainment data between vehicles, known as *vehicle-to-vehicle (V2V)* communications, or between vehicles and roadside units, known as *vehicle-to-infrastructure (V2I)* communications. There have been various VANET applications developed, from air-quality monitoring [1] to cooperative automotive [2], intelligent vehicles [3], and driving-data collection [4].

Multiple standards have been regulated for implementing VANETs. IEEE 802.11p is an extension of IEEE 802.11 that adds *wireless access in vehicular environments (WAVE)*. It defines the operations for MAC and physical layers in a vehicular network. Based on IEEE 802.11p, the IEEE 1609 family of standards defines operations at higher layers, which consider the architecture, management structure, communication model, and security for WAVE [5]. It is also the foundation of ITS-G5 [6], an ETSI standard for vehicular communications.

Different packet flows may be sent over a VANET to support various services. In general, there are four types of VANET services [7]. A *safety-critical service (SCS)* is used to reduce road accidents, and its flow is the most urgent to enable drivers to react early to accidents. A *traffic improvement service (TIS)* helps control the flow of cars and alleviate traffic jams to make road traffic better. A *driving-system monitoring service (DMS)* is concerned with monitoring the physiological condition of a driver and the car's status. An *infotainment service (INS)* offers entertainment to drivers and passengers during a trip, such as video sharing and Internet access. INS flows may require more bandwidth, but they can tolerate some delays.

As the density of vehicles increases, more vehicles compete for bandwidth to transmit packets, which raises the possibility of network congestion. This is more likely to occur in traffic jams. Besides, when an accident occurs, there may be multiple vehicles perceiving the accident. These vehicles will broadcast SCS packets to others for warning, making congestion worse and causing long delays or even the loss of important messages. So, it is critical to mitigate congestion on VANETs [8].

This paper proposes a *speed-aware flow management with packet classification (SFM-PC)* scheme for congestion control in VANETs. Depending on the priorities of services, SFM-PC divides packets into three classes. Then, each vehicle locally judges the traffic situation (e.g., smooth or jammed) by referring to its speed and adaptively adjusts the *transmission rate (TR)* of its packets according to the classification. Furthermore, SFM-PC prevents many SCS packets from worsening network congestion through packet consolidation. Our SFM-PC scheme is distributed in essence, as vehicles adjust TRs of packets and perform packet consolidation on their own without involving roadside units (or a central server). Simulation results reveal that our SFM-PC scheme can significantly reduce the *packet loss rate (PLR)*, as compared with existing solutions.

II. Related Work

In the literature, the congestion issue for VANETs has been widely discussed. Sailaja et al. [9] compare the performance of AODV and EDAODV (ED stands for "early congestion detection") routing in a VANET, and show that EDAODV has a lower PLR than AODV. The study [10] proposes a queue management scheme to control congestion, which schedules packets using a dual queue scheduler. Lyamin et al. [11] modify the decentralized congestion control mechanism proposed in ITS-G5 to avoid congestion and meet the age-of-information demand of an intelligent transportation system. Evidently, the above studies have different objectives from ours.

Assuming that vehicles broadcast beacons periodically, several studies mitigate congestion due to beacon transmission. The work [12] prioritizes beacons in the MAC's queue and adjusts their TRs via the LIMERIC algorithm [13]. The study [14] employs a maximum utility function to decide the period to send beacons to avoid network congestion. In [15], each vehicle chooses either mode to send beacons: 1) large coverage (with high power) but a low TR, or 2) small coverage (by decreasing power) and a high TR. However, these studies do not handle congestion caused by other (non-beacon) flows.

