Scalable beaconing for cooperative adaptive cruise control

Emiel Martijn van Eenennaam

Scalable Beaconing for Cooperative Adaptive Cruise Control

E.M. van Eenennaam

Graduation committee:

Chairman:	Prof. dr. ir. Ton Mouthaan
Secretary:	Prof. dr. ir. Ton Mouthaan
Promotor:	Prof. dr. ir. Boudewijn Haverkort
Assistant promotor:	Dr. ir. Geert Heijenk

Referee:	
Dr. ir. Georgios Karagiannis	University of Twente
Members:	
Prof. dr. ir. Kees Slump	University of Twente
Prof. dr. ir. Eric van Berkum	University of Twente
Prof. dr. ir. Bart van Arem	Delft University of Technology
Dr. Alexey Vinel	Halmstad University
Dr. ir. Nirvana Meratnia	University of Twente

CTIT Ph.D.-thesis Series No. 13-277
Centre for Telematics and Information Technology
University of Twente
P.O. Box 217 – 7500 AE Enschede, The Netherlands

ISSN 1381-3617
ISBN 978-90-365-3576-2
DOI 10.3990/1.9789036535762

Cover design: 🄯E.M. van Eenennaam, photo by Bart Klaassen.

This work is supported by the Dutch Senter Novem/HTAS (High Tech Automotive Systems) Project Connect&Drive, Project no. HTASD08002.

Printed by Gildeprint Drukkerijen, Enschede, The Netherlands.

SCALABLE BEACONING
FOR
COOPERATIVE ADAPTIVE CRUISE CONTROL

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
Prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties,
in het openbaar te verdedigen
op woensdag 20 november 2013 om 12:45 uur

door

Emiel Martijn van Eenennaam

geboren op 19 februari 1982
te Nijmegen, Nederland

Dit proefschrift is goedgekeurd door:
Prof. dr. ir. Boudewijn R. Haverkort (promotor)
Dr. ir. Geert Heijenk (assistent-promotor)

> Proofread carefully to see if you any words out.

Dave Barry

> People are so bad at driving cars that computers don’t have to be that good to be much better.

Marc Andreessen

> We live as we do
> to show the world what it could be
> What’s it mean to you?
> Is this a vision you can see?
> Heroes are those who
> don’t just accept the way things are
> Now, which one are you, the driver or the car?

Machinae Supremacy – Overwold

> Caffeine catalyses the scientific process, but only to a certain optimum.
> After that, the process becomes saturated and efficiency plummets.
> It is more-or-less comparable to road- or network traffic throughput.
> (see also Fig. 2.3 on page 16)

Martijn van Eenennaam

Abstract

Over the past two hundred years, automotive technology has evolved from mechanised horse carriage to high-tech systems which pack more computing power than the entire space program that put Neil Armstrong on the moon. Hand-in-hand with this evolution came a proliferation of ownership and use of cars. This enormous success causes one of modern society's largest problems: where many vehicles accumulate, traffic congestion occurs.

To a large degree, the cause of traffic congestion lies in the poor ability of the human driver to control the (longitudinal) motion of the vehicle under congested traffic circumstances. This leads to so-called string instabilities or shock waves, traveling against the flow of traffic. The traffic flow performance can be improved if the control of acceleration and deceleration is automated. Presently available solutions use radar or lidar to detect and measure the distance to the vehicle in front, and a cruise controller automatically reacts by adjusting the vehicle speed. However, the performance of these systems is not sufficient to prevent shock waves, predominantly due to the delay introduced by the sensors.

The Cooperative Adaptive Cruise Control (CACC) is a system which circumvents this by using wireless communication to exchange information about vehicle dynamics using the periodic transmission of so-called beacon messages. The technology proposed for this wireless communication is IEEE 802.11p, a modified version of the IEEE 802.11a designed for Wireless LAN applications. However, the wireless medium succumbs to a congested state in a similar fashion as the traffic on the road in response to an increase of the traffic density.

This dissertation focusses on the beaconing communication, used to generate a cooperative awareness in each vehicle. Given the real-time nature of the CACC system, it is important that the information in the cooperative awareness is accurate and fresh, even under an increasing number of communicating nodes in near vicinity. To this end, beaconing is evaluated through analytical modelling, discrete-event simulation and proof-of-concept implementations. The purpose is to determine the scalability limits of the IEEE 802.11p Medium Access Control mechanism when used for beaconing, and find and address bottlenecks.

In this disseration, detailed analytical models of the Distributed Coordination Function (DCF) and the Enhanced Distributed Channel Access (EDCA) are proposed, validated, and compared. Various mechanisms which impact the scalability of a beaconing system are described and evaluated using both these analytical and simulation models. In particular, an extensive comparison between the DCF and EDCA access mechanism variants of IEEE 802.11 is performed based on their performance

in the face of increasing traffic density. The conclusion is that, although IEEE 802.11p defines the EDCA to be used, the DCF is a more favourable access mechanism for the broadcast transmission of beacon messages.

Under both access methods, the use of the EIFS is found to be redundant because the beacon channel is broadcast-only. Furthermore, the periodic channel switching defined by IEEE 1609.4, which defines a way to use single-radio IEEE 802.11p on multiple channels in a time-division fashion, has a detrimental effect on beaconing performance. In addition, the way beacon messages are buffered and scheduled for transmission is evaluated. We conclude that it is beneficial to use a dropping policy which drops the oldest information in the queue, as opposed to the most recent arrival as is often implemented. This method is coined the Oldest Packet Drop (OPD) mechanism and is described and evaluated in detail.

The outlook of a CACC application operating on beacons transmitted using IEEE 802.11p is good. However, care has to be taken that the system does not become congested. This dissertation provides a set of tools to estimate when the channel becomes congested, and to evaluate the impact of various design choices on communication performance.

Samenvatting

In de afgelopen tweehonderd jaar heeft de automobieltechnologie enorme sprongen voorwaarts gemaakt, van gemechaniseerde koets tot technologisch-hoogwaardige systemen welke over meer rekenkracht beschikken dan het volledige ruimteprogramma dat Neil Armstrong op de maan zette. Hand in hand met deze evolutie, ging een groei in zowel bezit als gebruik. Dit enorme succes veroorzaakt één van de grootste problemen van onze hedendaagse maatschappij: waar veel auto's samenkomen, ontstaat inherent file.

De oorzaak van file is voor een groot gedeelte te herleiden tot het slechte vermogen van de menselijke bestuurder om de (longitudinale) beweging van het voertuig te beheersen onder toenemende verkeersdrukte. Dit heeft zogenaamde keten instabiliteiten tot gevolg, ook wel bekend als schokgolven, die zich tegen de verkeersstroom in bewegen. De verkeersdoorstroming zou verbeterd kunnen worden als controle over acceleratie en deceleratie van het voertuig wordt geautomatiseerd. Huidige systemen gebruiken radar of lidar om de afstand tot de voorligger te meten, en een cruise controller reageert automatisch door de snelheid aan te passen. Deze systemen presteren helaas niet toereikend om het ontstaan van schokgolven te voorkomen, hoofdzakelijk door de vertraging in de sensoren.

De Cooperative Adaptive Cruise Control (CACC) is een system dat deze vertraging omzeilt door de inzet van draadloze communicatie, om information uit te wisselen omtrent voertuigdynamica. Dit gebeurt doormiddel van periodiek verstuurde beacon (baken) berichten. De technologie die voorgesteld is om dit te doen is IEEE 802.11p, een aanpassing op IEEE 802.11a die ontworpen is voor Wireless LAN toepassingen. Echter, als reactie op een toename van de verkeersdruk geraakt het draadloze medium in een staat van overbelasting vergelijkbaar met het ontstaan van een file op de weg.

Dit proefschrift behandelt de beaconing communicatie die gebruikt wordt om een coöperatief bewustzijn te creëren in ieder voertuig. Gegeven de real-time aard van het CACC systeem is het belangrijk dat de informatie in dit bewustzijn accuraat en actueel is, zelfs onder een toenemend aantal communicerende voertuigen in de nabije omgeving. Hiertoe evalueren we het beaconing communicatie patroon door middel van analytische modellen, discrete-event simulatie, en prototype implementaties. Het doel is om de schaalbaarheidslimieten van het IEEE 802.11p Medium Access Control mechanisme in kaart te brengen, en knelpunten te vinden en aan te pakken.

Dit proefschrift beschrijft gedetaileerde analytische modellen van de Distributed Coordination Function (DCF) en Enhanced Distributed Channel Access (EDCA), valideert deze en beschrijft vervolgens een vergelijking tussen de DCF en EDCA

toegangsmechanismen. Verscheidene mechanismen die invloed hebben op de schaalbaarheid van een beaconing systeem worden beschreven en geëvalueerd aan de hand van de ontwikkelde analytische- en simulatiemodellen. In het bijzonder wordt een uitvoerige vergelijking van DCF en EDCA zoals gebruikt in IEEE 802.11p beschreven, op basis van de prestaties onder toenemende verkeersdichtheid. De conclusie is dat, ook al is het gebruik van EDCA in IEEE 802.11p voorgeschreven, de DCF betere prestaties blijkt te leveren voor de broadcast transmissies van beacon berichten.

Onder beide toegangsmechanismen is het gebruik van de EIFS overbodig, daar het beacon communicatiekanaal uitsluitend broadcast-berichten bevat. Verder wordt het periodiek van kanaal wisselen, zoals beschreven in IEEE 1609.4 en bedoeld om met één enkele radio toch meerdere kanalen “tegelijk” te kunnen gebruiken, geëvalueerd. Dit blijkt een zeer nadelig effect te hebben op de prestaties van het beaconing systeem. Daarnaast wordt de manier waarop beacon berichten gebufferd en transmissies gepland worden belicht. De conclusie van deze studie is dat het zinvol is om, in het geval de buffer vol zit, het oudste pakket weg te gooien. Dit in tegenstelling tot wat algemeen gangbaar is: het weggooien van het nieuwe pakket.

De vooruitzichten voor een CACC toepassing welke gebruikmaakt van beacon berichten verstuurd met IEEE 802.11p zijn goed. Maar niettemin moet er zorg voor gedragen worden dat het systeem niet overbelast raakt. Dit proefschrift reikt een set gereedschappen aan om een inschatting te maken van wanneer het kanaal overbelast raakt, en om de impact van ontwerpbeslissingen op de communicatie prestaties te kunnen evalueren.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Traditionally, automobiles rely on a human driver to function as both the sensory and control component, where the vehicle is just a machine accepting inputs from the driver such as throttle and brake directives. The human is completely *in the loop* and in full control of the vehicle. With this level of control, the performance of the vehicle is also limited by human limits in the field of vision, the delay between perception and action, and the correctness of this action.

Modern vehicles gradually shift some of the sensory and control tasks from the driver to the vehicle. This is done using on-board sensors such as radar, cameras, and ultrasound in conjunction with control systems running on increasingly powerful hardware. Examples include the Anti-lock Breaking System (ABS), Electronic Stability Program (ESP), and park assistance systems. Using these systems, vehicles have increased performance with respect to the granularity with which actuators can be engaged at just the right moment. One could say there is an increased *awareness* inside the vehicle's systems of what is going on inside the vehicle, and in its close proximity.

Despite the above improvements, on-board sensors have limits in both range and accuracy of this awareness when it comes to perceiving the world outside the vehicle. These limits can be dealt with by obtaining information from locations where it is available by means of wireless communication, rather than attempting to obtain it with just local sensors. These locations could be roadside devices, or other vehicles. In this concept, vehicles share information by means of small status messages called *beacons*, which they broadcast several times per second using radio communication. These status messages contain information such as the current velocity, acceleration, and position of the vehicle, and are defined in open standards so that every vehicle can understand them and make use of the information.

In addition to transmitting these messages, every vehicle also receives these messages from its surrounding vehicles. Based on these messages, a vehicle can create a so-called *cooperative awareness*—a virtual representation of the world around itself, beyond what would be possible when relying on just on-board sensors alone. Many applications can use this cooperative awareness, ranging from infotainment to sophisticated driver support systems.

But just as the human driver and on-board sensors are limited in range and field of view, the radio communication also has limits. This dissertation focusses on the performance of the communication system within these limits. The application dis-

Figure 1.1: A CACC-equipped vehicle automatically responds to the braking vehicle in front [1].

cussed in this work is Cooperative Adaptive Cruise Control (CACC), an Advanced Driver Assistance System (ADAS) which has particularly strong requirements on the underlying communication system because of its real-time nature. The reasoning is that if the requirements posed by CACC can be met, applications with more lenient requirements can readily be implemented.

CACC is a next-generation cruise control system which regulates the vehicle's velocity in cooperation with surrounding vehicles, achieving string stability in traffic flows. This is illustrated in Fig. 1.1, where a vehicle responds to deceleration of the vehicle in front. Its main purpose is to prevent the undesirable shockwaves traveling through traffic jams in the upstream direction; the typical stop-and-go motion every driver is familiar with. It is expected that with the adoption of CACC, traffic congestion can be significantly reduced. This will lead to a cleaner and more efficient transportation system.

1.1 Beaconing in vehicular networks

To have cars communicate with each other, they must all "speak the same language". For comparison, the enormous success of the Internet could not have been realised without open standards: devices operate according to standardised protocols, but their implementations leave plenty of freedom. Likewise, communication between vehicles should be according to open standards—only then can each vehicle communicate with every other vehicle, and can the power of the future smart car truely be leveraged.

The periodic broadcasting of status messages by future vehicles is known as *beaconing*. A beacon message contains static information, such as the vehicle length, and dynamic information, such as vehicle speed and acceleration. This is illustrated in Fig. 1.2: the awareness of vehicle i extends beyond what it measures with its on-board sensors. Although only illustrated in the longitudinal direction here, the awareness covers the full 360 degrees around a vehicle.

The communication technology envisioned to be used for this purpose is IEEE 802.11p, a member of the successful 802.11 family most people will be more familiar with under the name of Wireless LAN or WiFi [2]. Although originally designed for Internet and e-mail communication between computers in an office setup, the "p" amendment holds promise for application in the dynamic environment of vehicular networking.

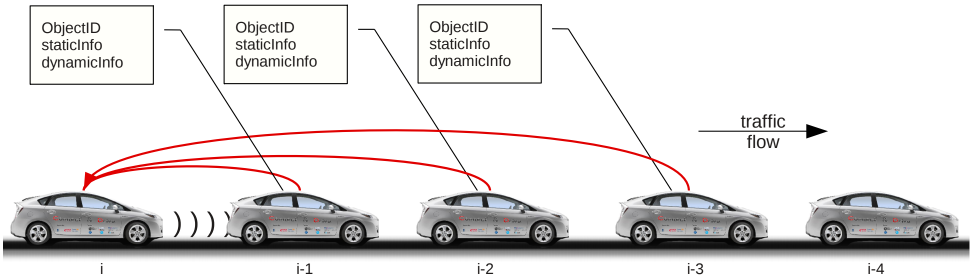


Figure 1.2: Beaconing: by cooperating, vehicles become awarene of their surroundings.

In the technology and architectures currently under standardisation, beaconing plays a central role. IEEE 1609 [3], ETSI ITS [4], and ISO CALM [5] all specify the use of these periodically transmitted status messages. Dedicated spectral resources have been reserved for vehicular networking both in the United States and in Europe, and many other countries are following suit. Within this frequency band, several channels are defined of which one is dedicated to the exchange of safety-critical, low-latency information. However, even with a dedicated channel it is still paramount to efficiently use the available resources in order for the communication to be effective. Applications such as CACC must operate properly in the face of the dynamic vehicular environment, in which a large number of nodes can be involved in the communications. In other words, the system must be scalable.

Transmisson range is a limiting factor in wireless networks, but given the nature of the CACC application, not the most critical one. If the other vehicle is far away, the relation between the two vehicles from a traffic point-of-view is weak. This relation—and the level of influence the lead vehicle can have on the following vehicle—increases in magnitude as the distance between vehicles diminishes. The vehicles are now well within each other's transmission range, and another factor limits the communication performance: the mechanism which governs access to the shared wireless medium. It is this Medium Access Control (MAC) which is the focus of this dissertation.

1.2 Research objectives and scope

The main goal of this dissertation is to evaluate the performance of IEEE 802.11p when used for beaconing in vehicular networks, and to develop and evaluate methods to improve scalability. In this respect, using the received information for real-time vehicle control is regarded as one of the most challenging applications. CACC is expected to have great potential to increase the performance, safety, and efficiency of transportation, and hence has great societal merit [6]. By focussing on the beaconing communication pattern, we consider only the communication of control and safety information, stating that this is the most challenging type of traffic. Infotainment applications are not considered in this work, it is assumed that their delay tolerance allows them to either use the other channels available for vehicular networking, or cellular technologies.

In a vehicular network, the beaconing system departs from traditional communication patterns used in the Internet or in cellular networks. The communication is completely ad hoc, which means that the channel access method is distributed, instead of centrally coordinated as is the case in typical cellular systems. In addition, the number of nodes can be significantly larger than in traditional Wireless LAN deployments.

In contrast to the Internet, which is largely based on the client-server paradigm using one-to-one communication, the messaging nature of beaconing is that of broadcast in a many-to-many scenario. The fact that the wireless technology envisioned for use in this context (IEEE 802.11p) is a member of the WiFi family, does not directly imply that what is known about IEEE 802.11 does also apply to beaconing in a vehicular network.

The main **goal** of this dissertation can be broken down into the following three objectives:

1. Determine the performance limits of beaconing in vehicular networking.
2. Study the performance of beaconing in vehicular networks within these limits.
3. Propose and evaluate methods to improve scalability and dependability of beaconing in vehicular networks.

To achieve these objectives, we will answer the following **research questions**:

RQ1: What are the requirements of CACC on the communication system?

RQ2: How do we evaluate whether a beaconing system is operating such that it meets the requirements? Which metrics can be used?

RQ3: What are the limiting factors of the performance of a beaconing system?

RQ4: How does IEEE 802.11p scale under varying traffic densities, and can it meet the requirements imposed by CACC?

RQ5: How do the scalability limits found in simulation, analysis and proof-of-concept implementations relate?

RQ6: What can be done to increase the scalability properties of IEEE 802.11p?

1.3 Research approach

This research combines three distinct methods to answer the posed research questions:

- **Stochastic modelling** of the channel state and the Medium Access Controller in each node by means of a Discrete Time Markov Chain (DTMC) to estimate key metrics.
- **Discrete-event simulation** of a beaconing system for the purpose of exploring its behaviour and of evaluating various proposed modifications.
- **Proof-of-Concept implementations** and measurements to show technical feasibility.

These three methods differ in approach and each focusses on different aspects of a beaconing system. Each method has certain benefits as well as drawbacks. Combination of the three allows a more complete, unbiased insight.

The research uses a top-down approach, starting with the application and then moving down to the network technology. Application-level performance is evaluated using CACC proof-of-concept implementations and simulation studies. An implementation in seven similar vehicles has been developed in the Connect&Drive project [7]. An implementation in one of ten heterogeneous vehicles has been developed in the Grand Cooperative Driving Challenge (GCDC) project [8]. Network-level performance is evaluated by means of analytical models which accurately predict the behaviour of a beaconing system, and extensive discrete-event simulation studies. Various standardisation efforts have been ongoing during the writing this text. The aim of this research is to provide an analysis of the concepts, independent of specific architectures. Where needed, assumptions on such an architecture are based on actual standards but can easily be generalised.

1.4 Contributions

This work focusses on the performance of beaconing using IEEE 802.11p and its application in real-time vehicle control. The performance of the CACC application can be better than that of a human driver, however, there is a strong dependence on the quality of the information fed into the control system. The CACC system is primarily designed to operate in dense traffic, with the aim of increasing traffic throughput in a regime where the human driver can no longer cope. This appears to be in contradiction to the wireless networking performance, which deteriorates with an increasing vehicle density. The contributions of this work and its related publications can be summarised as follows:

- Provide insight into the operation of IEEE 802.11p by means of detailed simulation studies, and describe phenomena typical when using this technology for CACC's communication needs.
- Provide a first step to application-level performance evaluation of a CACC system including wireless communication and vehicle motion.
- Provide an analytical model which accurately predicts the behaviour of beaconing using IEEE 802.11p DCF and EDCA. Based on the behaviour of a single node, modelled as a Discrete Time Markov Chain (DTMC), the behaviour of the entire system encompassing many nodes is approximated. In particular, this model allows success probability and delay estimations, in addition to estimation of the point where the 802.11 channel access mechanism starts to deteriorate.
- Evaluate IEEE 802.11p in large-scale context through simulation and analytical modelling.
- Describe weaknesses of IEEE 802.11p when used for the communication of real-time control information.
- Propose several improvements which could be applied to improve the performance of beaconing in vehicular networking.

Contributions of this dissertation were previously published in peer-reviewed conferences [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], and journals [20, 21], and several technical reports [22, 23, 24, 25, 26].

1.5 Outline of this dissertation

The remainder of this dissertation follows a top-down approach. An application-level focus evaluates CACC performance and its relation to networking performance. Then, the focus shifts to the performance of the wireless networking itself. Fig. 1.3 illustrates this structure, and also shows the publications on which the chapters are based. The figure also shows which research question is treated in each chapter.

Essential concepts are introduced in **Chapter 2: Background**. Using a top-down approach, the concept of Intelligent Transportation Systems (ITS) is introduced, and the application of CACC is described in detail. In order for many ITS applications

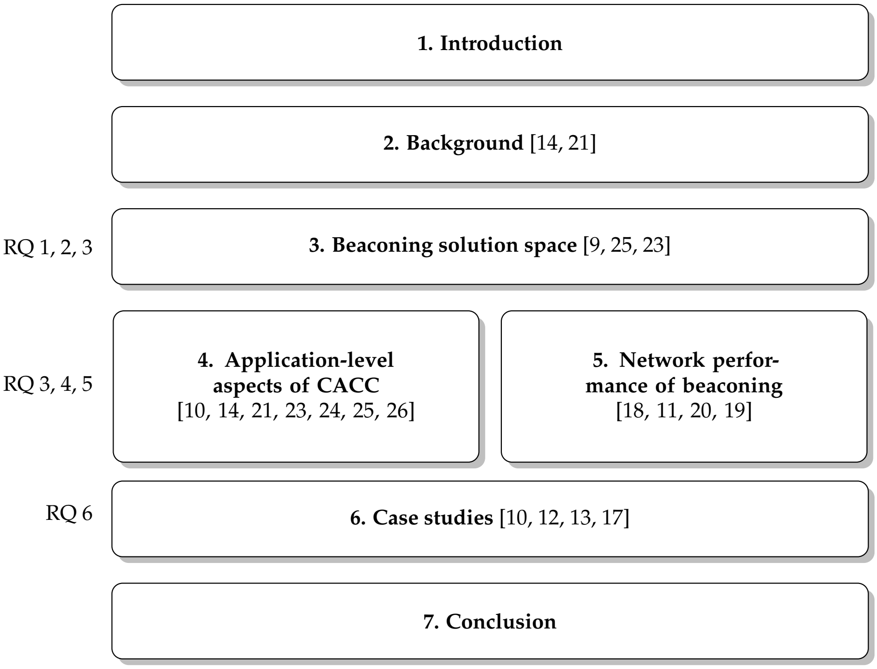

1. Introduction
2. Background [14, 21]
RQ 1, 2, 3
3. Beaconing solution space [9, 25, 23]
RQ 3, 4, 5
4. Application-level aspects of CACC [10, 14, 21, 23, 24, 25, 26]
5. Network performance of beaconing [18, 11, 20, 19]
RQ 6
6. Case studies [10, 12, 13, 17]
7. Conclusion

Figure 1.3: The structure of this dissertation.

to work, information needs to be exchanged. Vehicular networking is an enabling technology to exchange this information. Then, the means of achieving this vehicular networking are discussed, starting with Network Architectures. Because this dissertation focusses on the IEEE 802.11p MAC mechanism, the technology of the MAC and Physical layer are described in detail. Finally, this chapter introduces the simulation platforms and experimental setups used throughout this dissertation. The chapter concludes with a brief summary.

Chapter 3: Beaconing Solution Space defines beaconing and describes how beaconing can be used to construct a cooperative awareness in each vehicle. Then, the factors limiting the scalability of a beaconing system are treated in a conceptual manner. The requirements of a CACC system are covered, as are the metrics which can be used to gauge the performance of a beaconing system. Finally, a simple channel utilisation model is presented to estimate best-case performance and define the solution space within which beaconing operates. Chapter 3 answers research questions 1, 2, and provides a partial answer to research question 3.

Next, the viability of a beaconing system to support CACC is studied from application and network perspective in the following two chapters, and a refinement to research question 3 is provided. Additionally, research question 4 and 5 are answered. Starting in **Chapter 4: Application-level aspects of CACC**, a simple beaconing system is implemented in seven vehicles to build a CACC system in the Connect&Drive project [7]. In order to get another step closer to the multi-vendor, heterogeneous traffic environment encountered in real life, a CACC system is implemented in a single vehicle to participate in the Grand Cooperative Driving Challenge (GCDC) [8]. Because these small-scale deployments stay well within the scalability limits of the wireless networking technology, more vehicles are needed to study beaconing near the edges of the solution space. This is, however, a costly enterprise and very difficult to manage, so we turn to simulation. The impact of packet loss on CACC performance is studied in a simulation study, showing a significant sensitivity to networking performance.

Because simulations of this scale are very time-consuming and computationally intensive, an analytical model for beaconing in vehicular networks is developed in **Chapter 5: Network performance of beaconing**. This model allows quick numerical computation and is evaluated against simulation results, showing a good match. Detailed models of the DCF and EDCA MAC are presented, capturing system behaviour typical to these access mechanisms in the broadcast environment of vehicular networking. Next, a comparison of the DCF and EDCA access methods is performed.

Several aspects related to the scalability of beaconing are described in **Chapter 6: Case studies**, this chapter answers research question 6. Simulation studies report on the impact of the size of the contention window on beaconing performance. Then, the influence of way beacon messages are buffered and scheduled is studied. Next, the impact of the proposed multi-channel operations in IEEE 1609.4 when using a single radio is evaluated. Finally, **Chapter 7** reports the **Conclusions** of this dissertation, gives recommendations and provides an outlook on future work.

CHAPTER 2

Background

Chapter 1 briefly introduced beaconing in a vehicular network, and its application to realise a CACC system. The current chapter describes these concepts in detail and establishes a general background used throughout the rest of this dissertation.

Intelligent Transportation Systems (ITS) are expected to revolutionise movement of people and goods around the globe. A brief introduction to ITS and its societal impact is provided in Sec. 2.1. Automation, control and communication are central concepts in ITS. The Cooperative Adaptive Cruise Control (CACC), an ITS application expected to increase road traffic performance, is described in Sec. 2.2. This application is used because it is both iconic in the way it provides merits for transportation—to an extent not possible without the use of vehicular networking—and demanding in its requirements on the information exchanged by the wireless communication. The section concludes with describing two projects implementing CACC. In Sec. 2.3, the central role of vehicular networking in ITS is highlighted. This work focusses on Vehicle-to-Vehicle (V2V) communication. Then we zoom in on how V2V communication can be established. The ongoing standardisation in various standardisation bodies with respect to network layer architectures is briefly described. These upcoming standards allow for the use of IEEE 802.11p technology at the Medium Access Control (MAC) and Physical (PHY) layer.

The largest part of this chapter is dedicated to an in-depth description of medium access within the 802.11 family, starting in Sec. 2.4. The peculiarities of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) channel access scheme employed in IEEE 802.11 have great impact on the performance. We delve into the IEEE 802.11p MAC and PHY specifics in Sec. 2.5 and 2.6 in preparation of evaluations and discussions in the remainder of this dissertation.

Sec. 2.7 describes OMNeT++/MiXiM, the simulation framework used throughout this dissertation to evaluate various metrics in a beaconing system. In addition, several modifications and extensions have been made to perform simulation of an IEEE 802.11p vehicular network, as described in Sec. 5.5.1. A summary is provided in Sec. 2.8.

This chapter contains material previously published in [14, 21].

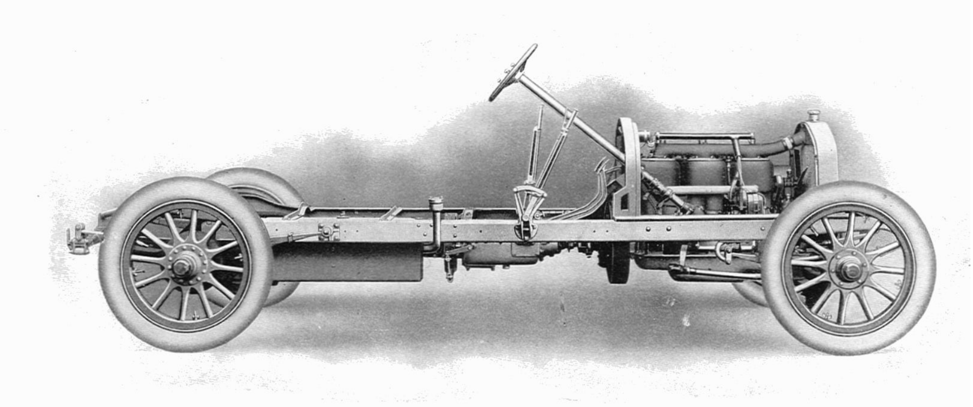


Figure 2.1: Daimler car chassis, dating back to 1905, copied from [27].

2.1 Intelligent transportation systems

Transport has transitioned from moving by foot or horse to the multitude of transportation systems available today. Paths between important locations were upgraded from dirt road, to brick, to asphalt. Transportation has moved towards a globally connected network, where a great number of people and a phenomenal quantity of goods cover large distances on a daily basis. Many systems were developed to meet these transportation needs: airplanes, trains, ships, and road vehicles. All of these systems are being continually improved and become increasingly more intelligent because of the application of information technologies. This trend is referred to as Intelligent Transportation Systems (ITS).

The main objective of ITS is to make transportation more efficient by reducing waste of resources, automating tasks, and improving coordination between hitherto separate systems. The focus of ITS can be on a single system, such as (partial) automation of traffic control systems for highway or railway networks. In a broader scope, the focus can also be on coordination between modes of transportation or with a focus on the traveler, who wants for example to seemlessly transfer from bus to train. Only the application of ITS in road vehicles is considered in this dissertation.

For many years since its inception over two hundred years ago [28], the automobile has been a frame with wheels and an engine. In a sense, it was a mechanised horse carriage. The human operator has to operate the vehicle and its components, and perceive the context such as the road and other road users. The role of the human here is to sense the environment—primarily through vision—and control the vehicle and its subsystems.

The past decades have seen various improvements in automotive technology, increasing both comfort, safety, and efficiency of the vehicles. *Comfort* is increased by on-board systems which take away arduous tasks from the driver; examples include Cruise Control, automatic gearboxes and automatic windshield wipers which turn on the moment the first drops of rain hit the windshield. *Safety* is increased by, for instance, introduction of the ABS which allows coordination of braking power beyond a human's capability, effectively reducing braking distance and preventing vehicles from skidding out of control. *Efficiency* is improved with various tweaks to the engine and drivetrain, such as automatic gearboxes which outperform a human in choosing the right moment to shift gears and the speed with which they do so. The recently introduced start-stop systems, which disable the engine to save fuel when the vehicle is not moving for a while, and the emergence of hybrid or fully electrical vehicles are other examples of an increase in efficiency. With increasing proliferation of information technology, the future vehicle can become even more comfortable. Safety can be increased and the entire system, not just a single vehicle, will become more efficient.

Throughout most of automotive history, the improvements have been vehicle-centric; they all rely on on-board sensors and actuators. Such a vehicle is perfectly able to operate in isolation. Purchasing such a system—usually at a premium price—directly endows the driver's car with pleasant features such as air conditioning and park assistance systems. Using sensors placed throughout the vehicle, the vehicle's performance can be boosted. When taken to the extreme, this results in so-called *autonomous vehicles*, as those developed for the DARPA Grand Challenges [29] and by Google [30], which can operate without a human driver. They perceive the environment using an extensive set of sensors and do not rely on any other entity.

However, most often cars are not used in isolation—they tend to be used in situations where many other cars accumulate. Vehicles are used by human beings whose lifes are increasingly becoming attached to all sorts of virtual services and communities. This trend also trickles into the automotive world. With the advent of computer networking comes the concept of information sharing; ubiquitous access through cellular technologies allows navigation systems to be updated on-the-fly. Wireless communication between vehicles and road-side installations allows the exchange of a wide range of information. This information, in turn, could allow better decisions to be made in a more timely manner. For example, one could think of the dissemination of information about road construction work, enabling an in-vehicle navigation system to reroute instead of finding a blocked or congested street. An example on a more critical timescale could be a vehicle driving on the highway receiving a warning about some debris on the left lane at a position a few hundred meters ahead. We refer to this approach as a network-centric one, yielding *cooperative vehicles*. In contrast to autonomous vehicles, cooperative vehicles share information by applying networking technologies. Whereas the autonomous vehicle could be thought of as an explorer trying to make her way through a jungle, the cooperative vehicle could be thought of as a member in a flock of birds. By sharing information, movements can be coordinated.

In contrast to on-board systems designed according to the vehicle-centric ideology, which provide direct merit in response to the investment, systems which rely on external information depend on the quality and availability of the external information. In addition, the ability to access this information also plays an important role. This departs from the traditional automotive way of doing business, where paying a premium price means getting something extra. In this light, the autonomous vehicle is in line with the traditional automotive busines model, whereas the cooperative vehicle is more in line with the philosophy behind the Internet, based on open communication using clearly defined interfaces. In this case, no matter how premium a price a consumer is willing to pay, there is a strong dependence on the degree at which his vehicle is able to cooperate with the environment.

ITS is a system-of-systems [31]—where at many different levels, systems from different manufacturers, owners, and administrators interact with each other in a trans-domain manner. The scope of such a system can be very broad. Consider for example a web-based platform which aids in exploiting multi-modality in personal travel, where public transit, taxi services and privately owned vehicles are all tied together in a choreography of traffic movements to provide transportation as a service to subscribers. This application would span many components, corporative and even legislative domains. The focus of ITS can also be very detailed, but limited in scope, such as an application in a vehicle which shows the status of the next traffic light on the current route.

The autonomous and cooperative vehicle concepts are not mutually exclusive. In a sense, every present-day vehicle is autonomous to a certain degree. By adding communication capabilities it becomes cooperative but can still function autonomously. In fact, with the addition of cooperation capabilities, vehicles will become even more able to perform driving tasks without interaction with the human driver. This move towards Automated Highway Systems (AHS) [32] has been in the domain of Science Fiction movies for decades, but is now becoming reality.

The remainder of this dissertation focusses on cooperative road vehicles, an important component of ITS.

2.2 Cooperative adaptive cruise control

Wirelessly communicated information can be applied in longitudinal vehicle control, also sometimes referred to as vehicle platooning, or in the context of this dissertation as CACC. CACC is an ITS application which can have great impact on traffic efficiency and safety, while also increasing comfort for the driver and passengers and reducing emissions. This application is the context within which all the analysis and evaluation steps in this dissertation have been performed.

2.2.1 String stability

Traditionally, the human driver controls the automobile. In the longitudinal domain this corresponds to perception of the distance to the vehicle in front, and subsequent control of the vehicle's engine and brakes. This paradigm is simple to implement and functions sufficiently well under low traffic densities; but as the traffic density on the road increases, the human driver increasingly becomes a bottleneck.

Anyone who has ever been a driver or passenger in dense traffic will be familiar with the phenomenon of traffic congestion. Often, this congestion is caused by the inability of the human driver to operate under these conditions: the delays inherent in human reactions are too large. When a disturbance is introduced into the traffic flow—for instance, by a braking vehicle—the following driver responds with some delay. In order to compensate for the delay, this driver brakes more than the previous one, and so does the next driver. This creates a shockwave which travels upstream, against the flow of traffic. Ultimately, this could result in traffic coming to a complete stand-still (best case) or head-tail collisions (worst case). This phenomenon is known as a *phantom traffic jam* [33]. Such behaviour is primarily caused by (un)stability in a dynamical system.

A stable dynamical system will, once perturbed from an equilibrium state, return to the equilibrium state. Manual traffic flow is generally unstable and can lead to spontaneous congestions [34], even in the absence of a bottleneck, causing phantom traffic jams. This sort of stability is refered to as local, string, or traffic flow stability, depending on the scope. *Local stability* is concerned with two vehicles following each other; *string (or platoon) stability* is concerned with propagation of disturbances in a string of multiple vehicles in the longitudinal direction. *Traffic flow stability* generalises this to a traffic flow comprising multiple lanes and interactions between the lanes. In this dissertation, we use string stability as defined in [34, 35]:

Definition 1. *String Stability: Any nonzero position, speed, and acceleration errors of an individual vehicle in a string do not amplify when they propagate upstream.*

Human drivers do not exhibit string stable behaviour when instructed to follow the vehicle in front, as clearly visible in the video referenced in [33]. This is due to limits in field of vision, but primarily due to the relatively long delay between sensing and the corresponding control response. The system will perform acceptable if the time-headway between two vehicles is sufficiently large to accomodate this long delay. A typical time-headway of a human driver is approximately 1.5 s [36], leading to the popular recommendation of two seconds of space to the vehicle in front. The acceptable time-headway differs among drivers and varies between 1s and 2 s for a range of traffic speeds [37]. Although a minimum time-headway of 1 s is within the limit of typical reaction time for braking by alert drivers, [37] identifies that this could lead to occasional accidents given variability in reaction times, decisions, and vehicle braking capabilities.

According to [6, 38], the vehicle speed should be taken as a basis for string stability, which is more relevant than distance error in view of traffic analysis. A simple scenario which can be used to explain string stability is illustrated in Fig. 2.2 (a) and (b). A string of vehicles is shown, moving from left to right. For a clear illustration we show only four vehicles, but the concept also holds for strings of more vehicles. The leading vehicle is denoted as 1^{st} while the last vehicle is denoted as 4^{th}. In each of these figures, below the shown string of vehicles, a speed vs. time plot for each vehicle is shown. As time goes by, the leading vehicle decelerates linearly and we can see different responses of the following vehicles in the platoon depending on whether the platoon is string stable or not.

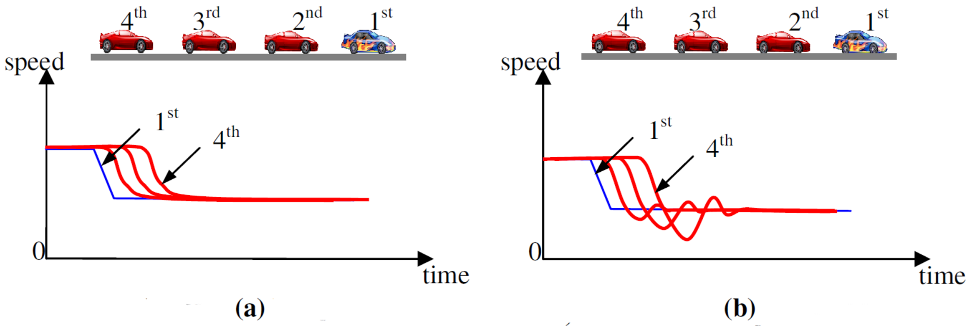


Figure 2.2: String Stability (a) stable and (b) unstable, copied from [34].

In Fig. 2.2 (a), the situation is shown where the platoon is string stable: the effect of the change in speed of the leading vehicle is not amplified through the following vehicles, the deceleration of following vehicles is smooth without any fluctuation of the speed. In Fig. 2.2 (b), the platoon is considered not string stable. It is said to be string unstable: the following vehicles decelerate even more than the leading vehicle, showing overshoot. Although finally the speeds of the following vehicles approach the leading vehicle's speed, there is a period in which speed fluctuates significantly. These fluctuations are amplified from vehicle to vehicle in the upstream direction. Actually, during the period of fluctuation, the distance between neighbouring vehicles also fluctuates and may become very small. As a result, head-tail collisions between vehicles are more likely to happen. This example shows a decelerating lead vehicle, but the concept holds similarly for an accelerating lead vehicle.

2.2.2 Achieving string stability with CACC

A control system deals with the behaviour of dynamical systems [39]. It regulates a variable to a reference value. Although first implementations of longitudinal vehicle control were targeted at maintaining cruise speed, more advanced incarnations slowly found their way onto the market over the past tens of years. The first commercially available car equipped with Cruise Control was the Chrysler Imperial in 1958. When set at a certain cruise speed by the driver, the vehicle maintains this speed. Though a comfort enhancing feature in sparse traffic, it is of little use in dense traffic. Moreover, it still relies on the human driver to judge the distance to, and acceleration of the lead vehicle and take corrective action—such as reducing speed when approaching a slower vehicle.

Around 1993 the Autonomous Intelligent Cruise Control [36] emerged. In 1995 Mitsubishi introduced the Diamante, using a lidar to measure the distance to the vehicle in front. Two years later, Toyota introduced a system based on radar in the Celsior model (sold in Europe as Lexus LS). These implementations of the Adaptive Cruise Control (ACC) were primarily comfort features. Even up to the date of this writing, implementations of ACC are sold as ’luxury’ add-ons at premium price.

In ACC, the control objective is to realize a desired distance to the preceding vehicle. This desired distance may be an increasing function of vehicle velocity in order to take safety aspects into account. The result is commonly referred to as a *constant time-headway spacing policy* [40, 41]. The acceleration of the preceding vehicle can be estimated from the distance and relative velocity as measured by radar or lidar, and this will cause a set-point change in the cruise controller. The result is automatic acceleration or deceleration in response to a change in velocity of the lead vehicle.

ACC is not string stable because the on-board sensors have a high latency in detecting relative velocity changes. There is additional phase delay in the estimated acceleration of the lead vehicle. An estimation algorithm is needed to translate the discrete range measurements to a metric of change in range over time, i.e., acceleration and deceleration of the lead vehicle. In order to make ACC string stable, a relatively long headway is required, especially at higher speeds. In practice, these time-headways are larger than the time-headways acceptable to drivers.

In the case of CACC, the objective is similar to that of ACC: maintain a desired distance to the preceding vehicle. The most distinctive difference between ACC and CACC is that besides the preceding vehicle’s relative speed and position used as inputs in ACC, the acceleration of the preceding vehicle as transmitted through the wireless channel is also adopted as input in CACC. This omits the phase delay of the estimation algorithm used in ACC. A benefit of this approach is that smaller time-headways can be maintained, while still remaining string stable. This has the potential to double the effective road capacity [6].

The controller’s main task is to reject disturbances caused by velocity variations of the preceding vehicles. An ideal feedback control system should be able to cancel out all errors, effectively mitigating the effects of any forces that might arise during operation, producing a response in the system that perfectly matches the designer’s wishes. In reality, this might be difficult to achieve when taking measurement errors in the sensors, delays in the controller, and imperfections in the control input into consideration. The tuning of the CACC control system has impact on the system’s string stability, but the nature of the information communicated over the wireless channel also is an important factor. This dissertation is concerned with the nature of imperfections of the wireless communication, and the impact on the CACC system.

A proof-of-concept implementation of a CACC controller structure has been developed within the Connect&Drive project [42, 43], to be discussed in Sec. 2.2.4, and shows promising results. More details about this proof-of-concept implementation can be found in Chapter 4 and [41].

2.2.3 Impact on traffic flow

A stable traffic flow exhibits an increasing traffic throughput or *flow* as a result of increased traffic density [44, 45], a behaviour also well-known in networking. The throughput increases with an increase in supplied load. When the supplied load is

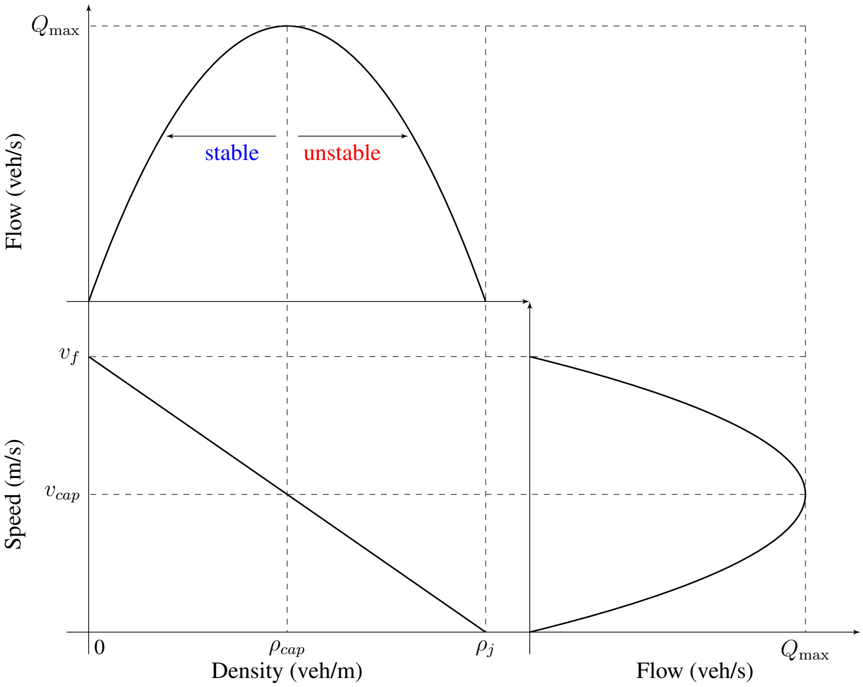


Figure 2.3: Traffic flow stability illustrated by flow-density, speed-density, and speed-flow relations, based on [46].

increased beyond a certain point, the efficiency decreases and the overall throughput actually diminishes. In this respect, the flow denoted by Q is the number of vehicles which pass a certain location on the road per time unit, for instance per hour. The density ρ in vehicles per kilometer per lane is inversely proportional to the vehicle spacing.

The relation between speed, flow and density is illustrated in Fig. 2.3. The most intuitive starting point is the relation between speed and density [46]. A single vehicle on a stretch of highway is not influenced by other vehicles and will travel at the *free-flow* speed v_f, expressed in meters per second. As more vehicles are driving on the road, the traffic density ρ will increase. The average speed of the vehicles will decrease because drivers slow down to accomodate the maneuvers of other vehicles (simplified to a linear relation in Fig. 2.3, in reality the low- and high-density regions are non-linear [46]). This relation is given in (2.1):

$$v = v_f \left(1 - \frac{\rho}{\rho_j}\right). \tag{2.1}$$

Ultimately, the density is increased to such a level that traffic will come to a stop. In this case, ρ is determined by the length of the vehicles and the space in between, yielding traffic jam density ρ_j. The relation between flow and density then becomes parabolic, depicted in the top left of Fig. 2.3, given by (2.2):

$$Q = \rho v. \tag{2.2}$$

Using this equation, we can obtain the maximum flow, denoted by Q_{max}, that a highway is capable of handling. This is also referred to as the *capacity*, with a corresponding density ρ_{cap}, the density which maximises flow. The capacity also has a corresponding speed v_{cap}, the speed which maximises flow as visible in the right part of Fig. 2.3.

When the string stability of a traffic flow is increased, the capacity density is increased: the traffic flow breaks down at a higher density. This means that vehicles are spaced closer to each other while they are still moving, yielding higher ρ_{cap} and resulting in an increased flow Q_{max}. This improvement is possible when the system is able to operate under tighter constraints, i.e., remain functional with smaller time headways in order to realise a higher ρ_{cap}. This can be achieved by reducing the delay in the control response.

The concept of automated car-following has been researched under many definitions, many of which overlap partially or completely. Frequently used terms are Automated Highway Systems (AHS) [32], cooperative driving [47], Cooperative Adaptive Cruise Control (CACC) [40, 41] or Integrated full-Range Speed Assistant (IRSA) [48]. Impact of CACC on traffic flow has been widely studied [6, 34, 49]. In a simulation study on the impact of CACC on traffic flow in [6] it was found that CACC has little effect on throughput at low penetration rates, with less than 40% of the vehicles equipped with CACC, in sparse traffic, but a reduction in shockwaves was observed. Already at low penetration rates, CACC can provide improvements to the traffic flow, provided that the "internal dynamics" of a block of non-communicating cars in between communicating cars are stable and reasonably smooth [23]. A higher penetration rate yields more improvements in terms of traffic flow stability and throughput, but shows dependency on the traffic density. In particular in high-density traffic, CACC shows a beneficial effect. In [6] the concept of a special lane for CACC-equipped vehicles is considered. it was concluded that this yields little beneficial effects and could even deteriorate the throughput of the combined lanes due to disturbances introduced by lane-changes.

2.2.4 Projects implementing CACC

During the course of this research, two proof-of-concept implementations of a CACC system were implemented: Connect&Drive (C&D) and the Grand Cooperative Driving Challenge (GCDC). In C&D the CACC was implemented in seven similar Toyota Prius cars; in GCDC ten vastly different vehicles were developed by independent teams which collaborated in an open-source fashion. These two different, yet similar projects show that it actually is practically feasible to perform real-time vehicle control based on beaconing with technology presently available.

Connect&Drive

Connect & Drive

C&D [42] was a research project which ran for $2\frac{1}{2}$ years, funded by the Dutch ministry of Economic Affairs under the High Tech Automotive Systems (HTAS) initiative. The project started in 2009 and was a cooperation between technical universities, industry and research institutes in The Netherlands. The aim of the project was the development and demonstration of a CACC system using identical vehicles [42].

The main motivation behind the project was finding a solution to traffic congestion. This is a large societal problem in the Netherlands, resulting in enormous financial losses due to wasted time and fuel, and unnecessary emissions. As described in Sec. 2.2, the goal of the CACC system is to reduce congestion by controlling vehicle dynamics [50]. Not only does CACC reduce congestion, it also reduces fuel consumption, emissions, and accidents because of a more stable traffic flow.

Focus of the project was on the Human Machine Interface (HMI), vehicle control, and wireless communication between vehicles. The C&D vehicles were equipped with layered functionality of traditional Cruise Control (CC), ACC by means of a forward-looking radar, and CACC systems. This approach enables the vehicles to drive in one of these modes in addition to manual driving. Under CC, ACC, or CACC, the system controls the vehicle's acceleration and deceleration, the driver is responsible for lateral control.

The layered approach enables the system to degrade gracefully and to operate under all circumstances to be expected in present-day and future traffic. If the system does not receive relevant information through its communication interface, it falls back to ACC behaviour. If no object is detected by the radar, the system falls back to CC behaviour. In conjuction with this fully integrated system, an advisory system—called the Cooperative Cruise Control (CCC)—was developed. Whereas the aim of (C)ACC is to automate longitudinal vehicle control, CCC's aim is to keep the human in the loop. As such, CCC is intended as a retrofit option. This part of the project will not be further discussed in this text.

Another topic covered by the C&D project which will not receive further attention in this dissertation is assisted merging. The reasoning why this would be needed is that CACC enables vehicles to drive closer together, eliminating gaps a merging vehicle could merge into. To solve this, a gap would need to be opened up when a merging vehicle is about to enter the highway. This topic is further covered in [51, 52, 53] and was also demonstrated in the prototype.

The transitional nature of the automotive market is a strong motivator to develop a system which can include both communicating and non-communicating vehicles, as well as vehicles with retrofit boxes. The project therefore does not only provide an "end product" of fully automated CACC, but also a transition path. The C&D system will be discussed in more detail in Sec. 4.2.

Grand Cooperative Driving Challenge

The Grand Cooperative Driving Challenge (GCDC) [8] was an international challenge set forth by the Dutch applied research and technology organisation TNO in March 2010. The challenge was for universities and research institutes to develop a vehicle which could cooperatively drive several defined traffic scenarios. CACC was a central component in the challenge. University of Twente, Fontys Automotive (Helmond, The Netherlands) and University of Waterloo (Canada) participated in the GCDC as Team FUTURUM[1].

Given a common interaction protocol and by that time commercially available IEEE 802.11p hardware, participants were free in the choise and design of their vehicle and control platform. Similar to C&D, team FUTURUM also implemented ad hoc platooning for participation in the GCDC, although some platoon management messages were part of the interaction protocol [54].

From a technology point of view, GCDC took one step further than C&D. In C&D, the seven similar vehicles all had the same communication and control platforms and parameters. In GCDC, only the air-interface was specified. Participants were free in the choice of vehicle, its instrumentation, and the type and configuration of controller they used, as long as the vehicle was able to respond to certain situations within certain limits. An example of this is the minimum acceleration capability of the vehicle. Using very different vehicles is one step closer to large-scale introduction of such systems: it can be expected that vendors will come up with different implementations. The fact that vehicles during the GCDC varied widely in size, weight and engine power proves that adoption of CACC in heterogeneous traffic is feasible. The FUTURUM system is described in more detail in Sec. 4.3.

2.3 Vehicular networking

An important aspect of being intelligent is the ability to acquire, interpret, and process information. If a system cannot obtain the information it needs to make its decisions, it may not make the right decisions—no matter how smart its design. Vehicular networking bridges this gap, allowing vehicles to get access to the information their ITS applications require to operate.

[1] Fontys University of Twente Universal Road Utility Maximiser

2.3.1 System architectures

Around the globe, many standardisation bodies and projects are or have been working to meet the challenges in vehicular networking and work is still ongoing. These efforts can largely be geographically grouped, because regulations and focus areas differ. In the USA, research and development mainly contributes to the standardisation of the IEEE 1609 protocol suite. In Europe, these activities result in contributions in European Telecommunications Standards Institute (ETSI) ITS [4, 55] and International Organisation for Standardisation (ISO) Communications Access for Land Mobiles (CALM) standardisation. Research and development in Japan contributes to the Association of Radio Industries and Businesses (ARIB) and ISO CALM standardisation. A complete overview is out of the scope of this work, we refer to [56, 57, 58].

The Car 2 Car Communication Consortium (C2C-CC) [59] is a consortium of European vehicle manufacturers, equipment suppliers, research organisations, and other partners. The discussion in this work is based on the C2C-CC reference architecture [60]. The C2C-CC works towards a compatible system usable throughout Europe, and great effort is put into harmonisation of standards on a global scale. This compatibility is important, because vehicles tend to cross regulatory borders. Another reason for standardisation is that huge investments are involved; it is not cost-effective to operate two competing systems side-by-side. In that case, vehicles would have to be equipped with both systems in order to reap the benefits of vehicular networking.

The C2C-CC specifies four key technical prerequisites and constraints for vehicular networks [60]:

Anonymity and data security. Local legislature with respect to privacy differs across many countries, even within the European Union. Efforts are being made to develop a system which can meet the requirements under all domains. In particular, technical issues regarding authentication, integrity, and privacy need to be solved. This is important in terms of system stability and availability; the system should continue to work under targeted or accidental adverse conditions. Anonymity and security is also important in terms of public acceptance. Public perception of a technology can make or break, or at least delay introduction of superior technology.

Effective protected frequency band. Vehicular networking needs to be robust. Safety-critical messages need to be exchanged within certain latency and reliability constraints. With a dedicated frequency band, the entire channel capacity will be available to the intended users. This limits the number of systems which will claim resources from the frequency band, and also limits their use of these resources, for instance by means of power regulations.

Mandatory sensor data. The C2C-CC specifies a certain minimal set of sensor data. How the sensor data is collected from the vehicle is out of the scope of the C2C-CC

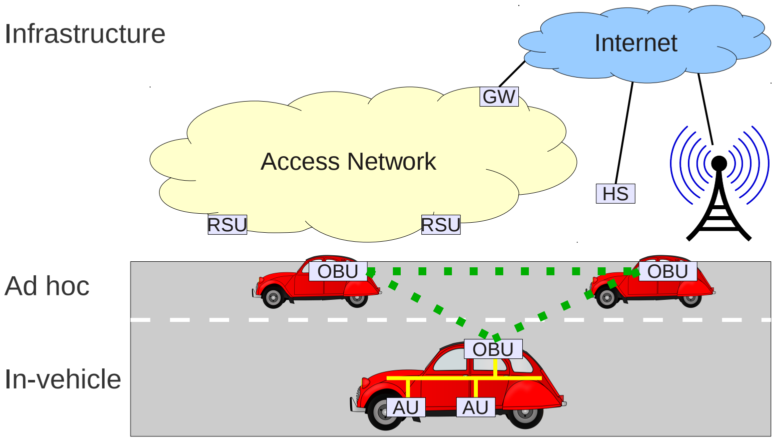


Figure 2.4: The C2C-CC Reference Architecture, based on [60].

and is left to the implementer. By exchanging this minimal set of data, every vehicle shares vital information which can be used by applications running in other vehicles or equipment alongside the road.

Scalability. Scalability is an important issue within vehicular networking. The system will have to work in a wide range of network circumstances, ranging from a few vehicles to a large number of neighbouring vehicles sharing the same channel. Another important aspect is that the system is also scalable in the temporal sense: once introduced onto the road, it should remain compatible with newer versions because vehicles often have a longer useful lifetime than, for instance, laptop computers or smart phones.

Out of these four technical prerequisites, the issue of scalability is addressed in this dissertation.

2.3.2 Communication domains

The C2C-CC reference architecture [60] identifies the three domains depicted in Fig. 2.4: in-vehicle, ad hoc, and infrastructure domain. These three domains will be discussed in this section.

The **in-vehicle networks** provide devices access to the services onboard the vehicle. Examples could be providing access to the Internet via bluetooth, managing the onboard entertainment system using a tablet PC, or streaming music from a smartphone to an in-vehicle entertainment system. An other form of in-vehicle network is the Controller Area Network (CAN) or Local Interconnect Network (LIN) bus which is used by the components of the vehicle to communicate. It is expected that some information from the sensors used in vehicle operation—such as wheel encoders and gyroscopes—will be available to other applications in the in-vehicle domain, although access should be properly proxied for safety reasons. Applications in these in-vehicle networks run on an Application Unit (AU) and can communicate with the outside world through a gateway in the On-Board Unit (OBU).

The **infrastructure networks** comprise all sorts of radio access technologies such as Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE), WiFi hotspots (depicted in the figure by the HS block), General Packet Radio Services (GPRS), WiMax, etc. Entities referred to as Road-Side Unit (RSU), provide access to resources located on the roadside. RSUs can be connected to an Access Network, which in turn can be connected to the Internet through gateways (depicted by the GW block). An RSU can be limited to the local scope, such as a traffic light which notifies nearby cars of its current status and timing information of future cycles. In a broader scope, an RSU provides information from a back-end network, e.g., a weather report which is periodically broadcast by an RSU. It may even provide full access to the Internet, as also provided by WiFi hotspots and 3G or LTE. Communication between vehicles and infrastructure networks is often referred to as Vehicle-to-Infrastructure (V2I) communications.

A significant number of ITS applications can be realised using the communication in the **ad hoc** domain. Communication between vehicles is known as Vehicle-to-Vehicle (V2V) communications, also frequently referred to as Inter-Vehicle Communication (IVC). It should be noted that some RSUs could also participate in the ad hoc network. In contrast to infrastructure networks, some nodes in the ad hoc networks are highly mobile and as a result, networking circumstances can change dramatically over a short timespan. It is possible to design applications which rely on the presence of a dense RSU network (such as the example of the traffic light broadcasting its status as given before) but the same architecture allows for completely nomadic applications solely existing in the ad hoc domain. Whereas widespread deployment of RSU networks will probably take a long time because of huge investments, obtaining a usable penetration degree of V2V-capable vehicles seems more realistic within the near future.

Within the ad hoc domain, the C2C-CC defines three data delivery schemes or communication patterns for V2V and V2I:

Geographical broadcast - to distribute data within a specified geographical area in a packet-centric way.

Event-driven, single-hop broadcast - to distribute data to the OBUs and RSUs within the direct wireless communication range.

Beaconing - to periodically distribute data to the OBUs and RSUs in the direct wireless communication range.

As depicted in Fig. 2.4, the OBU is responsible for Vehicle-to-Vehicle communications, and in Vehicle-to-Infrastructure communications it is the component in the vehicle which communicates with infrastructure components. The OBU can provide

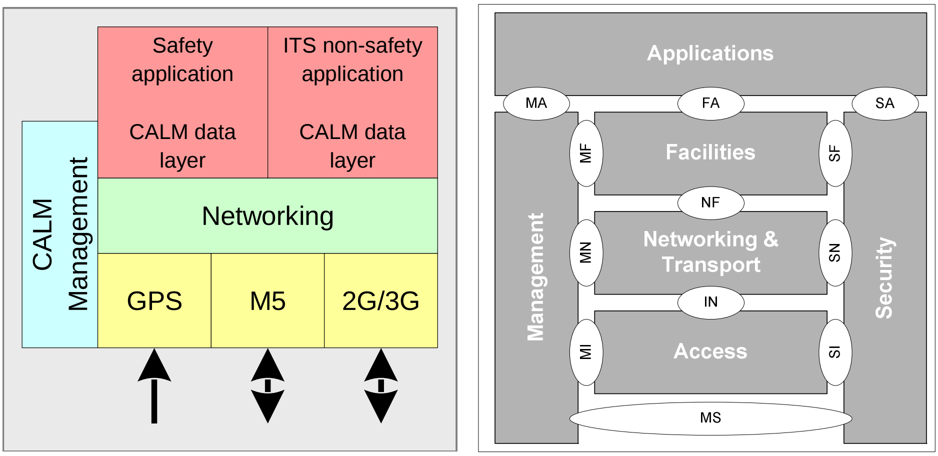
AUs access to the outside world through a variety of radio technologies, or present information received from the outside world to AUs.

2.3.3 Network architectures

The ISO CALM (TC204 WG16) [5] architecture combines multiple physical-layer technologies for V2V and V2I communications, taking input from various projects such as CVIS [61], COMeSafety[62], and Safespot [63]. At the core of the CALM architecture, illustrated in Fig. 2.5 (a), is a network layer which performs routing and media switching over various Medium Access technologies, also with the possibility to provide handovers. CALM also specifies a Non-IP stack known as CALM FAST, dedicated to communication in the ad hoc domain. CALM FAST uses CALM M5 [64], an adaptation of IEEE 802.11p to European regulations. A similar architecture is pursued in ETSI ITS [55, 4, 65], see Fig. 2.5 (b).

IEEE 1609 [3] is a standard which defines a network architecture on top of IEEE 802.11p. The combination of IEEE 1609 and IEEE 802.11p is referred to as Wireless Access in Vehicular Environments (WAVE). Sometimes this combination is also referred to as Dedicated Short Range Communication (DSRC), although the same term is used for Radio-Frequency Identification (RFID) communication in Electronic Toll Collection (ETC) applications.

CALM FAST and IEEE 1609 are intended to be harmonised as much as possible [66, 67]. In short, IEEE 1609 consists of four components:



(a) CALM station reference architecture, adapted from [5].

(b) ETSI ITS station reference architecture, copied from [55].

Figure 2.5: ETSI and CALM station architectures.

1609.1 - Resource Manager defines an application which enables multiplexing of messages between OBUs and remote applications which are potentially running in the infrastructure domain and can be accessed through an RSU.

1609.2 - Security Services defines message formats and processing methods, intended to protect WAVE communications against spoofing, eavesdropping, alteration and replay attacks. It also provides privacy to the users.

1609.3 - Networking Service specifies a TCP/IP stack based on IPv6, in parallel to a WAVE Short Message (WSM) stack. The TCP/IP stack is intended for use by non-safety applications on the Service Channel (SCH), whereas the WSM are to be used by safety applications. Only WSM traffic is allowed on the Control Channel (CCH). The WAVE Short Message Protocol (WSMP) is a lightweight protocol specific to the broadcast environment on the CCH and allows control of several physical parameters such as the transmission power, channel, and data rate. WSM messages can be either WAVE Service Advertisement (WSA) or WAVE Basic Safety Messages (BSM), which this work refers to as beacons or cooperative awareness messages. A service provider advertises its service and on which channel it is available by means of WSA messages.

1609.4 - Multi-Channel operations defines the use of a CCH and several SCH, parameters for priority, channel switching, and routing. The CCH is intended for use by WSM. IP datagrams are not allowed on the Control Channel. The channel-switching paradigm is described in more detail in Sec. 2.3.4.

IEEE 1609 and ISO CALM have many commonalities, especially around IEEE 802.11p and communication patterns in the V2V and V2I domains. CALM FAST could be seen as equivalent to the WSM stack. CALM does, however, have a broader scope with utilisation of various physical layer technologies, and cross-carrier handovers.

2.3.4 Channel switching

In general, the main objective of channel switching is to provide various services in parallel, using a smaller number of radios than the number of channels. This reduces complexity, hence cost of the required hardware and increases the *overall* capacity of the system.

In vehicular networking, IEEE 1609.4 enables single-radio devices to participate in communication on multiple channels in a time-division manner by means of synchronised interval timers. This is achieved by synchronised service intervals of 100 ms. A station resides on the CCH for 50 ms, then can spend 50 ms on a SCH to return again to the CCH, illustrated in Fig. 2.6. In order to do this, a global synchronisation by means of Coordinated Universal Time (UTC) derived from the

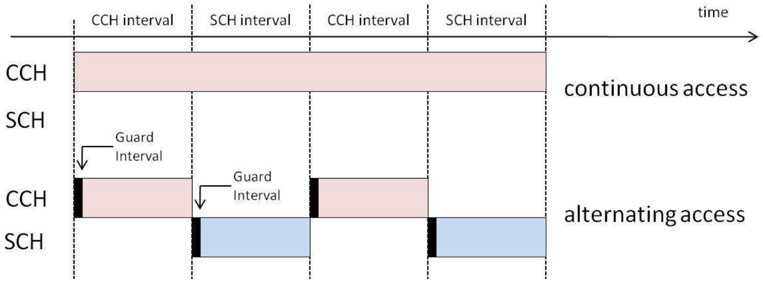


Figure 2.6: Continuous channel access in IEEE 802.11p, alternating channel access in IEEE 1609.4, and the immediate and extended access schemes; copied from [68].

Global Position System (GPS) is used. The CCH interval starts at a multiple of 100 ms after the UTC second boundary.

IEEE 1609.4 defines a Guard interval at the begininng of both the CCH and SCH interval, see Fig. 2.6. The Guard interval allows the node to switch its radio and accounts for small time differences between the nodes. The value of the guard interval is defined as `SyncTolerance` + `MaxChSwitchTime` [69]. `SyncTolerance` is the expected precision of a node's internal clock compared to the UTC. The time a node needs to tune its radio to another channel is defined by `MaxChSwitchTime`. Typical values for the guard interval are between 4 and 6 ms, resulting in an effective channel availability of 44 to 46 ms per 100 ms interval.

During the guard interval, nodes are not allowed to send or receive data so all on-going traffic is suspended. Also, a medium busy status is declared by the radio to make sure that nodes are not going to transmit simultaneously at the end of a guard interval. When the guard interval is over, all transmissions must first wait for the random backoff as prescribed by the IEEE 802.11 MAC, described in detail in Sec. 2.4. After performing backoff, the node starts new communication activities on the new channel or resumes any suspended transmissions for this channel.

The impact of IEEE 1609.4 channel switching is evaluated in Sec. 6.4.

2.3.5 Message sets: EIVP

It is important that not only the medium access and network part is standardised for interoperability: the messages which are exchanged should also be intelligible to all systems on or next to the road. This is completely in line with the open architecture on which the Internet is based, but is new in the automotive world where cooperation among products from different vendors is a completely new concept. Details on internal operation of components are often of a proprietary nature, and interfaces are not disclosed.

Several message sets have been defined, the most noteworthy being SAE J2735 - Dedicated Short Range Communications (DSRC) Message Set Dictionary [70] and European ITS VANET Protocol (EIVP) [71] developed in the EU COMeSafety and

SAFESPOT projects [62]. Because the EIVP was used within the C&D project, we will briefly explain its contents. The WAVE counterpart—based on the SAE message format—is not directly compatible but contains fields of a similar nature.

The EIVP message format [72] is modular in structure. It contains a Network Header, which in turn encapsulates a Transport Protocol Data Unit (TPDU) and a message body contained in the Protocol Data Unit (PDU). This structure is illustrated in Fig. 2.7, which represents the EIVP as used within the C&D project.

The Network Header contains various protocol control information fields, but more importantly it also contains a Netheader which contains various information elements relating to the operational status of a vehicle. Examples of fields include the `nodeHeading`, `nodeSpeed`, and `nodePosition`. The Cooperative Awareness Message (CAM) service PDU inside the TPDU contains various vehicle status fields such as the host vehicle's `yawRate` and `vehicleLength`.

The EIVP format allows adding information in a modular way, as is done with the C&D service PDU in Fig. 2.7, added in the C&D project. This element contains additional information, such as a vehicle's `radarRange` and the `radarRangeRate`, to signify range to the object in front and the rate of approach. The complete EIVP specification can be found in [72], the version used in C&D can be found in [73].

2.3.6 Vehicular networking: concluding remarks

IEEE 1609 is largely a U.S. standardisation effort, whereas CALM is primarily driven by European consortia contributing to ETSI standardisation. There are many similarities between both network architectures and harmonisation is attempted because the automotive market is a global one profiting greatly from economies of scale, a paradigm which favours harmonisation across legislative domains.

Message sets are under standardisation and are being used in prototype implementations. A message format such as the EIVP defines the exchange of a significant number of information elements concerning either vehicles or RSUs. By collecting the data received from surrounding nodes into a cooperative awareness, applications only need to select elements from this awareness and do not need to generate additional messages across the air interface. More details on cooperative awareness is presented in Sec. 3.2.

Regardless of the way applications are specified, the exact content of network packets or how they are formulated, at the MAC/PHY layer both IEEE 1609 and CALM allow the use of IEEE 802.11p technology (refered to as M5 [64] in the CALM architecture). Note that CALM also allows use of other MAC/PHY technologies in addition to M5. In the remainder of this work we abstract from upper-layer architectures, and focus on the performance of the beaconing communication pattern. Both approaches specify the use of beacon messages, generating a cooperative awareness in one way or another. Both specify beacons to have a certain content comprised of both static and dynamic data relating to nodes in the traffic context, either vehicle or RSU.

