Quick and Autonomous Platoon Maintenance in Vehicle Dynamics For Distributed Vehicle Platoon Networks

Ankur Sarker
Department of Computer Science
University of Virginia
as4mz@virginia.edu

Chenxi Qiu
Information Science & Technology
Pennsylvania State University
czq3@psu.edu

Haiying Shen
Department of Computer Science
University of Virginia
hs6ms@virginia.edu

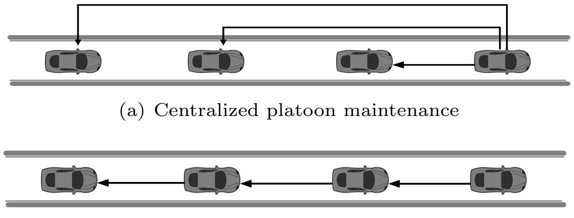
ABSTRACT

Platoon systems, as a type of adaptive cruise control systems, will play a significant role to improve travel experience and roadway safety. The stability of a platoon system is crucial so that each vehicle maintains a safety distance from its proceeding vehicle and can take necessary actions to avoid collisions. However, current centralized platoon maintenance method cannot meet this requirement. We suggest to use a decentralized platoon maintenance method, in which each vehicle communicates with its neighbor vehicles and self-determines its own velocity. However, a vehicle needs to know its distance from its preceding vehicle to determine its velocity, which is unavailable in vehicle communication disconnection caused by vehicle dynamics (i.e., node joins and departures). Thus, a formidable challenge is: *how to recover the platoon quickly in vehicle dynamics even when the distance information is unavailable?* To handle this challenge, we first profile a succeeding vehicle's velocity to minimize the time to recover the connectivity hole with its preceding vehicle and find that the profiles are almost the same at the beginning regardless of its current velocity and distance to its preceding vehicle. Accordingly, we devise a strategy, in which a succeeding vehicle uses its stored common velocity profile when it is disconnected from its preceding vehicle and then adjusts its velocity once the connection is built. Experimental results from simulation show the efficiency and effectiveness of our decentralized platoon maintenance method.

CCS CONCEPTS

•Networks →Cyber-physical networks; Mobile ad hoc networks; *Mobile networks;*

ACM Reference format:
Ankur Sarker, Chenxi Qiu, and Haiying Shen. 2016. Quick and Autonomous Platoon Maintenance in Vehicle Dynamics For Distributed Vehicle Platoon Networks. In *Proceedings of IoTDI '17, Pittsburgh, PA, USA, April 18-21, 2017,* 6 pages.
DOI: http://dx.doi.org/10.1145/3054977.3054998


(a) Centralized platoon maintenance

(b) Decentralized platoon maintenance

Figure 1: Centralized vs. Decentralized platoon maintenance.

1 INTRODUCTION

As a next-generation facility of land transportation systems [10, 12, 18], vehicle platoon systems have been drawn much attention in recent few years. In a vehicle platoon system, several vehicles follow one leader vehicle and run in a line to maintain a constant velocity. Also, each vehicle maintains a safety distance from its preceding vehicle. Because of the shorter inter-vehicle distance in one lane, platoon can provide higher roadway throughput and better traffic flow control [17]. It can also help to reduce energy consumption by avoiding unnecessary changes of acceleration [1]. In spite of these advantages, the shorter inter-vehicle distance brings about a safety issue, which requires the vehicles in a platoon always maintain the safety inter-vehicle distance. This requirement needs platoon maintenance for vehicle dynamics (i.e., vehicle joins and departures). When a vehicle enters a platoon, its succeeding vehicles need to decrease their velocities to leave enough space for the entering vehicle. Similarly, when a vehicle leaves the platoon, the velocities of the vehicles behind it need to be increased to maintain the platoon.

Previous platoon systems [2, 5, 7, 16] maintain an entire platoon using a centralized approach. As shown in Fig. 1(a), the leader vehicle periodically collects each follower vehicle's information, e.g., velocity and location, and calculates and notifies the velocity for each follower vehicle. However, the centralized platoon maintenance cannot provide timely velocity adjustment to guarantee vehicles' safety due to high transmission delay and packet drop rate, as well as the high computation overhead of the leader vehicle. First, the limited communication range of the leader vehicle and long communication distance between the leader vehicle and some follower vehicles may lead to packet drops and transmission delay due to retransmission. Second, because all follower vehicles need to communicate with the leader vehicle, which can only complete one single transmission in each time slot, it leads

---

IoTDI '17, Pittsburgh, PA, USA

DOI: http://dx.doi.org/10.1145/3054977.3054998

to much higher transmission delay. Third, the leader vehicle needs to periodically communicate with all of the follower vehicles in the platoon and calculates each vehicle's velocity. The high communication and computation overheads on the leader vehicle may prevent it from notifying the follower vehicles their velocities in time.

