Realistic Simulation of Cooperative Adaptive Cruise Control (CACC) Degradation Under Random Packet Loss

Cian Johnston

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Future Networked Systems)

Supervisor: Mélanie Bouroche

September 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Cian Johnston

September 14, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Cian Johnston

September 14, 2020

To my parents, Deirdre, Brian, Fergus, and Sacha.

To my loving wife, Sandra.

To my grandfather, Roy, and father-in-law, James.

You are in my heart, now and forever.

Acknowledgments

I would like to express my heartfelt gratitude to both my supervisor Mélanie, whose counsel and patience were instrumental in completing this work, and to the authors on which this work was based. I would also like to acknowledge the dedication and perseverence of both students and staff members this year, who strove bravely forward in the face of uncertainty and adversity.

Cian Johnston

University of Dublin, Trinity College
September 2020

Realistic Simulation of Cooperative Adaptive Cruise Control (CACC) Degradation Under Random Packet Loss

Cian Johnston, Master of Science in Computer Science

University of Dublin, Trinity College, 2020

Supervisor: Mélanie Bouroche

Abstract: Connected and Autonomous Vehicles (CAVs) are an active area of research due to their potential benefits to traffic flow on public roads. The vast majority of recent studies in this area utilise mixed-traffic simulations to gauge the potential effect of CAVs on traffic flow. These vary from microsimulations, which investigate the interactions between small numbers of vehicles in great detail, to macrosimulations, which investigate the emergent behaviours of thousands of vehicles.

This thesis analyses a number of recent studies of CAV effects on traffic flow, and evaluates them along various criteria, including the addition of realistic network simulation, and the complexity of the road network.

We find that CAV studies which include realistic network simulation alongside microscopic vehicle simulation tend to be based on less complex road network scenarios, and that CAV studies based on more complex road network scenarios omit realistic network simulation. We also find that no studies to date have been performed on the effect of packet loss on the traffic flow effects of CAVs in mixed-traffic scenarios including a complex road network. We hypothesise that there is no significant benefit of realism to be gained by performing network simulation in a complex road network,

and that varying the reliability of the network in such a simulation should have little effect on traffic flow.

To refute this hypothesis, a recent study involving mixed-traffic macrosimulation on a realistic road network is extended to include realistic network simulation.

This required evaluating a number of longitudinal CAV models, state-of-the-art vehicular and network simulation software, and then designing and running a number of large-scale simulation experiments designed to falsify this hypothesis. These experiments utilised a realistic vehicle simulator coupled with a realistic vehicular network simulation framework. The following parameters were all varied: amount of traffic (low/high), the longitudinal CAV controller, CAV market penetration rate, and packet drop rate.

These simulations required days of real computer time to run, and produced gigabytes of vehicle trajectory data. Some modifications to the underlying simulation software were also required in order to run these simulations.

The results of these simulations were then used to evaluate the hypothesis, and it was not found possible to reject the hypothesis based on these results. While no significant effects of packet loss on the rate of CAV traffic flow were found in this study, more investigation is required before further conclusions can be drawn.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent times, increased demands on national road networks and increased urbanization have contributed to issues with traffic congestion, causing wasted time for commuters and increased fuel consumption due to engine idling. Per census data from the Central Statistics Office (2016), over 200,000 commuters reported spending an hour or more in their daily commute. Connected and Autonomous Vehicles (CAVs) are expected to provide major improvements in traffic flow, thereby allowing more vehicles to share the existing road network and reducing congestion.

Traffic waves, also referred to as kinematic waves or stop-and-go waves, are of the most visible artifacts of traffic congestion at peak travel times. These are an emergent phenomenon that arise due to amplifications of speed perturbations by human drivers — one vehicle following another at close distance will be affected by changes in the leading vehicle's velocity due to the combination of the inter-vehicle spacing and human reaction time. Such changes in velocity could be due to another vehicle changing lane, or due to another vehicle performing an unexpected maneuver, or simply due to driver error. Edie (1961) studied this phenomenon in traffic entering and exiting the Lincoln tunnel in New York City and attempted to model this effect as a car-following model (CFM).

A later study by Sugiyama et al. (2008) showed that standing traffic waves can be reproduced very simply with a sufficiently dense ring of human-driven vehicles all attempting to keep a constant velocity. This experiment consisted of a ring of 22 human-driven vehicles driving at the relatively sedate speed of 30 km/h in a 230m

circle, with almost uniform inter-vehicle spacing. Small perturbations in vehicle velocities developed over time and were amplified by the human drivers. Larger induced variations in vehicle speed were also observed to cause this traffic wave propagation effect.

Recent research has shown that CAVs are able to reduce or eliminate these traffic waves entirely. Stern et al. (2018) repeated the experiments of Sugiyama et al. (2008) with the addition of a single vehicle with instrumentation sufficient to simulate the behaviour of a CAV. This experiment showed that this single CAV was capable of elminating the traffic waves that naturally arose from the human-driven vehicles.

However, such field experiments require not only trained professional drivers and specialised hardware, but also a sufficient space in which to perform these experiments safely. Such experimental setup requires substantial outlay and expense. A more fiscally efficient option is to perform simulations of such experiments instead. Dedicated vehicle simulation sofware such as AIMSUN and SUMO allow researchers to perform simulations of arbitrary numbers of vehicles on an arbitrarily defined road network. Work by Makridis et al. (2018) simulated a physically realistic road network using real traffic data, and found that CAVs were able to improve traffic flow, while autonomous vehicles (AVs) lacking vehicular communications were shown to have an overall negative effect on traffic flow.

A great number of such simulations have been peformed under varying conditions, but the effects of delays and/or packet loss is an important consideration while simulating CAVs. Performing accurate simulation of inter-vehicle communication (IVC) may allow greater fidelity and more accurate results, but this adds an extra layer of complexity and overhead to an already computationally expensive operation. It therefore is appropriate to ask under which circumstances it is neccessary to perform network-layer simulation in concert with a vehicular simulation.

The rest of this paper is organised as follows: Chapter 2 provides background information and related work, and poses the research question to be addressed in this work. Chapter 3 details the experimental methodology, setup, and simulations performed. Chapter **??** presents the results of these experiments, and Chapter 5 summarises conclusions and future work.

Chapter 2

Background and Related Work

This chapter is structured as follows: Section 2.1 provides a short overview of the background of autonomous vehicles (AVs), connected autonomous vehicles (CAVs), and vehicular networking. Section 2.2 draws a number of requirements from the background and related work. Section 2.3 evaluates related work against these requirements. Finally, Section 2.4 presents an open research question to be addressed in this work.

2.1 Background

The field of connected automated vehicles (CAVs) has been an active area of research in recent decades, owing to increases in both urban traffic density, recent advances in the miniaturization of computers and sensors, and in no small part to recent advances in the field of wireless communications. This section provides short overviews of developments in autonomous vehicles (2.1.1), connected autonomous vehicles (2.1.2), and vehicular networks (2.1.3). For further reading, Wang et al. (2019) provides a comprehensive survey of vehicular networking, and Wang et al. (2020) a comprehensive survey of developments in cooperative longitudinal controllers.

2.1.1 Autonomous Vehicles (AVs)

Autonomous vehicles might be defined as vehicles which are able to operate without requiring a human driver. However, this is an overly simplistic definition, as there are many partial levels of automation possible. For example, the now-ubiquitous cruise

control (CC) feature is one such method of partial automation, but CC on its own is not sufficient to operate a vehicle without a human driver either from a practical or from a legal standpoint. Per SAE International [1], autonomous vehicles can be classified in levels ranging from 0 to 5, with 0 representing complete manual control, and 5 representing full automation requiring no human control. Under this classification scheme, a vehicle with CC could be classified under level 1 (Driver Assistance). Newer vehicles such as the Tesla Model S with advanced driver assistance (marketed as "Autopilot") could be classified as level 2 (Partial Automation) or possibly level 2.5, with both the driver and vehicle taking responsibility for monitoring the driving environment.

Per Thrun (2010), the architecture of autonomous vehicles includes three major systems: perception, actuation, and control. The perception systems include sensors such as accelerometers, LASER range-finders, and cameras, all of which collect data in real-time from the vehicle's environment and deliver it to the control system. The actuation systems connect to vehicle components such as acceleration, braking, and steering, and can perform actions such as accelerating or decelerating, or changing the vehicle's direction. The control system decides the appropriate actuation levels based on inputs provided from the perception systems. The author also presents the DARPA Grand Challenges as a major driving force behind recent developments in autonomous vehicles. These competitions were held in the years of 2004, 2005, and 2007, and initially required autonomous vehicles to complete a course located in the Mojave desert, building to more complex scenarios in later iterations. A significant cash prize was also provided to the winning team.

However, autonomous vehicles on their own may actually have a detrimental effect on traffic flow. Makridis et al. (2018) performed simulations of the effect of various market penetrations of autonomous vehicles (modelled after Shladover et al. (2012)) on traffic flow. The authors determined that introducing AVs into a simulated road network caused a significant reduction in traffic flow, and attributed it to differences in behaviour between HDVs and AVs — human drivers will naturally take some calculated measures of risk while driving, while AVs would be engineered with "risk-averse" behaviours in order to avoid collisions wherever possible. Without the benefit of inter-vehicular communications, this necessitates a more conservative driving style.

[1] SAE International Standard J3016, available: `https://www.sae.org/standards/content/j3016_201806/`, accessed 2020-09-06

Additionally, autonomous vehicles are by no means infallible. A fatal accident involving a Tesla Model S and another vehicle occurred on May 7, 2016, when "neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky".[2] The NHTSA performed a through investigation and "did not identify any defects in the design or performance of the AEB or Autopilot systems of the subject vehicles." [3] and noted that "the tractor trailer should have been visible to the Tesla driver for at least seven seconds prior to impact." Drivers are ultimately responsible for ensuring they are fully in control of their vehicle at all times, regardless of the driver assistance systems available to them.

2.1.2 Connected Autonomous Vehicles (CAVs)

Connected vehicles were introduced as early as the late 20th century – Shladover (2007) details the work of the PATH project regarding CAVs and automated highway systems (AHS). Of particular note here is the work on coordinated longitudinal control of multiple vehicles, culminating in a demonstration of an eight-vehicle platooning system as part of the NAHSC's "Demo '97" event held in San Diego, in August 1997.

The available radio resources and available hardware were somewhat limited at the time of this experiment. Chang et al. (1991) describe the initial implementation of the inter-vehicular communications (IVC) in use by the PATH project. The 902–920 MHz unlicensed frequency bands were used for inter-vehicle communication, and spread spectrum transmission was used to minimise the effects of interference on IVC. However, the radio hardware used was limited to half-duplex mode, making IVC between larger numbers of vehicles complex due to timing issues. Bandwidth was also limited in comparison to modern standards, being limited to 19.2 Kbaud while operating in asynchronous mode.

Despite these limitations, the project did show that connected automated vehicles are able to synchronise their speed and acceleration closely, thereby increasing vehicle density while providing a "excellent ride quality comparable to that of extremely good human drivers" (Rajamani et al., 2000, p. 707).

[2]Statement from Tesla, Inc. available: `https://www.tesla.com/en_IE/blog/tragic-loss`, accessed 2020-09-06.

[3]NHTSA Investigation PE 16-007, available: `https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF`, accessed 2020-09-06.

Later experiments have shown that even relatively small numbers of CAVs can effect a measurable improvement on traffic flow. An experiment performed by Stern et al. (2018) reproduced the ring experiment of Sugiyama et al. (2008). A number of vehicles, including one CAV, were driven in a single-lane ring. The CAV in this experiment was programmed to travel at a weighted average of the preceding vehicle's speed and the ego vehicle's desired speed, with the weighting depending on the distance between the CAV and the leading vehicle. This experiment showed that one CAV was able to significantly dampen the speed perturbations of up to 21 human-driven vehicles in a simulated congested traffic environment. Reducing velocity perturbations of vehicles not only improves traffic flow, but also reduces unnecessary braking, thereby reducing fuel consumption and driver fatigue.

2.1.3 Vehicular Networks

The early work outlined in 2.1.2 showed the potential gains of connected vehicles. Efforts began soon thereafter to develop reliable, scalable, and low-latency open vehicular communications standards, with a view to enabling both researchers and automobile manufacturers to build novel connected mobility solutions.

Wireless vehicular networks pose a particular set of challenges. The network topology is constantly changing due to the mobility of user equipment, and therefore a particular communication link may only be active for a very short time interval. Additionally due to the safety-critical nature of the use case, low latency and tolerance of path losses are hard requirements for vehicular networks. The 900 MHz frequency bands used by the PATH project would have proven problematic for wide-scale commercial implementations owing to not only their limited throughput and bandwidth, but also due to potential interference from other low-powered radio operators.

In 1997, the Intelligent Transport Society (ITS) of America filed a petition with the FCC to allocate spectrum in the 5.9 GHz frequency bands specifically for use in vehicular communications. Following the US Congress passing legislation the next year, these frequency bands were allocated by the FCC.[4] The European Commission followed suit in 2008, allocating spectrum in the 5.8 — 5.9 GHz frequency bands to

[4] https://www.fcc.gov/wireless/bureau-divisions/mobility-division/dedicated-short-range-communications-dsrc-service

ensure that "spectrum used by ITS cooperative systems should be made available in a harmonised way".[5] The dedication of these radio resources was a crucial step for the continued development of vehicular networking systems.

