Evaluation and improvement of decentralized congestion control for multiplatooning application

Bai, Chumeng

Award date:
2018

Awarding institution:
Royal Institute of Technology

Evaluation and Improvement of Decentralized Congestion Control for Multiplatooning Application

Chumeng Bai

2018-08-01

Master's Thesis

Examiner
Gerald Q. Maguire Jr.

Academic adviser
Anders Västberg

KTH Royal Institute of Technology
School of Electrical Engineering and Computer Science
Department of ICT Innovation
SE-100 44 Stockholm, Sweden

Abstract

Platooning has the potential to be a breakthrough in increasing road capacity and reducing fuel consumption, as it allows a chain of vehicles to closely follow each other on the road. When the number of vehicles increases, platoons will follow one another in what is referred to as multiplatooning. Many Cooperative Intelligent Transportation Systems (C-ITS) applications rely on periodically exchanged beacons among vehicles to improve traffic safety. However, as the number of connected vehicles increases, the network may become congested due to periodically exchanged beacons. Therefore, without some congestion control method, safety critical messages such as Cooperative Awareness Messages (CAMs) may not be delivered on time in high vehicle density scenarios. Both the European Telecommunications Standards Institute (ETSI) and the Institute of Electrical and Electronics Engineers (IEEE) have been working on different standards to support vehicular communication. ETSI defined the Decentralized Congestion Control (DCC) mechanism which adapts transmission parameters (message rate, transmit data rate, and transmit power, etc.) to keep channel load under control. ETSI DCC utilizes a three-state machine with RELAXED, ACTIVE, and RESTRICTIVE states. In this thesis, we implemented this three-state machine by adapting the message rate based on the channel busy ratio (CBR). We name this message-rate based three-state machine DCC-3. DCC-3 has the ability to control channel load; however, it has unfairness and instability problems due to the dramatic parameter changes between states. Therefore, we divided the ACTIVE state of DCC-3 into five sub-states, and refer to this as DCC-7. We benchmarked DCC-3 against static beaconing (STB), dynamic beaconing (DynB), LInear MEssage Rate Integrated Control (LIMERIC), and DCC-7 using different evaluation metrics with different numbers of platoons. Our results from the Plexe simulator demonstrate that DCC-7 has the best performance when considering all evaluation metrics, including CBR, Inter-reception time (IRT), collisions, safe time ratio, and fairness. Furthermore, we found using transmit power control could greatly improve the performance of CBR and collision rates.

keywords: Wireless vehicular communication, decentralized congestion control, cooperative awareness messages, multiplatooning, simulation

Sammanfattning

Platooning (fordonskonvojer) har potential att bli ett genombrott i öka vägkapaciteten och minska bränsleförbrukning, eftersom det tillåter en kedja av fordon att noga följa varandra på vägen. När antalet fordon ökar, kommer att plutoner följa varandra i vad som benämns multiplatooning (konvoj av fordonskonvojer). Många kooperativ intelligenta transportsystem (C-ITS) tillämpningar förlitar sig på regelbundet utbytte beacons bland fordon att förbättra trafiksäkerheten. Dock som antalet uppkopplade fordon ökar, kan nätverket bli överbelastat på grund av regelbundet utbytte beacons. Utan någon trängsel kontrollmetod, får därför säkerhet kritiska meddelanden såsom kooperativ medvetenhet meddelanden (CAMs) inte levereras i tid i höga fordon densitet scenarier. Både Europeiska institutet för telekommunikationsstandarder (ETSI) och Institute el och elektroniska tekniker (IEEE) har arbetat på olika standarder för att stödja vehicular kommunikation. ETSI definieras den decentraliserade överbelastning kontroll (DCC) mekanism som anpassar överföring parametrar (meddelande hastighet, överföra datahastighet och sändningseffekt, etc.) för att hålla kanalen belastningen under kontroll. ETSI DCC använder en tre-state maskin med RELAXED, ACTIVE och RESTRICTIVE stater. I denna avhandling har genomfört vi denna tre-state maskin genom att anpassa meddelande hastighet baserat på kanal upptagen förhållandet (CBR). Vi nämna detta meddelande-hastighet baserat tre-state machine DCC-3. DCC-3 har förmågan att kontrollera kanal belastning; Det har dock otillbörlighet och instabilitet problem på grund av de dramatiska parameterändringar mellan stater. Därför vi indelat det ACTIVE tillståndet för DCC-3 i fem undertillstånd och hänvisar till detta som DCC-7. Vi benchmarkade DCC-3 mot statiska leda (STB), dynamisk leda (DynB), linjära MEssage Rate integrerad kontroll (LIMERIC) och DCC-7 med olika utvärdering statistik med olika antal plutoner. Våra resultat från Plexe simulator visar att DCC-7 har bästa prestanda när man överväger alla utvärdering statistik, inklusive CBR, mellan receptionen tid (IRT), kollisioner, säker tid baserat och rättvisa. Vi fann dessutom använda Sändareffektstyrning kan avsevärt förbättra prestanda för CBR och kollision priser.

Nyckelord: Trådlös fordonskommunikation, decentraliserad överbelastning kontroll, kooperativ medvetenhet meddelanden, multiplatooning, simulering

Acknowledgement

First of all, I would like to express my deepest gratitude to my supervisor, Ali Balador, for providing me with such an interesting research topic. I sincerely appreciate his assistance and guidance through this whole project. He was always pleasant and patient to help me solve any problems. I really appreciate his valuable comments and inspiring ideas in every meeting we have. Especially, I want to say thank you to my examiner, Gerald Q. Maguire Jr., for guiding and educating me. I really appreciate his patience in this process from the very beginning. He guided me to write a better master thesis and helped me improve my scientific writing skills.

I am sincerely grateful to RISE CISE for providing me with such a comfortable working environment, everyone is so nice and helpful at here.

Furthermore, I would like to thank Michele Segata for his assistant in helping me solve the problems with the simulation software Plexe. He also guided me how to start implementing codes in this software. I really appreciate his assistant and time.

Especially, I want to express my special gratitude to my teammate Zijie Liang. We always share our ideas and I can always get inspirations from him. His companion and encouragement help me spend a really happy time at RISE SICS.

Furthermore, I want to say thank you to my best friends, Qifan Dai and Xintong Wang for encouraging and helping me whenever I needed help.

Finally, I would like to express my gratitude to my parents and my boyfriend, whenever I have a hard time, they are always there listening to my complaints. Thank you for your understanding and support during my whole Master's degree project.

Stockholm, July 2018
Chumeng Bai

Contents

Abbreviations

ACC	Adaptive Cruise Control
ACK	Acknowledgement
AIFS	Arbitration Inter-Frame Space
AIFSN	Arbitration Inter-Frame Space Number
AIMD	Additive Increase Multiplicative Decrease
CACC	Cooperative Adaptive Cruise Control
CAMs	Cooperative Awareness Messages
CBI	Channel Busy Indicator
CBR	Channel Busy Ratio
CCA	Clear Channel Access
CCH	Control Channel
C-ITS	Cooperative Intelligent Transport Systems
CSMA/CA	Carrier-sense Multiple Access with Collision Avoidance
CW	Contention Window
DCC	Decentralized Congestion Control
DCC-7	Seven-state Machine DCC
DCC-3	Three-state Machine DCC
DENMs	Decentralized Environmental Notification Messages
D-FPAV	Distributed Fair Power Adjustment for Vehicular Environments
DR-DCC	Data-Rate DCC
DSC	DCC Sensitivity Control
DynB	Dynamic Beaconing
EDCA	Enhanced Distributed Channel Access
EMBARC	Error Model Based Adaptive Rate Control
ETSI	European Telecommunications Standards Institute
ETSI TC ITS	ETSI Technical Committee for Intelligent Transport Systems
IEEE	Institute of Electrical and Electronics Engineers
IRT	Inter-Reception Time
LIMERIC	LInear MEssage Rate Integrated Control
MAC	Medium Access Control
MD-DCC	Message and Data-Rate DCC
NIC	Network Interface

OFDM	Orthogonal Frequency Division Modulation
OMNeT++	Objective Modular Network Testbed in C++
PDR	Packet Delivery Ratio
PDR-DCC	Packet-count based Decentralized Data-Rate DCC
PHY	Physical
PLEXE	A Platooning Extension for Veins
PULSAR	Periodically Updated Load Sensitive Adaptive Rate
RSSI	Received Signal Strength Indicator
RTS/CTS	Request To Send/Clear To Send
SAE-DCC	Society of Automotive Engineers proposed Decentralized Congestion Control
SCHs	Service Channels
SIFS	Short Inter-Frame Space
STB	Static Beaconing
SUMO	Simulation of Urban Mobility
TAC	Transmit Access Control
TCP	Transmission Control Protocol
TPC	Transmit Power Control
TPC CLS	TPC with Proportional-integral Control Loop and Exchange of Channel Load Share Information
TPC PI	TPC with Proportional-integral Control Loop
TPRC	Transmit Power and Rate Control
TraCI	Traffic Control Interface
TRC	Transmit Rate Control
UDP	User Datagram Protocol
VANETs	Vehicular *ad hoc* Networks
VEINS	Vehicles in Network Simulation
V2V	Vehicle to Vehicle

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives an introduction to this thesis. Section 1.1 describes the background of the thesis project. Section 1.2 proposes the specific problems that this thesis address. The goals and contribution of this thesis project are summarized in Section 1.3 and Section 1.4 respectively. In Section 1.5, we briefly introduce our research methodology. Section 1.6 describes the delimitations of this thesis. Finally, an outline of the structure of the thesis report is given in Section 1.7.

1.1 Background

Although research has been conducted on Vehicular *ad hoc* Networks (VANETs), the implementation of Cooperative Intelligent Transport Systems (C-ITS) on the roads is still in its infancy. VANETs differ from other distributed wireless networks in terms of their high node mobility, highly dynamic topology, sensitivity to interference, and unpredictable radio conditions [1]. In addition, there are high requirements on timing and reliability posed by traffic safety applications, such as Cooperative Adaptive Cruise Control (CACC) and platooning, that cause difficulties for VANET communication and need to be addressed on a protocol design level [2]. VANETs based on wireless communication is a promising approach for C-ITS to reduce the number of traffic accidents and improve traffic efficiency. Many C-ITS applications rely on periodically exchanged beacons among vehicles (V2V) and between vehicles and roadside infrastructures to improve traffic safety. Both the European Telecommunications Standards Institute (ETSI) and the Institute of Electrical and Electronics Engineers (IEEE) have been working on different standards to support vehicular communication.

Road traffic congestion may cause a negative impact on both the environment and humans due to increased vehicle densities. Reducing inter-vehicle distance is an efficient method to increase road capacities [3]. Decreasing human reaction time is needed to reduce the inter-vehicle spacing, but since this is not possible the alternative is to use automated vehicles to increase the density of vehicles on the road. Platooning is a promising application to increase road capacity. One method of platooning is based on the vehicular communication. In this approach, a platoon is formed by a string of vehicles communicating with each other to maintain a desired intra-platoon distance. Reducing the platoon's intra-vehicle distance, reduces air-drag while increasing road capacity and reducing fuel consumptions [4]. In real traffic scenarios, as the number of vehicles increases, multiplatooning, a chain of platoons following each other is likely to happen [5].

1.1.1 IEEE and ETSI standards

The ETSI has standardized a profile of IEEE 802.11p adapted to 30 MHz of the frequency spectrum in the 5.9 GHz band allocated in Europe for vehicular communications. This is further divided into one control channel (CCH) and two service channels (SCHs). IEEE 802.11p [6], an amendment for Vehicle-to-everything communication to the IEEE 802.11 standard, defines specifications for the

physical (PHY) layer and the Medium Access Control (MAC) layer. IEEE 802.11p uses carrier-sense multiple access with collision avoidance (CSMA/CA) as the MAC layer method with support for quality of service.

Despite the built-in mechanisms of the CSMA/CA MAC protocol to prevent packet collisions, such as listen-before-talk and back-off mechanisms, packets may still collide and this can lead to unbounded channel access delays, especially under heavy channel load conditions. Many experiments have shown that the safety-critical applications, especially the platooning application, requires a regular beacon update frequency rate of 10 Hz [7]. Also, when safety messages are transmitted in the broadcast mode, no acknowledgement (ACK) messages or Request To Send/Clear To Send (RTS/CTS) are transmitted; thus it is uncertain that there has been a successful packet reception at the receiver. For this reason, the basic IEEE 802.11p MAC protocol is unable to meet the delay and reliability requirements of safety-critical applications, such as the platooning application. For this reason, ETSI has extended this protocol as described in the next section.

1.1.2 CAM and DENM

Typical C-ITS safety applications rely on the exchange of two basic types of message: periodic status updates and event-triggered hazard warnings. Therefore, ETSI ITS-G5 has standardized the Cooperative Awareness Basic Service and defined Cooperative Awareness Messages (CAMs) and Decentralized Event Notification Messages (DENMs). CAMs include status updates about a vehicle's position and speed, or other in-vehicle sensor data. CAMs can be periodically broadcast to provide input to a multitude of safety and efficiency applications, such as intersection collision warnings or overtaking warnings [8]. CAMs were generated periodically every 0.1 seconds and delivered to single-hop neighbors within an IEEE 802.11p network. At the same time, C-ITS systems require DENMs triggered by unforeseen events (e.g. a sudden deceleration through sudden breaking). The rate of DENMs generation varies from 1 Hz to 20 Hz. If necessary, DENMs can be broadcast to neighbors via multi-hop. More details of DENMs can be found in [9]. Both CAMs and DENMs carry highly time-critical data, which impose high demands on the timing and reliability of the underlying communication protocols.

On September 2014, the ETSI Technical Committee for Intelligent Transport Systems (ETSI TC ITS) realized the standard ETSI EN 302 637-2 and changed its strategy from periodic generation of status updates, to using triggering rules for beacon message generation in order to mitigate the unnecessary load on the common control channel. According to these rules, a CAM message will be triggered based on vehicle movements, with a frequency between 1 and 10 Hz. Next, the standard ETSI EN 302 637-2 is explained.

1.1.3 ETSI EN 302 637-2

The process of triggering CAMs can be described as follows: first, the time between two consecutive generated CAMs is controlled by T_{min} and T_{max}, where:

(1) $T_{min} = T_{GenCamMin} = 100$ ms, which is an upper bound corresponding to the maximum CAM generation rate of 10 Hz. As a result the time between two consecutive CAMs shall not be less than T_{min}; and

(2) $T_{max} = T_{GenCamMax} = 1000$ ms, which is an upper bound corresponding to the minimum CAM generation rate of 1 Hz. As a result the time between two consecutive CAMs shall not be larger than T_{max}.

A vehicle's kinematics should be checked periodically ($T_{CheckCamGen}$). This period should be less than or equal to $T_{GenCamMin}$. Each vehicle checks the deviation of its current speed, position, and direction from the measurements in the last triggered CAM, then a new CAM should be triggered if one of the following deviations has been observed:

(1) The time elapsed since the last CAM generation is greater than or equal to $T_{GenCamMax}$; or

(2) The time elapsed since the last CAM generation is greater than or equal to $T_{GenCamMin}$ and one or more of the following is true:

   (a) the absolute difference between the current heading direction and the heading direction in the previous CAM exceeds 4°;

   (b) the absolute difference between the current position and the position included in the previous CAM exceeds 4 m; or

   (c) the absolute difference between the current speed and the speed included in the previous CAM exceeds 0.5 m/s.

1.2 Problem Statement

The safety of vehicular communications in C-ITS is supported by periodically exchanged beacons informing neighbors of their vehicle kinematics. The static beaconing (STB) protocol was originally designed by ETSI to support safety applications by periodically broadcasting beacons every 0.1 seconds. However, in a high traffic density scenario, periodically broadcast beacons cause an overload of the communication channel due to the limited channel bandwidth. When the channel is saturated, the MAC layer may have a high number of packet collisions which leads to a high packet loss rate [10]. Therefore, the time critical messages may not be delivered in time, thus causing the communication among vehicles to become unreliable. The IEEE 802.11p MAC method is based on CSMA/CA where nodes listen to the channel before sending messages. As a result a node should defer its transmission when it detects the channel is busy. However, this may lead to unbounded transmission delay [11]. IEEE 802.11p as the standard MAC protocol has a high rate of packet collisions when the messages are broadcast (especially in the case of multi-hop) [12].

Keeping channel congestion under control reduces packet collision rates, hence this is of vital importance to guarantee the reliable and timely exchange of CAMs. Different methods attempt to keep channel load under control by adapting different transmission parameters. Congestion control can be either centralized or decentralized [13]. In the centralized approach, a central node decides upon transmission parameters for all the nodes in its communication range; whereas, in the decentralized approach, each node employs its own congestion control algorithm. However, since the communication overhead from a central node is possibly that from neighbor nodes, the delay requirement of traffic safety applications may not be satisfied [14]. Therefore, a decentralized congestion control method is preferred in VANETs for real-time applications.

ETSI proposed a framework for decentralized congestion control (DCC) which allows C-ITS nodes to adapt their transmission parameters to keep channel load under a pre-defined threshold. The central element of ETSI DCC is a three-state machine, whose states are RELAXED, ACTIVE, and RESTRICTIVE. There are many other DCC algorithms that adapt different transmission parameters, but the effectiveness of keeping the channel load below a target threshold is of vital importance for all methods. This is because too high an unsuitable channel load can lead to higher probability of packets transmission [15]. However, while maintaining the channel load under a target threshold, the reliability of safety application (in this thesis especially the safety of platooning) may not be satisfied. In terms of message rate, real-time applications require a beacon frequency of 10 Hz. However, DCC may lower the message rate to keep channel load under the threshold, hence this may violate the timing requirement of safety applications. Moreover, ETSI DCC has proven to have unfairness and instability problems due to unsuitable parameter settings [8, 16].

A platoon using a CACC controller needs to maintain the desired intra-platoon distance by receiving messages containing speed and acceleration through wireless communication from the lead vehicle and its immediate preceding vehicle on time. Michele Segata, et al. mentioned that ETSI DCC may be unsuitable for the platooning application under a harsh braking scenario where

vehicles may crash due to a high packet inter-reception time (IRT), when there is no LIDAR or RADAR safety system connected to the vehicle's braking system [17].

Based on our best knowledge, there are only a few papers evaluating DCC algorithms for the platooning application (these papers are summarized in Section 2.6 starting on page 14.). Moreover, the simulation scenarios that have been investigated in these earlier works are limited to constant vehicle speed, fixed acceleration, and fixed deceleration. Therefore, the performance of DCC under other simulation scenarios still needs to be studied. Also, in these papers only large numbers of platoons are considered; however, the number of platoons needs to vary within communication range and interference range. Since those platoons outside of the interference range neither interfere nor can they communicate with other platoons. Therefore, the behavior of nodes in the midst of enough platoons before and after it to occupy both the communication and interference ranges needs to be observed.

1.3 Goals

In this thesis, we aim to analyze and evaluate the performance and effectiveness of DCC protocols for the multiplatooning application with different numbers of platoons by using different evaluation metrics. Moreover, we aim to find whether the performance of ETSI DCC will be improved when adding more states to the three-state machine or when modifying state machine's parameter settings.

1.4 Contribution

The main contributions of this thesis are:

1) We implemented a three-state machine DCC (DCC-3)[1], seven-state machine DCC (DCC-7), Dynamic Beaconing (DynB), and LInear MEssage Rate Integrated Control (LIMERIC) protocols based on adjusting the message rate for the multiplatooning application.

2) We compared the performance of these protocols together with the baseline protocol STB using evaluation metrics of channel busy ratio (CBR), IRT, number of collisions, safe time ratio[2], and fairness with different numbers of platoons. The simulation scenario has the platoon leader accelerating and decelerating in a sinusoidal pattern (the sinusoidal pattern is summarized in Section 5.3 on page 31).

3) We reduced the transmit power of the intra-platoon communication and observed the performance of vehicles experiencing the maximum interference with the maximum number of platoons within communication range (the communication and interference range are summarized in Section 5.1 on page 29).

1.5 Research Methodology

Firstly, we implement the DCC-3 protocol using C++ in the OMNeT++ (Objective Modular Network Testbed in C++) simulator. Next, we conduct a simulation using the Plexe (A Platooning Extension for Veins) simulator to observe the effectiveness of DCC-3 in maintaining the channel load under control. Also, the unfairness problem of DCC-3 is analyzed. Then we implement several other congestion control protocols in the OMNeT++ simulator, namely, DynB, LIMERIC,

[1] In the rest of the thesis, we use DCC to represent all decentralized congestion control protocols. ETSI DCC refers to the three-state machine DCC protocol proposed by ETSI with various transmission parameter adaptations. DCC-3 and DCC-7 are the three-state machine and the seven-state machine DCC protocols implemented in this thesis by adapting the message rate.

[2] Safe time ratio is an application-level reliability metric. It is defined as the ratio of messages' delay smaller than the maximum allowable delay over all message delays.

and DCC-7 for comparison with DCC-3. A number of evaluation metrics were used to benchmark the performance of these different congestion control protocols. Moreover, we use transmit power adaption to analyze the behavior of vehicles in the middle of the platoon and who are also in the middle platoon of all the platoons which experience the maximum interference. Finally, we use Python to analyze the simulation results.

The detailed research methodology will be presented in Chapter 3.