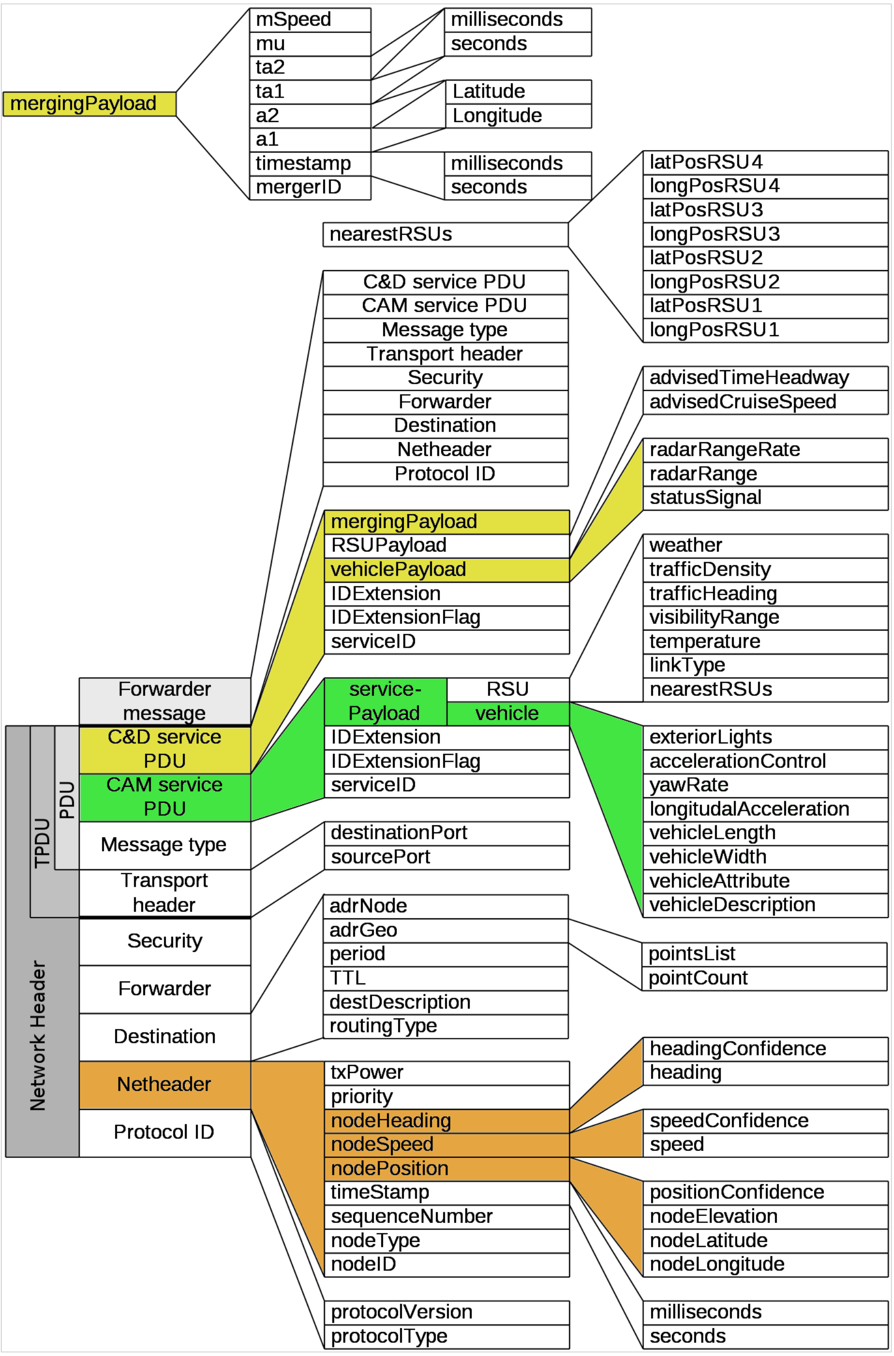


Figure 2.7: Schematic representation of the EIVP message format with C&D additions, adapted from [74].

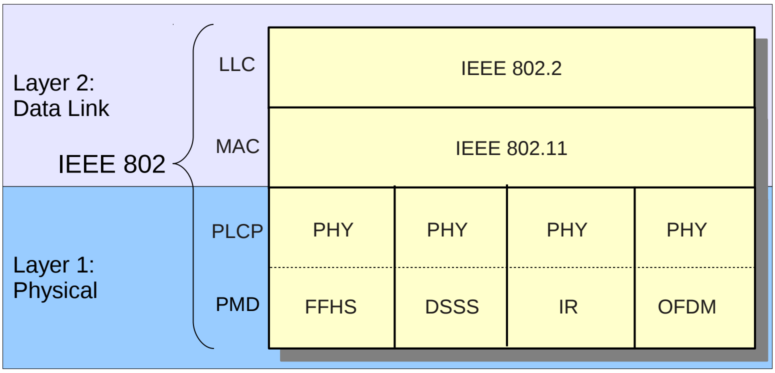


Figure 2.8: The IEEE 802.11 protocol stack allows use of various physical layers with the same MAC.

2.4 Medium access control layer: IEEE 802.11

Standardisation at the network layer and above is still ongoing, however, the MAC and PHY technology for V2V and V2I has converged to IEEE 802.11p, although several alternatives exist [75, 76]. The remainder of this work will focus on 802.11p and its particularities. In order to understand IEEE 802.11p, the following will first describe its basis, IEEE 802.11, in detail. IEEE 802.11 is a family of standards; prescribing a basic medium access mechanism which is then applied in various technologies. These technologies differ predominantly in their physical layer implementation, but also have subtle differences in their medium access mechanisms and parameters.

2.4.1 IEEE 802.11

By the end of the 1990's, wirelessly connecting computers to an Access Point (AP) which was in turn connected to a Local Area Network (LAN) became very popular with the advent of IEEE 802.11b, both in the home and the office environment. In this traditional setup, an AP is connected to a distribution system, typically Ethernet. The APs are then placed such that good coverage is achieved.

IEEE 802.11 essentially is an adaptation of the Ethernet MAC protocol (IEEE 802.3 [77]). It inherits many of the advantages of Ethernet, in particular its simplicity and fully distributed operation. The relative robustness and low cost of IEEE 802.11 technology are its major strengths. Especially the latter was an important factor in popular adoption in consumer electronics. The more complex HiperLAN [78] could not compete with the success of IEEE 802.11b, although a large portion of the physical layer of HiperLAN-2 [79] was later standardised in IEEE 802.11a.

IEEE 802.11 defines loosely coupled Medium Access Control and Physical layer specifications, as illustrated in Fig. 2.8. This modularity enables multiple physical layer technologies to be used with the same interface to upper layers. The Logical

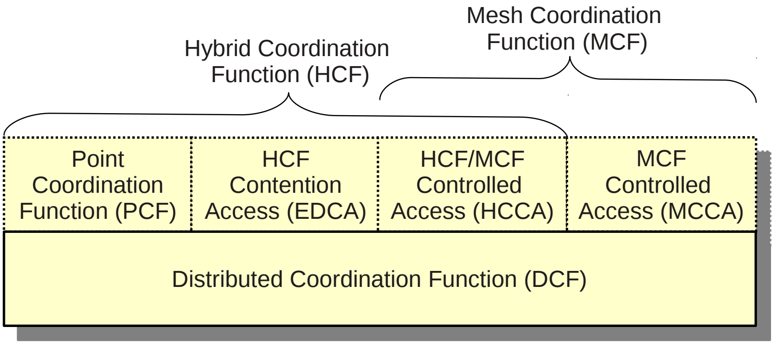

Mesh Coordination Function (MCF)
Hybrid Coordination Function (HCF)
Point Coordination Function (PCF)
HCF Contention Access (EDCA)
HCF/MCF Controlled Access (HCCA)
MCF Controlled Access (MCCA)
Distributed Coordination Function (DCF)

Figure 2.9: Relation of the coordination functions in the 802.11 MAC, based on [2].

Link Control (LLC) provides an interface similar for all 802 family members. The MAC defines how a node has to access the wireless channel and coordinate this access with other nodes. The Physical Layer Convergence Procedure (PLCP) sublayer is responsible for translating the MAC sublayer protocol data units (MPDU) to frames suitable for sending and receiving user data and management information using the associated Physical Medium Dependence (PMD) system [80]. This simplifies the interface between MAC and PMD, and provides a mapping of PMD to MAC primitives and vice-versa. The PMD provides a method to transmit and receive data using the associated wireless medium.

Over the years, various PHY technologies have been defined: Infra Red (IR) in legacy 802.11, and various Radio Frequency (RF) modulation technologies such as Frequency Hopping Spread Spectrum (FHSS) (legacy 802.11), Direct Sequence Spread Spectrum (DSSS) as used in 802.11b, and Orthogonal Frequency Division Multiplexing (OFDM) as used in 802.11a, g, and p. Later on, Multiple Input, Multiple Output (MIMO) capabilities to improve performance in multi-path environents were added in the "n" version.

Medium access is controlled by the Distributed Coordination Function (DCF), which is the fundamental access method of the IEEE 802.11 MAC, as illustrated in Fig. 2.9. On top of the DCF, the Point Coordination Function (PCF), Enhanced Distributed Channel Access (EDCA), HCF Controlled Channel Access (HCCA) and MCF Controlled Channel Access (MCCA) can reside. These functions rely on the DCF and provide additional services or override certain behaviour. The PCF and HCCA can be located in an Access Point, where the AP coordinates channel access among the stations associated with it to provide them with contention-free access. The Mesh Coordination Function (MCF) specifies additional coordination functions for use in a mesh environment. The PCF, HCCA, and MCCA will not be elaborated upon in this text, for more information see [2]. Because the DCF is the basis of 802.11, this will be described in further detail in the following subsections. After that, the EDCA as applied in IEEE 802.11p will be described in Sec. 2.5.

2.4.2 The distributed coordination function

The DCF is mandatory in all 802.11 stations, and specifies basic rules for medium access and contention resolution [80]. In addition, the DCF mandates a gap between contiguous signals on the medium, known as the Interframe Space (IFS).

When a station A wants to transmit a frame to another station B, this is transmitted by means of electro-magnetic signals traveling through the air. The receiving station's radio is tuned to the same frequency and will receive the frame. However, if a third station C in the vicinity is also transmitting during A's transmission, its signals may interfere and B will not be able to properly receive the signal as transmitted by A. This is called a *collision*.

The function of a medium access mechanism is to coordinate transmission attempts by multiple stations contending for access to the same wireless channel, and prevent collisions. To reduce the probability of collision, the DCF specifies the use of CSMA/CA. The Carrier Sense (CS) part prevents a node from transmitting when an other node is already transmitting: it is a simple "listen-before-talk" operation. The medium is found to be idle when the detected signal level is below the Carrier Sense threshold. In this case, a node may proceed to access the channel. If the channel is determined to be busy, the access attempt is deferred until the medium turns idle again. The PHY will continue to sense the channel. It may even be in the process of receiving a frame.

The Collision Avoidance (CA) part prevents collisions where they are most likely to occur: just after a transmission by an other node. As described above, CS mandates to defer access until the channel turns idle again. If multiple contending nodes are waiting for the channel to become idle, they are somehow synchronised and could cause a collision if they were to transmit immediately. To alleviate this, CA mandates use of a so-called backoff. When the medium turns idle, a node does not immediately begin transmission, but will wait a mandatory IFS and some random extra time by means of a backoff counter (bc). The bc is randomly drawn from the Contention Window (CW), a range $[0, \texttt{CWmin}]$, where the size of the contention window is always a power of two:

$$\texttt{CWmin} = 2^{\texttt{exponent}} - 1. \tag{2.3}$$

The size of the CW depends on the PHY, a typical exponent is 4 (OFDM PHY) or 5 (DSSS PHY). With the medium idle, nodes count down their bc for every timeslot that passes. The node with the first bc expiration will commence transmission; others will defer their access attempt and freeze their bc values. As soon as the channel is perceived idle and an IFS has passed, the nodes continue counting down.

CSMA/CA functions by reducing synchronisation; transmissions from contending stations are spread in time. Because of this temporal spreading and the initial CS operation, this mechanism is very effective under light load. It allows stations to transmit with minimal delay and with good reliability—especially when using the RTS/CTS mechanism described later in Sec. 2.4.3. This does not make the system collision-free; there is a positive probability that two nodes have their bc expire at the

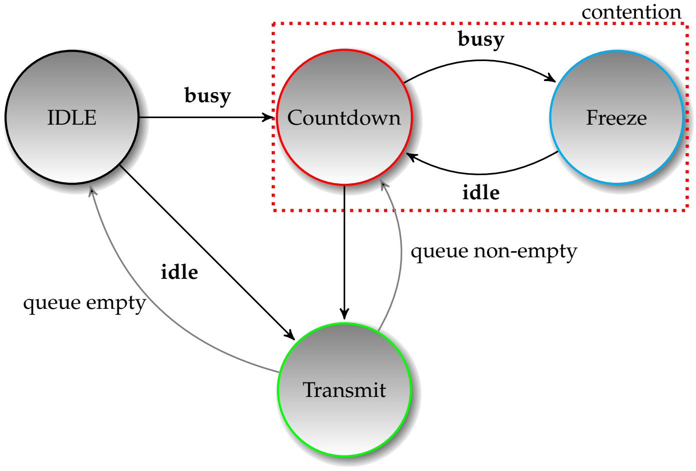


Figure 2.10: The CSMA/CA backoff process: states of the MAC.

same time, or that two stations simultaneously sense the medium to be idle. In both cases they will transmit at the same time, resulting in a collision. It does, however, make collisions less likely to occur.

The CSMA/CA mechanism in IEEE 802.11 is illustrated in Fig. 2.10, showing the states that the MAC will go throught. The MAC will perform the following operations once the network layer submits a packet to the MAC's transmission queue, assuming the station starts in the IDLE state:

1. Upon reception of a packet in the transmission queue, the MAC performs CS.
   (a) If the channel is found to be **idle** for a duration of at least one IFS, the transmission may commence immediately.
   (b) If the channel is determinded to be **busy**, the node will enter the contention state, which is divided into a countdown and freeze state.
2. The node uses a *bc* drawn from the CW. The *bc* is choosen from the uniform distribution [0,`CWmin`].
3. When the channel turns **idle**, the node waits one IFS.
4. Every timeslot σ in which the channel remains idle, the *bc* is decremented.
5. If the channel turns **busy** during Countdown, *bc* is frozen and the process continues from step 3.
6. When *bc* reaches 0, the node transmits the frame. Note that this may also happen if a node chooses 0 as *bc* in step 2.

The MAC is considered BUSY if it is servicing a packet, i.e., it is performing contention or transmission. If a packet arrives while the MAC is BUSY, it is enqueued and will be serviced after transmission of the current frame is complete.

Post-backoff

The post-backoff prevents unfair advantage by nodes which have just finished their transmission and still have packets to send. Following their own transmission, it is likely the other nodes in the system are caught up in counting down their bc values. This would imply that the node under consideration finds the medium idle, and can immediately transmit without performing backoff. In this scenario, a node with a very high traffic load could dominate the channel, resulting in starvation of the other nodes. The mechanism of post-backoff serves to prevent this starvation. Post-backoff functions similar to the backoff prior to performing a transmission when the medium was found to be busy during carrier sensing:

1. After completion of a transmission, the node draws a bc from the CW.
2. After an IFS during which the channel remains idle, for every timeslot σ in which the channel remains idle, the bc is decremented.
3. If the channel turns busy (by a transmission from another node) the bc is frozen and the process may only continue after the channel has turned idle, and remained idle for at least an IFS.
4. When bc reaches 0, the node transmits the frame if there was a frame to transmit. If the transmission queue is empty, the node remains IDLE.

2.4.3 Broadcast and unicast

The broadcast paradigm comes naturally to wireless communication. The radio waves are received by any receiver which is tuned to the correct frequency, and is able to demodulate the signal. This is the case when a node is within a distance such that the received signal is sufficiently stronger than the noise and interference. Fundamentally, at the physical layer every transmission is a broadcast transmission. At the MAC layer it is decided whether it was a Broadcast, meant for anyone who receives it, or a Unicast message, intended only for a specific receiver.

Unicast

In a Unicast transmission, the receiver's MAC address is included in the MAC header. Upon successful reception, determined using a Cyclic Redundancy Check (CRC), the receiver indicates success by means of a positive acknowledgement after waiting a Short Interframe Space (SIFS). This DATA-ACK sequence is called Basic Access and is the most fundamental form of Unicast. The Basic Access mechanism is illustrated in Fig. 2.11, transmissions above the horizontal lines are performed by the sender and below the line by the receiver. If the sender does not receive an acknowledgement within a timeout period, it will double the CW and choose a new *bc* for the retransmission. This process is known as Binary Exponential Backoff (BEB). With every retry, the CW will grow, up to a maximum of `CWmax` [80].

A node will retry up to a retry limit, after which the frame is dropped. In 802.11 this is indicated by the `longRetryLimit` and `shortRetryLimit`. Long frames claim the channel for a longer duration, and if multiple successive losses occur it is highly likely the channel is congested. This mechanism reduces the number of retransmissions of the expensive long frames. Whether the long or the short retry limit is used depends on the length of the frame.

A Unicast transmission is not only received at the intended receiver, but also at other nodes in the area due to the underlying broadcast nature of the wireless channel. Upon a successful CRC check, the Duration field in the frame's header (see Sec. 2.4.4) is used to set the Network Allocation Vector (NAV) in a station. This is the Virtual Carrier Sensing (VCS) function, and together with physical carrier sensing it is used to determine if the channel is busy. The result is that a node not involved in a DATA-ACK exchange will find the medium busy for the duration of the transmission and its acknowledgement, and will not interfere.

After setting the NAV, a station checks the destination address. If the destination address in the header does not equal a station's own MAC address, the frame will be discarded. The station which is the intended receiver of the frame will hand it up to its higher layer.

Unicast with RTS/CTS The Hidden Terminal problem is common in wireless networks. Consider Fig. 2.12. Here node A transmits a frame to B. Node C is not aware of A's transmission. This is because it is out of A's transmission range and cannot sense the ongoing transmission. The result is that node C finds the medium idle, and also transmits. This causes a collision at node B, even if only a small portion of the frames overlaps.

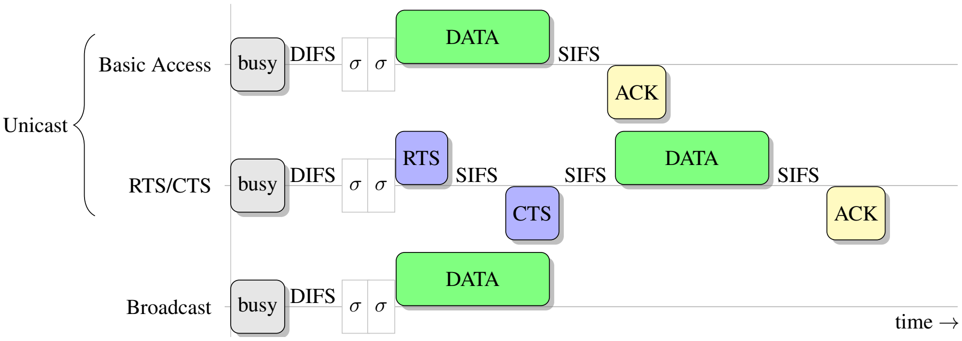


Figure 2.11: The IEEE 802.11 DCF access mechanisms: Unicast and Broadcast, here illustrated with a *bc* of 2.

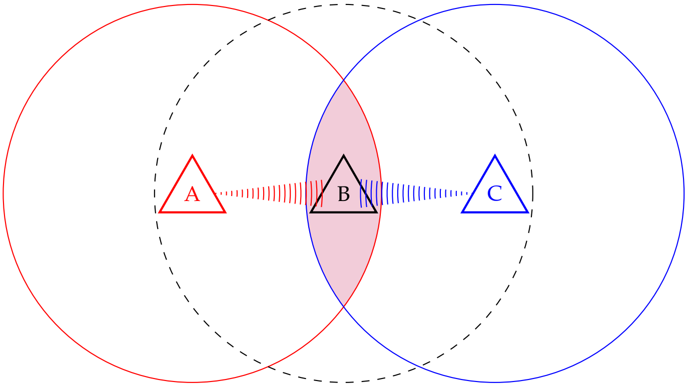


Figure 2.12: The Hidden Terminal Problem.

In order to counter the Hidden Terminal problem, Basic Access can be extended with the Request-To-Send Clear-To-Send (RTS/CTS) sequence. This improves the VCS for Basic Access and is illustrated as the second example in Fig. 2.11. If station A wants to transmit a frame, it first transmits a short Request-To-Send (RTS) frame. The RTS frame contains source and destination address, the duration of the frame the station wants to transmit, plus the time it takes the receiver to send an acknowledgement frame. The receiver (node B) will reply with a Clear-To-Send (CTS) frame which simply echoes the RTS contents. All stations which received either the RTS or the CTS frames (including node C in Fig. 2.12) are now aware of the planned transaction and its duration. These stations set their NAV for the remaining duration of the transaction. The effect is that nodes around the receiver will not interfere with the reception of the DATA frame, while nodes at the transmitter will not interfere with the reception of the ACK frame, because the channel has been "reserved" for the entire transaction.

This method greatly improves performance in scenarios where hidden terminals are common. Collisions, when they occur, are between the short RTS and CTS frames and only little time is lost in the collision because retransmission will take place quickly. However, the RTS/CTS sequence also introduces some overhead in the form of increased use of channel resources and latency, as illustrated in Fig. 2.11. The RTS/CTS sequence is more efficient for frames which are significantly larger than the RTS and CTS frames. Whether or not the RTS/CTS is used, depends usually on a size-dependent threshold defined by `RTSThreshold`.

Broadcast

By including a special Broadcast address into the frame's destination address, the transmitted data frame becomes a Broadcast frame. Any station which successfully receives it will pass it up to its network layer. This is the third example illustrated in Fig. 2.11, and consists only of a single transmission: that of the DATA frame.

IFS	Definition	Value
Slot time σ	(determined by `aAirPropagationTime`)	16μs
SIFS		32μs
DIFS	$2\sigma + \mathtt{SIFS}$	64 μs
EIFS	$\mathtt{SIFS} + \mathtt{DIFS} + \mathtt{T}_{\mathtt{phy}} + \mathtt{T}_{\mathtt{ack}}$	248 μs

Table 2.1: IEEE 802.11p DCF timing relations [2].

Given that the information conveyed in beacon messages in a vehicular network is of vital importance to *any* node in the vicinity, these beacons are transmitted as Broadcast messages. Broadcast transmissions do not use acknowledgements; if they would, many nodes would transmit and cause collisions with their acknowledgement frames. Also, Broadcast transmissions can not use the RTS/CTS sequence, for the same reason. Because the receiver does not explicitly acknowledge reception, the sender cannot verify if the frame was successfully received. Hence, it does not perform any retransmissions, or increase its CW size. This makes the Broadcast access method a very simple, yet fragile way of communication. The transmission has a higher probability of collision and the transmitting node has no way to infer correct reception at the receiver.

Interframe spacing In order to separate contiguous frames on the channel, stations must observe an Interframe Space (IFS) between frames. There are several types of IFS, each with a specific duration. By their difference in duration, one operation can obtain priority over an other because it can commence sooner, as explained in the discussion about the Short Interframe Space below.

The duration of the set of IFS-es also depends on the type of modulation used in the physical layer, because the IFS needs to be sufficiently large to perform proper physical CS. In line with IEEE 802.11's modularity, the physical layer determines various values, and other values in the MAC are derived from there. The values given in Table 2.1 are for IEEE 802.11p (discussed in detail in Sec. 2.5). These fields are also present in other flavours of 802.11, although they may have different values.

Slot time The slot time σ is choosen such that a station can detect if other stations have started a transmission. This depends on the modulation, the capabilities of the hardware, and the laws of physics. Formally, this amounts to [80]:

$$\begin{aligned}\mathtt{aSlotTime} = \;& \mathtt{aCCATime} + \mathtt{aRxTxTurnaroundTime} \\ & + \mathtt{aAirPropagationTime} \\ & + \mathtt{aMACProcessingDelay}.\end{aligned} \tag{2.4}$$

In this relation, aCCATime is the duration the PHY needs to determine if there is a signal on the channel which exceeds the CS threshold. The time the radio circuit needs to switch from receive to transmit is set by aRxTxTurnaroundTime. The signal takes some time to propagate through the channel, this is accounted for by aAirPropagationTime. The time the MAC needs to process is accounted for in aMACProcessingDelay.

Short interframe spacing The SIFS is the duration of time separating DATA-ACK pairs or dialogs, or the RTS-CTS-DATA-ACK frames as illustrated in Fig. 2.11, both only in Unicast transmissions. Effectively, SIFS gives priority to the receiving station to return the ACK (or perform the remainder of an RTS-CTS transaction), because all other stations (which may be competing for the medium as well) have to respect a longer IFS (at least a DIFS).

$$\begin{aligned}\mathtt{aSIFSTime} = \; & \mathtt{aRxRFDelay} + \mathtt{aRxPLCPDelay} \\ & + \mathtt{aMACProcessingDelay} + \mathtt{aRxTxTurnaroundTime}.\end{aligned} \tag{2.5}$$

Distributed (coordination function) interframe spacing Distributed IFS is the normal duration of time separating two contiguous transmissions on the medium. When a station performs CS, the channel must at least have been idle for DCF Interframe Space (DIFS) before transmission is allowed:

$$\mathtt{DIFS} = \; \mathtt{aSIFSTime} + 2 \cdot \mathtt{aSlotTime}. \tag{2.6}$$

Extended interframe spacing The Extended Interframe Space (EIFS) is the longest IFS. It is used in response to reception of a frame which failed its CRC check. The station was not able to extract the duration from the frame and cannot set its NAV for this duration. It is not even clear if the ongoing transmission was a broadcast or unicast transmission. To prevent this station from interfering with an ongoing transaction, it has to wait until after the intended recipient had an opportunity to return an ACK frame:

$$\mathtt{EIFS} = \; \mathtt{aSIFSTime} + \mathtt{ACKTxTime} + \mathtt{DIFS}, \tag{2.7}$$

where $\mathtt{ACKTxTime}$ consists of the duration of transmitting the physical layer header $\mathtt{T}_{\mathtt{phy}}$ and the Acknowledgement packet $\mathtt{T}_{\mathtt{ack}}$. Because of the nature of the wireless channel, it can occur that one node receives garbled bits while an other node is able to receive a frame perfectly. This implies that, in response to a collision, it is possible that not all nodes use an EIFS.

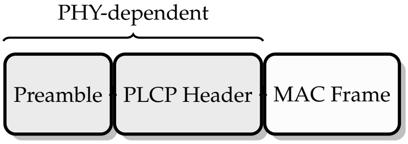


Figure 2.13: IEEE 802.11 PHY Frame Format, based on [80].

2.4.4 Frame formats

IEEE 802.11 contains a diverse set of frame formats. Only the *Data* frame and *Acknowledgement* frame are relevant in the scope of this work and will be described here. All other frames are detailed in the standard [2]. At the PHY, all 802.11 MAC frames are prepended by a Preamble (used for synchronisation and tuning of the reveiver) and a Physical Layer Convergence Protocol (PLCP) header containing necesarry information to receive the remaining part of the frame, shown in Fig. 2.13. The duration of the Preamble and PLCP Header are physical-layer dependent.

The PLCP Header is transmitted at the lowest data rate; this modulation is the most robust against interference. The PLCP Header contains, among other fields, information on the length and data rate of the encapsulated MAC Frame. This ensures that nodes in range become aware that a transmision is in progress, and the PHY will report the medium to be busy.

The MAC Frame may be transmitted at a higher data rate which is more susceptible to interference. If due to interference a remote node is not able to demodulate the MAC Frame, at least it may be aware that a transmission is going on if it can decode the header. More detail on the physical layer is covered in Sec. 2.6.

The contents and duration of the MAC Frame field depends on the type of MAC frame, the payload, and the data rate at which the frame is transmitted. The Data and Acknowledgement frame will be described below.

Data frame

The Data frame carries the actual data from the transmitting node to a receiving node and is used in both Broadcast and Unicast transmissions. This structure is depicted in Fig. 2.14. The frame is encapsulated into the MAC Frame of the PHY frame shown in Fig. 2.13. The MAC frame follows a similar header-data-trailer format. The header contains fields necessary to decode the body. The MAC Header comprises 34 bytes in 802.11 [80]. When using EDCA, another two bytes for QoS Control by APs are added.

The Frame Control field contains information necessary for operation of the protocol (such as version, frame type, etc.). The Duration/ID field, in the case of Data and Acknowdledgement frames, contains the duration of the remainder of the frame in μs. In case of Unicast transmissions, the Duration includes the IFS and the duration of an Ack, so receiving nodes can set their NAV for VCS as described in the section on Unicast in Sec. 2.4.3. The Sequence field is used to facilitate fragmentation

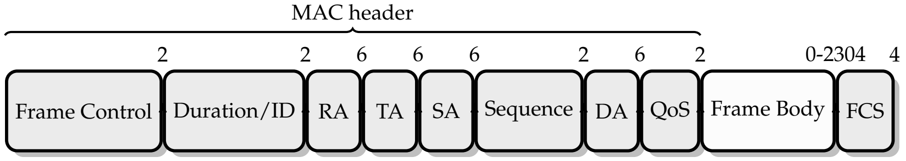


Figure 2.14: IEEE 802.11 MAC Data Frame Format, based on [80]. Field sizes are in bytes.

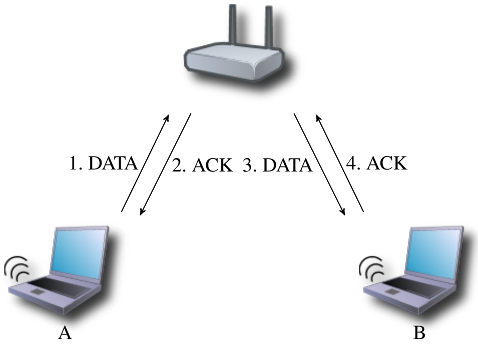


Figure 2.15: Node A sends data to node B through an AP.

Field	A → AP	AP → B
RA	AP	Node B
TA	Node A	AP
SA	Node A	Node A
DA	Node B	Node B

Table 2.2: Use of the IEEE 802.11 address fields for communication through an AP.

and reassembly. This is needed if the payload is too large to fit into a single MAC frame. Correct reception of the Frame Body can be checked by means of the Frame Check Sequence (FCS), which is a 32-bit CRC.

As illustrated in Fig. 2.14, the IEEE 802.11 MAC header has four address fields:

Receiver Address RA - The MAC address of the intended receiver of the frame. This field is used by a station to see if it is the intended recipient.

Transmitter Address TA - The MAC address of the station which transmitted the frame. In Unicast, the ACK should be returned to this address.

Source Address SA - The MAC address of the original source (in case of use of an Access Point)

Destination Address DA - The MAC address of the final destination (in case of use of an Access Point)

The SA and DA are important in case communication goes through an Access Point, this is illustrated in Fig. 2.15 for Basic Access. In case node A wants to communicate with node B, it will transmit a DATA frame with addresses as given in the second column of Table 2.2. The AP knows that it is the intended first-hop receiver of the frame (by checking the RA) so it will try to receive it. This interaction may involve several retransmissions and the AP will send an ACK to A. The AP now knows it needs to forward the Frame Body to Node B (present in the DA field). The AP will transmit a DATA frame with the address fields set as in the third column in Table 2.2, if needed with several retransmissions until the AP receives an ACK from B. Node B has now received the data and knows it comes from node A. Transmission of a response from higher-layer protocol entities in node B will follow a similar pattern in reverse order.

Acknowledgement frame

The Acknowledgement frame is used to acknowledge Unicast transmissions. The ACK is sent after a SIFS, before any other transmissions may commence. In a Broadcast scenario, this frame is not used but its duration is used to compute the duration of an EIFS.

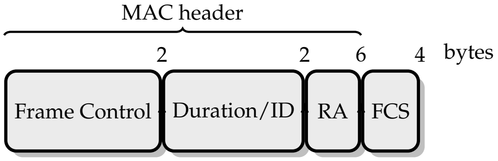


Figure 2.16: IEEE 802.11p Acknowledgement Frame Format, based on [80].

2.5 IEEE 802.11p

IEEE 802.11p is the MAC/PHY designed for use in vehicular networking to enable ITS [67] and is used for Wireless Access in Vehicular Environments (WAVE) [3] and as CALM-M5 / ISO 21215 [64], adapting it to European regulations. This particularly relates to spectrum use within various regulatory domains. IEEE 802.11p is an adaptation of IEEE 802.11a for the vehicular environment.

In traditional IEEE 802.11 networks, an AP is used to get access to a Distribution System (e.g. a LAN or the Internet), forming a Basic Service Set (BSS). Even if two stations are located next to each other, communication between them goes through the AP, as described in Sec. 2.4.4. This communication pattern is typical for Wireless Local Area Network (WLAN) situations, but absent in 802.11p. Because

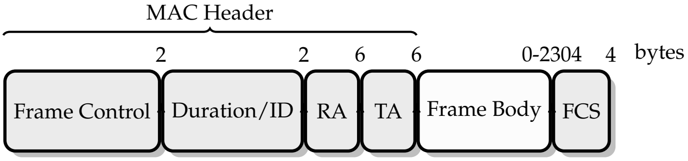


Figure 2.17: IEEE 802.11p MAC Data Frame Format, based on [80].

Variable	**802.11a**	**802.11b**	**802.11g**	**802.11p**
`aSlotTime` (μs)	13[2]	20	9, 20	16[3]
`aAirPropagationTime` (μs)	$\ll 1$	$\ll 1$	$\ll 1$	< 4
`aCCATime` (μs)	< 4	$\ll 15$	< 4	< 8
Symbol duration (μs)	4	n/a	4	8
Modulation	OFDM	DSSS	OFDM	OFDM
Min. data rate (Mbps)	6	1	1	3
Max. data rate (Mbps)	54	11	54	27
Frequency band (GHz)	5, 5.8	2.4	2.4	5.9
Bandwidth (MHz)	20	22	20	10
outdoor range (meters)	100	140	140	1000

Table 2.3: Differences between the various IEEE 802.11 flavours, based on [80, 82, 81].

communication does not go through an AP, 802.11p operates completely in ad hoc mode. The last two address fields of the standard 802.11 header (SA and DA) are absent in the 802.11p header because they are not needed. The MAC Header for 802.11p [81] then (including the FCS) has a size of 20 bytes, as depicted in Fig. 2.17.

Furthermore, the methods of active and passive scanning for access points, association and authentication with access points are not used. This is referred to as operating “outside the context of a BSS”. Instead of joining a BSS as is done in WLAN, stations use the “Wildcard BSSID”. Two stations can immediately exchange information without the need to perform a handshake, reducing access delay. An RSU may be connected to the Internet and provide this as a service, but it plays no role in the Medium Access Control in the way an AP does in traditional 802.11.

The DCF channel access mechanism used in IEEE 802.11 is replaced by the Enhanced Distributed Channel Access (EDCA) in order to allow Quality of Service (QoS) provisioning on a per-packet basis. Another major change is in the physical layer, described in Sec. 2.6. Due to 802.11’s loosely-coupled modular approach, changes in the physical layer have their impact on MAC layer parameters. The changes are summarised in Table 2.3. The change which has most impact on the MAC timing is the Slot time σ, which is 13μs in IEEE 802.11a. In order for IEEE 802.11p to operate at larger transmission ranges of up to 1000 m, `aAirPropagationTime` as explained in Sec. 2.4.3 is increased. This means that the slot time grows to 16μs for the intended transmission range. As a result, all timing values which depend on `aSlotTime`, such as the IFS, increase for 802.11p.

[2] 9μs in the 1999 version of the standard [82], 13μs in the 2007 [80] and 2012 version [2].

[3] Or 13μs when configured for short range [80]. In this work we assume maximum range.

AC	AIFSN	CWmin	CWmax	TXOP
0	9	aCWmin	aCWmax	0
1	6	aCWmin	aCWmax	0
2	3	$\frac{1}{2}$(aCWmin+1) -1	aCWmin	0
3	2	$\frac{1}{4}$(aCWmin+1)-1	$\frac{1}{2}$(aCWmin+1)-1	0

Table 2.4: EDCA parameters per Access Category [81].

2.5.1 EDCA in 802.11p

Initially, the EDCA was designed to provide QoS enhancements for delay-sensitive traffic such as Voice over IP (VoIP) and streaming multimedia in the WLAN setting. It was an amendment to IEEE 802.11, finalised in 2005 as IEEE 802.11e [83]. Given the importance of some messages in vehicular networking, it was reasoned that QoS-differentiation was needed in 802.11p. In EDCA, traffic is classified based on an Access Category (AC), and each AC has its own transmission queue, as illustrated in Fig. 2.18. The ACs for priority in Medium Access are ordered based on the tolerable elasticity of the delay of a certain category of traffic. Contention delay can constitute a significant portion of the total experienced delay, especially if multiple retransmissions are required. This method of QoS differention was designed to ensure that delay-sensitive traffic is, on average, in contention for a shorter period.

The service differentiation is defined among traffic from different stations, but also among traffic from different ACs within a single node. This is known as the Virtual Collision Handler (VCH). If two or more queues contain a packet ready for transmission, the station attempts to transmit the frame with the highest priority AC. The other AC experiences an “internal collision”, it performs another retry with an increased CW if the transmission is Unicast and its retry counter is incremented. Another difference between DCF and EDCA is the way backoff counters are decremented. This is covered in detail in Chapter 5.

Service differentiation methods

The service differentiation in EDCA is achieved through differentiation of the IFS, the CW, and Transmission Opportunity (TXOP). Every queue has its own values for these parameters, as illustrated in Fig. 2.18. TXOP is defined for use with Unicast transmissions. A station with a non-zero TXOP is allowed to send frames in rapid succession for a certain duration, separated by a SIFS instead of a DIFS or AIFS. Since TXOP is 0 for all ACs for Broadcast (see Table 2.4), it is not treated further in this work. The IFS and CW differentiation have great impact on the performance of Broadcast traffic and play an important role in the remainder of this text.

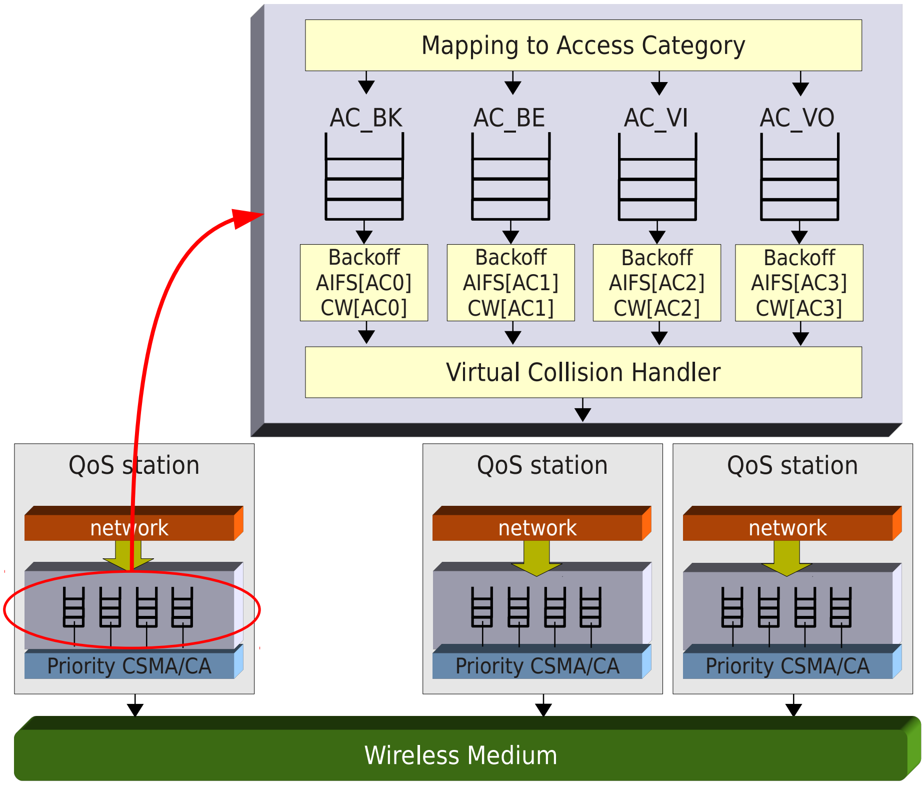

Mapping to Access Category
AC_BK
AC_BE
AC_VI
AC_VO
Backoff AIFS[AC0] CW[AC0]
Backoff AIFS[AC1] CW[AC1]
Backoff AIFS[AC2] CW[AC2]
Backoff AIFS[AC3] CW[AC3]
Virtual Collision Handler
QoS station
network
Priority CSMA/CA
QoS station
network
Priority CSMA/CA
QoS station
network
Priority CSMA/CA
Wireless Medium

Figure 2.18: EDCA QoS stations with four queues.

Arbitration interframe spacing (AIFS) As discussed in Sec. 2.4.3, the IFS functions as a priority mechanism, where the SIFS allows a receiving station to transmit an ACK frame before any other station can transmit, because these stations have to wait at least a DIFS. Similarly, EDCA allows prioritising DATA frames from one AC over the other, essentially by overriding the DIFS into durations equal to, or longer than the DIFS. This type of IFS is called the Arbitration Interframe Space (AIFS) and its duration depends on the AC of the frame currently in contention, see Table 2.4. The relation between the DCF timing and the EDCA timing is illustrated in Fig. 2.19. A higher-priority frame will have a smaller AIFS. This enables a station contending for a high priority AC to access the medium sooner than a station contending for a frame of lower priority. The actual duration of the AIFS for a certain AC follows from:

$$\texttt{AIFS}[\texttt{AC}] = \texttt{AIFSN}[\texttt{AC}] \cdot \texttt{aSlotTime} + \texttt{aSIFSTime}. \tag{2.8}$$

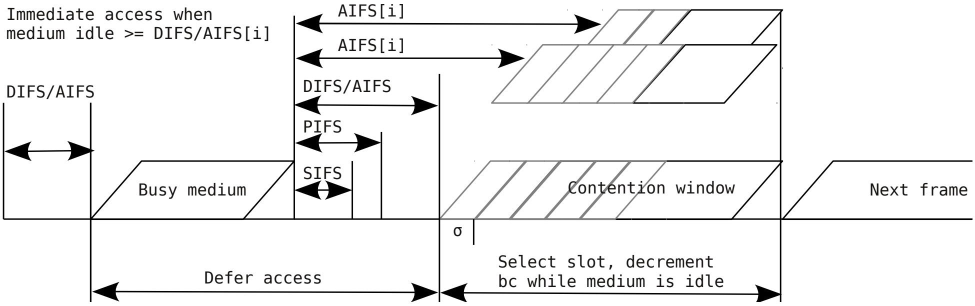


Figure 2.19: EDCA Interframe Spacing relations, based on [80].

The Extended Interframe Space (EIFS) presented in (2.7) is also superseded. The AIFS replaces the DIFS in the EIFS:

$$\begin{aligned} \mathtt{EIFS[AC]} &= \mathtt{EIFS} - \mathtt{DIFS} + \mathtt{AIFS[AC]} \\ &= \mathtt{AIFS} + \mathtt{SIFS} + T_{\text{phy}} + T_{\text{ack}}. \end{aligned} \tag{2.9}$$

This is because the station has no way to determine the AC of the observed signal on the channel. Note, however, that a station currently contending for a high-priority frame could use an EIFS which is shorter than the actual duration of the frame sent by a station acknowledging a low priority Unicast frame.

Contention window differentiation Whereas IEEE 802.11 defines a `CWmin` and a `CWmax`, and doubling of the CW in subsequent backoff stages for retransmissions, IEEE 802.11e (and hence p as well) defines a `CWmin[AC]` and `CWmax[AC]`, as given in Table 2.4. The result is a shorter average waiting time for high priority frames and a longer average waiting time for lower priority frames. This has two effects:

1. A station contending for a low priority frame spends more time counting down its *bc*.
2. A station contending for a low priority frame has a larger probability of encountering a busy medium during its contention, i.e., it must “yield” more often to higher priority traffic. This means the station must wait for the other transmission to complete, before it gets the opportunity to decrement its *bc*.

Ultimately, with increasing load this will lead to starvation of lower ACs but will ensure service to higher AC traffic. A downside of the smaller CW for high priority ACs, is that the collision avoidance characteristic, i.e., the breaking of the synchronisation for which the backoff procedure was designed, experiences reduced effectiveness when there are many stations with high priority frames. In that case, many stations will use a very small CW and the probability of two or more stations transmitting simultaneously increases.

2.6 Physical layer

Initially, IEEE 802.11 was designed for stationary use in office environments. A typical setup would be a couple of laptop computers near an AP, which in turn is connected to a LAN or the Internet. Experiments of standard 802.11b hardware in high Doppler environments [84] showed weaknesses of the Direct Sequence Spread Spectrum (DSSS) modulation scheme. A modification of 802.11a was evaluated for high mobility DSRC [85], referred to as 802.11a/RA (Roadside Access). This resulted in the physical layer used by IEEE 802.11p.

As mentioned in Sec. 2.4, IEEE 802.11 is designed for use with various physical layers because the MAC and PHY are loosely coupled; the same MAC can run on a wide variety of physical layers with only minor adaptation. The main differences between 802.11p and 802.11a include:

- Spectrum: 802.11p uses a frequency band dedicated to vehicular use, around 5.9 GHz.
- 10 MHz instead of 20 MHz bandwidth to address the increased RMS delay spread. At 20 MHz the guard interval is not long enough to prevent intersymbol interference [86].
- Symbol time: doubled from 4μs to 8μs.
- Bitrates of 3, 4.5, 6, 9, 12, 18, 24, and 27 Mbps.
- Range up to 1000 meters.

The halved bandwidth and doubled symbol time result in a halved data rate, but greatly improved robustness because the amount of energy per bit is increased. The transmission range is specified to be up to 1000 meters, although this depends on the modulation type (and its robustness) and the maximum allowed Equivalent Isotropically Radiated Power (EIRP), which depends on local regulations. In order to allow a larger propagation range, the allowable duration of the air propagation time `aAirPropagationTime`, hence, the slot time `aSlotTime` were increased (see Table 2.3).

2.6.1 Channel allocation

The 802.11b/g/n channels in the 2.4 GHz Industrial, Scientific and Medical (ISM) band partially overlap, as shown in Fig. 2.20. Although there are 14 channels spaced 5 MHz apart, only three can be used without overlap when using a DSSS modulation with a 22 MHz bandwidth. When using 802.11g, the Orthogonal Frequency Division Multiplexing (OFDM) uses a 20 MHz bandwidth, theoretically enabling use of four non-overlapping channels. However, not all channels are available in all countries due to government regulations. This overlap means that large-scale deployment—such as APs on a university campus—have to undergo carefull *site planning* in order to achieve maximum performance. This overlap also explains poor

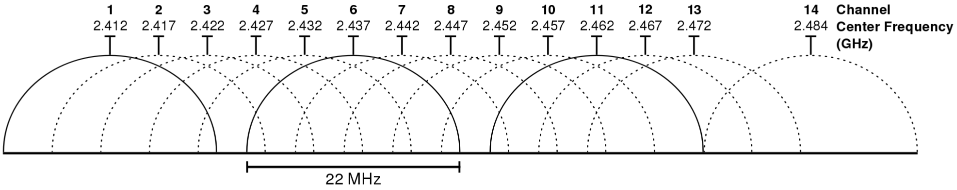

1 2 3 4 5 6 7 8 9 10 11 12 13 14 Channel
2.412 2.417 2.422 2.427 2.432 2.437 2.442 2.447 2.452 2.457 2.462 2.467 2.472 2.484 Center Frequency (GHz)
22 MHz

Figure 2.20: IEEE 802.11b,g channels in the 2.4 GHz ISM band, source: Wikimedia Commons.

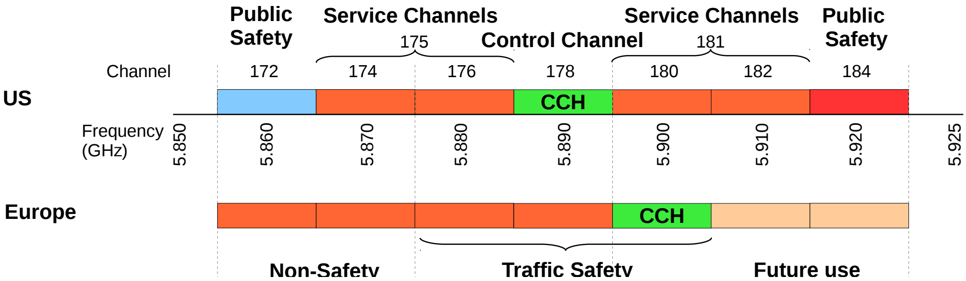


Figure 2.21: Channel allocation for ITS in both the US and the EU, based on [86, 65].

WLAN performance in densily populated areas such as apartment buildings, where deployment is not coordinated.

In 1999, the U.S. Federal Communications Commission (FCC) allocated 75 MHz of spectrum in the frequency band 5.85-5.925 GHz for V2V and V2I communications [86]. This band is divided in seven 10 MHz-wide channels. One channel is called the Control Channel (CCH) and the remaining six channels are the Service Channels (SCHs), as illustrated in Fig. 2.21. In Europe, a 50 MHz wide spectrum has been allocated for VANETs by ETSI [68]. The benefit of this non-overlapping channel allocation is that adjacent channels interfere less with each other, increasing both reliability and overall system capacity. This also eliminates the need for site planning. The OFDM modulation scheme can adhere to a steep spectrum mask, significantly reducing the amount of energy that leaks into adjacent channels.

2.6.2 Frame formats

The MAC data frame is encapsulated in an OFDM PLCP Data Unit (PPDU), illustrated in Fig. 2.22. The PLCP Preamble, see Fig. 2.23, is needed to perform various physical-layer tasks such as signal detection and synchronisation in order to perform demodulation.

The Signal and Data parts of the PPDU are illustrated in Fig. 2.24. The PLCP Preamble and PLCP Header are modulated at the most robust rate, BPSK 1/2

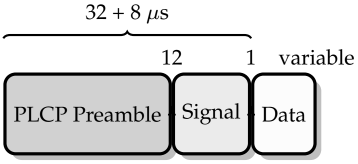


Figure 2.22: OFDM Physical PDU Format, sizes are in number of OFDM symbols, based on [2].

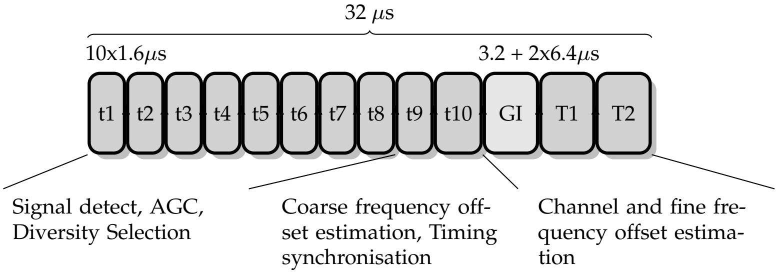

32 μs
10x1.6μs
3.2 + 2x6.4μs
t1
t2
t3
t4
t5
t6
t7
t8
t9
t10
GI
T1
T2
Signal detect, AGC, Diversity Selection
Coarse frequency offset estimation, Timing synchronisation
Channel and fine frequency offset estimation

Figure 2.23: OFDM PLCP Preamble, 12 symbols, based on [2].

(modulation is discussed further on in this section). The Data part has a length in octets, denoted by the Length field, and is coded at the Rate which is contained in the Signal fields. This rate can be a higher order modulation. The information contained in the Rate and Length fields in the Signal indicate the duration of the subsequent Data part, concerning only *this* frame.

Even if a station is not able to demodulate the Data part, it can predict the duration of the frame. The payload or Protocol Service Data Unit (PSDU) is encapsulated in the PSDU field in Fig. 2.24. The frame is padded to reach a total number of bits which is an integer multiple of the coded bits per OFDM symbol.

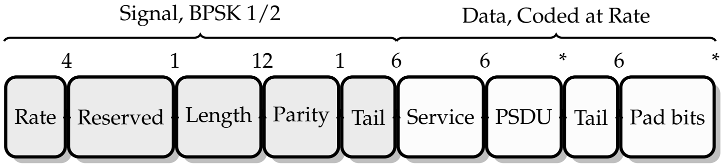


Figure 2.24: OFDM Logical PDU Format, sizes are in bits, based on [2]

OFDM is a multiplexing scheme which has very high spectral efficiency; subcarriers are orthogonal and can be placed close to each other. In a single-carrier system such as used in 802.11b DSSS, the carrier is modulated with the signal at a very high rate. In this case, a symbol occupies the entire bandwidth for a short time. OFDM splits the channel in several subcarriers, each of which is modulated at a low rate. The frequency of the carriers is chosen in such a way that they are orthogonal.

Transmission rate (Mbps)	Modulation	Coding rate	Coded bits per carrier	Coded bits per symbol	Data bits per symbol
3	BPSK	1/2	1	48	24
4.5	BPSK	3/4	1	48	36
6	QPSK	1/2	2	96	48
9	QPSK	3/4	2	96	72
12	16-QAM	1/2	4	192	96
18	16-QAM	3/4	4	192	144
24	64-QAM	2/3	6	288	192
27	64-QAM	3/4	6	288	216

Table 2.5: IEEE 802.11p modulations and coding rates, based on [81].

The result is that a subcarrier has a peak in the frequency domain where all other subcarriers are zero—they do not interfere with each other.

The subcarriers are sine waves, their phase and amplitude are modulated with the data. Four modulation schemes are available to 802.11 OFDM systems: Binary Phase Shift Keying (BPSK), Quaternary Phase Shift Keying (QPSK), and 16- or 64-Quadrature Amplitude Modulation (QAM). In ascending order, they code more data bits per symbol, achieving a greater data rate (see Table 2.5).

Every OFDM symbol has a cyclic prefix, which is a copy of the symbol prepended to the symbol during the Guard Interval. This alleviates Inter Symbol Interference (ISI) and Inter Carrier Interference (ICI). ISI occurs in a time-dispersive environment, where due to multi-path propagation symbols are dispersed in time and interfere with each other. ICI occurs under influence of the Doppler effect or poor synchronisation between transmitter and receiver, causing a slight shift in carrier frequency. This disrupts the orthogonality of the subcarriers and causes energy to leak from one subcarrier into another, increasing the Bit Error Rate (BER) [87]. The guard time in 802.11a is doubled from 0.8 μs to 1.6 μs in 802.11p. Multipath delay can cause a part of the previous symbol to arrive during the guard time. If the guard time is sufficiently large, these multipath components do not interfere with the symbol itself.

Symbol time is doubled from 4 to 8 μs. This implies that, for the same modulation and coding rate, the data rate of 802.11p is also halved. The halved data rate, however, is worth the gain in robustness because beaconing concerns very small messages and reliability is more important than throughput.

The various coding rates allow three kinds of Forward Error Correction (FEC) at the receiver. With FEC, a receiver can not only detect, but also correct errors without requiring retransmissions. Redundant bits are added to the original bitstream, at rates 1/2, 2/3 and 3/4. The data rate is the result of the type of modulation and FEC used, tabulated in Table 2.5. A convolutional encoder is used to provide n

output bits for m input bits, resulting in a coding rate of m/n. This coding spreads information from one input bit over multiple output bits, spreading itself over time.

Prior to transmission, the coded bits are interleaved over all subcarriers. In case of narrow-band interference the bit errors are spread over the bits in the input stream. This reduces the probability of having multiple bit errors close together after de-interleaving at the receiver. If there are errors in the bit stream, the FEC allows them to be corrected. This is done using a maximum-likelyhood decoder with a Viterbi-algorithm to reconstruct the original bits.

2.7 Simulation environment

An important evaluation technique applied in this research is simulation. In this approach, a model of the system under test is executed in an abstract and controllable environment. This enables study of complex behaviour and isolation of certain effects for study in more detail. This section describes the simulation tools and parameters used throughout this dissertation.

2.7.1 OMNeT++ discrete-event simulator

A discrete-event simulator [88, 89] essentially consists of a stack (i.e. an ordered list, usually in time) on which *events* are placed, illustrated in Fig. 2.25. The simulation engine takes the first event from the stack, and processes it. An event can comprise a certain action, such as consumption of resources, or scheduling of new events. State changes in the simulated systems occur at discrete points in time. These changes can be evenly spaced (*time-based*) with a focus on the progression of time, or with a focus on the events, and their relative ordering in time (*event-driven*). An important characteristic of discrete-event simulation is that time in the simulator (*simulated*

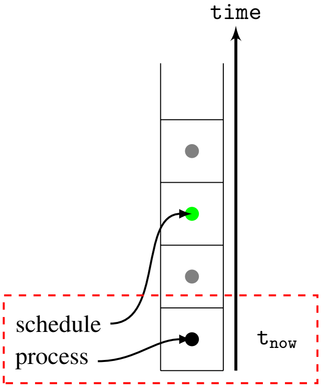


Figure 2.25: The event queue in discrete-event simulation. The processing of an event can cause scheduling of one or more events in the future.

time) does not pass the way it does in reality (referred to as *wall time*). A simple simulation, running on powerful hardware, can execute many simulated seconds per second of wall time.

While the engine is processing events, certain statistics can change over time (e.g., a count of the number of packets generated by an application). The process is bootstrapped by manually scheduling the first event or set of events, after that it can theoretically run for ever. In order for the simulator to terminate, a stop condition is defined. This stop condition can be to stop at time t, after creation of e events, or when some other condition is met.

The execution of a simulation *run* has three phases: initialisation or start, the event loop, and termination or finish. During *initialisation* (which can consist of multiple phases itself to allow creation of complex dependencies) the simulation models are constructed and parameters are set. For instance, a network module is connected to a host module, and its bitrate is set. These parameters are usually read from configuration files but can also be supplied at runtime. The bootstrap events are also scheduled in the initialisation phase.

In the *event loop* phase, the simulation engine continually loops while the stop condition is not satisfied. In each iteration, only a single event is treated, after which the stop condition is evaluated. When the stop condition is met, the simulation proceeds to the *termination* phase, and various statistical reports can be written.

A discrete-event simulator can simulate perfectly deterministic behaviour when using very naive models. In order to get real-world randomness into the simulation, pseudo-random number generators are used. The resulting random number stream can be used to simulate, e.g., drawing a bc from a CW when modelling CSMA/CA.

Most simulation studies in this work use the method of independent replications [88]. Under this approach, various *iteration variables* are defined, such as the number of nodes n and generation rate λ_g. Each combination of these iteration variables defines a scenario (n,λ_g), and each scenario is replicated a number of times. It quickly becomes evident that study of many iteration variables explodes the total number of runs which need to be executed. However, because every run is independent, this method allows distribution of the entire simulation experiment over many CPUs in a compute cluster.

Each replication is performed with a different seed for the random number generator. This has the benefit that, within a scenario, each run is unique, but reproducible. When performing a certain run with a defined combination of iteration variables over and over, the results will remain the same.

Each simulation experiment is performed with enough replications to get statistical significant results, and 95% confidence intervals were calculated (unless stated differently). If confidence intervals are not shown in the plots in the remainder of this work, they are well below 5% of the metric and omitted for readability.

The event-loop phase of each run starts with a warm-up period. This is necessary because the system needs to transition from empty to steady-state. During this transient period, buffers may not yet have reached a stable level, or some nodes may even still be inactive. In order to study the system under steady-state conditions, it is important that no results are collected in the transient period because these may taint the results. As soon as the warm-up period is over, measurements start and results are collected.

Simulation runs in this dissertation are time-bounded. A time-bounded stop condition has the benefit over an event-bounded condition that the simulation simply stops by freezing, instead of slowly coming to a halt because some nodes have already generated the required number of events. In case of event-boundedness, nodes will complete transmitting their number of beacons and stop one-by-one. This results in a system of $n-1$ nodes for the remaining nodes. This taints the results because this is transient behaviour, similar to that which the warm-up period addresses. The difference is that nodes are removed from the system instead of added, and buffers need to empty instead of fill.

Nodes in the simulator start generating beacons after a random initial delay to bootstrap the run. If nodes were to start at the same time, they would immediately be synchronised, which may affect the remainder of the run.

The discrete-event simulation platform used throughout this work is OMNeT++ [90] (Objective Modular Network Testbed) [91], a highly modular simulation platform written in C++. It dates back to 1992 but in its most recent incarnation it is integrated with the Eclipse IDE and encompasses a high-level system description language (NED) which allows composition of modules and behaviour. Many models and extensions are available.

2.7.2 MiXiM mobile and wireless framework

MiXiM [92] is a simulation framework for simulating wireless and mobile communication using the OMNeT++ simulation engine. Application areas include body-area networks, wireless sensor networks, and vehicular networks. It is a merger of several components, of which the Mobility Framework [93] is the most prominent.

MiXiM defines a `ChannelControl` module which manages the connections between nodes [94]. It does this dynamically during simulation run-time, depending on whether nodes are within eachother's interference range. For a transmission from one node to another, a `Signal` object is created and based on a propagation model, the time and frequency components for the receiver are calculated. In the free-space pathloss model, one of the most straight-forward propagation models, a signal's power decays with distance. Next, the `Signal`s of all other nodes in range are summed as interference. For a receiving node, the Signal-to-noise Ratio (SNR) is used to derive a bit-error probability. The BER model depends on the type of modulation, and the FEC. The random number stream and the BER model are used to stochastically decide whether the transmission could be decoded, or not.

This method of evaluation of the physical channel is computed for every transmitter-receiver pair, essentially resulting in a fully-connected graph. An optimisation to make the evaluation more scalable is to split the simulation area into a grid, and calculate—based on the free-space pathloss model—which nodes are within eachother's interference range. This grid is referenced to obtain the transmitter-receiver pairs which need to be evaluated. This improves the scalability of the simulator, because only possible communication between relevant nodes is evaluated.

MiXiM provides an IEEE 802.11 implementation (Mac80211) which models the IEEE 802.11b DCF for both Broadcast and Unicast, but for ad hoc mode only. The 802.11 state machine is modelled in a generic way, enabling mixture of Broadcast and Unicast packets depending on the address of the recipient. The RTS/CTS sequence, CW increment and retransmission for both long and short packets are implemented. Mac80211 does not generate management frames such as the beacon frames generated by an AP. Likewise, fragmentation and detection of duplicates are not implemented. Mac80211 has a single transmission queue, the length of which (in number of frames) can be configured. Data rates supported by the physical layer model are 1, 2, 5.5, and 11 Mbps. At the PHY, a bit error probability is computed based on BER models. Transmission power and the channel at which to transmit can be configured.

Various extensions were implemented for the simulation studies throughout this dissertation. These will be treated in Sec. 5.5.1.

2.8 Summary

This chapter discussed the background of this dissertation. The concept of Intelligent Transportation Systems was introduced and the vital role of vehicular networking was made clear. The application of Cooperative Adaptive Cruise Control was discussed, and how vehicular networking can realise significant performance gains in future road transportation by means of the CACC system. Two projects implementing a CACC system were briefly introduced, their technical details will be further discussed in Chapter 4.

Next, the Network, Medium Access Control and Physical layers used to enable vehicular networking were described. Network architectures were briefly introduced, in particular IEEE 1609, ETSI ITS, ISO CALM, and the C2C-CC reference architecture. Although standardisation at the network layer and above is still ongoing, the MAC and physical layer technology in these architectures converges to IEEE 802.11p, referred to as CALM M5 within the CALM architecture. The remainder of this work abstracts from specific network architectures, with focus primarily on the medium access mechanism and the beaconing communication pattern.

The inner workings of IEEE 802.11 and the modifications made in 802.11p were explained, because they play a central role in the remainder of this dissertation and have great impact on the performance of beaconing. In particular, the medium access mechanism, CSMA/CA, was described in detail, as was the EDCA service differentiation mechanism. Subsequently, a brief description of OFDM, the physical layer technology employed in IEEE 802.11p, was provided.

This chapter finished with a description of the OMNeT++ simulation engine and the MiXiM framework for wireless and mobile communication. Various extensions were implemented to simulate vehicular networks with IEEE 802.11p, these will be described in Sec. 5.5.1.

CHAPTER 3

Beaconing solution space

In the previous chapter, the background and context of vehicular networking were explained, and the CACC application was introduced. Also, a brief overview of the various standardisation efforts was provided. This chapter aims at a more conceptual and implementation-independent discussion about the principles used to enable real-time vehicle control by means of beaconing. To this end, we first more precisely define beaconing in Sec. 3.1. Next, we explain the value of the received information when collected in a structure we call the *cooperative awareness* in Sec. 3.2. Scalability and its facets are described in Sec. 3.3. A framework from the literature is applied to ease the discussion on particular aspects of vehicular networking and their impact on scalability. The information elements in the cooperative awareness are sufficient to enable many ITS applications and in particular CACC, if certain requirements are met. Sec. 3.4 discusses these requirements, and answers research question 1 as posed in Chapter 1 ("What are the requirements of CACC on the communication system?").

Next, we introduce a set of performance metrics in Sec. 3.5. These metrics are used to gauge if the performance of the beaconing system is adequately meeting the requirements. In addition, these metrics can be used to gain insight in the operation of the system, or can be used to isolate bottlenecks. This yields an answer to research question 2 ("How do we evaluate whether a beaconing system is operating such that it meets the requirements? Which metrics can be used? ").

In Sec. 3.6, a channel load model is described to evaluate the boundaries and ranges of parameters within which the beaconing system can exist to meet the requirements of the CACC application. This constitutes an upper-bound to the performance and provides insight in the limits of the system. A simple probabilistic expression is used to get a notion of the probability of performing a successful transmission within this solution space. This answers research question 3 ("What are the limiting factors of the performance of a beaconing system?").

A conclusion is provided in Sec. 3.7. This chapter is partly based on previously published material in [9, 25, 23].

3.1 Beaconing

According to the Merriam-Webster dictionary, the term *beacon* has the following meaning:

1. a signal fire commonly on a hill, tower, or pole
2. (a) a lighthouse or other signal for guidance
   (b) a radio transmitter emitting signals to guide aircraft
3. a source of light or inspiration

In the context of this dissertation, definition 2 (b) is the most appropriate; signals are emitted to guide vehicles. But whereas radio transmitters to guide aircraft have a fixed position, the vehicles in vehicular networking change position all the time. In the case of aviation, the device producing the signals is the beacon; transmitting beacon signals. In the vehicular context, a vehicle is equipped with a device which transmits beacon messages, although this device is not referred to as a beacon. The similarity with aviation becomes clear in the CACC context: the information received in the beacon messages is used for guidance of the vehicle. In the context of vehicular networking, we define beaconing as follows:

Definition 2. *Beaconing: the periodic broadcast transmission of short status messages by vehicles or other entities in vehicular networking, containing both static and dynamic information about these entities.*

There are several concepts in (wireless) networking which can be identified by the term beaconing. Their function is to spread information and generate a distributed awareness, but in the traditional Internet context their scope is limited to Data link, Network or Application layer, as illustrated in Fig. 3.1.

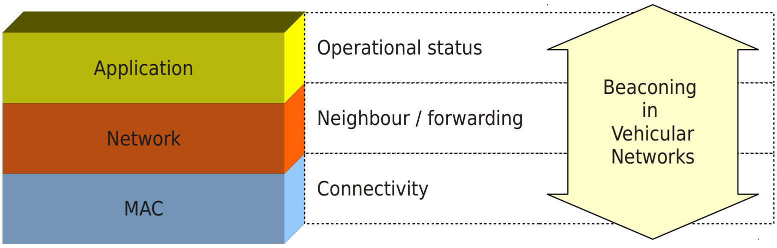


Figure 3.1: Beaconing in the protocol stack.

At the **MAC layer**, an IEEE 802.11 AP periodically announces its presence by broadcasting a Beacon frame [80] as part of the LLC function. This frame contains information necessary for a node to associate with the AP. In addition, it also provides a synchronisation mechanism and plays an important role in power saving functions in mobile devices. In 802.11 ad hoc mode, nodes transmit Beacon frames in a distributed manner [80]. At the **Network layer**, some routing protocols rely on one- or two-hop neighbour knowledge. This knowledge can be gathered when needed, like in the RREQ/RRESP procedure in Ad hoc On-demand Distance Vector (AODV) [95], or *a priori* as done in Destination-Sequenced Distance Vector (DSDV) [96] or Optimised Link State Routing (OLSR) [97] by keeping neighbour knowledge in a routing table. In the a priori case, the information in the routing table is kept up-to-date by means of the proactive exchange of small presence messages called beacons or *hello messages*. At the **Application layer**, applications typically exchange data related to their operational status. They could use the beaconing communication pattern to do so.

In a vehicular network, every operational node periodically transmits short status messages. These messages are referred to as routine messages, heartbeat messages, or beacons. Beaconing is one of the prominent communication patterns in vehicular networks, as described in Sec. 2.3. Each message is an atomic entity. This means that correct reception of a frame does not depend on successful reception of prior or subsequent frames. To be eligible to this atomicity, a beacon must be contained in a single frame on the wireless medium.

The objective of beaconing is to create a distributed, cooperative awareness among all nodes. Information contained in the beacons covers geographical topology, but also operational status of the vehicles. Spectrum resources are scarce, and if some piece of information is already present in a node it does not make sense to acquire it over the air interface. In this context, beaconing is of a cross-layer nature. A single information element (such as a node's position) can be required by many applications and protocol functions at different layers. This departs from the traditional networking approach used to great success in the Internet and in LAN deployments. In this approach, separation of concerns and encapsulation allow user data to be transparently transmitted from one protocol entity to the other. With the presence of beaconing in vehicular networks, many protocols and applications do not need to implement their own support functions. Dissemination protocols do not need to implement their own hello messages, because the beacons already contain information from which the network topology can be derived. An example is the work in [98], which describes using the information received from beacons to increase the performance of data dissemination schemes in a vehicular network.

A result of this cross-layer scope is that, in contrast to beaconing at the MAC or Network layer, beaconing in vehicular networks does not support a service, it *is* the service. It is a communication pattern which differs from those applied in the scope of the Internet and LANs. This means that lessons learned there need to be carefully reconsidered before being applied in the vehicular context.

Because of its cross-layer nature and how beaconing is embedded in its context, it becomes evident that beaconing, and the cooperative awareness generated by it, could probably satisfy most of the information requirements for many ITS applications. This reduces the need to exchange additional messages to support these applications, at least within the scope of the ad hoc domain.

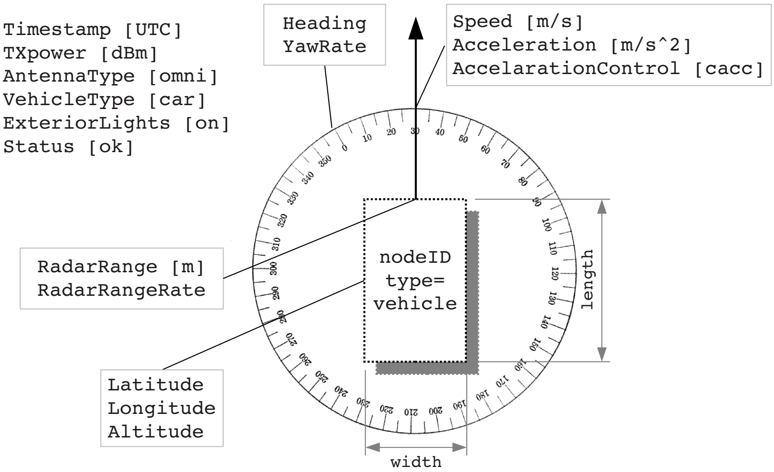


Figure 3.2: Information about a specific vehicle in the cooperative awareness as used in the C&D project [73], based on EIVP [71].

