Mälardalen University

Department of Computer Science and Engineering

Västerås, Sweden

Thesis for the Degree of Bachelor of Science in Network Engineering -

Computer Network Engineering, 15.0 credits

On the Impact of Delay Attacks on Connected and Automated Vehicles Braking

Sebastian Krzysztof Bober

sbr23006@student.mdu.se

Examiner: Giuseppe Nebbione
Mälardalen University, Västerås, Sweden

Supervisor: Shahriar Hasan
Mälardalen University, Västerås, Sweden

20/05/2026

Abstract

Connected and Automated Vehicles (CAVs) and associated with them Platooning Systems are considered key elements of future Intelligent Transportation Systems (ITS). Vehicle-to-Vehicle (V2V) communication allows vehicles in a platoon to maintain close distance gaps, which improves traffic efficiency, reduces fuel consumption and improves road safety. At the same time, platooning relies heavily on reliable and low-latency wireless communication, which is vulnerable to cyberattacks. Communication delays, interference, or targeted attacks can negatively impact string stability, and therefore safety of platoons. This thesis focuses on a specific type of attack: the delay attack. It analyzes its impact on CAVs braking, string stability of CAVs and different control models used for platooning. The experiment in this research was conducted using simulation frameworks. A series of experiments were conducted for different communication delay values, inter-vehicle distances, vehicle speeds, deceleration levels, and the number of vehicles in the platoon. Robustness of different vehicle control models in the platoon under delay-attack conditions was analyzed. Results show that communication delay can significantly affect platoon stability and collision occurrence, with the severity of its impact depending on the injected delay, attack timing, platoon size, and controller configuration. These findings demonstrate the importance of considering communication delays when evaluating safety, fault tolerance and robustness of automated vehicle platoons.

Acronyms

ACC	Adaptive Cruise Control
ACK	Acknowledgement
CACC	Cooperative Adaptive Cruise Control
CAV	Connected and Automated Vehicle
CAM	Cooperative Awareness Message
CC	Cruise Control
CDG	Constant Distance Gap
CLI	Command Line Interface
ComFASE	Communication-based Fault and Attack Simulation Engine
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CTG	Constant Time Gap
DoS	Denial of Service
DSRC	Dedicated Short-Range Communication
ESB	Enhanced Synchronized Braking
EXACT	EXtended ACTuator
GPS	Global Positioning System
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
ITS	Intelligent Transportation Systems
IVC	Inter-Vehicle Communication
LF	Leader-Following
LPF	Leader Predecessor-Following
LV	Leading Vehicle
MAC	Medium Access Control
ML	Machine Learning
OMNeT++	A network simulator
PD	Propagation Delay
PF	Predecessor-Following
PHY	Physical Layer
PKI	Public Key Infrastructure
PLEXE	A cooperative driving framework
PLOEG	A platooning controller
PRNG	Pseudo-Random Number Generator
RNG	Random Number Generator
SAE	Society of Automotive Engineers
SDR	Software-defined radio

SINR	Signal-to-Interference-plus-Noise Ratio
SoS	System-of-Systems
SUMO	Simulation of Urban Mobility
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular Ad Hoc Network
Veins	Vehicles in Network Simulation
VeReMi	Vehicular Reference Misbehavior Extension
WAVE	Wireless Access in Vehicular Environment
Wi-Fi	Wireless Fidelity

Table of Contents

1 Introduction **9**

2 Background **12**
2.1 Vehicular Ad Hoc Networks (VANETs) . . . 12
2.2 Connected and Automated Vehicles (CAVs) . . . 12
2.3 IEEE 802.11p . . . 12
2.4 String stability . . . 14
2.5 Longitudinal Control Models . . . 14
2.5.1 Constant Time Gap . . . 14
2.5.2 Constant Distance Gap . . . 14
2.5.3 Adaptive Cruise Control (ACC) . . . 15
2.5.4 Cooperative Adaptive Cruise Control (CACC) . . . 15
2.6 CACC Controllers . . . 16
2.6.1 Rajamani's CACC controller . . . 16
2.6.2 Ploeg's CACC controller . . . 18
2.7 Simulation Platforms for Platooning and Security Analysis . . . 20
2.7.1 OMNeT++ . . . 20
2.7.2 SUMO . . . 20
2.7.3 Veins . . . 21
2.7.4 PLEXE . . . 21
2.7.5 ComFASE . . . 21

3 Related work **23**
3.1 V2V Communication Security . . . 23
3.2 ComFASE: A Tool for Evaluating Communication Faults and Attacks on Vehicle Platoons . . . 24
3.2.1 Jamming as a Possible Attack Vector against Platoon Communication . . . 24
3.2.2 Modelling of Reactive-Jamming Effects and Delay attack . . . 25
3.3 Delay Impact on Braking Platoons . . . 26
3.4 Delay Attack Impact on Platoons . . . 27
3.5 VeReMi and VeReMi Extension datasets . . . 29
3.5.1 VeReMi NextGen dataset . . . 31
3.6 Summary . . . 31

4 Problem formulation **33**

5 Method **35**

6 Ethical considerations **38**

7 Simulation environment and experimental setup 39
7.1 A pre-campaign. Collision Threshold in the Braking Scenario 39
7.2 Run-level Dataset 40
7.3 Main campaign. Delay attacks simulations using ComFASE on Braking Scenario and Sinusoidal Scenario in Plexe-Veins 41
7.4 Experiment parameters 48
7.4.1 Deceleration 48
7.4.2 Speed 51
7.4.3 Platoon size 52
7.4.4 Inter Vehicle Distances 53
7.4.5 Injected Delay 53
7.5 Braking Dynamics in the Platoon 55
7.5.1 Complementary Beacon-Level Dataset 55
7.5.2 Message Delivery Delay 55
7.5.3 Physical Lag Response 56
7.5.4 Signal-to-Interference-plus-Noise Ratio (SINR) 57

8 Results 60
8.1 Delay Threshold in the Braking Scenario 60
8.2 Factors Affecting Platoon Safety under Delay Attacks 61
8.3 Time-Series Dataset for Machine Learning-Based Collision Prediction 66
8.3.1 Collision Prediction Models 66
8.3.2 Event-Level Evaluation 71

9 Discussion 73

10 Conclusions 75

11 Future Research Directions 76

References 77

A Simulation Environment and Research Data 82

List of Figures

1 Schematic representation of a CACC-equipped vehicle platoon based on the Leader Predecessor-Following communication topology. 16
2 Schematic representation of a CACC-equipped vehicle platoon based on the predecessor-following communication topology. 18
3 Research methodology adopted in this thesis. 35
4 Experimental workflow of the simulation campaign implemented using ComFASE. 43
5 Vehicle acceleration profiles in the Braking Scenario. The figure illustrates the dynamic response of the platoon to the braking maneuver initiated by the platoon leader. 47
6 Vehicle acceleration profiles in the Sinusoidal Scenario. The figure illustrates the dynamic response of the platoon to periodic speed variations of the platoon leader (oscillation). 47
7 Vehicle acceleration profiles during a delay attack in the Braking Scenario. The figure shows the delayed braking response and braking disturbance amplification in the platoon after emergency braking is initiated by the leader. 49
8 Vehicle acceleration profiles during a delay attack in the Sinusoidal Scenario. The figure shows the delayed braking response, acceleration and deceleration disturbance and consequently, gradual loss of string stability. 50
9 Vehicle acceleration profiles during a delay attack in the Sinusoidal Scenario. The figure shows the critical region preceding a collision, where the inter-vehicle distance decreases to an unsafe level, compromising platoon safety. 50
10 Per-beacon send-to-receive delay of leader messages during the Delay Attack. Each panel represents a direct transmission from the leader, `node[0]`, to one of the following vehicles. Every marker corresponds to one successfully received beacon and was calculated directly as $1000(t_{\text{receive}} - t_{\text{send}})$. The median observed delay was approximately 1004.483 ms for each receiver, consisting of the configured 1000 ms artificial delay superimposed on an existing message-delivery latency of approximately 4.483 ms. 56
11 Controller response and vehicle dynamics during the platoon braking maneuver. The figure presents the acceleration commanded by the controller, the actual vehicle acceleration, and the vehicle speed for the leader and the three following vehicles. The markers indicate the earliest detected moments at which the actual acceleration crossed the adopted braking threshold. The difference between the commanded and actual acceleration illustrates the gradual physical response represented by the modelled vehicle and powertrain dynamics. 58

12 SINR values of leader beacons received by the three following vehicles during the Delay Attack and braking interval. Each panel represents a direct communication link from the leader, `node[0]`, to one of the following vehicles. Each marker corresponds to one successfully received leader beacon. Breaks between line segments indicate intervals for which no corresponding leader-beacon records were present in the receiver-side CSV files. 59

13 Collision outcomes as a function of attack initiation time and injected communication delay. Fixed settings: CACC controller, four vehicles, 5 m spacing, initial speed of 80 km/h, and leader braking initiated at $t = 17$ s with a prescribed acceleration of -5 m/s^2. Each attack lasted 6 s. For every delay value, the same 60 attack initiation times were tested, from 16.0 to 21.9 s in steps of 0.1 s, giving 420 runs in total. Rows group tested initiation times with identical outcomes; the displayed ranges do not imply testing at intermediate times between the 0.1 s samples. 62

List of Tables

2 Information and control flow in the hierarchical longitudinal control architecture. 17
3 Comparison between VeReMi Extension and the dataset generated in the present ComFASE experiments. 30
4 Summary of conducted simulation experiments (campaign) 42
5 Configuration parameters for the simulation analysis. 45
6 Propagation of braking disturbance under increasing delay attack values 49
7 Propagation of braking disturbance under increasing delay values at 80 km/h . . 52
8 Propagation of braking disturbance under increasing delay values at 100 km/h . 52
9 Propagation of braking disturbance for different platoon sizes under identical simulation conditions 52
10 Collision outcomes for injected delays of 100–500 ms. Fixed settings: CACC controller, four vehicles, 5 m spacing, initial speed of 80 km/h, and leader braking initiated at $t = 17$ s with a prescribed acceleration of -5 m/s^2. Each attack lasted 6 s. For each delay, the same 60 attack initiation times were tested from 16.0 to 21.9 s in steps of 0.1 s, giving 300 runs in total. 61
11 Collision occurrence at different injected delays, with the evaluated configuration sets shown explicitly. 62
12 Collision occurrence for different spacing settings at an injected delay of 500 ms in eight-vehicle platoons. 63
13 Collision occurrence for different leader braking accelerations at an injected delay of 500 ms in eight-vehicle platoons. 64
14 Collision occurrence for different CACC platoon sizes at an injected delay of 500 ms. 65
15 Collision occurrence at two initial leader speeds for eight-vehicle platoons subjected to an injected communication delay of 500 ms. 65
16 Overall collision comparison between CACC and PLOEG controllers 66
17 Columns included in the beacon-level dataset. 66
18 Configuration of the CACC braking campaign with beacon-level data collection. 68
19 Model variants according to the source of input data. 69
20 Methods compared for each prediction horizon. 71
21 Collision-detection results at the event level. 72

1 Introduction

Connected and Automated Vehicles (CAVs) are now of interest to the logistics industry, vehicle industry, and other industries, because they are a key element of modern Intelligent Transport Systems (ITS), which aim to improve road safety, traffic efficiency, and sustainable transport development thanks to advanced communication and automation technologies. CAV-based systems can improve fuel efficiency, traffic flow, and even traffic safety [1]. Vehicle platooning refers to tightly coupled, coordinated groups of synchronized CAVs. Within a platoon, CAVs exchange data such as: speed, acceleration, position and steering angle with the platoon leader to maintain required distance and to track the leader's steering angle. In platooning, a group of CAVs follows the longitudinal and lateral movements of a Lead Vehicle (LV) using vehicle-to-vehicle (V2V) communication and onboard sensors [1]. Due to lower aerodynamic drag (resistance), vehicles traveling in a platoon behind the platoon leader or Leading Vehicle (LV) at a specified distance consume less fuel and thus reduce CO_2 emissions to the atmosphere. Platoon driving has the potential to reduce carbon emissions, which is one of the goals of Agenda 2030. According to Lammert et al., Tsugawa et al. demonstrated that a three-truck platoon cruising at 80 km/h (50 mph) with a 10-meter inter-vehicle gap achieved an average fuel consumption reduction of 14%, while the lead truck reduced its fuel use by 7.5% and the last truck achieved a 16% reduction [2]. Shorter inter-vehicle gaps can contribute to improving road efficiency by allowing more vehicles in the lane [1]. Reducing human errors can also lead to improved road safety. Control systems react faster than a human driver. In the event of an unexpected hazard, such as the occurrence of a person or animal on the road, they are able to react more smoothly and in a coordinated manner.

To enable longitudinal and lateral controls, vehicles use onboard sensors and V2V communications [1]. Vehicle platooning relies on V2V communication to maintain distances between vehicles and coordinate longitudinal movement. Effective platoon operation requires using V2V communication between vehicles and information collected by radar sensors. This allows vehicles to dynamically respond to changing traffic conditions and synchronize their maneuvers with other members of the platoon. Despite the potential advantages of platooning, maintaining safe and stable cooperation remains a significant challenge. Since platoons are strongly dependent on reliable and low-latency wireless communication, communication outages, delays, and packet losses or malicious interference can negatively affect coordinated driving behavior. Because CAVs in a platoon travel with short inter-vehicle gaps, transient communication outages can negatively impact coordinated braking and string stability. To be effective, the platoon should be resilient to such communications outages. If communication is interrupted or manipulated, it creates road safety threats, because driving control systems rely on timely and accurate data to make driving decisions.

The benefits of platooning are associated with strict requirements on coordinated vehicle con-

trol and periodic information exchange. Cooperative platooning systems combine information obtained from onboard sensors with information exchanged between vehicles in a platoon. Cooperative Adaptive Cruise Control (CACC) can use vehicle-state information from beacon messages communicated by other platoon members as an input to longitudinal control. Therefore platoon operation depends not only on the performance of vehicle controllers and sensing systems, but also on the availability and correctness of communicated information. The dependence on V2V communication introduces additional safety and cybersecurity challenges. Communication outages, packet losses, interference, communication delays, or data manipulation can affect the state information available to cooperative vehicle controllers. A controller receiving outdated information may therefore calculate its response using data that no longer represent the current state of the platoon. Communication delay is particularly relevant because a delayed message may contain valid, but outdated, information while representing an earlier state of another vehicle.

Previous studies investigated transient communication outages in V2V communication [1, 3]. Some research on delay attacks in platoons has been conducted [4, 5]. However, to the best of my knowledge, fewer studies have examined how delay attacks affect vehicle platoons during emergency braking, leaving a gap that this thesis aims to address. The state of the art includes several studies and frameworks addressing cooperative platooning, communication reliability, and communication-related attacks. Segata et al. developed PLEXE as a framework for simulating platooning and provided implementations of platoon control strategies, including CACC controllers based on the approaches proposed by Rajamani and Ploeg [6, 7, 8]. Hasan investigated safe degradation and transient communication outages in vehicle platoons, demonstrating the importance of communication availability and cooperative awareness for maintaining safe platoon operation [1]. Building on the PLEXE simulation environment, Malik et al. introduced ComFASE, a communication-based fault and attack simulation tool that enables the injection of communication faults and attacks, including delay attacks [4]. Furthermore, the VeReMi dataset was introduced to provide datasets for the development and evaluation of misbehavior detection mechanisms in VANETs [9].

The Goal of this thesis is to understand the effects of delay attacks on platoon braking in terms of safety and efficiency. Significant emphasis is placed on investigating vulnerabilities in vehicle platooning, analyzing delay attack on the communication channel, and evaluating its consequences. In the experimental part, the author simulates different platoon scenarios with various levels of delay attacks. This gives a better understanding of the effects of delay attacks on platoon behavior in terms of safety and efficiency. Conducted experiments provide insight into the effects of delay attacks on platoon behavior in terms of safety and efficiency. The analysis of the impact of delay attack on Connected and Automated Vehicles (CAVs) in vehicle platooning systems during braking is the academic contribution of this thesis. Additional contribution of this study is the assessment of system performance under delaying attack and a comparison of

the effectiveness of two widely adopted CACC control models in such scenario.

Generated datasets and results indicate that communication delays can significantly affect platoon safety and that the impact depends on both the controller architecture and the platoon configuration. The results also indicate that the timing of the attack is a critical factor, as delays introduced immediately before an emergency braking maneuver may have significantly more severe consequences than delays occurring during normal platoon operation. However, attack effectiveness depends jointly on both delay magnitude and attack timing. Further details are provided in the following chapters, where the presented experiments also provide evidence supporting these claims.

In summary, this thesis investigates the impact of communication delay attacks on platoon safety during emergency braking. In the experimental part of the research simulation campaign was conducted using ComFASE, a communication fault and attack simulation engine built on top of OMNeT++, SUMO, Veins, and PLEXE. Different delay values, braking intensities, platoon sizes, inter-vehicle distances, leader speeds, and controller architectures were systematically evaluated in order to answer the research questions presented in the Problem Formulation chapter. The study provides a vulnerability assessment of platooning systems and compares the robustness of Rajamani's and Ploeg's CACC controllers under artificially injected communication delay attacks. The datasets generated in this study lay the foundation for a possible future standardized benchmark and corresponding ground truth to evaluate communication delay attacks in cooperative vehicle platooning systems. The contribution of this thesis is the evaluation of the impact of the delay attack in different platoon scenarios and the generation of datasets that support further research on the communication security of CAVs.

Section 2 explains the key concepts related to cooperative driving in vehicle platoons and introduces the simulation frameworks used in the experimental workflow. Section 3 reviews previous research on security in VANETs, with particular emphasis on the ComFASE tool and the VeReMi datasets. Section 4 introduces the research questions addressed in this thesis, while Section 5 describes the adopted research methodology. Section 6 covers ethical considerations. Section 7 describes the execution of the experimental campaign, including the experimental workflow and simulation parameters. Section 8 reports the results of the experimental campaign, and Section 9 discusses their significance. Finally, Section 10 presents the conclusions, while Section 11 outlines possible directions for future research.

2 Background

This section introduces the theoretical foundations and concepts related to Connected and Automated Vehicles (CAVs) that are relevant to the research conducted in this thesis. Then it presents longitudinal control models for CAVs, string stability in vehicle platoons, and the simulation platform with frameworks used in platooning research.

2.1 Vehicular Ad Hoc Networks (VANETs)

Vehicular Ad Hoc Networks (VANETs) are mobile, wireless ad hoc networks supporting communication between vehicles (mobile nodes) and between vehicles and roadside infrastructure. VANETs differ from cellular technologies because they do not require base stations to ensure communication [10, p. 5]. Communication is carried out directly between nodes within their possible communication range. VANETs support safety and traffic-management applications, such as accident notifications, congestion monitoring, and alternative-route recommendations [11]. VANETs enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. These types of communication form part of the broader vehicle-to-everything (V2X) concept.

2.2 Connected and Automated Vehicles (CAVs)

Connected and Automated Vehicles (CAVs) facilitate connectivity between automated vehicles through inter-vehicle communications. A vehicle is considered connected if it is equipped with advanced communication technologies, such as Dedicated Short-Range Communication (DSRC) - in Europe deployed as ITS-G5 - and has V2X radiocommunication capabilities that enable cooperative V2X connectivity and support Advanced Intelligent Transportation Systems (ITS) applications [12, 6]. An Automated Vehicle (AV), in contrast, relies on on-board sensors, such as radar, cameras, and LiDAR, to support automated driving functions. A Connected and Automated Vehicle (CAV) therefore combines automated driving capabilities with cooperative V2X connectivity. A vehicle uses both on-board sensors and communication with other vehicles, road infrastructure, pedestrians, roadside units (RSUs), and other infrastructure elements. Radar and LiDAR allow it to independently perceive its surroundings and have the so-called perception. This facilitates cooperative car-following and coordinated manoeuvres between vehicles. To provide a standardized definition of vehicle automation, International Society of Automotive Engineers (SAE International) distinguishes six levels of driving automation.

2.3 IEEE 802.11p

Currently, several wireless vehicular communication technologies and standards are available, providing different communication ranges and supporting different application domains. IEEE

802.11p is used as a basis for wireless, ad hoc, vehicular communication [13]. In Europe, ITS-G5 is a vehicular communication protocol stack developed on top of IEEE 802.11p [14]. Several V2V communication technologies can support platooning, including IEEE 802.11p, C-V2X, and 5G NR-V2X. This thesis focuses on IEEE 802.11p-based V2V communication, as it is the communication technology implemented in the considered simulation environment. In IEEE 802.11p-based platooning systems, V2V communication enables CACC controllers to exchange vehicle state information between platoon members. Since tightly coupled platoons operate with short inter-vehicle gaps, delayed communication may significantly reduce the available reaction time during emergency braking maneuvers.

The IEEE 1609.4 operates at the MAC layer and provides mechanisms for coordinating upper-layer data transfers across multiple wireless channels while the IEEE 802.11p standard is used for Inter-Vehicle Communication (IVC) between vehicles in the Vehicular Ad Hoc Network (VANET) [15]. It is not the only standard, but it is a commonly adopted standard for Vehicular Ad Hoc Networks (VANETs) and Vehicle-to-Vehicle (V2V) communication. It is also the basis for wireless V2V communication implementation in vehicular safety communications simulator frameworks such as Veins.