1.6 Delimitations

We limited the simulation to no more than 80 platoons since such a large number of platoons is already unreasonable on a highway of length 10 km. Moreover, due to limited duration of this Master's project, the simulation scenario is limited to a sinusoidal acceleration/deceleration scenario of the platoon leader. Furthermore, we set the same transmit power for all vehicles in all platoons without adapting different transmit power for a platoon leader and its followers.

1.7 Thesis structure

The rest of the thesis is structured as follows: The literature review and the state-of-art are summarized in Chapter 2. The detailed research methods are described in Chapter 3. The implementation of congestion control algorithms is presented in Chapter 4. Chapter 5 summarizes the parameter settings of traffic, network, controller, DCC algorithms, etc. In Chapter 6, different evaluation metrics and methods of evaluating performance are presented followed by analysis of the results of the simulation. Future work will be suggested along with a statement of conclusions in Chapter 7.

Chapter 2

Background

This chapter describes the background information needed to carry out and to read this thesis. Section 2.1 introduces the platooning controllers: Adaptive Cruise Control (ACC) and CACC. The basic IEEE 802.11p protocol is introduced in Section 2.2. We mention the baseline STB protocol in Section 2.3 and the ETSI DCC architecture is introduced in Section 2.4. In Section 2.5, we introduce the evaluation metrics. Finally, the state of art is summarized in Section 2.6.

2.1 Platooning controller

Most popular platooning controllers are based on ACC or CACC. Previous research has shown that CACC outperforms ACC in terms of traffic efficiency due to a smaller inter-vehicle distance that can be realized with CACC in comparison with ACC. Furthermore, CACC performs better in terms of maintaining the required intra-vehicle spacing within the platoon [3].

2.1.1 ACC

In ACC, the vehicle uses RADAR or LIDAR to detect obstacles and to follow the velocity profile of its preceding vehicle in order to maintain a safe inter-vehicle distance. A string of vehicles with an ACC controller in one dimension is shown in Figure 2.1. In this figure, l and $\dot{x}$ represent the length and speed of a vehicle respectively. x is the inter-vehicle distance.

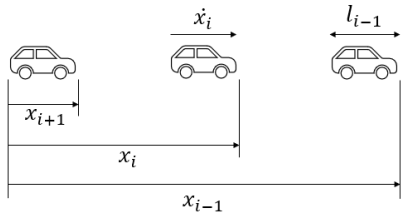


Figure 2.1: String of ACC control vehicles

The control equations for ACC are:

$$u_i = -\frac{1}{T}(\dot{\epsilon}_i + \lambda\delta_i) \tag{2.1}$$

$$\delta_i = x_i - x_{i-1} + l_{i-1} + T\dot{x}_i \tag{2.2}$$

$$\dot{\epsilon}_i = \dot{x}_i - \dot{x}_{i-1} \tag{2.3}$$

In equations 2.1 - 2.3, i and i-1 represent the vehicle i and its preceding vehicle respectively. u_i is the desired acceleration of vehicle i, $\dot{\epsilon}_i$ is the relative difference in speed between vehicle i and

i-1. $\dot{x}_i$ and $\dot{x}_{i-1}$ are the speeds of vehicle i and i-1 respectively. δ_i is the spacing error, i.e., the difference between the actual inter-vehicle distance and the desired distance. T represents the headway time which is the time necessary for vehicle i to travel the distance to its predecessor. T is usually set to be larger than 2 s for safety and reliability reasons, as ACC sensors and actuators need time for sensing and actuating [18]. Since a vehicle can only measure the distance to the preceding vehicle, ACC has difficulty adapting to rapidly changing traffic conditions.

2.1.2 CACC

ACC can be extended to support the platooning application. This extension is known as CACC. In contrast to ACC, CACC uses both RADAR and wireless communication to get information from the platoon leader and each vehicle's preceding vehicle. With intra-vehicle communication, CACC outperforms ACC with smaller headway time [19]. A string of vehicles with a CACC controller is shown in Figure 2.2.

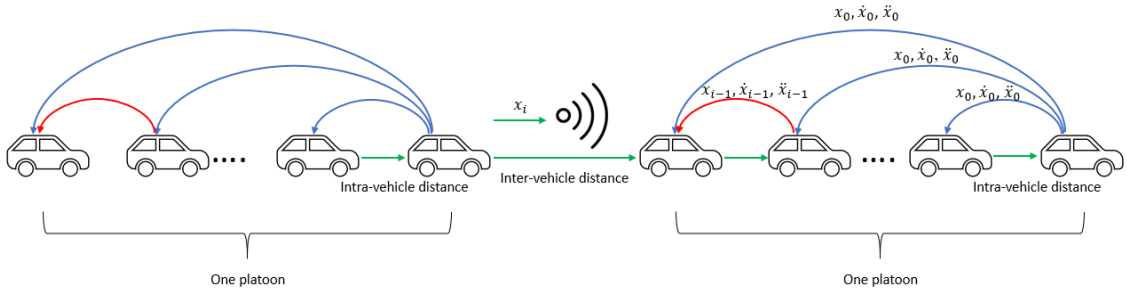


Figure 2.2: String of CACC control vehicles

The distance between a vehicle and its preceding vehicle is obtained by a radar, while other parameters are accessed through the wireless communication. The control equations of CACC are:

$$\dot{x}_i[n] = \frac{x_i[n] - x_i[n-1]}{\Delta t} \tag{2.4}$$

$$\ddot{x}_i[n] = \frac{\dot{x}_i[n] - \dot{x}_i[n-1]}{\Delta t} \tag{2.5}$$

$$\epsilon_i[n] = x_i[n] - x_{i-1}[n] + l_{i-1} + gap_{des} \tag{2.6}$$

$$\dot{\epsilon}_i[n] = \dot{x}_i[n] - \dot{x}_{i-1}[n] \tag{2.7}$$

$$u_i[n] = \alpha_1 u_{i-1}[n] + \alpha_2 u_0[n] + \alpha_3 \dot{\epsilon}_i[n] + \alpha_4(\dot{x}_i[n] - \dot{x}_0[n]) + \alpha_5 \epsilon_i[n] \tag{2.8}$$

$$\ddot{x}_i[n] = \beta u_i[n] + (1-\beta)\ddot{x}_i[n-1] \tag{2.9}$$

$$\beta = \frac{\Delta_t}{\tau + \Delta_t} \tag{2.10}$$

$$\alpha_1 = 1 - C1 \tag{2.11}$$

$$\alpha_2 = C1 \tag{2.12}$$

$$\alpha_3 = -(2\xi - C1(\xi + \sqrt{\xi^2 - 1}))w_n \tag{2.13}$$

$$\alpha_4 = -C1(\xi + \sqrt{\xi^2 - 1})w_n \tag{2.14}$$

$$\alpha_5 = -w_n^2 \tag{2.15}$$

In equations 2.4 - 2.15, Δt is the sampling time interval, n is the current sample's discrete time. ϵ_i represents spacing error and gap_{des} is the desired intra-vehicle distance which is usually set to 5 m for platooning application for cars. Each vehicle i knows the position x_{i-1}, the speed

$\dot{x}_{i-1}$, and the acceleration $\ddot{x}_{i-1}$ of its preceding vehicle. u_i is the desired acceleration used by the CACC controller to maintain the platoon's desired intra-vehicle distance. Due to an actuation lag (represented by τ) caused by the mechanical components, the real acceleration (represented by $\ddot{x}_i$) differs from u_i [17]. C1 is a weighting factor for the acceleration between a vehicle's preceding vehicle and the leader, which is set to 0.5 to achieve an equal weight between the speeds of the preceding vehicle and the leader, ξ is a damping factor, and w_n is the bandwidth of the controller.

When a vehicle is more than 20 m far away from its predecessor, the desired acceleration will be calculated by ACC, otherwise, it will be calculated by CACC [18]. Therefore, the inter-vehicle distance between platoons is usually maintained by the ACC controller, while the intra-vehicle distance within a platoon is controlled by the CACC controller.

According to S. E. Shladover, et al. CACC could significantly improve traffic capacity and reduce fuel consumption [20]. The headway time of CACC could be quite small, leading to a higher traffic throughput than ACC. Tightly following vehicles could significantly reduce their air resistance, hence reduce fuel consumption, which in turn reduces the emission of carbon dioxide by fossil fueled vehicles.

In the next section, the IEEE 802.11p protocol will be introduced.

2.2 IEEE 802.11p Protocol

Vehicular communication typically utilizes the IEEE 802.11p standard, as this standard is considered to be the main candidate for VANETs. This section explains the IEEE 802.11p PHY and MAC layers. IEEE 802.11p is an amendment to the IEEE 802.11a. IEEE 802.11a was initially devised for indoor communication.

2.2.1 Frequency band

IEEE 802.11p central frequency is 5.9 GHz according to a U.S. decision in 1999 to allocate part of the 5.9 GHz band for road safety. The decision of U.S. was replicated by the European Union in 2008.

The allocated bandwidth is 75 MHz (5.850 GHz-5.920 GHz) and it is divided into seven 10 MHz-width channels [21]. These channels consist of one CCH and six SCHs. The range of data rates is halved to 3 to 27 Mb/s from the original 6 to 54 Mb/s of IEEE 802.11a by reducing the channel bandwidth, providing greater robustness against fading. The reduction in per channel bandwidth relative to the 20 MHz of IEEE 802.11a allows for a more robust signal relative to fading and increases the tolerance for multi-path propagation [21]. The information about the channel subdivision can be found in Table 2.1 [21].

It has to be emphasized that different maximum transmission powers are allotted for the different channels. In particular channel 178, given its critical safety application, is allowed to use the highest transmit power level of 44.5 dBm. Detailed information about the IEEE 802.11p frequency band can be found in [21].

Table 2.1: Set of channels defined in IEEE 802.11p [21]

Channel number	Frequency (GHz)	Channel type	Application	Transmit power (dBm)
172	5.855-5.865	SCH	Non-safety	33
174	5.865-5.875	SCH	Non-safety	33
176	5.875-5.885	SCH	Traffic efficiency	33
178	5.885-5.895	CCH	Critical safety	44.8
180	5.895-5.905	SCH	Critical safety	23
182	5.905-5.915	SCH	Traffic efficiency	23
184	5.915-5.925	SCH	Traffic efficiency	40

2.2.2 PHY layer

The PHY is based on Orthogonal Frequency Division Modulation (OFDM), the same as IEEE 802.11a. This OFDM is based on 64-point Inverse Fast Fourier Transform, but only 48 sub-carriers for data are present, half that of IEEE 802.11a. The PHY layer is made up of two sub-layers [22]:

Physical Layer Convergence Protocol
Responsible for communicating with the MAC layer and transforming packet data arriving from the MAC layer to compose an OFDM frame.

Physical Medium Access
Interface with the physical transmission medium, performs data encoding and modulation.

2.2.3 MAC layer

The MAC layer is in charge of regulating the access of nodes to the medium. This plays a major part in guaranteeing the reliability of transmissions and low latency even in a dense and dynamically changing environment.

IEEE 802.11p implements the contention based Enhanced Distributed Channel Access (EDCA) MAC sub-layer protocol. IEEE 802.11p supports transmission rates from 3 Mb/s to 27 Mb/s at a range up to 1000 m. EDCA is based on CSMA/CA, hence the node first senses the medium. If the medium is not free for an Arbitration Inter-Frame Space (AIFS), then the node will defer its transmission by choosing a random back-off amount of time. AIFS is calculated as shown in equation 2.16.

$$AIFS = AIFSN * T_s + SIFS \tag{2.16}$$

In equation 2.16, $AIFSN$ is the Arbitration Inter-Frame Space number and T_s is the slot duration. $SIFS$ is Short Inter-Frame Space which represents the time to process a receive frame or the time to respond to a response frame. After waiting for $AIFS$ time, the channel becomes idle then the back-off procedure starts:

(1) The node selects a random back-off time inside the Contention Window (CW) interval [0, CW +1] where, on the first attempt, the initial value for CW is CW_{min};

(2) For CW, if the medium is sensed busy, the counter is frozen at its current value;

(3) Once the channel is idle, then the node waits for $AIFS$ and then the counter is resumed;

(4) Once the counter is zero, the packet is sent immediately;

(5) The interval size of CW is doubled if the transmission fails;

(6) The procedure repeats until the value CW_{max} is reached; and

(7) If the transmission limit is reached then the frame is dropped and the procedure is restarted from the beginning for the next frame.

An example of two nodes accessing the channel under CSMA/CA protocol is shown in Figure 2.3. Once node 1 begins its transmission, node 2 freezes its timer and waits AIFS then resumes its timer after node 1 completes its transmission. Once the timer of node 2 expires, node 2 begins its transmission.

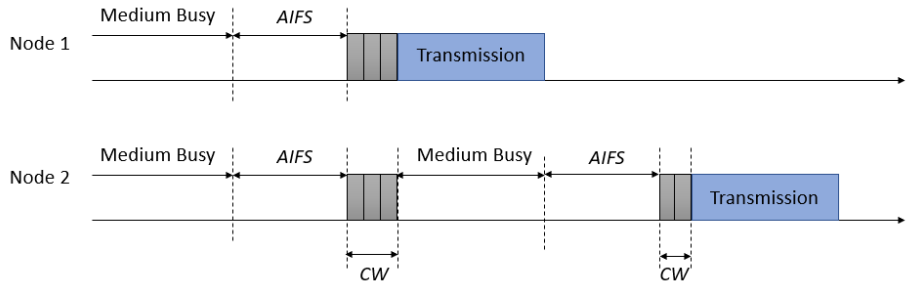


Figure 2.3: Example of nodes accessing the channel under CSMA/CA protocol

The protocol supports different priorities by implementing four different access classes, with each characterized by its AIFSN and range of CW values. The four access classes are background traffic, best effort traffic, video traffic, and voice traffic. The values of CW are listed in Table 2.2, an access class with a smaller AIFSN has a higher priority when accessing the channel.

Table 2.2: Contention parameters for different access classes [6]

Access Classes	CW_{min}	CW_{max}	$AIFSN$	$AIFS$ (μs)
Background Traffic	15	1023	9	208
Best effort Traffic	15	1023	6	148
Video Traffic	7	15	3	88
Voice Traffic	3	7	2	68

In the next section, we will introduce static beaconing which has the fixed message rate.

2.3 Static Beaconing

STB uses the IEEE 802.11p protocol based on CSMA/CA. IEEE 802.11p does not contain specific mechanisms to control channel load; therefore, the channel load increases as the number of nodes in the network increases. IEEE 802.11p focuses on achieving fair access to the communication channel for every node in the network. Therefore, nodes in the network randomly access to the channel. The message generation rate of STB is fixed to 10 Hz regardless of the channel congestion. A simplification of the STB algorithm is shown in Algorithm 1.

Algorithm 1 STB

Initialize():
  $beaconInterval = 0.1$;
  ScheduleAt ($sendBeacon$, beaconInterval);
sendBeacon():
  broadcast($platooningMsg$);
  ScheduleAt ($sendBeacon$, beaconInterval);

In Algorithm 1, the $beaconInterval$ is defined as the interval between sending two consecutive packets. Since the message generation rate is 10 Hz, the beacon interval is set to 0.1 s during the initializing phase. Therefore, the $sendBeacon$ event is scheduled every 0.1 seconds for broadcasting the platooning messages. In this Master's thesis, we slightly modified STB to have different nodes start sending beacons at a random time during the initializing phase. We use the modified STB as a baseline algorithm for comparison with other algorithms.

2.4 ETSI DCC architecture

Since our implementation and evaluation of DCC methods are based on ETSI ITS-G5 DCC, the architecture of ETSI DCC will be introduced in this section.

DCC is a cross-layer approach including one Access technology layer, one Networking and Transport layer, one Facilities layer, one Applications layer, one Security layer, and one Management layer. Different DCC components (DCC_acc, DCC_net, DCC_fac, and DCC_mgmt) are located in different layers. How DCC components are integrated into the ETSI protocol architecture for ITS is shown in Figure 2.4. These components are connected through DCC's interfaces. The specification of DCC's layers can be found in ETSI EN 302 665 [23].

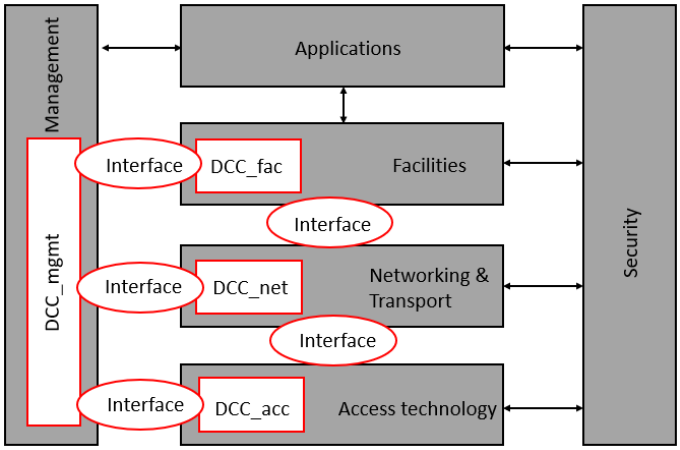


Figure 2.4: ETSI DCC architecture [24]

The important DCC layers for this Master's thesis are:

Access technology layer
This layer is also referred to as the ITS-G5 stack. This layer provides the functionality of PHY and MAC layers for C-ITS. The PHY and MAC layers are specified in the IEEE 802.11p protocol as described in Section 2.2.

Networking and Transport layer
This layer can be divided into two sub-layers: networking sublayer and transport sublayer. The networking protocols (such as the GeoNetworking protocol) are defined in the networking sublayer. The GeoNetworking protocol has the functionality to provide packet routing in an *ad hoc* network. Protocols for delivering messages such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are defined in the transport sublayer.

Facilities layer
This layer provides functions to support C-ITS applications. It handles data storage including receiving data from wireless communications or from sensors. The main function of this layer is to support the transmission of CAMs and DENMs.

Management layer
This layer ensures the messages exchanging between different layers.

The DCC components located in the above layers are:

DCC_acc
Located in the DCC Access technology layer, DCC_acc includes different DCC control methods, including transmit rate control (TRC), transmit power control (TPC), transmit data rate control, DCC access control, and DCC sensitivity control (DSC) [25].

DCC_net
Located in the Networking and Transport layer, DCC_net controls access to neighboring vehicles' information tables or exchanges transmission parameters with other vehicles.

DCC_fac
Located in the Facilities layer, DCC_fac calculates the benefits of forwarding received messages.

DCC_mgmt Located in the DCC Management layer, DCC_mgmt stores DCC background information so that other layers can access DCC state parameters through this component.

2.5 Reliability analysis for safety applications

As mentioned in the previous chapter, DCC algorithms may lower the message rate to maintain the channel load under a target threshold; but this may violate the timing requirement of safety applications. Therefore, the reliability analysis for safety applications (in this thesis especially platooning) is of vital importance for DCC algorithms. Segata et al. [17] mentioned network-level reliability metrics, including CBR and collision rate. He also proposed an application-level metric for the platooning application: safe time ratio. Bai and Krishnan [26] proposed several communication-level reliability metrics as well as application-level reliability metrics. The communication-level reliability focuses on whether packets are delivered successfully, since the number of packet collisions increases as the vehicle density increases. Application-level reliability focuses on end users, although some packets are lost in the network, the end user may not notice them or these packet loss may have little effect on them. Moreover, according to C. B. Math fairness for neighbor nodes is a new challenge for DCC [13].

2.5.1 Network-level reliability

CBR is the ratio of the time when the channel is busy to a given time interval. CBR is an important metric to increase the packet delivery ratio (PDR). According to the study by Yaser P. Fallah, et al. in [15], The PDR value reaches a peak when CBR is around 0.6 to 0.7. If CBR is lower than 0.6, there is a waste of channel resources. Conversely, for CBR larger than 0.7, ever higher packet collision rates will happen where several neighbors nodes attempt to access the channel at the same time. Another network-level metric, namely, number of packet collisions, counts the number of frames that could not be decoded in the PHY layer due to the interference [17]. Collisions happen due to more than one node starting to transmit a packet simultaneously in the same time slot. When the number of nodes increases, the number of collisions will also increase. Higher packet collision rates could lead to a lower system throughput [27].

2.5.2 Communication-level reliability

PDR is the most commonly used communication-level metric, as it represents the probability of successfully receiving packets sent by the sender. PDR is calculated as the ratio of the total number of received packets divided by the total number of packets transmitted during a given period. As PDR is an average value within a given period, even if PDR is high, the consecutive packet losses may still cause unreliability of the communication system. Therefore, the distribution of consecutive packet losses is used as another communication-level metric to represent packet drop bursts [26].

Another communication-level metric, namely, IRT, which represents the time interval between receiving two consecutive packets from the same sender [28]. IRT reflects the impact of packet collisions. In contrast, PDR is simply the ratio of successfully transmitted packets to packets sent. In this thesis, a platoon leader performs accelerating/decelerating in a sinusoidal pattern (the sinusoidal pattern is summarized in Section 5.3 on page 31), and CACC controller is used to maintain the intra-platoon distance. We do not consider vehicles having the ability to estimate

the moving trend. Therefore, IRT is more meaningful for the communication-level reliability than PDR in this thesis.

