Exploring Smart Infrastructure Concepts to Improve the Reliability and Functionality of Safety Oriented Connected Vehicle Applications

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Electrical Engineering

by
Anjan Narsingh Rayamajhi
August 2019

Accepted by:
Dr. James Martin, Committee Chair
Dr. Harlan Russell, Co-Committee Chair
Dr. Daniel Noneaker
Dr. Kuang Ching Wang
Dr. James Westall

Abstract

Cooperative adaptive cruise control (CACC), a form of vehicle platooning, is a well known connected vehicle application. It extends adaptive cruise control (ACC) by incorporating vehicle-to-vehicle communications. A vehicle periodically broadcasts a small message that includes in the least a unique vehicle identifier, its current geo-location, speed, and acceleration. A vehicle might pay attention to the message stream of only the car ahead. While CACC is under intense study by the academic community, the vast majority of the relevant published literature has been limited to theoretical studies that make many simplifying assumptions. The research presented in this dissertation has been motivated by our observation that there is limited understanding of how platoons actually work under a range of realistic operating conditions. Our research includes a performance study of V2V communications based on actual V2V radios supplemented by simulation. These results are in turn applied to the analysis of CACC. In order to understand a platoon at scale, we resort to simulations and analysis using the ns3 simulator. Assessment criteria includes network reliability measures as well as application oriented measures. Network assessment involves latency and first and second order loss dynamics. CACC performance is based on stability, frequency of crashes, and the rate of traffic flow. The primary goal of CACC is to maximize traffic flow subject to a maximum allowed speed. This requires maintaining smaller inter-vehicle distances which can be problematic as a platoon can become unstable as the target headway between cars is reduced. The main contribution of this dissertation is the development and evaluation of two heuristic approaches for dynamically adapting headway both of which attempt to minimize the headway while ensure stability. We present the design and analysis of a centralized and a distributed implementation of the algorithm. Our results suggest that dynamically adapting the headway time can improve the overall platoon traffic flow without the platoon becoming unstable.

Dedication

To my parents Shashi and Bindu Rayamajhi, brother Amir, and beloved wife Madhabi. I could not have done this without your unconditional love and support.

Acknowledgments

My deepest gratitude goes to my advisor Dr. Jim Martin for his immense guidance and motivation in every aspect of my graduate studies. I want to say a special thank you to Dr. James Westall for his guidance in my research. I would also like to thank Dr. Harlan Russell, Dr. Daniel Noneaker, and Dr. Kuang-Ching Wang for being in my committee. I also want to thank Dr. Mashrur Chowdhury for his guidance and giving me opportunities to collaborate in many research projects. I also want to thank Dr. Pierluigi Pisu and his student Roberto Merco for their help. Finally, I want to say thank you to my peers Manveen Kaur, Jianwei Liu, Dr. Kakan Dey, Dr. Mizanur Rahman, Mahfuz Islam, and Sakib Khan for their support and help which has made my graduate studies even more memorable.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The loss of life associated with transportation, both in the US and worldwide, is staggering. It is estimated that 40,000 deaths occur per year in the US due to injuries caused by vehicular accidents. The number of vehicular fatalities has remained at its peak of 40,000 deaths in 2018. This death rate is a 14% increase in deaths since 2014. The World Health Organization WHO estimated 1.25 millions deaths worldwide in road accidents [1]. Vehicular fatalities is the eighth leading cause of death in the world [2].

The United States Department of Transportation (US-DOT) along with the National Highway Traffic Safety Association (NHTSA) outlined two complementary transportation technology directions that could reduce the number of traffic fatalities and increase the efficiency of transportation systems. The two technology areas are Connected Vehicles (CV) and Autonomous Vehicles (AV). The two initiatives are collectively called Connected and Autonomous Vehicles (CAV). Connected vehicles is a concept that allows vehicles to communicate with other vehicles or with roadside infrastructure. The US-DOT has developed a complex distributed computing environment that manages different types of applications operating, at least in part, on participating vehicles. One category of vehicular application is safety applications which are intended to prevent crashes and save human lives. Another category of application is eco-friendly applications which are intended to improve the efficiency of transportation systems by either reducing the global fuel consumption or by increasing the rate of traffic flow across roadways. The eventual goal of CAV is to eliminate the primary cause of vehicular fatalities which is human error. The US-DOT has developed a staged approach towards replacing human decision making with autonomous machine decision making. These two technology directions have evolved independently for roughly the past decade. Only recently do we see significant activity towards a converged perspective of connected and autonomous vehicles (CAVs). This convergence brings together large, well established academic communities including Intelligent Traffic Systems (ITS), vehicular networking, automotive engineering, robotics, Lidar and computer-aided perception, along with new and quickly evolving fields such as machine learning and fog compute frameworks.

The research presented in this dissertation can be precisely located and defined within the broad scope of Cooperative Adaptive Cruise Control (CACC). Our research focuses on CACC which is a specific form of vehicular platooning. Along with other vehicular applications such as forward collision avoidance and lane change assistance, CACC is anticipated to be among the first set of CAV applications that will be widely deployed. It is particularly timely to study CACC as the research community is at the early stage of redefining CACC in an appropriate context fundamental to CAVs. Aspects of our research contributes to this effort.

The DOT has multiple large scale CV deployments underway in locations that include a mix of urban and rural areas including New York City, Tampa, Ann Arbor, and Wyoming. With few exceptions, every major car company is investing heavily in the R&D that will pave the way for autonomous vehicles. The DOT and NHTSA have recently published guidelines for AV [3] which identify a set of AV capabilities and define standards and guidance to ensure that equipment providers and technology vendors meet safety, mobility, and vehicular cyber-security requirements. The guideline categorizes the autonomy of vehicles into six distinct levels - level 0 being fully manual control of vehicles to level 5 which is fully autonomous control of vehicles. The research presented here focuses on CV but assumes the specific AV capabilities are experimental parameters. With reference to the level of autonomous capabilities mentioned in [3, 4], our research work considers the effect of vehicles with full automation capability as well as human controlled vehicles. With this approach we consider the safety and mobility requirements to include all forms of transportation models. It is natural to focus our research in the context of mixture of vehicles with different level of autonomy because that puts our work very relevant to next generation of transportation. In Chapter 4 we will explain in detail our CACC modeling approach including all assumptions. One crucial assumption is our simple modeling approach of providing a number of model control knobs that allow us to tailor the behavior of vehicles and the CACC algorithms with respect to their ability to model solely machine autonomy or to model human factors.

Figure 1.1 illustrates the CAV concept. The major components include -

- vehicles
- On-board unit (OBU)
- Road side unit (RSU)
- Transportation Authority
- Streams of sensors (radar, lidar, video).

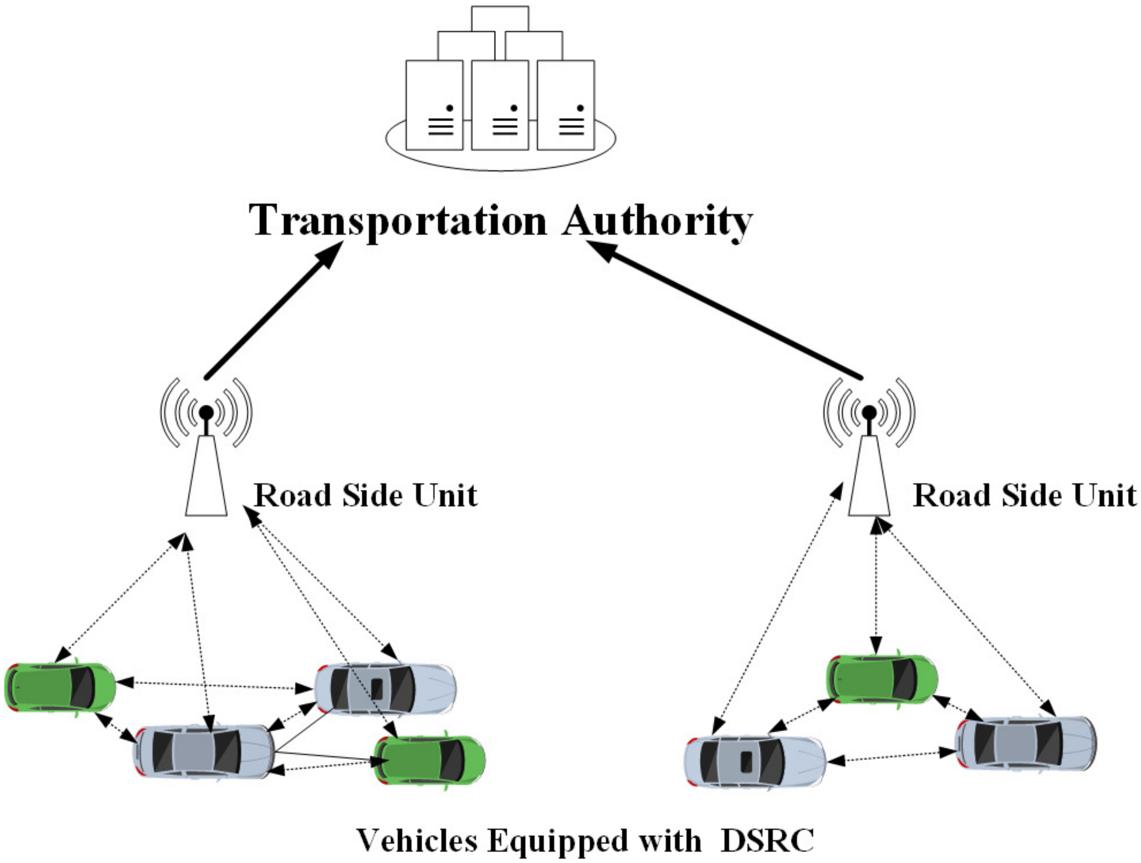


Figure 1.1: Current Connected Vehicle system

The system involves any number of vehicles that are able to communicate using wireless vehicle-to-vehicle (V2V), vehicle to roadside infrastructure (V2I) or vehicle-to-network (V2N) technologies collectively called vehicle-to-everything (V2X). The roadside infrastructure is connected to the Transportation Authority which can be a state DoT which, at least in the early deployment stage, will likely have RSU's interact with Vehicles passing through to collect vehicular information or to instruct vehicles in some manner. A vehicle might periodically broadcast a basic safety message (BSM) which contains a concise state of the vehicle such as speed, geographical location, steering wheel, and brake status. The nominal BSM transmission rate is ten messages

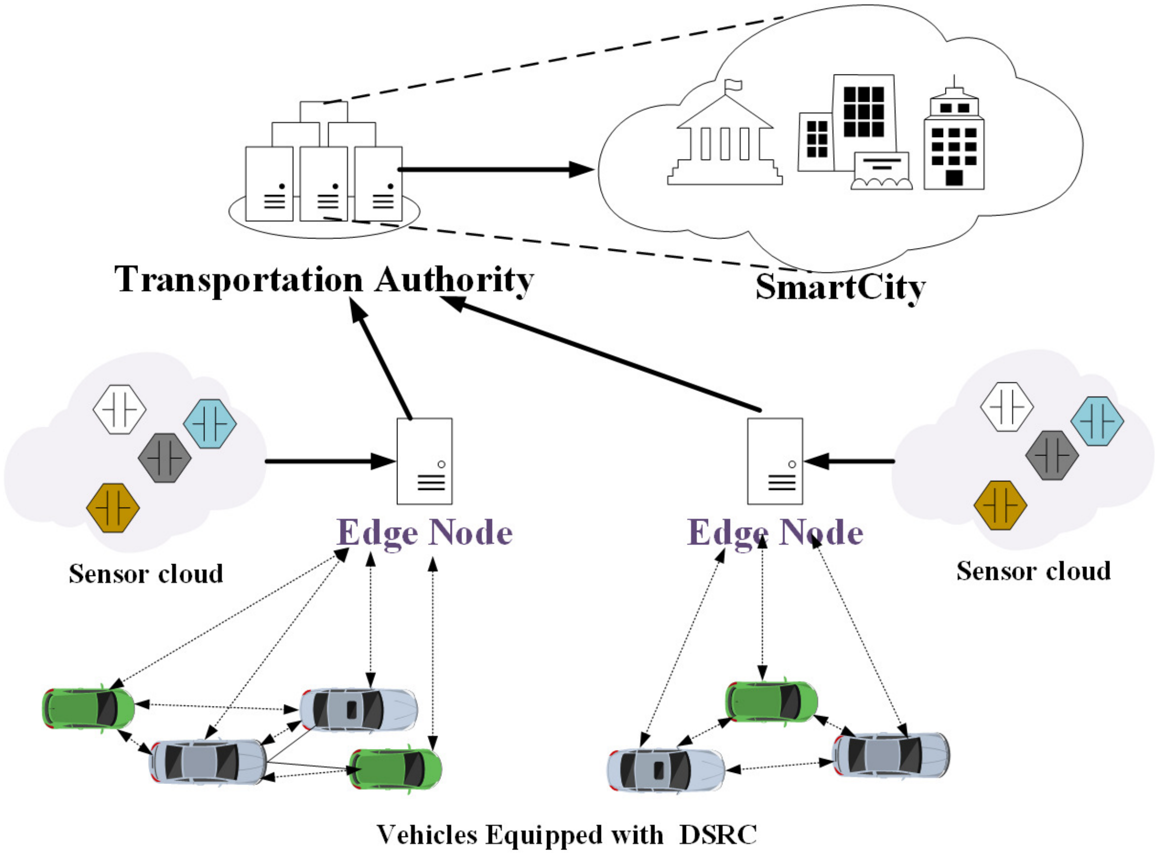


Figure 1.2: Future Intelligent Transportation System concept

per second. Current state DOT's run macroscopic traffic management processes on a city scale based on course grained information from in-road vehicle flow rate counters or more recently based on video analytics. If the CV concept progresses to large deployments, state DOTs will be able to drastically improve the efficiency of a given roadway system through access to many orders of magnitude of vehicular sensing data that can reach backend ITS systems within seconds. The larger societal impacts of CAVs will be the reduced rate of loss of life. This, in fact, is the primary motivation of our research - to contribute to the evolution of CACC from the prior conceptual world to practical CACC systems. Figure 1.2 shows an idea that involve the potential integration of CV with other domain of sensing applications that might address issues related to the climate, the environment, or to public safety. The idea can be regarded as an infusion of sensory data from a large number of intra-vehicular or inter-vehicular sensing devices. A smart infrastructure involving vehicular sensors can serve the needs of such domain. While terms like smarter city and edge computing are widely used (but not widely understood), we focus the proposed research on applying the concept of 'smart infrastructure' in a CV context and assume a range of AV capabilities involving sensors applicable in a vehicular environment.

The first step by the DOT is to require future vehicles to support a form of wireless communications referred to as dedicated short range communication (DSRC) [5]. A variant of the IEEE 802.11a wireless standard, DSRC is also referred to as 802.11p. This standard operates over seven channels each with bandwidth of 10 Mhz in a dedicated spectrum located at 5.9 Ghz. In addition DSRC supports roughly the same physical layer as 802.11a protocol with some modifications[6]. The link layer is enhanced to efficiently support an adhoc broadcast network with application traffic managed by specific application mappings and subsequent management through the IEEE 802.11e protocol for prioritizing different types of queues. Safety applications are given highest priority but the system supports non-safety applications (data collection, roadway condition warning, etc) to use other channels with a lower priority level. Technically the priority is defined based on the amount of time lagged at the queue on an average; shorter queue delay ensure faster service hence higher priority [5]. To further optimize V2V communications, an alternative to TCP/IP was developed referred to as wireless access for vehicular environment (WAVE) [5, 7, 8]. The scope of WAVE includes application discovery, security, and various management functions. We focus our research just on the necessary components of the fully designed system to allow us to develop and evaluate our ideas related to CAV applications for smart infrastructures. We focus on CACC applications using IP over DSRC but the same can be equally defined for WAVE over DSRC. The motivation for the choice of network protocol came from our experience using hardware implementation of DSRC technologies from many vendors and coming to face the same issues with independent abstraction of the underlying technology with little commonality. Also this approach is quite common as it removes the complications one needs to deal with when using proprietary WAVE implementations from different vendors.

For many years, DSRC has been designated as the wireless communication technology for vehicle to vehicle and vehicle to infrastructure communication. Implementations have been tested and proven to be capable of successful operation under high mobility and high traffic congestion conditions. Previous studies have investigated the performance and reliability of DSRC under different networking conditions [9–12] and provided substantive performance metrics. What lacks is the understanding of the effect wireless networks and its performance has on large scale vehicular networks and to develop counteractive solutions. The DOT has categorized and identified various connected vehicle applications using a blueprint architecture referred to as connected vehicle reference implementation architecture (CVRIA) [13]. The CVRIA provides details on many CV applications precisely defining their design and implementation. However, the descriptions are abstract and require significant interpretation in order for a realistic design and implementation to lead to a robust and safely deployed application. It is very apparent that at least some of the CVRIA application definitions would not lead to practical implementations. We again view this as a symptom of the many disciplines that must come together to produce a complex system that involves vehicles that communicate to support applications that have some manual as well as AV controls.

Prior research in this domain are limited with simplifying and sometimes unrealistic assumptions. Most of the previous work consider the loss in wireless component in vehicular networks as independent and uncorrelated process and assuming the effect of channel interference and noise to not span more than one packet transmission period, see [14–17]. There is a gap in literature towards understanding the performance of a platoon application under adverse channel condition that render significant loss in communication resulting in unavailability of cooperative messages. Many studies put forth the assumption of bounded latency and uncorrelated packet loss burst. Such assumption may not be accurate for channel access protocols like carrier sense multiple access (CSMA) in cases where the channel is kept always busy, with jamming attack on channel, leading to longer delays and packet drops. We assume that the channel loss process is not Bernoulli process (uncorrelated loss process) and provide evidence with real world tests in vehicular networks to support that claim. Recently platooning applications are being considered in a model predictive controller (MPC) form [18, 19] which subject platooning requirements through prediction and optimization. A MPC is designed to predict the state of a vehicle over a prediction horizon (future time) and use the predicted results and measured values to optimize an objective. Recently the optimization problem for MPC have been focused on lane changing and path finding problem for autonomous vehicles [20], minimizing energy consumption for platoon application [18] etc. The study in [18] looks at how channel degradation (e.g. packet loss) effect a controller like MPC but the result is limited to independent and uncorrelated loss processes. However, looking into MPC and its behaviour at burst packet loss process is beyond the scope of this study. Our study aims to provide a solution for a platooning application which, hopefully, can be interpolated to any form of such application. We focus on CACC which can be considered as a subset of MPC with optimized solution to minimize the distance separation error between two vehicles. We adhere to the original idea that CACC can be used for increasing traffic throughput and reduce inter-vehicle air friction energy loss by reducing the spacing between vehicles in a safe manner [14, 21]. It is evident that self driving cars and autonomous platooning trucks are already implemented in parts of the US and Europe. Our study focuses on autonomous trucks which are primarily used in freight transportation. We assume a single lane of freeway traffic so there are no intersections or traffic lights in our scenarios. We consider the vehicles to have homogeneous kinematics properties making them, ideally, copies of each other so that the driving profiles are similar. Our study considers the local sensors available in vehicles are not error free and we model our system based on erroneous sensor related data.

The performance and reliability of applications like CACC in actual DSRC-based CV deployments are not well understood. The emerging generation of these applications will support DSRC as well as future wireless networks. It is vital that systems operating these applications are better understood before they are deployed. The proposed research extends the emerging system concept to one that incorporates our ideas related to smart infrastructure where communication among vehicles and infrastructures can compliment the application. The main contribution of this research can be outlined as follows:

- We show that platoons are affected by outages in wireless networks and fall back mechanisms have to be developed to prevent a platoon from becoming unstable and to avoid crashes.

- We present the optimal approach to navigate a platoon with different number of vehicles under under different network scenarios. Our approach is to find the most safest and stable criteria.

- We develop and evaluate practical methods by which a platoon can adapt to better achieve maximum traffic flow while ensuring stability.

The rest of the dissertation is divided as follows: Chapter 2 provides literature review, Chapter 3 describes the wireless technology in vehicular systems, Chapter 4 describes the CACC application, Chapter 5 describes the simulation of large scale CACC, Chapter 6 describes the heuristic algorithm developed, and Chapter 7 concludes the dissertation.

Chapter 2

Literature Review

In this chapter we present a review of recent studies pertaining to CACC, WAVE and DSRC protocols. The DSRC protocol is based on 802.11a/e with quality of service enhancements. The underlying modulation schemes used is orthogonal frequency division modulation (OFDM) with BPSK, QPSK, QAM-16 and QAM-64 schemes [5–8, 22]. Authors in [5, 8, 22] describe in detail the functionality of 802.11p physical layer operation. The enhancements to make the previous 802.11a protocol robust towards high mobility vehicular communication include increasing the OFDM symbol duration from 4 μs to 8 μs (increasing the cyclic prefix from 0.8 μs to 1.6 μs) and introduction of wireless access for vehicular environments (WAVE) protocol that avoids the need of transport and network layers in the traditional OSI model. These standards are detailed in [6, 23]. There have been studies on the analytic modeling of DSRC protocol to understand the behavior in multiple node scenarios by quantifying the probabilistic nature of transmission and latency. Results in [24, 25] show the analysis based on 2-D Markov model that can be used to model throughput, congestion, and latency in an analytic manner. The studies show how congestion and large number of nodes affect the throughput, and latency in a 802.11p network. The studies also looks into probability of collision, probability of transmission, throughput and latency as number of nodes increase to measure the performance of 802.11p in dense network conditions. Works in [9–11] look at the performance of 802.11p in a realistic network scenarios with real DSRC compliant radios deployed in vehicles. The study in [9] concludes that line of sight and no line of sight significantly affect the performance of a DSRC network. In [10] a driving track for tests was created and performance of a DSRC radio in terms of throughput, latency, and packet loss was compared with that of 802.11a Wi-Fi protocol.

In recent years, most of the research in the network layer of vehicular networks have focused on IPv6 and WAVE [7, 8]. WAVE implementation of the network layer contain functions related to short message service (WSMP), multichannel MAC layer, security and network management as shown in figure 2.1. The 802.11p total bandwidth of 75 MHz is divided into seven channels of 10 MHz each (6 service channels SCH and 1 control channel CCH) and a mode in WAVE enables switching between two channels periodically over a period of 100 ms (50 ms for each channel) called switching mode or a continuous, unobstructed usage of one of the 7 channels called continuous mode. Works shown in [26, 27] show multichannel operation of DSRC based on IEEE 1609.4 protocol. The synchronization is based on GPS enabled 1 ppm signal that allows all nodes in the vicinity to convene at CCH every other 50 ms and switch to appropriate SCH in the next 50 ms period. While in CCH, only the road side units broadcast, and while in SCH, all the nodes contend based on CSMA for channel access. Several studies mentioned in [26, 28] look at TDMA based channel access in which the motivation towards selecting a CSMA based channel access protocol is mostly based on highly mobile vehicular environment with extremely fluctuating channel conditions and network topography. The system also supports IPv6 over SCH and can be configured to operate as an option to WAVE. Some works

have looked at employing routing and mobility management leveraged by IPv6 over V2V and V2I communication by using address reconfiguration [29]. In recent years, cellular V2X also known as C-V2X is being considered as an option to DSRC with huge support from the cellular companies [30]. Technically, C-V2X works with latest LTE (rev 14 or higher) using the proximity service in two modes, such as: [31, 32]
1. if the V2V synchronization and resource allocation is managed by the infrastructure 2. if the vehicles themselves manage the resources in cases where infrastructures become unavailable.

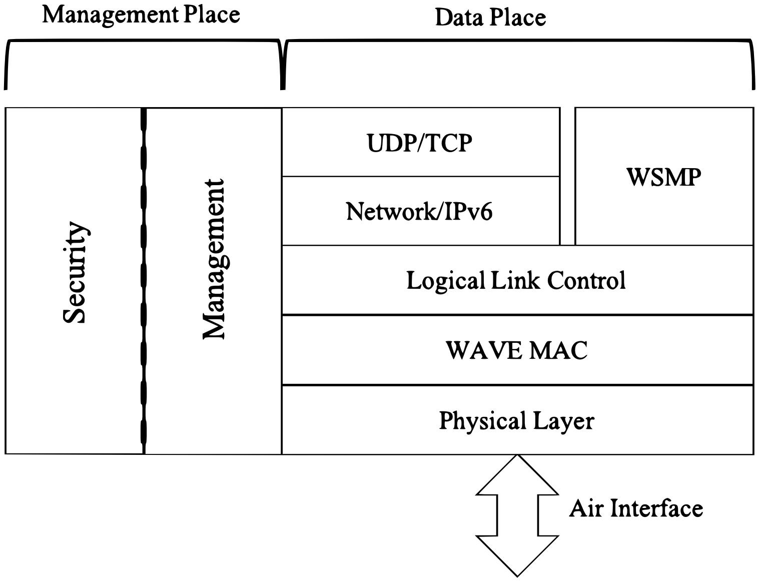


Figure 2.1: 802.11p WAVE stack

Our study has selected a safety/mobility applications - cooperative adaptive cruise control (CACC) as an illustrative CAV application. We believe that this application along with other ones like forward collision avoidance, lane assistance and queue warning will be prevalent in coming CAV Systems. Forward collision avoidance, also known as rear end collision avoidance, is a highly safety critical application which alerts vehicles, in real time, about possible collision or roadway breakage in the same lane and direction of travel [13, 33]. Present collision avoidance applications use radar and video cameras aligned at the front of the vehicle to detect physical objects in the direction of travel. These devices render a small view angle where DSRC can provide a complete 360 iew around a vehicle without any obstructions. Current autonomous vehicles use LiDAR or video cameras for visually rendering the surrounding environment such as Velodyne's LiDAR used in Google's self driving cars [34]. There have been studies in the past that look at forward collision avoidance using DSRC technology to alert the drivers of imminent collision [35] that show how effective DSRC can be in that scenario. One of the types of vehicle platooning application is CACC where the velocity and acceleration of the following vehicle in a vehicle pair are controlled to maintain proper headway distance from the leading vehicles. The information about preceding vehicles velocity and acceleration is transmitted using wireless broadcasts to the following vehicle. In contradiction to the earlier version of platooning called adaptive cruise control (ACC), CACC requires broadcasting of vehicle status messages to the following vehicle through a reliable wireless network. There has been tremendous amount of research in CACC [14, 16, 36–39] mostly focused in theoretical assumptions of a platoon as well as the characteristics of the wireless communication network.

Figure 2.2 shows CACC defined from a Cyber-Physical System (CPS) perspective that regulates the acceleration of an ego vehicle {vehicle in consideration} in a platoon based on the acceleration of preceding vehicle(s) received over a wireless vehicular adhoc network (VANET) [40] and local sensor readings of distance separating back bumper of leading vehicle to front bumper of the ego vehicle. The system can be easily viewed as an instance of a CPS because of how information collected from regulated vehicle (physical space) is used to calculate parameters (cyber space) that are recursively utilized to control the same physical system. This opens up aspects of our research where we develop controller module that act as a component to the cy-

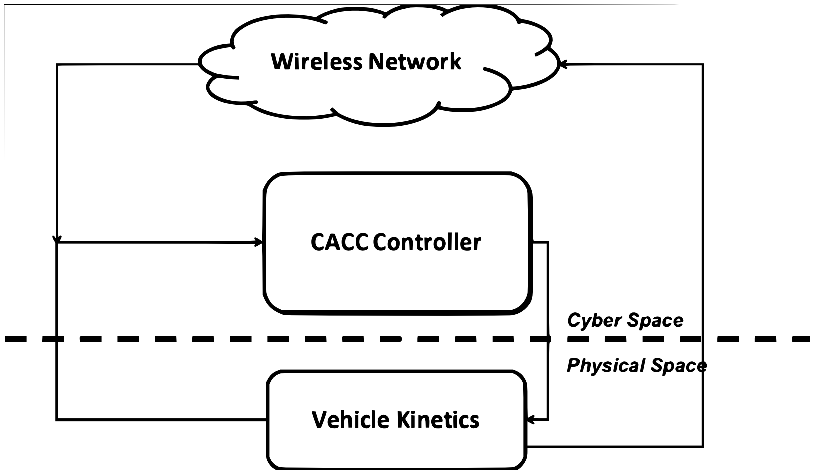


Figure 2.2: CACC as a Cyber-Physical System

ber space and helps to better regulate the vehicle. The wireless protocol in vehicular adhoc network is based on 802.11p or dedicated short range communication (DSRC) protocol and the local sensors considered are either LiDAR or radar sensors. The set of vehicles collaborating in a CACC are called a CACC platoon. A CACC provides better speed harmonization over the length of the platoon and increases safety by creating a tighter coherence between each vehicle and their movement pattern by constantly broadcasting the mobility information. The air drag is considerably reduced towards the tail of the platoon allowing smaller torque to be actuated from the vehicular mechanics which eventually reduces the aggregate energy consumption of the CACC platoon [36, 40]. In recent years, CACC and platooning applications are being considered for arterial network of roads and implemented mostly for reducing energy consumption (eco-driving) as well as safe stopping distance estimation at intersections. However our study concentrates on freeway, single lane truck platoons and defines CACC as a platooning application to improve highway throughput in a stable and safe manner.

It is evident in the past literature [14–17, 21] that availability of co-operative information of other (mainly immediate leader) vehicles is the most integral part of a stable and safe platooning application. In most cases the co-operative information is available through wireless networks and has been shown that lack of reliable communication could result into unstable and unsafe events (eg., crashes, pileups). The loss in communication due to packet drops has been studied and tested in real world test-bed [41] and results conclude that in reality significantly large number of consecutive packet loss occur more frequently than previously assumed. It was shown in [41] that lack of line of sight, presence of other vehicles in the vicinity, and the topography of the roadway can impact the performance of vehicular networks. The block packet loss can trigger instability in a platoon due to uncontrolled acceleration or oscillations in the parameters that CACC controllers are designed to mitigate e.g. amplification in the separation error, oscillations in measurable parameters like acceleration, velocity and jerk (rate of change of acceleration) which render the platoon unstable [37]. Such behavior can lead to instability in a platoon which is deemed very unsafe for traffic applications [42]. In practice, such scenarios of large packet loss can occur in a realistic platoon when it approaches an infrastructure which could be acting maliciously to prevent channel access for any vehicles in the platoon; this eventually causes the platoon to lose transmissions of all basic safety messages. The affect of wireless channel access is more prevalent in DSRC because of the protocol design. DSRC is based on random channel access protocol, its fairly comfortable to consider that an erroneous and disruptive infrastructure can over crowd the channels with large number of continuous broadcasts originating from its transmitter. Another scenario of interest could be a faulty communication unit in one or more of the participating vehicles.

The past literature reflects various control theoretic approaches to controller design for the CACC application [14, 38]. Most of the focus exists in the form of string stability analysis to understand the effect of changes in velocity and acceleration of the front vehicle and its rippling effect on the following vehicles. Some previous work exists where vehicles with autonomous or semi-autonomous capabilities have been configured to support CACC and tested for stability of platoon with varying network performance [43, 44]. Also many papers provide modeling of the DSRC link between two CACC platoon vehicles and try to evaluate the effect of network behavior on the stability of CACC system [14, 17, 39]. One of the assumptions made in the previous works is that the average packet loss rate of basic safety message (BSM) transmitted between two vehicles is very low and further, the loss process can be modeled as a Bernoulli random process. Therefore, the probability of a vehicle observing a gap in the BSM stream is negligible [17], we will show this statement as not always true. The related works also assume all vehicles synchronously update their velocity and acceleration to following vehicles. It is also assumed that all vehicles have a method for measuring the accurate distance and relative velocity of the car ahead. Further, it is also assumed that all vehicles are in a single lane and avoid scenarios common to platoon formation in CACC such as merging and splitting of a platoon.

Earlier works on the CACC platoon have looked into understanding and characterizing string stability for vehicles in a platoon [40, 42], understanding the affect of sensor failures, cyber-attacks and security threats on the vehicles in a platoon [14, 17] and finding ways to improve traffic throughput on an arterial roadway network using tightly coupled platoon of vehicles [45]. There is limited research that looks at how the system behaves during long periods of blackouts in the wireless network. We find that the wireless communication necessary for the cooperative aspect of platooning could come under harsh environment and malicious network behaviors that need to be accounted for in order to characterize a stable platoon. Authors of [16, 46, 47] have presented methods to maintain stability and control of a platoon by switching from one mode of data acquisition to another. Studies such as [46, 47] provide switching criteria and method to turn a CACC system into ACC system. However, it has been evident, and we provide evidences in forthcoming sections, that ACC systems are not stable or efficient for shorter headway time and longer platoon. We believe that integrating the local sensors to supplement for the loss of valuable information about preceding vehicle (acceleration, velocity, separation) could help in maintaining a stable platoon during spotty communications. Work shown in [16] develop an algorithm to estimate the acceleration of leading vehicle through distance and velocity readings from local sensors using an estimation technique. It is limited in providing a realizable formulation of the problem, as well as testing when the platoon is a mix of different types of controller – CACC, ACC or Manual Driving. We observed following limitations and assumptions in the prior research:

- Limitations due to single lane of vehicles assuming there are no cross-longitudinal driving patterns.
- Homogeneous controller behavior assumptions due to homogeneous vehicles.
- Ill-defined assumptions related to controllers that attempt to mimic human driving or driving characteristics acceptable by human riders. These assumptions may be ill-fitting in a CAV domain.
- Limitations on deliberating ACC as the fall back controller mechanism in case communication among vehicles becomes unavailable
- Difficulty in modeling and reproducing published results due to insufficient information provided in the past literature.

A possible deployment path for this application will be to support autonomous trucks running in dedicated platoon traffic lanes as studied in [44]. Minimizing the average headway between all trucks in the platoon leads to maximal fuel efficiency as well as roadway throughput efficiency. This application has been widely studied by the ITS area in its earlier form known as adaptive cruise control (ACC). In ACC, each vehicle requires a Lidar or radar device to maintain the distance between it and the vehicle ahead. CACC uses the DSRC network to share acceleration (in some cases position and speed) data among participating vehicles along with on-board sensors like LIDAR and radar which makes the inter-vehicle relative distance very small compared to ACC. Connectivity adds robustness to the application by permitting more precise information (the acceleration of the car in front). We conjecture that more benefits are readily accessible if further capabilities are incorporated including:

- vehicles making use of more information about the surrounding traffic rather than just what they learns from the car immediately ahead
- localized decision making being enhanced by distributed decision making by allowing vehicles to form clusters and have leader vehicles elected to provide coordinated decision making.
- distributed decision making being enhanced if vehicles are able to communicate with multiple types of networks. For example, it is reasonable to assume that future CVs will be able to communicate using both DSRC and LTE.

