23

A Reliable Wireless Protocol for Highway and Metered-Ramp CAV Collaborative Merging with Constant-Time-Headway Safety Guarantee

XUELI FAN and QIXIN WANG, Department of Computing, The Hong Kong Polytechnic University, China
JIE LIU, Harbin Institute of Technology (Shenzhen), China

To realize the grand vision of automated driving in smart vehicle *cyber-physical systems* (CPS), one important task is to support the merging of *connected automated vehicles* (CAVs) from a metered-ramp to highway. Certain safety rules must be guaranteed. However, this demand is complicated by the inherently unreliable wireless communications. In this article, we focus on the well adopted *constant-time-headway* (CTH) safety rule. We propose a highway and metered-ramp CAV collaborative merging protocol, and formally prove its guarantee of the CTH safety and liveness under *arbitrary wireless data packet losses*. These theoretical claims are further validated by our simulations. Furthermore, the simulation results also show significant improvements in the merging efficiency over other solution alternatives. Particularly, the merging success rates are more than 99% better in 11 out of 18 comparison pairs, and 0%(i.e., tied)∼ 71% better in the remaining 7 comparison pairs.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; Robotics; • **Networks** → Network reliability;

Additional Key Words and Phrases: CPS, wireless, reliability, hybrid automata, CAV

ACM Reference format:
Xueli Fan, Qixin Wang, and Jie Liu. 2023. A Reliable Wireless Protocol for Highway and Metered-Ramp CAV Collaborative Merging with Constant-Time-Headway Safety Guarantee. *ACM Trans. Cyber-Phys. Syst.* 7, 4, Article 23 (October 2023), 26 pages.
https://doi.org/10.1145/3609227

The research projects related to this article are supported in part by the Hong Kong RGC Theme-Based Research Scheme (TRS) under Grant T22-505/19-N (P0031331, RBCR; P0031259, RBCP), in part by RGC GRF under Grants PolyU 152002/18E (P0005550, Q67V) and PolyU 152164/14E (P0004750, Q44B), in part by RGC Germany/HK under Grant G-PolyU503/16 (P0001326, RAA8), in part by HKJCCT P0041424 (ZB5A), and in part by the Hong Kong Polytechnic University Fund under Grants P0045578 (CE1C), P0043884 (CD6R), P0043634 (TAB2), P0043647 (TABF), 49NZ, P0042701 (CE09), P0042699 (CE55), P0036469 (CDA8), P0042721 (ZVG0), P0013879 (BBWH), P0031950 (ZE1N), P0033695 (ZVRD), P0000157 (BBWC), P0009706 (YBXW), and 8B2V.
Authors' addresses: X. Fan and Q. Wang (corresponding author), Department of CompuEng, The Hong Kong Polytechnic University, 11 Yuk Choi Rd, Hung Hom, Hong Kong SAR, Kowloon, China; emails: xueli.fan@connect.polyu.hk, csqwang@connect.polyu.hk; J. Liu, Harbin InsEtute of Technology (Shenzhen), University Town of Shenzhen, Shenzhen, Guangdong, 518055, China; email: jieliu@hit.edu.cn.

2378-962X/2023/10-ART23
https://doi.org/10.1145/3609227

1 INTRODUCTION

To realize the grand vision of fully autonomous driving, one of the most promising directions is to first realize it in controlled environments, particularly in highways, where accesses are limited, and ***connected autonomous vehicles*** (**CAVs**) are collaborative [2, 13]. One important structure for such limited access highways is the *metered-ramp* [1, 15, 24, 41, 45], where local CAVs are stopped by a traffic signal (i.e., the *ramp meter*) before they enter the highway system via a ramp. Merging of CAVs from the metered-ramp to highway must be supported. The CAVs involved need to collaborate wirelessly to guarantee certain safety rules, as well as achieve good merging efficiency.

However, these demands are complicated by the inherently unreliable vehicular wireless communications. The solution will heavily depend on the targeted safety rule and the chosen wireless communication paradigm. A panacea solution is highly unlikely. In this article, we shall focus on a widely adopted safety rule, the ***constant time headway*** (**CTH**) safety [8, 11, 14, 30, 37, 39, 47].

Intuitively, CTH safety means at any time instance, any follower vehicle must maintain a constant temporal distance from its predecessor vehicle; i.e., the minimal spatial distance needed is proportional to the follower's current speed. For the wireless communications paradigm, there are two basic categories: ***vehicle to vehicle*** (**V2V**) and ***vehicle to infrastructure*** (**V2I**). Each has its pros and cons. Therefore, a mixed (i.e., V2I+V2V) approach, aka V2X, is gaining increasing attention recently [3, 12, 43, 46]. In this article, we shall also adopt the V2X approach: the design is centered on V2I, but V2V communications between line-of-sight neighboring CAVs along the highway lane are also exploited as an alternative for ranging.

In summary, this article shall focus on a wireless highway and metered-ramp CAV merging protocol, which guarantees the CTH safety under *arbitrary* wireless *data packet* (simplified as "*packet*" in the following) losses and achieves good merging efficiency.

Merging of vehicles is a hot research topic in smart vehicle ***cyber-physical systems*** (**CPS**). Besides the large volume of works based on the pure V2V or pure V2I wireless communications paradigm (which are to be elaborated in Section 2), V2X solutions are gaining increasing attention recently. Wang et al. [42] develop a merging algorithm using V2V and V2I communications to facilitate merging of CAVs. Virtual vehicles are mapped onto both the highway lane and the ramp to facilitate the merging of individual vehicles and platoons. Ntousakis et al. [32] propose a cooperative merging system model based on V2V and V2I communication which enables the effective handling of the available gaps between vehicles and evaluate its performance and impact on highway capacity by adopting a microscopic traffic simulator. Wang et al. [43] present a distributed consensus-based cooperative merging protocol, where ***road side unit*** (**RSU**) based infrastructure assigns sequence identifications to different vehicles based on their estimated arrival time (V2I communication), then vehicles apply distributed consensus protocol to adjust their velocity and positions in advance with V2V communications. Ahmed et al. [3] describe a freeway merge assistance system utilizing both V2V and V2I communication. The freeway merge assistance system uses an innovative three-way handshaking protocol and provides advisories to guide the merging sequence. However, the above works (including those based on pure V2V or pure V2I paradigm) do not discuss how to deal with arbitrary wireless packet losses.

There are also works focusing only on the application layer, and are independent of the underlying communication infrastructure (may it be V2V, V2I, or V2X—in another sense, this can be viewed as a more generic V2X) [6, 9, 22, 29, 31]. However, these works also assume the communication infrastructure is reliable, hence do not deal with arbitrary wireless packet losses.

Aoki et al. [5] present a safe highway and ramp merging protocol, which provides safety by using V2V communications and perception systems cooperatively and accommodates losses of wireless packets. In the protocol, packet losses can decrease traffic throughput, but cannot cause vehicle collisions. However, how to adapt this protocol to guarantee the CTH safety rule remains an open problem, as the protocol is not designed for the CTH safety rule to begin with.

In order to guarantee CTH safety rule under arbitrary wireless packet losses, we shall deploy a timeout (aka "lease" [18]) based approach. The basic idea is to properly configure certain timeout deadlines, so that if the corresponding wireless packets cannot arrive before the timeout deadlines, the distributed entities will independently reset themselves, hence implicitly reset the holistic system. Specifically, we make the following contributions.

(1) We propose a timeout-based CAV collaboration protocol for automatic highway and metered-ramp merging. We formally prove the safety (i.e., guarantee of the CTH safety rule) and liveness of our proposed protocol, even if there are arbitrary wireless packet losses.
(2) We carry out extensive simulations to further verify our proposed protocol. The results show that our protocol can always fulfill the CTH safety rule and liveness despite of arbitrary wireless packet losses.
(3) Furthermore, the simulation results also show significant improvements in the merging efficiency over other solution alternatives. Particularly, the merging success rates are more than 99% better in 11 out of 18 comparison pairs, and 0%(i.e., tied)∼ 71% better in the remaining 7 comparison pairs.

In the following, Section 2 presents related work. Section 3 formulates the problem. Section 4 proposes the protocol and formally proves its properties. Section 5 gives some important observations. Section 6 evaluates our protocol. Section 7 concludes the article and discusses future work.

2 RELATED WORK

Despite the V2X merging solutions listed in Section 1, there is a large volume of literature on purely V2V or purely V2I-based solutions.

In V2V-based approaches, Lu et al. [27, 28] propose a virtual vehicle-based approach to ensure sufficient distance for vehicles to merge into highway via a ramp. Hidas [20] classifies the merging maneuvers into "free", "forced", and "cooperative;" and studies their impact on the traffic flow, showing that "cooperative" merging, followed by "forced" merging, provides the greatest impact on the traffic flow. Xie et al. [44] develop an optimization-based ramp control strategy and a simulation platform to assess the potential safety and mobility benefits of V2V cooperative merging. Kazerooni et al. [23] and Heim et al. [19] present interactive protocol for merging, in which vehicles use both V2V communications and sensing for cooperation and safety guarantee.

In V2I-based approaches, Jiang et al. [21] use a V2I-based dynamic merge assistance method, to improve merging efficiency and safety. Letter et al. [26] present a longitudinal freeway merging control algorithm for maximizing the average travel velocity of CAVs. Raravi et al. [35] propose an approach for automatic merge control system, where an infrastructure node plans the merging sequences. Pueboobpaphan et al. [33] discuss an algorithm, where trajectories are planned with a safety zone around the ramp CAV. Adjustments based on the planning are continually relayed to the highway CAVs to accommodate the ramp CAV.

All of the above works, however, do not deal with arbitrary wireless packet losses; and how to adapt them to guarantee CTH safety for all vehicles at all time are still open problems.

There are various timeout (aka "lease") based distributed protocols [18, 38]. However, these protocols are not designed for highway and ramp merging, and neither for the CTH safety guarantee.

A less-than-two-full-page ***Work-in-Progress*** (**WiP**) abstract (not an article) of this work is published in a conference's WiP session [17]. Permission to reuse the contents of [17] are obtained (©2023 IEEE. Reprinted, with permission, from [17]). Compared to this article, the WiP abstract does not provide any intuitive explanation, narrative definition, theoretical proof, or evaluation of the proposed solution. The introduction, related work, and problem formulation are covered but very brief (about one page total, with only three references). Prototypes of Figures 1, 3–5, Definition 1, a conjecture of Theorem 1, as well as Ineq. (1), (6) have appeared in [17]. That is, this sentence occupies a unique paragraph by itself. The content of this article is also included in the PhD dissertation of the first author [16].

3 PROBLEM FORMULATION

In this section, we present the assumptions on CAV driving dynamics, describe the highway and metered-ramp merging scenario, and specify the demanded CTH safety.

3.1 Assumptions on Driving Dynamics

Vehicular driving dynamics modeling is nontrivial (interested readers can refer to [34]). Fortunately, we can make the following assumptions.

3.1.1 CAV Acceleration. We assume the CAV acceleration strategy along a straight lane is fixed. Specifically, given the initial speed (for straight lanes, we only need to discuss speed, instead of velocity) $v_{\mathrm{a}}^{\mathrm{low}}$ and the target speed $v_{\mathrm{a}}^{\mathrm{high}}$ (where $0 \leqslant v_{\mathrm{a}}^{\mathrm{low}} < v_{\mathrm{a}}^{\mathrm{high}}$, and note in this article, we assume vehicles cannot move backward), suppose currently the acceleration process has been going on for τ_{a} seconds ($\tau_{\mathrm{a}} \geqslant 0$) and has not yet finished, then the CAV's current acceleration value is fixed, and is a function of $v_{\mathrm{a}}^{\mathrm{low}}$, $v_{\mathrm{a}}^{\mathrm{high}}$, and τ_{a}. Denote this function as $\mathrm{acc}(v_{\mathrm{a}}^{\mathrm{low}}, v_{\mathrm{a}}^{\mathrm{high}}, \tau_{\mathrm{a}})$. This function implies that the current *speed* of the CAV is also a function of $v_{\mathrm{a}}^{\mathrm{low}}$, $v_{\mathrm{a}}^{\mathrm{high}}$, and τ_{a}, which can be denoted as $v_{\mathrm{a}}(v_{\mathrm{a}}^{\mathrm{low}}, v_{\mathrm{a}}^{\mathrm{high}}, \tau_{\mathrm{a}})$. This in turn implies that the total duration and distance needed to accelerate from $v_{\mathrm{a}}^{\mathrm{low}}$ to $v_{\mathrm{a}}^{\mathrm{high}}$ is a function of $v_{\mathrm{a}}^{\mathrm{low}}$ and $v_{\mathrm{a}}^{\mathrm{high}}$. We denote this duration and this distance to be respectively $\delta_{\mathrm{a}}(v_{\mathrm{a}}^{\mathrm{low}}, v_{\mathrm{a}}^{\mathrm{high}})$ and $d_{\mathrm{a}}(v_{\mathrm{a}}^{\mathrm{low}}, v_{\mathrm{a}}^{\mathrm{high}})$. Furthermore, we have

Assumption 1. *We assume the acceleration process as per* $\mathrm{acc}(v_{\mathrm{a}}^{\mathrm{low}}, v_{\mathrm{a}}^{\mathrm{high}}, \tau_{\mathrm{a}})$ *(where* $0 \leqslant v_{\mathrm{a}}^{\mathrm{low}} < v_{\mathrm{a}}^{\mathrm{high}}$*) is nonzeno (i.e.,* $\delta_{\mathrm{a}}(v_{\mathrm{a}}^{\mathrm{low}}, v_{\mathrm{a}}^{\mathrm{high}}) > 0$*) and strictly monotonic (i.e., speed will strictly monotonically increase from* $v_{\mathrm{a}}^{\mathrm{low}}$ *to* $v_{\mathrm{a}}^{\mathrm{high}}$ *over time).*