2.5.3 Application-level reliability

Platooning applications use the CACC controller to maintain the required intra-vehicle distance within the platoon. The CACC controller uses wireless communication to obtain information about the position, speed, and acceleration from its preceding and leader vehicles. Therefore, if a vehicle cannot receive vehicles' recent information, the CACC controller will perform 'blind' control, i.e., it uses the previous outdated information as input to compute the new acceleration, which may lead to vehicles crashing or platoon instability problems [17]. Therefore, Michele Segata et al. [17] defined the safe time ratio as an application-level reliability metric for the platooning application. This safe time ratio r_{safe} is defined as the ratio of messages' delay smaller than the maximum allowable delay over all message delays, shown in equation 2.18.

$$D_{safe} = \{d : d \in D \ \wedge d < \delta_{req} + \Delta\} \tag{2.17}$$

$$r_{safe} = \frac{\sum_{d_s \in D_{safe}} d_s}{\sum_{d \in D} d} \tag{2.18}$$

In equation 2.17 and 2.18, D is the set of all message delays of a vehicle. δ_{req} is the maximum allowable delay. D_{safe} is the set of all message delays in which the value is smaller than $\delta_{req} + \Delta$. Δ represents the sampling time of the CACC controller, during which the message arrival is still considered valuable. Δ is 10 ms for the CACC controller [7, 17]. Furthermore, different settings have different delay requirements. For example, if the deceleration rate is set to 2 m/s^2, then message arrival interval of 0.5 s is sufficient to avoid vehicles crashing. However, a message interval of 0.33 s may cause vehicle crashes when the deceleration rate is 8 m/s^2 [17].

2.5.4 Quality metric

Campolo et al. [29] mentioned that fairness is a quality metric which represents the ratio of channel access opportunity of all stations. C. B. Math et al. [13] stated that fairness for neighbor nodes is a new challenge for DCC. Fairness means equal resource allocation for all vehicles under the same channel load. Unfairness may cause variation in message rate, data rate, etc. Kuk and Kim [30] defined as a fairness performance indicator the number of beacon messages delivered per transmitting vehicle per second. The unfairness in message rate means some vehicles have "weaker" presence among their neighbors; thus disturb the awareness of some vehicles, therefore, raising safety issues [30].

2.6 The state of art

As the vehicular communication channel may become congested when vehicle density increases and the CSMA/CA back-off mechanism may lead to unbounded transmission delay when packets encounter a huge number of collisions, A. Weinfield, J. Kenney, and G. Bansal [31] propose only using a collision avoidance mechanism without any congestion control algorithm. Field tests carried out with 360 vehicles, show that there is 79.5% packet loss at links where the Received Signal Strength Indicator (RSSI) ranges from -80 to -85 dBm. This level of communication quality obviously cannot satisfy the requirements of the safety applications. Therefore, ETSI proposed the DCC methods and specifications that each node can use to control its communication parameters to keep the channel load under control and to reduce the probability of packet collisions.

The current DCC proposed by ETSI is under development and several papers proved that this DCC has many existing problems. S. Kuk, et al. [32] evaluated the performance of ETSI DCC, their simulation results show that even in non-congested conditions, safety critical messages cannot reach beyond immediately adjacent neighbors when vehicles enter into the RESTRICTIVE

state. They claimed that the problem stems from the unsuitable configuration of parameter values. However, they did not investigate how the DCC state machine parameters should be tuned. E. Cinque, et al. [8] proved DCC has instability and unfairness problems, and they found that unfairness problems can be solved by adapting DCC state machine parameters. S. Yang and H. Kim [33] solved instability and unfairness problems by adding two more states to the basic three-state machine based on Transmit Data Rate Control to reduce the parameter gaps between states. Their proposed solution is referred to as SDCC. Their simulation results show that the performance of SDCC is better than ETSI DCC in terms of stability and fairness. Zheyuan [34] adapted the number of DCC sub-states in the ACTIVE state based on TRC and evaluated the performance of CBR, velocity synchronization, and acceleration synchronization[1]. Zheyuan's results show that DCC with more sub-states in the ACTIVE state performs better than the three-state DCC. In [35], N. Lyamin, Q. Deng, and A. Vinel proved that DCC with more sub-states in the ACTIVE state can reduce fuel consumption. Furthermore, there are many other congestion control algorithms which adapt different transmission parameters.

TRC adapts the message rate to control the number of packets sent per second. LIMERIC [36] and Periodically Updated Load Sensitive Adaptive Rate (PULSAR) [37] are two popular methods which adapt the message rate. LIMERIC is a distributed and adaptive linear control algorithm that vehicles can use to adapt their message rates to ensure that the total channel load converges to a specified target value. LIMERIC has high throughput which is *independent* of the number of neighbor nodes. PULSAR adapts the message rate using Additive Increase Multiplicative Decrease (AIMD) based on comparing the CBR values from two-hop neighbors with the target CBR. If the measured value is below the target CBR, the increase in message rate will be doubled. In contrast, when the measured CBR is above the target threshold the increment in message rate will be halved. Error Model Based Adaptive Rate Control (EMBARC) [38] is based on LIMERIC and it has the ability to adapt the message rate based on vehicle dynamics. The simulation results in [38] proved that EMBARC not only has the advantage of having high throughput under dense vehicle scenarios (as does LIMERIC), but tracking vehicles also becomes easier by transmitting more messages in dense scenarios. However, a too low message rate may affect the reliability of real-time applications.

TPC adapts the transmit power to reduce the communication range in order to maintain channel load under the target threshold. M. Torrent-Moreno, et al. [39] proposed a distributed fair power adjustment for vehicular environments (D-FPAV) method which uses TPC to control the load due to periodic beacons, therefore beacons with higher priorities have sufficient bandwidth to transmit. Another TPC method proposed by G. Caizzone, et al. [40] controls transmit power so that the number of vehicles within the range of each vehicle is always within a minimum and maximum threshold. However, too short of a communication range will be unable to guarantee the delivery of relevant safety messages. A. A. Jimenez Luna [41] proposed two new methods related to DCC. The first one is TPC with a simple proportional-integral control loop (TPC PI), the other one is TPC with proportional-integral control loop and a third depends on the exchange of channel load share information (TPC CLS). Compared with ETSI DCC, TPC CLS and TPC PI perform better and they can solve the channel load oscillation problem. Moreover, TPC CLS has better performance than TPC PI in terms of PDR.

Transmit Data Rate Control modifies the data rate, hence controlling the maximum amount of data to be sent in a given time period. Packet-count based decentralized data-rate DCC (PDR-DCC) [42] and data-rate DCC (DR-DCC) [13] proposed by C. B. Math, et al. are two popular methods which adapt the data rate. DR-DCC adapts the data rate only based on CBR. The simulation results in [13] proved DR-DCC has a better performance than a TPC algorithm with aspect to IRT, but comparison of DR-DCC and other DCC mechanisms is missing from [13]. PDR-DCC uses a packet count (P_c) together with CBR measurements to adapt the data rate. P_c represents the number of packets sensed by a node during a given period. From the simulation results in [42], PDR-DCC has greater fairness and higher reliability compared with LIMERIC and

[1] Velocity synchronization and acceleration synchronization represent vehicles in the platoon have the same speed and acceleration with the platoon leader

DR-DCC.

Some other algorithms combine different transmission parameters. For example, joint Transmit Power and Rate Control (TPRC) [43] uses a combination of TPC and TRC. T. Tielert, et al. [43] evaluated the performance of TPRC w.r.t average packet IRT. A given transmit power and transmit data rate are selected based on the target distance and channel load, respectively. Their simulation results show that the transmit power which leads to the optimal performance is *independent* of vehicle density. Moreover, TPRC is easier to implement than pure TPC since a precise transmit power is difficult to achieve in real scenarios. In [44], Y. Wei evaluated the performance of the Society of Automotive Engineers International DCC (SAE-DCC) and compared it with LIMERIC and PDR-DCC. SAE-DCC adapts the message rate based upon the number of vehicles within communication range and adjusts the transmit power based on the measured CBR. Additionally, there is an algorithm which combines message rate and data rate, namely, MD-DCC [45]. The message generation rate adaption of MD-DCC is similar to LIMERIC which is based on CBR and forces the channel load to converge to a target value. Data rate adaption is based on the number of vehicles in communication range and the minimum beacon frequency which guarantees the channel has sufficient capacity of maintaining the message generation rate while keeping maximizing the communication range. Simulation results show that MD-DCC performs more reliably and leads to a better channel utilization than LIMERIC in high vehicle density scenarios. A. Autolitano, et al. [46] did a simulation by adapting each transmission parameter to estimate the overall performance of DCC methods (especially for TPC, TRC, and DSC). The performance was evaluated in terms of PDR, update delay, and CBR. Their simulation results indicate that TRC and TPC mainly affect the overall behavior of DCC and the parameter settings are highly sensitive. L. Le, et al. [47] compared the performance of TPC, TRC, and TPRC w.r.t CBR. Their results show that different DCC algorithms are suitable for different situations, TPRC is the most flexible among them. A. Vesco, et al. [48] also evaluated the performance of DCC with different transmission parameters. Their simulation utilized two scenarios: Line-Of-Sight and an urban environment. Similar to the results in [46], TRC mainly determines the performance of DCC.

Chapter 3

Research Methodology

This chapter provides an overview of the research method used in this thesis. Section 3.1 explains the research process. Section 3.2 introduces the simulation software. Section 3.3 describes the data collection methods.

3.1 Research process

The research process of this Master's project matches the goals mentioned in Section 1.3. The goals can be summarized as: analyze and evaluate the effectiveness of DCC protocols for multiplatooning application under different vehicle densities. Also, the research should figure out whether the performance of ETSI DCC will be improved when adding more states to the three-state machine.

In order to achieve the goals, we choose the quantitative simulation method, then the research process was conducted as follows:

(1) Since STB uses pure CSMA/CA with a fixed 10 Hz message rate without considering the channel conditions, ETSI proposed the DCC algorithm which allows vehicles to adapt their transmission parameters to keep the channel load under a target threshold. Therefore, we first implemented the DCC-3 algorithm based on the three-state machine using TRC.

(2) Secondly, the performance of how DCC-3 keeps channel load for the multiplatooning application is evaluated under different vehicle densities using the Plexe simulator and the performance of DCC-3 is compared with the baseline STB algorithm. Moreover, we observe whether DCC-3 has unfairness problems;

(3) We implemented a number of other DCC algorithms (namely, DynB, LIMEIRC, and DCC-7). Then benchmarked their performance in comparison with DCC-3 in terms of CBR, IRT, collision rates, safe time ratio, and fairness using the Plexe simulator;

(4) We adapted transmit power for all vehicles within the communication range and observe the performance of STB protocol in terms of CBR, IRT, and collision rates for those vehicles in the middle of the platoon and who are also in the middle platoon of all the platoons; and

(5) Finally, simulation data will be collected from the OMNeT++ simulator and simulation results will be plotted and analyzed using Python.

3.2 Software description

OMNeT++ [49] is a C++ and discrete event-based simulator for simulating network communications including wired and wireless communication. A discrete event simulator means state changes (events) happen at discrete instances and take zero time to react. OMNeT++ is also a component-based simulator, where each component (module) is written in C++ and the modules are combined

together using a higher level language (specifically, the NED language). Modules communicate through messages to form compound modules. Whenever an event is scheduled in a module, a message corresponding to this event will automatically be sent to this module.

SUMO (Simulation of Urban MObility) [50] is a widely used traffic simulator which can support large traffic scenarios. SUMO is a microscopic simulator where each vehicle is given a specific name, departure time, departure position, etc. SUMO is used to generate vehicles, mobility and each roads, traffic.

Veins (Vehicles in Network Simulation) [51] extends the OMNeT++ network simulator together with SUMO traffic simulator to provide a comprehensive vehicular communication framework. Veins is easy to download since it is an open source framework. Each vehicle in SUMO is defined as a node in OMNeT++. Each node is associated with a network stack including an IEEE 802.11p network interface (NIC). The higher level beaconing protocol is sit on top of NIC. OMNeT++ and SUMO are connected via a Traffic Control Interface (TraCI). Therefore, the mobility model of the related node is updated in OMNeT++ whenever a vehicle moves in SUMO. The structure of Veins is shown in Figure 3.1 [18].

Plexe is an extension of the Veins simulator to support platooning applications. The vehicle's dynamic information (such as speed, position, and acceleration, etc.) in SUMO can be sent to platooning applications in OMNeT++ via a TraCI. The received data can be passed back to SUMO to be used by a car following model, such as CACC.

Similar to Veins, Plexe provides each vehicle with an IEEE 802.11p NIC card and an application layer. Plexe implements SUMO by adding the car following model which enables longitudinal control to support the platooning application. We use the Plexe simulator in this thesis work since it supports a realistic simulation of the platooning application and it features ACC as well as CACC models. Furthermore, it is free to download.

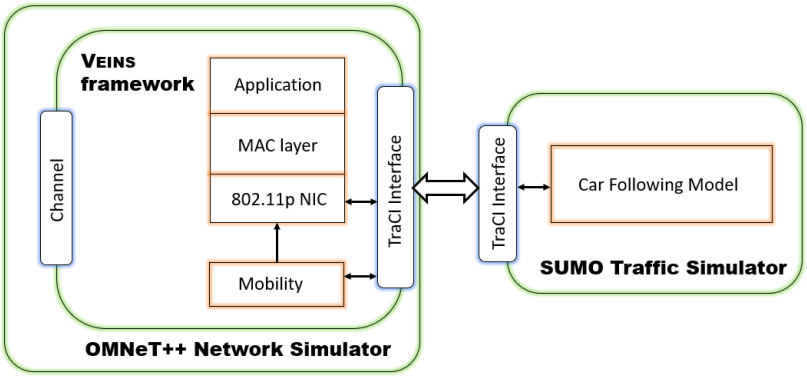


Figure 3.1: Schematic view of the Veins framework

3.3 Data collection

The simulation results of the Plexe simulator are stored in .vec and .sca files. Time series data (such as CBR values) is stored in .vec files . Summary data for a whole simulation (such as the total number of received messages) is stored in .sca files . Simulation files can directly viewed through .vec and .sca files using the OMNeT++ IDE. Since .vec and .sca files are both line-oriented text files, it is easy to use Python for analyzing the data. Moreover, coding and debugging is easier to do in Python than R. Also, Python is more suitable for dong data manipulation and repeated tasks. The Python libraries used frequently during the data analysis phase in this Master's thesis are:

Matplotlib

produces 2D graphs in a variety of formats. It is used to generate box plots and line plots

in this project.

Seaborn
it is a Python visualization library based on Matplotlib, and it can use a high-level interface to generate statistical graphs. It is used for assigning grouping columns in box plots in this project.

Numpy
provides functions for scientific computing. It is used for calculating the median and mean values for box plots in this project.

Chapter 4

DCC algorithms

This chapter describes the implementation of the DCC algorithms in detail. Section 4.1 describes the calculation of CBR. The implementation of the three-state machine DCC using TRC is explained in Section 4.2. Section 4.3 introduces DCC-7 with five sub-states of the ACTIVE state. Section 4.4 and Section 4.5 explain the implementation of DynB and LIMERIC respectively.

4.1 Channel busy ratio

Many congestion control protocols need measured CBR values to adapt transmission parameters. CBR is calculated as the percentage of channel busy duration over the measurement interval (T_m). The channel is indicated as busy when the following conditions are met:

(1) There is a frame decoding on the PHY layer; or

(2) Clear Channel Access (CCA) determines channel as busy.

We define a channel busy indicator (CBI) as a boolean to represent the channel state: busy or idle. CBI is calculated as shown in equation 4.1:

$$CBI = \begin{cases} 1 = if\ condition\ (1)\ or\ (2)\ is\ met \\ 0 = otherwise \end{cases} \tag{4.1}$$

The channel busy time over the duration can be calculated as shown in equation 4.2:

$$busyTime = \sum_{t-T_m}^{t} CBI \tag{4.2}$$

CBR is calculated as shown in equation 4.3.

$$CBR = \frac{busyTime}{T_m} \tag{4.3}$$

4.2 DCC-3

DCC-3 maintains channel load below a threshold to guarantee the throughput of successfully delivering packets in the network. DCC-3 performs as a three-state machine with transitions among RELAXED, ACTIVE, and RESTRICTIVE states based on the measured CBR as shown in Figure 4.1. As mentioned in Section 2.4, the DCC_acc component is located in the Access technology layer and it can utilize different congestion control methods. Among all of the control methods considered, TRC has the largest effect on the overall behavior as mentioned in Section

2.6. Therefore, in this Master's project, we focused on implementing and evaluating only those DCC protocols based on TRC for the platooning application.

The working principle of DCC-3 is to decrease the message rate when the communication channel becomes congested. Then CBR will decrease as fewer beacons are transmitted. In contrast, when the communication channel becomes idle the message rate should increase in order to transmit more beacons increasing the channel's utilization.

The three-state machine is based on CBR measurements, transition time measurements, and state transitions.

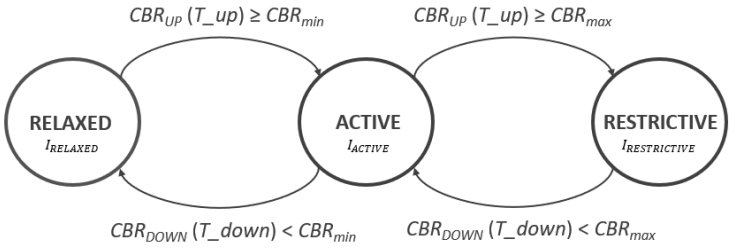


Figure 4.1: DCC-3 state machine

4.2.1 Three-state machine parameters

The state machine parameters shown in Figure 4.1 are explained in detail here based on ETSI TS 102 687 standard [25]. These parameter values are also summarized in Chapter 5.

$\boldsymbol{T\_up}$	A state change is triggered if CBR exceeds the threshold for a period of time. T_up represents how fast the control loop reacts when the measured CBR increases.
$\boldsymbol{T\_down}$	Represents how fast the control loop reacts when the measured CBR decreases.
$\boldsymbol{T\_sampling}$	Represents the minimum interval for subsequent checks of DCC states. $T\_sampling \leq T\_up \leq T\_down$
$\boldsymbol{CBR_{min}}$	Represents the minimum channel load below which the channel is assumed to be idle.
$\boldsymbol{CBR_{max}}$	Represents the maximum channel load above which the channel is assumed to be overloaded.
$\boldsymbol{CBR_{up}(T\_up)}$	Represents the minimum channel load for the past period T_up.
$\boldsymbol{CBR_{down}(T\_down)}$	Represents the maximum channel load for the past period T_down.
$\boldsymbol{I_{RELAXED}}$	Represents the message generation interval in the RELAXED state.
$\boldsymbol{I_{ACTIVE}}$	Represents the message generation interval in the ACTIVE state.
$\boldsymbol{I_{RESTRICTIVE}}$	Represents the message generation interval in the RESTRICTIVE state.

4.2.2 State transition

The initial state of DCC-3 is in RELAXED. The state transition rules are:

In the RELAXED state:
The state switches to ACTIVE, if $CBR_{up}(T\_up) \geq CBR_{min}$.

In the ACTIVE state:

- The state switches to RELAXED, if $CBR_{down}(T\_down) < CBR_{min}$.

- The state switches to RESTRICTIVE, if $CBR_{up}(T\_up) \geq CBR_{max}$.

In the RESTRICTIVE state:
The state switches to ACTIVE, if $CBR_{down}(T\_down) < CBR_{max}$.

The time interval between checking state transition rules should be no greater than $T\_sampling$.

4.2.3 Algorithm of DCC-3

The implementation of DCC-3 algorithm is shown as Algorithm 2.

Algorithm 2 DCC-3

```
Initialize():
  Define_module(DCC-3);
  Define three-state machine parameters;
  beaconInterval = 0.1;
  t_sent = -1;
  ScheduleAt (sendBeacon, Now + beaconInterval + uniform(0, 1));
  ScheduleAt (measureCBR, Now + uniform(0, 1));
  ScheduleAt (stateTransition, Now + uniform(0, 1));
while Simulation not ended do
  if message = sendBeacon then
    if t_sent + beaconInterval < Now then
      broadcast(platooningMsg);
    end if
    t_sent ← Now;
    ScheduleAt (sendBeacon , Now + beaconInterval);
  end if
  if message = measureCBR then
    Measure CBR;
    ScheduleAt (measureCBR, Now + 1 s);
  end if
  if message = stateTransition then
    Check state transition rules;
    if Rules satisfied then
      Perform state transition and update beaconInterval;
      if sendBeacon event has scheduled then
        ScheduleAt (sendBeacon, max(Now, t_sent + beaconInterval));
      end if
    end if
    ScheduleAt (stateTransition, Now + 1 s);
  end if
end while
```

In Algorithm 2, a module named DCC-3 is defined during Initialize(). The $beaconInterval$ represents the interval between sending two consecutive packets. Different states in the state machine have different message generation intervals as defined by $I_{RELAXED}$, I_{ACTIVE}, and $I_{RESTRICTIVE}$. t_sent represents the previous message sent time, which is initialized to -1 at the beginning. Each node schedules these three events at a random time initially to model the asynchronous nature of different nodes. Moreover, we schedule the $SendBeacon$ event later than the $measureCBR$ event in order to set the initial message generation interval based on the measured channel load. Now represents the current simulation time which is slightly larger than 1 s because simulation starts after the platoons are injected into the road at 1 s. After scheduling these events, messages named $sendBeacon$, $measureCBR$, and $stateTransition$ will

be automatically sent to the *DCC*-3 module itself, as mentioned in Section 3.2. The function of the three messages are:

SendBeacon This message informs the *DCC*-3 module to broadcast the platooning messages.

measureCBR This message informs the *DCC*-3 module to measure the current CBR.

stateTransition This message informs the *DCC*-3 module to check whether the current state needs to be switched based on the state transition rules.

