COOPERATIVE DRIVING FOR CONNECTED AND AUTOMATED VEHICLES ON DEDICATED LANES

FAN XUELI

PhD

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University
Department of Computing

Cooperative Driving for Connected and Automated Vehicles on Dedicated Lanes

Fan Xueli

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
April 2022

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Fan Xueli

Abstract

Autopiloting of smart vehicles is a hot topic in both industry and academia. Market for autopiloting smart vehicles is expected to reach over 500 billion US dollars by 2026. In academia, the *Society of Automotive Engineers* (SAE) defines five levels of automated drivig, with the ultimate goal of full scale (i.e. level 5) autopiloting. Expectedly autopiloting can eliminate human errors, hence greatly improve driving safety and reduce property and medical costs caused by transportation accidents.

One promising context to first realize autopiloting is in dedicated lanes, where human driven vehicles are forbidden, just like modern highways forbid horse carriages. In fact, the SAE level 4 automated driving is explicitly defined to be realized in limited spatial areas (in other words, dedicated lanes); and this level 4 is considered as an inevitable stepping stone toward level 5 (full scale) autopiloting.

By forbidding human driven vehicles (hence eliminating the unpredictability caused), the dedicated lanes make cooperative driving of *Connected Automated Vehicles* (CAVs) meaningful. Intuitively, cooperative driving of CAVs can make autopiloting easier, safer, and more efficient.

However, this vision is challenged by the inborn unreliability of wireless communications. Wireless communication failures, both transient and persistent, can happen randomly, due to various reasons, such as handover failures, jamming, large-scale path losses, multipath. Such failures can cause arbitrary wireless packet losses, leaving the CAV driving cooperation in inconsistent states, hence cause further failures, even

accidents.

As part of the endeavor to address this challenge, in this dissertation, we try to tackle the problem from protocol design and formal analyses perspective. By exploiting the design philosophy of time out (aka leasing), and the formal tool of hybrid automata, we propose two protocols, respectively for V2X (i.e. vehicle to everything wireless communications) highway and metered-ramp merging, and V2V (i.e. vehicle to vehicle wireless communications) highway lane change.

We formally prove the two protocols can guarantee the widely adopted *Constant Time Headway* (CTH) safety, as well as liveness (i.e. no deadlock), under arbitrary wireless packet losses.

These theoretical conclusions on safety and liveness are validated by our simulations. Furthermore, our simulations also show great performance improvements. For the highway metered-ramp merging protocol, our simulations show more than 99% merging success rate improvements in 11 out of 18 comparison pairs, and 0% (i.e. tied) to 71% merging success rate improvements in the remaining 7 comparison pairs. For the highway lane change protocol, our simulations show 8.5% to 81.8% (median: 36.9%, mean: 39.7%) lane change success rate improvements in 18 comparison pairs.

Publications Arising from the Thesis

TCAD22 He Z, Fan X, Peng Y, et al. EmPointMovSeg: Sparse Tensor Based Moving Object Segmentation in 3D LiDAR Point Clouds for Autonomous Driving Embedded System[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022.

ICCPSWiP20 Xueli Fan, Qixin Wang, Jie Liu, "Work-in-Progress Abstract: A Reliable Wireless Smart Vehicle Highway On-Ramp Merging Protocol with Constant Time Headway Safety Guarantee," in ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS'20) Work-in-Progress Session, pp.184-185, Sydney, Australia, April, 2020.

ICCPSWiP18 Jiajun Shen, Xueli Fan, Qixin Wang, "WiP Abstract: Underwater AUV localization with refraction consideration," in ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS'18) Work-in-Progress Session, pp.333-334, Porto, Portugal, April, 2018.

Submitted Journal Papers for Review

Merging21 Xueli Fan, Qixin Wang, Jie Liu, "A Reliable Wireless Protocol for Highway and Metered-Ramp CAV Collaborative Merging with Constant-Time-Headway

Safety Guarantee," submitted.

LaneChange22 Xueli Fan, Qixin Wang, "A CAV Lane Change Protocol with CTH Safety Guarantee for Cooperative Driving on Dedicated Highways," submitted.

Other Publications

TIMC18 Fan X, Yin X, An M, et al. Consensus in multi-agent system with communication constraints based on asynchronous dynamics[J]. Transactions of the Institute of Measurement and Control, 2018, 40(4): 1109-1126.

Acknowledgments

First of all, I would like to thank my Chief Supervisor, Dr. Qixin Wang, who supported me to pursue the Doctor of Philosophy. He has shown me the value of good problem formulations and intuitions in doing research. He encouraged me when I was in difficult situations and was supportive when I made decisions on the research directions. His insightful feedback pushed me to sharpen my thinking and brought my research to a higher level. He always help me to improve the scientific writing and presentation. His kind guidance and great help were indispensable for the achievement of this dissertation.

I am grateful for the short but fantastic visit in Germany. Thanks to Prof. Neeraj Suri and Dr. Stefan Winter for the time I spent at TU Darmstadt, and the great discussions that we had. It was an intensive, but quite enjoyable research experience.

I would particularly like to thank Prof. Lui Sha from the University of Illinois at Urbana-Champaign, and Prof. Jie Liu from Harbin Institute of Technology, for giving me helpful advice for my research. Many thanks to Prof. Guiling Wang, Prof. Minghua Chen, and Prof. Zhen Leng for spending their precious time to discuss my research.

I would like to acknowledge my groupmates, Dr. Zhijian He, Dr. Zhu Wang and Mr. Jiajun Shen, for their wonderful collaborations and patient support. When I was lost and confused, it is their immense care, support and encouragement that give me the courage to face all the difficulties.

In addition, I would like to thank my family for their immense love. Their understanding and support have been a great help to me during my PhD years in Hong Kong. They will accompany me for ups and downs, and they are my reasons to keep moving forward.

Finally, I would like to thank all of my friends in the Hong Kong Polytechnic University. They gave me suggestions and help for my dissertation and provided me happy memories in Hong Kong.

Table of Contents

Abstract **i**

Publications Arising from the Thesis **iii**

Acknowledgments **v**

List of Figures **xi**

List of Tables **xiii**

1 Introduction **1**

2 Background **5**

2.1 Vehicle Dynamic Models 5

2.1.1 Lateral Vehicle Dynamics 6

2.1.2 Longitudinal Vehicle Dynamics 8

2.2 Wireless Communication 10

2.3 Hybrid Automata 12

2.4 Chapter Summary 14

3 A Reliable Wireless Protocol for Highway and Metered-Ramp CAV Cooperative Merging **15**

3.1 Demand 15

3.2 Related Work 18

3.3 Problem Formulation 19

3.3.1 Assumptions on Driving Dynamics 20

3.3.2 Merging Scenario and CTH Safety Rule 21

3.4 Solution 25

3.4.1 Heuristics 25

3.4.2 Proposed Protocol 28

3.4.3 Analysis 34

3.5 Important Observations 40

3.6 Evaluation 41

3.6.1 Simulation Configuration 43

3.6.2 Safety 44

3.6.3 Liveness (Automatic Reset) 44

3.6.4 Merging Success Rate and Time Cost 45

3.7 Chapter Summary 46

4 A CAV Lane Change Protocol with CTH Safety Guarantee for Cooperative Driving on Dedicated Highways **50**

4.1 Demand 51

4.2 Related Work 53

4.3 Problem Formulation 56
4.3.1 Lane Change Scenario 56
4.3.2 Assumptions on CAV Driving Dynamics 57
4.3.3 Other Assumptions, CTH Safety Rule, and the Research Problem 60
4.4 Solution 62
4.4.1 Heuristics 62
4.4.2 Proposed Protocol 65
4.4.3 Analysis 80
4.5 Further Discussions 87
4.5.1 V2V Communication Failures between Target Lane CAVs are Irrelevant 87
4.5.2 Vehicle Body Shape 88
4.6 Evaluation 89
4.6.1 Simulation Configuration 89
4.6.2 Comparison Baselines 90
4.6.3 CTH-Δ^* Safety 91
4.6.4 Liveness (Automatic Resetting) 92
4.6.5 Lane Change Success Rate and Time Cost 93
4.7 Chapter Summary 95

5 Conclusions and Future Work **96**

5.1 Conclusions 96
5.2 Future Work 97

A Proofs for Chapter 3 **99**

A.1 List of Symbols Used in Chapter 3 99

A.2 Proof of Lemma 4 101

A.3 Proof of Lemma 5 103

A.4 Proof of Lemma 6 104

B Proofs for Chapter 4 **108**

B.1 Proof of Lemma 10 108

B.2 Proof of Lemma 11 111

B.3 Proof of Lemma 12 112

References **121**

List of Figures

2.1 Dynamics of Bicycle Model 6

2.2 V2X communication scenario 12

2.3 Thermostat automaton 14

3.1 The highway and metered-ramp merging scenario 22

3.2 Automata Sketches 26

3.3 Hybrid automaton A_{BS} for the base station 32

3.4 Hybrid automaton A_r for the ramp CAV 33

3.5 Hybrid automaton A_i for the highway CAV 35

4.1 Lane Change Scenario 56

4.2 Automata Sketches. 63

4.3 Hybrid automaton A_R 71

4.4 Hybrid automaton A_i 76

4.5 Statistics of time headways 91

4.6 Statistics of reset time costs 92

4.7 Statistics of lane change *success rates* (sr) and time costs 94

B.1 time-position trajectory diagram of V_{k-1} and V_k 109

B.2 time-position trajectory diagram of V_{k-1} and V_k 110

B.3 The entire lane change process: $d_0(L, R) < D_1$ and $d_0(R, F) \geq D_2$ . . 114

B.4 The entire lane change process: $d_0(L, R) \geq D_1$ and $d_0(R, F) < D_3$ . . 116

B.5 The entire lane change process: $d_0(L, R) < D_1$ and $d_0(R, F) < D_2$ . . 118

List of Tables

3.1 Simulation Results: Time Headway 47

3.2 Simulation Results: Reset Time Cost 48

3.3 Simulation Results: Merging Success Rate and Time Cost 49

Chapter 1

Introduction

Various studies throughout the late 1970s and 2010s show that about 90% of vehicle crashes are at least partially caused by human errors [71][70][89]. Particularly, as per the US Department of Transportation National Highway Traffic Safety Administration (NHTSA)'s 2015 nation-wide survey [71], as high as 94% of car crashes involving light vehicles can be attributed to human driver errors.

This piqued great interest to replace human drivers with automated driving, aka *autopiloting*. By eliminating human drivers, people hope autopiloting can eliminate all human errors. If this vision becomes true, it can save more than 30,000 lives each year in the United States alone, and reduce an annual medical cost of USD190 billion [81].

Therefore, it is no wonder that autopiloting has become a hot topic in both industry and academia. Annual market for autopiloting is expected to reach over 500 billion US dollars by 2026, and over 11 million units of sales by 2033 [103]. In academia, the *Society of Automotive Engineers* (SAE) defines six levels (from level 0 to level 5) of automated driving [42], with the ultimate goal of full scale (i.e. level 5) autopiloting.

One promising context to first realize autopiloting is on dedicated lanes. As Prof.

Fei-Yue Wang, the former president of the IEEE Intelligent Transportation Systems Society and editor-in-chief (2009-2016) of the IEEE Transactions on Intelligent Transportation Systems, put it in a keynote speech [94]: just as modern automotive highways forbids horse carriages, future autopiloting highways shall also forbid human drivers. Coinciding with this vision, the SAE six levels (level 0 to 5) of automated driving road map [42] also specifies level 4, a stepping stone toward level 5 (i.e. full scale autopiloting), to be conducted in limited spatial areas, in other words, dedicated lanes. The trend to deploy dedicated lanes for autopiloting is gaining increasing support in academia, industry, and society [24][25][52][75][35][1][43][96]. For example, the state of Michigan plans to build or assign dedicated lanes for automated vehicles on a 40-mile stretch of highway between Detroit and Ann Arbor [52]. Also, six Chinese cities (including Beijing) have announced roads designated for tests of automated vehicles. Particularly, in Beijing, approximately 206 miles of roads are assigned for testing [108].

By forbidding human drivers, hence eliminating the unpredictability caused, dedicated lanes make cooperative driving of *Connected Automated Vehicles* (CAVs) meaningful [76][63][60][67]. Intuitively, cooperative driving of CAVs can make autopiloting easier, safer, and more efficient [33][38][112][31][88].

However, this vision is challenged by the inborn unreliability of wireless communications (which are inevitable to connect moving vehicles). Wireless communication failures, both transient and persistent, can happen randomly, due to various reasons, such as handover failures, jamming, large-scale path losses, multipath. Such failures can cause arbitrary wireless packet losses, leaving the CAV driving cooperation in inconsistent states, hence cause further failures, even accidents.

In this paper, we aim to initiate an endeavor to address the wireless communications unreliability challenge in CAV coorperative driving in dedicated lanes.

The entire problem space of the challenge is huge, and cannot be completely addressed

in one single dissertation. As a first step, we focus on the following two problems, to explore approaches for future efforts.

The first problem is to guarantee the *Constant Time Headway* (CTH) safety rule, a well adopted driving safety rule [65][92][18][110][26][11][85], under arbitrary wireless packet losses, in the highway and *metered-ramp* [49][106][30][2][100] CAV cooperative merging scenario. Vehicle-to-infrastructure (V2I) communications are the main paradigm of wireless communications in this scenario, where CAVs are coordinated by a base station at the highway/metered-ramp merge point.

The second problem is to guarantee the CTH safety rule under arbitrary wireless packet losses, in the highway lane change scenario. Vehicle-to-vehicle (V2V) communications are the main paradigm of wireless communications in this scenario.

In line with previously defined problems, the main contributions of this thesis are:

1. For the first problem, 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. Specifically, the merging success rate are more than 99% better in 11 out of 18 comparison pairs, and 0% (i.e. tied) to 71% better in the remaining 7 comparison pairs.

2. For the second problem, we propose a cooperative lane change protocol, and formally prove its guarantee of the CTH safety and liveness under arbitrary wireless packet losses. These theoretical claims are further validated by our simulations. Furthermore, the simulation results also show the proposed protocol can perform better than other alternatives. Specifically, the lane change success rates are 8.5% to 81.8% better in all the 18 comparison pairs, with the median of 36.85% and mean of 39.7%.

The rest of this dissertation is organized into the following chapters.

Chapter 2 presents background knowledge used in this dissertation. These knowledge includes vehicle dynamic models (covering both lateral and longitudinal dynamics), CAV wireless communication technologies, and hybrid automata (the formal tool to specify protocols).

Chapter 3 proposes a timeout based dedicated lane CAV cooperative driving protocol for highway and metered-ramp merging. We formally prove the safety (i.e. guarantee of the CTH safety rule) and liveness of this protocol under arbitrary wireless packet losses; and carry out extensive simulations to further verify and evaluate this protocol.

Chapter 4 proposes a timeout based dedicated lane CAV cooperative driving protocol for lane change. We formally prove the CTH safety guarantee and liveness of this protocol under arbitrary wireless packet losses; and carry out simulations to verify and evaluate this protocol.

Chapter 5 concludes the dissertation and discusses the future work.

Chapter 2

Background

This chapter presents some basic theoretical foundations for the following chapters. As a basic theory of the vehicle industry, the vehicle dynamics plays an important role in the development of the vehicle industry. Vehicle dynamic models are introduced in section 2.1. Connected and Automated Vehicle (CAV) either communicates with each other or connects with traffic signals, signs, and other road items. The corresponding wireless communication technology is explained in section 2.2. CAV system is a hybrid system which combines discrete and continuous dynamics. We present hybrid automata as formal models that define behaviors of hybrid systems. Basic hybrid automata concepts are discussed in section 2.3.

2.1 Vehicle Dynamic Models

This section provides information on dynamics modeling of vehicle. We will use a nonlinear vehicle model including decoupled lateral and longitudinal dynamics, summarized from [78][48][50],

2.1.1 Lateral Vehicle Dynamics

According to [78][48][50], a "bicycle" model of the vehicle is considered, as shown in Fig.2.1. (X, Y) and (x, y) represent the global coordinates and body-fixed coordinates, respectively. We first consider the lateral vehicle dynamics represented by the vehicle lateral position y and the vehicle yaw angle ψ. The longitudinal velocity of the vehicle at the center of gravity (c.g.) is denoted by V_x. Consider a vehicle

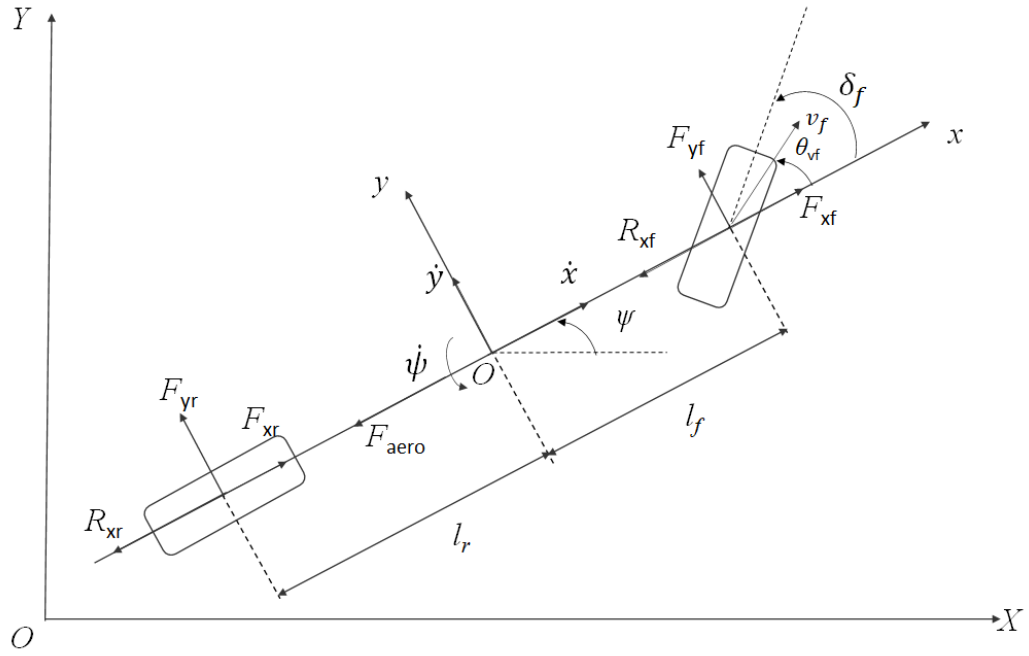


Figure 2.1: Dynamics of Bicycle Model

traveling with constant longitudinal velocity V_x, the differential equations applying Newton's second law are given by [78],

$$\begin{aligned} m(\ddot{y} + \dot{\psi} V_x) &= F_{\text{yf}} + F_{\text{yr}} \\ I_z \ddot{\psi} &= l_f F_{\text{yf}} - l_r F_{\text{yr}} \end{aligned} \tag{2.1}$$

where l_f and l_r are the distances of the front tire and the rear tire respectively from the c.g. of the vehicle. m and I_z denote the vehicle's mass and yaw moment of inertia,

respectively. F_{yf} and F_{yr} denote the lateral tire forces at the front and rear wheels, respectively.

The next step is to model the lateral tire forces. Experimental results show that the lateral tire force of a tire is proportional to the "slip-angle" for small slip-angles. The slip angle of a tire is defined as the angle between the orientation of the tire and the orientation of the velocity vector of the wheel(see Fig. 2.1). In Fig. 2.1, the slip angle of the front wheel is:

$$\alpha_f = \delta_f - \theta_{\mathsf{vf}}$$

where θ_{vf} is the angle that the velocity vector makes with the longitudinal axis of the vehicle and δ_f is the front wheel steering angle.

The rear slip angle is similarly given by:

$$\alpha_r = -\theta_{\mathsf{vr}}$$

Since in most vehicles the rear wheels cannot be steered, we assume $\delta_r = 0$.

The lateral tire force for the front and rear wheels of the vehicle can therefore be written as:

$$\begin{aligned} F_{\mathsf{yf}} &= 2C_{\alpha f}(\delta_f - \theta_{\mathsf{vf}}) \\ F_{\mathsf{yr}} &= 2C_{\alpha r}(-\theta_{\mathsf{vr}}) \end{aligned} \tag{2.2}$$

where the proportionality constants $C_{\alpha f}$ and $C_{\alpha r}$ are called the cornering stiffness of front and rear tire, respectively. The factor 2 accounts for the fact that there are two front wheels.

Using small angle approximations and using the notation $V_y = \dot{y}$, the following relations can be used to calculate θ_{vf} and θ_{vf}.

$$\begin{aligned} \theta_{\mathsf{vf}} &= \frac{\dot{y} + l_f\dot{\psi}}{V_x} \\ \theta_{\mathsf{vr}} &= \frac{\dot{y} - l_r\dot{\psi}}{V_x} \end{aligned} \tag{2.3}$$

Then substituting Eq. (2.3) into Eq. (2.2), the lateral tire forces can be written as:

$$\begin{aligned} F_{\mathsf{yf}} &= 2C_{\alpha f}(\delta_f - \frac{\dot{y} + l_f\dot{\psi}}{V_x}) \\ F_{\mathsf{yr}} &= 2C_{\alpha r}(-\frac{\dot{y} - l_r\dot{\psi}}{V_x}) \end{aligned} \tag{2.4}$$

Substituting Eq. (2.4) into Eq. (2.1), the state space model can be written as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{Bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{I_zV_x} & 0 & -\frac{2l_f^2C_{\alpha f} + 2l_r^2C_{\alpha r}}{I_zV_x} \end{bmatrix} \begin{Bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{Bmatrix} + \begin{Bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2l_fC_{\alpha f}}{I_z} \end{Bmatrix} \delta_f \tag{2.5}$$

The dynamic model described in Eq. (2.5) is based on body fixed coordinates. To obtain a global picture of the trajectory traversed by the vehicle, however, the time history of the body-fixed coordinates must be converted into trajectories in inertial space as the following transition.

$$\begin{aligned} \dot{Y} &= \dot{x}sin\psi + \dot{y}cos\psi \\ \dot{X} &= \dot{x}cos\psi - \dot{y}sin\psi \end{aligned} \tag{2.6}$$

where $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral velocity in the body frame, respectively.

2.1.2 Longitudinal Vehicle Dynamics

According to [78] [48] [50], consider a vehicle moving on a road without the influence of road bank angle as shown in Fig. 2.1. The longitudinal motion equation applying Newton's second law yields:

$$m\ddot{x} = F_{\mathsf{xf}} + F_{\mathsf{xr}} - F_{\mathsf{aero}} - R_{\mathsf{xf}} - R_{\mathsf{xr}} \tag{2.7}$$

where m is the mass of the vehicle, F_{xf} and F_{xr} are the longitudinal tire force at the front and rear tires, respectively. F_{aero} is the equivalent longitudinal aerodynamic drag force, and R_{xf} and R_{xr} are the force due to rolling resistance at the front and rear tires, respectively.

The next step is to model the longitudinal forces:

1. The longitudinal tire forces F_{xf} and F_{xr} are friction forces from the ground that act on the tires.

   - **Case 1** If the longitudinal slip ratio is small (typically less than 0.1 on dry surface) or it is during normal driving, the longitudinal tire force is proportional to the slip ratio. The tire force in this small-slip region can then be modeled as [78]:

     $$\begin{aligned} F_{\mathsf{xf}} &= C_{\sigma f}\sigma_{\mathsf{xf}} \\ F_{\mathsf{xr}} &= C_{\sigma r}\sigma_{\mathsf{xr}} \end{aligned} \tag{2.8}$$

     where $C_{\sigma f}$ and $C_{\sigma r}$ are called the longitudinal tire stiffness parameters of the front and rear tires respectively. σ_{xf} and σ_{xr} are the longitudinal slip ratio of the front and rear tires respectively.

     The longitudinal slip ratio of front tire is defined as:

     $$\sigma_{\mathsf{xf}} = \begin{cases} \dfrac{r_{\mathsf{eff}}w_{\mathsf{wf}} - \dot{x}}{\dot{x}}, & \text{during braking;} \\ \dfrac{r_{\mathsf{eff}}w_{\mathsf{wf}} - \dot{x}}{r_{\mathsf{eff}}w_{\mathsf{wf}}}, & \text{during accelerating.} \end{cases} \tag{2.9}$$

     where r_{eff} is the effective tire radius, w_{wf} is the rotational angular velocity of the front wheel, and $\dot{x}$ is the longitudinal vehicle velocity. The longitudinal slip ratio of rear tire can be defined similarly, just replace the w_{wf} to be w_{wr}.

   - **Case 2** If the longitudinal slip ratio is not small or if the road is slippery, then a nonlinear tire model needs to be used to calculate the longitudinal tire force. The Pacejka "Magic Formula" model or the Dugoff tire model

can be used to model tire forces in this case. Due to the limited space, interested readers can refer to [29].

2. The equivalent aerodynamic drag force on a vehicle can be represented as:

$$F_{\mathsf{aero}} = \frac{1}{2}\rho C_d A_F (\dot{x} + V_{\mathsf{wind}})^2 \tag{2.10}$$

where ρ is the mass density of air, C_d is the aerodynamic drag coefficient, A_F is the frontal area of the vehicle, which is the projected area of the vehicle in the direction of travel, $\dot{x}$ is the longitudinal vehicle velocity, V_{wind} is the wind velocity (positive for a headwind and negative for a tailwind).

3. The rolling resistance that acts to oppose the motion of the vehicle can be represented as:

$$\begin{gathered} R_{\mathsf{xf}} + R_{\mathsf{xr}} = f(F_{\mathsf{zf}} + F_{\mathsf{zr}}) \\ \text{where the front normal tire force is : } F_{\mathsf{zf}} = \frac{-F_{\mathsf{aero}} h_{\mathsf{aero}} - m\ddot{x}h + mgl_r}{l_f + l_r} \\ \text{where the rear normal tire force is : } F_{\mathsf{zr}} = \frac{F_{\mathsf{aero}} h_{\mathsf{aero}} + m\ddot{x}h + mgl_f}{l_f + l_r} \end{gathered} \tag{2.11}$$

where m is the mass of the vehicle, g is the acceleration due to gravity, f is the rolling resistance coefficient. h is the height of the c.g. of the vehicle. h_{aero} is the height of the location at which the equivalent aerodynamic force acts. l_f and l_r are the longitudinal distance of the front and rear axle from the c.g. of the vehicle, respectively.

2.2 Wireless Communication

Due to some physical limitations of the perception sensors, such as their limited range and field of view, or due to other important parameters such as their degraded performance because of bad weather conditions and their significant cost, wireless communication is examined in order to complement or even substitute these sensors [32].

There are three main types of communication used in vehicular environments [47]: vehicle to vehicle (V2V), vehicle to infrastructure (V2I) and vehicle to everything (V2X) communication.

Vehicle-to-vehicle (V2V) communication enables vehicles to transmit and receive information wirelessly between two vehicles. In this scenario, vehicles will exchange information about position, speed, direction, or other attributes with other surrounding vehicles. Then receiving vehicles will aggregate these messages and make smart decisions to avoid collision. Vehicle-to-Infrastructure (V2I) communication enables vehicles to send and receive information wirelessly between vehicles and traffic infrastructure or road side units (RSUs). In this scenario, the infrastructure will assist with coordination by collecting information on traffic and road conditions and subsequently recommending particular behaviors to a group of vehicles. Vehicle-to-everything (V2X) communication encompasses V2V, V2I, V2N (Vehicle to Network), V2P (Vehicle-to -Pedestrian) technology. Generally, V2X communication scenario involving V2V and V2I communication is shown in Fig. 2.2. In this scenario, traffic infrastructure and all road users, including cars, trucks, motorcycles and pedestrians securely exchange messages to mark their position, speed, direction and other attributes. V2X technology dramatically improve traffic safety and efficiency.

These communication technologies will use Dedicated Short Range Communications (DSRC) to exchange data packets, with nearby vehicles and entities between $300-500$ meters range. Messages will be sent up to 10 times per second providing a $360-$degree view of proximity, with on-board applications using the information for triggering alerts and warnings [36].

The insertion of the wireless communication in the CAVs makes the analysis and design of cooperative protocol complex. Some packets not only suffer transmission delay but, even worse, can be lost during transmission. Thus, how to design the cooperative protocol under arbitrary packet loss and how such packet losses affect the performance of CAVs must be considered.

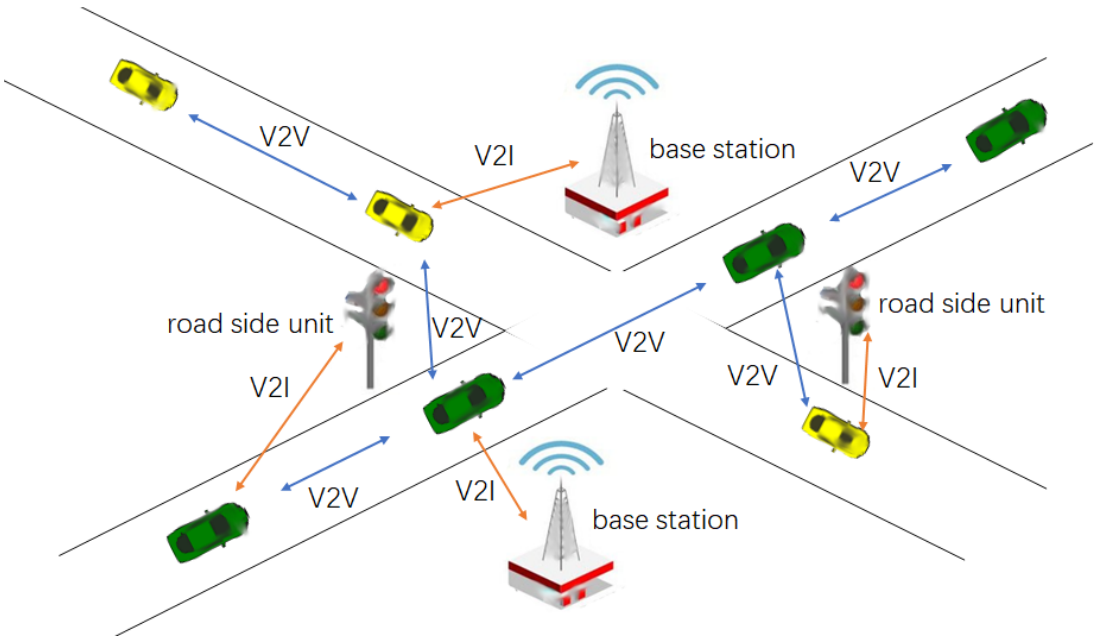


Figure 2.2: V2X communication scenario involving V2V and V2I communication.

2.3 Hybrid Automata

A Cyber Physical System (CPS) is an example of hybrid systems since it has both continuous and discrete variables associated with the physical process (the plant) and the logical dynamics (the control logic and external environment), respectively. A hybrid automaton is widely used as a model of hybrid systems [40]. In this section, we introduce the main concepts of hybrid automata for modeling hybrid systems [80].

A hybrid automaton $\mathcal{H}$ is a tuple of $(\mathsf{Mod}, \sum, \mathsf{Edge}, X, \mathsf{Init}, \mathsf{Inv}, \mathsf{Flow}, \mathsf{Jump})$, where:

- Mod is a finite set of control *modes* in the hybrid system.
- $\sum$ is a finite set of *event names* in the hybrid system.
- $\mathsf{Edge} \subseteq \mathsf{Mod} \times \sum \times \mathsf{Mod}$ is a finite set of *labeled edges*, which represents discrete changes of control mode in the hybrid system. Those changes are labelled by *event names* taken from the finite set of labels $\sum$.