IEEE 802.11p is a vehicular extension of the conventional IEEE 802.11 standard, developed to support low-latency communication in VANETs. It was designed specifically for VANETs, ITS, V2V and V2I communications. IEEE 802.11p specifies modifications to the IEEE 802.11 PHY and MAC layers to support communication in vehicular environments. One of the main characteristics of IEEE 802.11p is its ability to support direct broadcast communication between vehicles in vehicular ad hoc networks (VANETs) without the need to create a classic Wi-Fi network. In IEEE 802.11p, CSMA/CA uses carrier-sensing information from the physical layer to determine whether the wireless medium is busy or idle before attempting transmission. If the medium is busy, the transmission is deferred until the channel becomes available. This mechanism allows multiple vehicles to coordinate access to the shared wireless medium and reduces the probability of simultaneous transmissions [16]. IEEE 802.11p and the IEEE 1609.x protocol family form the wireless access in vehicular environments (WAVE) architecture, while CSMA/CA is the medium access mechanism used at the MAC layer of IEEE 802.11p [17].

For road safety reasons, time and fast flow of information transmission in the Vehicular Ad Hoc Network (VANET) is crucial. In the context of platooning, CSMA/CA is relevant because vehicles share the wireless medium and channel access is not instant. When the medium is busy, a transmission is deferred, which can contribute to communication latency. Since CACC relies on timely V2V information exchange for coordinated vehicle control, communication delays can result in vehicles reacting to outdated information. This makes communication delay particularly relevant to the safety and stability analysis performed in this thesis.

2.4 String stability

In his book Rajesh Rajamani defines string stability as the ability of a vehicle platoon to prevent disturbances from becoming larger as they propagate through the platoon [18]. This means that when the leading vehicle changes speed (accelerates or brakes), the following vehicles should respond to that in a coordinated way without amplifying spacing errors. Rajamani describes that the platoon loses string stability if speed or acceleration errors amplify as they propagate upstream or downstream in a string of CAVs [18]. Ploeg et al. characterize string stability as propagation of states over subsystems [7, p. 261].

2.5 Longitudinal Control Models

A vehicle controller is an algorithm operating within a closed-loop control system that processes input information from onboard sensors and, in connected vehicle systems, from V2V communication to determine appropriate actuation commands, such as acceleration, braking and steering. In platooning systems, controllers use information gained from the ego vehicle and other vehicles in the platoon to regulate longitudinal and lateral vehicle motion and maintain the desired inter-vehicle spacing. In platooning, controllers are responsible for determining vehicle acceleration and braking based on local measurements and data transmitted over VANETs and V2V communication, allowing the maintenance of desired inter-vehicle distances, platoon synchronization and string stability. A longitudinal controller's role is to adjust and maintain ego vehicle's speed and regulate gap between vehicles, while a lateral controller controls lateral movement: steering, lane-keeping, and overtaking [1].

2.5.1 Constant Time Gap

Constant Time Gap (CTG) is a spacing control strategy that maintains a constant time gap between successive vehicles. This gap is defined as the time interval between the rear bumper of the preceding vehicle and the front bumper of the following vehicle passing the same reference point on the road. As a result, the inter-vehicle distance, expressed in meters, varies with vehicle speed [1].

2.5.2 Constant Distance Gap

Constant Distance Gap (CDG) is a spacing control strategy that maintains a constant distance between two adjacent vehicles regardless of their speed. Consequently, the inter-vehicle distance, expressed in meters, remains constant, while the corresponding time gap varies with vehicle speed [1].

2.5.3 Adaptive Cruise Control (ACC)

ACC is a classic autonomous controller that utilizes, among other things, radar and/or local sensors. Ego vehicle's onboard radar detects the preceding vehicle and provides measurements of the inter-vehicle distance and relative speed. Based on this information, the ACC controller regulates the ego vehicle's acceleration to maintain the desired spacing and speed. If the ego vehicle is a platoon leader so there is no vehicle ahead, the controller uses Cruise Control (CC). It is an autonomous system, independent of V2V communication. Classical ACC relies on onboard sensor measurements for longitudinal control and does not require V2V communication. When no preceding vehicle is detected, the ACC system behaves similarly to conventional CC by maintaining the desired speed. ACC is described by the formula: [6]

$$\ddot{x}_{i,\text{des}} = -\frac{1}{T}\left(\dot{\epsilon}_i + \lambda\delta_i\right) \tag{1}$$

$\ddot{x}_{i,\text{des}}$ is the desired acceleration computed by the controller. T is the expected time headway between vehicles. $\dot{\epsilon}_i$ is the relative speed between the vehicle and the preceding vehicle. λ is a parameter defining the strength of the controller's reaction and the last parameter (δ_i) is the gap error, which is the difference between the actual and expected distance. At short time gaps, ACC-controlled vehicle strings may amplify disturbances as they propagate from one vehicle to the next upstream through the vehicle string [19].

2.5.4 Cooperative Adaptive Cruise Control (CACC)

CACC is an extension of ACC. ACC uses on-board sensors such as radar to measure the range to the preceding vehicle and the relative velocity. CACC expands this with V2V communication. A vehicle control system no longer needs to rely solely on its own sensors, as it can receive information about the braking manoeuvres of the preceding vehicle and the platoon leader before its own sensors detect the resulting changes in motion. This increases system situational awareness. In CACC an ego vehicle acquires data such as speed, acceleration and position through periodic V2V beacon messages. The Rajamani [8] controller uses information from both the platoon leader and the preceding vehicle, whereas the Ploeg [7] controller primarily relies on information received from the preceding vehicle. CACC is primarily used for automatic longitudinal control, i.e., maintaining speed and gap. CACC can utilize both CTG and CDG control strategies [20]. PLEXE framework, which is part of the research environment used in this thesis, implements (inter alia) two CACC controllers based on the control approaches proposed by Rajamani et al. and Ploeg et al. [6, 8, 7]. Therefore, in this work, thesis author focused on evaluating their security and resistance to delay attacks. Their presence in the PLEXE tool gives the opportunity to test these popular CACC controllers on many levels in different scenarios.

2.6 CACC Controllers

The type of longitudinal controller used in the platooning simulations is important and influences the research results, for example, the number of collisions. The Rajamani and Ploeg controllers rely on data transmitted wirelessly through V2V communication. [8] [7] Therefore, communication delay affects whether string stability is maintained and whether braking is synchronized. If the controllers do not have up-to-date values for all components of their control equations, their response depends on how missing messages are handled in the implementation. If outdated data are used, the desired acceleration is calculated based on an outdated vehicle state. This can degrade synchronized braking and string stability and, at small inter-vehicle gaps, can also increase the risk of collisions.

2.6.1 Rajamani's CACC controller

CACC controller by Rajamani et al utilizes information from both the preceding vehicle and the platoon leader. It is referred to Leader-Predecessor-Following CACC (LPF-CACC) [1]. Figure 1 illustrates the LPF communication topology typically associated with Rajamani's CACC controller.

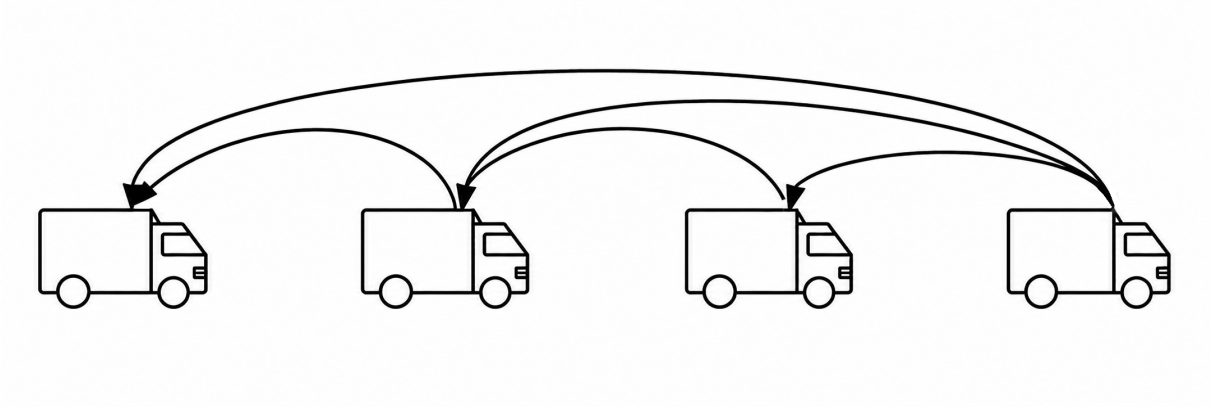

Figure 1: Schematic representation of a CACC-equipped vehicle platoon based on the Leader Predecessor-Following communication topology.

In the PLEXE implementation of the Rajamani CACC, information about the speed and acceleration of the leader and preceding vehicle is received via V2V communication, while the distance to the preceding vehicle is measured by radar [6]. Rajmani's longitudinal control system consists of two closely cooperating controllers: upper level and lower level controller [18]. The upper level controller determines the desired acceleration for each vehicle participating in the platoon. The lower level controller uses this data to determine throttle and/or brake commands. Then actuators issues these commands in order to keep the right acceleration profile. Table 2 illustrates the controller's operation scheme.

Rajamani's controller extends the traditional ACC by utilizing information about the acceleration of the vehicle in front and platoon leader transmitted via V2V communication, meaning it can

Table 2: Information and control flow in the hierarchical longitudinal control architecture.

Step	Component or signal	Function
1	Radar and V2V data	The onboard radar provides the inter-vehicle spacing, while V2V communication provides speed and acceleration from leader and predecessor.
2	Upper-level controller	Processes the available sensor and communication data and calculates the desired acceleration of the i-th vehicle.
3	Desired acceleration $\ddot{x}_{i,\text{des}}$	Represents the acceleration or deceleration that the i-th vehicle should achieve to maintain the desired spacing and coordinated platoon motion.
4	Lower-level controller	Translates the desired acceleration into the throttle and/or brake commands required to track it.
5	Throttle and brake actuators	Apply the control commands and determine the actual longitudinal motion of the vehicle.

receive information about acceleration changes before the sensors detect the change in distance and/or relative velocity. The two principal objectives of the upper-level controller are to maintain a constant small spacing between consecutive vehicles and to ensure the string stability of the platoon [8, pp. 696–697]. For the ego vehicle, the upper-level longitudinal control law is defined as follows:

$$\begin{aligned}\ddot{x}_{i,\text{des}} = {} & (1 - C_1)\ddot{x}_{i-1} + C_1\ddot{x}_\ell \\ & - \left[2\xi - C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\right]\omega_n\dot{\varepsilon}_i \\ & - \left(\xi + \sqrt{\xi^2 - 1}\right)\omega_n C_1\,(v_i - v_\ell) \\ & - \omega_n^2\varepsilon_i.\end{aligned} \tag{2}$$

where $\ddot{x}_{i,\text{des}}$ is the desired longitudinal acceleration of the ego vehicle [8, p. 695]. $\ddot{x}_{i-1}$ is the longitudinal acceleration of the preceding vehicle [8, p. 695]. C_1, ξ, and ω_n are control gains to be tuned in the upper-level longitudinal controller [8, p. 697]. These are often specified at the design stage of the controller. Parameter C_1 represents relative weight for platoon leader signal feedback compared to preceding vehcile signal feedback [8, p. 697]. Each vehicle in platoon receives acceleration information from the preceding car and the lead car via wireless communication. $\ddot{x}_\ell$ means longitudinal acceleration of the lead vehicle of the platoon [8, p. 695]. The longitudinal spacing error of the i-th vehicle is defined as $\varepsilon_i = x_i - x_{i-1} + L$, with L being the desired spacing [8, p. 695]. The variable v_i denotes the longitudinal velocity of the$i$-th vehicle, while v_ℓ denotes the longitudinal velocity of platoon leader [8].

The controller does not need to rely exclusively on radar information about relative distance to the preceding vehicle. It can directly use the communicated acceleration of the preceding and

lead vehicles when determining its desired longitudinal acceleration, $\ddot{x}_{i,\text{des}}$. This means that, under ideal wireless communication conditions, information about the lead vehicle's braking may reach, for example, the seventh or eighth vehicle in the platoon before its radar detects a change in the distance to the predecessor.

Intuitively, the control model proposed by Rajamani et al. appears to have greater communication dependency because vehicles located near the tail of the platoon may be more vulnerable to communication delays or temporary communication outages affecting data transmitted by the platoon leader. However this is more nuanced matter and therefore discussed in greater detail in Sections: 7, 8, 9.

In summary. The platoon leader provides its speed and acceleration through V2V communication, whereas the predecessor provides its acceleration. The ego vehicle locally measures the distance and relative speed to the predecessor using its onboard radar. These inputs are combined by the Rajamani controller to compute the desired acceleration.

2.6.2 Ploeg's CACC controller

Figure 2 illustrates a platoon using CACC proposed by Ploeg et al. [7]

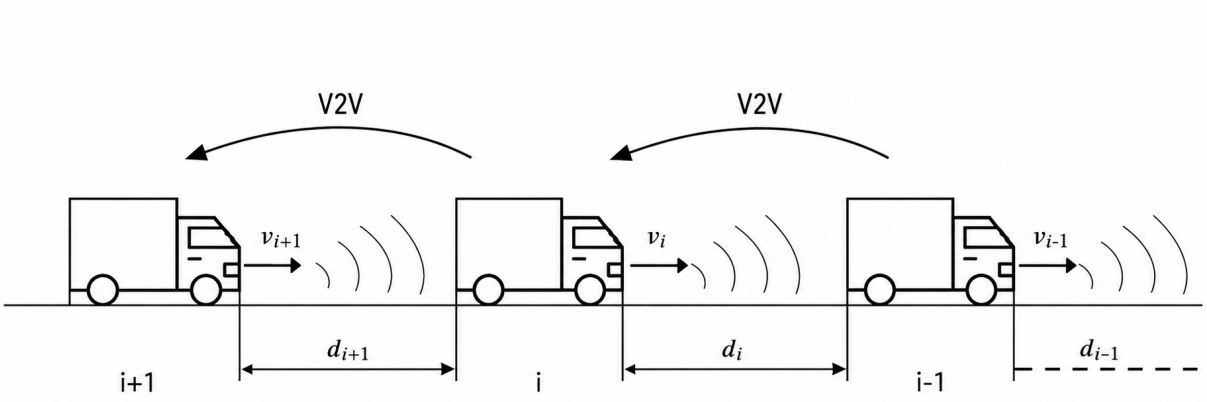

Figure 2: Schematic representation of a CACC-equipped vehicle platoon based on the predecessor-following communication topology.

Unlike Rajamani's LPF-CACC controller, Ploeg's controller utilizes a Constant Time Gap (CTG) policy and Predecessor-Following-Cooperative Adaptive Cruise Control (PF-CACC), which means the distance between vehicles depends on vehicle speed [7]. This means with the increase of speed the inter-vehicle gaps in meters increase as well, while the time gap (in seconds) remains constant. The disadvantage of CTG is that at high speeds the platoon becomes more stretched and inter-vehicle gaps increase. On the other hand, this solution may improve

platoon safety in the event of delays or communication loss. The Ploeg controller therefore does not require direct communication with the platoon leader. However, larger distances between vehicles reduce fuel efficiency. Ploeg et al. consider a platoon of m vehicles, where d_i represents the distance between vehicle i and its preceding vehicle $i-1$, and v_i is the velocity of vehicle i. Each following vehicle should maintain the desired distance $d_{r,i}$ from its preceding vehicle. For this purpose, the controller employs a constant-time-headway spacing policy, according to which the desired distance is defined as [7, p. 261]:

$$d_{r,i}(t) = r_i + hv_i(t), \qquad 2 \leq i \leq m, \tag{3}$$

where h is the time headway and r_i denotes the standstill distance [7, p. 261]. The desired distance increases proportionally to the velocity of vehicle i. Since the vehicle string is assumed to be homogeneous, the same time-headway value h is applied to all following vehicles.

The spacing error $e_i(t)$ is defined as the difference between the actual distance $d_i(t)$ and the desired distance $d_{r,i}(t)$:

$$e_i(t) = d_i(t) - d_{r,i}(t). \tag{4}$$

Because the actual distance is given by $d_i(t) = s_{i-1}(t) - s_i(t) - L_i$, the spacing error can be written as:

$$\begin{aligned} e_i(t) &= d_i(t) - d_{r,i}(t) \\ &= [s_{i-1}(t) - s_i(t) - L_i] - [r_i + hv_i(t)], \end{aligned} \tag{5}$$

where $s_{i-1}(t)$ and $s_i(t)$ denote the longitudinal positions of the preceding and following vehicles, respectively, and L_i denotes the length of vehicle i. A value of $e_i(t) = 0$ indicates that the actual distance is equal to the desired distance.

As mentioned, the time interval remains constant, but the actual distance in meters changes with speed. For example: with the time interval value defined in Veins as PloegH (or just H) = 0.5 s and a speed of 100 km/h the distance is approximately 13.9 meters. $100\,\mathrm{km/h} \approx 27.8\,\mathrm{m/s}$ and $27.8 \times 0.5 \approx 13.9\,\mathrm{m}$ This is because according to Ploeg's formula:

$$d_{r,i} = r_i + h_{d,i} \cdot v_i \tag{6}$$

where:

- $d_{r,i}$ is the desired inter-vehicle distance,
- r_i is the standstill spacing,
- $h_{d,i}$ is the desired headway time,

- v_i is the velocity of vehicle i.

2.7 Simulation Platforms for Platooning and Security Analysis

2.7.1 OMNeT++

OMNeT++ is an open source, C++-based, discrete event simulation library and framework in modular and object-oriented architecture [21]. It can simulate complex IT system models or hardware architectures. It is a generic framework and can therefore be applied across different problem domains, however, it is primarily used for building network simulators and for modelling communication networks and protocols. OMNeT++ can model networks and different wireless communication protocols [22]. Over the years, it has gained wide recognition as a platform for simulating computer networks thanks to its modular architecture, wide range of extensions, and open-source licensing. One of these extensions is Veins [23]. OMNeT++ alone does not provide complete mechanisms for modeling VANETs but thanks to the possibility of integration with frameworks such as Veins and SUMO it enables conducting advanced research on communication between vehicles.

OMNeT++ provides two main execution environments: Cmdenv and Qtenv. Cmdenv is a command-line interface (CLI) for running simulations in terminal with no graphical interface. Qtenv is a graphical user interface (GUI) that enables monitoring and interacting with simulations in real time.

2.7.2 SUMO

Simulation of Urban Mobility (SUMO) is an open source software for road traffic modeling. It was developed by employees of the Institute of Transportation Systems at the German Aerospace Center [24]. SUMO helps simulate vehicle traffic on roads, intersections and traffic lights, acceleration, braking, lane change, public transport, pedestrians and cyclists, large road networks (from a single intersection to entire cities), etc. [25]. As a microscopic simulator, SUMO models the movement and behavior of individual vehicles, which enables realistic simulation of vehicle dynamics and large-scale road traffic networks [24]. SUMO also supports bidirectional coupling with external tools, such as OMNeT++, allowing vehicle mobility and communication networks to be simulated together. This functionality is particularly important in this thesis, where SUMO provides the vehicle mobility and dynamics used by the platooning simulations. SUMO allows to import real maps and recreate realistic traffic scenarios and visualize them. SUMO also allows real maps to be imported in order to recreate and visualize realistic traffic scenarios. It also provides a user friendly graphic user interface (GUI).

2.7.3 Veins

Veins is an open-source simulation framework designed especially for VANETs research [26]. It integrates OMNeT++ with SUMO, allowing simulation of V2V communication and vehicle traffic simulations. Veins is a vehicular network simulation framework developed on top of OMNeT++ that provides a communication stack based on IEEE 802.11p [26, 23]. OMNeT++ and SUMO are connected through a TCP socket using the Traffic Control Interface (TraCI), which enables the exchange of mobility information and synchronization between the two simulation environments [27]. Information exchange between both simulators is performed via the TraCI. The Veins framework provides VANETs communication models and tools for visualization and analysis of simulated but realistic traffic scenarios. This makes Veins an excellent environment for research on Intelligent Transportation Systems (ITS).

2.7.4 PLEXE

PLEXE is a platooning extension for Veins [6]. While Veins already facilitates cooperative driving, PLEXE provides additional functionalities specifically for platooning, including platoon control, platoon formation, and longitudinal and lateral maneuvers. It introduces and integrates a platoon driving framework into SUMO and Veins. It uses both SUMO and Veins to simulate vehicle communication within a platoon and their movement in road traffic. PLEXE integrates the different control models used for platooning. Plexe is also an Open Source framework.

2.7.5 ComFASE

ComFASE is an open-source simulation framework that enables the injection of faults and malicious attacks into the communication channels of Connected and Autonomous Vehicles (CAVs) [4]. The Authors describe ComFASE as simulation engine capable of injecting various cyberatacks on wireless communication channel messages exchanged between CAVs in a platooning application. [4] It facilitates the evaluation of the behavior of CAVs in different cyberattack scenarios. ComFASE is a tool for testing and automating multiple attacks on platooning. It runs on top of a collection of open-source simulators. These are: Plexe (a cooperative driving framework), Veins (a vehicular network simulator), OMNeT++ (a network simulator) and SUMO (Simulation of Urban MObility) [4]. Some ComFASE functionalities and their practical use are described in more detail in other chapters (see section 3 and section 6).

ComFASE implements various attacks: Destructive Interference Attack, Barrage jamming Attack, Deceptive Jamming Attack, Delay Attack and DoS (Denial-of-Service) Attack. Malik et al.'s study results involve Delay Attack and DoS-attack [4]. Authors classify the results using quantitative parameters: deceleration profiles of all vehicles in the platoon and collision incidents [4]. ComFASE engine is provided together with handy Python scripts that allow data parsing and plotting. ComFASE logs simulation data from both SUMO and Veins. These

are very detailed traffic simulation information like vehicle IDs, acceleration profiles, collision times, attack start/end times etc.

