Multi-Vehicle Merge in Adaptive Decentralized Emergent Behavior PlaTooning

Shesha Sreenivasamurthy
Univ. of California, Santa Cruz
shesha@ucsc.edu

Katia Obraczka
Univ. of California, Santa Cruz
katia@soe.ucsc.edu

***Abstract*—In this paper, we describe the design, implementation and evaluation of a novel multi-vehicle merge maneuver for a decentralized platooning system named Adaptive Decentralized Emergent-behavior PlaTooning's (ADEPT), which is inspired by the "emergent behavior" of some processes found in nature. Performance evaluation results considering a variety of maneuvering scenarios show that the proposed emergent-based approach to multi-vehicle merging yields lower overall maneuver latency compared to centralized platooning merge. We also show that the proposed emergent-based multi-vehicle merge results in lower overhead when compared to centralized merge, and is thus able to more efficiently use network resources. Finally, due to its emergent-based bottom-up approach to platooning, maneuvers are significantly less complex to implement since they are based on a relatively small set of simple rules that can be used by all maneuvers.**

***Index Terms*—autonomous-platooning, decentralized, merge, bio-inspired, emergent**

I. Introduction and Background

According to the U.S. Department of Transportation (USDOT), *platooning* is defined as "a coordinated operation of two or more vehicles via cooperative adaptive cruise control (CACC)" [1]. Platooning provides a number of benefits, including improved fuel efficiency [2], road capacity [3], [4], and road safety. Early research on platooning and automated vehicular systems such as PATH [5], SARTRE [6] and Energy-ITS [7], mainly focused on steady-state cruising, which regulates speed and inter-vehicular spacing assuming the platoon had already been formed. "On-the-fly" platoon establishment and maneuvering such as formation and dissolution during the course of the trip has been less explored. Only recently, with the emergence of Cooperative Intelligent Transport Systems, has started to receive attention from researchers and practitioners [8], [9].

Generally, platooning systems can be classified as either centralized or decentralized. Centralized platooning is characterized by the presence of a *leader* responsible for coordinating all the maneuvers. This suffers from single point of failure, platoon length limitation and longer maneuver time as maneuvers need to be serialized [10], [11]. While decentralized platooning tries to mitigate the shortcomings of centralized platooning systems, it has its own limitations. Traditional decentralized platooning, also known as *deliberate decentralized platooning*, requires 1-to-1 communication among the maneuvering vehicles, thereby increasing communication overhead and demanding more from the underlying network [12], [13]. Additionally, each individual maneuver requires developing a specific protocol which increases the complexity of the overall platooning system.

To mitigate the drawbacks of traditional decentralized platooning, in [9] we proposed a decentralized platooning approach inspired by the emergent behavior of biological systems, such as ants and termites. We showed how maneuvers such as `JOIN` and `EXIT` can be performed by adopting strategies that "emerge" from basic, common rules.

In this paper, we present ADEPT's multi-vehicle `MERGE` maneuvering, executed when platooning vehicles change lanes and lanes merge. Multi-vehicle merge maneuvers proposed for centralized platooning [14] will not work in decentralized systems since they rely on a *leader* to coordinate maneuvers. While a few decentralized platooning approaches have been proposed, most of them focus on tail and single-vehicle merge [13], [15]. ADEPT intends to fill this important gap - its emergent-based multi-vehicle merging maneuver allows multiple vehicles to merge into a platoon at different positions, i.e., tail and in the middle of the platoon. We describe the design, implementation and evaluation of ADEPT's multi-vehicle `MERGE` maneuver. Our simulation results using a diverse set of maneuvering scenarios demonstrate that, when compared to centralized platooning, ADEPT's emergent-based approach to multi-vehicle merging ma-

978-1-6654-5530-5/23/$31.00

neuvers yields lower overall maneuver latency in most scenarios. It also incurs lower network overhead when compared to centralized merge and thus is able to use the network more efficiently. Additionally, ADEPT's emergent-based approach, which uses a relatively small set of simple rules, results in significantly lower maneuver implementation complexity.