- X is a finite set $\{x_1, x_2, \ldots, x_m\}$ of *real-valued variables.*
  $\dot{X}$ is a finite set $\{\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_m\}$ of the associated dotted variables (which represent first derivatives during continuous evolutions (inside a mode)).
  X' is a finite set $\{x'_1, x'_2, \ldots, x'_m\}$ of the associated primed variables, (which represent updates at the conclusion of discrete changes (from one control mode to another)).

- Init, Inv, and Flow are functions that assign three predicates to each mode.
  - Init(m) is a predicate whose free variables are from X. It states the possible *initial valuations* when the automaton starts in mode m;
  - Inv(m) is a predicate whose free variables are in X. It states the possible *valuations* when the control of the automaton lies in mode m;
  - Flow(m) is a predicate whose free variables are in $X \cup X'$. It states the possible *continuous evolutions* when the control of the automaton is in mode m;

- Jump is a function that assigns a predicate to each labelled edge.
  Jump(e) is a predicate whose free variables are in $X \cup X'$. It states when the *discrete jump* is possible and what is its effect on the continuous variables.

In the following, we use an example to illustrate the concepts of hybrid automaton components. The execution of a hybrid automaton results in continuous change (flows) and discrete change (jumps). The hybrid automaton of Fig 2.3 models a thermostat, the variable x represents the temperature.

- *Initially*, the heater is off and the temperature is $x = 20$ degrees.

- In *control mode* "Off" ,the heater is off. According to the *flow condition* $\dot{x} = -0.1x$, the temperature falls. According to the *jump condition* $x < 18$, the heater issues a "Turn on" *event* and *jumps* from "Off" to "On".

- In *control mode* "On", the heater is on. According to the *flow condition* $\dot{x} = 5 - 0.1x$, the temperature rises. According to the *jump condition* $x > 22$, the heater issues a "Turn off" *event* and *jumps* from "On" to "Off".

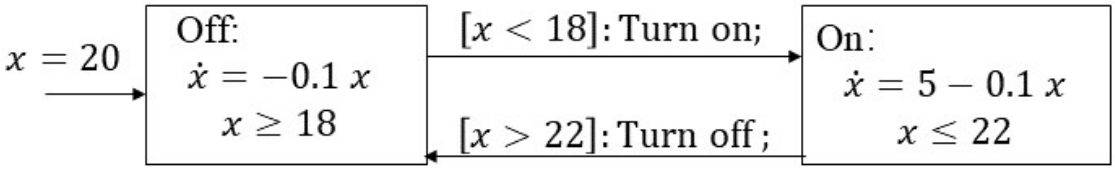


Figure 2.3: Thermostat automaton

2.4 Chapter Summary

This chapter discussed the basic theoretical foundations including vehicle dynamic models, connected vehicle technology, and hybrid automata.

The decoupled lateral and longitudinal vehicle dynamic models were formulated in Section 2.1. The dynamic models discussed in this chapter are useful for cooperative driving applications.

V2V, V2I, and V2X communication technology were introduced in Section 2.2. The wireless communication introduced in this chapter explains the communication between vehicles, transport infrastructure, and other road users.

The basic concepts of hybrid automata were discussed in Section 2.3. The hybrid automata model is able to model the discrete evolution of the controller and the continuous evolution of the vehicle.

Chapter 3

A Reliable Wireless Protocol for Highway and Metered-Ramp CAV Cooperative Merging

In this chapter, we focus on the dedicated lane CAV cooperative driving protocol design for highway and metered-ramp merging. This protocol shall guarantee the *Constant Time Headway* (CTH) safety under arbitrary wireless packet losses. Correspondingly, Section 3.1 presents the demand; Section 3.2 discusses related work; Section 3.3 formulates the problem. Section 3.4 proposes our protocol and formally prove its safety and liveness guarantees; Section 3.5 makes some important observations to relax our assumptions; and Section 3.6 evaluates our proposed protocol.

3.1 Demand

To realize the grand vision of fully automated driving, one of the most promising directions is to first realize it in controlled environments, particularly in highways, where accesses are limited, and *Connected Automated Vehicles* (CAVs) are collabora-

tive [24][3]. One important structure for such limited access highways is the *metered-ramp* [49][106][30][2][100], 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 achieving good merging efficiency.

However, these demands are complicated by the inherently unreliable vehicular wireless communications. Particularly, wireless packets can be lost arbitrarily due to various failures, such as handover failure, jamming, large-scale path loss, and multipath. The arbitrary wireless packet losses can leave the CAV cooperation in inconsistent states, which may lead to further failures, even accidents. Solutions to this problem will heavily depend on the targeted safety rule and the chosen wireless communication paradigm. A panacea solution is highly unlikely. In this chapter, we shall focus on a widely adopted safety rule, the *Constant Time Headway* (CTH) safety [65][92][18][110][26][11][85].

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 [109][20][102][4]. In this chapter, 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 chapter 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 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 3.2), V2X solutions are gaining increasing attention recently. Wang et al. [101] developed 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. [69] propose a cooperative merging system model based on V2V and V2I communication which enables the effective handling of the available gaps between vehicles, and evaluated its performance and impact on highway capacity by adopting a microscopic traffic simulator. Wang et al. [102] 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. [4] 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 reviewed as a more generic V2X) [64][45][66] [14][7]. However, these works also assume the communication infrastructure is reliable, hence do not deal with arbitrary wireless packet losses.

Aoki et al. [6] 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" [34]) 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 made 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 on 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)$\sim$ 71% better in the remaining 7 comparison pairs.

3.2 Related Work

Despite the V2X merging solutions listed in Section 3.1, there is large volume of literature on purely V2V or purely V2I based solutions.

In V2V based approaches, Lu et al. [58][59] propose a virtual vehicle based approach

to ensure sufficient distance for vehicles to merge into highway via a ramp. Hidas [41] classifies the merging maneuvers 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. [104] 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. [46] and Heim et al. [39] 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. [44] use a V2I-based dynamic merge assistance method, to improve merging efficiency and safety. Letter et al. [54] present a longitudinal freeway merging control algorithm for maximizing the average travel velocity of CAVs. Raravi et al. [79] propose an approach for automatic merge control system, where an infrastructure node plans the merging sequences. Pueboobpaphan et al. [74] 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, as mentioned in Section 3.1, 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 existing timeout (aka "lease") based distributed collaboration protocols [86][34]. However, these protocols are not designed for highway and ramp merging, and neither for the CTH safety guarantee.

3.3 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.

3.3.1 Assumptions on Driving Dynamics

Based on the vehicular driving dynamics modeling (see Section 2.1 of Chapter 2), we make the following assumptions.

CAV Acceleration

We assume the CAV acceleration strategy is fixed. Specifically, given the initial speed $v_{\mathsf{a}}^{\mathsf{low}}$ and the target speed $v_{\mathsf{a}}^{\mathsf{high}}$ (in this chapter, we assume vehicles cannot move backward, so unless otherwise denoted, we do not differentiate the concept of speed and velocity), suppose currently the acceleration process has been going on for τ_{a} seconds ($\tau_{\mathsf{a}} \geqslant 0$) and has not yet finished, then the CAV's current acceleration value is fixed, and is a function of $v_{\mathsf{a}}^{\mathsf{low}}$, $v_{\mathsf{a}}^{\mathsf{high}}$, and τ_{a}. Denote this function as $\mathsf{acc}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}, \tau_{\mathsf{a}})$. This function implies that the current *speed* of the CAV is also a function of $v_{\mathsf{a}}^{\mathsf{low}}$, $v_{\mathsf{a}}^{\mathsf{high}}$, and τ_{a}, which can be denoted as $v_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}, \tau_{\mathsf{a}})$. This in turn implies that the total duration and distance needed to accelerate from $v_{\mathsf{a}}^{\mathsf{low}}$ to $v_{\mathsf{a}}^{\mathsf{high}}$ is a function of $v_{\mathsf{a}}^{\mathsf{low}}$ and $v_{\mathsf{a}}^{\mathsf{high}}$. We can denote this duration and this distance to be respectively $\delta_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}})$ and $d_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}})$. Furthermore, we have the following assumption:

Assumption 1 *We assume in the acceleration process as per* $\mathsf{acc}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}, \tau_{\mathsf{a}})$*, the speed will strictly monotonically increase from* $v_{\mathsf{a}}^{\mathsf{low}}$ *to* $v_{\mathsf{a}}^{\mathsf{high}}$*.* ■

CAV Deceleration

Similar to the acceleration case, in this chapter, we assume the CAV deceleration strategy is also fixed. Specifically, given the initial speed $v_{\mathsf{d}}^{\mathsf{high}}$ and the target speed $v_{\mathsf{d}}^{\mathsf{low}}$, suppose currently the deceleration process has been going on for τ_{d} seconds ($\tau_{\mathsf{d}} \geqslant 0$) and has not yet finished, then the CAV's current acceleration value is fixed, and is a function of $v_{\mathsf{d}}^{\mathsf{high}}$, $v_{\mathsf{d}}^{\mathsf{low}}$, and τ_{d}. Denote this function as $\mathsf{dec}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}}, \tau_{\mathsf{d}})$. This function implies that the current *speed* of the CAV is also a function of $v_{\mathsf{d}}^{\mathsf{high}}$,

$v_{\mathsf{d}}^{\mathsf{low}}$, and τ_{d}, which can be denoted as $v_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}}, \tau_{\mathsf{d}})$. This in turn implies that the total duration and distance needed to decelerate from $v_{\mathsf{d}}^{\mathsf{high}}$ to $v_{\mathsf{d}}^{\mathsf{low}}$ is a function of $v_{\mathsf{d}}^{\mathsf{high}}$ and $v_{\mathsf{d}}^{\mathsf{low}}$. We can denote this duration and this distance to be respectively $\delta_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}})$ and $d_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}})$. Furthermore, we have the following assumption:

Assumption 2 *We assume in the deceleration process as per* $\mathsf{dec}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}}, \tau_{\mathsf{d}})$*, the speed will strictly monotonically decrease from* $v_{\mathsf{d}}^{\mathsf{high}}$ *to* $v_{\mathsf{d}}^{\mathsf{low}}$*.* ■

3.3.2 Merging Scenario and CTH Safety Rule

Fig. 3.1 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 metered-ramp intersect at point $\vec{p}_{\mathsf{merge}}$, which cuts the highway lane into two halves: the segment $(-\infty, \vec{p}_{\mathsf{merge}}]$ and the segment $(\vec{p}_{\mathsf{merge}}, +\infty)$. The metered-ramp, on the other hand, has a fixed entrance point $\vec{p}_{\mathsf{enter}}$, which is D_r away from $\vec{p}_{\mathsf{merge}}$ (where $D_r \stackrel{\mathsf{def}}{=} |\vec{p}_{\mathsf{merge}} - \vec{p}_{\mathsf{enter}}|$ is a given configuration constant). Any CAV merging into the highway lane via this metered-ramp must first stop at $\vec{p}_{\mathsf{enter}}$ to wait for permission to start.

Typically, $\vec{p}_{\mathsf{enter}}$ 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 $\vec{p}_{\mathsf{enter}}$ (e.g. assisted by GPS, or simple visual marks painted on the ramp at $\vec{p}_{\mathsf{enter}}$) and waits for a wireless permission message (from certain participants of the collaborative merging) to start.

For the time being, let us abstract every CAV as a point mathematically (see the $\circ$ dots in Fig. 3.1), 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 ($n < +\infty$)

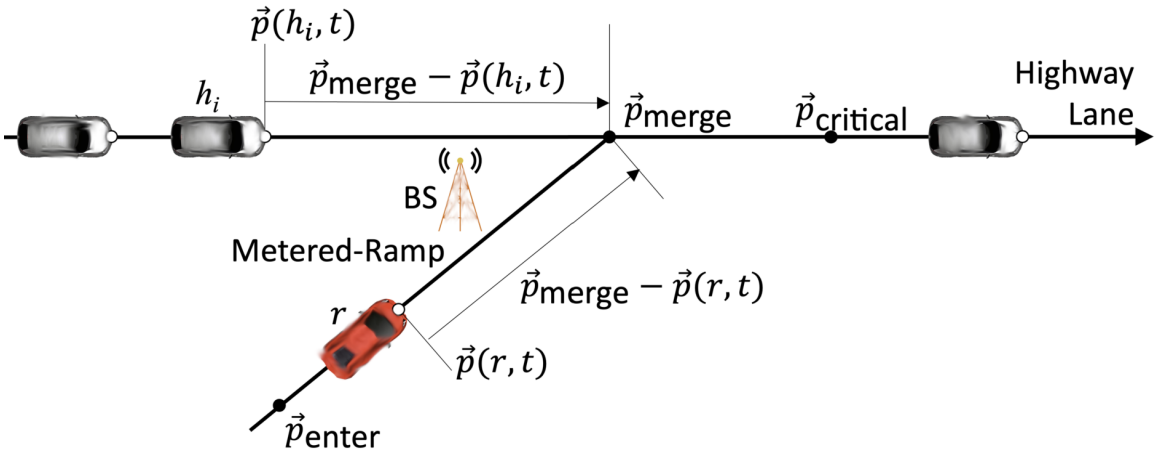


Figure 3.1: The highway and metered-ramp merging scenario. $\vec{p}(x, t)$ is CAV x's location at wall clock time t.

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 $\vec{p}_{\mathsf{enter}}$ 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 $\mathsf{acc}(0,\ v_{\mathsf{rm}},\ \tau_{\mathsf{a}})$ (acc is defined in Section 3.3.1) 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 $\vec{p}_{\mathsf{merge}}$. Correspondingly, Ineq. (3.1) is the configuration prerequisite to make this feasible:

$$d_{\mathsf{a}}(0, v_{\mathsf{rm}}) < D_r \stackrel{\mathsf{def}}{=} |\vec{p}_{\mathsf{merge}} - \vec{p}_{\mathsf{enter}}|, \tag{3.1}$$

where function d_{a} decides the distance needed to accelerate from the given initial speed to the given target speed (see Section 3.3.1, note the corresponding time cost is decided by the function δ_{a}). The duration cost for r from the start of acceleration

to reaching $\vec{p}_{\mathsf{merge}}$ hence is

$$\Delta_r \stackrel{\mathsf{def}}{=} \delta_{\mathsf{a}}(0, v_{\mathsf{rm}}) + \frac{D_r - d_{\mathsf{a}}(0, v_{\mathsf{rm}})}{v_{\mathsf{rm}}}. \tag{3.2}$$

We also give the following configuration prerequisite:

$$0 < v_{\mathsf{rm}} < v_{\mathsf{lim}}. \tag{3.3}$$

Correspondingly, once r reaches $\vec{p}_{\mathsf{merge}}$, it will accelerate again according to $\mathsf{acc}(v_{\mathsf{rm}}, v_{\mathsf{lim}}, \tau_{\mathsf{a}})$ to the speed of v_{lim}. The location on the highway lane where r first reaches v_{lim} hence is fixed. Denote it as $\vec{p}_{\mathsf{critical}}$ (see Fig. 3.1).

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 [83][98][53][15], particularly in the large volume of literature on *Cooperative Adaptive Cruise Control* (CACC) [8][26]. This practice prevails not only for its simplicity, but also for its safety and energy efficiency [53][15][8][26]. On the other hand, the ramp CAV r reaching $\vec{p}_{\mathsf{merge}}$ with v_{rm}, the minimum speed allowed on a highway lane, is a design out of caution. It includes the special case where $v_{\mathsf{rm}} = v_{\mathsf{lim}} - \varepsilon$, 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* $\vec{p}_{\mathsf{merge}}$*, which can coordinate the merging between* r *and* h_1*,* h_2*, ...,* 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* $\vec{p}_{\mathsf{merge}}$*) and speed of any CAV (this assumption will be relaxed in Section 3.5).* ■

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 to guarantee the *Constant Time Headway* (CTH) safety [65][92][18] [110][26][11][85] 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 $\delta(t) \stackrel{\mathsf{def}}{=} d(t)/v_y(t)$ 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 $\Delta^{**} + \frac{D_0}{v^{\mathsf{min}}}$, where Δ^{**} is the maximum duration needed to stop a vehicle at any speed $v \leqslant v^{\mathsf{max}}$ 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 anytime on the lane.

3.4 Solution

In this section, we propose a protocol to realize the aforementioned highway and metered-ramp merging (see Section 3.3.2), and prove its guarantee of the CTH safety and liveness, even under arbitrary wireless packet losses.

3.4.1 Heuristics

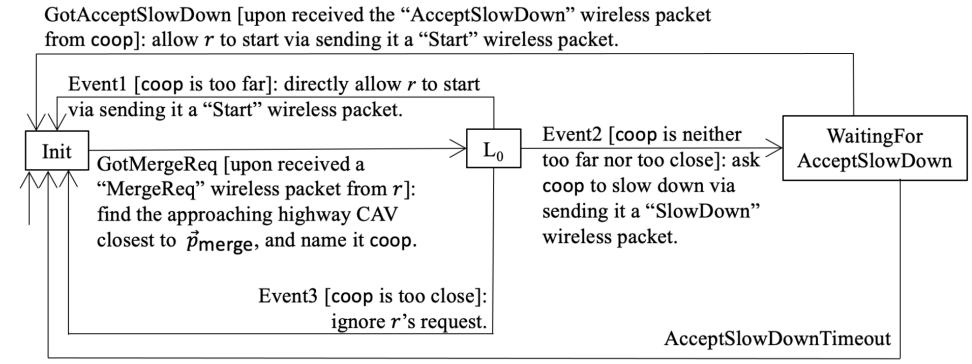
The heuristics of our proposed protocol is illustrated by the automata sketches of Fig. 3.2.

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 Fig. 3.2(b)). If BS receives this wireless packet, it triggers the "GotMergeReq" event (see event "GotMergeReq" in Fig. 3.2(a)). As the *action* (i.e. the handling routine) carried out by this event, BS finds the approaching highway CAV closet to $\vec{p}_{\mathsf{merge}}$, 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 $\vec{p}_{\mathsf{merge}}$. Specifically,

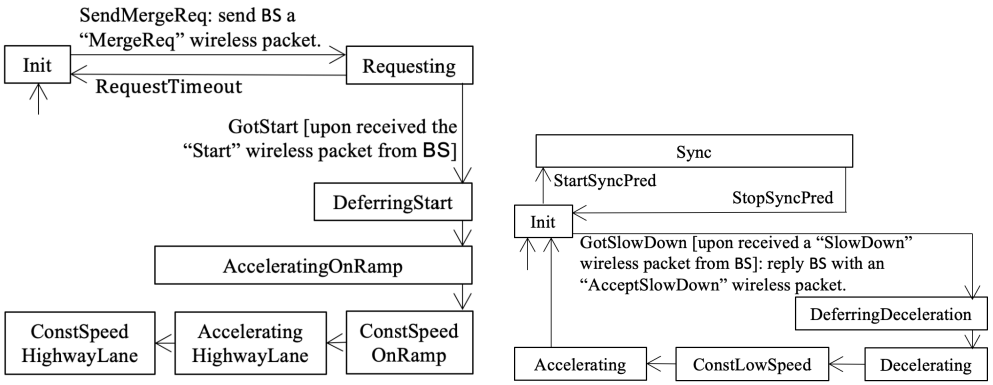
1. If coop is too far away from $\vec{p}_{\mathsf{merge}}$, BS will directly allow r to start by sending it a "Start" wireless packet (see "Event1" in Fig. 3.2(a)).

2. If coop is too close to $\vec{p}_{\mathsf{merge}}$, r's merge request is ignored and r has to request again in the future (see "Event3" in Fig. 3.2(a)).

3. If coop is neither too far away nor too close to $\vec{p}_{\mathsf{merge}}$ (see "Event2" in Fig. 3.2(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 a "AcceptSlowDown" wireless packet and start a deceleration routine (see "Got-



(a) Base Station BS



(b) Ramp CAV r (c) Highway CAVs h_i ($i \in \{1, 2, \ldots, n\}$)

Figure 3.2: 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.

SlowDown" event in Fig. 3.2(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 Fig. 3.2(a)). Upon reception of the "Start" wireless packet (see "GotStart" event in Fig. 3.2(b)), r will start and accelerate. Once r reaches $\vec{p}_{\mathsf{merge}}$, 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 Fig. 3.2(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 Section 3.4.2 and 3.4.3).

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 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 [86][34]. A "*lease*" is an agreement on timeout, contracted since the early stage of a distribute 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 Fig. 3.2, 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 Section 3.4.2 and 3.4.3. The efficiency is evaluated in Section 3.6.

3.4.2 Proposed Protocol

We propose our detailed protocol by expanding the automata sketches of Fig. 3.2 with the heuristics described in Section 3.4.1. The resulted full-fledged hybrid automata [5] A_{BS} (see Fig. 3.3), A_r (see Fig. 3.4), and A_i (see Fig. 3.5) 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 automata are respectively explained in the following (for readers' convenience, a list of the symbols used in this chapter is given in Appendix A.1).

Base Station BS *protocol behaviors (illustrated by hybrid automaton* A_{BS} *in Fig. 3.3)*:

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 $[0, \Delta_{\mathsf{BS}}^{\mathsf{min}}]$ (e.g. as per uniform distribution), where $\Delta_{\mathsf{BS}}^{\mathsf{min}} > 0$ 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 $\Delta_{\mathsf{BS}}^{\mathsf{min}}$ seconds (i.e. $\tau > \Delta_{\mathsf{BS}}^{\mathsf{min}}$), then BS triggers the "GotMergeReq" event. This event carries out the following action (see event "GotMergeReq" in Fig. 3.3):

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

Step2.2 set $\hat{\delta}_{\mathsf{coop}}$ to $|\vec{p}_{\mathsf{merge}} - \vec{p}(\mathsf{coop}, t_1)|/v_{\mathsf{lim}}$, where t_1 is the current wall clock time (i.e. $|\vec{p}_{\mathsf{merge}} - \vec{p}(\mathsf{coop}, t_1)|$ is the current distance between $\vec{p}_{\mathsf{merge}}$ 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 Fig. 3.3):

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

$$\Delta_1 \stackrel{\mathsf{def}}{=} \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}) - \frac{d_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})}{v_{\mathsf{lim}}}, \text{(note Ineq. (3.3) implies } \Delta_1 > 0) \quad (3.4)$$

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 $\vec{p}_{\mathsf{merge}}$, specifically, if $\Delta_r + \Delta^* + \Delta_1 > \hat{\delta}_{\mathsf{coop}} > \Delta_2$, where

$$\Delta_2 \stackrel{\mathsf{def}}{=} \Big(d_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) + v_{\mathsf{rm}}(\Delta_r + \Delta^* - \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}))\Big)/v_{\mathsf{lim}}, \quad (3.5)$$

then BS triggers "Event2." This event carries out the following sequential action: set δ_{defer} to $\hat{\delta}_{\mathsf{coop}} - \Delta_2$; 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" to wait for coop's reply.

Note we enforce the following configuration prerequisite:

$$\Delta^* < \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) < \Delta_r, \quad (3.6)$$

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

$$\Delta_r + \Delta^* + \Delta_1 > \Delta_2, \tag{3.7}$$

because (3.6)

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

Ineq. (3.7) ensures the guards for "Event1" and "Event2" (see Fig. 3.3) are valid and non-overlapping.

Case3 (Event3) Otherwise, i.e. if coop is too close to $\vec{p}_{\mathsf{merge}}$, specifically, $\hat{\delta}_{\mathsf{coop}} \leqslant \Delta_2$, 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 $[0, \mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}]$, where $\Delta_{\mathsf{nonzeno}} > 0$ 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 "AcceptSlowDownTimeout" events respectively in Fig. 3.3):

Case1 (GotAcceptSlowDown) If before τ exceeds $\mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$, an "AcceptSlowDown" wireless packet is received from coop, then BS triggers the "GotAcceptSlowDown" 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 $\mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$, 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 Fig. 3.4):

1. At any time instance, the ramp CAV r dwells in one of the following modes: "Init," "Requesting," "DeferringStart," "AcceleratingOnRamp," "ConstSpeed OnRamp," "AcceleratingHighwayLane," and "ConstSpeedHighwayLane."

2. Initially, r dwells in the "Init" mode, stops at $\vec{p}_{\mathsf{enter}}$ (i.e. $\vec{p}(r,t) = \vec{p}_{\mathsf{enter}}$), and has its local clock τ's initial value set to 0.

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 $\Delta_{\mathsf{nonzeno}}$, 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 $\Delta_{\mathsf{nonzeno}}$, 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 $\Delta_{\mathsf{nonzeno}}$, 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 $\mathsf{acc}(0, \mathsf{v}_{\mathsf{rm}}, \tau)$ ($\tau \in$

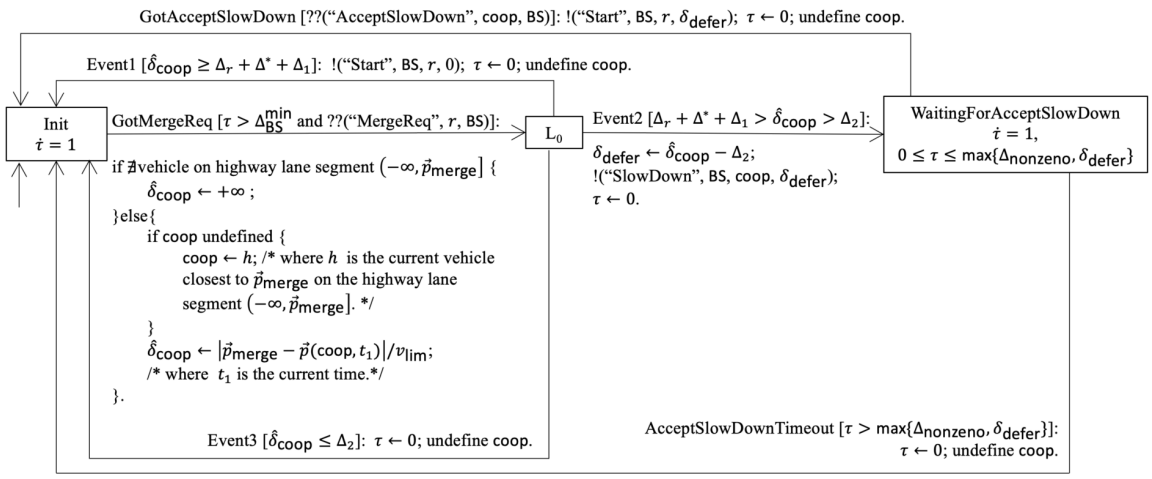


Figure 3.3: Hybrid automaton A_{BS} for the base station BS. Each rectangle box indicates a *hybrid automaton mode* (simplified as "*mode*" in the following). Inside a mode, the top line is the mode's name (it is local to the respective hybrid automata), the rest describes the constraints (e.g. a dwelling duration constraint like $0 \leq \tau \leq \Delta_{\mathsf{nonzeno}}$) 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 second, 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 $[0, \Delta^{\mathsf{min}}_{\mathsf{BS}}]$). 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 "$\leftarrow$" means value assignment. Same notational rules also apply to Fig. 3.4 and Fig. 3.5.

$[0, \delta_{\mathsf{a}}(0, v_{\mathsf{rm}})]$) to v_{rm}, (enter "ConstSpeedOnRamp" mode) maintain this speed till passed $\vec{p}_{\mathsf{merge}}$, (enter "AcceleratingHighwayLane" mode) accelerate as per $\mathsf{acc}(\mathsf{v}_{\mathsf{rm}}, \mathsf{v}_{\mathsf{lim}}, \tau)$ ($\tau \in [0, \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})]$) to v_{lim}, and (enter "ConstSpeed" mode) maintain v_{lim} on the highway lane, finishing the merging.

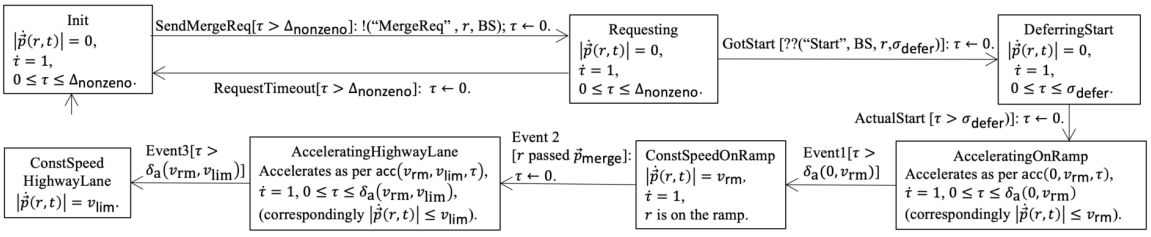


Figure 3.4: Hybrid automaton A_r for the ramp CAV r (τ's initial value is set to 0).

Highway CAV h_i ($i \in \{1, \ldots, n\}$) protocol behaviors (illustrated by hybrid automaton A_i in Fig. 3.5):

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 state 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 Fig. 3.5):

   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 Fig. 3.5): 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

$$D_1 \stackrel{\mathsf{def}}{=} v_{\mathsf{lim}}(\Delta_r + 2\Delta^* + \Delta_1 - \Delta_2), \quad (3.8)$$

(note Ineq. (3.7) implies $D_1 > 0$)

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 $\mathsf{dec}(v_{\mathsf{lim}}, v_{\mathsf{rm}}, \tau)$ ($\tau \in [0, \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}]$) to v_{rm}, (enter "ConstLowSpeed" mode without changing τ) maintain this speed till local clock τ exceeds $\Delta_r + \Delta^*$ seconds (note Ineq. (3.6) implies $\Delta_r + \Delta^* > \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}})$), (enter "Accelerating" mode with local clock τ reset to 0) accelerate as per $\mathsf{acc}(v_{\mathsf{rm}}, v_{\mathsf{lim}}, \tau)$ ($\tau \in [0, \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}]$) 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.

3.4.3 Analysis

We claim the following theorem.

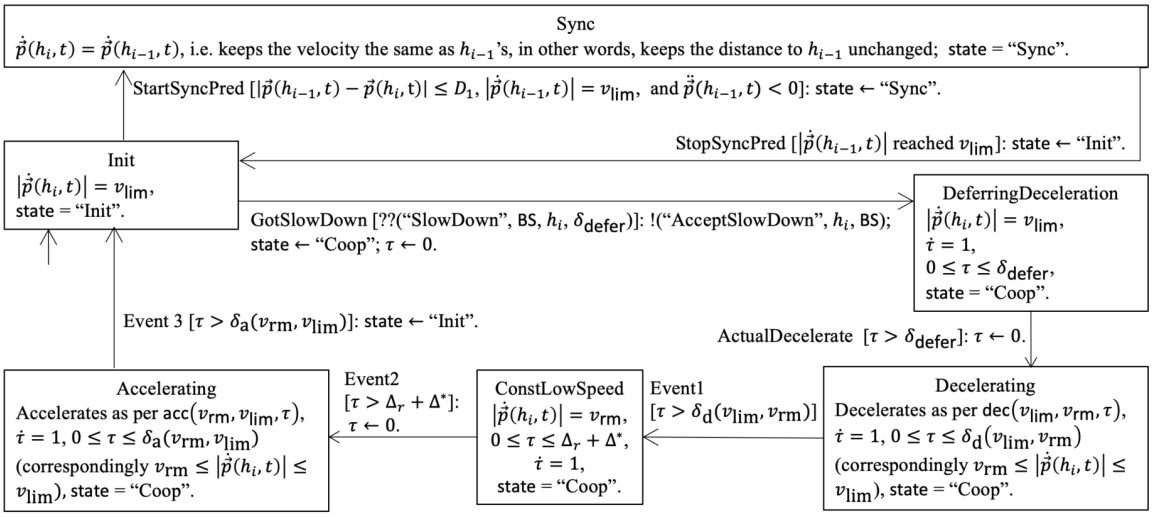


Figure 3.5: Hybrid automaton A_i for the highway CAV h_i ($i \in \{1, 2, \ldots, n\}$); note as h_0 does not exist, for h_1, the event "StartSyncPred" can never happen.