Most recent researches highlight a merging of connected and autonomous vehicle research with CACC. The set of connected vehicle applications that are now widely common as safety features in new vehicles are being explored in a connected and autonomous vehicle (CAV) environment. This reformulates CACC as a two dimensional control problem. While adding complexity, it adds the modeling dimensions necessary to explore CACC with scenarios involving lane change and interaction with infrastructure. As described in [48–52] control of autonomous vehicles involves complex optimization of large amounts of time sensitive, locally derived, sensing information. One approach used for reducing computational complexity is to engage a two level controller. A high level path planner generates a reference trajectory (a prediction horizon). A vehicular controller ensures the vehicle follows the path in an optimal manner. A model predictive control (MPC) technique is commonly used to deal with non-linear optimization aspects. The car following control from prior CACC is now inherently performed by the AV controller. References [53, 54] are illustrative of recent CACC research that considers longitudinal control ranging in lane changes and obstacle maneuvering. In [53], CACC lane change is explored by introducing a virtualized vehicle to guide the vehicle changing lanes through the process. The concept of artificial potential fields is applied in [54] to the car following problem with lane change. The work in [55] develops a method by which a signal controller can detect and characterize a platoon of vehicles that pass through the intersection (or presumably any RSU on a roadway). Observed V2V messages are analyzed, and applied to a CACC model to estimate the operating parameters for a platoon.

Chapter 3

Wireless Technology for Vehicular Communications

In this chapter we summarize V2X systems from an 802.11p perspective. We assume that a default configuration will involve a single radio in each vehicle. The DSRC protocol supports two types of channels namely service channel (SCH) and control channel (CCH). To run CACC, the radio must operate in switching mode where it alternates between supporting the CCH and a SCH channel consecutively. However, we assume that the CACC operates on the continuous mode where a SCH is being utilized by all platooning vehicles for broadcasts. We show measurement and simulation results hat demonstrate the following issues:

1. The loss process can be complex (and definitely not well modelled by a Bernoulli loss model)

2. V2X suffers from congestion

3. Different vendors have different behaviors. For example, when a Cohda unit loses GPS sync, it can not transmit. Further, Cohda nodes equipped with two

radios can not transmit at the same time.

4. Network outage occurs frequently.

5. DSRC is vulnerable to malicious attacks

Vehicular networks are formed with varieties of communication technologies to tether them together in vehicle-to-vehicle or vehicle-to-infrastructure communications. The DSRC has been the most widely used wireless communication technology for vehicular networks. A member of the IEEE 802.11 (Wi-Fi) family of protocols, DSRC combines several features of 802.11a and 802.11e. Its physical and link layers are specified in the IEEE 802.11p standard [5]. The protocol supports different modulation and coding schemes with BPSK, QPSK, QAM-16 and QAM-64 with rate 1/2, 2/3 and 3/4 convolution error correction codes. The default modulation scheme used is rate 1/2 QPSK that can theoretically support 6 Mbps of data rate. It operates on 10 MHz channels using orthogonal frequency division multiplexing (OFDM) over 64 sub-channels of which 48 carry the data. In order to facilitate low latency data communication, IEEE 1609 wireless access for vehicular environments (WAVE) standard has been defined. The WAVE standard is a light-weight non-routable protocol designed for high performance, low latency operation in a single broadcast domain [5, 8]. The frequency allocation available in the U.S. is at 5.9 GHz with a set of seven channels of bandwidth 10 Mhz of which six are service channels (SCH) and one in particular, channel number 178 (5.885 GHz$-$5.895 GHz), is used as control channel(CCH). The control channel is used as a rendezvous frequency for all DSRC radios to listen for access and timing information from application or services that they may be interested in. Two consecutive channels can be grouped to increase bandwidth to 20 MHz for reduced transmission time and collision but it becomes susceptible to background noise and inter symbol interference [8]. We only consider 10

MHz channels and use one of the service channels for our study. The Link layer used in DSRC is based on the QoS strategy employed in 802.11e protocol. Enhanced distributed channel access (EDCA) protocol which provides different access class (AC) with different collision window and arbitration inter-frame space (AIFS) for supporting different quality of service [56]. As seen in Figure 3.1 the different access class are associated with different Contention Window length resulting in a gradation of priority. There are mainly four queues dedicated for 802.11p protocol. The voice queue has the highest priority while the background queue has the lowest priority (see also Table 3.1). The 802.11p MAC employs random access protocol with carrier sense multiple access (CSMA). Some prior research has studied other forms of MAC protocols with DSRC networks mostly scheduled channel access such as time divisiion multiple access (TDMA). However, given the highly dynamic channel conditions and temporal nature of network topology, the random channel access scheme CSMA was determined as the most practical channel access technology [8].

3.1 Performance modeling of DSRC

In order to support a multitude of vehicular applications, from safety to traffic date collection, it is important to understand how the DSRC protocol performs. Throughput, latency, packet loss, and reliability are some of the pertinent metrics that are used to convey information about the performance of a wireless technology. End-to-End throughput can give a realistic measure of the throughput sustained by a wireless protocol and which can be used to determine its applicability in certain high data rate demanding applications. Most of the CAV application do not demand high datarate but they have strong constraint on latency and packet loss. In the following section, we consider the effective throughput achieved by 802.11p and define

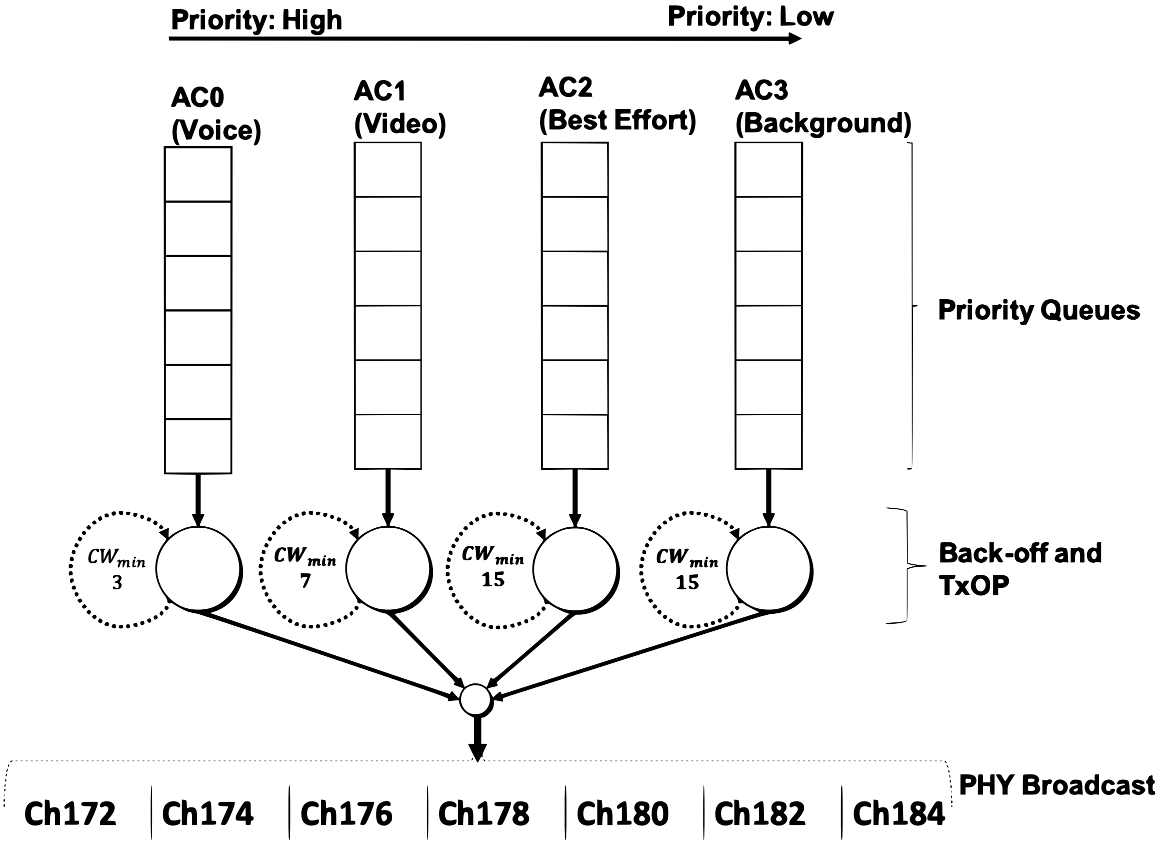


Figure 3.1: Access classes in 802.11p

the different metrics used in this study. Effective throughput is defined as the maximum achievable theoretic throughput after considering the overhead and congestion. Nominal throughput is defined as the physical layer throughput calculated by the information bit per symbol duration. The following discussions assume the use of UDP broadcast in IP mode over 802.11p and do not consider WAVE broadcasts over 802.11p.

3.1.1 Effective Throughput

Purpose of this discussion is to calculate the maximum effective throughput supported by different modulation and coding schemes in 802.11p protocol. Figure 3.1 shows different access class queues in operation in an 802.11p Link layer. As shown in figure 3.1, high priority queues have shorter AIFS duration and smaller congestion window size causing the packets to be sent much sooner than the low priority queues. This makes it perfectly applicable for safety and emergency services to use the high priority queues while other service applications use other low priority queues. Table 3.1 below shows different 802.11p physical and link layer parameters including the overhead bits, timing information and arbitration delays. Table 3.2 shows nominal and effective throughput (broadcast) attainable by different modulation and coding schemes in 802.11p for an application protocol data unit (APDU) of 1472 Bytes.

To calculate the effective throughput considering all the overhead bits and arbitration delays, we explore the problem by identifying the total number of OFDM symbols required to transmit a message of size x bytes and the total time it takes to transmit that many OFDM symbols. If we consider the APDU size is x bytes then the size of UDP datagram becomes $U_x = x + 8$ bytes. The network layer packet with UDP datagram and network headers is $N_x = U_x + 20$ bytes. The 802.11 protocol

Table 3.1: DSRC 802.11p physical and link layer parameters

802.11 Header + FCS (Bytes)	20
OFDM Service Bits (bits)	16
Tail Bits (bits)	6
SIFS	32 μs
Timeslot (t$_{slot}$	13 μs
PLCP Header	32 μs
Signal Field	8 μs
OFDM Symbol Guard time	1.6 μs
OFDM Symbol time	6.4 μs
OFDM data subcarrier number	48
OFDM pilot subcarrier number	4

Access Class[i]	**Voice** [0]	**Video**[1]	**Best Effort**[2]	**Background**[3]
AIFSN[i]	2	3	6	9
AIFS[i] = AIFSN[i] $\times t_{\ slot}$ + SIFS	58 μs	71 μs	110 μs	149 μs
CW$_{min}$[i]	3	7	15	15

header and frame check sequence added to the packets coming from network layer is of size $W_x = N_x + 20$ bytes. The total size of the link layer frames before padding for multiplexing into OFDM symbols is $P = W_x \times 8 + 16 + 6$ bits. If the information bits per modulation symbol is k, code rate is r, and total number of OFDM subcarriers per symbol is $O = 48$. Hence the information bits per OFDM symbol is equal to $M = k \times r \times O$. Also the total size in bits after padding for OFDM symbols is $S = \left(\lfloor \frac{P-1}{M} \rfloor + 1\right) \times M$. The OFDM symbol duration is $t_{OFDM} = 8$ μs and PLCP header duration (Preamble duration + Signal time duration) is $t_{PLCP} = 32 + 8$ μs [22]. Hence the total number of OFDM symbols required for transmission x bytes of APDU is $O_x = \frac{S}{M}$. The arbitration inter-frame separation for 802.11p (AIFS[AC$_i$]) is AIFSC[AC$_i$] $\times$ t_{slot} + SIFS and the arbitration time slot (t_{slot}) is 13 μs. If the minimum collision window size is CW_{min} then duration to transmit O_x OFDM symbols is calculate as $T_x = O_x * t_{OFDM} + t_{PLCP}$. And the total duration including the arbitration time is $T_{total} = T_x + AIFS[AC_i] + \frac{CW_{min}}{2} \times t_{slot}$. Eventually, the effective throughput for APDU of size x bytes (Th_{eff}) is calculated as $\frac{x \times 8}{T_{total}}$

Hence, the effective throughput (Th_{eff}), achievable through 802.11p protocol in the scenario containing two nodes only can be calculated as shown:

$$Th_{eff} = \frac{x}{\left\lfloor \frac{(x+OVHD)\times 8+ST-1}{M} + 1 \right\rfloor + \frac{t_{PLCP}+AIFS[i]+\frac{CW_{min}}{2}}{8}}$$

where, $OVHD$ is the overhead associated with different layers of protocol stack, ST is the length of OFDM service and tail bits, and x is the APDU in bytes. The values of these parameters are given in Table 3.1 and Table 3.2. Figure 3.2 shows the effective throughput, under *Best Effort* Access Class with different modulation and coding rates, calculated for different APDU. Comparing to the nominal throughput Th_{nom} given in Table 3.2, the efficiency of transmission decreases with $APDU$ size because of relatively large overhead needed for smaller APDU. Larger packet sizes provide higher throughput but smaller packet sizes are usually the norm for vehicular applications such as CACC.

3.1.2 Latency and Reliability

Latency can be defined in terms of delays related to application layer measurements or link layer measurements. The application level latency captures the delays associated with queuing of the packets at various network queues and channel access. We consider an average latency measurement as one form of the metric. Consider there are $L = l_1, l_2 ... l_n$ latency measurements in time window T, then latency metric τ can defined as :

$$\tau = \frac{\sum_{m=1}^{n} l_m}{n}$$

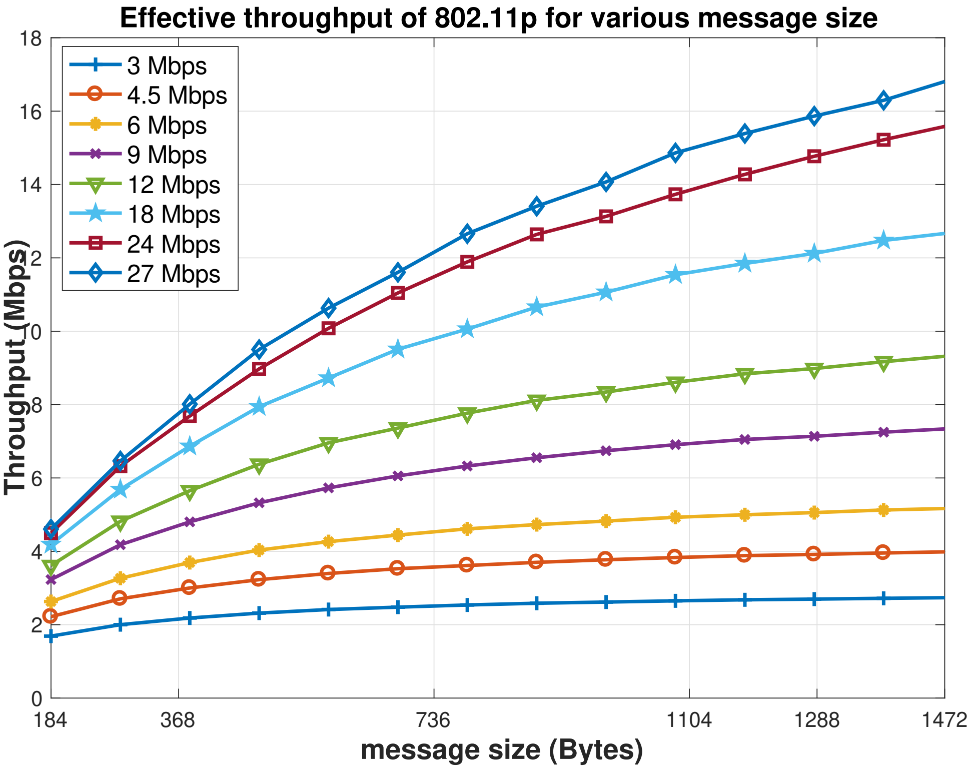


Figure 3.2: Effective throughput for various size APDU in best effort queue

Table 3.2: DSRC nominal and effective throughput (APDU = 1472 Byes)

MCS	**1/2 BPSK**	**3/4 BPSK**	**1/2 QPSK**	**3/4 QPSK**	**1/2 QAM16**	**3/4 QAM16**	**2/3 QAM64**	**3/4 QAM64**
M bits/symbol	24	36	48	72	96	144	192	216
Th$_{nom}$ (Mbps)	3	4.5	6	9	12	18	24	27
Th$_{eff}$(Mbps)								
Voice	2.82	4.16	5.48	7.97	10.39	14.77	18.71	20.53
Video	2.79	4.11	5.38	7.77	10.04	14.08	17.62	19.23
Best Effort	2.73	3.98	5.17	7.33	9.32	12.70	15.50	16.74
Background	2.71	3.93	5.08	7.15	9.04	12.18	14.75	15.86

where τ captures the effect of end-to-end delays adherent to contention, channel access and queuing.

Packet reception rate is defined as the ratio of number of packets received at the application end point of a receiving wireless node over the total number of packets sent from the application end point of the transmitting wireless node. If P_r is the packet reception rate, than $1 - P_r$ is called the packet loss rate. Packet loss rate (P_l) can capture the notion of losses in network stack due to queue overflow and losses in transmission due to channel noises and interference. In particular we are also interested in packet loss due to channel congestion created by large number of subsequent transmitting nodes or a malicious node hijacking the channel with high date rate transmissions.

Channel busy ratio (CBR) defines the amount of time a transmitting node found the channel to be busy during its arbitration period and had to back-off based on its Congestion Window size. This metric is helpful in understanding the normal operation of channel usage and compare it with abnormal behavior due to malicious node activities.

Mean burst length (MBL) defines the average length of burst packet loss observed in a communication network. A loss of more than or equal to two consecutive packets is called a burst loss. If the packet loss process considered is uncorrelated in nature based on Bernoulli loss process, then the MBL is equal to 0. This measure can provide some insights into how the packet loss actually occurs in a realistic wireless network. Mean good length (MGL), correspondingly, defines the average length of good packet burst between packet loss events. If $L_i, i = 2, 3, 4 \ldots$ be the number of loss packet burst of length i and $G_j, j = 2, 3, 4 \ldots$ be the number of good packet burst of length j then:

$$MBL = \frac{\sum_{i=2} L_i \times i}{\sum_{i} L_i}$$

and,

$$MGL = \frac{\sum_{j=2} G_j \times j}{\sum_{j} G_j}$$

Reliability (R) is defined very similar to T-window reliability metric given in [12].Reliability metric calculates the ratio of number of packets received within a given time over the maximum number of packet expected within the same time for a given transmission rate.

$$R \;\; = \;\; 1 - \frac{failed_N}{total_N}$$

where, $failed_N$ is the number of packets lost in time T and $total_N$ is the total number of packets that can be received in time T.

3.2 Testing and Results

The study presented here is categorized into different tests that were done to understand the performance of 802.11p networks in broadcast modes at various scenarios such as:

1. Test I: lab scenario with good line of sight and perfect channel condition,

2. Test II: Vehicle to Infrastructure tests using Clemson University test-bed [57].

The motivation for these tests was to study the real world performance of 802.11p protocol and to characterize the performance in terms of different metrics using test data. The metrics we report include: 1. channel busy ratio, 2. latency, 3. throughput 4. packet loss, 5. mean burst length and 6. reliability.

Packet arrival rate, latency and packet loss are particularly interesting in the case of vehicular networks because of the highly dynamic nature of the channel. Drastic changes in channel conditions can be triggered by sharp curves in the roadway, traffic congestion (especially when large trucks are involved), and even speed bumps. Also, due to the rapid changes in network conditions, the packet loss process is not intuitive. Losses are not always independent and uncorrelated as commonly assumed in the literature [14, 21].

The platform used in the study was Cohda Model MK5. A block diagram of the device is shown in Figure 3.3. It consists of an embedded Linux environment with two sets of DSRC radios connected to the Linux system via a high speed USB bus. Only one of the radios was employed in our test. Cohda platform provides a software development kit (SDK) that can be installed on a host computer and used to develop applications for the DSRC systems. The applications are transferred to the DSRC systems using the gigabit Ethernet interface as shown in Figure 3.4.

For our tests, the radios were configured to use a single fixed channel for all transmissions. Transmit power was set to 20 dBm (0.1 watt) for all tests.

3.2.1 Test I (Lab Test)

Understanding the performance of DSRC radios without incorporating channel noise was the motivation for the lab tests in test I. As shown in Figure 3.4 Two DSRC radios in line of sight and close range were used with a Constant Bit Rate (CBR) application to capture the performance metrics. During preliminary testing it was discovered that sending UDP data at rates approaching the maximum effective

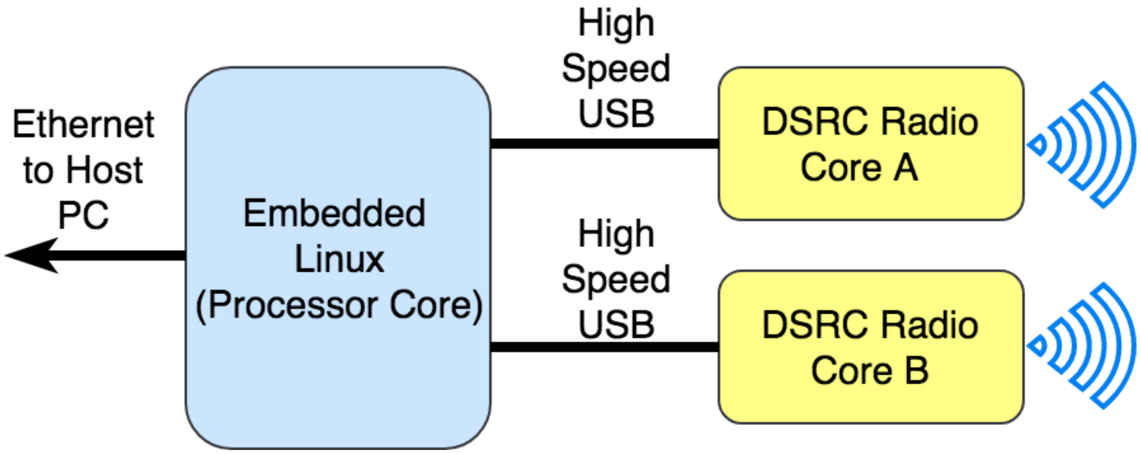


Figure 3.3: Testing Platform, DSRC Radio

throughput could queue overflow and packet loss within the sending DSRC unit. The lab tests were conducted to identify the maximum sending rate that the DSRC units can support.

3.2.2 Test II (V2I Test)

Test II consists of vehicle-to-infrastructure (V2I) testings done using a DSRC on-board unit (OBU) installed on a vehicle and driving in and out of a DSRC coverage zone of an existing road side unit (RSU) installation [57]. Our tests employed two of the eight supported signalling rates: 3/4 QPSK which provides a PHY layer bit rate of 9 Mbps; and 3/4 QAM-64 which provides a rate of 27 Mbps. One of the motivations for the set of measurements in test II is to understand the packet loss process in good and bad network conditions. As illustrated in Figure 3.5 the mobile OBU passes through a no-line-of-sight zone as it entered the wireless coverage of RSU followed by good line-of-sight zone and leave the coverage zone as it drove away. The tests were done on a clear day with moderate vehicular traffic conditions.

In this section we provide results and analysis based on the observation from

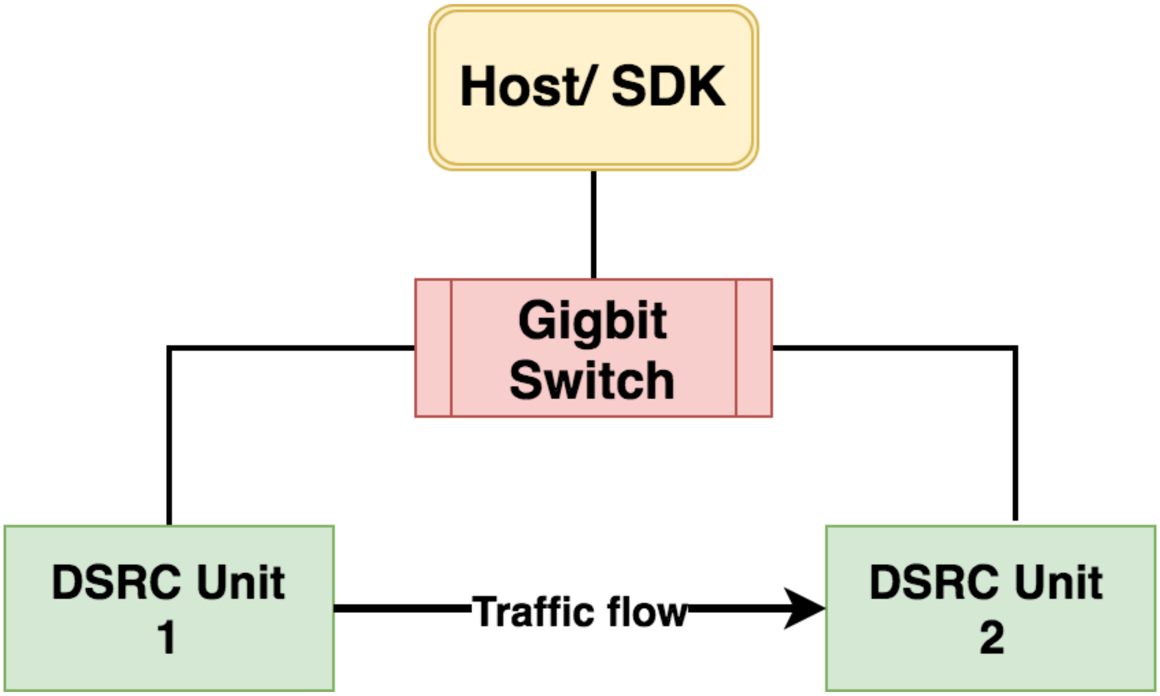


Figure 3.4: Testing Platform and setup for Test I

the tests performed during this study.

3.2.3 Results:Test I

Figure 3.6 below shows the broadcast throughput observed during lab tests with four different modulations. Transmit rates used were equal to the nominal throughput calculated in Section 3.2. The difference in nominal and observed throughput in higher modulation of QAM-16 and QAM-64 pertain to the packet loss. From Figure 3.6 it can be concluded that to avoid queue losses at the transmitter, its necessary to avoid transmitting at rates higher than the rates given in Figure 3.6. Another important metric of study in lab settings is the channel busy ratio (CBR), shown in Figure 3.7, expressed as average percentage of time the channel was found to be busy during a testing cycle. Figure 3.7 illustrates that higher modulation and coding can

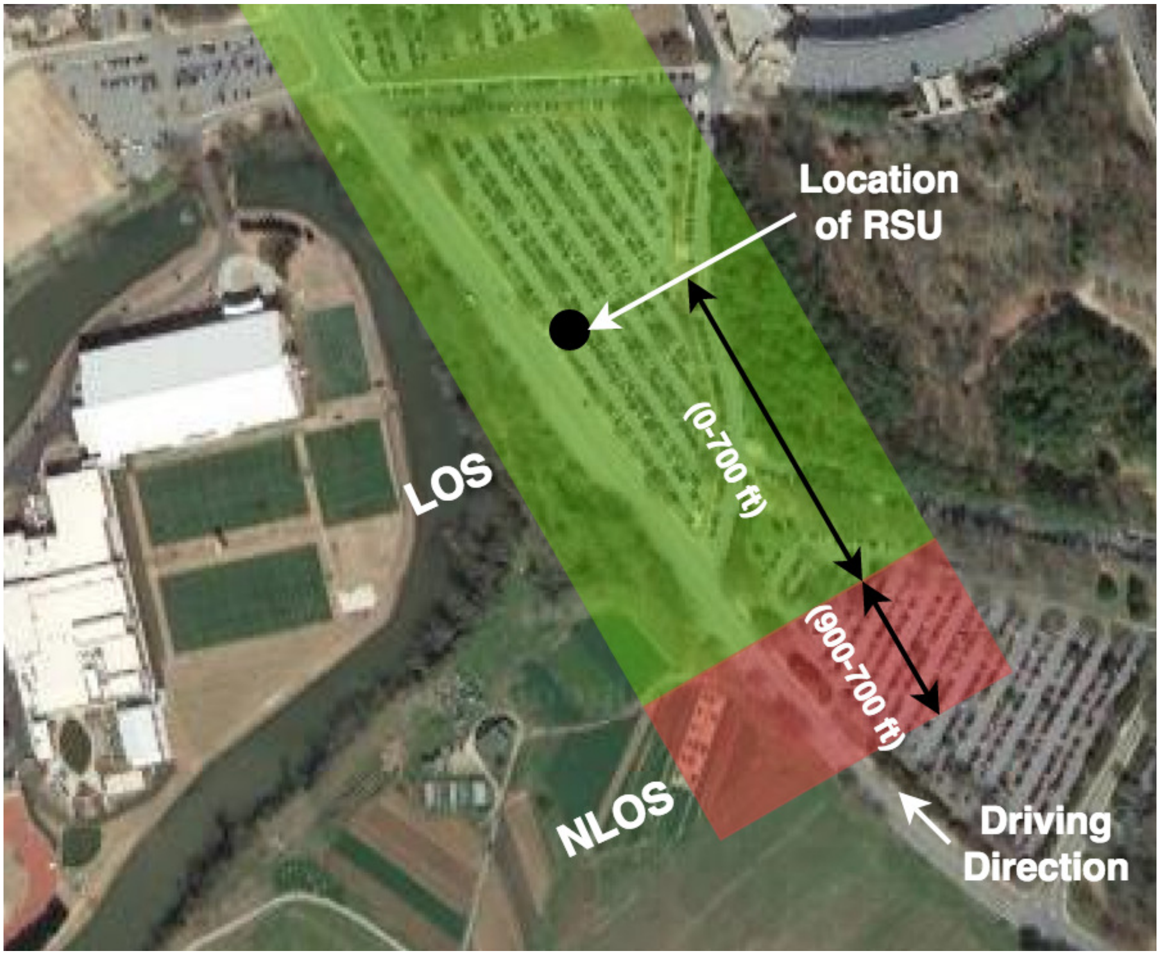


Figure 3.5: Test Scenario and Location for Test II

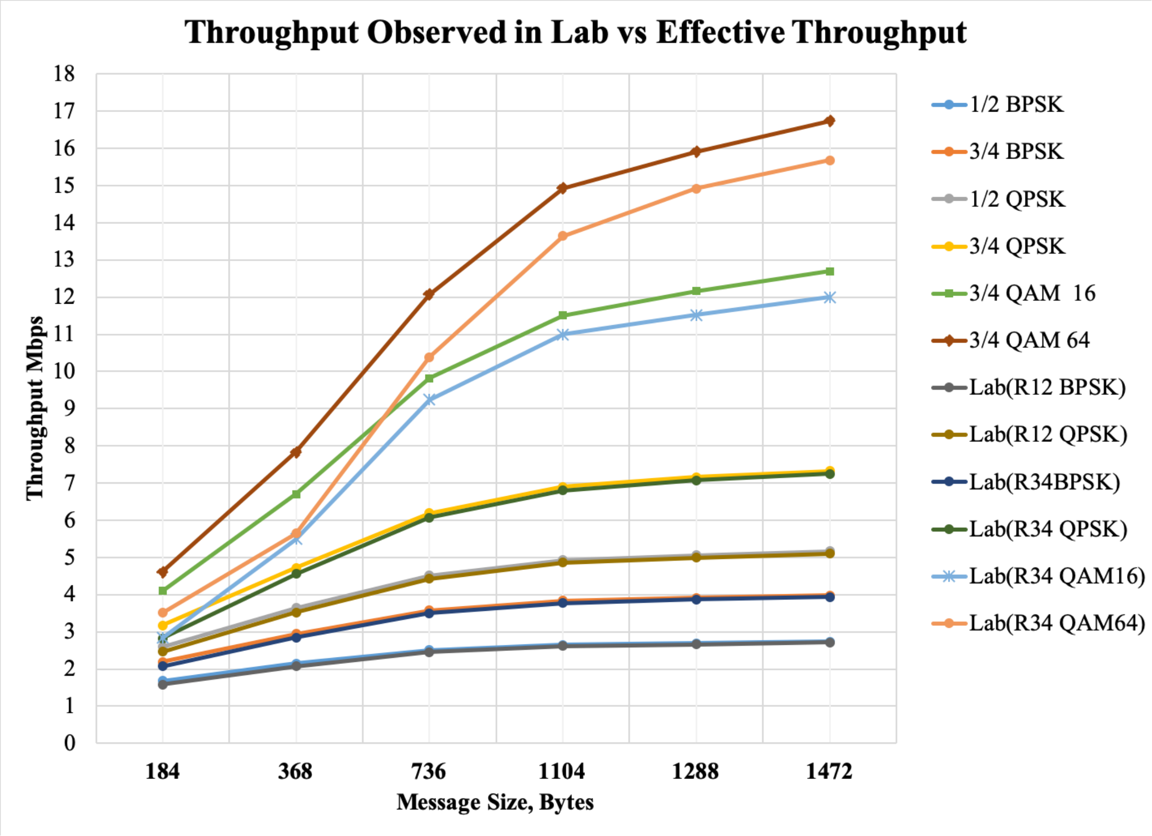


Figure 3.6: Throughput observed,Test I

reduce congestion and collision probabilities due to reduced channel utilization.

Nominal transmission time for different sizes of APDU is shown in Figure 3.8. As expected the transmission time increase with packet size as well we small modulation schemes. Figure 3.9 shows the observed end-to-end latency in test I where the CBR application was run at low transmission rates to avoid queue losses. The latency seen in Figure 3.9 is consistent as expected. Latency is calculated as an average observed latency of received packets at the application layer. With higher bit rates,latency decreases even with increased transmitted packet per second because of the reduction in packet transmission time. In summary, the throughput and latency results are consistent with results published in the earlier literature [5, 22].