3 Related work

Safety of CAVs depends on multiple interrelated factors of which communication security is important but not exclusive. This chapter reviews the current state of research concerning the safety of CAV platoons and the security and reliability of the V2V communication on which coordinated platooning depends. Particular attention is given to communication delays, jamming attacks, temporary information unavailability, and their potential effects on platoon safety, braking and string stability. Together with the Introduction and Background sections, this section establishes the state of the art and positions the present study in relation and with respect to the existing literature.

3.1 V2V Communication Security

Both american DSRC and european ITS-G5 employ physical access (PHY) and Media Access Control (MAC) method based on IEEE 802.11p, which is inherited from IEEE 802.11a [1, p. 20].

In the european Cooperative Intelligent Transport Systems (C-ITS) architecture, security header and certificate formats are specified in document: ETSI TS 103 097 [28]. This standard and its architecture adapts security mechanisms and data structures described in IEEE 1609.2 [28, p. 4]. According to ETSI TS 103 097 security process starts before transmission [29, p. 27]. An entity can request communication security and send data to the security service [29, p. 27]. Then data is returned as either: unsecured data or encrypted data or signed data. The subsequent transmission including network and transport layer encapsulation and MAC frame generation, is not specified by the security standard itself [29, p. 27].

What ETSI TS 103 097 specifies is that encrypted data means payload encryption by Advanced Encryption Standard (AES) with 128-bit keys (AES-128) which is also described in IEEE 1609.2 [28, p. 7]. Then AES-key is encrypted by ECIES using the recipient's public key. The public key encryption algorithm is applied using Elliptic Curve Integrated Encryption Scheme (ECIES) [28, p. 7]. However, the document states clearly that signed data should only be signatured using Elliptic Curve Digital Signature Algorithm (ECDSA) [28, p. 11]. As shown, this standard supports both data encryption and digital signing using private keys associated with Public Key Infrastructure (PKI) certificates.

ETSI TS 103 also points out that Cooperative Awareness Messages (CAMs) so beacon messages broadcasted by CAVs shall contain signed data [28, p. 12]. So usually CAMs are signed but their payload is not encrypted. Contents of CAMs can therefore be observed by receivers within communication range. This creates attack opportunities related primarily to eavesdropping. Additionally, the author suggests that access to unencrypted CAM data may allow an attacker to determine when a safety-critical manoeuvre is about to begin and such knowledge may be critical

to the effectiveness of an attack. This is also discussed in Section (Results and Discussion and Conclusion)

3.2 ComFASE: A Tool for Evaluating Communication Faults and Attacks on Vehicle Platoons

Still it is not evident how platoon behave as a whole while exposed to a delay attack in different traffic scenarios. Malik et al. conducted an extensive campaign comprising 11,250 experiments involving delay attacks using ComFASE tool they developed [4]. The attack initiation times were selected within a single oscillation cycle, between two peaks in the leader's acceleration profile, and varied from 17.0 to 21.8 seconds in increments of 0.2 seconds. For each initiation time, the attack duration was varied from 1 to 30 seconds in increments of 1 second [4]. Their results indicated that attacks lasting longer than 5 seconds were always classified as either benign or severe [4]. Moreover, the first seconds of the attack were particularly important, while extending the attack duration beyond 4 seconds did not produce a significant additional effect on the number of severe cases [4]. However, these findings applied only to the Sinusoidal Scenario. What was particularly relevant to the research questions addressed in this thesis was whether similar effects could also be expected in the Braking Scenario and which parameters had the greatest influence on the results obtained by Malik et al.

Following an experimental campaign involving delay and denial-of-service attacks in a Sinusoidal Platooning Scenario, Malik et al. emphasized the importance of the attack initiation time, attack duration, and vehicle driving pattern in causing collision incidents [4]. The findings of these scholars suggested that attack timing is an important factor determining the extent and scale of its effectiveness and severity. Both the attack initiation time and the operational state of the platoon were also important. As already mentioned knowing when a safety-critical manoeuvre is about to begin may be critical to the effectiveness of an attack. In this thesis author investigate if (and to what extend) results of Malik et al. research apply to Braking Scenario. Malik et al. findings were particularly important for formulating the third research question of this thesis and pointed towards highlighting the need to investigate the combined effects of simulation parameters, time and the relationships between them.

3.2.1 Jamming as a Possible Attack Vector against Platoon Communication

Ucar et al. highlighted that knowledge of the communication frequency used by the platoon may enable an attacker to jam the wireless channel. By transmitting interfering signals on this frequency, the attacker may prevent the platoon vehicles from maintaining reliable V2V communication [30]. This disrupts string stability and compromises the safety of CAVs. Ucar et al. proposed a hybrid security protocol combining IEEE 802.11p with Visible Light Communication (VLC) to improve the security and reliability of vehicle platoon communication. They empha-

sized the importance of redundant communication channels, as IEEE 802.11p communication may be vulnerable to jamming and disruption [30].

Jamming typically causes packet loss, which delays the availability of up-to-date data in V2V communication and may consequently threaten platoon safety. However, from the controller's perspective, the cause of the delay is secondary. Regardless of whether the delay is caused by reactive jamming, network congestion, or communication faults, the consequence for the CACC controller will be the same: it will operate using outdated information. This thesis focuses on the resulting impact of delayed data on platoon safety and stability rather than all possible implementations of such attacks in a real-world scenario. However, it is worth mentioning how they could be executed.

One possible approach is a mobile power-constrained reactive jamming attack using a drone flying above the platoon or a vehicle driving alongside the platoon. Reactive jamming leads to packet losses and communication disruptions, resulting in the controller operating on delayed and outdated state information. The term "reactive" means that the jammer emits an interference signal only after detecting a transmission from a legitimate vehicle. The objective is to prevent the receiver from successfully receiving and decoding packets. However, transmitted messages are not modified, falsified, or artificially injected in any way.

Reactive jamming can therefore be simulated as packet loss or burst packet loss. Alipour-Fanid et al. emphasize that CACC-based platooning systems are sensitive to jamming attacks and that CACC control models may be negatively affected by a reactive jammer [31]. Berger et al. state that reactive jamming is considered a sophisticated jamming technique [32, p. 107].

It is theoretically possible to jam or reactively jam beacons in V2V communication using a relatively inexpensive Software-defined radio (SDR) device such as HackRF One with PortaPack H2. HackRF One can potentially interfere with ITS-G5 beacon reception, because its transmit frequency range covers the relevant band. The effectiveness, however, would depend on the radio conditions. Reactive jamming is a more sophisticated technique that requires detecting a transmission and switching to transmit quickly enough to disrupt it. Theoretically, reactive-jamming attack could be implemented using a half-duplex device such as HackRF One. However, this remains an untested assumption that requires implementation and experimental validation.

3.2.2 Modelling of Reactive-Jamming Effects and Delay attack

It is important to distinguish between the attack implementation adopted in ComFASE and a real reactive jamming attack that can occur in a real traffic scenario. This distinction is reflected in the terminology used by Malik et al., who refer to the implemented attack as a *delay attack* [4]. ComFASE describes delay the attack as temporarily (for a limited time) blocking the transmission and reception of beacon messages associated with the specific node [4]. Delay can be implemented both on a specific node or a group of nodes. It is then possible to delay

every node in the platoon. When vehicle is selected as an attack target, its beacons are delayed before being delivered to the following vehicles. But in a physical wireless environment, reactive jamming would typically prevent jammed frames from being successfully decoded and cause packet loss rather than delaying the delivery of the same frames. Malik et al. suggest that a delay attack may be realized by first intercepting and storing a copy of a beacon and then using reactive jamming to prevent receiver from decoding the original beacon. The attacker can retransmit the copied message after a chosen time [4, p. 5]. This interception and retransmission process is an example of a possible physical attack. ComFASE simulates the Delay attack by increasing the message's propagation delay. It does not simulate the interception or retransmission steps.

Simulations conducted using simulation frameworks are subject to modelling limitations. Reactive jamming would typically cause the loss of an uncertain number of packets, depending on the attacker's capabilities and channel conditions. ComFASE abstracts this complex and potentially unpredictable packet behaviour by introducing a predefined communication delay. This is a deterministic representation of the unavailability of up-to-date information rather than a detailed physical model of packet loss caused by jamming. In other words, ComFASE models communication delays instead of packet loss. And this is correct and methodologically justified in terms of information available to the vehicle controller. Packet loss can potentially be simulated with communication delays, because both loss and delay result in the same thing: periodic unavailability of up-to-date information.

3.3 Delay Impact on Braking Platoons

Previous studies investigated communication outages [3], misbehavior [33], and delay attacks in VANETs [4].

Platoon safety was one of the research subjects investigated by Hasan [1]. Hasan's research focused on the safe degradation of platoons during transient communication outages and on emergency braking in hazardous situations. Although the research does not investigate deliberate delay attacks on platoons, it provides insight into transient communication outages and their impact on platoons. Communication outages and delay attacks are different phenomena; however, both can affect the timely availability of information exchanged through V2V communication. Hasan shows, inter alia, that when V2V communication quality becomes poor, a platoon can fall back to a radar-based ACC controller and require larger inter-vehicle gaps to maintain safety [1].

Khattak et al. showed that communication delays have a significant impact on CAVs string stability [34]. As delays increase, the variability of acceleration/deceleration increases, leading to less smooth driving and increasing the risk of dangerous situations. The negative impact of delays is particularly noticeable for vehicles further from the leader, where disturbances are amplified [34]. ACC and CACC were compared in the experimental part of Khattak et al.'s

research [34]. The experimental part of their research was conducted in real-world conditions. Results confirm that higher time gaps between vehicles and the use of CACC systems can reduce acceleration volatility and thus improve stability within a platoon. Nevertheless, it was not analyzed how the results change for different platoon sizes, i.e. platoons with different numbers of vehicles.

Zhao et al. used a mathematical model and Simulink simulations, to investigate how non-ideal communication within a vehicle platoon affects longitudinal control [35]. Modeled platoons used a constant time gap (CTG) spacing policy under non-ideal communication conditions, including random packet loss and time-varying communication delays. In their experiment, the platoon was cruising when the leader suddenly started to brake. Communication delays were randomly selected from a range of 0-200 ms and all vehicles stopped without any collisions. The researchers then increased communication delays to a range of 1-10 seconds, which resulted in degradation of platoon performance. Under the same delay conditions, a collision occurred when the vehicles relied only on information from their predecessors [35]. Zhao et al. did not consider deliberate delay attacks or investigate how the timing of the delay during emergency braking affects collision occurrence. In this thesis, the author focuses on the timing of the attack and its impact on collisions, examining how different parameters influence their occurrence.

Thunberg et al. developed a mathematical model to determine a feasible region of communication delays guaranteeing safe emergency braking in a platoon [36]. While Thunberg et al. analytically determined a feasible region of communication delays for safe emergency braking, this thesis uses simulation experiments to investigate how different platoon and attack parameters, such as inter-vehicle distance, platoon size, and attack timing, influence collision occurrence under deliberately introduced communication delays.

3.4 Delay Attack Impact on Platoons

Van der Heijden et al. compared the resilience of different CACC controllers against different cyber attacks. The authors proposed an evaluation framework for simultaneously comparing attack effectiveness and controller resilience [37]. Van der Heijden et al. evaluated three CACC controllers: the constant-spacing controller proposed by Rajamani et al., the loss-tolerant controller proposed by Ploeg et al., and a consensus-based controller [37]. However, in their implementation of the experiment, Ploeg's controller could predict the acceleration of the preceding vehicle if the information was not received [37]. This feature is, however, absent in my experiment. They conducted four attacks: jamming (which de facto simulates packet loss, i.e., a DoS attack), position injection, speed injection, and acceleration injection [37]. Van der Heijden et al. noticed that the Ploeg controller dealt well with a jamming attack (DoS), while the Rajamani-based CACC was unreliable when subjected to DoS [37]. They also pointed out a broad window of opportunity for attackers [37]. The attacker does not necessarily need to

initiate the attack at a specific point in time. This was, however, demonstrated for the sinusoidal scenario in PLEXE, not for a braking scenario [37]. This thesis focuses primarily on artificially injected delay attacks on vehicle platoons with different parameters in a braking scenario. The sinusoidal scenario is only used as a supplementary scenario to support the braking analysis.

Zhang et al. showed that a delay/replay attack can lead to a collision in a CAV platoon, particularly when delayed information is received while the leader is decelerating [5]. They showed that both prediction compensation and CACC-to-ACC degradation can mitigate the effects of the attack, although prediction compensation was considerably more effective in their delay/replay attack scenario. Without compensation, the minimum time-to-collision (TTC) of CAV 1 (first vehicle behind the leader) decreased to nearly 0 s during the delay/replay attack, indicating an imminent collision. With prediction compensation, the minimum TTC increased to approximately 12 s. With degradation compensation, it increased to approximately 4 s. Therefore, prediction compensation provided a larger safety margin than degradation compensation. Zhang et al. attributed lower safety performance of degradation compensation to its lack of prediction capability, which caused it to respond too late when the dangerous situation developed rapidly [5].

To predict the leader's acceleration, Zhang et al. used a lightweight autoregressive integrated moving average (ARIMA) model. ARIMA predicts the leader's acceleration based on previous acceleration observations [5]. The predicted acceleration is then compared with the acceleration received through V2V communication. The resulting prediction residual is used for cyberattack detection. Their work focused on cyberattack detection and the evaluation of prediction and degradation compensation strategies, while this thesis focuses on the influence of delay attack and platoon parameters on collision occurrence.

Sun et al. presented a survey of cyber-security threats in VANETs [38]. They described a replay attack as recording and then retransmitting valid packets at a later time. While their work presents and systematizes potential cyber-security threats and attack vectors, this thesis experimentally investigates the effects of delayed V2V communication on platoon safety and stability through simulation.

3.5 VeReMi and VeReMi Extension datasets

Van der Heijden et al. observed that previous studies on misbehavior detection in VANETs relied on individually designed simulations, making it difficult to compare different detection mechanisms [9]. Before VeReMi, there was no publicly available reference dataset enabling comparable evaluation of VANET misbehavior-detection mechanisms that would allow testing and verifying results of other scholars [33]. To address this issue simulation-based vehicular communication scenarios dataset were generated using VEINS, OMNeT++, and SUMO frameworks. VeReMi datasets can be therefore treated as a ground truth for detecting misbehavior of CAVs in ITS. Authors emphasize that communication delays in VANET environments can result from network congestion or hardware limitations [33]. The VeReMi and the VeReMi Extension datasets introduce systematization of security research in VANET.

To generate VeReMi Extension dataset, Kamel et al. employed the Framework for Misbehavior Detection (F2MD), an extension of Veins developed to reproduce and detect different misbehavior-detection use cases in vehicular networks [33]. In VeReMi one record represents one specific beacon message. Each received-message record in VeReMi Extension is represented in JSON format using the following field names:

> `type`, `rcvTime`, `sendTime`, `sender`, `senderPseudo`, `messageID`, `pos`, `pos_noise`, `spd`, `spd_noise`, `acl`, `acl_noise`, `hed`, and `hed_noise`.

These fields serve different purposes in the misbehavior-detection process. The `messageID`, `sender`, and `senderPseudo` fields identify the message and its origin. The `sendTime` and `rcvTime` fields describe when the message was transmitted and received. They allow the observed latency of message (k) to be calculated as:

$$L_k = t_{\text{rcv},k} - t_{\text{send},k} \tag{7}$$

What is important for this thesis is that VeReMi provides time-series data at the level of individual V2V messages, but it does not address communication in CACC-controlled platoons. On the other hand ComFASE enables communication attacks and faults to be introduced into a platoon and their effects to be evaluated, but the standard experimental results are aggregated at the simulation-run level rather than at the individual-beacon-message level. Knowing the most recent beacon messages and the current state of the platoon can make it possible to predict whether a collision will occur within the next few seconds. This thesis however defines the complete workflow for emergency-braking scenarios with relevant simulation and attack parameters. Thesis author collects the resulting data and evaluates the effects of delayed V2V communication on platoon safety using collision occurrence, collision count, and collision rate (see Table 3).

Table 3: Comparison between VeReMi Extension and the dataset generated in the present ComFASE experiments.

Feature	**VeReMi Extension**	**ComFASE experiment in this research**
Subject of analysis	V2X messages and vehicular misbehavior	Platoon behaviour under communication delays
Unit represented by one record	One message reception event	One complete simulation run
Data resolution	Message level	Simulation-run level
Temporal information	`sendTime` and `rcvTime` for individual messages	Attack initiation and termination times and, where applicable, collision time
Vehicle-motion data	Position, speed, acceleration, and heading reported in each message	Maximum acceleration and minimum deceleration recorded for individual platoon vehicles
Ground truth	Information identifying genuine, malfunctioning, and misbehaving behaviour	Configured delay-attack parameters and observed collision outcome
Primary research question	Can anomalous or misbehaving message behaviour be detected?	Does the configured communication delay compromise platoon safety?
Role of CACC platooning	Not the primary application under investigation	Central application under investigation
Role of emergency braking	Not the primary scenario under investigation	Central traffic scenario
Primary evaluation outcomes	Accuracy, precision, recall, and F1 score	Collision occurrence, collision count, and collision rate

Datasets can be used by different Machine Learning models to evaluate how efficient such models are in detecting cyberattacks on platoons. Real-world tests are generally more expensive and time-consuming than software-based simulations. Malik et al. already tried to address this issue by presenting communication fault and attack simulation engine - ComFASE - which allows users to plan and carry out various threat and misbehavior assessment campaigns [4]. However, Malik et al. did not present the experimental results in the form of datasets, which could be helpful in evaluating the results and in training the ML models. Such results could be used for training ML models if publicly available. In the future, this approach may support the detection of injections, DoS, and delay attacks in connected vehicle platooning systems or even predict collisions based on a few beacon messages. Creating datasets for braking scenarios in vehicle platooning systems may support future research in this direction.

3.5.1 VeReMi NextGen dataset

Hermann et al. introduced VeReMi NextGen, a dataset for evaluating Misbehavior Detection Systems (MDSs) in VANETs - which is the latest(2026) release of VeReMi [39]. The dataset includes 15 attack types, also a Time Delay Attack, in which messages are sent with a temporal offset, delaying the reception of vehicle data. VeReMi models delayed messages as messages containing valid data but delivered with a delay relative to the current time. Their evaluation using the CaTch MDS showed low F1-scores for detecting the Time Delay Attack across all evaluated traffic scenarios. The authors suggest that the low F1-scores are due to CaTch not sufficiently assessing temporal consistency. This thesis also provides simulation-generated datasets in the presence of communication delay attacks, but it specifically focuses on vehicle platoons.

3.6 Summary

Existing studies approach platooning safety from different perspectives. Although the mentioned studies differ in methodology and assumptions, they consistently demonstrate that communication reliability is a critical factor affecting platoon stability and therefore platoon safety. In summary, VeReMi contributes to the development of research on misbehavior and communication-related attacks in VANET environments by introducing structured attack models, ground truths and datasets. ComFASE extends these concepts by providing a framework for simulating and evaluating communication faults and cyberattacks in connected vehicle environments. Simulation frameworks such as OMNeT++, VEINS, SUMO, and PLEXE collectively provide an integrated workflow for modeling vehicular communication, traffic mobility, and platooning behavior.

Although communication delays, communication outages, and emergency braking in vehicle platoons have been investigated in previous studies [3, 5] the combined impact of delay attacks

and emergency braking under varying platoon configurations remains insufficiently addressed in the literature. Therefore, this thesis addresses this gap by systematically evaluating how variations in braking intensity, platoon size, inter-vehicle spacing, leader speed and controller architecture affect platoon safety and stability under communication delay attacks.

4 Problem formulation

Based on the literature review, the thesis author identified the following research gap: limited evidence is available on how different CACC controllers respond when an artificial delay is introduced into the V2V communication channel during braking. In particular, platoons controlled by the Rajamani or Ploeg CACC controllers have not been sufficiently compared under the same delay-attack and braking conditions. This thesis addresses this gap by examining how delayed V2V communication affects the responses of both controllers, coordinated braking of platoons and eventual collision occurrence.

In other words, the research problem is rooted in the lack of a systematic understanding of how communication delays, delay attacks and jamming attacks affect platoon braking. While existing datasets, such as VeReMi, are used to detect road safety issues in vehicular networks, they are not specifically designed for platooning applications and do not analyze the impact of communication attacks on control performance at the platoon level [9]. They also lack comprehensive data generated under varying traffic conditions, platoon size, and attack intensity. Existing datasets are intended for use with various AI/ML models or tools for misbehaviour detection in general vehicular ad hoc networks and do not capture the impact of communication attacks on platoon-level control performance, braking, string stability, or safety margins. Therefore, a structured investigation of the impact of delay attacks on the braking behavior of CAVs in controlled attack scenarios can contribute to filling this research gap. By running multiple scenarios with different delay values, it is possible to understand when delays are most damaging, how they reduce string stability and safety margins and under what conditions platooning becomes unstable or unsafe. This is important because vehicles in a platoon operate with short inter-vehicle gaps and limited safety margins. Communication disturbances can propagate and amplify along the string and therefore compromise platoon safety. Communication attacks may cause immediate visible failures, or they can gradually reduce safety margins and increase the risk of road collisions. Reliable anomaly detection for the early identification of abnormal communication patterns and implementation of appropriate safety measures is as important as the analysis of the causes and consequences of communication disturbances. Such analysis is necessary to understand their effects on controller behaviour, coordinated braking, string stability, and collision risk. This thesis addresses the identified research gap by assessing the responses of the Rajamani and Ploeg CACC controllers to delay attacks during emergency braking. It also contributes to further research by making the resulting simulation-generated datasets publicly available. The third and last contribution of this thesis is the design and execution of a simulation-based experimental campaign for assessing different CACC controllers under V2V communication delay attacks. Thesis author defined the experimental scenarios, selected the investigated parameters and their values, and determined the parameter combinations used in the simulation campaigns. The campaigns were executed using the existing ComFASE simulation environment, which provided the data-generation procedure and output metrics.