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.* (3.1), (3.3), (3.6), $\Delta^* > 0$, *and* $\Delta_{\mathsf{nonzeno}} > 0$;

(c2) $\Delta^{\mathsf{min}}_{\mathsf{BS}} > \Delta^{\mathsf{max}}_{\mathsf{coop}} + \Delta_{\mathsf{nonzeno}}$, *where* $\Delta^{\mathsf{max}}_{\mathsf{coop}} \stackrel{\mathsf{def}}{=} \delta^{\mathsf{max}}_{\mathsf{defer}} + \Delta_r + \Delta^* + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$, $\delta^{\mathsf{max}}_{\mathsf{defer}} \stackrel{\mathsf{def}}{=} \hat{\delta}^{\mathsf{max}}_{\mathsf{coop}} - \Delta_2$, *and* $\hat{\delta}^{\mathsf{max}}_{\mathsf{coop}} \stackrel{\mathsf{def}}{=} \Delta_r + \Delta^* + \Delta_1$;

(c3) $v_{\mathsf{rm}}\Delta_r \geqslant v_{\mathsf{lim}}\Delta^*$;

(c4) $\Delta_{\mathsf{nonzeno}} < \Delta_r + \Delta^* + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$.

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

$$\Delta^{\mathsf{max}}_{\mathsf{reset}} \stackrel{\mathsf{def}}{=} \Delta^{\mathsf{max}}_{\mathsf{coop}} + \Delta_{\mathsf{nonzeno}} + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}), \tag{3.9}$$

*then $\exists t_2 \in (t_1, t_1 + \Delta^{\mathsf{max}}_{\mathsf{reset}}]$ s.t. either (**Stable State 1**) at t_2, h_1, h_2, ..., 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, ..., 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 claims.

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 Fig. 3.5, the state must change back to "Init" at some finite t_2 (where $t_1 < t_2 \leqslant t_1 + \Delta^{\mathsf{max}}_{\mathsf{coop}}$, see (**c2**) for the definition of $\Delta^{\mathsf{max}}_{\mathsf{coop}}$). That is, $\forall t \in (t_1, t_2]$, state = "Coop";[1] and at t_2^+, state = "Init". We call $(t_1, t_2]$ a "coop-duration". Note as per Fig. 3.3 and 3.5, it is easy to see that $\Delta^{\mathsf{max}}_{\mathsf{coop}}$ 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 $t_1 \in [t_5 - \Delta^{\mathsf{max}}_{\mathsf{coop}}, t_5]$ and $t_3 \in [t_5 - \Delta^{\mathsf{max}}_{\mathsf{coop}}, t_5]$, therefore $|t_1 - t_3| \leqslant \Delta^{\mathsf{max}}_{\mathsf{coop}}$. This means BS sends two different "SlowDown" packets within $\Delta^{\mathsf{max}}_{\mathsf{coop}}$. This contradicts (**c2**), where $\Delta^{\mathsf{min}}_{\mathsf{BS}} > \Delta^{\mathsf{max}}_{\mathsf{coop}}$. □

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

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

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, ..., h_n) are in "Init" mode at t.* ■

Proof: According to Fig. 3.5, 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 A.2 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 A.3 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 $\vec{p}_{\mathsf{merge}}$ 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 A.4 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 $t_1 + \Delta_{\mathsf{BS}}^{\mathsf{min}}$.

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, $t_3 \overset{\mathsf{def}}{=} t_1 + \Delta_r + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$ 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 $t_4 \overset{\mathsf{def}}{=} t_1 + \Delta_{\mathsf{nonzeno}}$ is a valid time instance.

Case 2: "Event2" happens at t_1. Then by $t_1 + \mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$, 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 $t_5 \stackrel{\mathsf{def}}{=} t_1 + \delta_{\mathsf{defer}} + \Delta_r + \Delta^* + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$.

Meanwhile, as per (**c2**), $\exists \varepsilon \in (0, \Delta^{\mathsf{min}}_{\mathsf{BS}} - \Delta^{\mathsf{max}}_{\mathsf{coop}} - \Delta_{\mathsf{nonzeno}})$. Let $t_6 \stackrel{\mathsf{def}}{=} t_5 + \varepsilon$, and $t_7 \stackrel{\mathsf{def}}{=} t_6 + \Delta_{\mathsf{nonzeno}}$. Then we have $t_1 + \mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\} < t_5 < t_6 < t_7 < t_1 + \Delta^{\mathsf{min}}_{\mathsf{BS}}$ (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 $t_1 + \Delta^{\mathsf{min}}_{\mathsf{BS}}$. Hence due to Lemma 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 $t_1 + \mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$ and remains there till $t_1 + \mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\} + \Delta^{\mathsf{min}}_{\mathsf{BS}}$. 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 $[t_1, t_1 + \Delta^{\mathsf{min}}_{\mathsf{BS}}]$.

Let $t_8 \stackrel{\mathsf{def}}{=} t_1 + \mathsf{max}\{\Delta_{\mathsf{nonzeno}}, \delta_{\mathsf{defer}}\}$, $t_9 \stackrel{\mathsf{def}}{=} t_8 + \varepsilon$, $t_{10} \stackrel{\mathsf{def}}{=} t_9 + \Delta_{\mathsf{nonzeno}}$, where ε is the same ε chosen for **Case 2.1**. Then (**c2**) and (**c4**) imply $0 < t_8 < t_9 < t_{10} < t_1 + \Delta^{\mathsf{min}}_{\mathsf{BS}}$. 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 $t_{11} \overset{\mathsf{def}}{=} t_1 + \varepsilon$, $t_{12} \overset{\mathsf{def}}{=} t_{11} + \Delta_{\mathsf{nonzeno}}$, 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. ■

3.5 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 Fig. 3.3) which highway CAV is currently closest to $\vec{p}_{\mathsf{merge}}$ on segment $[\vec{p}_{\mathsf{merge}} - v_{\mathsf{lim}}(\Delta_r + \Delta^* + \Delta_1), \vec{p}_{\mathsf{merge}}]$, and (if it exists)

whether its current distance to $\vec{p}_{\mathsf{merge}}$ is no less than $v_{\mathsf{lim}}(\Delta_r + \Delta^* + \Delta_1)$, or no greater than $v_{\mathsf{lim}}\Delta_2$, 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 Fig. 3.5), and are only used between two consecutive highway CAVs (h_i and h_{i+1}, where $i = 1$, 2, ..., $n-1$) 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 $h_\imath$ ($\imath \in \{1, 2, \ldots, n\}$) that is in a coop-duration. This implies $h_\imath$ 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. ■

3.6 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. [6], and the *consensus-based protocol* from Wang et al. [102]. We choose these two protocols because their focus problem contexts are the most similar to ours.

Specifically, Aoki et al. [6] focus on the design of a safe highway metered-ramp merging protocol, with collision avoidance guarantee under arbitrary wireless packet losses. How to adapt their protocol to guarantee *CTH safety* under arbitrary wireless packet losses is still an open problem. Fortunately, Aoki et al. [6] mentioned a "baseline priority-based protocol" for comparisons purposes in their paper'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 3.4.2 (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 Fig. 3.3 as follows.

1. Expand Event3's guard to cover all cases where $\hat{\delta}_{\mathsf{coop}} < \Delta_r + \Delta^* + \Delta_1$;

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.

Wang et al. [102]'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. [6]'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.

3.6.1 Simulation Configuration

We follow the recommendations by the seminal textbook of [78] to configure our simulator. Specifically, CTH safety desired time headway $\Delta^* = 3$s; $\Delta^{\mathsf{min}}_{\mathsf{BS}} = 39.61$s; $\Delta_{\mathsf{nonzeno}} = 0.1$s; $D_r = 300$m; $v_{\mathsf{lim}} = 33.333$m/s; $v_{\mathsf{rm}} = 25$m/s; acceleration and deceleration strategy are set as per [78], which decides $\delta_{\mathsf{a}}(0, v_{\mathsf{rm}}) = 13.01$s, $d_{\mathsf{a}}(0, v_{\mathsf{rm}}) = 200.6840$m, $\delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}) = 12.20$s, $d_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}) = 362.3613$m, $\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) = 3.08$s, and $d_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) = 90.9735$m. The above configuration further decides other parameters, specifically, Δ_r (see Eq. (3.2)), Δ_1 (see Eq. (3.4)), Δ_2 (see Eq. (3.5)), D_1 (see Eq. (3.8))), and $\Delta^{\mathsf{max}}_{\mathsf{reset}}$ (see Eq.(3.9)). Particularly, $\Delta^{\mathsf{max}}_{\mathsf{reset}} = 51.2$s, which is used in Section 3.6.3 and Tab. 3.2.

Note the above configurations comply with the constraints demanded by Theorem 1, as well as the recommendations of the consensus-based protocol [102].

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 [−50000m, 0m], where the location at 0m is $\vec{p}_{\mathsf{merge}}$. 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 [−50000m, 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 locations 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 3.6.3) of a highway and metered-ramp merging scenario.

3.6.2 Safety

Theorem 1 **Claim 1** is on the CTH-Δ^* safety guarantee. To validate this claim, Tab. 3.1 shows the statistics of sampled time headways (relative to the respective immediate predecessor vehicles, see Definition 3) of all vehicles in all simulation trials (for each vehicle simulated, its time headway is sampled every 0.4s). According to Tab. 3.1, 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 3.6.1) 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 lightgray in Tab. 3.1.

3.6.3 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.

[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.

According to Tab. 3.2, for the proposed protocol, all reset time costs are within the theoretical bound of $\Delta_{\mathsf{reset}}^{\mathsf{max}} = 51.2$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.

Also note that normally each simulation trial lasts 10 minutes (in the simulated world). 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.

3.6.4 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 (10 minutes) 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 time. 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.

Tab. 3.3 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 paris.

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. $(n = 240, P = 0.9)$).

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.

3.7 Chapter Summary

In this chapter, 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.

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

Table 3.1: 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.

Table 3.2: 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 Tab. 3.1 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 $\Delta_{\mathsf{reset}}^{\mathsf{max}} = 51.2$s.

Table 3.3: 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 Tab. 3.1 for definitions of n and P; n.a.: not applicable.

Chapter 4

A CAV Lane Change Protocol with CTH Safety Guarantee for Cooperative Driving on Dedicated Highways

In this chapter, we focus on the dedicated lane CAV cooperative driving protocol design for highway lane change. This protocol shall guarantee the *Constant Time Headway* (CTH) safety under arbitrary wireless packet losses. Correspondingly, Section 4.1 presents the demand; Section 4.2 discusses related work; Section 4.3 formulates the problem; Section 4.4 proposes our protocol and formally prove its safety and liveness guarantees; Section 4.5 makes some important observations to relax our assumptions; and Section 4.6 evaluates our proposed protocol.

4.1 Demand

With the fast growth on the *Connected and Automated Vehicles* (CAVs) technologies, *cooperative driving* of CAVs on *dedicated highways* has become a (if not the) most promising context to fully realize the vision of smart vehicle autopiloting [24][25][52][75][35][1][43][96][3][73][56][95][93]. Even in this context, the realization of the vision is still non-trivial, and must be built upon individual cooperation protocols for specific driving scenarios. In this chapter, we focus on one of such scenarios: lane change.

Lane change is one of the most common driving scenarios in road traffic, and it has naturally become a hot topic for CAV driving [111][51][55][62][84][99][61][105]. However, how to guarantee the CAV lane change safety is challenged by the inherently unreliable wireless communications. Particularly, wireless packets can be arbitrarily lost due to various reasons, such as large scale path-loss, multipath, jamming, handover, contention etc. Solutions to address this challenge heavily depend on the *targeted safety rule* and the *chosen wireless communications paradigm*.

For the targeted safety rule, in this chapter, we focus on the widely adopted *Constant Time Headway* (CTH) safety rule [65][92][18][110][26][11][85][27]. This rule requires any two consecutive vehicles on a same lane (referred to respectively as the "*leader*" and the "*follower*") maintain a spatial distance proportional to the follower's current speed.

For the wireless communications paradigm, there are two major alternatives: *Vehicle to Vehicle* (V2V) or *Vehicle to Infrastructure* (V2I). In this chapter, we focus on the V2V paradigm, due to its lower demand on infrastructure investment.

In summary, this chapter shall propose a V2V CAV lane change protocol for cooperative driving on dedicated highways. This protocol shall guarantee the CTH safety under arbitrary wireless packet losses: a challenge still lacks attention nowadays. In the large volume of literature on lane change autopiloting [111][51][55][62][84][99][61][105],

works focusing on the challenge of wireless packet losses are relatively few.

Specifically, in Tsugawa et al.'s paper on the Demo 2000 cooperative driving of automated vehicles [90], wireless packet losses are reported. However, how to deal with the wireless packet losses is not the focus of the chapter, hence is not elaborated.

Sakr et al. [82] propose to use open-loop Kalman filter to predict the position of remote vehicles when wireless packets are lost. But the focus is to detect intention of lane change, instead of guaranteeing CTH safety.

Wang et al. [97] present two lane change protocols (PerLC and ConLC) for V2V CAVs. The main idea of the protocols is to use simple acknowledgements (ACKs) to fight against arbitrary wireless packet losses. If an ACK is not received within a predefined period, the wireless packet will be retransmitted. The mechanical lane change routine cannot start unless all the needed ACKs are received. Although arbitrary wireless packet losses are considered in [97], how to adapt the protocols to guarantee the CTH safety is still an open problem.

Different from the above works, this paper focuses on guaranteeing the CTH safety under arbitrary wireless packet losses. We shall exploit the timeout (aka leasing [87][34]) design philosophy for distributed systems to achieve our goal. The basic idea is to properly configure a set of timeout deadlines on each participant at the start of each collaboration. During the collaboration, if the needed wireless packets cannot arrive before the timeout deadlines, the participants will independently reset themselves, hence implicitly reset the whole system. Specifically, we made the following contributions.

1. We propose a timeout based lane change protocol for V2V CAVs on dedicated highways.

2. We formally prove the CTH safety guarantee and liveness of our proposed protocol, even under arbitrary wireless packet losses.

3. We carry out simulations to verify our proposed protocol. The results show that our protocol can always fulfill the CTH safety and liveness despite of arbitrary wireless packet losses. The results also show that compared to other alternatives, our protocol can achieve significantly higher lane change success rates under adverse conditions.

4.2 Related Work

Despite of the works by Tsugawa et al. [90], Sakr et al. [82], and Wang et al. [97], which are elaborated in Section 4.1, there are many other works on CAV lane change. These works can be divided into three categories [12].

The first category focuses on trajectory planning. Works in this category can be further divided into non-cooperative [22][19][23] and cooperative lane change. Non-cooperative lane change is not the focus of this chapter. In cooperative lane change, involved vehicles exchange information wirelessly to better plan the lane change trajectories. Xu et al. [105] propose a lane change dynamic model to emulate different lane change strategies and predict lane-change trajectories for collision prediction. Nilsson et al. [68] present a low-complexity lane change maneuver algorithm which determines whether a lane change maneuver is desirable, and if so, selects an appropriate inter-vehicle traffic gap and time instance to perform the maneuver, and calculates the corresponding longitudinal and lateral control trajectory. Liu et al. [57] build a general trajectory planning method not only for the parking problem but also for other vehicle motion planning problems (including lane change) for automated vehicles. Li et al. [55] develop a dynamic cooperative planning model for automated lane change, where vehicles collaboratively accelerate and/or decelerate to create proper gaps for lane changes. Ding et al. [28] propose an integrated lane-change trajectory planning method for CAVs. The planned trajectories holistically consider safety, time-cost, distance cost, and comfort. Recently, machine learning based solutions also attract

much attention. Imitation Learning [13] has shown some promising results. Ben et al. [10] propose a trajectory planning algorithm based on Hierarchical Reinforcement Learning (HRL). This solution allows an ego-car holistically consider high-level goals and low-level planner choices.

The second category focuses on controller design. Works in this category can be further divided into two approaches [16]. The first approach designs controllers to track given planned trajectories (aka reference trajectories, typically derived from solutions of the first category). Petrov et al. [72] present a two-layer nonlinear adaptive steering controller for automated lane change maneuver. The controller aims to track desired cycloidal trajectories planned in real-time. Yang et al. [107] propose a comparative study of *Model Predictive Control* (MPC) and robust $\mathcal{H}_\infty$ state feedback control for trajectory tracking. Besides, a double lane change test with low road adhesion is designed to find the maximum feasible (in the sense of stability guarantee) velocity for both controllers. Chen et al. [21] propose a linear time-varying MPC controller to maintain tracking performance and improve stability in high-speed low-road-tire-friction environments. The solution is further enhanced by a steering angle compensation controller, which runs receding horizon corrector algorithms. For the unified lateral guidance algorithm, the yaw rate generator provides the desired yaw rate, then the yaw rate controller track the desired yaw rate to achieve lane change [17][9]. Hatipoglu et al. [37] generate the steering angle commands using a reference yaw rate signal and then design a sliding mode yaw rate controller to implement the lane change. The second approach designs controllers that conduct lane change without trajectory planning. Wang et al. [99] propose a Reinforcement Learning (RL) based solution, which learns lane change maneuvers directly. The solution can change lane intelligently under diverse and even unforeseen scenarios.

The third category combines trajectory planning and controller design. Luo et al. [61] propose a dynamic automated lane change maneuver based on V2V communication. The trajectory planning method converts the planning problem into a constrained

optimization problem using the lane change time and distance, and a trajectory-tracking controller based on sliding mode control calculates the control inputs to make the host vehicle travel along the reference trajectory. However, this work only focuses on scenarios with two lanes and one lane-change vehicle. Luo et al. [62] focus on multi-vehicle V2V communication based cooperative automated lane-change. The focus scenarios have eight vehicles on three lanes. In these scenarios, same-direction and intersectant-direction cooperative lane changes are defined. Trajectory planning is treated as an optimization problem with the objective of maximizing safety, comfort, and lane-change efficiency, under the constraints of vehicle dynamics and the cooperative safety spacing model. An MPC trajectory tracking method is designed to minimize tracking errors and control increments. Rafat et al. [77] develop an adaptive lane change algorithm which provides all possible safe trajectories for any moment of maneuver. In this way, it is able to make a new decision and plan safe trajectories according to the new conditions of surrounding vehicles during the maneuver. Also, it guarantees collision avoidance at all-time via simultaneous longitudinal and lateral vehicle control. Vallon et al. [91] propose to use Support Vector Machine to make the lane change decision. After the lane change demand is generated, the maneuver is executed using a MPC. Krasowski et al. [51] present a framework for safeguarding a vehicle using a safety layer to verify whether the proposed actions are safe and provide a provably safe fail-safe controller. Shi et al. [84] propose a hierarchical reinforcement learning based architecture for decision making and control of lane change situations. Specifically, the Deep Q-network (DQN) is applied to decide when to conduct the maneuver, adhering to safety considerations. Subsequently, a Deep Q learning framework with quadratic approximator is designed for deciding how to complete the maneuver in longitudinal direction.

However, none of the above works, as mentioned in Section 4.1, focuses on CTH safety guarantee under arbitrary wireless packet losses.

4.3 Problem Formulation

In this section, we specify the lane change scenario, related assumptions, CTH safety rule, and the research problem.

4.3.1 Lane Change Scenario

Fig. 4.1 illustrates this chapter's focus lane change scenario.

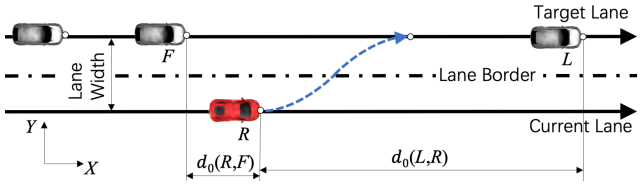


Figure 4.1: Lane Change Scenario

This scenario includes two neighboring lanes of interest along a dedicated highway. Without loss of generality, name one of the lanes as the "*current lane*," and the other lane as the "*target lane*" for the lane change. These two lanes are abstracted as two parallel axes in Fig. 4.1, whose positive direction defines the X-axis direction of our global fixed-to-ground X-Y coordinates. In this coordinate system, for simplicity and for the time being, let us abstract every CAV as a point mathematically (see the $\circ$ dots in Fig. 4.1), and let $\vec{p}(x,t)$ denote the location of CAV x at *world time instance* (simplified as "*time instance*" in the following unless otherwise denoted) t. When not conducting the mechanical lane change routine, a CAV point should drive along either the "current lane" axis or the "target lane" axis. Considerations on the vehicle body length are discussed in Section 4.3.3 (the paragraph right after Def. 3) and Section 4.5.2.

Suppose our lane change scenario starts from time instance t_0, when all CAVs on the dedicated highway are driving at the stable default speed v_{lim}, where v_{lim} is a preconfig-

ured constant. Note adopting such a v_{lim} is a popular practice by many collaborative CAV driving schemes [83][98][53][15], particularly in the large volume of literature on *Cooperative Adaptive Cruise Control* (CACC) [8][26]; this practice prevails not only for its simplicity, but also for its safety and energy efficiency [53][15][8][26].

Meanwhile, starting from t_0, a CAV R on the current lane intends to change to the target lane. We call R the "*requesting CAV*." On the target lane, at t_0, we assume there are n ($n \in \mathbb{N}^0$) CAVs, consecutively denoted as V_i ($i = 1, 2, \ldots, n$, with V_1 as the front most CAV on the target lane). We call the V_is the "*target lane CAVs*." From time to time, R needs to recognize the closest (along the X-axis) CAV before R, denote that target lane CAV as L ($L \stackrel{\mathsf{def}}{=} \varnothing$ if it does not exist); and recognize the closest (along the X-axis) CAV not before (i.e. after or at the same X coordinate as) R, denote that target lane CAV as F ($F \stackrel{\mathsf{def}}{=} \varnothing$ if it does not exist). We call L the "*leader CAV*," and F the "*follower CAV*." We denote the distances from R to L and F along the X-axis (i.e. distance projections on the X-axis) respectively as $d_0(L, R)$ and $d_0(R, F)$ ($d_0(L, R) \stackrel{\mathsf{def}}{=} +\infty$ if $L = \varnothing$, and $d_0(R, F) \stackrel{\mathsf{def}}{=} +\infty$ if $F = \varnothing$).

4.3.2 Assumptions on CAV Driving Dynamics

Vehicular driving dynamics modeling is nontrivial (interested readers can refer to the classic textbook of [78]). Fortunately, we can make the following reasonable and generic enough assumptions for our CAVs.

Mechanical Lane Change Routine

Suppose at time t^0_{lc}, a CAV starts the mechanical lane change routine with an initial speed of $v^0_{\mathsf{lc}} \in [v^{\mathsf{low}}_{\mathsf{lc}}, v^{\mathsf{high}}_{\mathsf{lc}}]$ (from the initial orientation of pointing toward the positive direction of the X-axis), where $v^{\mathsf{high}}_{\mathsf{lc}} > v^{\mathsf{low}}_{\mathsf{lc}} > 0$ are respectively the upper bound and lower bound of v^0_{lc}.

Based on the dynamic model of CAV in the classic textbook of [78], suppose currently the mechanical lane change routine has been going on for τ seconds ($\tau \geq 0$) and has not yet finished, then the CAV's current velocity (i.e. speed and direction) is determined by a function of v_{lc}^0 and τ. Denote this function as $\mathsf{lc}(v_{\mathsf{lc}}^0, \tau)$. This in turn implies the total duration and projected X-axis distance experienced by a completed mechanical lane change routine are respectively determined by functions of v_{lc}^0, which can be respectively denoted as $\delta_{\mathsf{lc}}(v_{\mathsf{lc}}^0)$ and $d_{\mathsf{lc}}^{\mathsf{X}}(v_{\mathsf{lc}}^0)$. Note the total projected Y-axis distance experienced by a completed mechanical lane change routine is always the lane width.

Furthermore, according to [78], we can make the following reasonable and generic enough assumption:

Assumption 6 *We assume throughout a mechanical lane change routine of a CAV, the CAV's speed (i.e. velocity magnitude) is constant, and the total duration $\delta_{\mathsf{lc}}(v_{\mathsf{lc}}^0)$ needed to complete the mechanical lane change routine is a strictly monotonically decreasing function of the initial speed v_{lc}^0. That is, the faster the initial speed, the faster the mechanical lane change routine completes. As $v_{\mathsf{lc}}^0 \in [v_{\mathsf{lc}}^{\mathsf{low}}, v_{\mathsf{lc}}^{\mathsf{high}}]$, we have $\delta_{\mathsf{lc}}(v_{\mathsf{lc}}^{\mathsf{high}}) < \delta_{\mathsf{lc}}(v_{\mathsf{lc}}^{\mathsf{low}})$ and $\delta_{\mathsf{lc}}(v_{\mathsf{lc}}^0) \in [\delta_{\mathsf{lc}}(v_{\mathsf{lc}}^{\mathsf{high}}), \delta_{\mathsf{lc}}(v_{\mathsf{lc}}^{\mathsf{low}})]$.* ■

The above assumption also implies the projected X-axis speed of the mechanical lane change routine first decreases from v_{lc}^0, and then increases back to v_{lc}^0, hence

$$v_{\mathsf{lc}}^0 \delta_{\mathsf{lc}}(v_{\mathsf{lc}}^0) > d_{\mathsf{lc}}^{\mathsf{X}}(v_{\mathsf{lc}}^0). \tag{4.1}$$

CAV Acceleration Routine

Besides the accelerations/decelerations conducted during mechanical lane change routines, a CAV may also conduct straight line (i.e. along the X-axis) accelerations/decelerations. *To simplify narration, in the following, unless explicitly denoted,*

the term "acceleration" and "deceleration" shall only refer to such straight line accelerations/decelerations.

We assume in dedicated highway coordinated driving of CAVs, all CAVs' acceleration strategies can be fixed, hence can be called "*acceleration routines.*" Specifically, given the initial speed $v_{\mathsf{a}}^{\mathsf{low}}$ and the target speed $v_{\mathsf{a}}^{\mathsf{high}}$, suppose currently the acceleration routine has been going on for τ seconds ($\tau \geqslant 0$) and has not yet finished, then the CAV's current acceleration value is fixed, and is a function of $v_{\mathsf{a}}^{\mathsf{low}}$, $v_{\mathsf{a}}^{\mathsf{high}}$, and τ. Denote this function as $\mathsf{acc}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}, \tau)$. This function in turn implies that the current speed of the CAV is a function of $v_{\mathsf{a}}^{\mathsf{low}}$, $v_{\mathsf{a}}^{\mathsf{high}}$, and τ, which can be denoted as $v_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}, \tau)$. This in turn implies that the total duration and distance needed to accelerate from $v_{\mathsf{a}}^{\mathsf{low}}$ to $v_{\mathsf{a}}^{\mathsf{high}}$ is a function of $v_{\mathsf{a}}^{\mathsf{low}}$ and $v_{\mathsf{a}}^{\mathsf{high}}$. We can denote this duration and this distance to be respectively $\delta_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}})$ and $d_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}})$ (for more sophisticated details on vehicle acceleration, interested readers can refer to [78]).

Furthermore, we have the following assumption:

Assumption 7 *Given* $0 \leqslant v_{\mathsf{a}}^{\mathsf{low}} < v_{\mathsf{a}}^{\mathsf{high}}$*, the acceleration routine as per* $\mathsf{acc}(\, v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}, \tau)$ *is always nonzeno (i.e.* $\delta_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}) > 0$*) and the speed will strictly monotonically increase from* $v_{\mathsf{a}}^{\mathsf{low}}$ *to* $v_{\mathsf{a}}^{\mathsf{high}}$*.* ■

CAV Deceleration Routine

Similar to acceleration, we also assume the CAV deceleration strategies are fixed, hence can be called "*deceleration routines.*" Specifically, given the initial speed $v_{\mathsf{d}}^{\mathsf{high}}$ and the target speed $v_{\mathsf{d}}^{\mathsf{low}}$, suppose currently the deceleration routine has been going on for τ seconds ($\tau \geqslant 0$) and has not yet finished, then the CAV's current acceleration value is fixed, and is a function of $v_{\mathsf{d}}^{\mathsf{high}}$, $v_{\mathsf{d}}^{\mathsf{low}}$, and τ. Denote this function as $\mathsf{dec}(v_{\mathsf{d}}^{\mathsf{high}}$, $v_{\mathsf{d}}^{\mathsf{low}}, \tau)$. This function in turn implies that the current speed of the CAV is a function of $v_{\mathsf{d}}^{\mathsf{high}}$, $v_{\mathsf{d}}^{\mathsf{low}}$, and τ, which can be denoted as $v_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}}, \tau)$. This in turn implies that the total duration and distance needed to decelerate from $v_{\mathsf{d}}^{\mathsf{high}}$ to $v_{\mathsf{d}}^{\mathsf{low}}$ is a function

of $v_{\mathsf{d}}^{\mathsf{high}}$ and $v_{\mathsf{d}}^{\mathsf{low}}$. We can denote this duration and this distance to be respectively $\delta_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}})$ and $d_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}})$.

Furthermore, we have the following assumption:

Assumption 8 *Given $v_{\mathsf{d}}^{\mathsf{high}} > v_{\mathsf{d}}^{\mathsf{low}} \geqslant 0$, the deceleration routine as per $\mathsf{dec}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}}, \tau)$ is always nonzeno (i.e. $\delta_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}}) > 0$) and the speed will strictly monotonically decrease from $v_{\mathsf{d}}^{\mathsf{high}}$ to $v_{\mathsf{d}}^{\mathsf{low}}$.* ■

4.3.3 Other Assumptions, CTH Safety Rule, and the Research Problem

Besides the mechanical driving capabilities, CAVs should also be able to sense their surrounding environments. Particularly, we assume the following.

Assumption 9 *Each CAV is equipped with redundant ranging sensors (e.g., laser, radar, supersonic, computer vision, V2V communications, and human driver as the last resort), so that at any intended time instance, the requesting CAV R can always recognize the leader CAV L (or that $L = \varnothing$, when it does not exist) and the follower CAV F (or that $F = \varnothing$, when it does not exist) on the target lane. Furthermore, R can sense the distance $d_0(L, R)$ and $d_0(R, F)$ to L and F respectively at the intended time instance ($d_0(L, R) \stackrel{\mathsf{def}}{=} +\infty$ when $L = \varnothing$, and $d_0(R, F) \stackrel{\mathsf{def}}{=} +\infty$ when $F = \varnothing$). Meanwhile, for any two consecutive CAVs x and y along a same lane (suppose x precedes y), y can instantly detect x's speed (e.g. based on y's own speed and the relative velocity to x detected by y's ranging sensors), and vice versa for x. Particularly, due to the redundancy, even if the V2V communications fail (so that x and y cannot exchange wireless packets), the ranging sensing can still function correctly.* ■

For the above lane change scenario, we aim to guarantee the *Constant Time Headway* (CTH) safety rule [26] as specified in the following.

Definition 3 (CTH Safety Rule) *Suppose two vehicles (in math point abstraction, see Fig. 4.1) x and y are driving in the same direction along a same lane (in math axis abstraction, see Fig. 4.1). Suppose x precedes y at time instance t. Denote the distance between x and y at t as $d(t)$, and y's speed at t as $v_y(t)$. Then we call $\delta(t) \stackrel{\mathsf{def}}{=} d(t)/v_y(t)$ as the* time headway *of y (relative to x) at t. If $\delta(t) \geqslant \Delta^*$, where $\Delta^* > 0$ is a given constant, 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.* ■

The importance of the CTH-Δ^* safety rule can be explained by the following fact. Suppose a lane has both a minimum speed limit v^{min} and a maximum speed limit v^{max}, we can set Δ^* to $\Delta^{**} + \frac{D_0}{v^{\mathsf{min}}}$, where Δ^{**} is the maximum duration needed to stop a CAV at any speed $v \in [v^{\mathsf{min}}, v^{\mathsf{max}}]$ using emergency braking (which should be monotonically decelerating, and can be different from the normal deceleration dec), and D_0 is the maximum vehicle body projection length on X-axis at any orientation (D_0 hence is no less than the vehicle body length). Under this setting, simple analysis can show that CTH-Δ^* safety guarantees y will never hit x, even if x abruptly stop at anytime on the lane.