Three main families of vehicular networking standards have emerged to address use cases for vehicular networking. The first two are IEEE 1609 Wireless Access in Vehicular Environments (WAVE), and ETSI ITS-G5. These both rely upon the IEEE 802.11p (IEEE Standards Association (2010)) physical layer standard, an amendment to the existing 802.11 standards commonly known as Wi-Fi. Explicit provisions are made to guarantee latency in the order of tens of milliseconds, and to avoid the need establish a Basic Services Set (BSS) prior to exchanging information. These networking standards enable not only communications between vehicles and roadside infrastructure (V2I), but also allow vehicles to establish peer-to-peer communications with other vehicles (V2V).

Lin et al. (2010) performed experiments to compare the performance characteristics of 802.11a and 802.11p using physical networking hardware operating under both LOS and NLOS conditions. The measured contact duration, packet loss distribution, and communication throughput showed that 802.11p was able to maintain contact longer than 802.11a with significantly lower packet losses under both LOS and NLOS conditions. It is also notable that, in these experiments, the authors recorded that 802.11a was unable to maintain a connection while the terminals were moving at relative speeds above 20 Km/h. The authors attribute these improvements to both the increased Guard Interval (GI) and lack of authentication/association process in 802.11p. Given the magnitude of these improvements, it is difficult to imagine further development of ITSes proceeding without the underlying 802.11p protocol to handle the physical realities of IVC.

The third major emerging standard is referred to as Cellular V2X (C-V2X) or LTE-V, and was introduced in 3GPP Release 14. In contrast to 802.11p, C-V2X is based upon Long Term Evolution (LTE) cellular networking standards. According to Wang et al. (2019), the major advantages of this approach compared to 802.11p-based include 1) that no additional infrastucture is required, 2) that a high level of coverage can be offered, and 3) that stringent Quality of Service (QoS) guarantees can be supported, and 4) that later evolution to 5G protocols is possible.

[5] `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32008D0671&from=en`

Mannoni et al. (2019) performed a detailed comparison of the performance of the ITS-G5 (based on IEEE 802.11p) and C-V2X physical layer protocols, finding that C-V2X provided greater performance than ITS-G5 at lower congestion levels, but suffering from greater performance degradation at higher levels of user density. The authors also note that 802.11p is a more widely-studied and mature technology, while trials of C-V2X are still in relatively early stages.

Section Summary

This section provided short overviews of recent developments in autonomous vehicles (2.1.1), connected autonomous vehicles (2.1.2), and vehicular networks (2.1.3). As discussed in this section, CAVs offer the prospect of improvements in traffic flow, fuel efficiency, and driver comfort. Recent advances in vehicular networking protocols offer low-latency, reliable, and scalable vehicular communications with or without supporting infrastructure, opening up new avenues of research and development in connected automotive applications.

2.2 Problem Definition and Requirements

This section defines the problem area with which this work concerns, and details a number of requirements against which the related work is evaluated in Section 2.3.

As discussed in 2.1, CAVs show the potential to radically alter the nature of road traffic. In order to ensure this change happens smoothly, it is necessary to determine the large-scale effects their introduction is likely to have on road traffic. The results of such studies will not only inform future research, but also provide information to automobile manufacturers implementing CAV systems. As shown in Section 2.3.1, performing large-scale field experiments is not a feasible solution to this problem. Instead, studies on the long-term effects of CAVs on traffic flow turn to traffic simulations instead. Traffic simulations are a cost-effective and time-efficient way to rapidly test CAV algorithms and implementations in a virtual environment.

In order for the results of a simulation to be meaningful, they must fulfil certain requirements. A number of requirements the author considers of paramount importance are defined below:

1. **Simulation of traffic**: the work must simulate the flow of traffic in some way. Such simulation may be microscopic (simulating individual vehicles) in nature, or macroscopic in nature (simulating flows of vehicles), and may be done in any numeric analysis software (e.g. MATLAB or R), or in specialised vehicle simulation software (e.g. SUMO). This requirement exists as physical experiments on CAVs are prohibitely time-consuming and expensive to peform without access to specialised hardware, vehicles, and trained drivers.

2. **Complex road network**: the work must model a non-trivial section of an existing road network for which realistic traffic data can be gathered to use as a baseline. While simple 'highway' scenarios are useful for determining the capabilities of vehicle models or communication models, more realistic scenarios can provide insights into how particular CAV implementations may perform in reality.

3. **Varied Market Penetration Rate (MPR)**: CAVs, like many other driver assistance features, will gradually appear on public roads as individuals purchase new vehicles. Therefore, it is important for any study of the effect of CAVs or AVs on traffic flow to perform mixed-traffic simulations. Mixed-traffic simulations vary the proportion of (C)AVs and HDVs in varying proportions to simulate gradual introduction of the new vehicle types into normal traffic flow, and allow researchers to determine the relative strengths of the various effects different vehicle types may have on normla traffic flow, as well as the minimum MPR for which such effects are likely to be seen.

4. **Simulation of network**: the work must simulate inter-vehicle communication, taking into account hardware capabilities, software capabilities, and the physical characteristics of the communication medium. Wireless communications can be interrupted by a number of physical effects, including shadowing by obstacles, interference from other wireless signals, and signal attenuation over distance (per Otto et al. (2009)). A study of the effects of CAVs on traffic flow should therefore incorporate a sufficiently realistic physical layer networking model. This work will mainly focus on studies using the more widely-implemented and well-studied IEEE 802.11p (or similar) physical layer standards.

5. **Varied Packet Error Rate (PER)**: performing realistic network simulation will necessarily give a more complete understanding into how simulated CAVs will perform in physical situations. Realistic physical models of wireless communication can also give accurate approximations of the performance of IVC in physical environments. However, hardware and software failures can occur, or malicious actors can perform jamming attacks on networks. Thus, it is important to determine the performance characteristics of CAVs under various levels of degraded connectivity.

2.3 Related Work

This section outlines the related work in this area, and evaluates said work against the requirements outlined in Section 2.2. Part 2.3.1 evaluates previous work involving CAV field experiments with physical hardware. Part evaluates previous large-scale simulations of the effects of CAVs on traffic flow. Part 2.3.3 evaluates previous smaller-scale CAV simulations focused mainly on the networking layer. Part 2.3.4 evaluates studies comparing the performance of various CACC models under various conditions. These studies are summarised in Table 2.1.

2.3.1 CAV Field Experiments

As measurement is a prerequisite to simulation, researchers working on CACC systems will often perform physical experiments to ensure that networking protocols, hardware, and coordination algorithms function as expected.

A large-scale field experiment was performed by Schakel et al. (2010) on a public road, using a string of 50 vehicles equipped with an "Acceleration Advice Controller (AAC)". The AAC includes a CACC controller which is used to compute a preferred acceleration based on the time headway delay of preceding vehicles, and presents this to the driver. Drivers cooperating with the AAC were able to partially dampen traffic shockwaves induced at the head of the string of vehicles. The results of this experiment showed up to a 13% reduction in variations of traffic density. No mixed-traffic experiments were performed, as "The AAC was designed for 100% penetration rate and is not suitable for mixed traffic." (Schakel et al., 2010, p. 761)

Work by Bu et al. (2010) (in co-operation with the Nissan Motor Company) extended a commercially available ACC implementation with CACC capabilities, utilising dedicated DSRC hardware. Field tests showed that CACC-equipped vehicles were able to maintain a stable formation at shorter time gaps than the commercially developed ACC system. These experiments were limited to two CACC-enabled vehicles, but other field tests showed that the CACC-enabled vehicles were able to succesfully dampen manually induced time gaps. However, the authors do not mention effects of packet loss or delay on string stability.

Ploeg et al. (2011) designed a novel longitudinal control model for CACC, and performed field tests on vehicles utilising 802.11a radios in ad-hoc mode for IVC. They also investigated the relationship between communication delay and the minimum allowable inter-vehicle distance while maintaining string stability. Stability of the model was confirmed both theoretically and in field experiments, which showed that the model is string-stable with a time headway of 0.7s, given a communication delay of 150ms. Crucially, the authors also show the relationship between communication delay and minimum stable headway for their longitudinal model.

As such physical experiments are time-consuming to set up, and are prohibitively costly to perform at large scale, these experiments are limited to the order of tens of vehicles. Performing experiments with larger numbers of vehicles would be prohibitively expensive. Therefore, a great deal of literature elects to perform simulations either to complement, or sometimes even in lieu of, physical experiments. While simulations allow a great deal of control and flexbility, one must always bear in mind that a simulation may not correspond completely to reality.

2.3.2 Simulations of the effects of AVs and CAVs on Traffic Efficiency and Safety

This part details a number of simulations performed on the possible effects of the introduction of CAVs on traffic efficiency. As there will necessarily be a gradual introduction of CAVs to public roads, many of these studies perform mixed-traffic simulations, where the proportion of CAVs (or other vehicle types) to HDVs is varied. The same experiment may be varied with a different mix of traffic. These simulations generally involve large numbers of vehicles using specialised vehicle simulation software (see: Section

3.2).

Wu et al. (2018) performed numeric simulations using the Intelligent Driver Model (IDM) (Treiber et al., 2000). The authors modelled an optimization problem to determine the minimum neccessary rate MPR of autonomous vehicles, and showed that as low as 6% AV MPR can reduce traffic waves in a single-lane scenario. However, the model did not take vehicular communications into account, and assumed all of the vehicles acting as autonomous agents.

Mena-Oreja and Gozalvez (2018) performed a large-scale simulation of CAV platooning maneuvers on traffic flow. A multi-lane circular highway was modelled, and the proportion of CAVs to human-driven vehicles was varied to produce a mixed traffic simulation. The open-source platooning simulator Plexe (Segata et al., 2014) was used to handle simulated platooning maneuvers and inter-vehicle communication. While Plexe would have simulated realistic IVC, the authors do not perform additional investigation on the effects of IVC degradation on CAV interactions.

Makridis et al. (2018) performed a large-scale CAV simulation using a realistic road network. Using data from OpenStreetMap, a realistic model of the Antwerp ring road was created and imported into the AIMSUN[6] vehicle simulator. Real data of traffic counts at peak rush hour was used to generate a representative traffic flow for the ring road. Additionally, the authors adjusted the proportions of HDVs, AVs, and CAVs in a number of different scenarios, resulting in approximately 189 hours of simulated road traffic. As discussed previously, the authors showed that AVs lacking communication capabilities had an overall negative effect on traffic flow, while CAVs had an overall positive effect on traffic flow at higher MPRs. However, reliable IVC was also assumed here and no explicit simulation of the underlying networking protocols was performed.

Liu et al. (2020) performed a mixed-traffic simulation of strings of CACC-enabled vehicles in a highway scenario including a merge bottleneck, using the commercial AIMSUN simulation software. The authors showed increased vehicle capacity and decreased fuel consumption at a 40% CAV MPR. However, reliable IVC is assumed here; the effects of channel congestion with larger numbers of vehicles, and shadowing due to other vehicles, may impact on the traffic flow improvements cited in this study.

Gueriau and Dusparic (2020) investigated the effects CAVs have on both road safety and traffic flow. Similar to Makridis et al. (2018), the authors generated multiple

[6]https://www.aimsun.com/

complex road network scenarios (highway, motorway, and city), and used open data to generate realistic traffic flows. The SUMO vehicle simulator was then used to perform a number of realistic simulations of varying MPRs of CAVs. The authors also publish their data and results freely online in order to allow other researchers to build upon their work. [7] Reliable communications were assumed in the case of CAVs.

In summary, the studies described in this section focused on large-scale mixed-traffic simulations of hundreds (or thousands) of vehicles in order to determine overall trends of traffic flow with the introduction of (C)AVs. While these studies were able to show a number of useful results, almost all did not perform explicit network simulation, and those that did also did not perform further investigation of the effect of random packet loss on (C)AVs.

2.3.3 CAV Network Simulations

Larger-scale evaluations of VANETs have recently been performed as simulations, as these are far simpler to perform than obtaining the necessary physical vehicles and hardware to perform physical experiments. Simulations allow researchers to create arbitrary scenarios with large numbers of vehicles and, crucially, to vary simulation parameters as required. Collecting experimental data is also generally a much simpler matter in these cases compared to a physical experiment. In contrast to the studies detailed in Part 2.3.2, these studies focus on a smaller number of vehicles while performing explicit simulation of the networking and co-ordinating aspects of CAVs.

Lei et al. (2011) performed a realistic network simulation of the effect of random packet loss on the string stability of a simulated platoon of vehicles, using the SUMO vehicle simulator and the MiXiM extension to the OMNeT++ network simulator. The road network modelled consisted of a single-lane highway scenario along which a single platoon of ten vehicles travelled. Packets were dropped independently according to a uniform distribution. At increased levels of packet loss, the simulated vehicles were observed to overshoot their desired velocity to a greater extent. This effect was found to be mitigable by increasing the beaconing frequency, which has the unfortunate drawback of increased network traffic and increased number of packet collisions. Mixed traffic was not simulated.