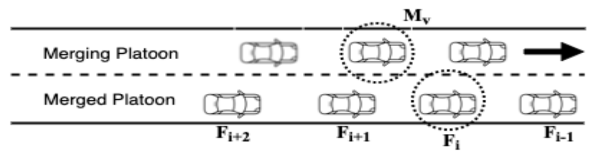
II. ADEPT'S MERGE MANEUVER

In this section, we describe in detail our design of ADEPT's multi-vehicle merge. We also describe the merge maneuver of a centralized platooning system which we will use as a baseline when evaluating ADEPT's merge performance. We start by defining some basic terminology, listing our assumptions and lastly, describing our system in detail.

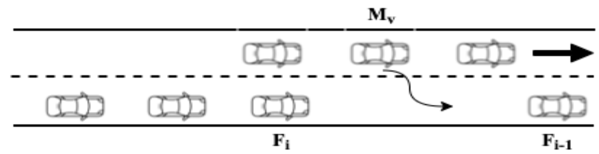
A. Terminology and System Model

We consider two platoons – The platoon from which vehicles intend to leave is called the `merging-platoon` and the platoon to which vehicles intend to merge is called the `merged-platoon`. One of the main steps during a merge maneuver is identifying the target position in the merged-platoon. As illustrated in Figure 1a, if the *merging-vehicle* M_v wants to merge, the target position for the merge is in front of vehicle F_i in the merged-platoon which is the closest and is in front of the `merging-vehicle` M_v. F_i will be the new `following-vehicle` after the merge. F_i creates a gap with F_{i-1} (as explained in Sections II-D and II-E) so that M_v merges in between F_i and F_{i-1} as shown in Figure 1b. Note that the closest rear vehicle F_{i+1} in the merged-platoon is not chosen because the position of M_v relative to F_i may be such that merging is not possible between F_{i+1} and F_i, even after F_{i+1} creates a gap with F_i as depicted in Figure 1c.

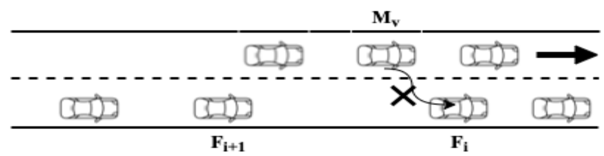
Alternatively, we could decelerate and reposition M_v to accommodate merging between F_{i+1} and F_i, which is more complex compared to decelerating only F_i as the former involves coordination among two vehicles while the latter involves only one. We reposition M_v when either F_{i-1} or both F_i and F_{i-1} are not present in the system. In the former case, i.e., the closest vehicle in the target platoon is the lead vehicle, M_v re-positions itself behind F_{i+1} because F_i becomes the first vehicle i.e., F_0, and it cannot create any gap with the vehicle in front of it. In the latter case, i.e., M_v is in front of the first vehicle in the target platoon, M_v will reposition itself behind F_{i+2}. In both the cases after repositioning, M_v will attempt to re-identify its new target merge position. The new following-vehicles will be F_{i+1} and F_{i+2}, in the former and latter cases, respectively.



(a) Platoons before the merge



(b) Merge gap creation



(c) Merge collision

Fig. 1: The merge maneuver

In the case the merged-platoon is a single-vehicle platoon, F_0 cannot create a gap in front of it for M_v to merge, nor M_v can reposition behind a vehicle that can create a gap for it to merge. In this case, M_v positions itself at a safe distance behind F_0 and merges into the merged-platoon. We do not accelerate M_v as it may be not possible to do so because of the existence of a vehicle in front of M_v in the merging-platoon. Though it is possible to accelerate M_v if it is the first vehicle in the merging-platoon, we chose not to do so for implementation simplicity.

In ADEPT, if the merging-platoon and the merged-platoon are in the same lane and M_v is behind the last vehicle of the merged-platoon we term the maneuver as `JOIN`. If the merging-platoon and the merged-platoon are in adjacent lanes, we term the maneuver as `MERGE`.