We hence assume the following, which implies the safety of the target lane before the lane change.

Assumption 10 *$\forall i \in \{1, 2, \ldots, n-1\}$, (V_i, V_{i+1}) is CTH-Δ^* safe at t_0.* ■

This naturally leads to our research problem: whether the lane change requested by R (see Fig. 4.1) will disrupt the safety of the target lane? Furthermore, as coordinations between CAVs need communications of wireless packets, and wireless communications are well-known unreliable, can we guarantee the safety when wireless packets are arbitrarily lost? More rigorously,

Research Problem 1 (CTH Safety Guarantee) *How to coordinate the CAV lane change under arbitrary wireless packet losses, so that CTH-Δ^* safety always holds in the target lane?* ■

Related, we are also concerned about the liveness and performance of our proposed coordination protocol. Formally,

Research Problem 2 (Liveness Guarantee) *Under arbitrary wireless packet losses, can the coordination protocol reset itself within bounded time?* ■

Research Problem 3 (Performance) *Under arbitrary wireless packet losses, what are the success rate and time cost of lane change?* ■

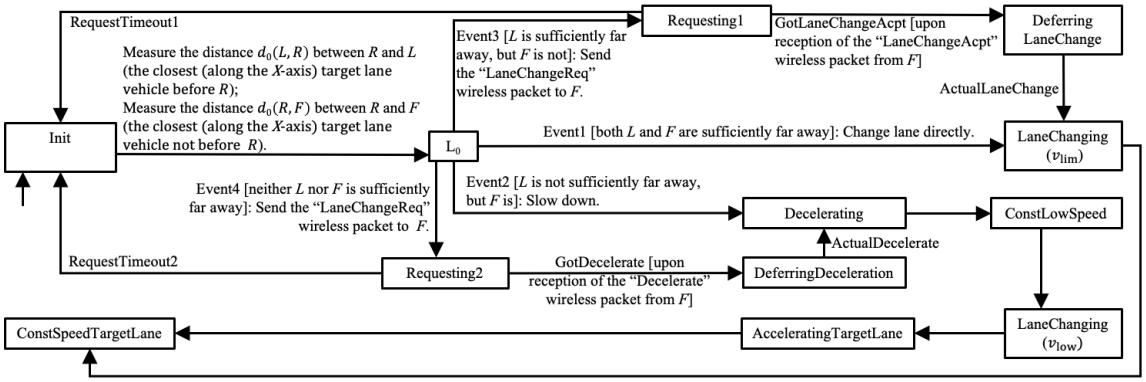
4.4 Solution

In this section, we propose our protocol to answer Research Problem 1 to 3.

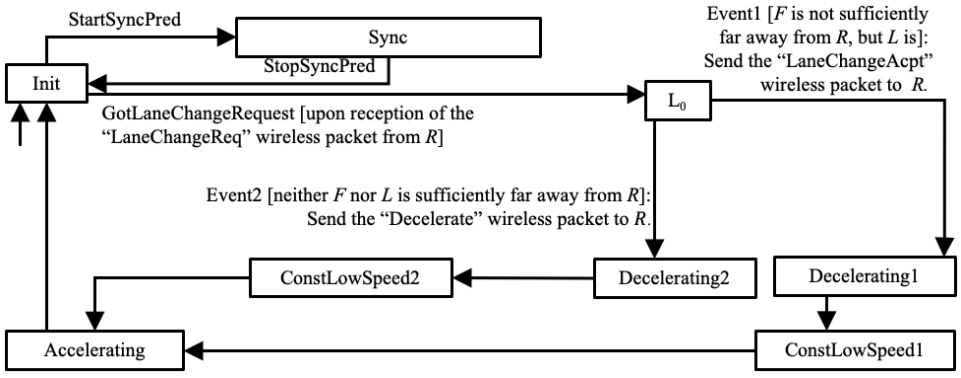
4.4.1 Heuristics

The heuristics of our proposed protocol is illustrated by the automata sketches of Fig. 4.2.

Initially, the target lane CAVs V_i ($i = 1, 2, \ldots, n$) and the requesting CAV R all reside in their respective "Init" mode. At t_0, R intends to change lane and triggers the first action: it identifies (with its ranging sensors) the closest (along the X-axis) target lane CAV before it (aka the *leader CAV*, denoted as L) and the closest (along the X-axis) target lane CAV not before (i.e. after or at the same X coordinate as) it (aka the *follower CAV*, denoted as F). If the leader (follower) CAV does not exist, we denote $L \stackrel{\mathsf{def}}{=} \varnothing$ ($F \stackrel{\mathsf{def}}{=} \varnothing$). Using the ranging sensors, R can measure its current



(a) CAV R



(b) CAVs V_i

Figure 4.2: 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.

distances to L and F along the X-axis: denote them respectively as $d_0(L, R)$ and $d_0(R, F)$ (see Fig. 4.1). Note $d_0(L, R) \stackrel{\mathsf{def}}{=} +\infty$ if $L = \varnothing$, and $d_0(R, F) \stackrel{\mathsf{def}}{=} +\infty$ if $F = \varnothing$. R then enters the intermediate mode of "L_0" to take further actions based on $d_0(L, R)$ and $d_0(R, F)$. Specifically,

1. If R is sufficiently far away from both L and F (see "Event1" in Fig. 4.2(a)), then it will directly start the mechanical lane change routine with the initial speed of v_{lim}.

2. If R is close to L but sufficiently far away from F (see "Event2" in Fig. 4.2(a)), then it will decelerate to v_{low} and maintain this speed to reach a safe distance from L. Then it will start the mechanical lane change routine with the initial speed of v_{low}. By the end of the routine, R will reach the target lane, and will then accelerate back to v_{lim}.

3. If R is sufficiently far away from L but close to F (see "Event3" in Fig. 4.2(a)), then it will first send a "LaneChangeReq" wireless packet to F. Upon receiving this packet, F will reply a "LaneChangeAcpt" wireless packet to R, and decelerate to v_{low} to reach a safe distance from R (see "Event1" in Fig. 4.2(b)). At R, upon reception of the "LaneChangeAcpt" wireless packet, R will wait for F to reach the safe distance, and then start the mechanical lane change routine with the initial speed of v_{lim}. By the end of the routine, R will reach the target lane, and F will accelerate back to v_{lim}. In addition, the target lane CAVs V_i after F will maintain the convoy by synchronizing their speeds with their respective predecessors if needed.

4. Otherwise (R is close to both L and F, see "Event4" in Fig. 4.2(a)), R will first send a "LaneChangeReq" wireless packet to F. Upon reception of this packet, F will reply a "Decelerate" wireless packet to R, and decelerate to v_{low} to reach a safe distance from R (see "Event2" in Fig. 4.2(b)). At R, upon reception of the "Decelerate" wireless packet, R will wait for F to reach the safe distance,

and then also decelerate to v_{low} to reach a safe distance from L. Then R will start the mechanical lane change routine with the initial speed of v_{low}. By the end of the routine, R will reach the target lane, and will then accelerate back to v_{lim}. Then F will also accelerate back to v_{lim}. In addition, the target lane CAVs V_i after F will maintain the convoy by synchronizing their speeds with their respective predecessors if needed.

For the above cases, how "far" is "sufficiently far" and how to configure the parameters to achieve the CTH-Δ^* safety are non-trivial questions. We will clarify them in the detailed protocol design and analysis (see Section 4.4.2 and 4.4.3).

Another challenge is the possibility of arbitrary wireless packet losses. What if the "LaneChangeReq," "Decelerate," "LaneChangeAcpt," wireless packets are lost? Can the CTH-Δ^* safety still sustain? Can the CAVs still reset themselves instead of stuck in a mode forever?

To address these concerns, we propose to deploy the "*lease*" design pattern for distributed systems [87]. A "*lease*" is an agreement on timeout, contracted since the early stage of a distribute 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 Fig. 4.2, 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 Section 4.4.2 and 4.4.3. The efficiency is evaluated in Section 4.6.

4.4.2 Proposed Protocol

We propose our detailed protocol by expanding the automata sketches of Fig. 4.2 with the heuristics described in Section 4.4.1.

Protocol Symbols

The protocol specifications (and follow up analyses) may use the following symbols.

First, as our proposed protocol deals with two and only two given steady state speeds: v_{lim} and v_{low}, where

$$v_{\mathsf{lim}} > v_{\mathsf{low}} > 0, \tag{4.2}$$

all straight line acceleration and decelerations are between these two speeds. Hence we adopt the following simplification notations:

$$\begin{aligned} \delta_{\mathsf{a}}^{\ddagger} &\stackrel{\mathsf{def}}{=} \delta_{\mathsf{a}}(v_{\mathsf{low}}, v_{\mathsf{lim}}), \qquad & d_{\mathsf{a}}^{\ddagger} &\stackrel{\mathsf{def}}{=} d_{\mathsf{a}}(v_{\mathsf{low}}, v_{\mathsf{lim}}), \\ \delta_{\mathsf{d}}^{\ddagger} &\stackrel{\mathsf{def}}{=} \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{low}}), \qquad & d_{\mathsf{d}}^{\ddagger} &\stackrel{\mathsf{def}}{=} d_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{low}}). \end{aligned} \tag{4.3}$$

Other simplification notations include

$$\widetilde{v} \stackrel{\mathsf{def}}{=} v_{\mathsf{lim}} - v_{\mathsf{low}} > 0 \text{ (due to Ineq. (4.2))}, \tag{4.4}$$

$$\widetilde{d}_{\mathsf{a}}(v_{\mathsf{lim}}) \stackrel{\mathsf{def}}{=} v_{\mathsf{lim}}\delta_{\mathsf{a}}^{\ddagger} - d_{\mathsf{a}}^{\ddagger} > 0 \text{ (due to Assumption 7)}, \tag{4.5}$$

$$\widetilde{d}_{\mathsf{a}}(v_{\mathsf{low}}) \stackrel{\mathsf{def}}{=} d_{\mathsf{a}}^{\ddagger} - v_{\mathsf{low}}\delta_{\mathsf{a}}^{\ddagger} > 0 \text{ (due to Assumption 7)}, \tag{4.6}$$

$$\widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) \stackrel{\mathsf{def}}{=} v_{\mathsf{lim}}\delta_{\mathsf{d}}^{\ddagger} - d_{\mathsf{d}}^{\ddagger} > 0 \text{ (due to Assumption 8)}, \tag{4.7}$$

$$\widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}}) \stackrel{\mathsf{def}}{=} d_{\mathsf{d}}^{\ddagger} - v_{\mathsf{low}}\delta_{\mathsf{d}}^{\ddagger} > 0 \text{ (due to Assumption 8)}, \tag{4.8}$$

$$\begin{aligned} \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) &\stackrel{\mathsf{def}}{=} v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) - d_{\mathsf{lc}}^{\mathsf{X}}(v_{\mathsf{lim}}) \\ &> 0 \text{ (due to Ineq. (4.1))}, \end{aligned} \tag{4.9}$$

$$\begin{aligned} \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{low}}) &\stackrel{\mathsf{def}}{=} v_{\mathsf{low}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) - d_{\mathsf{lc}}^{\mathsf{X}}(v_{\mathsf{low}}) \\ &> 0 \text{ (due to Ineq. (4.1))}. \end{aligned} \tag{4.10}$$

Our proposed protocol adopts the following configurable constants:

1. $\Delta^* > 0$, the desired time headway of CTH safety, see Def. 3.

2. $\Delta_{\mathsf{nonzeno}} > 0$, maximum waiting time for wireless reply.

These configurable constants further imply the following constants used in the protocol.

$$D_1 \stackrel{\text{def}}{=} v_{\mathsf{lim}}\Delta^* - \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}), \tag{4.11}$$

$$D_2 \stackrel{\text{def}}{=} 2v_{\mathsf{lim}}\Delta^* + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{low}}) + \widetilde{v}(\delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \delta^{\ddagger}_{\mathsf{a}}), \tag{4.12}$$

$$D_3 \stackrel{\text{def}}{=} v_{\mathsf{lim}}\Delta^* + \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}), \tag{4.13}$$

$$D_{\mathsf{Sync}} \stackrel{\text{def}}{=} \mathsf{max}\{D^{\mathsf{Event1,min}}_{\mathsf{Sync}}, D^{\mathsf{Event2,min}}_{\mathsf{Sync}}\}, \tag{4.14}$$

where

$$D^{\mathsf{Event1,min}}_{\mathsf{Sync}} \stackrel{\text{def}}{=} v_{\mathsf{lim}}\Delta^* + D_3 + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}}) + \widetilde{v}(\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) + \delta^{\ddagger}_{\mathsf{a}}), \tag{4.15}$$

$$D^{\mathsf{Event2,min}}_{\mathsf{Sync}} \stackrel{\text{def}}{=} v_{\mathsf{lim}}\Delta^* + D_2 + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + \widetilde{v}\delta^{\ddagger}_{\mathsf{a}}, \tag{4.16}$$

$$\Delta^{\mathsf{Event1,max}}_{\mathsf{Coop}} \stackrel{\text{def}}{=} \delta^{\ddagger}_{\mathsf{d}} + \delta^{\ddagger}_{\mathsf{a}} + \frac{D_3 + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta_{\mathsf{lc}}(v_{\mathsf{lim}}), \tag{4.17}$$

$$\Delta^{\mathsf{Event2,max}}_{\mathsf{Coop}} \stackrel{\text{def}}{=} \delta^{\ddagger}_{\mathsf{d}} + \delta^{\ddagger}_{\mathsf{a}} + \frac{D_2}{\widetilde{v}}, \tag{4.18}$$

$$\Delta^{\mathsf{max}}_{\mathsf{Coop}} \stackrel{\text{def}}{=} \mathsf{max}\{\Delta^{\mathsf{Event2,max}}_{\mathsf{Coop}} + \delta^{\ddagger}_{\mathsf{d}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \Delta^*, \Delta^{\mathsf{Event1,max}}_{\mathsf{Coop}}\}, \tag{4.19}$$

$$\Delta_{\mathsf{reset}} \stackrel{\text{def}}{=} \Delta^{\mathsf{max}}_{\mathsf{Coop}} + \Delta_{\mathsf{nonzeno}}. \tag{4.20}$$

A comprehensive symbol list is also provided below for reader's convenience (listed alphabetically: Greek before Latin, and upper case before lower case).

Δ^* : the desired time headway, see Def. 3.

$\Delta_{\mathsf{Coop}}^{\mathsf{Event1,max}}$, $\Delta_{\mathsf{Coop}}^{\mathsf{Event2,max}}$, $\Delta_{\mathsf{Coop}}^{\mathsf{max}}$: respectively see Eq. (4.17), (4.18), (4.19).

$\Delta_{\mathsf{nonzeno}}$: see Section 4.4.2 2).

Δ_{reset} : see Eq. (4.20).

$\delta_{\mathsf{a}}^{\ddagger}$, $\delta_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}})$: respectively see Eq. (4.3) and Section 4.3.2.

$\delta_{\mathsf{comm}}^{\mathsf{max}}$: see Theorem 2 (**c6**).

$\delta_{\mathsf{d}}^{\ddagger}$, $\delta_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}})$: respectively see Eq. (4.3) and Section 4.3.2.

$\delta_{\mathsf{defer}}^{\mathsf{Rdec}}$: see Section 4.4.2 Requesting CAV R Protocol Behaviors 13) Case1; Section 4.4.2 Target Lane CAV V_i Protocol Behaviors 4) Case2.

$\delta_{\mathsf{defer}}^{\mathsf{lc}}$: see Section 4.4.2 Requesting CAV R Protocol Behaviors 11) Case1; Section 4.4.2 Target Lane CAV V_i Protocol Behaviors 4) Case1.

$\delta_{\mathsf{lc}}(v_{\mathsf{lc}}^0)$: see Section 4.3.2.

$\delta_{\mathsf{low}}^{\mathsf{LR}}$: see Section 4.4.2 Requesting CAV R Protocol Behaviors "4) Case2," "13) Case1;" Section 4.4.2 Target Lane CAV V_i Protocol Behaviors 4) Case2.

$\delta_{\mathsf{low}}^{\mathsf{RF1}}$, $\delta_{\mathsf{low}}^{\mathsf{RF2}}$: respectively see Section 4.4.2 Target Lane CAV V_i Protocol Behaviors 4) Case1 and Case2.

τ : usually represents a local clock, please see the specific context for definition; particularly, in Section 4.4.2 Requesting CAV R Protocol Behaviors and Fig. 4.3, it is a local clock used by hybrid automaton A_R, and in Section 4.4.2 Target Lane CAV V_i Protocol Behaviors and Fig. 4.4, it is a local clock used by hybrid automaton A_i.

A_R : hybrid automaton for the requesting CAV R, see Section 4.4.2 Requesting CAV R Protocol Behaviors and Fig. 4.3.

A_i : hybrid automaton for the target lane CAV V_i, see Section 4.4.2 Target Lane CAV V_i Protocol Behaviors and Fig. 4.4.

D_1, D_2, D_3, D_{Sync}, $D_{\mathsf{Sync}}^{\mathsf{Event1,min}}$, $D_{\mathsf{Sync}}^{\mathsf{Event2,min}}$: respectively see Eq. (4.11), (4.12), (4.13), (4.14), (4.15), (4.16).

F : the follower CAV, see Section 4.3.1.

L : the leader CAV, see Section 4.3.1.

P : wireless packet loss rate, see Section 4.6.1.

R : the requesting CAV, see Section 4.3.1.

V_i : the ith target lane CAV, see Section 4.3.1.

$\mathsf{acc}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}}, \tau)$: see Section 4.3.2.

$d_0(L, R)$, $d_0(R, F)$: respectively the latest measured distance between L and R, and R and F, see Section 4.3.1.

$d_{\mathsf{a}}^{\ddagger}$, $d_{\mathsf{a}}(v_{\mathsf{a}}^{\mathsf{low}}, v_{\mathsf{a}}^{\mathsf{high}})$: respectively see Eq. (4.3) and Section 4.3.2.

$\widetilde{d}_{\mathsf{a}}(v_{\mathsf{lim}})$, $\widetilde{d}_{\mathsf{a}}(v_{\mathsf{low}})$: respectively see Eq. (4.5) and (4.6).

$d_{\mathsf{d}}^{\ddagger}$, $d_{\mathsf{d}}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}})$: respectively see Eq. (4.3) and Section 4.3.2.

$\widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}})$, $\widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})$: respectively see Eq. (4.7). and (4.8).

$\mathsf{dec}(v_{\mathsf{d}}^{\mathsf{high}}, v_{\mathsf{d}}^{\mathsf{low}}, \tau)$: see Section 4.3.2.

$\widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}})$, $\widetilde{d_{\mathsf{lc}}}(v_{\mathsf{low}})$: respectively see Eq. (4.9) and (4.10).

$d_{\mathsf{lc}}^{\mathsf{X}}(v_{\mathsf{lc}}^0)$: see Section 4.3.2.

$\mathsf{lc}(v_{\mathsf{lc}}^0, \tau)$: see Section 4.3.2.

n : total number of target lane CAVs, see Section 4.3.1.

$\vec{p}(x,t)$: location of CAV x at time instance t, see Section 4.3.1.

state : see Section 4.4.2 Target Lane CAV V_i Protocol Behaviors 2).

t : a time instance of the world.

t_0 : scenario start time instance, see Section 4.3.1.

t_1 : see Theorem 2 **Claim 2**.

t_2 : see Theorem 2 **Claim 2**.

$\widetilde{v}$: see Eq. (4.4).

v_{lim}, v_{low} : respectively the max and min steady state speeds along a highway lane, see Section 4.3.1 and Ineq. (4.2).

With the help of the above symbols, we can specify our proposed protocol in Section 4.4.2 Requesting CAV R Protocol Behaviors and 4.4.2 Target Lane CAV V_i Protocol Behaviors, respectively for the requesting CAV R protocol behaviors and the target lane CAV V_i ($i \in \{1, 2, \ldots, n\}$) protocol behaviors.

Requesting CAV R Protocol Behaviors

The requesting CAV R's protocol behaviors are defined by a hybrid automaton [5] A_R, which is illustrated by Fig. 4.3 and explained by the following text.

1. The hybrid automaton A_R has the following modes: "Init," "L_0," "Requesting1," "Requesting2," "DeferringLaneChange," "LaneChanging(v_{lim})," "DeferringDeceleration," "Decelerating," "ConstLowSpeed," "LaneChanging(v_{low})," "AcceleratingTargetLane," and "ConstSpeedTargetLane." At any time instance, R dwells in one of these modes. A_R also has a local clock for various timing purposes. *In this*

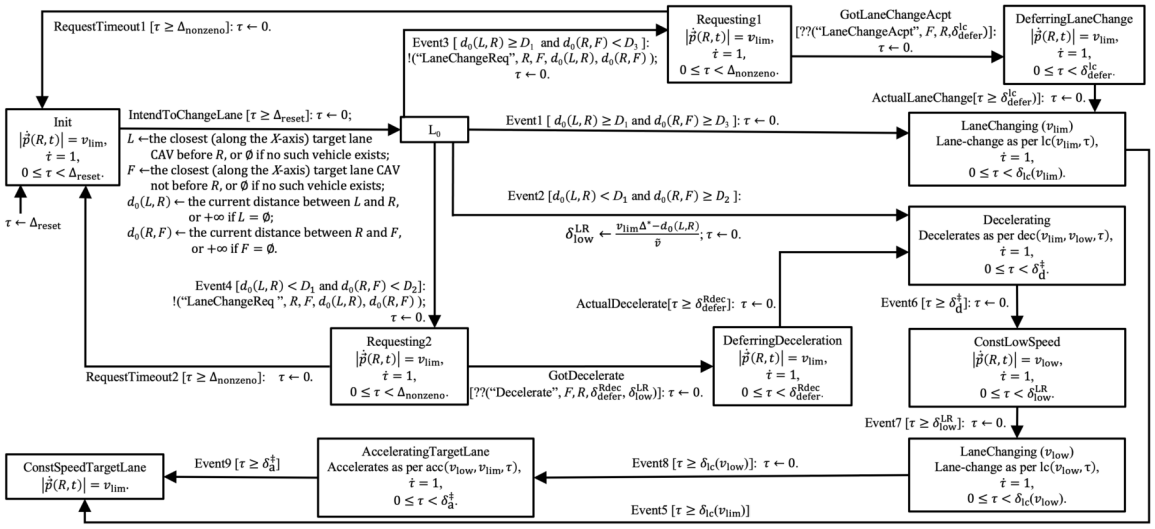


Figure 4.3: Hybrid automaton A_R (see Section 4.4.2 Requesting CAV R Protocol Behaviors) for the requesting CAV R. Each rectangle box represents a *hybrid automaton mode* (simplified as "*mode*" in the following). Inside a mode, the top line is the mode's name (which is local to the hybrid automaton, i.e. modes of same name of different hybrid automata are different modes), the rest describes the constraints (e.g. a dwelling duration constraint like $0 \leq \tau < \Delta_{\mathsf{reset}}$) and continuous domain dynamics (typically specified with differential equations, e.g. $\dot{\tau} = 1$) related to the mode. Note "L_0" is a dummy mode, whose maximum dwelling duration constraint is 0 second, 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. Other arrows represent discrete *events*. 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 (x is represented as 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 ":" are the instant actions to be taken when the event happens. Particularly, "!(y)" means a wireless packet y is sent once the event happens; and "$\leftarrow$" means value assignment. Same notational rules also apply to Fig. 4.4.

sub-sub-section and in Fig. 4.3, we use τ to represent this local clock of A_R, unless otherwise denoted.

2. Initially (i.e. at t_0), R dwells in the "Init" mode, and the initial value of τ is set to Δ_{reset}, where $\Delta_{\mathsf{reset}} > 0$ is a constant defined by Eq. (4.20).

3. When in mode "Init," R drives along the current lane at speed v_{lim} (i.e. $|\dot{\vec{p}}(R,t)| = v_{\mathsf{lim}}$). the local clock τ increases continuously (i.e. $\dot{\tau} = 1$), but must not exceed the range of $[0, \Delta_{\mathsf{reset}})$. Once τ satisfies $\tau \geqslant \Delta_{\mathsf{reset}}$, an "IntendToChangeLane" event is triggered (see Fig. 4.3 event "IntendToChangeLane"), which carries out the following action.

   1: Set τ to 0;
   2: Find the closest (along the X-axis) target lane CAV before R (i.e. the "leader CAV"), denote it as L (or set L to $\varnothing$ if there is no such vehicle);
   3: Find the closest (along the X-axis) target lane CAV not before R (i.e. the "follower CAV"), denoted it as F (or set F to $\varnothing$ if there is no such vehicle);
   4: Measure the distance to L, denote it as $d_0(L,R)$ (or set $d_0(L,R)$ to $+\infty$ if $L = \varnothing$);
   5: Measure the distance to F, denote it as $d_0(R,F)$ (or set $d_0(R,F)$ to $+\infty$ if $F = \varnothing$).

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

4. Mode "L_0" is a transient mode where R cannot stay. Upon entrance to "L_0," R immediately triggers one of the following events (see Fig. 4.3 "Event1," "Event2," "Event3," and "Event4").

   Case1 (Event1): If $d_0(L,R) \geqslant D_1$ and $d_0(R,F) \geqslant D_3$, then "Event1" of A_R is triggered. This event sets the local clock τ to 0. After this action, R enters the "LaneChanging(v_{lim})" mode.

Case2 (Event2): If $d_0(L,R) < D_1$ and $d_0(R,F) \geqslant D_2$, then "Event2" of A_R is triggered. This event carries out the following action.

1: Set $\delta_{\mathsf{low}}^{\mathsf{LR}}$ to $\frac{v_{\mathsf{lim}}\Delta^* - d_0(L,R)}{\widetilde{v}}$;

2: Set the local clock τ to 0.

After the above action, R enters the "Decelerating" mode.

Case3 (Event3): If $d_0(L,R) \geqslant D_1$ and $d_0(R,F) < D_3$, then "Event3" of A_R is triggered. This event carries out the following action.

1: Send a wireless packet "LaneChangeReq" to F, carrying the data payload of $d_0(L,R)$ and $d_0(R,F)$;

2: Set the local clock τ to 0.

After the above action, R enters the "Requesting1" mode to wait for F's reply.

Case4 (Event4): If $d_0(L,R) < D_1$ and $d_0(R,F) < D_2$, then "Event4" of A_R is triggered. This event carries out the following action.

1: Send a wireless packet "LaneChangeReq" to F, carrying the data payload of $d_0(L,R)$ and $d_0(R,F)$;

2: Set the local clock τ to 0.

After the above action, R enters the "Requesting2" mode to wait for F's reply.

5. When in mode "LaneChanging(v_{lim})," R changes the lane as per the mechanical lane change routine $\mathsf{lc}(v_{\mathsf{lim}}, \tau)$ (see Section 4.3.2), where τ is the local clock set to 0 at the start of the routine, and increases continuously (i.e. $\dot{\tau} = 1$). When τ reaches $\delta_{\mathsf{lc}}(v_{\mathsf{lim}})$ (i.e. the duration needed for the routine with initial speed v_{lim}, see Section 4.3.2), the mechanical lane change routine completes, and R enters the "ConstSpeedTargetLane" mode via "Event5" of A_R.

6. When in mode "ConstSpeedTargetLane," R has completed the lane change and is driving along the target lane at speed v_{lim} (i.e. $|\dot{\vec{p}}(R,t)| = v_{\mathsf{lim}}$).

7. When in mode "Decelerating," R decelerates from v_{lim} to v_{low} as per the decelera-

tion routine $\mathsf{dec}(v_{\mathsf{lim}}, v_{\mathsf{low}}, \tau)$ (see Section 4.3.2), where τ is the local clock set to 0 at the start of the routine and increases continuously (i.e. $\dot{\tau} = 1$). When τ reaches $\delta^{\ddagger}_{\mathsf{d}}$, the deceleration routine completes, and "Event6" of A_R is triggered. This event sets the local clock τ to 0. After this action, R enters the "ConstLowSpeed" mode.

8. When in mode "ConstLowSpeed," R drives at the speed of v_{low} (i.e. $|\dot{\vec{p}}(R,t)| = v_{\mathsf{low}}$) for $\delta^{\mathsf{LR}}_{\mathsf{low}}$ seconds (as $\dot{\tau} = 1$ and $0 \leqslant \tau < \delta^{\mathsf{LR}}_{\mathsf{low}}$, note τ is set to 0 by "Event6" of A_R and $\delta^{\mathsf{LR}}_{\mathsf{low}}$ is set by "Event2" of A_R or event "GotDecelerate"). When τ reaches $\delta^{\mathsf{LR}}_{\mathsf{low}}$, "Event7" of A_R is triggered. This event sets the local clock τ to 0. After this action, R enters the "LaneChanging(v_{low})" mode.

9. When in mode "LaneChanging(v_{low})," R changes the lane as per the mechanical lane change routine $\mathsf{lc}(v_{\mathsf{low}}, \tau)$ (see Section 4.3.2), where τ is the local clock set to 0 at the start of the routine, and increases continuously (i.e. $\dot{\tau} = 1$). When τ reaches $\delta_{\mathsf{lc}}(v_{\mathsf{low}})$ (i.e. the duration needed for the routine with initial speed v_{low}, see Section 4.3.2), the mechanical lane change routine completes, and "Event8" of A_R is triggered. This event sets the local clock τ to 0. After this action, R enters the "AcceleratingTargetLane" mode.

10. When in mode "AcceleratingTargetLane," R accelerates from v_{low} to v_{lim} as per the acceleration routine $\mathsf{acc}(v_{\mathsf{low}}, v_{\mathsf{lim}}, \tau)$ (see Section 4.3.2), where τ is the local clock set to 0 at the start of the routine and increases continuously (i.e. $\dot{\tau} = 1$). When τ reaches $\delta^{\ddagger}_{\mathsf{a}}$, the acceleration routine completes, and R enters the "ConstSpeedTargetLane" mode via "Event9" of A_R.

11. When in mode "Requesting1," R maintains the speed of v_{lim} (i.e. $|\dot{\vec{p}}(R,t)| = v_{\mathsf{lim}}$), the local clock τ increases continuously (i.e. $\dot{\tau} = 1$), and must not exceed its range constraint of $[0, \Delta_{\mathsf{nonzeno}})$, where $\Delta_{\mathsf{nonzeno}} > 0$ is a configurable constant (see Section 4.4.2 2)). In this mode, R may trigger one of the following events (see Fig. 4.3 event "GotLaneChangeAcpt" and "RequestTimeout1").

Case1 (GotLaneChangeAcpt): Under the local clock constraint of $\tau \in [0, \Delta_{\mathsf{nonzeno}})$, if a "LaneChangeAcpt" wireless packet is received from F, then from the packet's data payload, R gets the value for $\delta^{\mathsf{lc}}_{\mathsf{defer}}$. Meanwhile, the "GotLaneChangeAcpt" event is triggered. This event sets the local clock τ to 0. After this action, R enters the "DeferringLaneChange" mode.

Case2 (RequestTimeout1): If the local clock τ reaches $\Delta_{\mathsf{nonzeno}}$ without R receiving any "LaneChangeAcpt" wireless packet, then the "RequestTimeout1" event is triggered. This event sets the local clock τ to 0. After this action, R enters the "Init" mode.