[7] Available: `https://github.com/maxime-gueriau/ITSC2020_CAV_impact`

Ad-hoc networks differ from infrastructure networks in that there is no centralised point of access. Any user in range of the network may transmit to other users, but reception of a packet by all users of the network is not guaranteed. In such scenarios, vehicles may need to implement a Store-Carry-Forward (SCF) algorithm to propagate messages through the network. Simulations performed by Jia et al. (2014) studied the ability of platoons of vehicles to exchange information. Using the VEINS vehicular network simulator (Sommer et al., 2011), the researchers simulated a bidirectional highway scenario with a number of platoons of vehicles exchanging Basic Safety Messages (BSMs), measuring the delays transmitting BSMs between platoon leaders. Their findings showed that, in such a scenario, both the vehicle density and speed are important factors in the propagation of information throughout the network. However, measuring traffic flow efficiency was not an aim of this experiment, and a complex road network was not simulated.

Liu et al. (2018) analyzed the packet loss rate and throughput of a VANET operating under 802.11p using the open-source NS-2 network simulator. The number and speed of the vehicles were varied, and results showed a linear relationship between packet collisions and the number of vehicles in a VANET. Gonzalez and Ramos (2018) performed a similar study to investigate the loss characteristics of 802.11p-based VANETs. The authors used the NS-3 network simulator to simulate both a two-way highway and a crossroads intersection with varied numbers of vehicles (from 25 to 150), and varied the size of the CSMA/CA contention window (CW) [8] The authors similarly found that increased numbers of nodes on the network resulted in a higher packet loss percentage, with the majority of losses caused by hidden terminals. Increased packet losses entail retransmissions and delays, resulting in degraded communications for users of the network – an especially important consideration in vehicular networking.

Yao et al. (2020) propose distributed algorithms for controlling the roles assigned to vehicles in a platoon (i.e. leader or follower), and performs mixed-traffic simulations including realistic networking simulation at varying MPRs. A single lane ring road with one intersecting highway was modelled, and infrastructure-based communications (V2I) were used as opposed to peer-to-peer networking approaches (V2V).

[8]The contention window (CW) specifies a period for which a sender must wait for the transmission medium to be non-busy before attempting to transmit. A larger CW reduces the likelihood of collisions, but increases latency.

The simulations were evaluated along various dimensions including the average travel delay (the difference between the desired and actual arrival times) and the number of driving mode switches made by CAVs simulated (e.g. a platoon leader stepping down to become a follower). No random packet loss was simulated, and the simulated inter-vehicle communications could be classified as reliable owing to the infrastucture-assisted networking scenario.

Degraded communications may not just arise naturally from a congested communication medium — malicious actors may intentionally cause disturbances in a network for specific purposes. Additionally, faults in either hardware or software could unintentionally cause communications outages in vehicular networking scenarios. van der Heijden et al. (2017) investigated the effects of various attacks on platooning algorithms using Plexe (Segata et al., 2014). By performing jamming attacks on virtual platoons, as well as by feeding incorrect data to vehicles, the authors were able to cause numerous collisions between vehicles. The road network simulated was not complex, and the same proportion of attacker vehicles to subject vehicles was kept in all experiments. The work underlines the importance of security and redundancy when designing vehicular control systems.

In summary, simulations of VANETs have focused on smaller numbers of vehicles, but have placed particular note on the effects of communications degradation on the interactions between the networked vehicles. One should note the computational cost of network simulation — simulating n vehicles broadcasting packets at a frequency f results in a maximum of $n!f$ simulated network events to be processed per simulated second. Larger simulations with thousands of vehicles are computationally intensive and may take hours, or even days to run to completion. Given that simulations on this scale produce significant behavioural effects, it can be argued that at least approximate realistic network simulation should be performed with larger-scale CAV simulations.

2.3.4 Comparison of CACC models

The platooning experiments of the California PATH project were controlled by a specialised longitudinal control model. Rajamani (2011a) provides an in-depth explanation of this model, as well as proofs of string stability. Since then, other authors have designed alternative longitudinal control models, and compared the performance of these

models to those already extant.

Ploeg et al. (2013) proposed a novel CACC controller which aims to maintain string stability in the face of degraded inter-vehicle communications. The proposed novel controller performs an estimation of the leading vehicle's velocity, and uses this information in the case of communcation losses or delays from the leading vehicles. Numerical analysis and simulations are performed, and physical experiments are also performed to validate the stability and performance of the proposed controller. The performance of the proposed model was compared with the predecessor-following CACC model outlined by Ploeg et al. (2011). While numeric analyses were performed, neither vehicle simulations nor network simulations were performed in this work.

Terruzzi et al. (2017) investigated the effect of various platooning strategies on traffic flow, performing a comparison of the leader- and predecessor-following PATH CACC controller and the predecessor-following PLOEG controller. A multi-lane circular ring road was simulated with 1,080 vehicles travelling at varying target speeds, and varying levels of mixed traffic were simulated. The results of these simulations showed that a constant-time spacing policy was more effective in mitigating traffic shockwaves, while a constant-distance spacing policy effected a greater improvement on traffic flow. Realistic networking simulation was performed, but additional random packet loss was not simulated.

Santini et al. (2017) proposed a novel consensus-based CACC controller aimed at maintaining consistency in the face of network delays and packet losses. Numerical analyses were performed to confirm string-stability, and multiple simulations were performed using the Plexe vehicle platooning simulator. One such simulation included a road network consisting of a multi-lane highway, with existing flows of mixed traffic, alongside which a multiple-vehicle platoon was driven. The PATH CACC controller was used as a baseline for comparison. Both realistic vehicle and network simulation were performed, and random Bernoullian packet losses were introduced (up to 60%). However, larger numbers of platooning vehicles were not simulated, and the proportion of CAVs to HDVs was not varied.

Liu et al. (2019) investigated the performance characteristics of multiple CACC controllers under similar conditions. A scenario was constructed consisting of a straight one-way highway along which a single platoon of 8 vehicles travelled. The acceleration of the platoon leader was varied in a sinusoidal fashion, thereby creating traffic

shockwaves that propagated throughout the platoon. The abilities of CACC controllers to maintain the desired speed, acceleration, and inter-vehicle spacing in this scenario were compared. Realistic network conditions was performed, but no additional random packet loss was simulated.

From the above, it can be seen that a number of (simulated) performance comparisons have been made between different CACC implementations. These simulations have mostly been performed on a smaller scale, leaving open questions as to the behaviours of these models in larger-scale mixed-traffic scenarios. Additionally, the question remains of how different CACC degrade under increasing packet loss at a large scale.

2.4 Research Question

The preceding evaluation of related work alongside the requirements detailed in Section 2.3 is summarised in Table 2.1. To the best of the author's knowledge, no work exists that simultaneously fulfills all of the previously detailed requirements. It is evident that an opening exists for a new research question.

Given the computational complexity and cost of simulating network communications with large numbers of vehicles, it would be useful to define the specific circumstances under which it is important to perform detailed simulation of network communications. This thesis investigates the following hypothesis:

Realistic network simulation has no significant impact on large-scale simulations of the performance of connected and/or autonomous vehicles (CAVs) in physically realistic mixed traffic scenarios.

In order to test this hypothesis, the following chapter will focus on outlining an experimental methodology, building upon existing work.

Study	Traffic Simulation Performed?	Road Network Complexity	Varies MPR?	Network Simulation Performed?	Varies PER?
Lei et al. (2011)	Y	Simple	N	Y	Y
Jia et al. (2014)	Y	Simple	N	Y	N
Terruzzi et al. (2017)	Y	Complex	Y	Y	N
van der Heijden et al. (2017)	Y	Simple	N	Y	Y
Santini et al. (2017)	Y	Complex	N	Y	Y
Wu et al. (2018)	N	Simple	Y	N	N
Mena-Oreja and Gozalvez (2018)	Y	Simple	Y	Y	N
Makridis et al. (2018)	Y	Complex	Y	N	N
Liu et al. (2019)	Y	Complex	N	Y	N
Gueriau and Dusparic (2020)	Y	Complex	Y	N	N
Liu et al. (2020)	Y	Simple	Y	N	N
Yao et al. (2020)	Y	Simple	Y	Y	N

Table 2.1: Comparison of previous work according to requirements detailed in Section 2.2.

Chapter 3

Methodology

This chapter outlines the experimental methology to be used to test the hypothesis outlined in Section 2.4. Recall that this hypothesis depends on both network simulation and the behaviour of a CAV: both of these factors need to be accounted for in any experiment to test this hypothesis.

In order to simulate the behaviour of a CAV, a state-of-the-art model is required. Section 3.1 provides a brief description of a number of major CAV longitudinal controllers (also called CACC controllers), and outlines some common related terminology.

In order to test realistic network simulation, a realistic network simulator is required which is able to simulate physical effects such as signal attenuation and path loss. In order to test the behaviour of a simulated vehicle, specialised software is required which is able to effectively model and simulate the behaviour of a large number of vehicles in a given scenario. Section 3.2 provides an overview of a number of vehicle and network simulators, and details the specific choices made for this work.

Finally, as the behaviour of a CAV may depend on the information available at any given time, the network simulation and the vehicle simulation need to be integrated. Section 3.3 describes the experimental setup in detail, including any modifications made to the simulation software.

3.1 General Concepts

While many different CACC algorithms have been proposed in the last decades, they all share the same core concepts. This section outlines a number of general concepts relevant to CACC controllers. Part 3.1.1 describes *longitudinal and lateral control*. Part 3.1.1 describes *inter-vehicle spacing policies*. Part 3.1.2 describes *vehicle stability* and *string stability*. Part 3.1.3 describes *controller architectures*. Part 3.1.4 describes *controller topologies*. Finally, part 3.1.5 describes a number of major CACC controllers.

3.1.1 Latitudinal vs. Longitudinal Control

The overall goal of CACC is to minimise the inter-vehicle distances of a number of co-operating vehicles, also referred to as a *platoon*. This is generally accomplished in two phases: a *gap-closing* phase in which a vehicle accelerates to close the distance to its predecessor, and a *steady-state* phase in which a vehicle attempts to closely match the acceleration of its predecessor.

The above can be referred to as a *longitudinal control* strategy. *Latitudinal control* strategies seek to coordinate vehicles' latitudinal positions (such as lane-changing maneuvers), and are outside the scope of this paper. More general *platooning strategies* also seek to coordinate individual vehicles while performing various platoon-related operations, such as leaving, joining, or splitting a platoon, and are also outside the scope of this paper.

Inter-Vehicle Spacing Policy

Inter-vehicle spacing policies define the minimum spacing any two connected vehicles may maintain in a formation. These can be:

- **Constant Space Gap (CSG):** any given vehicle must not exceed a minimum safe distance from the vehicle in front, regardless of the speed of both vehicles. At higher speeds, this equates to less time for the following vehicle to react to a potential collision.
- **Constant Time Gap (CTG):** the minimum safe distance for a pair of vehicles is dependent on their relative velocities, analogous to the "two-second rule" for

human drivers described by the Road Safety Authority (2018). CTG policies result in increased inter-vehicular spacing at higher velocities, and consequently result in reduced traffic density. The function to calculate the desired spacing may incorporate multiple variables (such as the length of the preceding vehicle).

3.1.2 Vehicle and String Stability

It is important to differentiate between the concepts of *vehicle stability* and *string stability*. *Vehicle stability* refers to the ability of a controller to maintain a single vehicle at a constant velocity in stable environment, and *string stability* refers to the ability of a controller to dampen or eliminate speed perturbations of multiple vehicles.

Both ACC and CACC strategies seek to maintain a constant spacing (either distance or time) from the preceding vehicle whilst minimising the spacing error δ_i. Rajamani (2011b) defines this as

$$\delta_i = x_i - x_{i-1} + L_{des}$$

where x_i and x_{i-1} are the locations of the ego and preceding vehicles, respectively, measured from an inertial frame of reference, and L_{des} is the desired spacing for the ego vehicle.

A vehicle following controller provides individual vehicle stability if the spacing error converges to zero while the preceding vehicle maintains a constant velocity, satisfying the condition :

$$\ddot{x}_{i-1} \to 0 \Rightarrow \delta_i \to 0$$

where $\ddot{x}_{i-1}$ refers to the acceleration of the preceding vehicle.

Similarly, if the preceding vehicle is accelerating forwards or backwards, the spacing error is expected to be non-zero.

Where a number of vehicles controlled by CACC are travelling in sequence, it is important that the controller provide *string stability*. This guarantees that perturbations in spacing error are not amplified from a preceding vehicle to the following vehicle. Rajamani (2011b) provides in-depth explanations of the theory of string stability and the relevant proofs, and shows that ACC systems can provide string stability under a

CTG policy with a sufficient time gap, but cannot provide string stability under CSG policies, while CACC systems can provide string stability under both CTG and CSG policies.

3.1.3 Controller Architecture

(C)ACC systems generally operate under a multi-level controller system. For example, Rajamani (2011b) details an ACC controller consisting of:

- An upper level controller, which determines the desired vehicular acceleration that ensures both desired inter-vehicle spacing (according to sensor measurements) and string stability of the entire vehicle platoon are maintained.
- A lower-level controller calibrated to the vehicle's physical characteristics (e.g. torque, gear dynamics, etc.), and applies inputs as required by the upper-level controller. Rajamani (2011a) details an alternative version of this lower-level controller whish is able to adapt to unknown vehicle parameters online.