3.2 Building cooperative awareness with beacons

The information contained in received beacons can provide a vehicle with an extensive awareness of its surroundings, which can extend well beyond the range of on-board sensors or the range of vision of the driver. This awareness (also referred to as Local Dynamic Map (LDM) [99]) contains information received from vehicles, roadside units, traffic centers and on-board sensors. The LDM as described in [99] provides a reference architecture from which this work abstracts[4]. An interesting observation is that CACC is not a use case considered in the definition of the LDM [99], because its focus is limited to advisory systems. We define cooperative awareness as follows:

Definition 3. *Cooperative Awareness: a collection of static and dynamic information elements in a node, received from and shared with other entities in the vehicular network, concerning the environmental context and operational status of these entities.*

The cooperative awareness can be illustrated with another concept from aviation: a radar screen as used in air traffic control. A forward-looking radar or lidar used by an ACC system as described in Sec. 2.2 could be seen as an automotive counterpart of the radar system used in aviation: it detects presence of obstacles in front, and their relative position. Vehicular networking adds to these "blips" on the radar screen a rich scala of additional information, as illustrated in Fig. 3.2.

[4] Also, note that [99] uses "Co-operative Awareness" as an application. In this dissertation we use the term cooperative awareness in a broader scope.

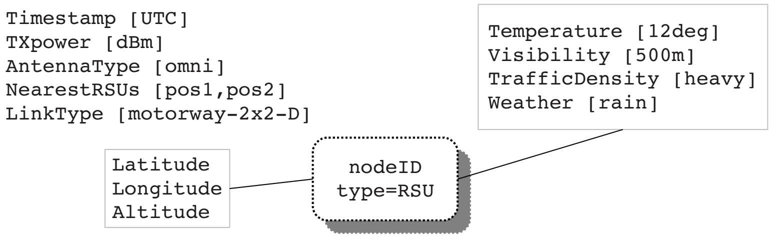


Figure 3.3: Information about a RSU in the cooperative awareness as used in the C&D project [73], based on EIVP [71].

For instance, consider the EIVP as used for beaconing in the C&D project to realise a CACC system as described in Sec. 2.2. Received beacons are used to build and maintain the cooperative awareness. New entities are detected and the status of their representation in the awareness is updated with every received beacon. As shown in Fig. 3.2, a node's position in 3D space, speed and acceleration are known, as is its heading and the yawrate. For each individual vehicle, a trajectory can be estimated, because this object must obey the laws of physics. A UTC timestamp conveys when this information was captured[5]. In addition, the information in Fig. 3.2 shows the object identified by `nodeID` is a car, with its exterior lights on and it is operational (and not crashed).

This is all information about vehicle `nodeID`, but this vehicle can also tell something about *its* environment, as present in the fields added within the C&D project: the `RadarRange` and the `RadarRangeRate`. Through these fields, this vehicle conveys whether it detects a vehicle in front, and at what rate it is approaching. If the vehicle driving in front of vehicle `nodeID` is a legacy vehicle, i.e., not instrumented, or it is simply out of our radio range, it can still be detected in some cases.

Collecting this information from many vehicles in the proximity allows selection of potential collision dangers, but also allows selection of a vehicle to use for a cooperative following application such as CACC. In this case, information about the dynamics of this vehicle is fed into the CACC system.

Cooperative awareness is not restricted to objects in the V2V domain, an RSU also becomes part of the cooperative awareness by means of the information it transmits. This is illustrated in Fig. 3.3. Within C&D, RSUs were envisioned to play a role in the coordination of merging on highways as described in detail in [51, 52, 53].

The information in the cooperative awareness does not necessarily originate solely from the beacon messages. Weather reports or traffic updates sent through other communication media can also contribute to the cooperative awareness, but are out of the scope of this work.

Many applications can be defined which can use the information in the coop-

[5] Or actually, when the beacon was generated. These two moments do not necessarily coincide.

erative awareness, without the need to generate additional communication on the network. An application such as the Electronic Emergency Brake Light (EEBL) [100] which, upon decelerating beyond a certain predefined threshold, transmits an EEBL report could be made to simply obtain this information from the cooperative awareness by tracking nearby vehicles, and raise an alarm if one of them performs an emergency brake action. This would eliminate the injection of additional messages into the network. In the GCDC project, this is aided by an optional `VehicleSpeedIntent` [54] information element in the beacon, which is essentially an array of (time, speed) tuples. Likewise, Intersection Collision Avoidance (ICA) [101] applications would not need hardware installations at every intersection, or introduction of additional messages into the network if they could rely on information in their cooperative awareness. However, with this last example we encounter a fundamental problem: non-instrumented vehicles are unlikely to show up in the cooperative awareness, unless detected by the on-board sensors. The same goes for pedestrians and bicyclists.

3.3 Scalability

The term scalability is often used as a favourable property of a system. However, scalability is difficult to formalise [102] and its characteristics are usually only apparent from the context. The concept of scalability is often understood as the ability to accomodate a growing number of elements or service requests, or to be easily expanded to greater capacity. This growth can affect processing, storage, or other system properties. A linear growth could result in similarly linear growth of resource usage (indicated as $\mathcal{O}(n)$) or accelerated resource usage, i.e., polynomial growth ($\mathcal{O}(n^c)$) or exponential ($\mathcal{O}(c^n)$), with $c > 1$. In general, a system is understood to be scalable if it remains practical within the range of expected growth.

Scalability is crucial to the long-term success of a system [103], so it comes as no surprise that the C2C-CC mentions it as one of the four most important technical prerequisites for vehicular networking [60], see Sec. 2.3. We use six aspects of scalability as defined by Bondi in [103], noting that two or more types may interact.

3.3.1 Load scalability

A system is *load scalable* if it can perform its task without undue delay or waste of resources at light, moderate or heavy loads. Factors identified in [103] that can undermine load scalability are (1) the *scheduling* of shared resources, (2) the scheduling of a class of resources in a manner that increases its own usage, referred to as *self-expansion*, and (3) inadequate exploitation of *parallelism*.

Beaconing and load scalability

In [103], the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) in Ethernet is used as an example of a system which is not load scalable. The

CSMA/CA in IEEE 802.11p follows a similar medium access pattern.

The number of nodes and the supplied work are related: adding more nodes increases the amount of supplied work. The amount of supplied work per node is, especially when compared to modern (Wireless) LANs, constant and very benign in quantity. With a generation rate λ_g of 10 Hz and an EIVP beacon of approximately 400 bytes, a single node only supplies a few kilobytes per second of workload.

As a vehicle navigates a route through traffic, the experienced traffic density can vary greatly and quickly. A certain road may have a low number of vehicles, but when this road intersects with a busier road, the number of nodes in range will suddenly increase. Many applications are defined to operate at these intersections, e.g., collision avoidance and intelligent traffic lights. These applications help the traffic system cope with the increased load in this situation, but this goes hand-in-hand with an increased load on the wireless networking capabilities of the nodes. It is imperative that the wireless networking does not become the bottleneck. Below we will cover the three aspects of load scalability described in the above: scheduling, self-expansion and parallelism.

Scheduling The MAC is involved in scheduling transmissions. It is well known that CSMA/CA's ability to successfully schedule transmissions deteriorates as the load increases. These collisions result in waste of resources and reduced quality of the cooperative awareness.

EDCA's priority mechanism enforces distinct load scalability for each AC (see Sec. 2.5.1). By dropping low priority traffic in the VCH, the load scalability of higher priority traffic is improved at the expense of lower priority packets. The dropping of packets when a queue is full improves load scalability, because it prevents using shared resources from the medium to attempt transmission for these frames, thereby affecting the loss and delay of other nodes. In beaconing however, dropping of new arrivals when a queue is full results in the transmission of old information which may be discarded at the receiver anyway. This form of wasteful scheduling is addressed by means of the Oldest Packet Drop (OPD) scheme described in detail in Sec. 6.3.

Self-expansion IEEE 802.11 has features which could be considered to be self-expanding. In particular in Unicast, a single packet generated by the network layer can cause a number of retransmissions on the medium, multiplying the effective load on the system. This behaviour is absent in beaconing because of the use of the Broadcast mode for beaconing. The Broadcast Storm phenomenon present in the flooding dissemination scheme [104] is another example of self-expansion. This is not present in the beaconing paradigm, but may be present in some other schemes which coexist with beaconing. An example is provided in Sec. 4.3.1 in the discussion of interactions in cooperative following.

Parallelism Exploitation of parallelism, in a wireless network, depends on the ability to realise spatial frequency reuse. If two nodes are outside of each other's range, their transmissions can be performed in parallel. With an increased transmission range in 802.11p, its exploitation of the potential to perform certain operations in parallel is reduced.

3.3.2 Space scalability

A system is *space scalable* if its memory requirements do not grow to intolerable levels with increase in the number of elements it supports [103]. E.g., if memory usage grows with $\mathcal{O}(n^2)$, resources may quickly become depleted. Space scalability may be achieved at the expense of load scalability, for instance by using compression.

Beaconing and space scalability

Space scalability is not a dominant aspect of beaconing, although one could reason the number of bytes in the payload of the beacon message to exhibit space scalability properties. Adding an element to the beacon format results in linear increase of the signal duration. Moreover, if all information is collected into a cooperative awareness as described in Sec. 3.2, adding an information element to the beacon description means this element will exist for all vehicles in the cooperative awareness.

3.3.3 Space-time scalability

A system is regarded as *space-time scalable* if it continues to function gracefully when the number of objects it encompasses increases by several orders of magnitude. An example is the time needed to search in a certain collection of data. The efficiency of an algorithm has great impact on space-time scalability. In this sense, a hash table is more space-time scalable than linearly going through all elements. It is mentioned in [103] that space scalability is a prerequisite for space-time scalability. For example, if the number of elements is excessive, searching through it may be a costly operation. In addition, if a system contains more elements this may result in increasing demands on its load scalability.

Beaconing and space-time scalability

As mentioned above for space scalability, the data collected in the cooperative awareness should be space scalable. Using, updating, and deleting this information should be space-time scalable. If some hardware is running estimator algorithms for vehicles in the cooperative awareness and the same hardware is responsible for generating the beacon messages, this subsystem should be space-time scalable enough to remain able to generate the beacon messages on time, independent of the number of elements in the cooperative awareness.

3.3.4 Structural scalability

The property of *structural scalability* means that a system is not limited by its implementation or standards in its ability to support the number of objects it encompasses, or at least not within the time frame and context in which the system is planned to be active. A good example here is the IPv4 addressing scheme; the number of elements in the system is limited by the size of the address space.

Beaconing and structural scalability

The channel switching mechanism proposed in IEEE 1609.4 which was described in Sec. 2.3.4 reduces structural scalability. The amount of resources available to a single-transceiver system is diminished, resulting in a reduction in load scalability because the system has fewer resources to perform its tasks. An evaluation of the impact is provided in Sec. 6.4. The use of EIFS in a Broadcast environment also causes reduced structural scalability, an evaluation is provided in Sec. 5.6.2.

3.3.5 Distance scalability

A protocol or system is said to be *distance scalable* if it performs equally well over long or short distances. An example is a timeout in a request-response sequence. This timeout should be sufficiently sized to encompass the entire round-trip time.

Beaconing and distance scalability

In light of distance scalablity, the 802.11p slot time σ needs to be adjusted for the increased transmission range. This in turn has effect on the guard spaces in time (the IFS) used by the medium access mechanism. This results in a (slight) increase in the overhead and decrease of the time spent performing useful work, evaluated in detail in Sec. 5.6.1.

3.3.6 Speed/distance scalability

A system is *speed/distance scalable* if it performs well over long and short distances both at high and low speeds.

Speed/distance scalability in vehicular networking

Physical phenomena like the Doppler effect impose a limit on the maximum acceptable relative speed two nodes may have before the physical layer becomes unable to successfully maintain synchronisation to the received signal. This makes it particularly difficult to correctly demodulate large frames, or frames transmitted by vehicles with a very high relative speed [105]. This is not a practical problem in the application of CACC, where relevant vehicles are considered to be moving in the same direction.

3.3.7 Scalability: concluding remarks

The scalability aspects discussed above impose certain limitations, constituting a solution space within which the beaconing system can exist. Some limitations may be reached before others, some limits may not even have impact within the scope of purpose and lifetime of the system. To judge the scalability of a beaconing system it is important to explore these limits. An evaluation of this is presented in Sec. 3.6, and the remainder of this dissertation will add more detail to these limits.

Load scalability can generally be enhanced by improving scheduling, avoiding self-expansion, or improving the exploitation of parallelism [103]. Poor load scalability manifests itself as an increase in waste and reduction of efficiency. The other scalability aspects are determined by architectural characteristics, standards, policies or legislation. These aspects can have a limiting effect on the system. If this limit is well outside the scope in which the system is perceived to function, these structural scalability issues will not manifest in reduced performance of the system.

These facets of scalability come from the automotive and wireless networking worlds, which converge under the concept of vehicular networking. Many of these scalability issues (such as coping with extreme temperatures or humidity) can be solved by the choice of materials and manufacturing procedure of the physical device. Dealing with local legislative domains is a matter of properly configured locale mechanisms or harmonised standardisation.

This dissertation is primarily concerned with load scalability, i.e., the growing number of nodes, but also addresses some structural scalability aspects.

3.3.8 Graceful degradation

Next to scalability, a favourable property of a system is its ability to *degrade gracefully* once its scalability limits are approaching or when faced with adverse conditions. In this respect, we define graceful degradation as follows:

Definition 4. *Graceful degradation: a system's ability not to fail catastrophically, but instead experience a gradually reduced ability to function, when it is operating near its scalability limits or affected by failures.*

In this sense, graceful degradation is a relaxed form of *fault tolerance*, where the system should remain working even under adverse conditions [106]. Complete fault tolerance may not be feasible because some aspects are out of the control of the system designers, or because of trade-offs between cost and performance.

The traffic flow on a road degrades to a traffic jam when more vehicles enter the road, as described in Sec. 2.2.3. Likewise, the beaconing system should also gracefully degrade as the number of nodes increases, as opposed to suddenly seizing to work. In case of a traffic jam, vehicle dynamics are low or vehicles even come to a complete stand-still. Since the wireless channel resources are limited, it succumbs to a similar congested state under an increasing number of nodes. Subsequently, performance degrades. Because of the decrease in traffic dynamics, the entire system is able to cope with a decrease in performance of the beaconing service.

Due to the nature of radio propagation, some percentage of messages may not arrive, even under light load on the channel. The result is that the applications should as well be able to deal with reduced performance of the wireless communication by degrading gracefully. An example is the CACC application described in detail in Sec. 4.2, which falls back to ACC when the received information does not meet the requirements.

3.4 Requirements for beaconing imposed by CACC

Real-time control applications such as CACC impose certain requirements on the communication system. The requirements presented here are based on those derived by the C&D project [7] described in Sec. 2.2.4. The deliverable containing the C&D CACC system requirements [23] covers many requirements for various aspects of the system. Here we only list the requirements relevant for the discussion in this dissertation.

CACC as implemented in C&D and considered in this dissertation depends solely on the beacon messages generated by the vehicles. No additional interactions are required and the communication is entirely stateless. The application does not depend on the presence of RSUs, the communication exists only in the ad hoc V2V domain. The received information is treated as indicative, the vehicle's control algorithms will decide if and which action to take.

3.4.1 Sample rate: number of beacons per second

The generation rate λ_g determines the number of beacons generated per second by a node. The initial requirement was a generation rate λ_g of 25 Hz [23], which was later in the implementation phase relaxed to 10 Hz [107], conform ETSI ITS specifications [4]. A beacon rate of 10 Hz yields a working system (see Chapter 4) confirmed by simulation experiments (see Sec. 4.5). Later, the GCDC project also specified 10 Hz as a generation rate (see Chapter 4).

Note that the rate at which samples are fed into the controller to a great extent determines the CACC performance. This rate is referred to in this dissertation as reception rate λ_r, defined in Sec. 3.5.3, and is not necessarily equal to λ_g due to packet loss caused by collisions on the wireless medium.

3.4.2 Delay: the freshness of received beacons

CACC aims to circumvent the relatively long delay in human reaction time and estimation algorithms in ACC. To accomplish this, the received information needs to be fresh. Once the signal has been sampled, it should not take excessively long before it is used in the control loop. Increased delay negatively influences the system's string stability. Within the C&D project, the delay requirements were specified as at most 200ms [23].

3.4.3 Range

Range is important for a CACC system for several reasons. The primary reason is that the cooperative awareness is used to enhance on-board sensors such as radar or lidar. The wireless communication must extend to *at least* the same range to augment the information supplied by the on-board sensors. In C&D, the range was defined as approximately 200 m or 15 vehicles, whichever comes first [23]. This notion of range is called *relevance range*. Depending on the control structure used (see Sec. 4.2.1), the relevance range could be vehicles in the driving direction, or also those behind the vehicle. From the vehicles in the cooperative awareness, the CACC will select the most relevant vehicle to follow.

The second reason is that wireless communication has the benefit over radar and lidar that it is not strictly line-of-sight. Wireless communication allows a vehicle to receive information from several vehicles ahead whereas the radar stops at the first vehicle in front. This allows to do more than simply ranging to the lead vehicle—the CACC system could take multiple cars as input, as discussed in Chapter 4. A favourable property of the broadcast nature of radio waves is that the beaconing mechanism can be constructed independently of the information flow structures used by the controller, as long as the vehicles in the relevance range have a sufficiently accurate representation in the cooperative awareness. When omnidirectionality is required like in CALM M5 [64], these requirements must be generalised to the entire system.

3.4.4 Communicated information

The information should be communicated using a standardised format, such that each vehicle is able to receive and interpret the information. The information required by CACC is typical to any safety application, and is almost fully covered by COMeSafety's EIVP Cooperative Awareness Message (CAM) [62]. This format has been chosen as the beacon message format. Several fields have been added (see Sec. 2.3.5), resulting in beacons of 400 bytes including security fields.

3.5 Performance metrics

Various metrics can be used to quantify the performance of a beaconing system, these will be addressed below. The experiments and evaluations carried out in the remainder of this dissertation evolve around quantifying these metrics.

In order to explain these metrics and their influence on the ability of the wireless communication system to transfer information from one node (denoted a) to another node (b), Fig. 3.4 is used as a reference in this section. In this figure, a local datastorage DS_a in vehicle a is refreshed with sample rate λ_s; we assume samples have zero delay at this moment. Beacons are generated at rate λ_g, without synchronisation between the sensors, datastorage and beacon generator. The beacon messages are inserted into the MAC queue. Next, they undergo contention and subsequent transmission.

In the receiver vehicle b, an evaluation algorithm which updates at rate λ_e takes the received information from the cooperative awareness denoted as CA_b. In this representation, each step from top to bottom could incur some delay and some steps have a loss probability. The samples in Fig. 3.4 have a sequence number, used to distinguish several scenarios. Although they are illustrated seperately here, these scenarios are not independent in practice. We divide the metrics in those related to *loss* in Sec. 3.5.1, *latency* in Sec. 3.5.2, and other metrics in Sec. 3.5.3.

3.5.1 Packet loss probability

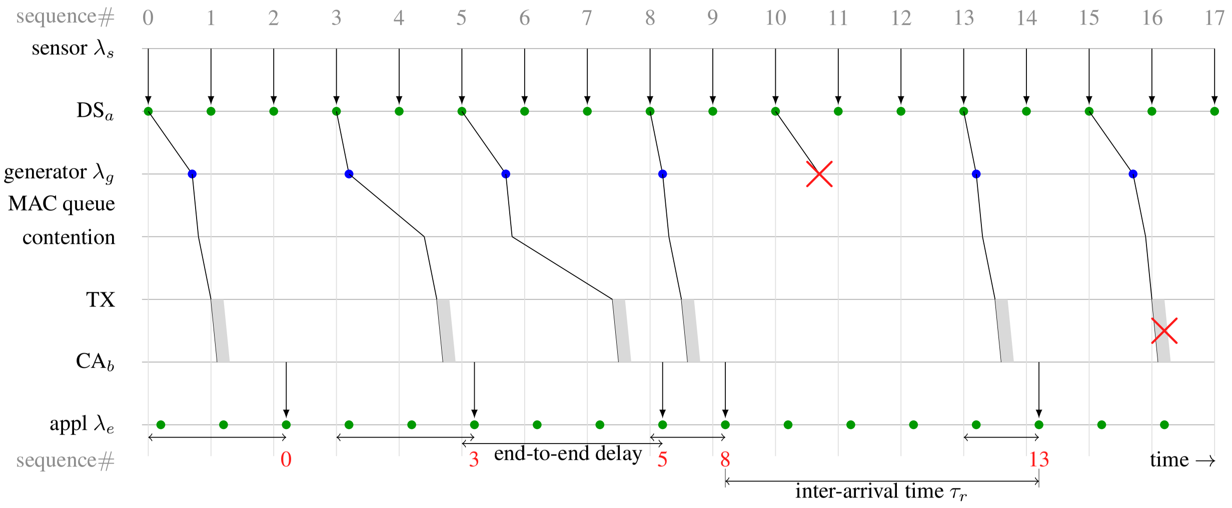
One of the most fundamental properties of a wireless multiple-access network is the probability of packet loss. Packet loss determines which ratio of the supplied load is not successfully delivered. Throughout the evaluation experiments presented later in this dissertation, the success probability P_s is often used as a metric. This is the probability that a beacon sent by node j is received by node i. The evaluation performed in Sec. 4.5 explicitly varies P_s to study sensitivity of the CACC application to packet loss. There are several factors which contribute to packet loss which manifest itself by the inability of the receiver to reconstruct the message as generated by the originator. These factors are treated below and summarised in Table 3.1.

Queueing loss is caused by the dropping of packets when a buffer of finite size is full. When another packet arrives, there are two options: drop the arriving packet, or drop a packet from the queue and add the new packet to the queue. Either way, a packet has to be dropped, contributing to packet loss. This is illustrated by the red cross on the path of sample 10 in Fig. 3.4.

Queueing loss can be expressed as a dropping probability P_{drop}. This is found as the number of beacons dropped out of the queue (thus not transmitted) at node i, divided by the number of beacons generated by node i at the network layer.

Collision loss is caused when two or more signals physically "collide" on the communication medium. Colliding signals can be of differing magnitude and phase, which means that one signal can modulate the other. When the resulting signal arrives at the receiver, it differs from the transmitted one. In the PHY, this will result in disturbed symbols in the OFDM signal. On the frame-level, this will manifest itself as flipped bits, some of which may be recovered by the FEC covered in Sec. 2.6.2. If the frame cannot be recovered, this results in failed CRC computations. It makes no sense for the MAC to hand the packet to upper layers, the contents are unpredictable. In Fig. 3.4 this is illustrated by the red cross on the path of the transmission with sequence number 15.

Propagation loss is caused by attenuation of the signal as it moves through the ether, or by interference from other RF sources. Attenuation causes the signal to become weaker at receivers further removed from the transmitter. As the signal



seq. #	Scenario
0	Normal operation
3	Long queueing delay
5	Long contention delay
5, 8	Short burst - two arrivals in rapid succession
10	Queuedrop
13	Normal operation
15	Collision

Figure 3.4: Latency and loss diagram: transferring data from a sensor at node a to an application at node b by means of beaconing, illustrating several scenarios.

Component	**Symbol**
Loss probability, Success probability	P_s
Queueing	P_{drop}
Collision	P_{coll}
Propagation	P_{prop}

Table 3.1: Loss components and their symbols as used in this dissertation.

becomes weaker, it is harder to receive and more sensitive to interference. In the extreme, propagation causes the signal to be attenuated to a level below the receiver's carrier sense threshold, rendering the receiver oblivious to the transmission. Interference causes the signal to be modulated by other signals or a delayed version of the signal itself. Examples include shadowing, fading and the ISI and ICI discussed in Sec. 2.6.2. The exact impact of these effects is out of the scope of this work.

These loss components are related as follows:

$$P_s = (1 - P_{drop})(1 - P_{coll})(1 - P_{prop}) \tag{3.1}$$

3.5.2 End-to-end delay

The time it takes for the information to travel from being measured at node a to being received at node b is an important metric, because many fields in the beacon are of a delay-sensitive nature. In Fig. 3.4, the end-to-end delay or *latency* incurred by the information by the time it is used in node b is illustrated by the interval between generation and "consumption" on the horizontal axis. Only packets which are correctly received have an end-to-end delay. Several components constitute this delay, they are treated below and summarised in Table 3.2.

Sampling delay depends on the sample rate of the sensor. If this is for instance 5 Hz, the information is between 0 and 200 ms old when the beacon generator reads it to construct a beacon, assuming the generator and the sensor are not synchronised. This delay can be reduced by sampling at a higher rate than the beacon generation, illustrated with 25 Hz in Fig. 3.4, or by applying filtering.

This factor is often ignored in analysis under the assumption that locally acquired sensor information has zero delay the moment it enters the MAC queue. Often instantaneous availabililty is assumed, as is also the case for all simulation and stochastic analyses performed in this dissertation. However, when chaining different systems to each other, the sampling delay can become significant. This is illustrated in Fig. 3.4, but the impact is treated in more detail in Sec. 4.2.4, based on measurements performed on a proof of concept implementation.

Processing delay is the sum of the several small processing steps in both the sender and the receiver. Many of these steps are present in the technology, such as `aMACProcessingDelay` in IEEE 802.11 as discussed in Sec. 2.4.3. It is common in analysis to assume this is negligible. The platform should be sufficiently provisioned to process the received stream of data, i.e., it should be space scalable as described in Sec. 3.3.2. This becomes even more vital when extensive processing is performed on the data in the cooperative awareness, such as running estimators for each neighbour vehicle or delivering certain data to other components in a publish-subscribe fashion.

Queueing delay is the time a frame spends in the transmission queue, waiting for the service of one or more previous frames to complete. Ideally, a frame should be transmitted as soon as it has been generated. Sample 3 in Fig. 3.4 undergoes long queueing delay.

Contention delay is accumulated by the frame currently in contention while the station is contending for medium access and counting down until its bc reaches zero. In IEEE 802.11 systems, contention delay depends on the MAC parameters such as the CW size, and the load on the network. With a larger CW, the average bc drawn from it will be larger. With increasing load on the channel, it becomes more probable a node has to defer channel access. This is illustrated for sample 5 in Fig. 3.4.

Propagation delay is incurred by the fact that radio waves travel at the speed of light. Even though this is a phenomenal velocity, depending on the distance seperating the transmitter and receiver this can be as much as 4 μs in IEEE 802.11p. This value is based on the upper-bound transmission range of approximately 1200m. This is represented in Fig. 3.4 by the slight slope of the line going from TX to CA$_b$.

On a per-packet basis this delay can often be considered negligible, however through the `aAirPropagationDelay`, for instance, this has influence on MAC parameters such as the slot time treated in Sec. 2.4.3, and hence affects system-level performance.

Transmission delay is the time needed to transmit the frame in its entirety. This is illustrated in Fig. 3.4 by the gray areas next to the line going from TX to CA$_b$. As the data rate becomes higher, the transmission delay will diminish (because the duration of a frame is reduced). When beacons become larger they will take longer to transmit, increasing transmission delay. With fixed data rate and beacon size, the transmission delay is constant.

Evaluation delay is the time between the moment information is received in CA$_b$ and the moment it is used by the application. In Fig. 3.4, the application evaluates the information in CA$_b$ at rate λ_e. This type of delay is similar to the sampling delay discussed above. Similar to sampling delay, the evaluation delay is often ignored

Component	**Type**	**Relation to load / n**
End-to-end	Varying	✓
Sampling	Varying, tight bounded	independent
Processing[6]	Constant, negligible	independent
Queueing	Varying	✓
Contention	Varying	✓
Propagation	Varying, tight bounded, negligible	independent
Transmission	Constant	independent
Evaluation	Varying, tight bounded	independent

Table 3.2: Summary of delay components.

in the evaluation of the communication, however, a λ_e of 25 Hz could introduce considerable delay compared to the other delay compents.

3.5.3 Other metrics

In addition to loss and latency, there are other metrics which can be used to evaluate the performance of a beaconing system. These are treated below and are summarised in Table 3.3. In addition to these network-level metrics, we also use the application-level metrics *velocity* and *acceleration* of a vehicle in Sec. 4.5.

Channel load μ is used as an indication of the utilisation of available resources. It is characterised by three fundamental components: the duration of a transmission T_{signal}, the generation rate λ_g, and the number of nodes n, under the following relation:

$$\mu = n\lambda_g T_{\text{signal}} \tag{3.2}$$

The number of nodes directly influences the load on the channel, but also the number of "competitors" that the MAC has to coordinate with. A larger number of nodes results in a larger probability of nodes having their backoff timer expire simultaneously, thus increasing the collision probability.

T_{signal} is determined by the size of a message in bytes, and the data rate at which it is transmitted:

$$T_{\text{signal}} = LR \tag{3.3}$$

with L the length of a signal in bits and R the data rate in bits per second. In the best case, the channel can only hold a certain number of messages without overlapping.

[6] Only considering PHY/MAC processing. When processing in the cooperative awareness is considered, this could be varying and non-negligible, and strongly related to n.

Metric	Symbol
Reception rate	λ_r
Throughput	X
Inter-arrival time	τ_r
Channel load	μ

Table 3.3: Other metrics and their symbols as used in this dissertation.

With an increasing data rate, more messages fit on the channel. The duration of a transmission also affects the period in which it is vulnerable to interference. A short frame has a smaller probability of encountering a burst of interference than a large frame.

Reception rate λ_r is the number of beacons per second node i receives from node j and is a measure of throughput. A vehicle receives beacons at the rate they are generated, multiplied by the probability that a beacon will be successfully received:

$$\lambda_r = \lambda_g P_s. \tag{3.4}$$

Throughput X is the total number of beacons received per second by node i. Since there are no retransmissions, the throughput corresponds to the *goodput*, which is the number of frames received at the application layer. Although beacon generation is periodic, beacon reception rate may differ in mean and distribution. X is treated in Sec. 5.5.2.

Inter-reception time τ_r is the time between two successional successful beacon receptions at vehicle i from vehicle j. This is the time the CACC controller operates on "stale" information (which could be alleviated but not fully mitigated by using filtering) before receiving "fresh" information. Performing extrapolations (such as dead-reckoning for a node's position) in a dynamic environment incurs errors which grow over time, reducing the estimation's confidence level. The arrival of fresh information can reset the estimation and increase confidence, illustrated by the vertical arrows from CA$_b$ to the application line in Fig. 3.4.

Note that if the system were deterministic and contained no loss, τ_r would equal $\frac{1}{\lambda_g}$, or the inter-generation time τ_g. However, delay and loss—successive loss in particular—can greatly increase τ_r. The mean τ_r is obtained as:

$$\tau_r = \frac{1}{\lambda_r}. \tag{3.5}$$

3.6 Borders of the beaconing solution space

Limitations of the trained human driver lie primarily in the range of vision, attention to, and understanding of the situation on the road, and swiftness of reaction. A vehicular network supporting CACC has similar but different limitations.

In traditional 802.11 networks, the Broadcast communication method is only used for management frames, such as those used for association. The broadcast method does not come with the mechanisms designed to increase robustness as applied in Unicast as described extensively in Sec. 2.4. As such, Broadcast is 802.11's most fragile method of communication. But owing to its one-to-many nature, Broadcast is also the most appropriate for the task of beaconing.

In order to explore the beaconing solution space, we derive a channel utilisation or medium busy-time model, based on n, λ_g, and T_{signal}. In this analysis, we assume 100% efficiency of the Medium Access Controller.

3.6.1 Model assumptions

This section specifies all assumptions that have been made for the models described in the remainder of this chapter. Unless specified differently, all assumptions have also been made in [108] and [109], research activities that also explore the boundaries of the beacon solution space.

Topology We assume a straight road with l lanes. All vehicles are similar and are spaced equidistantly on the road to achieve a certain density ρ in vehicles per kilometer per lane. We abstract from mobility under the assumption that every instance of communication—the transmission of a single beacon message—happens on a much smaller timescale than mobility and traffic can henceforth be considered static. This is illustrated in Fig. 3.5.

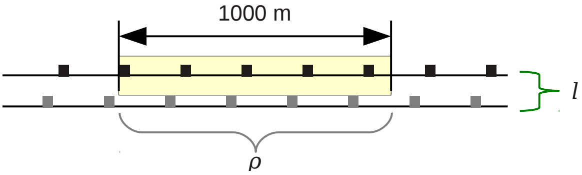


Figure 3.5: Node topology considered in the experiments.

Communication All vehicles can communicate using an IEEE 802.11p MAC and PHY. For simplicity, we assume a unit-disc propagation model: there is no signal attenuation up to the Communication Range (CR), beyond which there is 100% attenuation. The Carrier Sense (CS) range coincides with the CR.

The analysis is limited to data rates of 3 and 6 Mbps, because these modulation schemes are the most robust to interference and noise [110]. We consider beaconing

Component	Parameter	values
n	l	4 lanes
	ρ	1 – 100 veh/km/l
λ_g	1 – 40 Hz	only deterministic arrivals
T_{signal}	T_h	32 + 8 = 40μs
	L	100,200,400,1000,1400 bytes
	R	3 Mbps, 6 Mbps

Table 3.4: Model parameters and values [81].

in isolation and assume there is no other traffic or interference. We do not consider periodic channel switching as proposed for WAVE [69] (described in Sec. 2.3.4). A node *always* listens to the medium in order to maximise its cooperative awareness.

3.6.2 Modelling channel capacity boundaries

The normalised channel load μ in (3.2) can be used to obtain an upperbound for the maximum load supported by the medium. Under the assumption that channel utilisation can be 100%, we can find combinations of possible values by solving (3.2) for $\mu = 1$:

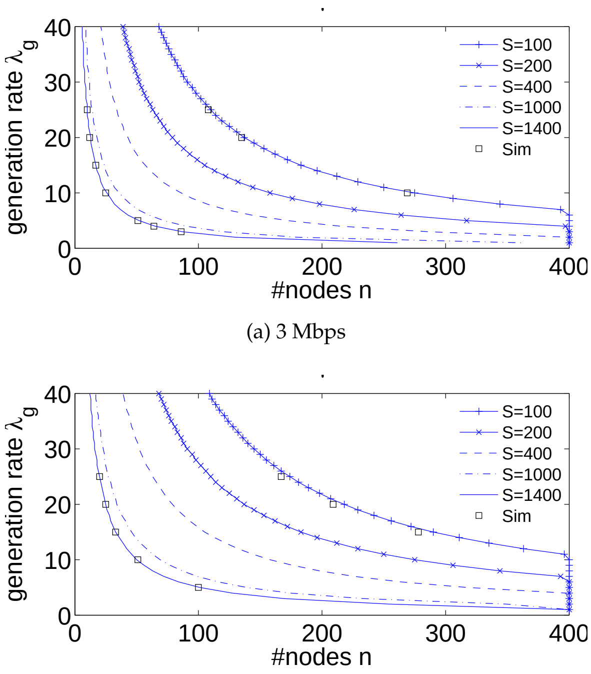
$$\lambda_g = \frac{1}{n\left(T_h + \frac{L}{R}\right)}, \tag{3.6}$$

The time T_{signal} a beacon message occupies the channel equals the duration of the header (T_h) plus the time it takes to transmit a beacon of length L in bits at bitrate R, in bits per second. Under ideal channel conditions and ideal efficiency of allocating resources, the channel load can go up to 1 without collisions, i.e., $P_s = 1$. The number of nodes n is defined by the number of nodes within the CS range. As such, ideally every node observes the channel in the same state. The number of nodes is obtained as follows:

$$n = 2CS \cdot \rho \cdot l \tag{3.7}$$

The CS range is assumed to be omnidirectional (hence the factor 2 for the straight road). Influence of the message duration T_{signal} on μ is twofold:

beacon size - By efficiently coding the beacon, the packet can be kept small, leaving more capacity for other nodes. Frame size also plays an important role in BER due to the reciprocal dependencies between previous and subsequent symbols. It is imperative that a future implementation codes beacons as efficiently as possible to minimise L.



(a) 3 Mbps

(b) 6 Mbps

Figure 3.6: Channel Capacity Boundaries.

data rate - As R increases, T_{signal} decreases. It seems tempting to increase the data rate but a larger R may have a negative impact on the probability of correct reception, as transmissions at higher data rates are more prone to interference and have a shorter effective range.

The channel utilisation model was verified using the OMNeT++/MiXiM simulator. Modifications were performed to meet the assumptions made for the model; contention is eliminated, there are no hidden nodes and transmissions are perfectly aligned in time (like a slotted Time-Division Multiple Access (TDMA) scheme). The simulator is described in more detail in Sec 2.7. In order to provide a numerical solution, values from Table 3.4 are used. These values were introduced in previous sections or are realistic assumptions.

The results of the simulation experiments are depicted by the squares in Fig. 3.6. These simulation experiments show that the values which are derived analytically coincide with the simulated behavior. This means that if the assumptions described above are considered, then (3.6) is a good approximation.

Fig. 3.6 shows a numerical solution to (3.6) for several values of S for data rates of both 3 and 6 Mbps. The plots show the channel utilisation boundaries for beacon sizes stated in Table 3.4, plotted in the same top-to-bottom order as they appear in the legend. It should be stressed that this figure shows the possible generation rates at maximum channel utilisation, assuming no collisions. The actual achievable throughput in terms of useful data will be considerably lower.

This deserves some clarification. The maximum useful channel utilisation—that is, time spent successfully transmitting packets—of Pure ALOHA, for instance, is 18%. Slotted ALOHA increases this to about 37%, whereas CSMA/CA Unicast used in IEEE 802.11 has a typical maximum channel utilisation of 54–66% in the absence of hidden terminals [111]. The exact useful channel utilisaton of IEEE 802.11p on a broadcast channel will be treated in Chapters 5 and 6 in this dissertation.

Observing Figs. 3.6 (a) and 3.6 (b), it is straight-forward to conclude that a beaconing system consist of either many nodes transmitting at a low rate, or few nodes transmitting at a high rate. In addition, the solution space becomes larger as the beacon size decreases. Using a data rate of 6 Mbps, as plotted in Fig. 3.6 (b), dramatically increases the solution space. The result is that larger n or λ_g can be achieved with the same S. This can be explained from (3.2); T_{signal} will significantly decrease. However, P_s will generally be lower at a higher data rate [110], so the greater capacity available when beaconing at 6 Mbps or higher data rates may be thwarted by poor signal propagation and reception probability, factors not taken into account by this model.

3.6.3 Modelling beacon reception probability

To model the probability of a successful beacon reception we use the probabilistic model presented by Yang *et al.* in [109]. This is a simplified model of the 802.11 DCF access method. This model is somewhat similar to Slotted ALOHA in the sense that time is divided into transmission slots, where the duration of one slot is equal to the time it takes to transmit a beacon, T_{signal}. The average time between generation of two beacons by a single node is defined as the reciprocal of the beacon generation rate: $\tau_g = \frac{1}{\lambda_g}$.

The model takes into account the probability that two nodes which are within each other's CS range may start transmitting at the same time and the probability that a hidden node may interfere with an ongoing transmission. The number of hidden nodes is set to the worst case scenario of $n/2$, as shown in Fig. 3.7. Here $n/2$ nodes are hidden terminals for the transmission of a message from transmitter j to receiver i. Considering that the relative distance between a transmitter and a receiver is not taken into account, effects like the capture effect [112] and signal propagation are not considered in this model.

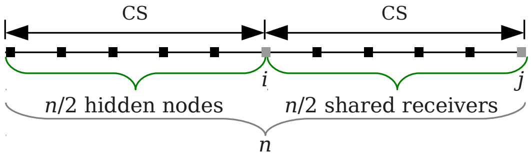


Figure 3.7: Worst-case hidden terminals on a straight road.

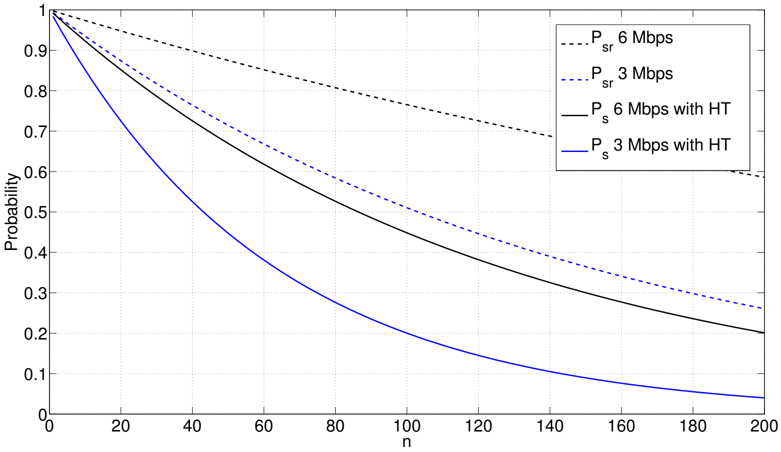


Figure 3.8: Probability of collision-free transmission according to (3.10) and (3.8).

Recall that P_s is the probability that vehicle i (in Fig. 3.7) successfully receives a beacon from vehicle j. Assuming synchronisation and abstracting from CSMA/CA's collision avoidance, the probability that in a given slot a node attempts a transmission can be expressed as T_{signal}/τ_g. Hence, the probability that $n/2$ nodes (the shared receivers) will *not* start a transmission at the same time in that time slot, P_{SR}, is then given as:

$$P_{SR} = \left(1 - \frac{T_{\text{signal}}}{\tau_g}\right)^{\frac{n}{2}}. \tag{3.8}$$

Assuming independence between the shared receivers and the hidden terminals, a transmission by j may experience interference from transmissions by hidden nodes. These transmissions can start during the time that j transmits, but also in the slot prior to j's transmission. As a result, the vulnerability period of a beacon is twice the beacon transmission time, making P_{HN}:

$$P_{HN} = \left(\left(1 - \frac{T_{\text{signal}}}{\tau_g}\right)^{\frac{n}{2}}\right)^2 = \left(1 - \frac{T_{\text{signal}}}{\tau_g}\right)^{n}. \tag{3.9}$$

Then, P_s is expressed as the probability that (given an idle medium and $n/2$ shared receivers) only vehicle j starts a new transmission (P_{SR}), multiplied by the probability that no hidden nodes will interfere with this transmission (P_{HN}):

$$P_s = \left(1 - \frac{T_{\text{signal}}}{\tau_g}\right)^{\frac{3n}{2}}. \tag{3.10}$$

To explore the impact of n on the successful reception probability. λ_g is fixed at 10 Hz, and S at 400 bytes, conform the C&D requirements presented in Sec. 3.4. Fig. 3.8 plots P_s (according to (3.10)) for varying number of nodes for both 3 and 6 Mbps (the two dashed lines, top line for 6 Mbps), other parameters are as in Table 3.4. Note that (3.10) assumes worst-case hidden terminals. The figure also plots P_{SR} (according to (3.8)) for the situation with no hidden terminals.

3.6.4 Verification of system requirements

We can now map the CACC requirements presented in Sec. 3.4 to the derived beaconing solution space. Consider beacons of 400 bytes, transmitted at 3 Mbps and λ_g=25 Hz, the generation rate initially pursued in the C&D project. If we solve (3.6) for these parameters we find $n \approx 34$ as an upperbound for the number of vehicles.

The C&D requirements [23] state that a communication range of 200 m or 15 vehicles per lane should be considered. Now, consider a highway with four lanes, where $\rho = 75$ (dense traffic). In this scenario, a vehicle has 15 vehicles in the same lane up to 200 m ahead, but the medium has to be shared by 120 vehicles because of the omnidirectionality of the antenna and the four lanes. Clearly, $\lambda_g = 25$ Hz cannot be maintained with $n = 120$ (also evident in Fig. 3.6 (a)) and should, according to (3.6), be lowered to $\lambda_g \approx 7$ Hz in this case. Based on this analysis, λ_g was relaxed to 10 Hz in the CACC prototype, yielding an upper bound of 85 vehicles.

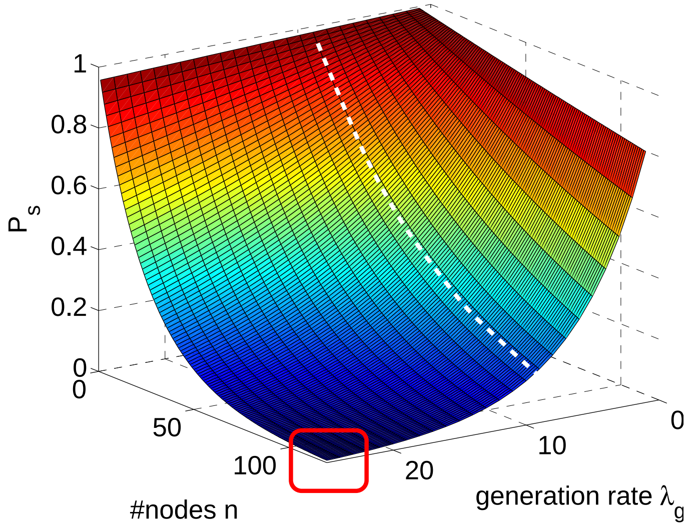


Figure 3.9: CACC Beaconing Solution Space.

Fig. 3.9 shows P_s as defined in (3.10), with n and λ_g varied. This figure is an indicator for the system's load scalability, as described in Sec. 3.3. This figure supports the conclusion that 120 vehicles beaconing at 25 Hz will result in very low P_s, the area indicated by the red box. In turn, this would result in very poor CACC string stability performance. In fact, for λ_g=10 Hz, indicated by the white dashed line in Fig. 3.9, there still is a small interval of n where P_s is high–the system cannot accomodate the growth of n expected under practical operation *and* have high P_s. It is important to find ways to ensure P_s remains acceptable as traffic density increases. Here we encounter a trade-off also identified in [109]: the number of vehicles we want to collect information about, and the temporal resolution of this information. This is reflected in n and the resulting λ_r. Whether the beaconing system must optimize for high n or high λ_r depends on the application. In case of CACC in its simplest implementation (one-vehicle look-ahead, discussed in Sec. 4.2), there is only a single relevant vehicle: the one in front. A sufficiently high update rate is required from this vehicle.

3.7 Conclusions & outlook

After defining beaconing, the powerful concept of cooperative awareness was described. The cooperative awareness is at the heart of ITS applications in the ad hoc domain. This is the central point where applications such as CACC draw their information from. The performance of these applications depends greatly on the quality of this information, and hence on the performance of the underlying beaconing system which supplies and refreshes the information. Next, the issue of scalability in this context was elaborated upon, because the quality of the cooperative awareness depends on the scalability of the beaconing system. The conclusion is that a beaconing system is primarily limited by its load scalability and by its structural scalability. Load scalability is limited by the scheduling of transmissions, resulting in a collision probability which increases with the number of nodes, and by the scheduling of beacons within a node. Structural scalability aspects can limit overall scalability by limiting design choices or standards. The remainder of this dissertation will revisit these factors, and quantify their impact on performance.

The communication requirements of CACC were presented and discussed. Applications should be able to degrade gracefully when requirements are no longer satisfied. The CACC application does so by falling back to ACC mode, as will be described in more detail in Chapter 4.

The performance metrics used throughout this work were described. An overview of the delay and loss metrics was provided. The remainder of this dissertation provides additional detail of these components.

A simple numerical analysis evaluates the performance of a beaconing system in terms of the number of nodes n which can be supported and the probability of successful communication P_s. Next, the requirements were verified against these findings. CACC is designed to outperform a human driver in dense traffic, hence the beaconing system should also remain operable under these circumstances.

Given the load generated by a beaconing system and the fact that the channel rapidly saturates with beacon frames under increasing traffic density, we emphasise the need for a dedicated beaconing channel instead of regarding beaconing as ”background traffic”. We reason that a single cross-layer beaconing system in a vehicular network could fulfill requirements of beaconing at all levels in the protocol stack, eliminating the need for many other, application-specific communications. These applications can, in turn, draw their information from the cooperative awareness present in each node.

This chapter provided basic insight in the beaconing solution space through simple analytical relations. The following chapters focus on gaining more insight in the performance of beaconing through three distinct approaches: by means of proof of concept implementations in Chapter 4, simulation experiments throughout the remainder of this dissertation, and more involved analytical modelling in Chapter 5.

CHAPTER 4

Application-level aspects of CACC

In Chapter 3 we obtained a coarse notion of the performance of a beaconing system using simple analytical relations, relying on restrictive assumptions. One method to find out if the performance of a beaconing system is sufficient, is to develop a proof-of-concept implementation and see what problems are encountered along the way. This is an explorative endeavor; not to create a product but to find out how to build it—or to find ways how *not* to build it.

In addition to proving technical feasibility, a proof-of-concept implementation also provides an opportunity to do measurements on a real-world system. The outcome of this can be used to validate models used for simulation or analytical evaluation of beacon message exchange in vehicular networks. Furthermore, there are also valuable lessons which can be learned by stepping out of the sandbox provided by the assumptions in both simulation and analytical modelling. There are factors which are not considered there, or are assumed to have no influence in the more abstract evaluation methods.

To begin with, the analytical relations formulated in Chapter 3 are applied in Sec. 4.1 to derive an expectation on the performance of the two proof-of-concept implementations. Next, Connect&Drive and the Grand Cooperative Driving Challenge are discussed in Sec. 4.2 and Sec. 4.3. Alongside the description of these two projects, relevant aspects which influenced their performance are discussed, culminating in a discussion on lessons learned in Sec. 4.4.

The performance of the wireless networking has significant influence on CACC performance, but is difficult to study in these two implementations. The sensitivity of the CACC application to network performance is evaluated in a simulation study in Sec. 4.5. This yiels an answer to research question 4 (“How does IEEE 802.11p scale under varying traffic densities, and can it meet the requirements imposed by CACC?”).

This chapter contains material previously published in [10, 14, 21, 23, 24, 25, 26].

4.1 Expected performance

Using the simple modelling approach presented in Chapter 3, a prognosis of the expected beaconing performance in both the C&D and the GCDC prototypes can be derived. Beacons have a size of 400 bytes and 20 bytes MAC header, resulting in

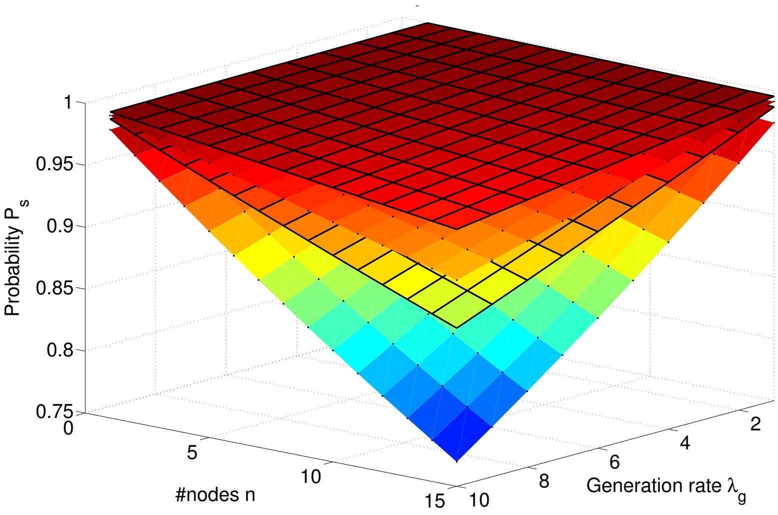


Figure 4.1: Expected probability of successful reception for 3 Mbps (with dots) and 6 Mbps (with mesh). Top surface is P_{SR} in (3.8), bottom is P_s in Eq. (3.10).

$T_s \approx 1336\mu s$ for 3 Mbps. Likewise we find T_s =802 μs for 6 Mbps. Furthermore, we use λ_g = 10 Hz. To get a notion of P_s, the numerical results of (3.10) introduced in the previous chapter are plotted for these parameters in Fig. 4.1. The 3 Mbps results are plotted with dots on the surface and the 6 Mbps results are plotted with a mesh. The top surface is without hidden terminals (best-case) and the bottom surface is with $n/2$ hidden terminals (assumed to be a worst case).

This analysis shows that, assuming ideal channel conditions, the expected P_s for the seven C&D vehicles is between 0.9409 and 0.9799 whereas the P_s for the 10 GCDC vehicles is between 0.9167 and 0.9714. Hence, both proof-of-concept implementations will not run into IEEE 802.11's scalability problems. P_s is sufficiently high and λ_r will only be marginally smaller than λ_g for the fixed generation rate of 10 Hz used in both projects.

Using Eq. (3.2), we can compute the expected channel utilisation. The results are plotted in Fig. 4.2 for both 3 and 6 Mbps. The surface on top is the utilisation of 3 Mbps beaconing, because messages are roughly doubled in duration (the duration of the header remains constant). These results confirm that scalability will not be an issue in systems with such a small number of nodes as used in C&D and GCDC. Channel Utilisation will be approximately 0.0595 and 0.08472 for the 7 and 10 test vehicles respectively. As a result, stations often find the medium idle and will perform a direct transmission. In fact, the channel utilisation will be so low that CSMA/CA's collision avoidance mechanism is hardly used. Because of this, delay due to queueing and contention can be expected to be negligible. The CACC performance will not be limited by the wireless communication.

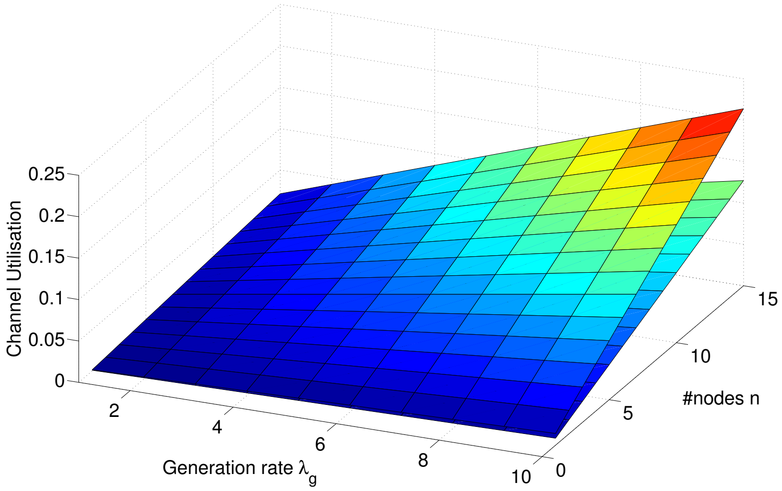


Figure 4.2: Expected channel utilisation for 3 Mbps (top surface) and 6 Mbps (bottom surface).

4.2 Connect&Drive

The Connect&Drive project was briefly introduced in Sec. 2.2.4. Here we will elaborate on the technical details of the project.

4.2.1 Control structures

It is possible to implement longitudinal vehicle control in various ways. To begin with, it can be centralised or distributed. From the outset of the project, it was reasoned that a distributed, ad hoc approach would have the best chance of success. However, within the distributed approach there are still various ways of implementation. For that reason, several information flows and control structures are considered in C&D. The radar information of the direct preceding vehicle is augmented with communicated data, see Fig. 4.4 (in which vehicles move from left to right). The control modes are defined as follows:

1. **Direct preceding vehicle** – in addition to tracking information derived from radar measurements, the vehicle uses the communicated information from the vehicle directly in front (one-vehicle look-ahead).

2. **N preceding vehicles** – the controller can take multiple vehicles into account (N-vehicle look-ahead). A benefit is that the system could anticipate in a way similar to a human driver seeing brake lights a few vehicles ahead.

3. **Preceding and following vehicle (bi-directional)** – the controller also takes into account vehicles behind it. This mode may be beneficial for heavy-duty

Figure 4.3: Connect&Drive Prius vehicles gracefully decelerate [1].

trucks to reduce aerodynamic drag by optimising the inter-vehicle distance.

4. **Leader vehicle only** – a certain vehicle is appointed to be the "platoon leader", the other vehicles will coordinate their motion with this vehicle.
5. **Preceding and leader vehicle** – combination of 1. and 4.
6. **Mini-platoon** – several clusters of vehicles driving in mode 4. or 5. are able to

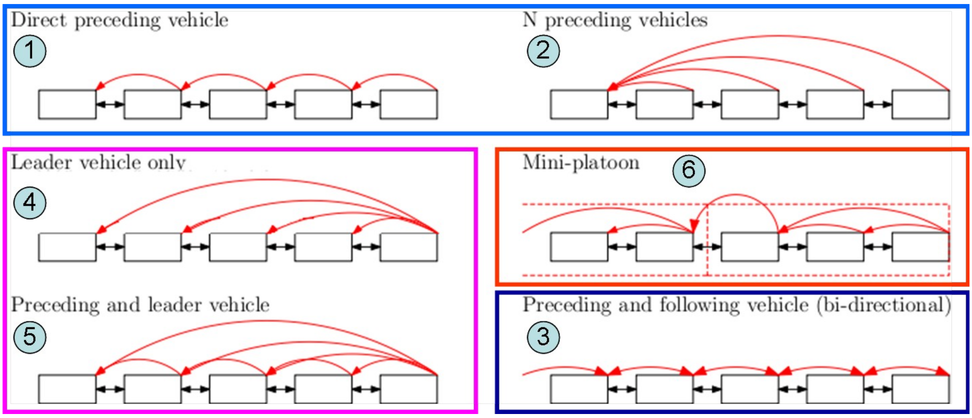


Figure 4: Miscellaneous, information flow structures for some potential *configuration modes*

Figure 4.4: C&D information flow structures, copied from [23].

coordinate their longitudinal behaviour.

These six structures include the two "schools" of platooning: *with* a leader vehicle (cases 4, 5) and *without* leader or ad hoc (1, 2, 3). Control structure (6) is a hybrid structure combining leader-based and ad hoc platooning. The leader-based structures focus on forming platoons behind an elected or configured leader vehicle; a scenario which may be relevant for a string of heavy-duty trucks. The ad hoc approach was deemed more flexible—and more capable in traffic flows dominated by individual vehicles—because explicit platoon management is absent. The externally observed behaviour is equivalent: vehicles drive in a close string, showing efficient following behaviour.

For the proof-of-concept implementation, only the control structure in scenario 1 was implemented. The one-vehicle-look-ahead topology is the simplest possible structure, and therefore it has the highest probability of being deployed. It is expected that scenario 2 can yield significant benefits in terms of string stability (as introduced in Sec. 2.2.1) and comfort. Additionally, the N preceeding vehicle mode may allow a certain degree of non-communicating vehicles.

Fig. 4.5 shows a functional decomposition of CACC as implemented in C&D. This figure relates input from the driver, surrounding vehicles, and infrastructure to the system in the host vehicle. The vehicles in the cooperative awareness are called object vehicles. The CACC block takes input from the vehicle itself ("ego motion data", "radar data") and from the object vehicles. It accepts configuration from the platoon control entity, which takes information from the object vehicles, from the human driver, and possibly from infrastructure. An example of input from the infrastructure could be an RSU transmitting a dynamic speed limit due to

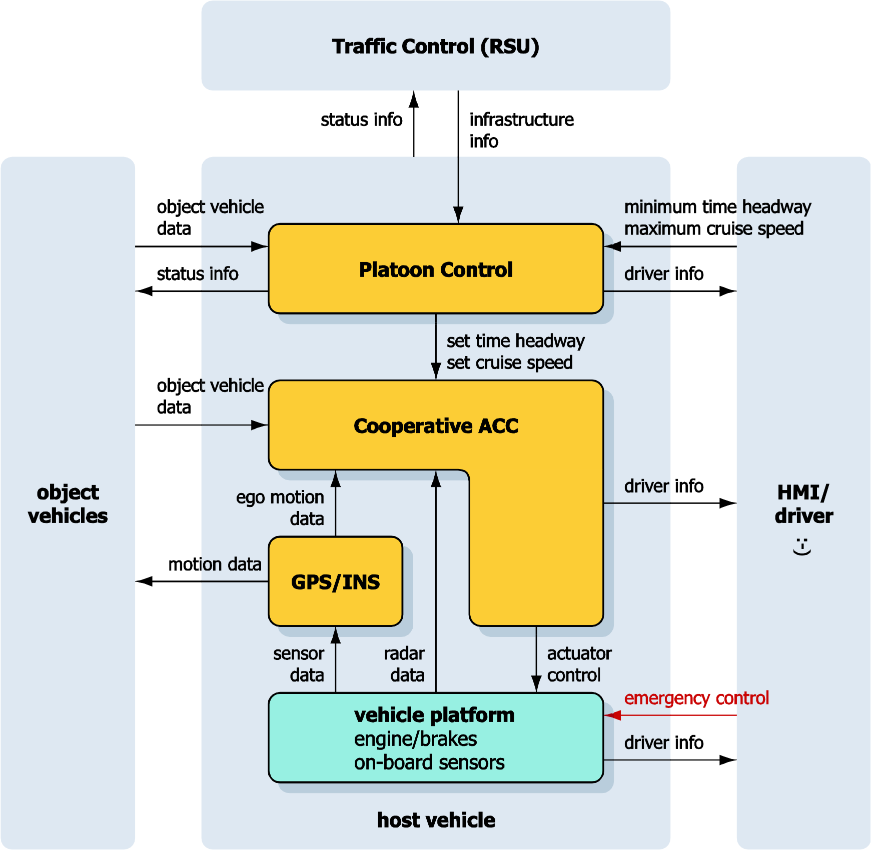


Figure 4.5: Functional decomposition of the (C)ACC system used in C&D [23].

bad weather. These three inputs allow the platoon control entity to derive a time headway and a desired cruise speed. The information exchange through beacon messages is present in the arrows between the host vehicle and object vehicles.

Note that this architecture puts the driver still in a position to take over full control by means of emergency control and, in a more moderate fashion, by supplying time headway and cruise speed settings through the HMI. The system continuously shows the driver what it is doing by means of the “driver info” information flows.

The system was implemented in several components, depicted in Fig. 4.6 in an arrangement similar to their functionality shown in Fig. 4.5. These components are:

1. The low-level control unit (developed in the MOVE project), visible central in Fig. 4.6. This is the interface between the vehicle and the CACC platform [43]. This box acts as a permanent “man-in-the-middle attack” between the stock radar and the engine management. It filters and injects messages on

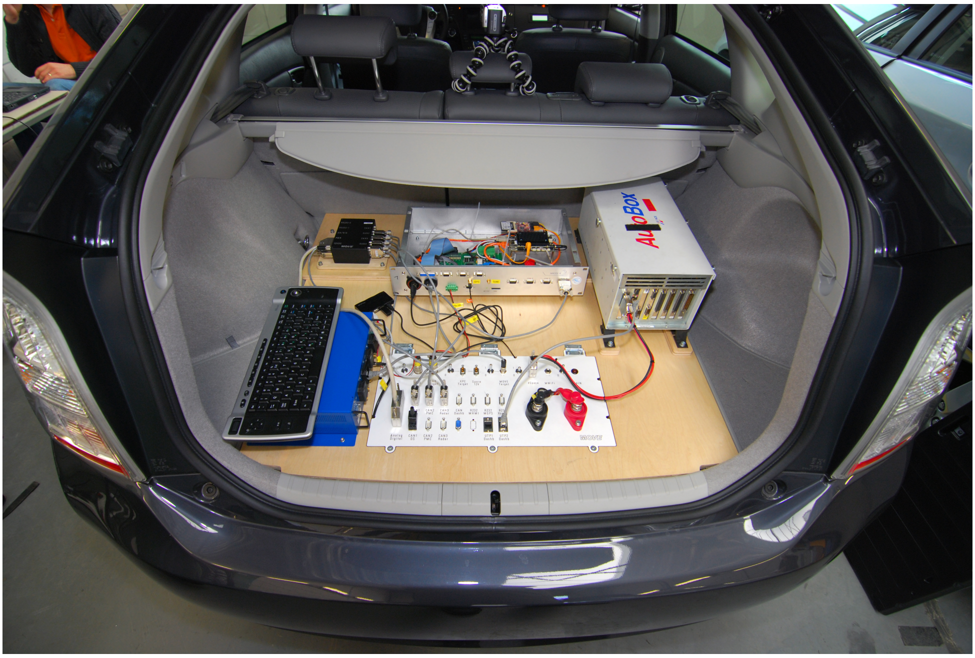


Figure 4.6: Instrumentation of the C&D vehicles [1].

the vehicle's CAN bus in order to control the vehicle, visible as the "vehicle platform" box in Fig. 4.5.

2. The high-level control platform (developed in the C&D project) visible at the top of Fig. 4.6. This component contains the three yellow functionality blocks depicted in the center of Fig. 4.5.

3. The FIGO box by Twente Institute for Wireless and Mobile Communications (Ti-WMC), the blue box visible on the left in Fig. 4.6. This component functions as the wireless gateway or OBU, and is responsible for interactions between the high-level control platform and object vehicles.

The AutoBox on the right in Fig. 4.6 was only present in vehicle 1 for the logging of low-level control signals.

4.2.2 Networking

When the project began in 2009, the 802.11p standard was still in draft phase and no hardware was commercially available. The wireless networking hardware choosen was Ubiquity SR5 WiFi cards (high-power IEEE 802.11a) tuned to channel 120 (5.6 GHz), with 30 dBm transmission power. The datarate was fixed at 6 Mbps. All communication uses the Broadcast method.

The network architecture was largely based on the work by the C2C-CC [60] and ISO CALM [5], taking into account the output of the SAFESPOT [63] and CVIS [61] projects. Information dissemination beyond one hop was considered but it was concluded that single hop V2V communication provides enough range to the cooperative awareness.

The lower part of the network layer is built on top of the layer 2 meshing algorithm Forwarding Layer for Meshing (FLAME) [113] because this was the framework present in the FIGO system. FLAME is a layer between the MAC and the network layer, IPv4 in this case. It creates a virtual Ethernet which could include nodes which are more than one hop away, eliminating the need for layer 3 (IP-based) routing such as OLSR and AODV. FLAME uses in-band signalling to maintain forwarding tables, adding an 18-byte header between the MAC header and Network PDU. The messages were encoded as EIVP messages (as discussed in Sec. 2.3.5) and these were encapsulated in User Datagram Protocol (UDP) broadcast packets to make use of the IPv4 stack available on the FIGO.

Fig. 4.7 shows the C&D Commmunication Gateway architecture, as implemented in the FIGO box. This design caters for a Global Positioning System (GPS) receiver to be connected to the wireless gateway, the output of which can be used internally (e.g., for Geo-addressing and timing synchronisation) or be provided to the control platform through a socket interface. The communication gateway provides a control interface to the control platform, e.g., to start and stop beaconing. The control platform delivers sensor readings in the "Data storage" component at a rate greater than the beaconing rate, 25 Hz in this implementation. The Beacon generator takes the data from this storage when constructing a beacon. Received beacons are passed up to the control platform.

The Network layer design contains the following components [107]:

- **The Channel Estimator** estimates the current load on the wireless channel. Exact implementation was not further explored. This subsystem provides load estimations to adaptive beaconing and congestion control algorithms [114, 109].

- **The Beacon Generator** decides when to gather data, create a beacon message and send it down to the 802.11 MAC layer.

- **The Receiver** accepts received messages from the 802.11 MAC and may perform some preliminary selection and filtering, and subsequently passes on received information to the CACC controller. The receiver is connected to the Forwarder in case of Network-layer routing and may also be connected to the Channel Estimator.

- **The Forwarder** forwards messages for multi-hop communication (if needed). This component is not further discussed here.

- **The Geo-addressing**, in conjunction with the Forwarder, could provide Georouting. It is not further discussed here.

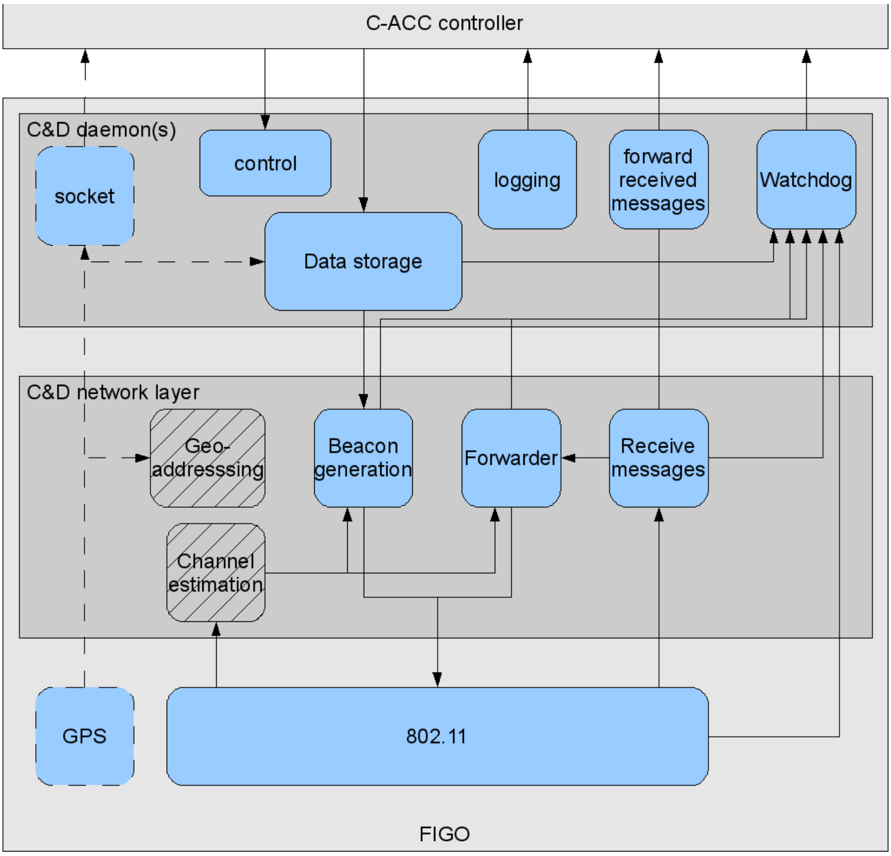


Figure 4.7: C&D Network layer architecture, copied from [107].

A design of an adaptive beaconing network layer is presented in Fig. 4.8. In this architecture, the Channel Estimator's primary goal is to determine the channel state (such as the medium busy time) by observing the MAC directly or by observing the received beacons. The state of the channel is useful for adaptively changing the beacon generator and the MAC configuration [10]. In the C&D network layer, the Channel Estimator is only used to gather information, it is not coupled to an adaptor, because adaptation is not considered. The channel information is collected as a logging feature. Because the Channel Estimator does not influence the generator, it may be turned off if processing overhead is too large. The main goal of the Channel Estimator in this implementation is to investigate whether the channel state can be derived, in practice, in a vehicle.