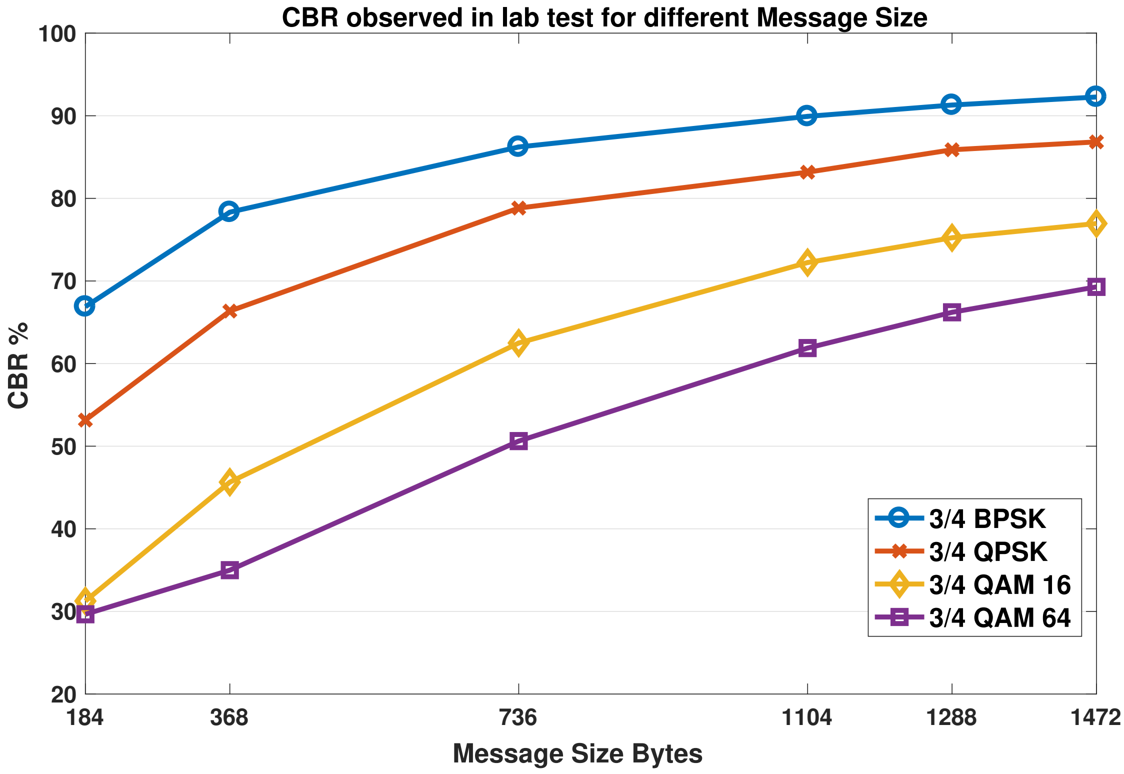


Figure 3.7: Channel Busy Ratio, Test I

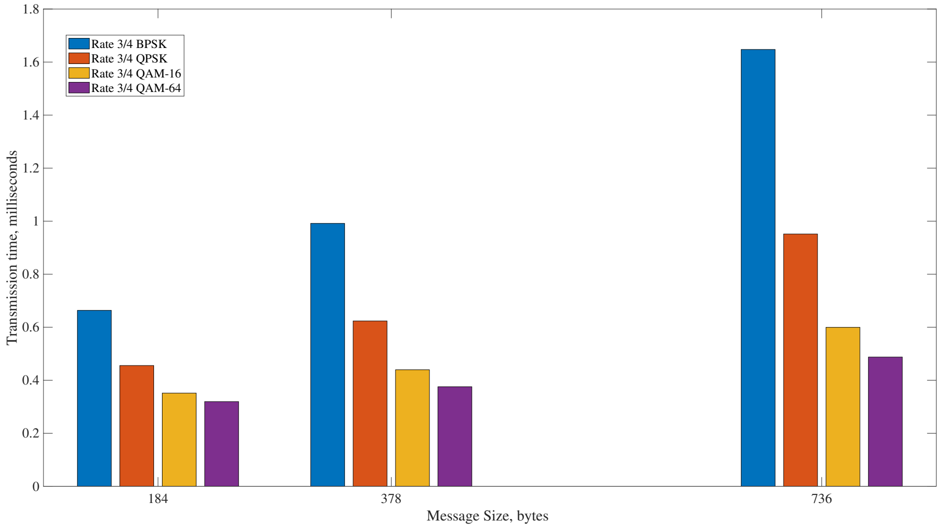


Figure 3.8: Test I: Nominal transmission time

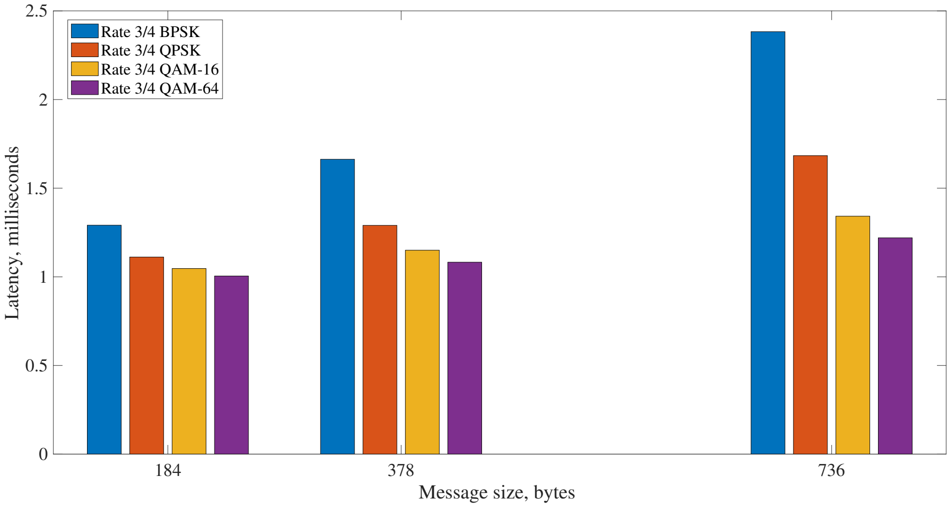


Figure 3.9: Test I: Observed end-to End latency

Table 3.3: Parameters used for Test II

MCS	Rate 3/4 QPSK	Rate 3/4 QAM-64
Transmit Rate	1.5 *Mbps*	2.5 *Mbps*
Message Size	184 Bytes	184 Bytes
Speed	30-35 mph	30-35 mph

3.2.4 Results:Test II

Test II (V2I) was conducted on a four-lane roadway segment on the campus of Clemson University [57]. RSU's were installed on lighting poles along the right-of-way as illustrated in Figure 3.5. Multiple runs of similar driving patterns were conducted. The vehicle started at a distance of about 900 ft in a non line-of-sight (NLOS) condition and moved towards the RSU through line-of-sight (LOS) condition and drove past the RSU. Test II was carried out by transmitting at rates to avoid transmit queue losses but much greater than 10 packets/second (Basic Safety Message transmit rate) in order to obtain large number of data points. A constant bit rate application was used and various physical, link, and application layer parameters were recorded. Table 3.3 describes the settings used in transmitter and receiver DSRC units for Test II.

We defined the reliability time slot T as 0.01 seconds and the minimum number of packets to be received N to be 5 packets. Another metric of interest is the mean burst length (MBL) which is defined as the average length of burst loss. A correlated burst noise process will show larger MBL than independent uncorrelated noise process. This metric is a good measure of how much burst noise effects the communication channel and also if the loss process exhibits correlation.

Figures 3.10 above shows the resulting plots of mean burst lengths, reliability metric, and packet loss rate for multiple test runs of rate 3/4 QAM-64 modulation and coding. Similar observations were found for rate 3/4 QPSK as well. Received

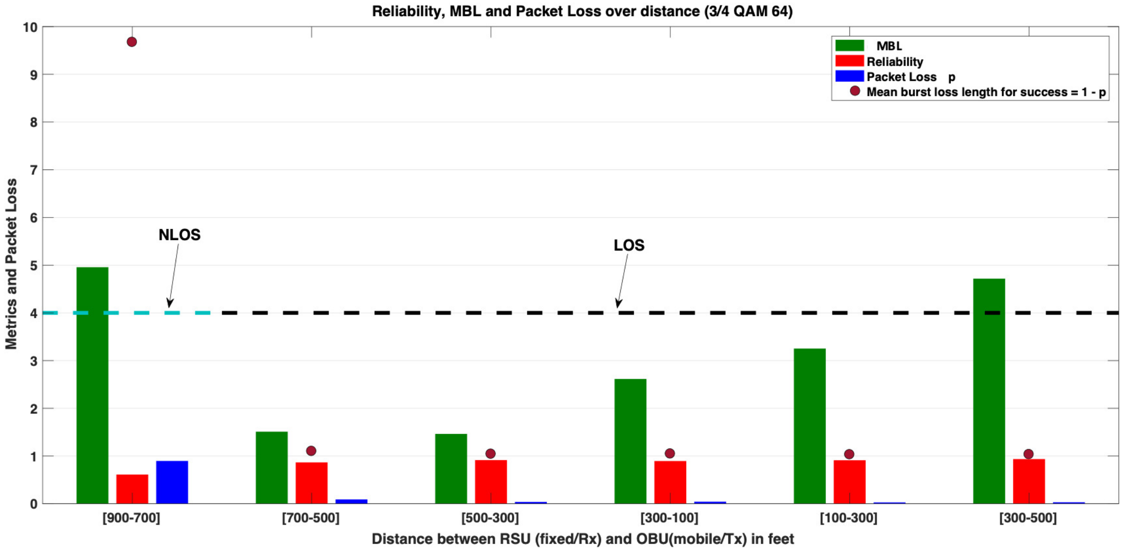


Figure 3.10: Metrics observed in Test II for rate 3/4 QAM-64

packets and loss events were grouped based on distance from the receiver as the vehicle moved towards and away from the RSU. Subsequently each bin of data was used for calculating the metrics. The NLOS and LOS indications on the plots show the region obstructed and unobstructed by foliage and roadway curve respectively. We can observe the effect of NLOS and the loss behavior through the reliability metrics degradation and increased MBL as well as increase in packet loss. The average MBL over all the data set was found to be 4.925 and average packet loss rate over the entire test was 0.2925. The average MBL value shows that Bernoulli random process is not accurate for implementing packet loss in vehicular networks. We also plotted the geometric distribution mean for a success probability of p = 1- packet loss probability observed in the test. The mean length of loss calculated based on the mean number of trials for a success in a geometric distribution is plotted as shown in Figure 3.10. The degradation in channel performance for the DSRC network in our study was due to distance and lack of line of sight, but factors such as as weather and high vehicular traffic congestion can also decrease the performance. The study of congestion involving larger numbers of vehicles is very difficult to evaluate in real testbeds. The vast majority of vehicular studies are therefore based on simulation. While our results are based on a single data set, it represents an interesting loss process sample path that includes correlation.

3.3 Simulation of DSRC in ns-3

In order to understand the affects of large number of nodes creating a congested network or a node acting in a malicious way (e.g. creating a Denial of Service attack) on the platooning application, a simulation of constant bit rate generator, similar to the one used earlier, was developed in ns-3 [58]. The ns-3 is widely used for simulating and analyzing large scale networking systems. In our case, we used ns-3 to simulate the following scenarios and evaluate 802.11p performance based on:

1. Effective throughput for different ranges between transmitter and receiver
2. Mean burst length(MBL) and mean good length (MGL) based on congested scenario
3. MBL and MGL based on malicious attack nodes

In order to analyze the effective end-to-end throughput observed at the application layer of a constant bit rate generator, a pair of 802.11p nodes were placed at different distances. To understand the affect of channel on the wireless broadcast along with implications of different modulation and coding schemes, the tests were conducted between two 802.11p nodes and effective throughput was captured while moving them further away from each other. The channel model used for the testing was urban scenario with Nakagami-m fast fading and log distance path loss

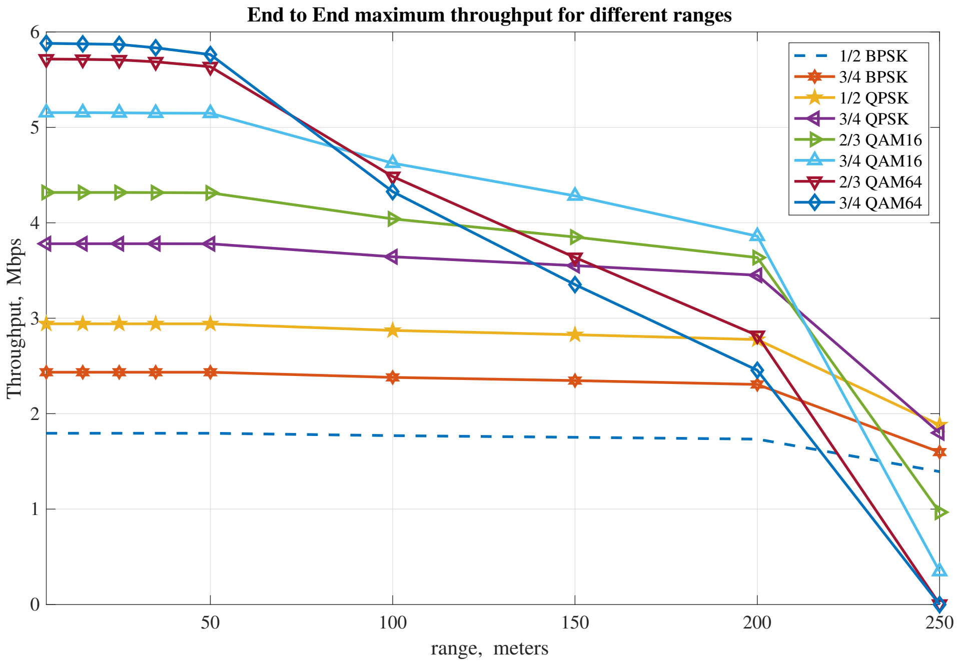


Figure 3.11: Maximum End to End throughput observed between two nodes at different separation

model based on [59]. Figure 3.11 shows the effective throughput observed between two 802.11p nodes separated by various distances. It can be observed that the higher modulation schemes suffer larger packet loss due to the increased probability of channel symbol error. In these tests the transmitter power was set equal and that causes more symbol error in higher modulation, like QAM-64, compared to lower modulation schemes like BPSK or QPSK. However, the bit rate is much smaller in lower modulation and coding schemes compared to higher modulation.

Similarly, to test the scenario of congestion, a pair of nodes with a designated flow (i.e. constant bit rate source and sink applications) were simulated with a varying number of nodes surrounding the two nodes. All of the nodes were broadcasting at 40 packets per second (the same rate used by CACC applications discussed in

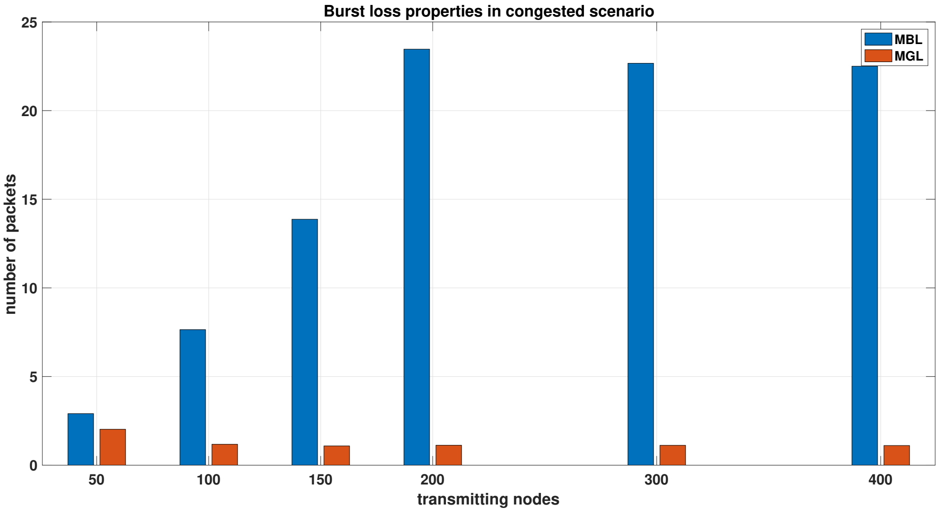


Figure 3.12: MBL and MGL observed between two nodes with different numbers of consecutive broadcasting nodes

next chapter). As the number of nodes were increased from 50 to 400, the effect of burst packet loss was observed. Figure 3.12 shows the values of mean burst length (MBL) and mean good length (MGL) observed at different number of surrounding broadcasting nodes is increased. The MBL and MGL seen here were recorded at the receiver hence they represent the effect on application metrics due to wireless channel conditions. The simulation was done using Rate 1/2 QPSK modulation with a square area of 200 by 200 meters. The default transmitting powers were used as well.

An attack scenario was simulated by using 51 nodes in total in an area of size 200 by 200 meters. The transmitting (node 0) and receiving node (node 49) were setup at 50 meters apart while the attack node (node 50) was set exactly mid way between the two. The rest of the nodes were randomly placed around the two transmitting and receiving nodes. All of the nodes except the attack node were broadcasting vehicular data (200 Bytes messages) at 40 packets per second. The different modulation and coding schemes were simulated with attack node designed

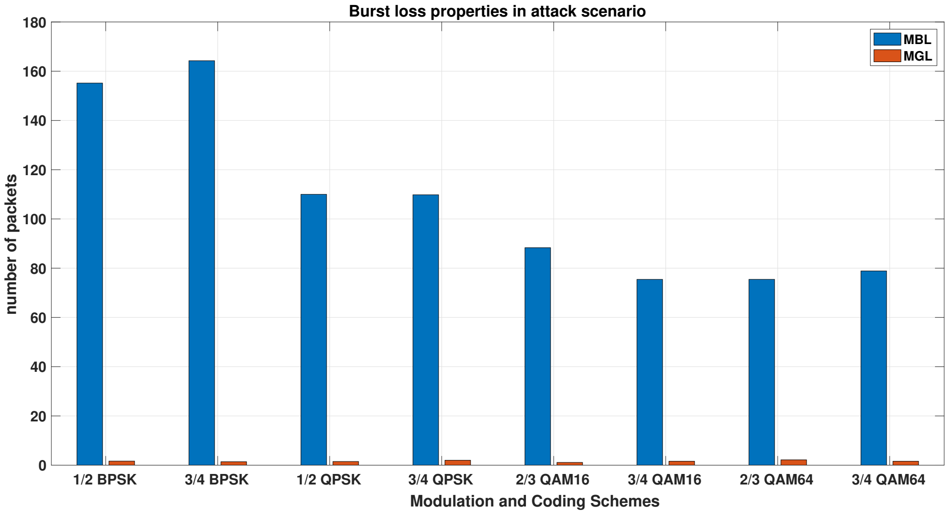


Figure 3.13: MBL and MGL observed between two nodes with third node as an attacker

to broadcast at the rate of maximum achievable throughput for each modulation and coding scheme. Therefore the attack node was able to keep the channel busy and cause collision during transmission. The MBL and MGL observed between node 0 and node 1 is illustrated in Figure 3.13. These observations are very helpful in designing different test scenarios for platooning application as shown in later chapters.

Chapter 4

Cooperative Adaptive Cruise Control

In this chapter we describe the operating details of a CACC controller. We present the model through the perspective of traditional CACC controller and present the issues created due to loss in wireless networks. The fall back mechanism and switching between the fall back controller and traditional CACC controller is also presented.

4.1 Introduction to CACC

Cooperative adaptive cruise control is a robust algorithm that actuates the control and coordination of acceleration of a vehicle based on the acceleration of preceding vehicles acquired through wireless broadcasts. A CACC system consists of two levels of control,

1. a sensing and regulation controller that issues acceleration/deceleration commands u_i, and

2. the vehicular dynamics controller that issues commands to the vehicle mechanics based on u_i.

. The platooning vehicles should be capable of merging and splitting maneuvers used in joining and leaving a platoon. Aspects of platoon formations, clustering and management as well as how it evolves over time are defined in [13, 60] and are found to depend on how vehicles communicate and process the information. Clustering strategies include:

- *Ad Hoc*, which might support ACC, or a platoon involving a mix of vehicular capabilities.

- *Local coordination*, where Vehicle to Vehicle (V2V) communication is used to form and manage the platoon.

- *Global coordination*, where Vehicles communicate with a centralized controller possibly using Vehicle to Infrastructure (V2I) networks.

In our study we consider the platoon is already formed by following any one of the above mentioned strategies. Our research looks at illustrating reliable CACC system using either local or global coordination. Along with the formation, stability and safety is another inherent criteria that separates a functioning platooning from those which eventually end up disastrous.

Cooperative Adaptive Cruise Control was developed as an extension to the existing platooning implementation called Adaptive Cruise Control [61]. Adaptive Cruise Control regulates the speed of the subject vehicle in a platoon based on the distance and relative speed of the vehicle in front and shows limitations in stability of the platoon and constraints towards ability to minimizing the inter vehicle separation. Minimizing the inter vehicle separation is important for two reasons: increasing traffic throughput and reducing air drag resistance to improve fuel efficiency. However there are issues mostly pertaining to safety and stability of the platoon that prevent the separation from becoming relatively smaller. It has been shown that CACC can improve string stability as well as reduce inter vehicle separation [14, 40, 45, 61]. Studies and tests conducted in California PATH program [38, 62] have shown practical CACC systems and compared the outcome with standard ACC systems. In many of the studies it is shown that ACC systems are string unstable and are unable to reduce separation gap to a relatively smaller value. In recent years, the research in CACC platoon have gathered more interest in the areas of arterial intersections and the interaction of platoon with infrastructures [40].

4.2 Types of CACC

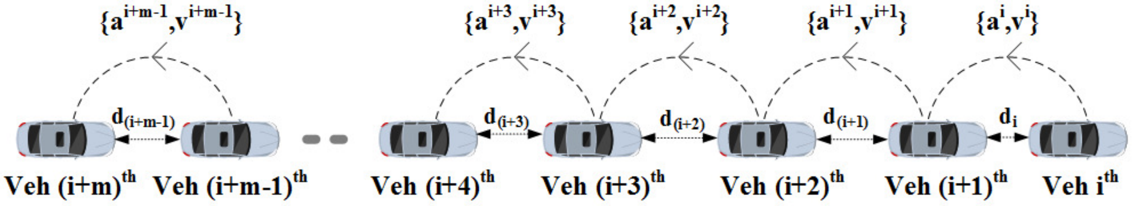


Figure 4.1: CACC platoon

A very simplified model of CACC platoon, such as given in Figure 4.1 consists of a string of finite number of vehicles in one platoon with a leader vehicle and a set of following vehicles. The platoon consists of a series of vehicles creating a chain leveraged by wireless communication and sensors. The vehicle in the front is called *leader* while the series of vehicles following the leader are called *followers*. The wireless communication between vehicles in a platoon is supported by DSRC 802.11p standard of 5.9 GHz frequency spectrum with OFDM and CSMA mac protocol [8]. There are different versions of CACC strategies depending on how the communication network is being utilized. We consider that the most relevant information is coming from the vehicle immediately preceding the *ego* vehicle (vehicle in consideration). There are a few assumptions motivating the simplification of this system,

- the number of vehicles correspond to the length of the platoon. The distance from the Leader vehicle to the tail vehicle is called the platoon length.
- the traditional CACC implements V2V communication (i.e., the communication is considered between vehicles only). However, in this study the decision making global controller can be located on the roadside edge compute node in which case the communication is V2I.
- the platoon operates on a single lane such as a platoon of autonomous trucks used in freight transportation. We assume the vehicles have homogeneous property in terms of their kinetics. Therefore mass, inertia, rolling friction, torque time constants and other attributes are same in every vehicle.

CACC is the result of an optimization problem that minimize the error in inter vehicle separation gap. Every time the CACC controller operates, its resulting target acceleration for the *ego* vehicle minimizes the error in inter vehicle separation. There are different CACC types based on the different strategies to minimize separation gap error between two consecutive vehicles. In recent literature most of the studies have revolved around devising strategies for gap regulation in a CACC platoon system:

- Constant time headway (h) - Time headway, h is defined as time taken by the following vehicle's front bumper to successively pass the point on the roadway that was previously passed by the back bumper of the immediate leading vehicle

[40]. In this strategy, the separation between two vehicles is dependent in the current velocity of the follower vehicle and the preset gap called stop distance (d_{stop}). In this study, terminology for time headway and headway are used interchangeably.

- Constant gap headway - the distance (in units of length) between two consecutive vehicles is fixed irrespective of their velocity and acceleration. Eco-driving strategy is one branch of constant gap headway strategy because the optimization on reduced fuel consumption results in reduced separation between vehicles at any speed.

- Constant safety factor criterion - this strategy is based on limiting the collision to one platoon and adheres to the idea that the affect of rear end collision never propagates to multiple platoons of CACC vehicles. The headway is calculated so that any collision or incident occurring in a platoon will be restricted to that platoon and the lead vehicles of immediate following platoon will have ample time to slow down or stop.

4.3 CACC System Model

In this section we provide details into the modeling of CACC controller. We consider the CACC as a platooning application where a set of vehicles are creating a cohesive chain by safely following the vehicle in front. First, we define the vehicle model similar to [14, 17, 42]

$$\begin{bmatrix} \dot{d}_i \\ \dot{v}_i \\ \dot{a}_i \end{bmatrix} = \begin{bmatrix} v_i - v_{i-1} \\ a_i \\ -\frac{1}{\tau}a_i + \frac{1}{\tau}u_i \end{bmatrix} \forall i \in \mathbf{V}_m \tag{4.1}$$

where a_i, d_i and v_i are acceleration, velocity and relative distance of i^{th} vehicle V_i where $\mathbf{V}_m$ is the set of all vehicles in a platoon. The acceleration is considered linear over the time constant of the drive-line dynamics which can be calculated as $\dot{a}_i$ = (u_i - a_i)/τ where u_i is the actuation signal (i.e., in case of CACC, the target acceleration of the vehicle) and τ is the time constant of the vehicular dynamics. For simplicity, we assume a piece-wise linear vehicular dynamics and the distance travelled (s$_i$), velocity (v$_i$) and acceleration (a$_i$) of any vehicle i in the platoon at time t can be calculated as follows :

$$\begin{aligned} a_i(t) &= a_i(t - t_0) + \dot{a}_i(t) \times t \\ d_i(t) &= v_i(t - t_0) \times t + a_i(t - t_0) \times \frac{t^2}{2} + \dot{a}_i(t) \times \frac{t^3}{6} \\ v_i(t) &= v_i(t - t_0) + a_i(t - t_0) \times t + \dot{a}_i(t) \times \frac{t^2}{2} \end{aligned} \tag{4.2}$$

where, t_0 is the earlier time instance when the values were calculated using equation 4.2.

Figure 4.1 shows a platoon in operation with m vehicles. An *ego* vehicle is the vehicle under consideration, *leader* vehicle is the vehicle in the head of the platoon and *lead* vehicle is the vehicle immediately in front of the ego vehicle. For example if V_3 is an *ego* vehicle in the platoon of Figure 4.1, V_0 is *leader* vehicle and V_2 is *lead* vehicle. The *leader* vehicle is designated as V_i and the last vehicle participating in the platoon is designated as V_{i+m}. We limit our analysis to one lane roadway without intersections or arterial inner city road networks. There is only one platoon active in a scenario with other vehicles on the roadway used for analyzing affect of large number of broadcasts. Figure 4.2 shows the modeling of a vehicle which is capable of CACC platooning. The Figure shows different components that make up an application like CACC. Sections below describe these components in detail.

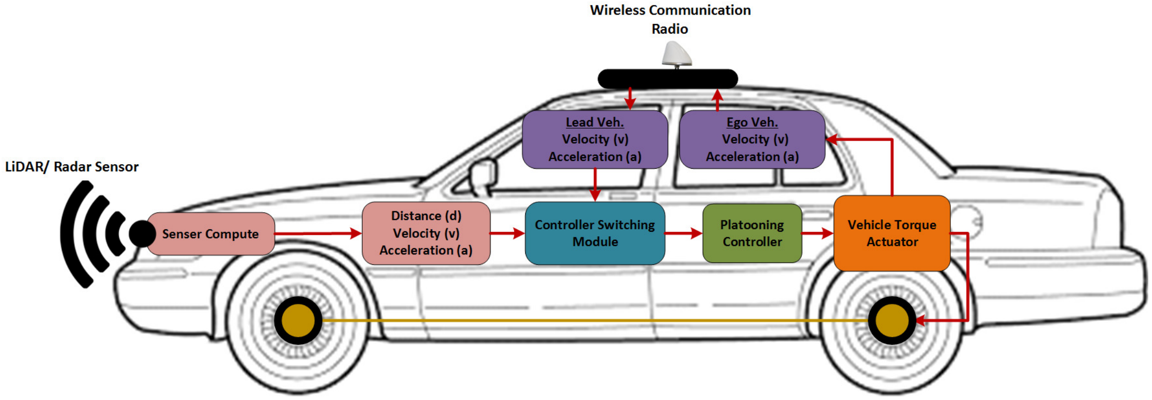


Figure 4.2: CACC Vehicular System Model

4.3.1 CACC Platoon Controller

In this section we provide details of the platoon controller as shown in Figure 4.2. The platoon controller is responsible for calculating the target acceleration for the ego vehicle. The equations governing the platoon controller are given below. The controller is optimized to minimize the distance separation error e_i. For a vehicle V_i the platoon controller equations are given as

$$d_{r,i}(t) = d_{stop} + h \times v_i(t) \tag{4.3}$$

$$e_i(t) = d_i(t) - d_{r,i}(t) \tag{4.4}$$

$$\dot{u_i}(t) = -\frac{1}{h}a_i(t) + \frac{1}{h}(k_p \times e_i(t) + k_d\dot{e}_i(t)) + \frac{1}{h}a^{i-1}(t) \tag{4.5}$$

where, $a^{i-1}(t)$ is the most recent acceleration update received from preceding vehicle V_{i-1} at or before time t. Similarly, $v_i(t)$, $a_i(t)$ and $\dot{u_i}(t)$ are velocity, acceleration V_i and the change in acceleration, also called *jerk* [63], calculated by the platoon controller at time t. The parameters k_p and k_d are calibrated based on stability criteria given in [42] ($h > 0$, $k_p > 0$, $k_d > 0$ and $k_d > k_p \times \tau$, $]tau$ is the vehicle dynamics time constant). The standstill distance between two vehicles is defined as d_{stop} and the time headway is defined as h seconds. At time t, the platoon controller computes the acceleration to be maintained by the vehicle at time $t+\Gamma$ where Γ is the time between successive CACC operation. Hence, $u_i(t+\Gamma)$ is the target acceleration calculated by CACC based on $\dot{u_i}(t)$. Typically, trucks and commercial vehicles have a limit on the maximum acceleration, $MaxA$ and minimum deceleration, $MinA$. Therefore, the target acceleration is subjected to a bounding limit of $MaxA$ and $MinA$. Hence, the value of $\dot{u_i}(t)$ is recalculated based on the values of $MaxA$ and $MinA$,

$$\dot{u_i}(t) = \begin{cases} \frac{MinA - a_i(t)}{\Gamma}, & \text{if } MinA > u_i(t+\Gamma) \\ \frac{MaxA - a_i(t)}{\Gamma}, & \text{if } MaxA < u_i(t+\Gamma) \end{cases} \tag{4.6}$$

In a stable and safe platoon, messages from a vehicle containing the acceleration are reliably received at the corresponding following vehicle. However, because of drastic network conditions and cases of malicious behaviors, the valid messages available through wireless network can be lost. In our study of the CACC we investigate different mechanisms of logical coupling between the vehicles as discussed below:

- Conventional ACC (ACC): This is the standard ACC coupling mechanism that has been studied and deployed for many years. The sensors such as LiDAR or Radar act as the only source of information returning the distance with respect to the back bumper of the vehicle in front. The ACC controllers use the separation gap and regulate the speed in order to minimize the separation error. In our study, we consider ACC as a baseline system for comparison purposes [61].

- CACC with acceleration updates (CACC): In this strategy either the current acceleration $a_i(t)$ or the target acceleration $u_i(t)$ [14, 42] of a vehicle is appended as a part of the periodic broadcasts. The platoon controller can utilize either of the two versions of acceleration as an input.

- CACC with local updates (CACC_L): This methodology assumes that the distance measuring sensors like LiDAR or Radar can provide, with some level of accuracy and errors, the relative velocity $r_i(t)$ or relative distance $d_i(t)$ between the consecutive vehicles in a platoon. The relative velocity and relative distances can be used to approximate the acceleration of the lead vehicle.

We assume that the acceleration of the vehicles that follow the leader are controlled by the CACC but the leader vehicle is either a completely autonomous vehicle or a human driven vehicle. For this study, we assume that the leader vehicle is stimulated by a predefined acceleration profiles. The acceleration profiles determine the acceleration of the leader vehicle at any time during the simulation. Table 4.1 shows different acceleration profiles used to define the movement of the leader in the platoon. As shown in Table 4.1 type $us6$ is available at [64] as speed profile in ft/s units which are converted to ft/s^2 acceleration profile, type $c2a$ is the acceleration profile recorded in a vehicle driving from Clemson, SC to Anderson, SC in normal day traffic.

Table 4.1: Acceleration profiles considered for leader vehicle

Type	Name	Acceleration profile	Duration
sin	sinusoidal	$5 \times sin(\frac{2\times\pi\times t}{T})$	$300s$
step	constant	$\begin{cases} 2.5 \times t, \text{if } 0 \geq t \leq 1s \\ 2.5, \text{if } 1s \geq t \leq 20s \\ -2.5 \times t, \text{if } 20s \geq t \leq 22s \end{cases}$	$300s$
lin	linear	$\begin{cases} 5 \times t, \text{if } 0 \geq t \leq 25s \\ -5 \times t, \text{if } 25s \geq t \leq 50s \end{cases}$	$300s$
*us*6	us06 [64]	predefined set of acceleration	$600s$
*c*2*a*		data collected from a vehicle driving from Clemson, SC to Anderson, SC	$1200s$

4.3.2 Wireless Communication

Each vehicle in a platoon is capable of communication with other vehicles in the vicinity using wireless broadcasts. We are considering DSRC (802.11p) wireless communication for our study. An ego vehicle is able to receive platooning specific broadcasts $u_i(t)$ or $a_i(t)$ from lead vehicle as well as other vehicles within the DSRC Range [8]. We assume that, for the sake of simplicity, the wireless channel used in DSRC broadcasts have preset modulation, coding, and power configurations. The wireless broadcasts allow any vehicle in a platoon or any roadside compute device to access the platoon behavior. This assertion can be used to determine if any vehicle in the platoon is undergoing malfunction such as large packet loss or malicious behavior such as denial of service attacks, data infusion etc. The vehicular network that operates over the CACC platoon consists of:

- Vehicle to Vehicle (V2V): the exchange of messages among vehicles in the platoon and also from other vehicles on the road. The networking requirements will determine the type of physical layer attributes like channel bandwidth, modulation and coding, the DSRC channel being used, and if the channel is single or dual mode [8].

- Vehicle to Infrastructure (V2I): the exchange of platoon specific information with the infrastructure which is evaluated and used to monitor the platoon. This communication can happen using DSRC between vehicles and road side units or through 3GPP wireless communication standards like 3G/4G/5G. For our study, we consider DSRC based V2I communications.

4.3.3 Local Sensor (Distance and Velocity Measurement)

The local sensor devices that measure relative distance and relative velocity play an important role in our study because of their usage in estimating the acceleration of the lead vehicle. The modeling of these sensors and the estimation methods used in the platooning system is presented in this section. Traditional ACC assumes a sensing device embedded into the vehicle that provides the controller with distance and velocity of the lead vehicle. For ACC, the device used is mostly long range RADAR (ranges from 50 m - 150 m) [65, 66]. For CACC a similar device is assumed to be used as the sensor. LiDAR is another technology making great leaps into autonomous vehicle domain [67]. LiDAR can provide a 360 rendering around the sensor where as RADAR have sharper and smaller beam width and can focus on a particular direction and point in space to measure relative distance and velocity [66, 67]. LiDAR is currently limited to distance measurements only and most of the recent work are limited to object detection where as, for platooning applications objects need to be identified as well. Therefore, LiDAR requires an extra level of sophistication to allow identification or selection of one particular object among others (i.e., the back bumper of the lead vehicle). RADAR on the other hand can be mounted such that they can readily measure the relative distance and velocity of any object in front of a vehicle. In recent years, truck platooning implementations are being commercialized using RADAR sensors in Europe and USA [68]. Figure 4.2 above illustrates a sensor attached to the front of the vehicle that is capable of measuring distance and relative velocity to the vehicle in front. The accuracy of LiDAR and RADAR also vary depending on the environmental factors (eg. fog, rain, or morning vs daylight) when the electromagnetic waves experience scattering and timing jitters to reflection from target surface [69]. The accuracy of LiDAR and RADAR is also discussed in [65–67]. Table 4.2 below lists some common sensors with their accuracy and range.