B. Assumptions

Our design makes the following assumptions: **(1)** The merging-platoon speed is greater than or equal to the merged-platoon speed. Otherwise, M_v should accelerate to match the speed of the merged-platoon. This may not be possible if M_v is not the first vehicle in the merging-platoon; **(2)** The platoon id is globally unique and is set by the first vehicle to join the platoon as described in [9]; **(3)** All vehicles in the system know and advertise their position through their beacon messages; **(4)** Vehicles do not engage in malicious behavior and thus do not generate malicious messages; **(5)** All vehicles are equipped with a controller capable of guiding the vehicle from one lane to another to complete the merge maneuver.

C. ADEPT's Communication

ADEPT uses a decentralized *emergent behavior* based platooning approach where participants communicate indirectly via the environment mimicking emergent systems in nature, such as ants and termites. Therefore, in ADEPT, all communication is done via broadcasting in lieu of direct, unicast communication.

In a platooning system, vehicle information, such as *vehicleId, speed* and *position*, required to maintain constant inter-vehicular spacing is periodically broadcasted as beacon messages. In ADEPT, additional information required for platoon maneuvers, highlighted gray in Figure 2, is piggybacked in the beacon messages. This eliminates the need for additional messages and results in reduced overall communication overhead as illustrated by our experimental results in Section IV. Note that, in our current ADEPT implementation, beacon messages are transmitted every $100ms$.

To cope with the dynamics of platooning systems (e.g., high mobility, communication channel unreliability, etc), vehicles only react after they receive a message for a predetermined number of times. This is a configurable parameter which can be pre-configured based on the expected dynamics of the platooning system, the operational environment and the underlying communication infrastructure.

D. ADEPT's Emergent Rules

Traditional platooning systems, also known as *Deliberate Systems* [16], usually adopt a top-down approach, where high-level objectives, in this case, maneuvers, are defined and then workflows (including message exchange) specific to each maneuver are developed. ADEPT draws inspiration from nature and adopts a bottom-up approach. The proposed *Emergent System*, also known as *Biologically-Inspired System* [17], first lays out basic rules of interaction, which vehicles use to carry out platooning maneuvers.

Below, we introduce the set of emergent rules that each vehicle in ADEPT follows to execute maneuvers in general and multi-vehicle merge in particular. Section II-E describes how these rules are used to execute ADEPT's multi-vehicle merge. The notation used in the description of the rules (see also Figure 1) is as follows: d is the minimum inter-vehicle gap, S_{Mv} is the merging-vehicle's speed, S_p is the merged-platoon's speed, l_{Mv} is the length of M_v, G_{Fi} is the longitudinal-gap between M_v and F_i, and l_{Fi} is the length of vehicle F_i.

- **Gap Maintenance (R1):** This rule is used by all vehicles to maintain constant inter-vehicle gap by accelerating when gap with the preceding vehicle $> (d + \delta)$ and decelerating when gap $< (d + \delta)$.
- **Speed Match (R2):** Used by M_v to match its speed with merged-platoon by decelerating while $S_{Mv} > (S_p + \delta)$.
- **Target Vehicle Identification (R3):** Used by M_v to determine F_i in the merged-platoon, i.e., the first vehicle that is longitudinally in front of M_v in the merged-platoon.
  - If F_i is a one-vehicle platoon, brake until $G_{Fi} > d + l_{Mv}$ and execute R6
  - If F_i is the first vehicle in multi-vehicle platoon, position behind the second vehicle of the merged-platoon and reinitiate R3.
  - Else execute R4.
- **Gap Wait (R4):** This rule is used by M_v to wait for F_i to create gap by adding F_i's ID in its beacon and waiting until F_i is at a safe distance behind it.
- **Gap Create (R5):** Rule used by F_i to create gap for M_v to merge by setting d, the inter-vehicle gap to be equal to $2d + l_{Mv} + G_{Fi} + l_{Fi}$ as long as it senses its ID in M_v's beacon. This gap is large enough for M_v with length l_{Mv} to merge safely.
- **Merge (R6):** This rule is used by M_v to perform the lane change.
- **End (R7):** Used by M_v to end the maneuver by not specifying F_i's ID in its beacon so that F_i can reset d to its original value that it modified in **R5**.