Messages are UDP packets containing an EIVP data payload. For communication from the high-level control platform to the wireless gateway, messages can be either EIVP or Control messages, in the reverse direction they can be EIVP messages or Log

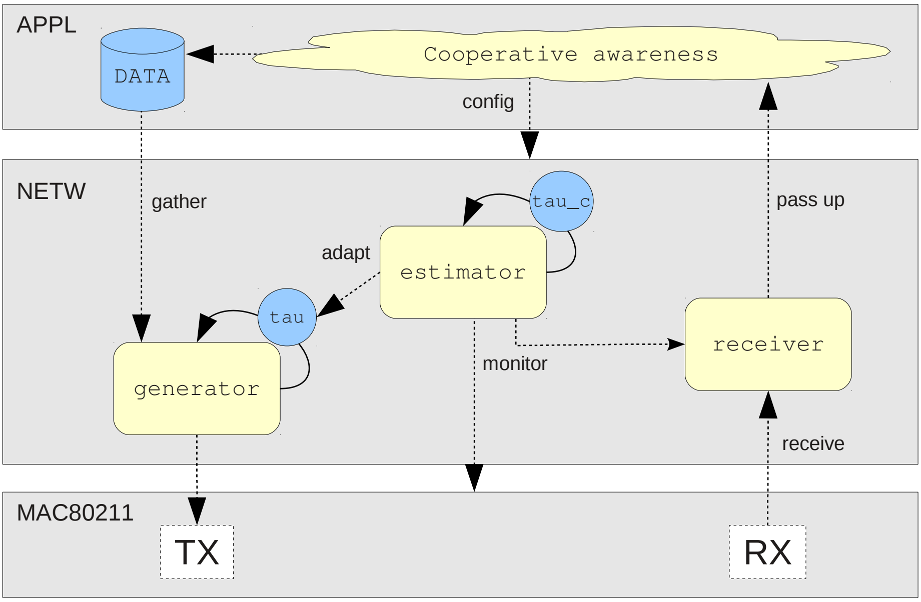


Figure 4.8: Adaptive beaconing network layer design, copied from [10].

messages. At the moment the C&D project ran, both the COMeSafety architecture and the EIVP message format were not yet finished. The EIVP was implemented to a format which resembles most closely the message format as intended by the EIVP designers [74].

4.2.3 System operation

The system was tested with seven Toyota Prius vehicles on the RDW test track near Lelystad, The Netherlands, in March 2011 during a three-day test. On the third day the system was presented to a selection of VIPs and video recordings were made.

A procedure for conducting the experiments was defined together with a list of items to log [115]. The logging functionality was implemented in every vehicle's high-level control platform. The experiments go through three phases: initialisation, run, and finish. The initialisation and finish provide snapshots of scalar values, whereas the run part is a time-series of values during system operation. Vehicles and experiments can be uniquely identified in their log files for offline analysis. Unfortunately, however, several factors prevented the logging of data.

For the visitors and from an overview perspective the demo was a success. The operation of CACC was successfully demonstrated and compared to ACC. Unfortunately, the component responsible for the logging of the data did not function properly during the demo. This bottleneck could be bypassed to continue with the demo, though a large part of the architecture was disabled in doing so. Part of

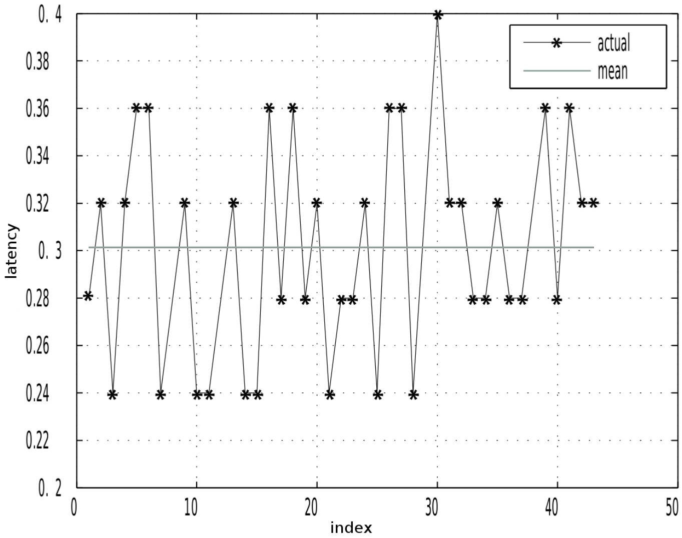


Figure 4.9: Measured two-way time delay (s) of the C&D platform, copied from [117].

the sacrificed functionality was the logging system. As a result, the demo did not generate any usable log files and measurements. As such, the initial research goals of study from a wireless communication perspective could not be realised. Furthermore, since the demo concluded the project there was no opportunity to further isolate and circumvent the bottlenecks and evaluate the performance of the CACC system as a whole or test sensitivity to changes in parameter settings. We were however able to make a movie showing the CACC system in action [116].

4.2.4 Delay measurements

Prior to the demonstration in March 2011, measurements between two vehicles were presented in an internal commissioning report [117]. It was found that a large variance in the delay was caused by the various sample rates of the components involved. In a simple experiment to estimate this delay, a 1 Hz sinusoid test signal was generated at a 25 Hz update rate. This is sampled at 10 Hz by the wireless system (because λ_g is 10 Hz) and then read by the control platform at 25 Hz from the receive buffer at the receiving node. During this test, the receiving control platform simply echoes the signal back: it is once more sampled at 10 Hz by the wireless system and finally logged by the C&D platform running, again, at 25 Hz.

Fig. 4.9 shows a snapshot of the measured delay of bouncing a signal back and forth from one C&D vehicle to another. Given the sampling intervals above, and no synchronisation between the components, a worst-case delay then becomes 2x100 ms + 2x40 ms = 280 ms, in addition to any delays caused by the MAC and other processing delay. This yields a mean delay around 300 ms as visible in the figure.

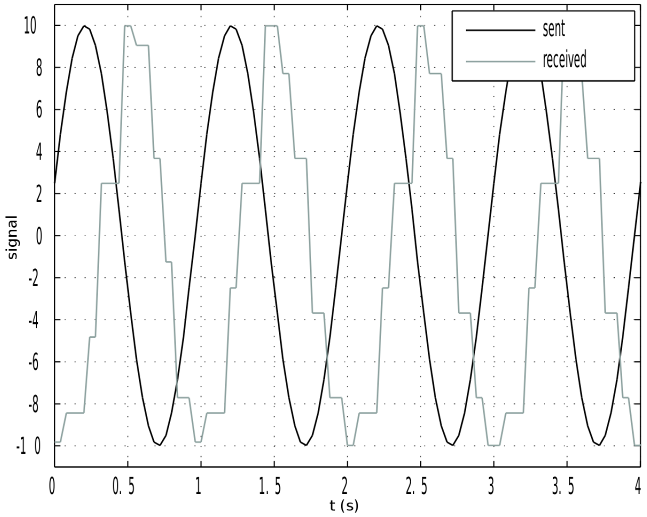


Figure 4.10: Time shift of a sine signal in the C&D system, copied from [117]

Observe that the signals are logged at 25 Hz, accounting for the 40 ms sampling steps of the delay visible in Fig. 4.9. This shows the large impact of the sampling delay on the total delay.

The resulting mean delay becomes visible when sending a sine wave and plotting it together with its returned version, which shows a time delay (see Fig 4.10). The plot shows a shift in time of about 300 ms, yielding a one-way delay of approximately 150 ms. Even though this is within the requirements presented in Sec. 3.4 and the system is able to operate appropriately, it is a significant delay considering that the estimated delay due to the communication (see Sec. 4.1) was marginal. In this case, a large portion of the delay budget is consumed by the sampling delay.

The internal update rate may have to be significantly larger due to requirements of some of the vehicle actuators [23]. This would also benefit the amount of sampling delay introduced into the total latency. Ideally, some means of synchronisation would need to be implemented among the control platform and the wireless gateway to further reduce the sampling delay.

In addition to the sampling delays introduced by the various components, the wireless gateway runs a layer-2 meshing algorithm (FLAME) between its MAC and Network layer. Since all transmissions were strictly broadcast intended only for one-hop neighbours, this could introduce some overhead in terms of processing delay and additional bytes for the in-band signalling. The FLAME algorithm is known to have a larger impact on delay than some other meshing algorithms when used in a multi-hop mesh network [118]. Compared to the sampling delay discussed above, however, the approximately 1 ms of additional delay found in [118] is negligible.

4.3 Grand Cooperative Driving Challenge

The participation of Team FUTURUM in the GCDC was briefly introduced in Sec. 2.2.4. This section elaborates on the details.

Figure 4.11: Team FUTURUM Smart cars.

4.3.1 A discussion on the two schools of platooning

The FUTURUM communication system is intended as pure V2V-approach. The reasoning behind this is that it is best to aim at an ad hoc solution, without the requirement of intermediate roadside infrastructure. First and foremost, applications like CACC are beneficial where large traffic densities accumulate. Given the fact that a certain percentage of drivers buys a new car every few years, a decent market penetration can be achieved in short time and will be in effect exactly there where it is needed: where vehicles accumulate. The alternative is to instrument all highways with roadside infrastructure, which is a process that will be both costly and lengthy, and then still the system only works on instrumented roads.

A second strong argument against primarily relying on roadside infrastructure is that vehicles are likely to move across regulatory domains. Though standardisation work is ongoing, harmonising every aspect, e.g., spectrum allocation, placement of roadside infrastructure etc., may not be straight-forward.

There exist two schools of platooning, being the *stateful* kind where a vehicle explicitly joins a platoon and the *stateless* kind, where a vehicle exhibits optimised following behaviour. The stateless kind was implemented in the C&D system, as discussed in Sec. 4.2. The GCDC Interaction Procotol [54] specifies a group of information elements concerning longitudinal platooning, detailing stateful platooning.

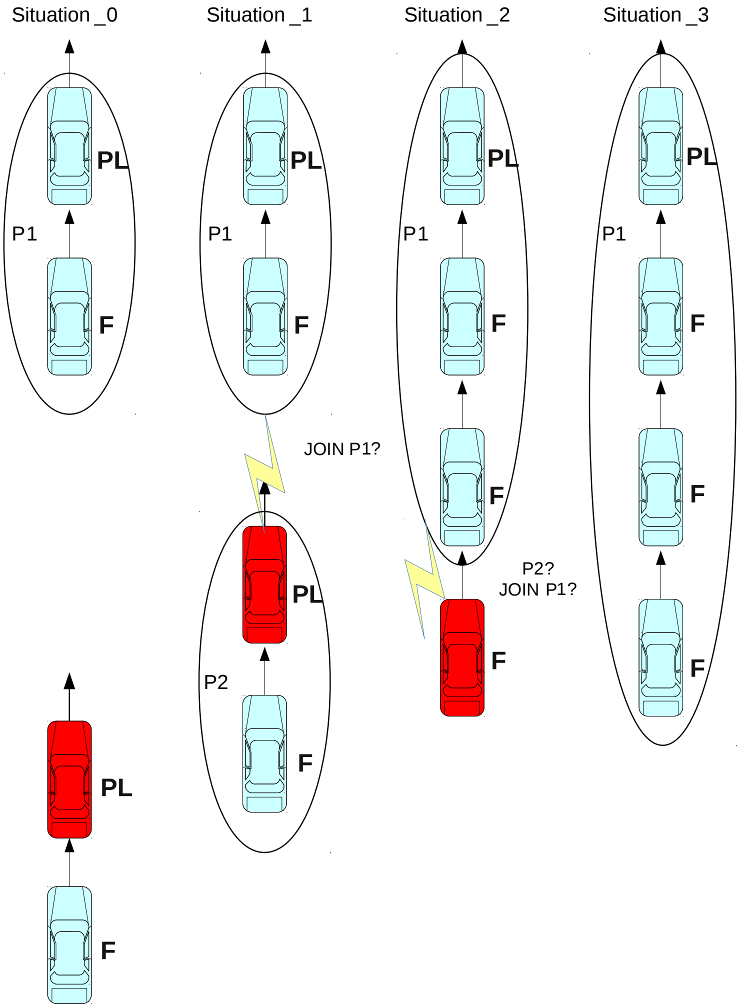


Figure 4.12: GCDC Platooning operations (longitudinal), copied from [54].

The rationale behind stateful platooning is that a vehicle is aware of the fact that it belongs to a platoon and, hence, can act accordingly. By definition of [54], a platoon is a string of vehicles which include the `platoonID` into their transmissions to show they are part of this platoon. Explicit interactions and messages are part of the protocol to facilitate vehicles to *join* or *split* a platoon. Joining and splitting are operations in the longitudinal domain. In the lateral domain, vehicles can *merge* into a platoon or *leave* a platoon, although this is not covered in this version of the GCDC Interaction Protocol.

Thus, by definition of the protocol, a platoon is a group of vehicles with the same platoon ID. A platoon has a *platoon leader*, which is the first vehicle in the platoon. Following vehicles adopt the leader's ID as their platoon ID. A single vehicle can be a platoon, by setting its own vehicle ID as its platoon ID.

The procedure of joining a platoon is illustrated in Fig. 4.12, where two platoons join to become a single, larger platoon. The states a vehicle goes through in this example, are illustrated in Fig. 4.13. Consider the the red vehicle in Fig. 4.12, which is leader of its own platoon (denoted by PL) in Situation_0. We will refer to this vehicle as the ego vehicle. If this platoon approaches a platoon in front, at some moment our ego vehicle will receive beacons from the vehicles in front. In Situation_1, the ego vehicle transmits a "requesting to join" message. It is now in the *pending* state, awaiting an acceptance message from the target platoon leader. As soon as the ego vehicle receives the acceptance, it will change state from platoon leader to follower as depicted in Situation_2. The follower of the ego vehicle receives beacons from its former platoon leader but now with the platoon ID of platoon 1. This vehicle, in turn, will then try to join platoon 1 as well.

In GCDC, only the joining of a platoon was part of the challenge. To this end, the beacon messages were extended with two fields: `platoonLeaderID` and `platoonState`. The PL acknowledges a request to join the platoon by means of a `PlatoonAction` message containing the ID of the follower, and a `platoonJoinGranted` field. The join request is guarded by a time-out window, so a vehicle does not indefinitely remain in the *pending* state.

The states in which a vehicle can be are depicted in Fig. 4.13. Here vehicles transition between being a platoon leader (Stable Leading) and follower (Stable Following). Besides maintaining some state, these operations do not significantly add anything to platooning performance. A platoon leader does not have a special role, other than accepting or rejecting join requests. There are some remarks concerning stateful platooning:

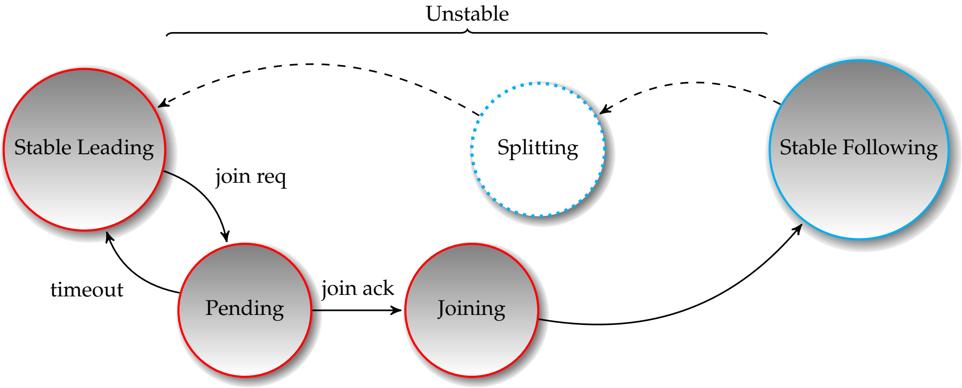


Figure 4.13: The GCDC Longitudinal Platooning states, based on [54].

- There seems to be no reason for vehicles to be part of a platoon and maintain this state, either from a traffic or a networking point of view. The vehicles can also be tracked individually in the cooperative awareness, the grouping in a platoon does not add new information.
- The additional interactions and payload add load to the communication channel. The increased use of resources does not seem justified if it has no function.
- Maintaining state in an ad hoc environment may be difficult. For a vehicle to join a platoon, its join request needs to reach the PL, and the PL's join acceptance needs to reach the vehicle. This symmetry may not always exist. Furthermore, this suggests the platoon length is limited by the communication range. The result could be that many vehicles reside in the "Unstable" part of the platoon state space illustrated in Fig. 4.13, because their PL moves in and out of range due to traffic movements.
- The presence of a platoon admission control system suggests there is a reason to involve the PL in the decision whether a vehicle is allowed to join a platoon or not; however it is not clear on what ground a PL could decide to reject or accept a join request.
- If the platoon is very long, why does a vehicle willing to join the platoon need to get acceptance from a vehicle driving several vehicles ahead? Both in networking and traffic context they are disjoint, up to the point of being completely isolated.
- The interaction of joining a platoon will cascade upstream (see Fig. 4.12). In this example two vehicles join P1, but if the second platoon would have a length of ten vehicles, ten join-accept interactions would be required. This cascading behaviour can be a serious threat to scalability.

In the end, the FUTURUM system was designed to participate in the platooning operations because they were mandatory. However, the increased overhead and the threat of cascading events in protocol interactions need to be carefully evaluated before future large-scale deployments, because they are self-expanding as described in Sec. 3.3 and impose a threat to scalability.

4.3.2 Networking

For the networking part, a communication stack was provided by the GCDC [119]. European FP6 project CVIS [61] had realised an open-source implementation of IEEE 802.11p based upon a Debian-Linux distribution, which was later released as a light-weight generic open-source implementation for Linux [120]. Safespot [63] defined message sets for the higher layers, but intellectual property rights prevented direct use of these. In order to arrive at a usable message set, an open interaction protocol [54] was defined, to which participants of the challenge also contributed.

All transmissions are Broadcast transmissions of the highest EDCA priority class. Each GCDC vehicle operates in the ITS CCH with 10 MHz bandwidth. The CCH is used continuously, there is no channel switching as defined in IEEE 1609.4. The communication uses CALM FAST, an implementation of which is provided by the GCDC organisation. The GCDC Interaction Protocol allows both V2V and V2I interactions, and the test road was equipped with many RSUs (also to monitor the challenge from the traffic control room).

The FUTURUM system is divided into two seperate physical devices; a real-time control platform (the *ControlBox*) and a general-purpose computing platform which handles the communication with the outside world (the *ComBox*). Not only does this two-box approach allow seperation-of-concerns in the design, it also allows two loosely coupled development tracks. The entities are defined as follows:

The ControlBox controls actuators and reads out sensors. It contains a cooperative awareness and CACC control logic.

The ComBox functions as an OBU and sends and receives GCDC messages (beacons) to and from other vehicles. The description of this device is covered below.

Fig. 4.14 shows the high-level design of the FUTURUM system, depicting the two components as two blocks. The ControlBox and the ComBox communicate through the FUTURUM Local Interaction Protocol (FLIP), see Appendix A. The ControlBox can issue commands and generate static and dynamic information packets based on sensor readings. The ComBox communicates with other participants through the IEEE 802.11p interface, on top of which the CALM FAST router [119] exchanges messages. The ControlBox supplies data into the DataStorage, from which the beacon generator draws data when building the next beacon (driven by a clock marked with “T” in Fig. 4.14). Received messages pass through a filter, which is a parameterised model of the real world, to discard information received from vehicles which are not of interest to the application. By doing this, the ComBox will only pass relevant information to the ControlBox. Performing filtering as early as possible lowers the burden on the real-time platform. In the ControlBox the messages are inserted into a cooperative awareness.

The ComBox consists entirely of Commercial-off-the-Shelf (COTS) parts. Hardware specifications are sufficiently over-dimensioned so as not to run into the same computation bottleneck as experienced in C&D. A secondary objective of Team FUTURUM was to show that you do not need a large budget to implement a CACC-capable vehicle, convivially dubbed *CACC on a shoestring*.

The ComBox connects to the ControlBox through Ethernet. In 2010, IEEE 802.11p hardware had just become commercially available. For this purpose UNEX DCMA-86P2 miniPCI cards based on the Atheros AR5414A-B2B chipset were used. A Laird Phantom 5.9 GHz antenna is mounted on the spoiler of the vehicle.

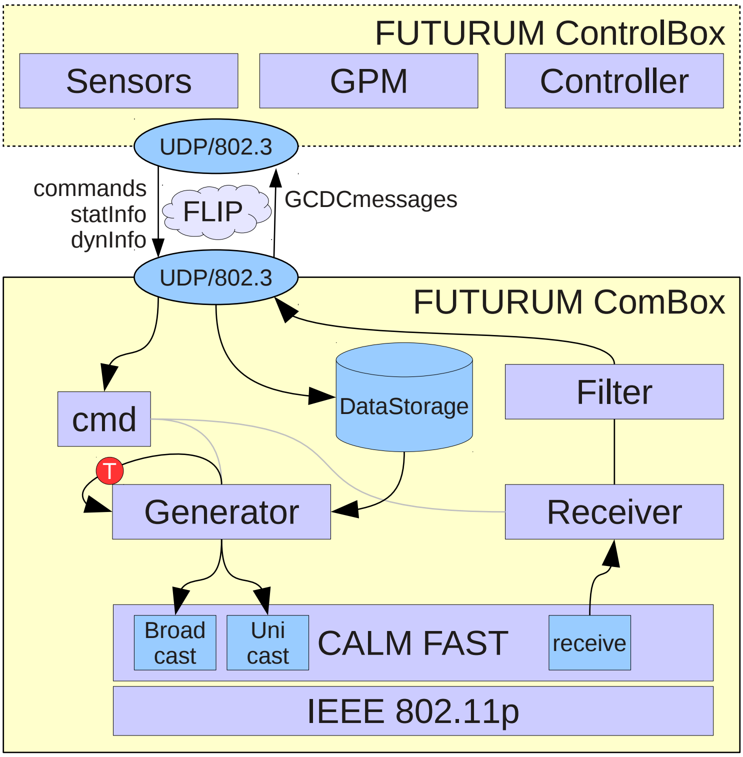


Figure 4.14: The FUTURUM Communication Architecture [26].

As modern cars are already filled with a great number of devices, adding yet-another-box may not be easy, especially in small cars. Though the ComBox design goal was not specifically to minimise the physical dimensions of the box, it should be noted that there is potential for significant down-sizing. By relying on COTS hardware it is possible to produce a relatively cost-effective IEEE 802.11p communication platform. This is also an important issue to consider; for swift market penetration it is unreasonable to expect six-thousand Euro cars to be equipped with communication and control hardware one third the price of the vehicle itself.

4.3.3 System operation

The Mini-ITX boards used in the ComBox are not equipped with a miniPCI slot. In order to use the miniPCI IEEE 802.11p module, a PCI risercard is connected to a PCI-to-miniPCI converter. One particular type of risercard caused constant freezes. We suspect this to be a timing issue on the PCI bus due to the increased length of the

PCI lines in the various interconnected Printed Circuit Board (PCB)s. A risercard from a different vendor functioned without problems. Ideally, the network hardware is integrated into the same PCB as the rest of the platform.

The same type of antennas were used as in the C&D project, however, in the GCDC—thanks to a bugfix to the driver supplied by the team from Karlsruhe Institute of Technology in Germany—it was possible to operate in the proper frequency band. No logging feature was implemented due to time constraints, however some output on the screen in the vehicle indicated beacon reception was good as long as vehicles were close by. In this respect, all teams concluded they had particular bad reception from the big truck TU Eindhoven participated with. The antenna was placed in the front on the roof, creating dead-zones around and behind the truck.

The Smart car used by Team FUTURUM has a glass roof. In line with the shoestring-paradigm introduced earlier, a piece of chicken fence was used to create a groundplate under the antenna array. The most noteworthy improvement this caused was a much cleaner signal from the GPS antenna.

In terms of scalability, the wireless communication was only a problem when driving in the dead zone behind one of the trucks, because the vehicle seemed to be obstructing propagation of its own signals. This caused vehicles driving behind it to operate in ACC mode. This is a problem that needs to be solved for future deployment, because it is not beneficial for vehicles to revert to ACC behaviour when they get closer to other vehicles: the ACC will respond with increasing the headway. This introduces instability into the traffic flow and does not result in string stable longitudinal behaviour. However, more dominant in the string stability performance evaluation of the FUTURUM car was its relatively poor acceleration capabilities, and the relatively long delay in its semi-automatic gearbox.

4.4 Lessons learned

Although few scientific results are available from these projects, some lessons learned can be destilled from either one or both projects. We saw in Sec. 4.2.4 that in the C&D prototype, sampling delay introduced a very large shift in time to the communicated signals. Though sample capture is often assumed to be instantaneous in simulation and analysis, it is a problem which exists in practice and must be solved. The resulting impact on overall delay can be larger than other components of the network-induced delay. A more detailed study on network-induced delay is performed in the following chapters by means of simulation (Chapters 5 and 6) and analytical modelling (Chapter 5).

Proper testing of components prior to integration is very important. In retrospect, the expected amount of data coming into the C&D platform from the Wireless Gateway was underestimated. This could have easily been tested with a simple test program which simulates output generated by an arbitrary number of vehicles. In the GCDC project this was tackled by several simulation tools shared by the teams which, for instance, fuzzed the input signals to see how well the system dealt with unexpected input or high load. Furthermore, the entire chain of components should be able to cope with the amount of data received by the communications hardware and should be load- and space-time scalable as discribed in Sec. 3.3.

The received information needs to be classified based on relevance. Safety critical information should come through, even at the expense of sacrificing other traffic or operations. In order to make the system robust, irrelevant information needs to be discarded as soon as possible in order not to clog up higher-level parts of the systems (such as applications which follow a publisher-subscriber pattern when observing the cooperative awareness).

Ultimately, it was delay which caused Team FUTURUM's vehicle to exhibit poor string stability. The networking and control platforms performed well, though the vehicle lacked the acceleration capability to respond sufficiently quick. The large delay in the gearbox was a dominating factor, the performance limiting factor of the system was mechanical in nature and does not lie in the wireless networking.

Both C&D and GCDC showed that the antenna placement is critical. This is still an issue to be solved because vehicles come in many shapes and sizes, and this can have dramatic impact on the radiation pattern of the antenna.

The C&D system is pure ad hoc and entirely stateless. The GCDC Interaction Protocol specifies additional, statefull interactions. These communication patterns need to be carefully designed. Interactions in a vehicular network should not have the possibility to cascade or follow self-expanding patterns; this triggers a chain of events which could seriously threaten scalability.

In the discussion about the Team FUTURUM system, a strong motivation was provided why the communication platform should be cheap. ITS, and CACC in particular, is a technology which benefits greatly from large penetration so the technology should also be available in budget vehicles.

Most problems encountered in these two implementations have little to do with the scalability limits found in the simulation experiments and analytical evaluation of IEEE 802.11p presented in this dissertation. On the one hand, this is because the implementations operated well within the scalability limits as found in both simulator and analysis. On the other hand, few of the problems encountered in the proof-of-concept implementations are part of the simulation and analytical models and as such do not manifest there. Delay in sensors and actuators (such as the gearbox-delay in the GCDC vehicle) or sampling delay due to components running at different update rates (responsible for the majority of delay in the C&D vehicles) are not part of simulation and analysis models. They can, however, be alleviated by changing the design of some components. This will raise the scalability limits *internal* to a vehicle. When these limits are sufficiently raised, the *external* scalability limits of the wireless networking will become dominant.

4.5 CACC and sensitivity to packet loss

The ultimate goal of beaconing is to facilitate an application such as CACC to work, under a wide range of network conditions. This section describes and analyses the relation between network conditions and CACC string stability performance, described in Sec. 2.2.1. This analysis is carried out by means of a simulation study, using the same controller prototype as used in the Connect&Drive project.

The network conditions are modelled by means of beacon generation rate λ_g and P_s. From this analysis, we aim to obtain a minimal update rate required for the application to remain functional, and evaluate the impact of network-level metrics. This study considers application-level metrics: the velocity and acceleration of vehicles, and how they progress over time in relation to the vehicle dynamics.

4.5.1 Simulation environment

The system is evaluated in a discrete event simulation framework composed of several simulation environments and models:

1. Wireless communication behaviour using OMNeT++/MiXiM (see Sec. 2.7).

2. Mobility behaviour of vehicles, modelled using Simulation of Urban Mobility (SUMO) [121].

3. Vehicle behaviour (the controller prototype based on the C&D system [41]), including the ACC and CACC models, which have been implemented using Simulink [122] and exported to a C++ shared library using the Real-Time Workshop tool in Simulink so that it can be integrated in SUMO.

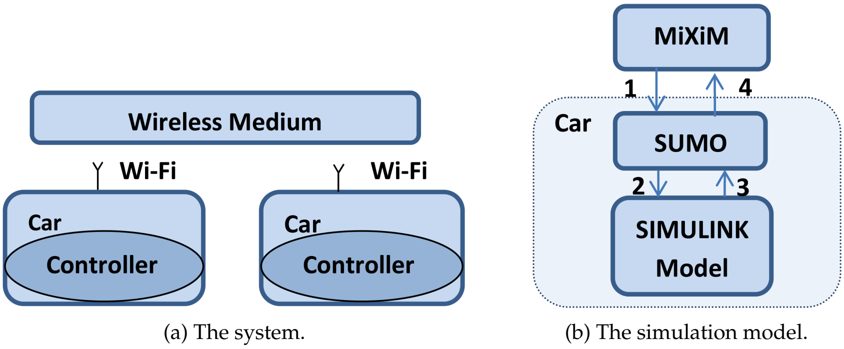
The system is illustrated in Fig. 4.15(a), while the corresponding simulation model is shown in Fig. 4.15(b). `Cars` are equipped with Controllers and communicate with each other through their Wi-Fi interfaces. The method described in [123, 124] is used for bidirectional coupling between OMNeT++/MiXiM and SUMO, where the two simulators communicate with each other through a traffic control interface (TraCI) using a TCP connection.

Every simulation step is divided into four smaller steps, illustrated in Fig. 4.15(b):

1. MiXiM sends the information in its cooperative awareness (e.g., acceleration of the preceding vehicle) to SUMO.

2. In the SUMO part, the acceleration received from MiXiM and the other parameters are used as inputs for the CACC controller for each vehicle to calculate `a_real` and v (see also Fig. 4.17).

3. The resulting velocity and position are used to simulate the movement of vehicles in SUMO.



(a) The system.

(b) The simulation model.

Figure 4.15: Experiment structure used to evaluate CACC sensitivity to packet loss.

4. After moving the vehicles, SUMO sends a trace to MiXiM which comprises the vehicles' acceleration, velocity and position as realised by the CACC controller. Then MiXiM moves its communication nodes according to this trace.

Asynchronous to the mobility timesteps, the transmission of a beacon can happen if a node's beacon timer has expired. Simulation of ACC only requires the SUMO model and the shared library converted from the Simulink model, as no communication is involved. More details on the implemented and used integrated simulation environment can be found in [125].

OMNeT++/MiXiM model

The vehicles in SUMO have their counterparts in the communication simulator. These nodes have the same positions as the vehicles in SUMO and exchange information by means beaconing. Simulation with the complete MAC and PHY model is impractical, because the number of nodes required to get to saturated channel conditions causes the total required computational effort to explode. In order to study the impact of packet loss, the physical layer model uses a stochastic packet loss model independent of n. For each beacon, the probability that a packet is lost is drawn from a uniform distribution, and is independent from that of other packets.

As described in Sec. 4.2.2, beacons are generated in the network layer in an event-driven fashion. The beacons contain information from a vehicle's sensors, such as the acceleration, velocity and possition obtained from SUMO. Received information is stored in the cooperative awareness.

SUMO model

The simple straight road modelled in SUMO [126] is shown in Fig. 4.16 together with a platoon of ten vehicles. Vehicles are numbered `veh0`, through `veh9`. The leading

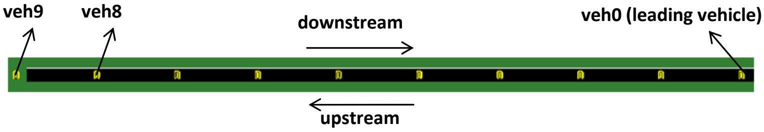


Figure 4.16: SUMO traffic model.

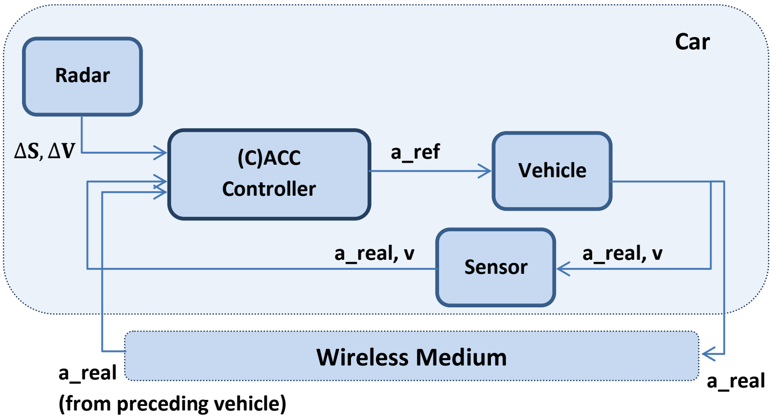


Figure 4.17: A vehicle's control system as modelled in Simulink.

vehicle (veh0) moves from left to right (the downstream direction) and the other nine vehicles equipped with ACC/CACC controllers follow.

Simulink model

The Simulink model of a Car module is shown in Fig. 4.17. This part of the system is driven by a clock event, which is scheduled at time $t_{now} + 10$ ms and rescheduled after processing of this event has completed. This causes a time-driven behaviour, stepping through evaluation of the control logic. The actual (C)ACC behaviour is modelled in the (C)ACC Controller block in a similar fashion as used in the C&D project [43]. At the beginning of each simulation step, the (C)ACC Controller reads the relative speed and distance to the preceding vehicle from the Radar module. This module models an automotive radar, and applies a realistic delay to these values. The host vehicle's acceleration and speed are read from the Sensor module. In addition, CACC obtains the acceleration of the preceding vehicle from the Wireless Medium through the Wi-Fi interface, whereas this input is not used for ACC.

The control objective is to realize a desired distance to the preceding vehicle, taking into account a pre-defined maximum speed, referred to as the cruise speed. Note that the cruise speed is a maximum speed when the vehicle operates in (C)ACC mode. If there is no target (preceding) vehicle, the system switches to Cruise Control mode, in which case the cruise speed becomes the target speed. The distance to the preceding vehicle is denoted as $D(t)$:

$$D(t) = D_{\text{standstill}} + hV(t), \tag{4.1}$$

where $D_{\text{standstill}}$ denotes the desired distance to the preceding vehicle at standstill. h is the desired time headway, the time it takes for vehicle i to reach the current position of its preceding vehicle $i-1$ if it continues to drive with a constant velocity [41]. $V(t)$ is the host vehicle velocity at time t. Based on these inputs, the (C)ACC controller calculates a reference acceleration `a_ref`, implementing a one-vehicle-lookahead control structure as described in Sec. 4.2. The `Vehicle` module takes this reference acceleration as input and mimics the response of a real vehicle (e.g., taking into account the vehicle's inertia). The output of the `Vehicle` module is the resulting (or real) acceleration, referred to as `a_real`.

This acceleration is then used to calculate the vehicle's resulting speed v, and hence resulting position s, at the *next* simulation time step. Next, `a_real` is sent to module `Wireless Medium` to model its wireless transmission. Note that when packet loss occurs and an updated acceleration value is lost, the CACC controller uses a previously received and stored acceleration value. This holds in general, as evaluation of the controller happens at a higher rate than reception of input through the wireless medium (the control mechanism is evaluated at 100 Hz). This "sample-and-hold" behaviour could be replaced with e.g. a Kalman filter to obtain more accurate estimates of the real value. This is applied in the CACC implementations discussed in Sec. 4.2 and 4.3, but not included in the simulation model. Additional (C)ACC controller details can be found in [41].

In these experiments, we consider only a one-vehicle-lookahead controller topology (as introduced in Sec. 4.2) and vehicles are following eachother, establishing Line-of-Sight (LoS) communication paths between relevant communication peers. Within the range of distances (headways) used in the experiments (7.7 - 40 m), P_s is not expected to vary significantly with distance for LoS communication. When studying scenarios with multiple-vehicle-lookahead, or non-communicating vehicles mixed with CACC-equipped vehicles, the wireless communication needs to be performed with a more detailed propagation model in order to account for non-LoS situations.

4.5.2 Experiment description

The aim of these experiments is to investigate the impact of P_s and λ_g on ACC and CACC string stability performance. Whether the disturbance of the leading vehicle is amplified upstream through the platoon can be investigated by observing the *speed* and *acceleration* of the following vehicles. The vehicle speed, as well as its under- or overshoot in situations of traffic disturbances, can be used as string stability performance metrics [21]. In this context, under- and overshoot can be defined as the absolute difference between the lowest (or highest) vehicle speed of the last following vehicle and the (target) speed of the leading vehicle. In a string

Parameter	**Value**
Time headway h	0.7 s
$D_{\text{standstill}}$	7.7 m
vehicle length	4.46 m
$a_{\max}$	2 m/s^2
$a_{\min}$	-9 m/s^2

Table 4.1: Simulation parameters for the CACC study.

stable traffic flow, the under- and overshoot are very small. We only observe the velocity response of the last following vehicle (`veh9`). When the platoon is not string stable, it is this vehicle that will experience the strongest disturbances.

Experiment setup

A platoon of ten vehicles is placed on a straight single lane road 5000 meters long, as illustrated in Fig. 4.16. The simulation parameters are given in Table 4.1, the varied parameters in Table 4.2. To study the impact of the communication performance on string stability performance, λ_g and P_s are varried. Increased time headway provides more "buffer" around vehicles, which can absorb instability. Varying h shows the impact of this controller parameter, and its relation with communication performance.

Varying $D_{\text{standstill}}$ is not of interest to the dynamic platoon behaviour because it has no effect on the dynamic behaviour, $D_{\text{standstill}}$ could be seen as a "virtual extension" of the vehicle. It is a margin a vehicle uses around its preceding vehicle. The distance between vehicles will change with different $D_{\text{standstill}}$ setting, which may impact communication performance. However, variations in the range [5,10] m are assumed not to be of much influence. Likewise, varying the length of a vehicle affects the distance between vehicles but has no effect on the dynamic behaviour.

In order to guarantee a high statistical accuracy of the obtained results, multiple runs have been performed and 90% confidence intervals have been calculated. For all experiments, the largest calculated confidence interval is ± 3 % of the shown calculated mean values. They are omitted from the plots for readability.

In order to validate the controller model and traffic model, the performance of ACC (without communication), and CACC with perfect communication (each vehicle can always get its preceding vehicle's acceleration without loss and delay) are compared. The results (not shown here) are very similar to the results obtained in [41] and it is concluded in [125] that the Simulink model as used in C&D and the combined SUMO-Simulink model are satisfactorily equivalent. In these baseline experiments, CACC outperforms ACC on string stability.

Parameter	Value
Beacon generation rate λ_g	5, 10, 20, 25 Hz
Reception probability P_s	1, 0.9, 0.8, 0.7, 0.6, 0.5
Time headway h	2, 1.5, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5 s

Table 4.2: Varied parameters for the CACC study.

Simulation scenarios

The leading vehicle starts with an initial speed of 20 m/s which remains constant until t = 80 s (i.e., up to the 8000th simulation time step). During this period, each following vehicle reaches a stable speed of 20 m/s and the distance between consecutive vehicles also stabilises.

The leading vehicle performs a pre-set motion pattern depending on the scenario—it either accelerates or decelerates, enabling study of these events in isolation. The metrics of interest are calculated relative to the behaviour of `veh0`. Using a step change of the lead vehicle acceleration heavily excites the system, such that the important dynamics become clearly visible.

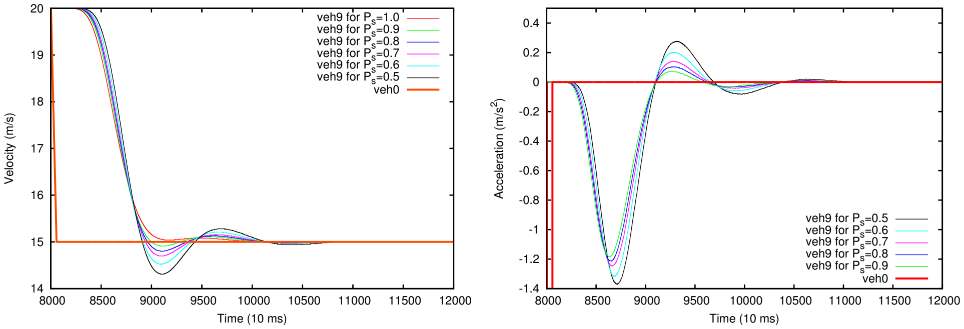
In the *deceleration* scenario, at time step 8000 `veh0` decelerates with -9 m/s^2, until it reaches the speed of 15 m/s. In the *acceleration* scenario, at time step 8000, `veh0` accelerates with 2 m/s^2 until its speed reaches the value of 25 m/s. For both scenarios, P_s and λ_g are varied as shown in Table 4.2. In addition, we also study these two scenarios with λ_g=15 Hz and P_s=0.8 while varying the time headway values (see Table 4.2).

4.5.3 Simulation results and analysis

For the deceleration as well as the acceleration scenario, the measured velocity, acceleration and over/undershoot of the velocity will be discussed below. In addition, velocity and acceleration under a varying time headway will be provided to show the impact of time headway.

Response to deceleration

The curves from top down at the 9000th time step of Fig. 4.18 (a) indicate P_s in descending order. For a constant value of λ_g (10 Hz) and h (0.7 s), the velocity fluctuations of `veh9` increase with decreasing P_s. This indicates that the disturbance introduced by `veh0` is amplified more through the platoon in the upstream direction. In other words, with decreasing P_s, the platoon is less string stable. Moreover, the undershoot of the velocity also increases in magnitude as P_s decreases, evident in Fig. 4.18 (a). Fig. 4.18 (b) shows the acceleration for this scenario; lower P_s results in a delayed, but larger response.



(a) Velocity of `veh0` and `veh9`.

(b) Acceleration of `veh0` and `veh9`.

Figure 4.18: Varying P_s in the deceleration scenario, with h=0.7 s, λ_g=10 Hz.

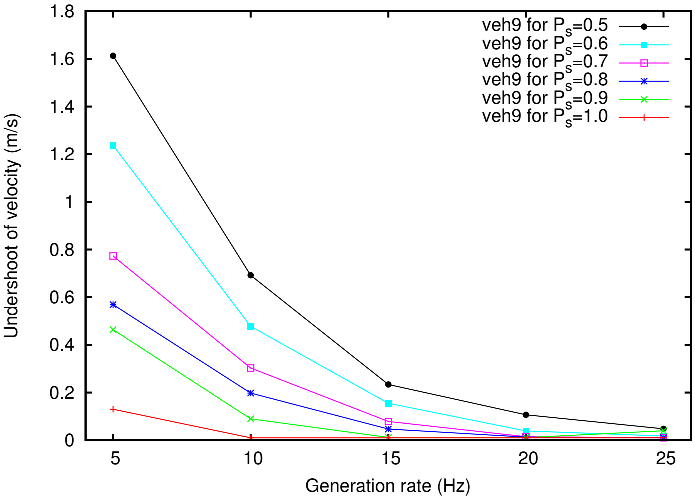
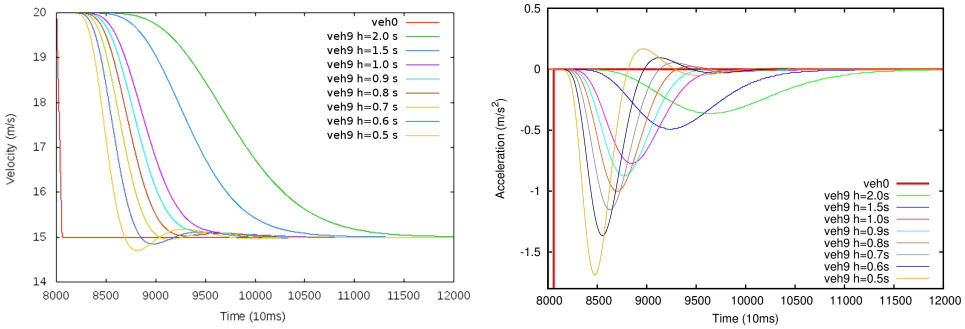


Figure 4.19: Undershoot for velocity of `veh9` in the deceleration scenario, with h=0.7s.

The undershoot of velocity for the last vehicle is shown for different combinations of selected λ_g and P_s in Fig. 4.19. This figure indicates that, with a selected value of λ_g and h, the undershoot of velocity for the last vehicle increases as P_s decreases. This means that the platoon becomes less string stable. It can also be observed that for a selected P_s, string stability becomes worse as λ_g is reduced.

A vehicle always uses the acceleration value which is most recently received from its preceding vehicle as the input of the CACC controller. Therefore, a higher λ_g results in a higher possibility of receiving fresh information under a constant P_s. Likewise, higher P_s can also result in a higher possibility of receiving fresh information under a constant λ_g. For λ_g=25 Hz, P_s has little influence because



(a) Velocity of `veh0` and `veh9`.

(b) Acceleration of `veh0` and `veh9`.

Figure 4.20: Varying time headway in the deceleration scenario, with λ_g=15 Hz, P_s=0.8

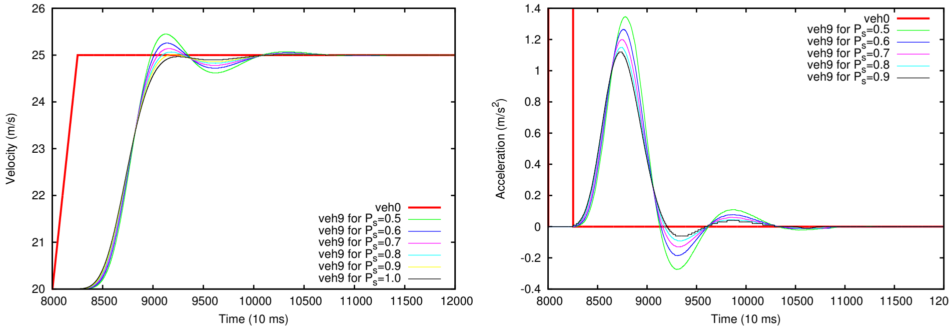
vehicles can still easily receive sufficiently fresh information–even with a P_s of 0.5. It should be noted, however, that at 25 Hz the wireless channel capacity will become a limiting factor when considering larger numbers of vehicles [9], as evaluated throughout this dissertation, so a high P_s is not always achievable.

Furthermore, observe in Fig. 4.19 that the situation λ_g=5 Hz, P_s=1 shows far lower undershoot than λ_g=10 Hz with a P_s of 0.5. This seems counter-intuitive, but can be explained by the distribution of beacon arrivals in time. In the first case, inter-arrival time τ_r is deterministic with 200 ms. In the second case, τ_r follows a geometric distribution multiplied by 200 ms intervals, where several "missed updates" can follow from consecutive loss. The magnitude of this difference suggests a sensitivity to consecutive loss.

Varying time headway

Fig. 4.20 (a) shows the velocity of the last vehicle for varying the value of h with λ_g of 15 Hz and P_s=0.8. Note that the curves from top to bottom relate to the headway in descending order. Fig. 4.20 (a) indicates that with the selected λ_g and P_s, as time headway increases, the platoon becomes more string stable, i.e., the velocity of the last vehicle decreases with fewer fluctuations, findings also reported in [41, 43]. Fig. 4.20 (b) shows the acceleration of `veh9` in response to the sudden deceleration of `veh0`. The figure clearly shows that, with smaller h, reaction of `veh9` is sooner, but also more aggressive than with larger h, even to such a degree that deceleration resembles an emergency stop.

Furthermore, with larger h, the relative distance between vehicles is larger. When a disturbance occurs on a leading vehicle, the following vehicles do not react as aggressively as when small time headways are used. Though large time headways may be beneficial for string stability, they are detrimental to road throughput and capacity as described in Sec. 2.2.3. Therefore, it is important though challenging to find the smallest h which guarantees string stability, while maximizing the road



(a) Velocity of `veh0` and `veh9`. (b) Acceleration of `veh0` and `veh9`.

Figure 4.21: Varying P_s in the acceleration scenario, with h=0.7 s, λ_g=10 Hz.

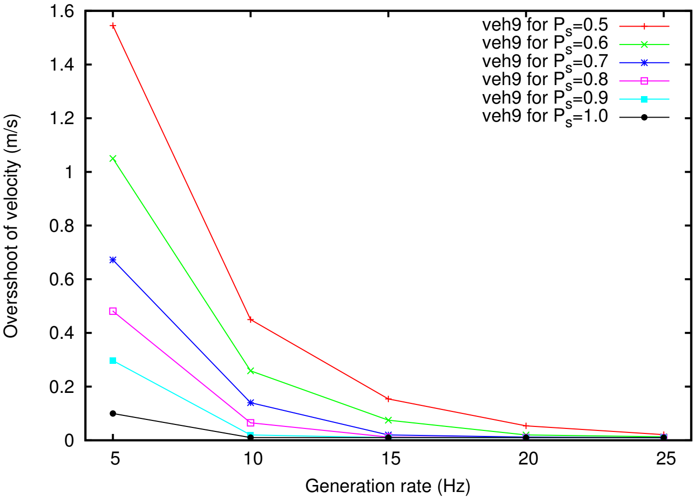


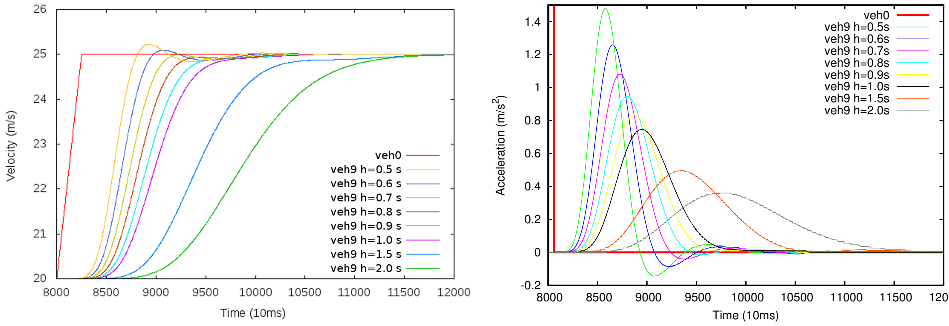
Figure 4.22: Overshoot for velocity of `veh9` in the acceleration scenario, with h=0.7 s.

capacity, especially in the face of traffic and network dynamics.

Response to acceleration

For the acceleration scenario, we again observe the dynamics of the last vehicle. The results of the simulation are plotted in similar fashion as the first scenario. For P_s=1, there is hardly any overshoot in velocity, see Fig. 4.21 (a). The acceleration of `veh9`, plotted in Fig. 4.21 (b), shows that with a lower P_s, reaction is later and more aggressive as also observed in the deceleration scenario.

Because this scenario deals with acceleration, Fig. 4.22 depicts the overshoot of



(a) Velocity of `veh0` and `veh9`.

(b) Acceleration of `veh0` and `veh9`.

Figure 4.23: Varying time headway in the acceleration scenario, with λ_g=15 Hz, P_s=0.8.

the velocity of `veh9`. From Figs. 4.21 (a) and 4.22 it becomes evident that for a given value of λ_g and h, the CACC controller's string stability performance decreases with a lower P_s. Accordingly, for a given P_s and h, the string stability becomes worse with lower λ_g. In addition, Fig. 4.22 shows the same sensitivity to the distribution of the arrival of beacons as observed in Fig. 4.19.

Varying time headway

The curves from top to bottom in Fig.4.23 (a) relate to time headway in ascending order for a λ_g of 15 Hz and P_s=0.8. From this figure, the same conclusions can be derived as for the deceleration scenario. In particular, it can be observed that string stability improves when h is increased. Recall, however, that an increased h yields a lower traffic flow, as described in Sec. 2.2.3. Fig. 4.23 (b) shows the corresponding acceleration curves for this experiment. These curves clearly show that, even in an acceleration scenario, string instability may result in sudden *deceleration* of vehicles, clearly visible for $h = 0.5s$.

ACC vs CACC

In order to get an idea of the performance gains of CACC over ACC, the acceleration and deceleration scenarios are also performed for an ACC control setup. The string stability performance of the ACC system is not affected by λ_g and P_s. This is because ACC does not use beaconing, but radar measurements for the calculation of time headway and acceleration, as modelled in the `Radar` block in Fig. 4.17.

The most interesting metrics in this respect are the under- and overshoot of velocity in the deceleration and acceleration scenarios. For CACC, the undershoot of the deceleration scenario (see Fig. 4.19) shows a maximum of 1.613 m/s for λ_g=5 Hz, P_s=0.5. ACC, under the same time headway setting of 0.7 s, has an overshoot of 11.14 m/s as reported in [21], independent from λ_g or P_s. This value is not plotted in Fig. 4.19 for readability.

The overshoot in the acceleration scenario for CACC (see Fig. 4.22) shows a maximum of 1.545 m/s under the same circumstances. The overshoot of ACC in this case is 11.06 m/s [21], also not plotted in the figure for readability. In particular for a P_s of 0, the CACC string stability overshoot and undershoot are more than 10 times smaller than those measured for the ACC system. Even for a P_s of 0.5, the over- and undershoot are more than 5 times smaller. This supports the conclusions of [38, 41] that the CACC feed-forward controller enables smaller time headways than the ACC feed-back controller, while still maintaining string stability.

4.5.4 Discussion on scalability of the simulation approach

As discussed in Sec. 2.7, the number of runs which need to be performed as part of a simulation study do not scale well when the number of iteration variables is increased, or when the number of elements per iteration variable is increased. When regarding the iteration variables and the number of runs as $1 \times N$ matrices, the resulting cartesian product is the space which comprises the entire experiment. The resulting set of scenarios scales polynomial with $\mathcal{O}(N^c)$, where c is the number of iteration variables and N is the number of values of a single iteration variable.

Within execution of a single run, MiXiM uses an optimisation to divide the simulated environment into partitions, as described in Sec. 2.7.2. This works well for sparse networks, because then many transmissions could *never* reach a subset of all nodes and their signal reception models do not need to be evaluated. However, when researching the scalability of IEEE 802.11p in vehicular networking, increasing the number of nodes n within range is an important aspect. The result approaches a fully connected graph, and every `Signal` needs to be treated for all edges of the graph, resulting in $\frac{n(n-1)}{2}$ operations. This causes a dramatic increase in wall-time duration of a single run, and hence the entire experiment through the relation described in the previous paragraph.

The evolution of metrics of interest in the traffic context occurs at a lower rate than those in the networking context. The shockwaves studied in the simulation experiments in this chapter, the simulated time is 120 seconds. As visible from the plots, the entire shockwave has a duration of more than 10 seconds. The time scale at which wireless communication operates is in the order of milli- or even microseconds. This greatly increases the number of events which needs to be processed in order for effect on metrics in the traffic context to manifest. This, in turn, increases the duration of these experiments, which does not scale well to begin with as described above. This limits the tractability of simulation experiments when evaluating traffic flow stability at a very large scale with the method used in this section. An alternative is replacing the communication simulator with a stochastic communication model, as presented in Chapter 5. The simulation platform used here will be used extensively for validation of these stochastic models, but also for evaluation of improvements studied in Chapter 6.

4.5.5 Conclusion

In this study, the relation between the network- and application performance was evaluated. The generation rate λ_g and success probability P_s have significant influence on the performance of the evaluated CACC system. Lower λ_g and/or lower P_s prevent vehicles from receiving fresh information from preceding vehicles, resulting in poor performance of the CACC application. This poor performance manifests as large overshoots in speed, indicating string instability. This leads to the conclusion that a high beacon reception rate λ_r is favourable for string stable CACC operation. However, CACC appears to degrade gracefully in response to a reduced λ_r and be relatively loss tolerant. Even for low λ_r, CACC will still outperform ACC. Moreover, the study suggests that the CACC system has potential to compensate for poor reception by means of adaptivity of some parameters.

The delay of the communicated information is minimal, as 10 vehicles do not load the wireless channel to such an extent that contention delay increases significantly. It should be noted that in an IEEE 802.11p system, low P_s is usually accompanied by increased delay due to its CSMA/CA medium access mechanism [12]. The relation between λ_g and P_s is assumed independent in this study, however in reality λ_g and other parameters influence P_s. This relation is studied in detail Chapter 5.

Another important conclusion is that sensitivity to P_s increases with a lower λ_g, but the difference between 5 and 10 Hz for 0% loss is small. This leads to the conclusion that CACC is relatively loss tolerant—it will still be able to function and outperform ACC if λ_r reduces slightly. In addition, when CACC experiences a reduced λ_r to such an extent that it threatens string stability, string stable behaviour may be recovered by increasing h. This suggests an adaptivity in the CACC design based on properties of the networking context.

These experiments focus on a string of ten vehicles, and packet loss is artificially introduced by means of an independent loss probability $(1 - P_s)$. In order to operate a CACC system near the limits of the beaconing system, many more communicating vehicles are required in order to get a more realistic collision model to reach the point where the system experiences a P_s of 0.5. In addition, the number of vehicles also needs to be increased in order to evaluate string stability in a broader scope (i.e., traffic flow stability [48, 34]).

4.6 Concluding remarks

Both the system developed in C&D and the system developed in GCDC have shown that it is possible to actually drive cars in a platoon by means of CACC. In fact, there is also good outlook for CACC in mixed traffic of passenger cars and trucks from different vendors, and it does not necessarily need to be expensive. It should be stressed that standardisation is key to cooperative ITS systems of any kind.

The implemented systems do not operate near the scalability limits of the wireless communication. To operate near these limits, more vehicles are needed. This impedes the practicality of the prototype-based approach for system validation. Not only are these systems financially costly, they are also difficult to manage when performing experiments with a focus on the wireless communication. These experiments are difficult and time consuming to reproduce. More tractable approaches to study beaconing in great detail are simulation and analytical modelling.

The scalability of performance evaluation by means of simulation study was discussed in Sec. 4.5.4. Because both the communication and traffic simulators experience an increase in wall-time duration for an increasing number of nodes, it is computationally attractive to perform (part of) the evaluation by means of analytical models. A detailed model of IEEE 802.11p beaconing will be provided in Chapter 5.

CHAPTER 5

Network performance of beaconing

In Chapter 3 we defined the performance limits of a beaconing system by means of two straight-forward expressions (3.6) and (3.10). It is, however, necessary to model the behaviour of the IEEE 802.11p MAC in more detail in order to obtain a more accurate model and gain more insight in the network performance of beaconing

Chapter 4 explored a prototype-based approach and a simulation-based approach to evaluate the performance of a beaconing system for a CACC application. It was also shown there that the prototype approach is costly and difficult to manage, and that simulation is a time-intensive approach. It would be beneficial to derive analytical models to express the beaconing performance, obtain insights, and explore the impact of parameters.

Detailed analytical models allow performance evaluation of a protocol [127, 128, 129, 130, 75]. These analytical models can also be used to enhance protocol design [131, 132, 133], to set or adapt protocol variables [134, 135], or to adaptively alter configuration in response to a change in demand or environment [109, 20]. Additionally, an accurate and tractable model of the MAC can be embedded in a model with a broader scope, e.g., taking into account the motion of vehicles. This is attractive to study situations where the vehicle mobility impacts the communication, and vice-versa. This is particularly enticing in situations where a simulation-based approach becomes too cumbersome because of the differences in time scale between events in the network and dynamics in traffic.

The IEEE 802.11 CSMA/CA is generally considered to be well-understood, and many detailed models are available in the literature. The focus of these models has historically been mostly on Unicast transmissions, as that is the prevailing mode of operation in a WLAN. However, this is not the communication pattern used in vehicular networks, as was described in Chapter 3. Current WLAN / Unicast models do not readily match with the vehicular networking scenario. In this chapter, the main contribution is to provide an analytical model for beaconing in vehicular networks, adapted to the specifics of the network traffic regime in a Broadcast environment with the potential to encompass many nodes. The assumption of independence among transmissions often made in WLAN models does not provide good estimates of the backoff counter (*bc*) blocking probability in the IEEE 802.11 DCF. We present a model of the DCF in this chapter, which addresses this issue by properly modelling *bc* decrement behaviour of the DCF. We provide analytical expressions for the phenomena we call Collision Multiplicity and Streak Length to calculate a better approximation of the expected service time and of the blocking probablity.

In addition, an analytical model of the EDCA is introduced, taking into account the specifics of the EDCA access method. The EDCA shows particularly different *bc* decrement behaviour, which results in fundamentally different behaviour under increasing number of communicating nodes.

Both models are validated and compared for application in a beaconing system. The models allow important performance metrics to be obtained, and also allow prediction of the point at which CSMA/CA breaks down with good accuracy. This may be beneficial in the development of congestion control mechanisms. The comparison study of the DCF and EDCA shows that the DCF has favourable properties for beaconing in a vehicular network.

This chapter expands the answer to research question 4 (”How does IEEE 802.11p scale under varying traffic densities, and can it meet the requirements imposed by CACC?”) as given in the previous chapter and also answers research question 5 (”How do the scalability limits found in simulation, analysis and proof-of-concept implementations relate?”).

This chapter is structured as follows. Related work is discussed in Sec. 5.1. Then, Sec. 5.2 introduces the high-level model into which the detailed MAC model is embedded. To obtain performance metrics of the MAC, the DCF and EDCA are modelled using a mean-field approach. The DCF model is described in detail in Sec. 5.3 and the EDCA model in Sec. 5.4. These models are validated in Sec. 5.5 and compared in Sec. 5.6.1. Sec. 5.8 concludes this chapter. This chapter contains material published in [18, 11, 20, 19].

5.1 Related work

Prior to describing our contributions in the area of 802.11 modelling in the rest of this chapter, this section describes related work and gives an overview of the various approaches which historically grew into the present state-of-the art. The main metrics of interest in most literature on wireless networks are throughput and delay. Especially in the days of 1 and 2 Mbps WLAN, throughput was a very important optimisation criterion. With the emergence of delay-sensitive applications such as VoIP and streaming video, the actual delay incurred by the wireless link became very relevant, leading to the development of priority-based access control in IEEE 802.11e EDCA.

In the context of vehicular networking, the Broadcast communication pattern of beaconing differs significantly from VoIP in a WLAN, and delay is found not to be the limiting factor [108, 12]. Given the messaging nature of beaconing, the probability to successfully receive a beacon is a more relevant metric. Though the topic of performance deterioration in a channel where broadcasts dominate has been extensively studied in the area of flooding and network-layer broadcasts [136, 137], the beaconing communication pattern does not generate a Broadcast Storm [104], because (in most implementations) beacon generations are independent and limited, (they are not self-expanding). The traffic generated by a single vehicle is low, but the number of nodes concurrently sharing the same channel can reach levels uncommon in WLAN situations.

The IEEE 802.11 family of standards has been widely studied over the past fifteen years. The models in literature can be classified based on the way they treat the transmission queue. Because throughput was a very important optimisation criterion in the early days, most early 802.11 models focus on saturated WLAN scenarios: the number of nodes is low, but the load supplied per node is generally high. These models are refered to in this text as saturation models. Another class of models, which assumes there are periods in which a station does not have a packet for transmission, are referred to in this discussion as models including the non-saturated regime.

Saturation models are described in Sec. 5.1.1, models including non-saturation are described in Sec. 5.1.2. Sec. 5.1.3 then focusses on models specifically designed for vehicular networking. Following that, two important aspects are discussed: Sec. 5.1.4 discusses the independence assumption and Sec. 5.1.5 discusses the assumption of saturation.

5.1.1 Saturation models

This class of models assumes the system to be in saturation: a station always finds a packet in the queue after completion of a transmission.

DCF models

In [129], Chhaya and Gupta study the DCF and RTS/CTS, looking at throughput and spatial fairness due to differences in signal power for different transmitter-receiver pairs. This is a physical-layer fairness aspect which is important for static WLAN situations using Unicast transmissions. The RTS/CTS sequence imposes overhead with respect to medium resource utilisation and delay, but greatly improves the resilience to hidden terminals. A critical frame length exists beyond which it is beneficial to use RTS/CTS; as it primarily depends on the length of the frame and secondarily on the distance between transmitter-receiver pairs. Furthermore, [129] makes the important observation that it is best to operate the 802.11 CSMA/CA under light or moderate load, since higher loads result in reduced throughput and increased delay.

Bianchi provides a foundational model of the IEEE 802.11 DFC [128, 138, 139]. Focus of the Bianchi model is on saturation throughput for the Basic Access mechanism and RTS/CTS in the DCF under ideal channel conditions for the 802.11b MAC. A Discrete Time Markov Chain (DTMC) $\{s(t), b(t)\}$ is provided to model the backoff stage $s(t)$ and backoff counter $b(t)$ for a single station at slot time t. This approach discretises time into either empty or busy slots, the latter of which are subdivided into successful or colliding transmissions. The stationary distribution of the chain, $b_{i,k}$, is found by normalisation towards $b_{0,0}$, the state in which a node will perform a transmission after its bc reaches zero in its first backoff stage. The probability that a station transmits in a generic slot is expressed as τ, which is found by summing over all $b_{i,0}$ states. The resulting system behaviour is found by averaging interaction with other stations of the same type, a method known as mean-field approximation, described in Sec. 5.2.

The Bianchi model does not include all features of the 802.11 DCF. First, it does not allow for bc freeze, i.e., bc decrement occurs even during a channel busy period. Secondly, there is no maximum number of retries: after transmission in the maximum backoff stage, it simply stays in this stage and does a new backoff. The result is a potentially unbounded delay because a frame can suffer many successive failed retransmissions. Thirdly, the EIFS, used in response to an observed collision on the medium, is also not part of the model.

Bianchi's model sparked a whole family of models, each adding protocol features or using the model to extract different metrics. Ziouva and Antonakopoulos [140] provide an update to Bianchi's model, to reflect bc freezing during channel busy periods by adding self-loops in the states where $bc > 0$. Wu *et al.* [133] and Chatzimisios *et al.* [135] each add frame retry limits to Bianchi's model. The model by Wu *et al.* was adapted by Wang *et al.* [141] to study delay of the 802.11 DCF. This model includes the EIFS, but lacks bc blocking while the channel is busy. Zhang *et al.* [142] improve the accuracy by including bc freezing into their model to study medium access delay of the DCF. Abdrabou and Zhuang [143] use Bianchi's model to derive a service time distribution for the DCF for Unicast with RTS/CTS. They show that the service time can be approximated by a geometric distribution because the system is near-memoryless under saturation conditions.

Tinnirello *et al.* [144] update the Bianchi model with backoff freezing and discuss its implications. It is identified that there exist *anomalous slots* in addition to the generic slots on which the modelling technique is based. It is concluded that the hypothesis of uncorrelation between consecutive channel slots and statistical homogeneity, as is very common in most models, is not generally true [144]. One anomalous slot is the one immediate following a successful transmission. [144] states that in Unicast, this slot cannot be used for transmissions by any other node, except the one which has just performed the successful transmission. Another anomalous slot occurs after the end of an EIFS, which will not be used by any node because only nodes currently in contention are considered in the discussion. These nodes have $bc > 0$, and will need an additional slot to decrement. These anomalous slots exhibit a different probability of transmission and collision than the other slots. However, this discussion does not consider new arrivals during the previous slot (in which the medium was busy) under the assumption that these are very rare. These new arrivals are, however, probable in a scenario encompassing many nodes. New arrivals are considered in the way bc freezing is modelled in our DCF model described in detail in Sec. 5.3.

EDCA models

Xiao [145] performs analysis of the IEEE 802.11e EDCA based on Bianchi's foundational model [138, 139] and the model by Ziouva and Antonakopoulos [140]. Xiao adds traffic class priority differentiation with respect to CW size and a finite retransmission limit, whereas both models by Bianchi and Ziouva and Antonakopoulos consider infinite retries. Furthermore, Xiao provides a simpler and more accurate delay model for Unicast than [140].

Hui and Devetsikiotis [130, 146] identify that there is a need to simplify and unify the three seemingly different models by Bianchi [139], Calì [131] and Tay [147], and provide a unified model and analysis method for EDCA. This model is then used to evaluate saturation throughput and and delay performance of EDCA.

Xiong and Mao [148] provide an analysis of the coexistence of DCF and EDCA stations, recognising that there may be a proliferation of EDCA capable stations alongside DCF deployments. Three DTMCs are provided: (1) a DCF station model, 2) an EDCA station model carrying voice or video traffic, and, (3) an EDCA station model carrying best-effort or background traffic. All three DTMCs assume saturation. The EDCA models incorporate the proper EDCA bc decrement rules: a station also decrements the bc after an AIFS has passed, but blocks when a transmission by a higher-priority station is performed.

Taher *et al.* [149] note that both DCF and EDCA analytical models are incrementally improved over the past years, but still not complete. They include the TXOPLimit into their model, in addition to the EDCA bc decrement rules. This is accomplished by explicitly including the state of the MAC into the model.

Tinnirello and Bianchi identify in [150] that most EDCA models rely on per-slot statistics, sharing a methodology originally proposed for the DCF. As an alternative, a modelling methodology is proposed which relies on the fixed-point computation of the whole (residual) bc distribution occurring after a generic transmission attempt. This approach achieves accurate results for high level metrics such as throughput, but also accurate low level results such as the bc distribution when compared to simulation experiments.

5.1.2 Models including non-saturation regime

In practice, data traffic is typically of a bursty nature, and mobile stations are generally far from being saturated [151]. For instance, an HTTP response usually exceeds the payload size of a single MAC frame, causing a burst of frames to enter the transmission queue. The class of models which aim to capture this behaviour can be referred to as models including the non-saturation regime. IEEE 802.11 networks usually operate in unsaturated traffic conditions with significant idle periods. During these periods, a node will perform the post-backoff and subsequently remain idle while the queue is empty. The node will await arrival of a packet in the transmission queue. Because beacons are generated at a very low rate compared to the maximum theoretical throughput of the link, the non-saturated regime is the most relevant for vehicular networking. The absence of idle periods in the saturation models limits their practical applicability [151], and the accuracy of many models is limited to a narrow range of parameters [152].

DCF models

Ho and Chen [127] evaluate the impact of the CA feature of the IEEE 802.11 DCF by using a 2-dimensional Markov Chain. This model focusses on the state the MAC is in. A node can either be idle (it has no packet to send), unbacklogged (it has a new packet to transmit), or backlogged (it has a packet to retransmit). The IEEE 802.11 MAC is modelled as a gated system with no buffer; arrivals after the beginning of the current contention period will not be served until the next contention period.

Malone *et al.* [153] extend Bianchi's model by adding post-backoff states, during which the queue remains empty. This model is used to analyse the Unicast performance of communicating pairs of nodes. Dao and Malaney use a dedicated Idle state for a DCF system in [154]. Felemban and Ekici [152] add bc freezing behaviour to Bianchi's model by using a separate DTMC to model the channel state. This DTMC models the Idle, Collision and Success state the channel can be in while a node is performing backoff, and is used to obtain a more accurate blocking probability for the DCF. The saturation model is adapted for use in the entire saturation range by iteratively deriving a probability P_0 that a node has no packet in its transmission queue from the saturation model. This probability is then used to estimate the probability mass function (PMF) of the number of nodes currently contending as binomially distributed, which in turn is used to obtain a service time estimation. Liu *et al.* [155] explicitly model the queue size in the third dimension of the Markov chain. The effect is that, rather than solving the complex service time distribution and the M/G/1 or G/G/1 queue separately, all the interesting 802.11 DCF system characteristics can be obtained directly from the Markov model.