The work [16] considers adapting the transmitted power of vehicles based on their density to reduce congestion, but how to estimate the density is not addressed. For each vehicle v_i, Patil et al. [17] compute a *channel busy ratio (CBR)*:

$$\Gamma_i = \sum\nolimits_{v_j \in \hat{\mathcal{N}}_i} l_j / \zeta, \tag{1}$$

where $\hat{\mathcal{N}}_i$ is v_i's neighbor set, l_j is the length of messages sent by v_j (to v_i), and ζ is the channel capacity. They divide vehicles into four groups based on CBR, and assign a fixed TR to vehicles in each group (specifically, 3, 6, 9, and 12 Mbps). In [18], a *speed-based distributed congestion control (SDCC)* method is used to change a vehicle's power based on its speed. If vehicles move slowly, they may encounter congestion, so their power will be reduced. Otherwise, vehicles raise power to improve throughput. As can be seen, none of them considers the service diversity of VANETs. This motives us to propose the SFM-PC scheme, which efficiently adjusts the TR of each vehicle according to its speed and packet classification.

III. System Model

We are given a city area $\mathcal{A}$, where each road R_j has a speed limit of $S_j^{\max}$. The distribution of vehicles in $\mathcal{A}$ is not uniform. In effect, traffic jams could occur on some roads (i.e., with too many vehicles), while car flows on some roads may be smooth due to only a few vehicles. Let s_i be the speed of each vehicle v_i, which may vary based on the traffic condition. Since most vehicles have speed sensors, it is easy to obtain parameter s_i.

As mentioned in Section I, a VANET may have four types of packet flows: SCS, TIS, DMS, and INS. SCS packets are sent only when an accident occurs (by those vehicles that perceive the accident), and they must be given precedence over other packets to minimize the accident's impact (e.g., bodily injury and vehicle damage). Though traffic safety is correlated with TIS and DMS flows, their packet delays may not necessarily have an immediate impact on safety. On the other hand, INS flows are for entertainment. Hence, the priority of TIS and DMS packets should be higher than that of INS packets (but lower than SCS packets).

Roadside units are not essential for a VANET. In fact, not all roads have deployed roadside units. Thus, our work aims at V2V communications, and discusses how to let each vehicle adjust its TR of packets to mitigate congestion (i.e., distributed congestion control). Since congestion is usually accompanied by massive packet loss, we employ the PLR as a performance metric, which is defined by $(n_{\texttt{TX}} - n_{\texttt{RX}})/n_{\texttt{TX}} \times 100\%$, where $n_{\texttt{TX}}$ is the number of packets sent by originators, and $n_{\texttt{RX}}$ is the number of packets received by destinations. Note that when an accident occurs, we should minimize the PLR of SCS flows.

TABLE I
Classification of packets.

class	C1	C2	C3
service	SCS	TIS and DMS	INS
packet size	256 bytes	256 bytes	512–1024 bytes
default TR	accidents occur	10 packets/s	10–20 packets/s
urgency	high	medium	low

IV. The Proposed SFM-PC Scheme

Our SFM-PC scheme has three parts. First, SFM-PC classifies packets and makes vehicles assess their traffic conditions. Then, each vehicle adaptively adjusts its TR of packets. Lastly, we discuss how to cope with congestion due to SCS flows.

A. Packet Classification & Traffic Assessment

Table I shows packet classification, where we consider three classes: C1, C2, and C3. Specifically, SCS packets belong to class C1, which are sent by the vehicles that detect accidents and have the highest urgency. TIS and DMS flows carry safety-related information, but their delays may not necessarily have an immediate impact on traffic safety. Hence, we classify their packets into class C2. Since INS flows are usually multimedia streaming, they have larger packet sizes and consume more bandwidth (which is reflected in the default TR). Their packets are classified as C3. Evidently, the order of priority will be C1 $>$ C2 $>$ C3.