- **RQ1:** How does the start time of a delay attack affect whether a collision occurs?
- **RQ2:** How does artificially injected communication delay affect the performance of different CACC controllers during braking in platoons of connected and automated vehicles?
- **RQ3:** Which operating parameters have the greatest impact on platoon safety under communication delay attack conditions?

5 Method

This thesis adopts a hypothetico-deductive research approach. This method is described by Dodig-Crnkovic [40]. Research hypotheses were formulated based on the literature review and converted into testable predictions concerning the effects of delayed V2V communication on platoon safety. Research methodology used in this thesis is presented in Figure 3.

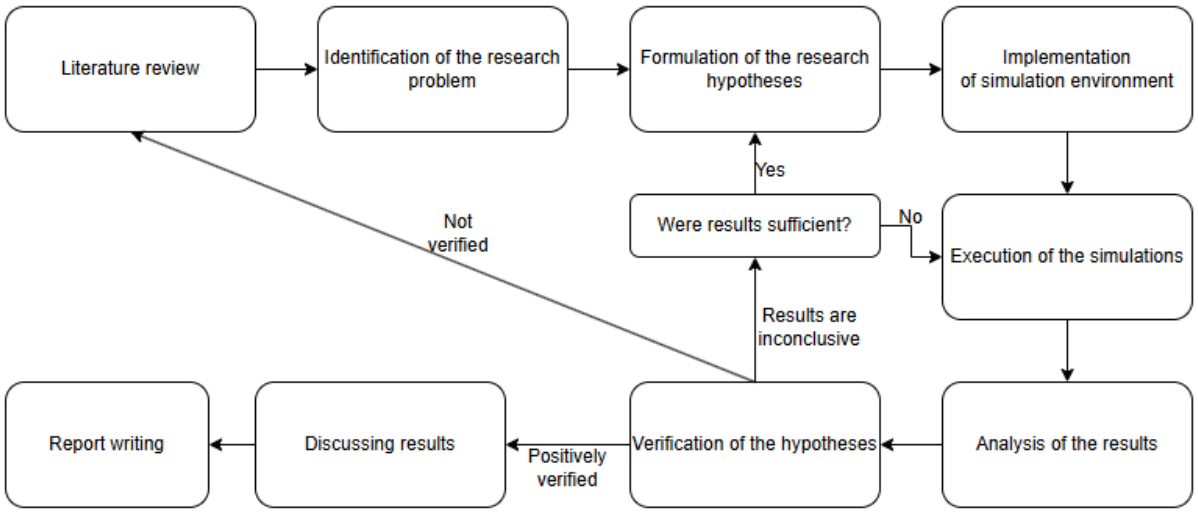

Figure 3: Research methodology adopted in this thesis.

The research process consisted of the following steps:

1. Literature review.
2. Identification of the research problem.
3. Formulation of the research hypotheses.
4. Execution of the simulations.
5. Analysis of the results.
6. Verification of the hypotheses.

The research process began with a review of the literature concerning security in VANETs and, in particular, the security of V2V communication in coordinated driving and its specific form known as platooning. The purpose of the literature review was to understand the research problem and to position this thesis among other studies of this kind with respect to the existing literature. Familiarity with different longitudinal driving control models and studies concerning V2V communication in VANETs led the author to consider the role of communication delay in the operational safety of a platoon. Based on the reviewed literature, the author formulated several research hypotheses that contribute to one general hypothesis. The general hypothesis is as follows:

H1: Artificially introduced communication delays negatively affect platoon safety by disrupting

coordinated braking during an emergency braking maneuver or degrading string stability in a platoon operating under a sinusoidal speed profile.

The following hypotheses developed further into this main hypothesis:

H2: The effectiveness of a communication delay attack depends on the time of its initiation relative to the start of the emergency braking maneuver.

H3: Increasing the magnitude of the injected communication delay increases the collision rate under emergency braking conditions.

The author identified a research problem that resulted from the lack of an in-depth analysis and understanding of delay attacks on platoons controlled by different control models under different emergency braking conditions or in situations requiring the maintenance of string stability within the platoon. The research method was selected according to the specific nature of the research problem. An empirical, simulation-based, experiment-driven methodology was adopted for this thesis. The controlled experiment was conducted through a series of simulations using ComFASE and evaluated using quantitative metrics such as collision occurrence (whether a collision occurred during the simulation campaign), collision count (how many collisions occurred during the simulation campaign), and collision rate (the proportion of collisions in the simulation campaign). After defining the quantitative metrics, it was also necessary to determine the simulation parameters. When selecting these parameters, the author considered, among other factors, the possibility of comparing the results with those obtained by Malik et al., as their research is the most closely related to the experimental campaign conducted in this thesis. The selection and adjustment of the parameters were intended to reproduce a realistic delay attack scenario involving vehicles in a platoon as accurately as possible. The simulation environment was then implemented, the experiments were executed, and the resulting data were analyzed. Then the hypotheses were verified and the research questions were answered.

H1: Was supported quantitatively for coordinated braking and qualitatively for string-stability degradation in the Sinusoidal Scenario. More datasets were generated from simulations of the braking scenario than from simulations of the sinusoidal scenario.

H2: Was supported. Delay attacks activated immediately before emergency braking were substantially more effective than attacks activated after the braking maneuver had begun.

H3: Was supported for both evaluated controllers. Collision rates increased as the injected communication delay increased from 500 ms to 1100 ms. However, the presented results indicate the non-linear nature of this phenomenon as it is dependent on the platoon configuration and parameters.

Simulations were automated with custom Python scripts, and datasets containing parameters like speed, acceleration or deceleration, inter-vehicle gaps, scenarios, platoon size, etc. were gener-

ated. Communication metrics were collected. The simulation based methodology was selected because it allows reproducible and controlled testing of safety-critical platooning scenarios. Part of this experiment consisted of simulation campaigns with various parameter configurations. Initially, value of only one parameter, was tested. When it was possible to determine the approximate value at which the platoon exhibited a specific trend (e.g., an increase or decrease in collision number), campaign was extended by adding another parameter. Thanks to this impact of each and every parameter was tested indivdually aswell as the impact of a group of parameters in correlation to each other (Table 4).

Figure 4 illustrates the complete experimental workflow. The simulation environment was first configured, after which the driving scenario (emergency braking or sinusoidal oscillation) and corresponding parameters were selected. Common parameters, including controller type, communication delay, platoon size, and leader speed, were then configured before executing the simulation campaign using ComFASE. Finally, the generated CSV and XML outputs were parsed and analyzed to evaluate string stability and collision occurrence.

6 Ethical considerations

This study was conducted entirely in a simulated environment and did not involve human participants, personal data, animals, physical vehicles, or experiments on public roads. The experiment did not expose individuals, vehicles, or infrastructure to physical or privacy related risks. Knowledge about delay or jamming attacks could be used both to improve the resilience of cooperative driving systems and to disrupt them. The intended purpose of this research is to contribute to improving the safety and resilience of vehicle platoons by identifying conditions under which communication delays may compromise platoon security and safety. All attacks were executed within a simulated environment. No personal or confidential data were used, collected, or processed in this study.

7 Simulation environment and experimental setup

This section provides a detailed description of the experimental environment used to simulate delay attacks on a platoon of CAVs. The Section presents the experiment step by step, including the simulation setup, scenarios, experimental parameters, configuration settings, and the reasons for selecting specific parameter values. At first, the author conducted the experiment to verify the hypothesis concerning the critical time interval during which the platoon is most vulnerable and to estimate a delay threshold below which collisions do not occur. VMware Workstation 17 Pro, version 17.6.2, and a virtual machine running Debian 13.6 Trixie were basis for the experiment. Then simulation platforms and frameworks required to conduct the experiment (described in the Background section) were subsequently installed in the virtual machine.

The experiment was divided into three stages. First, a preliminary campaign (pre-campaign) was conducted to identify the collision threshold. This was followed by the main simulation campaign (main campaign), which examined the combined effects of multiple parameters. Finally, supplementary simulation campaigns were conducted to investigate selected parameters and scenarios in greater detail.

In total, 35940 simulation runs were conducted and collected. The main experimental campaign contains 35100 runs, including the Rajamani and Ploeg CACC campaigns and the additional braking, platoon-size, delay, and attack-timing analyses. A further 840 runs were conducted to identify the collision threshold. Thesis author conducted first a pre-campaign to identify delay threshold in which collisions began to occur. In the pre-campaign, the injected delay and the attack initiation time were varied, while the controller, spacing policy, inter-vehicle gap, platoon size, braking deceleration, attack duration, and other simulation parameters were kept constant. The campaign was repeated for leader speeds of 80 and 100 km/h. After identifying the transition between collision-free and collision-producing delay values, the experimental campaign was extended to examine the effects of several parameters. The main simulation campaign examined the combined effects of several parameters in the Braking Scenario. Separate campaigns were conducted for the Rajamani and Ploeg CACC controllers, with 10,080 simulation runs performed for each controller. The main campaign therefore comprised 20160 simulation runs and examined collision occurrence across combinations of several simulation and attack parameters. The remaining simulations analyzed different deceleration levels in the Braking Scenario at leader speeds of 80 and 100 km/h. Finally, in addition to the main campaign, the author conducted individual simulations in the Sinusoidal Scenario to assess string stability.

7.1 A pre-campaign. Collision Threshold in the Braking Scenario

The collision threshold was investigated in the Braking Scenario using a single platoon consisting of four vehicles travelling on one lane. The vehicles were controlled by the Rajamani CACC controller, which used the LPF communication topology and the CDG policy. The desired

inter-vehicle gap was 5 m. The CACC-Rajamani controller parameters were $C_1 = 0.5$, $\xi = 1$, and $\omega_n = 0.2$ Hz, while the engine time constant was $\tau = 0.5$ s. The leader speed was set to 80, respectively 100 km/h, depending on the simulation run. The leader began emergency braking at $t = 17.0$ s with a deceleration of -5 m/s^2. Each simulation lasted 60 s.

The platoon communicated using the DSRC/WAVE stack based on IEEE 802.11p, with IEEE 1609.4 used at the MAC layer. Beacon messages were transmitted at 10 Hz. The carrier frequency was 5.89 GHz, the bit rate was 6 Mbps, and the packet size was 200 B. The transmission power was 100 mW, the receiver sensitivity was −94 dBm, and the thermal noise level was −95 dBm. Radio propagation was represented using the SimplePathlossModel with free-space propagation and a path-loss exponent of $\alpha = 2.0$. The configured ACK error rate and frame error rate were both zero.

A delay attack was introduced through ComFASE by adding an artificial delay to V2V beacon messages. Communication delays of 100, 200, 300, 400, and 500 ms were tested. For each delay, the attack start time was varied from $t = 16.0$ s to $t = 21.9$ s in increments of 0.1 s, resulting in 60 runs for each combination of delay and leader speed. Each attack lasted 6 s. Its end time varied from $t = 22.0$ s to $t = 27.9$ s according to its start time. Attacks on both the sender and receiver were enabled. The delay attack was enabled, while DoS and jamming attacks were disabled.

7.2 Run-level Dataset

This thesis presents a simulation-generated, run-level experimental dataset. Each row represents one simulation run and contains the simulation configuration, attack parameters, vehicle-acceleration metrics and collision outcome. It is neither a time-series nor a packet/beacon-level dataset. However, with this dataset it is possible to determine whether a simulation configuration produced a collision, calculate collision counts and collision rates, compare safety across attack initiation times, examine how delay magnitude, braking intensity, speed, spacing, controller type, and platoon size are associated with collision occurrence and on that basis identify potentially dangerous combinations of parameters.

The dataset collection can be used to determine whether a given simulation configuration produced a collision, calculate collision counts and collision rates, compare safety outcomes across attack initiation times and examine how delay magnitude, braking intensity, vehicle speed, inter-vehicle spacing, controller type, and platoon size are associated with collision occurrence. On this basis, potentially dangerous combinations of experimental parameters can be identified. The resulting CSV files are publicly available in the: https://github.com/sebastian91b/cav-platooning-delay-dataset.

7.3 Main campaign. Delay attacks simulations using ComFASE on Braking Scenario and Sinusoidal Scenario in Plexe-Veins

In this experiment the ComFASE tool [4] has been used, which is already described in this and previous Sections. The goal of the experiment was to analyze the behavior of vehicles traveling in a platoon using V2V communication under malicious delay attacks, with a strong emphasis on its impact on string stability and platoon safety during rapid emergency braking. Then generate methodologically and systematically organized datasets to recognize delay attack patterns and evaluate results. All exeprimnets were conducted in a simulated environment integrating the OMNeT++, Veins, Plexe, SUMO and ComFASE frameworks. For experiments the DSRC/WAVE protocol stack for vehicular communication model was used, and the free-space loss model was used as the wireless model. The main type of attack analyzed was the delay attack, implemented by manipulating the propagation delay parameter in V2V communication. A fixed, artificial delay value was added to each packet sent. ComFASE tool allows attacks on the sender or the receiver, as well as to attack both sides of communication simultaneously. In the platooning each vehicle is usually both the sender and the receiver of the message - except platoon leader (this vehicle is a sender only).So all the attacks in the experiment were carried out with the sender and receiver parameters set to “true”. To reflect and accurately simulate real platooning controller usually used by CAVs in a real traffic platoon scenario, Rajamani’s CACC controller with CDG strategy was set in OMNeT++ simulation configuration file (omnet.ini). So in this study the CACC controller was configured to use a constant distance between vehicles (with “caccSpacing” parameter), which corresponds to the CDG strategy commonly used in platooning. This is for realistic simulation of platoons environment. Additionally, in some simulations, a PLOEG controller based on the CTG strategy was used for comparison with CACC controller in terms of robustness and performance. The parameters that were changed in the test phase of the experiment are: distance between vehicles (5, 8, 10 meters), leader’s initial braking deceleration (-5, -6, -7, -8 m/s^2), controller (CACC, PLOEG), number of vehicles in the platoon (4, 5, 6, 7, 8) and cruising speed of the platoon, which is determined by platoon leader initial speed value (see Figure 4).

Simulations were automated with Python scripts, and datasets containing parameters like speed, acceleration (or negative acceleration, so deceleration), inter-vehicle gaps, scenarios, platoon size, etc. were generated. Communication metrics were collected. The simulation based methodology was selected because it allows reproducible and controlled testing of safety-critical platooning scenarios. Part of this experiment consisted of simulation campaigns with various parameter configurations. Initially, value of only one parameter, was tested. When it was possible to determine the approximate value at which the platoon exhibited a specific trend (e.g., an increase or decrease in collision number), campaign was extended by adding another parameter. Thanks to this impact of each and every parameter was tested indivdually aswell as the impact of a group of parameters in correlation to each other (Table 4).

Figure 4 below illustrates the complete experimental workflow. The simulation environment was first configured, after which the driving scenario (emergency braking or sinusoidal oscillation) and corresponding parameters were selected. Common parameters, including controller type, communication delay, platoon size, and leader speed, were then configured before executing the simulation campaign using ComFASE. Finally, the generated CSV and XML outputs were parsed and analyzed to evaluate string stability and collision occurrence.

Experiment category	Runs	Delay range	Controller
Braking scenarios	10080	500–1100 ms	CACC
Braking scenarios	10080	500–1100 ms	PLOEG
Platoon scalability study	5760	500 ms	CACC
Extended delay analysis	5760	600–900 ms	CACC
Attack timing analysis	3360	500–1100 ms	CACC
Detailed attack-phase analysis	60	500 ms	CACC
Total	35100	–	–

Table 4: Summary of conducted simulation experiments (campaign)

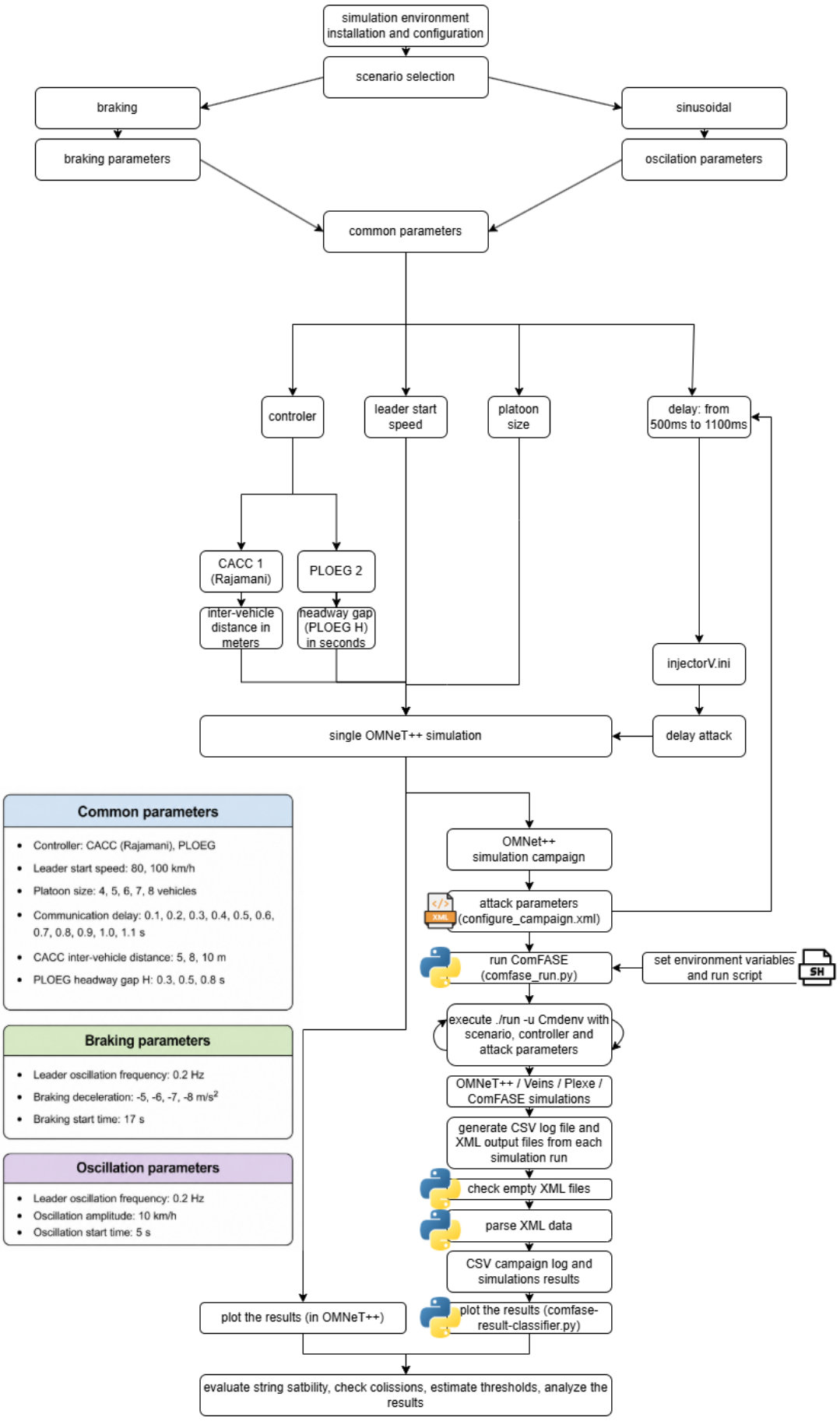

Figure 4: Experimental workflow of the simulation campaign implemented using ComFASE.

Data was collected methodically. In the initial testing phase only one simulation parameter was changed at the time and so that its impact on the platoon could be evaluated. Every parameter was tested in various configurations and scenarios(e.g., increasing the number of vehicles in the platoon by one). Both scenarios with the CACC controller in configurations with different levels of braking, speed and inter-vehicle distances. Both CACC controllers (RAJAMANI, PLOEG) were tested with this method.

Table 5 shows all the simulation parameters used in the experiment to make the experiment reproducible. ComFASE tool allows the injection of malicious attacks by changing settings in its injectorV.ini file. Deep understanding of these workflow environments and (especially omnet.ini and injectorV.ini) files was crucial to run simulations. Also Delay attack parameters have been changed in the ComFASE in injectorV.ini file. ComFASE tool provides python scripts for campaign automatization, so it possible to test different attacks in different scenarios. In this experiment both ComFASE scripts and other custom scripts were used. Propagation delay values were set between 500 and 1100 ms to simulate different levels of delay attack strength.

Table 5: Configuration parameters for the simulation analysis.