When receiving a *sendBeacon* message, before broadcasting the platooning message, the previous message sent time (*t_sent*) plus the current beacon interval is compared with current simulation time in Algorithm 2. If the message scheduled time has already passed, then the message will be immediately broadcast. Otherwise, the message will be re-scheduled because the current message interval has already changed.

When receiving the *measureCBR* message, the CBR value will be measured. This value is used for checking state transition rules. Then the next *measureCBR* event will be scheduled one second later, because the recommended measuring CBR interval is one second as described in ETSI TS 102 687 standard [25]. However, the results of message generation interval may be different as the *sendBeacon* and *measureCBR* events are scheduled independently in this thesis.

When receiving the *stateTransition* message, if the state transition rules are satisfied then state transitions will be performed. If the state switches to RELAXED, then *beaconInterval* is updated by $I_{RELAXED}$. Similarly, if the state switches to ACTIVE or RESTRICTIVE, the *beaconInterval* will be updated by I_{ACTIVE} or $I_{RESTRICTIVE}$ respectively. Then the *sendBeacon* event is re-scheduled since the message generation interval may have changed if a state transition happened. This explains why it is necessary to check whether the platoon message should be broadcast immediately or not when receiving a *sendBeacon* event. The *stateTransition* event is also scheduled once per second because *T_sampling* is no greater than *T_up* (*T_up* is 1 s defined in the ETSI TS 102 687 standard). Moreover, a too small *T_sampling* is meaningless since the control loop reacts every one second (*T_up*) or five seconds (*T_down*).

4.3 DCC-7

In the previous section, we described the three-state machine of DCC-3 with one ACTIVE state. However, as mentioned in Section 2.6, DCC with more sub-states in ACTIVE has better communication performance than the three-state DCC. Moreover, according to [35], fuel consumption will be reduced using multi-state DCC. Therefore, in this Master's thesis, we implemented DCC with five sub-states in ACTIVE based on ETSI TR 101 613 standard [52] to compare with DCC-3. We call this multi-state DCC with seven states DCC-7. DCC-7 is shown in Figure 4.2.

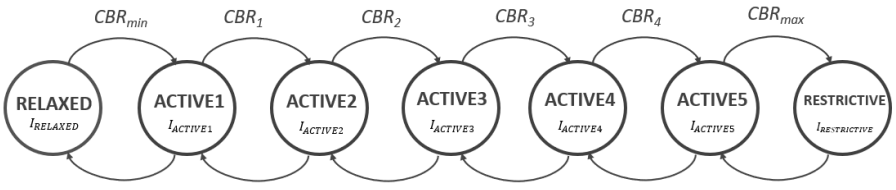


Figure 4.2: DCC-7 state machine

The basic working principle of DCC-7 is similar to DCC-3 (i.e. measuring CBR, performing state transitions, and send platooning messages). Therefore, there is no need to repeat the detailed state transitions and the description of the algorithm. We summarize the parameters of DCC-7 in Chapter 5.

4.4 Dynamic Beaconing

Since DCC-3 and DCC-7 have only limited numbers of message generation intervals for all channel congestion conditions, we consider a dynamic beaconing protocol which can dynamically adapt its message generation interval based on channel load. DynB targets maintaining CBR at a fixed value to reduce packet collisions. The idea of DynB is to increase the message generation interval whenever the network becomes denser (e.g. there are more neighboring vehicles). Therefore, DynB adapts the message generation interval more aggressive than DCC-3 and DCC-7 based on channel conditions. The message generation interval is calculated as equation 4.4 [53].

$$I_j = I_{des}(1 + r_j * N_j) \tag{4.4}$$

$$r_j = \max(0, \min(\frac{b_{m_j}}{b_{des}} - 1, 1)) \tag{4.5}$$

In equation 4.4, r is a parameter related to CBR and calculated as shown in equation 4.5. We use i and j to represent node i and j respectively. b_m is the measured CBR value and b_{des} is the desired CBR value. N is the number of one-hop neighbors. If node i receives a packet from node j within a pre-defined time interval, then node j is regarded as a neighbor of node i. The value of r is in the range of [0, 1]. When b_m is smaller than b_{des}, the current CBR is below the desired CBR, hence the minimum message generation interval I_{des} should be used to send as many packets as possible. When b_m is larger than b_{des}, the message generation interval should be increased to send fewer packets. The DynB algorithm is shown in Algorithm 3.

Algorithm 3 DynB

```
Initialize():
  Define_module(DynB);
  Define I = I_des, r, N, b_m, b_des;
  ScheduleAt (sendBeacon, Now + I + uniform(0, 1));
  ScheduleAt (measureCBR, Now + uniform(0, 1));
while Simulation not ended do
  if message = sendBeacon then
    update N; calculate I;
    if t_sent + I < Now then
      broadcast(platooningMsg);
    end if
    t_sent ← Now;
    ScheduleAt (sendBeacon , Now + I);
  end if
  if message = measureCBR then
    update b_m;
    ScheduleAt (measureCBR, Now + 1 s);
  end if
end while
```

In Algorithm 3, the I_{des} is set to 0.01 s to represent the minimum message generation interval [53]. Similar to DCC-3, the *sendBeacon* and *measureCBR* events for each node are initially scheduled at random times. When receiving a *sendBeacon* message, the number of neighbors will be updated. Then based on the measured CBR, a new message generation interval, namely, I is calculated based on equations 4.4 and 4.5. Similar to the DCC-3 algorithm, before broadcasting the platooning message, the previous message sent time plus the current beacon interval needs to be compared with the current simulation time. If the scheduled event time has already passed, the message needs to be broadcast immediately. Otherwise, the event will be re-scheduled after an interval I.

4.5 LIMERIC Beaconing

LIMERIC also targets keeping CBR at a desired threshold similar to DynB. However, the message rate of LIMERIC is based on the measured CBR value and the previous message rate. The message rate is calculated as equation 4.6 [36].

$$R_j(t) = (1 - \alpha) * R_j(t - I) + \beta * (CBR_T - CBR_j(t - T) \tag{4.6}$$

α is a contraction parameter which impacts the channel fairness and the convergence speed. β is a linear gain adaptive parameter, which impacts stability and also the convergence speed. Together they ensure a fair and stable convergence of the channel utilization. β plays a more important role than α in ensuring the algorithm's convergence [36]. CBR_T is the target channel load and $CBR_j(t)$ is the measured channel load by vehicle j. In steady state, all CBR values measured by vehicles converge to CBR_c and the message rate is fair for all vehicles. CBR_c is calculated as shown in equation 4.7. $CBR(t)$ is the total channel load of all vehicles measured at time t and calculated as shown in equation 4.8. The speed of convergence to fairness is calculated as shown in equation 4.9 [36].

$$CBR_c = \frac{n * \beta}{\alpha + n * \beta} * CBR_T \tag{4.7}$$

$$CBR(t) = CBR_c + (1 - \alpha - n\beta)^t (CBR(0) - CBR_c) \tag{4.8}$$

$$R_m(t) - R_k(t) = (1 - \alpha)^t (R_m(0) - R_k(0)) \tag{4.9}$$

The parameter n represents the number of vehicles within communication range. Parameters m and k represent vehicles m and k respectively. From equation 4.9, it can be seen that a larger value of α represents a faster convergence to fairness. From equation 4.7, it is obvious that a small enough α enables CBR_c almost independent n. Therefore, α is usually set to be 0.1 for balancing the convergence speed and the results of CBR_c [36, 54, 44]. When consider the value of β, from equation 4.7 we can see that a larger value can decrease the variation of CBR_c under different vehicle densities. From equation 4.8, when β is smaller than $(1 - \alpha)/n$, $(1 - \alpha - n\beta)$ is positive, the magnitude decreases which means a faster convergence when t increases. On the contrary, when *beta* is larger than $(1 - \alpha)/n$, the magnitude increases which leads to slower convergence. Therefore, β value is usually set to be 0.0066 [36] or 0.033 [54, 44] in most of the papers about LIMERIC. As a result, when n is small, CBR_c converges to a small fraction of CBR_T; when n is large, CBR_c is almost equal to CBR_T.

The LIMERIC algorithm is implemented as Algorithm 4. In Algorithm 4, two events are scheduled during initialization: *sendBeacon* and *measureCBR*. When the LIMERIC module receives a *sendBeacon* message, the message rate is calculated as equation 4.6, and the next *sendBeacon* event will be scheduled based on the current beacon interval. Similar to previous algorithms, the *measureCBR* event is scheduled periodically to measure CBR values. We set the period of measuring CBR to be 0.2 s instead of 1 s [44]. This is because LIMERIC calculates message rate using measured CBR values as shown in equation 4.6, therefore, a small measuring interval could provide more accurate information for LIMERIC to achieve channel fairness and convergence.

Algorithm 4 LIMERIC

```
Initialize():
  Define_module(LIMERIC);
  beaconRate = 10;
  beaconInterval = 0.1;
  ScheduleAt (sendBeacon, Now + beaconInterval + uniform(0, 1));
  ScheduleAt (measureCBR, Now + uniform(0, 1));
while Simulation not ended do
  if message = sendBeacon then
    if t_sent + beaconInterval < Now then
      broadcast(platooningMsg);
      calculate beaconRate;
      beaconInterval = 1/beaconRate;
    end if
    t_sent ← Now;
    ScheduleAt (sendBeacon , Now + beaconInterval);
  end if
  if message = measureCBR then
    measure CBR;
    ScheduleAt (measureCBR, Now + 0.2 s);
  end if
end while
```

Chapter 5

Simulation setup

In this Master's project, we use the Plexe simulator to evaluate the performance of DCC protocols for the platooning application. In this chapter, we summarize the parameter settings of traffic simulation, network simulation, platooning controllers, and DCC protocols. Section 5.1 describes the traffic simulation setup. The ACC and CACC controller parameters are summarized in Section 5.2. Section 5.3 introduces the sinusoidal simulation scenario. The network setup will be described in Section 5.4. Finally, the various DCC parameters will be summarized in Section 5.5.

5.1 Traffic simulation setup

As stated earlier we use the Plexe simulator together with SUMO to benchmark the performance of different DCC algorithms. We created a 4-lane highway scenario using the SUMO traffic simulator, as shown in Figure 5.1.

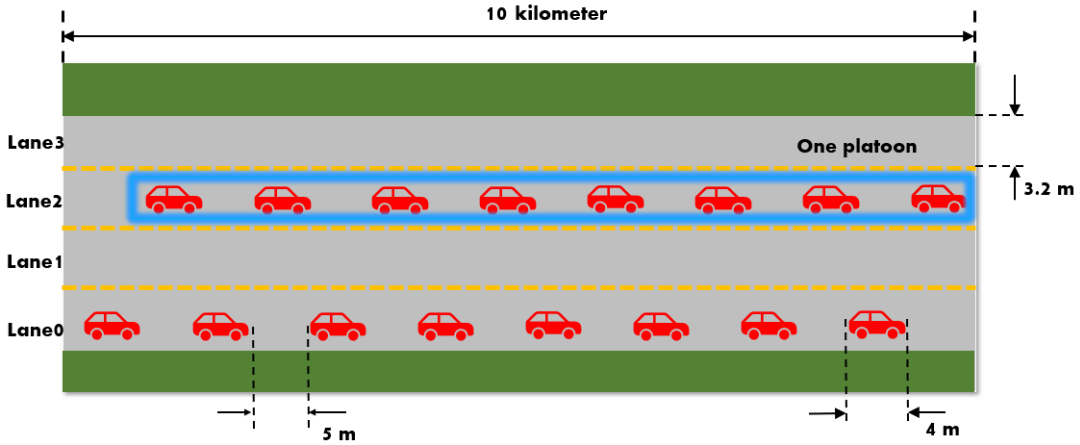


Figure 5.1: The highway scenario setup in SUMO

In Figure 5.1, each lane is 3.2 m in width and 10 kilometers in length. Vehicles in a platoon are injected into the road at 1 s with a fixed intra-vehicle distance of 5 m and inter-vehicle distance of 33 m. Each injected platoon contains eight vehicles of the same length (4 m). The number of injected vehicles should be a multiple of the platoon size multiplied by the total number of occupied lanes to ensure that each platoon has eight vehicles. For example, injecting 15 vehicles is not allowed, since there would be a platoon with seven vehicles. Additionally, the speed of the platoon leader is set to 100 km/h for a vehicle with 4 m length in this thesis[1].

[1] http://ec.europa.eu/transport/road_safety/going_abroad/sweden/speed_limits_en.htm

If there is only one platoon injected on the highway, then all vehicles in the platoon are injected in one lane, which is Lane0 in Figure 5.1. When there is more than one platoon injected, and the number of vehicles equals to 32, then the vehicles are injected in the way shown in Figure 5.2. The first vehicle is injected on Lane0, followed by Lane1, Lane2, and Lane3. Then the fourth vehicle is injected on Lane0 again.

Additionally, the communication range and interference range of IEEE 802.11p are 400 m [55] and 1000 m [56] respectively. Therefore, we choose the numbers of platoons within communication range and within interference range. Platoons outside of the interference range do not matter since they neither communication nor interfere with each other. As we know, each platoon has length of 67 m (8*4+7*5), and inter-platoon distance of 33 m. Therefore, four sets of nine platoons in parallel and 20 platoons in parallel will be the maximum number of platoons within communication range and within interference range of the middle platoon respectively. Therefore, in this thesis, we inject 1, 4, 8, 16, 20, 24, 36, 40, 60, and 80 platoons to evaluate the performance of different DCC protocols with different numbers of platoons. The simulation is set to be 20 s. Based on calculation, the lane length of 10 km is sufficient for 80 platoons to run 20 s on four lanes.

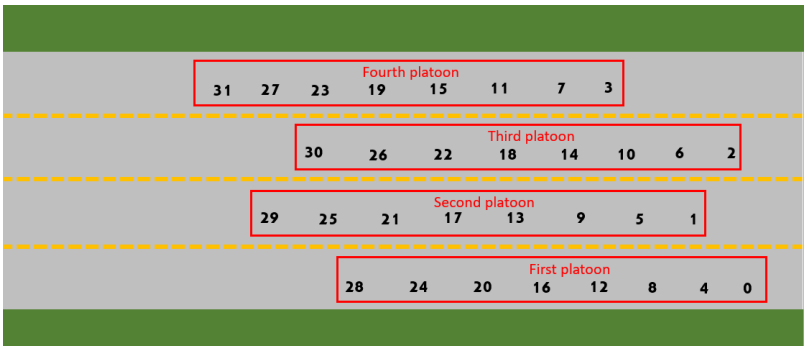


Figure 5.2: Sequence of injected vehicles, 32 vehicles

A summary of traffic simulation parameters is shown in Table 5.1.

Table 5.1: Road traffic simulation parameters

Parameters	Values
Numbers of platoons in total	1, 4, 8, 16, 20, 24, 36, 40, 60, and 80
Number of lanes	4
Lane width	3.2 m
Lane length	10 km
Intra-vehicle distance	5 m
Inter-vehicle distance	33 m
Platoon size	8 vehicles
Platoon injection start time	1 s
Vehicle length	4 m
Vehicle speed	100 km/h
Vehicle max speed	110 km/h
Simulation duration	20 s

5.2 CACC and ACC controllers setup

As mentioned in Section 2.1, CACC controller performs better than ACC in terms of traffic efficiency and safety aspects. Therefore, the CACC controller was chosen for intra-platoon

communication in our simulation. However, the inter-vehicle distance (33 m) needs to be controlled by ACC since 33 m is larger than the CACC's control range of 20 m, as mentioned in Section 2.1. Messages from the platoon leader and a vehicle's preceding vehicle are passed to the CACC controller. The controller parameters given in equations 2.1 to 2.15 are summarized in Table 5.2.

Table 5.2: Road traffic simulation parameters

Controller	Parameters	Value
ACC	T	1.2 s
	λ	0.1
CACC	C1	0.5
	ξ	1
	w_n	0.2 Hz
	τ	0.5 s

5.3 Simulation scenario setup

The sinusoidal scenario makes the platoon leader accelerate and decelerate with a 0.2 Hz frequency to simulate anticipation of the platoon to other traffic on the mixed highway [57]. Then a periodic disturbance is influencing the platoon leader's speed. The sinusoidal oscillation amplitude in speed is set to 10 km/h. A summary of traffic simulation parameters is shown in Table 5.3.

Table 5.3: Sinusoidal simulation parameters

Parameters	Value
Sinusoidal oscillation frequency	0.2 Hz
Sinusoidal oscillation amplitude	10 km/h
Start oscillation time	5 s

5.4 Network simulation setup

Plexe is an extension of Veins simulator which supports IEEE 802.11p PHY/MAC layer and IEEE 1609.4 DSCR/WAVE network layer for vehicular communication. We only use the CCH since it is used for the periodical transmission of control information. The carrier frequency in the PHY layer is 5.89 GHz, and the sensitivity and noise floor are both -95 dBm. The transmit power in the MAC layer is 100 mW, which equals to 20 dBm. The bit rate is set to 6 Mb/s. Our parameter settings are the same as those of Michele Segata in [17]. These setting have been chosen in order to compare the performance of different DCC protocols for the platooning application under the same network simulation setup.

Path loss represents the average received signal power which is related to the transmit signal power and the distance between the transmitter and the receiver. Consideration of path loss increases the modeling accuracy of the simulation. In our work, we consider only free space propagation which takes the distance between the sender and the receiver as well as the wavelength into account. The received power can be calculated as equation 5.1 [58].

$$P_r = \frac{P_t * G_r * G_t * \lambda^2}{(4 * \pi)^2 * d^{\alpha}} \tag{5.1}$$

P_r and P_t represent received signal power and transmit signal power respectively. G_r and G_t are the gain of the receiving antenna and the transmission antenna. α is the path loss exponent.

The distance between a transmitter and a receiver is represented by d. Transmit power of 100 mw is used to simulate the performance of different DCC protocols. Then we reduce the transmit power from 100 mw to 16 mw, 4 mw, and 1 mw to find a suitable transmit power which can over a whole platoon as well as reduce interferences of platoons in the communication range. Parameter f is the transmitted frequency which is 5.89 GHz in IEEE 802.11p PHY layer. The distance between transmitter and receiver is represented by d. λ is the wavelength calculated as equation 5.2, where c represents the speed of light which is $3 * 10^8$ m/s. Hence λ approximately equals to 0.051 m. The environment-dependent path loss exponent equals two in the free space model.

$$\lambda = \frac{c}{f} \tag{5.2}$$

A summary of network simulation parameters is shown in Table 5.4.

Table 5.4: Network simulation parameters

Parameters	Value
PHY model	IEEE 802.11p
MAC model	MAC 1609.4 CCH
Path loss model	Free space
Path loss exponent α	2
Frequency	5.89 GHz
Transmit power	100 mw, 16 mw, 4 mw, 1 mw
Sensitivity	-95 dBm
Noise floor	-94 dBm
CCA threshold	-95 dBm
Bitrate	6 Mb/s

5.5 DCCs' parameters

DCC-3 and DCC-7 parameters are presented in Table 5.5 and Table 5.6 respectively. Simulation parameters of all of the DCC protocols are summarized in Table 5.7. Moreover, the scheduled events for these protocols are summarized in Table 5.8.