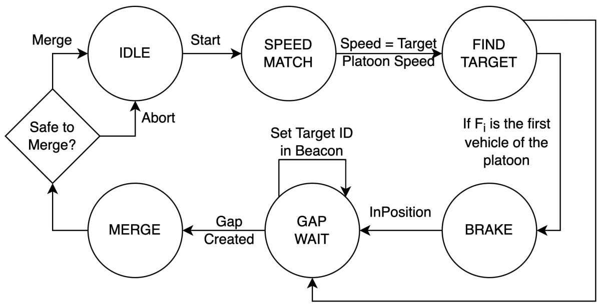
E. ADEPT's Emergent Merge

The state transition diagrams describing the merge maneuvers for vehicles M_v and F_i (see Figure 1) are shown in Figures 3a and 3b, respectively.

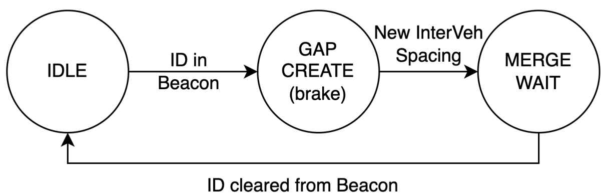
- In the `IDLE` state, all platoon vehicles maintain a minimum inter-vehicle gap of d using **R1** This is achieved by the vehicle's controller (in our current implementation, the PLOEG controller as described in Section III).
- When M_v is ready to carry out the merge, it moves to the `SPEED-MATCH` state to match the speed of the target platoon using **R2**. The speed of the target platoon is extracted from the beacon message from one of the merged-platoon members.
- When M_v is in sync with the speed of the merged-platoon, it transitions to the `FIND-TARGET` state and uses **R3** to identify F_i, the nearest vehicle in the merged-platoon in front of it. The beacon message from F_i contains the required *length* and *position*

vehicleId (u64)	speed (float)	pos-x (float)	pos-y (float)	veh-len (u8)	veh-role (u8)	laneId (u8)	mergeeId (u8)	controller (u8)	platoonId (UUID)

Fig. 2: ADEPT's Beacon Message



(a) M_v merge state transition



(b) F_i merge state transition

Fig. 3: ADEPT Merge State Transition Diagrams

information. M_v identifies F_i by maintaining a sorted list of positions of all the nearby merged-platoon vehicles.

- Once M_v identifies F_i as the target vehicle, its own ID (mergeeId) is set in its beacon message and M_v moves to the `GAP-WAIT` state where it uses **R4** to constantly check the position of F_i, which is advertised in F_i's beacon message.
- When F_i senses its ID in M_v's beacon message, it moves to the `GAP-CREATE` state and uses **R5** to create the required gap with F_{i-1}. As specified in **R5**, the gap is set as the minimum inter-vehicle gap whereby the vehicle's PLOEG controller (see Section III) slows down the vehicle until the appropriate gap with F_{i-1} is created. Once in the `GAP-CREATE` state, F_i processes beacons only from M_v and maintains the new gap while its ID is present in M_v's beacon message.
- M_v constantly monitors its environment by listening to beacon messages sent by nearby vehicles. Distance between M_v and other vehicles is determined using the position information it receives in the beacon message from other vehicles. The merge is deemed safe when M_v does not sense any other vehicle within the merging distance in the merged-platoon for a pre-determined period of time. The controller capable of guiding the vehicle from one lane to another is used to complete the merge maneuver.
- M_v stops specifying F_i's ID in its beacon message using **R7** and moves to the `IDLE` state either when it successfully completes the merge maneuver using **R6** or when it times out waiting for safe condition to merge. In the latter case, the merge operation is aborted and may be retried by M_v later.
- When F_i no longer senses its ID in M_v's beacon message, it transitions to the `IDLE` state and re-positions according to the original inter-vehicle gap. All vehicles in `IDLE` state maintain a minimum inter-vehicle gap with their preceding vehicle using **R1**.