EDCA models

Work by Engelstad and Østerø [156] extends Xiao's saturation model of the EDCA by adding AIFS differentiation and modifying the model to be representative of the whole saturation range, from non-saturated to fully saturated. In addition, post-backoff behaviour is added, which finally ends up in an Idle state. Whether or not to exhibit saturation behaviour is governed by a probability ρ that the queue contains another packet after transmission completes. This probability is obtained using the arrival rate of beacons and the service time of the MAC using Little's Law.

The model is used to evaluate throughput and estimate starvation of certain ACs. The throughput is evaluated and compared to simulation results. The findings are qualitatively good, with some inaccuracy in the semi-saturation part. It was noted that the numerical calculations in this area have difficulty converging. Throughput is overestimated in this region, but the points where lower ACs experience starvation are found with good accuracy.

In a WLAN situation, most traffic is asymmetric: a client requests a resource from the Internet through the AP, which is subsequently transferred from the AP to the client. Most of the service differentiation in this case is done in the Virtual Collision Handler (VCH) in the AP. Engelstad and Østerø extend the model to include the VCH in [157].

Later, a Z-transform approach is used to derive, in addition to the mean delay, higher-order moments of the delay [158]. The paper provides an evaluation of Delay and Throughput. The results show a similar over-estimation of the throughput, the same convergence problems occur in semi-saturation. Delay has a relatively good match with simulation results, where the delay of lower ACs goes to infinity to accomodate higher priority ACs—a clear sign of the starvation.

[159] analyses queueing delay of the EDCA by obtaining the Second-order moment of the delay by means of the Z-transform. In [160] a slightly modified Pollaczek-Khintchine formula is used to get the Z-transform of the queueing delay, and obtain its complimentary distribution. In [161], Engelstad and Østerø provide an analysis of the total delay as experienced by applications. This includes queueing-, contention-, and transmission delay.

Yang *et al.* [151] identify that the EDCA brings difficulties and complexities to the per-slot based Markov chain modelling techniques widely used for analysis of the DCF. One problem is that, due to AIFS differentiation, it is no longer possible to accurately define a common time scale across all nodes. This common time scale is a fundamental property of many Markov-chain based models. To cope with these problems, [151] uses a Channel access cycle-based modelling approach and adapt this for use in non-saturated conditions.

5.1.3 Modelling of IEEE 802.11p vehicular networks

A large degree of research in the field of vehicular networking is focussed on the challenges of network fragmentation, making routing and multi-hop forwarding very interesting and popular topics for study, see e.g. [162, 163, 164]. Another main challenge is that of large node density, bringing with it the scalability challenges associated with operation in a highly-connected network.

As opposed to IEEE 802.11 as used in WLAN situations where Unicast dominates, the large scale use of Broadcast as envisioned in vehicular networks has received little attention in the early modelling work. In Sec. 3.6.3 of this dissertation, the simple stochastic model proposed by Yang *et al.* in [109] is used. Chen *et al.* [165] analyse IEEE 802.11 broadcast performance using a one-dimensional Markov chain, modelling the *bc* decrementing behaviour of the DCF under saturation conditions. Ma and Chen [166] provide a model for broadcast in VANETs, including the presence of hidden terminals. Their model always performs backoff, exhibiting saturation behaviour without *bc* blocking. Rao *et al.* [167] propose a two-dimensional Markov model for a bufferless 802.11 DCF MAC, which is then extended with a finite queue. However, this model assumes complete independence among nodes. In [168], Vinel *et al.* address the trade-off between generation rate and network performance using deterministic arrivals, assuming backoff prior to transmission–also saturated behaviour. In [108] a Markov Chain is used to model the number of *active* stations and the number of transmissions in a slot under the assumption of a Bernoulli arrival process. An important assumption in [108] is that, when a new beacon arrives in the MAC queue and finds it non-empty, it replaces the current contents.

Campolo *et al.* provide a model for broadcast transmissions on the CCH under IEEE 1609.4 channel switching in [169]. This model has a per-cycle scope: each node will generate a beacon or WSA frame in a CCH period. There is no need to model the MAC queue, because at the end of each cycle remaining frames are dropped and freshly generated at the beginning of the next cycle. Unique to this environment is that, in addition to collision loss, a beacon can also *expire* if it has not been transmitted before the end of the CCH period. Based on a Bernoulli process, a collision probability is derived based on the assumption that beacon generation is synchronised at the start of the CCH period. Loss due to expiration is defined as a probability the frame is not transmitted by the end the CCH period ends. The conclusion is that expiry becomes significant under a large number of nodes ($n > 40$) and a large CW, until then collision loss dominates.

In [68], this analysis is expanded to uniform distribution of generation events to improve time diversity. It becomes evident that the tight synchronisation which occurs under 1609.4 channel switching reduces CSMA/CA's scalability because this temporarily overloads the system, causing a high packet loss probability. These findings are reinforced by the simulation study presented in Sec. 6.4. A suggestion to deal with this synchronisation provided in [68] is to increase CW, which effectively increases time diversity when nodes are synchronised to a common event. This event can be a transmission on the channel, but also expiry of a common timer when using a timeslot-based medium access approach. The latter is commonly referred to as the *timeslot boundary synchronisation* problem [170].

5.1.4 The independence assumption

Most of the models described above are based upon the assumption that a time slot is independent of the status of previous and subsequent time slots. This is not the case in practice, however, because a node's present state depends on what happened in previous slots. Foh and Tantra [171] extend the model by Ziouva and Antonakopoulos [140] by conditioning the transmission probability of a node on the status of the previous time slot.

Felemban and Ekici note in [152] that adding more states or scaling τ to expand the system into the non-saturated regime does not accurately model the system behaviour, because of the assumption that individual nodes are independent. This assumption is also referred to as a decoupling assumption [172]. In [173], it was proven that for random backoff algorithms, when n grows large, the system is indeed decoupled. This provides a theoretical justification for the decoupling assumption on the node level for large n.

Garetto and Chiasserini [174] take depencencies among nodes into account by taking the number of nodes currently in contention into account. However, the complexity of the model impedes its practical applicability—especially under scenarios with a large n. [152] does this by estimating the number of active contending nodes as a binomial distribution, which is then used to scale the collision probability, throughput, and channel access delay metrics.

Whereas the DCF nodes have a single queue and a single backoff state machine using the same parameters (a homogeneous system), [175] identifies that in EDCA, when using multiple ACs, the nodes can have multiple queues, separate backoff state machines and parameters, yielding a nonhomogeneous system. Even though a fixed point can be computed, it is found that there can be multiple unbalanced fixed points, suggesting multistability, which may result in short-term unfairness. Furthermore, [175] mentions that the fixed point approach is a heuristic that is found to work well in some cases.

5.1.5 The notion of saturation

Saturation, as used in the Bianchi-style models, refers to the state of the queueing station; the server is continuously serving arrivals. By assuming the queue always has a packet, the model can be simplified because the state of the transmission queue does not need to be modelled. If a station finds a new packet waiting for transmission after performing the previous transmission, the system is called saturated. This situation depends on two factors: (1) the number of nodes in the system, which governs a node's ability to serve arriving packets sufficiently quick (i.e., the service rate and service time, see Sec. 5.3.6), and (2) the arrival rate of the packets. If a small number of stations has a very high arrival rate (like in a typical WLAN situation as studied by Bianchi, Engelstad and others) a station has to contend with few others. In the extreme situation, a single transmitting node has an arrival rate high enough to saturate its queue. It waits an IFS, counts down post-backoff, transmits, and repeats. Queueing delay will grow excessively, whereas contention delay does not: there are no other stations to cause its *bc* to be frozen.

The channel can be saturated as well, but this does not necesarrily coincide with a saturated transmission queue. The single saturated station used as an example above still allows plenty of idle time on the channel by means of its PBO. Other stations can then find the medium idle and perform a direct transmission. The regime changes when we have many nodes with a low generation rate (as typical for beaconing scenarios). A node can experience periods of inactivity between successive packet arrivals in its queue. The channel will then be saturated before the transmission queue is.

5.2 A model for beaconing in vehicular networks

Reasoning in line with the discussion on the metrics presented in Sec. 3.5, we can model a beaconing system as a chain of components, where each component models a specific type of loss and delay, summing up to the total. A communication model such as this can, in turn, be embedded in models which evaluate the traffic context. This can be in simulation studies like [176], or model-based approaches such as described in [177].

Fig. 5.1 illustrates the high-level model of a single node, in which the detailed MAC model is embedded. Beacons travel from the *generator* (left) to the *application*

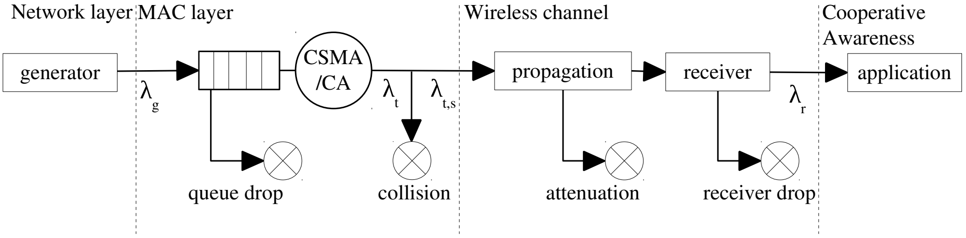


Figure 5.1: The beacon model for broadcasting in vehicular networks.

(right). They are generated at the network layer at rate λ_g and contain application data from which we abstract here. Beacons enter the *transmission queue* and will then be served by the IEEE 802.11p CSMA/CA MAC. The MAC takes a frame from the queue and then serves it: after contention, the frame will be transmitted. Beacons then propagate through the wireless channel, as modelled in the *propagation* component. The *receiver* component models whether a received signal can be succesfully decoded. This high-level model allows adding more detail because of its modularity.

Note that not all generated beacons will be received succesfully, because loss can occur at several stages in the process:

1. Packets can be dropped in case of a full queue. Queue drop is evaluated in Sec. 6.3.5.

2. In Unicast, frames are dropped by the MAC when the maximum number of retries has been reached. Broadcast frames are never dropped for this reason, hence, this type of loss is absent in Fig. 5.1.

3. Transmissions can collide because of the inability of the MAC to coordinate transmissions among nodes. This is evaluated throughout this dissertation as P_s, in particular in Sec. 5.5 and the case studies in Chapter 6. The effect of P_s on the CACC application was evaluated in Sec. 4.5.

4. Propagation loss is the result of attenuation of the transmitted signal. This could be modelled with a simple path loss expression which decays with distance, such as the Friis equation [178]. In this evaluation, all nodes are assumed to observe the channel in the same way. Note that this assumption coincides with Bianchi [128], and relevant other studies mentioned in Sec. 5.1.

5. Beacons can be lost at the receiver because of interference with hidden nodes.

In addition to a loss probablity in each component, the model also allows for end-to-end delay calculations to be obtained by summing the delay accumulated at the various components from generator to application. The resulting receive rate λ_r, and latency of the received information are important measures of interest, since they determine the quality of the ITS applications which use the information in the cooperative awareness.

Though attenuation and collisions with hidden terminals are surely limiting factors in the performance of a beaconing system [76], the focus of this work is on the MAC layer, and its ability to coordinate transmissions.

The MAC is modelled as an $M/G/1$ queueing station with successive beacon arrivals according to a Poisson process, i.e., exponentially distributed inter-arrival times with mean $1/\lambda_g$. The MAC consists of an infinite First In, First Out (FIFO) transmission queue and a single medium access "server", the behaviour of which is modelled in more detail in Sec. 5.3 and Sec. 5.4. This server cannot be assumed to have a negative exponentially distributed service time, but follows a general service time distribution with mean $\mathbb{E}[S]$, which is derived from the embedded detailed MAC model.

A mean-field approximation or interaction model [172] can be used to study the behaviour of large and complex stochastic models, such as n interacting MAC entities, by studying a simpler model. This approach allows a large number of entities which interact with each other through a common resource to be studied by approximating the single averaged effect of all other entities on any given entity. The underlying assumption is that a model of a single node exists in an independent environment, generated by other nodes of the same type [179]. In order to attain the independence to apply this modelling technique, an infinite queue and Poisson arrivals are required.

First, a model for the DCF MAC will be presented in Sec. 5.3, because the DCF MAC is the basic access method of IEEE 802.11. Sec. 5.4 then describes a model for the EDCA MAC, which is based on the DCF model and proposed for use by IEEE 802.11p [2]. After the MAC models are described, they will be validated in Sec. 5.5, to allow comparison in Sec. 5.6.1.

5.3 Modelling the DCF MAC

The DCF MAC will be modelled using a DTMC, the state space of which will be described in Sec. 5.3.2. This DTMC can be used to obtain metrics of interest, such as the probability that a node will transmit, after it has been solved. To achieve this, we obtain the steady state distribution in Sec. 5.3.3. The steady state equations are based on transition probabilities, which depend on several model variables. These are described in Sec. 5.3.4, and depend again on the solution of the DTMC through for instance the probability that a node will transmit. This chicken-and-egg problem can be solved by iteratively solving the resulting system of equations. These fixed-point iterations will converge to a solution, if a fixed-point exists.

The model variables are expanded upon in Sec. 5.3.5, where the probability of beacons entering the transmission queue is expressed. The service time of the MAC is obtained in Sec. 5.3.6, which depends on the collision multiplicity covered in Sec. 5.3.8 and the streak length in Sec. 5.3.7.

5.3.1 Preliminaries

As described in Sec. 2.5, the IEEE 802.11p standard prescribes the EDCA extensions for QoS differentiation. The EDCA is based on the DCF described in Sec. 2.4, and it is interesting to compare the two access methods for beaconing. In order to compare the DCF and the EDCA, we first model the DCF without the EDCA extensions. Later, in Sec. 5.4, we adapt the DCF model described here to comply with EDCA.

A vehicular network is a very dynamic environment which can comprise many nodes, but the amount of traffic generated by a single node is lower than a typical WLAN station. In order to correctly model it one should consider both saturation and non-staturation regimes. The model by Engelstad and Østerbø [156] provides a good starting point because it covers the entire saturation spectrum.

In the saturation case, a node always finds a packet in its queue to transmit. After transmission of one frame it will immediately start contention for the next frame. In the non-saturation case, it may happen that a node finds the queue empty after transmission. It will perform post-backoff (PBO) and subsequently remain IDLE, until a packet enters the queue, or slide from PBO into backoff (BO) when a packet arrives during PBO. The model abstracts from the actual number of packets in the queue, and only considers the queue to be either empty or non-empty.

The DCF is modelled as a DTMC that is a modified version of the one presented in [156, 159]. Time is discretised into generic slots. A slot is either idle or busy; a busy slot is either successful or a collision. In reality, these slots have different durations. We implicitly assume a node is always listening when it is not transmitting. This assumption is vital to the concept of cooperative awareness—not only does a node need to transmit its own beacons, it must also receive those sent by others.

The model in [156] allows for traffic within different ACs, including AIFS and CW differentiation. This work focusses on single-class beaconing and does not include traffic of other classes. Furthermore, [156] models multiple retries and CW increase per retry as defined by the standard. For a beaconing system using 802.11p we consider only Broadcast transmissions. Hence, there are no retransmissions and no CW increments.

To model Broadcast-specific behaviour, explicit differences from [156, 159] are presented in the following equations. In (5.16), q is adapted to more accurately model the packet arrival probability while IDLE. In (5.21), the bc blocking probability is improved. The service time of the MAC is obtained through (5.23), which is determined using the concepts of Streak Length in (5.29) and the Collision Multiplicity in (5.30). Foundational to this behaviour, is the way bc decrementing is performed in the DCF with strictly Broadcast transmissions.

5.3.2 State space

Fig. 5.2 shows the DTMC model for a beaconing node. The state space S of the DTMC consists of a finite set of states $\mathcal{S} = \{s_{j,k} | j \in \{0, 1\} \land k \in \{0, \ldots, W - 1\}\}$, where $j = 0$ holds for a node that is currently not in the process of accessing the medium (it is either in PBO or IDLE) and $j = 1$ means that the node is contending

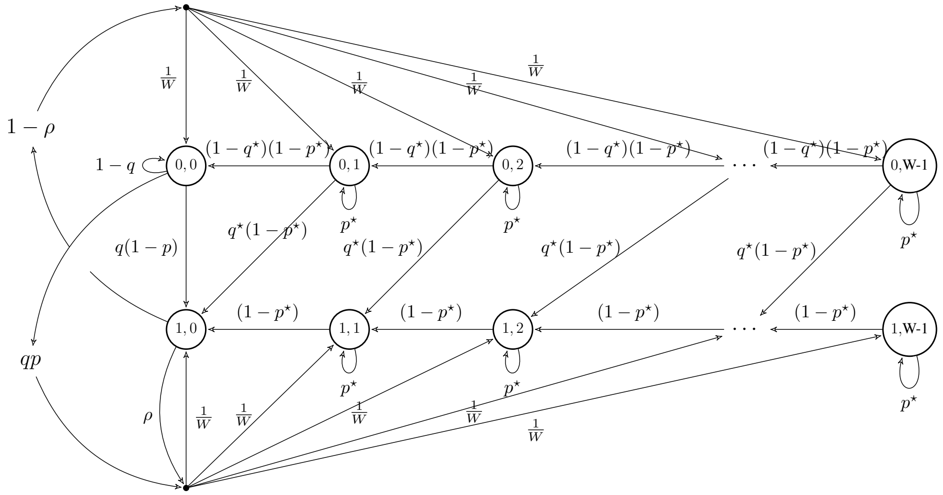


Figure 5.2: The Markov chain of the DCF model.

for medium access (BO), or actually transmitting. Parameter k denotes the current bc value, randomly choosen according to a uniform distribution when (1) a station takes a packet from the queue and starts its medium access attempt and finds the medium busy, or (2) when a station starts PBO.

When the bc reaches zero ($s_{1,0}$), the node transmits the frame it is currently serving. After transmission, with probability ρ the station finds another packet in its queue and will perform a new BO for medium access. With probability $1-\rho$ the queue is empty and the node will enter PBO. While in PBO, the bc is decremented for every empty timeslot, until the system reaches $s_{0,0}$. If a transmission by an other node is overheard (with probability $p^\star$) the bc is frozen. Countdown resumes when the channel turns idle again, with probability $1-p^\star$.

During PBO, with probability $q^\star$ a frame enters the transmission queue. The bc countdown will continue, in order to access the medium. This is modelled by the "diagonal transitions" in Fig. 5.2. While in BO (states $s_{1,k}$ for $k \in \{1 \ldots W-1\}$), new arrivals in the transmission queue are not considered, because the station is already serving a frame. The number of packets currently in the queue is not explicitly modelled. Note, however, that ρ accounts for arrivals which occur during the service time. When a node reaches $s_{0,0}$, which represents an IDLE node, it receives a packet in its transmission queue with probability q or remains IDLE with probability $1-q$. A node perceives the channel busy with probability p, hence, will perform a BO with probability qp, or a direct transmission with $q(1-p)$. In the latter case the carrier-sensing found the channel idle and transmission is allowed immediately. Both the direct transmissions and those mediated by BO will transition into $s_{1,0}$: the transmission state.

5.3.3 Steady state distribution

Let $b_{0,k}$ and $b_{1,k}$ denote the stationary probability of being in states $s_{0,k}$ and $s_{1,k}$, respectively, for the DTMC shown in Fig. 5.2. By working recursively from right to left, the following expression can be derived (a complete derivation is provided in Appendix B). The steady-state probability for a node in PBO, expressed in terms of $b_{1,0}$, is given by:

$$b_{0,k} = \frac{(1-\rho)b_{1,0}}{W(1-p^\star)} \frac{1-(1-q^\star)^{W-k}}{q^\star}, \text{ for } k = 1, \ldots, W-1. \tag{5.1}$$

The steady-state probability for a node that has completed PBO and is IDLE is given by:

$$b_{0,0} = \frac{(1-\rho)b_{1,0}}{Wq} \frac{1-(1-q^\star)^{W}}{q^\star}. \tag{5.2}$$

The steady-state probability for a node that is in BO to get access to the medium is given by:

$$\begin{aligned} b_{1,k} =& \frac{(1-\rho)b_{1,0}}{W(1-p^\star)} \left((W-k)\left(1 + \frac{\rho}{1-\rho} + \frac{p}{W}\frac{1-(1-q^\star)^W}{q^\star}\right) - \right. \\ & \left. \frac{1-(1-q^\star)^W}{q^\star} \right), \text{ for } k = 1, \ldots, W-1. \end{aligned} \tag{5.3}$$

We express the probability that a node will transmit in a generic slot as τ, which equals the probability of being in $s_{1,0}$. We can now find $b_{1,0}$ by normalising the above three expressions for $b_{j,k}$. The result expresses the probability that a node will transmit the frame in a generic slot:

$$\tau = b_{1,0} = \left(1 + \frac{(W-1)}{2(1-p^\star)} + \frac{(1-\rho)}{q} \frac{1-(1-q^\star)^W}{Wq^\star} \left(1 + \frac{(W-1)qp}{2(1-p^\star)} \right) \right)^{-1}. \tag{5.4}$$

Similar to (10) in [156], the first two terms in (5.4) dominate the saturation case, the second part the non-saturated case. The factor $(1-\rho)$ represents the probability that the queue is empty after a transmission. In this case, the system will enter PBO which is modelled by the geometric sum $\frac{1-(1-q^\star)^W}{Wq^\star}$. This expresses the probability that the queue remains empty during the PBO and we finally reach state $s_{0,0}$ instead of moving to the regular BO procedure that is represented by states $s_{1,0}, \ldots, s_{1,W-1}$. The last factor in (5.4) is quite vital, as it models 802.11's Carrier Sensing in state $s_{0,0}$; whether or not to perform a BO prior to transmission, or directly transmit.

We now have an expression for τ, the probability that a node transmits in a generic slot.

5.3.4 Model variables

The transition probabilities in the DTMC in Fig. 5.2 depend on q, p, $q^\star$, $p^\star$ and ρ. The resulting system of equations expresses the behaviour of a single node. In the following sections, these probabilities will be defined. During a slot, the medium has a certain status with a certain probability, determining the type of the slot. These probabilities are p_b, p, p_s, p_c and p_e. The expressions for these probabilities will be treated next, and are based on an independence assumption for the behaviour of a single node to be Markovian.

Let n be the number of coordinated nodes, i.e., those nodes which are in each other's carrier sensing range, in the system under consideration. The probability that the medium is busy in a generic slot, p_b, is the probability that *at least* one node transmits in a slot, i.e., 1 minus the probability none transmits:

$$p_b = 1 - (1-\tau)^n. \tag{5.5}$$

When a node is sensing the channel, it encounters a busy slot with probability p; at least 1 of the remaining $n-1$ other nodes is transmitting:

$$p = 1 - (1-\tau)^{n-1}. \tag{5.6}$$

The probability p_s that a generic slot contains a successful transmission can then be expressed as n-times the probability a node is transmitting, while the $n-1$ others are not:

$$p_s = n\tau(1-\tau)^{n-1}. \tag{5.7}$$

This is also the fraction of slots out of all slots in which a successful transmission is performed. The probability that a slot contains a collision is given by:

$$p_c = p_b - p_s, \tag{5.8}$$

and the probability that a generic slot is an empty slot is subsequently given by subtracting the probability a slot contains a successful transmission from the probability a slot contains a transmission at all:

$$p_e = 1 - p_b. \tag{5.9}$$

Let T_e be the duration of an empty slot, T_s the duration of a successful transmission and, T_c the duration of a collision, then:

$$\begin{aligned} T_e &= \texttt{aSlotTime}\ \sigma, && (5.10)\\ T_s &= \texttt{T}_{\texttt{phy}} + \texttt{T}_{\texttt{mac}} + L/R + \texttt{DIFS}, && (5.11)\\ T_c &= \texttt{T}_{\texttt{phy}} + \texttt{T}_{\texttt{mac}} + L/R + \texttt{EIFS}. && (5.12) \end{aligned}$$

T_s and T_c consist of a signal part and a guard space part. The former depends on physical- and MAC-layer headers and the payload length L which is transmitted at data rate R. The latter is the IFS, which is defined by the standard [2] to separate contiguous signals on the channel, described in detail in Sec. 2.4.3.

The EIFS is used after an *observed* collision; this excludes nodes whose transmission is part of the collision, and is defined as:

$$\texttt{EIFS} = \texttt{SIFS} + \texttt{T}_{\texttt{phy}} + \texttt{T}_{\texttt{ack}} + \texttt{DIFS}. \tag{5.13}$$

As a result, not all nodes use the EIFS in response to a collision. This behaviour is not reflected in the model and may introduce minor inaccuracy when many nodes are involved in a collision. Now we can express the mean slot duration as:

$$\mathbb{E}[T] = p_e T_e + p_s T_s + p_c T_c. \tag{5.14}$$

and the mean duration of a busy slot as:

$$\mathbb{E}[T_b] = \frac{p_s}{p_b} T_s + \left(1 - \frac{p_s}{p_b}\right) T_c, \tag{5.15}$$

5.3.5 Modelling packet arrivals

A generic slot in the model is composed of empty and busy slots. The busy slots are divided into successful and collision slots, each with their effective duration, see (5.10)–(5.12). The probability of receiving an arrival depends on the type (hence, duration) of a slot in which an arrival can occur.

Arrivals while in IDLE

When a node is in $s_{0,0}$, i.e., when it is IDLE, a packet arrival occurs in a generic slot with probability q. Because a generic slot can be empty, contain a successful transmission, or a collision, an arrival occurs in one of these slots with different probability depending on the type of slot. The mix of these types depends on the probability of occurence of such slots. For example, the probability of receiving a packet during an empty slot is $1 - e^{-\lambda_g T_e}$. The probability q that a packet arrival occurs in a generic slot, is given by a weighted Poisson arrival process with parameter λ_g:

$$q = 1 - \left((1 - p_b^\star) e^{-\lambda_g T_e} + p_s^\star e^{-\lambda_g T_s} + (p_b^\star - p_s^\star) e^{-\lambda_g T_c}\right), \tag{5.16}$$

where

$$p_s^\star = (n-1)\tau(1-\tau)^{n-2} \tag{5.17}$$

is the probability that the node under consideration observes a slot containing a successful transmission from one of the $n - 1$ others. That means that out of these other nodes, $n - 2$ do not transmit. The probability of observing a busy slot is obtained likewise, as:

$$p_b^\star = 1 - (1-\tau)^{n-1}. \tag{5.18}$$

The similarity with p is no coincidence, as both reflect a node observing a system of $n-1$ other nodes. The difference with the expression for q in [156] is that (5.16) considers a system of $n-1$ nodes, whereas [156] considers n nodes. This is incorrect, because one node is the subject for which this expression is evaluated, hence, it observes $n-1$ other nodes. Our expression in (5.16) is more accurate (especially for small n) because it models arrivals based on the state of the $n-1$ other nodes, given that the node under consideration is IDLE.

Arrivals during a busy slot

The probability of an arrival during a busy slot is given by:

$$q_b = 1 - \left(\frac{p_s}{p_b} e^{-\lambda_g T_s} + \left(1 - \frac{p_s}{p_b}\right) e^{-\lambda_g T_c} \right). \tag{5.19}$$

This probability is required in (5.26)–(5.28), where the state of the channel is conditioned on the state in a previous slot in order to obtain an estimator for the blocking duration.

Arrivals while in PBO

The probability of an arrival while in a PBO state, $q^\star$, was found in [158] to yield good accuracy when $q^\star = q$ was used for simplicity. In [160], a more detailed expression for $q^\star$ was derived:

$$\begin{aligned} q^\star &= \mathbb{P}(\text{ at least one arrival occurs during } bc \text{ blocking}) \\ &= 1 - \mathbb{P}(\text{ no arrival occurs during } bc \text{ blocking}) \\ &= 1 - \frac{\mathbb{P}(\text{ no arrival during } T_e)}{\mathbb{P}(\text{ no arrival during } T_b)} \\ &= 1 - \frac{(1-p^\star)e^{-\lambda_g T_e}}{1 - p^\star \left(\frac{p_s}{p_b} e^{-\lambda_g T_s} + \left(1 - \frac{p_s}{p_b}\right) e^{-\lambda_g T_c} \right)}. \end{aligned} \tag{5.20}$$

A PBO state includes the duration of an empty slot, plus the weighted duration of being blocked. This duration is weighted by the probability of facing an empty slot (with probability $(1-p^\star)$) over the probability of facing a busy slot ($p^\star$). No arrival should occur during the empty slot prior to being blocked, and no arrival should occur while blocked. Then, 1 minus this probability yields the probability of getting at least one arrival during one PBO state.

The $p^\star$ present in both numerator and denominator already relate to a system of $n-1$ nodes (see (5.21)). Like in [156], $p^\star$ is the probability of observing the channel busy during (P)BO. We do not model the AIFS differentiation because of the single-AC assumption. However, as also observed in [152], many models do not accurately model the backoff freezing process of the DCF. The expression provided in [156],

$p^\star = p$ without AIFS differentiation, yields increasingly inaccurate results for larger n. The reason for this is that transmissions occur in streaks with increasing number of nodes, and are not independent of each other as frequently assumed in literature. In [144] this is remedied by changing the time slots of the model. Our approach is to derive a slightly different expression for $p^\star$.

Based on the Streak Length described in Sec. 5.3.7, a better approximation of the blocking duration can be obtained. While in $s_{j,k}$ for $k \in \{1 \ldots W-1\}$, the progress towards $k-1$ could be seen as a series of Bernoulli trials, where success means to decrement the bc and failure to perform a self-loop. In (5.29) in Sec. 5.3.7 we approximate the Streak Length with a geometric distribution, and find $\frac{p}{1-p'}$ to be a good estimator of the mean Streak Length. Let $p^\star$ be the probability of staying in $s_{j,k}$: the channel is observed busy, possibly for consecutive slots. Then $p^\star$ should be defined such that:

$$\frac{p^\star}{1-p^\star} = \frac{p}{1-p'} \implies p^\star = \frac{p}{(1-p')+p}. \tag{5.21}$$

5.3.6 Service time

We can now derive an expression for the service time of the medium access server. This is the sum of the time it takes to transmit a frame (including the IFS), and the time spent in contention. Recall that whether or not to perform contention depends on the state of the channel upon arrival of a packet. The probability that a slot is observed busy is expressed as p, see (5.6). However, arrivals can happen at random moments in time (and not only on slot boundaries), so following the PASTA property, these Poisson arrivals see time averages. In this line, we need to find the observed real-time channel utilisation (μ) by multiplying the probability of encountering a busy slot with the duration of such a slot, and divide by the duration of a generic slot:

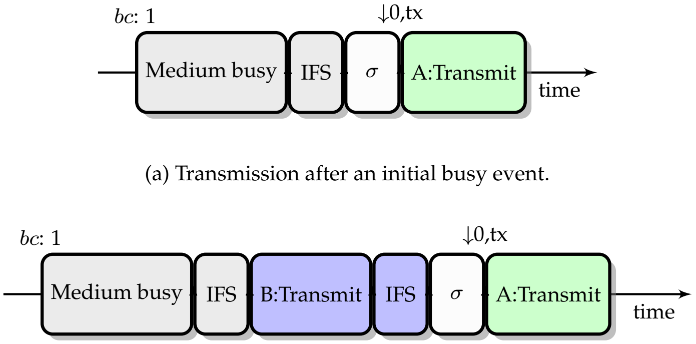
$$\mu = \frac{p\mathbb{E}[T_b]}{\mathbb{E}[T]}. \tag{5.22}$$

The expected service time $\mathbb{E}[S]$ can then be obtained as follows:

$$\mathbb{E}[S] = \mathbb{E}[T_b] + \mu\left(\frac{\mathbb{E}[T_b]}{2} + \frac{(W-1)}{2}\big(T_e + \mathbb{E}[T_b]\mathbb{E}[L]\big)\right). \tag{5.23}$$

Transmission of a message, including the IFS, has a duration of $\mathbb{E}[T_b]$. A station observes the medium busy and has to perform BO with probability μ. In this case, the event which caused it to BO has a mean remaining duration of $\frac{\mathbb{E}[T_b]}{2}$, after which BO can commence. On average, the station has to count down $\frac{(W-1)}{2}$ empty slots, and for every bc decrement there is a probability of finding the medium busy, freezing the bc. The duration of this freeze is the expected Streak Length $\mathbb{E}[L]$ (expressed in (5.29), see Sec. 5.3.7) multiplied by the mean duration of a busy slot.

Then, Little's Law can be used to obtain ρ:



(a) Transmission after an initial busy event.

(b) Transmission after an additional busy event. Note that the station waits for an empty slot before decrementing its bc and subsequently transmitting.

Figure 5.3: The channel divided into slots in DCF.

$$\rho = \lambda_g \mathbb{E}[S]. \tag{5.24}$$

[156] considers the PBO as part of the service time, and a correction is required to obtain ρ for the non-saturated regime. Because of our definition of $\mathbb{E}[S]$, we can directly use Little's Law.

5.3.7 Streak length

The DCF prescribes the bc to be decremented after an empty slot σ has been observed by a station, as described in Sec. 2.4.2. When the bc reaches zero, transmission takes place. This behaviour is illustrated in Fig. 5.3(a) where the actions relating to decrementing or transmitting are illustrated on top of the blocks. In this example, node A finds the medium busy and chooses a bc of 1. As soon as the medium turns idle for an IFS and the first empty slot has passed, bc is decremented to zero and node A transmits.

If another node, node B, had choosen $bc = 0$ at the same time, it transmits immediately after the IFS, as illustrated in Fig. 5.3(b). Node A with bc=1 now has to wait an additional transmission (plus the mandatory IFS) before it can decrement its bc, and then transmit. Note how these first two transmissions are following up eachother without intermediate σ. This consecutive series of transmissions without intermediate empty slots is referred to as a *streak* or *consecutive freeze process* [165].

The length of such a streak, denoted as L, is quite fundamental to the behaviour of the DCF, especially with increasing n. In [156] and [159], the share of slots where the countdown process is being blocked is expressed as follows under the assumption that a transmission is independent of previous or subsequent transmissions:

$$\text{mean number of blocked slots} = \frac{p}{1-p} \tag{5.25}$$

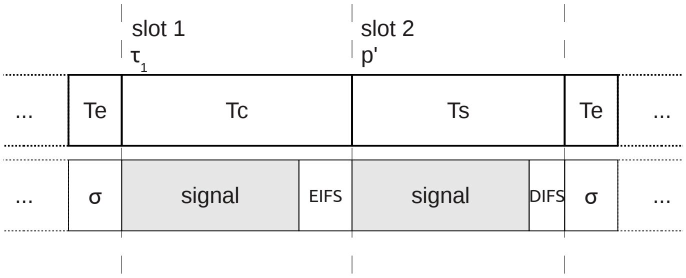


Figure 5.4: Streak length as modelled (top) and as on the physical channel (bottom).

This is a series of Bernoulli trials, finding the channel busy with probability p. This fraction follows from the mean of the geometric distribution and can be read as "number of successes until failure" (with a busy channel being success). However, transmissions are not independent because of the bc decrementing behaviour of the DCF, and the resulting streaks. The following will describe why. The resulting sequence of slots looks like the top part of Fig. 5.4. The lower part reflects the way a streak manifests on the channel. This process works as follows:

1. The medium is idle for an empty slot, indicated by T_e and σ in Fig. 5.4. All nodes currently in contention decrement their bc.

2. Nodes with $bc = 0$ transmit in slot 1 with probability τ_1, possibly causing a collision.

3. Next, in slot 2, nodes which had $bc \geq 1$ in slot 1 are *not* allowed to transmit. These nodes require an empty slot to decrement their bc. However, there are two groups of nodes which *can* transmit with a certain probability:

$\mathbb{G}_{TX}$ A node has performed a transmission in the previous slot. It finds its queue non-empty and chooses $bc = 0$ with probability $\frac{1}{W}$.

$\mathbb{G}_{IDLE}$ A node was IDLE in the previous slot, its queue was empty. At some moment during the previous slot, a packet arrives in the queue. The node senses the medium busy and chooses $bc = 0$ with probability $\frac{1}{W}$. In the next slot (after the IFS) it will transmit.

This process can repeat, and is more likely to repeat for larger n, resulting in multiple consecutive busy slots. Note the large influence of the CW size: if W is chosen large with respect to the number of nodes, the streak length will be reduced because the fraction of nodes which chooses a bc of zero will decline with each slot. Note that, in contrast to [144], we also include nodes which were IDLE while the medium was busy. Intuitively it is clear this will be a larger group for low λ_g as typical for beaconing; the number of nodes currently transmitting will be lower than the number of nodes currenlty IDLE.

The streak length has great impact on the duration a node finds its backoff countdown blocked: while other nodes gain access to the channel, those awaiting an empty slot accumulate contention delay. This is an important input in the modelling of the MAC, because an underlying assumption in many current models is that transmissions are independent. The manifestation of this streak behaviour clearly shows that the assumption of independence does not hold for large values of n, where the geometric distribution shows a particularly heavy tail. Hence, the blocking probability is not accurately estimated by (5.25). This is because of an increase in the number of nodes which collide in the first slot of a streak, causing many nodes to perform their service in parallel. This lowers the effective load in the remaining slots of this streak, and reduces the bc blocking duration as modelled in [156], because the transmissions are compressed into fewer slots. This effect is not visible in [156, 159], and [145] because these works do not consider a number of nodes which is large enough to show this effect, nor is it present in the models in [168] and [108], although the model by Felemban and Ekici [152] takes this effect into account.

The denominator of the share of slots where bc blocking occurs in [156] as given in (5.25) cannot be found as the mean of the geometric distribution, because of its dependency on what happened in the preceding slot. Hence, let p' be the probability that a slot is observed busy, given that the previous slot was busy. Then this situation refers to the second or later slots in a streak (see also Fig. 5.4). We express p' as 1 minus the probability that no nodes wil transmit in this situation, yielding:

$$p' = 1 - (1 - \Psi_{TX})(1 - \Psi_{IDLE}). \tag{5.26}$$

In these slots, a transmission can come from one of the two groups $\mathbb{G}_{TX}$ and $\mathbb{G}_{IDLE}$ as described above. Then Ψ_{TX} and Ψ_{IDLE} are the probabilities that at least one node from these groups will transmit. These probabilities can be obtained as follows:

$$\Psi_{TX} = \mathbb{E}[CM_1]\rho\frac{1}{W} \tag{5.27}$$

$$\Psi_{IDLE} = (n-1)b_{0,0}\frac{q_b}{W}. \tag{5.28}$$

In (5.27) we multiply the number of nodes which have performed a transmission in slot 1 of a streak, estimated using the concept of collision multiplicity (5.30), with the probability of finding the queue non-empty, and then choosing $bc = 0$ which happens with probability $\frac{1}{W}$. In (5.28) we perform a similar operation: multiply the number of nodes currently IDLE with the probability of receiving an arrival during a busy slot, *and* choosing $bc = 0$. We can now define the number of consecutive blocked slots as the mean streak length:

$$\mathbb{E}[L] = \frac{p}{1-p'}. \tag{5.29}$$

Essentially, this is the share of busy slots per empty slot while counting down the bc, taking the streak phenomenon into account.

5.3.8 Collision Multiplicity

In a beaconing system with an increasing n, it is increasingly likely that more nodes will be involved in collisions. The *collision multiplicity* (CM) expresses the number of nodes involved in a collision, as observed by a node. Observing $n-1$ other nodes from a certain node, we express the mean CM in slot s as a summation of the probability that the number of concurrent transmitters is i out of $n-1$.

The probability that i out of $n-1$ nodes transmit is approximated by a binomial distribution. The mean CM is then found as the weighted mean of all possible integer values $i \in [1, \cdots, n-1]$. The mean CM in slot s is given in (5.30). The fraction ensures that we only consider busy slots, where $i \geq 1$.

$$\begin{aligned}
\mathbb{E}[\mathrm{CM}_s] &= \sum_{i=1}^{n-1} i \cdot \mathbb{P}(i \text{ nodes transmit in this slot}) \\
&= \sum_{i=1}^{n-1} i \cdot \binom{n-1}{i} \tau_s^i (1-\tau_s)^{n-1-i} \cdot \frac{1}{1-(1-\tau_s)^{n-1}} \\
&= \frac{(n-1)\tau_s}{1-(1-\tau_s)^{n-1}}. \qquad (5.30)
\end{aligned}$$

The transmission probability of a node in slot s is expressed as τ_s, which is assumed to equal τ for $s > 1$. The first slot in a streak is a special case because it is known that the previous slot was empty. We obtain a special τ for this first slot, which we indicate as τ_1 to compute the CM for slot 1 in (5.27). τ_1 is the probability that a slot is busy, given that the previous slot was empty. This probability can be directly derived from the DTMC in Fig. 5.2. By dividing all paths leading to $s_{1,0}$ by the probability of *not* transmitting, we obtain:

$$\tau_1 = \frac{b_{1,1} + b_{0,1}q^\star + b_{0,0}q}{1 - b_{1,0}}. \qquad (5.31)$$

5.3.9 Relation between collision multiplicity and streak length

A high CM in a slot results in a large Ψ_{TX} in the next slot, but after these nodes have performed their transmission they have a large probability of becoming IDLE with an empty queue, or requiring an empty slot before transmitting. The result is that CM decreases with every slot in a streak.

Because fewer nodes are involved in the transmissions later in the streak, these transmissions are more likely to be collision-free. In other words, they have a lower mean CM. With a larger L, the duration of time in which the other nodes can receive an arrival of a newly generated beacon into their transmission queue increases. This implies that, after a long streak, more nodes will have a frame to transmit. A fraction of $\frac{1}{W}$ of these nodes will then transmit in the first slot after an empty slot.

These slots with high CM *absorb* the load supplied to the system, yielding (momentarily) lower loads in slots $s > 1$. Note that colliding transmissions, to the MAC "server", are still considered as serviced: the packets leave the system.

For increasing s, an increasing subset of nodes is not contending for slot $s + 1$: the nodes which were part of $\mathbb{G}_{TX}$ and $\mathbb{G}_{IDLE}$ have a large probability of *not* being eligible for a transmission opportunity during the current streak after they have performed their first transmission. These nodes enter a situation from which they can only transmit after one or more empty slots. This explains why, even under heavy saturation on the channel, some frames still get through succesfully.

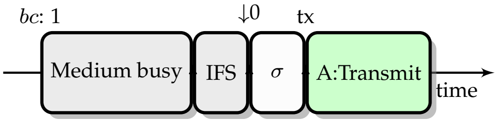
The above effects increase the load scalability of the system: instead of collisions in many slots, all collisions are "piled up" in a few slots. Another important thing to note here, is that the frames which have a high probability of colliding under this regime are the ones undergoing backoff. The frames transmitted by nodes $\mathbb{G}_{TX}$ and $\mathbb{G}_{IDLE}$ are all new arrivals which receive a "direct transmission" treatment, as explained above, resulting in very low latency for these beacons. This combination of high P_s and low delay for these nodes causes a collision bias is responsible for an actual *drop* of observed end-to-end delay at the receiver, as will be discussed in Sec. 5.5, and which will also be visible in the simulation study presented in Sec. 6.2.

5.4 Modelling the EDCA MAC

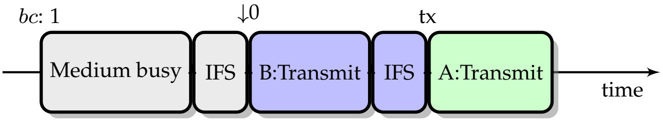
The EDCA was described in Sec. 2.5.1. Many early models of the EDCA mechanism did not model the complete standard, as described in Sec. 5.1. Incrementally, features were added in the various models. The model by Engelstad and Østerbø on which our model is based models AIFS differentiation by means of a scaled bc decrement probability [156], but is essentially DCF decrement behaviour with AIFS differentiation. Without the differentiation, the bc freeze probability equals the probability of observing a busy channel, or $p^{\star}$ in terms of the DCF model in Sec. 5.3.

In order to derive a model for beaconing using the EDCA, we use the DCF model from Sec. 5.3 as a basis, and describe the differences here. We assume all beacons have the same AC. As a result, the model does not include EDCA's AIFS and CW differentiation features.

There is a difference between EDCA and DCF which does not seem very fundamental but has a surprisingly strong impact on beaconing. This difference lies in the backoff counter decrement rules in EDCA, which differ from those used in DCF [2]. The way bc decrementing is treated in the EDCA has great influence on its behaviour, especially under increasing n. This is treated in Sec. 5.4.1, prior to defining the state space of the EDCA model in Sec. 5.4.2 and deriving the steady state distribution in Sec. 5.4.3. Because the bc decrementing rules have great influence on the blocking duration, the service time of the MAC differs from that of the DCF, and is defined for EDCA in Sec. 5.4.4.



(a) Transmission after an initial busy event. A slot boundary occurs after the IFS and after an empty slot. Note that the behaviour in terms of slots is equivalent to that of the DCF in Fig. 5.3(a).



(b) Transmission after an additional busy event, boundary after AIFS and second AIFS. Note that there is no empty slot required, like in Fig. 5.3(b).

Figure 5.5: The channel divided into slots in EDCA.

5.4.1 Backoff counter decrementing in EDCA

Whereas DCF decrements the bc after an empty slot σ and is allowed to transmit at the moment the bc reaches 0, the EDCA decrements the bc at slot boundaries. In this case, the bc is decremented at the beginning of the timeslot immediately following the IFS (either AIFS or EIFS), irrespective of the channel status in that slot [148]. However, it is not allowed to decrement the bc *and* start transmission simultaneously at the same slot boundary. This means that an EDCA node has to wait for the next slot boundary to transmit, once its bc reaches zero.

At a slot boundary, a station shall perform one, and only one, of the following actions [83]: start a transmission, decrement the bc, invoke the backoff procedures for an internal collision (VCH), or do nothing. Figs. 5.5(a) and 5.5(b) illustrate the same scenarios as covered for the DCF in Sec. 5.3 (in Figs. 5.3(a) and 5.3(b)), but now for the EDCA. Note the difference in when bc decrement takes place.

In scenario 1 (comparing Figs. 5.3(a) and 5.5(a)) the behaviour of DCF and EDCA is equivalent: even though the bc is decremented at the end of the IFS, node A is allowed to start transmission after the empty slot. Additionally, if the initial bc is chosen to be 0, both DCF and EDCA are allowed to transmit immediately after the IFS [148].

In scenario 2 (comparing Figs. 5.3b and 5.5b), it is clear that an EDCA station is able to access the medium earlier than a DCF station: it does not require an empty slot because it decrements at the IFS boundary after the first transmission. Station A can proceed with transmission after the next slot boundary, which is after the IFS following B's transmission. This provides EDCA a bc decrement advantage over DCF [148, 180]. Note, however, that an IFS needs to pass for the *current* AC. This means that lower priority ACs, which require a longer IFS, will not perform one of the four actions because they do not encounter a slot boundary if there was a node contending for an AC of higher priority. This node is able to claim the channel sooner and may start transmission before the node with lower priority traffic has experienced an IFS of adequate duration. This explains the starvation experienced by nodes contending for low priority frames [156]; they do not get the opportunity to decrement their bc. This form of starvation does not manifest with traffic of the same AC.

5.4.2 State space

Fig. 5.6 shows the resulting DTMC model for the EDCA MAC. Because an EDCA station always decrements at slot boundaries, bc countdown occurs in generic slots, irrespective of the medium condition. Thus, $p^\star = 0$, showing a striking resemblance to the original Bianchi model [128] where DCF bc freezing was modelled incorrectly. The result is that the (P)BO states have no self-loops and bc decrement occurs with probability 1. Because the countdown process counts down generic slots, arrivals during BO also take place in generic slots (the "diagonal transitions" in Fig. 5.6). Because the behaviour is homogeneous and no blocking occurs, the probability of receiving an arrival in a BO state equals q as defined in (5.16). Hence, $q^\star = q$.

After a transmission, bc decrement occurs directly in the IFS following the transmission. This means that a transition from TX to BO can choose a bc of $W-1$, and then decrement. The right-most state, as a result, is $s_{i,W-2}$. A station can choose $bc = 0$, and begin transmission immediately. When it chooses $bc = 1$, it will transition to $s_{1,0}$. Because of this way of bc decrementing, the same $W-2$ holds for the right-most PBO state. However, PBO is over when countdown reaches zero. This means that choosing a bc of 1 yields a 0, but a bc of 0 also yields a 0 (transition to the IDLE state $s_{0,0}$). The result is that the probability to get into this state is doubled, (probability $\frac{2}{W}$ in Fig. 5.6).

Additionally, bc decrement and TX never occur at the same boundary. To allow this, a special state $s_{1,-1}$, the transmission state, is added. It is possible to transition from IDLE ($s_{0,0}$) directly into the TX state if the channel is idle, and from $bc = 0$ ($s_{1,0}$) into the TX state upon encountering a slot boundary.

The probability of receiving a new beacon in the queue while a node is IDLE is modelled as q in the DCF model in Sec. 5.3 and in [156]. However, it is possible to model arrivals while in $s_{0,0}$ by using some additional information. The duration of a slot is determined by the state of the medium, p. As such, the mechanism which determines whether or not to perform backoff or to remain in $s_{0,0}$ is modelled with more detail by considering the duration of these slots. When an arrival occurs during a busy slot, it does not happen with probability qp, but $q_b p$, with q_b as expressed in (5.19). Likewise, whether or not a node performs a direct transmission, is governed by the probability of an arrival during an empty slot, given as $q_e(1-p)$, with:

$$q_e = 1 - e^{-\lambda_g T_e}. \tag{5.32}$$

Then, the probability of remaining in $s_{0,0}$, i.e., no arrival in this slot, is found as no

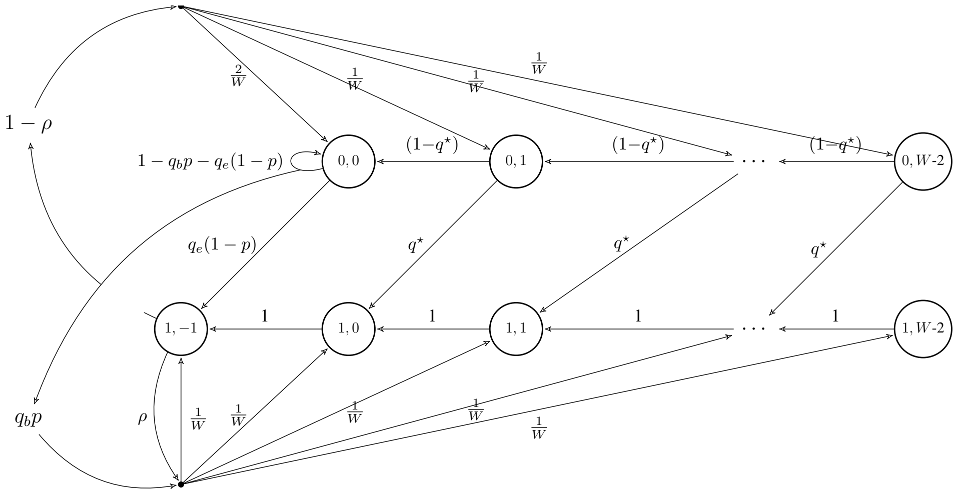


Figure 5.6: The Markov chain of the EDCA model.

arrival in busy nor in empty slots: $1 - q_bp - q_e(1 - p)$. This adds more accuracy in the region where the system transitions from mostly direct transmissions to mostly transmissions after contention.

5.4.3 Steady state distribution

Similar to the DCF model, the steady state distribution of the EDCA can be found by recursively working from right to left. Let $b_{0,k}$, $b_{1,k}$, and $b_{1,-1}$ denote the stationary probability of being in states $s_{0,k}$, $s_{1,k}$, and $s_{1,-1}$ for $k \in \{0, \dots, W-2\}$. Note from the DTMC in Fig. 5.6, that $s_{1,-1}$ is the state in which a node performs a transmission. In contrast to normalisation towards $s_{1,0}$ as used in the DCF, here we normalise towards $s_{1,-1}$. The complete derivation of the steady state equations is provided in Appendix C. The steady-state probability for a node in PBO is given by:

$$b_{0,k} = \frac{1-\rho}{W} b_{1,-1} \frac{1-(1-q^\star)^{W-k-1}}{q^\star}, \text{ for } k = 1, \dots, W-2. \tag{5.33}$$

A node is IDLE with steady-state probability:

$$b_{0,0} = \frac{(1-\rho)}{W(q_bp + q_e(1-p))} b_{1,-1} \left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right). \tag{5.34}$$

The steady-state probability for a node that is in BO is given by:

$$b_{1,k} = \frac{b_{1,-1}}{W}\left((W-k-1)\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right) + \right.$$
$$\left. (1-\rho)\left((W-k-1) - \frac{1-(1-q^\star)^{W-k-1}}{q^\star}\right)\right), \text{ for } k = 0, \ldots, W-2. \quad (5.35)$$

In a similar way to the DCF model in Sec. 5.3, the transmission probability τ equals the probability a node will transmit in a generic slot, which equals the probability of being in the state in which transmission is performed, which can be obtained by normalisation towards $s_{1,-1}$:

$$\begin{aligned} \tau =& b_{1,-1} \\ =& \left(1 + \tfrac{(W-1)}{2} + \left(\tfrac{1-\rho}{W(q_b p + q_e(1-p))}\left(1 - \tfrac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right)\left(1 + \tfrac{q_b p(W-1)}{2}\right)\right)^{-1}. \end{aligned} \quad (5.36)$$

Note that the system of equations does not depend on $p^\star$, the state of the medium while counting down, as opposed to the expression of τ for the DCF in (5.4). The system only depends on the state of the medium to determine whether to invoke a BO procedure or not. This is seen in the last term of the denominator of (5.36).

5.4.4 Service time

The AC of the beaconing traffic influences W, the AIFS, and the EIFS as tabulated in Table 2.4. As a result, the `AIFS[AC]` (2.8) and `EIFS[AC]` (2.9) replace their DCF counterparts in determining the duration of a busy and a collision slot:

$$T_s = \mathtt{T_{phy}} + \mathtt{T_{mac}} + \mathtt{S/R} + \mathtt{AIFS[AC]}, \quad (5.37)$$
$$T_c = \mathtt{T_{phy}} + \mathtt{T_{mac}} + \mathtt{S/R} + \mathtt{EIFS[AC]}. \quad (5.38)$$

In the DCF model, the streak length determines the blocking duration. Under EDCA, bc decrement occurs at each slot boundary, irrespective of the state of the medium. Instead of differentiating T_e and $\mathbb{E}[T_b]$ as defined in (5.15) for calculating the duration of the BO, EDCA counts down in generic slots of duration $\mathbb{E}[T]$, given in (5.14). As a result, the expected service time of the MAC then becomes:

$$\mathbb{E}[S] = \mathbb{E}[T_b] + \mu\left(\frac{\mathbb{E}[T_b]}{2} + \frac{(W-1)}{2}\mathbb{E}[T]\right), \quad (5.39)$$

using μ as defined in (5.22). This behaviour *only* manifests if there is no station contending for higher-AC traffic, possible because of the single-AC assumption. With this expression for $\mathbb{E}[S]$, Little's Law can be used to obtain ρ as done in (5.24) for the DCF in Sec. 5.3.

5.5 Validation

In the previous sections we described two MAC models, the DCF in Sec. 5.3 and the EDCA in Sec. 5.4. In this section, these models will be validated against results obtained from detailed simulation experiments. The validation will be based on μ, $\mathbb{E}[S]$, P_s and throughput.

In order to evaluate these models, the analytical results are obtained from the DTMC model by solving the system of equations using a fixed-point iteration approach using Matlab. The iterations terminate once $\tau - \tau_{new} < \varepsilon$, with $\varepsilon = 1 \times 10^{-6}$. The results are compared to results obtained from simulation experiments performed using OMNeT++ and MiXiM as described in Sec 2.7. Several extensions have been implemented to simulate vehicular networking, which will be presented in Sec. 5.5.1 prior to the validation. The DCF model will be validated in Sec. 5.5.2. The EDCA model will be validated in Sec. 5.5.4.

5.5.1 Extensions to MiXiM

The OMNeT/MiXiM platform is used as a basis, and several extensions have been made to simulate beaconing. These extensions are on the physical, MAC, and network level because beaconing in a vehicular network differs from WLAN Unicast, as modelled by the IEEE 802.11 implementation in MiXiM. For completeness, all of the features in the extentions are treated here, however, not all of them are used in the validation experiments. Some of these features are used in the simulation experiments performed in Chapter 6.

Going from bottom to top in the Open System Interconnection (OSI) reference model, a detailed physical layer module for 802.11p was implemented, using the 802.11p data rates. Also, the PHY includes OFDM BER curves [181] and the Decider module (responsible for the evaluation of success or failure of reception) has been extended to allow the physical layer to perform Frame Capture (FC), as described in detail in [181, 11]. FC is a phenomenon which occurs in 802.11 chipsets, where a difference in the arrival time and respective SNR among multiple signals at a receiver could still result in correct reception of one of the frames, as opposed to receiving none of the signals. This behaviour differs among chipsets from various manufacturers. In [181] we evaluate the Atheros and Prism behaviour using this implementation and find that FC can improve receiver performance for beaconing with approximately 20% compared to no FC at all. The Atheros behaviour performs an additional 5–8% better than Prism. In addition, FC was found to improve resilience against collisions with hidden terminals, because some of these collisions can still be recovered.

A MAC layer implementation was built based on MiXiM's Mac80211 model, denoted as Mac80211p. This MAC can switch between DCF and EDCA behaviour. Based on the type of MAC, the corresponding backoff counter decrement rules are used. When operating in EDCA mode, the MAC has four transmission queues for the different ACs. The AC is attached to a network packet's meta information and

Timestamp	Description
T_{creation}	set at creation of a beacon in the network layer
$T_{\mathrm{begin\_contention}}$	set in the MAC layer when it starts contending for this frame
$T_{\mathrm{begin\_transmission}}$	set when the frame begins transmission
$T_{\mathrm{reception}}$	set by the receiving node upon complete reception

Table 5.1: Timestamp fields attached to a beacon in simulation experiments.

the packet is entered in the correct queue when operating in EDCA mode.

Internal contention between the ACs (or VCH) has not been implemented. Whenever the MAC has completed transmission of a previous frame, it checks the queues for frames in descending order of priority. This is sufficient for Broadcast transmissions because CW is not increased across several transmission attempts. The situation is different in Unicast, where a transmission attempt could be a retransmission of a previous transmission. In this case the state may need to be maintained over multiple transmission attempts, with intermediate transmission attempts by higher-priority ACs.

AIFS and CW differentiation have been implemented based on the AC of the frame the MAC is currently contending for, conform Table 2.4. TXOP limit is not implemented because it is 0 for all ACs for Broadcast transmissions [2]. The MAC model is designed with several options to switch the buffering and scheduling mechanisms used by the queues in addition to the default FIFO, Newest Packet Drop (NPD) policy. The operation of these buffering and scheduling mechanisms is described and evaluated in Sec. 6.3.

As described in Sec. 4.2.2, beacons are generated at the network layer and their contents are drawn from a data storage. When using the EDCA MAC, the beacon generator attaches an AC to the beacon and inserts the beacon in the corresponding MAC queue. Received beacons are passed up to the application layer, where they are inserted in the cooperative awareness. Various beacon generators were implemented to study specific generator policies (deterministic, uniform, Poisson). Additionally, the network layer implementation can operate on a single channel, or perform multi-channel operations as defined in IEEE 1609.4. The impact of IEEE 1609.4 is described in Sec. 2.3.4 and evaluated in Sec. 6.4.

An application-layer module was designed to contain the cooperative awareness and to perform logging, in addition to several logging features in the other modules. The beacon frames are instrumented with timestamp fields. These do not add to the size of a beacon message because they are in the simulator's meta information. These timestamps are set at specific moments in the transmission process. The fields are given in Table 5.1.

Using these fields, we can derive the delay metrics for each frame at the receiver and at the transmitter side. Under the assumption that sampling delay is negligible, the end-to-end delay is the time between generation of the beacon in the network layer at node j and reception at the network layer of node i. It can be calculated as $T_{\text{reception}} - T_{\text{creation}}$. Queueing delay is found as $T_{\text{begin\_contention}} - T_{\text{creation}}$. Contention delay is calculated as $T_{\text{begin\_transmission}} - T_{\text{begin\_contention}}$. In addition, the service time of the MAC can be calculated at the transmitter as $(T_{\text{begin\_transmission}} + T_{\text{signal}}) - T_{\text{begin\_contention}}$, with T_{signal} being the duration a beacon message occupies the channel, as defined in Sec. 3.6.2. This is the time the MAC is busy performing the medium access "service" for a beacon.

Propagation Models

A simple free-space pathloss model is one of the propagation models present in MiXiM. This model applies Friisian attenuation to a signal [178]. A unit disc propagation model is created for comparison with the analytical models, which assumes all nodes observe the channel in a similar way. This model has zero attenuation within a certain range, and 100% attenuation outside it.

Node mobility

The motion of vehicles is an important aspect in the simulation study of CACC performance. First, motion determines relative distance, hence, physical layer aspects such as SNR of received signals. Secondly, the nature of the motion is a relevant application-level metric when evaluating CACC performance, as became apparent in Sec. 4.5.

Node mobility can be implemented by building a mobility model such as the Intelligent Driver Model (IDM) [182] inside OMNeT++/MiXiM, as is done in [183, 16]. Alternatively, a position trace can be obtained from a traffic simulator such as Quadstone Paramics [184], and replayed inside the simulator, as described in [185, 186]. However, the latter approach only allows the influence in one direction: traffic patterns determine the node positions and influence communication. This may be sufficient to study physical-layer aspects or routing decisions, but it is not when studying the application-level performance metrics if the communication influences mobility. A more dynamic approach is described in Sec. 4.5, where the traffic simulation tool SUMO is connected to OMNeT++ using a Traffic Control Interface (TraCI) [187]. These two simulators can then bi-directionally influence each other.

Simulator configuration for validation

To achieve a valid comparison, the unit disc propagation model was used in the simulation experiments and all nodes are in each other's reciprocal communication range to adhere to the same assumptions of the analytical model and isolate MAC-layer behaviour. No hidden terminals are assumed, and frame capture is disabled. For the validation experiments, the Poisson beacon generator was used and the MAC uses a FIFO queue sized to hold at most 14 frames, this is sufficiently large to make queue drops negligible. The experiments are performed using the method of

Parameter	Value	Parameter	Value
data rate R	3 Mbps[7]	payload S	3200 bits[8]
aSlotTime σ	16μs[9]	SIFS	32μs
W	16	DIFS[10]	64μs
$\texttt{T}_{\texttt{phy}}$	40μs	AIFS[AC0][10]	176μs
$\texttt{T}_{\texttt{ack}}$	112μs	$\texttt{T}_{\texttt{mac}}$	53μs[11]

Table 5.2: MAC parameters used in the experiments throughout this dissertation.

independent replications [88], and ten replications per datapoint were performed. No data is collected during the simulation's transient period in order to study stead-state behaviour. The duration of the transient period is estimated on based time-series inspection [188]. The simulated time in most experiments is 250 seconds, and simulation runs are time-bounded. The results have small 95% confidence intervals, as displayed in the plots.

5.5.2 Validation of the DCF model

The parameters used to validate the analytical model of the DCF are given in Table 5.2. The number of nodes is varied from $n = 1, \ldots, 200$, to show how the DCF responds to an increase in n and to evaluate its load scalability. The generation rate λ_g is kept constant at 10 Hz, the impact of varying λ_g is validated in Sec. 5.5.3 and studied in more detail in Sec. 5.6.1, where the DCF is compared to the EDCA. In the following, we compare the channel utilisation, service time of the MAC, success probability and throughput with the same metrics derived from the simulator.

Channel utilisation

The channel utilisation μ in (5.22) is compared to simulation results in Fig. 5.7. A busy slot in the model consists of a signal part and an IFS part, plotted as the top line in Fig. 5.7. μ approaches 1, resulting from the fact that when the channel is saturated it consists almost exclusively of busy slots.

In the simulator, the fraction of time there was signal on the medium was measured by a sniffer node. These measurements do not include the IFS. We derive a comparable analytical measure as follows:

$$\mu_{signal} = \frac{p_s(T_s - DIFS) + p_c(T_c - EIFS)}{\mathbb{E}[T]}, \tag{5.40}$$

[7] Default is specified as 6 Mbps; we use 3 Mbps because of greater robustness of the modulation.

[8] 400 bytes are sufficient for an EIVP CAM plus security fields [71].

[9] Duration of aAirPropagation is assumed 4μs to account for the increased transmission range, yielding σ=16μs opposed to default of 13μs.

[10] DIFS is used in the DCF, AIFS is used in the EDCA.

[11] Duration of 160 bits at 3 Mbps.

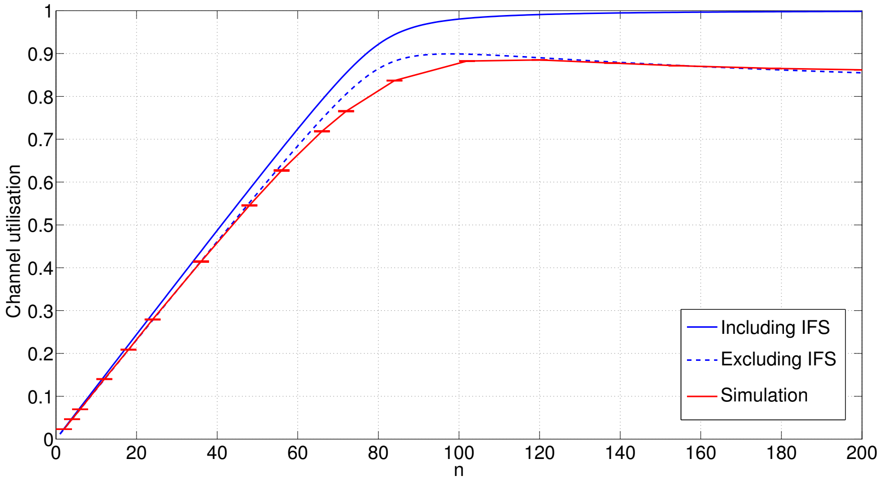


Figure 5.7: Channel utilisation of the DCF, analysis and simulation.

which divides the duration of the *signal* part of busy slots by the duration of a generic slot, thus deriving the fraction there is a signal in a generic slot. The resulting μ in Fig. 5.7 shows linear increase as more nodes are added to the system, and levels off as the system reaches saturation. The analytical results match very well with the values found through simulation, except for the slight inaccuracy in the semi-saturation part between n=60 and n=100. This inaccuracy was also observed in [156] and [159] (albeit for different values for n because the generated load differs). The numerical solution has difficulty converging in this area. This causes the number of iterations for convergence to peak. In [156] this was coined as the reason for the inaccuracy.

Although the fixed-point iteration still converges with our model, a method known as *relaxed* fixed-point iteration is applied in [152]. In this approach, $\tau^{(i)} = \tau^{(i-1)} + (1-\alpha)\tau_{new}$. In essence, this forces convergence, but not necesarrily to a fixed-point because the iteration can stabilise before having reached the fixed-point. Hence, this method is not applied in this research.

The inaccuracy described above is present in all metrics derived from the model. We verified that the fixed-point iteration converges indeed to the single fixed point. The possibility of numerical errors was reduced by by using Matlab's Symbolic Toolbox to perform most of the calculations in the symbolic domain. Hence, the inaccuracy stems from the modelling technique, and not from the numerical solution method. The inaccuracy is present in the region where the variance of T is largest. This leads to the conclusion that the inaccuracy is introduced by the mapping of time slots of differing size, to generic slots in order for the system to be Markovian. In this region, the system transitions from mostly T_e slots with sporadic transmissions, to mostly T_b slots with transmissions mediated by the backoff mechanism and sporadic T_e slots separating them.

From Fig. 5.7 it becomes clear that roughly 10% of the available channel resources

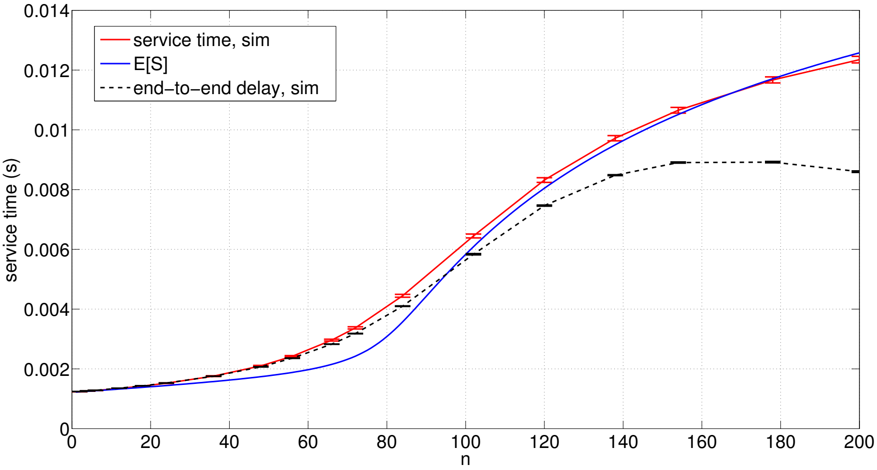


Figure 5.8: Service time of the DCF, analysis and simulation. Note how service time and end-to-end delay diverge with increasing n.

are used by the IFS, and this amount increases as collisions become more prominent. This is because nodes use an EIFS in response to observed collisions, as described in Sec. 2.4.3. Since a beacon channel consists solely of Broadcast transmissions, it would improve performance to remove the concept of EIFS in this case. This feature makes transmissions more Unicast-friendly, but is a waste of channel resources in a pure broadcast system. The impact is evaluated in Sec. 5.6.2. This waste is exacerbated by the small packet size compared to the relatively large IFS.

Service time and delay

Vinel *et al.* reported in [108] that delay requirements of ITS applications are more easily met than the reliability requirements. A simulation study in our earlier work confirmed this [12], as described in Sec. 6.2. For comparison, we obtain the service time from the simulator as described in Sec. 5.5.1. Fig. 5.8 shows that $\mathbb{E}[S]$ in (5.23) and the measured service time in the simulator match reasonably well, except for the inaccuracy in the semi-saturation region.

The average end-to-end delay in the simulator is measured only for successfully received frames, as described in Sec. 5.5.1. It is plotted as the dashed line in Fig. 5.8. A counter-intuitive result is that the average end-to-end delay actually *declines* with the number of nodes increasing beyond the point where the channel is saturated, an observation also described in the simulator study in Sec. 6.3. The explanation for this is that collisions apply a bias to which frames are received and which are not, as described in Sec. 5.3.9. This bias influences end-to-end delay as follows.