To overcome the drawbacks of the centralized platoon maintenance method, we propose to use a decentralized platoon maintenance method, in which each vehicle only needs to communicate with its succeeding vehicle without any explicit centralized control. As shown in Fig. 1(b), each vehicle periodically transmits a message containing its velocity and location to its succeeding vehicle, and determines its own velocity based on its current velocity and the received information from its preceding vehicle. However, though the decentralized platoon maintenance method can overcome all the aforementioned drawbacks, it brings about other problems. First, to determine the acceleration, the existing velocity control methods [2, 4, 5, 9] require the distance information from the succeeding vehicle (called accelerating vehicle) to the proceeding vehicle of the leaving node, which is unavailable when these two vehicles are disconnected in the decentralized platoon method. Since the communication range of each vehicle is about 80m, the leaving vehicle creates a connectivity hole inside the platoon.Second, it is desirable for a succeeding vehicle of a leaving vehicle or a joining vehicle to quickly adjust its velocity to maintain the platoon. A a result, a formidable challenge is *how to recover the platoon quickly in vehicle dynamics even when the distance information is unavailable?*

In this paper, we aim to handle this challenge in the decentralized platoon maintenance method. We first profile the accelerating vehicle's velocity to minimize the time to recover the connectivity hole with its preceding vehicle given its current velocity and its distance from its preceding vehicle with the consideration of the constraints of legal velocity, driver's convenience and mechanical control. A velocity profile records the vehicle's velocity in each time point over a future time period to cover a connectivity hole. By calculating the velocity profiles for different accelerating vehicle's current velocity and its distance from the preceding vehicle, we find that 1) the profiles are almost the same at the beginning, and 2) the time point that the profiles begin to be different is always after the time point that the accelerating vehicle is connected with its preceding vehicle. Accordingly, we call the common velocity profile part at the beginning *velocity accelerating profile (VACP)*, and call the subsequent velocity profile part *velocity adjusting profile (VADP)*. By taking advantage of the observed features, any accelerating vehicle can use the VACP regardless of its current velocity and distance to its preceding vehicle until it is connected with its preceding vehicle, and then use the VADP to adjust its velocity based on its current velocity and distance to its preceding vehicle. As a result, by maintaining the precalculated VACP and VADP in storage, each accelerating vehicle can immediately know its velocity to maintain the platoon upon vehicle leaving without any calculation. The same approach can be used for the vehicle joining the platoon.

Finally, we simulate our decentralized platoon method. The simulation results demonstrate even lacking in the absence of inter-vehicle distance information, our proposed decentralized platoon maintenance method performs similar to the existing centralized platoon maintenance method [19] in both aspects of platoon maintenance and safety violations. The simulation results also indicate the better performance of our platoon maintenance method compared to the previous method in terms of recovering connectivity holes.

The rest of this paper is organized as follows. Section 2 presents the research problem in detail. Section 3 presents our methods to solve the problem of the dynamic platoon's behaviors and Section 4 presents the performance of our methods in experimental study. Section 5 presents the related work. Section 6 concludes this paper with our remarks on the future work.