F. Centralized Merge Maneuver

For our comparative evaluation of ADEPT's merge (see Sections III and IV), we implemented the merge maneuver for the centralized platooning system described in [18]. A brief description of the centralized merge is provided below.

- M_v informs the leader of the merging-platoon its intention to exit. If M_v happens to be the leader, it transfers leadership to the second vehicle. If M_v is not able to get permission or transfer leadership, it aborts and retries after some random wait.
- Next, M_v requests the leader of the merged-platoon permission to merge. If it gets the go-ahead, the merged-platoon leader will also send the target merge position to M_v. The merge position is denoted by the vehicle id (F_i) behind which M_v should position itself ahead of the merge. F_i is trivially determined by the merged-platoon leader as it knows the positions of all its members and also the position of M_v that it receives in the request.
- If M_v is not given permission to merge, it revokes its intention to exit with the merging-platoon leader. If M_v was the leader of its original platoon, it continues as a single vehicle platoon and retries.
- As it adjusts its speed ahead of the merge, M_v constantly checks its current position against F_i's as reported in F_i's beacon message. It informs merged-platoon leader when it is behind F_i, ready to merge.
- The merged-platoon leader instructs F_i to create the appropriate gap for M_v to merge. F_i creates the gap

TABLE I: Simulation Parameters

	Parameter	Value
comm	TX pwr (centralized)	20 dBm
	TX pwr (emergent)	10 dBm
	Epoch (δt)	0.1s
mobility	Platoon size	Varying
	Desired gap	15m
	$MAXBOFF$	3s
	h	0.5s
	L_i	4m
	r	2m
	k_p, k_d, k_{dd}	0.2, 0.7, 0
	τ	0.5s

by braking and informs the leader when done. The merged-platoon leader then informs M_v that it is safe to merge.

- M_v merges into the merged-platoon and informs leaders of both platoons when the merge completes.
- Leaders of both platoons send updated platoon information to all their members.

III. Evaluation

In this section we describe the experimental methodology we used to evaluate ADEPT's merge maneuver.

A. Simulation Environment

We evaluate ADEPT's merge maneuver by comparing its performance against the centralized merge described in Section II-F). We conducted our experiments using the PLEXE 2.1 simulator [10], which in turn uses SUMO [19] for simulating vehicular mobility, and OMNeT++ [20] to simulate the underlying communication network. Table I summarizes our simulation parameters.

In our experiments, vehicles involved in centralized maneuvering support ACC and CACC using a centralized leader-predecessor controller [21], and vehicles in ADEPT use ACC and the PLOEG decentralized controller [22]. Due to the unavailability of path-planning controllers in our simulation environment, vehicles change lanes instantaneously, i.e., the "jump" from one lane to another when they are clear to merge in lieu of gradual lane change. We assume all vehicles are identical in terms of their physical characteristics (e.g., length, weight, power) and controllers, and when a vehicle exits the platoon, we assume that it exits the system (e.g., the freeway). In this paper, our experiments simulate flat roads with ideal dry weather conditions and assume that all vehicles are autonomous and can communicate with each other.