Category	Parameter	Value
communication	Communication protocol	DSRC/WAVE, IEEE 802.11p
	PHY model	IEEE 802.11p
	MAC model	IEEE 1609.4
	Path loss model	SimplePathlossModel/free-space
	Path loss exponent	$\alpha = 2.0$
	Tx Power	100 mW
	Packet size	200 B
	Bit rate	6 Mbps
	Sensitivity	−94 dBm
	Thermal noise	−95 dBm
	Frequency	5.89 GHz
	Beacon frequency	10 Hz
	ACK error rate	0
	Frame error rate	0
	Propagation delay	Enabled
mobility	Scenario types	BrakingScenario,SinusoidalScenario
	Leader speed	80 km/h
	Platoon size	4, 5, 6, 7, 8 vehicles
	Number of platoons	1
	Total number of lanes	1
	Platoon insert time	1 s
	Simulation time limit	60 s
	Leader oscillation frequency	0.2 Hz
	Oscillation amplitude	10 km/h
	Start oscillating	5 s
controller	Controllers	CACC, PLOEG
	CACC spacing	5, 8, 10 m
	C_1	0.5
	ξ	1
	ω_n	0.2 Hz
	Engine time constant τ	0.5 s
	PLOEG headway H	0.3, 0.5, 0.8 s
	k_p	0.2
	k_d	0.7
	ACC headway	1.2 s
braking	Braking scenario	Enabled
	Braking start time	17 s
	Braking deceleration	5, 6, 7, 8 m/s^2
	Packet loss rate	0
attack	Attack type	Delay attack
	Attack start time	16 s
	Attack end time	22 s
	Attack duration	6 s
	Injected propagation delay	500–1100 ms
	Attack target	Sender and receiver
	DoS attack	Disabled
	Jamming attacks	Disabled

Two main platooning scenarios were used in the experiment. The first was the Braking Scenario, in which the platoon leader performed sudden braking (see Figure 5). In this scenario, different leader braking deceleration values of -5 m/s², -6 m/s², -7 m/s², and -8 m/s² were analyzed. Figure 5 demonstrates the deceleration profiles of four vehicles in a platoon. At 17 s into the simulation time, the leading vehicle initiates a braking maneuver, and as a result, the following vehicles also start braking. Before the braking, the vehicles were cruising using the CACC controller. Due to the absence of any attack, all vehicles are able to perform the braking maneuver at the same time. As a result, at 23 s into the simulation time, the vehicles complete their braking maneuver.

The second scenario was the Sinusoidal Scenario, in which the platoon leader periodically accelerated and decelerated in a sinusoidal fashion (see Figure 6). The oscillation parameter in the SinusoidalScenario is responsible for the periodic acceleration and deceleration of the leader vehicle in a sinusoidal fashion, but oscillation parameter was not changed from the ComFASE default which was previously set to:

```
Leader Oscillation Frequency = 0.2 Hz
Scenario Oscillation Amplitude = 10 kmph
Scenario Start Oscillating = 5 s
```

The oscillation parameters define the frequency of sinusoidal changes in the platoon leader's speed, the duration of the acceleration and deceleration cycle, and the leader's speed relative to the base speed. The oscillation parameters determine the frequency, period, and amplitude of the sinusoidal variation in the platoon leader's speed. The parameter `startOscillating = 5 s` means that the leader begins following the sinusoidal speed profile 5 s after the start of the simulation. An oscillation frequency of 0.2 Hz corresponds to one complete speed-oscillation cycle every 5 s. An amplitude of 10 km/h means that the leader's speed varies by up to 10 km/h above and below the reference speed, resulting in a peak-to-peak variation of 20 km/h. For example, at a reference speed of 80 km/h, the leader's speed varies between 70 and 90 km/h.

```
Scenario CACC Spacing
to 5 m, 8 m, 10 m;

Scenario Braking Deceleration
to -5 m/s², -6 m/s², -7 m/s², -8 m/s²;

Scenario Controller
to CACC(1) and PLOEG(2);

Traffic Platoon Size
to 4, 5, 6, 7, 8 vehicles;
```

`Scenario Type`
`to BrakingScenario and SinusoidalScenario.`

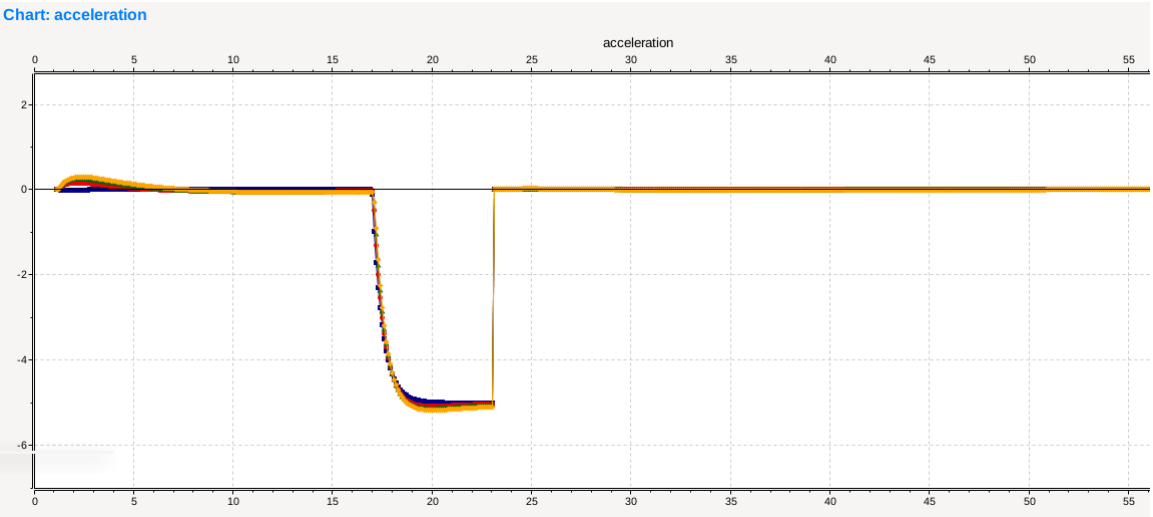

Figure 5: Vehicle acceleration profiles in the Braking Scenario. The figure illustrates the dynamic response of the platoon to the braking maneuver initiated by the platoon leader.

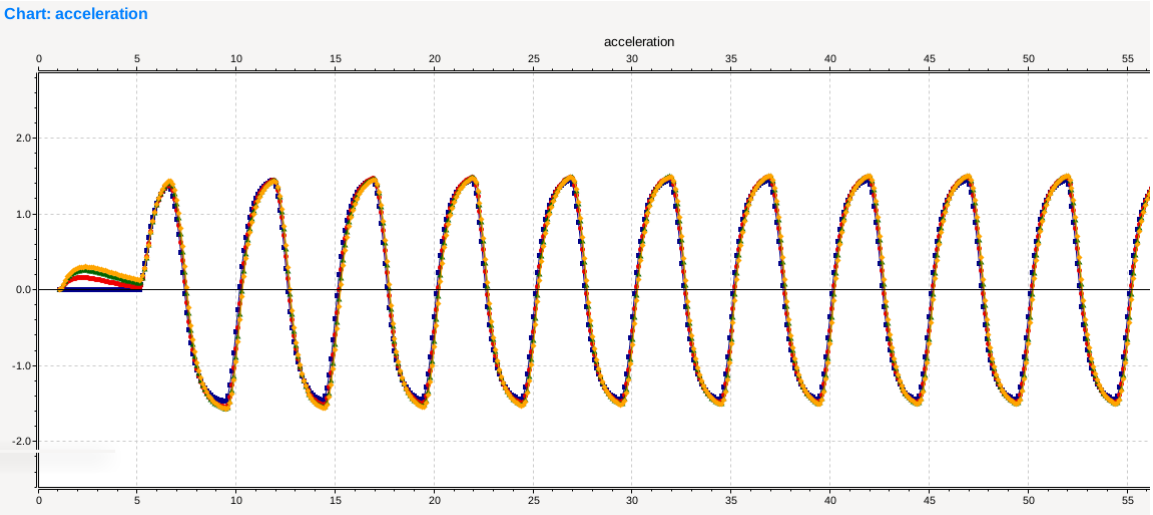

Figure 6: Vehicle acceleration profiles in the Sinusoidal Scenario. The figure illustrates the dynamic response of the platoon to periodic speed variations of the platoon leader (oscillation).

Malik et al. mention testing their tool on a platoon of 4 cars [4]. This experiment extends platoon size to 8 vehicles and then tests delay attack scenarios on respectively 4, 5, 6, 7 i 8 vehicles. Therefore it is possible to examine the effects of delay attack on larger platoons braking and string stability. Since all CAVs in a VANET share the same communication medium, size of the platoon could eventually be important for communication quality. CSMA/CA in IEEE 802.11p, uses random backoff. Bilstrup et al. showed that this access method cannot guarantee channel access before a specified deadline [13]. The nondeterministic nature of CSMA/CA media access may theoretically amplify communication delay within platoon together with the

increasing number of nodes (number of CAVs). However main limitation of CSMA/CA protocol is that there is no guarantee to access a channel, so vehicles can experience delays when there are many other channel users in traffic.

Cumulative delay and disturbance amplification involves also wireless propagation delay, controller delay and actuator delay. An additional problem with IEEE 802.11p-based platooning is the limitations of the communication medium itself. This standard is characterized by limited transmission range and bandwidth, which becomes particularly problematic in a dense traffic and/or in larger platoons, generating a large number of V2V messages. As a result, it is impossible to precisely determine the maximum transmission delay, as the channel access time depends on the current network load, the number of vehicles, and the level of contention between nodes. This can lead to increased communication delay, jitter, and packet loss, directly impacting the platoon's string stability and security. One of the key aspects of designing autonomous vehicle platoon systems is precisely determining the maximum tolerated communication delays for specific traffic parameters, such as vehicle speed, inter-vehicle gap, and platoon size. Because communication delays affect platoon safety, it is important to determine the maximum delay under which a given platoon configuration remains string-stable and avoids collisions. The author does not claim that this study identified a generic safety threshold. However, the experiment indicates that, for the specific scenario in which the platoon encounters no other road users and the communication conditions are favourable, it is possible to identify a threshold below which no collisions occur. This is further described in the Section 8.

Simulation campaign was mainly dedicated to the Braking Scenario and greatest emphasis in the experimental campaign was placed on the Braking Scenario. While the primary contribution of this thesis was based on the Braking Scenario, the experiments conducted using the Sinusoidal Scenario provided a complementary evaluation of the findings reported by Malik et al. All simulations considered a single-lane platoon without any additional road users representing surrounding road traffic.

7.4 Experiment parameters

7.4.1 Deceleration

The Table 6 and the Figure 7 show the propagation of braking disturbance in a custom braking scenario. Platoon consisted of four vehicles and the CACC (Rajamani) controller was used for longitudinal vehicle control in this experiment. For subsequent values of attack delay, i.e. artificially injected delay, each subsequent vehicle decelerates harder. Vehicle 0 maintained a deceleration of approximately -5 m/s^2, while Vehicle 3 reached deceleration amplitudes exceeding -8 m/s^2 under larger delay values in the platoon.

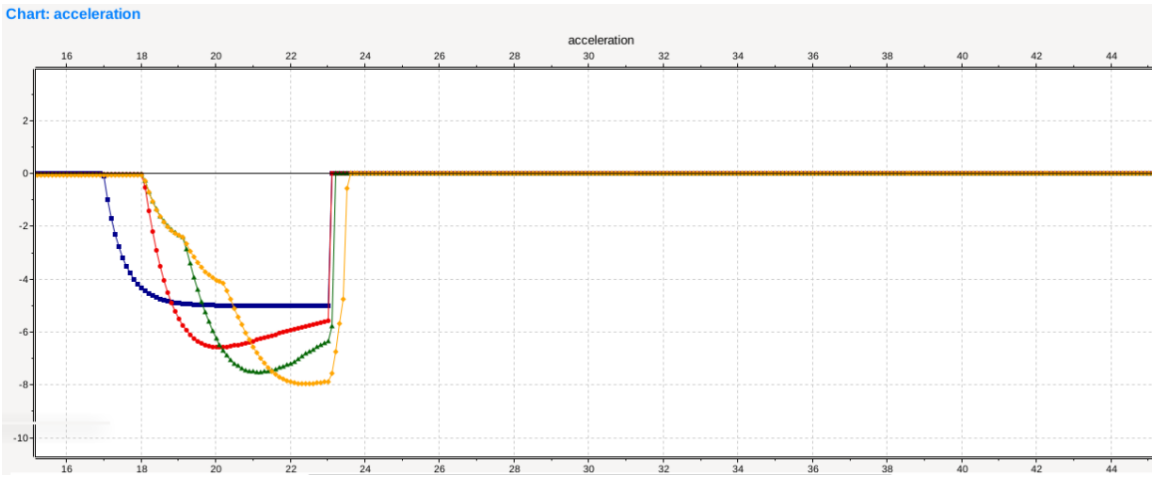

Figure 7: Vehicle acceleration profiles during a delay attack in the Braking Scenario. The figure shows the delayed braking response and braking disturbance amplification in the platoon after emergency braking is initiated by the leader.

Table 6: Propagation of braking disturbance under increasing delay attack values

Injected Delay (s)	**Vehicle 0**	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**
0.5	-5.00	-5.79	-6.20	-6.35
0.6	-5.00	-5.94	-6.45	-6.63
0.7	-5.00	-6.10	-6.70	-6.93
0.8	-5.00	-6.25	-6.96	-7.24
0.9	-5.00	-6.41	-7.23	-7.57
1.0	-5.00	-6.57	-7.51	-7.92
1.1	-5.00	-6.73	-7.80	-8.08

In the Sinusoidal scenario, vehicles in the platoon continuously accelerate and decelerate, resembling real-world traffic scenarios. As in the previous experiment, a platoon of four CAVs controlled by the CACC (Rajamani) controller was tested. The one second delay was injected starting at the 16th second of the simulation. As shown in Figure 8, the platoon leader began to decelerate at approximately the 17th second, whereas the first following vehicle initiated deceleration over a second later, despite the injected artificial communication delay being one second. As a result, the observed onset of braking may occur later than the communication delay due to controller and actuator response delays, as well as the vehicle's dynamic response. Therefore, braking may occur later than the communication delay itself. That is why it is difficult to determine the cumulative delay for each vehicle individually.

The potential collision risk increased when the predecessor exhibited relatively low acceleration, while the following vehicle accelerated more aggressively. As shown in Figure 8 and Figure 9, this behavior begins to appear at approximately 21 s and reaches its peak around 22.1 s of simulation time.

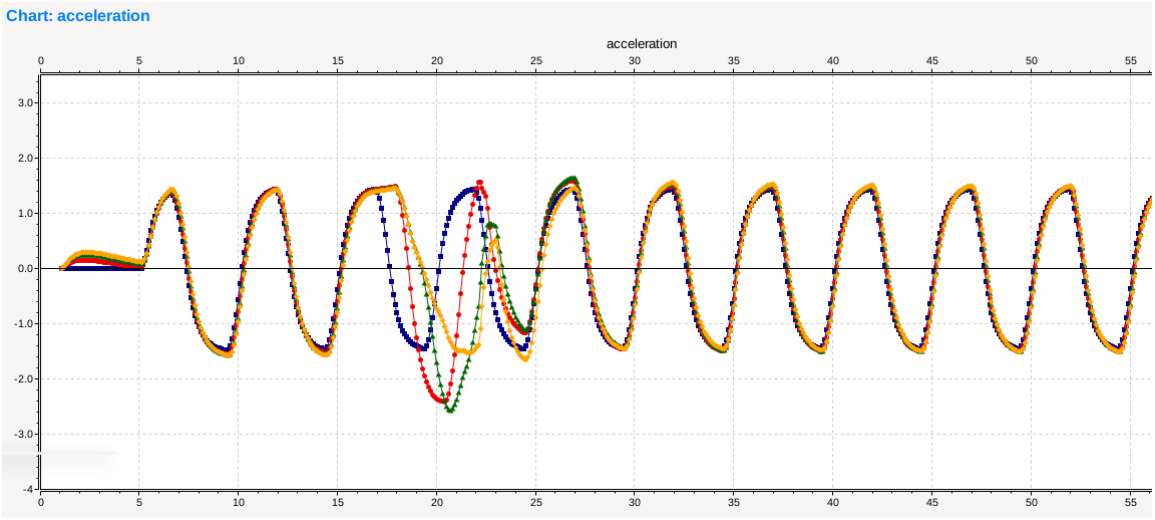

Figure 8: Vehicle acceleration profiles during a delay attack in the Sinusoidal Scenario. The figure shows the delayed braking response, acceleration and deceleration disturbance and consequently, gradual loss of string stability.

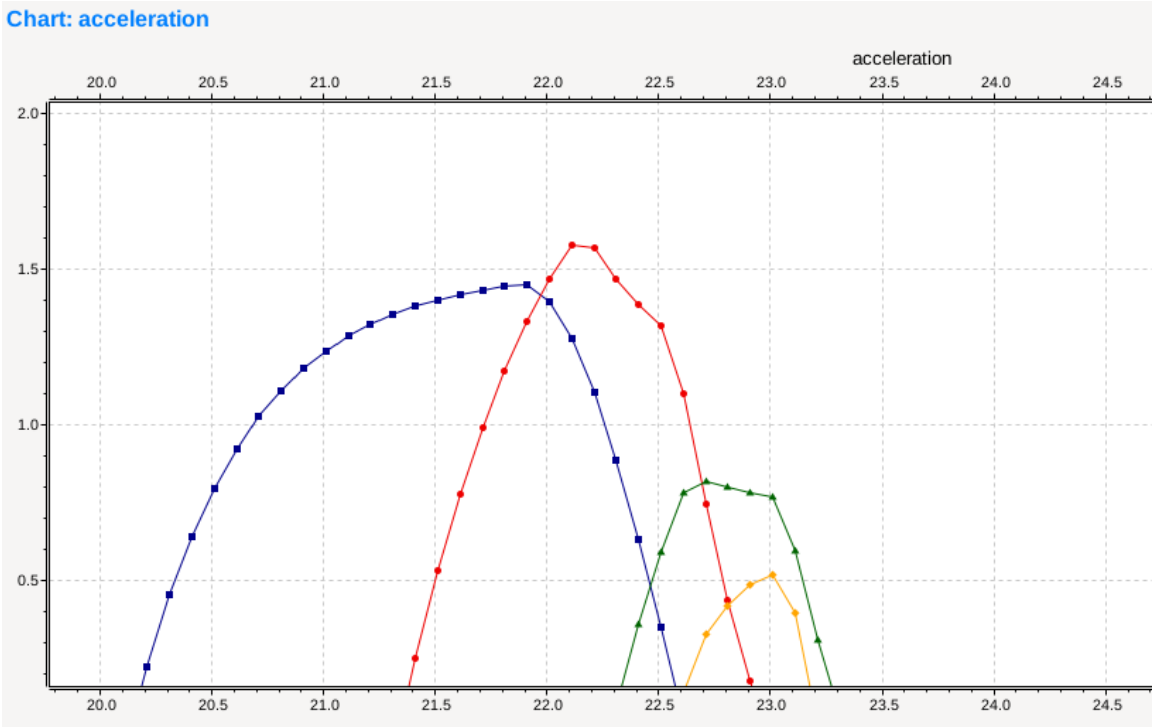

Figure 9: Vehicle acceleration profiles during a delay attack in the Sinusoidal Scenario. The figure shows the critical region preceding a collision, where the inter-vehicle distance decreases to an unsafe level, compromising platoon safety.

7.4.2 Speed

This section examine the impact of speed on platoon safety and string stability. As already noted, for platooning to be fuel-efficient, vehicles must maintain possible smallest gaps. This is not easy due to the obvious risk of collision in such tight conditions. Therefore, in the experimental part of the study, I focus on the impact of speed on platoon safety. In the simulations, initial leader speeds of 80 and 100 km/h were selected for several reasons. Primarily, these are realistic highway crusing speeds of fleets. Furthermore benefits of driving with reduced aerodynamic drag become noticeable at speeds of approximately 50–60 km/h. They are noticeable, but still relatively low at these speeds. At speeds exceeding 80 km/h, the benefits of slipstream driving (drafting) increase significantly, making drafting a highly effective solution in terms of reducing aerodynamic drag and energy consumption. Most European countries limit heavy goods vehicles and or trucks to around 90 km/h. Some European countries allow heavy goods vehicles to travel up to 100 km/h, making this the de facto upper limit for driving speeds in Europe. Local exceptions may apply, but it is generally accepted that the upper speed limit for the truck fleet will be around 100 km/h. For this reason, values of 80 and 100 km/h were considered appropriate for conducting realistic simulations. In simulated braking scenarios, vehicles adjust their speed to platoon leader's speed. It is then respectively 80 and 100 km/h.

Vehicle speed directly affects reaction time and braking dynamics in platooning systems. Tables 7 and 8 show that braking disturbances amplified as they propagated upstream through the platoon. Speed factor is therfore crucial for string stability in case of rapid and unexpected braking of platoon leader. As show leading vehicle maintained a constant braking deceleration of approximately -5.0 m/s^2, follower vehicles experienced increasingly aggressive braking responses under larger injected delay values.

Higher speeds reduce the available reaction time. Tight platoon with smaller inter-vehicular gaps lowers time window for follower vehicles during emergency braking maneuvers. This leads to a need of followers controllers stronger braking recation in order to maintain safe inter-vehicle gaps. If reaction window shrinks, and infomartion about acceleration is outdated, controller cannot react appropriately to a hazardous situation. In that particular case system responsiveness is unchanged and it even acts correctly in terms of matemathical and physical model. This is however not enough to ensure platoon safe operating in case of emergency braking. In a delay attack, data falsification is not necessary to compromise system security. This is why this particular attack vector is harder to identify, non-obvious, and hazardous. This case shows that the data the follower received was valid, not manipulated or injected and yet dangerous for the system.