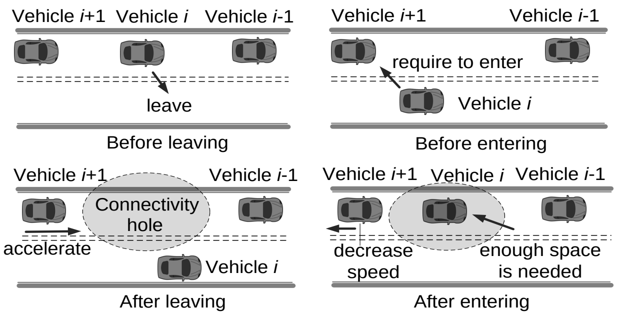
2 PROBLEM STATEMENT

2.1 Preliminaries

We consider n vehicles, denoted by $\{1, ..., n\}$, running on a road in a platoon. Let $x_i(t)$, $v_i(t)$, and $a_i(t)$ denote the location, the velocity, and the acceleration of vehicle i at time t, respectively. We use $d_{i,j}(t) = |x_i(t) - x_j(t)|$ to represent the Euclidean distance between vehicles i and j at time t. We use δ to denote the sum of the safety distance and vehicle's general length, called vehicle distance. That is, if there are m vehicles in the platoon, the platoon length is approximately δm. For avoiding complexity, we assume the lengths of all vehicles are same. Accordingly, we require $d_{i,i+1}(t) \geq \delta \ \forall i$. In addition, we assume that the mobile device in each vehicle has the same communication range R. To guarantee the connectivity of the platoon network, the distance between any two consecutive vehicles i and $i+1$ in the platoon must not exceed R, i.e., $d_{i,i+1}(t) \leq R$. As we indicated in Section 1, the decentralized platoon network can overcome the drawbacks of the centralized platoon network to increase the platoon length and vehicle safety, but also bring about one major problem. In Section 2.2 below, we will explain the problem in more detail, respectively. Then, in Section 3, we will present our solutions for the problem.

2.2 Platoon Maintenance

For platoon maintenance, we consider the following two scenarios: 1) when a vehicle leaves the platoon, and 2) when a vehicle enters the platoon. In the first case (Fig. 2(a)), vehicle i leaves the platoon, which generates a connectivity hole between vehicles $i-1$ and $i+1$ in the platoon. Then, vehicle $i+1$ needs to increase its velocity to recover the connectivity hole. In the second case (Fig. 2(b)), vehicle i, which is outside of the platoon, requests to enter the platoon between vehicles $i-1$ and $i+1$. After receiving the request from vehicle i, vehicle $i+1$ starts to decrease its velocity to generate enough space for vehicle i to move in. For both two scenarios, our



(a) Leaving (b) Entering

Figure 2: The leaving and entering cases.

objective is to determine a velocity profile for vehicle $i+1$, which records the vehicle's velocity in each time point over a time period, to generate enough space or to recover the hole as soon as possible. Then, immediate following vehicle $i+2$ adjusts its velocity accordingly (to generate space or recover hole) so that vehicle $i+2$ maintains the inter-vehicle safety distance between itself and vehicle $i+1$. This procedure goes on from next immediate following vehicle $i+3$ to last vehicle n. Let t_0 denote the time point that vehicle $i+1$ starts to change its velocity, then its velocity profile is represented as a series $\{v_{i+1}(t_0), v_{i+1}(t_0+1), v_{i+1}(t_0+2), ..., v_{i+1}(t_{\text{end}})\}$, where t_{end} denotes the end time point of leaving process or entering process. To determine vehicle $i+1$'s profile, we need to consider the following constrains.

Constraint 1. *Legal velocity constraint*: to guarantee vehicle operation within the legal velocity limits, we need to ensure $v_{\min} \le v_{i+1}(t) \le v_{\max}, \forall t$, where $v_{\min}$ and $v_{\max}$ are the minimum and the maximum velocity allowed in a road.

Constraint 2. *Convenience constraint*: for drivers and passengers' convenience, the acceleration of vehicle $i+1$, $a_i(t)$, cannot be too large. Hence, we have the limitations $a_{i+1}(t) \le a_{\max}, \forall t$, where $a_{\max}$ is the maximum acceleration of a vehicle such that drivers and passengers would not feel any sudden change of velocity.

Constraint 3. *Mechanical control constraint*: According to the control theory, the current status of each vehicle, i.e., velocity, acceleration and location, is constrained by the follow equations, which can be derived from [19] as shown in Appendix.

$$\begin{bmatrix} x_i(t+1) \\ v_i(t+1) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_i(t) \\ v_i(t) \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \end{bmatrix} a_i \quad (1)$$

$$a(t+1) = \text{sgn}(a_{\max}) \quad (2)$$

According to the above constraints, we formulate the following two optimal control problems:

Vehicle leaving problem: We require $d_{i+1,i-1}(t_{\text{end}}) \le R$, which means the vehicles $i-1$ and $i+1$ are within each other's transmission range at time t_{end}. The objective is to minimize the ending time of the leaving process.