3.1.2 CAV Deceleration. Similar to the acceleration case, in this article, we assume the CAV deceleration strategy along a straight lane is also fixed. Specifically, given the initial speed $v_{\mathrm{d}}^{\mathrm{high}}$ and the target speed $v_{\mathrm{d}}^{\mathrm{low}}$ (where $v_{\mathrm{d}}^{\mathrm{high}} > v_{\mathrm{d}}^{\mathrm{low}} \geqslant 0$), suppose currently the deceleration process has been going on for τ_{d} seconds ($\tau_{\mathrm{d}} \geqslant 0$) and has not yet finished, then the CAV's current acceleration value is fixed, and is a function of $v_{\mathrm{d}}^{\mathrm{high}}$, $v_{\mathrm{d}}^{\mathrm{low}}$, and τ_{d}. Denote this function as $\mathrm{dec}(v_{\mathrm{d}}^{\mathrm{high}}, v_{\mathrm{d}}^{\mathrm{low}}, \tau_{\mathrm{d}})$. This function implies that the current *speed* of the CAV is also a function of $v_{\mathrm{d}}^{\mathrm{high}}$, $v_{\mathrm{d}}^{\mathrm{low}}$, and τ_{d}, which can be denoted as $v_{\mathrm{d}}(v_{\mathrm{d}}^{\mathrm{high}}, v_{\mathrm{d}}^{\mathrm{low}}, \tau_{\mathrm{d}})$. This in turn implies that the total duration and distance needed to decelerate from $v_{\mathrm{d}}^{\mathrm{high}}$ to $v_{\mathrm{d}}^{\mathrm{low}}$ is a function of $v_{\mathrm{d}}^{\mathrm{high}}$ and $v_{\mathrm{d}}^{\mathrm{low}}$. We denote this duration and this distance to be respectively $\delta_{\mathrm{d}}(v_{\mathrm{d}}^{\mathrm{high}}, v_{\mathrm{d}}^{\mathrm{low}})$ and $d_{\mathrm{d}}(v_{\mathrm{d}}^{\mathrm{high}}, v_{\mathrm{d}}^{\mathrm{low}})$. Furthermore, we have

Assumption 2. *We assume the deceleration process as per* $\mathrm{dec}(v_{\mathrm{d}}^{\mathrm{high}}, v_{\mathrm{d}}^{\mathrm{low}}, \tau_{\mathrm{d}})$ *(where* $v_{\mathrm{d}}^{\mathrm{high}} > v_{\mathrm{d}}^{\mathrm{low}} \geqslant 0$ *) is nonzeno (i.e.,* $\delta_{\mathrm{d}}(v_{\mathrm{d}}^{\mathrm{high}}, v_{\mathrm{d}}^{\mathrm{low}}) > 0$*) and strictly monotonic (i.e., speed will strictly monotonically decrease from* $v_{\mathrm{d}}^{\mathrm{high}}$ *to* $v_{\mathrm{d}}^{\mathrm{low}}$ *over time).*

3.2 Merging Scenario and CTH Safety Rule

Figure 1 shows the highway and metered-ramp merging scenario in a bird's-eye view. We assume a metered-ramp leads to a straight highway lane. Mathematically, the highway lane is modeled as a real number axis. The metered-ramp is modeled as a half line. The highway lane and the

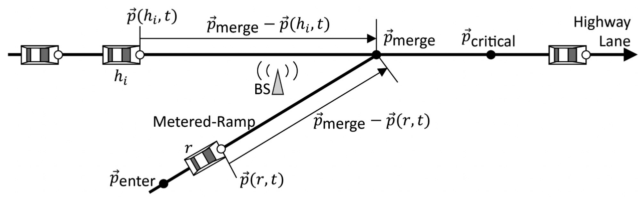


Fig. 1. The highway and metered-ramp merging scenario. $\vec{p}(x,t)$ is CAV x's location at wall clock time t. Drawn based on our intellectual property. Early version appeared in [17] Figure 1.

metered-ramp intersect at point $\vec{p}_{\text{merge}}$, which cuts the highway lane into two halves: the segment $(-\infty, \vec{p}_{\text{merge}}]$ and the segment $(\vec{p}_{\text{merge}}, +\infty)$. The metered-ramp, on the other hand, has a fixed entrance point $\vec{p}_{\text{enter}}$, which is D_r away from $\vec{p}_{\text{merge}}$ (where $D_r \stackrel{\text{def}}{=} |\vec{p}_{\text{merge}} - \vec{p}_{\text{enter}}|$ is a given configuration constant). Any CAV merging into the highway lane via this metered-ramp must first stop at $\vec{p}_{\text{enter}}$ to wait for permission to start. Typically, $\vec{p}_{\text{enter}}$ is where a physical infrastructural ramp meter (such as a red/green traffic light) is installed. But the ramp meter can also be virtual: the CAV simply stops at $\vec{p}_{\text{enter}}$ (e.g., assisted by GPS, or simple visual marks painted on the ramp at $\vec{p}_{\text{enter}}$) and waits for a wireless permission message (from certain participants of the collaborative merging) to start.

For the time being, we abstract every CAV as a point mathematically (see the ∘ dots in Figure 1), and let $\vec{p}(x,t)$ denote the location of CAV x at wall clock time t. Considerations on vehicle body length are discussed in the end of this subsection. Suppose the whole system starts at wall clock time t_0, when there are n ($n < +\infty$) CAVs driving at the speed limit v_{lim} (a given configuration constant, the maximum allowed speed on a highway lane, see discussions before Assumption 3) along the highway lane. Without loss of generality, denote the leading CAV to be h_1, which is followed by h_2, so on and so forth, till the last CAV h_n. We call h_is ($i = 1, 2, \ldots, n$) the "*highway CAVs*".

Also, at t_0, a CAV r is stopping at $\vec{p}_{\text{enter}}$ on the metered-ramp waiting for permission to start merging onto the highway lane. We call r the "*ramp CAV*". Once started, r should first accelerate as per $\text{acc}(0, v_{\text{rm}}, \tau_{\text{a}})$ (acc is defined in Section 3.1.1) to the speed of v_{rm} (a given configuration constant, the minimum speed allowed on a highway lane, see discussions before Assumption 3) and then maintains this speed to reach $\vec{p}_{\text{merge}}$. Correspondingly, Ineq. (1) is the configuration prerequisite to make this feasible:

$$d_{\text{a}}(0, v_{\text{rm}}) < D_r \stackrel{\text{def}}{=} |\vec{p}_{\text{merge}} - \vec{p}_{\text{enter}}|, \tag{1}$$

where $d_{\text{a}}(0, v_{\text{rm}})$ is the distance needed to accelerate from speed 0 to v_{rm} (see Section 3.1.1, note the corresponding time cost is $\delta_{\text{a}}(0, v_{\text{rm}})$). The duration cost for r from the start of acceleration to reaching $\vec{p}_{\text{merge}}$ hence is

$$\Delta_r \stackrel{\text{def}}{=} \delta_{\text{a}}(0, v_{\text{rm}}) + \frac{D_r - d_{\text{a}}(0, v_{\text{rm}})}{v_{\text{rm}}}. \tag{2}$$

We also give the following configuration prerequisite:

$$0 < v_{\text{rm}} < v_{\text{lim}}. \tag{3}$$

Correspondingly, once r reaches $\vec{p}_{\text{merge}}$, it will accelerate again according to $\text{acc}(v_{\text{rm}}, v_{\text{lim}}, \tau_{\text{a}})$ to the speed of v_{lim}. The location on the highway lane where r first reaches v_{lim} hence is fixed. Denote it as $\vec{p}_{\text{critical}}$ (see Figure 1).

Note we assume when the merging is completed, all CAVs on the highway shall drive at a same constant speed (specifically, v_{lim}). This is a popular practice adopted by many collaborative CAV driving schemes [10, 25, 36, 40], particularly in the large volume of literature on ***cooperative***

adaptive cruise control (**CACC**) [7, 14]. This practice prevails not only for its simplicity but also for its safety and energy efficiency [7, 10, 14, 25]. On the other hand, the ramp CAV r reaching $\vec{p}_{\text{merge}}$ with v_{rm}, the so-called minimum speed allowed on a highway lane, is a design out of caution. It covers the special case where $v_{\text{rm}} = v_{\text{lim}} - \varepsilon$, where $\varepsilon > 0$ is an arbitrarily small number.

We also assume the following about the CAVs and the road system for the time being.

Assumption 3. *The road system is equipped with V2I infrastructure. Particularly, a base station* BS *resides near* $\vec{p}_{\text{merge}}$*, which can coordinate the merging between* r *and* $h_1, h_2, \ldots, h_n$*. For the time being, we assume* BS *and the highway/metered-ramp lanes are equipped with sufficient wired infrastructure sensors, so that upon* BS*'s request, it can instantly know the distance (from* $\vec{p}_{\text{merge}}$*) and speed of any CAV (this assumption will be relaxed in Section 5).*

Assumption 4. *Each CAV is equipped with redundant ranging sensors (e.g., laser, radar, ultrasonic, computer vision, V2V communications, and human driver as the last resort), so that for any two consecutive CAVs along the highway lane, the follower CAV can instantly detect the predecessor CAV's speed (e.g., based on the follower's own speed and the relative velocity to the predecessor detected by the ranging sensors). Particularly, due to the redundancy, even if the V2V communications fail (so that the predecessor CAV cannot inform its speed via wireless packets to the follower CAV), the ranging sensing can still function correctly.*

For the above highway and metered-ramp merging scenario, we aim at guaranteeing the CTH safety [8, 11, 14, 30, 37, 39, 47] as specified in the following.

Definition 1 (CTH Safety). Suppose two vehicles (in math point abstraction) x and y are driving in the same direction along a same lane. Suppose x precedes y at time t. Denote the distance between x and y at t as $d(t)$, and y's speed at t as $v_y(t)$. We call $\delta(t) \stackrel{\text{def}}{=} d(t)/v_y(t)$ the *time headway* of y (relative to x) at t. If $\delta(t)$ is no less than a given constant $\Delta^* > 0$, aka the *desired time headway*, then we say the ordered tuple (x, y) is CTH-Δ^* safe at t. In other words, if $d(t) \geqslant v_y(t)\Delta^*$, then we say (x, y) is CTH-Δ^* safe at t.

Assumption 5. $\forall i \in \{1, 2, \ldots, n-1\}$, (h_i, h_{i+1}) *is CTH-*Δ^* *safe at* t_0.

Intuitively, suppose a lane has both a minimum speed limit $v^{\min}$ and a maximum speed limit $v^{\max}$, Δ^* should be set to $\Delta^{**} + \frac{D_0}{v^{\min}}$, where Δ^{**} is the maximum duration needed to stop a vehicle at any speed $v \leqslant v^{\max}$ using emergency braking (which could be different from the normal deceleration dec, but should be monotonic), and D_0 is the maximum vehicle body length. This way, CTH-Δ^* safety rule guarantees y will never hit x, even if x can abruptly stop at any time on the lane.

4 SOLUTION

In this section, we propose a protocol to realize the aforementioned highway and metered-ramp merging (see Section 3.2) and prove its guarantee of the CTH safety and liveness, even under arbitrary wireless packet losses.

4.1 Heuristics

The heuristics of our proposed protocol are illustrated by the automata sketches in Figure 2.

Initially, the base station BS, the ramp CAV r, and the highway CAVs h_i ($i = 1, 2, \ldots, n$) all dwell in their respective "Init" mode. Then the ramp CAV r requests permission to start the merging by sending a "MergeReq" wireless packet to BS (see event "SendMergeReq" in Figure 2(b)). If BS receives this wireless packet, it triggers the "GotMergeReq" event (see event "GotMergeReq" in Figure 2(a)). As the *action* (i.e., the handling routine) is carried out by this event, BS finds the approaching highway CAV closet to $\vec{p}_{\text{merge}}$, and names it coop (for "Cooperator"). BS then enters the transient mode of "L_0" to take further actions based on coop's distance to $\vec{p}_{\text{merge}}$. Specifically,

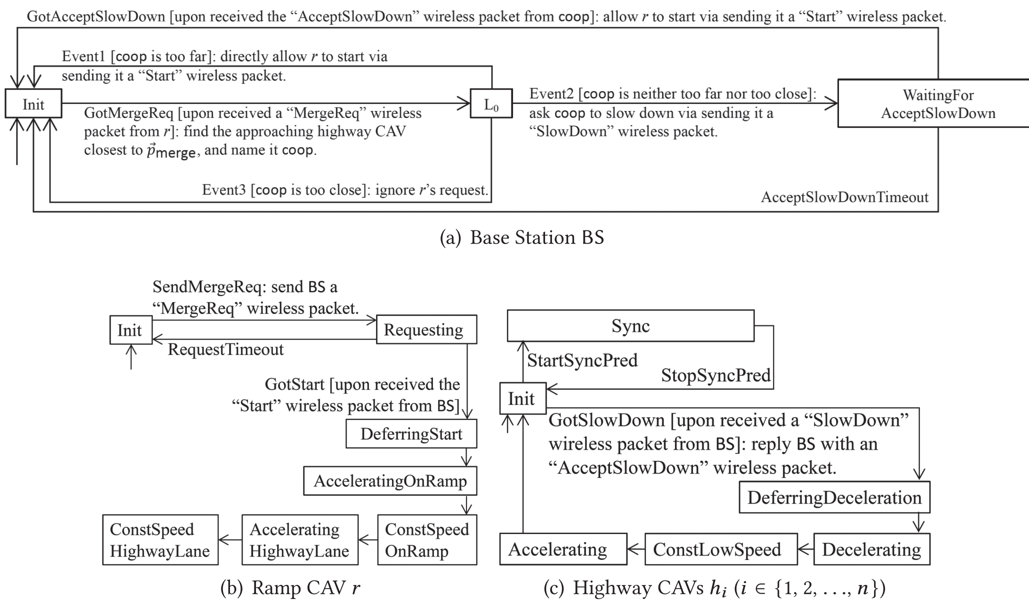


Fig. 2. Automata Sketches. Rectangles are modes, arrows between modes are events, and the arrow without source mode indicates the initial mode in the respective automata sketches. Texts in "[]" are the triggering conditions (aka *guards*) for the corresponding events; texts after the ":" are the *actions* to be carried out once the corresponding events happen.