Sensor Name	**Range**	**Accuracy**
Delphi SmartMicro (RADAR)	64 m - 175 m	0.25m - 0.5m
Velodyne HDL (LiDAR)	120 m	0.02 m

Table 4.2: Common RADAR and LiDAR devices specifications

Low cost LiDAR sensors are becoming available that can provide more contextual awareness. Due to the electronics involved and the weather/environmental factors, the sensors will have error affecting each readings [69]. Given the extreme range of sensing devices that might be considered, it is challenging to define accurate models for all possibilities. Therefore, we consider an inaccurate sensor device and develop a technique to estimate the co-operative information required for CACC platooning application. We assume that the measurement errors from the sensor equipment are a zero-mean additive Gaussian process. We also assume that the sensors are capable of recording distance d_i from the back bumper of the vehicle in front to the front bumper of the ego vehicle and the relative velocity r_i with respect to the ego vehicle. Methods to estimate the acceleration of the lead vehicle in a platoon is studied in [15, 16] but the process is limited to theoretic observations and solutions are provided in Laplace domain. Similarly, techniques presented in [70] assume a method to measure acceleration of the lead vehicle but most of the sensors currently in use are limited to measurements of range (distance) and velocities.

Since the local sensor measurements become corrupted with noise, our approach is to model specific classes of sensors that reflect the basic operating capabilities and present an estimation technique that utilizes noisy sensor data to accurately estimate the acceleration of lead vehicle. The measurements of the sensors are:

$$d_i = x_{i-1} - x_i + \mathcal{N}(\mu_d, \sigma_d) \tag{4.7}$$

$$r_i = v_{i-1} - v_i + \mathcal{N}(\mu_v, \sigma_v) \tag{4.8}$$

where, x_i and v_i are the true position and velocity of the vehicle i, and (μ_d, σ_d) and (μ_v, σ_v) are mean and standard deviation of Gaussian random process added as sensor noises. It is assumed that the sensors independently measure distance and velocity and the noises are added independently to each sensor readings. Here, μ_d and μ_v are assumed to be zero. The following assumption about the sensors are made:

- The sensors have an operating range defined as (R_{min}, R_{max}) and the sensors can produce results under a give stochastic process with certain sample errors.

- The sensor periodically measures new samples and a new sample will overwrite the existing reading. We define the maximum rate at which new samples are available from the sensor as S_{max} defined as number of samples per second. If the controller demands samples faster than S_{max}, the sensor will return the previous measured data. We assume that the sensors are capable of meeting the requirements set forth by platooning applications.

The platoon controller requires, at the least, the knowledge of relative distance to the lead vehicle and also an estimation of lead vehicle's acceleration from the sensor devices in cases when wireless network becomes unreliable. Therefore, we devise two sets of estimation models based on Kalman Filter techniques [71] to estimate the relative distance and the acceleration of the lead vehicle. A Kalman filter is an estimation technique that can be used to recursively estimate the state of a system in a way that minimizes the mean squared error [71]. A Kalman filter has two sub-processes executing recursively in a loop: prediction phase and correction/estimation phase. The prediction phase periodically predicts the state variables where as the correction phase tries to correct the predicted state variable values using the measurements. If x is the state of a system to be estimated, u is control input, z is the measurements done periodically, then a Kalman filter is designed to address the general problem of estimating the state $x_k \in \mathcal{R}^n$ with measurements $z \in \mathcal{R}^m$ of a process defined by the discrete time equations:

$$x_k = \mathbf{A}x_{k-1} + \mathbf{B}u_{k-1} + w_{k-1}$$

$$z_k = \mathbf{H}x_k + v_k$$

The random variables w_k and v_k are assumed independent zero-mean Gaussian random variables that represent process and measurement noise respectively. The covariance matrix of the process noise and measurement noise are **Q** and **R**. In practice the values for **Q** and **R** are determined by trial and error method where the parameters are tuned by repeatedly testing for smaller mean squared error. The $n \times n$ matrix **A** relates the past estimate of the state x_{k-1} to the current state x_k in the absence of noise and control input u_k. Similarly, $n \times 1$ matrix **B** relates the control input u_k to the state x_k. And the $m \times n$ matrix **H** relates the state x_k to the measurement variables z_k. In practice the various matrices **A**, **B** and **H** vary with time but we assume they remain constant for our study.

$\underline{PredictionPhase}$

$$\hat{x}_k^- = \mathbf{A}\hat{x_{k-1}} + \mathbf{B}u_{k-1}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$

where, $\hat{x}_k^-$ is the priori estimate of the state x_k at time step k and $\hat{x}_k$ is the posteriori estimate of state x_k at time step k. $\mathbf{P}_k^-$ is the a priori estimate error covariance and $\mathbf{P}_{k-1}$ is the a posteriori estimate error covariance. The goal of the Kalman filter design process is to find an equation that computes a posteriori estimate $\hat{x}_k$ as a linear combination of an a priori estimate $\hat{x}_k^-$ and a weighted difference between an actual measurement z_k and a prediction $\mathbf{H}\hat{x}_k^-$ [71].

$\underline{Correction Phase}$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R})^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k(z_k - \mathbf{H}\hat{x}_k^-)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$$

where, the $n \times m$ matrix $\mathbf{K}_k$ is the gain that minimizes a priori error covariance $\mathbf{P}_k^-$ at time step k.

4.3.3.1 Estimation of the relative distance

In order to filter the noises added during the measurement of relative distance from the local sensor, an estimation technique was developed based on Kalman filter as shown above. The relative distance ($d_{k,sensor}$) is used as the state of the estimator and difference between the measured relative distance with noise and the last estimate of the relative distance is used as the control input u_k . In this case m=n=1 and the

Kalman filter equations become:

$$\mathbf{A} = \mathbf{B} = \mathbf{H} = [1]$$

$$\mathbf{Q} = 10^{-5}$$

$$\mathbf{R} = 1$$

$$z_k = d_{k,sensor}$$

$$u_k = d_{k,sensor} - x_{k-1}$$

4.3.3.2 Estimation of the acceleration of lead vehicle

Since the piece wise linear acceleration model is used to represent the mobility of each vehicle in the platoon, the same behavior can be imported to design an estimation technique based on sensor measurements. A Kalman filter technique is designed to estimate the acceleration of the vehicle in front using information gathered from the ego vehicle and mounted sensor devices. For this estimation model, we assumed that the sensors are capable of measuring new readings every dt seconds. At time step k, ego vehicle (V_i) is able to make following measurements:

- relative distance to the lead vehicle (V_{i-1}) = $d_i[k]$, where $d_i[k] = x_{i-1}[k] - x_i[k] + \mathcal{N}(0, \sigma_d)$
- relative velocity to the lead vehicle = $r_i[k]$, where $r_i[k] = v_{i-1}[k] - v_i[k] + \mathcal{N}(0, \sigma_v)$
- distance traversed by ego vehicle in dt time = $s_i[k]$
- approximate distance traversed by lead vehicle in dt time= $\hat{s}_{i-1}[k] = s_i[k] + d_i[k] - \hat{s}_{i-1}[k-1]$, where $\hat{s}_{i-1}[k-1]$ is the prior estimated distance traveled by the lead vehicle

- approximate velocity of lead vehicle = $\hat{v}_{i-1}[k] = r_i[k] + v_i[k]$

- approximate acceleration of lead vehicle = $\hat{a}_{i-1}[k] = \dfrac{\hat{v}_{i-1}[k] - v_{i-1}[k-1]}{dt}$, where $v_{i-1}[k-1]$ is the prior estimated velocity of lead vehicle.

- approximate rate of change of lead vehicle's acceleration = $\hat{m}_{i-1}[k]$. For our calculations, we assume $\hat{m}_{i-1}[k]$ remains unchanged between estimates and is fixed at zero.

- hence, the state variable $x[k]$ defined as

$$x[k] = \begin{bmatrix} m_{i-1}[k] \\ s_{i-1}]k] \\ v_{i-1}[k] \\ a_{i-1}[k] \end{bmatrix}$$

At any time step k, the prediction and correction Kalman filter model parameters used for the estimation are as given,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{dt} \\ \frac{dt^3}{6} & 0 & dt & \frac{dt^2}{2} \\ \frac{dt^2}{2} & 0 & 1 & dt \\ dt & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$u = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$z = \begin{bmatrix} \hat{m}_{i-1}[k] \\ \hat{s}_{i-1}[k] \\ \hat{v}_{i-1}[k] \\ \hat{a}_{i-1}[k] \end{bmatrix}$$

Following values of covariance metrics were used in the estimation process.

$$\mathbf{Q} = \begin{bmatrix} 0.1^{-4} & 0 & 0 & 0 \\ 0 & 0.01^{-4} & 0 & 0 \\ 0 & 0 & 0.01^{-4} & 0 \\ 0 & 0 & 0 & 0.01^{-4} \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

4.3.4 Switching Controller Model

In this study, scenarios with large packet loss, sudden stoppage of a vehicle in a platoon, or malicious wireless broadcasts from a vehicle participating in the platoon are considered. It is evident that such scenarios can disturb the stability and safety of the platoon application as well as increase probability of crashes. The traditional CACC platooning model switches to ACC when the communication is invalid on unavailable. Therefore the motivation behind using a switching controller, as shown in Figure 4.3, is to facilitate a smoother and safer adoption between the two different platoon implementations. The switching controller can weigh the information received from distance measuring equipment (LiDAR/Radar) against the data received from wireless broadcasts from Lead vehicle and use the most reliable version of information between the two.

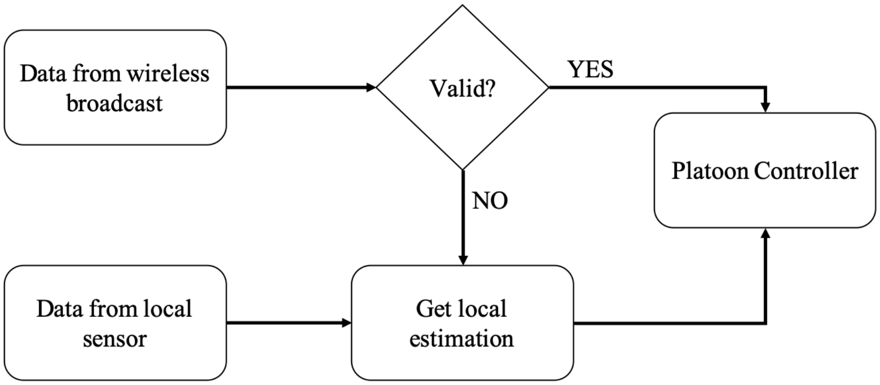


Figure 4.3: Switching Controller

4.3.5 Stability Analysis of platoons

Stability is inherent to a platoon and can be used to differentiate a safe platoon from an unsafe one. The stability analysis is very important for understanding the behavior of platoon because:

- The analysis can provide insights into successive amplification of any mobility parameter (eg., acceleration, jerk, velocity, distance separation error) towards the tail of the platoon,
- If any of the parameters amplify over time and towards the end of the platoon then it becomes unstable,
- Onset of instability can help predict occurrence of a crash in a platoon and take preemptive actions.

In particular, the scenario of most interest is when an intermediate vehicles in the platoon has to brake suddenly or if one of the vehicles in the platoon malfunctions and suddenly applies hard brake. Actions of a malicious vehicle in a platoon that can disrupt the communication (Denial of Service Attacks) or broadcast forged vehicular information are possible reasons for a platoon to swerve from normality. The platoon is considered stable if such scenarios are not amplified towards the tail of the platoon. A platoon is defined to be *string stable* if sudden changes in velocity of any intermediate vehicle is not amplified downstream towards the tail. In terms of distance separation error, the system is also string stable if the distance separation error never expand towards the tail of the platoon. The string stability has been analyzed by works in [16, 37, 42] and is restated here:

$$||\mathcal{T}_i(j\omega)_{H_\infty}|| = max_i \left[\frac{\int_0^W v_i^2(t)}{\int_0^W v_{i-1}^2(t)} \right]^{1/2} \tag{4.9}$$

For string stable platoon,

$$||\mathcal{T}_i(j\omega)_{H_\infty}|| <= 1 \tag{4.10}$$

Equation 4.9 defines stability in terms of energy dissipation towards the back of the platoon. Let $\hat{v}_i(s)$ represent the Laplace transform of $v_i(t)$ and let $\mathcal{T}_i(s)$ represent a transfer function, defined in the Laplace transform domain from the transform of $v_{i-1}(t)$ to the transform of $v_i(t)$.

That is,

$$\hat{v}_i(s) = \mathcal{T}_i(s)\hat{v}_{i-1}(s)$$

or

$$\mathcal{T}_i(s) = \frac{\hat{v}_{i-1}(s)}{\hat{v}_i(s)}$$

It is stated in [42] that the following relationship holds from linear system theory.

$$||\mathcal{T}(j\omega)||_{H_\infty} = max_i \frac{||v_i(t)||_{L_2}}{||v_{i-1}(t)||_{L_2}} \tag{4.11}$$

Here the H_∞ norm of $\mathcal{T}(s)$ taken along the imaginary axis is the supremum of the values of $\mathcal{T}(s)$ in that domain and

$$||v_i(t)||_{L_2} = (\int_0^T v_i(t)^2 dt)^{\frac{1}{2}}$$

The kinetic energy of car i is $E = mv^2/2$. Consequently the L_2 norm of $v_i(t)$ represents a normalized representation of the energy of car i. Instability occurs when energy increases toward the rear of the platoon. Consequently

$$||\mathcal{T}(j\omega)||_{H_\infty} <= 1$$

is used as a sufficient condition for stability. This equation requires initialization of velocity $v(0) = v(W) = 0$ (i.e. the velocity should come back to zero before this equation can be evaluated). That requirement is not feasible for real-time platoon applications because in many cases the velocity may not come back to zero. Trucks platooning on a single lane in a highway is one such example of a realistic scenario for which this metric is not applicable. Therefore, we have developed metrics that report the platoon stability in real time and prevent possible crashes in the near future.

Another version of stability analysis formulation incorporates error propagation along the platoon as shown in equation 4.12. For a stable platoon, at any given time, the maximum absolute value of the distance separation error should decrease towards the tail of the platoon. Any events where that parameter is increasing towards the tail implies the platoon is becoming unstable.

$$max_t\,|e_n(t)| <= max_t\,|e_{n-1}(t)| <= max_t\,|e_{n-2}(t)| \ldots <= max_t\,|e_1(t)| \tag{4.12}$$

Most of the metrics used in the past research that estimate the stability are unable to determine the onset of instability. We propose a metric that can be incorporated in real time to illustrate the onset of instability of a platoon. The idea is intuitively based on the amount of jerk a vehicle is experiencing in real time and preventing extreme oscillations in the jerk to keep the platoon from diverting towards instability. Equation 4.5 in Section 4.3.1 defines the jerk also called the rate of change of acceleration [63]. The quantity $\dot{u}_i(t)$ is calculated by the platoon controller and input to the vehicle control as a target change in acceleration for the vehicle. Therefore, changes in the value of $\dot{u}_i(t)$ can quantify the changes in acceleration of any ego vehicle. We devise a metric based on the behavior of $\dot{u}_i(t)$ and calculate the mean absolute value of $\dot{u}_i(t)$, $\mathcal{M}(t)$ over renewal periods of $\dot{u}_i(t)$. A renewal period is defined as the time span between two consecutive switch in polarity of $\dot{u}_i(t)$ from negative to positive $(-/+)$ i.e. one full cycle of $\dot{u}_i(t)$. The equation for mean absolute jerk is given as:

$$\mathcal{M}(t) = \frac{\int_{t-T_s}^{t} |\dot{u}_i(t)|}{t - T_s} \tag{4.13}$$

where, T_s is the start of a renewal period. We state that a system inclines towards instability if the value of $\mathcal{M}(t)$ is oscillating larger than $MaxA$.

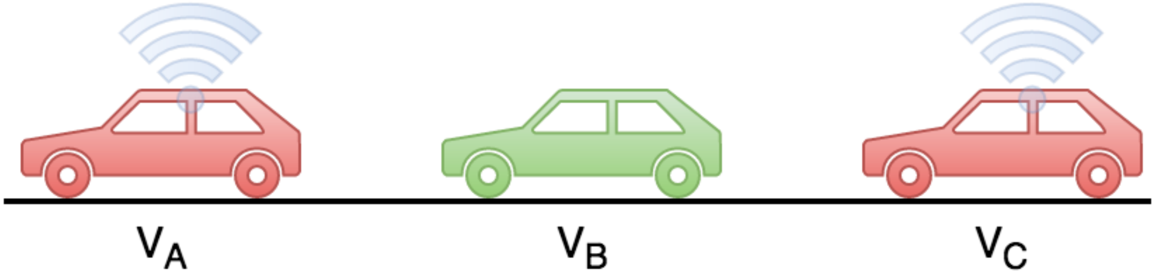


Figure 4.4: Setup for Test III

4.4 Testing CACC application with two vehicles

Section 3.1 described tests conducted in a connected vehicle test-bed. Along with those tests, *Test III* was also conducted where a two vehicle platoon (V_A and V_C) was used to study the loss process in V2V between the vehicles at speeds of 25-35 mph as shown in Figure 4.4. A third vehicle (V_B) was emulated offline using the data collected during the two vehicle test run. In this test, vehicle V_B was CACC capable so it could properly emulate a platoon vehicle following V_A. The modulation and coding used in the study was rate 3/4 QAM-64. On Board Units were installed on the roof of the two vehicles, and they were driven along the pathway as shown in Figure 4.5 .

The test roadway is located on the campus of Clemson University. It provides varying signal conditions from good signal with line of sight (LOS) to non line of sight (NLOS). The results were than used in a CACC emulation application to explore the impact of network impairments on a hypothetical vehicle operating in the platoon.

As described earlier, only vehicles V_A and V_C were equipped with DSRC Radios. The presence of emulated vehicle (V_B) in the middle provided a mechanism to test with different distance between V_A and V_B. Vehicle V_C obtained distance from vehicle V_A through GPS position sent from V_A but it could also be done with a Lidar

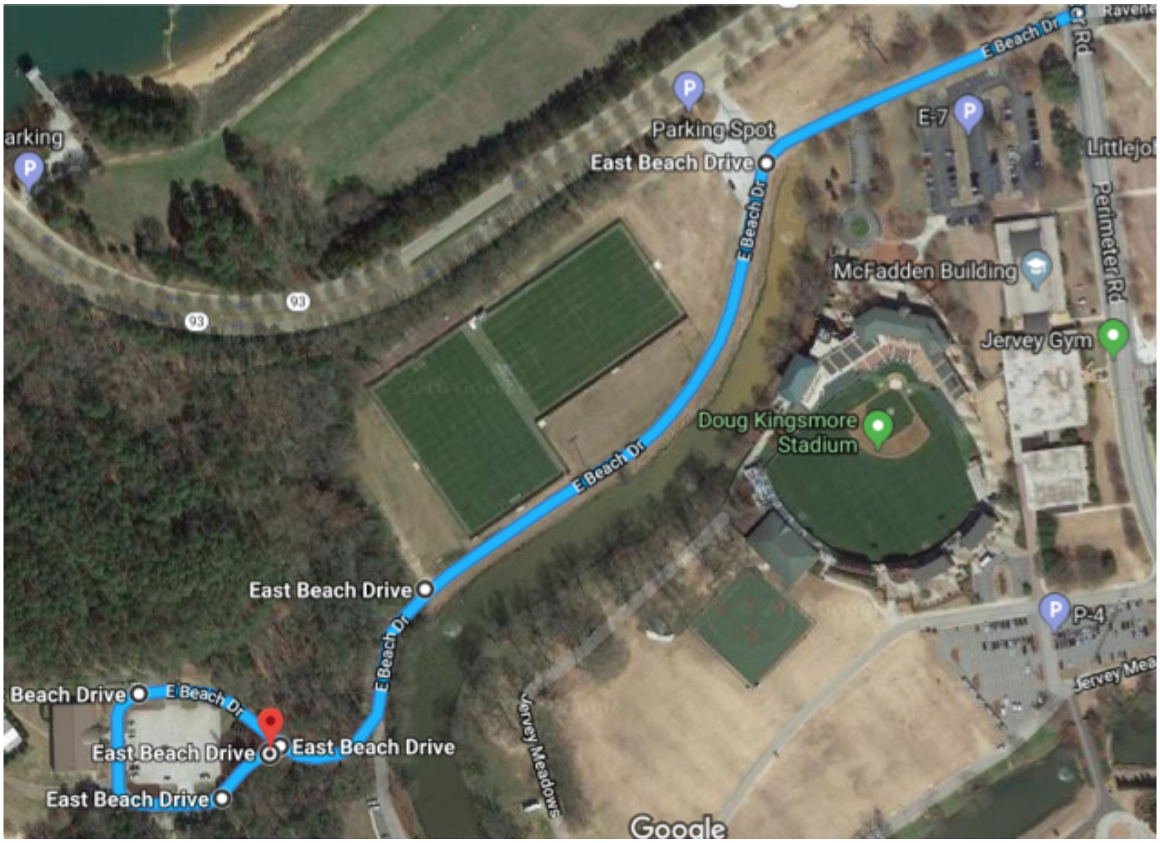


Figure 4.5: Driving track used for Test III

or a Radar. The test roadway selected contained a curved section with a large number of tress resulting in a NLOS zone and a clear road section with good LOS. The objectives of these tests were to study the loss process associated with two moving DSRC radios and also to understand the effect of the loss process on a modular CACC application.

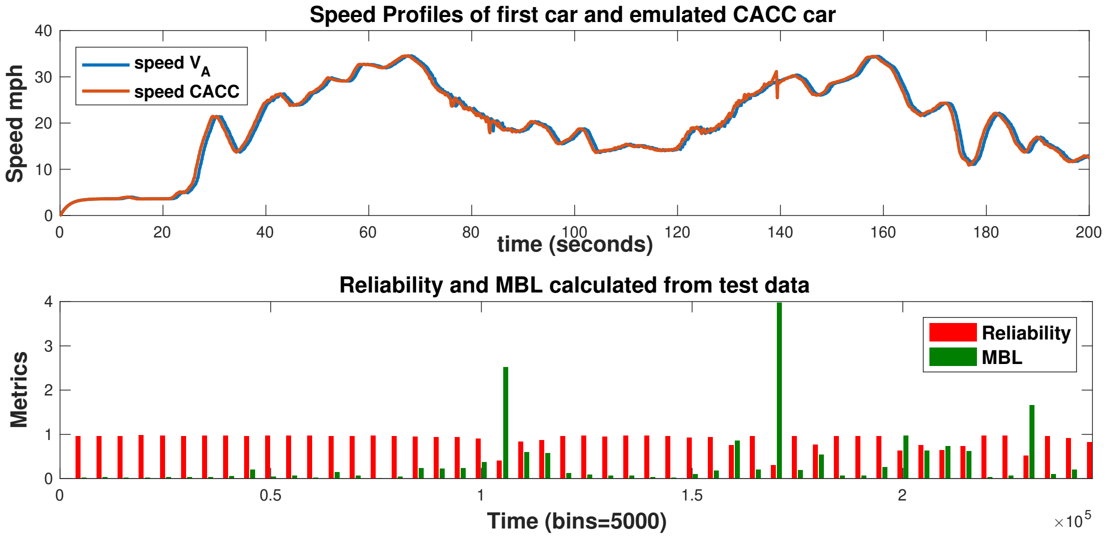


Figure 4.6: Test III observation for Rate 3/4 QAM 64 modulation (a) Speed Profile (b) Metrics based on timed bins

We created a modular CACC application based on [14] that can be used offline and in situations where autonomous or semi autonomous vehicles are not available. The emulation settings used were K_p=0.2,K_d=0.7 and h=1 seconds as described in [37]. By recording the speed and distance between two vehicles using GPS coordinates and feeding that information in the CACC emulation we can model the output from the controller with and without the presence of noise. The input to emulated CACC vehicle V_B was embedded with the observed loss process. Figure 4.6 above shows the results obtained from Test III with Rate 3/4, QAM-64 modulation and coding scheme. Part (a) shows the speed profile of V_A and calculated speed profile when the velocity and acceleration from V_A was used as input to emulated CACC controller in V_B. The resulting speed profile shows a rapid fluctuation in speed which occurs relatively close to the high burst loss event indicated by large MBL and smaller reliability. From these tests, the effect of network conditions on velocity of a platooning vehicle can be observed. One limitation of this approach is that the tests were limited to two vehicle platoons. To answer questions related to longer platoons and burst packet loss due to congestion and malicious attacks, simulations were carried out and they are described in the next chapter.

Chapter 5

Simulation of CACC

The simulation of CACC and results supporting this research are presented in this chapter. The results presented showcase implications of wireless network reliability on a CACC controller and applicability of the fall back controller. Simulations were conducted in *ns-3*, a network simulation software [58] and Matlab©. Over the years ns-3 has been catering mostly to the networking and communications research community. However, for our study it became a necessity to model a platooning application along with the overlaying wireless communication for cooperative message broadcasts. Therefore, the simulation framework developed in this study is used for investigating the effect of channel congestion (heavy traffic) as well as burst packet loss due to denial of service attacks on a platoon of vehicles. The choice of ns-3 was based on it being an actively developed state of the art simulation environment for advanced packet radio communication technologies as well as its object oriented framework and ease of implementing the vehicular mobility models.

5.1 CACC Simulation Components

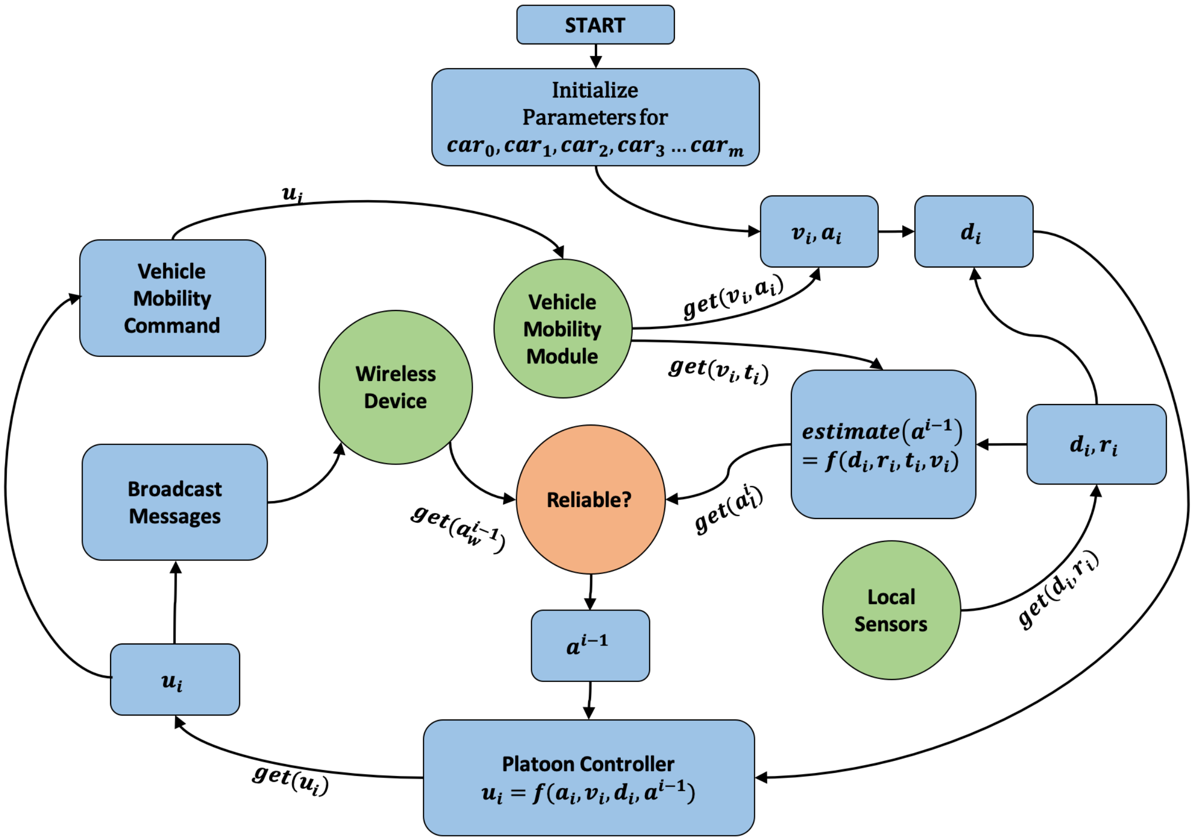
The simulation framework consisted of following main components:

- Platoon controller
- Vehicle kinetics
- Leader vehicle mobility
- Communication component
- Local sensors and estimation
- Simulating burst packet loss

The flowchart in Figure 5.1 shows the major components and their interactions during the simulation of a mobility application like CACC in ns-3 or Matlab. The overall simulation framework is divided into components that emulate the vehicle motion, broadcast messages, and control of the platoon. In the following section the components are discussed in brief and the simulation methodology is presented.

The main components of the simulation are as follows:

- *Platoon Controller* - This module operates independently at rate f_{pc} Hz ($f_{pc} = \frac{1}{T_{pc}}$) and at each event it calculates the target acceleration u_i to be achieved at the end of next T_{pc} seconds and the rate of change of acceleration $\dot{u}_i(t)$ based on Equations 4.5 and 4.6. The acceleration of the lead vehicle is either received from the wireless interface a_w^{i-1} or estimated from the local sensor a_l^{i-1}. The reliability module in Figure 5.1 determines the validity of data coming from wireless interface and is able to inject sensor estimation data where ever there are packet losses.



$\mathbf{v_i} =$ velocity of $\mathrm{car_i}$

$\mathbf{a_i} =$ acceleration of $\mathrm{car_i}$

$\mathbf{d_i} =$ distance between $\mathrm{car_i}$ and $\mathrm{car_{i-1}}$

$\mathbf{t_i} =$ distance travelled by $\mathrm{car_i}$ in time $\mathrm{T_{dme}}$

$\mathbf{r_i} =$ relative velocity of $\mathrm{car_i}$ w. r. t $\mathrm{car_{i-1}}$

$\mathbf{a_l^i} =$ locally estimated acceleration of $\mathrm{car_{i-1}}$

$\mathbf{a_w^{i-1}} =$ acceleration of $\mathrm{car_{i-1}}$ received by wireless

$\mathbf{a^{i-1}} =$ acceleration of $\mathrm{car_{i-1}}$

$\mathbf{u_i} =$ target acceleration of $\mathrm{car_i}$

Figure 5.1: Flow chart for simulation of CACC

- *Vehicle kinetics* - Vehicles in a platoon move in a continuous fashion. This component is used to calculate the position, speed, and velocity of any vehicle in a platoon during the simulation. For simplicity the vehicles are designed to move in a horizontal (x-axis) direction only. The acceleration is assumed to be piece-wise linear based on Equations 4.1 and 4.2. During the simulation, vehicle mobility module calculates the kinematics every $T_{vm}=\frac{1}{f_{vm}}$. Once the distance is calculated, the $Mobility\ Module$ of ns-3 reallocates the position of the node (vehicle) - hence moving the vehicle.

- *Leader vehicle Mobility* - The leader vehicle (V_0) mobility component calculates the acceleration of the first vehicle and is based on the predefined acceleration profile as shown in Table 4.1. Figure 5.3 below shows the various acceleration profiles used in modeling the kinetics of V_0. The acceleration profiles are designed such that the maximum velocity and acceleration of any vehicles following these profile were realistic. In this case, the velocity based on the profiles given in Table 4.1 was below 85 *mph*.

- *Communication Module* - As described in earlier chapters, the wireless communication module is used for broadcasting vehicle status messages. For the simulation purposes we assume that the vehicles are capable of 802.11p communication. The messages are transmitted in IP protocol with the link and physical layer implemented in 802.11p. The *YansWifiPhyHelper* and *Wifi80211pHelper* available in ns-3 were used to model the wireless radio interface in each vehicle. The broadcast rate (f_{cm}) is categorized to vary from the standard BSM rate of 10 Hz to 40 Hz.

- *Local Sensors* - The local sensors measure the distance and the relative velocity to the back bumper of the lead vehicle. As discussed in Section 4.3.3, the sensor noises are modeled as Gaussian random processes. The detailed working of the sensor mechanism, noise considerations, and the estimation techniques are presented in Section 4.3.3. The sensor module is capable of undertaking measurements at $T_{ls} = \frac{1}{f_{ls}}$ seconds.

- *Simulating burst packet loss* - In order to add the burst loss to the wireless networks, a two state Markov model was used to emulate burst loss process. The simulation operated in two distinct phases of wireless behavior. *Loss Less Mode* where the burst loss was absent and *Loss Mode* where a predefined loss process acted on the wireless network. The two state Markov model during the Loss Mode is shown in Figure 5.2. In Figure 5.2, state 1 is the no packet loss state and state 0 is the packet loss state (i.e., packets are lost when the system is in state 0 and received correctly when it is in state 1). As long as the simulation exists in Loss Less Mode, there are no burst packet losses. When the simulation enters the Loss Mode, than a predefined burst packet loss process acts on the packet communication network, causing application to drop packets in simulation. This allows for investigation of performance as the platoon travels through regions where high volumes of data are being affected by either congested channel conditions or malicious wireless device behavior. The probabilities shown in Figure 5.2 are defined as p, the probability of leaving a no-packet-loss state and q, the probability of leaving the packet-loss state. Given the average length of loss burst b_L and average length of good burst g_L:

$$p = \frac{1}{g_L} \tag{5.1}$$

$$q = \frac{1}{b_L} \tag{5.2}$$

In order to test the system under burst loss and to generate the different loss conditions, following set of average length (i.e., number of consecutive packets) of good and loss bursts were used as shown in Table 5.1. If p_r is the packet rate considered for the application then the average length of burst loss time, $Loss_{period} = p_r \times b_L$ seconds.