The various control strategies detailed later in this chapter can be viewed as upper level controllers, as they focus mainly on the upper-level dynamics of determining the desired acceleration of the ego vehicle while leaving the specific kinematic requirements of an individual vehicle to a lower-level controller. The specifics of lower-level vehicle controllers are out of scope of this work.

3.1.4 Controller Topologies

When calculating the desired velocity of an individual vehicle, there may be multiple sources of information available. For example, given a string of 4 consecutive vehicles, the fourth vehicle may have information about not only the speed and position of the third vehicle, but also the leading vehicle. The third vehicle may have information about both the second and fourth, but may not have information about the leading vehicle.

The CACC strategies detailed later in this chaper may fall into one or more of the following controller topologies:

- Leader- and predecessor-following, where information from both the platoon leader and the vehicle preceding the ego vehicle are taken into account when determining the ego vehicle's desired acceleration.
- Predecessor-following, where only information from the preceding vehicle is taken into account.
- Bidirectional, where both information from the preceding and following vehicles are taken into account.

Direct propagation of information from the leader to all followers may not be possible via vehicular communications due to path loss. For example, a platoon of vehicles turning around a corner in an urban environment may experience path loss due to shadowing from an intervening structure. Therefore, multi-hop routing may be required for a leader- and predecessor-following topology, where members of the vehicle platoon store, carry, and forward packets bound for known platoon members.

3.1.5 Overview of major CACC controllers

A brief description of a number of major CACC controllers with open-source implementations is provided in this section. The following is not an exhaustive list, but all of the below strategies are proven to be string-stable (refer to the respective publications for proofs).

PATH CACC

CACC as described in Rajamani et al. (2000) operates under a leader- and predecessor-following strategy, where information from both the platoon leader and the predecessor vehicle are used to determine the optimal speed and acceleration of the ego vehicle. This control strategy operates under a constant spacing policy, and was successfully demonstrated on an eight-vehicle platoon running on a dedicated 7.6 mile-long highway over a period of three weeks for 6–8 hours daily. It was shown to reliably maintain a constant inter-vehicle distance within an accuracy of 20 cm.

The PATH CACC control law defines the desired acceleration $\ddot{x}_{i_{des}}$ of the i-th vehicle

as follows (Rajamani et al., 2000):

$$\begin{aligned}\ddot{x}_{i_{des}} = (1 - C_1)\ddot{x}_{i-1} + C_1\ddot{x}_l \\ - (2\xi - C_1(\xi + \sqrt{\xi^2 - 1}))\omega_n\dot{\epsilon}_i \\ - (\xi + \sqrt{\xi^2 - 1})\omega_n C_1(\nu_i - \nu_l) - \omega_n^2\epsilon_i\end{aligned}$$

The three relevant control gains C_1, ξ, and ω_n control the weighting of the lead vehicle's speed, the damping ratio, and the controller bandwidth, respectively.

Ploeg CACC

Ploeg et al. (2011) outline a simpler CACC strategy which only takes the velocity of the preceding vehicle into account and seeks to establish a constant-time headway as opposed to a constant spacing. The authors opt for a constant-time spacing policy instead of a constant-distance spacing policy due to the improved string stability offered by constant-time spacing. The control law here seeks to minimise the spacing error of the i-th vehicle with respect to its predecessor $e_i(t)$, which is defined as:

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

In the above, $s_i(t)$ refers to the position of vehicle i at time t, L_i refers to the length of vehicle i, d_i refers to the distance between vehicles i and $i-1$, $d_{r,i}(t)$ refers to the desired distance between the ego vehicle and predecessor vehicle at time t, h refers to the desired time headway for all vehicles (assuming a homogeneous platoon), and $v_i(t)$ refers to the velocity of vehicle i at time t. Note that the above equation does not refer to the full control law; Ploeg et al. (2011) provides additional formulations of the control law based on error dynamics and linear systems theories, both of which are out of scope of this work.

Flatbed CACC

Ali et al. (2015) propose a leader- and predecessor-following CACC model based around the concept of a flatbed tow truck. In this model, vehicles in a given platoon are connected by virtual monodirectional spring-dampers. Each such spring-damper applies

a force to the follower vehicle that is proportional to the difference in velocities of the follower vehicle and the leader vehicle. The spacing policy is described as a "modified constant-time headway", which is proportional to the speed of the vehicle relative to the speed of the virtual flatbed towtruck.

The control law for the Flatbed CACC model defines the control input W_i as follows:

$$W_i = -k_a \ddot{x}_i + k_v \dot{e}_i + k_p \delta_i$$

In the above, x_i refers to the spacing of the i-th vehicle, e_i refers to the spacing error between the i-th vehicle and its predecessor, δ_i refers to the weighted spacing error of the i-th vehicle, and k_a, k_v and k_p are weighting constants. The weighted spacing error δ_i is defined as a function of the spacing error e_i, the raw inter-vehicular distance ΔX_i, the length of the vehicle L, the time headway term hv_i, and the speed of a virtual flatbed tow truck V:

$$\begin{aligned} \delta_i &= e_i - h(v_i - V) \\ &= \Delta X_i - L - h(v_i - V) \\ e_i &= \Delta X_i - L \\ \Delta X_i &= x_{i-1} - x_i \end{aligned}$$

The velocity of the leader vehicle is disseminated regularly to its followers. In the event of communication loss, the follower vehicles switch to a normal ACC CTG policy. Note that the model as described assumes that the lower-level dynamics of all vehicles involved are homogeneous, and does not address the string stability of a platoon of *heterogeneous* vehicles.

Consensus

Santini et al. Santini et al. (2017) outline a distributed algorithm based on the framework of consensus in dynamic networks Chen and Lewis (2011). The communication topology of the platoon members is modelled as a weakly-connected directed graph, and messages passed between platoon members are timestamped in order to handle heterogeneous communication delays.

Santini et al. (2017) define the control law for this controller in terms of a high-order conensus problem – the absolute position of the i-th vehicle r_i and its speed v_i can be defined:

$$r_i(t) \to \frac{1}{\Delta_i}\left\{ \sum_{j=0}^{N} a_{ij} \cdot (r_j(t) + d_{ij}) \right\}$$
$$v_i(t) \to v0$$

In the above, d_{ij} is the desired distance between the i-th and j-th vehicles, a_{ij} is an adjacency matrix modelling the communication topology of the vehicles, Δ_i is the number of communication links available tothe i-th vehicle, or its *degree*, and $v0$ is the speed of the leading vehicle. The authors transform this into a decentralized control action with corresponding delay terms to take communications delay into account.

The authors prove closed-loop stability of the controller using the Lyapunov approach, and also perform realistic simulations using PLEXE (Segata et al., 2014). In a high-density mixed traffic simulation, the consensus-based controller is shown to be able to preserve string stability at a Bernoullian PER of 60%, whereas the original PATH CACC controller becomes string-unstable. Additionally, the effect of different topologies (bidirectional, leader and predecessor, and predecessor only) is also investigated.

This concludes an overview of four major CACC controllers described in recent literature. While these controllers all have the same overall goal, their internals and control laws differ significantly. This, in turn, leads to variations in behaviour between controllers in different situations.

3.2 Overview of Simulation Software

This section provides an overview of the various choices of simulation software available at the time of this work. Part 3.2.1 describes and evaluates vehicle simulators. Part 3.2.2 describes and evalutes network simulators. Finally, Part 3.2.3 describes vehicular network simulators.

Before describing in detail the various software choices available, we define requirements that will inform our choices:

- **Open-Source:** the source code for the software must be freely available and open to modification and extension.
- **Free of charge:** the software must be available at no charge for academic usage.
- **Fit for purpose:** the software must be capable of performing simulations that satisfy all of the requirements detailed previously in Section 2.2 with little to no modification. If it is not capable of fulfilling all of the simulation requirements on its own, it must be interoperable with other software that can do so, and must be able to perform its respective simulation component *in parallel* with the other components of the simulation.
- **Active:** the project must be maintained and under active development.

3.2.1 Vehicle Simulator

The following vehicle simulators were evaluated:

- SUMO (Lopez et al., 2018) is an open-source microscopic vehicle simulator, available free-of-charge and actively maintained under the guidance of the Eclipse Foundation[1]. A great deal of the work referenced in Section 2.3 leverages SUMO for vehicle simulation.
- AIMSUN is a commercial mobility modelling solution, developed by AIMSUN SLU[2]. It is closed-source, and is available for academic purposes upon request.
- VISSIM is a commercial traffic simulator developed by PTV AG.[3] It is closed-source, and is available for academic purposes upon request.
- MOTUS[4] is an open-source microscopic vehicle simulator. It does not appear to be actively maintained, and its capabilities are limited.

[1] `https://www.eclipse.org/`
[2] AIMSUN homepage: `https://www.aimsun.com/`, accessed 2020-09-10
[3] VISSIM homepage: `https://www.ptvgroup.com/en/solutions/products/ptv-vissim/`, accessed 2020-09-10.
[4] MOTUS homepage: `http://homepage.tudelft.nl/05a3n/`, accessed 2020-09-10.

- FLOW (Wu et al., 2017) is a deep learning framework for simulating autonomous vehicles. It is not a standalone vehicle simulator, but integrates with existing vehicle simulators such as SUMO. It is open-source, available free of charge, and is under active development.

3.2.2 Network Simulators

The following network simulators were evaluated:

- OMNeT++[5] is an open-source discrete-event network simulation library and framework. It is under active development, available free of charge, and has been used to create a wide array of network simulations and models.[6]

- NS-3[7] is another open-source discrete-event network simulation tool. It is also under active development and available free of charge. A number of extensions for various networking protocols (e.g. LoraWAN, 5G-LENA) are available.[8]

3.2.3 Vehicular Network Simulators

- VEINS[9] (Sommer et al., 2011) is an open-source simulation framework for running vehicular network simulations, and builds on top of SUMO and OMNeT++. It is available free-of-charge, and is actively maintained. A great deal of the work referenced in Section 2.3.3 leverages VEINS for vehicular network simulation.

- PLEXE[10] (Segata et al., 2014) is an extension to VEINS that enables simulating vehicle platoons. It is open-source, available free of charge, and is actively maintained.

The choice of software to use for this work was partially informed by the integrations available between various simulators, and partially by previous work. The decision was made to use a combination of SUMO, OMNeT++, and VEINS/PLEXE for this work,

[5]OMNeT++ homepage: `https://omnetpp.org/`, accessed 2020-09-10.
[6]OMNeT++ models: `https://omnetpp.org/download/models-and-tools`, accessed 2020-09-10.
[7]NS-3 homepage: `https://www.nsnam.org/`, accessed 2020-09-10.
[8]NS-3 App Store: `https://apps.nsnam.org/`, accessed 2020-09-10.
[9]VEINS homepage: `http://veins.car2x.org/`, accessed 2020-09-10.
[10]PLEXE homepage: `http://plexe.car2x.org/`, accessed 2020-09-10.

as this was the same combination used by the vast majority of related work referenced in Section 2.3. All of the platooning models described in Section 3.1.5 have been implemented as part of PLEXE, and these implementations were recently merged into the SUMO project itself[11].

3.3 Experimental Setup

This section details the exact experimental setup used for this work. Part 3.3.1 details the experimental architecture and the scenario used in this work. Part 3.3.2 details the versions of the software used, as well as the process of calibrating the simulation environment. Part 3.3.3 details the various iterations of the experimental scenario run, as well as the parameters used in these simulations. Part 3.3.4 details modifications made to the simulation software to support this work. Finally, Part 3.3.5 details the data analysis tools used in this work.

3.3.1 Simulation Architecture & Scenario

As detailed in Section 3.2, a combination of the SUMO vehicle simulator (Lopez et al., 2018)), the VEINS vehicular network simulator (Sommer et al., 2011), and the PLEXE platooning simulator (Segata et al., 2014) were used to perform experiments related to this work. VEINS and PLEXE both leverage the open-source discrete event simulator OMNeT++.