(1) If coop is too far away from $\vec{p}_{\text{merge}}$, BS will directly allow r to start by sending it a "Start" wireless packet (see "Event1" in Figure 2(a)).
(2) If coop is too close to $\vec{p}_{\text{merge}}$, r's merge request is ignored and r has to request again in the future (see "Event3" in Figure 2(a)).
(3) If coop is neither too far away nor too close to $\vec{p}_{\text{merge}}$ (see "Event2" in Figure 2(a)), BS first sends a "SlowDown" wireless packet to coop to request it to decelerate (i.e., to yield). If coop receives this packet, it will acknowledge BS with an "AcceptSlowDown" wireless packet and start a deceleration routine (see "GotSlowDown" event in Figure 2(c)). Upon reception of the "AcceptSlowDown" wireless packet, BS will send a "Start" wireless packet to r to start its merging routine (see "GotAcceptSlowDown" event in Figure 2(a)). Upon reception of the "Start" wireless packet (see "GotStart" event in Figure 2(b)), r will start and accelerate. Once r reaches $\vec{p}_{\text{merge}}$, it will accelerate to v_{lim}, and later coop will also accelerate to v_{lim}. In addition, when a highway CAV sees its close (current distance is within a certain threshold) predecessor CAV decelerates or accelerates, it will do the same (i.e., "*synchronize*" with the predecessor, see mode "Sync" in Figure 2(c)).

For the above cases, how "far" is "too far," how "close" is "too close," and how to configure the parameters to achieve the CTH-Δ^* safety are non-trivial problems. We will clarify them in the detailed protocol design and analysis (see Sections 4.2 and 4.3).

Another challenge is the possibility of arbitrary wireless packet losses. What if the "MergeReq," "SlowDown," "AcceptSlowDown," and/or "Start" wireless packets are lost? Can the CTH-Δ^* safety still sustain? Can the CAVs still reset themselves, instead of being stuck in a mode forever? Can the CAVs still merge efficiently?

To address these concerns, we propose to deploy the "*lease*" design philosophy for distributed systems [18, 38]. A "*lease*" is an agreement on timeout, contracted since the early stage of a distributed collaboration. After the lease is contracted, if wireless packets are lost, the affected entities can reset themselves when the agreed timeout is reached (by looking at their respective local clocks, hence need no more communications). In Figure 2, nearly every mode has its timeout configuration. The exact configurations to choose are also non-trivial problems that affect the CTH-Δ^* safety, system liveness, and efficiency. The details and analysis are also elaborated in Sections 4.2 and 4.3. The efficiency is evaluated in Section 6.

4.2 Proposed Protocol

We propose our detailed protocol by expanding the automata sketches of Figure 2 with the heuristics described in Section 4.1. The resulting full-fledged hybrid automata [4] A_{BS} (see Figure 3), A_r (see Figure 4), and A_i (see Figure 5), respectively, define the protocol behaviors of the base station BS, the ramp CAV r, and the highway CAV h_i ($i = 1, 2, \ldots, n$). These behaviors are explained as follows; a symbol list is also provided in Appendix A for the reader's convenience.

Base Station BS *protocol behaviors (illustrated by hybrid automaton A_{BS} in Figure 3)*:

(1) At any time instance, the base station BS dwells in one of the following modes: "Init," "L_0," and "WaitingForAcceptSlowDown."

(2) Initially, BS dwells in the "Init" mode, and has its local clock τ's initial value set randomly from $[0, \Delta_{\text{BS}}^{\text{min}}]$ (e.g., as per uniform distribution), where $\Delta_{\text{BS}}^{\text{min}} > 0$ is a configuration constant.

(3) When dwelling in mode "Init", if a "MergeReq" wireless packet is received from the ramp CAV r, and BS has been continuously dwelling in "Init" for at least $\Delta_{\text{BS}}^{\text{min}}$ seconds (i.e., $\tau > \Delta_{\text{BS}}^{\text{min}}$), then BS triggers the "GotMergeReq" event. This event carries out the following action (see event "GotMergeReq" in Figure 3):

Step1 **IF** currently there is no highway CAV approaching BS (i.e., if $\nexists$ vehicle on highway lane segment $(-\infty, \vec{p}_{\text{merge}}]$) **THEN** set $\hat{\delta}_{\text{coop}}$ to $+\infty$.

Step2 **ELSE**

Step2.1 **IF** coop is undefined, **THEN** set coop as the current closest highway CAV approaching BS (i.e., the current vehicle closest to $\vec{p}_{\text{merge}}$ on the highway lane segment $(-\infty, \vec{p}_{\text{merge}}]$);

Step2.2 set $\hat{\delta}_{\text{coop}}$ to $|\vec{p}_{\text{merge}} - \vec{p}(\text{coop}, t_1)|/v_{\text{lim}}$, where t_1 is the current wall clock time (i.e., $|\vec{p}_{\text{merge}} - \vec{p}(\text{coop}, t_1)|$ is the current distance between $\vec{p}_{\text{merge}}$ and the coop).

After the above action, BS enters the transient mode "L_0."

(4) Mode "L_0" is a transient mode that BS cannot stay. Upon entrance to "L_0," BS immediately triggers one of the following events (see "Event1," "Event2," and "Event3," respectively, in Figure 3):

Case1 (Event1) If the highway CAV coop is too far from the merging point $\vec{p}_{\text{merge}}$, specifically, if $\hat{\delta}_{\text{coop}} \geqslant \Delta_r + \Delta^* + \Delta_1$, where

$$\Delta_1 \overset{\text{def}}{=} \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}) - \frac{d_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})}{v_{\text{lim}}}, \quad \text{(note Ineq. (3) implies } \Delta_1 > 0\text{)}, \tag{4}$$

then BS triggers "Event1." This event carries out the following sequential action: send a "Start" wireless packet (with the data payload of 0) to the ramp CAV r, telling r to start immediately (i.e., with 0 delay); set the local clock τ to 0; undefine coop.

After the above action, BS returns to mode "Init."

Case2 (Event2) If coop is neither too far nor too close to $\vec{p}_{\text{merge}}$, specifically, if $\Delta_r + \Delta^* + \Delta_1 > \hat{\delta}_{\text{coop}} > \Delta_2$, where

$$\Delta_2 \overset{\text{def}}{=} (d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) + v_{\text{rm}}(\Delta_r + \Delta^* - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})))/v_{\text{lim}}, \tag{5}$$

then BS triggers "Event2." This event carries out the following sequential action:
set δ_{defer} to $\hat{\delta}_{\text{coop}} - \Delta_2$; send a "SlowDown" wireless packet (with the data payload of δ_{defer}) to coop, telling it to slow down in δ_{defer} seconds; set the local clock τ to 0.
After the above action, BS enters mode "WaitingForAcceptSlowDown" for coop's reply.
Note we enforce the following configuration prerequisite:

$$\Delta^* < \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) < \Delta_r, \tag{6}$$

which implies $\Delta_2 > 0$, and also implies

$$\Delta_r + \Delta^* + \Delta_1 > \Delta_2, \tag{7}$$

because (6)

$$\begin{aligned} \Rightarrow \quad & (v_{\text{lim}} - v_{\text{rm}})(\Delta_r + \Delta^*) + v_{\text{lim}}\delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}) + v_{\text{rm}}\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) \\ & > v_{\text{lim}}\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) + v_{\text{lim}}\delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}) > d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) + d_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}) \\ \Rightarrow \quad & \Delta_r + \Delta^* + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}) - d_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})/v_{\text{lim}} = \Delta_r + \Delta^* + \Delta_1 \\ & > (v_{\text{rm}}(\Delta_r + \Delta^* - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})) + d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}))/v_{\text{lim}} = \Delta_2. \end{aligned}$$

Ineq. (7) ensures the guards for "Event1" and "Event2" (see Figure 3) are valid and non-overlapping.

Case3 (Event3) Otherwise, i.e., if coop is too close to $\vec{p}_{\text{merge}}$, specifically, $\hat{\delta}_{\text{coop}} \leqslant \Delta_2$, then BS triggers "Event3." This event carries out the following sequential action:
set the local clock τ to 0; undefine coop.
After the above action, BS returns to mode "Init."

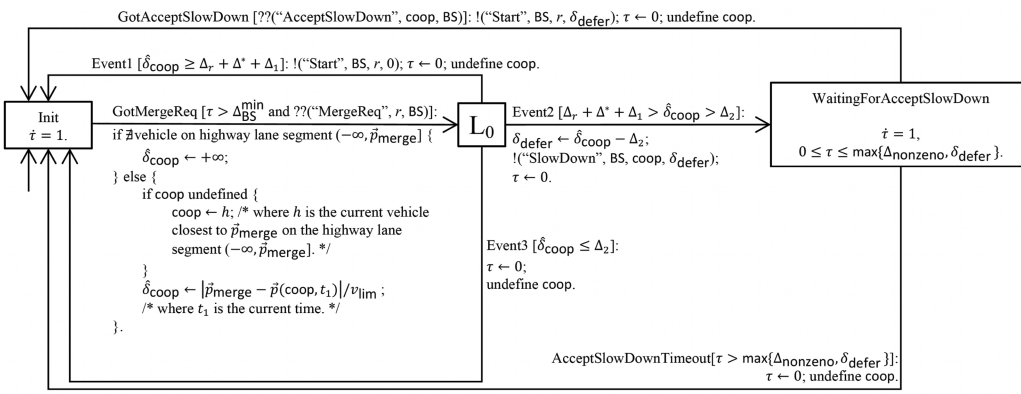
(5) When dwelling in mode "WaitingForAcceptSlowDown," the local clock τ grows continuously (i.e., $\dot{\tau} = 1$), and must not exceed its range constraint of $[0, \max\{\Delta_{\text{nonzeno}}, \delta_{\text{defer}}\}]$, where $\Delta_{\text{nonzeno}} > 0$ is a configuration constant, and δ_{defer} is set by "Event2." In this mode, BS may trigger one of the following two events (see "GotAcceptSlowDown" and "AcceptSlowDown-Timeout" events respectively in Figure 3):

Case1 (GotAcceptSlowDown) If before τ exceeds $\max\{\Delta_{\text{nonzeno}}, \delta_{\text{defer}}\}$, an "AcceptSlow-Down" wireless packet is received from coop, then BS triggers the "GotAcceptSlow-Down" event. This event carries out the following sequential action:
send a "Start" wireless packet to r, telling r to start in δ_{defer} seconds (with the packet data payload of δ_{defer}); set the local clock τ to 0; undefine coop.
After the above action, BS returns to mode "Init."

Case2 (AcceptSlowDownTimeout) If local clock τ exceeds $\max\{\Delta_{\text{nonzeno}}, \delta_{\text{defer}}\}$, then BS gives up waiting for the "AcceptSlowDown" wireless packet from coop, and triggers the timeout event "AcceptSlowDownTimeout." This event carries out the following sequential action:
set the local clock τ to 0; undefine coop.
After the above action, BS returns to mode "Init."

Ramp CAV r protocol behaviors (illustrated by hybrid automaton A_r in Figure 4):

(1) At any time instance, the ramp CAV r dwells in one of the following modes: "Init," "Requesting," "DeferringStart," "AcceleratingOnRamp," "ConstSpeedOnRamp," "Accelerat-ingHighwayLane," and "ConstSpeedHighwayLane."
(2) Initially, r dwells in the "Init" mode, stops at $\vec{p}_{\text{enter}}$ (i.e., $\vec{p}(r, t) = \vec{p}_{\text{enter}}$), and has its local clock τ's initial value set to 0.



Legends: Each rectangle box indicates a *hybrid automaton mode* (simplified as "*mode*" in the following). Inside a mode, the top line is the mode's name (it is local to the respective hybrid automata), the rest describes the constraints (e.g., a dwelling duration constraint like $0 \le \tau \le \Delta_{\text{nonzeno}}$) and continuous domain dynamics (typically specified with differential equations, e.g., $\dot{\tau} = 1$) related to the mode.
"L_0" is a transient mode, whose maximum dwelling duration constraint is 0 seconds, i.e., when the execution enters "L_0", it must exit "L_0" immediately (via a qualified event).
The arrow without source mode indicates the starting mode of execution (τ's initial value is uniformly sampled from $[0, \Delta_{\text{BS}}^{\min}]$). Other arrows represent discrete *events* for the system.
Annotations to each event arrow have the following meanings. Before the ":" is the optional event name and the *guard* (quoted by the brackets "[]"), i.e., the triggering condition for the event. Particularly, "??(x)" means the event is triggered upon the reception of a wireless packet "(x)" (a wireless packet (x) is a tuple of three or four elements, respectively the type, sender, intended receiver, and optional data payload of the packet). Note a sent wireless packet is not always received: the packet could be lost arbitrarily. After the ":" is the action carried out by the event. Particularly, "!(y)" means a wireless packet (y) is sent; and "←" means value assignment.
Same legends also apply to Figures 4 and 5.

Fig. 3. Hybrid automaton A_{BS} for the base station BS. Drawn based on our intellectual property. Early version appeared in [17] Figure 2.

(3) When dwelling in mode "Init," r is stopping (i.e., $|\dot{\vec{p}}(r,t)| = 0$) and the local clock τ grows continuously (i.e., $\dot{\tau} = 1$). But when τ exceeds Δ_{nonzeno}, r triggers the "SendMergeReq" event. This event carries out the following sequential action: send a "MergeReq" wireless packet to BS; reset τ to 0.
After the above action, r enters mode "Requesting" to wait for BS's reply.
(4) When dwelling in mode "Requesting," r is stopping (i.e., $|\dot{\vec{p}}(r,t)| = 0$) and τ grows continuously. If before τ exceeds Δ_{nonzeno}, the reply from BS, i.e., a "Start" wireless packet (with the data payload of value σ_{defer}), is received, then r triggers the "GotStart" event, resets τ to 0, and enters the "DeferringStart" mode. Otherwise, if no reply from BS is received till τ exceeds Δ_{nonzeno}, then r triggers the "RequestTimeout" event, resets τ to 0, and returns to mode "Init," giving up waiting for BS's reply.
(5) Once r enters the "DeferringStart" mode, r will first wait for σ_{defer} seconds, then (enter "AcceleratingOnRamp" mode) accelerate as per $\text{acc}(0, v_{\text{rm}}, \tau)$ ($\tau \in [0, \delta_{\text{a}}(0, v_{\text{rm}})]$) to v_{rm}, (enter "ConstSpeedOnRamp" mode) maintain this speed till passed $\vec{p}_{\text{merge}}$, (enter "AcceleratingHighwayLane" mode) accelerate as per $\text{acc}(v_{\text{rm}}, v_{\text{lim}}, \tau)$ ($\tau \in [0, \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})]$)

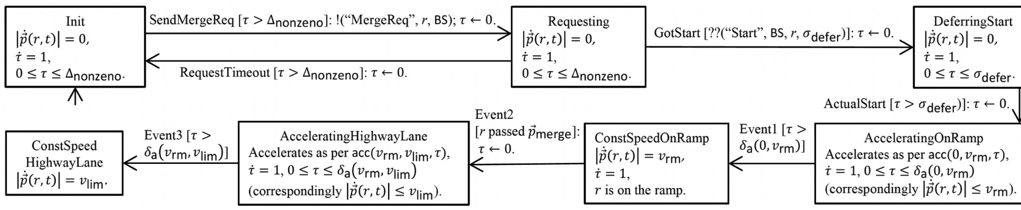


Fig. 4. Hybrid automaton A_r for the ramp CAV r (τ's initial value is set to 0). Drawn based on our intellectual property. Early version appeared in [17] Figure 3(a).

to v_{lim}, and (enter "ConstSpeed" mode) maintain v_{lim} on the highway lane, finishing the merging.