$$\begin{aligned} \min \quad & t_{\text{end}} - t_0 && (3) \\ \text{s.t.} \quad & \text{Constraints 1-3 are satisfied} && (4) \\ & d_{i+1,i-1}(t_{\text{end}}) \le R && (5) \end{aligned}$$

Vehicle entering problem: We require that $d_{i+1,i-1}(t_{\text{end}}) \ge 2\delta$, i.e., vehicle $i+1$ has left enough space for vehicle i to move in at time t_{end}. The objective is to minimize the ending time of the entering process:

$$\begin{aligned} \min \quad & t_{\text{end}} - t_0 && (6) \\ \text{s.t.} \quad & \text{Constraints 1-3 are satisfied} && (7) \\ & d_{i+1,i-1}(t_{\text{end}}) \ge 2\delta && (8) \end{aligned}$$

For both problems, given the inputs $v_{i+1}(t_0)$ and $d_{i-1,i+1}(t_0)$ (i.e., the velocity of vehicle $i+1$ and the distance between vehicles $i+1$ and $i-1$ at time t_0), we can get the output (vehicle $i+1$'s velocity profile) using the existing method in [19]. The problem described in [19] is how to change a vehicle's speed to make the inter-vehicle distance equal to the safety distance when the inter-vehicle distance is smaller or larger than the safety distance. Both our entering problem and leaving problem are special cases of the problem in [19], and we can directly use the method in [19] to solve our problems. Specifically, we iteratively derive the velocity in each time spot in the profile. That is, in each iteration, we derive the velocity at time $t+1$ from the velocity at time t using Equ. (1) and Equ. (2) and then adjust the calculated velocity by Constraints 1 and 2. Unfortunately, in the decentralized platoon network, if vehicle $i+1$ is disconnected with vehicle $i-1$, vehicle $i+1$ cannot get $d_{i-1,i+1}(t_0)$, which is assumed to be known in this method. Further, to calculate the velocity in each time spot, this method needs to call for Equation (1), which has high computation time complexity. Also, the number of time spots in each velocity profile is usually larger than 100 [19], which leads to a long delay for the platoon application. Then, the question is:

- *how to recover the platoon quickly in vehicle dynamics even when the distance information is unavailable?*

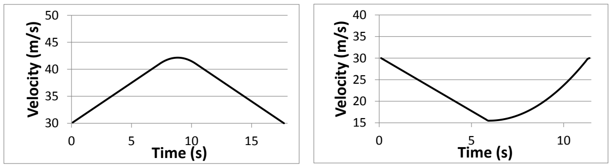
3 SYSTEM DESIGN

In this section, we introduce how to solve the platoon maintenance problem in the decentralized platoon network. At first, we present the concept of velocity profile. Then, we introduce our solution for platoon maintenance.

3.1 Platoon Maintenance

In this section, we aim to overcome the challenges indicated in Section 2.2 to solve platoon maintenance problem. We first profile the velocity of the succeeding vehicle of the leaving or joining vehicle to recover the platoon by solving the optimal problems in Section 2.2. Then, based on our observations from the velocity profiles, we propose our solution.

3.1.1 Observation of Velocity Profiles. We first calculate the velocity profiles for both optimal control problems in Section 2.2 (the vehicle entering problem and the vehicle leaving problem) by directly using the method introduced in [19], which aims to calculate the vehicle profile to make the inter-vehicle distance equal to the safety distance. Here,



(a) Vehicle leaving (b) Vehicle entering

Figure 3: Velocity profile.

one profile records the vehicle's velocity in each time point over a time period, given the vehicle's current velocity and its distance from its preceding vehicle. By referring to [13], we set $v_{\max} = 42m/s$, $v_{\min} = 20m/s$ and $a_{\max} = 2.5m/s$.

Fig. 3(a) and Fig. 3(b) respectively show the derived velocity profiles for the vehicle leaving problem and the vehicle entering problem when the vehicle's velocity is 30m/s (the average velocity on high way) and its distance from its preceding vehicle is 50m [13]. As we expect, the velocity of vehicle in Fig. 3(a) first increases from 30m/s to 43m/s, and then decreases to 30m/s. It is because when a vehicle is required to recover the hole generated by the leaving vehicle, it first accelerates until it reaches the preceding vehicle's communication range, and then decreases the velocity to the original level to keep the safety distance. In contrast, in Fig. 3(b), the velocity of the vehicle first decreases from 30m/s to 15m/s, and then increases back to 30m/s. It is because when a vehicle is required to leave space for the entering vehicle, it first decreases its velocity, and after the space is large enough for the entering vehicle, it needs to recover the velocity to the original level.