Table 7: Propagation of braking disturbance under increasing delay values at 80 km/h

Injected Delay (s)	Vehicle 0	Vehicle 1	Vehicle 2	Vehicle 3
0.5	-5.0	-5.79	-6.20	-6.35
0.6	-5.0	-5.94	-6.45	-6.63
0.7	-5.0	-6.10	-6.70	-6.93
0.8	-5.0	-6.25	-6.96	-7.24
0.9	-5.0	-6.41	-7.23	-7.57
1.0	-5.0	-6.57	-7.51	-7.92
1.1	-5.0	-6.73	-7.80	-8.08

Table 8: Propagation of braking disturbance under increasing delay values at 100 km/h

Injected Delay (s)	Vehicle 0	Vehicle 1	Vehicle 2	Vehicle 3
0.5	-5.0	-5.79	-6.21	-6.36
0.6	-5.0	-5.95	-6.46	-6.64
0.7	-5.0	-6.10	-6.71	-6.94
0.8	-5.0	-6.26	-6.97	-7.25
0.9	-5.0	-6.41	-7.24	-7.58
1.0	-5.0	-6.58	-7.52	-7.93
1.1	-5.0	-6.74	-7.81	-8.09

7.4.3 Platoon size

Table 9 shows that platoon size is another factor affecting platoon safety in terms of emergency braking and string stabilty. Braking disturbance propagates downstream, more vehicles respond to delayed braking information and cumulative disturbance amplification becomes larger. This supports claim that disturbance amplification grows with platoon size.

Table 9: Propagation of braking disturbance for different platoon sizes under identical simulation conditions

Platoon Size	First Vehicle Deceleration	Last Vehicle Deceleration
5 vehicles	-5.0	-6.35
6 vehicles	-5.0	-6.72
7 vehicles	-5.0	-7.28
8 vehicles	-5.0	-8.08

Data from selected braking scenario indicate the nonlinear nature of this phenomenon. Based

on the simulations, it was observed that increasing the platoon size from 5 to 6 and from 7 to 8 vehicles did not significantly increased the number of collisions within the platoon. One the other hand Larger platoons casued larger amount of colisions in comaparsion to platoons consisting of 5 and 6 vehicles - which in major cases (but not all cases) remained collision-free.

7.4.4 Inter Vehicle Distances

In the simulations following values were used for the distances between vehicles behind the leader. In experimenet these values refers to inter vehicle gaps. Following gaps were used in simulations with the CACC (Rajamani) controller that uses CDG strategy.

5 m — Aggressive spacing: the shortest evaluated inter-vehicle gap and the most collision-prone configuration, indicating degraded spacing margin;

8 m — Intermediate spacing: an intermediate inter-vehicle gap that provides a compromise between close platoon formation and spacing margin;

10 m — Safer spacing: the largest evaluated inter-vehicle gap, generally providing a greater spacing margin but potentially reducing the aerodynamic benefits of close platooning.

These gaps are used in simulations with the CACC (Rajamani) controller that uses CDG strategy.

For the Ploeg controller, author used the following times: 0,3, 0,5 and 0,8 s.

7.4.5 Injected Delay

In ComFASE, the Delay attack is implemented by modifying the Propagation Delay parameter, effectively introducing an additional artificial communication delay between vehicles in the platoon. Instead of blocking or dropping packets, the attack artificially delays the delivery of V2V (vehicle-to-vehicle) messages. As a result, vehicles continue to communicate, but receive outdated information. Technically, the attack is implemented by manipulating the propagationDelay parameter. However, throughout this thesis, it is interpreted as an injected communication delay rather than the physical propagation delay of a radio signal. The Veins framework calculates the packet transmission time using the propagation delay (PD), which depends on the distance between the sender and the recipient. Propagation delay is calculated accordingly.

$$Propagation\ Delay = \frac{Distance}{Speed\ of\ Light}$$

ComFASE intercepts communication in the Veins ChannelAccess.cc source file, specifically in the calculatePropagationDelay() function. This function calculates the standard transmission time for a packet between vehicles. When the Delay Attack is active, this function is used by ComFASE tool to retrieve the calculated propagation delay value. Function does not yet return

the value but passes it insetad to the ComFASE attack module. The attack module then checks whether the Delay attack is active, whether the current simulation time is within the attack window, whether the sender or recipient belongs to the attacked vehicles, and what delay value should be added. If all conditions are met, ComFASE adds an artificial delay which is stored in propagation delay variable. The delay-attack injection algorithm is illustrated in Algorithm 1.

Algorithm 1: Delay Attack injection

Input: Packet transmitted from Vehicle A to Vehicle B

Output: Final packet propagation delay

Vehicle A wants to send a packet to Vehicle B;

Veins calls `calculatePropagationDelay()`;

Retrieve the position of Vehicle A;

Retrieve the position of Vehicle B;

Compute the normal propagation delay:

$$\texttt{normalDelay} = \frac{\texttt{distance(A,B)}}{SpeedOfLight}$$

Find the ComFASE module;

Check whether the Delay Attack is active;

if *Delay Attack is NOT active* **then**

    return `normalDelay`;

if *Delay Attack is active* **then**

    Check whether the current simulation time is inside the attack window;

    Check whether Vehicle A or Vehicle B is a target vehicle;

    Check whether the attack targets the sender or the receiver;

    if *all attack conditions are satisfied* **then**

        `finalDelay` = `normalDelay` + `injectedDelay`

        return `finalDelay`;

    else

        return `normalDelay`;

The packet is delivered after the calculated delay;

7.5 Braking Dynamics in the Platoon

The data from individual simulations provided information about the number of collisions occurring in scenarios involving the injection of artificial delays into different longitudinal driving-control models. This data can be used to observe the effects of communication delays and to compare the number of collisions in platoons using the Rajamani's CACC and Ploeg's CACC controllers. However, without time-series data, it is not possible to observe changes in the sequence of events associated with the dynamics of the vehicles in the platoon. Instead of basing the analysis exclusively on the value specified in the injector configuration, the effective delay recorded by the message receiver can be observed step by step. For this reason, a beacon-level analysis was conducted for a representative run of the braking scenario.

7.5.1 Complementary Beacon-Level Dataset

The investigated run included four vehicles traveling in a single platoon. The leader was configured to begin braking at $t = 17\,\mathrm{s}$ with a specified deceleration of $5\,\mathrm{m/s^2}$. The Delay Attack implemented by ComFASE was active between $t = 16\,\mathrm{s}$ and $t = 22\,\mathrm{s}$ and introduced an additional delay of $1.0\,\mathrm{s}$. Platooning messages were generated at intervals of $0.1\,\mathrm{s}$. For each correctly received message, the sender and receiver IDs, message sequence number, generation and reception times, sender and receiver states, Signal-to-Interference-plus-Noise Ratio (SINR) and received signal power were recorded according to the methodology proposed in VeReMi. This was a single simulation based on the Rajamani's CACC model with a CDG policy. The purpose of the analysis was to investigate the following sequence of events:

injected delay → increase in CAM message age → outdated information about the predecessor → delayed follower response → increase in relative velocity → platoon compression → dangerous and rapid reduction in inter-vehicle gaps resulting in a collision

7.5.2 Message Delivery Delay

For each correctly received message, its age at the time of reception was calculated as:

$$A_{\mathrm{CAM}} = t_{\mathrm{receive}} - t_{\mathrm{generate}},$$

where t_{generate} denotes the time at which the platooning message was generated by the sender, whereas t_{receive} denotes the time at which the message was delivered to the receiver's protocol. The value of A_{CAM} should not be interpreted exclusively as the physical propagation delay of radio waves. It represents the total observed time between the generation of a message and its delivery to the receiver's protocol. It therefore includes the natural processing and transmission time through the simulated communication stack and, when the attack was active, the delay introduced by ComFASE. Figure 10 presents the difference between the message transmission

and reception times for the nodes in the platoon in this particular simulation. The communication delay itself did not accumulate along the platoon. Leader's beacons were received by all three followers with almost identical send-to-receive delays of approximately 1004.483 ms. However, the artificially injected delay was only 1 s, which means that the configured Delay Attack was imposed on the existing message-delivery latency.

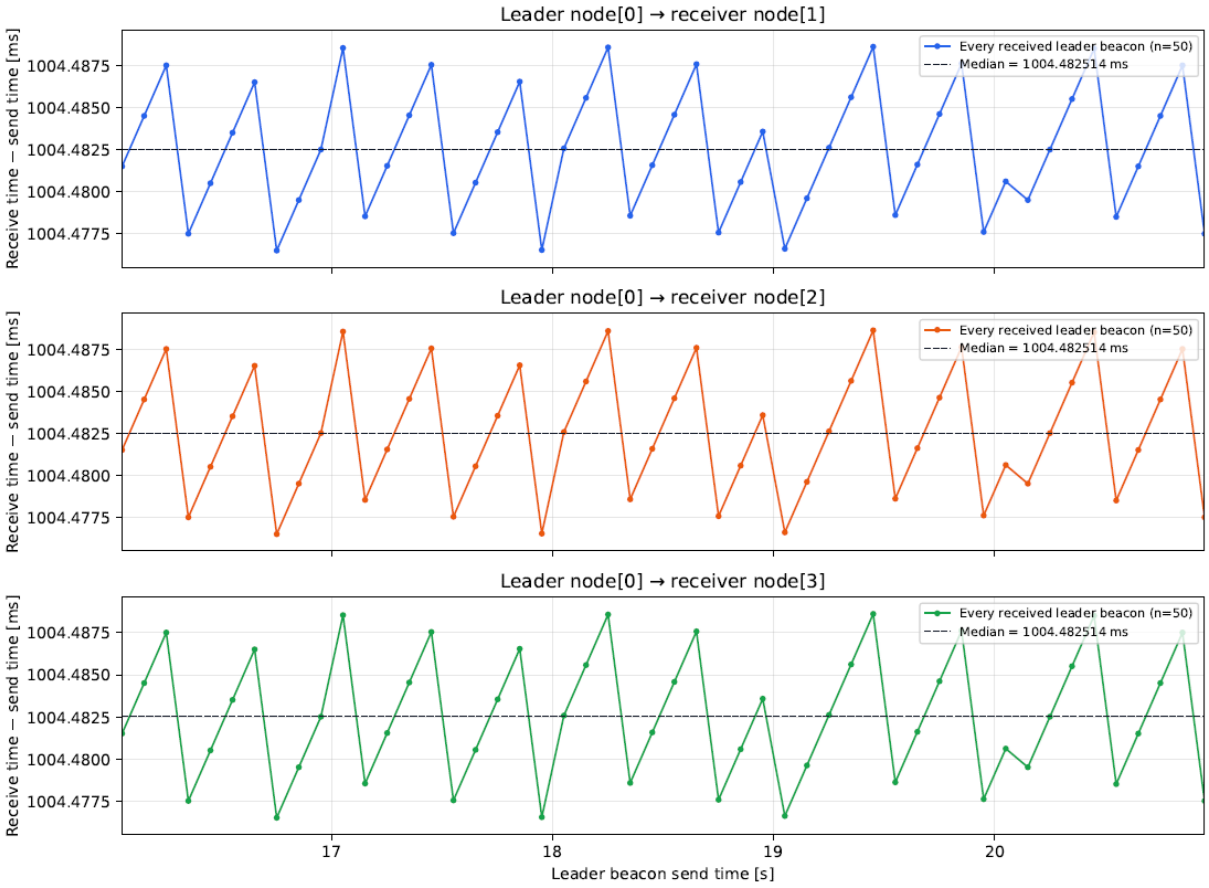

Figure 10: Per-beacon send-to-receive delay of leader messages during the Delay Attack. Each panel represents a direct transmission from the leader, `node[0]`, to one of the following vehicles. Every marker corresponds to one successfully received beacon and was calculated directly as $1000(t_{\text{receive}} - t_{\text{send}})$. The median observed delay was approximately 1004.483 ms for each receiver, consisting of the configured 1000 ms artificial delay superimposed on an existing message-delivery latency of approximately 4.483 ms.

7.5.3 Physical Lag Response

The acceleration commanded by the controller is not identical to the physical acceleration of the ego vehicle. The controller determines the desired acceleration and, on this basis, issues commands to the actuators. Vehicle and actuator dynamics cause the actual acceleration not to change immediately after the controller issues a command. In this context, the role of the `engineTau` parameter, which was set to 0.5 s, should be emphasized. Plexe models the

response time of the powertrain and the actuating system using the engine time constant τ. This means that the engine, brakes, and the entire actuation system respond gradually. The powertrain time constant τ describes how quickly the vehicle achieves the acceleration or deceleration commanded by the controller. The effect of the time constant τ can be observed in Figure 11, where the controller acceleration results directly from the configured common spacing parameters, whereas the actual acceleration shows a smoother change corresponding to the vehicle dynamics. Dots in Figure 11 indicate the earliest detected moment of actual braking, which, as can be observed, occurs later than the moment specified in the configuration. This means that the platoon responds with a delay to the scheduled braking event at $t = 17$ s. Probably due to the modeled vehicle dynamics.

7.5.4 Signal-to-Interference-plus-Noise Ratio (SINR)

In Figure 12, the SINR at the time of the attack remains similar to its pre-attack value. Moreover, `node[3]`, which is located farthest from the leader, records an even higher SINR level than the other vehicles at certain point. The high SINR provides highly favorable communication conditions, yet a collision between vehicles still occurs. This is because the problem is not the quality of the communication channel, but the outdated information delivered to the controllers.

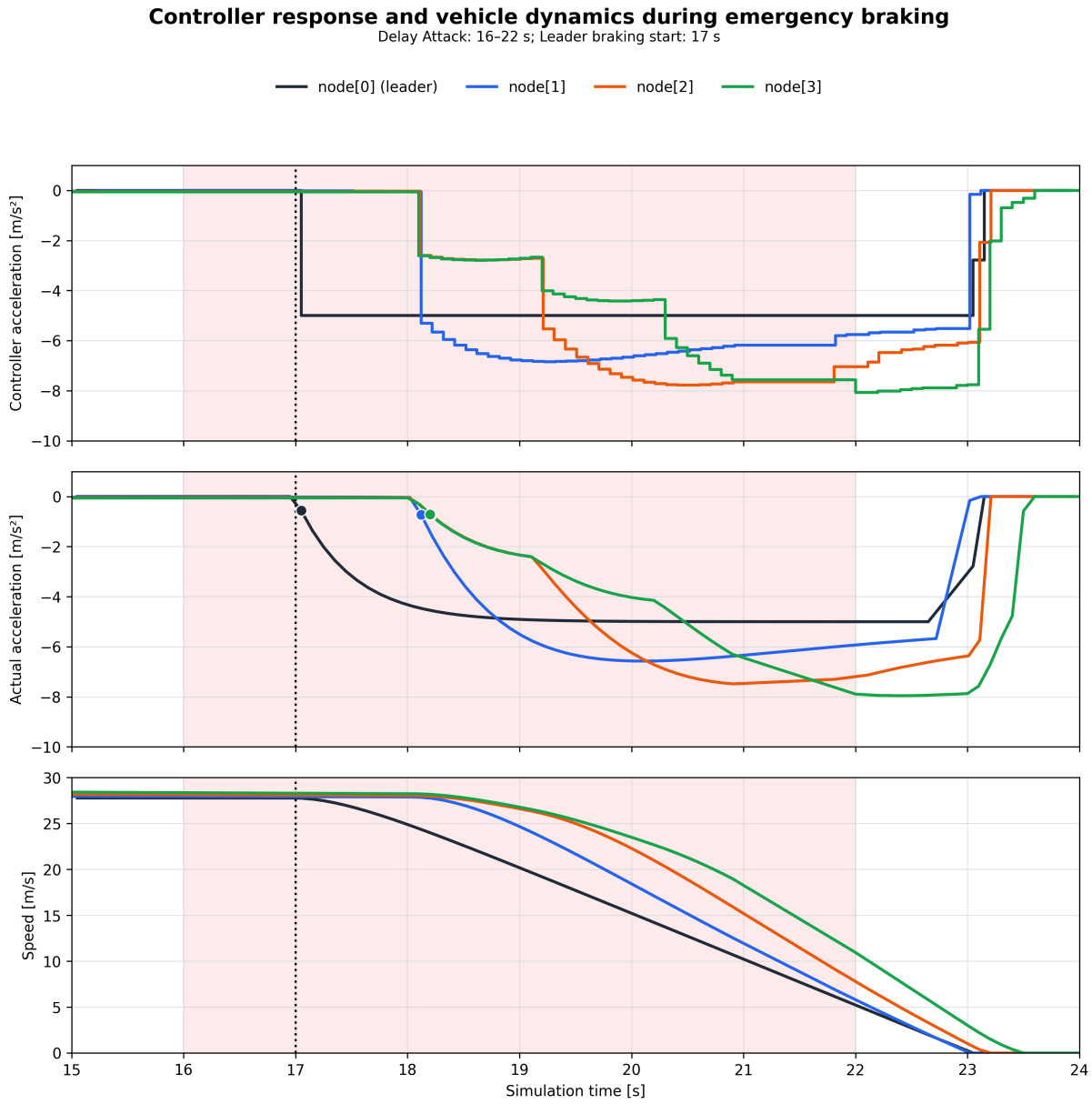

Figure 11: Controller response and vehicle dynamics during the platoon braking maneuver. The figure presents the acceleration commanded by the controller, the actual vehicle acceleration, and the vehicle speed for the leader and the three following vehicles. The markers indicate the earliest detected moments at which the actual acceleration crossed the adopted braking threshold. The difference between the commanded and actual acceleration illustrates the gradual physical response represented by the modelled vehicle and powertrain dynamics.

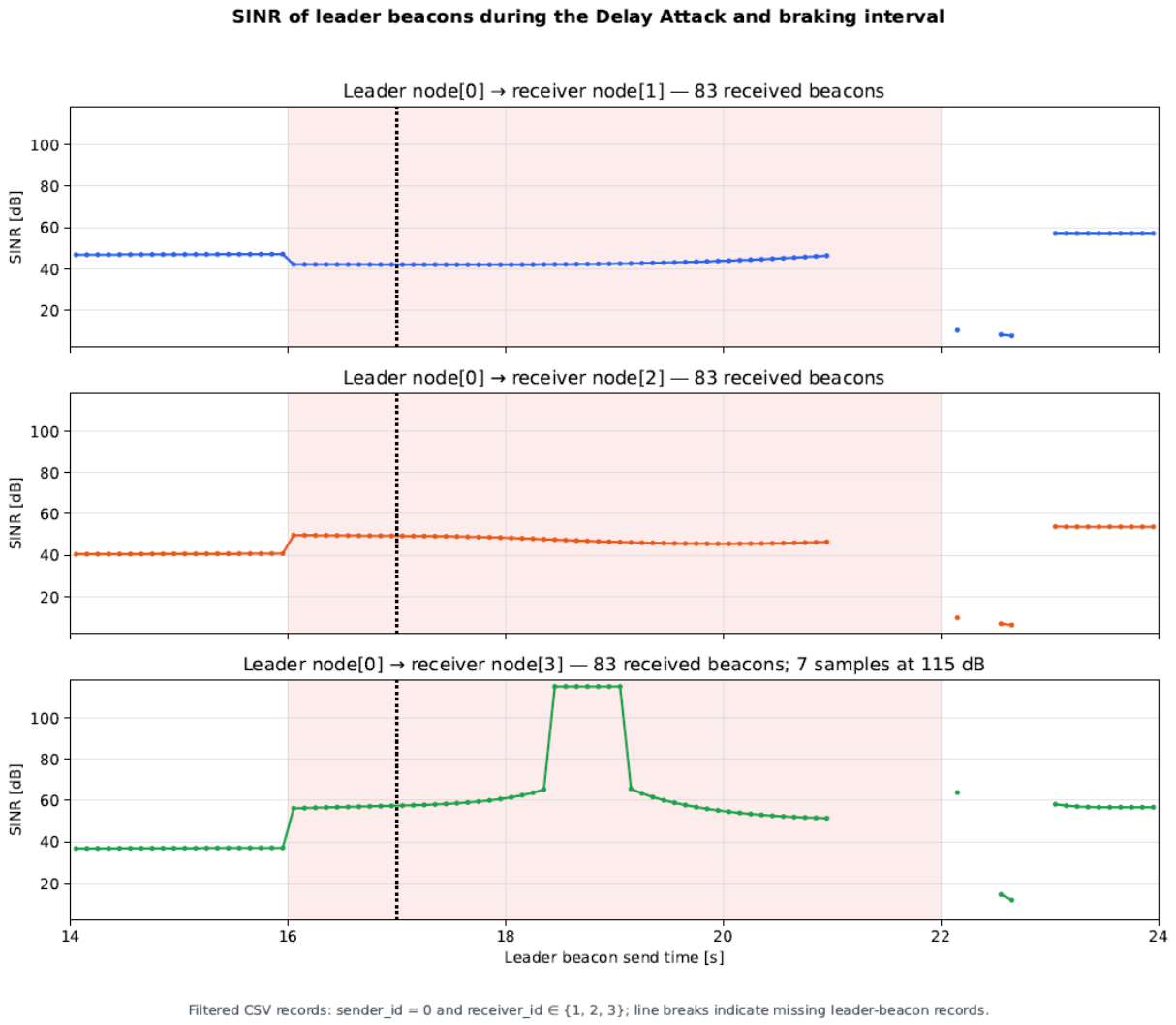

Figure 12: SINR values of leader beacons received by the three following vehicles during the Delay Attack and braking interval. Each panel represents a direct communication link from the leader, `node[0]`, to one of the following vehicles. Each marker corresponds to one successfully received leader beacon. Breaks between line segments indicate intervals for which no corresponding leader-beacon records were present in the receiver-side CSV files.