Table 5.1: Simulation Parameters: Burst loss process

Loss Mode	b_L	g_L
No Loss	1	100000
Complete Outage	100000	1
Congested	30	1
Malicious	150	1
On-Off	50	50
Long Burst	1000	300

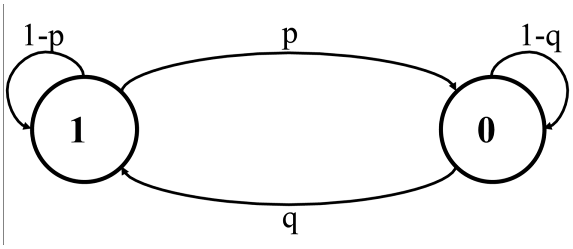


Figure 5.2: Two state Markov burst loss Model (state 1 corresponds to no packet loss and state 0 to burst packet losses

- *Reliability Module* The reliability of wireless network that comes under the affect of different channel conditions and malicious attacks is calculated by the reliability module. Based on the transmit rate, if the wireless network is capable of delivering at most N_p packets over the duration of T_p seconds, then the reliability, r_i measures the ratio of number of packets received every T_p period to N_p. We assume that the broadcast rate for a typical CACC platoon will be approximately 40 packets per second with 200 bytes of messages per

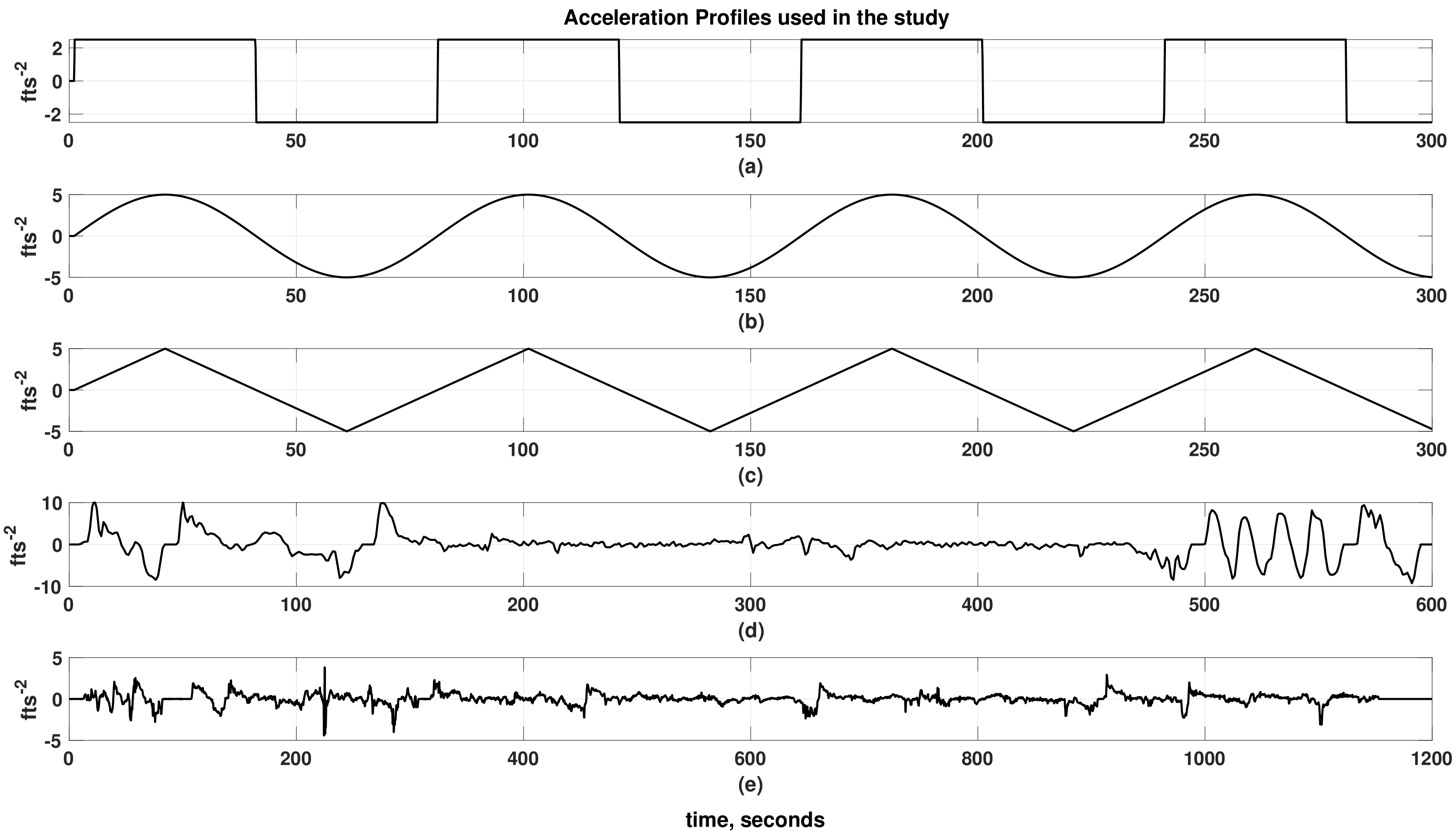


Figure 5.3: Acceleration profiles used for simulations: (a) Step, (b) Sinusoidal, (c) Linear, (d) US06 and (e) c2a

packet. Therefore, $N_p = 40$ packets and $T_p = 1$ second. Since the reliability is calculated in real time every T_p seconds:

$$\begin{aligned} r_{i,s}(t) &= \left[\frac{N_{received}}{N_p}\right]_{t=nT_p} \\ r_{i,a}(t) &= (1-\alpha_r) \times r_{i,a}(t-\tau) + \alpha_r \times r_{i,s}(t) \end{aligned} \tag{5.3}$$

where α_r is chosen to give larger weights to newer values of $r_{i,s}(t)$, typically α_r = 0.8. $r_{i,a}$ gives a moving weighted average of reliability metric updated in real time.

5.2 Simulation Results and Analysis

In this section simulation results are presented and analyzed. Following parameters are defined for each simulation -

- **N**: number of vehicles participating in a platoon
- σ_d and σ_v: standard deviation of zero mean Gaussian noise added to distance and relative velocity measurement devices respectively
- V_0[ACCLN]: Property of stimulating acceleration that drives V_0
- *maxA, minA*: maximum acceleration and minimum deceleration allowed for any vehicle respectively. This research assumes vehicles have similar properties so *maxA* and *minA* is same in all participating vehicles.
- **h**: headway time in seconds
- b_L, b_G: mean burst packet loss length and mean burst good length respectively.

In addition, this research considers following set of results to showcase the difference in performance based of different values of parameters given in Table 5.2. The results include following parameters of interest -

- $\dot{u}_i(t)$ = rate of change in target acceleration.
- $d_i(t)$ = distance separation from front bumper of V_i to the back bumper of V_{i-1} in a platoon
- $C(t)$ = time of first crash event in a platoon. A crash is said to have occurred with vehicle V_i crashing into the back bumper of vehicle V_{i-1} if $d_i(t) < 0$. The simulation stops after the occurrence of a crash.
- $||\mathcal{T}_i(j\omega)||_{H_\infty}$ = string stability analysis
- $\mathcal{F}(t)$ = Flow calculated as the number of vehicles passing over a point on the road per unit second. Mathematically, it is calculated as the ratio of number of vehicles (N) and time between vehicles V_0 and V_{N-1} passing the same point x in the road ($T_{0,N-1}$).

$$\mathcal{F}(t) = \frac{N}{T_{0,N-1}} \tag{5.4}$$

$$\tag{5.5}$$

5.2.1 Traditional ACC

It is necessary to study the behavior of traditional ACC systems and use that as a basis for other results presented here. Many studies in the past tend to consider traditional ACC as a fallback mode (i.e., to establish the platoon controller behavior

Parameter	**Values**
Fallback Mode	ACC or CACC$_l$
h	0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.5, 2.0
N	10
d$_{ref}$	10 ft
k$_p, k_d$	0.66, 0.7
Mean Loss Burst Length(b$_l$)	see Table 5.1
Mean Good Burst Length (g$_l$)	see Table 5.1
σ_d	0.2
σ_v	0.12
Total Simulation time	300 or 600 or 1200 seconds
f$_{pc}$	20 Hz
f$_{vm}$	100 Hz
f$_{cm}$	10 Hz or 40 Hz
f$_{ls}$	50 Hz

Table 5.2: Simulation parameters settings

based on traditional ACC models in cases where the wireless network becomes unreliable or absent). It may be feasible to consider ACC in scenarios where the packet losses are uncorrelated and delays are finite, but the system tends to fail when the packets are lost in bursts and the delays are not bounded (i.e., the packet is discarded by the MAC layer because of longer queue delays due to the channel being heavily contested making probability of transmission negligible). We show here that the traditional ACC may not be the best mode for fallback.

Figure 5.4 shows the time of first crash in the platoon using traditional ACC controller with different values of headway time and different leader vehicle acceleration profiles. For smoother acceleration profiles such as sinusoidal, the chances of crash are considerably lower, however, such acceleration profiles are unrealistic. It can be seen in Figure 5.4 the time of crash increases with the time headway until it eventually becomes crash free. It can also be noted from Figure 5.4 that the crash free region for all types of acceleration profiles considered in this study starts at $h \geq 2s$. Therefore, it would be risky to push the time headway below that value in traditional

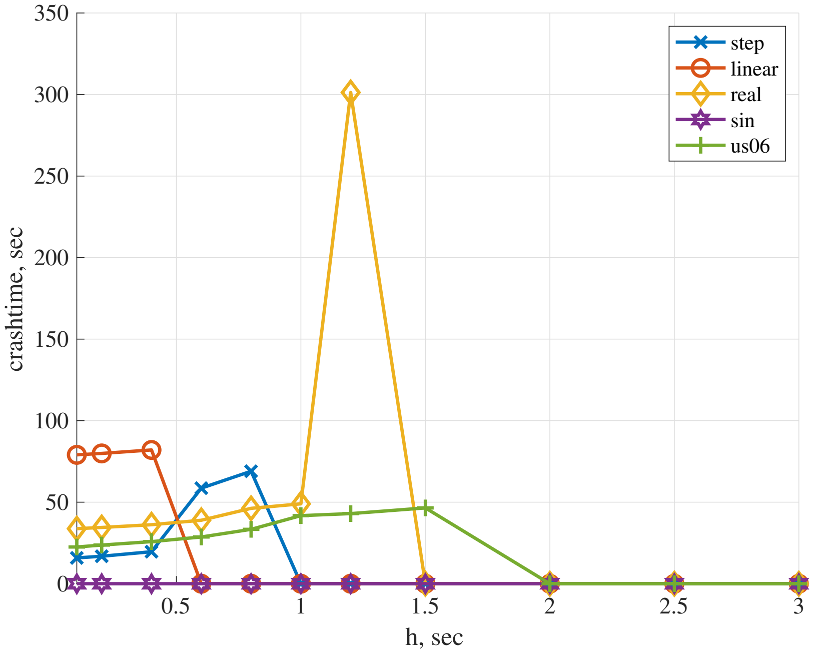


Figure 5.4: Crash time for traditional ACC based controller

ACC based controller.

Figure 5.5 illustrates the max $||\mathcal{T}_i(j\omega)||_{H_\infty}$ $\forall i = 1, 2, 3 \ldots N - 1$ for different acceleration profiles. From Equation 4.9, for a string stable platoon,

$$||\mathcal{T}_i(j\omega)||_{H_\infty} \leq 1.0$$

therefore,

$$\max_i ||\mathcal{T}_i(j\omega)||_{H_\infty} \leq 1.0$$

for all vehicles, $i = 1, 2, 3 \ldots N - 1$ is a valid metric to understand if any vehicle surpasses the stability limit given in equation 4.9. The stability metric $||\mathcal{T}(j\omega)||_{H_\infty}$ maximized over all vehicles shows that for ACC controller operation the system reaches stability condition only for $h \geq 2.5s$ in all acceleration profiles considered. Based on these observations, the best value of time headway, h, suited for stable and safe ACC

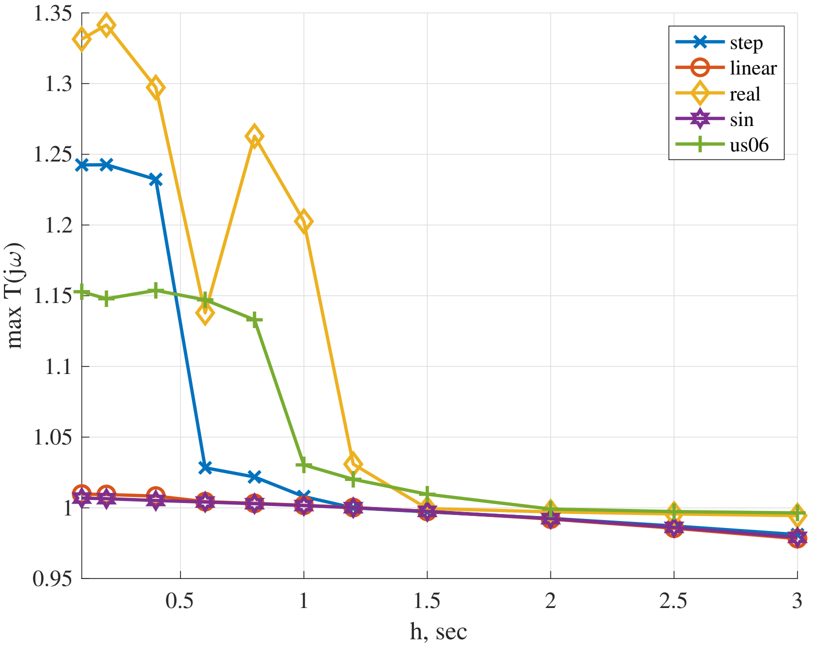
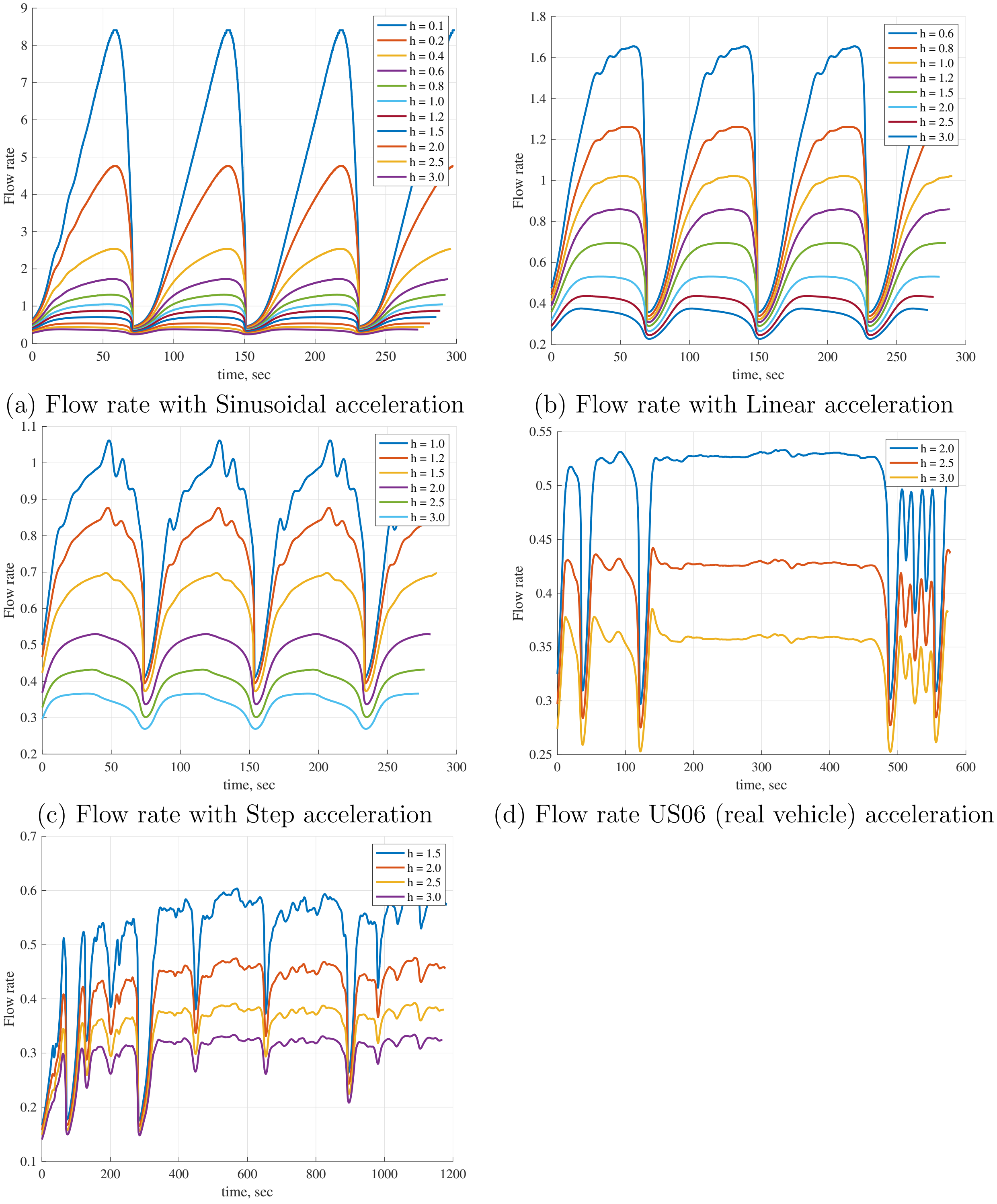


Figure 5.5: max $||\mathcal{T}(j\omega)||_{H_\infty}$ for traditional ACC controller

controller operation is $h \geq 2.5s$.

Figures 5.6 (a-e) illustrate the flow rate observed during the simulation of ACC controller in operation. The flow rates are only shown for those platoons which were found to be crash free. The figures show that the flow rate increases with smaller headway but from the observations made earlier, the system cannot operate for smaller values of h. It can be seen that for a stable ACC platoon the maximum flow rates that can be achieved for different acceleration profiles is $\mathcal{F}(t) < 0.4$ vehicles per second, which corresponds to $h \geq$ 2.5s. We can conclude that the traditional ACC is not suited as a fall back mode for CACC platoons because of larger headway requirements and poor flow rate.



(a) Flow rate with Sinusoidal acceleration

(b) Flow rate with Linear acceleration

(c) Flow rate with Step acceleration

(d) Flow rate US06 (real vehicle) acceleration

(e) Flow rate with C2A (real) acceleration

Figure 5.6: Flow rates for different acceleration profiles with traditional ACC controller

5.2.2 CACC$_l$ fallback

In this section we present the results for CACC$_l$, the local sensor estimation of lead vehicle acceleration, as the fall back in case of unreliable wireless networks. In this mode the Kalman filter based estimation technique is used to substitute the required acceleration information in case the packets are lost in wireless broadcasts. The reliability module shown in Figure 5.1 tracks when a packet was not received as expected and substitutes with locally estimated copy of lead vehicle acceleration. The affect of different types of packet loss in the system as shown in Table 5.1 is analyzed in this section.

5.2.2.1 No Packet Loss

In the scenario where no packets are lost in the process, the system behaves as a properly operating CACC platoon. This section provides insights into the different artifacts of the system and how they behave when everything is considered to be working well. Figure 5.7 illustrates the $||\mathcal{T}_i(j\omega)||_{H_\infty}$ maximized over all the vehicles in a platoon. As can be seen, the string stability criteria is safely satisfied for all vehicles with time headway $h >= 0.4s$. Therefore, the system can be considered to be string stable for much smaller headway time than traditional ACC platoon controllers. We found no crashes in this scenario. This guarantees that, under no packet loss, the platoons can have smaller value of h. Figure 5.8 (a-e) shows the flow rate observed with this scenario. As expected smaller values of h deliver larger number of vehicles per second. It can be stated that the platoon becomes a lot safer and stable with improved flow rate when used with the estimated acceleration fall back compared to traditional ACC fallback. The increase in flow rate going from traditional ACC to CACC with no packet loss is nearly eight to ten folds.

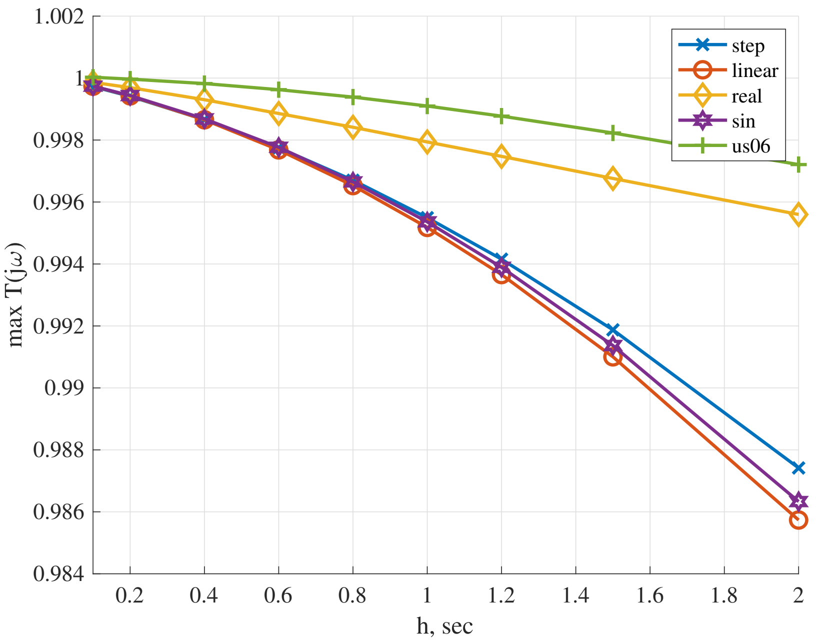
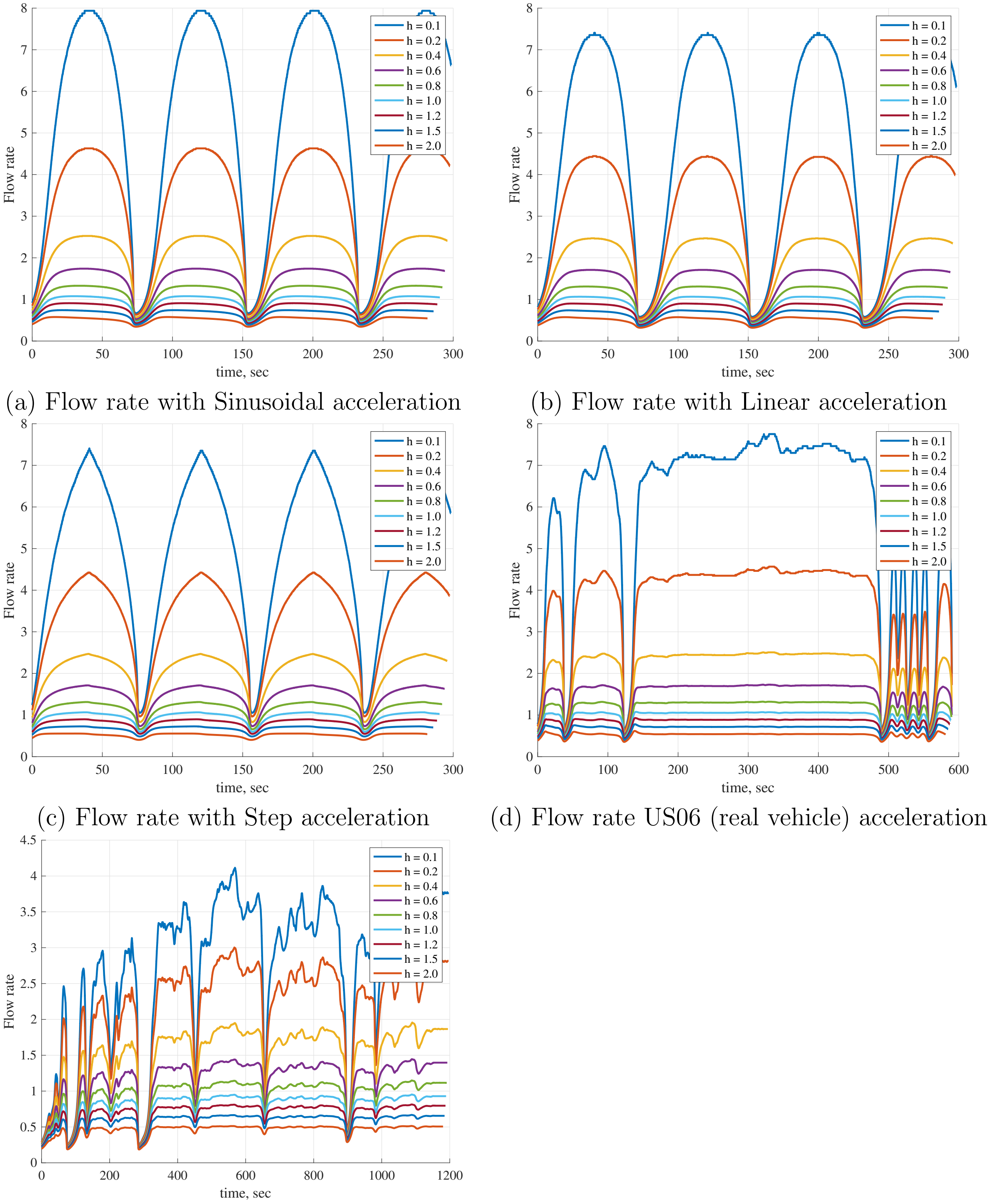


Figure 5.7: max $||\mathcal{T}(j\omega)||_{H_\infty}$ for CACC with NO packet loss scenario

5.2.2.2 Complete Outage

Situations where the wireless network fails completely is considered in this section. Reasons for such failures could be a malfunctioning radio hardware, or a jammer system working in the same channel used by the platoon for broadcasts. In this case, the simulation works without wireless networks therefore the estimation of acceleration of lead vehicle using relative distance and relative velocity measurements become more relevant. This study allows us to understand the efficacy of estimated acceleration input to the platoon controller.

Figures 5.9 and 5.10 show the time of crash and stability criteria for different values of h. The system is not stable below $h = 1.5s$ which is still smaller compared to the results of traditional ACC (Figure 5.5). This validates, within the basis of the acceleration profiles used, the improvement to traditional ACC controller by implementing the local estimation of lead vehicles acceleration. In this scenario, the



(a) Flow rate with Sinusoidal acceleration

(b) Flow rate with Linear acceleration

(c) Flow rate with Step acceleration

(d) Flow rate US06 (real vehicle) acceleration

(e) Flow rate with C2A (real) acceleration

Figure 5.8: Flow rates for different acceleration profiles with NO packet loss scenario

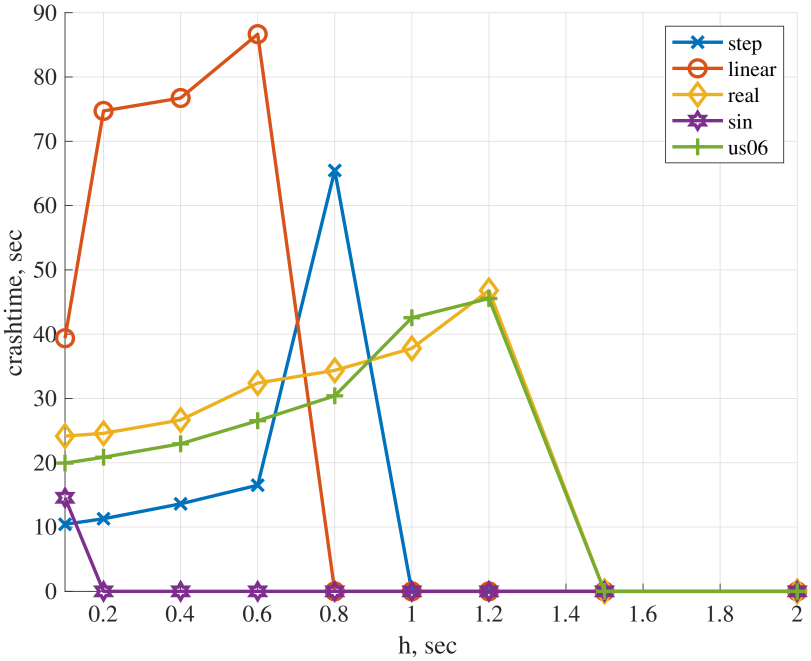


Figure 5.9: Crash time for complete outage scenario

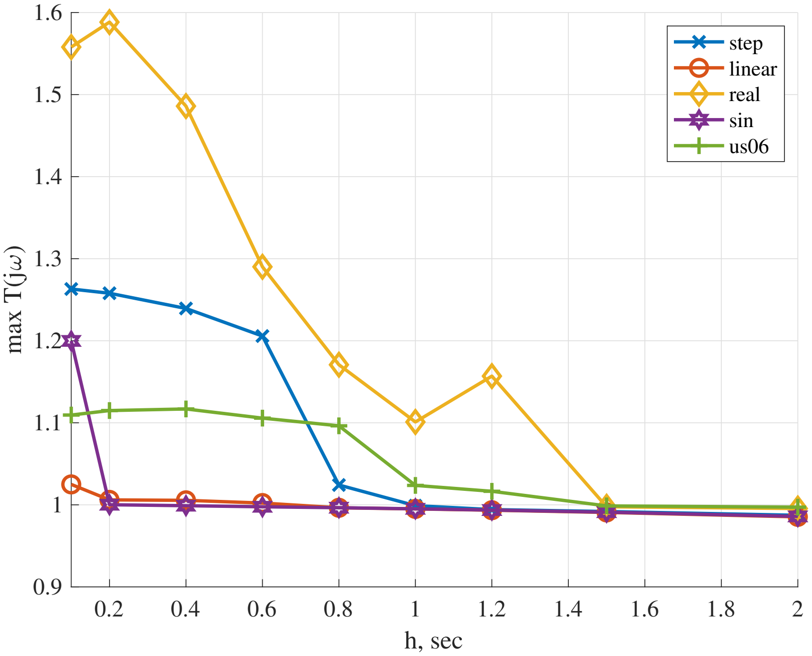


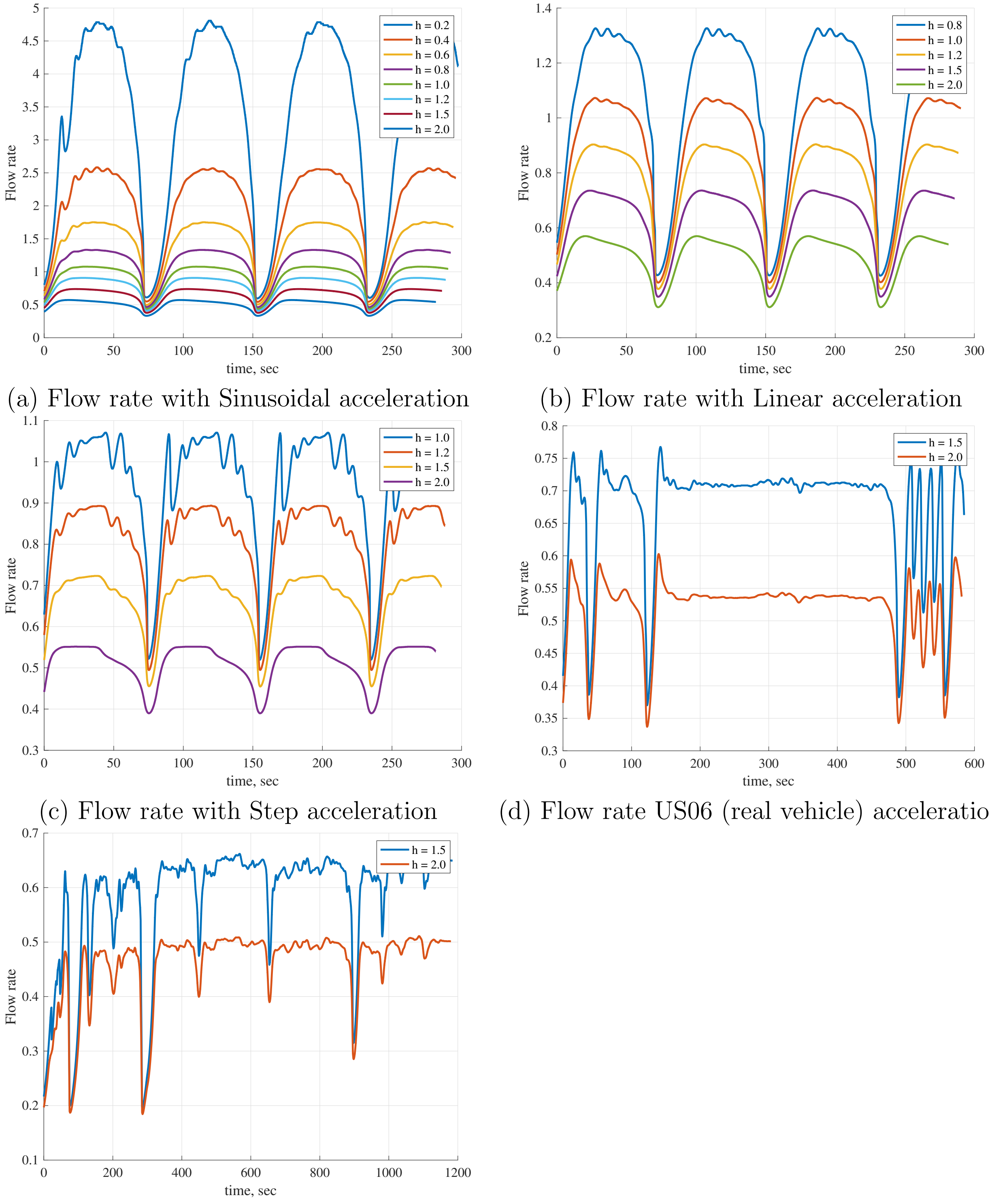
Figure 5.10: max $||\mathcal{T}(j\omega)||_{H_\infty}$ for CACC with complete outage scenario

wireless network never becomes a part of the controller and therefore, the platoon operates by using locally estimated values. Figure 5.11 shows the flow rates observed for the different acceleration profiles in cases where there were no crashes recorded. As can be seen in Figure 5.9 and Figure 5.10 that for realistic acceleration profiles, time headway cannot be lowered below $h = 1.5s$. These results show that if there is no communication and the local estimated acceleration is used as fall back, it performs better than traditional ACC. The minimum headway for safe and stable platoon is $h = 1.5s$ and corresponding flow rate is 0.7 vehicles per second.

5.2.2.3 Congested

We simulated congested scenario by embedding impairments to the wireless networks in a platoon application similar to observed in Figure 3.12. The good and bad burst lengths are given in Table 5.1 and the affect of such loss process on a platooning application is presented here. The scenario is tested for rate 1/2 QPSK modulation and coding scheme.

Figures 5.12 and 5.13 illustrate the crash time and stability criteria for the platoon at different time headway. When the platoon passes through a congested area with relatively longer loss burst of packet and very short good bursts, the platoon becomes more susceptible to instability. Whenever there is a packet lost, the reliability module substitutes the estimated acceleration obtained from the local sensors. As seen on the figures, the good value of h that avoids crashes throughout the simulation is 1.5s. Similarly, Figure 5.14 shows the flow rate observed for crash free cases in congested scenario. As expected the best case scenario is when time headway $h >= 1.5s$. Therefore in cases where the congestion affects the platoon behavior, the safe headway corresponds to an increase flow rate compared to traditional ACC.