The versions of the various software used are as follows:

- SUMO version `1.6.0`
- VEINS master branch commit `a367f827` (August 12, 2020)
- PLEXE master branch commit `6d5f1ede` (Jul 6, 2020)
- OMNeT++ `5.5.1`
- GCC `10.2.1` under Linux (Fedora 32)

[11] SUMO Merge Request 5102: `https://github.com/eclipse/sumo/pull/5102`, accessed 2020-09-11

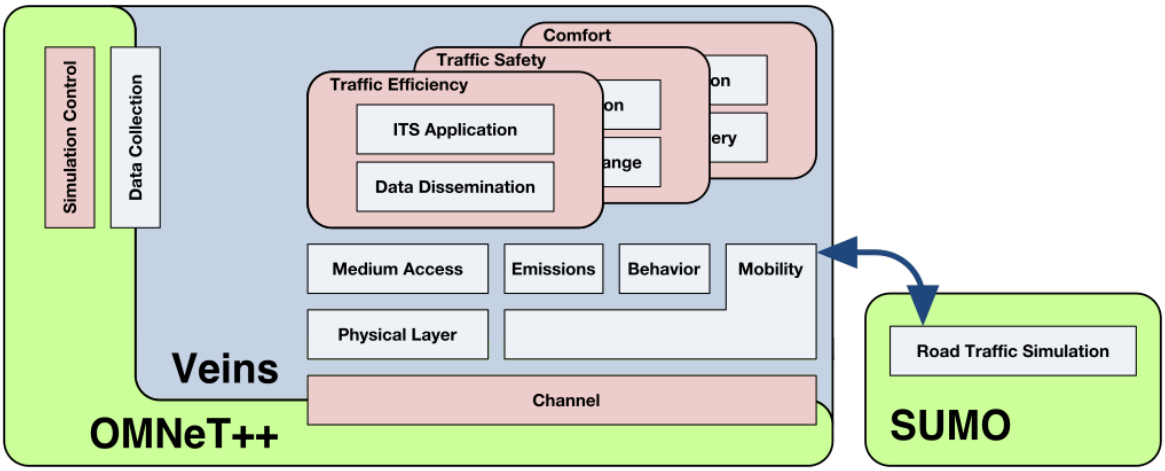

Figure 3.1: VEINS Simulation Architecture, taken from VEINS website: `http://veins.car2x.org/documentation/veins-arch.png`

These particular versions were chosen after various compatibility tests, and based on recommendations from the author of PLEXE.[12]. Additionally, the latest version of VEINS was used due to the recent addition of a feature which allows the user to control the rate at which a packet will be dropped (commit `12170fe5`).

When performing a simulation with VEINS/PLEXE, both SUMO and VEINS/PLEXE (via OMNeT++) are running simultaneously. Upon startup, VEINS starts SUMO with the parameters defined in the scenario configuration, including a directive for SUMO's *Traffic Control Interface* (TraCI) to listen at a specified port on `localhost`. Figure 3.1 shows a diagram of the architecture of VEINS, illustrating the interaction between VEINS and SUMO.[13] The PLEXE simulator is simply an additional library loaded by VEINS at runtime.

VEINS will then perform periodic synchronisation with SUMO over an interval defined as the *timestep*. SUMO will advance the simulation one *timestep*, and then wait for the next directive to advance its simulation. VEINS will then fetch the updated simulation state from SUMO, perform its own internal network simulation logic, and send TraCI commands to SUMO over TCP to perform various actions on the simulated vehicles. This process continues until the simulation time limit defined in the simulation configuration is reached.

[12] `https://github.com/michele-segata/plexe-veins/issues/14`
[13] VEINS website: `http://veins.car2x.org/documentation/`, accessed 2020-09-12.

This flexible architecture allows the developers of both SUMO and VEINS/PLEXE to develop the respective applications independantly; all communication between the two applications happens over TraCI. However, it has a number of unfortunate drawbacks: a large amount of data is transferred between the two processes, with the corresponding overhead of serializing and deserializing data. Additionally, both SUMO and VEINS/PLEXE must move in lock-step with each other, each waiting for the other to finish computing the current timestep. A valuable avenue of future work would be to investigate the possibilty of performing both vehicular simulation and network simulation inside the same process.

It is also important to note that both SUMO and OMNeT++ are *single-threaded* applications, and do not benefit from the large number of cores available in modern hardware. Work to enable scaling SUMO's vehicular simulation across multiple cores is ongoing.[14] To work around this, it is possible to use either `make`[15] or `parallel` (Tange, 2018) to automate the process of scheduling multiple simulations simultaneously across multiple cores. This has implications on recording simulation data, which are detailed in Section 3.3.5.

3.3.2 Scenario & Calibration

The National road network scenario created by Gueriau and Dusparic (2020) was used as a basis for this work. This road network consists of a realistic recreation of a 17.1 km stretch of the N7 (E-20) National Road. A rendering of the road network is provided in Figure 3.2. The traffic flows used by the original authors were generated using open data provided by the Irish Government[16], and consist of a total of 177,417 vehicles travelling through the network over a 24-hour period. All of the road networks, traffic data, and simulation results from the original authors are available freely online[17]. Two simulation time windows were chosen: 0400–0430 (freeflow), and 0700–0730 (congested). The time window studied was reduced from 1 hour to 30 minutes to reduce simulation overhead. Simulation of the entire 24-hour time period was not performed.

[14]SUMO issue: parallelize microsim: `https://github.com/eclipse/sumo/issues/4767`, accessed 2020-09-12.

[15]GNU Make: `https://www.gnu.org/software/make/`, accessed 2020-09-12

[16]NRA Traffic Data: `https://www.nratrafficdata.ie/`, accessed 2020-09-10

[17]Data and results for Gueriau and Dusparic (2020): `https://github.com/maxime-gueriau/ITSC2020_CAV_impact`

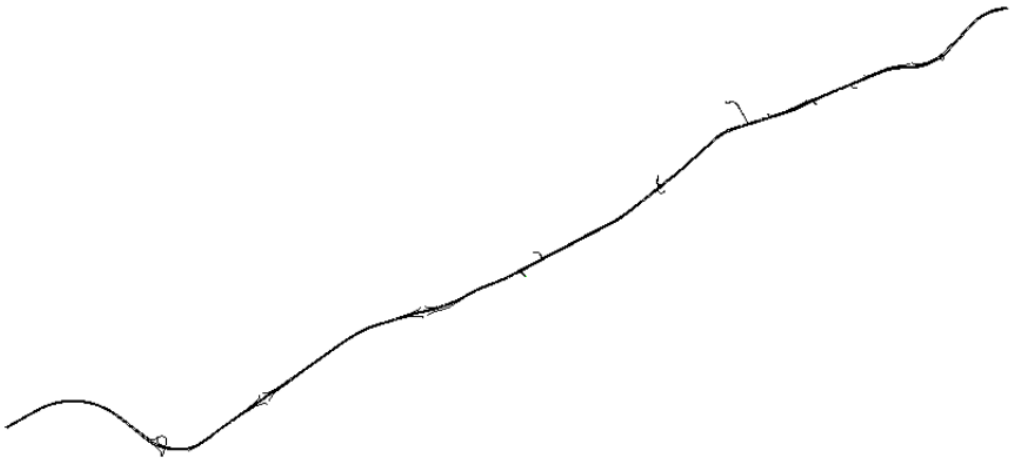

Figure 3.2: Rendering of the National road network scenario

The SUMO configuration files for scenarios A, D, and F provided by Gueriau and Dusparic (2020) were imported into a new PLEXE scenario. To validate the experimental setup, a subset of the original scenario A was run. In this test run, PLEXE/VEINS were configured to take no actions upon vehicles. On an edge-by-edge basis, the number of vehicles entering, exiting, and the average speeds of vehicles traversing the edge were compared with the results of the original authors. A two-tailed Kolmogorov-Smirnov test was used to compare the per-edge distributions, and were not found to be significantly different. The results of this baseline comparison are shown in Figure 3.3.2.

The purpose of performing this initial comparison was to ensure that further simulations performed in this environment are comparable to those of the original authors.

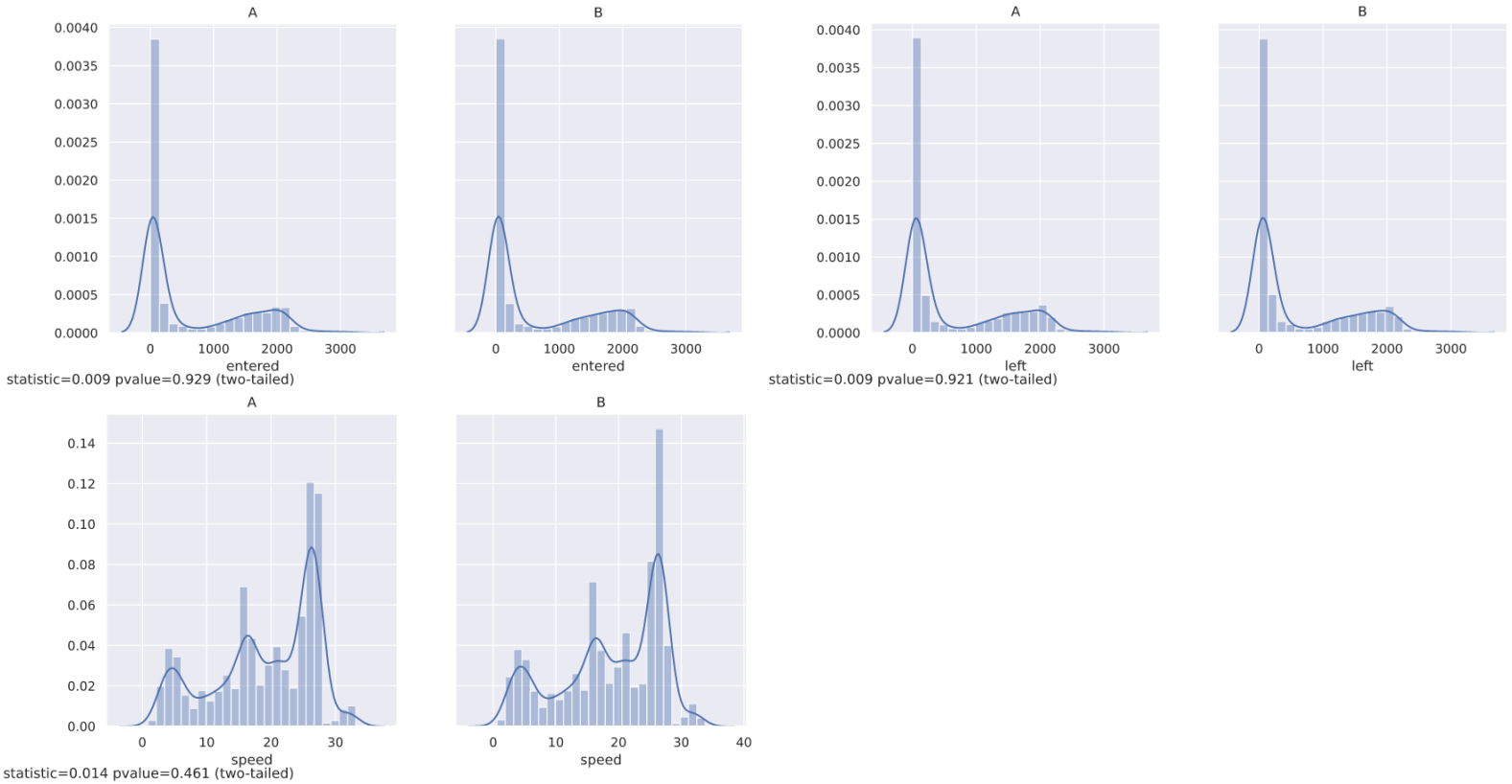

Figure 3.3: Baseline comparison between Gueriau and Dusparic (2020) (A) and reproduction (B). Number of vehicles entering and exiting each edge, and average speed of vehicles traversing each edge are compared. Proportion of the total population is measured on the y-axis. Comparison is made by a two-tailed Kolmogorov-Smirnov test.

3.3.3 Evaluation

Recalling the hypothesis introduced in Section 2.4:

Realistic network simulation has no significant impact on large-scale simulations of the performance of connected and/or autonomous vehicles (CAVs) in physically realistic mixed traffic scenarios.

To test this hypothesis, multiple large-scale phyiscally realistic mixed-traffic simulations of CAVs were performed, varying the probability of successful packet delivery. This hypothesis can be rejected if there is a significant difference between the behaviours of vehicles in the simulation at low packet error rates, and at high packet error rates. As the networked controllers being tested here are *longitudinal* controllers, the relevant metrics to compare will be:

- Vehicle speed/acceleration: how much does the speed of vehicles in the simulation vary over time? This will show the ability of the CACC controller to dampen traffic shockwaves.

- Edge Travel Rate: for each edge in the simulated road network, how long, on average, does it take to traverse each edge? This will allow pinpointing traffic flow bottlenecks in the simulation, (e.g. at lane merges).

Metrics involving the number of lane-changes, or metrics relating to individual lanes of a given edge are out of scope of this work, as these will measure the effects of *lateral control*.

A number of individual experiments were run with varying parameters, using the scenario described in Section 3.3.2 as a basis. The parameters of these experiments are detailed in Table 3.3. These parameters are designed to align with scenarios A, D, and F defined in the work of Gueriau and Dusparic (2020). Only one repetition of each individual experiment was performed of each experiment due to time constraints; future work is to perform multiple repetitions of these parameters.

When performing a simulation, sufficient time must be provided for the road network to load fully before taking measurements. In this work, a simulation parameter `manager.firstStepAt` is leveraged, which specifies the time at which VEINS/PLEXE should preform its initial synchronisation with SUMO. This allows the user to advance the simulation to a specified point in time without the additional overhead of network

simulation, and then proceed to perform the rest of the network simulation logic with a fully loaded traffic network.