8 Results

Collected data do not allow to define a general safe-delay threshold for all braking platoons in different traffic conditions. However the data do allow to define such threshold for a specific simulation run. Thesis author emphasizes that a single safe-delay threshold for braking platoons would be an oversimplification. Nevertheless resulst obtained in this reserach points out towards the existence of scenario-specific safe-delay thresholds.

The results show that platoon safety cannot be described by a single universal delay threshold. The collision risk depended simultaneously on the delay value, attack initiation time, controller, spacing, braking intensity, and platoon length. Vehicle speed had the smallest impact. This section supports these hypotheses with collected data and describes the results in greater detail.

8.1 Delay Threshold in the Braking Scenario

In the Braking Scenario, 500 ms was the lowest tested communication delay at which collisions occurred. No collisions were observed for the tested delay values of 100, 200, 300, and 400 ms. Therefore, for configuration specified below, the transition between collision-free and collision-producing delay values lies within the interval of $400 - 500$ ms.

In the Braking Scenario delay of 500 ms represents the threshold at which vehicle collisions begin to occur. Below this threshold no collisions are observed. As shown in Fig. 10, the platoon of four CAVs using Rajamani's CACC model with CDG startegy, leader deceleration of -5 m/s^2, maintaining 5m gaps (inter-vehicle distance of 5 m) is able to tolerate communication delays up to 0.4 s without causing vehicle collisions. The platoon was cruising with the leader speed set to 100 km/h, respectively 80km/h. 500 ms threshold applies to both cruising speeds. At a delay of 500 ms, 11 of the 60 runs resulted in a collision at 80 km/h, and 11 of the 60 runs resulted in a collision at 100 km/h. For both speeds, collisions occurred when the attack started between $t = 16.0$ s and $t = 17.0$ s, inclusive. No collisions occurred when the 500 ms attack started between $t = 17.1$ s and $t = 21.9$ s. The collision occurred later at 100 km/h than at 80 km/h.

For the attack to be effective, it must be initiated before the platoon leader begins the braking maneuver. In the example presented above, attacks injected after the leader had already started braking were ineffective Fig. 10.

For the Braking scenario, a critical communication delay threshold of 0.5 s was identified. This threshold defines the maximum communication delay that can be tolerated without causing vehicle collisions. In this particular scenario, i.e. in a sudden braking situation, exceeding this threshold leads to unsafe platoon operation and the possible occurrence of vehicle collisions.

Table 10: Collision outcomes for injected delays of 100–500 ms. Fixed settings: CACC controller, four vehicles, 5 m spacing, initial speed of 80 km/h, and leader braking initiated at $t = 17$ s with a prescribed acceleration of -5 m/s^2. Each attack lasted 6 s. For each delay, the same 60 attack initiation times were tested from 16.0 to 21.9 s in steps of 0.1 s, giving 300 runs in total.

Attack start time [s]	Relative to braking onset [s]	Start times per delay	**Injected delay [ms]**				
			100	**200**	**300**	**400**	**500**
16.0–17.0	−1.0 to 0.0	11	N	N	N	N	**C**
17.1–21.9	+0.1 to +4.9	49	N	N	N	N	N
Runs with a collision / total runs		60	0/60	0/60	0/60	0/60	11/60
Collision rate			0%	0%	0%	0%	18.33%

C: collision recorded; **N**: no collision recorded. Each row groups tested initiation times with identical outcomes. Ranges refer only to the sampled times at 0.1 s intervals. Collision rate denotes the proportion of the 60 tested initiation times that produced a collision, not an estimated real-world collision probability.

8.2 Factors Affecting Platoon Safety under Delay Attacks

Experimental data indicate that vehicle spacing has the significant impact on collision during delaying attacks. While a 10-meter spacing eliminated collisions in smaller platoons (4, 5 and 6 vehicles), larger platoons (7, 8 vehicles) still experienced non-zero collision rates. Multiple simulations of the braking scenario showed that the timing of the attack injection is crucial. In the braking scenario, the platoon leader began braking after 17th second of the simulation, but the Delay attack started already in the 16th second. So the system is then attacked before a critical braking maneuver occurs. The data shows that in simulations (with artificial delay of 500 ms and above) starting between 16th and 17th second delay always led to a collision (Fig 13). This was a collision between the leader and the second car in the platoon. Therefore, there was a very short but critical window before braking began, when the platoon was most vulnerable and susceptible to delay attack. This was proven by over 35000 simulations in PLEXE on both CACC (Rajamani) controller and PLOEG controller.

Communication delay had the second greatest impact on platoon safety. For the CACC controller, the collision rate increased from 10.56% at a delay of 500 ms to 18.89% at 900 ms, while for the PLOEG controller, it increased from 49.86% to 72.29%. The results indicate that increasing delay significantly impairs the vehicles's ability to maintain stability and a desired distance during emergency maneuvers (Table 11).

Attack start time [s]	Relative to braking onset [s]	Start times per delay	**Injected communication delay [ms]**						
			500	**600**	**700**	**800**	**900**	**1000**	**1100**
16.0–17.0	−1.0 to 0.0	11	C	C	C	C	C	C	C
17.1	+0.1	1	N	N	N	N	C	C	C
17.2–21.9	+0.2 to +4.9	48	N	N	N	N	N	N	N
Runs with a collision / total runs		60	11/60	11/60	11/60	11/60	12/60	12/60	12/60

C: collision recorded; **N**: no collision recorded.

Figure 13: Collision outcomes as a function of attack initiation time and injected communication delay. Fixed settings: CACC controller, four vehicles, 5 m spacing, initial speed of 80 km/h, and leader braking initiated at $t = 17$ s with a prescribed acceleration of -5 m/s^2. Each attack lasted 6 s. For every delay value, the same 60 attack initiation times were tested, from 16.0 to 21.9 s in steps of 0.1 s, giving 420 runs in total. Rows group tested initiation times with identical outcomes; the displayed ranges do not imply testing at intermediate times between the 0.1 s samples.

Controller	***Delay***	Vehicles	Spacing / headway	Initial speed (km/h)	Leader braking acceleration (m/s^2)	Runs	Runs with collisions	Collision rate
CACC	***500 ms***	8	5, 8, 10 m	80, 100	−5, −6, −7, −8	1440	152	10.56%
CACC	***600 ms***	8	5, 8, 10 m	80, 100	−5, −6, −7, −8	1440	203	14.10%
CACC	***700 ms***	8	5, 8, 10 m	80, 100	−5, −6, −7, −8	1440	240	16.67%
CACC	***800 ms***	8	5, 8, 10 m	80, 100	−5, −6, −7, −8	1440	263	18.26%
CACC	***900 ms***	8	5, 8, 10 m	80, 100	−5, −6, −7, −8	1440	272	18.89%
PLOEG	***500 ms***	8	0.3, 0.5, 0.8 s	80, 100	−5, −6, −7, −8	1440	718	49.86%
PLOEG	***600 ms***	8	0.3, 0.5, 0.8 s	80, 100	−5, −6, −7, −8	1440	819	56.88%
PLOEG	***700 ms***	8	0.3, 0.5, 0.8 s	80, 100	−5, −6, −7, −8	1440	894	62.08%
PLOEG	***800 ms***	8	0.3, 0.5, 0.8 s	80, 100	−5, −6, −7, −8	1440	945	65.63%
PLOEG	***900 ms***	8	0.3, 0.5, 0.8 s	80, 100	−5, −6, −7, −8	1440	986	68.47%
PLOEG	***1000 ms***	8	0.3, 0.5, 0.8 s	80, 100	−5, −6, −7, −8	1440	1016	70.56%
PLOEG	***1100 ms***	8	0.3, 0.5, 0.8 s	80, 100	−5, −6, −7, −8	1440	1041	72.29%

Common settings: Leader braking begins at 17.0 s. Attack duration: 6 s. Attack initiation times: 16.0–21.9 s in 0.1 s increments (60 initiation times).

Runs per row: 3 spacing settings × 2 speeds × 4 braking accelerations × 60 attack initiation times = 1440. Within each controller's campaign, the same parameter combinations are used at every delay.

Controllers: CACC denotes Rajamani's controller with constant-distance spacing; PLOEG denotes Ploeg's controller with time-headway spacing.

Collision rate: Runs with at least one collision divided by all runs in the row, multiplied by 100%. CACC results for 1000 and 1100 ms are omitted because corresponding verified eight-vehicle data were unavailable.

Table 11: Collision occurrence at different injected delays, with the evaluated configuration sets shown explicitly.

The third most significant factor was the inter-vehicle gap. For the CACC controller, reducing the distance between vehicles from 10 m to 5 m increased the collision rate from 5% to 18.33%, corresponding to a 266.67% increase. The data show that small gaps significantly increase the platoon's susceptibility to the effects of communication delays (Table 12).

Controller	***Gap / headway***	Injected delay (ms)	Vehicles	Initial leader speed (km/h)	Leader braking acceleration (m/s^2)	Runs	Runs with collisions	Collision rate
CACC	***5 m***	500	8	80, 100	−5, −6, −7, −8	480	88	18.33%
CACC	***8 m***	500	8	80, 100	−5, −6, −7, −8	480	40	8.33%
CACC	***10 m***	500	8	80, 100	−5, −6, −7, −8	480	24	5.00%
PLOEG	***0.3 s***	500	8	80, 100	−5, −6, −7, −8	480	243	50.63%
PLOEG	***0.5 s***	500	8	80, 100	−5, −6, −7, −8	480	278	57.92%
PLOEG	***0.8 s***	500	8	80, 100	−5, −6, −7, −8	480	197	41.04%

Common attack settings: Attack initiation times: 16.0–21.9 s in 0.1 s increments (60 distinct times). Each attack lasts 6 s.

Runs per row: 60 attack initiation times × 2 speeds × 4 braking accelerations = 480. Within each controller's campaign, every spacing setting is evaluated using the same combinations of these parameters.

Controllers and spacing: CACC denotes Rajamani's controller with constant-distance spacing in metres. PLOEG denotes Ploeg's controller with time-headway settings in seconds, not fixed distances.

Collision rate: Percentage of runs in the row with at least one recorded collision.

Table 12: Collision occurrence for different spacing settings at an injected delay of 500 ms in eight-vehicle platoons.

Braking intensity also had a significant impact. For CACC, the collision rate increased from 8.33% at 5 m/s^2 braking to 13.06% at 8 m/s^2, while for PLOEG, it increased from 39.44% to 55.28%. More aggressive braking therefore led to more collisions (Table 13).

Controller	***Leader braking acceleration (m/s^2)***	Injected delay (ms)	Vehicles	Spacing / headway	Initial leader speed (km/h)	Runs	Runs with collisions	Collision rate
CACC	**−5**	500	8	5, 8, 10 m	80, 100	360	30	8.33%
CACC	**−6**	500	8	5, 8, 10 m	80, 100	360	36	10.00%
CACC	**−7**	500	8	5, 8, 10 m	80, 100	360	39	10.83%
CACC	**−8**	500	8	5, 8, 10 m	80, 100	360	47	13.06%
PLOEG	**−5**	500	8	0.3, 0.5, 0.8 s	80, 100	360	142	39.44%
PLOEG	**−6**	500	8	0.3, 0.5, 0.8 s	80, 100	360	183	50.83%
PLOEG	**−7**	500	8	0.3, 0.5, 0.8 s	80, 100	360	194	53.89%
PLOEG	**−8**	500	8	0.3, 0.5, 0.8 s	80, 100	360	199	55.28%

Common settings: Leader braking begins at 17.0 s. Attack initiation times: 16.0–21.9 s in 0.1 s increments (60 distinct times). Each attack lasts 6 s.

Runs per row: 60 attack initiation times × 2 speeds × 3 spacing settings = 360. Within each controller's campaign, every braking acceleration is evaluated using the same combinations of these parameters.

Controllers and spacing: CACC denotes Rajamani's controller with constant-distance spacing in metres. PLOEG denotes Ploeg's controller with time-headway settings in seconds.

Collision rate: Percentage of runs in the row with at least one recorded collision.

Table 13: Collision occurrence for different leader braking accelerations at an injected delay of 500 ms in eight-vehicle platoons.

Another factor was platoon size. Increasing the number of vehicles from 5 to 8 increased the collision rate for CACC from 6.88% to 10.56%, corresponding to a 53.54% increase (Table 14). In general, larger platoons exhibited higher collision rates. However, this trend was not observed in all cases. For example, the collision rate remained unchanged when the platoon size was increased from 5 to 6 vehicles. Thus, increasing platoon size did not always result in higher collision rates. The results suggest that the impact of platoon size should not be considered in isolation, but rather together with other factors such as communication delay, braking intensity, inter-vehicle spacing, and speed.

Controller	***Platoon size (vehicles)***	Injected delay (ms)	Spacing (m)	Initial leader speed (km/h)	Leader braking acceleration (m/s^2)	Runs	Runs with collisions	Collision rate
CACC	***5***	500	5, 8, 10	80, 100	−5, −6, −7, −8	1440	99	6.88%
CACC	***6***	500	5, 8, 10	80, 100	−5, −6, −7, −8	1440	99	6.88%
CACC	***7***	500	5, 8, 10	80, 100	−5, −6, −7, −8	1440	152	10.56%
CACC	***8***	500	5, 8, 10	80, 100	−5, −6, −7, −8	1440	152	10.56%

Controller: Rajamani's CACC controller with constant-distance spacing. Platoon size includes the leader.
Common settings: Leader braking begins at 17.0 s. Attack initiation times: 16.0–21.9 s in 0.1 s increments (60 distinct times). Each attack lasts 6 s.
Runs per row: 60 attack initiation times × 3 spacing settings × 2 speeds × 4 braking accelerations = 1440. The same combinations of these parameters are evaluated for each platoon size.
Collision rate: Percentage of runs in the row with at least one recorded collision.

Table 14: Collision occurrence for different CACC platoon sizes at an injected delay of 500 ms.

For eight-vehicle platoons subjected to an injected delay of 500 ms, changing the initial leader speed from 80 to 100 km/h produced only small differences in the aggregate collision rates. For CACC, the collision rate increased from 10.00% to 11.11%, while for PLOEG it decreased from 50.14% to 49.58%. Each speed was evaluated using 720 runs with the same controller-specific combinations of spacing settings, braking intensities, and attack initiation times (Table 15).

Controller	***Initial leader speed (km/h)***	Injected delay (ms)	Vehicles	Spacing / headway	Leader braking acceleration (m/s^2)	Runs	Runs with collisions	Collision rate
CACC	***80***	500	8	5, 8, 10 m	−5, −6, −7, −8	720	72	10.00%
CACC	***100***	500	8	5, 8, 10 m	−5, −6, −7, −8	720	80	11.11%
PLOEG	***80***	500	8	0.3, 0.5, 0.8 s	−5, −6, −7, −8	720	361	50.14%
PLOEG	***100***	500	8	0.3, 0.5, 0.8 s	−5, −6, −7, −8	720	357	49.58%

Common settings: Leader braking begins at 17.0 s. Attack initiation times: 16.0–21.9 s in 0.1 s increments (60 distinct times). Each attack lasts 6 s.
Runs per row: 60 attack initiation times × 3 spacing settings × 4 braking accelerations = 720. Within each controller's campaign, both speeds are evaluated using the same combinations of these parameters.
Controllers and spacing: CACC denotes Rajamani's controller with constant-distance spacing in metres. PLOEG denotes Ploeg's controller with time-headway settings in seconds.
Collision rate: Percentage of runs in the row with at least one recorded collision.

Table 15: Collision occurrence at two initial leader speeds for eight-vehicle platoons subjected to an injected communication delay of 500 ms.

The analysis also revealed significant differences between controllers. For the 8-vehicle platoon, the collision rate for CACC was 10.56%, while for PLOEG it reached 49.86%, which means 372.37% more collisions in the case of PLOEG (Table 16).

Controller	Runs	Collisions	Collision rate
CACC (8 cars, 500 ms)	1440	152	10.56%
PLOEG (8 cars, 500 ms)	1440	718	49.86%

Table 16: Overall collision comparison between CACC and PLOEG controllers

8.3 Time-Series Dataset for Machine Learning-Based Collision Prediction

This section investigates whether machine-learning models can predict an upcoming collision using beacon-level time-series data collected from a four-vehicle CACC platoon. Separate predictions were generated to determine whether an ego vehicle would collide with its immediate predecessor within the next 1, 2, or 3 seconds. The analysis considered local radar measurements, the current state of the ego vehicle, and delayed information received through V2V communication. The section presents the collected data, evaluated prediction models and their performance. Beacon campaign tests if ML models can predict collisions in 1s, 2s, and 3s using radar and outdated beacon data. Collected data involved only CACC (with CDG) and for four vehicles-platoon.

8.3.1 Collision Prediction Models

Each received beacon was stored as a separate row in the time-series dataset. In addition to information transmitted by the sender, every row contained the state of the receiving vehicle, radio-channel measurements, beacon timing information, and local radar measurements recorded by the receiver. Time series data contains the following columns: Table 17.

Table 17: Columns included in the beacon-level dataset.

No.	Column name
1	`run_id`
2	`receiver_id`
3	`sender_id`
4	`message_id`
5	`send_time_s`
6	`receive_time_s`
7	`message_age_s`
8	`sender_position_x`
9	`sender_position_y`
10	`sender_speed_mps`
11	`sender_acceleration_mps2`

Continued on the next page

Table 17 – continued from the previous page

No.	Column name
12	sender_controller_acceleration_mps2
13	receiver_position_x
14	receiver_position_y
15	receiver_speed_mps
16	receiver_acceleration_mps2
17	sinr_linear
18	sinr_db
19	received_power_dbm
20	bitrate_bps
21	radar_gap_m
22	radar_relative_speed_mps

Each row represents a single beacon received by the node. Run ID corresponds to the experiment ID from ComFASE (Ex ID) run-level CSV file. IDs and timestamps were used to associate messages with simulation runs, senders, receivers, and reception events. The sender and receiver fields described the vehicle states available when a beacon was received, whereas the SINR and received-power fields characterized the corresponding V2V transmission. The final two columns contained the local inter-vehicle gap and relative-speed measurements obtained from the radar of the receiving vehicle.

The collected beacon-level data were used to investigate whether information available to following vehicles could support the prediction of an upcoming collision. The author conducted the analysis using custom Python scripts. The objective was to determine whether machine-learning models could predict whether an ego vehicle would collide with its immediate predecessor within the next one, two, or three seconds using information from the ego vehicle's radar and potentially delayed V2V beacons.

The analysed campaign comprised 7200 simulation runs. Its configuration is presented in Table 18. The available input data included the current radar measurements, the current state of the ego vehicle, the currently received beacon, the five previously received beacons from the immediate predecessor, derived features such as time-to-collision (TTC) and the rate of gap reduction. A new sample was generated whenever an ego vehicle received a beacon from its immediate predecessor. Depending on the evaluated model variant, the sample contained radar-based features, V2V-based features, or a combination of both.

Separate binary labels were generated to indicate whether the ego vehicle would be involved as the collider in the first recorded collision within the next one, two, or three seconds. Samples

Table 18: Configuration of the CACC braking campaign with beacon-level data collection.

Parameter	**Value**
Scenario	Braking
Controller	Rajamani CACC
Spacing policy	Constant distance gap
Vehicles per platoon	4
Number of platoons	1
Number of lanes	1
Leader braking start time	17 s
Commanded braking magnitude	$\{5, 6, 7, 8\}$ m/s^2
Initial speed	$\{80, 100\}$ km/h
Desired inter-vehicle gap	$\{5, 8, 10\}$ m
Attack type	Delay attack
Configured injected delay	$\{100, 300, 500, 700, 900\}$ ms
Attack initiation time	16.0–21.9 s, in steps of 0.1 s
Number of attack initiation times	60
Attack duration	6 s
Attack end time	Attack initiation time + 6 s
Sender-side attack enabled	Yes
Receiver-side attack enabled	Yes
Controller parameter C_1	0.5
Damping ratio ξ	1
Configured `cacc` ω_n	0.2 Hz
Engine time constant τ	0.5 s
Beacon generation interval	0.1 s
Transmission power	100 mW
Bitrate	6 Mbit/s
Receiver sensitivity	−94 dBm
Configured noise floor	−95 dBm
Simulation time limit	60 s
Simulation runs	$60 \times 5 \times 4 \times 2 \times 3 = 7200$
Beacon-to-experiment association	`run_id` equals the experiment number

were generated for node[1], node[2], and node[3]. For each ego vehicle, only beacons received from its immediate predecessor were used:

- node[0] → node[1];
- node[1] → node[2];
- node[2] → node[3].

Three main types of models were developed according to the source of the input data, and each model was trained separately for prediction horizons of 1, 2, and 3 seconds. The three primary variants were based on radar data, V2V communication data, and a combination of radar and V2V data, as summarized in Table 19. Random Forest models were additionally trained for the radar-based and combined variants.

Table 19: Model variants according to the source of input data.

Model type	Input data	Objective
radar	Ego radar and ego state	Physical-state-based collision prediction
v2v	Information received in V2V beacons	V2V-data-based collision prediction
fused	Ego radar, ego state, and V2V data	Collision prediction using both sources

A single sample corresponded to the moment in which the ego vehicle received a beacon from its direct predecessor. In that moment, script processed the delayed V2V information transmitted by the preceding vehicle, the current radar measurement obtained by the ego vehicle, the current state of the ego vehicle, and the history of the five previously received beacons.