3.1.2 Solution to Recover Platoon. Though existing methods [2–5, 9] can be used to calculate the velocity profiles, the computation time of these methods is relatively high for higher order discrete differential equations discussed in Section 2.2, where the current state depends on previous states, and highly dynamic traffic scenarios are considered. For example, when $v_{i+1}(t_0) = 30$ m/s and $d_{i-1,i+1}(t_0) = 50$m, it takes about 1.96 seconds to run the algorithm proposed in [4] on a computer with Intel core-i3 and 4GB RAM. Apparently, such computation delay is unacceptable for the platoon application, in which collision avoidance is the primary concern. Also, previous vehicle control methods [2, 9] assume that by collecting the location information of the preceding vehicle, vehicle $i+1$ knows its distance from vehicle $i-1$, $d_{i-1,i+1}$, which is required to calculate vehicle $i+1$'s velocity profile. However, when vehicle $i+1$ is disconnected with vehicle $i-1$, vehicle $i+1$ cannot learn $d_{i-1,i+1}$.

Our proposed approach based on our observations from our velocity profiling can handle these two challenges. First, we consider the vehicle leaving case. The velocity profile is represented as a time series $\{v_{i+1}(t_0)$, ..., $v_{i+1}(t_c)$,..., $v_{i+1}(t_{\text{end}})\}$, where $v_{i+1}(t_c)$ denotes the time that vehicles $i+1$ and $i-1$ are reconnected. We call the same velocity profile part of all the profiles, $\{v_{i+1}(t_0), ..., v_{i+1}(t_c)\}$, *velocity accelerating profile (VACP)* and call the subsequent velocity profile part based on the distance information obtained from vehicle i-1 after the reconnection, $\{v_{i+1}(t_c), ..., v_{i+1}(t_{\text{end}})\}$, *velocity adjusting profile (VADP)*. Note that VACP only depends on the velocity of verbile $i+1$ while VADP depends on both the velocity verbile $i+1$ and its distance from its preceding vehicle.

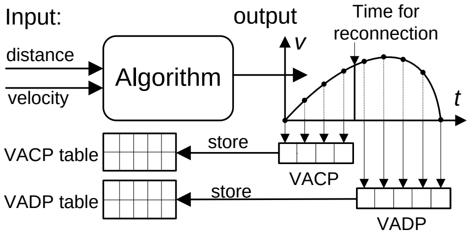
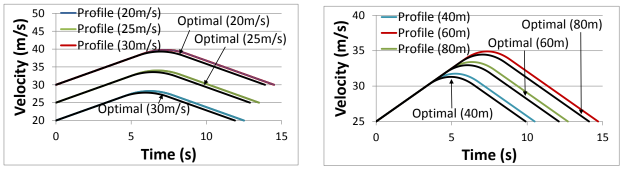


Figure 4: Process of building the VACP and VADP profiles.

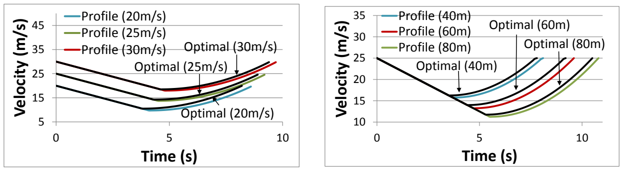
We first build VACP and VADP by profiling. We store the VACP for each current velocity input and VADP for each inputs of current velocity and distance into the VACP table and the VADP table, respectively. To obtain the whole table of VADP, we conduct the profiling by enumerating all possible inputs. In particular, we vary the distance from $d_{\min}$ to $d_{\max}$ and vary the velocity from $v_{\min}$ to $v_{\max}$. Once the tables are built, each vehicle does not need to update the table.