Other simulation parameters were unchanged from their original values, or left at their default values. The parameters in table 3.1 relate to the driver models used by SUMO for traffic simulation. Explanations of these parameters are provided in the SUMO documentation[18], and are briefly summarised below:

- `model` defines the car-following model used to model the vehicle's behaviour.
- `tau` defines to the minimum desired headway between vehicles.
- `sigma` controls to the driver's imperfection, where 0 denotes perfect driving.
- `speedDev` defines the factor by which vehicles may exceed per-lane speed limits.
- `minGap` controls to the desired gap (in metres) between the ego vehicle and the leading vehicle.
- `lcModel` specifies the desired lane-changing model, with the SUMO LC2013 model (Erdmann, 2015) being the default.
- `lcStrategic` defines the eagerness with which the ego vehicle will attempt to change lanes.

The parameters in table 3.2 relate to the network simulation parameters used by OMNeT++/VEINS, and parameters used by PLEXE to define the behaviour of the CAV model. Brief explanations of these parameters are provided below, and in Section 3.1.5 where applicable. Additional information is also available in the PLEXE API documentation.[19]

- 802.11p Center Frequency defines the center frequency used by the simulated 802.11p model.
- Packet size defines the size of the simulated packets, in bytes.
- Packet priority defines the priority of the messages sent by CAVs.

[18]SUMO Documentation: `https://sumo.dlr.de/docs/Definition_of_Vehicles,_Vehicle_Types,_and_Routes.html`, accessed 2020-09-12

[19]PLEXE API Documentation: `http://plexe.car2x.org/api/`, accessed 2020-09-12.

SUMO vType	Parameter Name	Parameter Value
HDV	model	SUMO Default (Krauss)
	tau	1.2
	sigma	0.5
	speedDev	0.1
	minGap	2.5
	lcModel	SUMO Default (LC2013)
	lcStrategic	0.5
HDT	model	SUMO Default (Krauss)
	tau	1.5
	sigma	0.5
	speedDev	0.1
	minGap	2.5
	lcModel	SUMO Default (LC2013)
	lcStrategic	0.5
CAV2	model	CC, fallback to Krauss
	tau	0.8
	sigma	0.05
	speedDev	0.05
	minGap	2.5
	lcModel	SUMO Default (LC2013)
	lcStrategic	0.5
CAT2	model	CC, fallback to Krauss
	tau	0.8
	sigma	0.05
	speedDev	0.05
	minGap	2.5
	lcModel	SUMO Default (LC2013)
	lcStrategic	0.5
CAV4	model	IDM
	tau	0.6
	sigma	0.05
	speedDev	0.05
	minGap	1.0
	lcModel	SUMO Default (LC2013)
	lcStrategic	1.0
CAT4	model	IDM
	tau	0.6
	sigma	0.05
	speedDev	0.05
	minGap	1.0
	lcModel	SUMO Default (LC2013)
	lcStrategic	1.0

Table 3.1: Common SUMO parameters for all experiments.

- Tx power defines the transmission power of the radio transmitters used by the simulated CAVs.
- Bitrate defines the bitrate at which the simulated 802.11p protocol operates.
- Minimum Power Level is the minimum power level at which the simulated radio transmitters may operate.
- Noise Floor defines the background noise present in the simulated network. This has a direct effect on the transmission range between vehicles.
- Antenna Type specifies a given antenna definition. This is left at the default value `monopole`, which uses measurements from Kornek et al. (2010).
- Analogue Model defines the path loss model used in the simulated physical layer. The path model used here is a simple model where the signal attentuation depends only on the distance between the sender and the receiver. A possible avenue of future work would be to reassess these simulations using a path loss model which takes interference from other vehicles into account.
- CACC C_1, xi, and ω_n define the respective parameters of the PATH controller.
- Ploeg K_p, K_d, and H define the respective parameters of the Ploeg controller.
- Engine τ defines the engine lag time constant.

More information on these parameters can be found in Sommer et al. (2019).

3.3.4 Modifications

A number of modifications needed to be made to the original scenario, as well as to both PLEXE and VEINS, in order to support this work. This section details the modifications made.

Scenario

The only major modification made to the base scenario was to alter the `vType` declaration of the `CAV2` vehicle type to use the `CC` controler instead of `IDM`. This vehicle

VEINS simulation parameters	
802.11p Center Frequency	5.89 GHz
802.11p Packet Size	200 bytes
802.11p Packet Priority	4
802.11p Tx Power	20mW
802.11p Bitrate	6Mbps
802.11p Minimum Power Level	-110dBm
802.11p Noise Floor	-98 dBm
Antenna Type	`monopole`
Analogue Model	SimplePathLossModel with thresholding
Beaconing Interval	0.1s
PLEXE CACC parameters	
CACC C_1	0.5 (default)
CACC ξ	1 (default)
CACC ω_n	0.2 Hz (default)
CACC Spacing	5m (default)
Engine τ	0.5s (default)
Ploeg H	0.5s (default)
Ploeg K_p	0.2 (default)
Ploeg K_d	0.7 (default)

Table 3.2: Common VEINS and PLEXE parameters for all experiments

type represents a L2 CAV, i.e. a vehicle capable of partial autonomous control. This decision was made because of the implementation of the `CC` vehicle controller in SUMO, which is designed to be able to switch between driver-operated control, ACC control, and CACC control. The human driver model in this case is SUMO's default *Krauss* driver model, and is not configurable. Other SUMO parameters were left unchanged.

PLEXE

Normally, PLEXE simulations involve the `TrafficManager` component of PLEXE inserting preconfigured vehicles into the SUMO simulation via TraCI. Since the vehicles in the base simulation are already defined in the SUMO configuration, this approach is not possible. Additionally, online platoon formation is a non-trivial task, and is an area of active research. Instead, a simple naïve CACC application (`CaccApp`) was created in PLEXE to support this experiment, extending the PLEXE `BaseApp` class. This application causes CACC-enabled vehicles to perform the following actions:

- Upon insertion, set the desired cruise control speed to the speed limit of the

Time Period	CACC Controller	%HDV	%CAV L2	%CAV L4	%PER
Low Traffic (0400–0430)	N/A	100%	0%	0%	0%
High Traffic (0700–0730)	N/A	100%	0%	0%	0%
Low Traffic (0400–0430)	PATH	80%	15%	5%	0%
Low Traffic (0400–0430)	PATH	80%	15%	5%	50%
Low Traffic (0400–0430)	PLOEG	80%	15%	5%	0%
Low Traffic (0400–0430)	PLOEG	80%	15%	5%	50%
Low Traffic (0400–0430)	PATH	30%	50%	20%	0%
Low Traffic (0400–0430)	PATH	30%	50%	20%	50%
Low Traffic (0400–0430)	PLOEG	30%	50%	20%	0%
Low Traffic (0400–0430)	PLOEG	30%	50%	20%	50%
High Traffic (0700–0730)	PATH	80%	15%	5%	0%
High Traffic (0700–0730)	PATH	80%	15%	5%	50%
High Traffic (0700–0730)	PLOEG	80%	15%	5%	0%
High Traffic (0700–0730)	PLOEG	80%	15%	5%	50%
High Traffic (0700–0730)	PATH	30%	50%	20%	0%
High Traffic (0700–0730)	PATH	30%	50%	20%	50%
High Traffic (0700–0730)	PLOEG	30%	50%	20%	0%
High Traffic (0700–0730)	PLOEG	30%	50%	20%	50%

Table 3.3: Summary of experiments run, and the parameters of these experiments.

simulated road network (100 km/h), and engage ACC.

- Continuously send beacons every beaconing interval, including information about the vehicle's position, speed, and angle.
- Continuously listen for beacons emitted by other CACC-enabled vehicles.
- Ignore beacons received from a vehicle that is travelling in a diffferent direction.
- Ignore beacons received from a vehicle that is travelling in a diffferent lane.
- Ignore beacons received from a vehicle that is behind the ego vehicle.
- Ignore beacons received from a vehicle that is more than 50m ahead of the ego vehicle.
- If the beacon was not ignored due to any of the above conditions, and the sender is directly ahead of the ego vehicle (i.e. with no intervening vehicles), engage the CACC controller and synchronise speed with that vehicle.
- Periodically check if the currrently leading vehicle is still within 50m of the ego vehicle. If not, disengage CACC and engage ACC.

This approach has several limitations: it does not specify a dedicated platooning lane, and it does not allow forming platoons of more than 2 vehicles in length. Due to this, evaluation of the Flatbed (Ali et al., 2015) and Consensus (Santini et al., 2017) controllers was not possible. A possible avenue of future work would be to repeat this experiment using a state-of-the-art online platoon formation algorithm.

In order to account for non-CACC-enabled vehicles, some small modifications needed to be made to the `BaseApp` class to allow suppressing both the sending and receipt of beacons for such vehicles. This was necessary as an instance of the PLEXE application is created for each SUMO vehicle retrieved via TraCI at runtime. To the best of the author's knowledge, it is not currently possible to make this instantiation conditional on the SUMO vehicle type, or other runtime parameter. As this results in unnecessary overhead in mixed traffic simulations, a possible avenue of future work would be to extend VEINS/PLEXE to allow the user to specify constraints on the SUMO vehicles that should be synchronised at each simulation timestep.

VEINS

While implementing the above modifications to PLEXE, several TraCI methods were found not implemented in the VEINS `TraCICommandInterface` class, which is the main TraCI API client used for communication between PLEXE/VEINS and SUMO. These were implemented as per the SUMO TraCI specification, which required both judicious use of `gdb` to inspect the binary data sent between VEINS and SUMO, and an element of educated guesswork. The methods implemented were:

- `getSpeed`: returns the speed of the ego vehicle.
- `getAcceleration`: returns the acceleration of the ego vehicle.
- `getAngle`: returns the angle of the ego vehicle.
- `getDistanceTravelled`: returns the total distance travelled by the vehicle so far in the current simulation.
- `getLeader`: returns details of the vehicle directly preceding the ego vehicle.

3.3.5 Data Analysis

This section details the process of extracting and processing the results of the simulations detailed above.

As detailed in Section 3.3.1, a number of experiments were run in parallel, producing over $200GB$ of data in the OMNeT++ vector format (`*.vec`). SUMO can be configured to provide useful output for each edge of the simulated road network[20]. Unfortunately, it was not found possible to override the path of this output while running multiple simulations in parallel. This would have resulted in multiple simulation runs writing to the same file, resulting in data corruption. Additionally, it was found that creating the desired per-edge output of SUMO resulted in significant runtime and memory overhead. Therefore, any required data from SUMO was requested via TraCI and emitted via VEINS/PLEXE to a named output vector.

Initially, the `scavetool` program supplied with OMNeT++ was used to export the data to CSV format. This proved to be untenable, both due to the size of the resulting

[20] `https://sumo.dlr.de/docs/Simulation/Output/Lane-_or_Edge-based_Traffic_Measures.html`

files and due to the time taken to perform the conversion. To work around this, a small Python script `vec2sql.py` was created, which reads a vector file sequentially and exports the data to a SQLite[21] database. This allowed both more effective exploration and transformation of the data, as well as more space-efficient storage on-disk.

In order to reconstruct the per-edge vehicle data that SUMO is capable of emitting, another Python script `sql2edgedata.py` was written to reconstruct an approximation of the edge data. This involved several steps:

- Reading the original SUMO network file and storing the geometry of each edge.
- Constructing an *r-tree* spatial index to allow easily querying edges by coordinate.
- Iterating through each row of the database created in the previous step, matching the position of each vehicle to the respective edge by position, and aggregating per-edge metrics.

However, the coordinate reference systems between SUMO and VEINS differ. SUMO allows negative coordinates in both axes, while OMNeT++ requires that they be positive. This impelementation detail is handled by logic in VEINS[22] to translate between the two coordinate systems. This was ported to Python, using the `shapely` library (Gillies et al., 07) for all geometry-related operations.

The per-edge traffic measures computed in this step include:

- Average, minimum, maximum, and standard deviation of speeds of vehicles traversing a given edge, measured in metres per second.
- Travel Rate, defined as the inverse of the average speed of vehicles traversing a given edge. This is converted to minutes per kilometre for comparison with the original results from Gueriau and Dusparic (2020).
- Congestion Index, defined by Hamad and Kikuchi (2002) as $\frac{t_{actual}-t_{freeflow}}{t_{freeflow}}$, where t_{actual} is the actual transit time for a given edge, and $t_{freeflow}$ is the freeflow transit time for an edge. A value of 0 indicates low congestion, and a value greater than 2 indicates high congestion.

[21] SQLite website: `https://www.sqlite.org/`, accessed 2020-09-12.
[22] Refer to the `TraCICoordinateTransformation` class in the VEINS codebase.

Additional data analysis and graphing was performed using the SciPy suite of Python libraries (Virtanen et al., 2020). The above scripts and the generated per-edge result data are available at the following URL: `https://gitlab.com/johnstcn/dissertation`.

Summary

In this section, we motivated the requirements for testing the hypothesis outlined in Section 2.4, and described the design and implementation of a number of experiments to test this hypothesis. In the next section, we present and analyse the results of running the experiments as described.

Chapter 4

Results

As previous studies (e.g. Makridis et al. (2018)) have shown, effects of CAVs are highly pronounced in dense traffic scenarios, and much less pronounced in low-density traffic due to the limited interactions between vehicles. Therefore, we consider the effect of PER on both traffic rate and congestion index in a free-flow low traffic scenario as a baseline, but ultimately investigate the effect of PER in a high-density traffic environment.