B. Scenarios

We evaluate six different scenarios involving multiple vehicles, multiple platoons and different variations of the merging maneuver, including vehicles merging the platoon at the end, vehicles merging from one platoon to another and vehicles exiting the freeway, all happening simultaneously. Since it is impractical to test all possible scenario variations, we selected a diverse set that aims at representing most real-world merge settings. In all scenarios tested, vehicles in the merging-platoon and merged-platoon are traveling in two adjacent lanes. All vehicles of a platoon travel in the same lane one behind the other. The scenarios used in our experiments are as follows – **(1)** This is the simplest scenario that involves merging of a single vehicle from one platoon to another. The merging-platoon consists of 2 vehicles and the merged-platoon consists of 3 vehicles. The last vehicle of merging-platoon merges in front of the second vehicle in the merged-platoon. On completion, the merging platoon consists of 1 vehicle while the merged-platoon has 4. **(2)** This scenario involves two operations, i.e., merging and exit. The merging-platoon consists of 3 vehicles and merged-platoon has 4, where the first vehicles of both platoons exit the freeway, forcing leadership transfer in case of centralized merging. Simultaneously, the last vehicle of the merging platoon merges into the merged-platoon. On completion, the merging platoon has 1 vehicle while the merged-platoon has 4. **(3)** This scenario too involves two operations, namely merging and joining the platoon at the rear. The merging-platoon has 3 vehicles and merged-platoon has 4. Two single vehicle platoons, one in the merging-platoon lane and another in the merged-platoon lane join the respective platoons at the rear. Simultaneously, the second vehicle from the merging-platoon merges into the merged-platoon. On completion, the merging platoon has 3 vehicles while the merged-platoon has 6. **(4)** This scenario is more complex compared to the previous ones, involving join, exit and merge operations. The merging-platoon has 7 vehicles and the merged-platoon 4. 2 vehicles from the merging-platoon exit the freeway while 1 vehicle merges into the merged-platoon. Additionally, 2 vehicles in the merging-platoon lane and 2 vehicles in the merged-platoon lane attempt to join the respective platoons at the rear. All the above-mentioned operations are triggered simultaneously. On completion, the merging-platoon has 6 vehicles, merged-platoon 7, while 2 vehicles exit the freeway. **(5)** This scenario illustrates multi-vehicle merging, where the whole merging-platoon merges into the merged-platoon to form one single platoon. The merging-platoon has 4 vehicles and merged-platoon 5. All merging-platoon vehicles merge into the merged-platoon one by one. On completion, the merged-platoon consists of all 9 vehicles. **(6)** This is a similar scenario to the previous one where we illustrate the merging of the whole platoon. In this case, the leaders of the two platoons in centralized merging can coordinate and create gaps in their respective platoons so that all the vehicles of merging-platoon can merge simultaneously. However, this requires a complete specialized protocol. The reason for this scenario is to bring about the trade-off between maneuver efficiency in the case of centralized merging at the cost of increased complexity and overhead.

IV. RESULTS

The results presented in this section are average of 10 runs. In each run a different random seed is used which affects the position of the vehicles and the timing of their messages. In a particular run, the same random seed is used for both ADEPT and centralized platooning. The error bars in the graphs show the standard deviations. We evaluate ADEPT's merge using the following metrics:

- **Merge Maneuver Time:** Total time taken by the system to complete a set of platooning maneuvers.
- **Communication Overhead:** Average number of data bytes sent by each vehicle.

A. *Merge Maneuver Time*

The maneuver time for different scenarios is shown in Figure 4 where the x-axis shows the 6 different scenarios described in Section III-B and the y-axis, the maneuver time in seconds. Note that in scenario (1) which involves merging of a single vehicle, the time taken by centralized merge is slightly lower (22%) than ADEPT. Recall that in ADEPT, communication happens indirectly by sensing the environment and having to wait for a pre-specified period to address the dynamics of the environment, thus the resulting increase in latency.

In scenarios (2) to (4), centralized takes considerably more time (85%, 203% and 325% respectively) as exit and join maneuvers happen in parallel with the merge maneuver in ADEPT, whereas in the centralized platooning merge, all maneuvers are serialized. The more there are concurrent maneuvers, the more efficient ADEPT's approach is when compared to centralized platooning.