Table 5.5: DCC-3 parameters

States	CBR	message generation interval (s)	message rate (Hz)
RELAXED	$0 \leq$ CBR < 0.15	0.04	25
ACTIVE	$0.15 \leq$ CBR < 0.4	0.5	2
RESTRICTIVE	CBR ≥ 0.4	1	1

Table 5.6: DCC-7 parameters

States	CBR	message generation interval (s)	message rate (Hz)
RELAXED	$0 \leq$ CBR < 0.19	0.06	16.7
ACTIVE1	$0.19 \leq$ CBR < 0.27	0.10	10.0
ACTIVE2	$0.27 \leq$ CBR < 0.35	0.18	5.6
ACTIVE3	$0.35 \leq$ CBR < 0.43	0.26	3.8
ACTIVE4	$0.43 \leq$ CBR < 0.51	0.34	2.9
ACTIVE5	$0.51 \leq$ CBR < 0.59	0.42	2.4
RESTRICTIVE	CBR ≥ 0.59	0.46	2.2

Table 5.7: DCCs parameters

Protocols	Parameters	Values
STB	$beaconInterval$	0.1 s
DCC-3 and DCC-7 common	T_{up}	1 s
	T_{down}	5 s
	$T_{sampling}$	1 s
DCC-3 only	CBR_{min}	0.15
	CBR_{max}	0.4
	$I_{RELAXED}$	0.04 s
	I_{ACTIVE}	0.5 s
	$I_{RESTRICTIVE}$	1 s
DCC-7 only	CBR_{min}	0.19
	CBR_1	0.27
	CBR_2	0.35
	CBR_3	0.43
	CBR_4	0.51
	CBR_{max}	0.59
	$I_{RELAXED}$	0.06 s
	$I_{ACTIVE1}$	0.1 s
	$I_{ACTIVE2}$	0.18 s
	$I_{ACTIVE3}$	0.26 s
	$I_{ACTIVE4}$	0.34 s
	$I_{ACTIVE5}$	0.42 s
	$I_{RESTRICTIVE}$	0.46 s
DynB	I_{des}	0.01 s
	b_{des}	0.25
LIMERIC	α	0.1
	β	0.033
	CBR_T	0.7

Table 5.8: DCC protocols' scheduled events

Protocols	Scheduled events	Events scheduled interval (s)	Events initial scheduled time (s)
STB	$sendBeacon$	0.1	≈1.1 + uniform(0, 1)
DCC-3	$sendBeacon$	3 fixed message intervals	≈1.1 + uniform(0, 1)
	$measureCBR$	1	≈1 + uniform(0, 1)
	$stateTransition$	1	≈1 + uniform(0, 1)
DCC-7	$sendBeacon$	7 fixed message intervals	≈1.1 + uniform(0, 1)
	$measureCBR$	1	≈1 + uniform(0, 1)
	$stateTransition$	1	≈1 + uniform(0, 1)
DynB	$sendBeacon$	Varies	≈1.01 + uniform(0, 1)
	measureCBR	≈1	≈1 + uniform(0, 1)
LIMERIC	$sendBeacon$	Varies	≈1.1 + uniform(0, 1)
	$measureCBR$	0.2	≈1 + uniform(0, 1)

Chapter 6

Performance evaluation

In this chapter, we compare the performance of DCC-3 with four congestion control algorithms (STB, DynB, LIMERIC, and DCC-7). The simulation results using different evaluation metrics are summarized and analyzed. Section 6.1 describes different evaluation metrics. Section 6.2 evaluates the effectiveness of DCC-3 keeping channel load under control as the number of platoons increases compared with the baseline algorithm STB. Next, the unfairness problem of DCC-3 is observed in Section 6.3. In Section 6.4, we benchmark DCC-3 in a comparison with STB, DynB, LIMERIC, and DCC-7 using those evaluation metrics mentioned in Section 6.1 under different vehicle densities to analyze the effectiveness of these algorithms. In Section 6.5, we observe the middle platoon with the maximum number of platoons within the communication range by adapting transmit power. Finally, we give a summary of the simulation results in Section 6.6.

6.1 Evaluation metrics

In this section, we present four categories of evaluation metrics:

(1) Network-level metrics: CBR and the number of packet collisions;

(2) Communication-level metrics: IRT and message generation interval;

(3) Application-level metrics: safe time ratio; and

(4) Quality metrics: Fairness.

6.1.1 CBR

CBR represents the fraction of time when the channel is busy. Since PDR reaches its peak when CBR is around 0.6 to 0.7 as mentioned in Section 2.5.1, we set the desired CBR to 0.7 in this thesis. Then we will evaluate the effectiveness of different DCC protocols maintaining CBR to the target threshold. We sample and record CBR values every 0.1 seconds for all protocols since the minimum interval for CBR measurements is 0.2 s (as shown in Table 5.8). Moreover, we discard those values measured before 2.2 s in order to have data for all vehicles and protocols since the latest initial CBR measurement time is at approximately 2.1 s (as shown in Table 5.8).

DCC-3 was selected as an example in order to explain how the DCC state machine works. We observe the states transitions of the three-state machine as a function of CBR values with different platoon numbers. Although the box plots of CBR contain data from all vehicles, the state transition with CBR needs to be analyzed for a specific vehicle, since different vehicles perform different state transitions due to the unfairness problem. Therefore, we choose to observe a vehicle in the first platoon on Lane0, because when we only inject one platoon to the road, the state transition with CBR can still be analyzed since the first platoon on Lane0 is always the first platoon to be injected, as shown in Figure 5.2. We choose the fourth vehicle in the first platoon

because the leader and tail vehicles are 'special' vehicles due to their positions in a platoon, and we want to mitigate their 'special' status (as they have either no preceding or trailing vehicle).

6.1.2 Collisions

The number of collisions per second counts the number of frames that could not be decoded in the PHY layer due to interference during one second. Therefore, we sample this metric once per second. When the number of vehicles increases, the number of collisions may also increase, we observe the number of collisions per second for different protocols under different vehicle densities. Results of collisions are sampled in box plots. Since the computation of collisions in IEEE 802.11p is quite complex, there are no simple models that can calculate it [53]. Therefore, in our work, we just analyze the trend in collisions (computed by the Plexe simulator) based on CBR and the message generation intervals rather than analyzing specific collisions per second values.

6.1.3 IRT and message generation interval

IRT is one of the most important communication-level metrics for safety applications, such as the platooning application. IRT is defined as the time interval between two successful beacon receptions from the same sender. IRT values depend on CBR, the number of collisions per second, and the message generation interval. We will measure the 95^{th} percentile IRT for these algorithms under different vehicle densities. The 95^{th} percentile IRT is defined as the IRT value which is larger than 95% of the measured IRT values. Furthermore, IRT and the message generation interval are also analyzed through box plots.

6.1.4 Safe time ratio

The application-level metric safe time ratio r_{safe} is defined as the ratio of message delays smaller than the maximum allowable delay to all message delays. The maximum allowable delay δ_{req} depends on vehicles' maximum acceleration or deceleration. In our simulation, we set the platoon leader's accelerations and decelerations in a sinusoidal pattern with the amplitude of 10 km/h. However, the controller needs time to stabilize due to the cruise control law and vehicle dynamics. Therefore, we can see from Figure 6.1 that the actual acceleration and speed of the platoon leader stabilize after the first loop ([5 s, 10 s]). The speed and acceleration are plotted as blue and red lines respectively. From Figure 6.1 we can see that the maximum acceleration and deceleration are roughly 1.5 m/s^2. Michele Segata et al. [17] mentioned that 0.5 s of δ_{req} with an acceleration/deceleration of 2 m/s^2 can avoid vehicles' crashing into each other. Therefore, for a maximum acceleration of 1.5 m/s^2 in our simulation of the platooning application, δ_{req} to 0.5 s can also avoid vehicle's crashing. Moreover, we observe r_{safe} under the worst case scenario where δ_{req} is set to 0.1 s as was done in [17].

6.1.5 Fairness

DCC unfairness problems may cause variations of neighbor nodes with regard to message rate, data rate, etc. In our simulation, we observe the message rate for two nodes in the same platoon. We use scatter plots to show the distribution of message rate as a function of simulation time. If two neighboring nodes have the same message rate or their message rates converge to similar values, then the two vehicles get similar information from the platoon leader and each vehicle's preceding vehicle. Unfairness in message rate will disturb the awareness of some vehicles; therefore, raising safety issues. We think DCC protocols having fairness of the message rate provide fair resource allocation for neighboring nodes. Otherwise, we conclude that this DCC protocol has unfair resource allocation.

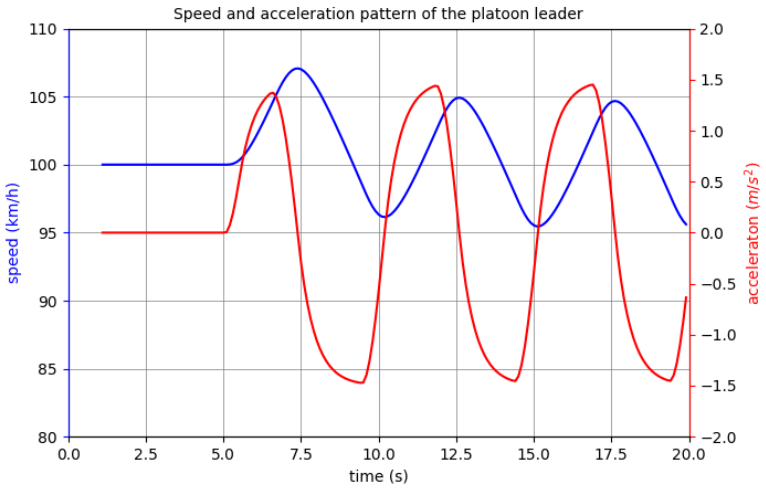


Figure 6.1: The speed and acceleration pattern of the platoon leader

6.2 Comparison of CBR for DCC-3 and STB

In this section, we compare the performance of DCC-3 with the baseline algorithm STB as a function of CBR to benchmark the effectiveness of DCC-3 maintaining the channel load under control. The performance comparison is shown in Figure 6.2. The working principle of the DCC-3 state machine is shown in Figure 6.3. We now present a detailed analysis.

For better understanding, we introduce the concept of box plots, since a box plot is a simple way to see data distribution. The central box spans the first quartile (Q1) to the third quartile (Q3). Median value ($Q2$) is a line inside the box. The inter-quartile range, represented by ΔQ, is calculated as $Q3 - Q1$. The upper quartile and lower quartile values are calculated by $Q3 + 1.5\Delta Q$ and $Q3 - 1.5\Delta Q$ respectively, and they are represented by whiskers. In this paper, we consider outliers as values which are smaller than the lower whisker or larger than the upper whisker. The box plots of all evaluation metrics in this thesis do not plot outliers for simplification.

In Figure 6.2, CBR_T represents the desired CBR under which PDR reaches its peak. CBR_T is set to 0.7 in all simulations. From Figure 6.2, we can see that for STB, when the number of platoons is smaller than 20, the network is not overloaded, because the CBR under this situation is smaller than CBR_T. At 24 platoons, CBR increases to roughly 0.8. When the number of platoons increases to 36, the channel saturates at almost 0.9 of CBR. As the number of platoons continues to increase the CBR remains around 0.9. This is because 36 platoons is the largest number of platoons within the communication range. Therefore, adding more platoons within the interference range does not have much influence on CBR values. We conclude that STB does not have the ability to keep channel load under control since CBR greatly exceeds CBR_T at high platoon numbers. For better illustration later, we define the number of platoons {36, 40, 60, 80} to be large number of platoons since CBR values are almost 0.9 in these cases. Moreover, from this box plot, we see STB always has larger CBR values than the upper whisker of DCC-3 when the number of platoon is larger than eight. On the contrary, the CBR values of STB are always smaller than or equal to the upper whisker values of DCC-3 when platoon numbers are {1, 4, 8}. Therefore, we refer to the number of platoon in the range of {1, 4, 8} to be few number of platoons and {16, 20, 24} to be moderate number of platoons.

Additionally, from Figure 6.2, we can see that the median CBR values of DCC-3 keep increasing as the number of platoons increases. However, DCC-3 has the ability to maintains CBR under 0.55 even when there are 80 platoons. Also, we can observe that the median CBR of DCC-3 are basically smaller than those of STB except for the case of a single platoon. It also can be seen

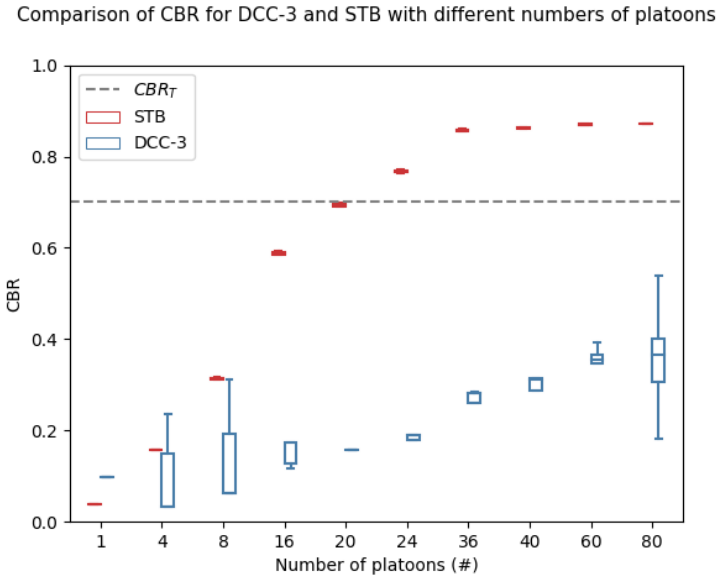


Figure 6.2: CBR of DCC-3 and STB with different numbers of platoons

that the inter-quartile range of the box tends to decrease as the number of platoons increases from one to 20 and then increase afterward. We found two interesting observations, and we summarize them as follows:

Observation 1 The median CBR value of DCC-3 with one platoon is larger than that of STB; and

Observation 2 The trend of the inter-quartile range of DCC-3 first decreases and then increases as the number of platoons increases.

In order to explain these two observations, we briefly summarize the working principle of DCC-3 again. DCC-3 acts as a three-state machine, where different states have different message rates. State transition happens based on measured CBR values. When the CBR value increases, the control loop reacts once per second. Conversely, when the CBR value decreases, the control loop reacts once per five seconds. The measured CBR value is compared with CBR_{min} and CBR_{max}, which are 0.15 and 0.4 respectively. The detailed state transitions rules ware described previously in Section 4.2.2. In this section, the state transitions as a function of CBR are shown in Figure 6.3. We choose cases of 1, 4, 20, and 80 platoons to explain the above two observations. In Figure 6.3 state and CBR are represented by red and blue lines respectively. States are numbered "0" (RELAXED), "1" (ACTIVE), and "2" (RESTRICTIVE) in this figure. Moreover, in order to clearly show the state transition rules of the three-state machine, we plot CBR values at an interval of 1 s. The detailed explanations of the above two observations are:

(1) From the first subplot in Figure 6.3, we can see that when there is one platoon, CBR values are around 0.1 and do not vary at all. Therefore, the state is RELAXED all the time because the CBR values of 0.1 are smaller than CBR_{min}, hence no state transitions occur. The message rate of DCC-3 in the RELAXED state is 25 Hz, which is larger than for STB. We know that a larger message rate will generate more packets into the communication channel, therefore, the channel busy time increases as more packets are transmitted in the channel, which leads to a higher channel load. Therefore, the median CBR value of DCC-3 under one platoon density is larger than that of STB, which explains observation 1.

(2) The second subplot in Figure 6.3 is the state transition with 4 platoons. In this figure, CBR is roughly 0.25 at around 2.2 s, then the state switches to ACTIVE since 0.25 is larger than

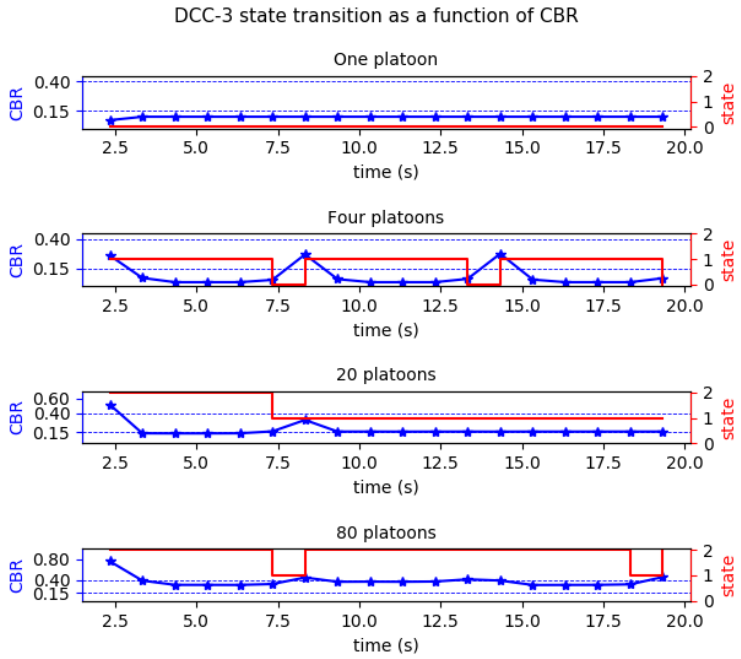


Figure 6.3: DCC-3 state transitions with changes in CBR plotted as function of simulation time

CBR_{min}. In the ACTIVE state, as the message rate decreases to 2 Hz, then CBR also decreases since fewer packets are transmitted in the channel. This can be seen from the period (2.2 s, 7.2 s], where CBR continues to decrease. Since the maximum value of CBR is smaller than 0.15 during this period, the state changes back to RELAXED at around 7.2 s. In the RELAXED state, the message rate increases back to 25 Hz, which leads CBR increasing again. When the next measurement occurs at around 8.2 s the CBR value exceeds CBR_{min}, therefore, the state switches to ACTIVE at around 9.2 s. We notice that this state transition pattern is repeated every 6 seconds where CBR increases for one second and decreases for five seconds until the end of 20 s.

(3) As shown in the third subplot in Figure 6.3, for 20 platoons, the state changes to RESTRICTIVE at around 2.2 s since CBR of 0.6 is larger than CBR_{max}. Then during the next 5 seconds, CBR values keep decreasing, and maximum CBR is smaller than 0.4 in the period (2.2 s, 7.2 s]. Therefore, the state switches to ACTIVE at around 7.2 s. In the next second, CBR increases due to the sudden message rate increase in the ACTIVE state; however, the state stays in ACTIVE since the CBR value does not reach 0.4 until around 8.2 s. From 8.2 s until 20 s, the state stays in ACTIVE since CBR values are slightly larger than 0.15 all the time. Since the state does not go to RELAXED in this situation, CBR values fluctuate less compared with the previous scenario. The CBR values at around 2.2 s and 8.2 s are treated as outliers since most CBR values are around 0.15 in this scenario.

(4) The state transition w.r.t CBR under 80 platoons is shown in the last subplot in Figure 6.3. In the beginning, CBR reaches 0.8 which causes the state switches to RESTRICTIVE. During the next 5 s, CBR keeps decreasing and the maximum CBR value is smaller than 0.4 during the period (2.2 s, 7.2 s], thus the state changes to ACTIVE at around 7.2 s. After 1 s, CBR exceeds 0.4 again, therefore, the state changes back to RESTRICTIVE. Then the state stays in RESTRICTIVE until 18.5 s because the maximum CBR in every 5 s interval is always larger than CBR_{max} (0.4). In this subplot, we can see that most CBR values vary within the range of approximately [0.3, 0.44], which is basically same as the inter-quartile range of box plots in Figure 6.2 with 80 platoons. Therefore, the inter-quartile range of the box with 80

platoons is greater than that with 20 platoons. Observation 2 is explained together with the explanation of (2), (3), and (4).

In summary, from Figure 6.2 and Figure 6.3, we conclude that that STB does not have the ability to keep the channel load under control, and the channel saturates at almost 0.9 when there are larger number of platoons. In contrast, DCC-3 has the ability to maintain CBR below 0.55 even with large numbers of platoons when using the three-state machine.

6.3 DCC-3 unfairness problem

In this section, we will explain the unfairness problem of DCC-3, as shown in Figure 6.4.

ETSI DCC has unfairness and instability problems as mentioned in Section 2.6. Fairness represents the number of messages delivered per second of different nodes converging to a similar value. In order to explain the unfairness problem of DCC-3 clearly, we observe CBR, state transitions, and message rate for two vehicles: vehicle 1 and 4 in the first platoon on Lane1 under 80 platoons. In the network, each vehicle is treated as a node with the corresponding ID. Here, we use vehicle with ID =1, and 4 in the network to represent the second vehicle and the fifth vehicle in this first platoon.

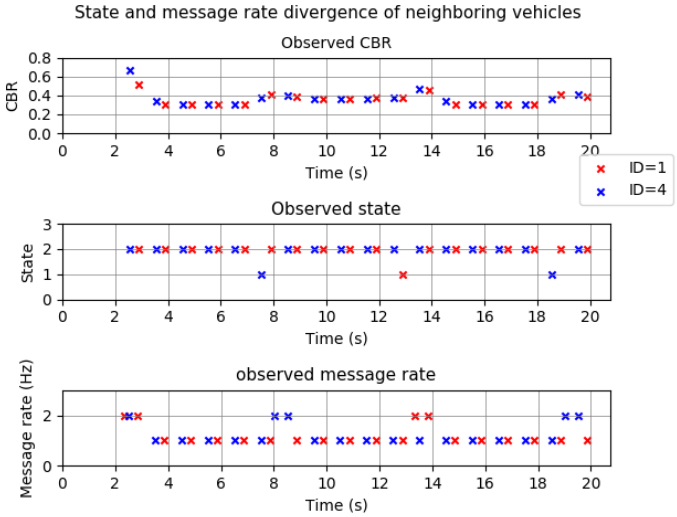


Figure 6.4: DCC-3 State divergence of neighboring vehicles

From Figure 6.4, we notice that node 4 starts measuring CBR at around 2.4 s which is earlier than node 1 at around 2.8 s, and the CBR values for node 4 and node 1 are roughly 0.7 and 0.5 respectively at the beginning. Since CBR values of 0.7 and 0.5 are larger than CBR_{max}, then both nodes switch to the RESTRICTIVE state. In the RESTRICTIVE state, the message rate decreases to 1 Hz which means fewer packets are transmitted in the channel. Therefore, we can see at 3.4 s and 3.8 s, the CBR values of node 1 and node 4 are below 0.4. Then at 7.4s , node 4 switches to the ACTIVE state since all CBR values in the period (2.4 s, 7.4 s] are smaller than CBR_{max} of 0.4. However, the CBR value measured by node 1 at 7.8 s is larger than 0.4. Then node 1 still stays in the RESTRICTIVE state since the maximum CBR in the period of (2.8 s, 7.8 s] is larger than CBR_{max} of 0.4. Next, in the period of (7.4 s, 8.4 s], CBR values keep increasing because the ACTIVE state has a message rate of 2 Hz which is larger than that in the RESTRICTIVE state. At 8.4 s, the CBR value of node 4 is larger than 0.4, therefore, node 4 switches back to the RESTRICTIVE state. As for node 1, CBR values keep decreasing and node 1 switches to the ACTIVE state at 12.8 s since the maximum CBR value in the period (7.8 s, 12.8 s] is smaller than 0.4. Then finally, node 4 switches to the ACTIVE state at 18.4 s while node

1 stays in the RESTRICTIVE state. State divergence of node 1 and node 4 causes message rate divergence, as shown in the last subplot in Figure 6.4.