12. When in mode "DeferringLaneChange," R maintains the speed of v_{lim} (i.e. $|\dot{\vec{p}}(R,t)| = v_{\mathsf{lim}}$), the local clock τ increases continuously (i.e. $\dot{\tau} = 1$), and must not exceed its range constraint of $[0, \delta^{\mathsf{lc}}_{\mathsf{defer}})$, where $\delta^{\mathsf{lc}}_{\mathsf{defer}}$ is a previously received value from wireless packet "LaneChangeAcpt." If τ reaches $\delta^{\mathsf{lc}}_{\mathsf{defer}}$, then the "ActualLaneChange" event is triggered. This event sets the local clock τ to 0. After this action, R enters the "LaneChanging(v_{lim})" mode.

13. When in mode "Requesting2," R maintains the speed of v_{lim} (i.e. $|\dot{\vec{p}}(R,t)| = v_{\mathsf{lim}}$), the local clock τ increases continuously (i.e. $\dot{\tau} = 1$), and must not exceed its range constraint of $[0, \Delta_{\mathsf{nonzeno}})$, where $\Delta_{\mathsf{nonzeno}} > 0$ is a configurable constant (see Section 4.4.2 2)). In this mode, R may trigger one of the following events (see Fig. 4.3 event "GotDecelerate" and "RequestTimeout2").

Case1 (GotDecelerate): Under the local clock constraint of $\tau \in [0, \Delta_{\mathsf{nonzeno}})$, if a "Decelerate" wireless packet is received from F, then from the packet's data payload, R gets the value for $\delta^{\mathsf{Rdec}}_{\mathsf{defer}}$ (the first data payload element) and the value for $\delta^{\mathsf{LR}}_{\mathsf{low}}$ (the second data payload element). Meanwhile, the "GotDecelerate" event is triggered. This event sets the local clock τ to 0. After this action, R enters the "DeferringDeceleration" mode.

Case2 (RequestTimeout2): If the local clock τ reaches $\Delta_{\mathsf{nonzeno}}$ without R receiving any "Decelerate" wireless packet, then the "RequestTimeout2" event is triggered. This event sets the local clock τ to 0. After this action, R enters the "Init" mode.

14. When in mode "DeferringDeceleration," R maintains the speed of v_{lim} (i.e. $|\dot{\vec{p}}(R,t)| = v_{\mathsf{lim}}$), the local clock τ increases continuously (i.e. $\dot{\tau} = 1$), and must not exceed its range constraint of $[0, \delta^{\mathsf{Rdec}}_{\mathsf{defer}})$, where $\delta^{\mathsf{Rdec}}_{\mathsf{defer}}$ is a previously received value from wireless packet "Decelerate." If τ reaches $\delta^{\mathsf{Rdec}}_{\mathsf{defer}}$, then the "ActualDecelerate" event is triggered. This event sets the local clock τ to 0. After this action, R enters the "Decelerating" mode.

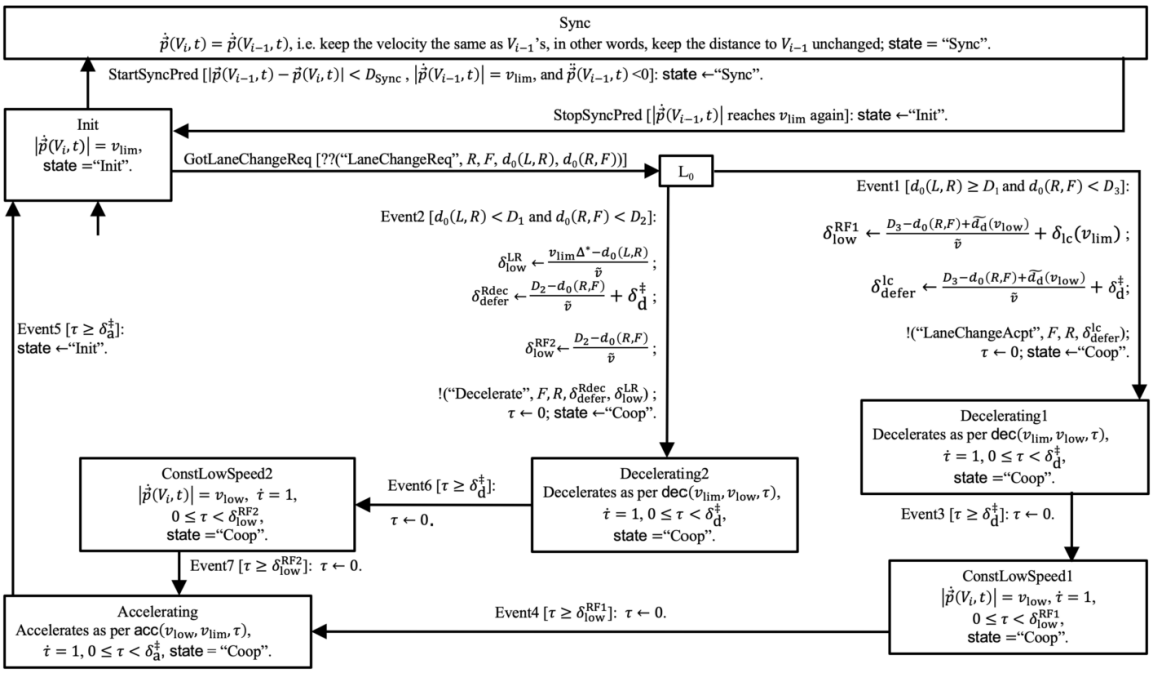


Figure 4.4: Hybrid automaton A_i (see Section 4.4.2 Target Lane CAV V_i Protocol Behaviors) for the target lane CAV V_i ($i \in \{1, 2, \ldots, n\}$); note as V_0 does not exist, for V_1, the event "StartSyncPred" can never happen.

Target Lane CAV V_i Protocol Behaviors

A target lane CAV V_i's ($i \in \{1, 2, \ldots, n\}$) protocol behaviors are defined by a hybrid automaton A_i, which is illustrated by Fig. 4.4 and explained by the following text.

1. The hybrid automaton A_i has the following modes: "Init," "L_0," "Decelerating1," "Decelerating2," "ConstLowSpeed1," "ConstLowSpeed2," "Accelerating," and "Sync." At any time instance, V_i dwells in one of these modes. A_i also has a local clock for various timing purposes. *In this sub-sub-section and in Fig. 4.4, we use τ to represent this local clock of A_i, unless otherwise denoted.* A_i also has another local variable, state, whose value ("Init", "Coop," and "Sync") is determined by the current mode.

2. Initially (i.e. at t_0), V_i dwells in the "Init" mode, and the initial value of state is set to "Init."

3. When in mode "Init," V_i maintains the speed of v_{lim} (i.e. $|\dot{\vec{p}}(V_i, t)| = v_{\mathsf{lim}}$), and state maintains the value of "Init." In this mode, V_i may trigger one of the following events (see Fig. 4.4 event "GotLaneChangeReq" and "StartSyncPred").

   Case1 (GotLaneChangeReq): If a "LaneChangeReq" wireless packet is received from the requesting CAV R, then it implies that V_i is currently the follower F for the lane change. From the packet's data payload, V_i gets the value for $d_0(L, R)$ (the first data payload element) and the value for $d_0(R, F)$ (the second data payload element). Meanwhile, the "GotLaneChageReq" event is triggered, which switches V_i to the transient mode "L_0."

   Case2 (StartSyncPred, only for those V_i whose $i > 1$): If i) the current distance to V_{i-1} is less than D_{Sync} (i.e. $|\vec{p}(V_{i-1}, t) - \vec{p}(V_i, t)| < D_{\mathsf{Sync}}$, where $D_{\mathsf{Sync}} > 0$ is a constant defined by Eq. (4.14)), and ii) V_{i-1} is currently decelerating from v_{lim} (i.e. $|\dot{\vec{p}}(V_{i-1}, t)| = v_{\mathsf{lim}}$ and $\ddot{\vec{p}}(V_{i-1}, t) < 0$), then the "StartSyncPred" event

is triggered. This event sets state to "Sync." After this action, V_i enters the "Sync" mode.

4. Mode "L_0" is a transient mode where V_i cannot stay. Upon entrance to "L_0," V_i immediately triggers one of the following events (see Fig. 4.4 "Event1" and "Event2").

   Case1 (Event1): If $d_0(L,R) \geqslant D_1$ and $d_0(R,F) < D_3$ (where $d_0(L,R)$ and $d_0(R,F)$ are previously received values from wireless packet "LaneChangeReq," while D_1 and D_3 are constants defined by Eq. (4.11) and (4.13)), then "Event1" of A_i is triggered[1], which carries out the following action.

   1: Set $\delta_{\mathsf{low}}^{\mathsf{RF1}}$ to $\frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta_{\mathsf{lc}}(v_{\mathsf{lim}})$;
   2: Set $\delta_{\mathsf{defer}}^{\mathsf{lc}}$ to $\frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta_{\mathsf{d}}^{\ddagger}$;
   3: Send a "LaneChangeAcpt" wireless packet to R, carrying the data payload of $\delta_{\mathsf{defer}}^{\mathsf{lc}}$, which tells R to change lane after $\delta_{\mathsf{defer}}^{\mathsf{lc}}$ seconds;
   4: Set the local clock τ to 0;
   5: Set state to "Coop."

   After the above action, V_i enters mode "Decelerating1."

   Case2 (Event2): If $d_0(L,R) < D_1$ and $d_0(R,F) < D_2$ (where $d_0(L,R)$ and $d_0(R,F)$ are previously received values from wireless packet "LaneChangeReq," while $D_1$and D_2 are constants defined by Eq. (4.11) and (4.12)), then "Event2" of A_i is triggered, which carries out the following action.

   1: Set $\delta_{\mathsf{low}}^{\mathsf{LR}}$ to $\frac{v_{\mathsf{lim}}\Delta^* - d_0(L,R)}{\widetilde{v}}$;
   2: Set $\delta_{\mathsf{defer}}^{\mathsf{Rdec}}$ to $\frac{D_2 - d_0(R,F)}{\widetilde{v}} + \delta_{\mathsf{d}}^{\ddagger}$;
   3: Set $\delta_{\mathsf{low}}^{\mathsf{RF2}}$ to $\frac{D_2 - d_0(R,F)}{\widetilde{v}}$;
   4: Send a "Decelerate" wireless packet to R, carrying the data payload of $\delta_{\mathsf{defer}}^{\mathsf{Rdec}}$ and $\delta_{\mathsf{low}}^{\mathsf{LR}}$, which tells R to decelerate in $\delta_{\mathsf{defer}}^{\mathsf{Rdec}}$ seconds and (after

[1]Note "Event$j$" of A_i and "Event$j$" of A_R ($j = 1, 2, \ldots$) are different events.

deceleration) maintain low speed v_{low} for $\delta_{\mathsf{low}}^{\mathsf{LR}}$ seconds before starting the mechanical lane change routine;

5: Set the local clock τ to 0;

6: Set state to "Coop."

After the above action, V_i enters mode "Decelerating2."

5. When in mode "Decelerating1," state maintains the value of "Coop;" meanwhile, V_i decelerates from v_{lim} to v_{low} as per the deceleration routine $\mathsf{dec}(v_{\mathsf{lim}}, v_{\mathsf{low}}, \tau)$ (see Section 4.3.2), where τ is the local clock set to 0 at the start of the routine and increases continuously (i.e. $\dot{\tau} = 1$). When τ reaches $\delta_{\mathsf{d}}^{\ddagger}$, the deceleration routine completes, and "Event3" of A_i is triggered. This event sets the local clock τ to 0. After this action, V_i enters the "ConstLowSpeed1" mode.

6. When in mode "ConstLowSpeed1," state maintains the value of "Coop;" meanwhile, V_i drives at the speed of v_{low} (i.e. $|\dot{\vec{p}}(V_i, t)| = v_{\mathsf{low}}$) for $\delta_{\mathsf{low}}^{\mathsf{RF1}}$ seconds (as $\dot{\tau} = 1$ and $0 \leqslant \tau < \delta_{\mathsf{low}}^{\mathsf{RF1}}$, note τ is set to 0 by "Event3" of A_i and $\delta_{\mathsf{low}}^{\mathsf{RF1}}$ is set by "Event1" of A_i). When τ reaches $\delta_{\mathsf{low}}^{\mathsf{RF1}}$, "Event4" of A_i is triggered. This event sets the local clock τ to 0. After this action, V_i enters the "Accelerating" mode.

7. When in mode "Accelerating," state maintains the value of "Coop;" meanwhile, V_i accelerates from v_{low} to v_{lim} as per the acceleration routine $\mathsf{acc}(v_{\mathsf{low}}, v_{\mathsf{lim}}, \tau)$ (see Section 4.3.2), where τ is the local clock set to 0 at the start of the routine and increases continuously (i.e. $\dot{\tau} = 1$). When τ reaches $\delta_{\mathsf{a}}^{\ddagger}$, the acceleration routine completes, and "Event5" of A_i is triggered. This event sets state to "Init." After this action, V_i enters the "Init" mode.

8. Mode "Decelerating2" behavior is the same as that of mode "Decelerating1," except that when the deceleration routine completes, "Decelerating2" triggers "Event6" of A_i instead of "Event3" of A_i. "Event6" of A_i sets the local clock τ to 0. After this action, V_i enters the "ConstLowSpeed2" mode.

9. When in mode "ConstLowSpeed2," state maintains the value of "Coop;" meanwhile, V_i drives at the speed of v_{low} (i.e. $|\dot{\vec{p}}(V_i,t)| = v_{\mathsf{low}}$) for $\delta_{\mathsf{low}}^{\mathsf{RF2}}$ seconds (as $\dot{\tau}=1$ and $0 \leqslant \tau < \delta_{\mathsf{low}}^{\mathsf{RF2}}$, note τ is set to 0 by "Event6" of A_i and $\delta_{\mathsf{low}}^{\mathsf{RF2}}$ is set by "Event2" of A_i). When τ reaches $\delta_{\mathsf{low}}^{\mathsf{RF2}}$, "Event7" of A_i is triggered. This event sets the local clock τ to 0. After this action, V_i enters the "Accelerating" mode.

10. Mode "Sync" is only applicable to V_i where $i > 1$. When in this mode, state maintains the value of "Sync;" and V_i keeps its speed the same as that of V_{i-1}, in other words, V_i keeps the distance to V_{i-1} unchanged (see Section 4.5.1 on more discussions). When V_{i-1} recovers the speed of v_{lim}, the "StopSyncPred" event is triggered. This event sets state to "Init." After this event, V_i enters the "Init" mode.

4.4.3 Analysis

We now analyze the proposed protocol. Specifically, we claim the following theorem, which answers Research Problem 1 and 2.

Theorem 2 *Suppose the following constraints hold:*

(c1) $v_{\mathsf{lim}} > v_{\mathsf{low}} > 0$, *i.e. Ineq.* (4.2)*;*

(c2) $\Delta^* > 0$ *and* $\Delta_{\mathsf{nonzeno}} > 0$*;*

(c3) $\Delta^* > \delta_{\mathsf{lc}}(v_{\mathsf{low}}) > \delta_{\mathsf{lc}}(v_{\mathsf{lim}}) > \delta_{\mathsf{d}}^{\ddagger}$*;*

(c4) $\delta_{\mathsf{d}}^{\ddagger} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) \geqslant \delta_{\mathsf{a}}^{\ddagger}$*;*

(c5) $\Delta_{\mathsf{nonzeno}} < \mathsf{min}\{\Delta_{\mathsf{Coop}}^{\mathsf{Event1,max}}, \Delta_{\mathsf{Coop}}^{\mathsf{Event2,max}}\}$*;*

(c6) *when no packet is lost, maximum one hop wireless packet request and reply delay* $\delta_{\mathsf{comm}}^{\mathsf{max}} << \mathsf{min}\{\Delta_{\mathsf{nonzeno}}, \Delta^*, \delta_{\mathsf{a}}^{\ddagger}, \delta_{\mathsf{d}}^{\ddagger}, \delta_{\mathsf{lc}}(v_{\mathsf{lim}}), \delta_{\mathsf{lc}}(v_{\mathsf{low}})\}$*, hence is negligible.*

Then we have the following claims.

Claim 1 *(Safety): $\forall t \in [t_0, +\infty)$, for any two CAVs x and y on the target lane, one and only one of the following sustains:*

(i) (x, y) is CTH-Δ^* safe at t, or

(ii) (y, x) is CTH-Δ^* safe at t.

Claim 2 *(Liveness (Automatic Resetting)): Suppose at $t_1 \in [t_0, +\infty)$, R leaves hybrid automaton A_R mode "Init," while V_is ($i = 1, 2, \ldots, n$) are all residing in the respective hybrid automaton A_i mode "Init." Then $\exists t_2 \in (t_1, t_1 + \Delta_{\mathsf{reset}}]$ s.t.*

*(**Stable State 1**) at t_2, V_1, V_2, ..., V_n, and R are in respective hybrid automata mode "Init", or*

*(**Stable State 2**) at t_2, V_1, V_2, ..., V_n are in respective hybrid automata mode "Init" and R is in A_R's mode "ConstSpeedTargetLane".*

■

In order to prove Theorem 2, we need to first propose/prove several definitions and claims.

Definition 4 (Coop-Duration) *For a target lane CAV V_i ($i \in \{1, 2, \ldots, n\}$), suppose its hybrid automaton (i.e. A_i) variable,* state*, changes from "Init" to "Coop" at $t_3 \in [t_0, +\infty)$, then as per Fig. 4.4, the* state *must change back to "Init" at some finite t_4 (where $t_3 < t_4 \leqslant t_3 + \Delta^{\mathsf{max}}_{\mathsf{Coop}}$, see Eq. (4.19)(4.17)(4.18) for the definition of $\Delta^{\mathsf{max}}_{\mathsf{Coop}}$). That is, $\forall t \in (t_3, t_4]$,* state *= "Coop";[2] and at t_4^+,* state *= "Init". We call $(t_3, t_4]$ a "*coop-duration*". Note as per Fig. 4.4, it is easy to see that $\Delta^{\mathsf{max}}_{\mathsf{Coop}}$ is a loose upper bound to the time length of a coop-duration.* ■

[2]Note, if we regard hybrid automaton discrete variables' values are left continuous along time axis, then at t_3, we regard state ="Init".

Lemma 7 *Any two coop-durations* $(t_5, t_6]$ *and* $(t_7, t_8]$ *respectively belonging to two different target lane CAVs can never overlap nor connect. Formally, i.e.,* $[t_5, t_6] \cap [t_7, t_8] = \varnothing$. ■

Proof: Suppose $[t_5, t_6] \cap [t_7, t_8] \neq \varnothing$ and suppose $t_9 \in [t_5, t_6] \cap [t_7, t_8]$. Then $t_5 \in [t_9 - \Delta^{\mathsf{max}}_{\mathsf{Coop}}, t_9]$ and $t_7 \in [t_9 - \Delta^{\mathsf{max}}_{\mathsf{Coop}}, t_9]$, therefore $|t_5 - t_7| \leqslant \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. This means R sends two different "LaneChangeReq" packets within $\Delta^{\mathsf{max}}_{\mathsf{Coop}}$. This contradicts $\Delta_{\mathsf{reset}} > \Delta^{\mathsf{max}}_{\mathsf{Coop}}$ (see Fig. 4.3 "Init" mode and Eq. (4.20)). □

Lemma 8 *Any two coop-durations* $(t_{10}, t_{11}]$ *and* $(t_{12}, t_{13}]$ *can never overlap nor connect. Formally, i.e.,* $[t_{10}, t_{11}] \cap [t_{12}, t_{13}] = \varnothing$. ■

Proof: In addition to Lemma 7, applying similar reasonings, we can prove coop-durations of a same target lane CAV V_i cannot overlap nor connect. □

Lemma 9 $\forall t \in [t_0, +\infty)$*, if no target lane CAV* V_i *is in coop-duration at* t*, then all target lane CAVs (i.e.* V_1*,* V_2*, …,* V_n*) are in "Init" mode at* t*.* ■

Proof: According to Fig. 4.4, if $\exists V_i$, whose state = "Sync" at t, then there must be an V_j in a coop-duration at t. □

Lemma 10 *Suppose* $(t_{14}, t_{15}] \subseteq [t_0, +\infty)$ *is the first ever happened coop-duration, then* $\forall t \in [t_0, t_{15}]$, $\forall i \in \{1, 2, \ldots, n-1\}$, (V_i, V_{i+1}) *is CTH-*Δ^* *safe at* t*.* ■

Proof: See Appendix B.1 for details. □

Lemma 11 $\forall t \in [t_0, +\infty)$, $\forall i \in \{1, 2, \ldots, n-1\}$, (V_i, V_{i+1}) *is CTH-*Δ^* *safe at* t*.* ■

Proof: See Appendix B.2 for details. □

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

Proof: Due to Lemma 11, the first swapping never happens. □

Lemma 12 *Suppose CAV* R *finishes a mechanical lane change routine at* $t_{16} \in [t_0, +\infty)$*, then* $\forall i \in \{1, 2, \ldots, n\}$*, one and only one of the following claims sustain.*

Claim 1 (V_i, R) *is CTH-*Δ^* *safe throughout* $[t_{16}, +\infty)$*.*

Claim 2 (R, V_i) *is CTH-*Δ^* *safe throughout* $[t_{16}, +\infty)$*.* ■

Proof: See Appendix B.3 for details. □

Now we are ready to prove Theorem 2.

Proof of Theorem 2 **Claim 1**:

In case $x, y \in \{V_1, V_2, \ldots, V_n\}$, the claim sustains due to Lemma 11 (in case x and y are not consecutive, e.g. $x = V_i$ and $y = V_{i+k}$, where $k > 1$, then the distance between x and y is no less than the distance between V_{i+k-1} and y, hence the CTH-Δ^* safety rule still sustains for (x, y)).

In case $x \in \{V_1, V_2, \ldots, V_n\}$ and $y = R$, or the reverse, the claim sustains due to Lemma 12.

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

Proof of Theorem 2 **Claim 2**:

Based on if the requesting CAV R sends the "LaneChangeReq" wireless packet at t_1, there are two cases.

Case 1: R does not send "LaneChangeReq" to the follower CAV $F = V_i$ ($i \in \{1, 2, \ldots, n\}$) at t_1. This includes two further cases.

Case 1.1: R triggers "Event1" of A_R (see Fig. 4.3) at t_1. Then R will enter the "ConstSpeedTargetLane" mode at $t_{17} \overset{\mathsf{def}}{=} t_1 + \delta_{\mathsf{lc}}(v_{\mathsf{lim}}) < t_1 + \Delta_{\mathsf{reset}}$. Meanwhile, all target lane CAVs $V_1 \sim V_n$ remain in their respective "Init" mode throughout $[t_1, +\infty)$. Therefore, t_{17} is a time instance that satisfies the claim's description (we call such a time instance a "*valid time instance*" in the following).

Case 1.2: R triggers "Event2" of A_R (see Fig. 4.3) at t_1.

Then R will enter the "ConstSpeedTargetLane" mode at

$$\begin{aligned} & t_{18} \overset{\mathsf{def}}{=} t_1 + \delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{LR}}_{\mathsf{low}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \delta^{\ddagger}_{\mathsf{a}} \\ = \;& t_1 + \delta^{\ddagger}_{\mathsf{d}} + \frac{v_{\mathsf{lim}}\Delta^* - d_0(L,R)}{\widetilde{v}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \delta^{\ddagger}_{\mathsf{a}} \\ \leqslant \;& t_1 + \delta^{\ddagger}_{\mathsf{d}} + \frac{v_{\mathsf{lim}}\Delta^*}{\widetilde{v}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \delta^{\ddagger}_{\mathsf{a}} \\ = \;& t_1 + \delta^{\ddagger}_{\mathsf{d}} + \delta^{\ddagger}_{\mathsf{a}} + \frac{v_{\mathsf{lim}}\Delta^* + \widetilde{v}\delta_{\mathsf{lc}}(v_{\mathsf{low}})}{\widetilde{v}} \\ \leqslant \;& t_1 + \delta^{\ddagger}_{\mathsf{d}} + \delta^{\ddagger}_{\mathsf{a}} + \frac{D_2}{\widetilde{v}} \\ \leqslant \;& t_1 + \Delta^{\mathsf{Event2,max}}_{\mathsf{Coop}} \leqslant t_1 + \Delta^{\mathsf{max}}_{\mathsf{Coop}} < t_1 + \Delta_{\mathsf{reset}} \end{aligned}$$

(see Eq. (4.18)(4.19)).

Meanwhile, all target lane CAVs $V_1 \sim V_n$ remain in their respective "Init" mode throughout $[t_1, +\infty)$. Therefore, t_{18} is a valid time instance.

Combining **Case 1.1** and **Case 1.2**, **Case 1** complies with the claim.

Case 2: R sends "LaneChangeReq" to the follower CAV $F = V_i$ ($i \in \{1, 2, \ldots, n\}$) at t_1.

This also implies that R will not send another "LaneChangeReq" wireless packet during $(t_1, t_1 + \Delta_{\mathsf{reset}}]$ (see Fig. 4.3) (†)

Meanwhile, at the receiver's end, we can have two cases.

Case 2.1: $F = V_i$ receives the "LaneChangeReq" wireless packet at t_1 (see event "GotLaneChangeReq" in Fig. 4.4), which leads to two possible cases.

Case 2.1.1: The reception of the "LaneChangeReq" packet triggers "Event1" of A_i (see Fig. 4.4) at t_1. Then a coop-duration starts at t_1 and ends at $t_{19} \overset{\mathsf{def}}{=} t_1 + \delta_{\mathsf{d}}^{\ddagger} + \delta_{\mathsf{low}}^{\mathsf{RF1}} + \delta_{\mathsf{a}}^{\ddagger}$. Let $t_{20} \overset{\mathsf{def}}{=} t_1 + \Delta_{\mathsf{Coop}}^{\mathsf{Event1,max}}$, and $t_{21} \overset{\mathsf{def}}{=} t_{20} + \Delta_{\mathsf{nonzeno}}$. Then we have $t_{19} \leqslant t_{20} < t_{21} \leqslant t_1 + \Delta_{\mathsf{reset}}$ (due to Eq. (4.17)(4.20)(4.19) and the definition of $\delta_{\mathsf{low}}^{\mathsf{RF1}}$, see Fig. 4.4 "Event1").

Meanwhile, due to (†), a second coop-duration will not start during $(t_1, t_1 + \Delta_{\mathsf{reset}}]$. (††)

Due to (††), V_i is in "Init" at $\forall t \in (t_{20}, t_{21})$. Without loss of generality, pick $t_{22} \overset{\mathsf{def}}{=} \frac{t_{20}+t_{21}}{2}$. Meanwhile, (††) also implies no other target lane CAV is in a coop-duration at t_{22}. Hence due to Lemma 9, $V_1 \sim V_n$ are all in "Init" mode at t_{22}. († † †)

In order to determine R's mode at t_{22}, note V_i's "Event1" sends the "LaneChangeAcpt" wireless packet to R. This can have two cases.

Case 2.1.1.1: If R receives the "LaneChangeAcpt" packet at t_1^+, then it will enter the "ConstSpeedTargetLane" mode by $t_1 + \delta_{\mathsf{defer}}^{\mathsf{lc}} + \delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = t_1 + \delta_{\mathsf{low}}^{\mathsf{RF1}} + \delta_{\mathsf{d}}^{\ddagger} \leqslant t_{20} < t_{22}$.

Case 2.1.1.2: If R loses the "LaneChangeAcpt" packet at t_1^+, then at t_{22}, R must be in mode "Init" (see Fig. 4.3). This is because $t_1 + \Delta_{\mathsf{reset}} > t_{22} > t_{20} = t_1 + \Delta_{\mathsf{Coop}}^{\mathsf{Event1,max}} > t_1 + \Delta_{\mathsf{nonzeno}}$ (due to (**c5**)).

Combining **Case 2.1.1.1**, **Case 2.1.1.2**, and († † †), we see t_{22} is a valid time instance, hence **Case 2.1.1** complies with the claim.

Case 2.1.2 The reception of the "LaneChangeReq" packet triggers "Event2" of V_i (see Fig. 4.4) at t_1. Then a coop-duration starts at t_1 and ends at $t_{23} \overset{\mathsf{def}}{=} t_1 + \delta_{\mathsf{d}}^{\ddagger} + \delta_{\mathsf{low}}^{\mathsf{RF2}} + \delta_{\mathsf{a}}^{\ddagger}$. Let $t_{24} \overset{\mathsf{def}}{=} t_1 + \Delta_{\mathsf{Coop}}^{\mathsf{Event2,max}} + \delta_{\mathsf{d}}^{\ddagger} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \Delta^*$ and $t_{25} \overset{\mathsf{def}}{=} t_{24} + \Delta_{\mathsf{nonzeno}}$. Then we have $t_{23} \leqslant t_{24} < t_{25} \leqslant t_1 + \Delta_{\mathsf{reset}}$ (due to Eq. (4.18)(4.20)(4.19) and the definition of $\delta_{\mathsf{low}}^{\mathsf{RF2}}$, see Fig. 4.4 "Event2").

Meanwhile, due to (†), a second coop-duration will not start during $(t_1, t_1 + \Delta_{\mathsf{reset}}]$. († † ††)

Due to († † ††) V_i is in "Init" at $\forall t \in (t_{24}, t_{25})$. Without loss of generality, pick $t_{26} \overset{\mathsf{def}}{=} \frac{t_{24}+t_{25}}{2}$. Meanwhile, († † ††) also implies no other target lane CAV is in a coop-duration at t_{26}. Hence due to Lemma 9, $V_1 \sim V_n$ are all in "Init" mode at t_{26}. († † † † †)

In order to determin R's mode at t_{26}, note V_i's "Event2" sends the "Decelerate" wireless packet to R. This can have two cases.

Case 2.1.2.1: If R receives the "Decelerate" packet at t_1^+, then it will enter the "ConstSpeedTargetLane" mode by

$$\begin{aligned}
& t_1 + \delta^{\mathsf{Rdec}}_{\mathsf{defer}} + \delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{LR}}_{\mathsf{low}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \delta^{\ddagger}_{\mathsf{a}} \\
= \; & t_1 + \frac{D_2 - d_0(R,F)}{\widetilde{v}} + \delta^{\ddagger}_{\mathsf{d}} + \delta^{\ddagger}_{\mathsf{d}} + \frac{v_{\mathsf{lim}}\Delta^* - d_0(L,R)}{\widetilde{v}} \\
& + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \delta^{\ddagger}_{\mathsf{a}} \\
= \; & t_1 + \frac{D_2}{\widetilde{v}} + \delta^{\ddagger}_{\mathsf{d}} + \delta^{\ddagger}_{\mathsf{a}} + \delta^{\ddagger}_{\mathsf{d}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) \\
& + \frac{v_{\mathsf{lim}}\Delta^* - (d_0(R,F) + d_0(L,R))}{\widetilde{v}} \\
\leqslant \; & t_1 + \Delta^{\mathsf{Event2,max}}_{\mathsf{Coop}} + \delta^{\ddagger}_{\mathsf{d}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \frac{v_{\mathsf{lim}}\Delta^* - v_{\mathsf{low}}\Delta^*}{\widetilde{v}} \\
& (\text{due to Lemma 11 and } F\text{'s speed} \geqslant v_{\mathsf{low}}) \\
= \; & t_1 + \Delta^{\mathsf{Event2,max}}_{\mathsf{Coop}} + \delta^{\ddagger}_{\mathsf{d}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) + \Delta^* \\
= \; & t_{24} < t_{26}
\end{aligned}$$

Case 2.1.2.2: If R loses the "Decelerate" packet at t_1^+, then at t_{26}, R must be in mode "Init" (see Fig. 4.3). This is because $t_{26} > t_{24} > t_1 + \Delta^{\mathsf{Event2,max}}_{\mathsf{Coop}} > t_1 + \Delta_{\mathsf{nonzeno}}$ (due to (**c5**)) and $t_{26} < t_{25} \leqslant t_1 + \Delta_{\mathsf{reset}}$.