The radar-based model used the following input features:

- radar gap (m);
- radar relative speed (mps);
- ego-vehicle speed and acceleration;
- previous values of the radar-measured gap and relative speed;
- the rate of gap reduction;
- an approximate time-to-collision (TTC).

The V2V-based model used features derived from the received beacons and the receiver state, including:

- message age (s);

- the speed and acceleration reported by the sender;
- the acceleration requested by the sender's controller;
- the receiver's current state;
- signal-to-interference-plus-noise ratio (SINR);
- received signal power;
- the history of previously received beacons.

The combined model, denoted as `fused`, simultaneously received the radar-based and V2V-based features.

For each observation time t, three binary labels indicated whether a collision would occur within the next 1, 2, and 3 s, respectively. For example, if the collision was recorded at $t_{\text{collision}} = t + 1.7$ s, the labels were:

- collision within the next 1 s: 0 (no);
- collision within the next 2 s: 1 (yes);
- collision within the next 3 s: 1 (yes).

The collision time was known only to the data-preparation script because it analysed completed simulations and used this information to generate the ground-truth labels. The collision time, collision state, and other future information were not provided to the models as input features during either training or testing.

The dataset was divided using complete simulation runs rather than individual samples. Consequently, all samples originating from the same simulation run were assigned exclusively to the training, validation, or test set. No simulation run was shared between these subsets.

A start-time-based split was subsequently applied. The models were trained using runs associated with selected attack initiation times and tested using runs with different attack initiation times that had not been observed during training. This procedure reduced the risk of data leakage between the training and test sets and provided a more demanding evaluation of the models' ability to predict collisions under previously unseen temporal conditions.

The complete dataset contained 7,200 simulation runs, of which 571 resulted in a collision. Approximately 70% of all runs were assigned to the training set, 15% to the validation set, and 15% to the test set. Among the 571 collision runs, 408 were assigned to the training set, 102 to the validation set, and 61 to the test set. This corresponded to the following distribution:

- training set: $408/571 \approx 71.45\%$;

- validation set: $102/571 \approx 17.86\%$;
- test set: $61/571 \approx 10.68\%$.

Thus, 571 represents the total number of collision runs in the complete dataset, whereas 61 represents the number of collision runs used for the final evaluation of the models. The distribution of collision runs did not exactly follow the intended 70/15/15 split because the data were divided according to complete attack initiation times rather than the collision outcome.

For each prediction horizon of 1, 2, and 3 seconds, the six methods listed in Table 20 were evaluated.

Table 20: Methods compared for each prediction horizon.

Method	**Description**
`ttc`	Simple physics-based rule
`radar_tree`	Single decision tree using radar data
`v2v_tree`	Single decision tree using V2V data
`fused_tree`	Single decision tree using radar and V2V data
`radar_random_forest`	Random Forest using radar data
`fused_random_forest`	Random Forest using radar and V2V data

Therefore, the evaluation covered a total of

$$3 \text{ prediction horizons} \times 6 \text{ methods} = 18 \text{ method–horizon combinations.} \tag{8}$$

8.3.2 Event-Level Evaluation

In addition to sample-level performance, the methods were evaluated at the level of complete collision events. A collision event was considered detected if the model generated at least one timely warning for the colliding vehicle within the corresponding prediction horizon. Table 21 presents the number of detected collisions, event-level recall, false alarms, and mean warning lead time for TTC, the radar-based decision tree, and the radar-based Random Forest.

The test set contained 61 collision events and 3,240 run–vehicle pairs. The radar-based decision tree detected all 61 collisions for each prediction horizon. Its mean first-warning lead time was 0.929 s, 1.673 s, and 2.017 s for the 1-, 2-, and 3-second horizons, respectively. False alarms occurred in 42 (1.30%), 47 (1.45%), and 129 (3.98%) run–vehicle pairs.

The radar-based Random Forest detected 59 of 61 collisions for the 1-second horizon and 58 of 61 collisions for the 2- and 3-second horizons. Its corresponding mean first-warning lead times were 0.908 s, 1.610 s, and 1.879 s. False alarms occurred in 10 (0.31%), 26 (0.80%), and 46 (1.42%) run–vehicle pairs.

Table 21: Collision-detection results at the event level.

Model	Horizon	Detected collisions	Event recall	False alarms*	Mean warning lead time
TTC	1 s	61/61	100%	53 (1.64%)	0.922 s
Radar tree	1 s	**61/61**	**100%**	42 (1.30%)	0.929 s
Radar RF	1 s	59/61	96.72%	**10 (0.31%)**	0.908 s
TTC	2 s	61/61	100%	109 (3.36%)	1.634 s
Radar tree	2 s	**61/61**	**100%**	47 (1.45%)	1.673 s
Radar RF	2 s	58/61	95.08%	**26 (0.80%)**	1.610 s
TTC	3 s	61/61	100%	154 (4.75%)	1.980 s
Radar tree	3 s	**61/61**	**100%**	129 (3.98%)	2.017 s
Radar RF	3 s	58/61	95.08%	**46 (1.42%)**	1.879 s

* Number of non-collision run–vehicle pairs for which at least one false warning was generated; 3,240 test pairs were evaluated.

The radar-based Random Forest achieved sample-level F1-scores of 0.913, 0.879, and 0.803. The V2V-based decision tree achieved F1-scores of 0.097, 0.184, and 0.173. The fused Random Forest achieved F1-scores of 0.894, 0.824, and 0.749 for the 1-, 2-, and 3-second horizons, respectively.

9 Discussion

The experimental results indicate that communication delay, gap/headway, and braking deceleration were more important than vehicle speed. Platooning safety is more dependent on available reaction space than on vehicle speed itself. Even with aggressive braking at 8 m/s^2, a larger gap of 10 m significantly improved platooning safety.

In the braking scenario, the timing of the injection attack was clearly important. Attacks injected before braking resulted in more collisions between the leader and the vehicle behind it. Attacks injected after braking initiated significantly fewer collisions in the simulation with the CACC (Rajamani) controller and in the simulation with the PLOEG controller. The PLOEG controller, although designed for robustness and resistance to communication outages, is less resistant than CACC to delay attacks. Increasing inter-vehicle distances improves the security of the platoon using the CACC (Rajamani) controller. In the case of the PLOEG controller, increasing the H parameter (i.e., the time gap), results in a less significant improvement in terms of platoon safety under simulated delay attack conditions compared to the CACC (Rajamani) controller.

The experimental results indicate that the impact of platoon size on platoon safety depended strongly on other operating conditions, such as inter-vehicle spacing, vehicle speed, braking intensity, and injected communication delay. The results did not show that larger platoons always produced more collisions in every simulation scenario.

In some scenarios with a 5-meter inter-vehicle distance, all parameters except platoon size were left unchanged. In these scenarios with relatively small inter-vehicle distances, different platoon size values generated identical collision counts. This suggests that the platoon was already operating under safety-compromised conditions where the influence of platoon size became less important than the extremely limited reaction time available during emergency braking maneuvers.

In some scenarios increasing inter-vehicle distances (for example from 8 m to 10 m) resulted in lowering collision occurrence in the smaller platoons (up to 6 vehicles) to zero while larger platoons still generated collisions under the same simulation parameters in the same braking scenario. This suggests that larger platoons are more vulnerable to braking disturbance propagation during delayed communication conditions. Larger platoons appeared to be more sensitive to disturbance amplification and degradation of string stability during emergency braking maneuvers.

At the same time, the results suggest that platoon size alone should not be treated as an isolated, single safety factor. It should be rather analyzed as a correlated factor. Results of this research shows that focusing on each parameter independently can lead to non-factual conclusions and some of these could be eventually oversimplified when analyzing parameters separately.

The results also show that the timing of the attack is a critical factor, as delays introduced immediately before an emergency braking maneuver may have significantly more severe consequences than delays occurring during normal platoon operation. The vulnerability of the platoon is highly dependent on the operational state of the system at the time of attack execution. Attacks injected immediately before emergency braking were more effective than attacks launched during the braking maneuver itself. In some simulations, contrary to the initial expectation, collisions did not predominantly emerge at the tail of the platoon as a result of disturbance amplification. Instead, collision events most frequently involved the first follower vehicle and the platoon leader, indicating that the immediate reaction to delayed braking information may be more critical than downstream propagation effects in the evaluated scenarios. Disturbance amplification does not necessarily imply rear-end collisions at the tail.

A distinct threshold effect was observed with respect to attack activation time. Almost all attacks launched before or at the onset of leader braking (17.0 s) resulted in collisions, while attacks activated at 17.1 s or later resulted in almost no collisions. This observation suggests that the most safety critical communication messages are those transmitted during the transition from cruising to emergency braking. Delaying these initial braking messages appears to be more disruptive than delaying messages exchanged after the braking maneuver has already begun.

In case of Ploegs controller, the results indicate that increasing inter-vehicle spacing alone may be insufficient to compensate for severe communication delays. Under larger delays (above 700 ms and more), vehicles react based on outdated state information, which limits the safety benefits normally associated with higher headway values.

This thesis provides new insights into the vulnerability of cooperative vehicle platoons to communication delay attacks by demonstrating that attack effectiveness depends not only on the magnitude of the communication delay, but also on the operational state of the platoon at the time of attack execution. The results revealed a critical vulnerability window immediately preceding emergency braking, during which communication delays of 500 ms and exceeding 500 ms (i.e. 500ms and above) consistently resulted in collisions, whereas attacks launched after braking had already begun were substantially less effective. Collisions frequently occurred between the platoon leader and the first following vehicle, indicating that the initial response to delayed braking information may be more critical than disturbance propagation under the evaluated conditions when the whole platoon is under attack. Furthermore, increasing the inter-vehicle spacing alone did not guarantee a proportional improvement in resilience against sophisticated delay attacks, suggesting that safety depends on the interaction between communication delay, attack timing, deceleration or braking intensity, controller design, vehicle speed, and platoon size rather than on any single parameter. Although the findings are based on simulation experiments using ComFASE, they identify mechanisms that can support the design of more resilient cooperative driving controllers and motivate future research on attack-aware control strategies and communication delay mitigation in connected automated vehicle platoons.

10 Conclusions

This thesis demonstrates that:

- the effectiveness of communication delay attacks depends not only on the magnitude of the communication delay but also on the operational state of the platoon.
- a critical vulnerability window exists immediately before a braking maneuver, during which communication delays exceeding approximately 500 ms consistently resulted in collisions under the evaluated conditions.
- platoon safety is the result of the interaction of multiple parameters.
- the Ploeg controller does not always provide the expected improvement in safety.
- the Rajamani CACC controller exhibited greater tolerance to communication delay attacks than the Ploeg controller under the evaluated conditions.
- a larger platoon does not always result in more collisions.
- communication delay attacks lead to braking disturbance amplification.
- a comprehensive experimental campaign was conducted, and an experimental framework was developed using ComFASE.
- more than 35000 simulations were executed, producing datasets that can be used for training and validating machine learning models, as well as for the development and evaluation of more resilient cooperative driving controllers.

In summary, the research has shown that the CACC (Rajamani) is more resistant to delay attacks during critical braking maneuvers. The PLOEG loses stability more quickly. A larger gap increases the robustness of the CACC(Rajamani) controller. Speed is a measurable, but not the dominant factor in determining the security of the platoon under sophisticated delay-type cyber attacks. The other mentioned factors showed a greater impact on the platoon's safety. Vehicle speed influenced collision occurrence, but its impact was smaller than that of communication delay, inter-vehicle spacing, and braking intensity. This study also proved that increasing the distances between vehicles to improve platoon safety and string stability under delay attack conditions has poor effectiveness when using a PLOEG controller. However, to better understand the nature and use cases of both controllers, more scenarios should be tested. The scope of this work has been narrowed to two very specific road scenarios. So further research is needed, which could potentially result in the creation of a generic tool for malicious attack and misbehavior prediction. Research strategies that can contribute to the development of such tools are described in the Future Research Directions section.

11 Future Research Directions

Future Research Directions

Future work plans to expand the current research on collision risk in autonomous vehicle platooning during a delay attack. Thesis author plans to train models on data from one controller (CACC) and then test them on data from a second controller (PLEOG). This will allow us to assess the feasibility of building universal threat detection systems independently of the controller used in a specific platooning implementation. Also more data needs to be collected to train a model capable of distinguishing between natural communication delays and sophisticated delay attacks on CAV platoons. Ultimately, the developed system could be used as part of an intelligent Intrusion Detection System (IDS) for platooning autonomous vehicles, capable of continuous analysis of traffic parameters and the prediction of potential security threats in real time. To achieve this, an evaluation of a Real-Time Intrusion Detection System (RT-IDS) using machine learning is required. The computational efficiency, accuracy, and precision of classical ML and deep learning models should be evaluated in future work.

References

[1] S. Hasan, "On transient communication outages among collaborating connected and automated vehicles," Ph.D. dissertation, Mälardalen University, Västerås, Sweden, 2023. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:mdh:diva-63845

[2] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson, "Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass," *SAE International Journal of Commercial Vehicles*, vol. 7, no. 2, 2014, DOI: 10.4271/2014-01-2438.

[3] S. Hasan, S. Girs, and E. Uhlemann, "Characterization of transient communication outages into states to enable autonomous fault tolerance in vehicle platooning," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 101–129, 2023, DOI: 10.1109/OJITS.2023.3237958.

[4] M. Malik, M. Maleki, P. Folkesson, B. Sangchoolie, and J. Karlsson, "ComFASE: A tool for evaluating the effects of V2V communication faults and attacks on automated vehicles," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2022, pp. 185–192, DOI: 10.1109/DSN53405.2022.00029.

[5] Y. Zhang, Z. Xu, K. Ma, X. Chou, and J. Mu, "A real-time resilient framework for cav platoons under cyberattacks: Integrating vmd-assisted detection with compensation strategies," *Vehicular Communications*, vol. 60, p. 101035, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209626000379

[6] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for veins," in *Proceedings of the 6th IEEE Vehicular Networking Conference (VNC 2014)*, Paderborn, Germany, Dec. 2014, pp. 53–60, DOI: 10.1109/VNC.2014.7013309.

[7] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 260–265, DOI: 10.1109/ITSC.2011.6082981.

[8] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, Jul. 2000, DOI: 10.1109/87.852914.

[9] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "VeReMi: A dataset for comparable evaluation of misbehavior detection in VANETs," in *Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 254. Cham, Switzerland: Springer, 2018, pp. 318–337, DOI: 10.1007/978-3-030-01701-9_18.

[10] European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Performance Evaluation of Self-Organizing TDMA as Medium Access Control Method Applied to ITS; Access Layer Part," ETSI, Sophia Antipolis, France, Technical Report ETSI TR 102 862 V1.1.1, Dec. 2011. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/102800_102899/102862/01.01.01_60/tr_102862v010101p.pdf

[11] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Networks*, vol. 61, pp. 33–50, Jun. 2017, DOI: 10.1016/j.adhoc.2017.03.006.

[12] International Telecommunication Union, "Connected automated vehicles," ITU Radiocommunication Sector, Geneva, Switzerland, Report ITU-R M.2534-0, 2023. [Online]. Available: https://www.itu.int/pub/R-REP-M.2534-2023

[13] K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, "On the ability of the 802.11p MAC method and STDMA to support real-time vehicle-to-vehicle communication," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 902414, 2009, DOI: 10.1155/2009/902414.

[14] European Telecommunications Standards Institute, "Intelligent transport systems (its); its-g5 access layer specification for intelligent transport systems operating in the 5 ghz frequency band," ETSI, ETSI EN 302 663 V1.3.1, 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.03.01_60/en_302663v010301p.pdf

[15] IEEE, "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-Channel Operation," IEEE, Tech. Rep. IEEE Std 1609.4-2016, 2016, DOI: 10.1109/IEEESTD.2016.7435228.

[16] H. Zhang, W. Tian, and J. Liu, "Improving edca for efficient channel access in vehicular communications," *IEEE Communications Magazine*, 2018, DOI: 10.1109/MCOM.2018.1800121.

[17] S. A. Malik, M. A. Shah, S. A. Khan, M. Jahanzeb, U. Farooq, and M. A. Khan, "Performance evaluation of ieee 802.11p mac protocol for vanets," *Australian Journal of Basic and Applied Sciences*, vol. 4, no. 8, pp. 4089–4098, 2010. [Online]. Available: https://www.ajbasweb.com/old/ajbas/2010/4089-4098.pdf

[18] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. New York, NY, USA: Springer, 2012, DOI: 10.1007/978-1-4614-1433-9.

[19] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-stable CACC design and experimental validation: A frequency-domain approach," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4268–4279, Nov. 2010, DOI: 10.1109/TVT.2010.2076320.

[20] C. Flores and V. Milanés, "Fractional-order-based acc/cacc algorithm for improving string stability," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 381–393, 2018, DOI: 10.1016/j.trc.2018.07.026.

[21] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, Marseille, France, 2008, pp. 1–10, DOI: 10.4108/ICST.SIMUTOOLS2008.3027.

[22] A. Varga, "The omnet++ discrete event simulation system," in *Proceedings of the European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, Jun. 2001. [Online]. Available: https://www.researchgate.net/publication/228460521_The_OMNET_discrete_event_simulation_system

[23] D. Eckhoff and C. Sommer, "A multi-channel ieee 1609.4 and 802.11p edca model for the veins framework," in *Proceedings of the 5th ACM/ICST International Conference on Simulation Tools and Techniques*. ACM, 2012. [Online]. Available: https://www.david-eckhoff.net/pdf/eckhoff2012multichannel.pdf

[24] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582, DOI: 10.1109/ITSC.2018.8569938.

[25] K. G. Lim, C. H. Lee, R. K. Y. Chin, K. B. Yeo, and K. T. K. Teo, "Sumo enhancement for vehicular communication development," *ICTACT Journal on Communication Technology*, vol. 8, no. 4, pp. 1625–1631, Dec. 2017, DOI: 10.21917/ijct.2017.0239.

[26] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011, DOI: 10.1109/TMC.2010.133.

[27] Veins Project. Veins documentation. [Online]. Available: https://veins.car2x.org/documentation/

[28] European Telecommunications Standards Institute (ETSI), "Intelligent transport systems (its); security; security header and certificate formats; release 2," ETSI, ETSI Technical Specification ETSI TS 103 097 V2.2.1, Mar. 2026, reference RTS/ITS-005122. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103000_103099/103097/02.02.01_60/ts_103097v020201p.pdf

[29] *IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Application and Management Messages*, IEEE Std. IEEE Std 1609.2-2022, DOI: 10.1109/IEEESTD.2023.10075082.

[30] S. Ucar, S. C. Ergen, and O. Ozkasap, "Ieee 802.11p and visible light hybrid communication based secure autonomous platoon," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8667–8681, 2018, DOI: 10.1109/TVT.2018.2840846.

[31] A. Alipour-Fanid, M. Dabaghchian, H. Zhang, and K. Zeng, "String stability analysis of cooperative adaptive cruise control under jamming attacks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, 2017, pp. 157–162, DOI: 10.1109/HASE.2017.39.

[32] D. S. Berger, F. Gringoli, N. Facchi, I. Martinovic, and J. Schmitt, "Gaining insight on friendly jamming in a real-world IEEE 802.11 network," in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 105–116, DOI: 10.1145/2627393.2627403.

[33] J. Kamel, M. Wolf, R. W. van der Heijden, A. Kaiser, P. Urien, and F. Kargl, "VeReMi Extension: A dataset for comparable evaluation of misbehavior detection in VANETs," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6, DOI: 10.1109/ICC40277.2020.9149132.

[34] Z. H. Khattak, J. Rios-Torres, and M. D. Fontaine, "Impact of communications delay on safety and stability of connected and automated vehicle platoons: Empirical evidence from experimental data," *IEEE Access*, vol. 11, pp. 128 549–128 568, 2023, DOI: 10.1109/ACCESS.2023.3324056.

[35] C. Zhao, X. Duan, L. Cai, and P. Cheng, "Vehicle platooning with non-ideal communication networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 18–32, 2021, DOI: 10.1109/TVT.2020.3046165.

[36] J. Thunberg, N. Lyamin, K. Sjöberg, and A. Vinel, "Vehicle-to-vehicle communications for platooning: Safety analysis," *IEEE Networking Letters*, vol. 1, no. 4, pp. 168–172, 2019, DOI: 10.1109/LNET.2019.2929026.

[37] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (cacc)," in *2017 IEEE Vehicular Networking Conference (VNC)*, 2017, pp. 45–52, DOI: 10.1109/VNC.2017.8275598.

[38] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (cavs)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6240–6259, 2022, DOI: 10.1109/TITS.2021.3085297.

[39] A. Hermann, J.-N. Remmers, D. Eisermann, B. Erb, and F. Kargl, "Veremi nextgen: A dataset for evaluating misbehavior detection systems in vanets," in *2026 IEEE Vehicular Networking Conference (VNC)*, 2026, pp. 1–8, DOI: 10.1109/VNC69225.2026.11629123.

[40] G. Dodig-Crnkovic, "Scientific methods in computer science," in *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden*, Skövde, Sweden, 2002, pp. 126–130. [Online]. Available: https://www.es.mdu.se/publications/375-Scientific_Methods_in_Computer_Science

A Simulation Environment and Research Data

The installation guide, simulation configuration notes, and run-level dataset are available in the project repository:
https://github.com/sebastian91b/cav-platooning-delay-dataset.

The beacon-level dataset is available from the repository's Releases page:
https://github.com/sebastian91b/cav-platooning-delay-dataset/releases.