From Figure 6.4 we see that a slightly difference in the measured CBR values causes neighboring nodes to have state divergence. This is because the measured CBR values are only compared with CBR_{min} and CBR_{max} when making decisions about state transitions. Moreover, the dramatic changes of message rate in different states cause neighboring nodes to have different CBR values at roughly the same time. This unfairness in message rate will disturb the awareness of some vehicles; therefore, raising safety issues [30].

6.4 Comparison of different congestion control algorithms

In this section, we benchmark DCC-3 with STB, DynB, LIMERIC, and DCC-7 using different evaluation metrics and different numbers of platoons. In Section 6.4.1, we present the comparison results with respect to CBR. In Section 6.4.2, the comparison of IRT in different scenarios is analyzed. The comparison of the number of packet collisions per second will be summarized in Section 6.4.3. Then the safe time ratio results are analyzed in Section 6.4.4. Finally, we compare the fairness of message rate for DCC-3, DCC-7, and LIMERIC in Section 6.4.5.

6.4.1 Comparison of CBR

The comparison of CBR for DCC-3 and STB was analyzed previously in Section 6.2. In this section, we compare the performance of CBR for DCC-3 with DynB, LIMERIC, and DCC-7. These results are as shown in Figure 6.5.

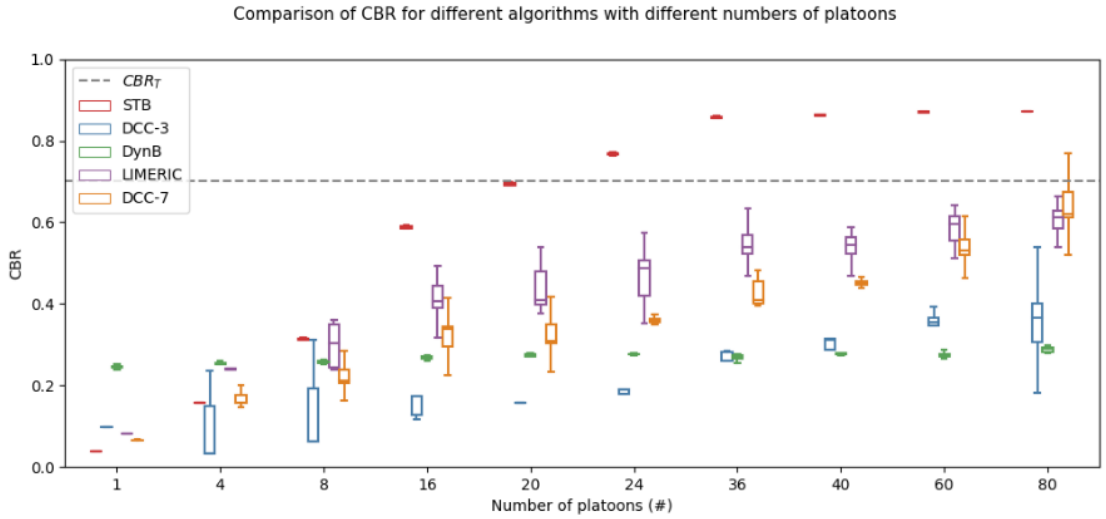


Figure 6.5: Comparison of CBR for different algorithms with different numbers of platoons

As mentioned in Section 6.2, DCC-3 has the ability to maintain CBR values under 0.55 even when there are many platoons, while using the three-state machine. However, DCC-3 has unfairness problems due to the dramatic parameter changes between states. Therefore, we implemented DynB, which can dynamically adjust the message rate based on the channel load. The performance of DynB for controlling the channel load is shown in Figure 6.5.

From Figure 6.5, we can see that DynB keeps the CBR values at around 0.25 under all numbers of platoons. This is because DynB increases the message generation interval when the channel becomes busier than the desired value, namely, b_{des}, which is defined as 0.25 in Table 5.7. Therefore, DynB has the ability to maintain CBR at b_{des}; but at the cost of increasing the message generation interval.

Next, it can be seen from Figure 6.5 that LIMERIC has higher channel utilization than DCC-3, and it has the ability to maintain CBR lower than 0.7 even with 80 platoons. The aim of LIMERIC it to target CBR values at 0.7, however, even with large numbers of platoons, the actual CBR values cannot reach 0.7. Yongyi Wei [44] also observed that CBR values of LIMERIC is about 10% lower than CBR_T. This means LIMERIC does not have a 100% efficiency in maintaining CBR at 0.7. This is because of the fixed β and α settings for all the situations considered in this thesis. In order to find the suitable β and α values for all the situations considered, there are a series of rules to follow [36]. However, finding the optimal values of these parameters for LIMERIC is not the main focus of this Master's project.

Moreover, in Figure 6.5 we notice that the average CBR values of DCC-7 are generally larger than that of DCC-3, which means DCC-7 has a higher channel utilization compared with DCC-3. This is because DCC-7 reduces the parameter gaps among states, also the message rate in the RESTRICTIVE state was increased compared with DCC-3 in order to increase the channel utilization. However, DCC-7 generally has a lower channel utilization than LIMERIC. This is because LIMERIC dynamically adapts the message rate to target CBR at 0.7, while DCC-7 only has fixed message rates based on the state machine. Finally, we can see that the maximum CBR value of DCC-7 with 80 platoons slightly exceeds CBR_T.

We summarize the median CBR values for different protocols with different numbers of platoons in Table 6.1.

Table 6.1: Median CBR values for different protocols with different numbers of platoons

	Number of platoons									
Algorithms	1	4	8	16	20	24	36	40	60	80
STB	0.04	0.16	0.31	0.58	0.69	0.77	0.86	0.86	0.87	0.87
DCC-3	0.09	0.04	0.06	0.13	0.16	0.19	0.28	0.31	0.35	0.37
DynB	0.24	0.25	0.26	0.27	0.27	0.28	0.27	0.28	0.28	0.29
LIMERIC	0.08	0.24	0.30	0.38	0.41	0.43	0.47	0.55	0.60	0.62
DCC-7	0.07	0.16	0.21	0.34	0.31	0.36	0.41	0.45	0.53	0.62

In summary, from Figure 6.5 and Table 6.1, we conclude that STB does not have the ability to keep channel load under control since the CBR values reach almost 0.9 with large numbers of platoons. DCC-3 can dynamically control channel load via the three state machine; however, the maximum CBR value is only 55% which is far from CBR_T. Therefore, the channel utilization of DCC-3 is not the optimal. DynB maintains CBR at a fixed pre-defined value 0.25, by dynamically adjusting the message rate. LIMERIC targets CBR at 0.7; however, CBR cannot reach this value due to unsuitable convergence parameters settings in this thesis. However, it still has the highest channel utilization compared with STB, DCC-3, DynB, and DCC-7 in most cases. For 80 platoons, DCC-7 has higher average channel utilization than LIMERIC. Finally, DCC-7 increases channel utilization compared with DCC-3 due to the parameter settings of the state machine.

6.4.2 Comparison of IRT

In this subsection, we compare the performance of the 95 percentile IRT for different DCC protocols, presented in Figure 6.6. Moreover, we analyze the box plots of IRT and message generation interval in Figure 6.7 and Figure 6.8 respectively.

From Figure 6.6, we can see that STB generally has the smallest 95% IRT values under most scenarios except for the cases of one and four platoons. DCC-7 and LIMERIC have slightly larger 95% IRT values than STB in most cases, and DCC-7 generally has smaller values than LIMERIC. 95% IRT values of DCC-3 and DynB are much larger than that of STB, DCC-7, and LIMERIC in most cases. Moreover, DCC-3 has smaller 95% IRT values than DynB with few platoons. In contrast, DynB has larger values than DCC-3 with moderate numbers of platoons. The 95% IRT value for DynB with 80 platoon reaches roughly 2.5 s.

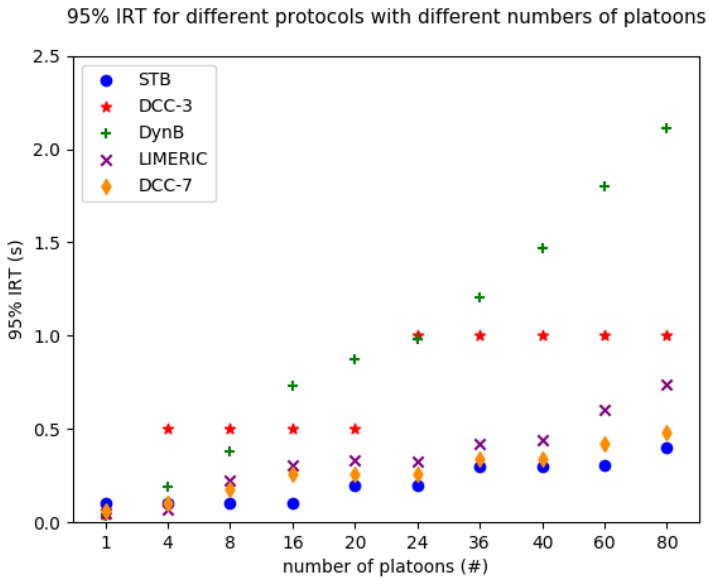


Figure 6.6: Comparison of 95% IRT for different protocols with different numbers of platoons

Next, we analyze the box plots of IRT and message generation interval in Figure 6.7 and Figure 6.8 (respectively). First, we start with the simulation results with few platoons. From Figure 6.7, we can see that with few platoons especially when there is only one platoon, STB has larger IRT values than other protocols since the message generation interval for STB is the largest, (as shown in Figure 6.8). This also explains why STB has the largest 95% IRT value when there is only one platoon, as shown in Figure 6.6. Moreover, we can see from Figure 6.7 that when there are four or eight platoons, DCC-3 has the largest fluctuation of IRT values, and the upper whiskers reach 1 s. We notice that the message generation rate with four or eight platoons for DCC-3 is in the range of [0.04 s, 0.5 s] (as shown in Figure 6.8), which means there is some packet loss due to packet collisions. For DynB, the maximum IRT values in the box plot keep increasing as number of platoons increases from one to eight (as shown in Figure 6.7), because the message generation intervals are increased to strictly target CBR at 0.25 shown in Figure 6.8. The IRT values for LIMERIC are always smaller than that of STB under light vehicle densities, since the message generation intervals of LIMERIC are smaller. It can also be seen that the inter-quartile range of DCC-7 in the box plots increases as the number of platoons increases; however, the inter-quartile range of DCC-7 is always smaller than that of DCC-3 due to the parameter settings of the state machine.

Additionally, from Figure 6.7 it can be seen that with moderate number of platoons, ΔQ of DCC-3 in the box plots for 16 platoons is larger than that for 20 and 24 platoons. This is because CBR values with 16 platoons in the box plots have larger fluctuations compared with the cases of 20 and 24 platoons, as shown in Figure 6.5. IRT values of DynB and LIMERIC also increase as message generation intervals are increased to maintain target channel load under control. DCC-7 has smaller ΔQ than DCC-3 and the maximum IRT value of DCC-7 is smaller than for DCC-3. Moreover, we notice that IRT of STB remains at around 0.1 s, because STB always utilizes a 10 Hz message rate and the channel is not congested with moderate numbers of platoons as CBR does not exceed CBR_T as shown in Figure 6.5.

Furthermore, from Figure 6.7 we notice that under large numbers of platoons, IRT values of STB increases to around 0.3 s as the channel becomes quite congested as shown in Figure 6.5 due to the fixed 10 Hz message rate. DynB increases message generation interval significantly to maintain CBR at 0.25, shown in Figure 6.8. The maximum IRT value of DynB with 80 platoons exceeds 2.5 s, which may cause the real-time applications to become unreliable. For DCC-3, we can

see the message generation interval generally varies between 0.5 s and 1 s with 60 and 80 platoons (as shown in Figure 6.8). Therefore, in Figure 6.7, it can be seen that the maximum IRT values of DCC reach 1.5 s since there are some packet losses. Also, from Figure 6.7, we notice that DCC-7 has larger Δ Q than LIMERIC because LIMERIC adapts message rate dynamically while DCC-7 only has fixed and limited message rates. From Figure 6.7 we can also see that LIMERIC has larger IRT values than STB when there are large numbers of platoons. This is because LIMERIC targets CBR at 0.7, therefore, it increases the message generation interval to keep channel load under the target threshold, while STB maintains a 10 Hz message rate regardless of the channel load, as shown in Figure 6.8.

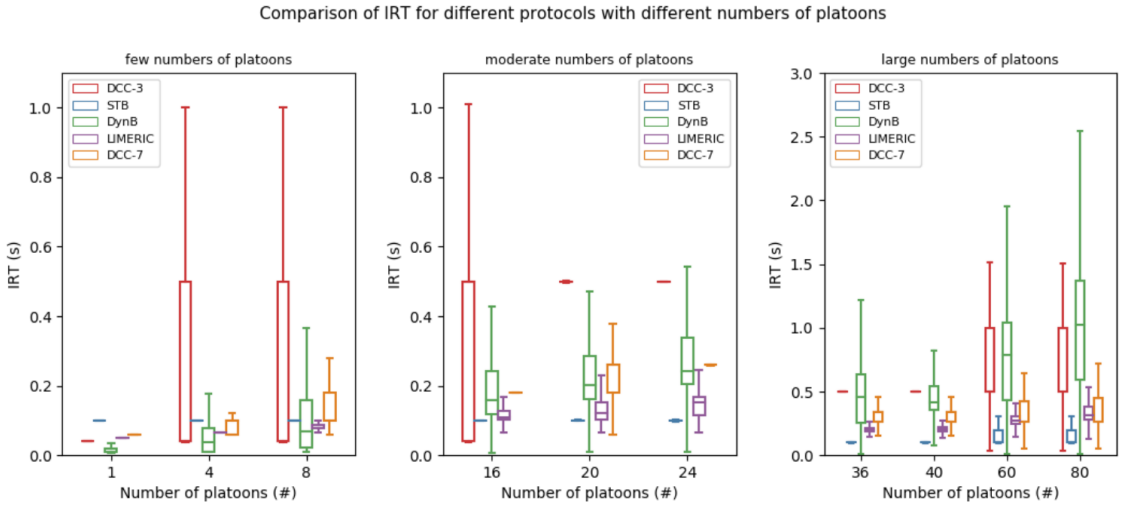


Figure 6.7: Comparison of IRT for different protocols with different numbers of platoons

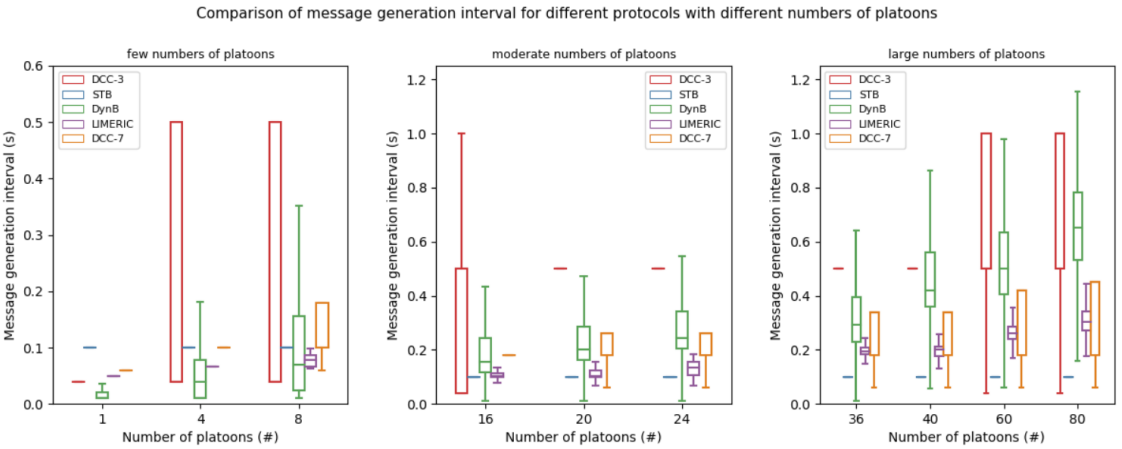


Figure 6.8: Comparison of message generation interval for different protocols with different numbers of platoons

We summarize the average, median, and 95% IRT values for different protocols with different numbers of platoons in Table 6.2.

In summary, from Figure 6.6, 6.7, 6.8, and Table 6.2, we conclude that STB generally has the lowest IRT values due to its fixed 10 Hz message generation rate. However, taking CBR into consideration, STB is unsuitable in scenarios with large numbers of platoons. DynB controls too aggressive on CBR, leading to a 95% IRT of as much as 2.7 s which may not be suitable for

real-time applications. IRT of DCC-3 has the largest fluctuation due to the dramatic parameter changes between states. The 95% IRT values of DCC-3 increase to 1.5 s with large numbers of platoons. Although LIMERIC and DCC-7 have larger 95% IRT values than STB, they have the ability to keep channel load under control. Moreover, DCC-7 has smaller 95% IRT values than LIMERIC in most cases.

Table 6.2: Average, median, and 95% IRT values (s) for different protocols with different numbers of platoons

Algorithms	calculation	Number of platoons									
		1	4	8	16	20	24	36	40	60	80
STB	average	0.10	0.10	0.10	0.11	0.11	0.11	0.13	0.14	0.15	0.15
	median	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
	95%	0.10	0.10	0.10	0.10	0.20	0.20	0.30	0.30	0.30	0.40
DCC-3	average	0.04	0.19	0.25	0.36	0.40	0.47	0.49	0.49	0.59	0.72
	median	0.04	0.04	0.04	0.50	0.50	0.50	0.50	0.50	0.50	1.00
	95%	0.04	0.50	0.50	0.50	0.50	1.00	1.00	1.00	1.00	1.00
DynB	average	0.02	0.06	0.12	0.23	0.28	0.33	0.51	0.55	0.83	1.07
	median	0.01	0.04	0.07	0.16	0.20	0.24	0.46	0.41	0.79	1.02
	95%	0.04	0.19	0.38	0.73	0.88	0.98	1.21	1.47	1.80	2.11
LIMERIC	average	0.05	0.06	0.11	0.14	0.16	0.17	0.23	0.23	0.31	0.37
	median	0.05	0.06	0.08	0.11	0.12	0.15	0.20	0.20	0.27	0.32
	95%	0.05	0.06	0.22	0.31	0.34	0.32	0.42	0.44	0.60	0.73
DCC-7	average	0.06	0.08	0.13	0.18	0.22	0.23	0.29	0.29	0.35	0.38
	median	0.06	0.10	0.10	0.18	0.26	0.26	0.34	0.34	0.42	0.45
	95%	0.06	0.10	0.18	0.26	0.26	0.26	0.34	0.34	0.42	0.48

6.4.3 Comparison of numbers of collisions per second

The number of packet collisions per second is influenced by CBR and message generation interval. The comparison of collision rates for different protocols with different numbers of platoons is shown in Figure 6.9. In this section we present a detailed analysis.

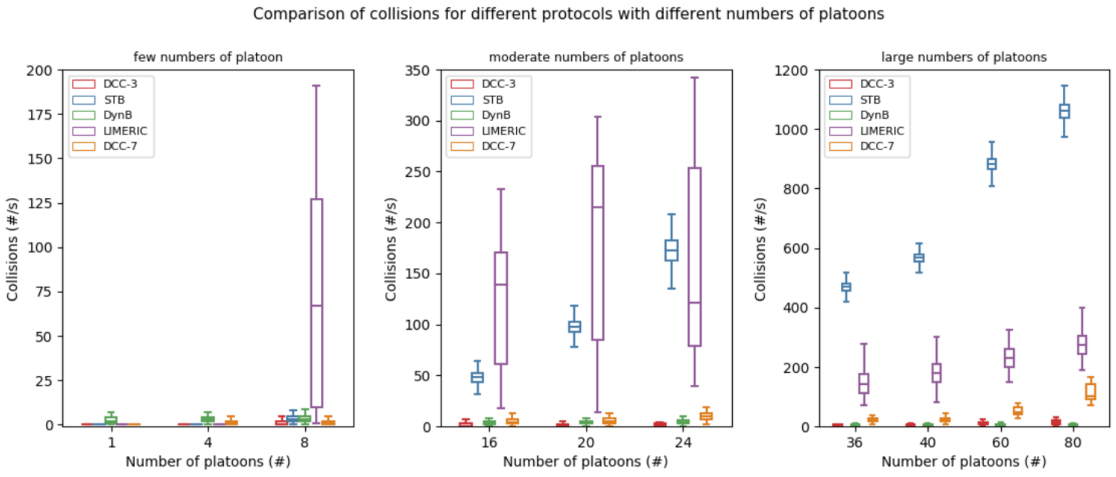


Figure 6.9: Comparison of number of collisions for different protocols with different numbers of platoons.