Suppose that a vehicle v_i is on a road R_j. Then, v_i decides the traffic condition by the relationship between its speed s_i and R_j's speed limit $S_j^{\max}$. To do so, we use two thresholds δ_{H} and δ_{L} to divide traffic conditions into three levels: 1) *smooth:* $s_i \geq \delta_{\text{H}} \times S_j^{\max}$, 2) *moderate:* $\delta_{\text{L}} \times S_j^{\max} \leq s_i < \delta_{\text{H}} \times S_j^{\max}$, and 3) *jammed:* $s_i < \delta_{\text{L}} \times S_j^{\max}$, where $0 < \delta_{\text{L}} < \delta_{\text{H}} < 1$. For example, we can set $\delta_{\text{L}} = 0.5$ and $\delta_{\text{H}} = 0.9$.

B. Adaptive TR Adjustment

Based on the classification of packets and the level of traffic conditions, vehicles can adaptively adjust their TRs to mitigate congestion. Because the packets of class C1 are sent only when an accident occurs, our discussion on TR adjustment focuses on classes C2 and C3. In Section IV-C, we will detail how to mitigate congestion caused by class C1 (i.e., SCS) packets.

Let $r_i(t)$ be the TR of a vehicle v_i at time t. If the traffic condition becomes smooth, we set $r_i(t) = r_i^{\texttt{S}}$ (i.e., default TR in Table I). Otherwise, there are two cases to be discussed:

Case 1. The traffic condition starts becoming moderate at time t. We set $r_i(t) = r_i^{\texttt{M}} = \lambda_{\texttt{M}} \times r_i^{\texttt{S}}$, where $0 < \lambda_{\texttt{M}} < 1$. At time $t + k$, where $k \in \mathbb{Z}^+$, the TR is adjusted to

$$r_i(t+k) = \min\{r_i(t+k-1) \times \mu_{\texttt{M}}(t+k), r_i^{\texttt{S}}\}, \tag{2}$$

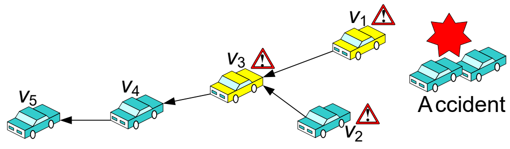


Fig. 1. Example of packet consolidation to handle bursty congestion.

where

$$\mu_{\mathtt{M}}(t+k) = \begin{cases} \mu_{\mathtt{M}}(t+k-1) + (-1)^{\gamma} \times \beta_2^{\mathtt{M}} & \text{for C2,} \\ \mu_{\mathtt{M}}(t+k-1) + (-1)^{\gamma} \times \beta_3^{\mathtt{M}} & \text{for C3,} \end{cases} \quad (3)$$

where $0 < \beta_2^{\mathtt{M}} < \beta_3^{\mathtt{M}} < 0.1$ and $\beta_2^{\mathtt{M}} \approx \beta_3^{\mathtt{M}}$. Note that $\mu_{\mathtt{M}}(t) = 1$. The exponent γ depends on the change of CBR in Eq. (1):

$$\gamma = \begin{cases} 0 & \text{if } \Gamma_i(t+k-1)/\Gamma_i(t+k-2) < 1, \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

Let us discuss the rationale. In Eq. (2), the new TR $r_i(t+k)$ at time $t+k$ is the product of $r_i(t+k-1)$ (i.e., the previous TR) and a *scale factor* $\mu_{\mathtt{M}}(t+k)$. However, we should avoid $r_i(t+k)$ exceeding $r_i^{\mathtt{S}}$ (i.e., the TR when the traffic condition is smooth). The scale factor is also dynamic, as computed by Eq. (3). Specifically, if CBR reduces (i.e., $\Gamma_i(t+k-1)/\Gamma_i(t+k-2) < 1$), the exponent γ is set to 0, so the scale factor $\mu_{\mathtt{M}}(t+k)$ will be increased by a small amount of $\beta_2^{\mathtt{M}}$ or $\beta_3^{\mathtt{M}}$ if v_i sends classes C2 or C3 packets, as compared with the previous scale factor $\mu_{\mathtt{M}}(t+k-1)$. In this situation, v_i's channel load reduces, so it is safe to increase v_i's TR to improve throughput. Otherwise, we shall reduce v_i's TR (as its channel load rises) to mitigate congestion. Therefore, we set $\gamma = 1$ and $\mu_{\mathtt{M}}(t+k)$ will be decreased by an amount of $\beta_2^{\mathtt{M}}$ or $\beta_3^{\mathtt{M}}$ if v_i sends classes C2 or C3 packets, as compared with $\mu_{\mathtt{M}}(t+k-1)$.