Combining **Case 2.1.2.1**, **Case 2.1.2.2**, and († † † † †), we see t_{26} is a valid time instance, hence **Case 2.1.2** complies with the claim.

Combining **Case 2.1.1** and **Case 2.1.2**, **Case 2.1** compiles with the claim.

Case 2.2: $F = V_i$ loses the "LaneChangeReq" packet at t_1 (see event "GotLaneChangeReq" in Fig. 4.4). Then, due to (†), nothing happens to $V_1 \sim V_n$ during $[t_1, t_1 +$

Δ_{reset}). Due to Lemma 9, $\forall t \in [t_1, t_1 + \Delta_{\mathsf{reset}})$, $V_1 \sim V_n$ are all in "Init" mode at t. († † † † ††)

Let $t_{27} \stackrel{\mathsf{def}}{=} t_1 + \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. Then R must be in "Init" mode at t_{27}, as $t_1 + \Delta_{\mathsf{nonzeno}} < t_{27} < t_1 + \Delta_{\mathsf{reset}}$. Meanwhile, due to († † † † ††), t_{27} is a valid time instance. Hence **Case 2.2** compiles with the claim.

Combining **Case 2.1** and **Case 2.2**, **Case 2** complies with the claim.

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

Due to (∗) and (∗∗), the theorem sustains. ■

4.5 Further Discussions

4.5.1 V2V Communication Failures between Target Lane CAVs are Irrelevant

V2V communications between the target lane CAVs (if used) are only used in the "Sync" mode (see Fig. 4.4); and are only used between two consecutive target lane CAVs (V_i and V_{i+1}, where $i = 1, 2, \ldots, n-1$) for three possible cases: to trigger the "StartSyncPred" event, to let V_i inform V_{i+1} of the former's current distance/velocity/acceleration, or to trigger the "StopSyncPred" event. For all these three cases, the corresponding V2V communications can be replaced by V_{i+1}'s local ranging sensors (see Assumption 9). Hence the correponding V2V communications failures are irrelevant.

In case the ranging sensors need *line-of-sight*, we have the following observations. All the target lane CAVs that should be in "Sync" at any time instance t must be following a unique target lane CAV $V_\imath$ ($\imath \in \{1, 2, \ldots, n\}$) that is in a coop-duration. This implies $V_\imath$ must be behind R at t, if R is after all on the target lane at t.

Therefore, it is impossible that R resides between two *speed synchronizing* target lane CAVs (i.e. the predecessor CAV is in a coop-duration, while the follower CAV is in "Sync"; or both are in "Sync") at t. Therefore, the line-of-sight between two speed synchronizing target lane CAVs is available at t.

4.5.2 Vehicle Body Shape

In Section 4.3.3, right after Def. 3, we discussed how to take into consideration of vehicle body length in same lane CTH-Δ^* safety design. Now let us discuss how to consider vehicle body shape in a mechanical lane change routine.

So far, we have abstracted the target lane as an axis (simplified as the "*target lane axis*" in this subsection), and CAVs as points (simplified as the "*CAV points*" in this subsection). Suppose the earliest time instance that the requesting CAV R point touches the target lane axis is t_{45}, and (on the target lane axis) the touch-point's X-coordinate is X_1. Meanwhile, suppose the earliest time instance that R's body touches the lane border between the current and target lane (see Fig. 4.1) is t_{46}. As we only allow limited number of mechanical lane change routines, there is an upper bound to $|t_{45} - t_{46}|$, denote it as $\Delta_{\mathsf{half\_lc}}$.

Theorem 2 ensures at t_{45}, the requesting CAV R point is at least $v_{\mathsf{low}}\Delta^*$ ahead of the follower CAV F point. Therefore $\forall t \in [t_{46}, t_{45}]$, the follower CAV F point's X-coordinate is no greater than $X_1 - v_{\mathsf{low}}\Delta^* - v_{\mathsf{low}}\delta$, where $\delta \stackrel{\mathsf{def}}{=} t_{45} - t$, hence $0 \leqslant \delta \leqslant \Delta_{\mathsf{half\_lc}}$. Meanwhile, as v_{lim} is also the maximum projected speed on the X-axis during any mechanical lane change routine, at t, any point on the CAV R's body has an X-coordinate of no less than $X_1 - v_{\mathsf{lim}}\delta - D_0$, where D_0 is the maximum vehicle body projection length on X-axis at any orientation.

Hence at t, any point on the CAV R's body is at least $X_1 - v_{\mathsf{lim}}\delta - D_0 - (X_1 - v_{\mathsf{low}}\Delta^* - v_{\mathsf{low}}\delta) \geqslant v_{\mathsf{low}}\Delta^* - D_0 - \widetilde{v}\Delta_{\mathsf{half\_lc}}$ ahead of the follower CAV F point along the X-axis.

4.6 Evaluation

Next, we carry out simulations to verify our proposed protocol, especially the CTH-Δ^* safety , liveness (automatic resetting) , and performance of the lane change protocol.

4.6.1 Simulation Configuration

Our simulator adopts the following configurations typical to a vehicle (see the classic textbook of [78]): $\Delta_{\mathsf{nonzeno}} = 0.1$s; $\Delta^* = 6$s; $v_{\mathsf{lim}} = 25$m/s; $v_{\mathsf{low}} = 20$m/s; lane-change, acceleration, and deceleration strategies are set as per [78], which imply $\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = 4.51$s, $d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) = 112.5573$m, $\delta_{\mathsf{lc}}(v_{\mathsf{low}}) = 4.72$s, $d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) = 94.1975$m, $\delta^{\ddagger}_{\mathsf{a}} = 4.65$s, $d^{\ddagger}_{\mathsf{a}} = 105.0914$m, $\delta^{\ddagger}_{\mathsf{d}} = 1.97$s, and $d^{\ddagger}_{\mathsf{d}} = 44.955$m. The above configurations further decide other parameters: $D_1 \sim D_3$ (see Eq. (4.11)(4.12)(4.13)), D_{Sync} (see Eq. (4.14)), and Δ_{reset} (see Eq. (4.20)).

The simulator also involves a wireless packet loss rate parameter P, which is set to 0.1 (i.e. 10%), 0.5 (i.e. 50%), and 0.9 (i.e. 90%) respectively to evaluate our protocol under mild, moderate, and severe wireless packet losses. We assume there are n target lane CAVs along a target lane segment [−5000m, 5000m]. We set n to 10, 20, 30 respectively to evaluate our protocol under mild, moderate, and heavy traffic load. For each given P and n value combination, we run 100 simulation trials. Each trial simulates 10 minutes (unless in some exception cases, see the notes in Section 4.6.4) of a lane change scenario. At the beginning of a simulation trial, the n target lane CAVs are generated along the target lane segment [−5000m, 5000m] as per pseudo uniform distribution, which takes into consideration of Assumption 10. Specifically, the pseudo code is as follows:

Step 1 Set $\mathcal{V}$ to $\varnothing$;

Step 2 If $|\mathcal{V}| \geqslant n$, terminate; otherwise

Step 2.1 Randomly pick a candidate target lane CAV initial location p on the target lane segment [−5000m, 5000m] as per uniform distribution;

Step 2.2 If p does not violate CTH-Δ^* safety rule with the elements already in $\mathcal{V}$, then add p into $\mathcal{V}$; otherwise, ignore p;

Step 2.3 Go back to **Step 2**.

The generated $\mathcal{V}$ is the initial locations for the target lane CAVs for the trial.

4.6.2 Comparison Baselines

As mentioned in Section 4.1 and 4.2, there are not many works covering CTH safety guarantee under arbitrary wireless packet losses. Perhaps the closest work is that of Wang et al. [97]'s PerLC and ConLC protocols for V2V CAVs (see Section 4.1), where ACKs from the target lane leader CAV L and follower CAV F are obligatory for starting the mechanical lane change routine. ConLC improves PerLC in terms of using multicast instead of one by one unicast, and brakes the requesting CAV R to change the relationship to L and F (include letting the current L and F pass, and change to a new pair of L and F).

However, CTH safety guarantee is not covered by Wang et al. [97]; and how to adapt the PerLC and ConLC protocols to guarantee CTH safety under arbitrary wireless packet losses are still open problems. Nevertheless, in order to provide comparison baselines to evaluate our proposed protocol, we propose our version of adaptations to the PerLC and ConLC protocols, respectively named PerLC+ and ConLC+. PerLC+ is basically a subset of our proposed protocol. The target lane CAV hybrid automata A_i ($i \in \{1, 2, \ldots, n\}$, see Fig. 4.4) remain unchanged. The requesting CAV R's hybrid automaton A_R (see Fig. 4.3) has the "Event2" and "Event4" disabled (returns to "Init" mode with τ reset to 0 in case of "Event2" and "Event4"). We call the resulted hybrid automaton A'_R. ConLC+ evolves PerLC+. Specifically, it evolves A'_R

to A''_R by replacing the "RequestTimeout1" event with the following routine:

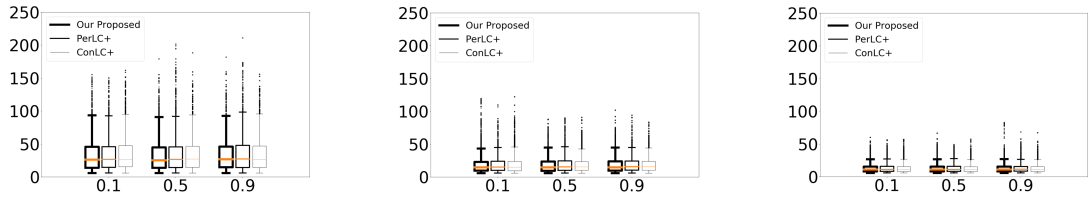
Step 1 Decelerate to v_{low};

Step 2 Wait for $\Delta^{\mathsf{Event1,max}}_{\mathsf{Coop}} + \Delta_{\mathsf{nonzeno}}$, then find the new leader CAV L and follower CAV F and measure the new $d_0(L,R)$ and $d_0(R,F)$;

Step 3 If $d_0(L,R) \geqslant v_{\mathsf{lim}}(\Delta^* - \delta_{\mathsf{lc}}(v_{\mathsf{low}})) + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}})$ and $d_0(R,F) \geqslant v_{\mathsf{lim}}(\Delta^* + \delta_{\mathsf{lc}}(v_{\mathsf{low}})) - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) + \widetilde{d}_{\mathsf{a}}(v_{\mathsf{lim}})$, change lane via mechanical lane change routine $\mathsf{lc}(v_{\mathsf{low}}, \tau)$ ($\tau \in [0, \delta_{\mathsf{lc}}(v_{\mathsf{low}})]$) and then accelerate to v_{lim}; else go to **Step 2**.

Following the same proof for Theorem 2 **Claim 1**, we can prove both PerLC+ and ConLC+ guarantee the CTH-Δ^* safety.

4.6.3 CTH-Δ^* Safety



(a) (b) (c)

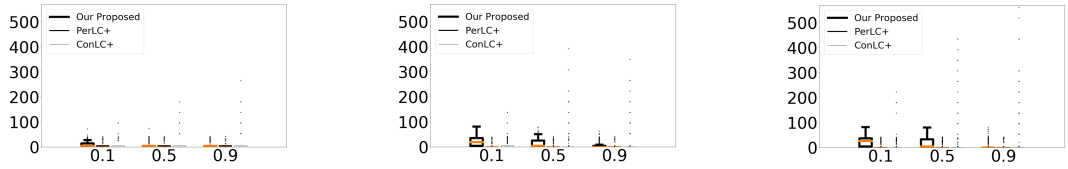
Figure 4.5: Statistics of time headways: (a) $n = 10$ (n is the number of target lane CAVs); (b) $n = 20$; (c) $n = 30$. X-axes: wireless packet loss rate (0.1, 0.5, and 0.9 respectively mean the wireless packet loss rate P is 10%, 50%, and 90%). Y-axes: time headway (unit: second)

We first verify Theorem 2 **Claim 1** on CTH-Δ^* safety guarantee, which answers Research Problem 1.

Fig. 4.5 plots the statistics of sampled time headways (relative to the respective immediate predecessor CAVs, see Def. 3) of all CAVs in all simulation trials (for each CAV simulated, its time headway is sampled every 0.8s).

According to Fig. 4.5, for all the three protocols (our proposed, PerLC+, ConLC+), under all three wireless packet loss rates ($P = 10\%$, 50%, 90%) and all three traffic loads ($n = 10$, 20, 30), the time headways are always no less than 6.0s. This means the CTH-Δ^* safety (remember Δ^* is set to 6.0s, see Section 4.6.1) always holds[3].

4.6.4 Liveness (Automatic Resetting)



(a) (b) (c)

Figure 4.6: Statistics of reset time costs: (a) $n = 10$ (n is the number of target lane CAVs); (b) $n = 20$; (c) $n = 30$. X-axes: wireless packet loss rate (0.1, 0.5, and 0.9 respectively mean the wireless packet loss rate P is 10%, 50%, and 90%). Y-axes: reset time costs (unit: second)

Next, we verify Theorem 2 **Claim 2** on liveness (i.e. automatic resetting), which answers Research Problem 2.

The claim regards two states of the holistic system to be stable: **Stable State 1** and **Stable State 2**. **Stable State 1** is the initial state of the holistic system; while **Stable State 2** is the desired end state of the holistic system, where the requesting CAV R successfully changes lane into the target lane, and together with all target lane CAVs regain the constant speed of v_{lim}.

Theorem 2 **Claim 2** basically says every time the holistic system becomes unstable (say at time $t_1 \in [t_0, +\infty)$), it will reset itself either back to **Stable State 1** or to **Stable State 2** within finite duration $(t_2 - t_1)$, where $t_2 \in (t_1, t_1 + \Delta_{\mathsf{reset}}]$, and Δ_{reset}

[3]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.

is a constant defined by Eq. (4.20). For ease of narration, we call $(t_2 - t_1)$ the "*reset time cost*."

Fig. 4.6(a), (b), and (c) respectively show that under light, mild, and heavy traffic loads, following our proposed protocol, the maximum reset time costs observed are respectively 72.2s, 82.7s and 82.2s (when packet loss rate $P = 10\%$); 73.7s, 79.1s, and 80.3s (when $P = 50\%$); 41.1s, 62.1s, and 81.6s (when $P = 90\%$). All are below $\Delta_{\mathsf{reset}} = 89.68$s as per our simulation configuration (see Section 4.6.1 and Eq (4.20)). This verifies Theorem 2 **Claim 2**.

Note for $P = 90\%$, our proposed protocol's reset time costs are statistically shorter than those of $P = 10\%$ and 50%. This is because under extremely poor packet loss rates, most resets return to the unwanted **Stable State 1**, which costs less time.

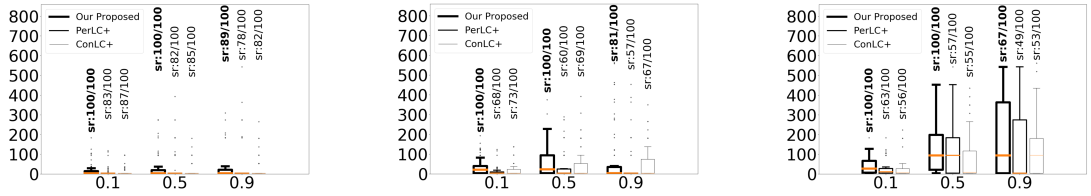
According to Fig. 4.6, the reset time cost statistics of PerLC+ exhibit similar features. This is expected, as PerLC+ fundamentally is a subset of our proposed protocol. ConLC+, on the other hand, can have much bigger reset time costs. This is because instead of returning to **Stable State 1**, it will keep trying until both the leader CAV and follower CAV are far away (in the worst case, this can loop till all the n target lane CAVs have passed away).

Also note that normally each trial of our simulation lasts 10 minutes (in the simulated world). But in case by the end of the 10th minute, the system is still waiting for a reset to take place, the simulation will go on until the reset takes place (i.e. reaching **Stable State 1** or **Stable State 2**).

4.6.5 Lane Change Success Rate and Time Cost

Finally, we evaluate the lane change success rates and time costs, which answers Research Problem 3.

A *lane change success rate* refers to the proportion of simulation trials (under given



(a) (b) (c)

Figure 4.7: Statistics of lane change *success rates* (sr) and time costs:(a) $n = 10$ (n is the number of target lane CAVs); (b) $n = 20$; (c) $n = 30$. X-axes: wireless packet loss rate (0.1, 0.5, and 0.9 respectively mean the wireless packet loss rate P is 10%, 50%, and 90%). Y-axes: lane change time costs (unit: second), with success rates (sr) labeled at the top of each data column.

protocol, n, and P) that end with **Stable State 2** (see Section 4.6.4 and Theorem 2 **Claim 2**); while a *lane change time cost* refers to the duration from the start of a simulation trial to the time instance when the holistic system reaches **Stable State 2** in the simulated world.

Fig. 4.7 show the lane change *success rates* (sr) and time costs of all three protocols (our proposed, PerLC+, ConLC+) under different traffic loads ($n = 10$, 20, and 30) and wireless packet loss rates ($P = 10\%$, 50%, 90%).

According to the figure, for each given traffic load n ($n \in \{10, 20, 30\}$), our proposed protocol's success rates outperform those of PerLC+ and ConLC+; and the difference becomes more significant when the wireless packet loss rate deteriorates. For example, under mild traffic load (see Fig. 4.7(b)), when P deteriorates from 10%, to 50%, to 90%, our proposed protocol's success rates are respectively 100%, 100%, and 81%; while PerLC+'s success rates are respectively 68%, 60%, 57%; and ConLC+'s success rates are respectively 73%, 69%, 67%.

For each given wireless packet loss rate P ($P \in \{10\%, 50\%, 90\%\}$), our proposed protocol's success rates also outperform those of PerLC+ and ConLC+; and the difference becomes more significant when the traffic load deteriorates. For example,

under mild wireless packet loss rate of $P = 50\%$, when n deteriorates from 10, to 20, to 30, our proposed protocol's success rates are respectively 100%, 100%, and 100%; while PerLC+'s success rates are respectively 82%, 60%, and 57%; and ConLC+'s success rates are respectively 85%, 69%, and 55%.

For those trials that succeed in the end, our proposed protocol's time costs are not always the shortest. However, this comparison is biased. As the success rate comparisons show, both PerLC+ and ConLC+ mainly succeed under benign driving conditions, hence lane change time costs are likely low; while our proposed protocol succeeds in both benign and harsh driving conditions, the latter usually cost more time to handle (e.g. require more complex driving coordinations). Encouragingly, even under such biased comparisons, the time cost statistics of our proposed protocol are still comparable to those of PerLC+ and ConLC+.

Also note that the lane change time cost is tightly related to the choice of the configuration parameters of the system (see Section 4.6.1). However, how to optimize the choice of the configuration parameters is not the focus of this chapter. Instead, our simulation adopts classic configurations for vehicles and drivings in our simulations (see Section 4.6.1 and [78]).

4.7 Chapter Summary

In this chapter, we propose a protocol to realize the safe lane change of automatic driving of CAVs in dedicated highway lanes. 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 the lane change time costs are satisfactory. Comparing to the other two lane change protocols, our proposed protocol can achieve significantly better success rates, particularly under adverse wireless packet loss or traffic conditions.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this dissertation, we discuss that individual CAVs can benefit from connectivity and autonomy. By exploiting the state-of-the-art hybrid automata modeling, we propose cooperative driving protocol to achieve more safe and efficient merging and lane changing. The main works can be summarized as follows:

First, 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.

Second, we propose a protocol to realize the safe lane-changing of automatic driving intelligent vehicles. We formally prove that the protocol can always guarantee the CTH safety and liveness, even under arbitrary wireless packet losses. These theoretical claims are validated by our simulations, which show that our proposed approach significantly improves the existing approach.

5.2 Future Work

As future work, we will do the following:

(**FW1**) Take into consideration of wireless transmission and communication delay. Throughout this dissertation, it has been assumed that information between vehicles is shared instantly. This assumption might not be applicable in practical implementation. Therefore, a time-delay factor can be incorporated in the future study while evaluating the performance.

(**FW2**) Allow more variations in the various physical parameters. In this dissertation, we consider special speed distribution, such as v_{lim}, v_{rm} and v_{low}. For the random speed distribution, the collision avoidance and cooperative control of vehicles need to be concerned with in the future to achieve the driving safety. Fortunately, the continuous nature of the physical world means that this dissertation's rigorous math abstractions are still meaningful in the physical world.

(**FW3**) A mixed traffic flow scene which contains connected and automated vehicles (CAVs) and conventional human-driving vehicles(HDVs) is to be established. In this dissertation, the vehicles are considered to be homogeneous, which means the vehicles have the same dynamics. Of course, the vehicles in the real-time scenario can have different properties. Thus, this research can be extended for heterogeneous vehicles in future studies.

(**FW4**) More sophisticated solutions involving parallel Cyber-Physical transactions and mutual exclusions are to be explored. In this thesis, the protocol design does not consider vehicles following the requesting CAV on its original lane. When changing lane, the speed of the requesting CAV may change on its original lane, how it may affect the following vehicles should be discussed.

(**FW5**) More simulations are to be added. In this thesis, simulation settings follow the assumptions made for analysis, performance of the protocol should be evaluated

in more practical scenarios. Moreover, it is desirable to add evaluation results in terms of loss of specific messages, besides evaluation under a general wireless packet loss rate parameter.

Appendix A

Proofs for Chapter 3

A.1 List of Symbols Used in Chapter 3

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

1. Δ_1, Δ_2, $\Delta_{\mathsf{BS}}^{\mathsf{min}}$, $\Delta_{\mathsf{nonzeno}}$, Δ_r, Δ^* are all configuration constants with positive values, see Eq. (3.4), (3.5), Section 3.4.2-"Base Station BS protocol behaviors"-(2)(5), Eq. (3.2), Definition 3 respectively.

2. $\Delta_{\mathsf{coop}}^{\mathsf{max}}$, see Theorem 1 (**c2**).

3. $\delta_{\mathsf{a}}(v_1, v_2)$ is the total time needed to accelerate from v_1 to v_2 (where $0 \leqslant v_1 \leqslant v_2$), see Section 3.3.1.

4. $\hat{\delta}_{\mathsf{coop}}$ is a runtime variable local to A_{BS}. It is used to estimate the time distance of the current coop CAV to reach $\vec{p}_{\mathsf{merge}}$.

5. $\delta_{\mathsf{d}}(v_2, v_1)$ is the total time needed to decelerate from v_2 to v_1 (where $v_2 \geqslant v_1 \geqslant 0$), see Section 3.3.1.

6. δ_{defer} is a runtime variable created by A_{BS} at "Event2" (note Ineq. (3.7) ensures $\delta_{\mathsf{defer}} > 0$), 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 Fig. 3.4). It is the data payload parameter received via the "Start" packet. In case the packet is sent by BS via "Event1" (see Fig. 3.3), $\sigma_{\mathsf{defer}} = 0$. In case the packet is sent by BS via the "GotAcceptSlowDown" event, $\sigma_{\mathsf{defer}} = \delta_{\mathsf{defer}}$. 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 $[0, \Delta_{\mathsf{BS}}^{\mathsf{min}}]$ (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 Eq. (3.8), Ineq. (3.1) respectively.

10. acc is the predefined acceleration strategy, see Section 3.3.1.

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. $d_{\mathsf{a}}(v_1, v_2)$ is the total distance needed to accelerate from v_1 to v_2 (where $0 \leqslant v_1 \leqslant v_2$), see Section 3.3.1.

13. $d_{\mathsf{d}}(v_2, v_1)$ is the total distance needed to decelerate from v_2 to v_1 (where $v_2 \geqslant v_1 \geqslant 0$), see Section 3.3.1.

14. dec is the predefined deceleration strategy, see Section 3.3.1.

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 local 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 3.3.2 and Ineq. (3.3) for more information.

A.2 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. $(\star)$

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 Fig. 3.5), 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 $(t_1, t_2]$. $(\star\star)$

This can be proved iteratively.

For $h_{\imath+1}$, throughout $(t_1, t_1+\delta_{\mathsf{defer}}]$, both $h_\imath$ and $h_{\imath+1}$ remain at v_{lim}; throughout $(t_1+\delta_{\mathsf{defer}}, t_2]$, $h_{\imath+1}$ synchronizes its speed with $h_\imath$ according to mode "Sync" (see Fig. 3.5); Hence throughout $(t_1, t_2]$, $(h_\imath, h_{\imath+1})$ remains CTH-Δ^* safe.

Same reasoning can be applied to $h_{\imath+2}$, $h_{\imath+3}$, ..., 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) $\forall t \in (t_1, t_1+\delta_{\mathsf{defer}}]$, 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) $\forall t \in (t_1+\delta_{\mathsf{defer}}, t_1+\delta_{\mathsf{defer}}+\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}})]$, h_{k-1} synchronizes its speed with h_{k-2}, h_{k-3}, ..., $h_\imath$, hence keeps decelerating from v_{lim} to v_{rm}; while h_k remains in "Init" (as $|\vec{p}(h_{k-1}, t_1+\delta_{\mathsf{defer}}) - \vec{p}(h_k, t_1+\delta_{\mathsf{defer}})| > D_1$, event "StartSyncPred" will not happen at $t_1+\delta_{\mathsf{defer}}$ to h_k, and during $(t_1+\delta_{\mathsf{defer}}, t_1+\delta_{\mathsf{defer}}+\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}})]$ the event will neither happen to h_k as h_{k-1}'s speed is below v_{lim}). Meanwhile, for the entire deceleration process, $|\vec{p}(h_{k-1}, t) - \vec{p}(h_k, t)| > D_1 + d_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) - v_{\mathsf{lim}}\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) > v_{\mathsf{lim}}\Delta^*$ (see D_1's definition in Eq. (3.8)). This means (h_{k-1}, h_k) is CTH-Δ^* safe at t.

(iii) $\forall t \in (t_1+\delta_{\mathsf{defer}}+\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}), t_1+\delta_{\mathsf{defer}}+\Delta_r+\Delta^*]$ (note according to (3.6), $\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) < \Delta_r + \Delta^*$), h_{k-1} remains synchronizing its speed with h_{k-2}, h_{k-3}, ..., $h_\imath$, hence keeps the speed of v_{rm}, while h_k remains in mode "Init" (as $|\dot{\vec{p}}(h_{k-1}, t)| < v_{\mathsf{lim}}$, event "StartSyncPred" will not happen to h_k). Meanwhile for this entire constant speed process, $|\vec{p}(h_{k-1}, t) - \vec{p}(h_k, t)| > D_1 + d_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) - v_{\mathsf{lim}}\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) - (v_{\mathsf{lim}} - v_{\mathsf{rm}})(\Delta_r + \Delta^* - \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}})) > v_{\mathsf{lim}}\Delta^*$ (see D_1's definition in Eq. (3.8)). This means (h_{k-1}, h_k) is CTH-Δ^* safe at t.

(iv) $\forall t \in (t_1+\delta_{\mathsf{defer}}+\Delta_r+\Delta^*, t_1+\delta_{\mathsf{defer}}+\Delta_r+\Delta^*+\delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})]$ (note $t_1+\delta_{\mathsf{defer}}+\Delta_r+$

$\Delta^* + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$ 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}, ..., $h_\imath$, 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, $|\vec{p}(h_{k-1}, t) - \vec{p}(h_k, t)| > D_1 + d_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) - v_{\mathsf{lim}}\delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) - (v_{\mathsf{lim}} - v_{\mathsf{rm}})(\Delta_r + \Delta^* - \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}})) - (v_{\mathsf{lim}}\delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}}) - d_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})) \geqslant v_{\mathsf{lim}}\Delta^*$ (see Δ_1 and D_1's definition in Eq. (3.4)(3.8)). This means (h_{k-1}, h_k) is CTH-Δ^* safe at t.

Combining (i)$\sim$(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**$\sim$**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. □

A.3 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. □

A.4 Proof of Lemma 6

Proof: According to A_r (see Fig. 3.4), if r reaches $\vec{p}_{\mathsf{merge}}$ at t_1, then it must have received the "ActualStart" event at $t_2 \stackrel{\mathsf{def}}{=} t_1 - \Delta_r$, 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 Fig. 3.3) at t_2.

Then first, this means the most recent "Event2", the only event that can trigger a coop-duration, (if it ever happened) must be before $t_2 - \Delta_{\mathsf{BS}}^{\mathsf{min}}$ (note there can be no more "Event2" after t_2, as r has received "Start"). Due to (**c2**), $\Delta_{\mathsf{BS}}^{\mathsf{min}} > \Delta_{\mathsf{coop}}^{\mathsf{max}} > \Delta_r$, 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" 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 $(-\infty, \vec{p}_{\mathsf{merge}}]$. This means at t_2^-, h_n is at highway lane segment of $(\vec{p}_{\mathsf{merge}}, +\infty)$. So by t_1, h_n is at least $v_{\mathsf{rm}}\Delta_r \geqslant v_{\mathsf{lim}}\Delta^*$ (due to (**c3**)) ahead of r. Due to Corollary 1, this implies all highway

CAVs are at least $v_{\mathsf{rm}}\Delta_r \geqslant v_{\mathsf{lim}}\Delta^*$ 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 $(-\infty, \vec{p}_{\mathsf{merge}}]$. Suppose the one closest to $\vec{p}_{\mathsf{merge}}$ is h_i ($i \in \{1, 2, \ldots, n\}$). Then because BS sends "Start" packet via "Event1", we know

$$\hat{\delta}_{\mathsf{coop}} = |\vec{p}_{\mathsf{merge}} - \vec{p}(h_i, t_2^-)|/v_{\mathsf{lim}} \geqslant \Delta_r + \Delta^* + \Delta_1 \tag{A.1}$$

Meanwhile, as per A_r, r shall reach $\vec{p}_{\mathsf{critical}}$ (the location where r first reaches speed v_{lim}, see Fig. 3.1) at $t_3 \stackrel{\mathsf{def}}{=} t_1 + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$, and $|\vec{p}_{\mathsf{critical}} - \vec{p}_{\mathsf{merge}}| = d_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$.

Due to (†), h_i reaches $\vec{p}_{\mathsf{merge}}$ at $t_2 + \hat{\delta}_{\mathsf{coop}} \geqslant t_2 + \Delta_r + \Delta^* + \Delta_1$ (due to (A.1)) $= t_1 + \Delta^* + \Delta_1$. This means (r, h_i) is CTH-Δ^* safe at t_1.

Furthermore, h_i reaches $\vec{p}_{\mathsf{critical}}$ at $t_2 + \hat{\delta}_{\mathsf{coop}} + d_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})/v_{\mathsf{lim}} \geqslant t_2 + \Delta_r + \Delta^* + \Delta_1 + d_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})/v_{\mathsf{lim}} = t_1 + \Delta^* + \delta_{\mathsf{a}}(v_{\mathsf{rm}}, v_{\mathsf{lim}})$ (see the definition of Δ_1 in (3.4)) $= t_3 + \Delta^*$. This means (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 $(\vec{p}_{\mathsf{merge}}, +\infty)$ 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 hence 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" in A_{BS} (see Fig. 3.3) at $t_2 - \delta_{\mathsf{defer}}$. Immediately before it, BS must have sent ("SlowDown", BS, h_{coop}, δ_{defer}) packet to h_{coop} at $t_2 - \delta_{\mathsf{defer}}$ and received h_{coop}'s "AcceptSlowDown" packet, where

$\mathsf{coop} \in \{1, 2, \ldots, n\}$. Without loss of generality, suppose $\mathsf{coop} = i$.