Highway CAV h_i ($i \in \{1, \ldots, n\}$) protocol behaviors (illustrated by hybrid automaton A_i in Figure 5):

(1) At any time instance, highway CAV h_i ($i \in \{1, \ldots, n\}$) dwells in one of the following modes: "Init," "DeferringDeceleration," "Decelerating," "ConstLowSpeed," "Accelerating," and "Sync."
(2) Initially, h_i dwells in mode "Init," drives at speed v_{lim}, and the state local variable is set to "Init."
(3) When dwelling in mode "Init," h_i may trigger one of the following two events (see "GotSlowDown" and "StartSyncPred" events, respectively, in Figure 5):
    Case1 (GotSlowDown) If a "SlowDown" wireless packet is received from BS (with the data payload of value δ_{defer}), then h_i triggers the "GotSlowDown" event. This event carries out the following sequential action (see event "GotSlowDown" in Figure 5): send the "AcceptSlowDown" wireless packet to BS; set state to "Coop"; set local clock τ to 0. After the above action, h_i enters mode "DeferringDeceleration."
    Case2 (StartSyncPred, only applicable for $i > 1$) If h_{i-1} is no more than

$$D_1 \overset{\text{def}}{=} v_{\text{lim}}(\Delta_r + 2\Delta^* + \Delta_1 - \Delta_2) \qquad \text{(note Ineq. (7) implies } D_1 > 0\text{)} \tag{8}$$

    distance ahead of h_i and starts to decelerate from speed v_{lim}, then h_i triggers the "StartSyncPred" event, sets state to "Sync," and enters mode "Sync."
(4) Once h_i enters the "DeferringDeceleration" mode, h_i will first wait for δ_{defer} seconds,
    then (enter "Decelerating" mode with local clock τ reset to 0) decelerate as per dec$(v_{\text{lim}}, v_{\text{rm}}, \tau)$ ($\tau \in [0, \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}])$ to v_{rm},
    (enter "ConstLowSpeed" mode without changing τ) maintain this speed till local clock τ exceeds $\Delta_r + \Delta^*$ seconds (note Ineq. (6) implies $\Delta_r + \Delta^* > \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})$),
    (enter "Accelerating" mode with local clock τ reset to 0) accelerate as per acc$(v_{\text{rm}}, v_{\text{lim}}, \tau)$ ($\tau \in [0, \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}])$ to v_{lim},
    and return to mode "Init" (with state reset to "Init").
(5) Once h_i enters the "Sync" mode, h_i keeps its speed the same as h_{i-1}'s, until h_{i-1} recovers its speed of v_{lim}. At that moment, h_i triggers the "StopSyncPred" event, sets state to "Init," and returns to mode "Init."

We claim the above protocol for BS, r, and h_i ($i \in \{1, \ldots, n\}$) guarantees CTH safety and liveness (i.e., entities will not stuck in any mode), even under arbitrary wireless packet losses. In the next subsection, we shall rigorously describe and prove these properties.

4.3 Analysis

We claim the following theorem on sufficient conditions for safety and liveness.

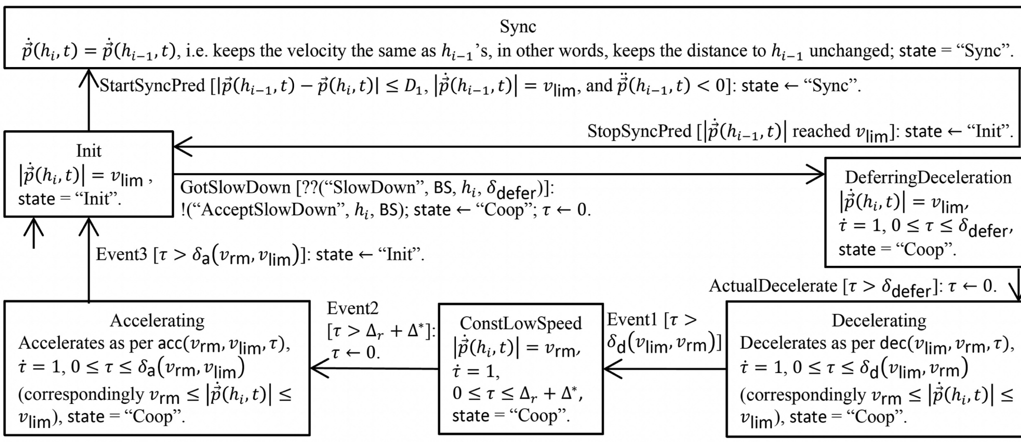


Fig. 5. Hybrid automaton A_i for the highway CAV h_i ($i \in \{1, 2, \ldots, n\}$); note as h_0 does not exist, for h_1, the event "StartSyncPred" can never happen. Drawn based on our intellectual property. Early version appeared in [17] Figure 3(b).

Theorem 1. *Suppose configuration constants of A_{BS}, A_r, and A_i ($i = 1, 2, \ldots, n$) comply with the following constraints:*

(c1) *aforementioned constraints: Ineq. (1), (3), (6), $\Delta^* > 0$, and $\Delta_{\text{nonzeno}} > 0$;*
(c2) $\Delta_{\text{BS}}^{\min} > \Delta_{\text{coop}}^{\max} + \Delta_{\text{nonzeno}}$, *where*
$$\Delta_{\text{coop}}^{\max} \stackrel{\text{def}}{=} \delta_{\text{defer}}^{\max} + \Delta_r + \Delta^* + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}),\ \delta_{\text{defer}}^{\max} \stackrel{\text{def}}{=} \hat{\delta}_{\text{coop}}^{\max} - \Delta_2,\ \textit{and}\ \hat{\delta}_{\text{coop}}^{\max} \stackrel{\text{def}}{=} \Delta_r + \Delta^* + \Delta_1;$$
(c3) $v_{\text{rm}}\Delta_r \geqslant v_{\text{lim}}\Delta^*$;
(c4) $\Delta_{\text{nonzeno}} < \Delta_r + \Delta^* + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})$.

Then we have the following claims.

Claim 1 (Safety). *$\forall t \in [t_0, +\infty)$, for any two CAVs x and y on the highway lane, one and only one of the following sustains: (x, y) is CTH-Δ^* safe at t, or (y, x) is CTH-Δ^* safe at t.*

Claim 2 (Liveness (Automatic Reset)). *suppose at $t_1 \in [t_0, +\infty)$, the base station* BS *leaves hybrid automaton A_{BS} mode "Init", while highway CAV h_is ($i = 1, 2, \ldots, n$) are all dwelling in respective A_i mode "Init", let*

$$\Delta_{\text{reset}}^{\max} \stackrel{\text{def}}{=} \Delta_{\text{coop}}^{\max} + \Delta_{\text{nonzeno}} + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}), \tag{9}$$

*then $\exists t_2 \in (t_1, t_1 + \Delta_{\text{reset}}^{\max}]$ s.t. either (**Stable State 1**) at t_2, $h_1, h_2, \ldots, h_n$,* BS*, and the ramp CAV r are in respective hybrid automata mode "Init"; or (**Stable State 2**) at t_2, $h_1, h_2, \ldots, h_n$, and* BS *are in respective hybrid automata mode "Init" and r is in A_r's mode "ConstSpeedHighwayLane".*

In order to prove Theorem 1, we need to first propose/prove several definitions and lemmas.

Definition 2 (Coop-duration). For a highway CAV h_i ($i \in \{1, 2, \ldots, n\}$), suppose its hybrid automaton variable, state, changes from "Init" to "Coop" at $t_1 \in [t_0, +\infty)$, then as per Figure 5, the state must change back to "Init" at some finite t_2 (where $t_1 < t_2 \leqslant t_1 + \Delta_{\text{coop}}^{\max}$, see (**c2**) for the definition of $\Delta_{\text{coop}}^{\max}$). That is, $\forall t \in (t_1, t_2]$, state ="Coop";[1] and at t_2^+, state = "Init". We call $(t_1, t_2]$

[1]Note, if we regard hybrid automaton discrete variables' values are left continuous along the time axis, then at t_1, we regard state ="Init".

a "*coop-duration*". Note as per Figures 3 and 5, it is easy to see that $\Delta_{\text{coop}}^{\text{max}}$ is the maximum possible time length for a coop-duration.

Lemma 1. *Any two coop-durations $(t_1, t_2]$ and $(t_3, t_4]$ respectively belonging to two different CAVs can never overlap nor connect, i.e., $[t_1, t_2] \cap [t_3, t_4] = \varnothing$.*

Proof. Suppose $[t_1, t_2] \cap [t_3, t_4] \neq \varnothing$ and suppose $t_5 \in [t_1, t_2] \cap [t_3, t_4]$. Then $t_1 \in [t_5 - \Delta_{\text{coop}}^{\text{max}}, t_5]$ and $t_3 \in [t_5 - \Delta_{\text{coop}}^{\text{max}}, t_5]$, therefore $|t_1 - t_3| \leqslant \Delta_{\text{coop}}^{\text{max}}$. This means BS sends two different "SlowDown" packets within $\Delta_{\text{coop}}^{\text{max}}$. This contradicts (**c2**), where $\Delta_{\text{BS}}^{\text{min}} > \Delta_{\text{coop}}^{\text{max}}$. □

Lemma 2. *Any two coop-durations $(t_1, t_2]$ and $(t_3, t_4]$ can never overlap nor connect, i.e., $[t_1, t_2] \cap [t_3, t_4] = \varnothing$.*

Proof. In addition to Lemma 1, applying similar reasonings, we can prove coop-durations of a same highway CAV cannot overlap nor connect. □

Lemma 3. $\forall t \in [t_0, +\infty)$, *if no highway CAV is in coop-duration at t, then all highway CAVs (i.e., $h_1, h_2, \ldots, h_n$) are in "Init" mode at t.*

Proof. According to Figure 5, if $\exists h_i$, whose state = "Sync" at t, then there must be an h_j in a coop-duration at t. □

Lemma 4. *Suppose $(t_1, t_2] \subseteq [t_0, +\infty)$ is the first ever happened coop-duration, then $\forall t \in [t_0, t_2]$, $\forall i \in \{1, 2, \ldots, n-1\}$, (h_i, h_{i+1}) is CTH-Δ^* safe at t.*

Proof. See Appendix B for details. □

Lemma 5. $\forall t \in [t_0, +\infty)$, $\forall i \in \{1, 2, \ldots, n-1\}$, (h_i, h_{i+1}) *is CTH-Δ^* safe at t.*

Proof. See Appendix C for details. □

Corollary 1. *Throughout $[t_0, +\infty)$, there is no spatial swapping between h_i and h_j ($\forall i, j \in \{1, 2, \ldots, n\}$, $i \neq j$) along the highway lane.*

Proof. Due to Lemma 5, the first swapping never happens. □

Lemma 6. *Suppose ramp CAV r reaches $\vec{p}_{\text{merge}}$ at $t_1 \in [t_0, +\infty)$, then for each $i \in \{1, 2, \ldots, n\}$, one and only one of the following claims sustain: (**Claim 1**) (h_i, r) is CTH-Δ^* safe throughout $[t_1, +\infty)$; (**Claim 2**) (r, h_i) is CTH-Δ^* safe throughout $[t_1, +\infty)$.*

Proof. See Appendix D for details. □

Now we are ready to prove Theorem 1.

Proof of Theorem 1 **Claim 1**:

In case $x, y \in \{h_1, h_2, \ldots, h_n\}$, the claim sustains due to Lemma 5 and Corollary 1 (in case x and y are not consecutive, e.g., $x = h_i$ and $y = h_{i+k}$, where $k > 1$, then due to Corollary 1, the distance between x and y is no less than the distance between h_{i+k-1} and y, hence the CTH-Δ^* safety rule still sustains for (x, y)).

In case $x \in \{h_1, h_2, \ldots, h_n\}$ and $y = r$, or the reverse, the claim sustains due to Lemma 6.

Combining the above two cases, the claim sustains. □ (‡)

Proof of Theorem 1 **Claim 2**:

Case 1: "Event1" happens at t_1. Then at t_1^+, BS returns to "Init" and remains there till at least $t_1 + \Delta_{\text{BS}}^{\text{min}}$.

Case 1.1: If r receives the "Start" packet at t_1, then it will be in "ConstSpeedHighwayLane" by $t_1 + \Delta_r + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}) < t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$ (due to (**c2**)). Meanwhile, all $h_1 \sim h_n$ remain in "Init" from t_1 to $t_1 + \Delta_r + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$. Therefore, $t_3 \stackrel{\text{def}}{=} t_1 + \Delta_r + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$ is a time instance that matches the claim's description (we call such a time instance a "*valid time instance*" in the following).
Case 1.2: If r did not receive the "Start" packet at t_1. Then, as r sent the "MergeReq" packet at t_1, it will be at "Init" at $t_1 + \Delta_{\mathsf{nonzeno}} < t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$ (due to (**c2**)). Meanwhile, $h_1 \sim h_n$ remains in "Init" at $t_1 + \Delta_{\mathsf{nonzeno}}$. Hence $t_4 \stackrel{\text{def}}{=} t_1 + \Delta_{\mathsf{nonzeno}}$ is a valid time instance.
Case 2: "Event2" happens at t_1. Then by $t_1 + \max\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$, BS should have returned to "Init" and remain there till at least $t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$.