As class C3 packets are sent by INS flows that spend more bandwidth, we let $\beta_3^{\mathtt{M}}$ be larger than $\beta_2^{\mathtt{M}}$. Thus, we can prevent class C3 packets from worsening the channel load if CBR rises (i.e., to avoid potential congestion). On the contrary, when the channel load decreases, INS flows can be given slightly more bandwidth (by raising the TR of class C3 packets).

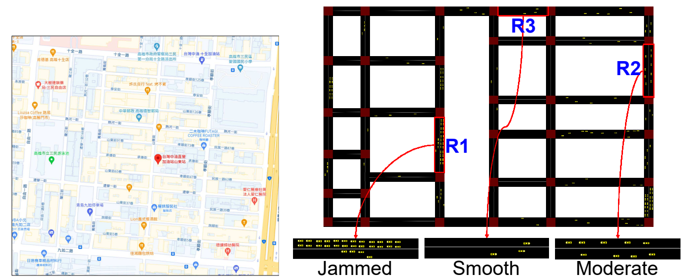
Case 2. The traffic condition starts becoming jammed at time t. We set $r_i(t) = \lambda_{\mathtt{J}} \times r_i^{\mathtt{S}}$. Here, $\lambda_{\mathtt{J}}$ should be smaller than $\lambda_{\mathtt{M}}$ since congestion could occur (and $\lambda_{\mathtt{J}} > 0$). Then, at time $t+k$, where $k \in \mathbb{Z}^+$, the TR is adjusted to

$$r_i(t+k) = \min\{r_i(t+k-1) \times \mu_{\mathtt{J}}(t+k), r_i^{\mathtt{M}}\}, \quad (5)$$

where

$$\mu_{\mathtt{J}}(t+k) = \begin{cases} \mu_{\mathtt{J}}(t+k-1) + (-1)^{\gamma} \times \beta_2^{\mathtt{J}} & \text{for C2,} \\ \mu_{\mathtt{J}}(t+k-1) + (-1)^{\gamma} \times \beta_3^{\mathtt{J}} & \text{for C3,} \end{cases} \quad (6)$$

where $\beta_3^{\mathtt{M}} < \beta_2^{\mathtt{J}} < \beta_3^{\mathtt{J}} < 0.1$, $\beta_2^{\mathtt{J}} \approx \beta_3^{\mathtt{J}}$, and $\mu_{\mathtt{J}}(t) = 1$. The adjustment of TR is similar to case 1, but with two differences. First, we impose an upper bound $r_i^{\mathtt{M}}$ (rather than $r_i^{\mathtt{S}}$) on TR in Eq. (5). Second, the change amplitude (i.e., $\beta_2^{\mathtt{J}}$ or $\beta_3^{\mathtt{J}}$) in the scale factor $\mu_{\mathtt{J}}(t+k)$ is larger than that of $\mu_{\mathtt{M}}(t+k)$. Doing so can lower the TR faster to avoid worsening congestion.



(a) street map of Kaohsiung (b) roads and vehicle distribution

Fig. 2. City area $\mathcal{A}$ considered in the simulation.