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

During $(t_2, t_2 + \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}})]$, h_i decelerates from v_{lim} to v_{rm} (note due to (3.6), $t_2 + \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) < t_2 + \Delta_r = t_1$) and drives $v_{\mathsf{lim}}\delta_{\mathsf{defer}} + d_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}) = \hat{\delta}_{\mathsf{coop}} v_{\mathsf{lim}} - v_{\mathsf{rm}}(\Delta_r + \Delta^* - \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}))$ distance since $t_2 - \delta_{\mathsf{defer}}$.

During $(t_2 + \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}), t_2 + \Delta_r + \Delta^*]$, h_i remains at v_{rm}. Note $t_1 = t_2 + \Delta_r \in (t_2 + \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}), t_2 + \Delta_r + \Delta^*)$. This means, at t_1, h_i is in the "ConstLowSpeed" mode, maintaining the speed of v_{rm}. Therefore, at t_1, $\vec{p}(r, t_1) - \vec{p}(h_i, t_1) = \vec{p}_{\mathsf{merge}} - \vec{p}(h_i, t_1) = \hat{\delta}_{\mathsf{coop}} v_{\mathsf{lim}} - (\hat{\delta}_{\mathsf{coop}} v_{\mathsf{lim}} - v_{\mathsf{rm}}(\Delta_r + \Delta^* - \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}})) - v_{\mathsf{rm}}(t_1 - t_2 - \delta_{\mathsf{d}}(v_{\mathsf{lim}}, v_{\mathsf{rm}}))) = v_{\mathsf{rm}}\Delta^* > 0$. This means, at t_1, r is ahead of h_i by $v_{\mathsf{rm}}\Delta^*$; 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 $\vec{p}_{\mathsf{merge}}$. 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 $\vec{p}_{\mathsf{merge}}$ 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 $[t_1 + \Delta^*, +\infty)$, by applying integration to calculate the distance between r and h_i, and noticing i) the time-location curves are Δ^* away shifted along the time axis and ii) the acceleration process is monotonic, we can see (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 $t_2 - \delta_{\mathsf{defer}}$, when BS sends "SlowDown"

packet to h_i, h_{i-1} must be on segment $(\vec{p}_{\mathsf{merge}}, +\infty)$. Also, notice as coop-durations cannot overlap nor connect, and BS sends no more "SlowDown" packet after $t_2 - \delta_{\mathsf{defer}}$. This means throughout $[t_2 - \delta_{\mathsf{defer}}, +\infty)$, h_{i-1} is in "Init". Then use 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 $\forall j \in \{1,\ 2,\ \ldots,\ i-1\}$, (h_j, r) is CTH-Δ^* safe throughout $[t_1, +\infty)$.

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

Appendix B

Proofs for Chapter 4

B.1 Proof of Lemma 10

First, as $(t_{14}, t_{15}]$ is the first ever coop-duration, due to Assumption 10, V_1, V_2, …, V_n all reside in hybrid automata mode "Init" throughout $[t_0, t_{14}]$, hence $\forall t \in [t_0, t_{14}]$, (V_i, V_{i+1}) $(i = 1, 2, \ldots, n-1)$ is CTH-Δ^* safe. $(\star)$

Suppose the coop-duration $(t_{14}, t_{15}]$ belongs to $V_\imath$ $(\imath \in \{1, 2, \ldots, n\})$. Then we have the following cases.

Case 1: First we discuss V_j, where $j < \imath$. As coop-durations cannot overlap nor connect, V_j $(\forall j \in \{1, 2, \ldots, \imath - 1\})$ must remain in mode "Init" throughout $(t_{14}, t_{15}]$. That is, for any V_j and V_{j+1} $(j \in \{1, 2, \ldots, \imath - 2\})$, throughout $(t_{14}, t_{15}]$, both keep the speed of v_{lim}. Hence (V_j, V_{j+1}) is CTH-Δ^* safe throughout $(t_{14}, t_{15}]$. For $V_{\imath-1}$ and $V_\imath$, as $V_{\imath-1}$ keeps the maximum allowed speed (v_{lim}) throughout $(t_{14}, t_{15}]$, $(V_{\imath-1}, V_\imath)$ is hence CTH-Δ^* safe throughout $(t_{14}, t_{15}]$.

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

Case 2.1: Suppose at t_{14}, $V_k (k > \imath)$ is the first CAV after $V_\imath$, s.t. $|\vec{p}(V_{k-1}, t_{14}) - \vec{p}(V_k, t_{14})| \geq D_{\mathsf{Sync}}$. Then we have the following cases.

Case 2.1.1: $\forall j \in \{\imath+1,\ \imath+2,\ \ldots,\ k-1\}$, we have (V_{j-1}, V_j) is CTH-Δ^* safe throughout $(t_{14}, t_{15}]$. $(\star\star)$

This can be proved iteratively.

For $V_{\imath+1}$, it synchronizes its speed with $V_\imath$ according to mode "Sync" (see Fig. 4.4); Hence throughout $(t_{14}, t_{15}]$, $(V_\imath, V_{\imath+1})$ remains CTH-Δ^* safe.

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

Case 2.1.2: For V_k, we have (V_{k-1}, V_k) is CTH-Δ^* safe throughout $(t_{14}, t_{15}]$. $(\star\star\star)$

This can be proved via two cases.

Case 2.1.2.1: $V_\imath$ enters the coop-duration via "Event1" in Fig. 4.4. Then the coop-duration of $(t_{14}, t_{15}]$ consecutively experiences and only experiences the "Decelerating1," "ConstLowSpeed1," and "Accelerating" modes respectively for $\delta^\ddagger_{\mathsf{d}}$, $\delta^{\mathsf{RF1}}_{\mathsf{low}}$, and $\delta^\ddagger_{\mathsf{a}}$ durations. Throughout this coop-duration, V_{k-1} synchronizes its speed with V_{k-2}, V_{k-3}, ..., $V_\imath$ and V_k remains in its "Init" mode. Hence the V_{k-1} and V_k distance shrinks from t_{14} to t_{15}. The scenario is illustrated by Fig. B.1. Therefore $\forall t \in (t_{14}, t_{15}]$,

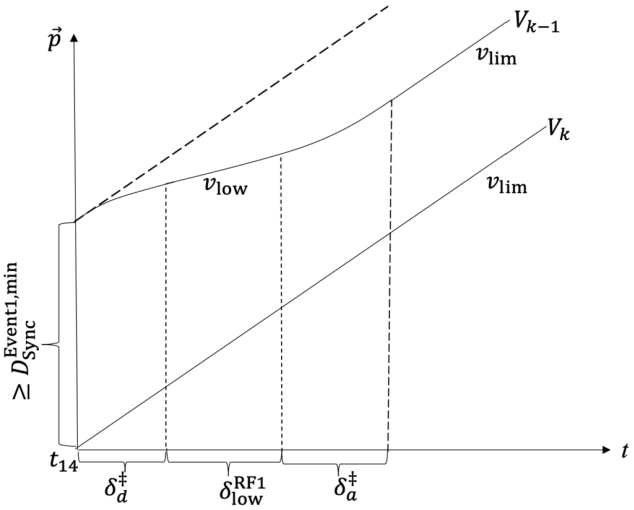


Figure B.1: time-position trajectory diagram of V_{k-1} and V_k

$$|\vec{p}(V_{k-1},t)-\vec{p}(V_k,t)| \geqslant D_{\mathsf{Sync}}^{\mathsf{Event1,min}} - \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) - \widetilde{v}(\delta_{\mathsf{low}}^{\mathsf{RF1}} + \delta_{\mathsf{a}}^{\ddagger})$$
$$= v_{\mathsf{lim}}\Delta^* + d_0(R,F) \geqslant v_{\mathsf{lim}}\Delta^*.$$

This means (V_{k-1}, V_k) is CTH-Δ^* safe at t.

Case 2.1.2.2: V_i enters the coop-duration via "Event2" in Fig. 4.4.

The scenario is illustrated by Fig. B.2. Apply similar reasoning as **Case 2.1.2.1**, we

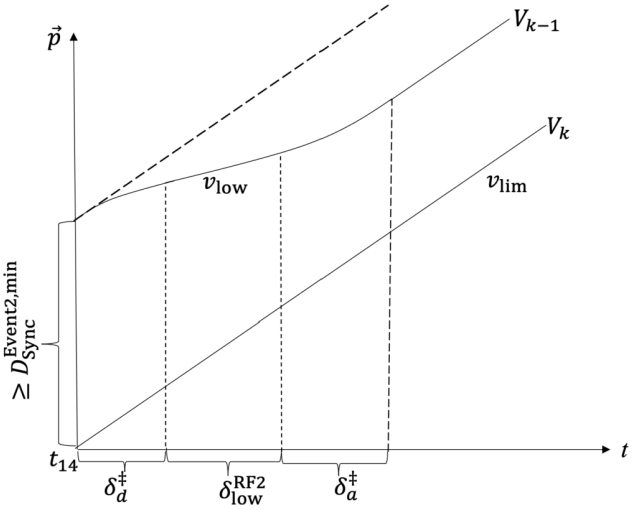


Figure B.2: time-position trajectory diagram of V_{k-1} and V_k

have $\forall t \in (t_{14}, t_{15}]$,

$$|\vec{p}(V_{k-1},t)-\vec{p}(V_k,t)| \geqslant D_{\mathsf{Sync}}^{\mathsf{Event2,min}} - \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) - \widetilde{v}(\delta_{\mathsf{low}}^{\mathsf{RF2}} + \delta_{\mathsf{a}}^{\ddagger})$$
$$= v_{\mathsf{lim}}\Delta^* + d_0(R,F) \geqslant v_{\mathsf{lim}}\Delta^*.$$

This means (V_{k-1}, V_k) is CTH-Δ^* safe at t.

Combining **Case 2.1.2.1** and **Case 2.1.2.2**, $(\star\star\star)$ sustains.

Case 2.1.3: For V_j $(j = k+1, k+2, \ldots, n)$, as V_k remains in mode "Init" throughout $(t_{14}, t_{15}]$, V_{k+1} also remains in mode "Init" throughout $(t_{14}, t_{15}]$, so on and so forth. Hence the CTH-Δ^* safety of (V_{j-1}, V_j) since t_0 is unchanged.

Combining **Case 2.1.1**$\sim$**Case 2.1.3**, we see in **Case 2.1**, $\forall j \in \{\imath+1, \imath+2, \ldots, n\}$, $\forall t \in (t_{14}, t_{15}]$, (V_{j-1}, V_j) is CTH-Δ^* safe at t.

Case 2.2: Suppose at t_{14}, $\forall j \in \{\imath+1, \imath+2, \ldots, n\}$, $|\vec{p}(V_{j-1}, t_{14}) - \vec{p}(V_j, t_{14})| < D_{\mathsf{Sync}}$, then follow the same proving method for **Case 2.1.1**, we can prove $\forall t \in (t_{14}, t_{15}]$, (V_{j-1}, V_j) is CTH-Δ^* safe at t.

Combining **Case 2.1** and **Case 2.2**, we see in **Case 2**, $\forall j \in \{\imath+1, \imath+2, \ldots, n\}$, $\forall t \in (t_{14}, t_{15}]$, (V_{j-1}, V_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.

B.2 Proof of Lemma 11

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

Case 2: If infinite coop-duration(s) happen. Suppose $(t_{28}, t_{29}] \subseteq [t_0, +\infty)$ is the first coop-duration ever happens. Then due to Lemma 10, this lemma trivially sustains for the duration $[t_0, t_{29}]$. At t_{29}^+, due to Lemma 8 and Lemma 9, all CAVs have returned to node "Init", and $\forall i \in \{1, 2, \ldots, n-1\}$, (V_i, V_{i+1}) is CTH-Δ^* safe at t_{29}^+. Thus we can regard t_{29} as the new t_0, and apply the same technique to prove Lemma 10, 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.

B.3 Proof of Lemma 12

Proof: According to A_R (see Fig. 4.3), if R finishes the mechanical lane change routine at t_{16}, then it must have started the routine at $t_{30} \stackrel{\mathsf{def}}{=} t_{16} - \delta_{\mathsf{lc}}(v_{\mathsf{lim}})$ (if the routine is triggered via R's "Event1" or event "GotLaneChangeAcpt"); or at $t_{31} \stackrel{\mathsf{def}}{=} t_{16} - \delta_{\mathsf{lc}}(v_{\mathsf{low}})$ (if the routine is triggered via R's "Event2" or event "GotDecelerate") and then accelerates (from v_{low}) to v_{lim} at $t_{32} \stackrel{\mathsf{def}}{=} t_{16} + \delta^{\ddagger}_{\mathsf{a}}$.

Next, we shall discuss these four cases.

Case 1: If the mechanical lane change routine is triggered via R's "Event1," then the event must have happened at t_{30}.

Due to (**c6**) and $\Delta_{\mathsf{reset}} > \Delta^{\mathsf{max}}_{\mathsf{Coop}}$ (see Eq. (4.20)), we can prove with contradiction that no target lane CAV V_i ($i \in \{1, 2, \ldots, n\}$) is in a coop-duration at t_{30}. [1] Due to Lemma 9, this means all target lane CAVs are in mode "Init" at t_{30}.

As R's "Event1" does not incur sending of the "LaneChangeReq" wireless packet, all target lane CAVs will remain in mode "Init" throughout $[t_{30}, +\infty)$. Suppose at t_{30}, the recognized follower CAV $F = V_i$ ($\imath \in \{1, 2, \ldots, n\}$). Then at t_{16}, $\vec{p}(R, t_{16}) - \vec{p}(V_i, t_{16}) = d_0(R, F) + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) \geqslant D_3 + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = v_{\mathsf{lim}}\Delta^* + \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) - \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) = v_{\mathsf{lim}}\Delta^*$. This means (R, V_i) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{\imath, \imath+1, \ldots, n\}$, (R, V_i) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Meanwhile, if $\imath > 1$, then the leader CAV L exists and $L = V_{\imath-1}$. At t_{16}, $\vec{p}(V_{\imath-1}, t_{16}) - \vec{p}(R, t_{16}) = d_0(R, L) + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) \geqslant D_1 + \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) = v_{\mathsf{lim}}\Delta^* - \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) +$

[1] Specifically, suppose V_i is in coop-duration $(t_{37}, t_{38}]$, s.t. $t_{30} \in (t_{37}, t_{38}]$. Then $t_{37} < t_{30}$, hence $|t_{37} - t_{30}| \leqslant \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. But in order to trigger the coop-duration at t_{37}, R must have fired "Event3" (or "Event4") at t_{37} (due to (**c6**)). Meanwhile, R's "Event1" is fired at t_{30}. As "Event3" (or "Event4") and R's "Event1" must be separated by at least $\Delta_{\mathsf{reset}} + \Delta_{\mathsf{nonzeno}}$, this means $|t_{37} - t_{30}| > \Delta_{\mathsf{reset}}$. This implies $\Delta^{\mathsf{max}}_{\mathsf{Coop}} > \Delta_{\mathsf{reset}}$, which contradicts Eq. (4.20).

$\widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) = v_{\mathsf{lim}}\Delta^*$. This means $(V_{\imath-1}, R)$ is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{1, 2, \ldots, \imath-1\}$, (V_i, R) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Case 2: If the mechanical lane change routine is triggered via R's "Event2," then the event must have happened at $t_{33} \stackrel{\mathsf{def}}{=} t_{31} - \delta^{\mathsf{LR}}_{\mathsf{low}} - \delta^{\ddagger}_{\mathsf{d}}$.

Similar to **Case 1**, we can prove all target lane CAVs are in mode "Init" at t_{33}. [2] The entire lane change process is illustrated by Fig. B.3.

As R's "Event2" does not incur sending of the "LaneChangeReq" wireless packet, all target lane CAVs will remain in mode "Init" throughout $[t_{33}, +\infty)$. Suppose at t_{33}, the recognized follower CAV $F = V_\imath$ ($\imath \in \{1, 2, \ldots, n\}$). Then at t_{16}, $\vec{p}(R, t_{16}) - \vec{p}(V_\imath, t_{16}) = d_0(R, F) + (d^{\ddagger}_{\mathsf{d}} + v_{\mathsf{low}}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}})) - v_{\mathsf{lim}}(\delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{LR}}_{\mathsf{low}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}})) = d_0(R, F) + d^{\ddagger}_{\mathsf{d}} - \widetilde{v}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) - v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{d}} - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) \geqslant D_2 + d^{\ddagger}_{\mathsf{d}} - \widetilde{v}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) - v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{d}} - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) = v_{\mathsf{lim}}\Delta^* + \widetilde{v}\delta^{\ddagger}_{\mathsf{a}} + d_0(L, R) > v_{\mathsf{lim}}\Delta^*$.

After t_{16}, R accelerates from v_{low} to v_{lim}, hence $\forall t \in [t_{16}, t_{32}]$, $\vec{p}(R, t) - \vec{p}(V_\imath, t) \geqslant \vec{p}(R, t_{16}) - \vec{p}(V_\imath, t_{16}) - \widetilde{v}\delta^{\ddagger}_{\mathsf{a}} \geqslant v_{\mathsf{lim}}\Delta^* + d_0(L, R) \geqslant v_{\mathsf{lim}}\Delta^*$. That is, $(R, V_\imath)$ is CTH-Δ^* safe at t. This will sustain after t_{32}, as R reaches v_{lim}. Hence, $(R, V_\imath)$ is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{\imath, \imath+1, \ldots, n\}$, (R, V_i) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Meanwhile, if $\imath > 1$, then the leader CAV L exists and $L = V_{\imath-1}$. At t_{16}, $\vec{p}(V_{\imath-1}, t_{16}) -$

[2] Specifically, suppose $\exists$ target lane CAV V_i ($i \in \{1, 2, \ldots, n\}$) in a coop-duration $(t_{39}, t_{40}]$, s.t. $t_{33} \in (t_{39}, t_{40}]$. Then $t_{39} < t_{33}$, hence $|t_{39} - t_{33}| \leqslant \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. But in order to trigger the coop-duration at t_{39}, R must have fired "Event3" (or "Event4") at t_{39} (due to (**c6**)). Meanwhile, R's "Event2" is fired at t_{33}. As "Event3" (or "Event4") and R's "Event2" must be separated by at least $\Delta_{\mathsf{reset}} + \Delta_{\mathsf{nonzeno}}$, this means $|t_{39} - t_{33}| > \Delta_{\mathsf{reset}}$. This implies $\Delta^{\mathsf{max}}_{\mathsf{Coop}} > \Delta_{\mathsf{reset}}$, which contradicts Eq. (4.20). Therefore, at t_{33}, no target lane CAV is in a coop-duration. Due to Lemma 9, this means all target lane CAVs are in mode "Init" at t_{33}.

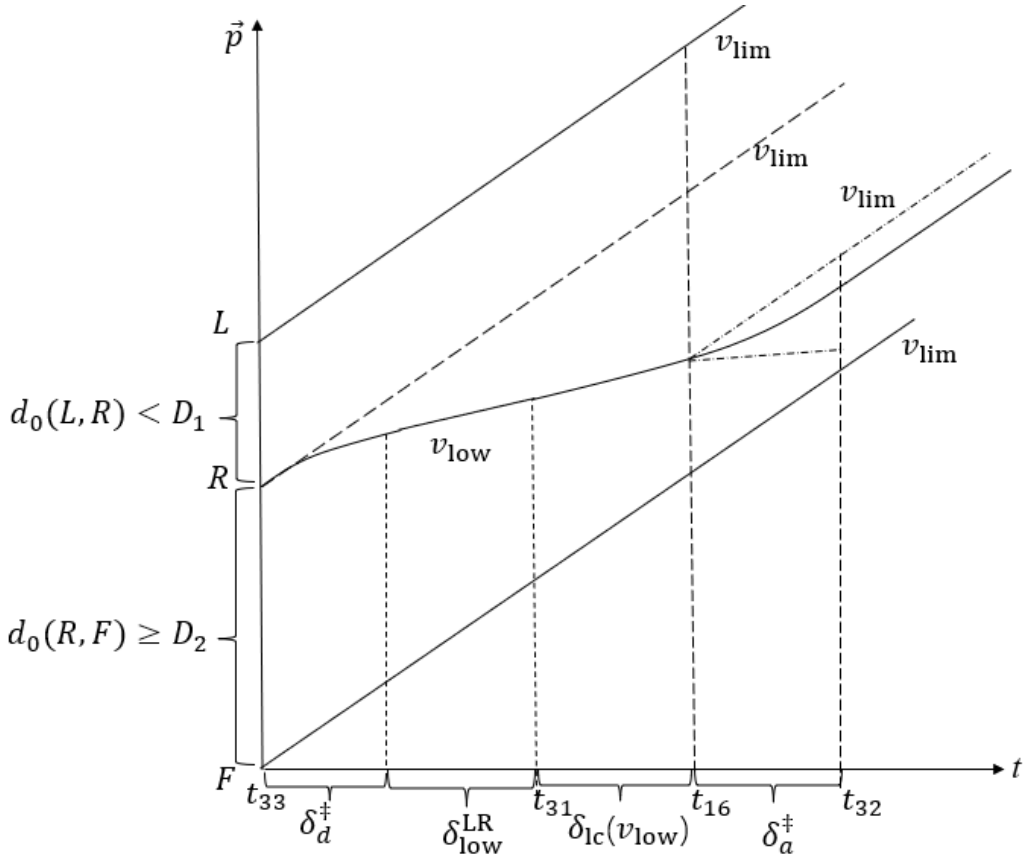


Figure B.3: The entire lane change process: $d_0(L,R) < D_1$ and $d_0(R,F) \geq D_2$

$\vec{p}(R,t_{16}) = d_0(L,R) + v_{\mathsf{lim}}(\delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{LR}}_{\mathsf{low}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}})) - (d^{\ddagger}_{\mathsf{d}} + v_{\mathsf{low}}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}})) = d_0(L,R) + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{d}} + \widetilde{v}\delta^{\mathsf{LR}}_{\mathsf{low}} + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\ddagger}_{\mathsf{d}} - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) = v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{d}} + v_{\mathsf{lim}}\Delta^* + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\ddagger}_{\mathsf{d}} - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) = v_{\mathsf{lim}}\Delta^* + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \geqslant v_{\mathsf{lim}}\Delta^* + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{low}}) \geqslant v_{\mathsf{lim}}\Delta^*$. After t_{16}, R accelerates from v_{low} to v_{lim}, then $\forall t \in [t_{16}, t_{32}]$, $\vec{p}(V_{\imath-1},t) - \vec{p}(R,t) \geqslant \vec{p}(V_{\imath-1},t_{16}) - \vec{p}(R,t_{16}) + v_{\mathsf{lim}}(t-t_{16}) - v_{\mathsf{lim}}(t-t_{16}) \geqslant v_{\mathsf{lim}}\Delta^*$. Hence, $(V_{\imath-1}, R)$ is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{1, 2, \ldots, \imath-1\}$, (V_i, R) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Case 3: If the mechanical lane change routine is triggered via event "GotLaneChangeAcpt," then the event must have happened at $t_{34} \overset{\mathsf{def}}{=} t_{30} - \delta^{\mathsf{lc}}_{\mathsf{defer}}$.

Similar to **Case 1**, we can prove all target lane CAVs are in mode "Init" at t_{34}. [3]

Suppose at t_{34}, the recognized follower CAV $F = V_{\imath}$ ($\imath \in \{1, 2, \ldots, n\}$). After t_{34}, the mode change time instances of R and F have the following relationships: i) $\delta^{\ddagger}_{\mathsf{d}} < \delta^{\mathsf{lc}}_{\mathsf{defer}} = \frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta^{\ddagger}_{\mathsf{d}} < \delta^{\mathsf{RF1}}_{\mathsf{low}} = \frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta_{\mathsf{lc}}(v_{\mathsf{lim}})$ (due to (**c3**)); ii) $\delta^{\mathsf{lc}}_{\mathsf{defer}} + \delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = \frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta^{\ddagger}_{\mathsf{d}} + \delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = \delta^{\mathsf{RF1}}_{\mathsf{low}} + \delta^{\ddagger}_{\mathsf{d}}$, particularly, this means

[3] Specifically, suppose $\exists$ target lane CAV V_i ($V_i \neq F$, the follower of the triggering "GotLaneChangeAcpt" event) in a coop-duration $(t_{41}, t_{42}]$, s.t. $t_{34} \in (t_{41}, t_{42}]$. Then $t_{41} < t_{34}$, hence $|t_{41} - t_{34}| \leqslant \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. But in order to trigger the coop-duration at t_{41}, R must have fired "Event3" (or "Event4") at t_{41} (due to (**c6**)) to $V_i \neq F$. Meanwhile, "Event3" is fired at t_{34} to F (due to (**c6**)). As "Event3" (or "Event4") to V_i and "Event3" to F must be two different events, hence must be separated by at least $\Delta_{\mathsf{reset}} + \Delta_{\mathsf{nonzeno}}$, this means $|t_{41} - t_{34}| > \Delta_{\mathsf{reset}}$. This implies $\Delta^{\mathsf{max}}_{\mathsf{Coop}} > \Delta_{\mathsf{reset}}$, which contradicts Eq. (4.20). On the other hand, suppose F is in a coop-duration $(t_{41}, t_{42}]$, s.t. $t_{34} \in (t_{41}, t_{42}]$. Then $t_{41} < t_{34}$, hence $|t_{41} - t_{34}| \leqslant \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. But in order to trigger the coop-duration at t_{41} for F, R must have fired "Event3" (or "Event4") at t_{41} (due to (**c6**)) to F. Meanwhile, "Event3" is fired at t_{34} to F (due to (**c6**)). As $t_{41} < t_{34}$, "Event3" (or "Event4") to F at t_{41} and "Event3" to F at t_{34} must be two different events, hence must be separated by at least $\Delta_{\mathsf{reset}} + \Delta_{\mathsf{nonzeno}}$, this means $|t_{41} - t_{34}| > \Delta_{\mathsf{reset}}$. This implies $\Delta^{\mathsf{max}}_{\mathsf{Coop}} > \Delta_{\mathsf{reset}}$, which contradicts Eq. (4.20). Therefore, at t_{34}, no target lane CAV is in a coop-duration. Due to Lemma 9, this means all target lane CAVs are in mode "Init" at t_{34}.

at t_{16}, F starts to accelerate.

Therefore, the entire lane change process is illustrated by Fig. B.4

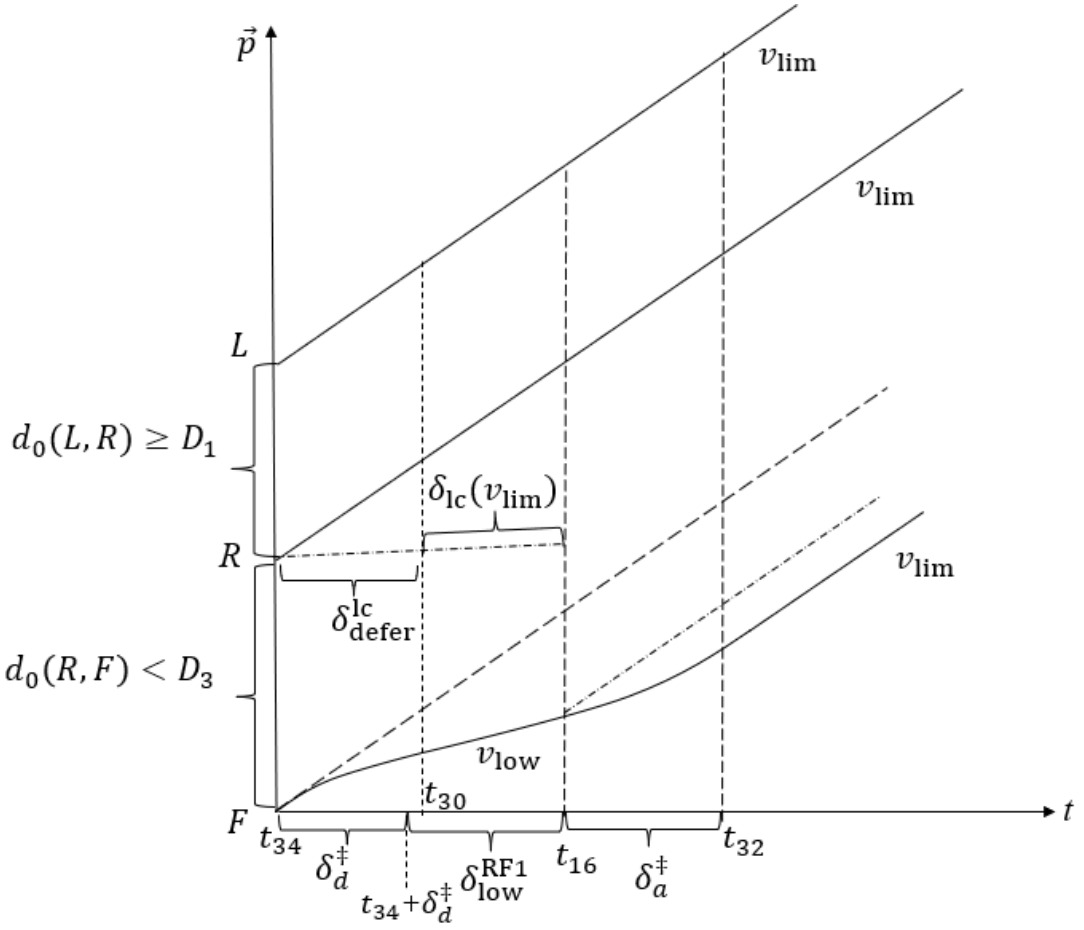


Figure B.4: The entire lane change process: $d_0(L,R) \geq D_1$ and $d_0(R,F) < D_3$

Hence $\forall t \in [t_{16}, +\infty)$, $\vec{p}(R,t) - \vec{p}(V_i,t) \geqslant \vec{p}(R,t_{16}) - \vec{p}(F,t_{16}) = d_0(R,F) + v_{\mathsf{lim}}\delta^{\mathsf{lc}}_{\mathsf{defer}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) - d^{\ddagger}_{\mathsf{d}} - v_{\mathsf{low}}\delta^{\mathsf{RF1}}_{\mathsf{low}} = d_0(R,F) + v_{\mathsf{lim}}(\frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta^{\ddagger}_{\mathsf{d}}) + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) - d^{\ddagger}_{\mathsf{d}} - v_{\mathsf{low}}(\frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + \delta_{\mathsf{lc}}(v_{\mathsf{lim}})) = d_0(R,F) + \widetilde{v}\frac{D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}})}{\widetilde{v}} + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{d}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) - d^{\ddagger}_{\mathsf{d}} - v_{\mathsf{low}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = d_0(R,F) + D_3 - d_0(R,F) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}}) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) - v_{\mathsf{low}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = v_{\mathsf{lim}}\Delta^* + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}}) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) - v_{\mathsf{low}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) = v_{\mathsf{lim}}\Delta^* + \widetilde{v}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{low}}) + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) \geqslant v_{\mathsf{lim}}\Delta^*$. Hence (R, V_i) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{\imath, \imath+1, \ldots, n\}$, (R, V_i) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Meanwhile, if $\imath > 1$, then the leader CAV L exists and $L = V_{\imath-1}$. Furthermore, we have $\forall t \in [t_{16}, +\infty)$, $\vec{p}(V_{\imath-1}, t) - \vec{p}(R, t) = \vec{p}(L, t_{16}) - \vec{p}(R, t_{16}) = d_0(L, R) + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{lim}}) - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{lim}}) = d_0(L, R) + \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) \geqslant D_1 + \widetilde{d_{\mathsf{lc}}}(v_{\mathsf{lim}}) = v_{\mathsf{lim}}\Delta^*$. Hence, $(V_{\imath-1}, R)$ is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{1, 2, \ldots, \imath-1\}$, (V_i, R) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Case 4: If the mechanical lane change routine is triggered via event "GotDecelerate," then the event must have happened at $t_{35} \overset{\mathsf{def}}{=} t_{33} - \delta^{\mathsf{Rdec}}_{\mathsf{defer}}$.