Meanwhile, it will not send another "SlowDown" packet during $(t_1, t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}]$ at least. (♣)
Case 2.1: h_{coop} receives the "SlowDown" packet at t_1. Then the coop-duration starts at t_1 and ends at $t_5 \stackrel{\text{def}}{=} t_1 + \delta_{\mathsf{defer}} + \Delta_r + \Delta^* + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$.

Meanwhile, as per (**c2**), $\exists \varepsilon \in (0, \Delta_{\mathsf{BS}}^{\mathsf{min}} - \Delta_{\mathsf{coop}}^{\mathsf{max}} - \Delta_{\mathsf{nonzeno}})$, s.t. $\varepsilon < \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$. Let $t_6 \stackrel{\text{def}}{=} t_5 + \varepsilon$, and $t_7 \stackrel{\text{def}}{=} t_6 + \Delta_{\mathsf{nonzeno}}$. Then we have $t_1 + \max\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\} < t_5 < t_6 < t_7 < t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$ (due to (**c2**), (**c4**)). Hence BS is in "Init" at t_6 and t_7.

Due to (♣), a second coop-duration will not start till after $t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$. Hence due to Lemmas 2 and 3, we know $h_1 \sim h_n$ are all in "Init" at t_6 and at t_7.
Case 2.1.1 BS receives "AcceptSlowDown" at t_1^+, it sends ("Start", BS, r, δ_{defer}) at t_1^+.

(**a**) r receives the "Start" packet at t_1^+. Then it reaches "ConstSpeedHighwayLane" at $t_1 + \delta_{\mathsf{defer}} + \Delta_r + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}) < t_6$. Hence t_6 is a valid time instance.

(**b**) r did not receive the "Start" packet at t_1^+. Then at t_6, it must be in "Init" or "Requesting". In this case, if r is in "Init" at t_6. Then t_6 is a valid time instance; If r is in "Requesting" at t_6. Then r must have switched to "Init" at t_7. Then t_7 is a valid time instance.

Combining **a** and **b**, **Case 2.1.1** complies with the claim.
Case 2.1.2 BS does not receive "AcceptSlowDown" at t_1^+. Then it returns to "Init" at $t_1 + \max\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$ and remains there till $t_1 + \max\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\} + \Delta_{\mathsf{BS}}^{\mathsf{min}}$. No "Start" packet was sent.

Then similar to the analysis of item (**b**), if r is in "Init" at t_6. Then t_6 is a valid time instance; If r is in "Requesting" at t_6. Then t_7 is a valid time instance.

Combining **Case 2.1.1** and **Case 2.1.2**, **Case 2.1** complies with the claim.
Case 2.2 h_{coop} does not receive "SlowDown" at t_1. Then nothing happens to $h_1 \sim h_n$ during $[t_1, t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}]$.

Let $t_8 \stackrel{\text{def}}{=} t_1 + \max\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$, $t_9 \stackrel{\text{def}}{=} t_8 + \varepsilon$, $t_{10} \stackrel{\text{def}}{=} t_9 + \Delta_{\mathsf{nonzeno}}$, where ε is the same ε chosen for **Case 2.1**. Then (**c2**) and (**c4**) imply $0 < t_8 < t_9 < t_{10} < t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$. Hence at t_9 and t_{10}, BS and $h_1 \sim h_n$ are in "Init". Considering r, we have the following two cases.
Case 2.2.1 r is in "Init" at t_9. Then t_9 is a valid time instance.
Case 2.2.2 r is in "Requesting" at t_9. Then t_{10} is a valid time instance.

Combining **Case 2.2.1** and **Case 2.2.2**, **Case 2.2** complies with the claim.

Combining **Case 2.1** and **Case 2.2**, **Case 2** complies with the claim.
Case 3 "Event3" happens at t_1. Then **BS** returns to "Init" at t_1^+. Nothing happens to $h_1 \sim h_n$ till $t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$.

Let $t_{11} \stackrel{\text{def}}{=} t_1 + \varepsilon$, $t_{12} \stackrel{\text{def}}{=} t_{11} + \Delta_{\mathsf{nonzeno}}$, where ε is the same ε chosen for **Case 2.1**. Then (**c2**) and (**c4**) imply $t_1 < t_{11} < t_{12} < t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$. Hence at t_{11} and t_{12}, BS and $h_1 \sim h_n$ are in "Init". Considering r, we have the following two cases.
Case 3.1 r is in "Init" at t_{11}. Then t_{11} is a valid time instance.

Case 3.2 r is in "Requesting" at t_{11}. Then t_{12} is a valid time instance.

Combining **Case 3.1** and **Case 3.2**, Case **Case 3** complies with the claim.

Combining **Case 1**, **Case 2**, and **Case 3**, the claim sustains. □ (‡‡)

Due to (‡) and (‡‡), the theorem sustains.

5 IMPORTANT OBSERVATIONS

We have two important observations regarding Theorem 1's validity based on the design of the proposed protocol and the proof of the theorem.

Relaxation on Assumption 3. BS only needs to be able to instantly know (upon reception of a "MergeReq" packet, see Figure 3) which highway CAV is currently closest to $\vec{p}_{\text{merge}}$ on the segment $[\vec{p}_{\text{merge}} - v_{\text{lim}}(\Delta_r + \Delta^* + \Delta_1), \vec{p}_{\text{merge}}]$, and (if it exists) whether its current distance to $\vec{p}_{\text{merge}}$ is no less than $v_{\text{lim}}(\Delta_r + \Delta^* + \Delta_1)$, or no greater than $v_{\text{lim}}\Delta_2$, or otherwise.

V2V Communication Failures are Irrelevant. V2V communications (if used) are only used in the "Sync" mode of the highway CAV hybrid automaton (see Figure 5), and are only used between two consecutive highway CAVs (h_i and h_{i+1}, where $i = 1, 2, \ldots, n-1$) for three possible cases: to trigger the "StartSyncPred" event, to let h_i inform h_{i+1} of the former's current ranging/velocity/acceleration or to trigger the "StopSyncPred" event. For all these three cases, the V2V communications can be replaced by h_{i+1}'s local ranging sensors (see Assumption 4). Hence V2V communications failures are irrelevant. In case the ranging sensors need line-of-sight, we have the following observations. All the highway CAVs that should be in "Sync" at any time instance t must be following a unique highway CAV h_ι ($\iota \in \{1, 2, \ldots, n\}$) that is in a coop-duration. This implies h_ι must be behind r, if r is after all on the highway lane at t. Therefore, it is impossible that r resides between two *speed synchronizing* highway CAVs (i.e., the predecessor highway CAV is in a coop-duration, while the follower highway CAV is in "Sync"; or both are in "Sync") at t. Therefore, the line-of-sight between two speed synchronizing highway CAVs is available at t.

6 EVALUATION

We carry out simulations to verify the proposed protocol, particularly on the CTH-Δ^* safety guarantee, the liveness (automatic reset) guarantee, and the success rates and time costs of merging.

We also compare the proposed protocol with two other protocols: the *priority-based protocol* adapted from Aoki et al. [5], and the *consensus-based protocol* from Wang et al. [43]. We choose these two protocols because their focus problem contexts are the most similar to ours.

Specifically, Aoki et al. [5] focus on the design of a safe highway metered-ramp merging protocol, with collision avoidance guarantee under arbitrary wireless packet losses. As mentioned in Section 1, how to adapt their protocol to guarantee *CTH safety* under arbitrary wireless packet losses is still an open problem. Fortunately, Aoki et al. [5] mentioned a "baseline priority-based protocol" for comparison purposes in their article's evaluation section. We found a way to adapt this "baseline priority-based protocol" to guarantee CTH safety under arbitrary wireless packet losses. Specifically, the adapted protocol (referred to as the "*priority-based protocol*" in the following) looks exactly the same as our proposed protocol of Section 4.2 (referred to as "*the proposed protocol*" in the following), except that the base station no longer requests highway CAVs to yield. Formally, this means to adapt the hybrid automaton A_{BS} of Figure 3 as follows:

(1) Expand Event3's guard to cover all cases where $\hat{\delta}_{\text{coop}} < \Delta_r + \Delta^* + \Delta_1$;
(2) Delete mode "WaitingForAcceptSlowDown" and event "Event2," "GotAcceptSlowDown," "AcceptSlowDownTimeout."

The proof of CTH guarantee under arbitrary wireless packet losses of the above priority-based protocol follows the corresponding proof for the proposed protocol, as the priority-based protocol is basically a subset of the proposed protocol.

The other comparison alternative, Wang et al. [43]'s consensus-based protocol, is a highway and ramp merging protocol using V2X communications. The protocol can achieve good CTH safety statistically, but it does not focus on CTH *guarantee* under arbitrary wireless packet losses. We choose to compare with this protocol because it covers V2X communications, highway and ramp merging, and CTH safety. Similar to Aoki et al. [5]'s work, the focus problem context does not exactly match ours but is among the closest.

Next, we shall discuss the simulator configurations and the evaluation results.

6.1 Simulation Configuration

We follow the recommendations by the seminal textbook of [34] to configure our simulator. Specifically, CTH safety desired time headway $\Delta^* = 3$s; $\Delta_{\mathrm{BS}}^{\min} = 39.61$s; $\Delta_{\mathrm{nonzeno}} = 0.1$s; $D_r = 300$m; $v_{\lim} = 33.333$m/s; $v_{\mathrm{rm}} = 25$m/s; acceleration and deceleration strategy are set as per [34], which decides $\delta_{\mathrm{a}}(0, v_{\mathrm{rm}}) = 13.01$s, $d_{\mathrm{a}}(0, v_{\mathrm{rm}}) = 200.6840$m, $\delta_{\mathrm{a}}(v_{\mathrm{rm}}, v_{\lim}) = 12.20$s, $d_{\mathrm{a}}(v_{\mathrm{rm}}, v_{\lim}) = 362.3613$m, $\delta_{\mathrm{d}}(v_{\lim}, v_{\mathrm{rm}}) = 3.08$s, and $d_{\mathrm{d}}(v_{\lim}, v_{\mathrm{rm}}) = 90.9735$m. The above configuration further decides other parameters, specifically, Δ_r (see Equation (2)), Δ_1 (see Equation (4)), Δ_2 (see Equation (5)), D_1 (see Equation (8))), and $\Delta_{\mathrm{reset}}^{\max}$ (see Equation(9)). Particularly, $\Delta_{\mathrm{reset}}^{\max} = 50.4$s, which is used in Section 6.3 and Table 2.

Note the above configurations comply with the constraints demanded by Theorem 1, as well as the recommendations of the consensus-based protocol [43].

At the beginning of each simulation trial, our simulator generates n (n = 120, 180, or 240, respectively for light, mild, and heavy traffic; n's value is fixed for each individual simulation trial) highway CAVs along the highway lane segment [−50, 000m, 0m], where the location at 0m is $\vec{p}_{\mathrm{merge}}$. The exact initial locations of the n highway CAVs are randomly chosen as per a pseudo-uniform distribution, which takes into consideration of Assumption 5. Specifically, the pseudo-code is as follows:

- Step1 initialize H to empty set;
- Step2 **IF** ($|H| \geqslant n$) **THEN** terminate; **ELSE**
  - Step2.1 randomly choose a point p on the highway lane segment [−50, 000m, 0m] as per uniform distribution;
  - Step2.2 **IF** p does not violate CTH-Δ^* safety rule with the points already in H **THEN** add p into H; **ELSE** ignore p;
  - Step2.3 go back to Step2.

The generated H is the initial location for the highway CAVs for the trial.

Our simulator also adopts a wireless packet loss rate parameter P, whose value is set to 0.1 (i.e., 10%), 0.5 (i.e., 50%), or 0.9 (i.e., 90%) to evaluate the proposed protocol under mild, moderate, and severe wireless packet losses (P's value is fixed for each individual simulation trial).

For each given n and P values, we run 25 simulation trials. Each trial simulates 10 minutes (unless in some exception cases, see the last paragraph of Section 6.3) of a highway and metered-ramp merging scenario.

6.2 Safety

Theorem 1 **Claim 1** is on the CTH-Δ^* safety guarantee. To validate this claim, Table 1 shows the statistics of sampled time headways (relative to the respective immediate predecessor vehicles, see Definition 1) of all vehicles in all simulation trials (for each vehicle simulated, its time headway

Table 1. Simulation Results: Time Headway

Protocols	n	P	Time headway statistics (s)				
			min	median	max	average	std
The Proposed Protocol	120	0.1	**3.0**	9.4	70.3	12.4	9.1
		0.5	**3.0**	9.5	73.1	12.4	9.4
		0.9	**3.0**	9.7	81.1	12.5	9.3
Priority-based Protocol		0.1	**3.0**	9.6	81.9	12.4	9.0
		0.5	**3.0**	9.7	80.5	12.4	9.2
		0.9	**3.0**	9.6	72.8	12.4	9.2
Consensus-based Protocol		0.1	**1.9**	9.7	97.0	12.5	9.3
		0.5	**1.1**	9.7	73.2	12.5	9.1
		0.9	**0.4**	9.9	73.1	12.4	9.0
The Proposed Protocol	180	0.1	**3.0**	6.8	49.9	8.3	5.2
		0.5	**3.0**	6.8	58.0	8.3	5.1
		0.9	**3.0**	6.9	42.2	8.3	5.1
Priority-based Protocol		0.1	**3.0**	6.8	45.3	8.3	5.0
		0.5	**3.0**	6.8	44.3	8.3	5.1
		0.9	**3.0**	6.8	43.0	8.3	5.0
Consensus-based Protocol		0.1	**2.6**	6.8	58.1	8.3	5.0
		0.5	**3.0**	6.8	54.8	8.3	5.1
		0.9	**3.0**	6.9	47.5	8.3	5.1
The Proposed Protocol	240	0.1	**3.0**	5.5	31.2	6.2	2.9
		0.5	**3.0**	5.4	33.8	6.2	2.9
		0.9	**3.0**	5.5	33.7	6.2	3.0
Priority-based Protocol		0.1	**3.0**	5.5	41.3	6.3	3.0
		0.5	**3.0**	5.5	35.3	6.3	3.0
		0.9	**3.0**	5.5	40.2	6.3	3.0
Consensus-based Protocol		0.1	**3.0**	5.5	27.4	6.3	3.0
		0.5	**3.0**	5.5	27.7	6.2	2.9
		0.9	**3.0**	5.5	34.6	6.2	3.0

n: initial number of highway CAVs on the highway segment [−50km, 0km]; P: wireless packet loss rate; and $\Delta^* = 3$s.

is sampled every 0.4s). According to Table 1, for the proposed protocol, the time headways are always no less than 3.0s, which means the CTH-Δ^* safety (remember Δ^* is set to 3s, see Section 6.1) holds.[2] For the priority-based protocol, which basically is a subset of the proposed protocol, the CTH-Δ^* safety also holds. For the consensus-based protocol, the time headways cannot always satisfy CTH-Δ^* safety. Corresponding failures are highlighted in light gray in Table 1.