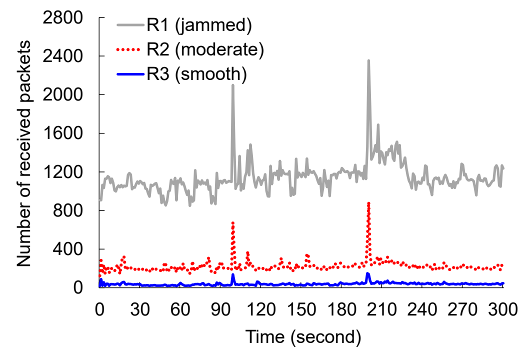


Fig. 3. Average number of packets received by a vehicle.

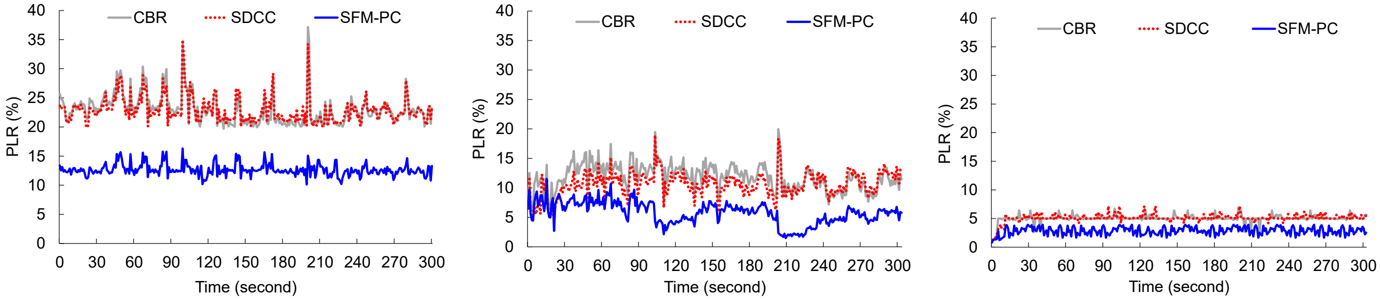
C. Handling Congestion Caused by SCS Flows

When an accident occurs, multiple vehicles may detect the accident (probably with some time difference). They successively broadcast SCS packets to warn others, and the network will soon be flooded with SCS packets that describe the same accident. Since SCS packets belong to class C1 (i.e., with the highest priority), they could quickly consume bandwidth. This is called *bursty congestion* [19]. Fig. 1 gives an example. Three vehicles detect the accident with a sequence of v_1, v_2, and v_3. Hence, v_1, v_2, and v_3 broadcast SCS packets sequentially, which wastes bandwidth and causes bursty congestion.

To solve the problem, we adopt packet consolidation. When it is the first time that a vehicle v_i receives an SCS packet p_j that describes an accident, v_i records p_j (for a short time) as a reference and resends p_j to neighbors. Then, if v_i gets other SCS packets that depict the same accident as p_j, v_i directly drops these packets (as the information has been disseminated to neighbors). Moreover, if v_i itself detects the accident after sending p_j, v_i will not generate SCS packets. Take Fig. 1 as an example. When v_3 receives an SCS packet from v_1, it will send the packet to neighbor v_4. However, v_3 neither resends v_2's SCS packet nor generates its own SCS packet when v_3 itself detects the accident later.

V. Performance Evaluation

To evaluate system performance, we employ both SUMO and OMNeT++ to construct a simulation. Specifically, SUMO is an open-source package that allows for the modeling of practical roads and car traffic [20]. In our simulation, we select



(a) road R1 (jammed) (b) road R2 (moderate) (c) road R3 (smooth)

Fig. 4. Comparison of PLR between different methods.

a $2000\,\mathrm{m} \times 1000\,\mathrm{m}$ area $\mathcal{A}$ from the downtown of Kaohsiung City, Taiwan, as presented in Fig. 2(a). More concretely, we first obtain the street map through OpenStreetMap. By using SUMO's road-network importer, namely *netconvert*, we can deploy streets and lanes on the map [21]. The speed limit of each road in $\mathcal{A}$ is 50 km/h or 60 km/h. Afterward, we decide the locations of traffic lights, where their cycles are set to 60 seconds. There will be around 200 to 300 vehicles moving in $\mathcal{A}$, where they follow the Manhattan grid mobility model [22].