To increase the time efficiency for velocity determination in the platoon recovery in vehicle dynamics, the VACP table and the VADP table are stored in each vehicle when it joins the platoon. A vehicle receives these tables from its preceding vehicle after it joins the platoon. The VACP table and VADP table are kept in each vehicle's storage. Since the vehicle's velocity follows a range of values where that range is static for a time, once the table is built, each vehicle does not need to change the VACP table and VADP table anymore. To determine the velocity, vehicle $i+1$ first checks the VACP table only based on its current velocity, and then changes its velocity according to the velocity profile in the VACP table. Once vehicle $i+1$ is connected with vehicle $i-1$ and gets the distance information $d_{i-1,i+1}$, it checks the VADP table according to its velocity and the distance $d_{i-1,i+1}$ and then changes its velocity according to the velocity profile in the VADP table. Then, vehicle $i+1$ notifies vehicle $i+2$ that it is going to speed up according to the VADP table. To maintain the connectivity between vehicle $i+1$ and vehicle $i+2$, vehicle $i+2$ increases its velocity based on the VADP table as vehicle $i+1$ does. Vehicle $i+2$ also sends notification its following vehicle $i+3$. This procedure goes on for other following vehicles until last vehicle n receives the notification and increases its velocity.

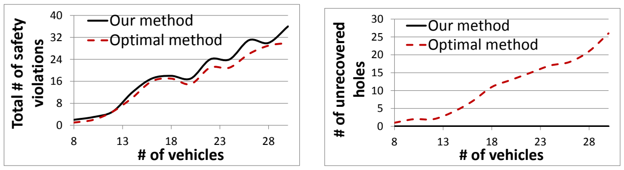
As for the vehicle entering case, because the distance information $d_{i-1,i+1}$ is always available to vehicle $i+1$, we do not need to decompose the profile. In similar way, we store all the possible profiles in a table, called *entering velocity profile* (*EVP*) table. The building process of the EVP table is similar to that of the VACP table and the VADP table. We use the constraint (3) to output the velocity profiles given



(a) Profiles with different velocity inputs (leaving)
(b) Profiles with different distance inputs (leaving)



(c) Profiles with different velocity inputs (entering)
(d) Profiles with different distance inputs (entering)



(e) Safety violations
(f) Unrecovered holes

Figure 5: Comparison of the effectiveness of the platoon maintenance methods.

different distances and velocities. Then, we store all the velocity profiles in the (EVP) table in each vehicle in the same way as the VACP table and VADP table. To determine its velocity when vehicle i joins the platoon, vehicle $i+1$ changes its velocity according to the velocity profile in its EVP table. Then, vehicle $i+1$ sends a notification to vehicle $i+2$ that it is going to deaccelerate with a certain velocity. To maintain the inter-vehicle safety distance from vehicle $i+1$, vehicle $i+2$ decreases its velocity. Also, vehicle $i+2$ notifies its following vehicle $i+3$. This procedure continues to other following vehicles until last vehicle n receives the notification and changes its velocity accordingly.

4 PERFORMANCE EVALUATION

In this section, we compare the performance of our distributed platoon network with several state-of-the art methods by conducting both simulation and real-world experiment. In the simulation, we compared different methods using MatLab.

4.1 Simulation

In the simulation, we had one leader vehicle and 30 follower vehicles, each of which was equipped with a short range (80 meters) communication device (IEEE 802.11b). The velocity of each vehicle was changed from 8m/s to 30m/s, and according to the traffic policy [19], the safety distance between vehicles was varied from 47.5m to 80m. Since the communication range of each vehicle is larger than the safety distance but smaller than the twice of the safety distance, the communication range of each vehicle covers at most two vehicle, i.e., its two neighboring vehicles, and hence each vehicle can only communicate with its neighboring vehicles. In a velocity profile, each vehicle changes its velocity at every 0.1 second [19]. In the following, we compare our method with previous method in following aspect of platoon maintenance explained below.

Platoon maintenance. For this case, we chose the method described in [19] for doing comparison, in which the distance information between the accelerating vehicle and its preceding vehicle is assumed to be known. Here, we used Matlab for this test.

We compared the effectiveness and time-efficiency of our platoon maintenance method upon vehicle entering or leaving with the an existing method [19] that provides the optimal solution with the assumption that each vehicle has the distance information from the preceding vehicle, which however is not always true. We tested two scenarios: vehicle joining and vehicle leaving. We first suppose this assumption is true for the method in [19] to show the effectiveness of our method compared to this optimal method. Then, we remove this assumption to show the drawback of the method in [19].