(a) Flow rate with Sinusoidal acceleration

(b) Flow rate with Linear acceleration

(c) Flow rate with Step acceleration

(d) Flow rate US06 (real vehicle) acceleration

(e) Flow rate with C2A (real) acceleration

Figure 5.11: Flow rates for different acceleration profiles in complete outage scenario

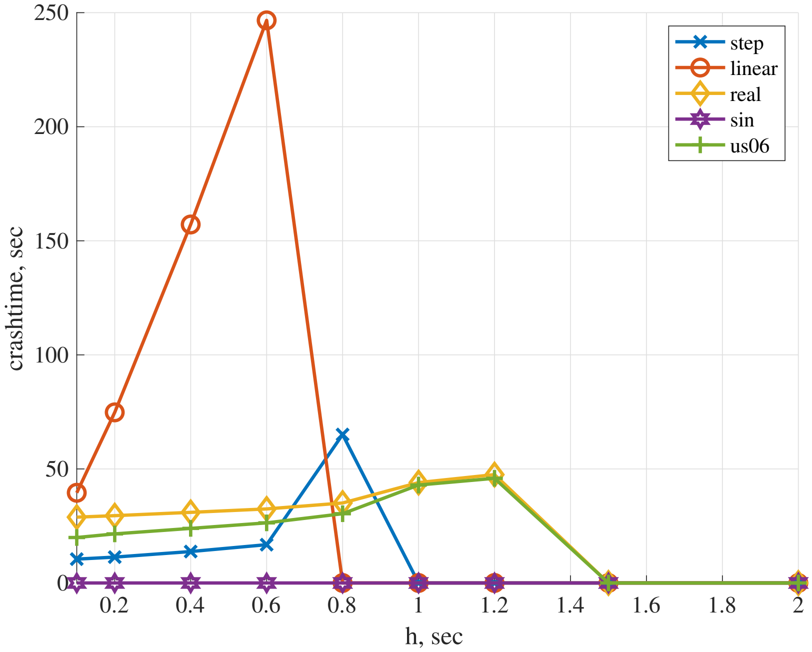


Figure 5.12: Crash time for congested scenario

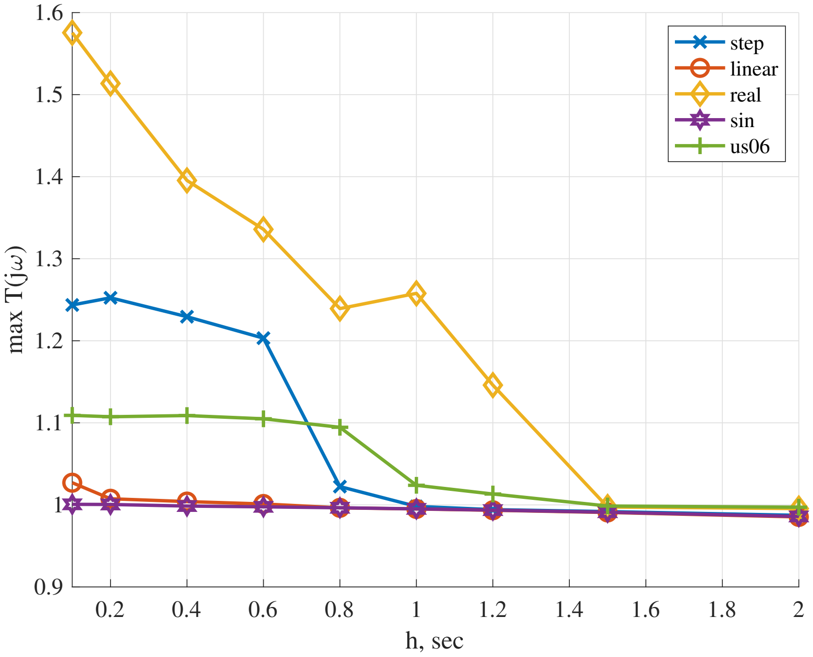
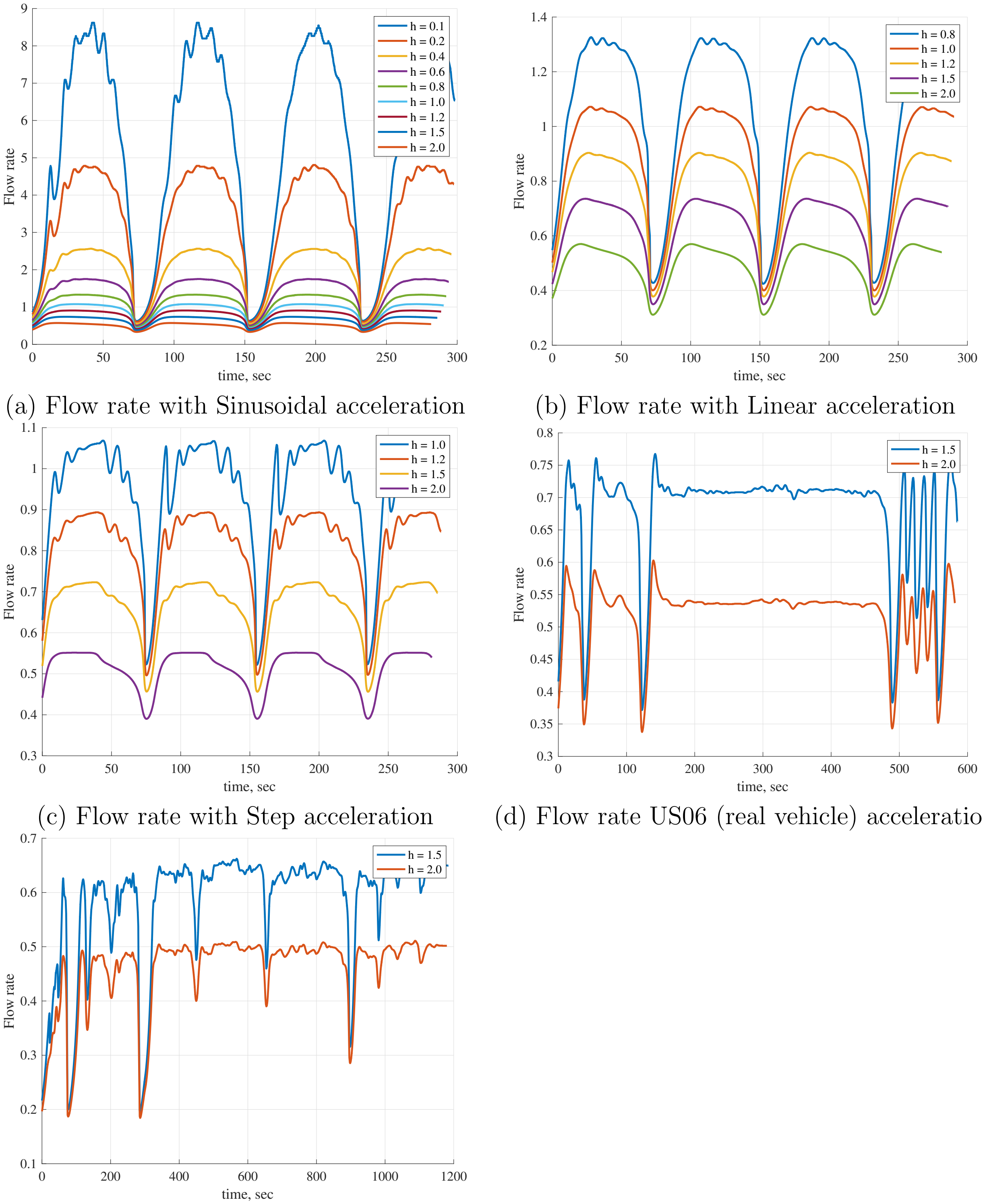


Figure 5.13: max $||\mathcal{T}(j\omega)||_{H_\infty}$ for CACC in a congested scenario



(a) Flow rate with Sinusoidal acceleration

(b) Flow rate with Linear acceleration

(c) Flow rate with Step acceleration

(d) Flow rate US06 (real vehicle) acceleration

(e) Flow rate with C2A (real) acceleration

Figure 5.14: Flow rates for different acceleration profiles with congested scenario

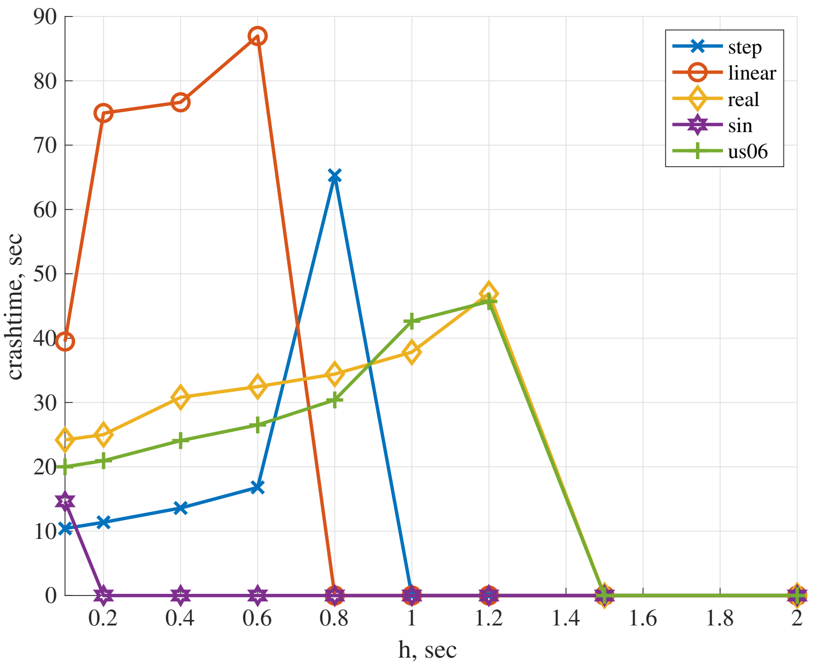


Figure 5.15: Crash time for malicious scenario

5.2.2.4 Malicious

In this cases we consider a node in the platoon or an infrastructure by the roadside that acts maliciously by creating a Denial of Service attack on the channel that is being used by the platooning vehicles. The affect of such an attack on all the message broadcasts in the DSRC channel is studied in this scenario by using the malicious scenario for rate 1/2 QPSK modulation and coding scheme obtained from earlier ns-3 tests.

As seen in Figure 5.15, the platoon remains crash free at $h >= 1.5s$. Also Figure 5.16 shows the stability is valid after $h >= 1.5s$ as well. It can be observed that as the value of h increases, the platoon becomes more robust. This result is better than the traditional ACC model if we compare the maximum flow rate achieved in the two cases where it increases from 0.4 to 0.6 vehicles per second. Figure 5.17 shows the flow rate observed for crash free cases in the malicious node scenario.

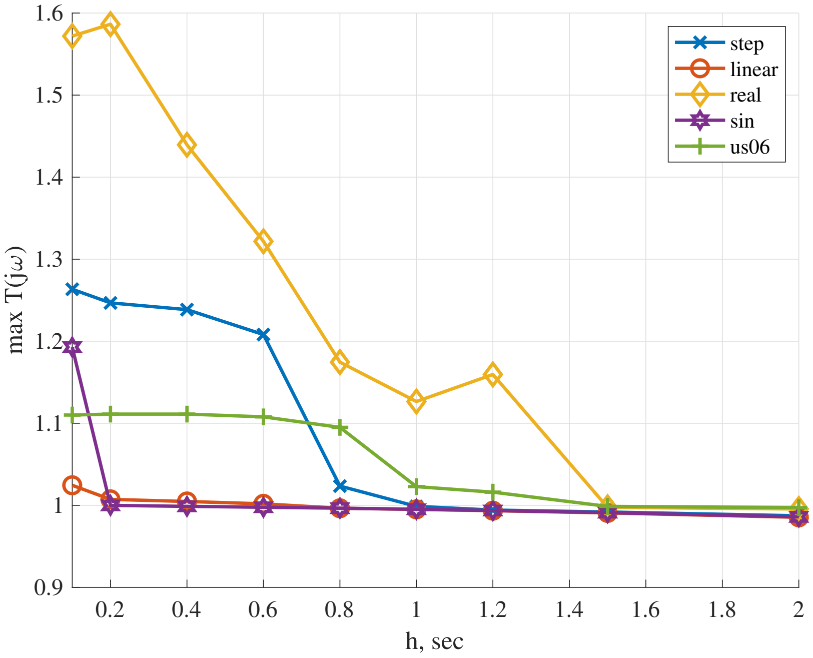


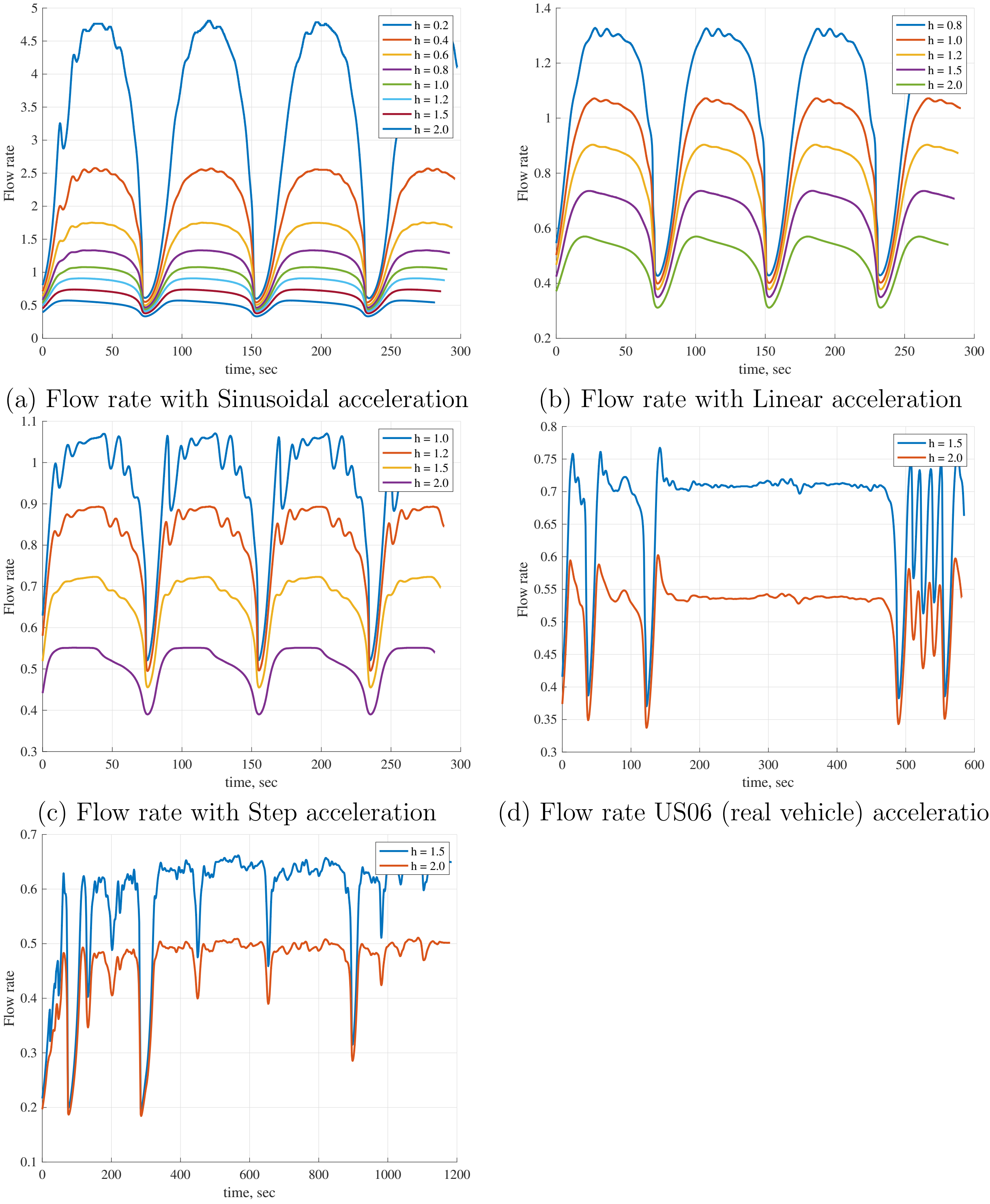
Figure 5.16: max $||\mathcal{T}(j\omega)||_{H_\infty}$ for CACC in malicious scenario

5.2.2.5 On-Off

An on and off like packet loss burst is simulated in this case. It is assumed that the length of good packet burst is equal to the bad packet burst length (i.e., $b_L = g_L$). The length of the burst used in this scenario is given in Table 5.1. It can be seen in Figure 5.18 that the most favorable time headway h is 1s. The acceleration profiles used for the leader vehicle show that the platoon is not stable below $h \leq 1s$. Figure 5.20 shows the observed flows in crash free cases of the On-Off scenario.

5.2.2.6 Long Burst

Another simulated scenario concerning longer burst length with burst lengths $b_l = 1000, g_L = 300$ packets. As seen in Figure 5.21, the platoon experience no crashes for $h \geq 1.5s$ and the stability criteria is also satisfied for the same value of time headway. Similarly, Figure 5.23 illustrates the flows observed for different



(a) Flow rate with Sinusoidal acceleration

(b) Flow rate with Linear acceleration

(c) Flow rate with Step acceleration

(d) Flow rate US06 (real vehicle) acceleration

(e) Flow rate with C2A (real) acceleration

Figure 5.17: Flow rates for different acceleration profiles with malicious scenario

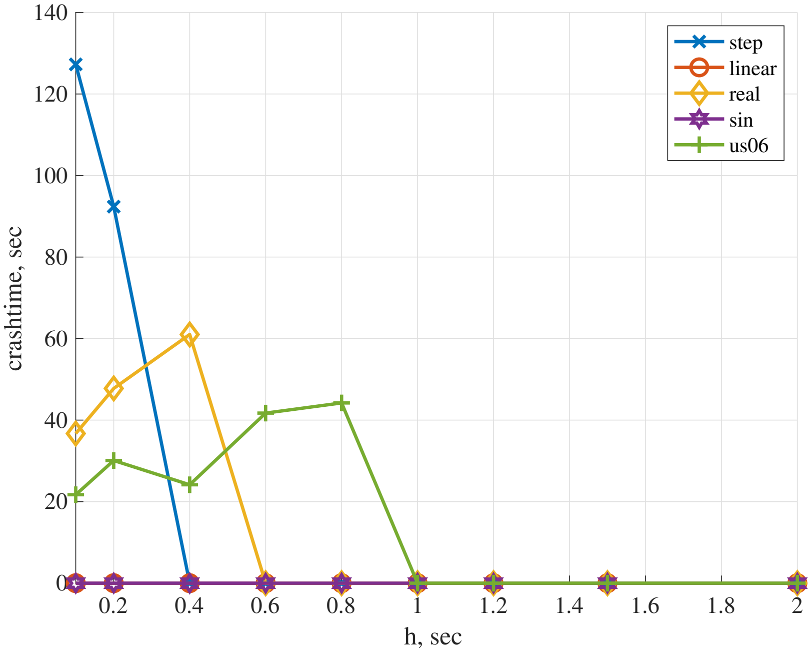


Figure 5.18: Crash time for OnOff Scenario

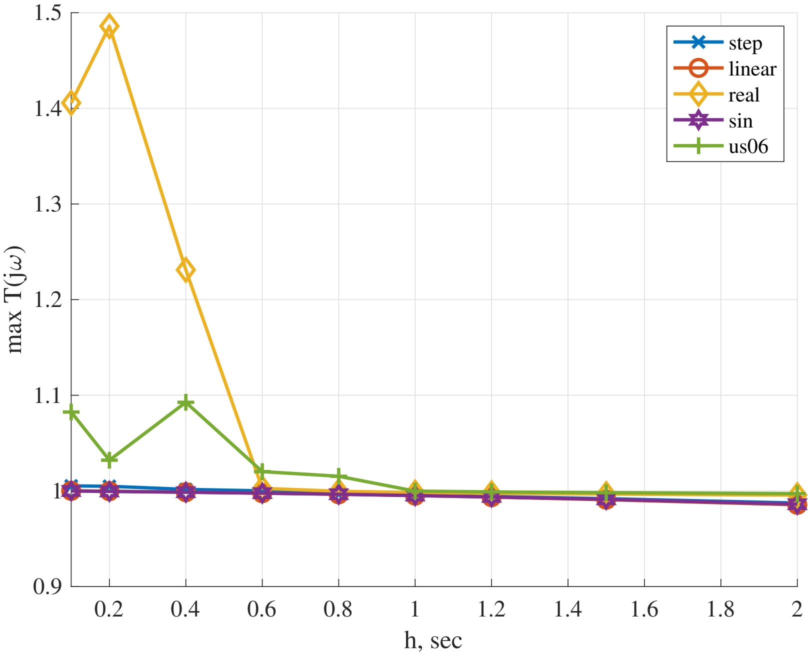
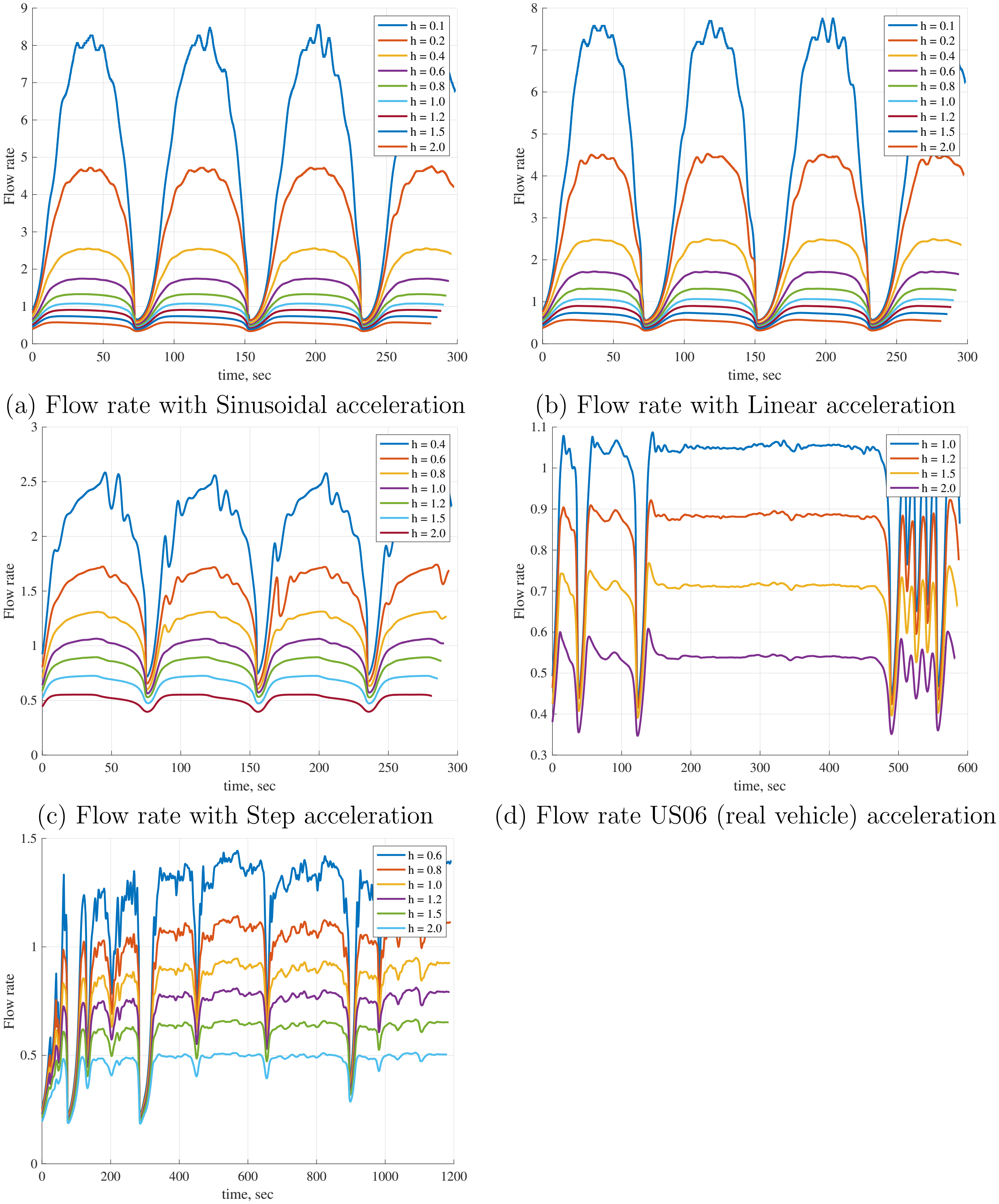


Figure 5.19: max $||\mathcal{T}(j\omega)||_{H_\infty}$ for CACC in On-Off scenario



(a) Flow rate with Sinusoidal acceleration

(b) Flow rate with Linear acceleration

(c) Flow rate with Step acceleration

(d) Flow rate US06 (real vehicle) acceleration

(e) Flow rate with C2A (real) acceleration

Figure 5.20: Flow rates for different acceleration profiles with On-Off scenario

acceleration profiles in crash free cases. We observed that in long burst loss scenario, the flow rate is increased compared to traditional ACC.

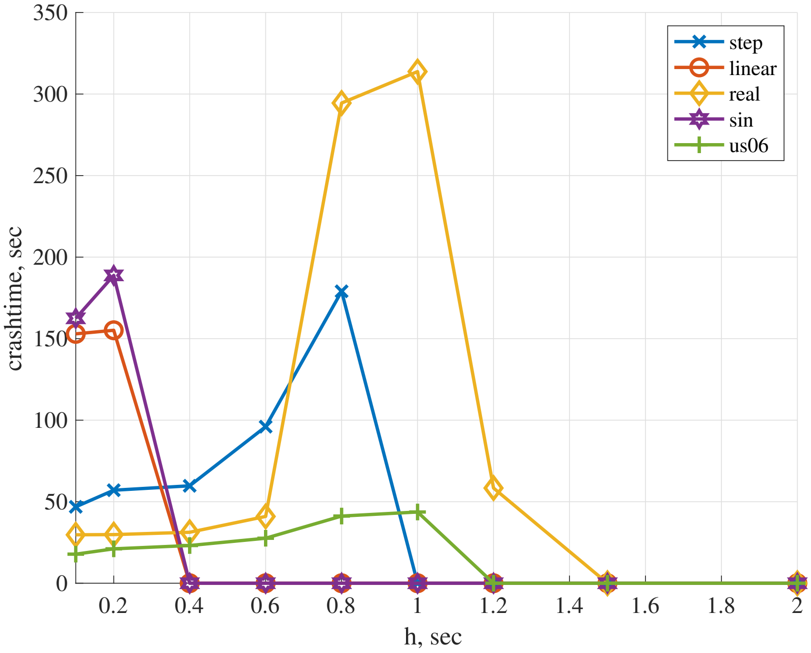


Figure 5.21: Crash time for long burst Scenario

5.3 Time Headway and Reliability

From the observations made with crash free and stable time headway and reliability in different scenarios, we come to the conclusion on how network reliability affects the selection of time headway. Reliability can be calculated from the burst loss process using Equation 5.3. As show in Table 5.3, the time headway can be pushed as low as 0.4s for system with very good reliability (r=1) and needs to be bounded above 1.5s for system with outage in wireless network (r=0). Figure 5.24 shows the line drawn based on the data points between reliability and time headway for N=10 vehicle platoon. The line connects the two limiting values of time headway. The equation for the line is as shown:

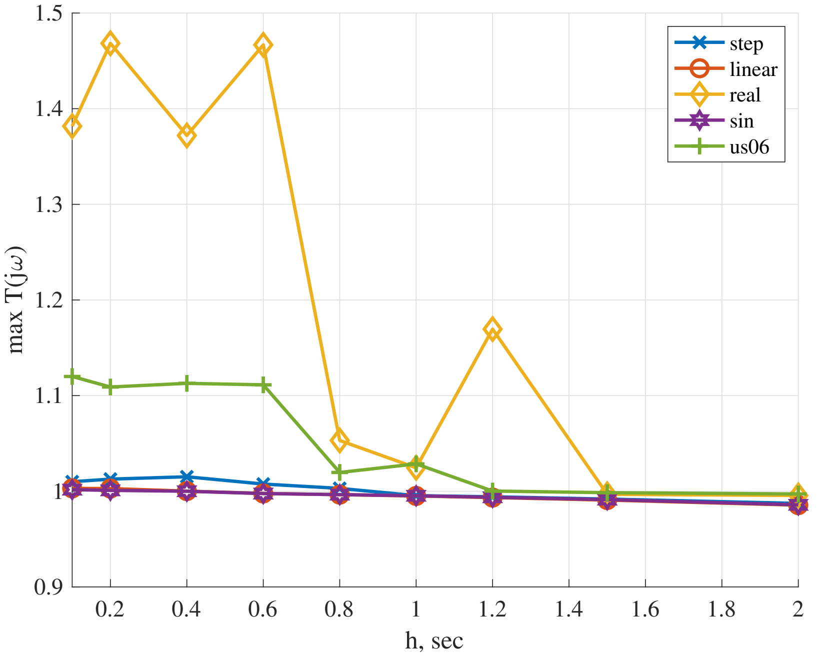


Figure 5.22: max $||\mathcal{T}(j_{H_\infty}$ for CACC in long burst scenario

$$h = f(r) = (L - U)r + U$$

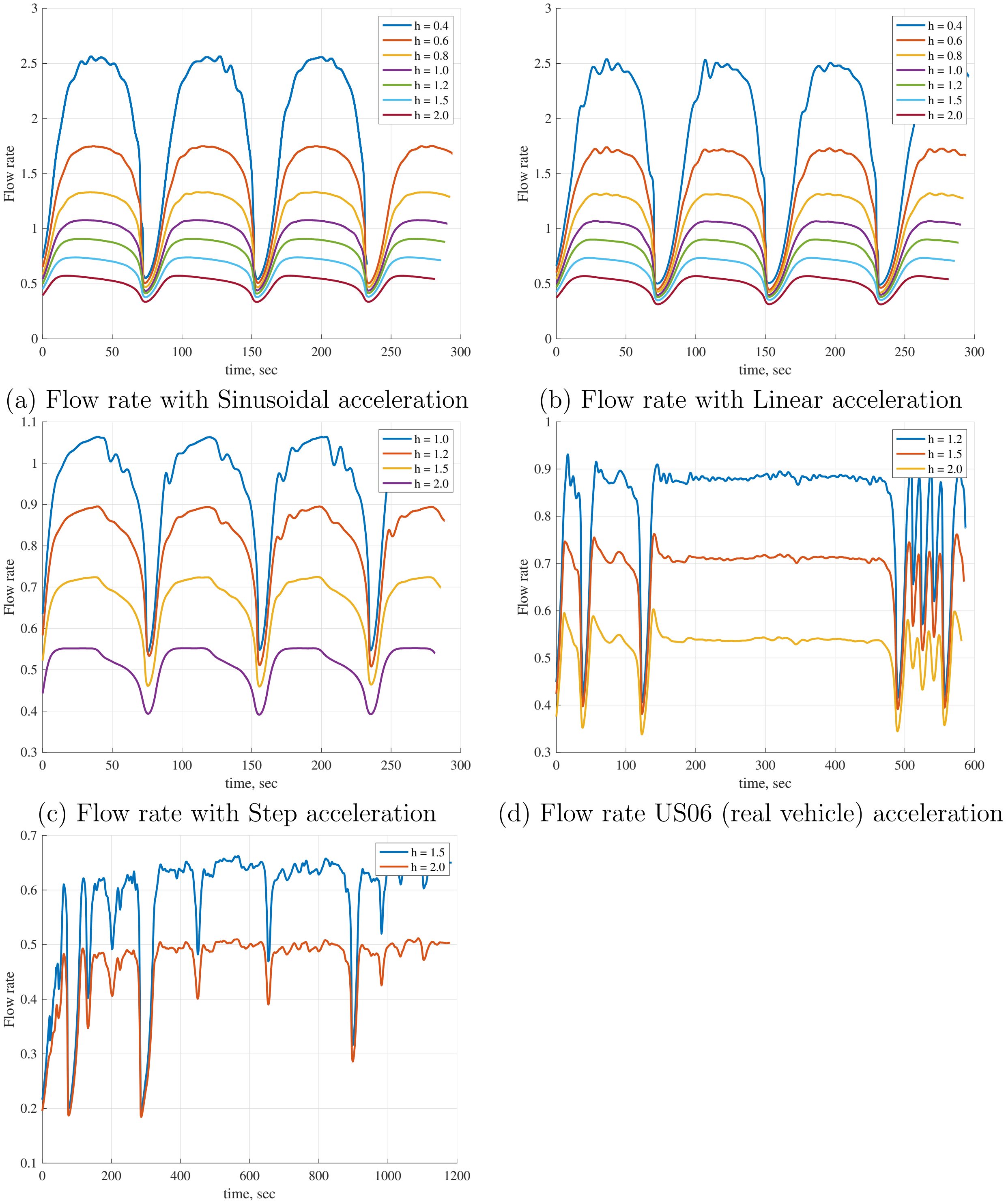
where, U = Upper bound on value of h

L = Lower bound on value of h

Similarly, multiple tests were performed with different number of vehicles in a platoon to observe the most safest values of time headway for the profiles given in 5.3. Figure 5.25 shows the line drawn between the upper and lower values of time headway from the data collected for different length of platoon. This figure shows the relation between the length of platoon and the minimum and maximum values of safe and stable time headway for a given network reliability.

reliability	headway
0	1.5s
1	0.4s

Table 5.3: Bounds on time headway vs reliability for 10 vehicle platoon



(a) Flow rate with Sinusoidal acceleration

(b) Flow rate with Linear acceleration

(c) Flow rate with Step acceleration

(d) Flow rate US06 (real vehicle) acceleration

(e) Flow rate with C2A (real) acceleration

Figure 5.23: Flow rates for different acceleration profiles with long burst scenario