From Figure 6.9, we notice that when there are few platoons, although CBR values for LIMERIC is under 0.4 shown in Figure 6.5, the collision rate increases to almost 200 when there

are eight platoons. This is because the message generation interval for LIMERIC is smaller than 0.1 s all the time, as shown in Figure 6.8. Although the channel is not saturated in this situation, a small enough message generation interval may cause a huge number of collisions in the PHY layer. DynB, DCC-3, and DCC-7 also have message generation interval smaller than 0.1 s, however, these three protocols sometimes increase message generation intervals to larger than 0.1 s to reduce the packet collision rates, as shown in Figure 6.8. Therefore, we can see from Figure 6.9 that DynB, DCC-3, and DCC-7 have limited packet collision rates. As for STB, since the channel is not saturated and it does not send messages as frequently as LIMERIC, thus STB also has a very limited amount of collisions when there are few platoons, as shown in Figure 6.9.

Moreover, we can see from the middle plots in Figure 6.9 that when there are a moderate number of platoons, LIMERIC still has the highest collision rates. This is because the median message generation intervals of LIMERIC is around 0.1 s, and there as several instances of intervals smaller than 0.1 s as shown in Figure 6.8. The collision rates for STB shows a linear increase due to the fixed 10 Hz message rate and the linear increase in CBR values, as shown in Figure 6.5. Moreover, the median value of box plots of STB with 20 platoons starts to become larger than that of LIMERIC. As for DynB, DCC-3, and DCC-7, they can still keep collision rates at a limited level as they dynamically increase message generation interval to keep channel load under the target threshold.

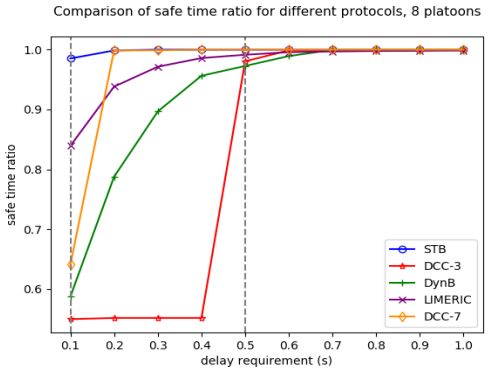
It can be seen from the right most plot in Figure 6.9, in scenarios with a large number of platoons, STB starts to have the highest collision rates because CBR values exceed 0.7 in this situation, as shown in Figure 6.5. A CBR value larger than 0.7 is considered as channel saturated which will cause a large number of packet collisions. DCC-3 and DynB have the ability to keep CBR and collision rates under control even with large numbers of platoons at cost of large IRT values, as shown in Figure 6.6. LIMERIC targets CBR at 0.7, while DCC and DynB maintain CBR at 0.4 and 0.25 respectively. Therefore, LIMERIC does not need to increase the message generation interval to such a large value as DynB and DCC, as shown in Figure 6.8. Therefore, LIMERIC always has a higher collision rate than DCC and STB due to its smaller message generation interval. However, compared with STB, LIMERIC has a significantly lower collision rate as shown in Figure 6.9. DCC-7 also has a relatively small collision rate even with 80 platoons; although the value is larger than that of DCC-3 and DynB. This is because the median message generation interval of DCC-7 is smaller than DCC-3 and DynB, as shown in Figure 6.8.

In conclusion, with large numbers of platoons, STB has the highest collision rate since it does not have the ability to keep the channel load under the desired threshold. With few or a moderate number of platoons, the collision rates for LIMERIC is the highest because its message generation interval is too small. DCC-3, DCC-7, and DynB are capable of keeping CBR and collision rates under control regardless of the number of platoons.

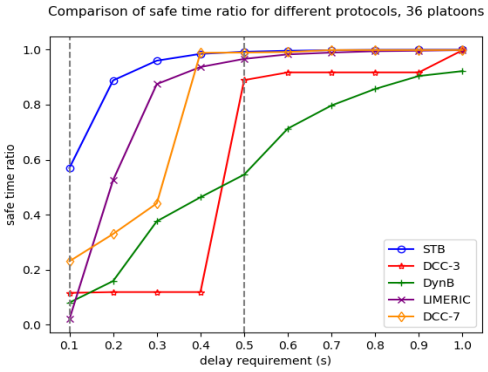
6.4.4 Comparison of safe time ratios

The results of the safe time ratio for different protocols with 24, 36, 60, and 80 platoons are shown in Figure 6.10. As we mentioned previously, the maximum allowable delay δ_{req} depends on vehicles' maximum acceleration or deceleration. In our simulation, the platoon leader has a maximum acceleration and deceleration rate of roughly 1.5 m/s^2 and other vehicles in each platoon use wireless communication to follow the leader's speed and acceleration. Moreover, for maximum acceleration and deceleration rates of 1.5 m/s^2, packet delays smaller than 0.5 s are considered as safe for the platooning application (as was described in Section 6.4.4).

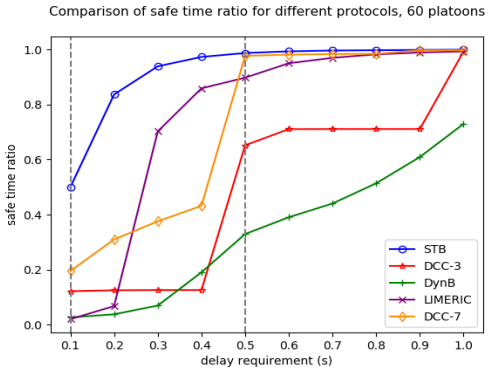
From Figure 6.10 (a), we can see that with eight platoons, all protocols can ensure the platooning application in safe conditions almost 100% of the time for δ_{req} equals to 0.5 s. However, for δ_{req} smaller than 0.5, DCC-3 only has around 0.5 safe time ratio, which may affect the platooning application if several vehicles have a maximum acceleration or deceleration rate larger than 1.5 m/s^2. In the worst case (considered here) for a delay requirement of 0.1 s, STB and LIMERIC still have safe time ratio of roughly 100% and 70% respectively. For 36 platoons in Figure 6.10 (b), the safe time ratio of all protocols except STB are generally below 0.2 for a delay requirement of 0.1 s. For a delay requirement of 0.5 s, DynB only has roughly 50% safe time ratio.



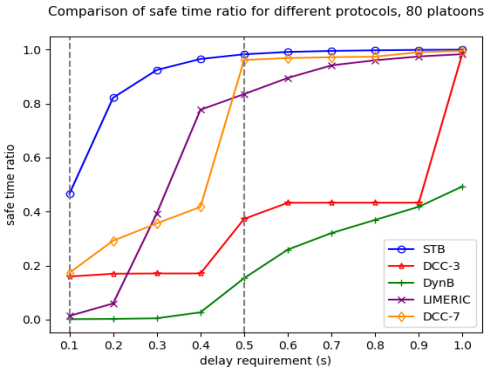
(a) safe time ratio for 8 platoons



(b) safe time ratio for 36 platoons



(c) safe time ratio for 60 platoons



(d) safe time ratio for 80 platoons

Figure 6.10: Comparison of safe time ratio for different protocols with different numbers of platoons

Additionally, in Figure 6.10 (c) it can be seen that DCC-3 and DynB have roughly 70% and 30% safe time ratio for δ_{req} equal to 0.5 s. In contrast, DCC-7, STB, and LIMERIC have rough 100% safe time ratio for a delay requirement of 0.5 s. However, in the most demanding scenario of δ_{req} equals to 0.1 s, STB can achieve a roughly 50% safe time ratio, while the safe time ratio for the others are clearly unacceptable.

Finally, From Figure 6.10 (d) we notice that for a delay requirement of 0.5 s, DCC-3 and DynB only have roughly 40% and 20% safe time ratio. DCC-7 and STB can still achieve almost 100% safe time ratio when δ_{req} equals to 0.5 s. As for LIMERIC, the safe time ratio is roughly 80%. Moreover, even for a delay requirement of 1 s, DynB can only achieve roughly 50% safe time ratio.

In summary, for a delay requirement of 0.5 s in our simulation scenario, DCC-7, STB, and LIMERIC can achieve safe time ratio above 80% of the time even in the 80 platoons scenario. DynB, in contrast, only has a 50% safe time ratio even for a delay requirement of 1 s. As for DCC-3, it only has roughly 40% safe time ratio for a delay requirement of 0.5 s with 80 platoons, which indicates that it is also not safe for the platooning application. Finally, for the worse case scenario of a delay requirement equals to 0.1 s, none of these protocols can achieve a safe time ratio higher than 80% with moderate or large numbers of platoons.

6.4.5 Comparison of fairness

In Section 6.3, we introduced and explained the unfairness problem of DCC-3. In this subsection, we will compare the fairness of DCC-3 with LIMERIC and DCC-7. These results are shown in Figure 6.11. We choose to observe vehicle 1 and 4 in the first platoon on Lane1 with 80 platoons,

which is the same situation as shown in Figure 6.4 (hence the values for vehicles 1 and 4 for DCC-3 are the same as those shown in the earlier figure).

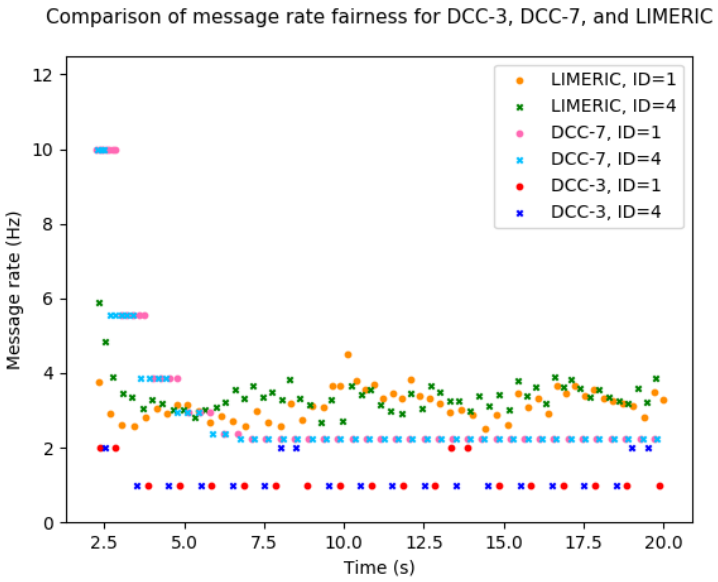


Figure 6.11: Comparison of fairness for DCC-3, LIMERIC, and DCC-7

As we can see from Figure 6.11, LIMERIC tends to converge the message rate of different nodes to similar values as time goes by. It can be seen that in the period [10 s, 20 s], the two nodes basically have the similar message rate. This is because LIMERIC adapts a vehicle's message rate based on channel load, hence it attempts to share the channel resources evenly for all vehicles. In the case of DCC-7, we can see that both nodes have the same message rate of roughly 2 Hz in the period [6 s, 20 s]. This is because DCC-7 reduces the parameter gaps among states compared with the three-state machine. Therefore, we conclude that LIMERIC has better fairness than DCC-3 as a function of message rate since LIMERIC attempts to share the channel resources evenly for all vehicles. However, LIMERIC still lacks fairness due to unsuitable and fixed parameter settings of α and β, which greatly influence the convergence speed and fairness. DCC-7 has a better fairness message rate compared with DCC-3 and LIMERIC, as it provides the most stable division of the channel resources (even if the message rate is below that of LIMERIC).

6.5 Performance with TPC

As we mentioned in Section 5.1, four sets of nine platoons in parallel is the maximum number of platoons within the communication range. However, the aggregate communication metrics will only be valid when computed for the 4 platoons in the middle. Since all of the others have few vehicles within communication range. Moreover, the transmit power of 100 mw is too high for intra-platoon communications, since communications range (400 m) is nearly 5 times the length of the platoon (67 m) and the interference range (1000 m) is nearly 15 times the length of the platoon. Therefore, in this section, we analyze the evaluation metrics of CBR, collisions, and IRT for the middle vehicles (except leader and tail vehicle in this platoon) in middle platoon on Lane1 with 36 platoons using different transmit power.

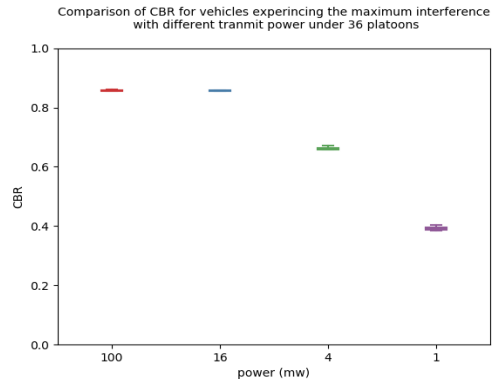
We choose transmit power of 1 mw, 4 mw, 16 mw to compare the performance of the baseline STB protocol with 100 mw. As we know, the communication range and interference range go as $1/r$ with transmit power in a free space model. Therefore, for transmit power of 1 mw, the communication and interference range decrease to 40 m and 100 m. We calculate the

communication and interference range as a function of transmit power, as shown in Table 6.3. Moreover, due to influences caused by noise, interference, and fading effects, etc., packets may not be decoded even within communication range.

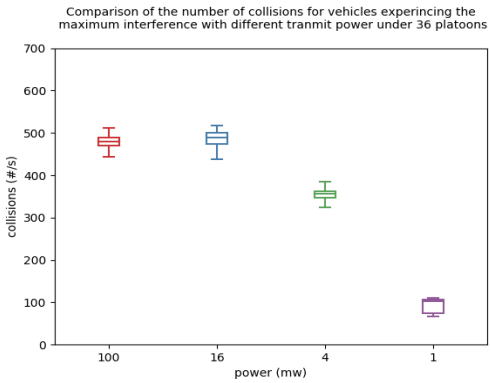
The comparison of CBR, number of collisions, front message delays, and leader message delays with different transmit power are shown in Figure 6.12.

Table 6.3: Communication and interference range as a function of transmit power

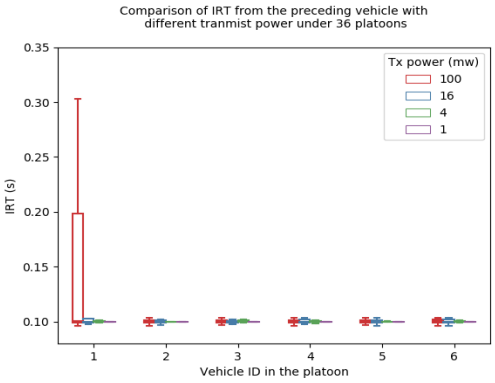
Transmit power (mw)	100	16	4	1
Communication range (m)	400	160	80	40
Interference range (m)	1000	400	200	100



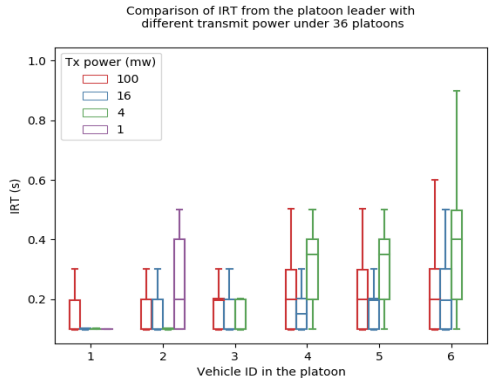
(a) CBR with different transmit power



(b) Number of collisions per second with different transmit power



(c) Front delay with different transmit power



(d) Leader delay with different transmit power

Figure 6.12: Comparison of CBR, number of collisions, front message delays, and leader message delays with different transmit power

From Figure 6.12 (a) we can see that CBR values are generally the same under 100 mw and 16 mw. The reason for this is all platoons are still within the interference range with transmit power of 16 mw. In contrast, CBR values decrease significantly when transmit power are 4 mw and 1 mw, since the vehicles within the communication range and interference range are reduced. From Figure 6.12 (b), we can see that collision rates are also decreased when transmit power are 4 mw and 1mw.

Since the communication range are shorter than the platoon size with transmit power of 1 mw, therefore, communicating with the platoon leader may become impossible. Therefore, we analyze

the IRT values of vehicles with ID from 1 to 6 (platoon leader has ID=0, platoon tail has ID=7) in the middle platoon with 36 platoons within the communication range. Moreover, we separate IRT values sent by the platoon leader and each vehicle's predecessor as shown in Figure 6.12 (c) and (d). From Figure 6.12 (c) we notice that STB with transmit power of 16 mw, 4 mw, and 1 mw can reduce IRT values for the vehicle with ID=1 sent by the predecessor. From Figure 6.12 (d) we can see that with transmit power of 1 mw, vehicles with ID in the range of{3, 6} cannot communicate with the platoon leader as the communication range is only approximately 40 m. Although vehicles can receive messages from the platoon leader with transmit power of 4 mw, the IRT values are the largest for the vehicle with ID=6. The IRT values of both leader and preceding vehicles are the best when the transmit power is 16 mw.

Next, we compare the fairness of DCC-3 with transmit power of 100 mw and 16 mw under 640 platoons. These results are shown in Figure 6.13. We choose to observe vehicle 1 and 4 in the first platoon on Lane1 with 80 platoons, which is the same situation as shown in Figure 6.4 (hence the values for vehicles 1 and 4 for DCC-3 with transmit power of 100 mw are the same as those shown in the earlier figure).

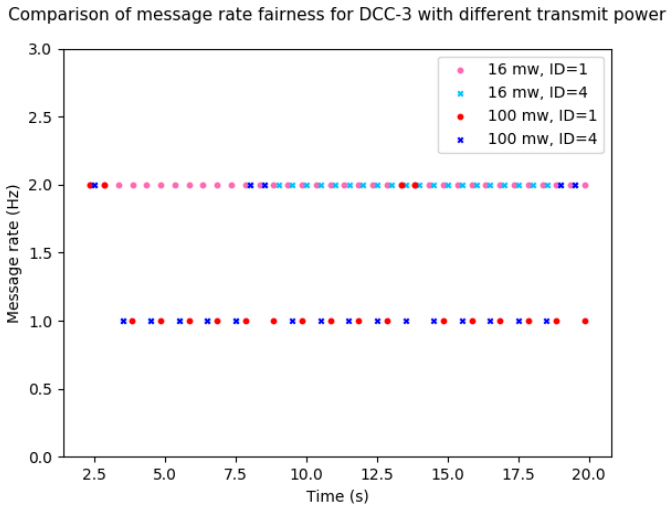


Figure 6.13: Comparison of fairness for DCC-3 with different transmit power

From Figure 6.13, we can see that both nodes have the same message rate of 2 Hz with transmit power of 16 mw; whereas, two nodes have divergence in message rate with transmit power of 100 mw. Therefore, the fairness in message rate is improved when the transmit power is reduced as the number of vehicles in the communication range and interference range is reduced.

In summary, when reducing power for all vehicles in the communication and interference ranges, CBR, collision rates, IRT values, and fairness could be improved. However, communicating with the platoon leader may become impossible. Therefore, we can set different transmit power for the platoon leader and its followers to make sure all vehicles in the platoon could communicate with the leader as well as reducing packet collisions in this platoon.

6.6 Summary

In this section, first we compared the effectiveness of DCC-3 and STB with respect to maintaining channel load under control, the results show that STB does not have the ability to control channel load as the channel load saturates to almost 0.9 of CBR with large numbers of platoons. In contrast, DCC-3 has the ability to keep the channel load under 0.55 because of its three-state machine. Next, we compared DCC-3, DynB, LIMERIC, and DCC-7 w.r.t CBR, IRT, collision rates, and safe time ratio. We noticed that DCC-3 has a low collision rate; however, the 95% IRT

value increases to 1.5 s with a large number of platoons. DynB strictly maintains CBR at 0.25, but at the cost of increasing IRT to around 2.5 s with 80 platoons. LIMERIC has a better channel utilization than other protocols mentioned in this thesis; however, its 95% IRT values are larger than that of STB. LIMERIC has a much lower collision rate than STB with a large number of platoons. When there are few platoons, LIMERIC has the highest collision rates due to the fixed α and β parameter settings. DCC-7 has a slightly smaller channel utilization than LIMERIC; moreover, it has smaller 95% IRT values and collision rates than LIMERIC. We analyzed the application-level metric safe time ratio. We noticed that for a delay requirement of 0.5 s, DCC-7, LIMERIC, and STB can achieve safe time ratio above 80% of the time even when there are many platoons. DCC-3 can only achieve 40% safe time ratio with a delay requirement of 0.5 s for the case of 80 platoons. DynB only has roughly 50% safe time ratio even for a delay requirement of 1 s. Moreover, we compared the fairness of DCC-3 with LIMERIC and DCC-7. The results show that DCC-7 has greater fairness of the message rate among neighbor nodes due to the parameter settings of the state machine. LIMERIC shows greater fairness than DCC-3; however, LIMERIC cannot achieve greater fairness due to the use of fixed α and β parameter settings for all numbers of platoons. Finally, by reducing transmit power for vehicles in the communication range, the performance of CBR, collisions, IRT, and fairness are improved; however, communicating with the platoon leader may become impossible if the communication range are smaller than the platoon size.

Chapter 7

Conclusions and future work

This chapter describes the conclusions and future work of this thesis. Section 7.1 gives the conclusions of this Master's thesis. Section 7.2 mentions the future work. Section 7.3 describes the economic, sustainability, and ethical aspects of this thesis.