In scenario (5), since all vehicles attempt to merge at the same time, all vehicles in the merged-platoon create gaps also at the same time. This enables multiple vehicles to merge in parallel in case of ADEPT and therefore leads to shorter maneuver time (13%). However, sometimes vehicles in the merged-platoon will be positioned such that it is unsafe for the merging-platoon

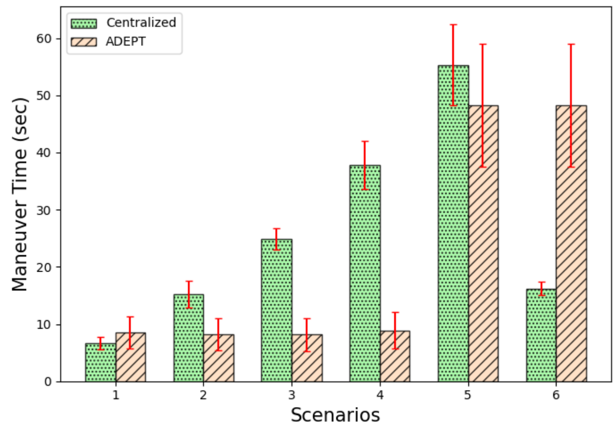


Fig. 4: Maneuver time for different scenarios

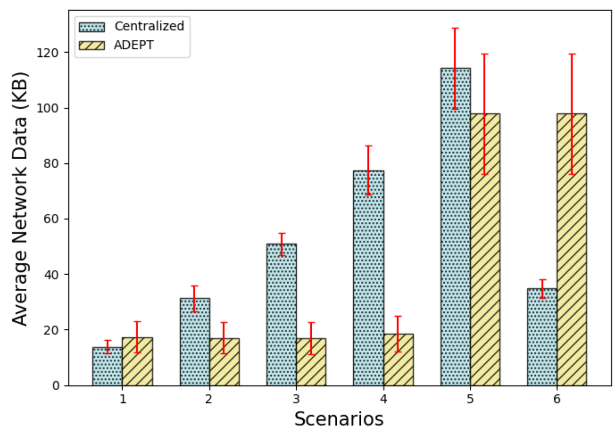


Fig. 5: Average data transmitted by each vehicle

vehicles to merge due to multi-vehicle movement. In those cases, the merge is aborted and retried later, which leads to higher maneuver time. This is evidenced by scenario (5)'s higher standard deviation.

In scenario (6), the maneuver is coordinated by the leader of both platoons in the centralized merge. As such, the merge in centralized platooning takes less than half the time compared to ADEPT (66%). So, while the centralized merge is well suited for merging entire platoons, ADEPT is better equipped to support multiple simultaneous maneuvers, in addition to providing increased resilience and robustness since it avoids single-point-of-failure issues.

B. *Communication Overhead*

Maneuvers in ADEPT leverage the broadcast nature of wireless communication and information required for platooning is piggybacked on beacon messages that are periodically broadcasted. In our current implementation, piggybacked information necessary for maneuvers adds a total of 21 bytes to the beacon message, highlighted in gray in Figure 2. Centralized platooning, on the other hand, utilizes specialized unicast messages, specific to each maneuver. Figure 5 shows the average number of bytes of data sent by each vehicle for each scenario. Even though ADEPT adds extra bytes in its beacon messages for maneuvering, vehicles transmit significantly less data (45%, 66%, 76%, 14%) overall for scenarios (2) to (5).

Additionally, centralized maneuvers require higher transmit power as the leader must communicate with the last vehicle of the platoon. The higher the transmit power, the higher the interference, likely resulting in higher data loss. This may lead to longer maneuvering time which adds to the fact that maneuvers have to be serialized. In the case of scenario (6), the leader in the centralized platooning approach coordinates merging of the whole platoon and, as previously discussed, this yields to lower maneuvering time and transmit less data (64%) compared to maneuvers in ADEPT.