Figures 5(a), 5(b), 5(c), and 5(d) respectively show the velocity profiles for an entering vehicle and a leaving vehicle with different velocity and distance inputs. We find that in both leaving and entering scenarios, our method has almost the same velocity profiles with the optimal solution. As mentioned before, our method can maintain the platoon network connectivity without any distance information, since it keeps a collection of pre-calculated velocity profiles, which are almost the same to the optimal velocity profile. Fig. 5(e) compares the total number of safety violations of the two methods during the whole simulation time, which demonstrates that our method has almost the same performance with the optimal solution for ensuring vehicles' safety. The performance of our method closely reaches the performance of the optimal solution in the aspect of safety violations, because the total number of safety violations is determined by the velocity profiles of the vehicles, and the velocity profiles calculated by our method is very close to the optimal velocity profiles. Then, we removed the assumption that each vehicle has the distance information from the preceding vehicle in node leaving. Without this information, the optimal method cannot recover the connectivity holes. Fig. 5(f) compares the total number of unrecovered holes of the two methods during the simulation time. We find that the optimal method has more unrecovered holes than our method and our method can always recover the holes. This is because that both methods lack the distance information when some vehicles leave from the platoon, but our method can recover the holes even when the distance information is unavailable, while the optimal method cannot.

5 RELATED WORK

Vehicle networks. During these years, many works have been proposed to improve the performance of vehicle networks in controlling traffic congestion [13–15] or decreasing packet drop rate [6, 8, 11]. Most of these works consider a centralized platoon network, which can only support a limited number of vehicles in reality due to the leader vehicle's limited communication capability and cannot support vehicle safety. In contrast to the centralized platoon network, our proposed decentralized platoon network can support more vehicles with lower packet drop rate and packet transmission delay, which increase vehicle safety.

Velocity control. A large set of research works [2, 5, 9] have been proposed to increase the stability of connected vehicles for platoons. Desjardins and Chaib-draa [2] investigated the Cooperation Adaptive Cruise Control (CACC) in terms of line following behavior. They presented a vehicle architecture with its automatic cruise control (ACC) subsystem, and used back propagation neural network to control the longitudinal distance of all vehicles. However, their method cannot perform well with dynamic behaviors of vehicles. Another work [5] proposes the sample-data control scheme with the consideration of sensor failures and it uses feedback controllers that can stabilize the CACC system. Morbidi *et al.* [9] presented two decentralized optimal strategies for CACC system. First, a quadratic regular is synthesized and stability is enforced by the controller's feedback and forward gain in presence of disturbances. Second, desired group behavior and string stability are achieved by the compensator blending model. However, they did not consider communication latency, which affects collision avoidance. In our method, we consider decentralized vehicle control in terms of dynamic behaviors of vehicles and communication latency.

6 CONCLUSIONS

The centralized platoon maintenance method cannot quickly maintain a platoon in vehicle dynamics and hence cannot provide high vehicle safety due to the limited communication range and high overhead of the leader vehicle. To overcome this problem, we suggested to use a decentralized platoon maintenance method that only requires the communication between neighboring vehicles. However, such a method brings about the challenge of maintaining the connectivity of platoons in vehicle dynamics with the absence of the distance information in vehicle disconnection. To handle this challenge, we conducted profiling study and observed that the velocity profiles of succeeding vehicles of leaving vehicles are almost the same at the beginning regardless of their current velocities and distances to preceding vehicles. By leveraging this observation, we proposed a velocity control strategy, in which velocity profiles are maintained in each vehicle to recover the platoon connection upon vehicle joining and leaving without the inter-vehicle distance information. Thus, a platoon can be quickly and autonomously maintained to enhance vehicle safety. Our simulation results show that without inter-vehicle distance information the decentralized platoon maintenance method shows similar performance as the centralized method in terms of low recovery time and less number of safety violations. In our future work, we will further consider different lengths of vehicles, multiple vehicles' randomized movements in our design.

ACKNOWLEDGEMENTS

This research was supported in part by U.S. NSF grants ACI-1719397 and CNS-1733596, and Microsoft Research Faculty Fellowship 8300751.

REFERENCES

[1] Arturo Davila, Eduardo del Pozo, Enric Aramburu, and Alex Freixas. 2013. *Environmental Benefits of Vehicle Platooning.* Technical Report. SAE Technical Paper.