The first slot in a streak has a high average CM and a high collision probability. Subsequent slots in a streak all have lower CM, because of the smaller probability that an other node will transmit in these subsequent slots: collisions are not independent. Furthermore, transmissions in the first slot in a streak often go through contention because they were waiting for an empty slot to pass. In contrast, transmissions in later slots in a streak incur little delay because they are directly transmitted as soon as the medium turns idle (without intermediate T_e). These are the transmissions performed by nodes from groups $\mathbb{G}_{TX}$ and $\mathbb{G}_{IDLE}$ as described in Sec. 5.3.7. The result of this unfairness is a lower mean delay of successfully received information with increasing n. The resulting conclusion is that the end-to-end delay of received information cannot be directly derived from the service time.

A *medium access delay* metric can be derived for all packets submitted to the MAC queue; this would then equal the queueing delay plus the service time. From the application perspective, only the delay accumulated by succesfully received beacons is relevant. However, knowing how much time the MAC spends performing wasteful operations is relevant for protocol optimisation, as will become clear in the comparison between the DCF and EDCA in Sec. 5.6.1. It is possible to derive an estimate for the end-to-end delay based on CM and the streak concept. This is left as future work.

Another important observation from Fig. 5.8, is that the system will maintain progress and does not contend for channel access indefinitely: $\mathbb{E}[S]$ does not go to infinity. Contention is followed by a transmission. Regardless of success or collision, the MAC can go on and service the next frame in the queue. This is not the case in Unicast, where retransmissions of the same frame are performed with increasing CW, hence increasing contention delay per retry in a self-expanding manner as was described in Sec. 3.3.1.

Eventually, a streak will be over and many nodes will perform their service in parallel (causing a collision, but safeguarding the progress). Note that in this case the 802.11 MAC is able to exploit parallellism to increase scalability: collisions occur synchronised, safeguarding productive use of channel resources in other slots.

Some studies evaluate channel access delay, which essentially coincides with the service time of the MAC plus any queueing delay incurred by a packet while in the transmission queue. This metric may not be a relevant one in a vehicular networking context, because, as became clear above, this is not necessarily the latency of the correctly received information in the cooperative awareness.

Success probability

The successful reception probability is an important performance metric in a vehicular network. We can express the probability that no other node transmits, given that this node transmits, as:

$$P_s = (1 - \tau)^{n-1}. \tag{5.41}$$

Fig. 5.9 shows that P_s remains high with increasing number of nodes (resulting in increased load on the channel, showing almost linear growth in Fig. 5.7), and then starts to drop as the channel reaches saturation and collisions become more

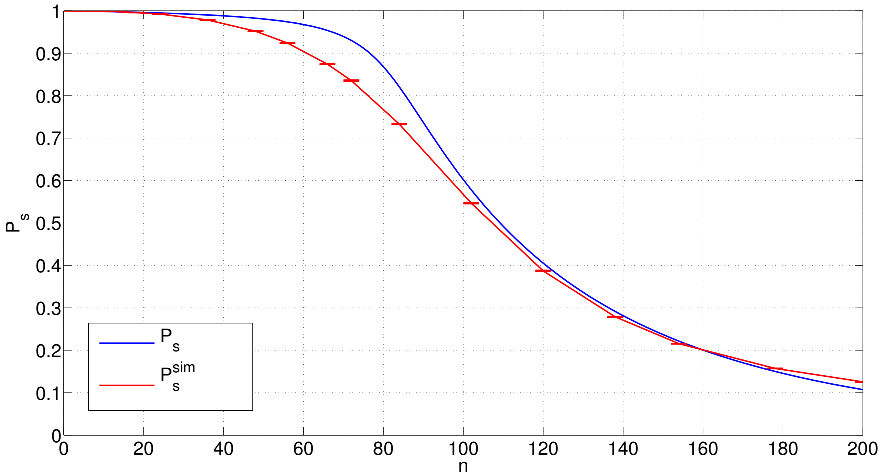


Figure 5.9: Success probability of the DCF, analysis and simulation.

prominent. Under light load, the CSMA/CA mechanism is able to coordinate transmissions. Beacons are generated at random moments in time, but if the channel is busy the MAC will defer and transmit later when the channel has become idle again. Instead of transmitting at random moments, transmissions are aligned in time by selectively delaying some transmissions. This shift in time is visible in Fig. 5.8, where $\mathbb{E}[S]$ increases: a beacon spends more time in contention, awaiting the transmissions by other nodes to complete.

The temporal spread provided by CSMA/CA allows two or more nodes to have their bc expire in the same timeslot, causing a collision. For low load this can occur sporadically (explaining the slow reduction in P_s in response to an increase in n). But this becomes more probable as n increases further, accelerating the decline of P_s as is visible halfway the figure. As the n increases even further, the system starts to behave like slotted ALOHA. Every slot is busy, but success depends on the probability that a slot is taken by only a single node, resembling the simple expression in (3.8).

The simulation results show a good match with P_s, although the inaccuracy in the semi-saturated area is visible. The simulation results decline more gradually, whereas (5.41) suddenly plummets.

Throughput

A classic metric in IEEE 802.11 modelling is throughput. The throughput X, in received beacons per second, can be derived as follows. First, the total number of slots per second can be found as $\frac{1}{\mathbb{E}[T]}$, dividing by the mean duration of a generic slot. Then, we multiply with p_s, which is the probability a slot contains a successful transmission. This yields the number of slots containing a successful transmission per second:

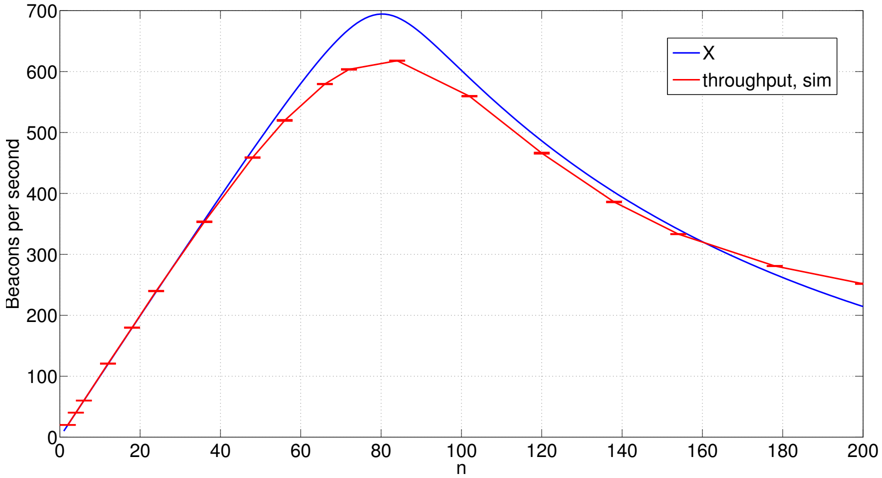


Figure 5.10: Throughput of the DCF, analysis and simulation.

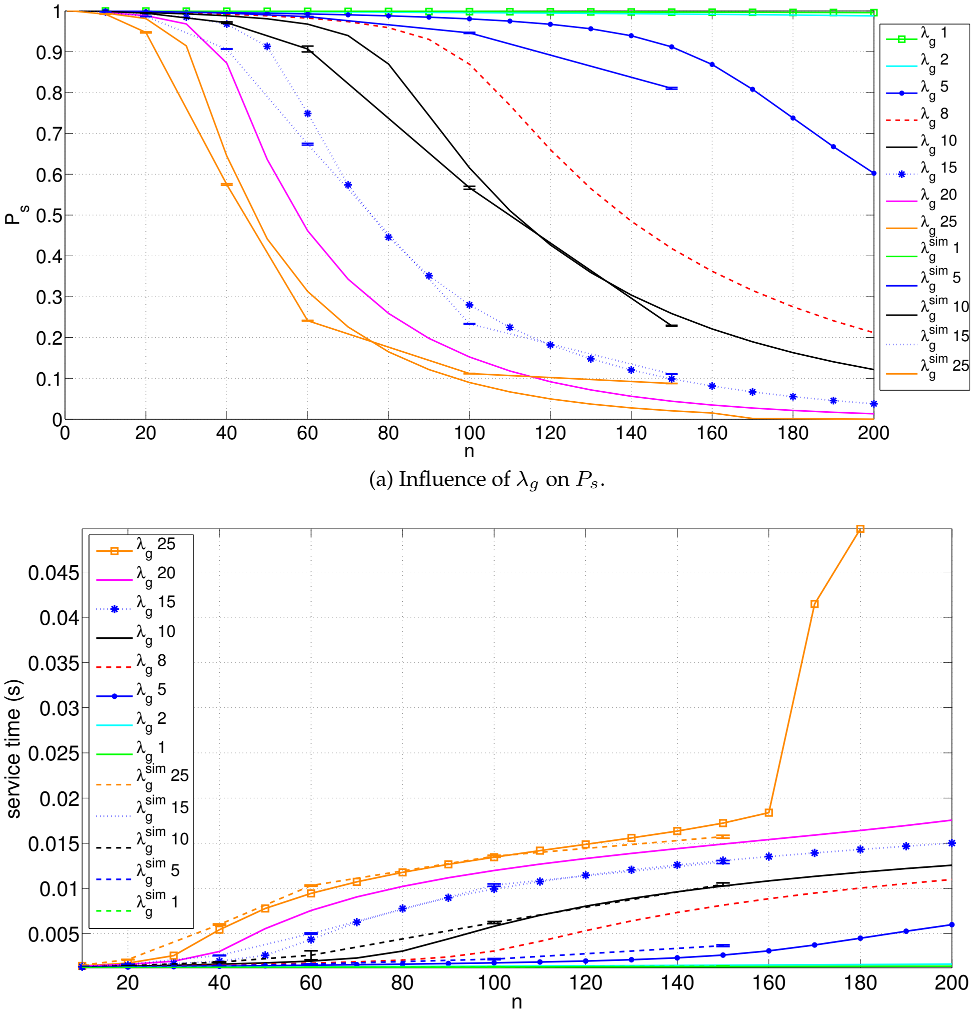
$$X = \frac{p_s}{\mathbb{E}[T]}. \tag{5.42}$$

X is plotted in Fig. 5.10 against the average number of beacons received per second, as measured by the sniffer node in the simulator. The model matches well with simulation results, although the inaccuracy in the semi-saturated area is clearly visible. This is also visible in the throughput plots in [156] and [159], on which our model is based. There, the throughput is also overestimated in the semi-saturated area. Contrary to [158], our model does show a good throughput estimation with a large number of nodes.

An important observation is that the peaks of the throughput occur at the same value of n (albeit at different magnitude). This indicates that the point where the channel becomes saturated is accurately captured by the model. This is a useful input for the design of adaptive congestion control mechanisms [189, 190, 191]; since beyond this point performance degrades.

5.5.3 Validation of DCF under varied generation rate

So far we have only studied the impact of varying n on the beaconing system. Here we also vary λ_g, and validate the resulting P_s and $\mathbb{E}[S]$. The DCF model is computed for n from 10 to 200 and λ_g from 1 to 25. Alongside the values obtained from the model are values obtained from simulation experiments similar to those used in Sec. 5.5.2. Parameters varied in the numerical analysis and the simulation experiments are given in Table 5.3. The fixed MAC parameters are equal to those provided in Table 5.2.



(a) Influence of λ_g on P_s.

(b) Influence of λ_g on $\mathbb{E}[S]$.

Figure 5.11: DCF under varied generation rate.

Success probability

Fig. 5.11(a) shows the P_s curve as also plotted for λ_g=10 Hz in Sec. 5.5.2, but also includes P_s for λ_g ranging from 1 to 25 Hz. In addition, the corresponding P_s results from simulation are plotted.

It becomes evident that using a larger λ_g results in a more rapid decline of performance when the number of nodes increases, because the channel becomes overloaded. A lower λ_g shows a more moderate decline in P_s under increasing n. The lines plotting the simulation results show that under varying λ_g, the DCF model retains a good fit with simulation results. The inaccuracy in semi-saturation present in the model also exists for higher and lower generation rates.

An exception to the good fit is the datapoint n=150 for λ_g=25 Hz, where P_s is significantly higher in the simulator. The explanation is that, thanks to the streak phenomenon, performance does not completely drop to zero. However, under these circumstances the system is so far into saturation that it is questionable whether this regime has any practical use.

Service time

The service time of the MAC is plotted in Fig. 5.11(b) for different values of λ_g, together with the service time as obtained from the simulator. The analytical results match well with the simulation results, except for the inaccuracy in the semi-saturated area as noted before.

With an increasing generation rate, the channel reaches saturation sooner and delay starts to increase at lower n. For λ_g=25 Hz, the service time explodes as it approaches the same order of magnitude as the inter-arrival time τ_g of the beacons (0.04 s for 25 Hz). The model becomes unstable (in terms of the model, $\rho = 1$), yielding unreliable results. Note that this is also the region in which the entire beaconing system is overloaded, and packets are being dropped at the queue.

5.5.4 Validation of the EDCA model

The EDCA MAC model is solved using a similar fixed-point iteration method as the DCF model in Sec. 5.5.2. The MAC parameters are similar to those presented in Table 5.2, with the exception that AIFS[AC0] overrides the DIFS and EIFS[AC0] overrides EIFS (see Table 2.4).

In the following, we will show that the EDCA model fits well with results obtained from detailed simulation experiments under veried n. The channel utlisation, service time of the MAC, success probability and throughput will be compared with the same metrics derived from the simulator. In Sec. 5.5.5, the EDCA MAC is validated for varied λ_g.

Channel utilisation

The channel utilisation μ can be calculated in a similar way to that presented in Sec. 5.5.2, with the exception that the DIFS and EIFS are replaced by their EDCA

Parameter	**Analysis**	**Simulation**
n	10–200 in steps of 10	2, 10, 20, 40, 60, 100, 150
λ_g	1,2,3,4,5,6,7,8,9,10,15,20,25	1, 5, 10, 15, 25

Table 5.3: Varied parameters used to study impact of varying generation rate.

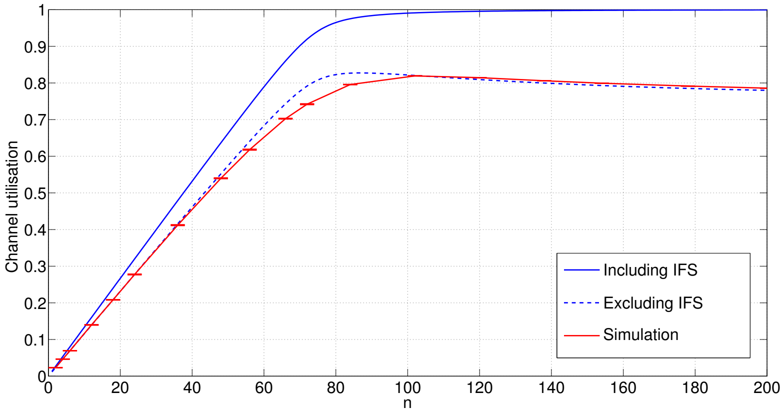


Figure 5.12: Channel utilisation for the EDCA, analysis and simulation.

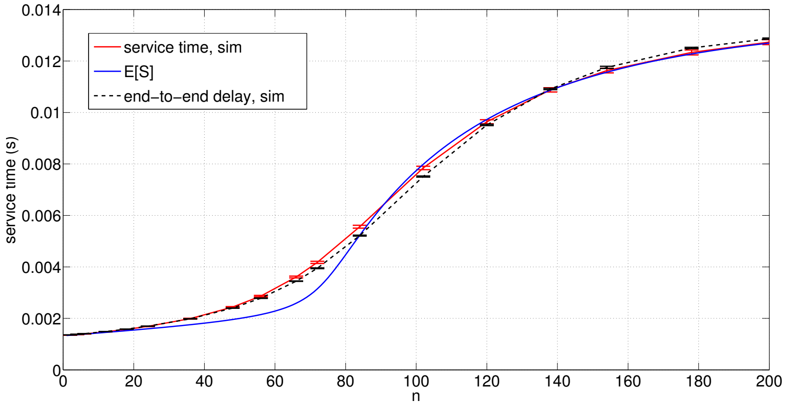


Figure 5.13: Service time of the EDCA, analysis and simulation. Note how service time and delay overlap for the simulation results.

counterparts to subtract the duration of the IFS from a timeslot:

$$\mu_{signal} = \frac{p_s(T_s - AIFS[AC]) + p_c(T_c - EIFS[AC])}{\mathbb{E}[T]}. \tag{5.43}$$

Fig. 5.12 plots μ (5.22) and μ_{signal} (5.43) in conjunction with the channel utilisation measured in the simulator. The expression for μ shows good match with simulation results, except for the same inaccuracy in the semi-saturated region as present in the DCF model. Also visible in the figure, is that the EDCA exhibits a reduction in μ due to the increase in IFS because of the increasing use of the EIFS as collisions become more prominent.

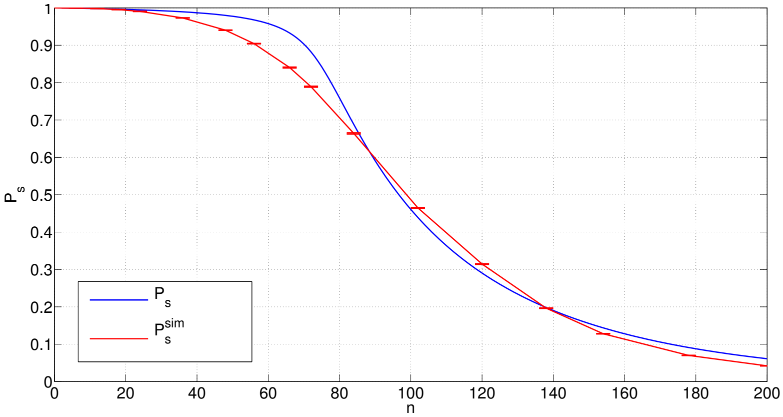


Figure 5.14: Success probability of the EDCA, analysis and simulation.

Service time and delay

Fig. 5.13 plots $\mathbb{E}[S]$ (5.39) as described in Sec. 5.4.4 in conjunction with service time and end-to-end delay as measured in the simulator. Service time in the simulator matches well with the simulation results, except for the semi-saturated area as was also the case for the DCF model.

Observe that the EDCA bc decrement behaviour has a seriously different impact on the end-to-end delay of the received beacons compared to the DCF in Fig. 5.8. The service time and the end-to-end delay are almost equivalent, as visible in the simulation results. This is because the bias which applies by the interplay of CM and the streak phenomenon as found for the DCF, is not present in the EDCA. The exclusive probability of transmission in a streak for beacons which arrived while the station was in its IDLE state does not exist in the EDCA.

In effect, EDCA shows a more Slotted ALOHA-like behaviour because every slot is equally likely to contain transmissions. The probability of collision in the EDCA is independent of the moment when a beacon enters the queue and the state of the MAC. In contrast, the DCF exhibits a lower collision probability for beacons arriving while the MAC is IDLE.

The result for the EDCA is that the service time and the end-to-end delay accumulated by an average beacon, are similar because there is no collision bias. This equality holds because queueing delay is negligible; a beacon has been transmitted before the next one arrives with high probability. Queueing delay is part of the end-to-end delay, but not part of the service time. This explains the slight rise of the end-to-end delay in Fig. 5.13 for large n, where beacons start to accumulate queueing delay. For the EDCA, the service time of the MAC can be used as a good estimator for the age of the received information, i.e., the latency the application will experience.

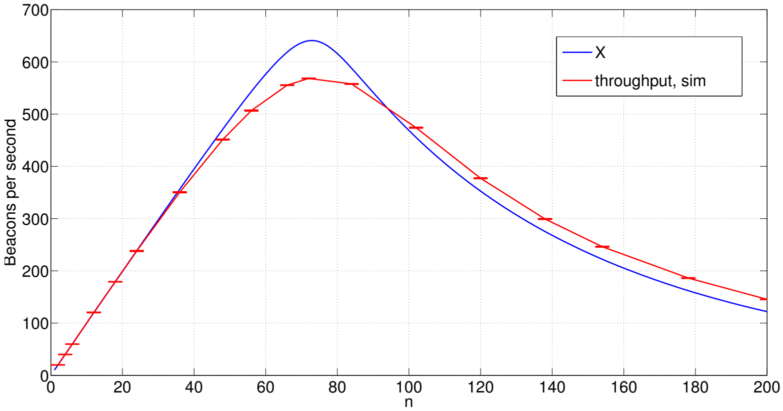


Figure 5.15: Throughput of the EDCA, analysis and simulation.

Success probability and throughput

The P_s for the EDCA is calculated using (5.41) in a same way as for the DCF. P_s is plotted in Fig. 5.14, and shows a good match with simulation results, clearly capturing the impact of collision loss on the success probability of each transmitted beaconing. For larger n, a small deviation starts to manifest. Just like in the other metrics, the inaccuracy in semi-saturation also manifests here.

Throughput

The throughput for beaconing using the EDCA is calculated using (5.42) and plotted in Fig. 5.15 together with the mean number of beacons received per second in the simulator. Similar to the throughput of the DCF, the peaks occur for the same value of n but are quantitatively different due to the inaccuracy in this region. X shows an underestimation of the number of received beacons per second for large n, which corresponds with the inaccuracy of P_s in this region.

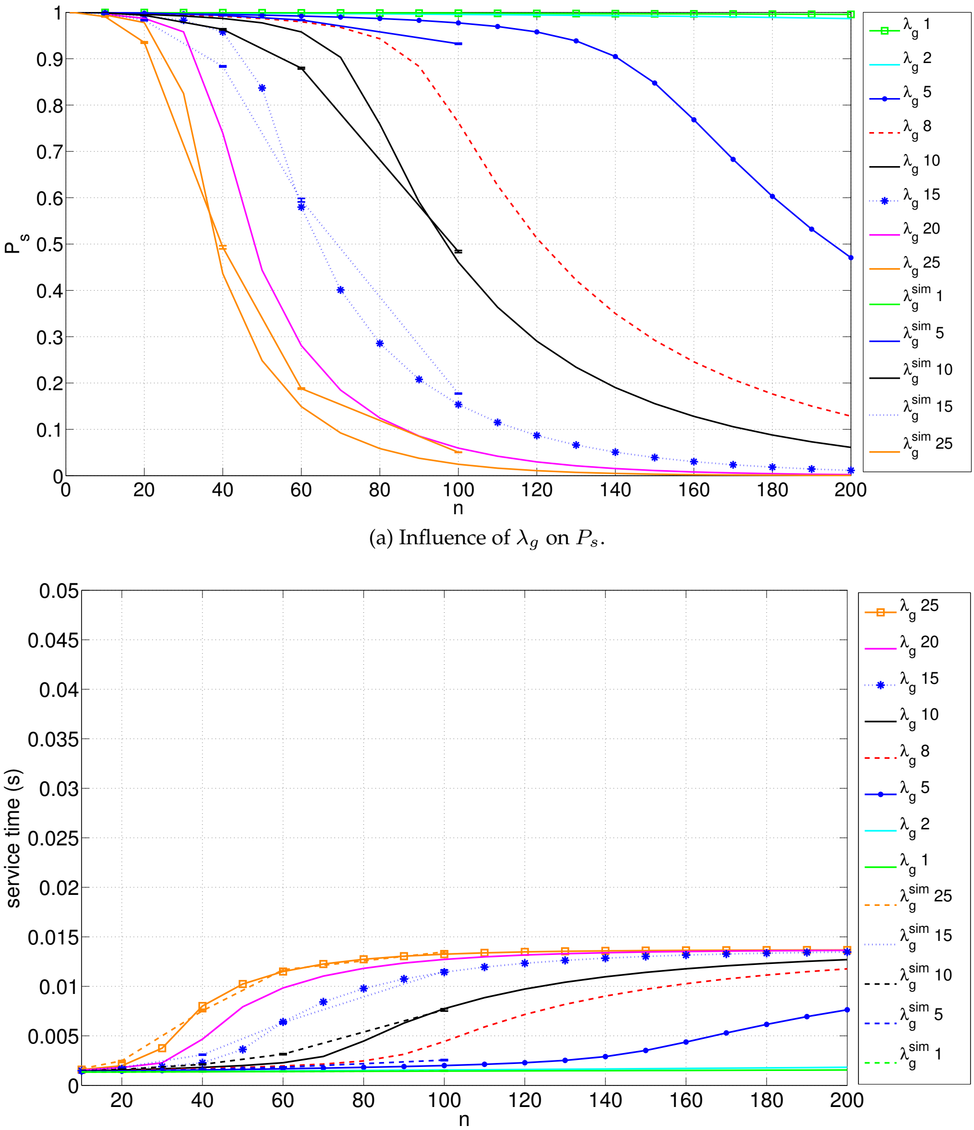
5.5.5 Validation of EDCA under varied generation rate

The EDCA model is validated for varied λ_g in a manner similar to the validation of the DCF in Sec. 5.5.3. We use the parameters tabulated in Table 5.3, and plot both the analytical values and the values obtained from simulation experiments.

Success probability

The success probability of the EDCA under varied λ_g is plotted in Fig. 5.16(a). This figure also contains the value λ_g =10 Hz seen before in Sec. 5.5.4.

Similar to the DCF, a larger λ_g results in a more rapid decline of performance when the traffic density increases. The lines plotting the simulation results show a

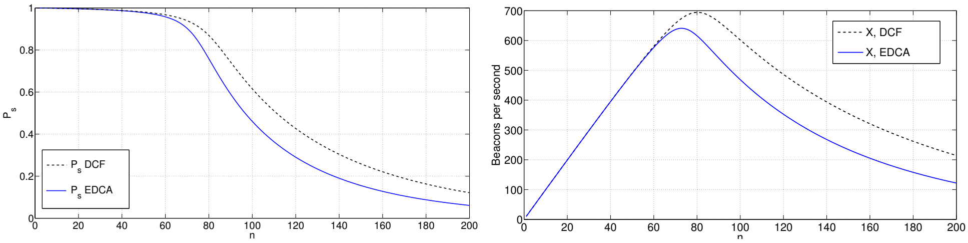


(a) Influence of λ_g on P_s.

(b) Influence of λ_g on $\mathbb{E}[S]$.

Figure 5.16: EDCA under varied generation rate.

good fit with the EDCA model. As was the case with the DCF model, the EDCA model also retains the inaccuracies in the semi-saturated areas under varied λ_g.



(a) Success probability P_s.

(b) Throughput X.

Figure 5.17: P_s and X, DCF vs. EDCA.

Service time

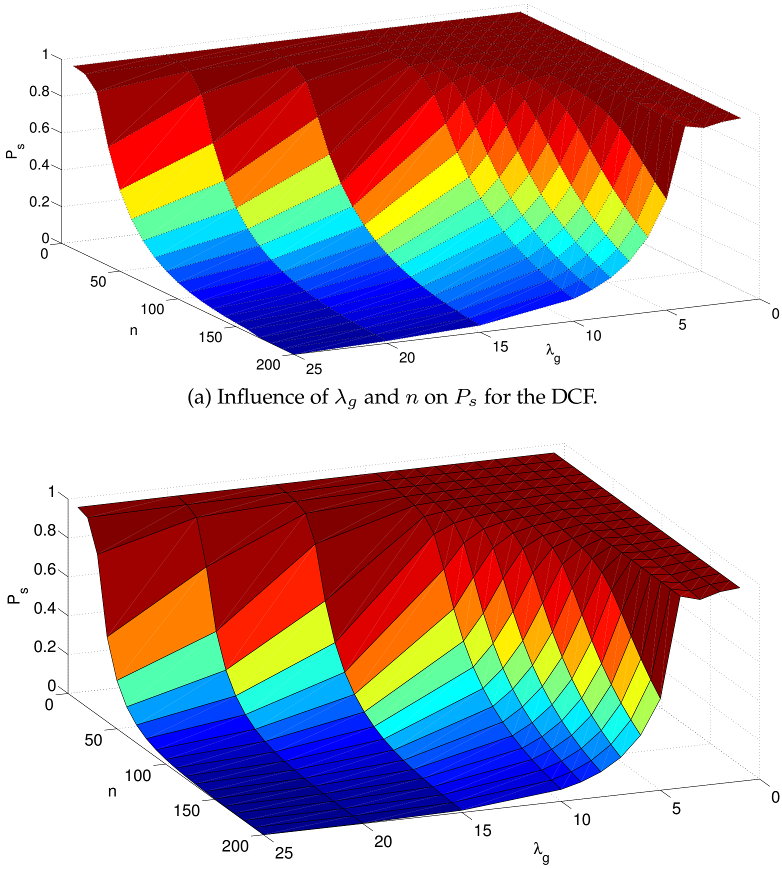
The service time of the MAC is plotted in Fig. 5.16(b), together with service time as measured during the simulation experiments. Note that the y-axis is plotted in the same scale as the figure for the DCF service time (Fig. 5.11(b)). The results obtained from the analytical model match well with the simulation results.

An interesting observation is that the EDCA model does not become unstable for λ_g =25 Hz and large n. This is because the bc blocking under DCF can force a node to defer increasingly longer and exceed the inter-generation time τ_g, while the service time under EDCA is limited by the contention window size: it will decrement every slot. See Sec. 5.6.1 for a detailed comparison between DCF and EDCA service time.

5.6 Performance analysis of DCF and EDCA

In the previous section, the DCF and EDCA models were validated against results obtained from simulation experiments. Both models show a good match with the simulation results, and can be used to study the performance of beaconing, explore sensitivity to parameters, or perform comparisons.

The IEEE 802.11p standard defines use of the EDCA based upon the assumption that a vehicular network is a "difficult" environment and that some messages may require priority over others. Though the environment may be difficult, this is primarily at the physical level and will affect both DCF and EDCA because both use the same OFDM physical layer technology as described in Sec. 2.6. The performance of DCF and EDCA are compared in Sec. 5.6.1. In this comparison, we look at P_s, $\mathbb{E}[S]$, X, μ, and τ_r. Next, in Sec. 5.6.2 the impact of the EIFS is evaluated, and Sec. 5.6.3 looks into which AC to choose when using EDCA for beaconing.



(a) Influence of λ_g and n on P_s for the DCF.

(b) Influence of λ_g and n on P_s for the EDCA.

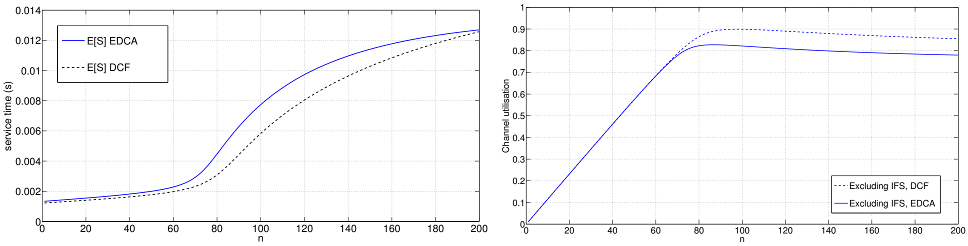
Figure 5.18: P_s of DCF and EDCA under varied generation rate.

5.6.1 Comparison of DCF and EDCA

It is interesting to compare the DCF and EDCA, and see if the EDCA is indeed the right access mechanism for the purpose of beaconing. This comparison is performed in this section, using the MAC parameters in Table 5.2 and Table 2.4.

Success probability and throughput

Fig. 5.17(a) shows P_s for both DCF and EDCA, as also plotted before in the section on validation. It becomes evident that the DCF is able to achieve a significantly larger P_s with increasing n than the EDCA. For the DCF, reduction in P_s occurs for larger n than the EDCA and remains higher, up to being nearly double the P_s of EDCA for $n > 150$.



(a) Service time $\mathbb{E}[S]$.

(b) Channel utilisation μ.

Figure 5.19: $\mathbb{E}[S]$ and μ of the DCF and EDCA.

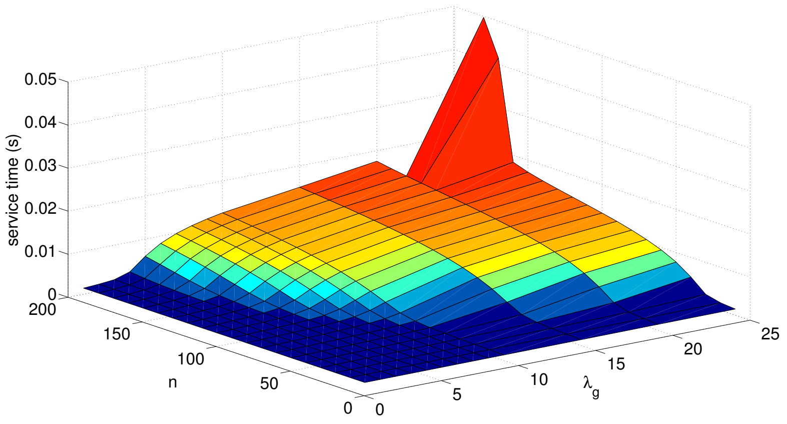
The same difference is also visible in Fig. 5.17(b), which shows X for DCF and EDCA. The DCF is able to achieve a larger throughput than the EDCA, and its saturation point occurs at larger n. This can partially be attributed to the smaller IFS of the DCF, which leaves more channel resources for effective use, but also to the streak behaviour caused by the bc decrementing behaviour in the DCF. To get a notion of the behaviour of the EDCA with the same IFS as the DCF, see Sec. 5.6.3. There an AIFSN of 2 is used, and shows a signifcantly lower P_s for large n than the DCF; the difference lies in the streak behaviour.

We now compare the DCF and EDCA under varied λ_g to explore the impact P_s. The explored parameters are equal to those tabulated in Table 5.3. In Fig. 5.18(a), P_s is plotted versus n and λ_g for the DCF. Fig. 5.18(b) shows P_s for the EDCA. Comparing these figures with Fig. 3.9 derived with the simple model in Sec. 3.6.4, it is clear that the detailed models show a more optimistic decay of P_s under increasing n or λ_g. This results in a flat surface around the axes, which is absent in the simpler model. This is because the simple model does not include the performance gain provided by the coordination of CSMA/CA, which is effective in this area. A large part of the solution space has an acceptable P_s for CACC. Still, beaconing at λ_g=25 Hz is, as was concluded in Chapter 3, a solution with very poor scalability.

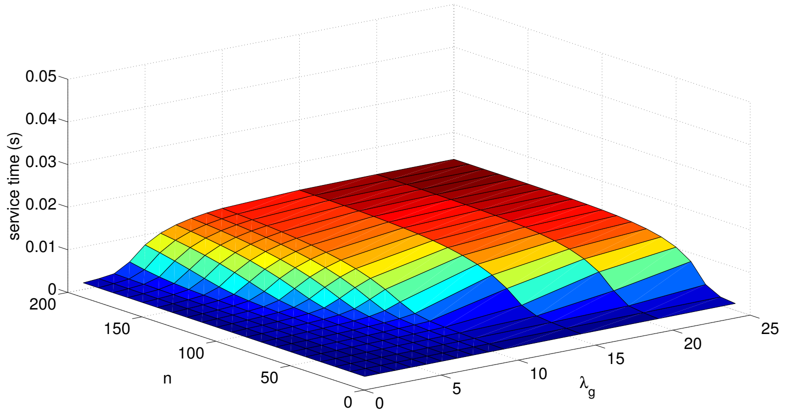
Service time, delay and channel utilisation

As described in Sec. 5.5.2, $\mathbb{E}[S]$ of the DCF does not coincide with the end-to-end delay found in the simulator, because of the bias applied by the streak behaviour caused by the bc decrementing behaviour. The EDCA in Sec. 5.5.4 exhibits completely different end-to-end delay behaviour when compared to the DCF, almost similar to $\mathbb{E}[S]$ because of the EDCA bc decrement rules, as was visible in the simulation results in Fig. 5.13.

Fig. 5.19(a) shows $\mathbb{E}[S]$ for the DCF and the EDCA together. The EDCA reaches the saturation point at a lower value of n as observed in Fig. 5.19(b) comparing the μ of DCF and EDCA. This causes $\mathbb{E}[S]$ to rise sooner for the EDCA. The IFS in the DCF is smaller than in the EDCA, which explains the higher μ which can be achieved by the DCF: less time is spent in the guard spaces. This is also visible in the $\mathbb{E}[S]$ for the



(a) Influence of λ_g and n on $\mathbb{E}[S]$ for the DCF.



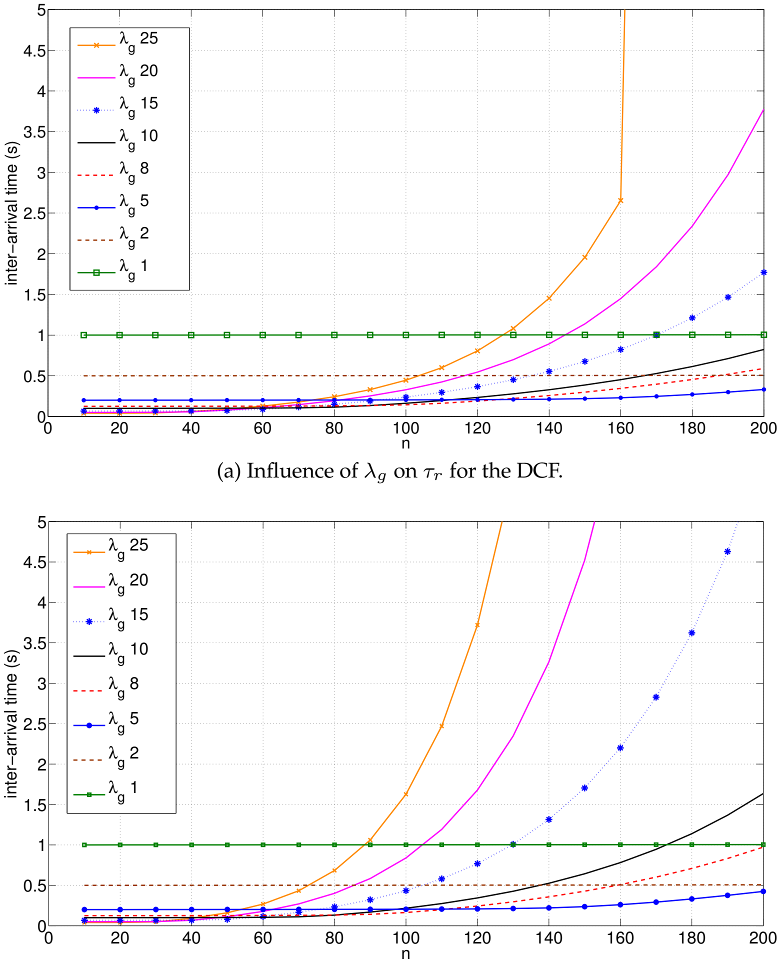
(b) Influence of λ_g and n on $\mathbb{E}[S]$ for the EDCA.

Figure 5.20: $\mathbb{E}[S]$ of DCF and EDCA under varied generation rate.

DCF for low values of n, which is slightly lower than that of the EDCA.

Fig. 5.20(a) plots $\mathbb{E}[S]$ versus n and λ_g, showing a low delay for low n and / or λ_g. The figure shows the similar spike in service time for a combination of large n and λ_g as observed in Fig. 5.11(b). However, this is a regime which should be avoided at all times.

Fig. 5.20b plots $\mathbb{E}[S]$ for the EDCA. When compared to the DCF in Fig. 5.20(a), it is clear that service time rises quicker upon increasing either n or λ_g, but remains lower than that of the DCF.



(a) Influence of λ_g on τ_r for the DCF.

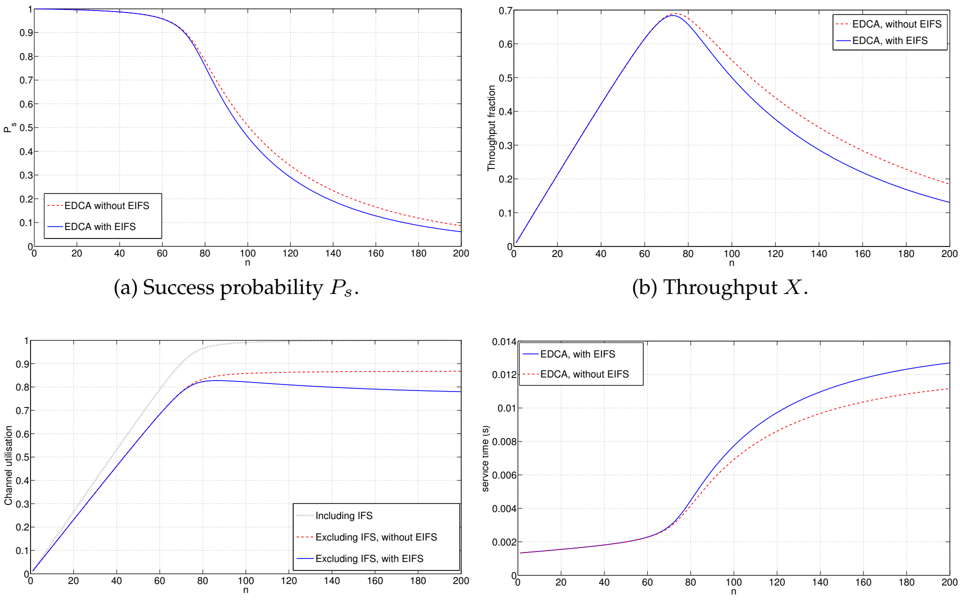
(b) Influence of λ_g and n on $\mathbb{E}[S]$ for the EDCA.

Figure 5.21: The inter-reception time τ_r under DCF and EDCA.

Inter-reception time

Beacons are generated at rate λ_g. Because of packet loss, they are received at rate λ_r, as expressed in (3.4). This rate can then be used to obtain the mean inter-reception time τ_r using (3.5), which is plotted for various values of λ_g for the DCF in Fig. 5.21(a) and for the EDCA in Fig. 5.21(b).

For the DCF using $\lambda_g = 1$ Hz, τ_r is 1 s and this is unaffected by an increase in n because the channel does not become saturated and P_s remains close to 1, even at large n. The same holds for λ_g =2 Hz, but a generation rate of 5 Hz starts to show an increase in τ_r for high n. Larger generation rates have a smaller τ_r under low n,



(a) Success probability P_s.

(b) Throughput X.

(c) Channel utilisation μ.

(d) Service time $\mathbb{E}[S]$.

Figure 5.22: Evaluating the EDCA, with and without EIFS.

but rapidly saturate the channel under increasing n, resulting in an increase in τ_r.

Fig. 5.21(b) shows τ_r for the EDCA. Compared to the DCF, τ_r rises sooner under increasing n and λ_g. This is because of the significantly lower P_s of the EDCA (see Fig. 5.17(a)).

Judging from the results for τ_r, the DCF clearly is the best option for beaconing for CACC: information in the cooperative awareness is refreshed more often. For instance, consider λ_g =10 Hz and n =120, then τ_r is approximately 250 ms for DCF and nearly 400 ms for EDCA.

5.6.2 The impact of the EIFS on beaconing

The EIFS is designed as a courtesy feature towards Unicast transmissions. If a station was not able to decode the received Unicast message because it contains bit errors, it is possible that the intended recipient was able to receive it properly. This receiver should then have the opportunity to send an acknowledgement frame. As described in Sec. 2.4, the NAV is set to delay channel access for the duration of EIFS after an observed collision.

In a system where all transmissions are of a Broadcast nature, using the EIFS just consumes precious channel resources and serves no purpose. There is no Unicast traffic which could benefit from the delayed channel access.

In the following, the model of the EDCA is used with and without EIFS. The EDCA uses a larger IFS than the DCF and clearly shows the impact of the EIFS. The DCF shows similar behaviour, although of a less severe nature because the gain by disabling EIFS is smaller. When the EIFS is disabled, an AIFS is used after an observed collision.

The results are provided in Fig. 5.22. A higher P_s and throughput can be realised without the EIFS. In fact, P_s can be increased by up to 40% for $n = 200$, as visible in Fig. 5.22(a). In addition, Fig. 5.22(c) shows that the channel can be used more efficiently, indicated by a higher μ when disabling the EIFS. This more efficient use of the channel goes hand-in-hand with a reduced $\mathbb{E}[\mathbb{S}]$ visible in Fig. 5.22(d); a node gets access to the channel sooner if it only has to wait an AIFS, and bc countdown is quicker.

The $\mathtt{EIFS}[\mathtt{AC0}]$ is more than twice $\mathtt{AIFS}[\mathtt{AC0}]$, i.e., 360 μs vs. 176 μs. Following up on the reasoning that the AIFS is already long compared to the DIFS—and accounts to a large degree for the difference in performance between the DCF and EDCA—the use of EIFS is particularly wasteful in an EDCA Broadcast channel.

The difference between using the AIFS or EIFS manifests when collisions become more probable, in Fig. 5.22(c) it is clearly visible this starts at n=80. The system then enters a regime under which CSMA/CA shows poor performance, a situation idealy avoided in practice.

5.6.3 Which AC to use for beaconing?

When using the EDCA, an important question is which AC to use for beaconing. The service differentiation is implemented using different CW sizes and AIFS durations per AC, as described in Sec. 2.5.1. It would be straight-forward to designate beaconing as the highest priority traffic given its importance, but what is the impact on performance? To answer this question, we use the EDCA model described in Sec. 5.4 and evaluate the four scenarios resulting from W=$\{4, 16\}$ and AIFSN=$\{2, 9\}$.

AC0 has a CW size of 16 and a long AIFS (with AIFSN=9). AC3 has a CW size of 4 and AIFSN=2, the AIFS duration equal to DIFS. To show the impact, we also evaluate the large CW with the short AIFS, and the small CW in combination with the long AIFS.

The resulting P_s is plotted in Fig. 5.23. What becomes evident from Fig. 5.23, is that increasing W from 4 to 16 has only a marginal effect. However, changing AIFSN has significant effect, as also found in [146]. In the comparison with the DCF, we observed that its smaller IFS (the size of a DIFS) enabled a larger μ and hence improved X and P_s compared to the EDCA. AC3 also has a similar benefit over AC0: it has a significantly smaller IFS and hence more available channel resources.

The small impact of W on beaconing was also observed in [12] and is described in the simulation study described in Sec. 6.2.3. The reason why W has such small effect is two-fold: firstly, as the number of nodes rises, the system succumbs to a slotted ALOHA-like regime and the system is fully synchronised. Secondly, slots under the EDCA are homogeneous and the EDCA lacks streak-effects because of

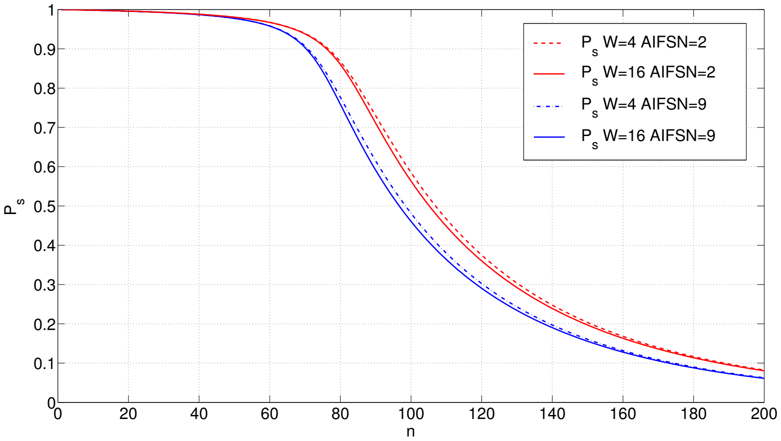


Figure 5.23: Comparing EDCA AC0 and AC4 for use in beaconing.

its bc decrement behaviour. This causes a transmission to face the same collision probability, irrespective if it had performed a long backoff, a short backoff, or an immediate transmission. The backoff mechanism is designed to temporarily *exclude* a share of nodes from channel access by defering. This works well in high rate WLAN situations with bursty arrivals and a low number of nodes. However, in a large beaconing scenario there are many other nodes which take the place of those currently deferring access.

By performing beaconing in a lower AC, EDCA essentially *reserves* opportunities for higher priority transmissions to be performed with lower delay and larger P_s—provided that it is only a minority which receives this priority. Hence, by accepting a penalty in scalability due to the larger IFS, the system is able to perform beaconing *and* allow transmission of high priority information. It may even be beneficial to differentiate priority on a per-beacon level, because the AIFS differentiation divides the n nodes into groups, some of which cannot interfere with transmissions of higher-priority frames. This method is also applied in [192] to isolate ACs. As a result, these transmissions will experience better P_s which will in turn increase the beaconing system's load scalability.

5.7 Discussion on n, λ_g, and beaconing performance

In this chapter, it became clear that under increasing n the performance quickly deteriorates. By varying λ_g in the experiments in Sec. 5.6.1, the impact of the generation rate on beaconing performance was illustrated. This shows the great potential for methods to either throttle generation rate in response to the observed number of neighbours [20], or throttle the number of neighbours by means of power control [193, 109], or combine both methods.

Though some works consider a fixed λ_g of 10 Hz [4, 3], we motivate in [9] that *generation rate adaptation* as a network-layer mechanism is one of the instruments to increase scalability. Increasing λ_g results in a higher temporal resolution. But this comes at the price of an increase in channel load, hence, collision probability, especially in dense traffic. Decreasing λ_g in such scenarios reduces the load, causing fewer collisions. A drawback of this approach is that if the rate is too low, this also affects the cooperative awareness as it does not get updated often enough [108]. An example of this is using a λ_g below 5 Hz, visible in Fig. 5.21a to be virtually insensitive to an increase in load but result in a low λ_r even under perfect P_s.

In case of severe channel congestion caused by large n, one could reason that it is better to at least have *some* information, than no information at all. In [9] we suggest adapting λ_g based on channel estimations and road traffic dynamics as collected in the cooperative awareness. An overview of situation-adaptive beaconing schemes is provided in [194]. Beacon rate control is used in [195] to minimise the age of the information in the cooperative awareness. Low information age is particularly important for real-time applications such as CACC [196]. Controlling λ_g in a vehicular network is possible for several reasons [10]:

1. Inherent unreliability of the wireless channel: the application must already cope with this by applying filtering (e.g. interpolation, Kalman estimates). Due to the nature of wireless communication, upper layers already need to deal with loss probabilities, even under the best of circumstances.

2. More load leads to more loss. Under certain conditions, λ_r may be raised by lowering λ_g. This is a well-known phenomenon in networks, where raising the supplied load beyond a certain point will lead to a reduction in throughput due to loss. In the case of 802.11 Broadcast, there is no reaction to load in the MAC so this will need to be implemented at the Network layer.

3. As traffic dynamics decrease (e.g., in traffic congestion, speeds are significantly lower than in free-flow), a high λ_g is not always necessary for the application to operate. This influences the relevance of communicated information: if a node has not moved, it does not need to notify its neighbours. This suggests the use of a situation-dependent minimum required λ_g.

As a means to vary n, *transmission power adaptation* is a good method to increase the scalability of a dense vehicular network [193, 110]. The goal of transmission power control is to increase spatial frequency reuse, this increases the system's load scalability because it can more effectively exploit parallelism. Reducing transmission power under an increasing vehicle density is a logical measure to preserve scalability, because vehicles are closer and traffic dynamics are reduced.

The aim of these mechanisms is to adapt n or λ_g to keep P_s in the flat area visible near the axes in Fig. 5.18. When the system operates in this region, the service time remains low and both end-to-end delay and τ_r will also be low.

5.8 Conclusions

In a vehicular network, the large number of nodes, in combination with the low supplied load per node and the strict use of the Broadcast transmission method, result in a behaviour typical to this environment. Current IEEE 802.11 models have a focus on Unicast in WLAN, and are not readily able to capture the behaviour of a vehicular network. Many of these models do not implement the complete standard, or are based on simplifying assumptions such as saturation of the transmission queue, which do not reflect the actual system behaviour. The assumption of independence among consecutive transmissions often made does not provide good estimates of the duration of bc blocking and transmission probability, especially not under an increasing number of nodes.

In this chapter, two MAC models have been proposed which allows fast computation of metrics of interest and can easily be embedded in models of a broader scope, for instance considering the traffic context.

A detailed model of the IEEE 802.11 DCF MAC was provided. To obtain a good estimate of DCF performance, the concepts of collision multiplicity and streak length are used to obtain a better estimate of the bc blocking probability and service time. In addition, we find that whether a transmission is performed in the first or subsequent slots in a streak impacts its reception probability and end-to-end delay. We motivate that the service time and channel access delay for the DCF MAC cannot directly be used to obtain the end-to-end delay of the received information at the receiver, because of the collision bias caused by the streak phenomenon.

Based on the DCF model, a detailed model of the EDCA was presented, incorporating the EDCA bc decrementing rules. Because of the way the bc is decremented in the EDCA, the service time of the MAC can be used to estimate the end-to-end delay of the received information at the receiver. In contrast to the DCF, collisions under EDCA are independent; there is no bias as found in the DCF.

Analytical results are obtained from the two MAC models through fixed point iteration and are validated against results obtained from detailed simulations using the platform described in Sec 2.7 with the extensions described in Sec. 5.5.1. We observe that the probability of successful reception and service time of the MAC match well, and the system behaviour is captured well by the models. Additionally, the saturation point of the channel can accurately be predicted for both DCF and EDCA. This is an important input in the development of congestion control mechanisms.

The DCF and EDCA models presented in this chapter have a greater level of detail and complexity than the simple model presented in Chapter 3. However, they do not capture semi-saturated behaviour with high accuracy. This is due to the modelling technique, which generalises slots of differing duration and assumes independence among these types of slots in order for the system to be Markovian.

Next, the DCF and EDCA MAC were compared for use for beaconing. The DCF is found to show better performance than the EDCA under increasing n. This can be attributed to:

1. The large AIFS in the EDCA, which causes more channel resources to be wasted

as guard space.

2. The lack of streak behaviour in EDCA due to its *bc* decrementing behaviour, which causes inefficient use of consumed resources.

3. In contrast to the EDCA, the fraction of beacons which *are* received when using the DCF has a favourable delay characteristic because of a collision bias. The streak behaviour causes frames transmitted from contention to have a higher collision probability than those undergoing direct transmission. The result is that the end-to-end delay of the DCF is significantly better than that of the EDCA.

The IEEE 802.11p standard mandates the use of the EDCA, however we have shown this access method to have drawbacks when compared to the DCF for the traffic on the beaconing channel. An observation relating to both DCF and EDCA is that use of the EIFS is obsolete in a Broadcast-only environment, and unnecessarily consumes channel resources.

In this chapter we focussed on modelling the behaviour of the medium access mechanism, and the DCF and EDCA in particular. Though these models are powerful tools, there are certain studies which are easier performed using a simulation-based approach. These will be presented in Chapter 6.

CHAPTER 6

Case Studies

In Chapter 5, a comparison of the DCF and EDCA was performed and it became clear that the DCF has some features which make it the more scalable of the two. This chapter investigates three distinct features which could influence scalability. In doing so, an answer is provided to research question 6 ('What can be done to increase the scalability properties of IEEE 802.11p?').

Although the analytical approach presented in Chapter 5 is very powerful, it has a few limitations, mentioned below, which make simulation an attractive alternative in some cases. Some additional metrics could be obtained from the simulation study used to validate the models in Sec. 5.5.2 and 5.5.4. One such metric is the distribution of the end-to-end delay, which will be described in Sec. 6.1.

The simulation platform described in Sec 2.7 is then used to perform the following studies. These studies use the DCF *bc* blocking behaviour described in Sec. 5.3.7, as this was found to outperform the EDCA. Sec. 6.2 analyses the impact of the contention window size on beaconing, and shows that the CW affects performance differently in beaconing than it does in WLAN Unicast. Alternative buffering and scheduling methods for beaconing are evaluated in Sec. 6.3, showing that there is merit in using an Oldest Packet Drop policy as opposed to the Newest Packet Drop policy frequently used in WLAN. The modelling approach in Chapter 5 relies on the assumption of an infinite FIFO queue, in the simulator the nature and implementation of the queue can easily be changed.

Sec. 6.4 looks into the beaconing performance under IEEE 1609.4 channel switching constraints, and shows that performance degrades more than 50% when IEEE 1609.4 is applied. Performing such an evaluation is difficult with the models in Chapter 5, because the models consider steady-state behaviour. By performing channel switching, transient behaviour manifests in the beginning and end of the CCH periods, which is not captured by the analytical models. Finally, Sec. 6.5 concludes this chapter. This chapter contains material previously published in [10, 12, 13, 17].

6.1 Distribution of the end-to-end delay

In this section, we treat the distribution of the end-to-end delay, the time between generation of a beacon at a source node and reception at a receiving node, as defined in Sec. 3.5.2. This data is recorded in the simulation experiments which were used for the validation experiments in Sec. 5.5.2 and Sec. 5.5.4. The histograms are used to display an estimate of the end-to-end delay distribution or density.

6.1.1 End-to-end delay distribution of the DCF

Fig. 6.1 shows the emperical distribution of the end-to-end delay for the DCF for a number of nodes ranging from 4 to 200. In the simulator, the end-to-end delay of received beacons is recorded in 1 ms intervals or bins. The value of such a bin then is a measure of the frequency of occurence of observed end-to-end delays which were within this interval.

As visible in Fig. 6.1(a), with only a few nodes the delay is low and hence the share of the smallest bin is nearly 100%. As n increases, the average end-to-end delay goes up, reflected in a rise of the share of bins with larger delays. The tail becomes longer, but short delays still remain very prominent, with approximately 42% of delay of received beacons below 3 ms for $n = 200$ in Fig. 6.1(f).

The distribution is truncated at bin 49–∞, any overflow is added to this last bin. This explains the small rise of the last bin barely visible in Fig. 6.1(f). The data of the last three bins for n=200 is tabulated in Table 6.1, and shows that about 0.046% of the beacons has a delay which exceeds 49 ms. Note the relatively large 95% confidence intervals. These are caused by the fact that receiving a beacon with excessive end-to-end delay is a rare event. In order to obtain better confidence intervals in this range, a significantly larger number of beacon transmissions needs to be observed.

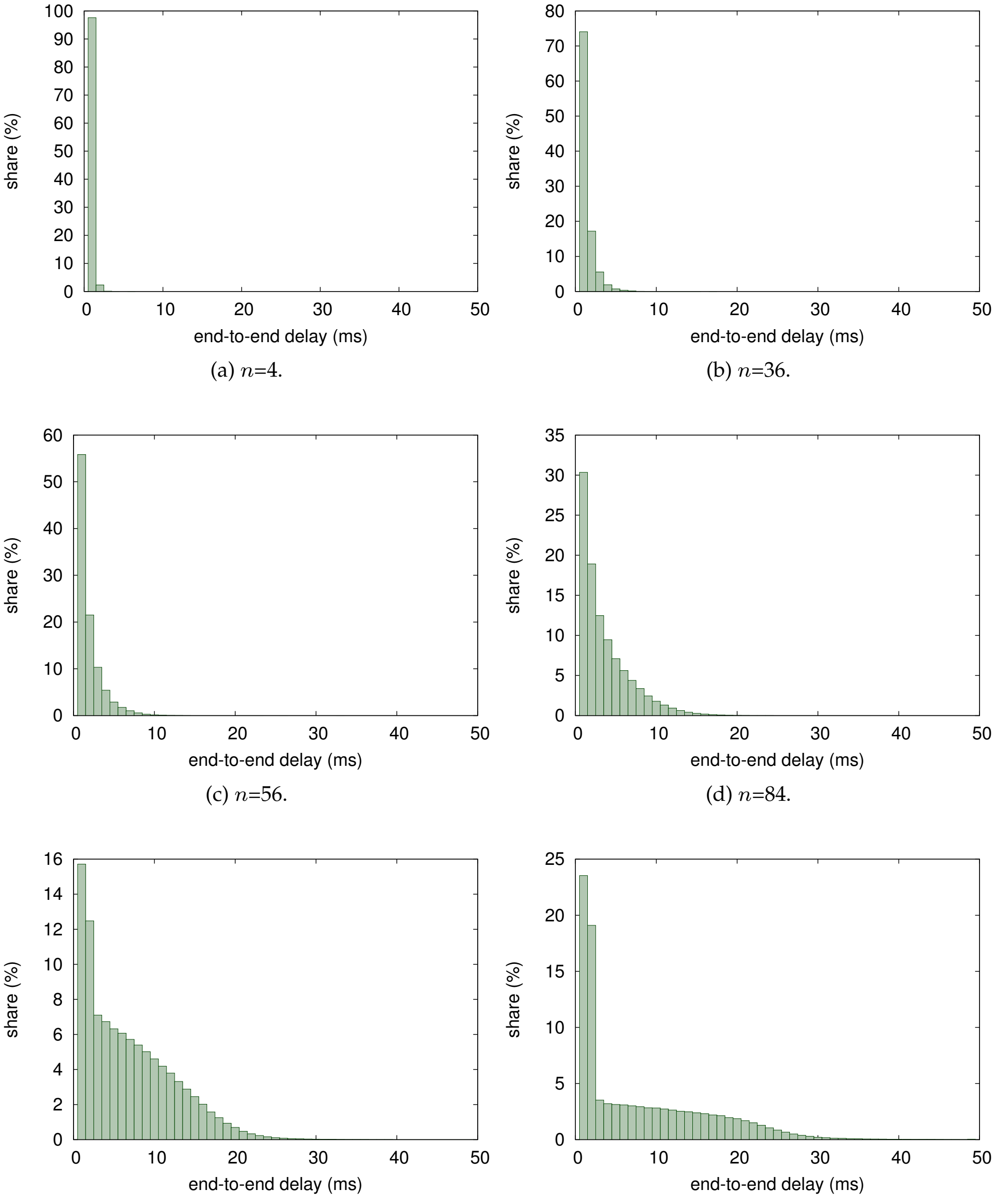
The histograms in Fig. 6.1 show resemblance to the distributions found analytically by Engelstad and Østerbø in [160], although magnitudes differ because of different parameters.

Bin (ms)	**Share (%)**	**95% CI**
47–48	0.0104	0.0105
48–49	0.0092	0.0057
49–∞	0.0463	0.0229

Table 6.1: Last three bins of the end-to-end delay distribution of the DCF.

6.1.2 End-to-end delay distribution of the EDCA

Similar to the end-to-end distribution of the DCF, the share of the smallest bin is largest for low n under the EDCA, as visible in Fig. 6.2(a). Note, however, the different progression of the distribution as n increases. Where the DCF groups many received beacons into the low end-to-end delay bins, the EDCA shows a significantly larger probability that a received beacon has a large delay. This coincides with the findings in Chapter 5, where the mean delay of received beacons under EDCA



(a) n=4.

(b) n=36.

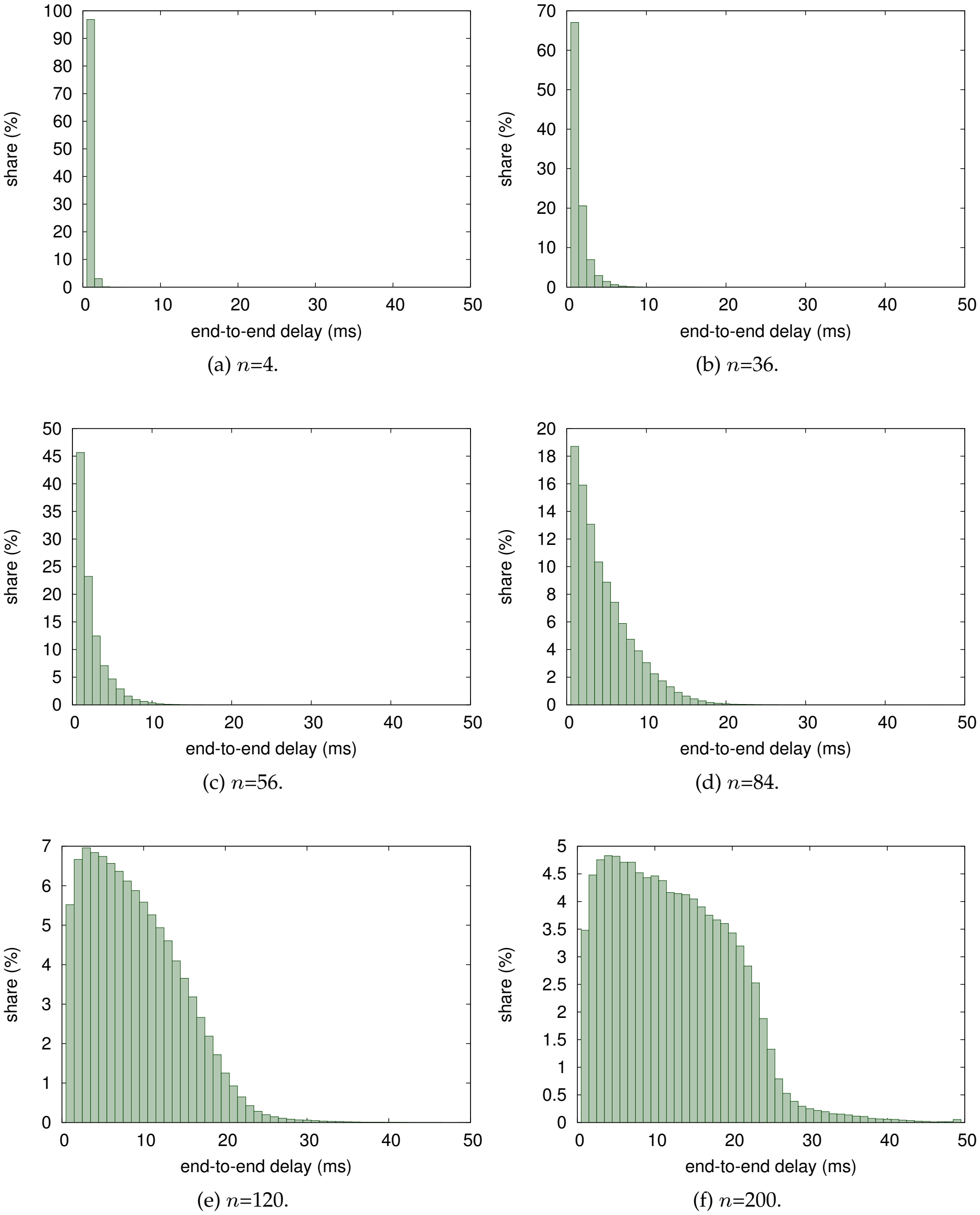
(c) n=56.

(d) n=84.

(e) n=120.

(f) n=200.

Figure 6.1: Distribution of the end-to-end delay of the DCF.



(a) n=4.

(b) n=36.

(c) n=56.

(d) n=84.

(e) n=120.

(f) n=200.

Figure 6.2: Distribution of the end-to-end delay of the EDCA.

was found to be larger than under the DCF. This difference in distribution can be attributed to the *bc* blocking and the streak behaviour as described in Sec. 5.3.7.

As opposed to the 42% of beacons with a delay < 3 ms in the DCF for n=200 (see Fig. 6.1 (f)), this is only approximately 8% in the EDCA, as visible in Fig. 6.2(f). In addition, whereas the DCF shows a clear peak for the lowest bins, the peak in the DCF shifts to the right with increasing n. Exceptionally long delays are slightly more likely in the EDCA, as visible in Table 6.2 where the last three bins are tabulated. Here too, the last bin contains any overflow, showing that approximatly 0.055% of the received beacons has a delay exceeding 49 ms.

As opposed to the DCF, the histograms in Fig. 6.2 differ from the distributions found analytically by Engelstad and Østerbø in [160]. It was observed in Sec. 5.4 that this model is essentially DCF decrement behaviour with AIFS differentiation. The distribution of the end-to-end delay of the EDCA is of a different nature.

Bin (ms)	**Share (%)**	**95% CI**
47–48	0.0146	0.0307
48–49	0.0155	0.0223
49–∞	0.0554	0.0440

Table 6.2: Last three bins of the end-to-end delay distribution of the EDCA.

6.2 Impact of contention window size on beaconing

In Unicast, the CW is increased when a transmitting node does not receive an acknowledgement. Either the data frame or its acknowledgement was involved in a collision, or suffered from propagation-related loss. By means of doubling of the CW, the subsequent access attempts by the nodes involved in the collision are spread in time, reducing the collision probability for the *next* transmission attempt. As such, the collision avoidance mechanism by means of the CW desynchronises nodes which previously were synchronised by scheduling some transmissions further away into the future. Additionally, doubling of the CW has a load-balancing effect: a node spends more time in contention for a single frame, reducing the number of frames this node transmits into the channel per second.

In [10] we observed that there may be merit in adaptively changing various MAC layer properties in response to an increased channel load because of large traffic densities. The size of the CW was one of the suggested properties. Here, we evaluate the impact of increasing the CW and find it to be surprisingly low, and explain why.

The default initial CW value (`CWmin`) for the DCF is 15, yielding a CW size of 16. This value is equal to the CW for AC0, the low priority AC in the EDCA. For high priority traffic in the EDCA, CW is 3 (see Table 2.4). This is good to prioritise one frame over the other but does not provide adequate spread in backoff duration

Parameter	Values
Number of nodes n	4, 20, 40, 80, 120, 150
Poisson generation rate (λ_g)	1, 5, 10, 25
Contention Window (CW)	3, 7, 15, 63, 255, 1023
Beacon size	3200 bits
Data rate	3 Mbps
Transmission queue	14 frames

Table 6.3: Simulation parameters for the CW study.

when the system approaches saturation conditions, especially with a large number of stations. This was also observed for transmission of beacons in [190]. This leads to a counter-productive situation, because high priority traffic has a small CW to obtain low delay, but also a very high probability of collision with other nodes when synchronised to the same channel-busy event. Low priority traffic, on the other hand, has a larger CW. This results in a larger delay, but also in a lower probability of collision.