In $\mathcal{A}$, we select three roads with different traffic conditions, as shown in Fig. 2(b). In particular, the densities of vehicles on roads R1, R2, and R3 are high, medium, and low, respectively. As mentioned in Section IV-A, the traffic conditions on roads R1, R2, and R3 correspond to jammed, moderate, and smooth, respectively. In Fig. 3, we show the average number of packets received by a vehicle on these three roads (without applying any method to mitigating congestion). It is apparent that the higher the density of vehicles, the more packets are generated in the network. Hence, the vehicle will receive more packets. In addition, accidents happen at the 100th and 200th seconds, so each line in Fig. 3 has two obvious peaks.

To emulate V2V communications between vehicles, we use OMNeT++ [23], an extensible, modular, and component-based network simulator. The underlying protocol is IEEE 802.11p, whose operating band is 5.9 GHz. Since vehicles move, their wireless signals will be affected by fast fading, and we adopt the Rayleigh fading model [24].

Vehicles will generate class C2 packets (for TIS and DMS flows) or class C3 packets (for INS flows), whose parameters are given in Table I. If a vehicle detects an accident, it produces class C1 packets (for SCS flows). We compare our proposed SFM-PC scheme with two methods: CBR [17] and SDCC [18]. As discussed in Section II, CBR uses Eq. (1) to estimate the channel load for each vehicle and decides its TR accordingly. SDCC adjusts each vehicle's power based on its speed. When vehicles move slowly, their power is reduced, and vice versa. In SFM-PC, we set $\lambda_{\mathtt{M}} = 0.6$, $\lambda_{\mathtt{J}} = 0.2$, $\beta_2^{\mathtt{M}} = 0.01$, $\beta_3^{\mathtt{M}} = 0.02$, $\beta_2^{\mathtt{J}} = 0.04$, and , $\beta_3^{\mathtt{J}} = 0.05$. The simulation time is 300 s.

Fig. 4(a) presents the PLR of vehicles on road R1 as time goes by, where the traffic condition is jammed. Evidently, both CBR and SDCC perform about the same, with their PLRs always above 20% and vibrating significantly. Moreover, there are two manifest peaks at the 100th and 200th seconds. That is because accidents occur, and numerous vehicles broadcast SCS packets, thereby causing congestion. By contrast, SFM-PC can substantially reduce packet loss, and its PLR is relatively stable (in particular, between 10% and 15%). When accidents occur, SFM-PC uses packet consolidation to avoid bursty congestion caused by SCS packets. Hence, unlike CBR and SDCC, SFM-PC's line has no obvious peak. This experimental result reveals that our SFM-PC scheme outperforms others in a traffic jam, and its performance can stay stable.

TABLE II
AVERAGE PLR OF EACH METHOD.

method	R1	R2	R3
CBR	22.97%	11.83%	5.02%
SDCC	22.85%	10.64%	5.09%
SFM-PC	12.66%	5.79%	2.85%

Fig. 4(b) shows the PLR of vehicles on road R2 along the time axis. Since the traffic condition becomes moderate (that is, there are fewer vehicles on road R2 than on road R1), the PLR of each method will decrease. In this case, the difference in speeds between vehicles also rises. Hence, SDCC can better adjust the power of each vehicle based on the speed difference. That explains why SDCC has a lower PLR than CBR. On the other hand, our SFM-PC scheme takes advantage of packet classification and consolidation to alleviate congestion, so it can further reduce the PLR as compared with SDCC.

Fig. 4(b) gives the PLR of vehicles on road R3 at different times, whose traffic condition is smooth. On this road, there are a few vehicles competing for bandwidth to send packets. Thus, all methods have low PLRs (i.e., below 6.5%), and their lines have no apparent peaks. In other words, the transmission of SCS packets due to the occurrence of accidents does not cause congestion on road R3. Among all methods, our SFM-PC scheme can always maintain the lowest PLR.