[2] Charles Desjardins and Brahim Chaib-draa. 2011. Cooperative adaptive cruise control: A reinforcement learning approach. *IEEE Trans. on ITS* 12, 4 (2011), 1248–1260.

[3] Ahmed Elbagoury, Moustafa Youssef, Dongyue Xue, and Eylem Ekici. 2013. Demo v(t) CSMA: A Link Scheduling Algorithm in Wireless Networks with Improved Delay Characteristics.. In *Proc. of Mobicom.*

[4] Ali Ghasemi, Reza Kazemi, and Shahram Azadi. 2013. Stable Decentralized Control of a Platoon of Vehicles With Heterogeneous Information Feedback. *IEEE Trans. on VT* 62, 9 (2013), 4299–4308.

[5] Ge Guo and Wei Yue. 2014. Sampled-data cooperative adaptive cruise control of vehicles with sensor failures. *IEEE Trans. on ITS* 15, 6 (2014), 2404–2418.

[6] C Lei, EM van Eenennaam, W Klein Wolterink, G Karagiannis, G Heijenk, and J Ploeg. 2011. Impact of packet loss on CACC string stability performance. In *Proc. of ITST.* 381–386.

[7] Shengbo Li, Keqiang Li, Rajesh Rajamani, and Jianqiang Wang. 2011. Model predictive multi-objective vehicular adaptive cruise control. *IEEE Trans. on CST* 19, 3 (2011), 556–566.

[8] Zhuozhao Li and Haiying Shen. 2016. Learning Network Graph of SIR Epidemic Cascades Using Minimal Hitting Set based Approach. In *Proc. of ICCCN.*

[9] Fabio Morbidi, Patrizio Colaneri, and Thomas Stanger. 2013. Decentralized optimal control of a car platoon with guaranteed string stability. In *Proc. of ECC.* 3494–3499.

[10] Chenxi Qiu, Haiying Shen, Ankur Sarker, Vivekgautham Soundararaj, Mac Devine, and Egan Ford. 2016. Towards Green Transportation: Fast Vehicle Velocity Optimization for Fuel Efficiency. In *Proc. of IEEE CloudCom.*

[11] Ankur Sarker, Chenxi Qiu, and Haiying Shen. 2016. A Decentralized Network with Fast and Lightweight Autonomous Channel Selection in Vehicle Platoons for Collision Avoidance. In *Proc. of MASS.*

[12] Ankur Sarker, Chenxi Qiu, Haiying Shen, Andrea Gil, Joachim Taiber, Mashrur Chowdhury, Jim Martin, Mac Devine, and AJ Rindos. 2016. An Efficient Wireless Power Transfer System To Balance the State of Charge of Electric Vehicles. In *Proc. of ICPP.*

[13] Björn Scheuermann, Wenjun Hu, and Jon Crowcroft. 2007. Near-optimal co-ordinated coding in wireless multihop networks. In *Proc. of CoNEXT.* ACM, 9.

[14] Björn Scheuermann, Matthias Transier, Christian Lochert, Martin Mauve, and Wolfgang Effelsberg. 2007. Backpressure multicast congestion control in mobile ad-hoc networks. In *Proc. of CoNEXT.* ACM, 23.

[15] Haiying Shen, Shenghua He, Lei Yu, and Ankur Sarker. 2017. Prediction-based Redundant Data Elimination with Content Overhearing in Wireless Networks. In *Proc. of IEEE PerCom.*

[16] Christoph Sommer, Stefan Joerer, Michele Segata, Ozan Tonguz, Renato Lo Cigno, and Falko Dressler. 2013. How shadowing hurts vehicular communications and how dynamic beaconing can help. In *Proc. of Infocom.* 110–114.

[17] Bart Van Arem, Cornelie JG Van Driel, and Ruben Visser. 2006. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Trans. on ITS* 7, 4 (2006), 429–436.

[18] Li Yan, Haiying Shen, Juanjuan Zhao, Chengzhong Xu, Feng Luo, and Chenxi Qiu. 2017. CatCharger: Deploying Wireless Charging Lanes in a Metropolitan Road Network through Categorization and Clustering of Vehicle Traffic. In *Proc. of INFOCOM.*

[19] Kyongsu Yi and Young Do Kwon. 2001. Vehicle-to-vehicle distance and speed control using an electronic-vacuum booster. *JSAE review* 22, 4 (2001), 403–412.