V. Conclusion

This paper introduced a novel multi-vehicle merge maneuver for our emergent-behavior based decentralized platooning approach called ADEPT (Adaptive Decentralized Emergent-behavior PlaTooning) and described its design, implementation, and evaluation. Our simulation results using a diverse set of maneuvering scenarios demonstrated that, when compared to centralized platooning, ADEPT's emergent-based approach to multi-vehicle merging maneuvers yields lower overall maneuver latency in most scenarios. We also show that ADEPT demonstrates lower communication overhead compared to centralized merge and thus is able to use network resources more efficiently. Finally, due to its emergent-based, bottom-up approach to platooning, ADEPT's maneuvers are less complex to implement since they are based on a relatively small set of simple rules that can be used by all maneuvers.

References

[1] J. Loftus, "Truck Platooning The State of the Industry and Future Research Topics," *U.S. Department Of Transportation*, Jan 2018.

[2] USDOT, "Automated Vehicles: Truck Platooning ITS Benefits, Costs, and Lessons Learned: 2018 Update Report," 2018. https://www.itsknowledgeresources.its.dot.gov/its/bcllupdate/TruckPlatooning.

[3] S. E. Shladover, "Highway capacity increases from automated driving," *California PATH Program*, 2012.

[4] R. Margiotta and S. Washburn, "Simplified highway capacity calculation method for the highway performance monitoring system," tech. rep., Federal Highway Administration, 2017.

[5] J. GULDNER, S. Patwardhan, H.-S. Tan, and W.-B. Zhang, "Coding of road information for automated highways," *Journal of Intelligent Transportation System*, vol. 4, no. 3-4, pp. 187–207, 1999.

[6] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson, "Challenges of platooning on public motorways," in *17th world congress on intelligent transport systems*, pp. 1–12, 2010.

[7] S. Tsugawa, S. Kato, and K. Aoki, "An automated truck platoon for energy saving," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4109–4114, IEEE, 2011.

[8] J. Heinovski and F. Dressler, "Platoon formation: Optimized car to platoon assignment strategies and protocols," in *2018 IEEE Vehicular Networking Conference (VNC)*, pp. 1–8, IEEE, 2018.

[9] S. Sreenivasamurthy and K. Obraczka, "Towards biologically inspired decentralized platooning for autonomous vehicles," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–7, 2021.

[10] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *6th IEEE Vehicular Networking Conference (VNC 2014)*, (Paderborn, Germany), pp. 53–60, IEEE, 12 2014.

[11] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by vanet," *Vehicular communications*, vol. 2, no. 2, pp. 110–123, 2015.

[12] F. Michaud, P. Lepage, P. Frenette, D. Letourneau, and N. Gaubert, "Coordinated maneuvering of automated vehicles in platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 437–447, 2006.

[13] T. Renzler, M. Stolz, and D. Watzenig, "Decentralized dynamic platooning architecture with v2v communication tested in omnet++," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 1–6, IEEE, 2019.

[14] S. Maiti, S. Winter, L. Kulik, and S. Sarkar, "The impact of flexible platoon formation operations," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 229–239, 2019.

[15] A. Sharma, D. Agrawal, N. Roy, S. Bhichar, and R. B. Battula, "Potent-decentralized platoon management with heapify for future vehicular networks," in *International Conference on Advanced Information Networking and Applications*, pp. 387–398, Springer, 2022.

[16] H. Mintzberg and J. A. Waters, "Of strategies, deliberate and emergent," *Strategic management journal*, vol. 6, no. 3, pp. 257–272, 1985.

[17] A. Broggi, M. Cellario, P. Lombardi, and M. Porta, "An evolutionary approach to visual sensing for vehicle navigation," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 18–29, 2003.

[18] M. Segata, B. Bloessl, S. Joerer, F. Dressler, and R. L. Cigno, "Supporting platooning maneuvers through ivc: An initial protocol analysis for the join maneuver," in *2014 11th Annual conference on wireless on-demand network systems and services (WONS)*, pp. 130–137, IEEE, 2014.

[19] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, ThinkMind, 2011.

[20] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, p. 60, ICST (Institute for Computer Sciences, Social-Informatics and telecommunications engineering), 2008.

[21] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[22] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 260–265, IEEE, 2011.