Table II summarizes the average PLR of each method, where we take the average value of PLRs over 300 seconds. As can be seen, SFM-PC has significantly lower PLRs than both CBR and SDCC under different traffic conditions. Specifically, our SFM-PC scheme can reduce packet loss by 46.4% and 44.8%, as compared with the CBR and SDCC methods, which shows

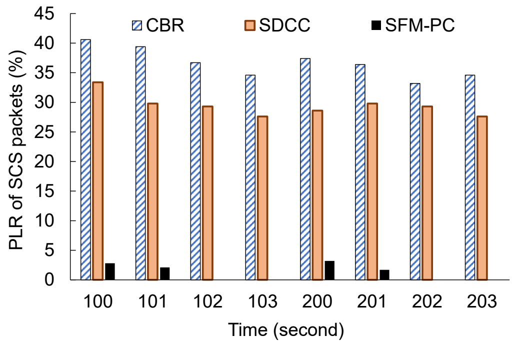


Fig. 5. Comparison of PLR for SCS packets on road R1.

its effectiveness in mitigating congestion.

As mentioned earlier in Section IV-C, when one accident occurs, multiple vehicles may detect the accident and broadcast SCS packets for warning. In a traffic jam, doing so could lead to bursty congestion. Fig. 5 gives the PLR of vehicles on road R1 during the occurrence of accidents (i.e., the 100th–103rd seconds for the first accident and the 200th–203rd seconds for the second accident), where we count only SCS packets. Since SDCC will lower the power of vehicles in a traffic jam, it has a lower PLR than CBR. Thanks to packet consolidation, our SFM-PC scheme reduces the dissemination of duplicate SCS packets in the VANET. Hence, SFM-PC can efficiently handle bursty congestion, where only a few SCS packets are lost.

VI. Conclusion

In this paper, we propose the SFM-PC scheme to manage packet flows in a VANET. Considering the diversity of VANET services, we categorize packets into three classes. Each vehicle gauges the traffic conditions based on its speed and the road's speed limit, and adaptively adjusts its TR to mitigate network congestion. To address the bursty congestion problem caused by accidents, vehicles carry out packet consolidation to avoid sending duplicate SCS packets that describe the same accident. By using SUMO and OMNeT++ to build the simulation, we show that our SFM-PC scheme substantially reduces the PLR (especially for SCS packets) as compared with existing CBR and SDCC methods, which verifies that SFM-PC can mitigate congestion efficiently in a VANET.

Acknowledgment

This work was supported by National Science and Technology Council, Taiwan under Grants 111-2221-E-110-023-MY2 and 112-2634-F-110-001-MBK, and the Information Security Research Center at National Sun Yat-sen University, Taiwan.

References

[1] S. C. Hu, Y. C. Wang, C. Y. Huang, and Y. C. Tseng, "Measuring air quality in city areas by vehicular wireless sensor networks," *Journal of Systems and Software*, vol. 84, no. 11, pp. 2005–2012, 2011.

[2] A. Ghosh, V. V. Paranthaman, G. Mapp, O. Gemikonakli, and J. Loo, "Enabling seamless V2I communications: toward developing cooperative automotive applications in VANET systems," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 80–86, 2015.

[3] M. Haris, M. A. Shah, and C. Maple, "Internet of intelligent vehicles (IoIV): an intelligent VANET based computing via predictive modeling," *IEEE Access*, vol. 11, pp. 49 665–49 674, 2023.

[4] Y. C. Wang, "Mobile sensor networks: system hardware and dispatch software," *ACM Computing Surveys*, vol. 47, no. 1, pp. 12:1–12:36, 2014.

[5] IEEE Standards Association, "IEEE guide for wireless access in vehicular environments (WAVE) architecture," IEEE Std 1609.0-2019, 2019.