6.3 Liveness (Automatic Reset)

Theorem 1 **Claim 2** is on liveness guarantee, particularly in the sense of automatic reset. It proves the boundedness of reset time. This is confirmed by our simulations. According to Table 2, for the proposed protocol, all reset time costs are within the theoretical bound of $\Delta_{\text{reset}}^{\text{max}} = 50.4$s.

Note for all protocols, for given n, as wireless packet loss rate P rises, more resets return to **Stable State 1** instead of **Stable State 2** (see Theorem 1-**Claim 2** for definitions). The former can happen as fast as a sub-second software reset (though not always); while the latter must involve physical movement, hence usually costs tens of seconds.

[2]Note our computer simulation's time granularity is 0.01s, hence our minimum time headway value is rounded to one digit after the floating point.

Table 2. Simulation Results: Reset Time Cost

Protocols	n	P	Reset time cost statistics (s)				
			min	median	max	average	std
The Proposed Protocol	120	0.1	0.1	0.1	**37.9**	5.6	11.8
		0.5	0.1	0.1	**37.7**	3.4	9.5
		0.9	0.1	0.1	**35.3**	0.7	3.8
Priority-based Protocol		0.1	0.1	0.1	29.2	2.9	8.5
		0.5	0.1	0.1	29.2	1.6	6.5
		0.9	0.1	0.1	29.2	0.3	2.2
Consensus-based Protocol		0.1	0.1	0.1	56.5	2.8	7.8
		0.5	0.1	0.1	45.4	1.9	6.3
		0.9	0.1	0.1	24.3	0.4	2.3
The Proposed Protocol	180	0.1	0.1	0.1	**37.6**	2.1	7.9
		0.5	0.1	0.1	**37.6**	1.6	6.9
		0.9	0.1	0.1	**33.8**	0.4	2.4
Priority-based Protocol		0.1	0.1	0.1	29.2	0.7	4.2
		0.5	0.1	0.1	29.2	0.5	3.5
		0.9	0.1	0.1	0.1	0.1	0
Consensus-based Protocol		0.1	0.1	0.1	30.8	0.4	2.7
		0.5	0.1	0.1	20.7	0.3	1.9
		0.9	0.1	0.1	22.8	0.2	1.2
The Proposed Protocol	240	0.1	0.1	0.1	**36.0**	0.4	3.2
		0.5	0.1	0.1	**34.9**	0.4	3.0
		0.9	0.1	0.1	**0.1**	0.1	0
Priority-based Protocol		0.1	0.1	0.1	29.2	0.2	1.5
		0.5	0.1	0.1	0.1	0.1	0
		0.9	0.1	0.1	0.1	0.1	0
Consensus-based Protocol		0.1	0.1	0.1	0.1	0.1	0
		0.5	0.1	0.1	0.1	0.1	0
		0.9	0.1	0.1	0.1	0.1	0

See Table 1 for definitions of n and P. Note according to Theorem 1-**Claim 2**, the reset time costs of the proposed protocol shall be upper bounded by $\Delta_{\text{reset}}^{\max} = 50.4$s.

Also, note that normally each simulation trial lasts 10 minutes (in the simulated universe). But in case by the end of the 10th minute, the system is still waiting for a reset to happen, the simulation will go on till the reset happens.

6.4 Merging Success Rate and Time Cost

Besides safety and liveness guarantees, we are also concerned about the merging success rates and time costs. Merging success means before the end of the simulation trial, the ramp CAV is merged into the highway lane, all vehicles on the highway lane reach speed of v_{lim}, *and* the CTH-Δ^* safety is maintained at all times. Merging time cost is the total time cost from the start of the merging scenario to the first time instance when merging success is achieved. For a simulation trial where merging success is never achieved, merging time cost is not applicable.

Table 3 shows the merging success rates and merging time cost statistics. According to the table, for any given n and P (referred to as "(n, P) *combination*" or simply "*combination*" in the following), we have 2 comparison pairs: the proposed protocol versus the priority-based protocol, and the proposed protocol versus the consensus-based protocol. Hence for all the 9 combinations of n and P (light, mild, and heavy traffic versus low, mild, and high wireless packet loss rates), we have $9 \times 2 = 18$ comparison pairs.

Table 3. Simulation Results: Merging Success Rate and Time Cost

Protocols	n	P	succ. rate	Merging time cost statistics (s) min	median	max	avg	std
The Proposed Protocol	120	0.1	**24**/25	39.5	200.7	462.1	215.5	117.6
		0.5	**17**/25	42.9	235.5	516.5	282.9	134.8
		0.9	**3**/25	73.7	277.9	485.1	278.9	168.0
Priority-based Protocol		0.1	**19**/25	37.0	213.6	569.2	239.4	154.7
		0.5	**14**/25	48.6	325.1	535.6	300.0	164.9
		0.9	**2**/25	186.2	293.8	401.4	293.8	107.6
Consensus-based Protocol		0.1	**14**/25	23.6	199.0	548.8	219.2	138.8
		0.5	**7**/25	215.8	278.2	326.6	271.5	42.1
		0.9	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
The Proposed Protocol	180	0.1	**14**/25	36.7	146.5	580.1	205.4	159.8
		0.5	**5**/25	48.2	374.6	479.8	282.3	185.8
		0.9	**1**/25	431.9	431.9	431.9	431.9	0
Priority-based Protocol		0.1	**7**/25	46.4	269.0	551.4	315.5	165.3
		0.5	**5**/25	81.8	352.6	561.2	310.8	182.8
		0.9	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
Consensus-based Protocol		0.1	**3**/25	93.6	152.0	539.9	261.8	198.1
		0.5	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
		0.9	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
The Proposed Protocol	240	0.1	**3**/25	90.5	159.0	197.0	148.9	44.0
		0.5	**2**/25	140.0	352.4	564.8	352.4	212.4
		0.9	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
Priority-based Protocol		0.1	**1**/25	155.6	155.6	155.6	155.6	0
		0.5	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
		0.9	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
Consensus-based Protocol		0.1	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
		0.5	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.
		0.9	**0**/25	n.a.	n.a.	n.a.	n.a.	n.a.

See Table 1 for definitions of n and P; n.a.: not applicable.

Out of these 18 comparison pairs, there are 11 of them, where the proposed protocol's merging success rates are more than 99% better than the comparison counterpart's.[3]

This improvement is mainly because the proposed protocol focuses on two aspects simultaneously. It not only guarantees CTH-Δ^* safety under arbitrary wireless packet losses but also proactively coordinates the highway CAVs and the ramp CAV: when traffic is heavy, it asks the highway CAVs to yield to the ramp CAV. In comparison, neither of the other two protocols focuses on both of the aforementioned aspects.

More specifically, for all the 9 combinations of n and P, the consensus-based protocol fails all the 25 trials (i.e., success rate = 0) for 6 combinations; the priority-based protocol fails all the 25 trials (i.e., success rate = 0) for 3 combinations; while the proposed protocol only fails all the 25 trials (i.e., success rate = 0) for 1 combination, which corresponds to the heaviest traffic and highest wireless packet loss rate (i.e., $(n = 240, P = 0.9)$).

[3] In case the proposed protocol's success rate is positive, while the comparison counterpart's is 0, we also count the case as "the proposed protocol is more than 99% better".

Also, for (n, P) combinations where the consensus-based protocol succeeds for some trials (i.e., success rate > 0), the proposed protocol's merging time cost statistics are all comparable with (and usually better than) those of the consensus-based protocol's. Same is for the priority-based protocol.

7 CONCLUSION AND DISCUSSION

In this article, we propose a protocol to realize the safe merging of CAVs on highway and metered-ramp. We formally prove that the protocol can always guarantee the CTH safety and liveness, even under arbitrary wireless packet losses. These theoretical claims are verified by our simulations, which also show significant performance improvements over other alternatives.

This article also exemplifies the importance of introducing *cyber-physical transactions* and hybrid automata in the design and analysis of CPS. Particularly, Lemma 2 isolates possible combinations of discrete events and continuous maneuvres into mutually exclusive coop-durations (i.e., cyber-physical transactions). This greatly simplifies the design and analysis. Meanwhile, the specifications of the protocol and the formal proof of the CTH safety guarantee would be difficult (if not impossible) without the help of hybrid automata.

In future work, we will do the following.

(FW1) Take into consideration of wireless transmission and propagation delay. This delay shall be in the order of magnitude of 10 μs, which corresponds to distance errors in the order of magnitude of millimeters.

(FW2) Carry out *sensitivity study*: allow more variations in the various physical parameters, such as CAV velocities. As demonstrated by this article, the analyses are expected to be nontrivial and deserve multiple articles. Fortunately, the continuous nature of the physical world will bind the deviations from this article's formal models. This can help us to speculate the sensitivity. For example, suppose the CAV velocity error is bounded by V m/sec; as a coop-duration is bounded by $\Delta_{\text{reset}}^{\text{max}}$, (suppose CTH is satisfied at the start of the coop-duration) then we can expect that the inter-CAV distance error from CTH safety is bounded by $2V\Delta_{\text{reset}}^{\text{max}}$ when the coop-duration ends.

(FW3) Analyze the impacts when the number of packet losses is bounded. For example, whether there can be a time bound on the success of the merging.

(FW4) Derive necessary conditions for safety and liveness. We may start by negating the sufficient conditions listed in Theorem 1.

APPENDICES

A SYMBOL LIST

Symbols used in the article are listed alphabetically (Greek before Latin, and upper case before lower case) in the following.

(1) Δ_1, Δ_2, $\Delta_{\text{BS}}^{\text{min}}$, Δ_{nonzeno}, Δ_r, Δ^* are all configuration constants with positive values, see Equations (4), (5), Section 4.2-"Base Station BS protocol behaviors"-(2) (5), Equation (2), Definition 1, respectively.
(2) $\Delta_{\text{coop}}^{\text{max}}$, see Theorem 1 (**c2**).
(3) $\delta_{\text{a}}(v_1, v_2)$ is the total time needed to accelerate from v_1 to v_2 (where $0 \leqslant v_1 < v_2$), see Section 3.1.1.
(4) $\hat{\delta}_{\text{coop}}$ is a runtime variable local to A_{BS}. It is used to estimate the time distance of the current coop CAV to reach $\vec{p}_{\text{merge}}$.
(5) $\delta_{\text{d}}(v_2, v_1)$ is the total time needed to decelerate from v_2 to v_1 (where $v_2 > v_1 \geqslant 0$), see Section 3.1.2.

(6) δ_{defer} is a runtime variable created by A_{BS} at "Event2" (note Ineq. (7) ensures $\delta_{\text{defer}} > 0$), but may be sent to A_i ($i \in \{1, 2, \ldots, n\}$) in case h_i is chosen as the coop. It basically requests h_i to start deceleration (i.e., to yield) in δ_{defer} seconds.
(7) σ_{defer} is a runtime variable for A_r (see Figure 4). It is the data payload parameter received via the "Start" packet. In case the packet is sent by BS via "Event1" (see Figure 3), $\sigma_{\text{defer}} = 0$. In case the packet is sent by BS via the "GotAcceptSlowDown" event, $\sigma_{\text{defer}} = \delta_{\text{defer}}$. Upon reception of a "Start" packet, r will defer σ_{defer} seconds before actually starting the acceleration (i.e., entering the "AcceleratingOnRamp" mode of A_r).
(8) τ represents a runtime timer; it is a local variable to each hybrid automaton. Note for A_{BS}, the initial value of τ (when the system starts, i.e., at t_0) can be any value in $[0, \Delta_{\text{BS}}^{\text{min}}]$ (e.g., randomly chosen as per uniform distribution from this range); for A_r and A_i ($i = 1, 2, \ldots, n$), the initial value of τ is 0.
(9) D_1, D_r are both configuration constants with positive values, see Equation (8), Ineq. (1), respectively.
(10) acc is the predefined acceleration strategy, see Section 3.1.1.
(11) coop is a runtime variable for hybrid automaton A_{BS} only, whose value can only be "undefined" or h_i ($i \in \{1, 2, \ldots, n\}$). Intuitively, it refers to the closest approaching highway CAV toward the base station BS.
(12) $d_{\text{a}}(v_1, v_2)$ is the total distance needed to accelerate from v_1 to v_2 (where $0 \leqslant v_1 < v_2$), see Section 3.1.1.
(13) $d_{\text{d}}(v_2, v_1)$ is the total distance needed to decelerate from v_2 to v_1 (where $v_2 > v_1 \geqslant 0$), see Section 3.1.2.
(14) dec is the predefined deceleration strategy, see Section 3.1.2.
(15) $\vec{p}(x, t)$ is the location of vehicle x at wall clock time t. Correspondingly, $|\dot{\vec{p}}(x, t)|$ is the speed of vehicle x at t, and $\ddot{\vec{p}}(x, t)$ is the acceleration (deceleration) of x at t.
(16) state is a runtime variable for A_i ($i = 1, 2, \ldots, n$) only, whose value can only be "Init", "Coop", or "Sync". Note the initial value of state is set to "Init".
(17) t is the current wall clock time; it is a global variable.
(18) v_{lim} and v_{rm} are configuration constants related to CAV speed. They are, respectively, the maximum and minimum allowed speed on the highway lane. See Section 3.2 and Ineq. (3) for more information.

B PROOF OF LEMMA 4

Proof: First, as $(t_1, t_2]$ is the first ever coop-duration, due to Assumption 5 and Lemma 3, $h_1, h_2, \ldots, h_n$ all reside in hybrid automata mode "Init" throughout $[t_0, t_1]$, hence $\forall t \in [t_0, t_1]$, (h_i, h_{i+1}) ($i = 1, 2, \ldots, n-1$) is CTH-Δ^* safe. (★)

Suppose the coop-duration $(t_1, t_2]$ belongs to $h_\imath$ ($\imath \in \{1, 2, \ldots, n\}$). Then we have the following cases.