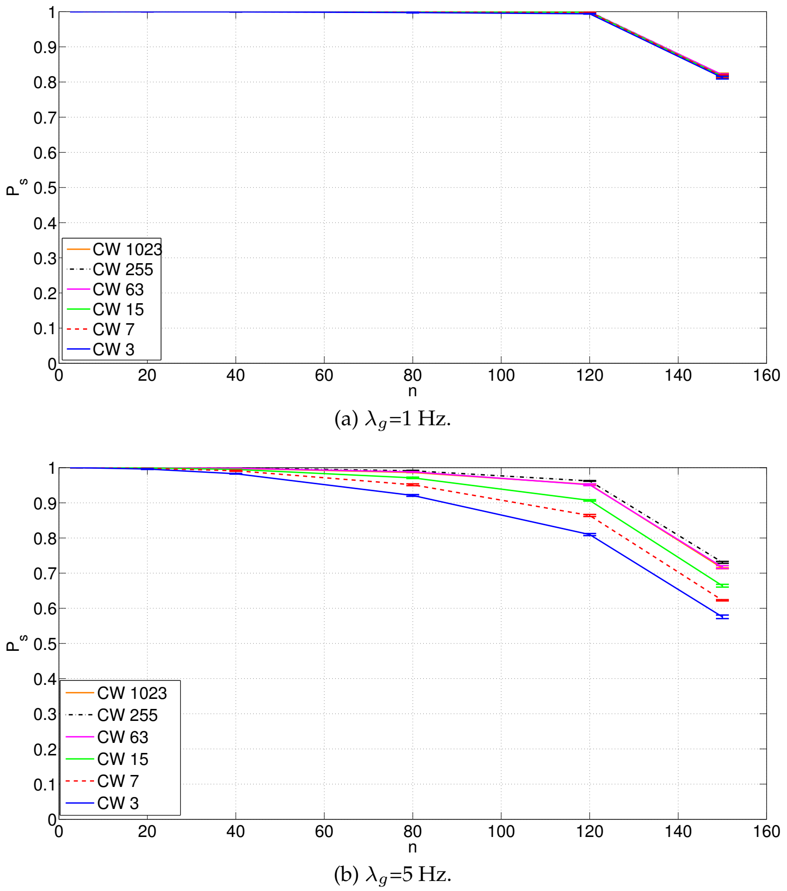
The experiments will be described in Sec. 6.2.1. The results of this study will be presented in Sec. 6.2.2. A discussion on CW adaptation for beaconing is provided in Sec. 6.2.3, and Sec. 6.2.4 presents the conclusions of this study.

6.2.1 Experiment description

This simulation study evaluates various generation rates and contention window sizes under varying number of nodes. The simulation platform described in Sec. 2.7 is used, together with the extensions covered in Sec. 5.5.1. The experiments simulate 200 seconds of beaconing, *after* the simulation reaches a steady state, determined by visual inspection. During the transient phase no results are collected. The experiments are conducted using the method of independent replications [88]. 95% confidence intervals are given in the figures.

The experiments are conducted using a unit disc propagation model to isolate collision loss, and all nodes are within range of each other. We are interested in isolating IEEE 802.11p’s ability to coordinate transmissions without causing collisions. By disabling propagation loss, all loss can be attributed to collisions. Because of these assumptions, the results provide valuable insight of the operation on the MAC level but actual performance in reality may differ.

Beacons are generated with Poisson rate λ_g, which is varied in the different simulation experiments. The number of nodes and CW are varied as well. An overview of the different values used is shown in Table 6.3. Note that these parameters include the EDCA default values from Table 2.4. The number of nodes ranges from very sparse traffic (4 nodes in range) to congested (150 nodes in range), a bumper-to-



(a) λ_g=1 Hz.

(b) λ_g=5 Hz.

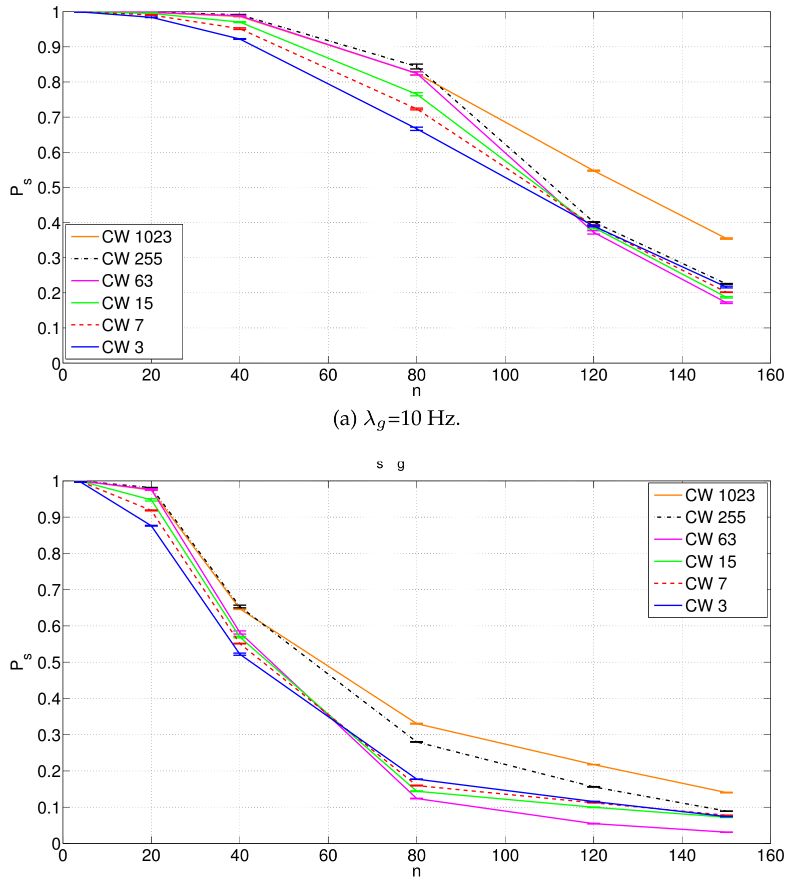
Figure 6.3: P_s for λ_g=1 and 5 Hz and various CW sizes.

bumper scenario. The nodes do not perform 1609.4 channel switching, but reside on the same channel continuously to maximise their cooperative awareness.

6.2.2 Simulation results and analysis

The metrics used in this study were introduced in Sec. 3.5, and are the following. First, P_s will be studied under varied λ_g and CW, where the dropping probability P_{drop} is used to explain the effect of a large CW on P_s. Next, the delay accumulated by the beacons is studied.

Reception probability Fig. 6.3 shows P_s for each of the considered CW values for λ_g= 1 and 5 Hz, in relation with the number of nodes. Fig. 6.3(a) shows P_s for λ_g= 1 Hz, and there is no difference between the studied CW sizes. This is primarily



(a) λ_g=10 Hz.

(b) λ_g=25 Hz.

Figure 6.4: P_s for λ_g=10 and 25 Hz and various CW sizes.

because the channel load is low for all considered values of n. Fig. 6.3(b) shows beaconing at 5 Hz. Up to 40 nodes there is no difference between the different CW values. As n increases further, the smaller CW sizes suffer a greater decline in P_s than the larger CW sizes.

Also, note that λ_g =1 Hz, n=150 and λ_g =5 Hz, n=30 result in the same number of transmissions on the channel per second (namely 150), but Figs. 6.3(a) and 6.3(b) show differing success probability. This illustrates that the CSMA/CA mechanism is better able to deal with few nodes and high rates, than many nodes and a low rate. The reason for this is that the MAC can better coordinate access attempts by nodes which are synchronised (more likely under a higher λ_g) than the less-coordinated transmissions resulting from the random beacon arrivals under low λ_g.

Fig. 6.4 shows P_s for λ_g=10 and 25 Hz, for the studied CW values in relation

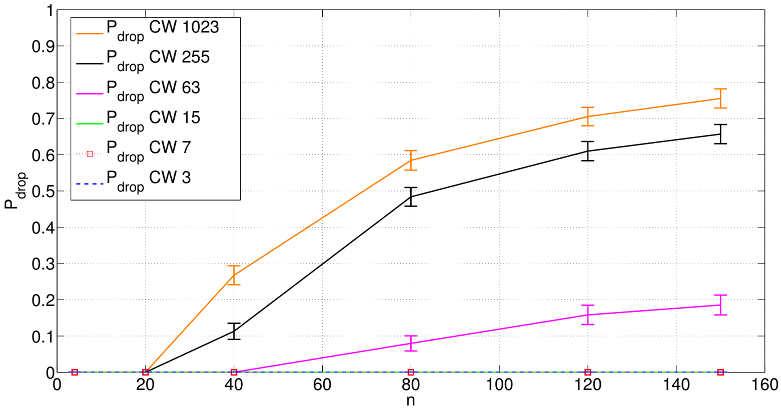


Figure 6.5: P_{drop} for λ_g=25 Hz.

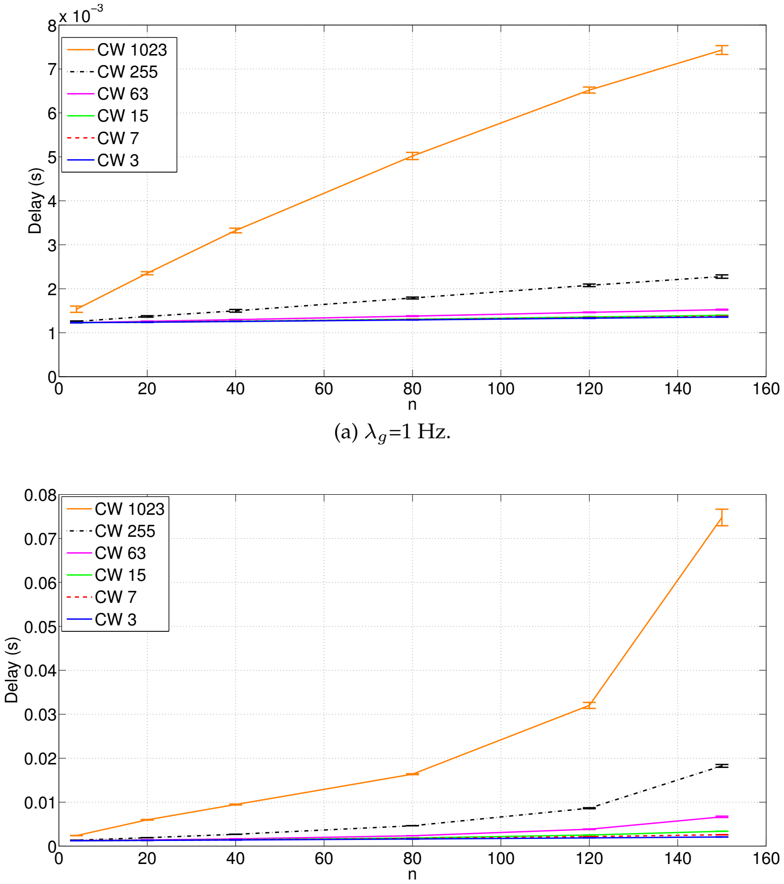
with n. The scenario with λ_g=10 Hz is shown in Fig. 6.4(a), and decline sets in for lower n than observed for lower λ_g: the system is saturated under a lower number of nodes because of the larger supplied load per node. CW 3 shows a nearly linear decline, wheareas larger CW sizes have a more prominent bump before declining and perform better on this metric. All studied CW sizes seem to converge beyond n=120, except for CW=1023 which shows an apparent remarkably good performance. An explanation for this is provided later on.

When $\lambda_g = 25$ Hz, decline already sets in for very low n, visible in Fig. 6.4(b). Note that in addition to CW=1023, CW=255 now also seems to significantly outperform the other CW values.

The acceptable reception probability depends on the type of application supported by the cooperative awareness. Ma and Chen specified in [166] that P_s should not drop below 0.99 for safety-critical applications. As evident in the results, this requirement is readily violated as traffic density increases. For other ITS applications, such as CACC, the reception probability can be lower than 99%, depending on the loss tolerance of the CACC controller [23], as shown in Sec. 4.5. Loss tolerance can for these applications be improved by means of, e.g., Kalman filtering.

The simulation results show that a P_s of 99% is only achieved if the number of nodes is very low, or if λ_g is below 5 Hz. If n is below 20, P_s is around 99% for nearly all λ_g values and for all considered CW values. It seems from these results that, by increasing CW, P_s can be improved. However, we should not rely on a single metric, since the CW size has some other effects, as will be treated below.

But why then does a large CW show good P_s? To answer this, consider Fig. 6.5, which shows the dropping probability, as described in Sec. 3.5.1, for λ_g=25 Hz. As clearly visible, P_{drop} rapidly increases for the large CW sizes, up to a value of over 0.7 for CW=1023. In the following, the relation between delay and CW size will be addressed and it will become clear that queue buildup starts to occur for larger CW sizes. This causes packets to be dropped, because the transmission queue is often



(a) λ_g=1 Hz.

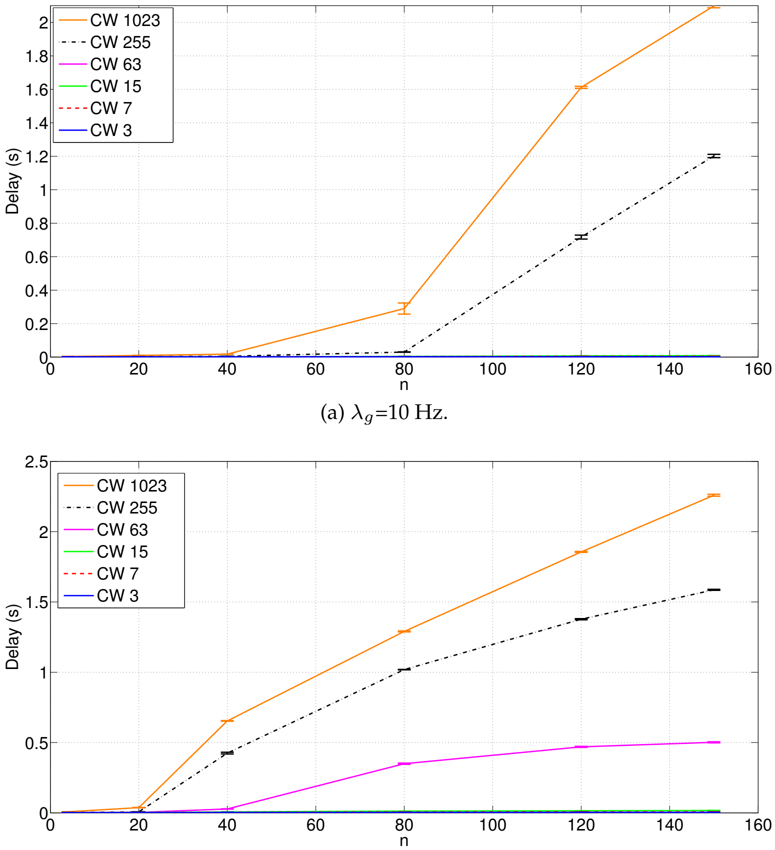
(b) λ_g=5 Hz.

Figure 6.6: Mean end-to-end delay for λ_g=1 and 5 Hz and various CW sizes.

encountered to be full by arriving packets. This, in turn, lowers the supplied load on the channel, yielding an actual gain in P_s.

Delay analysis Fig. 6.6 and 6.7 (note the differences in y-scale) shows the mean delay experienced by beacons under various traffic densities for the considered CW values at various generation rates. The average delay is larger for a larger CW (see Fig. 6.6(a)) and rises as the number of nodes increases. This rise can primarily be attributed to the fact that, when a backoff is to be performed, the average bc drawn from the CW is larger and a node spends more time counting down.

When λ_g is increased to 5 Hz, Fig. 6.6(b) shows that the large CW sizes show an increase in delay. With a λ_g of 10 Hz (see Fig. 6.7(a)) or 25 Hz (see Fig. 6.7(b)) it becomes evident that end-to-end delay increases significantly with an increase in CW



(a) λ_g=10 Hz.

(b) λ_g=25 Hz.

Figure 6.7: Mean end-to-end delay for λ_g=10 and 25 Hz and various CW sizes.

size, up to values of well over a second, unacceptable to just about any application in the ITS domain.

The delays found here coincide with the findings in [108], a study which only considered a small CW. The reported delays were up to approximately 8 ms with $\lambda_g = 10$ Hz, visible in Fig. 6.8, which shows the data of Fig. 6.7(b) without the two largest contention windows.

Work on a CACC system in [41] models a communication delay of 10 ms. From our results, the assumed communication delay in the CACC system is not violated, unless very large CW values are used or the system enters the saturated regime. This is what happens to the large CW values in Fig. 6.7 (a), where the effect of queue-buildup becomes eminent: beacons are sometimes (but increasingly often for increasing n) generated faster than they can be transmitted. A beacon then not only has a contention delay, but also accumulates queueing delay. The relation between

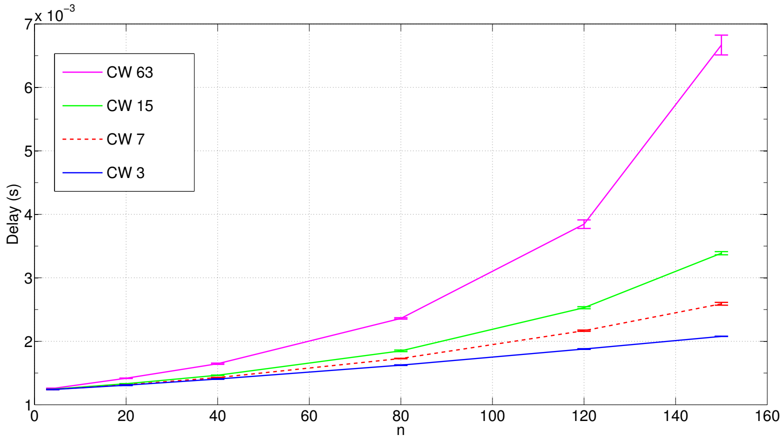


Figure 6.8: Mean end-to-end delay for λ_g=10 Hz in detail, ignoring CW 255 and 1023 (visible in Fig. 6.7(a)).

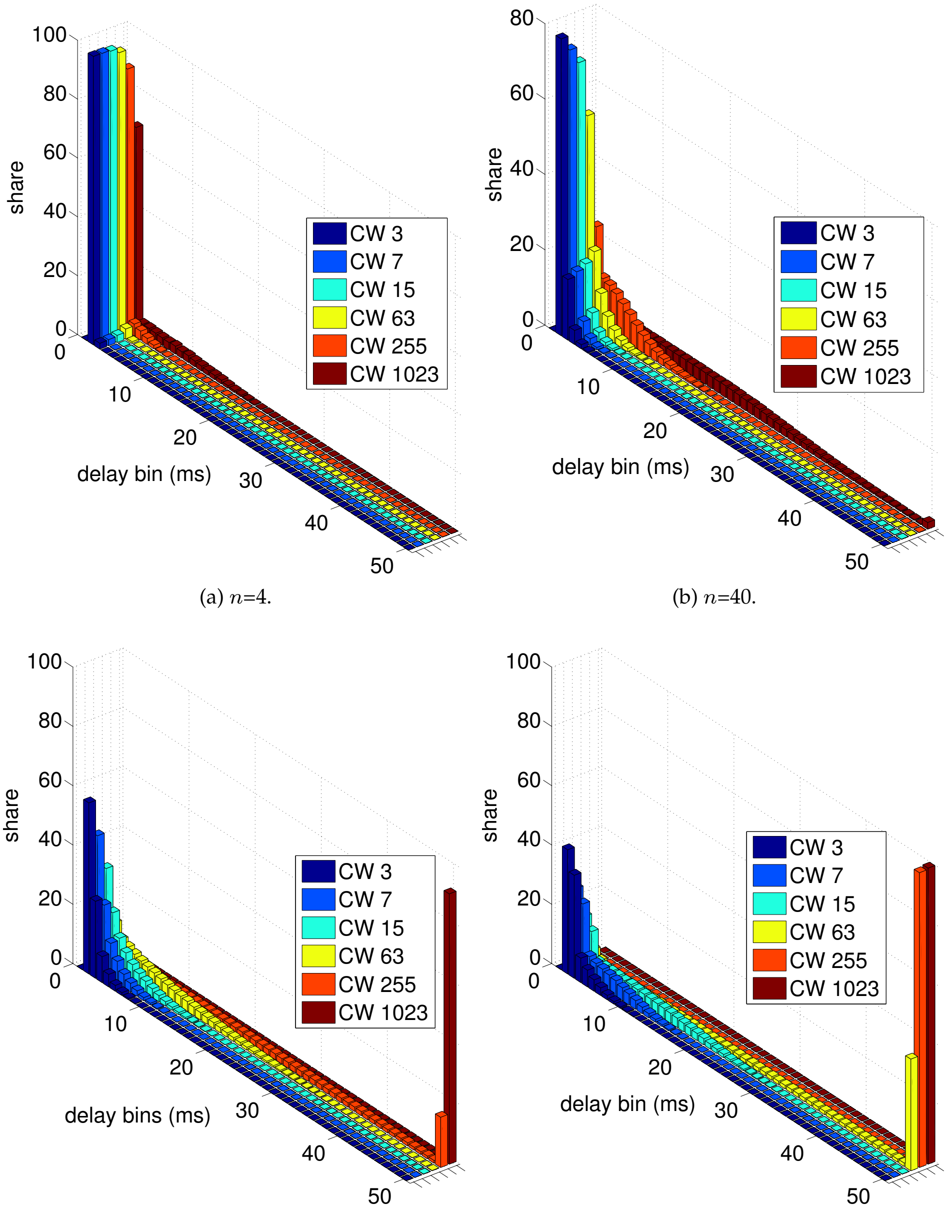
contention- and queueing delay is treated in detail in Sec. 6.3.

Our IEEE 802.11p implementation with a buffer size of 14 shows this only for a CW of 255 and 1023 for $\lambda_g = 10$ Hz, but as λ_g is raised further, the other CW values also start to show a rising delay. As a result, a node is only transmitting old information, even though newer information is available in the buffer. The study described in Sec. 6.3 addresses this issue.

We now address the delay distribution. For that purpose, all end-to-end delay measurements are grouped into bins of 1 ms, up to a maximum of 50 ms for the figures to remain tractable. Any overflow is grouped in the last bin, as was used before in Sec. 6.1.

Fig. 6.9 provides a set of histograms showing the distribution of delay for λ_g=10 Hz for various values of n. Note that a node with a larger CW (hence, larger mean backoff counter) has more opportunities of being interrupted while performing backoff countdown. As a result, the delay will be larger. The delay histograms plotted in Fig. 6.9(a) and (b) resemble a typical long-tailed distribution. Most received beacons have a delay below 2 ms. These are cases where a station performs carrier sensing, finds the channel idle, and immediately transmits. The tail contains transmissions which did not directly find the channel idle, and backoff was performed. This backoff may have been interrupted by other transmissions, a node will then freeze its bc. The result is a (potentially) very long delay if CW is large.

Now consider Fig. 6.9(b). We can observe that the smaller CW values have a greater proportion of their transmissions in the first interval. This indicates they experience a very short contention delay. In the case of larger CW values, more transmissions fall into later bins. The result is that the share in the first bin is smaller. Note the slight rise for CW=1023 in the last bin, in this case values exceeding 50 ms were measured.



(a) n=4.

(b) n=40.

(c) n=80.

(d) n=150.

Figure 6.9: End-to-end delay distribution, λ_g=10 Hz.

Figs. 6.9(c) and (d) show the delay distributions of 80 and 150 vehicles, respectively. What can be observed is that a node increasingly finds the channel busy and performs backoff instead of a direct transmission. Hence, the peak present near the origin in Fig. 6.9(a) lowers and spreads. The long tail, however, becomes heavier: larger delays occur more often.

This is praticular prominent for larger CW values, which have a very low percentage of short delays, indicated by the heavy weight of the tail. In these scenarios, delays in the order of seconds were observed (see Fig. 6.7), which are grouped into the last bin, which peaks in this figure. Note from Fig. 6.9(d), that large delays become even more prominent. However, the CW values 3, 7, and 15 still maintain a low delay, suggesting that larger CW values should not be used.

When using small CW values, an important observation also made in [108], is that delay requirements are easier to satisfy than P_s requirements. Delay requirements for safety-critical information are specified as 500 ms [166], which seems relaxed when compared with the 200 ms requirements of a real-time CACC system as described in Sec. 3.4. However, even 200 ms is not challenging in single-hop broadcasts, judging from the delay analysis described above.

6.2.3 Discussion on CW adaptation

The CW values studied in this experiment show an increase in P_s for larger CW values, which at a first glance makes large CW values seem desirable. However, it was shown that in these cases the gain in P_s is primarily due to the increase in P_{drop} caused by queue buildup. This discards some of the load supplied to the channel, and improves the success probability for those beacons that do not get dropped from the queue. The large CW sizes tend to cause a very long delay, resulting in queue buildup and excessive latency of the received information, to the point of beacons being outdated on arrival.

We can conclude that the currently used default CW size (i.e., 15) for beaconing is an adequate value: larger values impede the system's scalability because of an increase in end-to-end delay. Smaller values were found to have a reduced P_s because of the increased probability of choosing the same bc and causing collisions.

CW increase in Unicast works under the assumption that load on the channel is not constant, but of a bursty nature. If a node finds the channel busy, it will defer increasingly longer so the burst of transmissions by other nodes is at some time over. After that, there is a larger probability of finding the channel idle. In beaconing, however, the load is relatively constant. If a node is to perform carrier sensing at arbitrary moments, it would find the channel idle with the same probability. What a bc of 1023 actually does, is allow (or force) a node to wait 1023 slottimes (and potentially, equally many or more transmissions by other nodes) and then transmit with the same probability of success as when the backoff counter would have been 0, with only a moderate gain in P_s because of the reduced load caused by packets which are dropped from the queue.

The reason why a significant difference in P_s under varied CW was found in, for instance, [169, 190], is that these studies assume the periodic channel swichting proposed by IEEE 1609.4. This imposes a synchronisation point on a system that operates best under low synchronisation. All nodes are now synchronised, and rely on the CW to become desynchronised. In this situation, the CW increases time diversity by spreading the transmissions in time. This is more effective with larger CW size, because of the reduced probability of choosing the same bc and causing a collision. The impact of channel switching is further explored in Sec. 6.4.

In contrast to the suggestion in [10], adaptation of the CW does not yield the expected beneficial effects on P_s. However, it should be noted that in this research *all* nodes in a scenario used the same CW settings. This situation differs from Unicast, because there nodes will not all have the same CW at the same time, as such the different CW values give one node an advantage over the other. This is also the case if beacons would be transmitted in varying ACs, each with its respective parameters.

On average, delay is found to be short. However, in case of a congested channel, a node can spend a long time in contention. This time can even be so long (values in the order of seconds have been observed, confirming findings in [197]) that a new beacon has been generated in the mean time. This beacon has been added to the MAC queue and is waiting for transmission. In this case, from an application perspective, the beacon currently in contention is obsoleted by the new one in the queue. The result is an excessive delay and an increase in P_{drop}—a newly generated beacon finds the queue full and is dropped. This problem can be addressed in several ways: (1) make the queue small and drop an old beacon in stead of the new one (to be evaluated in Sec. 6.3), (2) make a preemptive backoff procedure, or (3) only generate a new beacon if the old one has left the MAC queue. In turn, a solution to this problem would also lower the supplied load on the channel, because the node does not have to perform contention for both the new *and* the old beacons. In a beaconing system, the default FIFO MAC queue with tail-drop policy reinforces the performance drop with respect to delay and packet loss.

Delay can be large, but for small CW values this only holds for a small percentage of the beacons. Most of the probability mass lies well within the inter-generation time τ_g as introduced in Sec. 3.5.3. Note however, that under increasing n the average end-to-end delay will rise. Whereas for sparse traffic the delay distribution approximates an exponential distribution, the distribution for more dense traffic is of a different kind with an increasingly heavier tail. This is important to keep in mind when analytically modelling delay of a beaconing system.

6.2.4 Conclusion

This study investigated whether increasing the contention window in reaction to an increased traffic density could improve beaconing performance. The effects on P_s and delay were researched in a simulation study. The conclusion is that the default CW of 15 does perform very well in practise. Larger values cause significant delay, although they do provide a slight increase in P_s due to an increase in P_{drop}. Smaller values reduce P_s due to the increased collision probability.

The analysis of P_s shows that IEEE 802.11p Broadcast operates best when the load on the channel is low, because performance drops rapidly with increasing channel utilisation. This occurs when either λ_g or the number of vehicles sharing the same collision domain becomes excessive. The delay analysis shows that, if a beacon is successfully received, it is on average less than 10ms old which satisfies the requirements of ITS applications.

An important observation is that the periodicity of beaconing implicitly retransmits data. A real-time system using samples derived from beacons should be able to operate under a wide variety of traffic densities if it can replay samples with a known delay and interpolate missing samples.

6.3 Buffering and scheduling beacon transmissions

In Chapter 5 we saw that the DCF shows significantly better performance in face of an increase in the number of nodes than the EDCA when assuming single-AC traffic for beaconing. This makes the DCF the better choice for dense traffic scenarios. One reason why the EDCA was adopted in the IEEE 802.11p specification is because it allows service differentiation based on ACs, giving one class of traffic an advantage (in both P_s and delay) over other classes. However, both the DCF and the EDCA have the same problem with respect to the order in which frames are transmitted, which is not solved by EDCA's QoS features. This problem is that, within a single queue, the transmitted frames will still be the oldest ones in the queue.

The general purpose FIFO queue as used 802.11 WLAN is part of the system's LAN legacy, where every frame will receive fair best effort service. Under EDCA, the amount of effort is differentiated, but it is still based on the same fairness principles.

Given the fact that a vehicular network differs from a LAN, we reason that it may be worthwile to adopt a special queueing strategy for beacon messages. Beacons have a strong temporal relevance, particularly with respect to the critical constraints of ITS applications such as CACC as described in Sec. 2.2. If frames must be sacrificed due to congestion on the channel and limited queue size, it is best to discard the oldest beacons in the queue, while trying to transmit the most recent ones. In order to address this problem, we propose a different queueing and scheduling paradigm for beacon frames. This section evaluates the impact of the default and the proposed queueing and scheduling paradigms of the transmission queue by means of a simulation study. The reason why we turn to simulation is that, although the model presented in Sec. 5 allows fast and detailed performance evaluation of the medium access mechanism, it depends on the assumption of an infinite FIFO queue.

First, the buffering and scheduling mechanisms will be described in Sec. 6.3.1 and Sec. 6.3.2. Related work is covered in Sec. 6.3.3. Next, the simulation experiments are described in Sec. 6.3.4. Results are described in Sec. 6.3.5, and Sec. 6.3.6 provides a discussion on the findings. Sec. 6.3.7 concludes this case study.

6.3.1 Buffering mechanisms

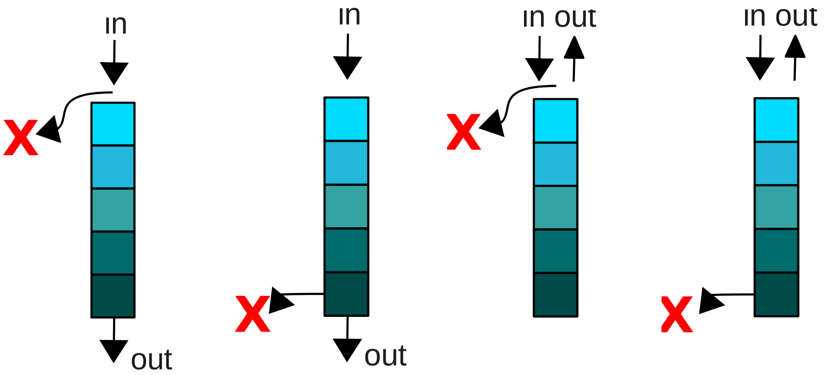
A buffering mechanism prescribes how an arriving element, e.g., a beacon frame, is handled depending on the state of the queue. Dropping a new arrival when the queue is full is called a tail-drop or drop-tail approach [198]. Another method often used in the Internet is Random Early Detection (RED) [199]. Under this regime, new arrivals are marked or dropped with a certain probability, depending on the utilisation of the queue. An empty queue will accept the packet with high probability. As the utilisation of the queue increases, the probability the packet is dropped increases as well. Although used successfully as a congestion avoidance method, it is still dropping new arrivals. We refer to these methods as Newest Packet Drop (NPD): out of all packets in the queue plus the new arrival, the newest one is dropped. In beaconing, this is very undesirable behaviour. It means that older packets are kept and packets containing updated information are discarded. The information carried by the newly arrived beacon is much more important than the information carried by the packets already in the queue [17]. The Oldest Packet Drop (OPD) [13] mechanism addresses this issue by dropping the oldest packet when the queue is full and a newer packet arrives.

6.3.2 Scheduling disciplines

A scheduling discipline prescribes the order in which buffered packets are served. Two well-known scheduling disciplines are FIFO and Last In, First Out (LIFO). A FIFO scheduler will serve the elements in the order in which they arrived; a property often advantageous in LANs and the Internet because of its implicit fairness. A LIFO scheduler will serve the most recently arrived element; in line with the goal presented above: to always transmit the most recent information from the queue. Note that the LIFO scheduler serves the most recently arrived element, and then takes the next-most recently arrived element. This could mean the next packet is older than the previous one. This information could still be useful, as the CACC may be replaying the received (timestamped) samples with a known delay for its estimators.

6.3.3 Related work

The analytical modelling approaches discussed in Sec. 5.1 all assume a G/G/1 or M/G/1 queue, or no queue at all. This is based on the assumption that the IEEE 802.11 system uses a FIFO queue. Liu *et al.* [155] observe that both infinite and queueless models are inaccurate, and instead explicitly model the queue length. However, they do not consider different buffering and scheduling disciplines than FIFO with NPD. Eichler evaluated the performance of IEEE 802.11p using the standard EDCA parameters in a quantitative way [200] and concluded that long queues can cause serious performance degradation. Suleiman *et al.* [197] note that a small transmission queue in IEEE 802.11 can increase saturation throughput in a WLAN. Based on the



(a) FIFO, NPD (b) FIFO, OPD (c) LIFO, NPD (d) LIFO, OPD

Figure 6.10: The four studied configurations.

number of neighbours, nodes around a base station can optimise their saturation throughput. This study also only considers a FIFO, NPD queue.

Kaul *et al.* [195] address the issue of minimising end-to-end delay by means of a rate adaptation mechanism above the MAC layer, based on the observation that minimal delay can be achieved at an optimal operation point with respect to channel load. A brief comparison of FIFO and Latest Out (LO), a regime which always selects the most recent information for transmission, shows potential gains in minimising queueing delay. In [195] it is reasoned that hardware adaptations are needed in order for a LO scheme to work, so it is treated as hypothetical.

6.3.4 Experiment description

A transmission buffer can be implemented in two ways: (a) the network layer places packets in a queue and the MAC takes the packets one by one from the queue. It will perform contention, hand the packet to the physical layer as soon as it has won contention and will then take the next packet from the queue to contend for, or (b) the packet remains in the queue while the MAC performs contention. Once the MAC has won contention, it removes the packet from the queue and sends it to the physical layer. The Mac80211 implementation in MiXiM (see Sec. 2.7) uses method (b). As such, one could reason the head of the queue to be a special place, as it can hold a packet awaiting service, or a packet receiving service if one is to regard the contention as the service.

For these simulation experiments the simulation platform described in Sec. 2.7 is used in conjunction with the extensions covered in Sec. 5.5.1. The four studied scenarios are combinations of the two scheduling disciplines FIFO and LIFO, and the two buffer mechanisms OPD and NPD, as illustrated in Fig. 6.10. In this figure, the scenarios differ in from which position in the queue a packet is retrieved for

Parameter	**Value**
Beacon generation rate λ_g	25 Hz
Beacon size	3200 bits
CW_{min}	15
Data rate	3 Mbps
IFS	176 μs

Table 6.4: Simulation parameters for the buffering and scheduling study.

Parameter	**Value**
Number of nodes n	[10,20,40,80,120,160]
Queue size	[1,2,5,10]
Scheduling	[FIFO, LIFO]
Buffering	[NPD, OPD]

Table 6.5: Varied parameters for the buffering and scheduling study.

service, and which packet is dropped when the queue is full, indicated by the red X. The scenario (FIFO, NPD) can be considered to be the default DCF and EDCA behaviour. Note that for a frame, entering the queue is not a guarantee of being transmitted in case of OPD: if the frame resides in the queue for an extended period of time, it can be dropped in favour of more recent information.

Also note that we do not implement pre-emption, in which case even a frame in contention could be dropped (or rather, be replaced) in favour of a new arrival. Pre-emption may increase performance even further, but would make an unfair comparison with the NPD scenarios, as NPD would never drop the frame currently in contention. Given the special nature of the head of the queue as implemented in MiXiM, the studied OPD mechanism does not drop the packet currently undergoing contention, but it drops the packet next in line.

For these experiments we do not consider the periodic channel switching as proposed in IEEE 1609 [3], but assume instead that a station always resides on the same channel. The capacity of the channel is already very limited without the channel switching [9, 17], see also Sec. 6.4.

Beacons are generated at a rate of 25 Hz [23] by means of a Poisson process. This generation rate yields a high utilisation, where the effects of the different scenarios is best visible. Other properties are the same as used in earlier studies in this dissertation, and listed in Table 6.4 or specified in the standard [81].

Nodes remain stationary and are within each others range, i.e., they form a fully-connected network. There is no chance of loss due to propagation effects (i.e.,

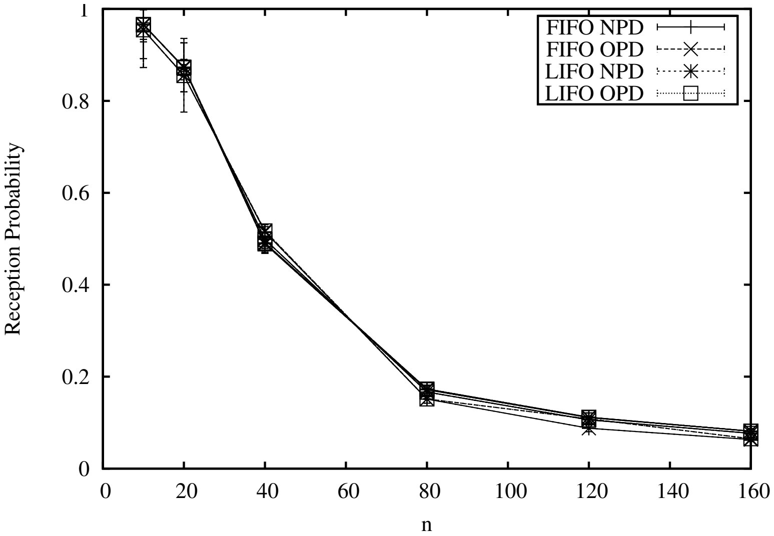


Figure 6.11: P_s for all studied queue sizes, buffering and scheduling mechanism, and varied number of nodes.

we use a Unit-Disc propagation model). Thus, we have a scenario without hidden terminals and the only possibility of loss after transmission is collision loss. Though not directly applicable to reality, it does give insight in how CSMA/CA and the studied buffering and scheduling mechanisms perform.

For all simulation runs the parameters that are not changed during the simulations are listed in Table 6.4. The varied parameters are listed in Table 6.5. The number of vehicles is varied and ranges from 10 to 160 nodes, where 160 nodes within range is a realistic assumption for congested highway scenarios.

6.3.5 Simulation results and analysis

The metrics used in this study were introduced in Sec. 3.5, here we evaluate the reception probability P_s, the mean end-to-end delay and its components: the mean queueing- and contention delay. In addition, we evaluate P_{drop}, the probability that a frame is dropped because of a full queue. Every scenario is simulated multiple times using the method of independent replications [88]. No results are collected during the simulation's transient phase, and 95% confidence intervals are very small, in the plots barely visible.

Reception probability As n increases, so does the supplied load to the channel. The result is an increased collision and dropping probability. This causes a reduced probability of successful reception, similar to observations in Chapter 5.

Fig. 6.11 shows P_s for the four studied scenarios. The lines only show minor

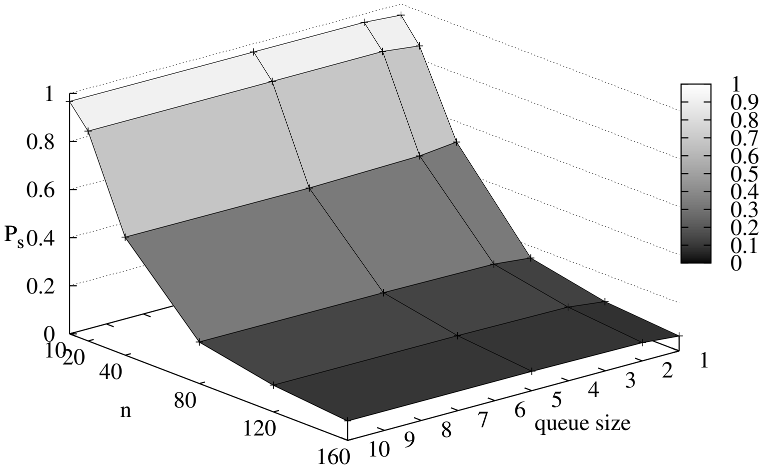


Figure 6.12: P_s for FIFO NPD, varied queue size and number of nodes.

differences which are not statistically significant, leading to the conclusion that the nature of the queue has little impact on P_s. A simple explanation is that, regardless of which frame is selected to be dropped or to be transmitted, the circumstances in the studied scenarios with respect to channel load are equal. Regardless of dropping frame A or B, a frame will be dropped. P_s only concerns whether beacons are lost or received, and not which ones.

This means that the average frame undergoes the same probability of collision and queue drop as any other frame, regardless of the time spent in the queue. Fig. 6.12 shows P_s versus n and queue size, specific for the FIFO, NPD configuration. The plot shows that the size of the queue has very little impact on P_s. The results for the other three configurations are very similar.

Mean end-to-end delay Fig. 6.13 shows the average end-to-end delay of the received beacons for a queue of size 10, Fig. 6.14 shows the mean end-to-end delay for a queue of size 2. As n increases—and hence the congestion on the channel—delay also increases. Comparing the plots, it becomes clear that a small queue leads to low end-to-end delay.

Fig. 6.13 indicates a difference between FIFO and LIFO for high n, however, the difference between NPD and OPD is small. Discarding the oldest or the newest packet upon queue drop has little effect because of the relatively large number of other packets in the queue collecting delay. With increasing n, the order in which packets are transmitted begins to dominate. Here we see that both LIFO schemes exhibit larger mean end-to-end delay than the FIFO schemes: they allow packets to age at the back of the queue while transmitting fresh information. This is because of the depth of the queue, and because the load on the channel is relatively constant.

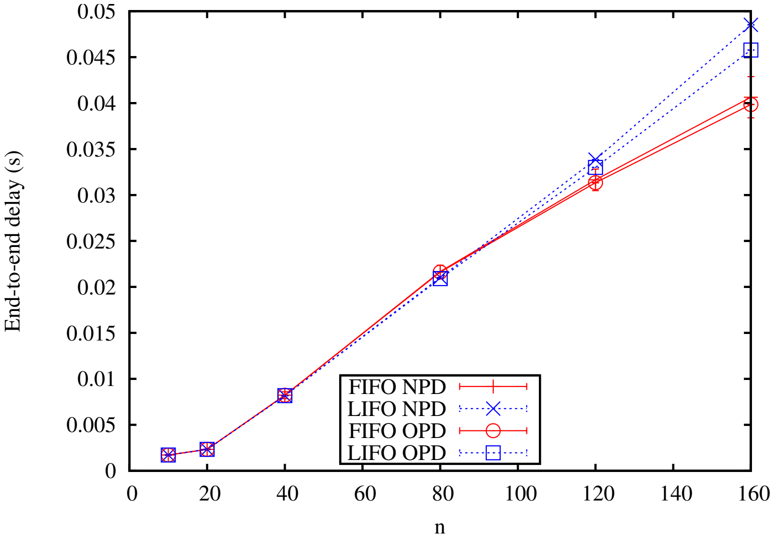


Figure 6.13: Mean end-to-end delay for queue size=10.

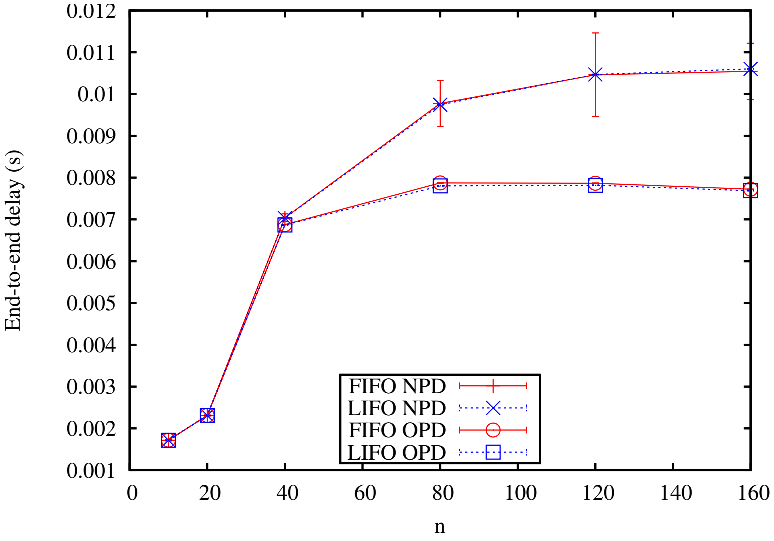


Figure 6.14: Mean end-to-end delay for queue size=2.

As a result, a node hardly gets an opportunity to transmit old packets.

The difference between NPD and OPD clearly manifest for the small queue of size 2 in Fig. 6.14: the OPD configurations level at a value significantly lower than NPD. The levelling of the delay can be explained by the very small buffer, which causes many packets to be dropped at the queue. While influx of packets in a node remains constant under increasing n, the total service time (the duration of the contention and transmission) increases due to the increase in bc freezes. This results in fewer packets being transmitted onto the channel. Dropping packets from the queue reduces congestion, and explains why [197] was able to improve throughput by reducing the size of the buffer: the supplied load to the channel was reduced by dropping packets from the queue, and hence P_s increased. A similar behaviour was observed in Sec. 6.2 for large CW values.

The difference between OPD and NPD in Fig. 6.14 can be attributed to *which* packets are dropped from the queue. In case of NPD we drop the new arrival, hence allowing the packets remaining in the queue to age further before transmission. In the OPD case, the oldest packet is removed from the queue and replaced by the new arrival; hence, the OPD scheme is able to achieve a lower end-to-end delay.

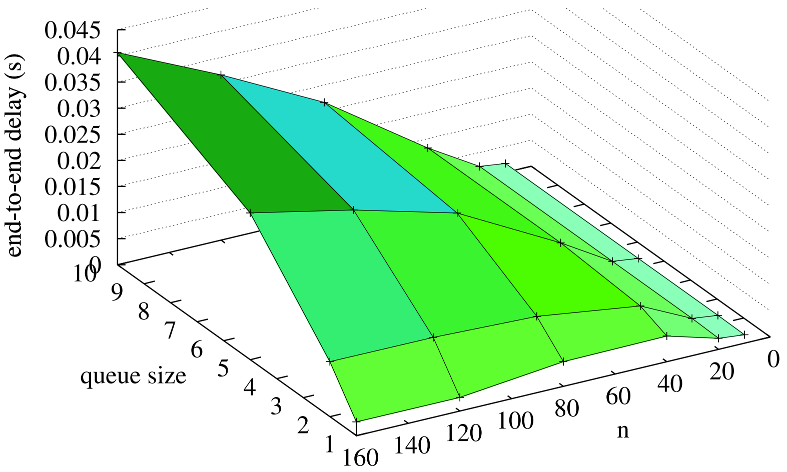
Figs. 6.15 and 6.16 show the mean end-to-end delay of the FIFO and LIFO scenarios under varied n and queue size. Observe the rising delay for larger queues and the leveling delay for small queues, or even diminish for a queue of size 1, this is due to the collision bias described in Sec. 5.3.9.

Queueing delay The difference in mean end-to-end delay between the four configurations can be attributed to the time a beacon spends in the queue, as we will show here. Under the OPD regime, the oldest beacon is dropped in favour of the newest when queue drop occurs. The result is a shorter average time spent in the queue for beacons which are transmitted when using a small queue, as visible in Fig. 6.17. This figure plots contention delay (denoted as Cdel) and queueing delay (denoted as Qdel) together to show their relative magnitude. Note that the difference FIFO / LIFO is not as significant as the difference between NPD and OPD. In this case, OPD provides a reduction of queueing delay of up to 30% compared to NPD.

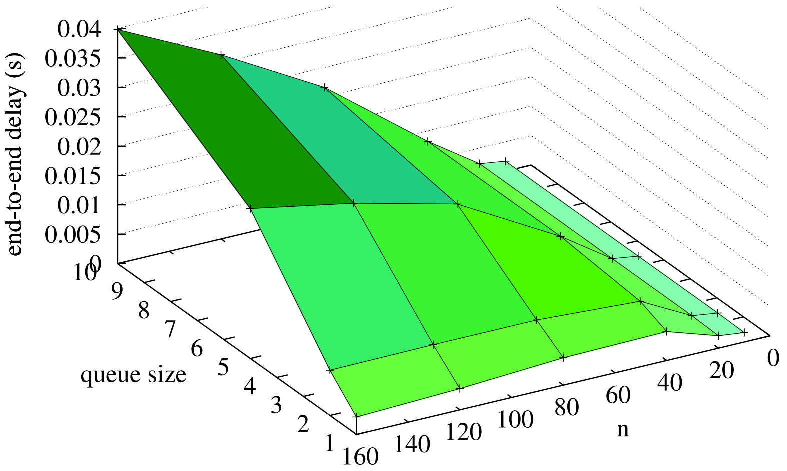
Instead of dropping the new arrival and allowing packets in the queue to age, the OPD scheme drops the oldest packet in the queue, resulting in a lower average age upon being serviced (i.e., being taken by the MAC to perform contention for this packet). Results for larger queues show that queueing delay increases with the depth of the queue, visible in Fig. 6.18 for a queue of size 10. The difference between OPD and NPD is less pronounced and FIFO obtains better results than LIFO, as also found in the discussion of end-to-end delay above, visible in Fig. 6.13.

Contention delay The queueing and scheduling mechanisms have little to no impact on the contention delay, as visible in Figs. 6.17 and 6.18, where all four lines overlap. Note that contention delay increases with increase of the supplied load on the channel as the number of nodes increases, and drops and levels when the channel goes into saturation and frames start being dropped at the queue.

This reduction in contention delay can be attributed to the bias present in the DCF, caused by the streak phenomenon described in Sec. 5.5.2, because this metric is only calculated for received beacons. Also note that for smaller queues, contention delay dominates and for larger queues, queueing delay dominates the end-to-end delay.



(a) FIFO NPD.

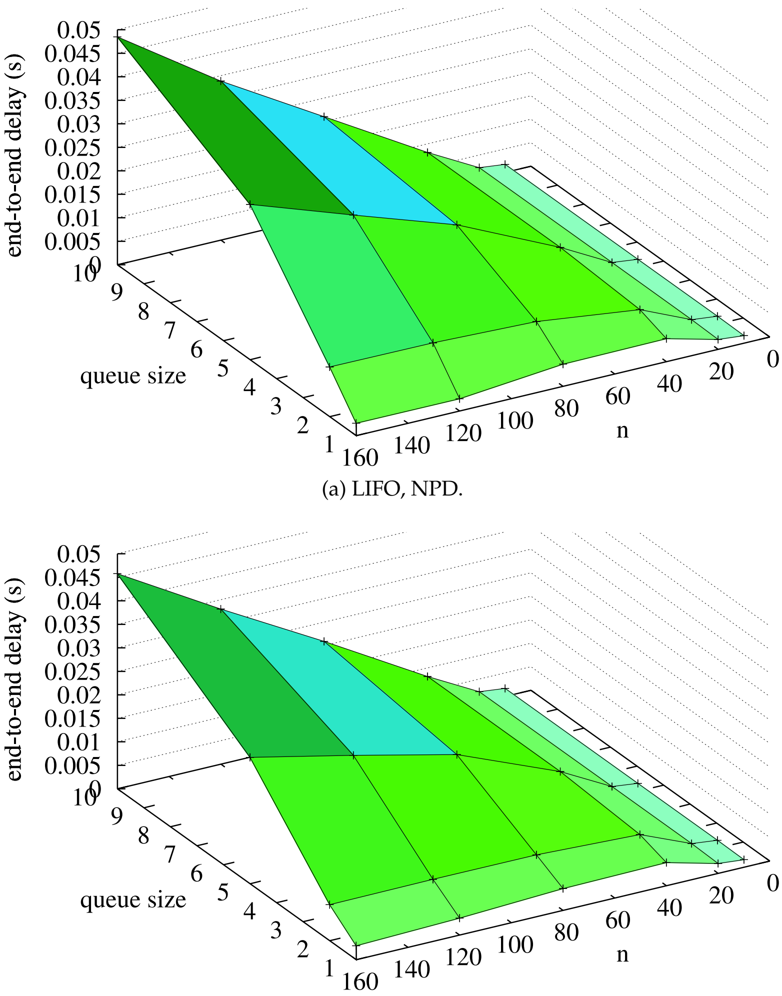


(b) FIFO OPD.

Figure 6.15: Mean end-to-end delay for the two FIFO scenarios.

Dropping probability Since OPD selects different packets for dropping than NPD, it is in situations with increasing dropping probability where the resulting effects manifest. The dropping probability is shown in Fig. 6.19. All four configurations exhibit similar dropping probability behaviour. This is because all scenarios experience the same channel conditions; the resulting congestion on the channel causes an increase in service time as observed in Sec. 5.5.2. As n increases, P_{drop} rises because contention takes longer and queue buildup occurs, causing packets to be dropped.

It should be noted that a high dropping probability has a congestion-controlling effect; using OPD ensures that the beacons which *do* get through still have low mean



(a) LIFO, NPD.

(b) LIFO, OPD.

Figure 6.16: Mean end-to-end delay for the two LIFO scenarios.

end-to-end delay. Considering Fig. 6.3 and 6.4, which show that P_s was insensitive to queue size, and Fig. 6.19, which shows that a smaller queue has a larger dropping probability, it becomes clear that in order to maintain the same P_s, a reduced collision probability must hold for the situations where queue drops occur. This becomes apparent when plotting the total precentage of time a node spends transmitting, presented in Fig. 6.20. Here, it is visible that for a smaller queue with increasing n, the time a node is transmitting is reduced: packets are dropped before they enter contention, and will not be transmitted onto the channel.

Not only do smaller queues reduce the average delay of the received beacons,

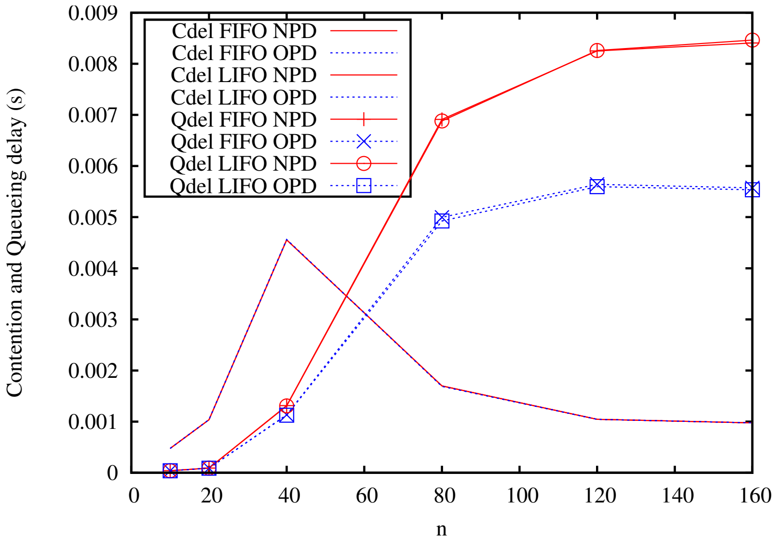


Figure 6.17: Mean contention and queueing delay for received beacons, queue size=2.

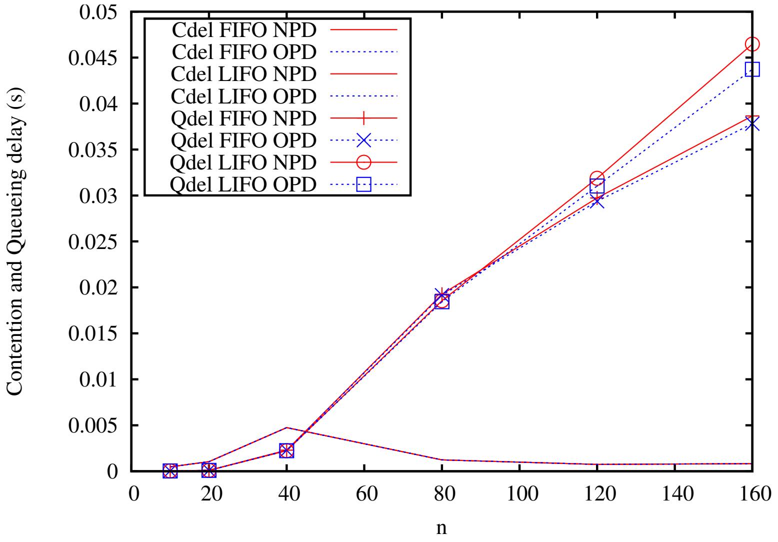
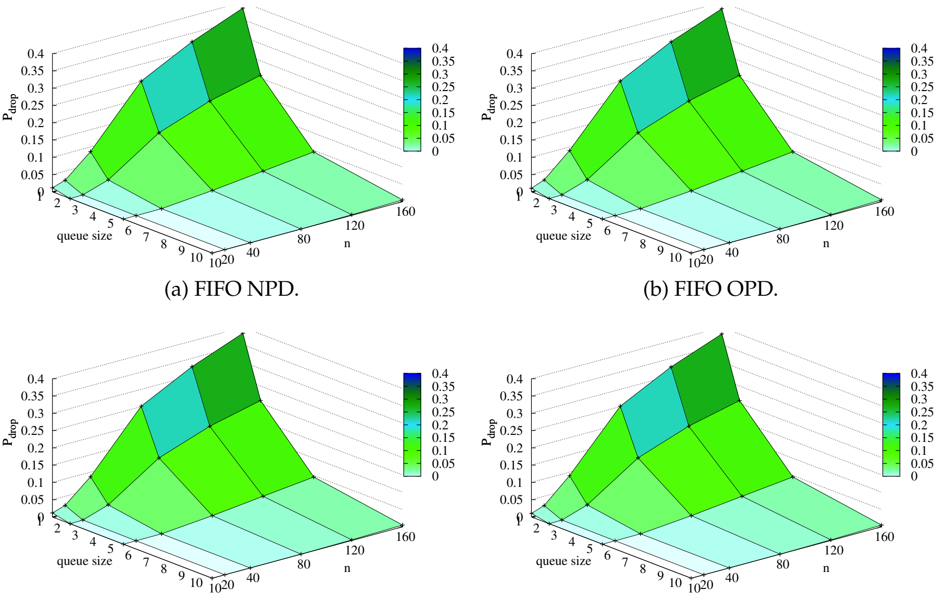


Figure 6.18: Mean contention and queueing delay for received beacons, queue size=10.

they also allow better scalability because of their congestion-mitigating properties: if a beacon is dropped from the queue, it does not influence successful reception of other beacons.



(a) FIFO NPD.

(b) FIFO OPD.

(c) LIFO NPD.

(d) LIFO OPD.

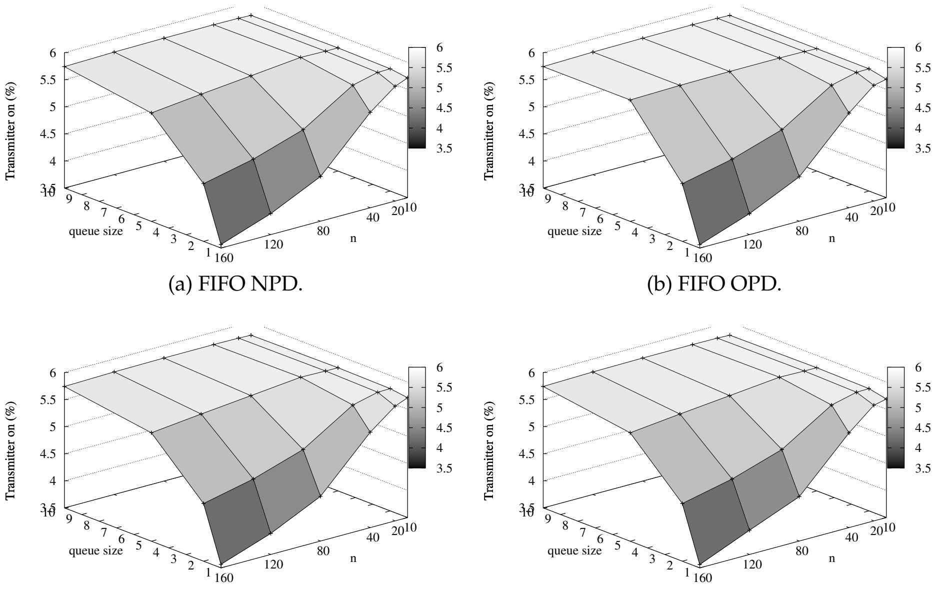
Figure 6.19: Dropping probability, varied n and queue size, for the four scenarios.

6.3.6 Discussion on scheduling and buffering strategies

Many works assume the IEEE 802.11p MAC to be fixed and set in stone, and try to solve the problems of congestion and delay at the network layer, because otherwise changes to hardware or drivers will be required. This is regarded as an obstacle, but should be considered if the gain is worth the effort.

Comparing LIFO and OPD to the default FIFO NPD in the context of beaconing, we reason there is an opportunity for a significant performance gain if we were to venture into the necessary changes and add a special beacon queue with its own properties. A small queue size is desirable, as queueing delay increases with the depth of the queue. Furthermore, a small queue which is full because of congestion on the medium performs implicit congestion control by dropping packets at the queue. The OPD policy sees to it that, when packets need to be dropped, it is *not* the newest packet which is discarded.

The question which then arrises is, why not remove the queue altogether? A queue serves to accept burst arrivals, but in the context of beaconing, it can also accept the difference in service time due to the dynamic nature of vehicular networks, and the more-or-less periodic arrival of messages from the network layer. If there is no queue, a node would throw away many beacons while those that get transmitted have a probability of collision. Keeping in mind the use of estimators by the control applications, it may be worthwile to transmit more information if the current channel



(a) FIFO NPD.

(b) FIFO OPD.

(c) LIFO NPD.

(d) LIFO OPD.

Figure 6.20: Average percentage of time spent transmitting, varied n and queue size.

state allows for it – the estimator may be able to also incorporate some recent history in order to obtain a better estimate.

We could not find a clear distinction between FIFO and LIFO, as the metrics observed in this research considered mean values. It is to be expected that the distributions of the delay will show a difference between FIFO and LIFO. In essence, a LIFO queue always sends the most recent packet for transmission; however the packets which remain in the queue can accumulate considerable delay. Future work should consider the distribution of the delay.

If a network layer-level congestion control scheme detects congestion on the medium and is reducing its λ_g, the buffer may still contain packets to transmit. As such, there is a lag in the reaction to the load: there may be a reduced influx of arrivals but those present in the queue still await service. A possible way to solve this would be to dynamically shrink and grow the transmission queue, or drop arrivals as is done in RED [199]. If, in order to achieve the new specified size, a packet must be dropped, the system should drop the oldest packet. However, this needs further investigation.

6.3.7 Conclusion

We provided an analysis of the impact of the standard FIFO NPD behaviour of the IEEE 802.11p MAC transmission queue on the performance of the exchange of beacon messages for use in networked control systems like CACC. A different scheduling discipline (LIFO) and a different buffering mechanism (OPD) were evaluated, studying various transmission queue sizes. The systems were evaluated under a varied number of nodes, simulating traffic densities as they can arrise in day-to-day traffic.

The OPD buffering mechanism compares favourable to the NPD mechanism with respect to delay of the received beacons, primarily for small queues. This is because of a reduction in queueing delay under near-saturation conditions when the wireless channel becomes congested. As such, the transmitted beacons are always the freshest available. In case of a full queue, it is the oldest information which is dropped. This is favourable, as this information may have already lost its relevance and may be dropped at the receiver anyway.

The LIFO scheduler compared to the FIFO scheduler makes little difference with respect to delay for small queue sizes, although for larger queues the depth of the LIFO queue causes accumulation of large delays on packets deep in the queue. From all evaluated metrics, the application of OPD results in an improvement in the end-to-end delay while keeping the other metrics around the original level.

The OPD buffering mechanism in conjunction with a FIFO scheduler and a small transmission queue can improve the end-to-end delay of received information for beaconing by means of a reduction in queueing delay for a number of nodes ranging from small to large.

6.4 Beaconing under channel switching constraints

The method of operating on multiple channels by means of channel switching as defined in IEEE 1609.4 was described in Sec. 2.3.4. A single transceiver spends 50 ms on the Control Channel (CCH), then spends 50 ms on a Service Channel (SCH), and then it returns to the CCH. Intuitively, channel switching with a single radio implies less time on a single channel. The reduced time a node spends on a single channel has a negative impact on the performance of communication on that channel.

An interesting observation in the literature on the impact of channel switching is that none of the solutions in literature focus on minimising the impact of the IEEE 1609.4 channel switching on the CCH performance. All methods focus on optimising for the SCH [17], some even at the expense of CCH performance.

For example, studies in e.g., [201, 202, 203], consider that the IEEE 1609.4 channel switching scheme causes a *bandwidth wastage* problem for the SCH. What is happening on the CCH in the mean time is disregarded. The reasoning is as follows. When a packet transmission is planned to take place near the end of the SCH interval, and the estimated transmission time for this packet exceeds the residual time of the current interval, then the IEEE 1609.4 standard recommends that the transmitting node should prevent sending out this packet during the current interval. Instead, it should send the packet during the *next* interval, leaving the end of every SCH interval underutilised, and causing delay in the transmission.

Parameter	Value
Beacon generation rate	10 Hz
Data rate	3 Mbps
EDCA class	AC0
Beacon size	3200 bits
CWmin	15
Scheduling	FIFO

Table 6.6: Simulation parameters for the IEEE 1609.4 study.

The real problem, however, occurs on the CCH, and causes a serious reduction in beaconing performance. Campolo *et al.* [169, 68] analytically evaluate the impact of channel switching on beaconing performance. Initially the evaluation was based on the assumption that all nodes immediately try to transmit their beacon at the beginning of the CCH. Later, in [68] this synchronised approach is compared to scheduling at a random moment in the CCH, the moment drawn from a uniform distribution. The uniform jitter shows good performance compared to the synchronised approach. In [204] a method to coordinate the temporal spread of access attempts, as opposed to random assignment, is addressed using a so-called sowing calendar. A schedule is derived from GPS data to minimise the collision probability.

But what exactly is the impact of channel switching in the first place? This section provides an overview of the simulation experiments in which the impact of the IEEE 1609.4 channel switching procedures on the beaconing performance is analysed. The experiment setup is described in Sec. 6.4.1. The results are presented in Sec. 6.4.2, followed by a discussion in Sec. 6.4.3. Sec. 6.4.4 concludes this section.

6.4.1 Experiment description

In these experiments, the simulation platform described in Sec. 2.7 is used in conjunction with the extensions described in Sec. 5.5.1. It is assumed that all nodes are within each other's range, their position does not change and there are no hidden terminals. Due to the Unit-Disc propagation model, relative position and distances between nodes have no impact. Also, just like in the previous simulation studies in this chapter, it is assumed that the communication channel is a 'perfect channel', no bit errors can occur during transmission. This means that packets can only be lost due to a collision.

Two scenarios are analysed in these experiments: Scenario one implements the operation of the CCH as defined in IEEE 802.11p. In this scenario a node resides on the CCH permanently for beaconing, denoted as the *continuous* scenario. In the second scenario, a node switches every 50 ms between the CCH and the SCH, as specified in IEEE 1609.4, yielding 100 ms intervals. This is the *alternating* scenario.

-

Parameter	Value
Number of nodes	[10, 20, 30, 40, 60, 80, 120, (160, 200 only for continuous mode)]
Channel mode	[continuous, alternating]

Table 6.7: Varied parameters for the IEEE 1609.4 study.

The simulation experiments are performed using the parameters tabulated in Table 6.6 and Table 6.7. Beacons are generated with $\lambda_g = 10$ Hz, coinciding with the C&D and GCDC projects and sufficient to operate CACC on. In the continuous scenario, the beacons are generated at a moment randomly drawn from a uniform distribution $[0, 100\text{ms}]$ every interval. In the alternating scenario, the moment of beacon generation is restricted to the CCH guard and CCH periods of an interval, minus the duration of one transmission. This is described in detail in Sec. 2.3.4.

In the alternating scenario, the moment of generation is drawn randomly from a uniform distribution $[0, T_{cch} - T_s]$. Here T_{cch} stands for the CCH interval time, namely 50 ms. and T_s for the duration of the transmission of a single beacon frame, see also [68]. This method is similar to the one used in [169]. It is not necessary to model traffic in the SCH, because the CCH and SCH have separate transmission queues and their time-division nature prevents any interference [69].

A data rate of 3 Mbps is used for all nodes, and beacons are of EDCA class AC0, but using DCF bc blocking behaviour. Transmissions are performed using FIFO scheduling. The number of nodes is varied from 10 to 120 nodes, where 120 nodes is a realistic number of nodes in range during rush hour on a highway. Note that for the continuous scenario 160 and 200 nodes were also considered to achieve comparable metrics.

6.4.2 Simulation results and analysis

In this section the results of the simulation experiments are reported. In this study we evaluate the impact of channel switching on P_s, mean end-to-end delay, and channel utilisation, as introduced in Sec. 3.5. The reported plots include very benign 95% confidence intervals.

Reception probability As discussed earlier throughout this thesis, the load on the channel increases as n increases. This results in an increased collision probability.

In Fig. 6.21, the reception probability is plotted against n, for the continuous and alternating scenarios. As can be observed, the choice of channel access scheme has a significant impact on the reception probability. In the alternating scenario, P_s is lower than in the continuous scenario, showing a drop starting from only 10 nodes. This observation was expected and is due to the fact that the effective load on the

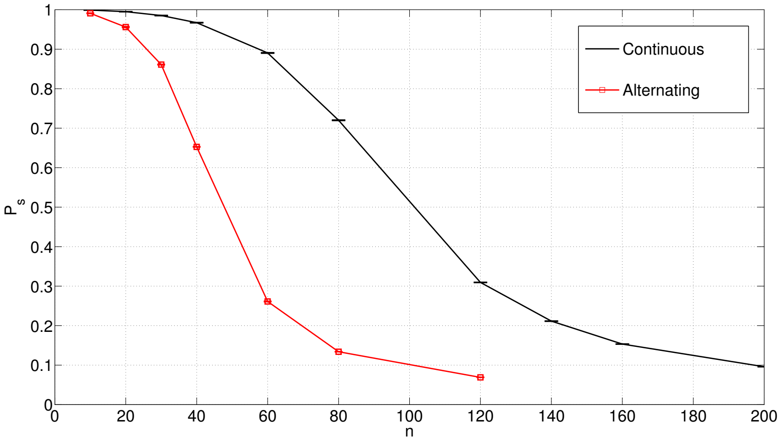


Figure 6.21: Reception probability P_s for Continuous and Alternating access.

channel significantly increases when the alternating scenario is used, since a node has to transmit the same number of beacons in less than halve the time compared to the continuous scenario. This results in a reception probability which appears to decrease twice as fast for the alternating scenario. In other words, alternating between CCH and SCH dramatically reduces the structural scalability of a beaconing system when compared to continuous access.