In this chapter, the results of the simulations detailed in Chapter 3.3 are presented. Section 4.1 presents the effects of Packet Error Rate (PER) on the simulations at free-flow. Section 4.2 presents the effects of PER on the simulations in a congested state. Section 4.3 presents a high-level summary and analysis of results from all scenarios.

4.1 Low Traffic Scenario

Results for the Low Traffic scenario are summarised in Table 4.1. We can see very little difference in the numbers of vehicles inserted into the simulation, which indicates little to no congestion inside the road network. As the traffic is free-flow, there is no effect of increased PER. This is an expected result — at low levels of traffic and market penetration rate, the likelihood of two CACC-enabled vehicles being within 50m of each other is low, and therefore the opportunities for forming platoons limited. Figure 4.1 shows boxplots of average vehicle speed per edge, comparing the effects of PER on the controllers tested. We see no effects of packet error rate during this free-flow period.

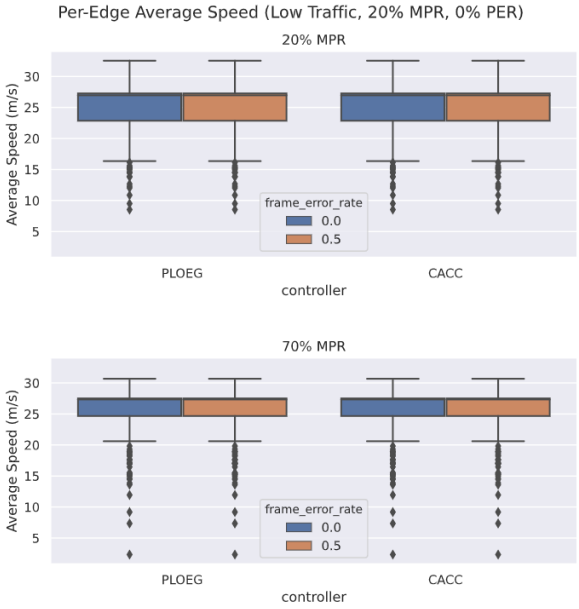

Figure 4.1: Comparison of the effects of PER on the average speed of CACC controllers in low traffic.

Controller	**MPR (%)**	**PER (%)**	**TR (min/km)**	**CI**	**N**
HDV	N/A	N/A	0.710	0.032	1268
CACC	20%	0%	0.704	0.020	1267
		50%	0.704	0.020	1267
	70%	0%	0.710	0.043	1271
		50%	0.710	0.043	1271
PLOEG	20%	0%	0.704	0.020	1267
		50%	0.704	0.020	1267
	70%	0%	0.710	0.043	1271
		50%	0.710	0.043	1271

Table 4.1: Summary of results for Low Traffic scenario (from the time period 0400–0430), means over all edges of Packet Drop Rate (PER), Travel Rate, (TR), Congestion Index (CI), and vehicle count (N).

As described previously in 3.3.4, the custom platooning scenario is configured to engage ACC with the ACC desired speed being equal to the maximum allowable on the network (100 km/h). This is the most likely cause of the larger proportion of vehicles travelling at the specific speed of $27.78m/s$ in this scenario.

4.2 High Traffic Scenario

Results for the High Traffic scenarios (0700–0730) are summarised in Table 4.2. We can see the effects of CAVs on traffic density by comparing the numbers of vehicles SUMO was able to insert into each simulation: at a 20% MPR, it was possible to insert more vehicle into the simulation compared to the HDV baseline.

We can also see the dramatically reduced congestion index (CI) for the scenarios in which CAVs are present, in comparison to those of HDVs. This is partially an artifact of the preconfigured default ACC speed detailed in Section 4.3. Remarkably, at a higher MPR, the number of vehicles inserted into the network dropped. The most likely explanation for this lies in the implementation of the naïve platooning strategy detailed in Section 3.3.4, as the CACC-enabled vehicles initially drive in ACC mode at a preconfigured speed until a beacon is successfully received.

No major differences are visible with regard to the effect of the packet error rate (PER) on travel rate (TR) and congestion index (CI) on both the PATH (CACC) and Ploeg controllers from this experiment. Figure 4.2 shows a comparison of the effects of PER on various controllers in the high traffic scenario. The PATH controller (CACC) is seen to be slightly more sensitive to the effects of PER, but this effect is small at the higher market penetration rate.

The traffic flow can be visualised more easily by plotting each edge of the network with colour mark corresponding to the average travel rate of vehicles along each edge. Figure 4.3 compares the per-edge travel rates of the PATH CACC and Ploeg controllers at 70% MPR with 0% forced PER, and with 50% forced PER. The bottlenecks on particular edges are clearly visible, but no significant changes in travel rate are visible with increased PER.

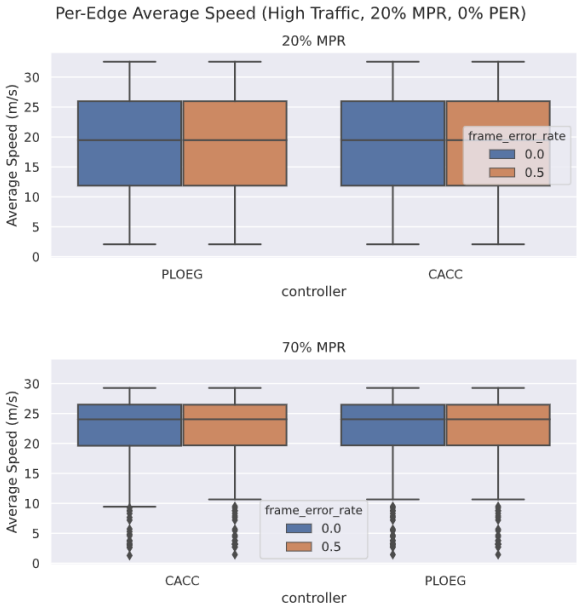

Figure 4.2: Comparison of the effects of PER on the average speed of CACC controllers in high traffic.

Controller	**MPR (%)**	**PER (%)**	**TR (min/km)**	**CI**	**N**
HDV	N/A	N/A	2.787	2.91	9098
CACC	20%	0%	1.181	0.78	9266
		50%	1.181	0.78	9266
	70%	0%	1.002	0.495	8591
		50%	0.997	0.485	8735
PLOEG	20%	0%	1.181	0.78	9266
		50%	1.181	0.78	9266
	70%	0%	0.996	0.485	8735
		50%	0.996	0.485	8735

Table 4.2: Summary of results for High Traffic scenario (from the time period 0700–0730), means over all edges of Packet Drop Rate (PER), Travel Rate, (TR), Congestion Index (CI), and vehicle count (N).

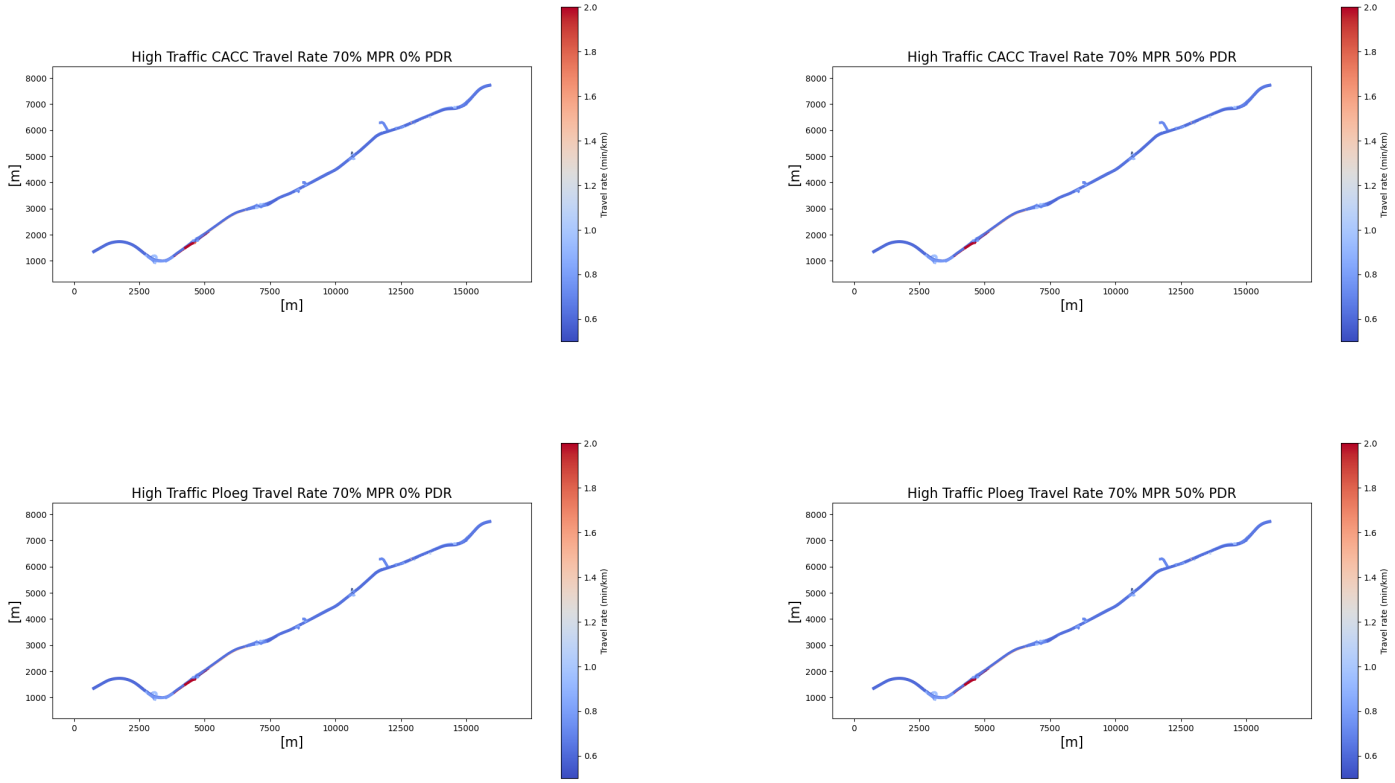

Figure 4.3: Comparison of 0% vs 50% PER on Travel Rates of PATH CACC and Ploeg controller at high traffic, 70% MPR.

4.3 Summary

The results as presented above appear to show that the effect of PER on the performance of longitudinal CACC controllers (CAVs) is nonexistent, and we are thus unable to refute the hypothesis raised in Section 2.4.

However, this should not be interpreted to mean that realistic network simulation has no effect on the performance of CAVs, as other work has shown this to not be the case in smaller-scale simulations. Most dramatically, work by van der Heijden et al. (2017) has shown in simulation that malicious actors are able to effect vehicle collisions by performing actions on the network, while work by Santini et al. (2017) has shown that longitudinal controllers become string-unstable at high PERs.

A number of issues are evident with the experimental scenario as presented:

- Per Santini et al. (2017), noticeable degradation of other controllers only occurred at PERs above 60%. It is entirely possible that the chosen upper bound for forced PER in this experiment was too conservative.
- The mechanism in which CACC-enabled vehicles choose the leading vehicle (3.3.4) is sub-optimal. Additionally, due to the CACC-enabled vehicles initially engaging ACC at a preconfigured speed, the overall speed distributions for all scenarios are effectively artificially distorted. This should be addressed in future work.
- An additional oversight exists in the results: there is no logging of the percentage of time a CACC-enabled vehicle spends in CACC mode, versus ACC mode or HDV mode. This makes it difficult to gauge the effect of packet loss on longitudinal controller performance.
- Multiple merge bottlenecks are present in the road network used, but the effects of lane-changing manoeuvres are not taken into account in this study. This should be addressed in future work.

Additional graphs and data are available separately in the linked Git repository: `https://gitlab.com/johnstcn/dissertation`.

Chapter 5

Conclusions

In this chapter, a brief summary of the work undertaken is provided (Section 5.1), the various challenges encountered while carrying out this work are summarised (Section 5.2), and possible avenues of future work are outlined (Section 5.3).

5.1 Summary

In this work, an array of previous studies of the effects of CAVs in simulated scenarios were evaluated on numerous criteria, including the presence or absence of realistic vehicular simulation, realistic network simulation, and the complexity of the road network (Section 2.3).

The studies evaluated which focus on the larger-scale effects of CAVs on traffic flow tended to simulate complex and/or realistic road networks, but omitted performing realistic network simulation. Conversely, the studies evaluated which focus on the performance and/or implementation of networked CAV models tended to choose less complex road networks, but as a rule will perform realistic network simulation to ensure that the CAV model performs well in a physical environment. However, few of these studies addressed the rate at which CAV models degrade with increased packet loss.

The hypothesis was posed (Section 2.4) that realistic network simulation has no significant effect on the performance of CAVs in large realistic scenarios. A series of large-scale mixed-traffic simulations were designed to test this hypothesis, building on top of previous work by Gueriau and Dusparic (2020).

These experiments (Section 3.3) involved a large realistic road network, realistic vehicle simulation coupled with realistic network simulation using state-of-the-art software, and varied the rate at which packets would be forcibly dropped. Traffic measures including Travel Rate and Congestion Index were measured on each edge of the simulated road network.

The results of the above simulations (Section **??**) were unable to refute the hypothesis: no significant effect on the travel rate of vehicles in the simulation was observed when the probability of packets being arbitrarily dropped was varied. Multiple issues were found in the experimental implementation, and identified as avenues for future work.