Case 1: First we discuss h_j, where $j < \imath$. As coop-durations cannot overlap nor connect, $\forall j \in \{1, 2, \ldots, \imath - 1\}$, as per A_j (see Figure 5), throughout $(t_1, t_2]$, h_j must remain in mode "Init". That is, for any h_j and h_{j+1} ($j \in \{1, 2, \ldots, \imath - 2\}$), throughout $(t_1, t_2]$, both retain the speed of v_{lim}. hence (h_j, h_{j+1}) is CTH-Δ^* safe throughout $(t_1, t_2]$. For $h_{\imath-1}$ and $h_\imath$, as $h_{\imath-1}$ retains the maximum allowed speed, v_{lim}, throughout $(t_1, t_2]$, hence $(h_{\imath-1}, h_\imath)$ is CTH-Δ^* safe throughout $(t_1, t_2]$.

Case 2: Now we discuss h_j, where $j > \imath$.

Case 2.1: Suppose at t_1, h_k ($k > \imath$) is the first highway CAV after $h_\imath$ s.t. $|\vec{p}(h_{k-1}, t_1) - \vec{p}(h_k, t_1)| > D_1$. Then we have the following cases.

Case 2.1.1: $\forall j \in \{\imath + 1, \imath + 2, \ldots, k - 1\}$, we have (h_{j-1}, h_j) is CTH-Δ^* safe throughout $(t_1, t_2]$. (★★)

This can be proved iteratively.

For $h_{\imath+1}$, throughout $(t_1, t_1 + \delta_{\text{defer}}]$, both $h_\imath$ and $h_{\imath+1}$ remain at v_{lim}; throughout $(t_1 + \delta_{\text{defer}}, t_2]$, $h_{\imath+1}$ synchronizes its speed with $h_\imath$ according to mode "Sync" (see Figure 5); Hence throughout $(t_1, t_2]$, $(h_\imath, h_{\imath+1})$ remains CTH-Δ^* safe.

Same reasoning can be applied to $h_{\imath+2}, h_{\imath+3}, \ldots, h_{k-1}$. Hence $(\star\star)$ sustains.

Case 2.1.2: For h_k, we have (h_{k-1}, h_k) is CTH-Δ^* safe throughout $(t_1, t_2]$. $(\star\star\star)$

We prove this step by step.

(i) $\forall t \in (t_1, t_1 + \delta_{\text{defer}}]$, both h_{k-1} and h_k retain the speed of v_{lim}, hence h_k remains in "Init" and $|\vec{p}(h_{k-1}, t) - \vec{p}(h_k, t)|$ remains unchanged.

(ii) $\forall t \in (t_1 + \delta_{\text{defer}}, t_1 + \delta_{\text{defer}} + \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})]$, h_{k-1} synchronizes its speed with $h_{k-2}, h_{k-3}, \ldots, h_\imath$, hence keeps decelerating from v_{lim} to v_{rm}; while h_k remains in "Init" (as $|\vec{p}(h_{k-1}, t_1 + \delta_{\text{defer}}) - \vec{p}(h_k, t_1 + \delta_{\text{defer}})| > D_1$, event "StartSyncPred" will not happen at $t_1 + \delta_{\text{defer}}$ to h_k, and during $(t_1 + \delta_{\text{defer}}, t_1 + \delta_{\text{defer}} + \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})]$ the event will neither happen to h_k as h_{k-1}'s speed is below v_{lim}). Meanwhile, for the entire deceleration process, $|\vec{p}(h_{k-1}, t) - \vec{p}(h_k, t)| > D_1 + d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) - v_{\text{lim}}\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) > v_{\text{lim}}\Delta^*$ (see D_1's definition in Equation (8)). This means (h_{k-1}, h_k) is CTH-Δ^* safe at t.

(iii) $\forall t \in (t_1 + \delta_{\text{defer}} + \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}), t_1 + \delta_{\text{defer}} + \Delta_r + \Delta^*]$ (note according to (6), $\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) < \Delta_r + \Delta^*$), h_{k-1} remains synchronizing its speed with $h_{k-2}, h_{k-3}, \ldots, h_\imath$, hence keeps the speed of v_{rm}, while h_k remains in mode "Init" (as $|\dot{\vec{p}}(h_{k-1}, t)| < v_{\text{lim}}$, event "StartSyncPred" will not happen to h_k). Meanwhile for this entire constant speed process, $|\vec{p}(h_{k-1}, t) - \vec{p}(h_k, t)| > D_1 + d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) - v_{\text{lim}}\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) - (v_{\text{lim}} - v_{\text{rm}})(\Delta_r + \Delta^* - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})) > v_{\text{lim}}\Delta^*$ (see D_1's definition in Equation (8)). This means (h_{k-1}, h_k) is CTH-Δ^* safe at t.

(iv) $\forall t \in (t_1 + \delta_{\text{defer}} + \Delta_r + \Delta^*, t_1 + \delta_{\text{defer}} + \Delta_r + \Delta^* + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})]$ (note $t_1 + \delta_{\text{defer}} + \Delta_r + \Delta^* + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})$ is when the coop-duration ends, i.e., it equals to t_2), h_{k-1} remains synchronizing its speed with $h_{k-2}, h_{k-3}, \ldots, h_\imath$, hence keeps accelerating from v_{rm} to v_{lim}; while h_k remains in "Init" (as h_{k-1} is accelerating, event "StartSyncPred" will not happen to h_k). Meanwhile for this entire acceleration process, $|\vec{p}(h_{k-1}, t) - \vec{p}(h_k, t)| > D_1 + d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) - v_{\text{lim}}\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) - (v_{\text{lim}} - v_{\text{rm}})(\Delta_r + \Delta^* - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})) - (v_{\text{lim}}\delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}}) - d_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})) \geqslant v_{\text{lim}}\Delta^*$ (see Δ_1 and D_1's definition in Equations (4) and (8)). This means (h_{k-1}, h_k) is CTH-Δ^* safe at t.

Combining (i)~(iv), we see $(\star\star\star)$ sustains.

Case 2.1.3: For h_j $(j = k+1, k+2, \ldots, n)$, as h_k remains in mode "Init" throughout $(t_1, t_2]$, h_{k+1} remains in mode "Init" throughout $(t_1, t_2]$, so on and so forth.

Combining **Case 2.1.1~Case 2.1.3**, we see in **Case 2.1**, $\forall j \in \{\imath+1, \imath+2, \ldots, n\}$, $\forall t \in (t_1, t_2]$, (h_{j-1}, h_j) is CTH-Δ^* safe at t.

Case 2.2 Suppose at t_1, $\forall j \in \{\imath+1, \imath+2, \ldots, n\}$, $|\vec{p}(h_{j-1}, t_1) - \vec{p}(h_j, t_1)| \leqslant D_1$, then follow the same proving method for **Case 2.1.1**, we can prove $\forall t \in (t_1, t_2]$, (h_{j-1}, h_j) is CTH-Δ^* safe at t.

Combining **Case 2.1** and **Case 2.2**, we see in **Case 2**, $\forall j \in \{\imath+1, \imath+2, \ldots, n\}$, $\forall t \in (t_1, t_2]$, (h_{j-1}, h_j) is CTH-Δ^* safe at t.

Combining **Case 1** and **Case 2**, together with the claim $(\star)$ proven at the very beginning, the lemma sustains. □

C PROOF OF LEMMA 5

Proof: **Case 1**: If coop-duration never happens, then all highway CAVs always remain in hybrid automata mode "Init". The lemma trivially sustain.

Case 2: If infinite coop-duration(s) happen. Suppose $(t_1, t_2] \subseteq [t_0, +\infty)$ is the first coop-duration ever happens. Then due to Lemma 4, this lemma trivially sustains for the duration $[t_0, t_2]$. At t_2^+, due to Lemma 2 and Lemma 3, all highway CAVs have returned to mode "Init", and $\forall i \in \{1, 2, \ldots, n-1\}$, (h_i, h_{i+1}) is CTH-Δ^* safe at t_2. Regard t_2 as the new t_0, and apply the same technique

to prove Lemma 4, we can prove this lemma sustains to the end of the second coop-duration, so on and so forth, until we cover time instance t. The lemma shall sustain.
Case 3: If finite coop-duration(s) happen. Then we can apply the proving technique of **Case 2**, and (if needed) after the last coop-duration ends, we can apply the proving technique of **Case 1**, until we cover time instance t. The lemma shall sustain.

Combining **Case 1** to **Case 3**, the lemma sustains. □

D PROOF OF LEMMA 6

Proof: According to A_r (see Figure 4), if r reaches $\vec{p}_{\text{merge}}$ at t_1, then it must have received the "ActualStart" event at $t_2 \stackrel{\text{def}}{=} t_1 - \Delta_r$, which is caused by a "Start" packet from the BS. There can be two cases.
Case 1: The "Start" packet is sent by BS via "Event1" in A_{BS} (see Figure 3) at t_2.

Then first, this means the most recent "Event2" of A_{BS}, the only event that can trigger a coop-duration, (if it ever happened) must be before $t_2 - \Delta_{\text{BS}}^{\min}$ (note there can be no more "Event2" of A_{BS} after t_2, as r has received "Start"). Due to (**c2**), $\Delta_{\text{BS}}^{\min} > \Delta_{\text{coop}}^{\max} > \Delta_r$, there is no coop-duration overlapping or connecting with $[t_2, +\infty)$. Due to Lemma 3, all highway CAVs hence should remain in "Init" throughout $[t_2, +\infty)$. (†)

Second, the "Event1" of A_{BS} at t_2 could be due to two cases at t_2^-, when BS receives a ("MergeReq", r, BS) packet.
Case 1.1: At t_2^-, there is no CAV on the highway lane segment of $(-\infty, \vec{p}_{\text{merge}}]$. This means at t_2^-, h_n is at highway lane segment of $(\vec{p}_{\text{merge}}, +\infty)$. So by t_1, h_n is at least $v_{\text{rm}}\Delta_r \geqslant v_{\text{lim}}\Delta^*$ (due to (**c3**)) ahead of r. Due to Corollary 1, this implies all highway CAVs are at least $v_{\text{rm}}\Delta_r \geqslant v_{\text{lim}}\Delta^*$ ahead of r at t_1. (††)

Conclusion (†) and (††) imply that $\forall t \in [t_1, +\infty)$, (h_i, r) $(i = 1, 2, \ldots, n)$ is CTH-Δ^* safe at t.
Case 1.2: At t_2^-, there is/are highway CAVs on the highway lane segment $(-\infty, \vec{p}_{\text{merge}}]$. Suppose the one closest to $\vec{p}_{\text{merge}}$ is h_i $(i \in \{1, 2, \ldots, n\})$. Then because BS sends "Start" packet via "Event1", we know

$$\hat{\delta}_{\text{coop}} = |\vec{p}_{\text{merge}} - \vec{p}(h_i, t_2^-)|/v_{\text{lim}} \geqslant \Delta_r + \Delta^* + \Delta_1 \tag{10}$$

Meanwhile, as per A_r, r shall reach $\vec{p}_{\text{critical}}$ (the location where r first reaches speed v_{lim}, see Figure 1) at $t_3 \stackrel{\text{def}}{=} t_1 + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})$, and $|\vec{p}_{\text{critical}} - \vec{p}_{\text{merge}}| = d_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})$.

Due to (†), h_i reaches $\vec{p}_{\text{merge}}$ at $t_2 + \hat{\delta}_{\text{coop}} \geqslant t_2 + \Delta_r + \Delta^* + \Delta_1$ (due to (10)) $= t_1 + \Delta^* + \Delta_1$. This means (r, h_i) is CTH-Δ^* safe at t_1.

Furthermore, h_i reaches $\vec{p}_{\text{critical}}$ at $t_2 + \hat{\delta}_{\text{coop}} + d_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})/v_{\text{lim}} \geqslant t_2 + \Delta_r + \Delta^* + \Delta_1 + d_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})/v_{\text{lim}} = t_1 + \Delta^* + \delta_{\text{a}}(v_{\text{rm}}, v_{\text{lim}})$ (see the definition of Δ_1 in (4)) $= t_3 + \Delta^*$. Hence (r, h_i) is CTH-Δ^* safe at t_3.

As r reaches v_{lim} after t_3, we hence conclude (r, h_i) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

Furthermore, due to Lemma 5 and Corollary 1, we can conclude $\forall j \in \{i, i+1, \ldots, n\}$, (r, h_j) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

Another important CAV is h_{i-1}. As it is on segment $(\vec{p}_{\text{merge}}, +\infty)$ at t_2^-, using the same reasoning for **Case 1.1**, we know (h_{i-1}, r) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

Due to Corollary 1, we can conclude $\forall j \in \{1, 2, \ldots, i-1\}$, (h_j, r) is CTH-Δ^* safe throughout $[t_1, +\infty)$.
Case 2: The "Start" packet is sent by BS via "Event2" (followed by "GotAcceptSlowDown") in A_{BS} (see Figure 3) at $t_2 - \delta_{\text{defer}}$. Immediately before it, BS must have sent ("SlowDown", BS, h_{coop}, δ_{defer}) packet to h_{coop} at $t_2 - \delta_{\text{defer}}$ and received h_{coop}'s "AcceptSlowDown" packet, where coop $\in \{1, 2, \ldots, n\}$. Without loss of generality, suppose coop $= i$.

Then during $[t_2 - \delta_{\text{defer}}, t_2]$, h_i remains at v_{lim} and drives $v_{\text{lim}}\delta_{\text{defer}} = \hat{\delta}_{\text{coop}}v_{\text{lim}} - d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) - v_{\text{rm}}(\Delta_r + \Delta^* - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}))$ distance since $t_2 - \delta_{\text{defer}}$.

During $(t_2, t_2 + \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})]$, h_i decelerates from v_{lim} to v_{rm} (note due to (6), $t_2 + \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) < t_2 + \Delta_r = t_1$) and drives $v_{\text{lim}}\delta_{\text{defer}} + d_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}) = \hat{\delta}_{\text{coop}}v_{\text{lim}} - v_{\text{rm}}(\Delta_r + \Delta^* - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}))$ distance since $t_2 - \delta_{\text{defer}}$.