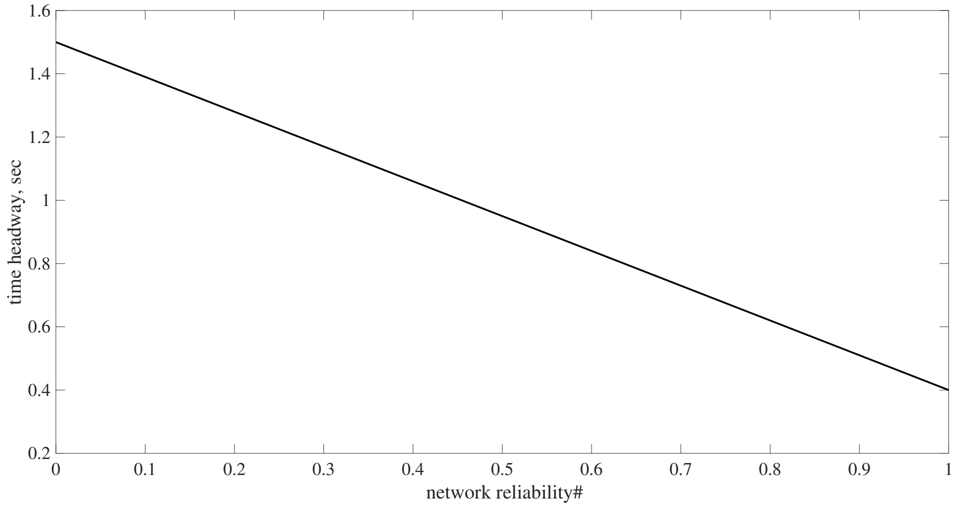


Figure 5.24: Time headway vs reliability for 10 vehicle platoon for $T_p = 1$s

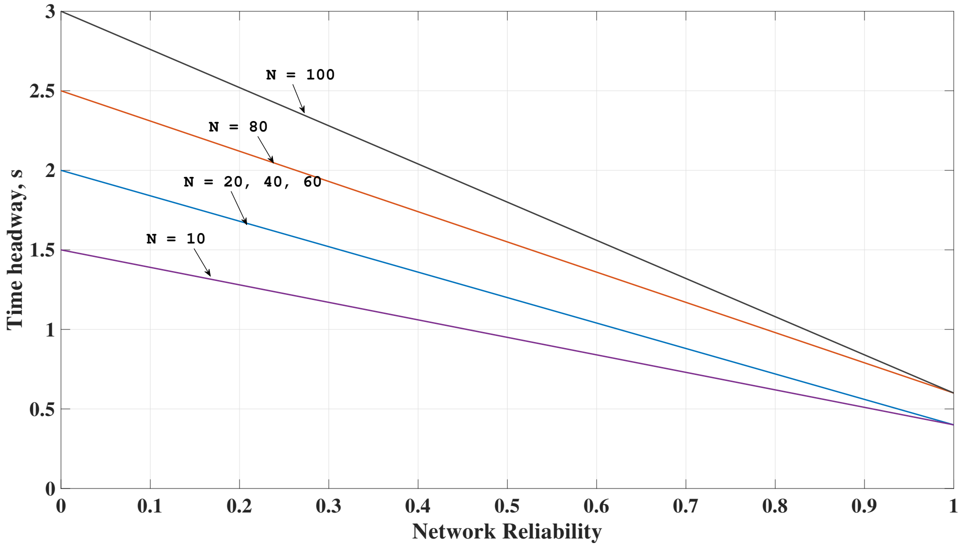


Figure 5.25: Time headway vs reliability for different length platoons for $T_p = 1$s

Chapter 6

Dynamic Headway Assignment

In this chapter we focus on the improvement to the traditional CACC systems by deploying a dynamic time headway algorithm. The dynamic headway algorithm is motivated by one of the main advantages of CACC i.e. to increase roadway throughput. Hence the algorithm is designed to improve flow rate in a safe and stable manner. The motivation behind this idea is as follows :

- reliability of wireless network is inherently important to make the platooning applications more stable and safer
- improving roadway throughput by maximizing flow rate is one of the motivation for platooning application
- flow rate can be maximized if the platoon can adjust, in real time, their time headway accordingly to the changes in reliability of the underlying wireless network

In this research we propose two algorithms to enhance the flow rate of a platoon by dynamically adjusting the time headway. The dynamic headway assignment can

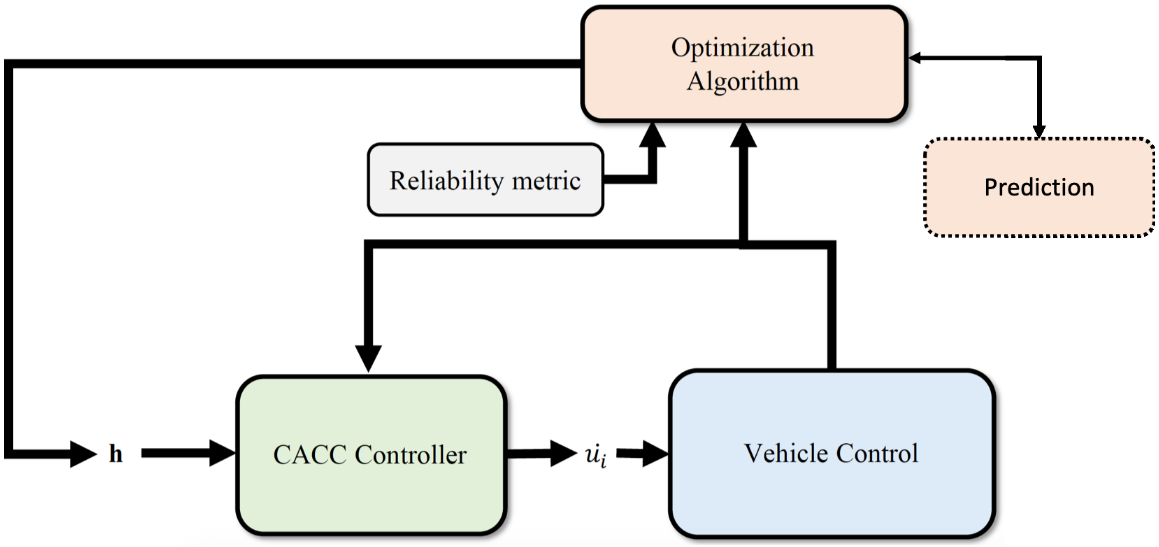


Figure 6.1: Dynamic headway controller

be implemented locally as well as globally. In a CACC controller, time headway is the only variable that can be adjusted in real time. Other inputs to the controller such as, velocity, acceleration and separation provide no mechanism to have an external entity control their behavior in real time except those variables are outcome of the vehicular motion. The notion of dynamically changing the time headway is comparable to the idea of Model Predictive Control presented (MPC) in [18–20]. While MPC systems predict and optimize the vehicular dynamics based on utility function that result in target acceleration that is injected to the vehicle's torque control, CACC systems are already optimized to reduce the distance separation gap error (d_{error}) and the result is also a target acceleration for each vehicle. The basic idea of the dynamic headway controller presented in this research is shown in figure 6.1.

The model suggests that the optimization algorithm can be deployed globally in a centralized method or embedded in each vehicle for local decision making in a decentralized method. Following assumptions are taken to describe the operation of the dynamic headway controller.

1. Platooning vehicles periodically broadcast, along with the vehicle status messages used for CACC operation, the control messages that include information on reliability of the wireless network $r_i(t)$, $\mathcal{M}(t)$ and current time headway $h_i(t)$.

2. The centralized global module for dynamic headway can be embedded in one of the vehicles or placed in a network reachable infrastructure node. Hence the implementation is either local or global based on the location of the nodes where information converge and decision is made.

3. We assume that the control messages are exchanged using either a different wireless technology (LTE/4G/5G) or exchanged in a different DSRC channel not being used by the platoon. Hence the channel conditions affecting the communication remain constrained to the wireless channel used by platooning vehicles to exchange vehicle status messages.

6.1 Problem Statement

The main objective of the dynamic headway assignment algorithm is to increase the flow rate $\mathcal{F}$ of vehicles in a roadway in a stable manner.

If N is the total number of vehicles V in a platoon $\mathcal{P} : V_i \in \mathcal{P} \ \forall \ i : 0 \leq i \leq N-1$. Let $v_i(t)$, $a_i(t)$, $x_i(t)$ and $\dot{u}_i(t)$ be defined as the velocity (units in fts^{-1}), acceleration (units in fts$^-2$), position (units in ft) and change in acceleration or jerk (units in fts^{-3}) of vehicle V_i at time t respectively. Similarly, h_i be defined as the time headway for vehicle V_i such that $L \leq h_i \leq U \ \forall i$. Then the maximization problem can be stated as follows:

$$\begin{aligned}
\max_{i} \mathcal{F} &= \frac{N}{\sum_{i=1}^{N-1} h_i} \\
\text{s.t.} h_i &\geq L \ \forall i \\
h_i &\leq U \ \forall i \\
h_i &\geq f(r_i) \ \forall i \\
x_{i-1} - x_i &\geq d_{stop} \ \forall i \\
d_{stop} &> 0
\end{aligned} \tag{6.1}$$

Using equations 4.5 from chapter 4, the the position of a vehicle at any time $t + \tau$ can be calculated as:

$$x_i(t+\tau) = x_i(t) + v_i(t) \times \tau + a_i(t) \times \frac{\tau^2}{2} + \dot{u}_i(t) \times \frac{\tau^3}{6} \tag{6.2}$$

Equation 6.2 is based on piece wise linear acceleration behavior of the vehicles movement and the jerk $\dot{u}_i(t)$ is calculated based on equation 4.5. Hence, the optimization solution would find a value of time headway $h_i(t)$ based on how the platoons move constrained by equation 6.2 i.e. the best value of time headway that would increase the flow rate while making sure there are no crashes i.e. $x_{i-1} > x_i$. Therefore, the optimization will forecast the movement of the platoon in the prediction phase based on the behavior of vehicular movement for all vehicles in platoon following the *Ego* vehicle. The problem can amplify and become unmanageable with longer platoon and unpredictable driving behavior. Therefore, we present following heuristic algorithms to address these issues.

6.2 Global Headway Controller

The dynamic time headway algorithm is implemented in a centralized node which could be located in either one of the followings:

- An infrastructure with low latency network to the platoon, e.g. a Road Side DSRC Unit operating in a different channel than the one being used by the platoon or a wireless infrastructure node that has LTE/4G/5G connectivity to the vehicles.
- A participating platoon vehicle that can receive all of the broadcast messages and facilitate the algorithm. The vehicle can be selected based on wireless network coverage to all the platooning vehicles for example a s vehicle in the middle of the platoon that can efficiently receive broadcasts from the *Leader* as well as the tail end vehicle.

The global headway controller (or, global controller) makes decision based on message collected from all of the vehicles. The mean absolute jerk $\mathcal{M}(t)$ which quantifies the rate of change of acceleration and can be used to identify onset of instability and the reliability measured at each vehicles are parts of the message broadcast by the platoon vehicles. The algorithm given in Heuristic 1 shows the steps taken by a centralized entity to make safe decisions on the headway assignment to all the vehicles in a platoon. The decisions are based on reliability metric and the mean absolute jerk over the renewal periods calculated by each vehicle. The changes in the value of time headway h is allowed only once per renewal period of at least one of the vehicles. This is done to prevent the $\mathcal{M}(t)$ recursively amplify with the change in time headway. The global controller adapts the same value of headway to all the vehicles.

The reliability is calculated each T_p period while $\mathcal{M}(t)$ is calculated after the

Heuristic 1: Global dynamic headway assignment algorithm

```
initialization k,$\alpha$,$\Delta$,L,U,T$_p$, A$_{max}$;
begin h=U;
while do
    while in each vehicle ($V_i$) do
        $r_{i,s}[k] = \frac{N_{i,received}}{N_{i,p}}$;
        $r_{i,a}[k+1] = (1-\alpha)r_{i,a}[k] + \alpha r_{i,s}[k]$;
        $\mathcal{M}_i[k] = \frac{\sum_{t-Ts}^{t} |\dot{u}_i[t = mT_{pc}]|}{t - T_s}$;
        if renewal in any Vehicle then
            $P_{renew} = True$;
        end
    end
    while in global node do
        $h_{new} = f(\min r_{i,a}[k+1])$;
        if $P_{renew} == True$ then
            if $h - h_{new} > \Delta$ then
                if $\max_i \mathcal{M}_i[k] \leq A_{max}$ then
                    if $h - \Delta \geq L$ then
                        $h = h - \Delta$;
                    else
                        $h = L$
                    end
                end
            else
                if $h - h_{new} \leq \Delta$ then
                    if $h_{new} \leq U$ then
                        $h = h_{new}$;
                    else
                        $h = U$
                    end
                end
            end
            $P_{renew} = False$;
        end
    end
    k=k+1;
end
```

start of a renewal period of $\dot{u}_i(t)$ over $t - T_s$ period where T_s is the start time of the latest renewal period. The vehicles broadcast the updates on reliability and $\mathcal{M}(t)$ every T_p period. The global headway controller calculates the minimum of the reliability updates and maximum of the mean absolute jerk. The new headway h_{new} is calculated using the minimum reliability. The new headway is further used for adaptation only if the maximum $\mathcal{M}(t) \leq A_{max}$, where A_{max} is the maximum acceleration achievable by the vehicles and at least one renewal period in any vehicle has passed since the last time headway was dynamically changed. If the new headway is larger than the current headway, then the new headway is updated to all the vehicles where as if the new headway is smaller than the current headway by at least the value of Δ ($\Delta \leq 0.02$), then $h - \Delta$ is updated to the vehicles.

Parameters	Values
N	10
α	0.8
Δ	0.02
T_p	1 sec
$N_{i,p}$	40
L	0.4s
U	1.5s
A_{max}	10 fts^{-2}

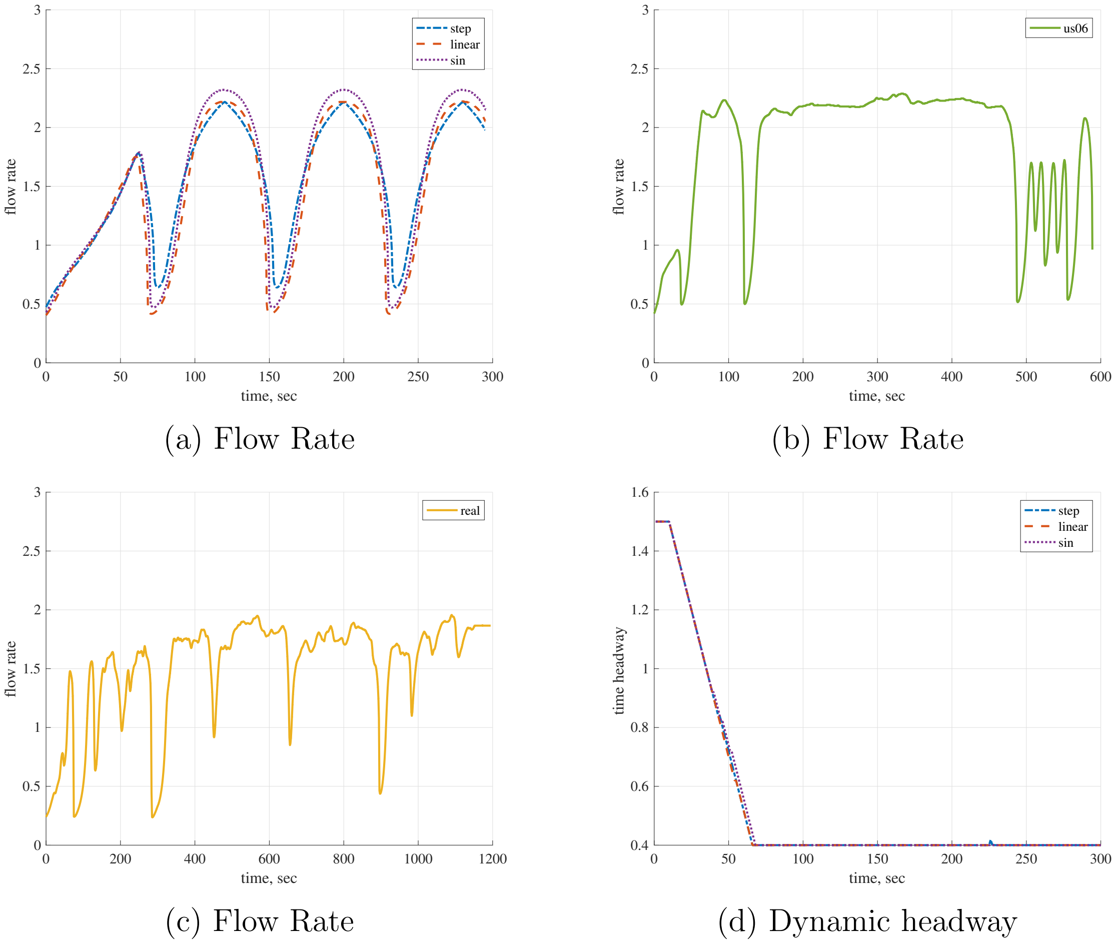
Table 6.1: Parameters setup for simulating heuristic algorithms

6.2.1 Results with Global Headway Controller

In this section, the results of simulating the global headway controller in a 10 vehicle platoon is presented. The tests are carried out with the same acceleration profiles as before (see figure 5.3). However, the simulation time is divided into three equal sections. The first and last sections of the simulation time have no packet loss. During the middle section, the platoon undergoes through different packet losses as given in table 5.1. The parameters settings used in the simulation are given in table 6.1.

Figure 6.2 (a-c) shows the flow rate for different acceleration profiles and the average time headway is shown in (d). As seen here, the headway is initialized to U and it adapts itself to reliable wireless network and converges to an applicable value that maximizes flow rate. The global controller collects the periodic information on reliability metric and mean absolute jerk, which are then used to estimate the best headway for a given situation. This eventually leads to the adaptation of time headway over the length of the platoon targeted for safe maximization of flow rate.

Similarly, figure 6.3 shows the flow rates and average headway observed during the simulation of global controller in a congested scenario. As expected, the global controller adapts the same headway to each vehicle based on global information of reliability and mean absolute jerk. In congested scenario, the vehicle(s) with worst reliability metric and mean absolute jerk control the adaptation algorithm. Figure 6.4 represents the observed flow rate and average headway over time in a scenario with $b_g = 400$ and $b_l = 100$. In this case the system adapts to changing reliability and modulates headway based on the worst reliable wireless network. Figure 6.4 (d) illustrates the headway adapted based on globally collected information. Also figure 6.5 shows the flow rate and headway observed for scenario with $b_g = 100$,$b_l = 100$. It can be seen that the global controller takes the worst reliability observed from each vehicle and modulates headway based on that.



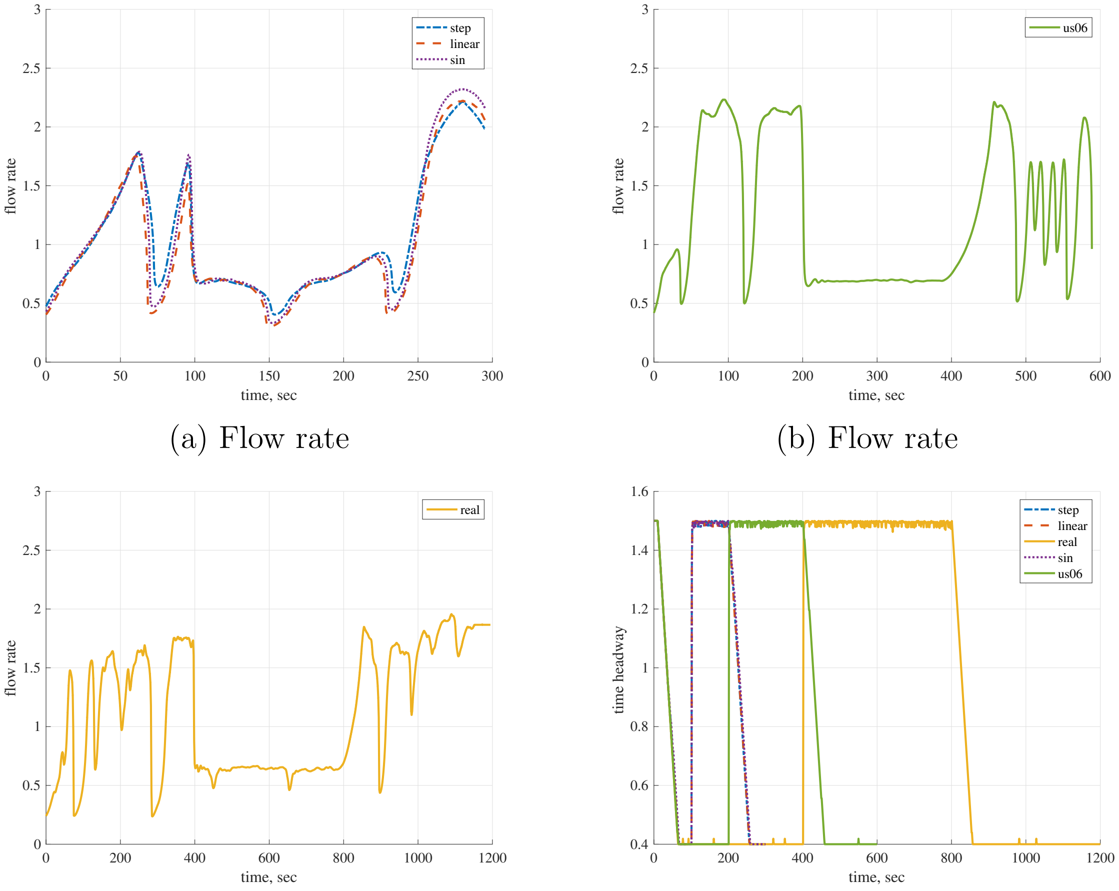
(a) Flow Rate

(b) Flow Rate

(c) Flow Rate

(d) Dynamic headway

Figure 6.2: Results with global dynamic headway controller in no packet loss



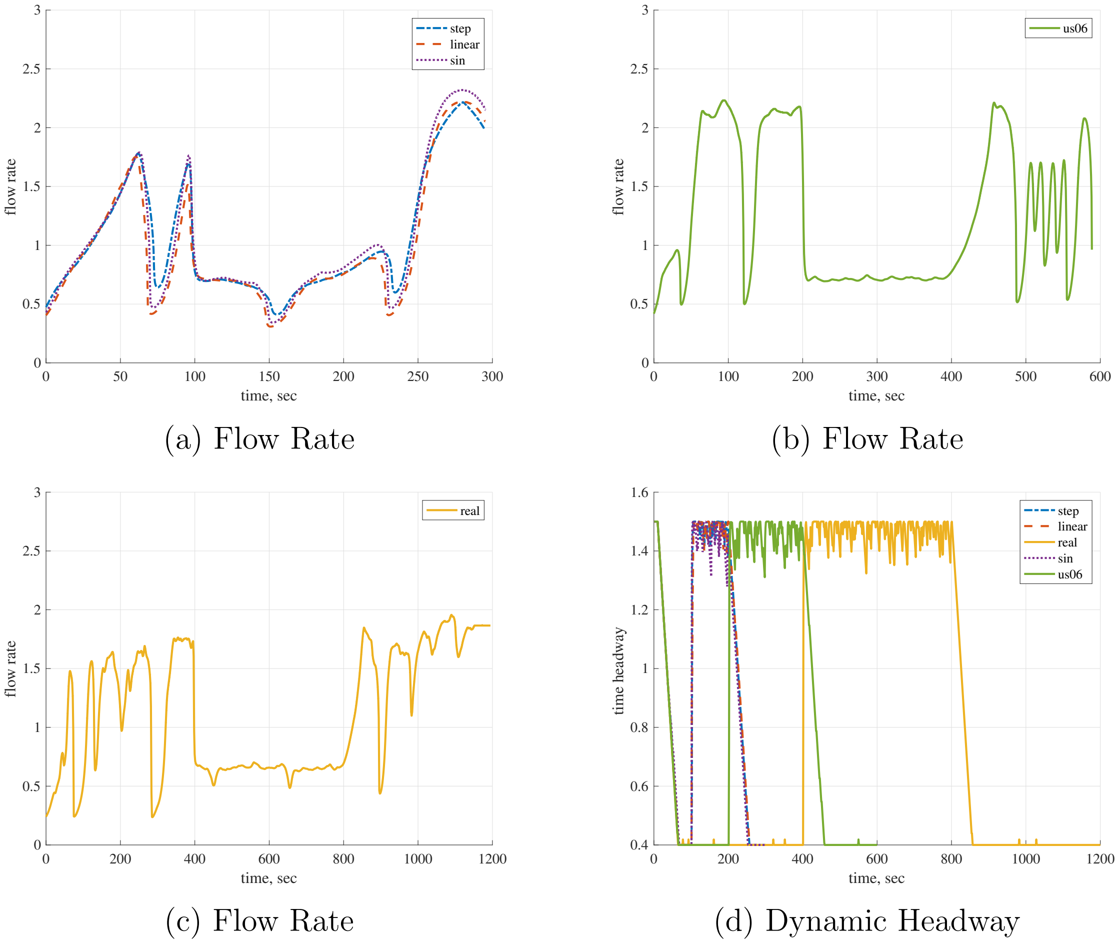
(a) Flow rate

(b) Flow rate

(c) Flow rate

(d) Dynamic headway

Figure 6.3: Results with global dynamic headway controller in Congested Scenario



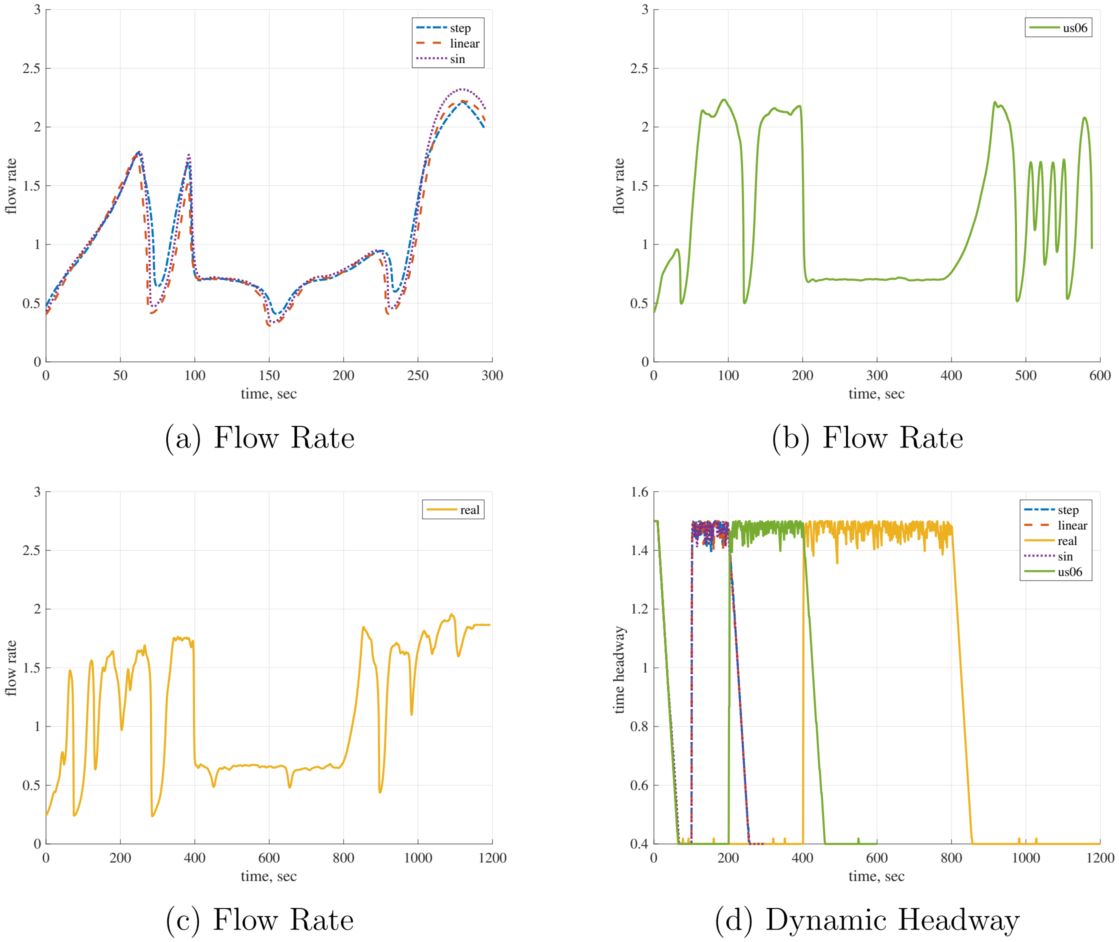
(a) Flow Rate

(b) Flow Rate

(c) Flow Rate

(d) Dynamic Headway

Figure 6.4: Results with global dynamic headway controller in scenario with $b_g = 400, b_L = 100$ packets



(a) Flow Rate

(b) Flow Rate

(c) Flow Rate

(d) Dynamic Headway

Figure 6.5: Results with global dynamic headway controller in scenario with $b_g = 50, b_L = 50$ packets

6.3 Local Headway Controller

Analogous to the global controller, the local controller can be implemented in every vehicle and select a safe headway value in order to maximize the flow rate of the vehicles. We assume that the local controller is a part of the vehicle and works in tandem with the platoon controller. The local controller of an *Ego* vehicle collects the locally evaluated reliability and mean absolute jerk values and computes the heuristic algorithm given in heuristic 2. The adapted value of headway is used in the *Ego* vehicle only. In this method, the headway h is updated individually, therefore every vehicle can have distinct time headway.

The adaptation of headway is carried out by the local controllers in each platooning vehicles. As shown in the algorithm in heuristic 2, each vehicle calculates reliability metric over T_p time and the mean absolute jerk metric $\mathcal{M}(t)$ over each renewal period of $\dot{u}(t)$. These information are used by the local controller to adapt based on if $\mathcal{M}(t) \leq A_{max}$ and if the headway had been updated within the last renewal period. The headway is allowed to increase as needed till the upper bound of U but it is only allowed to be decreased by Δ as long as it stays above the lower bound L. The decisions are made every T_p period.

6.3.1 Results with Local Headway Controller

In this section the results for local controller are presented for different acceleration profiles and burst loss similar to the ones used in local controller. The simulation comprises of three equal sections- first and last one-third of the simulation time has no packet loss where as the middle one-third section has different burst losses. The parameters used in the simulation are given in table 6.1.

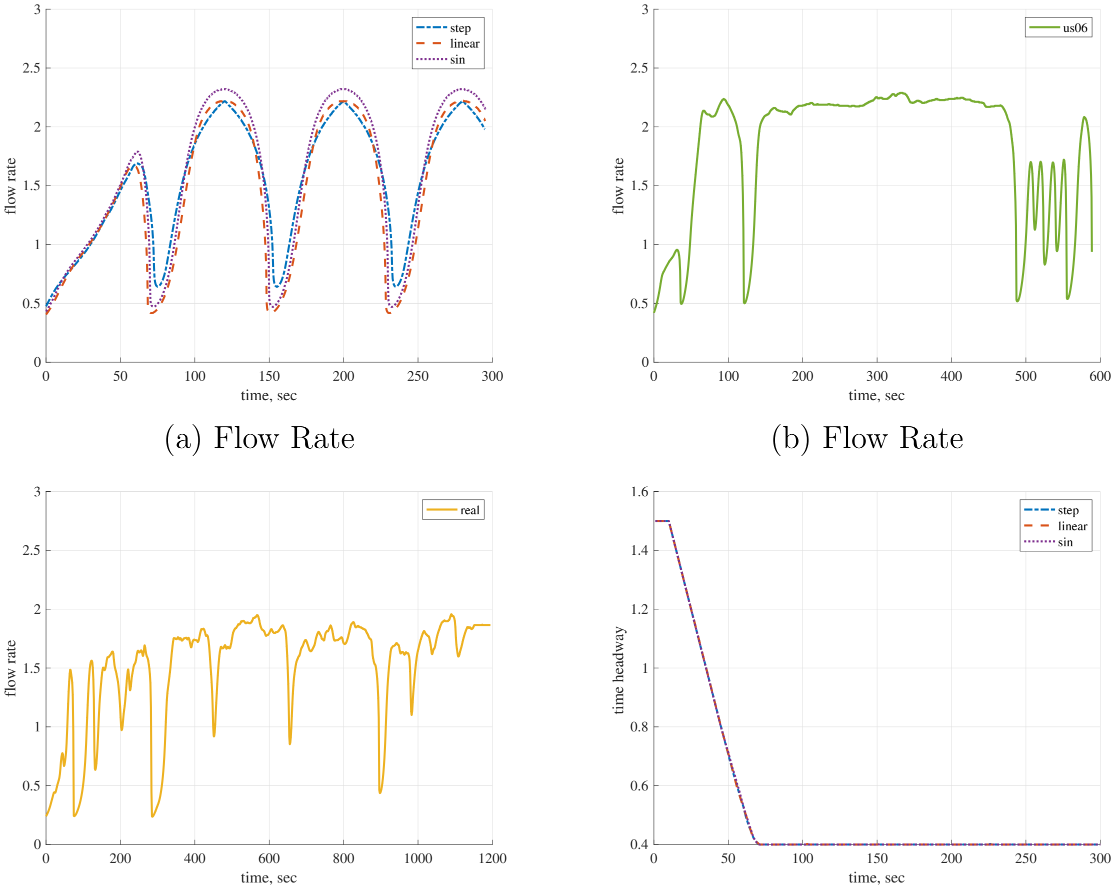
Figure 6.6 (a-c) show flow rates observed with local controller implementation

Heuristic 2: Local Dynamic Headway Assignment Algorithm

```
initialization k,$\alpha$,$\Delta$,L,U,T$_p$, A$_{max}$;
begin h=U;
while in each vehicle ($V_i$) do
    $r_{i,s}[k] = \frac{N_{i,received}}{N_{i,p}}$;
    $r_{i,a}[k+1] = (1-\alpha)r_{i,a}[k] + \alpha r_{i,s}[k]$;
    $\mathcal{M}_i[k] = \frac{\sum_{t-Ts}^{t} |\dot{u}_i[t = mT_{pc}]|}{t - T_s}$;
    if renewal in $V_i$ then
        $P_{i,renew} = True$;
    end
    $h_{i,new} = f(r_{i,a}[k+1])$;
    if $P_{i,renew} == True$ then
        if $h_i - h_{i,new} > \Delta$ then
            if $\mathcal{M}_i[k] \leq A_{max}$ then
                if $h_i - \Delta \geq L$ then
                    $h_i = h_i - \Delta$;
                else
                    $h_i = L$
                end
            end
        else
            if $h_i - h_{i,new} \leq \Delta$ then
                if $h_{i,new} \leq U$ then
                    $h_i = h_{i,new}$;
                else
                    $h_i = U$
                end
            end
        end
    end
    if $h_i \neq h_{i,new}$ then
        $P_{renew} = False$;
    end
    k=k+1;
end
```

in a 10 vehicle platoon. The flow rates match well with the observed flow rate in global controller scheme. Figure 6.6 (d) is the mean headway adapted for all vehicles. In this scenario, each vehicle has different headway values so the plot shows only the mean. As expected the headway slowly adapts from U to L in very reliable wireless network. Similarly figure 6.7 shows the flow rates and average headway for a congested scenario. In a congested scenario, the flow rates match very closely to the global controller scheme. Figure 6.8 show the flow rates and headway time for scenario with $b_g = 400$, $b_l = 100$. Figure 6.9 illustrates the flow rate and headway for system $b_g = 100$,$b_l = 100$. The mean headway of all vehicles is smaller than the one observed for global controller during the same scenario. This shows that the headway for local controller is smaller than the global controller for similar conditions. It can also be seen that the headway is basically similar in cases where $r_i = 0$ or $r_i = 1$.

In this chapter we presented the development and analysis of two heuristic algorithms for maximizing the traffic flow in a platooning application. The motivation lies in determining more realistic operating conditions for a platoon and adapting the time headway in order to maximize the flow of vehicles while ensuring that the platoon remains stable and safe. The analysis presented consisted of simulation and demonstration of real time headway adaptation in different acceleration and burst packet loss scenarios.