[6] ETSI, "Intelligent transport systems (ITS); ITS-G5 access layer specification for intelligent transport systems operating in the 5 GHz frequency band," EN 302 663 V1.3.1, 2019.

[7] N. H. Hussein, C. T. Yaw, S. P. Koh, S. K. Tiong, and K. H. Chong, "A comprehensive survey on vehicular networking: communications, applications, challenges, and upcoming research directions," *IEEE Access*, vol. 10, pp. 86 127–86 180, 2022.

[8] A. Balador, E. Cinque, M. Pratesi, F. Valentini, C. Bai, A. A. Gomez, and M.Mohammadi, "Survey on decentralized congestion control methods for vehicular communication," *Vehicular Communications*, vol. 33, pp. 1–17, 2022.

[9] P. Sailaja, B. Ravi, and T. Jaisingh, "Performance analysis of AODV and EDAODV routing protocol under congestion control in VANETs," in *International Conference on Inventive Communication and Computational Technologies*, 2018, pp. 945–948.

[10] A. Shwetha and P. Sankar, "Queue management scheme to control congestion in a vehicular based sensor network," in *International Conference on Inventive Systems and Control*, 2018, pp. 917–921.

[11] N. Lyamin, B. Bellalta, and A. Vinel, "Age-of-information-aware decentralized congestion control in VANETs," *IEEE Networking Letters*, vol. 2, no. 1, pp. 33–37, 2020.

[12] S. Sharma, M. Chahal, and S. Harit, "Transmission rate-based congestion control in vehicular ad hoc networks," in *Amity International Conference on Artificial Intelligence*, 2019, pp. 303–307.

[13] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: a linear adaptive message rate algorithm for DSRC congestion control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, 2013.

[14] H. Song, Y. Wang, P. Hong, and L. Chen, "A self-selected congestion scheme based on CAMs maximum utility function," in *International Conference on Virtual Reality and Intelligent Systems*, 2020, pp. 1007–1010.

[15] J. Choi and H. Kim, "A QoS-aware congestion control scheme for C-V2X safety communications," in *IEEE Vehicular Networking Conference*, 2020, pp. 1–4.

[16] O. Akinlade, I. Saini, X. Liu, and A. Jaekel, "Traffic density based distributed congestion control strategy for vehicular communication," in *International Conference on Distributed Computing in Sensor Systems*, 2019, pp. 195–197.

[17] A. Patil, M. Deeksha, N. Shekar, V. Shet, and M. Kulkarni, "Transmit data rate control based decentralized congestion control mechanism for VANETs," in *International Conference on Data Science and Communication*, 2019, pp. 1–5.

[18] C. Facchina and A. Jaekel, "Speed based distributed congestion control scheme for vehicular networks," in *IEEE Symposium on Computers and Communications*, 2020, pp. 1–4.

[19] Y. C. Wang and C. T. Wei, "Lightweight, latency-aware routing for data compression in wireless sensor networks with heterogeneous traffics," *Wireless Communications and Mobile Computing*, vol. 16, no. 9, pp. 1035–1049, 2016.

[20] SUMO, https://www.eclipse.org/sumo/.

[21] Y. C. Wang and G. W. Chen, "Efficient data gathering and estimation for metropolitan air quality monitoring by using vehicular sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7234–7248, 2017.

[22] W. H. Yang, Y. C. Wang, Y. C. Tseng, and B. S. P. Lin, "Energy-efficient network selection with mobility pattern awareness in an integrated WiMAX and WiFi network," *International Journal on Communication Systems*, vol. 23, no. 2, pp. 213–230, 2010.

[23] OMNeT++, https://omnetpp.org/.

[24] W. K. Lai, Y. C. Wang, H. C. Lin, and J. W. Li, "Efficient resource allocation and power control for LTE-A D2D communication with pure D2D model," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3202–3216, 2020.