During $(t_2+\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}), t_2+\Delta_r+\Delta^*]$, h_i remains at v_{rm}. Note $t_1 = t_2+\Delta_r \in (t_2+\delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}), t_2+\Delta_r + \Delta^*)$. This means, at t_1, h_i is in the "ConstLowSpeed" mode, maintaining the speed of v_{rm}. Therefore, at t_1, $\vec{p}(r, t_1) - \vec{p}(h_i, t_1) = \vec{p}_{\text{merge}} - \vec{p}(h_i, t_1) = \hat{\delta}_{\text{coop}}v_{\text{lim}} - (\hat{\delta}_{\text{coop}}v_{\text{lim}} - v_{\text{rm}}(\Delta_r + \Delta^* - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}})) + v_{\text{rm}}(t_1 - t_2 - \delta_{\text{d}}(v_{\text{lim}}, v_{\text{rm}}))) = v_{\text{rm}}\Delta^* > 0$. This means, at t_1, r is ahead of h_i by $v_{\text{rm}}\Delta^*$; and as h_i's speed at t_1 is v_{rm}, the above means (r, h_i) is CTH-Δ^* safe at t_1.

After t_1, r accelerates from v_{rm} to v_{lim}, while h_i remains at v_{rm} till $t_1 + \Delta^*$, when it reaches $\vec{p}_{\text{merge}}$. Then h_i carry out the same acceleration process as that of r to reach v_{lim}. Therefore, the two time-location curves (time as the x-axis, and location as the y-axis) of r and h_i above the location of $\vec{p}_{\text{merge}}$ are parallel and Δ^* away shifted along the time axis. Note the acceleration process is monotonic (the speed keeps monotonically increasing until the target speed is reached, see Assumption 1), and finally both CAVs stabilize at v_{lim}. By observing the time-location curves, we can see that during $[t_1, t_1 + \Delta^*]$, (r, h_i) is CTH-Δ^* safe; and during $[t_1 + \Delta^*, +\infty)$, (r, h_i) is also CTH-Δ^* safe. So in summary, (r, h_i) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

Furthermore, due to Lemma 5 and Corollary 1, we can conclude $\forall j \in \{i, i + 1, \ldots, n\}$, (r, h_j) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

Another important CAV is h_{i-1} (if $i > 1$). At $t_2 - \delta_{\text{defer}}$, when BS sends "SlowDown" packet to h_i, h_{i-1} must be on segment $(\vec{p}_{\text{merge}}, +\infty)$. Also, notice as coop-durations cannot overlap nor connect, and BS sends no more "SlowDown" packet after $t_2 - \delta_{\text{defer}}$. This means throughout $[t_2 - \delta_{\text{defer}}, +\infty)$, h_{i-1} is in "Init". Then using the same reasoning for **Case 1.1** for h_n, we can prove (h_{i-1}, r) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

Due to Corollary 1, we conclude $\forall j \in \{1, \ldots, i - 1\}$, (h_j, r) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

Combining **Case 1** and **Case 2**, we conclude the lemma sustains. □

ACKNOWLEDGMENTS

We also thank Dr. Shaheer Muhammad, Mr. Shiyu Zhang, and Mr. Zhihao Zhao anonymous reviewers for their comments to improve this article.

REFERENCES

[1] Federal Highway Administration. 2014. *Ramp Metering: A Proven, Cost-Effective Operational Strategy - A Primer*. US Department of Transportation.

[2] Michael Aeberhard, SebasEan Rauch, Mohammad Bahram, Georg Tanzmeister, Julian Thomas, Yves Pilat, Florian Homm, Werner Huber, and Nico Kaempchen. 2015. Experience, results and lessons learned from automated driving on Germany's highways. *IEEE Intelligent Transportation Systems Magazine* 7, 1 (2015), 42–57.

[3] Md Salman Ahmed, Mohammad A. Hoque, Jackeline Rios-Torres, and Asad Khauak. 2018. A cooperative freeway merge assistance system using connected vehicles. arXiv:1805.00508. Retrieved from https://arxiv.org/abs/1805.00508

[4] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. 1992. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. *Hybrid Systems* LNCS v736 (1992), 209–229.

[5] Shunsuke Aoki and Ragunathan Rajkumar. 2017. A merging protocol for self-driving vehicles. In *Proceedings of the 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 219–228.

[6] Tanveer Awal, Lars Kulik, and Kotagiri Ramamohanrao. 2013. Optimal traffic merging strategy for communication- and sensor-enabled vehicles. In *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 1468–1474.

[7] Jakob Axelsson. 2017. Safety in vehicle platooning: A systematic literature review. *IEEE Transactions on Intelligent Transportation Systems* 18, 5 (2017), 1033–1045.

[8] Mario di Bernardo, Alessandro Salvi, and Stefania SanEni. 2014. Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays. *IEEE Transactions on Intelligent Transportation Systems* 16, 1 (2014), 102–112.

[9] Wenjing Cao, Masakazu Mukai, Taketoshi Kawabe, Hikaru Nishira, and Noriaki Fujiki. 2015. Cooperative vehicle path generation during merging using model predictive control with real-time optimization. *Control Engineering Practice* 34, January (2015), 98–105.

[10] David J. Chang and Edward K. Moriok. 2005. Vehicle speed profiles to minimize work and fuel consumption. *Journal of Transportation Engineering* 131, 3 (2005), 173–182.

[11] Hossein Chehardoli and Ali Ghasemi. 2018. Adaptive centralized/decentralized control and identification of 1-D heterogeneous vehicular platoons based on constant time headway policy. *IEEE Transactions on Intelligent Transportation Systems* 19, 10 (2018), 3376–3386.

[12] Tianyi Chen, Meng Wang, Siyuan Gong, Yang Zhou, and Bin Ran. 2021. Connected and automated vehicle distributed control for on-ramp merging scenario: A virtual rotation approach. arXiv:2103.15047. Retrieved from https://arxiv.org/abs/2103.15047

[13] Laurene Claussmann, Marc Revilloud, Dominique Gruyer, and Sebastien Glaser. 2020. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems* 21, 5 (2020), 1826–1848.

[14] Kakan C. Dey, Li Yan, Xujie Wang, Yue Wang, Haiying Shen, Mashrur Chowdhury, Lei Yu, Chenxi Qiu, and Vivekgautham Soundararaj. 2016. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC). *IEEE Transactions on Intelligent Transportation Systems* 17, 2 (2016), 491–509.

[15] Jianbang Du, Qing Li, Fengxiang Qiao, and Lei Yu. 2018. Estimation of vehicle emission on mainline freeway under isolated and integrated ramp metering strategies. *Environmental Engineering and Management Journal* 17, 5 (2018), 1237–1248.

[16] Xueli Fan. 2022. *Cooperative Driving for Connected and Automated Vehicles on Dedicated Lanes*. Ph. D. Dissertation. The Hong Kong Polytechnic University.

[17] Xueli Fan, Qixin Wang, and Jie Liu. 2020. Work-in-progress abstract: A reliable wireless smart vehicle highway on-ramp merging protocol with constant time headway safety guarantee. In *Proceedings of the 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS) WiP Session*. 184–185.

[18] Cary Gray and David Cheriton. 1989. Leases: An efficient fault-tolerant mechanism for distributed file cache consistency. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles*. 202–210.

[19] Scou Heim, Rory Alston, Daniel Pierce, Patrick Smith, Robert Lyle, and David Bevly. 2019. *Automated Platoon Manipulation in Merging Scenarios Using Trajectory Estimation of Connected Vehicles*. Technical Report. US ARMY CCDC GVSC (FORMERLY TARDEC) WARREN United States.

[20] Peter Hidas. 2005. Modelling vehicle interactions in microscopic simulation of merging and weaving. *Transportation Research Part C: Emerging Technologies* 13, 1 (2005), 37–62.

[21] Xiaowen Jiang, Peter J. Jin, Xia Wan, and Yizhou Wang. 2017. A V2I (vehicle-to-infrastructure) based dynamic merge assistance method based on instantaneous virtual trajectories: A microscopic implementation of gap metering. In *Proceedings of the Transportation Research Board 96th Annual Meeting*.

[22] Igor Kabashkin. 2017. Reliable v2x communications for safety-critical intelligent transport systems. In *Proceedings of the 2017 Advances in Wireless and Optical Communications (RTUWO)*. IEEE, 251–255.

[23] Elham Semsar Kazerooni and Jeroen Ploeg. 2015. Interaction protocols for cooperative merging and lane reduction scenarios. In *Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 1964–1970.

[24] Stefan R. Klomp, Victor L. Knoop, Henk Taale, and Serge P. Hoogendoorn. 2021. Ramp metering with microscopic gap detection algorithm design and empirical acceleration verification. *Transportation Research Record* 2676, 1 (2021), 91–104.

[25] Stephanie Lefevre, Chao Sun, Ruzena Bajcsy, and Christian Laugier. 2014. Comparison of parametric and non-parametric approaches for vehicle speed prediction. In *Proceedings of the American Control Conference (ACC)*. 3494–3499.

[26] Clark Letter and Lily Elefteriadou. 2017. Efficient control of fully automated connected vehicles at freeway merge segments. *Transportation Research Part C: Emerging Technologies* 80, July (2017), 190–205.

[27] Xiao-Yun Lu and J. Karl Hedrick. 2003. Longitudinal control algorithm for automated vehicle merging. *International Journal of Control* 76, 2 (2003), 193–202.

[28] Xiao-Yun Lu, Han-Shue Tan, Steven E. Shladover, and J. Karl Hedrick. 2004. Automated vehicle merging maneuver implementation for AHS. *Vehicle System Dynamics* 41, 2 (2004), 85–107.

[29] Haigen Min, Yukun Fang, Xia Wu, Guoyuan Wu, and Xiangmo Zhao. 2021. On-ramp merging strategy for connected and automated vehicles based on complete information static game. *Journal of Traffic and Transportation Engineering (English Edition)* 8, 4 (2021), 582–595.

[30] Fernando V. Monteiro and Petros Ioannou. 2021. Safe lane change and merging gaps in connected environments. *IFAC-PapersOnLine* 54, 2 (2021), 69–74.
[31] América Morales and Henk Nijmeijer. 2016. Merging strategy for vehicles by applying cooperative tracking control. *IEEE Transactions on Intelligent Transportation Systems* 17, 12 (2016), 3423–3433.
[32] Ioannis A. Ntousakis, Kallirroi Porfyri, Ioannis K. Nikolos, and Markos Papageorgiou. 2014. Assessing the impact of a cooperative merging system on highway traffic using a microscopic flow simulator. In *Proceedings of the ASME 2014 International Mechanical Engineering Congress and Exposition.* American Society of Mechanical Engineers, V012T15A024–V012T15A024.
[33] Rattaphol Pueboobpaphan, Fei Liu, and Bart van Arem. 2010. The impacts of a communication based merging assistant on traffic flows of manual and equipped vehicles at an on-ramp using traffic flow simulation. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems.* IEEE, 1468–1473.
[34] Rajesh Rajamani. 2012. *Vehicle Dynamics and Control.* Springer.
[35] Gurulingesh Raravi, Vipul Shingde, Krithi Ramamritham, and JaEn Bharadia. 2007. Merge algorithms for intelligent vehicles. In *Proceedings of the Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems.* Springer, 51–65.
[36] Marwan K. Shalaby, Amr Farag, Omar M. AbdelAziz, Dalia M. Mahfouz, Omar M. Shehata, and Elsayed I. Morgan. 2019. Design of various dynamical-based trajectory tracking control strategies for multi-vehicle platooning problem. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC).* 1631–1637.
[37] Darbha Swaroop and K. R. Rajagopal. 2001. A review of constant time headway policy for automatic vehicle following. In *Proceedings of the 2001 IEEE Intelligent Transportation Systems.* IEEE, 65–69.
[38] Feng Tan, Yufei Wang, Qixin Wang, Lei Bu, and Neeraj Suri. 2015. A lease based hybrid design pattern for proper-temporal-embedding of wireless cps interlocking. *IEEE Transactions on Parallel and Distributed System* 26, 10 (2015), 2630–2642.
[39] Christophe Viel, Ulysse Vautier, Jian Wan, and Luc Jaulin. 2019. Platooning control for heterogeneous sailboats based on constant time headway. *IEEE Transactions on Intelligent Transportation Systems* 21, 5 (2019), 2078–2089.
[40] Meng Wang. 2018. Infrastructure assisted adaptive driving to stabilise heterogeneous vehicle strings. *Transportation Research Part C* 91, June (2018), 276–295.
[41] Yibing Wang, Elias B. Kosmatopoulos, Markos Papageorgiou, and Ioannis Papamichail. 2014. Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study. *IEEE Transactions on Intelligent Transportation Systems* 15, 5 (2014), 2024–2039.
[42] Yunpeng Wang, E. Wenjuan, Wenzhong Tang, Daxin Tian, Guangquan Lu, and Guizhen Yu. 2013. Automated on-ramp merging control algorithm based on internet-connected vehicles. *IET Intelligent Transport Systems* 7, 4 (2013), 371–379.
[43] Ziran Wang, Guoyuan Wu, and Matthew Barth. 2018. *Distributed Consensus-Based Cooperative Highway On-Ramp Merging Using V2X Communications.* Technical Report. SAE Technical Paper.
[44] Yuanchang Xie, Huixing Zhang, Nathan H. Gartner, and Tugba Arsava. 2017. Collaborative merging strategy for freeway ramp operations in a connected and autonomous vehicles environment. *Journal of Intelligent Transportation Systems* 21, 2 (2017), 136–147.
[45] Guangchuan Yang and Zong Tian. 2019. An exploratory investigation of the impact of ramp metering on driver acceleration behavior. *IATSS Research* 42, 4 (2019), 277–285.
[46] Masashi Yoshida, Hiromu Asahina, Hiroshi Shigeno, and Iwao Sasase. 2020. A scheduling scheme for cooperative merging at a highway on-ramp with maximizing average speed of automated vehicles. In *Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall).* IEEE, 1–5.
[47] Xiaohai Yu, Ge Guo, and Hongbo Lei. 2018. Longitudinal cooperative control for a bidirectional platoon of vehicles with constant time headway policy. In *Proceedings of the 2018 Chinese Control And Decision Conference (CCDC).* IEEE, 2427–2432.

Received 4 October 2021; revised 13 May 2023; accepted 19 June 2023