Similar to **Case 1**, we can prove all target lane CAVs are in mode "Init" at t_{35}. [4]

Suppose at t_{35}, the recognized follower CAV $F = V_\imath$ ($\imath \in \{1, 2, \ldots, n\}$). After t_{35}, the mode change time instances of R and F have the following relationships:

i) $\delta^{\mathsf{Rdec}}_{\mathsf{defer}} = \delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{RF2}}_{\mathsf{low}}$, which implies at $t_{33} = t_{35} + \delta^{\mathsf{Rdec}}_{\mathsf{defer}}$, R starts to decelerate simultaneously as F starts to accelerate;

[4] Specifically, suppose $\exists$ target lane CAV V_i ($V_i \neq F$, the follower of the triggering "GotDecelerate" event) in a coop-duration $(t_{43}, t_{44}]$, s.t. $t_{35} \in (t_{43}, t_{44}]$. Then $t_{43} < t_{35}$, hence $|t_{43} - t_{35}| \leqslant \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. But in order to trigger the coop-duration at t_{43}, R must have fired "Event3" (or "Event4") at t_{43} (due to (**c6**)) to $V_i \neq F$. Meanwhile, "Event4" is fired at t_{35} to F (due to (**c6**)). As "Event3" (or "Event4") to V_i and "Event4" to F must be two different events, hence must be separated by at least $\Delta_{\mathsf{reset}} + \Delta_{\mathsf{nonzeno}}$, this means $|t_{43} - t_{35}| > \Delta_{\mathsf{reset}}$. This implies $\Delta^{\mathsf{max}}_{\mathsf{Coop}} > \Delta_{\mathsf{reset}}$, which contradicts Eq. (4.20). On the other hand, suppose F is in a coop-duration $(t_{43}, t_{44}]$, s.t. $t_{35} \in (t_{43}, t_{44}]$. Then $t_{43} < t_{35}$, hence $|t_{43} - t_{35}| \leqslant \Delta^{\mathsf{max}}_{\mathsf{Coop}}$. But in order to trigger the coop-duration at t_{43} for F, R must have fired "Event3" (or "Event4") at t_{43} (due to (**c6**)) to F. Meanwhile, "Event4" is fired at t_{35} to F (due to (**c6**)). As $t_{43} < t_{35}$, "Event3" (or "Event4") to F at t_{43} and "Event4" to F at t_{35} must be two different events, hence must be separated by at least $\Delta_{\mathsf{reset}} + \Delta_{\mathsf{nonzeno}}$, this means $|t_{43} - t_{35}| > \Delta_{\mathsf{reset}}$. This implies $\Delta^{\mathsf{max}}_{\mathsf{Coop}} > \Delta_{\mathsf{reset}}$, which contradicts Eq. (4.20). Therefore, at t_{35}, no target lane CAV is in a coop-duration. Due to Lemma 9, this means all target lane CAVs are in mode "Init" at t_{35}.

ii) F recovers the speed v_{lim} at $t_{36} \overset{\mathsf{def}}{=} t_{33} + \delta_{\mathsf{a}}^{\ddagger} \leqslant t_{16} = t_{33} + \delta_{\mathsf{d}}^{\ddagger} + \delta_{\mathsf{low}}^{\mathsf{LR}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}})$ (due to (**c4**)). Hence, $\forall t \in [t_{16}, +\infty)$, F's speed is v_{lim}.

Therefore, the entire lane change process is illustrated by Fig. B.5

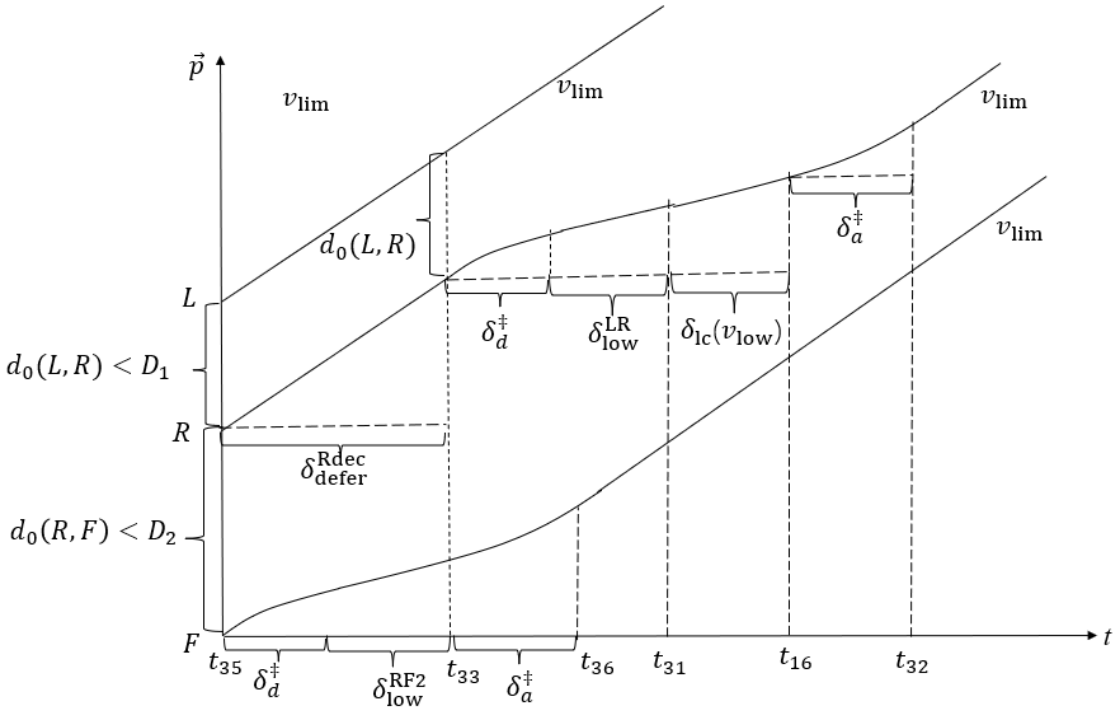


Figure B.5: The entire lane change process: $d_0(L,R) < D_1$ and $d_0(R,F) < D_2$

Hence $\forall t \in [t_{16}, +\infty)$,

$$
\begin{aligned}
& \vec{p}(R,t) - \vec{p}(V_i,t) = \vec{p}(R,t) - \vec{p}(F,t) \\
\geqslant\ & \vec{p}(R,t_{16}) + v_{\mathsf{low}}\delta^{\ddagger}_{\mathsf{a}} - (\vec{p}(F,t_{16}) + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{a}}) \\
=\ & \vec{p}(R,t_{16}) - \vec{p}(F,t_{16}) - \widetilde{v}\delta^{\ddagger}_{\mathsf{a}} \\
=\ & d_0(R,F) + v_{\mathsf{lim}}\delta^{\mathsf{Rdec}}_{\mathsf{defer}} + d^{\ddagger}_{\mathsf{d}} + v_{\mathsf{low}}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) - \\
& (d^{\ddagger}_{\mathsf{d}} + v_{\mathsf{low}}\delta^{\mathsf{RF2}}_{\mathsf{low}} + d^{\ddagger}_{\mathsf{a}} + v_{\mathsf{lim}}(t_{16} - t_{36})) - \widetilde{v}\delta^{\ddagger}_{\mathsf{a}} \\
=\ & d_0(R,F) + v_{\mathsf{lim}}(\delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{RF2}}_{\mathsf{low}}) + v_{\mathsf{low}}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \\
& - v_{\mathsf{low}}\delta^{\mathsf{RF2}}_{\mathsf{low}} - d^{\ddagger}_{\mathsf{a}} - v_{\mathsf{lim}}(\delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{LR}}_{\mathsf{low}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}}) - \delta^{\ddagger}_{\mathsf{a}}) \\
& - \widetilde{v}\delta^{\ddagger}_{\mathsf{a}} \\
=\ & d_0(R,F) + \widetilde{v}\delta^{\mathsf{RF2}}_{\mathsf{low}} - \widetilde{v}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \\
& - d^{\ddagger}_{\mathsf{a}} - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{a}} - (v_{\mathsf{lim}} - v_{\mathsf{low}})\delta^{\ddagger}_{\mathsf{a}} \\
=\ & d_0(R,F) + \widetilde{v}\delta^{\mathsf{RF2}}_{\mathsf{low}} - \widetilde{v}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\ddagger}_{\mathsf{a}} \\
& - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) + v_{\mathsf{low}}\delta^{\ddagger}_{\mathsf{a}} \\
\\
=\ & d_0(R,F) + D_2 - d_0(R,F) - v_{\mathsf{lim}}\Delta^* + d_0(L,R) \\
& + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\ddagger}_{\mathsf{a}} - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) + v_{\mathsf{low}}\delta^{\ddagger}_{\mathsf{a}} \\
=\ & 2v_{\mathsf{lim}}\Delta^* + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + v_{\mathsf{low}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \\
& + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{a}} - v_{\mathsf{low}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) - v_{\mathsf{low}}\delta^{\ddagger}_{\mathsf{a}} \\
& - v_{\mathsf{lim}}\Delta^* + d_0(L,R) + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\ddagger}_{\mathsf{a}} - v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) \\
& + v_{\mathsf{low}}\delta^{\ddagger}_{\mathsf{a}} \\
=\ & v_{\mathsf{lim}}\Delta^* + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{a}} + d_0(L,R) - d^{\ddagger}_{\mathsf{a}} \\
=\ & v_{\mathsf{lim}}\Delta^* + \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + \widetilde{d}_{\mathsf{a}}(v_{\mathsf{lim}}) + d_0(L,R) \geqslant v_{\mathsf{lim}}\Delta^*.
\end{aligned}
$$

Hence (R, V_i) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{\imath, \imath+1, \ldots, n\}$, (R, V_i) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Meanwhile, if $\imath > 1$, then the leader CAV L exists and $L = V_{\imath-1}$. Furthermore, we have $\forall t \in [t_{16}, +\infty)$,

$$\begin{aligned}
& \vec{p}(V_{\imath-1}, t) - \vec{p}(R, t) \geqslant \vec{p}(L, t_{16}) - \vec{p}(R, t_{16}) \\
= \;& d_0(L, R) + v_{\mathsf{lim}}(t_{16} - t_{33}) - (d^{\ddagger}_{\mathsf{d}} + v_{\mathsf{low}}\delta^{\mathsf{LR}}_{\mathsf{low}} + d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}})) \\
= \;& d_0(L, R) + v_{\mathsf{lim}}(\delta^{\ddagger}_{\mathsf{d}} + \delta^{\mathsf{LR}}_{\mathsf{low}} + \delta_{\mathsf{lc}}(v_{\mathsf{low}})) \\
& - d^{\ddagger}_{\mathsf{d}} - v_{\mathsf{low}}\delta^{\mathsf{LR}}_{\mathsf{low}} - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \\
= \;& d_0(L, R) + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{d}} + \widetilde{v}\delta^{\mathsf{LR}}_{\mathsf{low}} + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) \\
& - d^{\ddagger}_{\mathsf{d}} - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \\
= \;& d_0(L, R) + v_{\mathsf{lim}}\delta^{\ddagger}_{\mathsf{d}} + v_{\mathsf{lim}}\Delta^* - d_0(L, R) + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) \\
& - d^{\ddagger}_{\mathsf{d}} - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \\
= \;& \widetilde{d}_{\mathsf{d}}(v_{\mathsf{lim}}) + v_{\mathsf{lim}}\Delta^* + v_{\mathsf{lim}}\delta_{\mathsf{lc}}(v_{\mathsf{low}}) - d^{\mathsf{X}}_{\mathsf{lc}}(v_{\mathsf{low}}) \geqslant v_{\mathsf{lim}}\Delta^*.
\end{aligned}$$

Hence, $(V_{\imath-1}, R)$ is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Due to Lemma 11 and Corollary 2, we have $\forall i \in \{1, 2, \ldots, \imath - 1\}$, (V_i, R) is CTH-Δ^* safe throughout $[t_{16}, +\infty)$.

Combining **Case 1** to **Case 4**, we conclude the lemma sustains.

References

[1] Are self-driving cars about to get their own lane? *Leonard*, May 16 2019.

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

[3] M. Aeberhard et al. Experience, results and lessons learned from automated driving on germany's highways. *IEEE Intelligent Transportation Systems Magazine*, 7(1):42–57, Spring 2015.

[4] M. Ahmed et al. A cooperative freeway merge assistance system using connected vehicles. *preprint arXiv:1805.00508*, 2018.

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

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

[7] T. Awal et al. Optimal traffic merging strategy for communication-and sensor-enabled vehicles. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1468–1474. IEEE, 2013.

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

[9] Shahian-Jahromi Babak, Syed A Hussain, Burak Karakas, and Sabri Cetin. Control of autonomous ground vehicles: a brief technical review. In *IOP Conference Series: Materials Science and Engineering*, volume 224, page 012029. IOP Publishing, 2017.

[10] Kaleb Ben Naveed, Zhiqian Qiao, and John M Dolan. Trajectory planning for autonomous vehicles using hierarchical reinforcement learning. *arXiv e-prints*, pages arXiv–2011, 2020.

[11] D. Bernardo et al. Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays. *IEEE Trans. on Intelligent Transportation Systems*, 16(1):102–112, 2014.

[12] David Bevly, Xiaolong Cao, et al. Lane change and merge maneuvers for connected and automated vehicles: A survey. *IEEE Transactions on Intelligent Vehicles*, 1(1):105–120, 2016.

[13] Peide Cai, Yuxiang Sun, Yuying Chen, and Ming Liu. Vision-based trajectory planning via imitation learning for autonomous vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2736–2742. IEEE, 2019.

[14] W. Cao et al. Cooperative vehicle path generation during merging using model predictive control with real-time optimization. *Control Engineering Practice*, 34:98–105, 2015.

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

[16] Wonshik Chee. Unified lateral control system for intelligent vehicles. *SAE transactions*, pages 531–539, 2000.

[17] Wonshik Chee and Masayoshi Tomizuka. Unified lateral motion control of vehicles for lane change maneuvers in automated highway. *Info:*.

[18] H. Chehardoli et al. Adaptive centralized/decentralized control and identification of 1-d heterogeneous vehicular platoons based on constant time headway policy. *IEEE Trans. on Intelligent Transportation Systems*, 19(10):3376–3386, 2018.

[19] Cheng Chen, Yuqing He, et al. Quartic bézier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6108–6113. IEEE, 2014.

[20] Tianyi Chen, Meng Wang, Siyuan Gong, Yang Zhou, and Bin Ran. Connected and automated vehicle distributed control for on-ramp merging scenario: A virtual rotation approach. *arXiv preprint arXiv:2103.15047*, 2021.

[21] Xiaolong Chen, Bing Zhou, and Xiaojian Wu. Autonomous vehicle path tracking control considering the stability under lane change. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, page 0959651821991357, 2021.

[22] Ji-wung Choi, Renwick Curry, and Gabriel Elkaim. Path planning based on bézier curve for autonomous ground vehicles. In *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pages 158–166. IEEE, 2008.

[23] Yong-Geon Choi, Kyung-Il Lim, and Jung-Ha Kim. Lane change and path planning of autonomous vehicles using gis. In *2015 12th International Conference*

on Ubiquitous Robots and Ambient Intelligence (URAI), pages 163–166. IEEE, 2015.

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

[25] L.C. Davis. Optimal merging into a high-speed lane dedicated to connected autonomous vehicles. *Phsica A*, 555, 2020.

[26] K. Dey et al. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc). *IEEE Trans. on Intelligent Transportation Systems*, 17(2):491–509, February 2016.

[27] Kakan C Dey, Li Yan, et al. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc). *IEEE Transactions on Intelligent Transportation Systems*, 17(2):491–509, 2015.

[28] Yang Ding, Weichao Zhuang, et al. Safe and optimal lane-change path planning for automated driving. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, page 0954407020913735, 2020.

[29] Moustapha Doumiati, Ali Charara, Alessandro Victorino, and Daniel Lechner. *Vehicle dynamics estimation using Kalman filtering: experimental validation.* John Wiley & Sons, 2012.

[30] J. Du et al. Estimation of vehicle emission on mainline freeway under isolated and integrated ramp metering strategies. *Environmental Engineering and Management Journal*, 17(5):1237–1248, May 2018.

[31] David Elliott, Walter Keen, and Lei Miao. Recent advances in connected and automated vehicles. *Journal of Traffic and Transportation Engineering (English Edition)*, 6(2):109–131, 2019.

[32] Azim Eskandarian. *Handbook of intelligent vehicles*, volume 2. Springer, 2012.

[33] Hossam Gaber, Ahmed M Othman, and Abul Hasan Fahad. Future of connected autonomous vehicles in smart cities. In *Solving Urban Infrastructure Problems Using Smart City Technologies*, pages 599–611. Elsevier, 2021.

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

[35] Subhrajit Guhathakurta and Amit Kumar. When and where are dedicated lanes needed under mixed traffic of automated and non-automated vehicles for optimal system level benefits? Technical report, Georgia Institute of Technology and University of Texas at San Antonio, 2019.

[36] Maanak Gupta, James Benson, Farhan Patwa, and Ravi Sandhu. Secure v2v and v2i communication in intelligent transportation using cloudlets. *IEEE Transactions on Services Computing*, 2020.

[37] Cem Hatipoglu, Umit Ozguner, and Keith A Redmill. Automated lane change controller design. *IEEE transactions on intelligent transportation systems*, 4(1):13–22, 2003.

[38] Jianhua He, Zuoyin Tang, Xiaoming Fu, Supeng Leng, Fan Wu, Kaisheng Huang, Jianye Huang, Jie Zhang, Yan Zhang, Andrew Radford, et al. Cooperative connected autonomous vehicles (cav): research, applications and challenges. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–6. IEEE, 2019.

[39] S. Heim et al. Automated platoon manipulation in merging scenarios using trajectory estimation of connected vehicles. Technical report, US ARMY CCDC GVSC (FORMERLY TARDEC) WARREN United States, 2019.

[40] Thomas A Henzinger. The theory of hybrid automata. In *Verification of digital and hybrid systems*, pages 265–292. Springer, 2000.

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

[42] SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles (sae j3016_202104). April 30 2021.

[43] Jordan Ivanchev, Alois Knoll, Daniel Zehe, Suraj Nair, and David Eckhoff. Potentials and implications of dedicated highway lanes for autonomous vehicles. Technical report, Technische Universität München, Germany, Sep. 25 2017.

[44] Xiaowen Jiang, Peter J Jin, Xia Wan, and Yizhou Wang. A v2i (vehicle-to-infrastructure) based dynamic merge assistance method based on instantaneous virtual trajectories: A microscopic implementation of gap metering. In *Transportation Research Board 96th Annual Meeting*, 2017.

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

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

[47] Geon-Hee Kim, Dong-Sung Pae, Woo-Jin Ahn, Koung-Suk Ko, Myo-Tae Lim, and Tae-Koo Kang. Vehicle positioning system using v2x that combines v2v and v2i communications. In *IOP Conference Series: Materials Science and Engineering*, volume 922(1), page 012009. IOP Publishing, 2020.

[48] Jinsoo Kim, Jahng-Hyon Park, and Kyung-Young Jhang. Decoupled longitudinal and lateral vehicle control based autonomous lane change system adaptable to driving surroundings. *IEEE Access*, 9:4315–4334, 2020.

[49] Stefan R. Klomp, Victor L. Knoop, Henk Taale, and Serge P. Hoogendoorn. Ramp metering with microscopic gap detection algorithm design and empirical acceleration verification. *Transportation Research Record*, August 2021.

[50] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099. IEEE, 2015.

[51] Hanna Krasowski, Xiao Wang, and Matthias Althoff. Safe reinforcement learning for autonomous lane changing using set-based prediction. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020.

[52] Tom Krisher, David Eggert, and Joe McDonald. Michigan plans dedicated road lanes for autonomous vehicles. *ABC News*, August 14 2020.

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

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

[55] Tingting Li, Jianping Wu, Ching-Yao Chan, Mingyu Liu, Chunli Zhu, Weixin Lu, and Kezhen Hu. A cooperative lane change model for connected and automated vehicles. *IEEE Access*, 8:54940–54951, 2020.

[56] Bin Liu, Fei-Yue Wang, Jason Geng, Qingming Yao, Hui Gao, and Buqing Zhang. Intelligent spaces: An overview. In *Proc. of IEEE International Conference on Vehicular Electronics and Safety*, pages 1–6, 2007.

[57] Wei Liu, Zhiheng Li, Li Li, and Fei-Yue Wang. Parking like a human: A direct trajectory planning solution. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3388–3397, 2017.

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

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

[60] Aki Lumiaho and Fanny Malin. Road transport automation: road map and action plan 2016-2020. *Liikenneviraston tutkimuksia ja selvityksiä*, 2016.

[61] Yugong Luo, Yong Xiang, et al. A dynamic automated lane change maneuver based on vehicle-to-vehicle communication. *Transportation Research Part C: Emerging Technologies*, 62:87–102, 2016.

[62] Yugong Luo, Gang Yang, Mingchang Xu, Zhaobo Qin, and Keqiang Li. Cooperative lane-change maneuver for multiple automated vehicles on a highway. *Automotive Innovation*, 2(3):157–168, 2019.

[63] Shannon Sanders McDonald and Caroline Rodier. Envisioning automated vehicles within the built environment: 2020, 2035, and 2050. In *Road Vehicle Automation 2*, pages 225–233. Springer, 2015.

[64] H. Min et al. 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):582–595, 2021.

[65] F. Monteiro et al. Safe lane change and merging gaps in connected environments. *IFAC-PapersOnLine*, 54(2):69–74, 2021.

[66] América Morales and Henk Nijmeijer. Merging strategy for vehicles by applying cooperative tracking control. *IEEE Trans. on Intelligent Transportation Systems*, 17(12):3423–3433, 2016.

[67] Amirreza Nickkar and Young-Jae Lee. Evaluation of dedicated lanes for automated vehicles at roundabouts with various flow patterns. *arXiv preprint arXiv:1904.07025*, 2019.

[68] Julia Nilsson, Jonatan Silvlin, Mattias Brannstrom, Erik Coelingh, and Jonas Fredriksson. If, when, and how to perform lane change maneuvers on highways. *IEEE Intelligent Transportation Systems Magazine*, 8(4):68–78, 2016.

[69] Ioannis A Ntousakis, Kallirroi Porfyri, Ioannis K Nikolos, and Markos Papageorgiou. Assessing the impact of a cooperative merging system on highway traffic using a microscopic flow simulator. In *ASME 2014 International Mechanical Engineering Congress and Exposition*, pages V012T15A024–V012T15A024. American Society of Mechanical Engineers, 2014.

[70] United States Department of Transportation National Highway Traffic Safety Administration. Nationall motor vehicle crash causation survey (report to congress). *DOT HS 811 059*, July 2008.

[71] United States Department of Transportation National Highway Traffic Safety Administration. A brief statistical summary: critical reasons for crashes investigated in the national motor vehicle crash causation survey. *DOT HS 812 115*, February 2015.

[72] Plamen Petrov and Fawzi Nashashibi. Adaptive steering control for autonomous lane change maneuver. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 835–840. IEEE, 2013.

[73] Judith Princeton and Simon Cohen. Impact of a dedicated lane on the capacity and the level of service of an urban motorway. *Procedia Social and Behavioral Sciences*, 16:196-206, 2011.

[74] Rattaphol Pueboobpaphan, Fei Liu, and Bart van Arem. 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 *Proc. of the 13th Intl. IEEE Conf. on Intelligent Transportation Systems*, pages 1468–1473. IEEE, 2010.

[75] Solmaz Razmi Rad, Haneen Farah, Henk Taale, and Bart van Arem. Design and operation of dedicated lanes for connected and automated vehicles on motorways: A conceptual framework and research agenda. *Transportation Research Part C*, 117, 2020.

[76] Solmaz Razmi Rad, Haneen Farah, Henk Taale, Bart van Arem, and Serge P Hoogendoorn. Design and operation of dedicated lanes for connected and automated vehicles on motorways: A conceptual framework and research agenda. *Transportation research part C: emerging technologies*, 117:102664, 2020.

[77] Mohsen Rafat and Shahram Azadi. A novel adaptive lane change method in transient dynamic traffic conditions for highly automated vehicles. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, page 14644193211024800, 2021.

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

[79] G. Raravi et al. Merge algorithms for intelligent vehicles. In *Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems*, pages 51–65. Springer, 2007.

[80] Jean-François Raskin. An introduction to hybrid automata. In *Handbook of networked and embedded control systems*, pages 491–517. Springer, 2005.

[81] John Rowley, Ann Liu, Steven Sandry, Joshua Gross, Melvin Salvador, Chris Anton, and Cody Fleming. Examining the driverless future: An analysis of human-caused vehicle accidents and development of an autonomous vehicle communication testbed. *Proc. of Systems and Information Engineering Design Symposium (SIEDS)*, pages 58–63, April 27 2018.

[82] Ahmed Hamdi Sakr, Gaurav Bansal, et al. Lane change detection using v2v safety messages. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3967–3973. IEEE, 2018.

[83] M. Shalaby et al. Design of various dynamical-based trajectory tracking control strategies for multi-vehicle platooning problem. In *Proc. of the IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1631–1637, October 2019.

[84] Tianyu Shi, Pin Wang, Xuxin Cheng, Ching-Yao Chan, and Ding Huang. Driving decision and control for automated lane change behavior based on deep reinforcement learning. In *2019 IEEE intelligent transportation systems conference (ITSC)*, pages 2895–2900. IEEE, 2019.

[85] Darbha Swaroop and KR Rajagopal. A review of constant time headway policy for automatic vehicle following. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, pages 65–69. IEEE, 2001.

[86] F. Tan et al. A lease based hybrid design pattern for proper-temporal-embedding of wireless cps interlocking. *IEEE Trans. on Parallel and Distributed System*, 26(10):2630–2642, 2015.

[87] Feng Tan, Yufei Wang, Qixin Wang, et al. A lease based hybrid design pattern for proper-temporal-embedding of wireless cps interlocking. *IEEE Transactions on Parallel and Distributed Systems*, 26(10):2630–2642, 2014.

[88] Danyang Tian, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J Barth. Performance measurement evaluation framework and co-benefit\/tradeoff analysis for connected and automated vehicles (cav) applications: A survey. *IEEE Intelligent Transportation Systems Magazine*, 10(3):110–122, 2018.

[89] J.R. Treat, N.S. Tumbas, S.T. McDonald, Shinar D., Hume R.D., Mayer R.E., R.L. Stansifer, and N.J. Castellan. Tri-level study of the causes of traffic accidents executive summary. *United States Department of Transportation National Highway Traffic Safety Administration DOT HS-034-3-535*, May 1979.

[90] Sadayuki Tsugawa, Shin Kato, et al. A cooperative driving system with automated vehicles and inter-vehicle communications in demo 2000. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, pages 918–923. IEEE, 2001.

[91] Charlott Vallon, Ziya Ercan, Ashwin Carvalho, and Francesco Borrelli. A machine learning approach for personalized autonomous lane change initiation and control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1590–1595. IEEE, 2017.

[92] Christophe Viel, Ulysse Vautier, Jian Wan, and Luc Jaulin. Platooning control for heterogeneous sailboats based on constant time headway. *IEEE Trans. on Intelligent Transportation Systems*, 21(5):2078–2089, 2019.

[93] Fei-Yue Wang. Agent-based control for networked traffic management systems. *IEEE Intelligent Systems*, 20(5):92-96, 2005.

[94] Fei-Yue Wang. Transportation 5.0 in cpss: Autonomous vehicles, parallel driving, intelligent traffic, and smart cities. *Keynote in IEEE Fellow Cloud Forum*, August 26 2021.

[95] Fei-Yue Wang, Daniel Zeng, and Liuqing Yang. Smart cars on smart roads: An ieee intelligent transportation systems society update. *IEEE Pervasive Computing*, 5(4):68-69, Oct-Dec 2006.

[96] Fei-Yue Wang, Nan-Ning Zheng, Dongpu Cao, Clara Marina Martinez, Li Li, and Teng Liu. Parallel driving in cpss: A unified approach for transport automation and vehicle intelligence. *IEEE/CAA Journal of Automatica Sinica*, 4(4):577-587, 2017.

[97] Le Wang, Renato F Iida, and Alexander M Wyglinski. Coordinated lane changing using v2v communications. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2018.

[98] Meng Wang. Infrastructure assisted adaptive driving to stabilise heterogeneous vehicle strings. *Transportation Research Part C*, 91:276–295, April 2018.

[99] Pin Wang, Ching-Yao Chan, and Arnaud de La Fortelle. A reinforcement learning based approach for automated lane change maneuvers. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1379–1384. IEEE, 2018.

[100] Y. Wang et al. Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study. *IEEE Trans. on Intelligent Transportation Systems*, 15(5):2024–2039, October 2014.

[101] Yunpeng Wang, E Wenjuan, Wenzhong Tang, Daxin Tian, Guangquan Lu, and Guizhen Yu. Automated on-ramp merging control algorithm based on internet-connected vehicles. *IET Intelligent Transport Systems*, 7(4):371–379, 2013.

[102] Ziran Wang, Guoyuan Wu, and Matthew Barth. Distributed consensus-based cooperative highway on-ramp merging using v2x communications. Technical report, SAE Technical Paper, 2018.

[103] Maggie Weeks. Solving the autonomous vehicle safety challenges. https://rtr.ai/solving-the-autonomous-vehicle-safety-challenges/, 2019.

[104] Y. Xie et al. Collaborative merging strategy for freeway ramp operations in a connected and autonomous vehicles environment. *Journal of Intelligent Transportation Systems*, 21(2):136–147, 2017.

[105] Guoqing Xu, Li Liu, et al. Dynamic modeling of driver control strategy of lane-change behavior and trajectory planning for collision prediction. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1138–1155, 2012.

[106] Guangchuan Yang and Zong Tian. An exploratory investigation of the impact of ramp metering on driver acceleration behavior. *IATSS Research*, 42(4):277–285, December 2019.

[107] Kai Yang, Xiaolin Tang, Yechen Qin, Yanjun Huang, Hong Wang, and Huayan Pu. Comparative study of trajectory tracking control for automated vehicles via model predictive control and robust h-infinity state feedback control. *Chinese Journal of Mechanical Engineering*, 34(1):1–14, 2021.

[108] Balaji Yelchuru. Dedicating lanes for priority or exclusive use by connected and automated vehicles. Technical report, NCHRP Research Report 891, 2018.

[109] M. Yoshida et al. A scheduling scheme for cooperative merging at a highway on-ramp with maximizing average speed of automated vehicles. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–5. IEEE, 2020.

[110] Xiaohai Yu, Ge Guo, and Hongbo Lei. Longitudinal cooperative control for a bidirectional platoon of vehicles with constant time headway policy. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 2427–2432. IEEE, 2018.

[111] Nur Ezzati Yunus, Siti Fatimah Abdul Razak, Sumendra Yogarayan, and Mohd Fikri Azli Abdullah. Lane changing models: A short review. In *2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC)*, pages 110–115. IEEE, 2021.

[112] Shuya Zong. How connected autonomous vehicles would affect our world? a literature review on the impacts of cav on road capacity, environment and public attitude. In *MATEC Web of Conferences*, volume 296, page 01007. EDP Sciences, 2019.