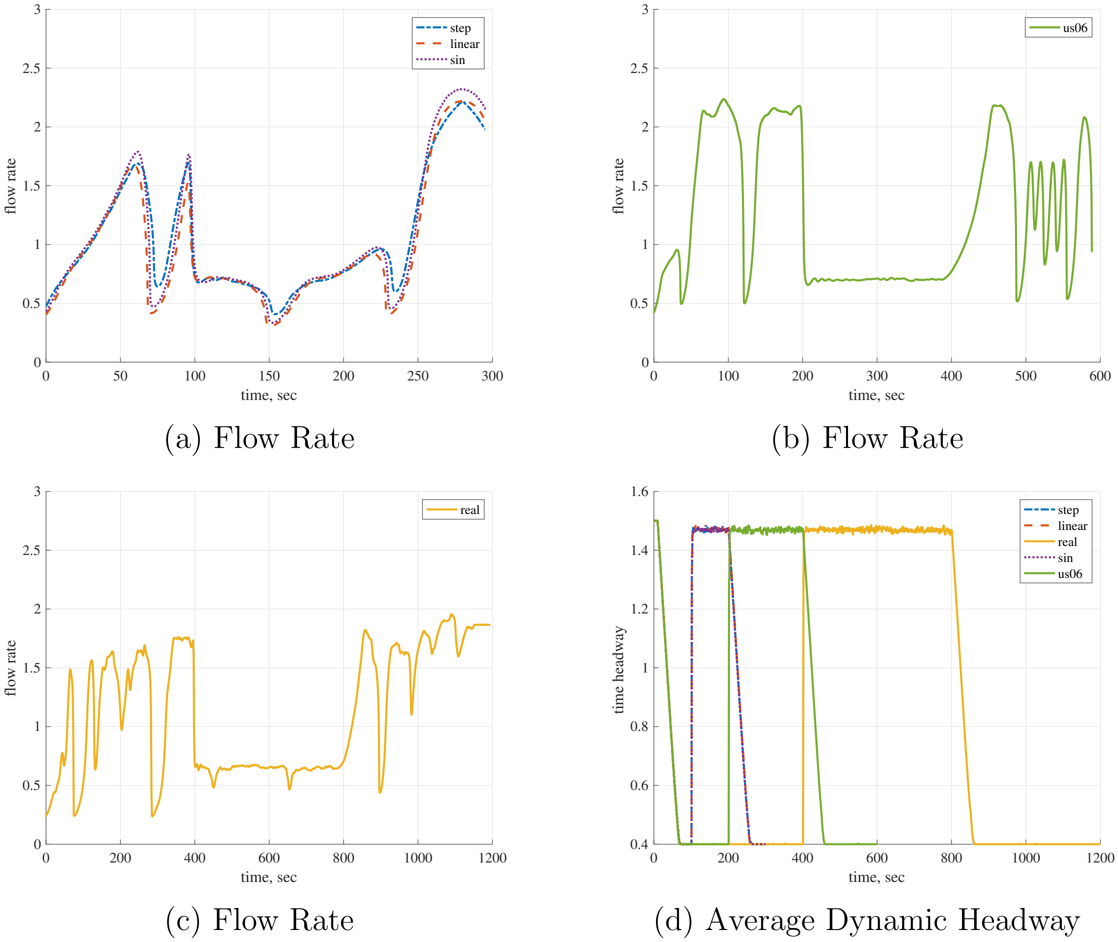
(a) Flow Rate

(b) Flow Rate

(c) Flow Rate

(d) Average Dynamic Headway

Figure 6.6: Results with local dynamic headway controller in scenario with no packet loss



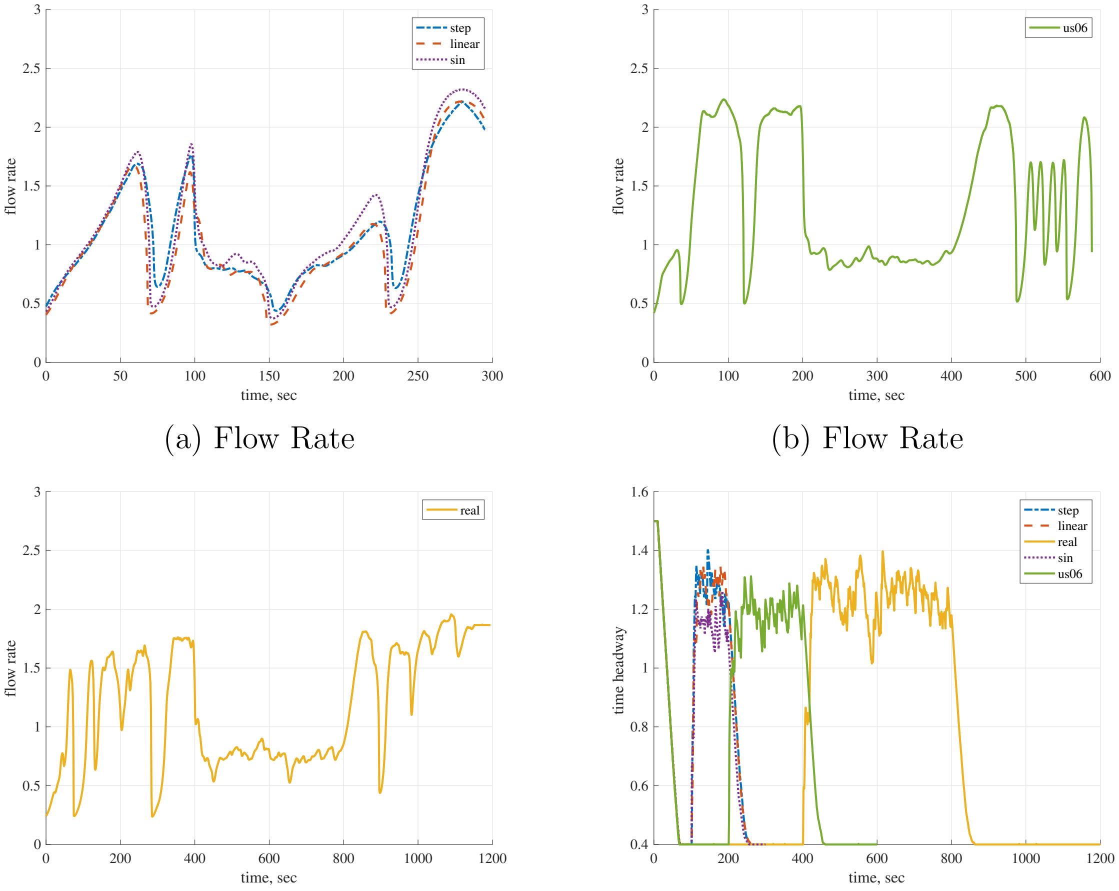
(a) Flow Rate

(b) Flow Rate

(c) Flow Rate

(d) Average Dynamic Headway

Figure 6.7: Results with local dynamic headway controller in scenario with congestion



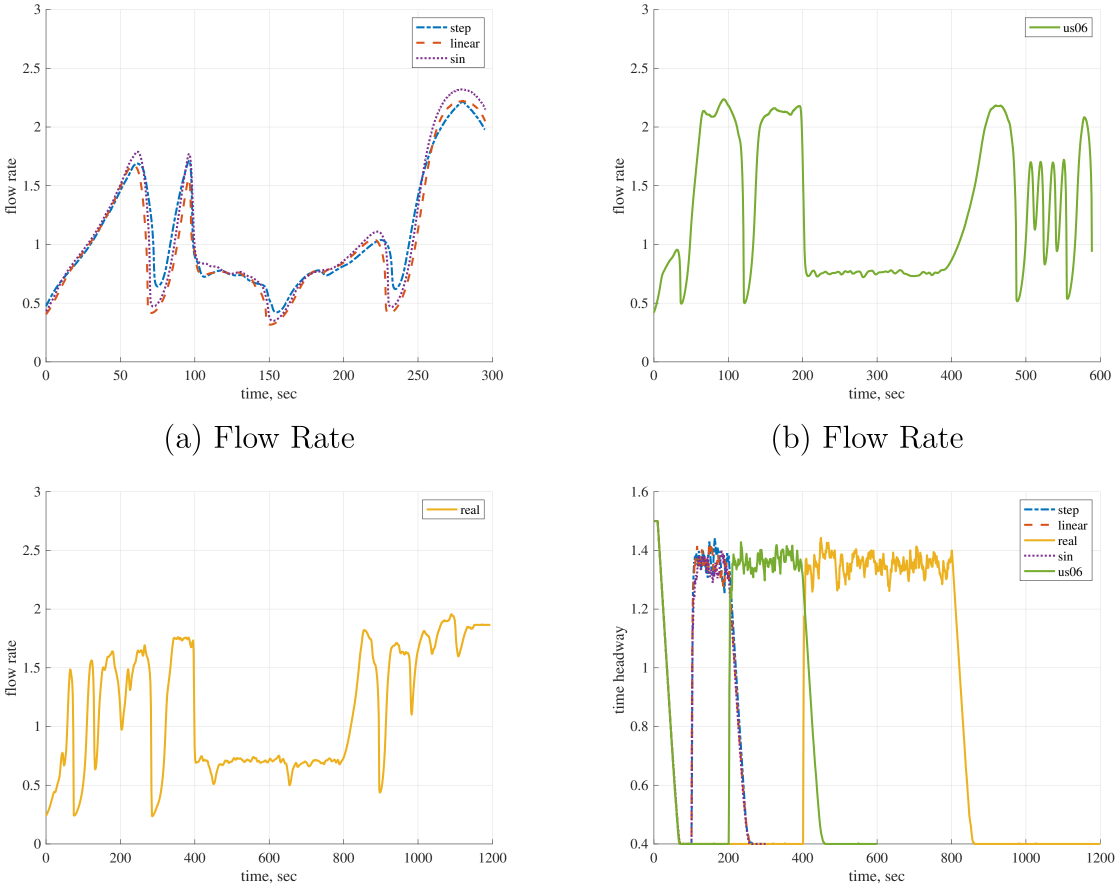
(a) Flow Rate

(b) Flow Rate

(c) Flow Rate

(d) Average Dynamic Headway

Figure 6.8: Results with local dynamic headway controller in scenario $b_g = 400, b_L = 100$ packet



(a) Flow Rate

(b) Flow Rate

(c) Flow Rate

(d) Average Dynamic Headway

Figure 6.9: Results with local dynamic headway controller in scenario with $b_g = 50, b_L = 50$ packets

Chapter 7

Conclusion

Cooperative Adaptive Cruise Control can improve the highway throughput, increase safety in our roadways and decrease fuel consumed by the vehicles. It has many benefits that have bolstered it to be an important Connected and Autonomous vehicle application heavily favored to be pushed out in near future. Its evident that if properly designed and deployed, the CACC can be a driving force in future transportation system that is seemingly going to be overrun by autonomous and connected vehicles. Therefore, a good understanding of various implications of CACC and many factors that play vital role in governing its safe and stable form is timely.

We have studied CACC by observing the system into various aspects of its operation and by simulating as well as testing in real world applications. Wireless network is an important component of the platoon and many previous studies lack in understanding the system from the perspective where the wireless network is unreliable, congested or under a malicious attack.

In this research, we have developed techniques to prevent the system from being unstable in case when wireless network becomes unavailable. The Kalman filter based estimated acceleration serves as an efficient fallback technique. We developed the switching method for selecting the most reliable version of acceleration. Additionally we developed a heuristic algorithm for increasing flow rate of a realistic platoon in a manner that avoids crash and improves stability. We developed two types of such algorithm- a centralized (global) algorithm that can be applied either in one of the platooning vehicles or in an infrastructure agent, and a decentralized (local) algorithm that can be implemented in each of the platooning vehicles and avoids the need of centralized data collection. We tested the algorithm in varieties of scenarios and present the result that show that the flow rates can be improved in a safe manner.

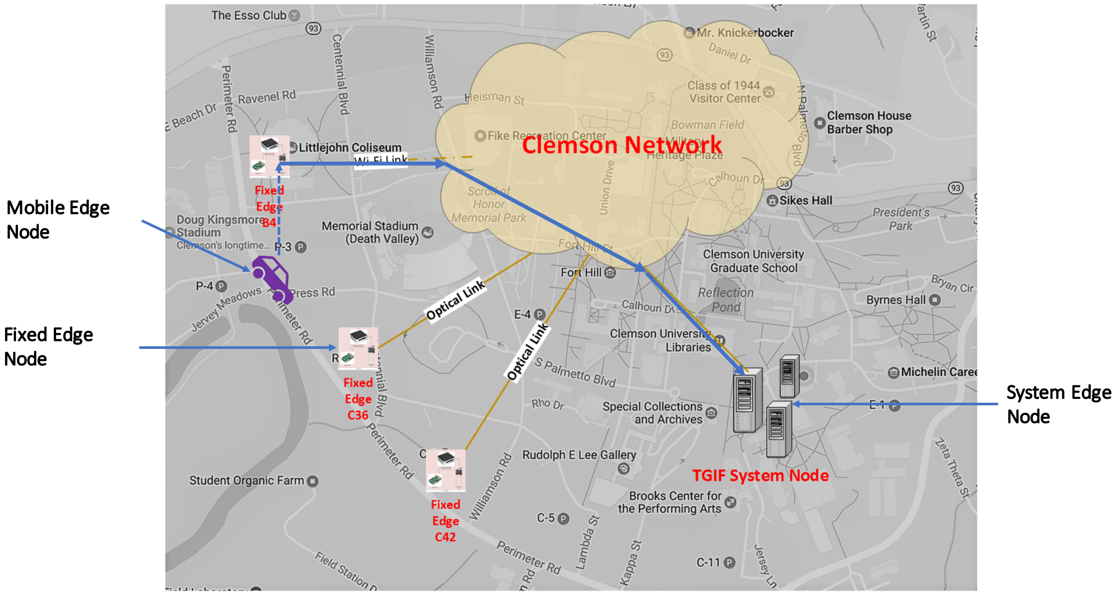
Appendices

Appendix A List of Abbreviations and Usage

Acronyms	Terms
PHY	Physical Layer
MAC	Medium Access and Control
DSRC	Dedicated Short Range Communication
OFDM	Orthogonal Frequency Division Multiplexing
EDCA	Enhanced Distributed Channel Access
Mhz	Mega Hertz
Ghz	Giga Hertz
AIFSN	Arbitration Inter-Frame Space Number
CSMA	Carrier Sense Multiple Access
TDMA	Time Division Multiple Access
AC	Access Class

Table 1: List of Abbreviations

Appendix B Clemson University Connected Vehicle Test-bed



Clemson University Connected Vehicle Test-bed was setup to test and deploy advanced wireless networks for vehicular applications. Located along the perimeter road surrounding the beautiful Clemson University campus are three infrastructure installations on light posts that house a DSRC wireless radio and a powerful compute node, collectively called Fixed Edge Node (FEN). A server node located in McAdams Hall in Clemson University to abstract as an end point for all data and message flows from the Fixed Edge Nodes is called the System Edge Node (SEN). Mobile Edge Nodes (MEN) is defined as a set of DSRC wireless radio and a compute node placed in a moving vehicle capable of communicating with any infrastructure node along the test bed. The FEN's are located such that there is overlap in coverage zone between two consecutive radios along the perimeter road. This allowed for continuous wireless connectivity between a MEN moving along the perimeter road and the infrastructure.

Two of the three FENs are connected to the SEN via Clemson University's optical fiber network while one of the FEN is backhauled via Wi-Fi. The radios used in the test-bed are configurable to 802.11p WAVE or IP modes. In IP modes, they are configured as an ad-hoc Wi-Fi interface capable of broadcasts.

The radios are configurable in the type of Modulation and Coding schemes to use, transmit power, the bandwidth (10 MHz or 20 MHz) and the Access Class to be used. Typically in WAVE mode the WSMP packet header contains information regarding the bandwidth, transmit power Modulation and Coding, and the Access Class for each individual packets so that the underlying Link layer assigns the queues and transmit the packets as requested. Some of the applications developed in the test-bed include :

- Network Performance Analysis
- Adaptive Queue Warning Prediction
- Forward Collision Avoidance
- Traffic Data Collection
- Cooperative Adaptive Cruise Control

One of the issues we faced in the test-bed involved proprietary WAVE stack firmware that were dependent in the vendors. Therefore, the initial testings used in the research were conducted in IP mode.

Appendix C Running Simulations in ns-3 and Matlab©

This section briefly describes the steps taken to simulate the CACC application in ns-3 and Matlab©.

C.1 Simulating in ns-3

The source code for simulating the platoon in ns-3 can be obtained as shown:

```
$ git clone buffet@buffet.cs.clemson.edu:arayama/ns3-dev
```

The clone folder contains ns-3 ver. 3.24 and a script **installns3.sh** that can be run as shown to install ns-3 in a Linux (tested in Ubuntu 16.04 LTS) system

```
$ sh installns3.sh
```

This script checks for prerequisites and installs them before install ns-3 in the *build* folder. The necessary scripts for simulation are placed in *scratch* folder and can be compiled by using

```
$ ./waf
```

To simulate the CACC application:

The arguments are defined as shown:

- RANDSEED = Random Seed set to the random number generator in ns-3
- NumberofNodes = Total number of vehicles in the platoon to simulate starting with front vehicle 0
- Datarate = bit rate for broadcast network in kbps
- PacketSize = size of broadcast message to use (typically 200 B)

- SimTime = total length of time for simulation to run

- Phymode = ns-3 specific defines that abstract the type of Modulation and Coding to be used in 802.11p network (typically, Rate 1/2 QPSK is given as OfdmRate6MbpsBW10MHz)

- PlatoonLength = number of vehicles starting from the front vehicle counted as 0 to actually participate in a platoon (usually PlatoonLength = NumberofNodes)

- HeadwayTime = set h value in seconds

- Accln Profile = type of acceleration profile to be used for the *Leader* vehicle (the acceleration profile files are named with '.acc' extension

- Caccupdatetime = time span between consecutive CACC updates

- Moveupdatetime = time span between consecutive piece wise linear acceleration updates used for vehicle mobility

- Leaderupdatetime = time span between consecutive *Leader* vehicle's acceleration updates

- Dref = reference distance in feet.

- ACC_basic, CACC_U, CACC_A, CACC_DU and CACC_RU = selection on type of CACC (with wireless or without wireless and only using sensors) and ACC to simulate

- DMEupdatetime = time span between consecutive Sensor updates

- LogfileLocation = folder location to store all the log files generated per simulation

- LossStartTime = start time in seconds w.r.t the simulation start time when the loss process acting on the wireless network becomes active
- LossDuration = length of time since the LossStartTime till the loss process is active
- numLossVehicles = number of vehicles affected by the loss process
- MeanBurstLength = MBL for loss process simulation
- MeanGoodLength = MGL for loss process simulation
- BackupMode = fall back CACC mode to be used in case the wireless network drops packets
- distnNoise = standard deviation used to simulate noise in distance measurements
- velnoise = standard deviation used to simulate noise in velocity measurements

After the simulation finishes, the log files will be stored in LogLocation. A set of different log files contain mobility information (acceleration, velocity, position) about each platooning vehicles, their communication logs (received time, sent time, packet counter and RSSI in dBm), and CACC related logs (distance error, target acceleration).

C.2 Simulating in Matlab©

The simulation files for Matlab© can be cloned using

```
$ git clone https://buffet.cs.clemson.edu/vcs/u/arayama/matlabscripts/
```

The Matlab© scripts already has preset all the acceleration profiles and different h values to use during the simulation. To run the scripts following command need to be executed

```
>> usimALLcacc2(foldername,appendstring,N,sd,sv,BG,BL,usekalman)
```

where,

- foldername = name of folder to store the result logs
- appendstring = unique string to append to each subfolder that hold results for different values of h
- N = number of platooning vehicle
- sd = standard deviation of noise in distance measurements
- sv = standard deviation of noise in velocity measurements
- BG = Mean Good Length
- BL = Mean Burst Length
- usekalman = 1 if Kalman estimation of *Lead* vehicle's acceleration is to be used and 0 if no estimation is to be used.

Bibliography

[1] Road traffic injuries. http://www.who.int/mediacentre/factsheets/fs358/en. Accessed: 2018-2-19.

[2] WHO, the top 10 causes of death. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. Accessed: 2019-4-30.

[3] National Highway Traffic Safety Administration. Automated driving systems: A vision for safety 2.0. 2017.

[4] NHTSA. Automated vehicles for safety. URL https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety. Accessed: 2018-8-10.

[5] Y. L. Morgan. Notes on DSRC amp; WAVE standards suite: Its architecture, design, and characteristics. *IEEE Communications Surveys Tutorials*, 12(4): 504–518, Fourth 2010. ISSN 1553-877X. doi: 10.1109/SURV.2010.033010.00024.

[6] Ieee standard for telecommunications and information exchange between systems - lan/man specific requirements - part 11: Wireless medium access control (mac) and physical layer (phy) specifications: High speed physical layer in the 5 ghz band. *IEEE Std 802.11a-1999*, pages 1–102, Dec 1999. doi: 10.1109/IEEESTD.1999.90606.

[7] Yunxin Jeff Li. An overview of the DSRC/WAVE technology. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 544–558. Springer, 2010.

[8] John B Kenney. Dedicated short-range communications (DSRC) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[9] Xianan Huang, Ding Zhao, and Huei Peng. Empirical study of DSRC performance based on safety pilot model deployment data. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[10] Seungbae Lee and Alvin Lim. An empirical study on ad hoc performance of DSRC and wi-fi vehicular communications. *International Journal of Distributed Sensor Networks*, 9(11):482695, 2013.

[11] Sebastian Gräfling, Petri Mähönen, and Janne Riihijärvi. Performance evaluation of IEEE 1609 WAVE and IEEE 802.11 p for vehicular communications. In *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*, pages 344–348. IEEE, 2010.

[12] Fan Bai and Hariharan Krishnan. Reliability analysis of DSRC wireless communication for vehicle safety applications. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 355–362. IEEE, 2006.

[13] Connected vehicle reference implementation architecture. `https://local.iteris.com/cvria/`. Accessed: 2017-10-15.

[14] Zoleikha Abdollahi Biron, Satadru Dey, and Pierluigi Pisu. On resilient connected vehicles under denial of service. 05 2017.

[15] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer. Graceful degradation of cacc performance subject to unreliable wireless communication. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1210–1216, Oct 2013. doi: 10.1109/ITSC.2013.6728397.

[16] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer. Graceful degradation of cooperative adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):488–497, Feb 2015. ISSN 1524-9050. doi: 10.1109/TITS.2014.2349498.

[17] Z Abdollahi Biron and P Pisu. Sensor and actuator fault detection in connected vehicles under a packet dropping network. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(6):1114–1120, 2016.

[18] Hao Zhou, Romesh Saigal, Francois Dion, and Li Yang. Vehicle platoon control in high-latency wireless communications environment: Model predictive control method. *Transportation Research Record*, 2324(1):81–90, 2012. doi: 10.3141/2324-10. URL `https://doi.org/10.3141/2324-10`.

[19] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick. Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Transactions on Control Systems Technology*, 25(3):899–910, May 2017. ISSN 1063-6536. doi: 10.1109/TCST.2016.2594588.

[20] C. Huang, F. Naghdy, and H. Du. Model predictive control-based lane change control system for an autonomous vehicle. In *2016 IEEE Region 10 Conference (TENCON)*, pages 3349–3354, Nov 2016. doi: 10.1109/TENCON.2016.7848673.

[21] Zoleikha Abdollahi Biron and Pierluigi Pisu. Distributed fault detection and estimation for cooperative adaptive cruise control system in a platoon. 10 2015.

[22] Abdeldime M. S. Abdelgader and Wu Lenan. The physical layer of the ieee 802.11p wave communication standard: The specifications and challenges. 2014.

[23] Ieee standard for wireless access in vehicular environments (wave) – networking services. *IEEE Std 1609.3-2016 (Revision of IEEE Std 1609.3-2010)*, pages 1–160, April 2016. doi: 10.1109/IEEESTD.2016.7458115.

[24] A. F. M. S. Shah and N. Mustari. Modeling and performance analysis of the ieee 802.11p enhanced distributed channel access function for vehicular network. In *2016 Future Technologies Conference (FTC)*, pages 173–178, Dec 2016. doi: 10.1109/FTC.2016.7821607.

[25] Q. Wu and J. Zheng. Performance modeling of the ieee 802.11p edca mechanism for vanet. In *2014 IEEE Global Communications Conference*, pages 57–63, Dec 2014. doi: 10.1109/GLOCOM.2014.7036784.

[26] Illa Ul Rasool, Yousaf Bin Zikria, and Sung Won Kim. A review of wireless access vehicular environment multichannel operational medium access control protocols: Quality-of-service analysis and other related issues. *International Journal of Distributed Sensor Networks*, 13(5):1550147717710174, 2017. doi: 10.1177/1550147717710174.

[27] Caixia Song. Performance analysis of the ieee 802.11p multichannel mac protocol in vehicular ad hoc networks. *Sensors*, 17(12):2890, Dec 2017. ISSN 1424-8220. doi: 10.3390/s17122890. URL `http://dx.doi.org/10.3390/s17122890`.

[28] Mohamed Hadded, Paul Muhlethaler, Anis Laouiti, Rachid Zagrouba, and Leila Azouz Saidane. TDMA-based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis and Open Research Issues. *Communications Surveys and Tutorials, IEEE Communications Society*, 2015. doi: 10.1109/COMST.2015.2440374. URL `https://hal.archives-ouvertes.fr/hal-01211437`.

[29] Wang Xiaonan and Qian Huanyan. Mobility management solution for ipv6-based vehicular networks. *Computer Standards Interfaces*, 36(1):66 – 75, 2013. ISSN 0920-5489. doi: https://doi.org/10.1016/j.csi.2013.07.013. URL `http://www.sciencedirect.com/science/article/pii/S0920548913000792`.

[30] Cellular cv2x vs dsrc. `https://www.autonews.com/mobility-report/new-connected-car-battle-cellular-vs-dsrc`. Accessed: 2018-5-1.

[31] M. Boban, K. Manolakis, M. Ibrahim, S. Bazzi, and W. Xu. Design aspects for 5g v2x physical layer. In *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–7, Oct 2016. doi: 10.1109/CSCN.2016.7785161.

[32] R. Molina-Masegosa and J. Gozalvez. Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications. *IEEE Vehicular Technology Magazine*, 12(4):30–39, Dec 2017. ISSN 1556-6072. doi: 10.1109/MVT.2017.2752798.

[33] Liang Li, G. Lu, Yunpeng Wang, and Daxin Tian. A rear-end collision avoidance system of connected vehicles. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 63–68, Oct 2014. doi: 10.1109/ITSC.2014.6957667.

[34] Velodynes latest lidar lets driverless cars handle high-speed situations. `https://www.theverge.com/2017/11/29/16705674/velodyne-lidar-128-autonomous-vehicles-driverless-cars`. Accessed: 2018-2-19.

[35] Kakan Chandra Dey, Anjan Rayamajhi, Mashrur Chowdhury, Parth Bhavsar, and James Martin. Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network performance evaluation. *Transportation Research Part C: Emerging Technologies*, 68:168 – 184, 2016. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2016.03.008. URL `http://www.sciencedirect.com/science/article/pii/S0968090X16300018`.

[36] G. Naus, R. Vugts, J. Ploeg, R. v. d. Molengraft, and M. Steinbuch. Cooperative adaptive cruise control, design and experiments. In *Proceedings of the 2010 American Control Conference*, pages 6145–6150, June 2010. doi: 10.1109/ACC.2010.5531596.

[37] Jeroen Ploeg, Nathan Van De Wouw, and Henk Nijmeijer. Lp string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, 22(2):786–793, 2014.

[38] V. Milans, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):296–305, Feb 2014. ISSN 1524-9050. doi: 10.1109/TITS.2013.2278494.

[39] C. Lei, E. M. van Eenennaam, W. K. Wolterink, G. Karagiannis, G. Heijenk, and J. Ploeg. Impact of packet loss on cacc string stability performance. In *2011 11th International Conference on ITS Telecommunications*, pages 381–386, Aug 2011. doi: 10.1109/ITST.2011.6060086.

[40] Nowakowski C. Lu X. Shladover, S. E. Using cooperative adaptive cruise control (cacc)to form high-performance vehicle streams. definitions, literature review and operational concept alternatives. *UC Berkeley: California Partners for Advanced Transportation Technology*. URL https://escholarship.org/uc/item/3w6920wz.

[41] Anjan Rayamajhi, Zoleikha Abdollahi Biron, Roberto Merco, Pierluigi Pisu, James M Westall, and Jim Martin. The impact of dedicated short range communication on cooperative adaptive cruise control. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

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

[43] Vicente Milanés and Steven E Shladover. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transportation Research Part C: Emerging Technologies*, 48:285–300, 2014.

[44] Song Gao, Alvin Lim, and David Bevly. An empirical study of DSRC v2v performance in truck platooning scenarios. *Digital Communications and Networks*, 2(4):233–244, 2016.

[45] Jennie Lioris, Ramtin Pedarsani, Fatma Yildiz Tascikaraoglu, and Pravin Varaiya. Platoons of connected vehicles can double throughput in urban roads. *Transportation Research Part C: Emerging Technologies*, 77:292 – 305, 2017. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2017.01.023. URL http://www.sciencedirect.com/science/article/pii/S0968090X17300384.

[46] Y. Abou Harfouch, S. Yuan, and S. Baldi. An adaptive switched control approach to heterogeneous platooning with inter-vehicle communication losses. *IEEE Transactions on Control of Network Systems*, pages 1–1, 2018. ISSN 2325-5870. doi: 10.1109/TCNS.2017.2718359.

[47] P. Wang, C. Jiang, X. Deng, L. Wang, H. Deng, and Z. He. A multi-mode cooperative adaptive cruise switching control model for connected vehicles considering abnormal communication. In *2017 6th Data Driven Control and Learning Systems (DDCLS)*, pages 739–744, May 2017. doi: 10.1109/DDCLS.2017.8068165.

[48] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, June 2011. doi: 10.1109/IVS.2011.5940562.

[49] H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal. Moving obstacle avoidance for large, high-speed autonomous ground vehicles. In *2017 American Control Conference (ACC)*, pages 5568–5573, May 2017. doi: 10.23919/ACC.2017.7963821.

[50] J. Siegel, D. Erb, and S. Sarma. Algorithms and architectures: A case study in when, where and how to connect vehicles. *IEEE Intelligent Transportation Systems Magazine*, 10(1):74–87, Spring 2018. ISSN 1939-1390. doi: 10.1109/MITS.2017.2776142.

[51] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes. Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. *IEEE Transactions on Control Systems Technology*, 25(4):1204–1216, July 2017. ISSN 1063-6536. doi: 10.1109/TCST.2016.2599783.

[52] Shih-Chieh Lin, Yunqi Zhang, Chang-Hong Hsu, Matt Skach, Md E. Haque, Lingjia Tang, and Jason Mars. The architectural implications of autonomous driving: Constraints and acceleration. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '18, pages 751–766, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-4911-6. doi: 10.1145/3173162.3173191. URL `http://doi.acm.org/10.1145/3173162.3173191`.

[53] Klaus Schmidt. Cooperative adaptive cruise control for vehicle following during lane changes * *this work was supported by the scientific and technological research council of turkey (tubitak) [award 115e372]. 50:12582–12587, 07 2017.

[54] E. Semsar-Kazerooni, K. Elferink, J. Ploeg, and H. Nijmeijer. Multi-objective platoon maneuvering using artificial potential fields. *IFAC-PapersOnLine*, 50(1): 15006–15011, 7 2017. ISSN 2405-8963. doi: 10.1016/j.ifacol.2017.08.2570.

[55] Kamonthep Tiaprasert, Yunlong Zhang, and Xin Ye. Platoon recognition using connected vehicle technology. *Journal of Intelligent Transportation Systems*, 0 (0):1–16, 2018. doi: 10.1080/15472450.2018.1476146. URL `https://doi.org/10.1080/15472450.2018.1476146`.

[56] Stephan Eichler. Performance evaluation of the IEEE 802.11 p WAVE communication standard. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2199–2203. IEEE, 2007.

[57] Anjan Rayamajhi et al. Things in a fog: System illustration with connected vehicles. In *Vehicular Technology Conference, 2017. VTC-2017 Spring. IEEE 85th (In Print)*.

[58] Network simulator 3. `https:www.nsnam.org`. Accessed: 2018-2-19.

[59] J. Benin, M. Nowatkowski, and H. Owen. Vehicular network simulation propagation loss model parameter standardization in ns-3 and beyond. In *2012 Proceedings of IEEE Southeastcon*, pages 1–5, March 2012. doi: 10.1109/SECon.2012.6196929.

[60] Mani Amoozadeh, Hui Deng, Chen-Nee Chuah, H Michael Zhang, and Dipak Ghosal. Platoon management with cooperative adaptive cruise control enabled by vanet. *Vehicular communications*, 2(2):110–123, 2015.

[61] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer US, 2011. ISBN 9781461414339. URL `https://books.google.com/books?id=cZJFDox4KuUC`.

[62] California PATH Program. Cooperative adaptive cruise control. URL `https://path.berkeley.edu/research/connected-and-automated-vehicles/cooperative-adaptive-cruise-control`. Accessed: 2018-8-10.

[63] Soon Yong Jeong, Yun Jong Choi, PooGyeon Park, and Seung Gap Choi. Jerk limited velocity profile generation for high speed industrial robot trajectories. *IFAC Proceedings Volumes*, 38(1):595 – 600, 2005. ISSN 1474-6670. doi: https://doi.org/10.3182/20050703-6-CZ-1902.01369. URL `http://www.sciencedirect.com/science/article/pii/S1474667016373815`. 16th IFAC World Congress.

[64] Emission Test Cycles. US06 supplemental federal test procedure (sftp). URL `https://www.dieselnet.com/standards/cycles/ftp_us06.php`. Accessed: 2018-8-10.

[65] Fabian De Ponte Muller. Survey on ranging sensors and cooperative techniques for relative positioning of vehicles. *Sensors (Basel, Switzerland)*, 17:2, August 2018.

[66] Delphi esr 2.5 radar. `https://autonomoustuff.com/product/delphi-esr-2-5-24v/`. Accessed: 2018-4-15.

[67] Velodyne lidar. `https://www.velodynelidar.com/index.html`. Accessed: 2018-4-15.

[68] Peloton tech truck platooning system. `https://peloton-tech.com/how-it-works/`. Accessed: 2018-4-15.

[69] Wenhua Song, JianCheng Lai, Zabih Ghassemlooy, Zhiyong Gu, Wei Yan, Chunyong Wang, and Zhenhua Li. The effect of fog on the probability density distribution of the ranging data of imaging laser radar. *AIP Advances*, 8(2):025022, 2018. doi: 10.1063/1.5011781. URL `https://doi.org/10.1063/1.5011781`.

[70] P. Tu and J. Kiang. Estimation on location, velocity, and acceleration with high precision for collision avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):374–379, June 2010. ISSN 1524-9050. doi: 10.1109/TITS.2010.2043098.

[71] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D): 35–45, 1960.