7.1 Conclusions

In this Master's thesis, we evaluated the effectiveness of DCC-3 and we compared its performance with STB, DynB, LIMERIC, and DCC-7 using different evaluation metrics under different numbers of platoons for the platooning application.

Using the Plexe simulator, we injected different numbers of platoons into the road to simulate different numbers of platoons. The simulations considered 1, 4, 8, 16, 20, 24, 36, 40, 60, and 80 platoons in the simulated road. We set the platoon leader's acceleration and deceleration to follow in a sinusoidal pattern with the oscillation amplitude of 10 km/h to represent the influence of other vehicles on the mixed highway.

First, we evaluated the effectiveness of DCC-3 with respect to keeping channel load under control. From the simulation results from the Plexe simulator, we proved that DCC-3 has the ability to maintain channel load under 0.55 for all numbers of platoons, while the channel saturates to 0.9 of CBR for STB with large numbers of platoons. However, DCC-3 has an unfairness problem as neighboring nodes have different message rates.

Next, we compared the performance of DCC-3 with STB, DynB, LIMERIC, and DCC-7 with different numbers of platoons. With respect to CBR, LIMERIC achieves higher channel utilization in most cases. However, due to fixed parameter settings of LIMERIC for all numbers of platoons, it cannot achieve a higher channel utilization. DynB keeps CBR at 0.25 for all numbers of platoons to reduce packet collision rates. DCC-7 has a slightly lower channel utilization compared with LIMERIC; however, DCC-7 greatly improves CBR compared with DCC-3. When considering IRT and collision rates, DCC-3 and DynB always have the smallest collision rates at the cost of increasing 95% IRT to roughly 1.5 s and 2.5 s respectively. STB has the smaller IRT values under all numbers of platoons, however, it has the highest packet collision rate due to the fixed 10 Hz message rate. LIMERIC has larger 95% IRT values than STB with larger numbers of platoons, however, it has a lower collision rate than STB. Nevertheless, LIMERIC has the highest collision rate for small numbers of platoons compared with other protocols due to the unsuitable parameter settings of α and β. Although DCC-7 has slightly smaller channel utilization than LIMERIC, it has a much lower collision rate and 95% IRT value compared with LIMERIC. Next, from the perspective of safe time ratio, we conclude that STB, DCC-7, and LIMERIC can achieve at least an 80% safe time ratio when the delay requirement is 0.5 s. In contrast, DCC-3 and DynB only have a safe time ratio of roughly 40% and 20% which is unsafe for the platooning application. Then, we analyzed the fairness of DCC-3, DCC-7, and LIMERIC. The results show that DCC-7 has a better message rate fairness than LIMERIC and DCC-3, which represents a better division

of the channel resources.

Finally, we reduced the transmit power for all vehicles in the communication range and we observe CBR, collision rate, and IRT values sent by the platoon leader and each vehicle's predecessor for the middle platoon with 36 platoons. We found that CBR, collision rates and IRT values decrease as the transmit power decreases; however, a too small transmit power makes it impossible for vehicles in the middle platoon to communicate with the leader. Therefore, setting different transmit powes for the platoon leader and its followers is the best option. Moreover, we also observed the fairness in message rate of DCC-3 is improved with transmit power of 16 mw compared with 100 mw.

In summary, DCC-7 performs the best when taking CBR, IRT, collision rates, safe time ratio, and fairness together into consideration.

7.2 Future work

Based on our current research, there are several research studies that could be conducted in the future work to make a comprehensive evaluation of congestion control protocols. We summarize this future work as:

(1) In our DCC algorithms, different events (i.e. *sendBeacon* and *measureCBR*) are scheduled independently. The message interval may be different when the sequence of two scheduled events biased. Therefore, in the future, one may attempt to schedule one event based on other events, therefore, the events would be scheduled in a fixed order.

(2) Although LIMERIC targets CBR at 0.7 and aims to achieve fairness of message rates, the current fixed parameter settings of α and β is unsuitable for achieving this goal. In the future one might attempt to find suitable dynamic α and β values based on vehicle densities.

(3) We set the platoon leader to decelerate and accelerate in a sinusoidal pattern where the maximum acceleration and deceleration is 1.5 m/s^2. The safe time ratio of DCC-7, LIMERIC, and STB can reach 80% with the maximum delay requirement of 0.5 s. However, if the platoon leader has a higher acceleration or deceleration, the safe time ratio will decrease. Therefore, simulations under various simulation scenarios needs to be conducted.

(4) From the simulation results, DCC-7 performs much better than DCC-3 w.r.t CBR, IRT, safe time ratio, etc. This is because DCC-7 has five sub-states in ACTIVE which reduces parameter gaps within the state machine compared to DCC-3. However, we are unsure if the number of sub-states in ACTIVE were increased, would the performance of DCC become better? Therefore, it would be interesting to evaluate the performance of DCC with more sub-states in ACTIVE in future work.

(5) We reduced the transmit power for all vehicles in the communication range to reduce collision rates and channel load. However, vehicles in the middle platoon within the communication range cannot communicate with their leader when the transmit power is reduced to 1 mw. Therefore, it would be possible to set larger transmit power for the platoon leader and lower transmit power for its followers for improving the overall performance.

7.3 Reflections

Real-time applications such as the platooning application are supported by the transmission of safety critical messages; therefore, low latency communication is of vital importance to guarantee the safety of the platooning application. Timely messages exchanging could maintain the intra-platoon distance at a target value, therefore, avoiding vehicle crashes. DCC-7 has better overall performance compared with other DCC protocols implemented in this thesis since it can keep channel load under control as well as achieve a high safe time ratio, etc. Therefore, DCC-7 can

enhance the safety of the platooning application which leads to fewer traffic deaths. However, real experiments are costly and difficult to realize; therefore, it is difficult to prove the implementation correctness of DCC protocols even they were evaluated by simulation.

Moreover, platooning application is innovative and full of challenges. Platooning could improve traffic safety and reduce fuel consumptions. The future development of platoon follows different steps, where the first step is to have the leading vehicle in a platoon controlled by driver. Then in the next step, a platoon will be controlled by fully automated vehicles which will further improve traffic efficiency and flows. Moreover, in a mixed highway scenario, platoons are influenced by other vehicles on the road which may influence the delivery of safety critical messages. Therefore, platoons with mixed traffic scenarios need further research.

Bibliography

[1] M. S. Frigau, "Fair Decentralized Congestion and Awareness Control for Vehicular Networks," in *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, Dec. 2015, pp. 172–180. [Online]. Available: http://dx.doi.org/10.1109/ICPADS.2015.30

[2] A. Balador, A. Böhm, C. T. Calafate, and J. C. Cano, "A reliable token-based MAC protocol for V2V communication in urban VANET," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2016, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/PIMRC.2016.7794823

[3] "Comparison of Longitudinal Control of Vehicles with and without Inter-vehicle Communication," *Section for Automation and Control, Department of Electronic Systems, Aalborg University*. [Online]. Available: http://kom.aau.dk/group/09gr723/tidligere%20p7%20projekter/platoon-article.pdf

[4] D. Lu, Z. Li, and D. Huang, "Platooning as a service of autonomous vehicles," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2017, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2017.7974353

[5] H. Peng, D. Li, K. Abboud, H. Zhou, H. Zhao, W. Zhuang, and X. . Shen, "Performance Analysis of IEEE 802.11p DCF for Multiplatooning Communications With Autonomous Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2485–2498, Mar. 2017. [Online]. Available: http://dx.doi.org/10.1109/TVT.2016.2571696

[6] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1109/IEEESTD.2012.6178212

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

[8] E. Cinque, F. Valentini, A. Iovine, and M. Pratesi, "An adaptive strategy to mitigate instability in the ETSI DCC: Experimental validation," in *2017 15th International Conference on ITS Telecommunications (ITST)*, May 2017, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/ITST.2017.7972223

[9] "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," *Technical Report, ETSI TS 102 637-3 V1.1.1*, 2010. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102600_102699/10263703/01.01.01_60/ts_10263703v010101p.pdf

[10] M. v. Eenennaam, A. v. d. Venis, and G. Karagiannis, "Impact of IEEE 1609.4 channel switching on the IEEE 802.11p beaconing performance," in *2012 IFIP Wireless Days*, Nov. 2012, pp. 1–8. [Online]. Available: https://doi.org/10.1109/WD.2012.6402853

[11] K. Bilstrup, E. Uhlemann, E. G. Strom, and U. Bilstrup, "Evaluation of the IEEE 802.11p MAC Method for Vehicle-to-Vehicle Communication," in *2008 IEEE 68th Vehicular Technology Conference*, Sep. 2008, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/VETECF.2008.446

[12] X. Jiang and D. H. C. Du, "PTMAC: A Prediction-Based TDMA MAC Protocol for Reducing Packet Collisions in VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9209–9223, Nov. 2016. [Online]. Available: http://dx.doi.org/10.1109/TVT.2016.2519442

[13] C. B. Math, A. Ozgur, S. H. d. Groot, and H. Li, "Data Rate based Congestion Control in V2V communication for traffic safety applications," in *2015 IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov. 2015, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/SCVT.2015.7374229

[14] C. Chen, H. Xiang, C. Sun, and Q. Pei, "A Safety Enhancement Broadcasting Scheme Based on Context Sensing in VANETs," *Mobile Information Systems*, vol. 2016, p. 17, 2016, Art. ID 2630439. [Online]. Available: http://dx.doi.org/10.1155/2016/2630439

[15] Y. P. Fallah, C. L. Huang, R. Sengupta, and H. Krishnan, "Analysis of Information Dissemination in Vehicular Ad-Hoc Networks With Application to Cooperative Vehicle Safety Systems," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 233–247, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1109/TVT.2010.2085022

[16] F. Marzouk, R. Zagrouba, A. Laouiti, and P. Muhlethaler, "An empirical study of Unfairness and Oscillation in ETSI DCC," *International Conference on Performance Evalutaion and Modeling in Wired and Wireless networks PEMWN' 2015*, Nov. 2015, hammamet, Tunisia. [Online]. Available: http://arxiv.org/abs/1510.01125

[17] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Toward Communication Strategies for Platooning: Simulative and Experimental Evaluation," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5411–5423, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1109/TVT.2015.2489459

[18] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for Veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec. 2014, pp. 53–60. [Online]. Available: http://dx.doi.org/10.1109/VNC.2014.7013309

[19] Q. Xu and R. Sengupta, "Simulation, analysis, and comparison of ACC and CACC in highway merging control," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)*, Jun. 2003, pp. 237–242. [Online]. Available: http://dx.doi.org/10.1109/IVS.2003.1212915

[20] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative Adaptive Cruise Control (CACC) Definitions and Operating Concepts," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2489, pp. 145–152, 2015. [Online]. Available: https://www.researchgate.net/publication/273204405_COOPERATIVE_ADAPTIVE_CRUISE_CONTROL_CACC_DEFINITIONS_AND_OPERATING_CONCEPTS

[21] S. Gräfling, P. Mähönen, and J. Riihijärvi, "Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications," in *2010 Second International Conference on Ubiquitous and Future Networks (ICUFN)*, Jun. 2010, pp. 344–348. [Online]. Available: http://dx.doi.org/10.1109/ICUFN.2010.5547184

[22] A. Abdelgader and L. Wu, "The Physical Layer of the IEEE 802.11p WAVE Communication Standard: The Specifications and Challenges," ser. World Congress on Engineering and Computer Science, WCECS 2014, San Francisco, USA, 22 - 24 October, 2014. Hong Kong: IAENG, International Association of Engineers, 2014. [Online]. Available: https://www.researchgate.net/publication/279474688_The_Physical_Layer_of_the_IEEE_80211_p_WAVE_Communication_Standard_The_Specifications_and_Challenges

[23] "Intelligent Transport Systems (ITS); Communications Architecture," *ETSI EN 302 665 V1.1.1*, Sep. 2010. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf

[24] R. Bossom, R. Brignolo, T. Ernst, K. Evensen, A. Frötscher, W. Höfs, J. Jääskeläinen, Z. Jeftic, P. Kompfner, T. Kosch, I. Kulp, A. Kung, A. K. Mokaddem, A. Schalk, E. Uhlemann, and C. Wewetzer, "European ITS Communication Architecture-Overall Framework-Proof of Concept Implementation," *EC FP7 Deliverable, EC Information Society Technologies Programme, Tech. Rep*, Mar. 2009. [Online]. Available: http://www.transport-research.info/sites/default/files/project/documents/20130515_125913_29633_COMeSafety_DEL_D31_EuropeanITSCommunicationArchitecture_v2.0_01.pdf

[25] "Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," *ETSI TS 102 687 V1.1.1*, Jul. 2011. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102600_102699/102687/01.01.01_60/ts_102687v010101p.pdf

[26] F. Bai and H. Krishnan, "Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications," in *2006 IEEE Intelligent Transportation Systems Conference*, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1109/ITSC.2006.1706767

[27] S. B. R. Gottapu, M. Tatineni, and M. Kumar, "Performance analysis of collision alleviating distributed coordination function protocol in congested wireless networks - a Markov chain analysis," *IET Networks*, vol. 2, no. 4, pp. 204–213, Dec. 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6669867/

[28] T. ElBatt, S. K. Goel, G. Holland, H. Krishnan, and J. Parikh, "Cooperative Collision Warning Using Dedicated Short Range Wireless Communications," in *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '06. New York, NY, USA: ACM, 2006, pp. 1–9. [Online]. Available: http://doi.acm.org/10.1145/1161064.1161066

[29] C. Campolo, A. Molinaro, and R. Scopigno, Eds., *Vehicular ad hoc Networks: Standards, Solutions, and Research*. Springer International Publishing, 2015. [Online]. Available: //www.springer.com/fr/book/9783319154961

[30] S. Kuk and H. Kim, "Preventing Unfairness in the ETSI Distributed Congestion Control," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1222–1225, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2014.2320893

[31] A. Weinfield, J. Kenney, and G. Bansal, "An Adaptive DSRC Message Transmission Rate Control Algorithm," *18th World Congress on Intelligent Transport Systems, Intelligent Transportation Society of America*, Oct. 2011. [Online]. Available: https://www.researchgate.net/publication/268097754_An_Adaptive_DSRC_Message_Transmission_Rate_Control_Algorithm

[32] S. Kuk, S. Yang, Y. Park, and H. Kim, "On the Connectivity Problem of ETSI DCC Algorithm," Mar. 2017, arXiv: 1703.05174. [Online]. Available: http://arxiv.org/abs/1703.05174

[33] S. Yang and H. Kim, "Configuring ETSI DCC parameters for better performance and stability," *Electronics Letters*, vol. 52, no. 8, pp. 667–669, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7451450/

[34] L. Zheyuan, "Evaluation of platoon Application Enabled by Contemporary ETSI ITS-G5 Standards," Master's thesis, Halmstad University, Jun. 2015. [Online]. Available: http://www.diva-portal.se/smash/get/diva2:862533/FULLTEXT01.pdf

[35] N. Lyamin, Q. Deng, and A. Vinel, "Study of the Platooning Fuel Efficiency under ETSI ITS-G5 Communications," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 551–556. [Online]. Available: http://dx.doi.org/10.1109/ITSC.2016.7795608

[36] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: A Linear Adaptive Message Rate Algorithm for DSRC Congestion Control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1109/TVT.2013.2275014

[37] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi, and H. Hartenstein, "Design methodology and evaluation of rate adaptation based congestion control for Vehicle Safety Communications," in *2011 IEEE Vehicular Networking Conference (VNC)*, Nov. 2011, pp. 116–123. [Online]. Available: http://dx.doi.org/10.1109/VNC.2011.6117132

[38] G. Bansal, H. Lu, J. B. Kenney, and C. Poellabauer, "EMBARC: error model based adaptive rate control for vehicle-to-vehicle communications," in *Proceeding of the Tenth ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, ser. VANET '13. New York, NY, USA: ACM, 2013, pp. 41–50. [Online]. Available: http://doi.acm.org/10.1145/2482967.2482972

[39] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3684–3703, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1109/TVT.2009.2017545

[40] G. Caizzone, P. Giacomazzi, L. Musumeci, and G. Verticale, "A power control algorithm with high channel availability for vehicular ad hoc networks," in *IEEE International Conference on Communications, 2005. ICC 2005. 2005*, vol. 5, May 2005, pp. 3171–3176 Vol. 5. [Online]. Available: http://dx.doi.org/10.1109/ICC.2005.1494999

[41] A. A. Jimenez Luna, "Evaluation of and Improvement to Decentralized Congestion Control via Transmit Power Control with Information Exchange in ETSI GeoNetworking," Master's thesis, TU Delft, Electrical Engineering, Mathematics and Computer Science, 2016. [Online]. Available: http://resolver.tudelft.nl/uuid:71f4d491-a97e-4bea-a88f-067f9af2923c

[42] C. B. Math, H. Li, S. H. d. Groot, and I. Niemegeers, "Fair decentralized data-rate congestion control for V2V communications," in *2017 24th International Conference on Telecommunications (ICT)*, May 2017, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/ICT.2017.7998229

[43] T. Tielert, D. Jiang, H. Hartenstein, and L. Delgrossi, "Joint power/rate congestion control optimizing packet reception in vehicle safety communications," in *Proceeding of the Tenth ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, ser. VANET '13. New York, NY, USA: ACM, 2013, pp. 51–60. [Online]. Available: http://doi.acm.org/10.1145/2482967.2482968

[44] Y. Wei, "SAE-DCC evaluation and comparison with popular congestion control algorithms of V2X communication," Aug. 2017, Master's thesis, Eindhoven University of Technology, Aug. 2017. [Online]. Available: https://pure.tue.nl/ws/files/89010044/1035011_Yongyi_Wei_Master_thesis.pdf

[45] C. B. Math, H. Li, S. H. d. Groot, and I. Niemegeers, "A combined fair decentralized message-rate and data-rate congestion control for V2V communication," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov. 2017, pp. 271–278.

[46] A. Autolitano, C. Campolo, A. Molinaro, R. M. Scopigno, and A. Vesco, "An insight into Decentralized Congestion Control techniques for VANETs from ETSI TS 102 687 V1.1.1," in *2013 IFIP Wireless Days (WD)*, Nov. 2013, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WD.2013.6686471

[47] L. Le, R. Baldessari, P. Salvador, A. Festag, and W. Zhang, "Performance Evaluation of Beacon Congestion Control Algorithms for VANETs," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Dec. 2011, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2011.6133897

[48] A. Vesco, R. Scopigno, C. Casetti, and C. F. Chiasserini, "Investigating the effectiveness of decentralized congestion control in vehicular networks," in *2013 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 1314–1319. [Online]. Available: http://dx.doi.org/10.1109/GLOCOMW.2013.6825176

[49] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *Proceedings of the European Simulation Multiconference (ESM2001), Prague, 6-9*, Jun. 2011, pp. 1–7. [Online]. Available: https://www.omnetpp.org/download/docs/papers/esm2001-meth48.pdf

[50] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (Simulation of Urban MObility) - An open-source traffic simulation," in *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, 2002, pp. 183–187. [Online]. Available: http://sumo.dlr.de/pdf/dkrajzew_MESM2002_SUMO.pdf

[51] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1109/TMC.2010.133

[52] "Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for operation in the ITS G5A and ITS G5B medium; Validation set-up and results," *ETSI TR 101 613 V1.1.1*, Sep. 2015. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/101600_101699/101613/01.01.01_60/tr_101613v010101p.pdf

[53] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. L. Cigno, and F. Dressler, "How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1411–1421, Jul. 2015. [Online]. Available: https://doi.org/10.1109/TMC.2014.2362752

[54] G. Bansal, B. Cheng, A. Rostami, K. Sjoberg, J. B. Kenney, and M. Gruteser, "Comparing LIMERIC and DCC approaches for VANET channel congestion control," in *2014 IEEE 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014)*, Sep. 2014, pp. 1–7. [Online]. Available: https://doi.org/10.1109/WIVEC.2014.6953217

[55] C. Song, "Performance Analysis of the IEEE 802.11p Multichannel MAC Protocol in Vehicular Ad Hoc Networks." [Online]. Available: http://www.mdpi.com/1424-8220/17/12/2890

[56] L. Stibor, Y. Zang, and H. J. Reumerman, "Evaluation of Communication Distance of Broadcast Messages in a Vehicular Ad-Hoc Network Using IEEE 802.11p," in *2007 IEEE Wireless Communications and Networking Conference*, Mar. 2007, pp. 254–257. [Online]. Available: https://doi.org/10.1109/WCNC.2007.53

[57] S. Santini, A. Salvi, A. S. Valente, A. Pescapé, M. Segata, and R. L. Cigno, "A Consensus-Based Approach for Platooning with Intervehicular Communications and Its Validation in Realistic Scenarios," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1985–1999, Mar. 2017. [Online]. Available: https://doi.org/10.1109/TVT.2016.2585018

[58] Y. Jing, Z. Heng, L. Zetao, D. Dan, L. Xuemei, L. Yi, Z. Yuzhi, W. Xiao, Y. Xin, and C. Hua, "Propagation characteristics in corridor scenario about typical radio wave," in *2017 9th International Conference on Advanced Infocomm Technology (ICAIT)*, Nov. 2017, pp. 5–9. [Online]. Available: https://doi.org/10.1109/ICAIT.2017.8388818