5.2 Challenges

A number of challenges were encountered while completing this work:

- The simulation software used is implemented in C++, which allows for great efficiency performing intensive simulations. However, C++ is a complex language in which it is easy to make programming errors.

- Both VEINS and PLEXE have little in the way of unit tests, meaning that contributors implementing new features must perform manual testing. In contrast, SUMO's codebase is well-tested and includes a number of automated integration tests. Manual testing is prone to human error; addition of more automated test suites would allow future contributors to iterate more rapidly while avoiding the addition of errors into the existing codebase.

- The simulation software used has a huge array of parameters to control every possible aspect of the simulation. These parameters are not all documented; many can only be found by inspecting the source code of the respective application. Changes to these parameters may result in unexpected emergent behaviour that may not arise in all scenarios, and recreating the circumstances under which such behaviour emerges is a time-consuming process.

- Performing realistic network simulation on such a scale is extremely resource-intensive. For example, the simulations for the high-traffic scenario at 70% MPR

took over 48 hours of real time to complete. Additionally, an enormous volume of data is produced by simulations of such scale. In the case of the high-traffic scenario, over 39.5 million rows of data were produced. Analysing such a volume of data is a daunting task.

- Due to the single-threaded nature of the simulation software used, multiple simulations needed to be run in parallel. However, as explained in Section 3.3.5, emitting the required per-edge data directly from SUMO was not possible to perform in a parallel fashion. Therefore, the data needed to be emitted in a different format, and reconstructed afterwards. This was a non-trival task which involved transforming co-ordinates between the different reference systems of SUMO and OMNeT++.

5.3 Future Work

Due to time constraints, a number of potential avenues of improvements were identified but not pursued in this work. These are enumerated here.

- The simulation software used in this work is limited to single-threaded execution, and cannot take advantage of the multiple cores present in modern hardware. This work could increase simulation speeds by an order of magnitude.

- SUMO and VEINS' mode of commmunication incurs a large amount of overhead and a significant amount of waiting, as described in Section 3.3.1. Integrating both the vehicle and network simulation into the same execution context could obviate the requirement for lock-step parallel execution between SUMO and VEINS, and thereby greatly increase simulation speeds.

- The online platoon forming algorithm used in this work is extremely naïve and does not represent the state-of-the-art. Repeating this work with the addition of a state-of-the-art platoon formation algorithm would provide a more realistic result.

- The impact of lane-changing manoeuvres was not investigated in this work. These have a significant effect on traffic flow and stability. Authors such as An and

Talebpour (2019) investigate the effects of co-ordinated lane-changing manoeuvres. As packet loss is likely to have a significant effect on such mechanisms, incorporating such research into this work would provide greater insights into the robustness of such algorithms in the face of communication degradation.

- The level of packet loss investigated was capped at 50%. As other studies (e.g. Santini et al. (2017)) have shown, this may have been too low a level to cause significant degradation of longitudinal controller performance.

- The effects of packet loss on simulated surrogate safety measures, such as Post-Encroachment Time (PET) were not investigated in this work. As road traffic accidents claim a large number of human lives every year, this should also be investigated.

- The path loss model used in these experiments only factors in the distance between the sender and the receiver. Additional models may take other obstacles into account, such as buildings or other vehicles, when computing the success or failure of the delivery of a packet. This would further ground this work in reality.

- Two other scenarios were provided by Gueriau and Dusparic (2020): a motorway scenario, and a city centre scenario. The city centre scenario, in particular, would be interesting to investigate. The high number of buildings and other obstacles would likely have a strong impact on the performance of simulated CAVs, assuming the use of the appropriate path loss model.

- Only vehicle-to-vehicle communication was investigated in this work. Investigating the impact of the addition of vehicle-to-infrastructure communications would be worthwhile. In non-line-of-sight scenarios, such as the city centre scenario, it is likely that infrastructure-based communications would be able to partially compensate for path loss between vehicles.

Bibliography

Ali, A., Garcia, G., and Martinet, P. (2015). The Flatbed Platoon Towing Model for Safe and Dense Platooning on Highways. *IEEE Intelligent Transportation Systems Magazine*, 7(1):58–68. Conference Name: IEEE Intelligent Transportation Systems Magazine.

An, G. and Talebpour, A. (2019). Lane-Changing Trajectory Optimization to Minimize Traffic Flow Disturbance in a Connected Automated Driving Environment. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1794–1799.

Bu, F., Tan, H.-S., and Huang, J. (2010). Design and field testing of a cooperative adaptive cruise control system. In *Proceedings of the 2010 American Control Conference*, pages 4616–4621. IEEE.

Central Statistics Office (2016). Census of Population 2016 – Profile 6 Commuting in Ireland.

Chang, K. S., Li, W., Shaikhbahai, A., Assaderaghi, F., and Varaiya, P. (1991). A preliminary implementation for vehicle platoon control system. In *1991 American Control Conference*, pages 3078–3083. IEEE.

Chen, G. and Lewis, F. L. (2011). Leader-following control for multiple inertial agents. *International Journal of Robust and Nonlinear Control*, 21(8):925–942.

Edie, L. C. (1961). Car-Following and Steady-State Theory for Noncongested Traffic. *Operations Research*, 9(1):66–76. Publisher: INFORMS.

Erdmann, J. (2015). SUMO's Lane-Changing Model. In Behrisch, M. and Weber, M., editors, *Modeling Mobility with Open Data*, Lecture Notes in Mobility, pages 105–123, Cham. Springer International Publishing.

Gillies, S. et al. (2007–). Shapely: manipulation and analysis of geometric objects.

Gonzalez, S. and Ramos, V. (2018). A Simulation-Based Analysis of the Loss Process of Broadcast Packets in WAVE Vehicular Networks. ISSN: 1530-8669 Library Catalog: www.hindawi.com Pages: e7430728 Publisher: Hindawi Volume: 2018.

Gueriau, M. and Dusparic, I. (2020). Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic. In *23rd IEEE International Conference on Intelligent Transportation Systems*.

Hamad, K. and Kikuchi, S. (2002). Developing a measure of traffic congestion: Fuzzy inference approach. *Transportation research record*, 1802(1):77–85.

IEEE Standards Association (2010). Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. *IEEE standard*, 802.

Jia, D., Lu, K., and Wang, J. (2014). On the network connectivity of platoon-based vehicular cyber-physical systems. *Transportation Research Part C: Emerging Technologies*, 40:215–230.

Kornek, D., Schack, M., Slottke, E., Klemp, O., Rolfes, I., and Kürner, T. (2010). Effects of antenna characteristics and placements on a vehicle-to-vehicle channel scenario. In *2010 IEEE International Conference on Communications Workshops*, pages 1–5. IEEE.

Lei, C., van Eenennaam, E. M., Wolterink, W. K., Karagiannis, G., Heijenk, G., and Ploeg, J. (2011). Impact of packet loss on CACC string stability performance. In *2011 11th International Conference on ITS Telecommunications*, pages 381–386.

Lin, W.-Y., Li, M.-W., Lan, K.-c., and Hsu, C.-H. (2010). A comparison of 802.11 a and 802.11 p for v-to-i communication: A measurement study. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 559–570. Springer.

Liu, F., Long, Y., and Zhou, H. (2019). Analysis of Autonomous Vehicle Platoon with Disturbance Based on PLEXE. In *2019 Chinese Automation Congress (CAC)*, pages 134–139. ISSN: 2688-0938.

Liu, H., Shladover, S. E., Lu, X.-Y., and Kan, X. (2020). Freeway vehicle fuel efficiency improvement via cooperative adaptive cruise control. *Journal of Intelligent Transportation Systems*, pages 1–13.

Liu, Y., Shi, K., Xu, G., Lin, S., and Li, S. (2018). Analysis of Packet Loss Characteristics in VANETs. In *2018 8th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 219–222. ISSN: 2377-8431.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE.

Makridis, M., Mattas, K., Ciuffo, B., Raposo, M. A., Toledo, T., and Thiel, C. (2018). Connected and Automated Vehicles on a freeway scenario. Effect on traffic congestion and network capacity. page 10.

Mannoni, V., Berg, V., Sesia, S., and Perraud, E. (2019). A comparison of the v2x communication systems: Its-g5 and c-v2x. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5. IEEE.

Mena-Oreja, J. and Gozalvez, J. (2018). On the Impact of Platooning Maneuvers on Traffic. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–6.

Otto, J. S., Bustamante, F. E., and Berry, R. A. (2009). Down the Block and Around the Corner: The Impact of Radio Propagation on Inter-vehicle Wireless Communication. In *2009 29th IEEE International Conference on Distributed Computing Systems*, pages 605–614. IEEE.

Ploeg, J., Scheepers, B. T. M., van Nunen, E., van de Wouw, N., and Nijmeijer, H. (2011). Design and experimental evaluation of cooperative adaptive cruise control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 260–265. ISSN: 2153-0017.

Ploeg, J., Semsar-Kazerooni, E., Lijster, G., van de Wouw, N., and Nijmeijer, H. (2013). Graceful degradation of CACC performance subject to unreliable wireless

communication. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1210–1216. ISSN: 2153-0017.

Rajamani, R. (2011a). *Vehicle dynamics and control*, chapter 7. Longitudinal Control for Vehicle Platoons, pages 187–220. Springer Science & Business Media.

Rajamani, R. (2011b). *Vehicle dynamics and control*, chapter 6. Adaptive Cruise Control, pages 153–183. Springer Science & Business Media.

Rajamani, R., Tan, H.-S., Law, B. K., and Zhang, W.-B. (2000). Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons. *IEEE Transactions on Control Systems Technology*, 8(4):695–708.

Road Safety Authority (2018). *Rules of the Road*, chapter 8. Speed Limits, pages 110–111. O'Brien Press Ltd.

Santini, S., Salvi, A., Valente, A. S., Pescapé, A., Segata, M., and Lo Cigno, R. (2017). A Consensus-Based Approach for Platooning with Intervehicular Communications and Its Validation in Realistic Scenarios. *IEEE Transactions on Vehicular Technology*, 66(3):1985–1999. Conference Name: IEEE Transactions on Vehicular Technology.

Schakel, W. J., van Arem, B., and Netten, B. D. (2010). Effects of Cooperative Adaptive Cruise Control on traffic flow stability. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 759–764. ISSN: 2153-0009.

Segata, M., Joerer, S., Bloessl, B., Sommer, C., Dressler, F., and Cigno, R. L. (2014). Plexe: A platooning extension for Veins. In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 53–60, Paderborn, Germany. IEEE.

Shladover, S. E. (2007). Path at 20—history and major milestones. *IEEE Transactions on intelligent transportation systems*, 8(4):584–592.

Shladover, S. E., Su, D., and Lu, X.-Y. (2012). Impacts of Cooperative Adaptive Cruise Control on Freeway Traffic Flow. *Transportation Research Record*, 2324(1):63–70.

Sommer, C., Eckhoff, D., Brummer, A., Buse, D. S., Hagenauer, F., Joerer, S., and Segata, M. (2019). Veins – the open source vehicular network simulation framework.

In Virdis, A. and Kirsche, M., editors, *Recent Advances in Network Simulation*. Springer.

Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15. Conference Name: IEEE Transactions on Mobile Computing.

Stern, R. E., Cui, S., Delle Monache, M. L., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Haulcy, R., Pohlmann, H., Wu, F., Piccoli, B., Seibold, B., Sprinkle, J., and Work, D. B. (2018). Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205–221.

Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S.-i., and Yukawa, S. (2008). Traffic jams without bottlenecks - experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10:33001.

Tange, O. (2018). *GNU Parallel 2018*. Ole Tange.

Terruzzi, L., Colombo, R., and Segata, M. (2017). Poster: On the effects of cooperative platooning on traffic shock waves. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 37–38. ISSN: 2157-9865.

Thrun, S. (2010). Toward robotic cars. *Communications of the ACM*, 53(4):99.

Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805.

van der Heijden, R., Lukaseder, T., and Kargl, F. (2017). Analyzing attacks on cooperative adaptive cruise control (CACC). In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 45–52. ISSN: 2157-9865.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern,

R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E. W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Wang, J., Liu, J., and Kato, N. (2019). Networking and Communications in Autonomous Driving: A Survey. *IEEE Communications Surveys Tutorials*, 21(2):1243–1274. Conference Name: IEEE Communications Surveys Tutorials.

Wang, Z., Bian, Y., Shladover, S. E., Wu, G., Li, S. E., and Barth, M. J. (2020). A Survey on Cooperative Longitudinal Motion Control of Multiple Connected and Automated Vehicles. *IEEE Intelligent Transportation Systems Magazine*, 12(1):4–24. Conference Name: IEEE Intelligent Transportation Systems Magazine.

Wu, C., Bayen, A. M., and Mehta, A. (2018). Stabilizing Traffic with Autonomous Vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, Brisbane, QLD. IEEE.

Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., and Bayen, A. M. (2017). Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv preprint arXiv:1710.05465*, page 10.

Yao, S., Shet, R. A., and Friedrich, B. (2020). Managing connected automated vehicles in mixed traffic considering communication reliability: a platooning strategy. *Transportation Research Procedia*, 47:43–50.