Mean end-to-end delay Fig. 6.22 plots the average end-to-end delay for received beacons versus n for the continuous and alternating scenarios. The average beacon end-to-end delay increases as the number of nodes increases. In the alternating scenario, the average end-to-end delay increases faster than in the continuous scenario. This is because in the alternating scenario, nodes have to send the same number of beacons in less than halve the time, resulting in bursts of congestion on the channel. This, in turn, causes an increase in contention delay. Also, it can happen that a beacon arrives in the queue at the end of an interval, and has to wait for the next CCH interval to finally be transmitted.

For the alternating scenario, the mean end-to-end delay decreases as n exceeds 60 in Fig. 6.22. The delay is even lower for the alternating scenario compared to the continuous scenario when more than 120 nodes are using the channel. This effect can be explained if the results are compared to P_s in Fig. 6.21. Only beacons which are received successfully contribute to the mean end-to-end delay. This counter-intuitive phenomenon was described in detail in Sec. 5.5.2, and can be attributed to the collision bias caused by the streak behaviour caused by the DCF bc blocking. In this case, transmissions from contention have a higher collision probability than those transmitted directly, and these have acquired less delay.

The same effect also occurs for continuous mode (see Fig. 6.22), but at a much larger value of n. This means the saturation point of the channel has shifted. The con-

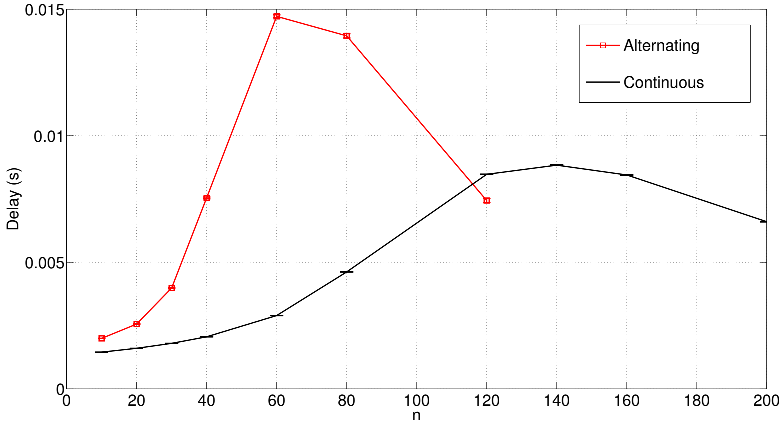


Figure 6.22: Average end-to-end delay for Continuous and Alternating access.

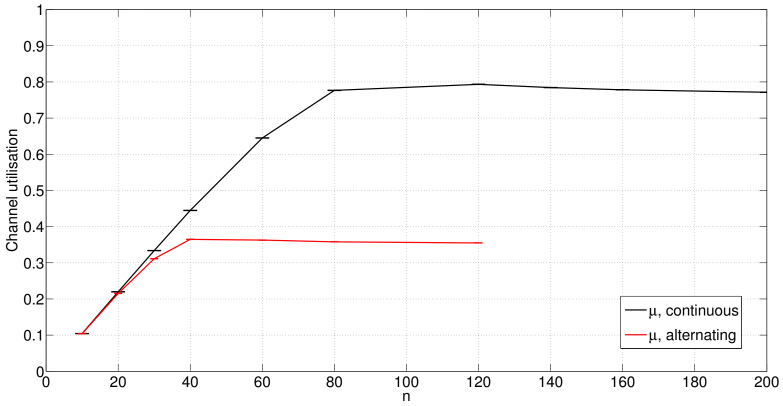


Figure 6.23: Channel utilisation for Continuous and Alternating access.

tinuous scenario uses the available channel resources more efficiently and therefore congestion occurs for larger values of n.

Channel utilisation The inefficient use of the channel becomes particularly clear in Fig. 6.23, which shows the mean μ for both the continuous and the alternating scenario. This figure also shows that the alternating scenario is saturated at lower n than the continuous scenario, because μ levels. In addition, the figure makes clear that the continuous scenario is able to achieve a higher utilisation of the available channel, the alternating scheme wastes more than 60% of the channel resources. Note that in the time domain the alternating scenario shows alternating busy and idle intervals, which is averaged into the mean in Fig.6.23.

6.4.3 Discussion on IEEE 1609.4 channel switching

Channel switching severely reduces beaconing performance. As opposed to the approximately 80 nodes which could have acceptable reception rate λ_r under continuous channel access, this number is reduced to below 40. In addition, it is interesting to note that all proposed methods in the literature [17] focus on increasing SCH performance, and some do so at the expense of CCH performance. In the light of beaconing performance, this can have catastrophic impact on the quality of the cooperative awareness. It should be stressed that a node not only needs to transmit its own beacon, but it should also be able to receive beacons sent by *all* neighbouring nodes for the concept of cooperative awareness to work. This is the *raison d'être* for the synchronised "common" CCH. Some of the schemes discussed in [17] seem to have been designed without this requirement in mind.

Accurate information on the dynamics of the vehicle in front is imperative for ITS applications for collision-avoidance or cooperative following to work. This fact renders methods optimising SCH performance at the cost of CCH performance useless in the dynamic vehicular network, where receiving up-to-date beacons is vital. In fact, the entire concept of channel switching becomes questionable when taking its significant drawbacks into account.

The problem of multi-channel access with a single-radio device while still maintaining high availability of the CCH does not exist in a dual-radio setup, where one radio is continuously tuned to the CCH and the other can be used for activities on any of the service channels. Although challenges related to increased cost and adjacent channel interference exist [205], it is advisable to explore other solutions than alternating between channels in a time-division manner because this severely limits the capacity of a system which already is capacity-limited in dense traffic. The use of dual-radio devices is considered by ETSI [206].

6.4.4 Conclusion

We performed an analysis of the beaconing performance of IEEE 802.11p when using the channel switching procedures proposed in IEEE 1609.4. Both the continuous scenario and alternating scenario were evaluated in a simulation study, using a varied number of nodes.

The periodic channel switching results in significantly lower P_s, which is reduced by up to 75% compared to continuous access. In addition, an increase of mean end-to-end delay of beacons by a factor of four is observed, already for a small number of nodes. This can be attributed to the reduction in effectively available channel resources, which leaves only 46% of the channel compared to continuous channel access. Under the CSMA access mechanism, a μ of only 0.36 can be realised, versus a μ of 0.8 under continuous access. The conclusion is that channel switching severely impacts the structural scalability of a beaconing system. This renders it less capable of handling the increased load associated with high vehicle densities which can arise in practise.

A recommendation for future work is to develop a scheme that could minimise the impact of the channel switching and improve the beaconing (i.e., CCH) performance, if needed at the cost of SCH. Better even, is to use multi-channel radios or wide-band receivers, and abandon channel switching with a single radio completely. This seems more in line with meeting scalability and dependability requirements of the applications which increase traffic performance and safety, and require a good cooperative awareness to do so.

6.5 Concluding remarks

This chapter reported on the studies published in [12, 13, 17]. Although expected in [10], changing the CW to deal with an increase in n in an adaptive beaconing system holds little promise. This is because the CW is more effective in Unicast with burst arrivals than a beaconing scenario with more or less constant load resulting from the periodic Broadcast transmissions. Although larger CW values *appear* to result in a higher P_s, this is because a significant share of frames is dropped. Those beacons which are received exhibit a very large end-to-end delay, exceeding values of 1 second. Using heterogeneous CW values was not studied and may be interesting future work.

A small transmission queue seems optimal for beaconing. In case of congestion on the channel, dropping beacons at the transmission queue has an implicit congestion-controlling effect. Transmitting this frame would have a high probability of collision; preventing transmission by dropping the frame from the queue reduces the collision probability of other transmissions.

Because of the real-time nature of beacon messages, a new arrival is an implicit retransmission of the previous beacon. The new arrival should replace the frame currently in the queue, and the old information should be discarded. This procedure departs from the traditional NPD policy and is referred to as Oldest Packet Drop (OPD) in [13]. We have shown a significant performance gain of OPD over NPD.

The analysis of the impact of channel switching on beaconing clearly shows that it severely impedes P_s and raises delay. Research in the direction of multi-radio systems is encouraged as an alternative to channel switching with a single radio.

CHAPTER 7

Conclusions

In this dissertation, the performance of a beaconing system for cooperative adaptive cruise control has been evaluated by means of simulation experiments, analytical modelling, and proof-of-concept implementations. The goal was to obtain a comprehensive insight in performance-limiting factors. The envisioned application is that of real-time vehicle control, or CACC. This is an application with particularly stringent requirements. The reasoning is that if these requirements can be met, we can easily design ITS applications with more lenient requiremens if they are following the same cooperative awareness paradigm. This chapter presents an overview of the results of the research in this dissertation in Sec. 7.1. Sec. 7.2 provides recommendations how the scalability of vehicular networking can be improved. Finally, Sec. 7.3 presents directions for future work.

7.1 Results and conclusions

The automobile industry is currently undergoing a paradigm shift from autonomous, vehicle-centric to cooperative, network-centric vehicles. In a world where people and devices are networked to an ever-increasing degree, their transportation systems were slow to follow but are rapidly catching up. Vehicular networking is an important enabling factor in realising the intelligent car of the future, embedded in, and communicating with, the world around it.

A vehicular network is not a typical general purpose network. Even though the proposed MAC and physical-layer technology is based on standards applied with great success in the WLAN domain, its behaviour in vehicular networking is significantly different.

Chapter 3 described the construction of a cooperative awareness by means of beaconing. A cross-layer approach is required, as is an open communication protocol, where every information element is used optimally and redundant communication is prevented. In this sense, beaconing does not support a service like the hello-messages used by some routing protocols, it *is* the service. From the cooperative awareness created with received beacons, many applications can draw the information elements they require to operate. This makes it unnecessary to perform additional transmissions and interactions on the channel.

The performance of an application depends to a great extent on the quality of the information in the cooperative awareness. An important determining factor in this regard is the performance of the underlying wireless communication conveying this information, which is studied in great detail in this dissertation.

A traffic flow experiences congestion in response to an increase of the number of vehicles on the road. A similar phenomenon occurs in beaconing in a vehicular network: the system gradually becomes congested. The CACC application is intended to increase performance of a traffic flow, to keep traffic going under densities which would lead to congestion under manual longitudinal control. Under these dense traffic circumstances, it is important that the wireless communication system on which the CACC relies remains operational.

In Chapter 3 it was concluded that a CSMA/CA system like the IEEE 802.11p DCF or EDCA works optimally under low loads, and breaks down as the load increases beyond a certain point. In this sense, an increase in traffic density is one of the greatest threats to the performance of a beaconing system, warranting investigation of the system's scalability.

The proof-of-concept implementations discussed in Chapter 4 show that it is practically feasible to operate a CACC system based on beaconing with multiple vehicles, even if these vehicles differ in physical properties and the applied control architecture. These implementations also indicate that there are limitations outside the networking domain which can deteriorate CACC performance, the effect of which can be several orders of magnitude more prominent than network-induced performance limitations. The large delay introduced by the sampling of the sensors in C&D, and the large delay in the gearbox actuator in the GCDC vehicle, impede application-level performance. These limitations can be overcome in future implementations by applying different sensors and actuators, by applying synchronisation between components, or by other improvements to optimise the entire chain of subsystems. Despite the sampling delay, the overal latency in the C&D system was sufficiently small to outperform driving in ACC or manual mode.

In addition to these performance limits which are internal to a vehicle, the focus of the remainder of the dissertation is on the limits external to a vehicle: those present in the networking. It was found in Sec. 4.5 that the CACC is able to deal with a reduced influx of new information and can degrade gracefully because of its relative loss tolerance. CACC can still outperform a human driver in longitudinal vehicle control, even under moderately impaired wireless communication.

Most analytical models of IEEE 802.11 available in literature focus on a WLAN situation, where communication is performed in Unicast interactions. In Chapter 5 we provided detailed analytical models of the IEEE 802.11 DCF and EDCA for the evaluation of beaconing. The probability of succesful communication and service time of the MAC match well with simulation results. Additionally, the saturation point of the channel can accurately be predicted, which is an important input in the development of congestion control mechanisms. These models allow quick numerical evaluation and can in turn be embedded in modelling approaches which focus on, for instance, micro- or macroscopic traffic flow effects.

The DCF and EDCA were compared using the models developed in this dissertation. The large AIFS used in the EDCA causes more channel resources to be wasted as guard space, accounting for a significantly larger number of nodes to be achieved for similar probability of succesful communication under the DCF. Another weak point of the EDCA is the lack of streak behaviour present in the DCF due to the way the backoff counter is decremented. This causes a less efficient use of availabe resources than the DCF. The streak behaviour is able to synchronise collisions in the first slot of a streak, thereby *absorbing* part of the load and lowering collision probability for transmissions in subsequent slots. Furthermore, in contrast to the EDCA, the fraction of beacons which are received when using the DCF has a favourable delay characteristic because of a collision bias applied by the streak behaviour.

Various parameters influence the communication performance. The EIFS, for instance, is unnecessary on a channel with Broadcast-only traffic and forms a structural limitation to scalability. Increasing the size of the contention window was found to have little positive influence because of the nature of the beaconing channel. A larger contention window appears to have a beneficial effect on the success probability at first glance, but this is because many frames are dropped at the queue and those beacons that do arrive experience a very long delay.

For the beaconing communication pattern, a small transmission queue was found to be optimal because of its implicit congestion controlling effect. Whenever a frame needs to be dropped from the queue, the oldest one should be dropped, according to the Oldest Packet Drop policy described in this dissertation. This provides improved load scalability of the beaconing system.

The simulation study on the impact of channel switching on beaconing showed a performance drop of over 50%, rendering channel switching a severe threat to structural scalability. This leads to the conclusion that a node should continuously observe the channel used for the exchange of beacons.

Throughout this disseration, various networking aspects were discussed which impact application-level performance. In the end, it is the application which determines to a great extent whether the networking should be optimised for responsiveness, stability, cost, etc. This work provides the tools to make these trade-offs quantifiable.

7.2 Recommendations

Based on the findings presented in this dissertation, we can make the following recommendations to increase the scalability of beaconing for cooperative adaptive cruise control.

- Sampling delay accounts for a large part of the delay budget in the C&D system, as became evident in Sec. 4.2.4. This can be alleviated by synchronisation of the sampling and generation of beacons inside a vehicle, sampling at a higher rate, or by providing a good estimator. The contents of the beacon should be as fresh and accurate as possible upon generation. This is particularly important for highly dynamic metrics such as velocity and acceleration, because these are used as inputs in the CACC control loop. In the design phase, it is important to regard all the systems involved in the chain, and not solely focus on a single component.

- The use of complex application interactions as discussed in Sec. 4.3.1 is detrimental to beaconing performance. An example of such interactions could be where success of a certain beacon depends on correct reception of previous or subsequent beacons. It is better to exploit the messaging-nature of beaconing, by making every beacon an atomic entity. Additionally, these interactions should not be self-expanding, i.e. scheduling resources in a manner that increases its own usage. This reduces load scalability.

- The DCF was found in Sec. 5.6.1 to exhibit superior load scalability compared to the EDCA when used for beaconing in a single AC. When used for broadcast transmission of beacon messages, the DCF exhibits typical behaviour referred to in this dissertation as streaks. This causes several succesful transmissions to follow an initial collision in which many nodes are involved. This behaviour is absent in the EDCA because of the way backoff counter decrementing is handled. Under increased load, the EDCA succumbs to Slotted ALOHA-like behaviour. Based on these findings, use of the DCF for beaconing needs to be considered because of its greater success probability and significantly better delay characteristics.

- An observation relating to both DCF and EDCA is that use of the EIFS is unnecessary in a Broadcast-only environment, as studied in Sec. 5.6.2. In this case, the EIFS unnecessarily consumes channel resources. Disabling the use of EIFS on the CCH provides a performance benefit in both probability of succesful communication and delay and allows better medium utilisation, increasing the load scalability of the beaconing system.

- In Sec. 6.3, the use of the Oldest Packet Drop buffering mechanism for beaconing was found to perform better than the Newest Packet Drop policy often used in generic networks. OPD significantly reduces the queueing delay accumulated by a beacon by up to 30%, especially under increasing number of nodes. This holds in particular when using a dedicated queue of limited size, which was found to have an implicit congestion-reducing effect.

- Finally, the IEEE 1609.4 channel switching was shown in Sec. 6.4 to be detrimental to the structural scalability of a beaconing system, reducing success probability by up to 75%, and raising end-to-end delay by a factor of four. Furthermore, channel utilisation is reduced to 0.36 compared to 0.8 under continuous access. Research in the use and adoption of multi-radio or wide-band receiver solutions is strongly encouraged.

7.3 Future work

This dissertation explored the performance of wireless communication in the context of cooperative adaptive cruise control. Various directions of further research can be identified:

- It would be interesting to further investigate the impact of networking aspects on CACC, beyond the simulation study in Sec. 4.5. Directions could be the impact of correlated (burst) losses, other types of CACC controller topologies and their sensitivity to loss and delay, which overshoot/undershoot thresholds are acceptable for different types of traffic scenarios, and which exact network configuration is optimal under which circumstances. This could result in a set of rules to judge when CACC operation is possible or when the system should fall back to ACC behaviour. In addition, it was identified that there may be opportunity to tune the CACC based on the properties of the networking context, this could be further explored.

- The ultimate goal is development of a CACC system which is able to operate under any traffic regime it may encounter during operation. Exploring the impact of CACC including realistic communication on the traffic flow is the next logical step. Such research can be performed using the bi-directionally coupled SUMO and OMNeT++ discrete-event simulators, or analytically using the models presented in Chapter 5. These analytical models can be embedded in macro- or microscopic traffic models or simulated application behaviour, and provide a less computationally intensive alternative to discrete-event simulation. In order to evaluate this, an adequately provisioned analysis approach is required, the analytical models presented in this dissertation provide suitable building blocks.

- This work has shown that there is merit in using an OPD buffering mechanism as opposed to an NPD approach. Furthermore, it was suggested there may be benefit in growing and shrinking the queue size or the dropping probability in response to the channel state. Also, pre-emptive scheduling may be an interesting optimisation, where the beacon currently in contention is replaced if a newer one is generated in the mean time. Consequences of pre-emptive scheduling need to be investigated.

- So far, proof-of-concept implementations were only performed on very limited scale with only a few vehicles. It is essential to evaluate if a beaconing system can still operate under the networking regime which coincides with the traffic regime for which the CACC is designed to be most valuable: very dense traffic. The studies performed in this dissertation provide a good outlook as well as the opportunity for gains to be made, however a large-scale test involving hundreds of vehicles will be the ultimate test. The first steps are being taken today, in the SimTD project [207] in Germany and the Score@F project [208] in France.

Appendices

APPENDIX A

The FUTURUM Local Interaction Protocol

The ControlBox and the ComBox described in Sec. 4.3 communicate through Ethernet using a simple Header-Length-Value protocol dubbed “FLIP”. FLIP messages are simple UDP broadcast packets, transmitted on the in-vehicle Ethernet LAN. FLIP’s purpose is to communicate status and command messages and function as a

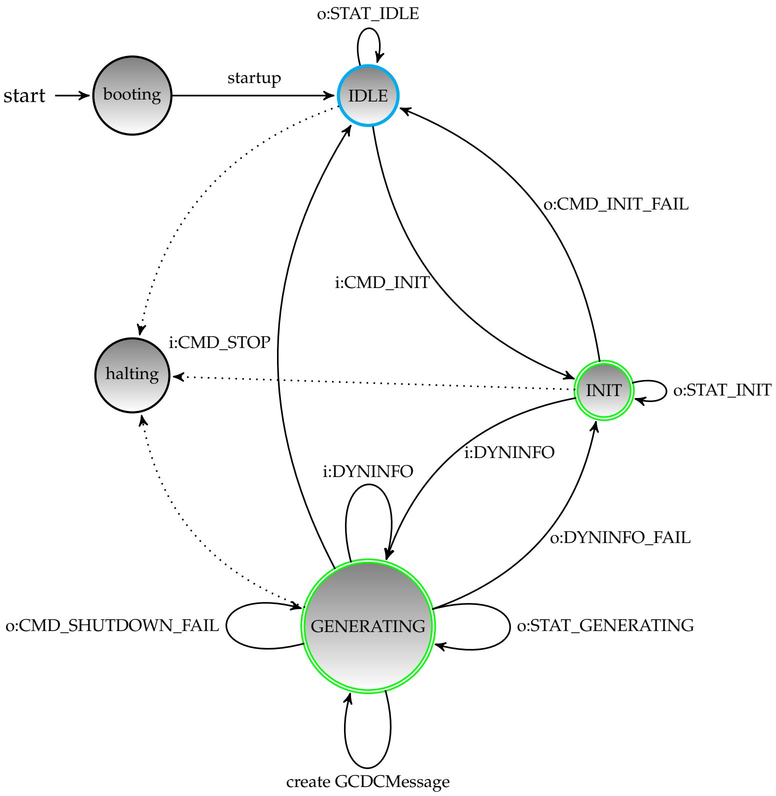


Figure A.1: States of the FUTURUM ComBox daemon, dotted transitions have label i:CMD_SHUTDOWN o:CMD_SHUTDOWN_OK.

container protocol for messages defined by the GCDC Interaction Protocol.

Fig. A.1 shows the states the ComBox can be in, and transitions between them. The states are defined based upon the ComBox's transmission function, the reception function is implicity present in states INIT and GENERATING.

IDLE Upon startup, the daemon is in state IDLE. It periodically sends STAT_IDLE FLIP messages to the ControlBox.

INIT Upon reception of a CMD_INIT message with a valid StaticVehicleInfo element in the payload, control moves to the INIT state. If the CMD_INIT is invalid (for instance, sent without StaticVehicleInfo) it will respond with a CMD_INIT_FAIL and remain in IDLE. In the INIT state, it will periodically send a STAT_INIT FLIP message. When in state INIT, it will forward received GCDC messages to the ControlBox. It is now passive and only listens to messages sent by others.

GENERATING As soon as the ControlBox starts reading its sensors and provides DYNINFO FLIP messages (containing DynamicVehicleInfo elements) the ComBox will transition to the GENERATING state. It will start to generate the GCDCMessages as defined by the GCDC interaction protocol. It will also forward received GCDC messages to the ControlBox, after filtering if needed. While in GENERATING, the ComBox wil periodically send STAT_GENERATING messages. It is now actively participating in the communication. Upon reception of the CMD_STOP command, the daemon will go back to IDLE, and flush its local DataStorage. A sequence CMD_STOP, CMD_INIT resets the system. The system transitions to INIT in case DYNINFO messages are no longer received: the generator does no longer have the required data to produce valid messages, switching back to passive mode. If the daemon receives the CMD_SHUTDOWN it will stop receiving and transmitting messages. It will finish writing to log files and will reply with a CMD_SHUTDOWN_OK message, terminate the daemon and halt the system.

Whenever the ComBox is in INIT or in GENERATING, it will receive messages transmitted by other nodes. As such, reception behaviour does not differ. While in INIT, the ComBox is passive (i.e. only receiving) and in GENERATING it is active (i.e. it receives and transmits messages).

The ASN.1 definition of FLIP is as follows:

```
FLIPMessage ::= SEQUENCE {
     msgType                  MsgType,     --byte denoting type
     msgLength        INTEGER(0..MTU),     --16bit unsigned integer, length of
          payload in bytes
     payload              FLIPpayload
}
msgType | msgLength | payload

 //msgType is to be encoded as a byte
 //some messages can carry payload, if not msgLenght=0
```

```
MsgType ::= OCTET {
    RESERVED              (000|xxxxx),     --Not used
    RESERVED              (101|xxxxx),     --Not used
    RESERVED              (110|xxxxx),     --Not used
    RESERVED              (111|xxxxx),     --Not used
    CMD class             (001|xxxxx),     --(1) Command messages
      GCDCMESSAGE              |00001),    -- (1) Contains GCDCMessage
      DYNINFO                  |00010),    -- (2) Contains DynamicVehicleInfo,
           start beacon cycle
      CMD_INIT                 |00011),    -- (3) Command to Init, contains
           StaticVehicleInfo
      CMD_STOP                 |00101),    -- (5) Command to Stop generating
           messages
      CMD_SHUTDOWN             |00111),    -- (7) Command to shutdown the
           ComBox
      CMD_SPEED_INTENT         |01000),    -- (8) Command to send
           VehicleSpeedIntent
    REP class             (010|xxxxx),     --(2) Reply messages
      CMD_SHUTDOWN_OK          |00111),    -- (7) Shutdown ok
      CMD_SPEED_INTENT_OK      |01000),    -- (8) ComBox will send
           VehicleSpeedIntent
    STAT class            (011|xxxxx),     --(3) State information
      STAT_IDLE                |00000),    -- (0) ComBox is IDLE
      STAT_GENERATING          |00010),    -- (2) ComBox is GENERATING
      STAT_INIT                |00011),    -- (3) ComBox is in INIT, ready to
           start GENERATING
    ERR class             (100|xxxxx),     --(4) Error messages
      DYNINFO_FAIL             |00010),    -- (2) ComBox trying to generate,
           but DYNINFO too old
      CMD_INIT_FAIL            |00011),    -- (3) Combox cannot properly init
      CMD_STOP_FAIL            |00101),    -- (5) ComBox cannot properly stop
      CMD_SHUTDOWN_FAIL        |00111),    -- (7) ComBox cannot properly
           shutdown
      CMD_SPEED_INTENT_FAIL    |01000),    -- (8) ComBox cannot send
           VehicleSpeedIntent
      STAT_ERROR               |11111)     -- (32) ComBox is in an error state
}

 //payload
FLIPpayload ::= CHOICE {
      GCDCMessage,
      DynamicVehicleInfo,
      StaticVehicleInfo,
      VehicleSpeedIntent
}
 //FLIPpayload elements are directly reused from GCDC interation protocol
      ASN.1 specification.
```

Listing A.1: FLIP

APPENDIX B

Steady state distribution of the DCF

This appendix derives the steady-state equations given in Sec. 5.3 for the DCF MAC.

B.1 Post backoff towards idle

We start in the upper-right corner of the chain in Fig. 5.2 by finding the balance equation for $s_{0,W-1}$, and then work to the left. Finaly, the expression for $b_{0,1}$ is used to find $b_{0,0}$.

Finding a generic expression for $b_{0,k}$.

$$\begin{aligned}
b_{0,W-1} &= \frac{1-\rho}{W(1-p^\star)} b_{1,0}(1-q^\star)^0 \\
b_{0,W-2} &= \frac{1-\rho}{W(1-p^\star)} b_{1,0}\left((1-q^\star)^0 + (1-q^\star)^1\right) \\
b_{0,W-3} &= \frac{1-\rho}{W(1-p^\star)} b_{1,0}\left((1-q^\star)^0 + (1-q^\star)^1 + (1-q^\star)^2\right) \\
b_{0,W-4} &= \frac{1-\rho}{W(1-p^\star)} b_{1,0}\left((1-q^\star)^0 + (1-q^\star)^1 + (1-q^\star)^2 + (1-q^\star)^3\right)
\end{aligned} \tag{B.1}$$

Results in:

$$\begin{aligned}
b_{0,k} &= \frac{1-\rho}{W(1-p^\star)} b_{1,0} \sum_{i=0}^{W-k-1} (1-q^\star)^i \text{ for } k = 1, \ldots, W-1 \\
&= \frac{1-\rho}{W(1-p^\star)} b_{1,0} \frac{1-(1-q^\star)^{W-k}}{1-(1-q^\star)} \\
&= \frac{1-\rho}{W(1-p^\star)} b_{1,0} \frac{1-(1-q^\star)^{W-k}}{q^\star}
\end{aligned} \tag{B.2}$$

□

The summation is a standard summation (geometric series starting at 1) so we can write it as a fraction:

$$\sum_{i=0}^{n-1} x^i = \frac{1-x^n}{1-x} \tag{B.3}$$

In case of $k = 1, \ldots$,W-1, the $\frac{1}{(1-p^\star)}$ and $(1-p^\star)$ do not cancel eachother, like they do for $b_{0,0}$. For $b_{0,0}$ the recursion traverses the full chain, for $b_{0,k}$ the recursion stops after $W-k$ steps.

Finding an expression for $b_{0,0}$. From the chain in Fig. 5.2 we find the balanced equation for $s_{0,0}$:

$$\begin{aligned}
b_{0,0} &= \left(\frac{1-\rho}{W} b_{1,0} + (1-q^\star)(1-p^\star) b_{0,1}\right)\frac{1}{q} \\
&= \left(\frac{1-\rho}{W} b_{1,0} + (1-q^\star)\cancel{(1-p^\star)}\left[\frac{1-\rho}{W\cancel{(1-p^\star)}} b_{1,0} \sum_{i=0}^{W-k-1}(1-q^\star)^i\right]\right)\frac{1}{q} \\
&= \frac{1-\rho}{Wq} b_{1,0} + \frac{1-\rho}{Wq} b_{1,0}(1-q^\star)\sum_{i=0}^{W-k-1}(1-q^\star)^i \\
&= \frac{1-\rho}{Wq} b_{1,0} + \frac{1-\rho}{Wq} b_{1,0}(1-q^\star)\frac{1-(1-q^\star)^{W-k}}{q^\star} \\
&= \frac{1-\rho}{Wq} b_{1,0}\left(1 + \frac{(1-q^\star)-(1-q^\star)^W}{q^\star}\right) \\
&= \frac{1-\rho}{Wq} b_{1,0}\left(\frac{\cancel{q^\star} + 1 \cancel{-q^\star} - (1-q^\star)^W}{q^\star}\right) \\
&= \frac{1-\rho}{Wq} b_{1,0}\frac{1-(1-q^\star)^W}{q^\star}
\end{aligned} \tag{B.4}$$

We substitute $b_{0,1}$ found earlier and substitute $1 = \frac{q^\star}{q^\star}$ in order for the $q^\star$ and the $-q^\star$ to cancel eachother.

□

B.2 Backoff towards transmission attempt

The state $s_{1,W-1}$ is a special state in that there is no transition into it from the first row:

$$\begin{aligned}
b_{1,W-1} &= \frac{b_{1,0}\frac{\rho}{W} + b_{0,0}\frac{qp}{W}}{1-p^\star} \\
&= \frac{1}{W(1-p^\star)}(\rho b_{1,0} + qp b_{0,0})
\end{aligned} \tag{B.5}$$

This contains getting into $s_{1,W-1}$ after a transmission attempt (the queue was nonempty and we immediately content for the next frame) and the arrival of a new frame in the queue while the node is idle ($s_{0,0}$). The transition out of $s_{1,W-1}$ occurs with probability $(1-p^\star)$. We can now find an expression for $b_{1,k}$ for $k = 1 \ldots W-1$:

Finding an expression for $b_{1,k}$.

$$\begin{aligned}
b_{1,W-1} &= \left(b_{1,0}\frac{\rho}{W} + b_{0,0}\frac{qp}{W}\right)\frac{1}{1-p^\star} \\
&= \left(b_{1,0}\frac{\rho}{W} + \left[\frac{1-\rho}{Wq}b_{1,0}\frac{1-(1-q^\star)^W}{q^\star}\right]\frac{qp}{W}\right)\frac{1}{1-p^\star} \\
&= \frac{b_{1,0}}{W(1-p^\star)}\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right) \qquad \text{(B.6)}
\end{aligned}$$

$$\begin{aligned}
b_{1,W-2} &= \left(b_{1,0}\frac{\rho}{W} + b_{0,0}\frac{qp}{W} + b_{1,W-1}(1-p^\star) + b_{0,W-1}q^\star(1-p^\star)\right)\frac{1}{(1-p^\star)} \\
&= b_{1,0}\frac{\rho}{W(1-p^\star)} + b_{0,0}\frac{qp}{W(1-p^\star)} + b_{1,W-1} + b_{0,W-1}q^\star \\
&= b_{1,0}\frac{\rho}{W(1-p^\star)} + \left[\frac{1-\rho}{Wq}b_{1,0}\frac{1-(1-q^\star)^W}{q^\star}\right]\frac{qp}{W(1-p^\star)} + \\
&\quad \left[\left(b_{1,0}\frac{\rho}{W} + b_{0,0}\frac{qp}{W}\right)\frac{1}{(1-p^\star)}\right] + \left[\frac{1-\rho}{W(1-p^\star)}b_{1,0}\right]q^\star \\
&= b_{1,0}\frac{1}{W(1-p^\star)}\left(\rho + \cancel{q}p \cdot \frac{1-\rho}{W\cancel{q}}\frac{1-(1-q^\star)^W}{q^\star} + \right. \\
&\quad \left. \rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star} + q^\star(1-\rho)\right) \\
&= \frac{b_{1,0}}{W(1-p^\star)}\left(2\rho + \frac{2p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star} + q^\star(1-\rho)\right) \qquad \text{(B.7)}
\end{aligned}$$

$$
\begin{aligned}
b_{1,W-3} &= \left(b_{1,0}\frac{\rho}{W} + b_{0,0}\frac{qp}{W} + b_{1,W-2}(1-p^\star) + b_{0,W-2}q^\star(1-p^\star)\right)\frac{1}{(1-p^\star)} \\
&= b_{1,0}\frac{\rho}{W(1-p^\star)} + b_{0,0}\frac{qp}{W(1-p^\star)} + b_{1,W-2} + b_{0,W-2}q^\star \\
&= b_{1,0}\frac{\rho}{W(1-p^\star)} + \left[\frac{1-\rho}{Wq}b_{1,0}\frac{1-(1-q^\star)^W}{q^\star}\right]\frac{qp}{W(1-p^\star)} + \\
&\quad \left[\frac{b_{1,0}}{W(1-p^\star)}\left(2\rho + \frac{2p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star} + q^\star(1-\rho)\right)\right] + \\
&\quad \left[\frac{1-\rho}{W(1-p^\star)}b_{1,0}\frac{1-(1-q^\star)^{W-(W-2)}}{\cancel{q^\star}}\right]\cancel{q^\star} \\
&= \frac{b_{1,0}}{W(1-p^\star)}\left(3\rho + \frac{3p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star} + \right. \\
&\quad \left. q^\star(1-\rho) + (1-\rho)(1-(1-q^\star)^{W-(W-2)})\right) \qquad \text{(B.8)}
\end{aligned}
$$

Here we note linear increase of the first two terms with $(W-k)$ and a summation of the last terms which are multiplied with $(1-\rho)$. Also note the $\left((1-(1-q^\star)^{W-(W-1)}\right)$ evaluates to $q^\star$ for $b_{1,W-2}$. For $b_{1,W-1}$ this term is zero (i.e. $1-(1-q^\star)^0 = 1-1$). The result is a summation over W-k-1 multiplied by $(1-\rho)$. A generic expression can be found by rewriting the summation (it is a geometric series as seen earlier):

$$
\begin{aligned}
b_{1,k} &= \frac{b_{1,0}}{W(1-p^\star)}\left((W-k)\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right) + \right. \\
&\quad \left.(1-\rho)\sum_{i=0}^{W-k-1} 1-(1-q^\star)^{W-(W-i)}\right) \\
&= \frac{b_{1,0}}{W(1-p^\star)}\left((W-k)\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right) + \right. \\
&\quad \left.(1-\rho)\left((W-k) - \sum_{i=0}^{W-k-1}(1-q^\star)^i\right)\right) \\
&= \frac{b_{1,0}}{W(1-p^\star)}\left((W-k)\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right) + \right. \\
&\quad \left.(1-\rho)\left((W-k) - \frac{1-(1-q^\star)^{W-k}}{q^\star}\right)\right) \qquad \text{(B.9)}
\end{aligned}
$$

□

B.3 Normalisation

We now know $b_{0,0}, \cdots, b_{0,W-1}$ and $b_{1,1}, \cdots, b_{1,W-1}$. By normalisation we can find an expression for $b_{1,0}$, and express all states in $b_{1,0}$.

Finding an expression for $b_{1,0}$.

$$\begin{aligned}1 &= b_{0,0} + b_{0,k} + b_{1,0} + b_{1,k}, \forall k \in \{1, \ldots, W-1\}\\ &= b_{0,0} + \sum_{k=1}^{W-1} b_{0,k} + b_{1,0} + \sum_{k=1}^{W-1} b_{1,k}\\ &= \left[\frac{1-\rho}{Wq} b_{1,0} \frac{1-(1-q^\star)^W}{q^\star}\right] + \sum_{k=1}^{W-1}\left[\frac{1-\rho}{W(1-p^\star)} b_{1,0} \frac{1-(1-q^\star)^{W-k}}{q^\star}\right] + b_{1,0} +\\ &\sum_{k=1}^{W-1}\left[\frac{b_{1,0}}{W(1-p^\star)}\left((W-k)\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right) +\right.\right.\\ &\left.\left.(1-\rho)\left((W-k) - \frac{1-(1-q^\star)^{W-k}}{q^\star}\right)\right)\right]\end{aligned} \tag{B.10}$$

Isolating $\frac{1}{b_{(1,0)}}$:

$$\begin{aligned}\frac{1}{b_{1,0}} = &\left[\frac{1-\rho}{Wq}\frac{1-(1-q^\star)^W}{q^\star}\right] + \sum_{k=1}^{W-1}\left[\frac{1-\rho}{W(1-p^\star)}\frac{1-(1-q^\star)^{W-k}}{q^\star}\right] + 1 +\\ &\sum_{k=1}^{W-1}\left[\frac{1}{W(1-p^\star)}\left((W-k)\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right) +\right.\right.\\ &\left.\left.(1-\rho)\left((W-k) - \frac{1-(1-q^\star)^{W-k}}{q^\star}\right)\right)\right]\end{aligned} \tag{B.11}$$

We need to get rid of the summations, so we rewrite the second term:

$$\begin{aligned}b'_{0,k} &= \sum_{k=1}^{W-1}\left[\frac{1-\rho}{W(1-p^\star)}\frac{1-(1-q^\star)^{W-k}}{q^\star}\right]\\ &= \frac{1-\rho}{W(1-p^\star)}\sum_{k=1}^{W-1}\frac{1-(1-q^\star)^{W-k}}{q^\star}\\ &= \frac{1-\rho}{q^\star W(1-p^\star)}\left(W - \sum_{k=1}^{W-1}(1-q^\star)^k\right)\end{aligned} \tag{B.12}$$

Note how the result resembles the standard summation for the geometeric series. Just need to start at $k=0$, which is no problem because $1-(1-q^\star)^0 = 0$, summing a zero.

$$
\begin{aligned}
b'_{0,k} &= \frac{1-\rho}{q^{\star}W(1-p^{\star})}\left(W - \sum_{k=0}^{W-1}(1-q^{\star})^{k}\right) \\
&= \frac{1-\rho}{q^{\star}W(1-p^{\star})}\left(W - \frac{1-(1-q^{\star})^{W}}{q^{\star}}\right) \\
&= \frac{(1-\rho)}{q^{\star}(1-p^{\star})} - \frac{(1-\rho)}{q^{\star}(1-p^{\star})}\frac{1-(1-q^{\star})^{W}}{Wq^{\star}}
\end{aligned}
\tag{B.13}
$$

Next, rewrite the fourth term:

$$
\begin{aligned}
b'_{1,k} = \quad & \sum_{k=1}^{W-1}\left[\frac{1}{W(1-p^{\star})}\left((W-k)\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^{\star})^{W}}{q^{\star}}\right) + \right.\right. \\
& \left.\left. (1-\rho)\left((W-k) - \frac{1-(1-q^{\star})^{W-k}}{q^{\star}}\right)\right)\right] \\
= \quad & \frac{1}{W(1-p^{\star})}\left(\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^{\star})^{W}}{q^{\star}}\right)\sum_{k=1}^{W-1}(W-k) + \right. \\
& \left. (1-\rho)\left(\sum_{k=1}^{W-1}(W-k) - \sum_{k=1}^{W-1}\frac{1-(1-q^{\star})^{W-k}}{q^{\star}}\right)\right) \\
= \quad & \frac{1}{W(1-p^{\star})}\left(\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^{\star})^{W}}{q^{\star}}\right)\sum_{k=1}^{W-1}(W-k) + \right. \\
& \left. (1-\rho)\left(\sum_{k=1}^{W-1}(W-k) - \frac{1}{q^{\star}}\left(W - \sum_{k=1}^{W-1}(1-q^{\star})^{k}\right)\right)\right)
\end{aligned}
\tag{B.14}
$$

We take everything not related to the summations outside the summations. Because $\sum_{k=1}^{W-1}(W-k) = \sum_{k=1}^{W-1}k$ (change order of summation) we arrive at an arithmetic series $\sum_{k=1}^{N}k = \frac{N(N+1)}{2}$ which we can write as $\frac{(W-1)W}{2}$ instead. Also, we perform the by now familiar conversion of the geometric series:

$$\begin{aligned}
b'_{1,k} = {} & \frac{1}{W(1-p^\star)}\left(\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right)\frac{(W-1)W}{2} + \right. \\
& \left. (1-\rho)\left(\frac{(W-1)W}{2} - \frac{1}{q^\star}\left(W - \frac{1-(1-q^\star)^W}{q^\star}\right)\right)\right) \\
= {} & \frac{1}{W(1-p^\star)}\left(\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right)\frac{(W-1)W}{2} + \right. \\
& \left. \frac{(1-\rho)(W-1)W}{2} - \frac{(1-\rho)W}{q^\star} + \frac{(1-\rho)}{q^\star}\frac{1-(1-q^\star)^W}{q^\star}\right)
\end{aligned} \tag{B.15}$$

Now we can put it all together, and proceed with simplifying:

$$\begin{aligned}
\frac{1}{b_{1,0}} = {} & \frac{1-\rho}{Wq}\frac{1-(1-q^\star)^W}{q^\star} + \frac{1-\rho}{q^\star W(1-p^\star)}\left(W - \frac{1-(1-q^\star)^W}{q^\star}\right) + 1 + \\
& \frac{1}{W(1-p^\star)}\left(\left(\rho + \frac{p(1-\rho)}{W}\frac{1-(1-q^\star)^W}{q^\star}\right)\frac{(W-1)W}{2} + \right. \\
& \left. \frac{(1-\rho)(W-1)W}{2} - \frac{W(1-\rho)}{q^\star} + \frac{(1-\rho)}{q^\star}\frac{1-(1-q^\star)^W}{q^\star}\right) \\
= {} & \frac{1-\rho}{Wq}\frac{1-(1-q^\star)^W}{q^\star} + \frac{W(1-\rho)}{q^\star W(1-p^\star)} - \frac{(1-\rho)}{q^\star W(1-p^\star)}\frac{1-(1-q^\star)^W}{q^\star} + 1 + \\
& \frac{\rho(W-1)W}{2W(1-p^\star)} + \frac{p(1-\rho)(W-1)W}{2W}\frac{1}{W(1-p^\star)}\frac{1-(1-q^\star)^W}{q^\star} + \\
& \frac{(1-\rho)(W-1)W}{2W(1-p^\star)} - \frac{W(1-\rho)}{q^\star W(1-p^\star)} + \frac{(1-\rho)}{q^\star(1-p^\star)}\frac{1-(1-q^\star)^W}{Wq^\star} \\
= {} & \frac{1-\rho}{q}\frac{1-(1-q^\star)^W}{Wq^\star} + \frac{(1-\rho)}{q^\star(1-p^\star)} - \frac{(1-\rho)}{q^\star(1-p^\star)}\frac{1-(1-q^\star)^W}{Wq^\star} + 1 + \\
& \frac{\rho(W-1)}{2(1-p^\star)} + \frac{p(1-\rho)(W-1)}{2(1-p^\star)}\frac{1-(1-q^\star)^W}{Wq^\star} + \\
& \frac{(1-\rho)(W-1)}{2(1-p^\star)} - \frac{(1-\rho)}{q^\star(1-p^\star)} + \frac{(1-\rho)}{Wq^\star(1-p^\star)}\frac{1-(1-q^\star)^W}{q^\star} \\
= {} & \frac{1-\rho}{q}\frac{1-(1-q^\star)^W}{Wq^\star} + \frac{p(1-\rho)(W-1)}{2(1-p^\star)}\frac{1-(1-q^\star)^W}{Wq^\star} + 1 + \\
& \frac{(1-\rho)}{q^\star(1-p^\star)}\frac{1-(1-q^\star)^W}{Wq^\star} - \frac{(1-\rho)}{q^\star(1-p^\star)}\frac{1-(1-q^\star)^W}{Wq^\star} \\
& \frac{\rho(W-1)}{2(1-p^\star)} + \frac{(1-\rho)(W-1)}{2(1-p^\star)} + \frac{(1-\rho)}{q^\star(1-p^\star)} - \frac{(1-\rho)}{q^\star(1-p^\star)}
\end{aligned} \tag{B.16}$$

After shuffling, the sixth and seventh term cancel eachother. Also, the fourth and fifth and sixth can be merged because $\rho + (1-\rho) = 1$. The result:

$$\frac{1}{b_{1,0}} = \frac{1-\rho}{q}\frac{1-(1-q^\star)^W}{Wq^\star} + 1 + \frac{(W-1)}{2(1-p^\star)} + \frac{p(1-\rho)(W-1)}{2(1-p^\star)}\frac{1-(1-q^\star)^W}{Wq^\star} \tag{B.17}$$

Rewriting in the same manner as done in [156] yields:

$$\frac{1}{b_{1,0}} = 1 + \frac{(W-1)}{2(1-p^\star)} + \frac{1-\rho}{q}\frac{1-(1-q^\star)^W}{Wq^\star}\left(1 + \frac{(W-1)qp}{2(1-p^\star)}\right) \tag{B.18}$$

□

APPENDIX C

Steady state distribution of the EDCA

This appendix derives the steady-state equations given in Sec. 5.4 for the EDCA MAC. The derivation goes along the same lines as that presented in Appendix B.

C.1 Post backoff towards idle

We start in the upper-right corner of the chain in Fig. 5.6 by finding the balance equation for $s_{0,W-2}$, and then work to the left.

Finding a generic expression for $b_{0,k}$*,* $k \in \{1, \ldots, W-2\}$*.*

$$\begin{aligned}
b_{0,W-2} &= \frac{1-\rho}{W} b_{1,-1} \cdot \text{“}(1-q^\star)^0\text{”} \\
b_{0,W-3} &= \frac{1-\rho}{W} b_{1,-1} \left((1-q^\star)^0 + (1-q^\star)^1\right) \\
&\vdots \\
b_{0,W-15} &= \frac{1-\rho}{W} b_{1,-1} \Big((1-q^\star)^0 + (1-q^\star)^1 + \cdots + (1-q^\star)^{W-k-2)})\Big) \qquad \text{(C.1)}
\end{aligned}$$

Results in:

$$\begin{aligned}
b_{0,k} &= \frac{1-\rho}{W} b_{1,-1} \sum_{i=0}^{W-k-2} (1-q^\star)^i \text{ for } k = 1, \ldots, W-2 \\
&= \frac{1-\rho}{W} b_{1,-1} \frac{1-(1-q^\star)^{W-k-1}}{q^\star} \qquad \text{(C.2)}
\end{aligned}$$

□

Finding an expression for $b_{0,0}$. From the chain in Fig. 5.6 we find the balanced equation for $s_{0,0}$:

$$\begin{aligned}
b_{0,0} &= \left(\frac{2(1-\rho)}{W} b_{1,-1} + (1-q^\star) b_{0,1}\right) \frac{1}{q_b p + q_e(1-p)} \\
&= \left(\frac{2(1-\rho)}{W} b_{1,-1} + (1-q^\star)\left[\frac{1-\rho}{W} b_{1,-1} \frac{1-(1-q^\star)^{W-2}}{q^\star}\right]\right) \frac{1}{q_b p + q_e(1-p)} \qquad \text{(C.3)} \\
&= \frac{2(1-\rho)}{W(q_b p + q_e(1-p))} b_{1,-1} + \frac{1-\rho}{W(q_b p + q_e(1-p))} b_{1,-1}(1-q^\star)\frac{1-(1-q^\star)^{W-2}}{q^\star}
\end{aligned}$$

Multiplication with $(1-q^\star)$ adds a 1, resulting in $W-1$:

$$\begin{aligned}
&= \frac{(1-\rho)}{W(q_b p + q_e(1-p))} b_{1,-1}\left(2 + \frac{(1-q^\star) - (1-q^\star)^{W-1}}{q^\star}\right) \\
&= \frac{(1-\rho)}{W(q_b p + q_e(1-p))} b_{1,-1}\left(\frac{q^\star + \cancel{q^\star} + (1\cancel{-q^\star}) - (1-q^\star)^{W-1}}{q^\star}\right) \qquad \text{(C.4)} \\
&= \frac{(1-\rho)}{W(q_b p + q_e(1-p))} b_{1,-1}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right) \qquad \text{(C.5)}
\end{aligned}$$

We substitute $b_{0,1}$ found earlier in Eq. (C.3) and substitute $2 = \frac{q^\star + q^\star}{q^\star}$ in order for the $q^\star$ and the $1-q^\star$ to cancel eachother in Eq. (C.4). The remaining $\frac{q^\star}{q^\star}$ then becomes the 1 in front of the fraction. □

C.2 Backoff towards transmission attempt

The state $s_{1,W-2}$ is a special state in that there is no transition into it from the first row. This contains getting into $s_{1,W-2}$ after a TX (the queue was nonempty and we immediately content for the next frame) and the arrival of a new frame in the queue while the node is IDLE ($s_{0,0}$). The transition out of $s_{1,W-2}$ occurs with probability 1. We can now find an expression for $b_{1,k}$ for $l = 0, \dots, W-2$:

Finding an expression for $b_{1,l}$.

$$\begin{aligned}
b_{1,W-2} &= b_{1,-1}\frac{\rho}{W} + b_{0,0}\frac{q_b p}{W} \\
&= b_{1,-1}\frac{\rho}{W} + \left[\frac{1-\rho}{W(q_b p + q_e(1-p))} b_{1,-1}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right]\frac{q_b p}{W} \\
&= \frac{b_{1,-1}}{W}\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right) \qquad \text{(C.6)}
\end{aligned}$$

Added to this expression is a "$+(1-\rho)(1-(1-q^\star)^0)$" which evaluates to 0 (the following will make clear why).

$$\begin{aligned}
b_{1,W-3} &= b_{1,-1}\frac{\rho}{W} + b_{0,0}\frac{q_b p}{W} + b_{1,W-2} + b_{0,W-2}q^\star \\
&= b_{1,-1}\frac{\rho}{W} + \left[\frac{1-\rho}{W(q_b p + q_e(1-p))} b_{1,-1}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right]\frac{q_b p}{W} + \\
&\quad \left[\frac{b_{1,-1}}{W}\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right)\right] + \\
&\quad \left[\frac{1-\rho}{W} b_{1,-1}\frac{1-(1-q^\star)^{W-(W-2)-1}}{\cancel{q^\star}}\right]\cancel{q^\star} \\
&= \frac{b_{1,-1}}{W}\left(2\rho + \frac{2q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right) + \right. \\
&\quad \left. (1-\rho)(1-(1-q^\star)^1)\right)
\end{aligned} \tag{C.7}$$

$$\begin{aligned}
b_{1,W-4} &= b_{1,-1}\frac{\rho}{W} + b_{0,0}\frac{qp}{W} + b_{1,W-3} + b_{0,W-3}q^\star \\
&= b_{1,-1}\frac{\rho}{W} + \left[\frac{1-\rho}{W(q_b p + q_e(1-p))} b_{1,-1}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right]\frac{q_b p}{W} + \\
&\quad \left[\frac{b_{1,-1}}{W}\left(2\rho + \frac{2q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right) + \right.\right. \\
&\quad \left.\left.(1-\rho)(1-(1-q^\star)^1)\right)\right] + \left[\frac{1-\rho}{W} b_{1,-1}\frac{1-(1-q^\star)^{W-(W-3)-1}}{\cancel{q^\star}}\right]\cancel{q^\star} \\
&= \frac{b_{1,-1}}{W}\left(3\rho + \frac{3q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right) + \right. \\
&\quad \left. (1-\rho)\left((1-(1-q^\star)^1) + (1-(1-q^\star)^2)\right)\right)
\end{aligned} \tag{C.8}$$

Here we note linear increase of the first two terms with $(W-l-1)$: for $l = 14$ or $W-2$, this yields $16-14-1 = 1$, summing only once for $b_{1,W-2}$. For $l = 0$, $16-0-1 = 15$. This makes sense, as there are 15 paths to get from TX or IDLE into $s_{1,0}$: one direct and 14 through the other BO states.

Also, there is a summation of $\left((1-(1-q^\star)^{W-(W-l)-1}\right)$, for $b_{1,W-2}$ this term is zero (i.e. $1-(1-q^\star)^0 = 1-1$). The result is a summation over $W-l-2$ multiplied by $(1-\rho)$. A generic expression can be found by rewriting the summation (it is a geometric series as seen earlier):

$$b_{1,l} = \frac{b_{1,-1}}{W}\left((W-l-1)\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right) + \right.$$
$$\left.(1-\rho)\sum_{i=0}^{W-l-2} 1-(1-q^\star)^{W-(W-i)}\right)$$

a summation $\sum_{i=0}^{N} c = (N+1)c$, so we can take c out of the summation

$$= \frac{b_{1,-1}}{W}\left((W-l-1)\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right) + \right.$$
$$\left.(1-\rho)\left((W-l-1) - \sum_{i=0}^{W-l-2}(1-q^\star)^i\right)\right)$$

and then rewrite the geometeric series:

$$= \frac{b_{1,-1}}{W}\left((W-l-1)\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right) + \right.$$
$$\left.(1-\rho)\left((W-l-1) - \frac{1-(1-q^\star)^{W-l-1}}{q^\star}\right)\right) \tag{C.9}$$

□

C.3 Normalisation

We now know $b_{0,0}, \cdots, b_{0,W-2}$ and $b_{1,0}, \cdots, b_{1,W-2}$. By normalisation we can find an expression for $b_{1,-1}$, and express all states in terms of $b_{1,-1}$.

Finding an expression for $b_{1,-1}$.

$$1 = b_{0,0} + b_{0,k} + b_{1,-1} + b_{1,l}, k \in \{1,\ldots,W-2\}, l \in \{0,\ldots,W-2\}$$
$$= b_{0,0} + \sum_{k=1}^{W-2} b_{0,k} + b_{1,-1} + \sum_{l=0}^{W-2} b_{1,l} \tag{C.10}$$

Isolating $\frac{1}{b_{1,-1}}$:

$$\begin{aligned}
\frac{1}{b_{1,-1}} = & \left[\frac{(1-\rho)}{W(q_b p + q_e(1-p))}\left(1+\frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right]+ \\
& \sum_{k=1}^{W-2}\left[\frac{1-\rho}{W}\frac{1-(1-q^\star)^{W-k-1}}{q^\star}\right]+1+ \\
& \sum_{l=0}^{W-2}\left[\frac{1}{W}\left((W-l-1)\left(\rho+\frac{q_b p(1-\rho)}{W(q_b p+q_e(1-p))}\left(1+\frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right)+\right.\right. \\
& \left.\left.(1-\rho)\left((W-l-1)-\frac{1-(1-q^\star)^{W-l-1}}{q^\star}\right)\right)\right] \qquad \text{(C.11)}
\end{aligned}$$

We need to get rid of the summations, so we rewrite the second term:

$$\begin{aligned}
b'_{0,k} &= \sum_{k=1}^{W-2}\left[\frac{1-\rho}{W}\frac{1-(1-q^\star)^{W-k-1}}{q^\star}\right] \\
&= \frac{1-\rho}{W}\sum_{k=1}^{W-2}\frac{1-(1-q^\star)^{W-k-1}}{q^\star} \\
&= \frac{1-\rho}{Wq^\star}\left((W-2)-\sum_{k=1}^{W-2}(1-q^\star)^k\right) \qquad \text{(C.12)}
\end{aligned}$$

Note how the result resembles the standard summation for the geometeric series, which starts at $k=0$. This is no problem because $1-(1-q^\star)^0=0$, summing a zero.

$$\begin{aligned}
b'_{0,k} &= \frac{1-\rho}{Wq^\star}\left((W-1)-\sum_{k=0}^{W-2}(1-q^\star)^k\right) \\
&= \frac{1-\rho}{Wq^\star}\left((W-1)-\frac{1-(1-q^\star)^{W-1}}{q^\star}\right) \\
&= \frac{(1-\rho)(W-1)}{Wq^\star}-\frac{(1-\rho)}{q^\star}\frac{1-(1-q^\star)^{W-1}}{Wq^\star} \qquad \text{(C.13)}
\end{aligned}$$

Next, we rewrite the fourth term:

$$
\begin{aligned}
b'_{1,l} = \sum_{l=0}^{W-2} \Bigg[\frac{1}{W} \Bigg(&(W-l-1) \left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))} \left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star} \right) \right) + \\
&(1-\rho) \left((W-l-1) - \frac{1-(1-q^\star)^{W-l-1}}{q^\star} \right) \Bigg) \Bigg]
\end{aligned}
$$

We take everything not related to the summations outside the summations:

$$
\begin{aligned}
&= \frac{1}{W} \Bigg(\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))} \left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star} \right) \right) \sum_{l=0}^{W-2} (W-l-1) + \\
&\qquad (1-\rho) \left(\sum_{l=0}^{W-2} (W-l-1) - \sum_{l=0}^{W-2} \frac{1-(1-q^\star)^{W-l-1}}{q^\star} \right) \Bigg) \\
&= \frac{1}{W} \Bigg(\left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))} \left(1 + \frac{1-(1-q^\star)^{W-1}}{q^\star} \right) \right) \sum_{l=0}^{W-2} (W-l-1) + \\
&\qquad (1-\rho) \left(\sum_{l=0}^{W-2} (W-l-1) - \frac{1}{q^\star} \left((W-1) - \sum_{l=0}^{W-2} (1-q^\star)^{l+1} \right) \right) \Bigg) \qquad \text{(C.14)}
\end{aligned}
$$

Because $\sum_{l=1}^{W-2}(W-l-1) = \sum_{l=1}^{W-2} l-1$ (change order of summation) we arrive at what resembles an arithmetic series $\sum_{l=1}^{N} l-1 = \frac{N(N+1)}{2}$ which we can write as $\frac{(W-2)(W-1)}{2}$ instead. $\sum_{l=1}^{W-1} l$, hieruit volgt een $\frac{W(W-1)}{2}$.

Also, we perform the by now familiar conversion of the geometric series:

$$
\begin{aligned}
b'_{1,k} = \frac{1}{W} \Bigg(&\frac{(W-1)W}{2} \left(\rho + \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))} \left(1 - \frac{1-(1-q^\star)^{W-1}}{q^\star} \right) \right) + \\
&(1-\rho) \left(\frac{(W-1)W}{2} - \frac{1}{q^\star} \left((W-1) - \frac{1-(1-q^\star)^{W-1}}{q^\star} \right) \right) \Bigg) \\
= &\frac{\rho(W-1)}{2} + \frac{(W-1)}{2} \frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))} \left(1 - \frac{1-(1-q^\star)^{W-1}}{q^\star} \right) + \\
&\frac{(1-\rho)(W-1)}{2} - \frac{(1-\rho)(W-1)}{W q^\star} + \frac{(1-\rho)}{W q^\star} \frac{1-(1-q^\star)^{W-1}}{q^\star} \qquad \text{(C.15)}
\end{aligned}
$$

Now we can put it all together, and proceed with simplifying:

$$\begin{aligned}\frac{1}{b_{1,-1}} = {} & \frac{(1-\rho)}{W(q_b p + q_e(1-p))}\left(1+\frac{1-(1-q^\star)^{W-1}}{q^\star}\right)+\frac{(1-\rho)(W-1)}{Wq^\star}- \\ & \frac{(1-\rho)}{q^\star}\frac{1-(1-q^\star)^{W-1}}{Wq^\star}+1+\frac{\rho(\mathbf{W}-\mathbf{1})}{\mathbf{2}}+ \\ & \frac{(W-1)}{2}\frac{q_b p(1-\rho)}{W(q_b p + q_e(1-p))}\left(1+\frac{1-(1-q^\star)^{W-1}}{q^\star}\right)+ \\ & \frac{(\mathbf{1}-\rho)(\mathbf{W}-\mathbf{1})}{\mathbf{2}}-\frac{(1-\rho)(W-1)}{Wq^\star}+\frac{(1-\rho)}{q^\star}\frac{1-(1-q^\star)^{W-1}}{Wq^\star}\end{aligned} \tag{C.16}$$

Observe that the third and tenth term cancel eachother, as do the fourth and eleventh. Also, the sixth and ninth can be merged because $\rho + (1-\rho) = 1$. The result:

$$\begin{aligned}\frac{1}{b_{1,-1}} = {} & \frac{(1-\rho)}{W(q_b p + q_e(1-p))}\left(1+\frac{1-(1-q^\star)^{W-1}}{q^\star}\right)+1+ \\ & \frac{(W-1)}{2}+\frac{q_b p(W-1)}{2}\frac{(1-\rho)}{W(q_b p + q_e(1-p))}\left(1+\frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\end{aligned} \tag{C.17}$$

Rewriting yields:

$$\frac{1}{b_{1,-1}} = 1+\frac{(W-1)}{2}+\left(\frac{1-\rho}{W(q_b p + q_e(1-p))}\left(1-\frac{1-(1-q^\star)^{W-1}}{q^\star}\right)\right)\left(1+\frac{q_b p(W-1)}{2}\right) \tag{C.18}$$

□

Compare to the same equation for the DCF:

$$\frac{1}{b_{1,0}} = 1+\frac{(W-1)}{2(1-p^\star)}+\frac{1-\rho}{q}\frac{1-(1-q^\star)^{W}}{Wq^\star}\left(1+\frac{qp(W-1)}{2(1-p^\star)}\right) \tag{C.19}$$

The system does no longer depend on $p^\star$, the state of the medium while counting down. It only depends on the state of the medium to determine whether to invoke BO or not.

APPENDIX D

Online content

The research described in this dissertation resulted in, or has been part of, the following movies, available online:

- Connect&Drive Demo 2011 http://youtu.be/OoRuE7OqFEs
- Connect&Drive Demo 2011 Extra http://youtu.be/FTkWKVBLnL0
- Connect&Drive at InterTraffic 2010 http://youtu.be/3LEyOUmFfuk
- C,MM,N and Connect&Drive http://youtu.be/BVYc_uQOJDQ

Bibliography

[1] B. Klaassen, http://www.bartmediadesigns.com, 2011.

[2] "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, 2012.

[3] "Wireless Access in Vehicular Environments (WAVE): IEEE 1609 Standards Series," *IEEE 1609*, Mar 2012.

[4] " Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions," European Telecommunications Standards Institute: Technical Committee Intelligent Transport System (ITS), Tech. Rep., 6 2009.

[5] International Standardization Organization (ISO) TC 204 Communications Access for Land Mobiles (CALM), http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=54706&development=on, July 2009.

[6] B. A. van, C. J. D. van, and R. Visser, "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, December 2006.

[7] "Connect & drive (C&D), project description," http://www.utwente.nl/ctit/research/projects/concluded/national/senternovem/connect-drive.doc/, July 2009, last checked: 2013-03-06.

[8] "Grand cooperative driving challenge website," http://www.gcdc.net, TNO Science and Industry, last checked: 2013-03-06.

[9] E. M. van Eenennaam, W. Klein Wolterink, G. Karagiannis, and G. J. Heijenk, "Exploring the Solution Space of Beaconing in VANETs," *IEEE Vehicular Networking Conference, VNC, Tokyo, Japan*, Oct 2009.

[10] ——, "Towards Scalable Beaconing in VANETs," *Fourth ERCIM Workshop on e-Mobility, Luleå, Sweden*, pp. 103–108, May 2010.

[11] P. van Wijngaarden and E. M. van Eenennaam, "Frame capture in IEEE 802.11p vehicular networks," *Welcome the Wireless World: Second Workshop on the Pervasive Application of Wireless Technologies*, September 2010.

[12] R. Reinders, E. M. van Eenennaam, G. Karagiannis, and G. J. Heijenk, "Contention window analysis for beaconing in VANETs," in *Seventh IEEE International Wireless Communications and Mobile Computing conference, IWCMC, Istanbul, Turkey*. IEEE Com.Soc., July 2011.

[13] E. M. van Eenennaam, L. Hendriks, G. Karagiannis, and G. J. Heijenk, "Oldest packet drop (OPD): a buffering mechanism for beaconing in IEEE 802.11p VANETs," in *Third IEEE Vehicular Networking Conference, VNC 2011, Amsterdam, The Netherlands*. IEEE Communications Society, November 2011, pp. 252–259.

[14] C. Lei, E. M. van Eenennaam, W. Klein Wolterink, G. Karagiannis, G. Heijenk, and J. Ploeg, "Impact of packet loss on cacc string stability performance," *Proceedings of the Eleventh International Workshop on ITS Telecommunications*, pp. 381–386, Aug 2011.

[15] E. M. van Eenennaam, "microslotted 1-persistence flooding in vanets," in *Welcome the Wireless World: Workshop on the Pervasive Application of Wireless Technologies, Enschede, The Netherlands*. Enschede: Twente University Press, November 2009.

[16] E. M. van Eenennaam and G. J. Heijenk, "Providing over-the-horizon awareness to driver support systems," *Proc. of 4th IEEE Workshop on Vehicle to Vehicle Communications (V2VCOM)*, June 2008.

[17] E. M. van Eenennaam, A. van de Venis, and G. Karagiannis, "Impact of IEEE 1609.4 channel switching on the IEEE 802.11p beaconing performance," in *Wireless Days (WD), 2012 IFIP*, nov. 2012, pp. 1 –8.

[18] E. M. van Eenennaam, A. K. I. Remke, and G. J. Heijenk, "An analytical model for beaconing in VANETs," in *IEEE Vehicular Networking Conference (VNC) 2012, Seoul, South Korea*, nov. 2012, pp. 9 –16.

[19] G. J. Heijenk, A. K. I. Remke, and E. M. van Eenennaam, "Performance Comparison of IEEE 802.11 DCF and EDCA for Beaconing in Vehicular Networks (submitted)," in *IEEE Vehicular Networking Conference (VNC) 2013, Boston, U.S.A.*, dec. 2013, pp. 9 –16.

[20] K. Z. Ghafoor, K. A. Bakar, E. M. van Eenennaam, R. H. Khokhar, and A. J. Gonzalez, "A Fuzzy Logic Approach to Beaconing for Vehicular Ad hoc Networks," *Telecommunication Systems Journal*, pp. 1–11, January 2011, 10.1007/s11235-011-9466-8. [Online]. Available: http://dx.doi.org/10.1007/s11235-011-9466-8

[21] C. Lei, E. M. van Eenennaam, W. Klein Wolterink, J. Ploeg, G. Karagiannis, and G. Heijenk, "Evaluation of CACC string stability using SUMO, Simulink, and OMNeT++," *EURASIP Journal on Wireless Communications and Networking*, 2012.

[22] E. M. van Eenennaam, G. J. Heijenk, G. Karagiannis, and B. van Arem, "Over-the-horizon awareness for advanced driver assistance systems: the trafficfilter and microslotted 1-persistence flooding," http://eprints.eemcs.utwente.nl/20528/, Centre for Telematics and Information Technology University of Twente, Enschede, Technical Report TR-CTIT-11-21, September 2011.

[23] Connect & Drive project, Project no. HTASD08002, "Connect&Drive: WP-1 Requirements for Cooperative Adaptive Cruise Control based on Wireless Communication," Tech. Rep., April 2009.

[24] E. M. van Eenennaam, G. J. Heijenk, G. Karagiannis, W. Klein Wolterink, and T. Lippmann, "Connect&drive deliverable: Communication test & demo evaluation," University of Twente, Twente Institute for Wireless and Mobile Communications B.V., Tech. Rep., March 2011.

[25] ——, "Connect&drive deliverable: Beaconing algorithms & analysis," University of Twente, Twente Institute for Wireless and Mobile Communications B.V., Tech. Rep., March 2011.

[26] E. M. van Eenennaam, "FUTURUM GCDC communication architecture," University of Twente - Design and Analysis of Communication Systems group, Team FUTURUM, Tech. Rep., Nov 2010.

[27] R. Kennedy, *The Book of Modern Engines and Power Generators (Vol. III ed.)*. London: Caxton Publishing Company, 1912 edition of 1905 book, *public domain*.

[28] M. Rauck and J. Cugnot, *1769–1969: der Urahn unseres Autos fuhr vor 200 Jahren*. München: Münchener Zeitungsverlag, 1969.

[29] D. G. Challenge, http://www.darpagrandchallenge.com/, last checked: 2013-03-06.

[30] E. Guizzo, "How google's self-driving car works," http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works, 18 Oct 2011, last checked: 2013-07-23.

[31] M. Maier, "Architecting principles for system of systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[32] P. E. Ioannou, *Automated Highway Systems*. Springer, Jan 1997.

[33] University of Nagoya, Japan, "Shockwave traffic jams recreated for the first time," http://youtu.be/Suugn-p5C1M, last checked: 2013-03-06.

[34] R. Pueboobpaphan and B. van Arem, "Understanding the Relation between Driver/Vehicle Characteristics and Platoon/Traffic Flow Stability for the Design and Assessment of Cooperative Adaptive Cruise Control," *Proc. Transportation Research Board 89th Annual Meeting, Washington DC, USA*, 2010.

[35] A. Bose and P. Ioannou, "Analysis of traffic flow with mixed manual and intelligent cruise control vehicles: Theory and experiments," *California PATH Research Report UCB-ITS-PRR-2001-13*, april 2001.

[36] P. Ioannou and C. Chien, "Autonomous intelligent cruise control," *Vehicular Technology, IEEE Transactions on*, vol. 42, no. 4, pp. 657–672, Nov 1993.

[37] T. Ayres, L. Li, D. Schleuning, and D. Young, "Preferred time-headway of highway drivers," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, 2001, pp. 826 –829.

[38] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-stable cacc design and experimental validation: A frequency-domain approach," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 4268–4279, Nov 2010.

[39] S. Kumarawadu, *Control systems : theory and implementation*. Alpha Science International, 2010.

[40] A. Bodhankar, "Cooperative adaptive cruise control (CACC)," *KReSIT, IIT Bombay*, Nov 2004.

[41] G. Naus, R. Vugts, J. Ploeg, R. van de Molengraft, and M. Steinbuch, "Cooperative adaptive cruise control, design and experiments," pp. 6145 –6150, Jul 2010.

[42] Connect&Drive consortium, "Cooperative Adaptive Cruise control based on WiFi communication between vehicles and infrastructure, Project Plan HTAS," HTAS Driving Guidance Priority, Tech. Rep., July 2008.

[43] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and Experimental Evaluation of Cooperative Adaptive Cruise Control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, oct. 2011, pp. 260 –265.

[44] J. Zhou and H. Peng, “Range policy of adaptive cruise control vehicles for improved flow stability and string stability,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 2, pp. 229 – 237, june 2005.

[45] A. D. May, *Traffic Flow Fundamentals*. Prentice-Hall, 1992.

[46] F. L. Mannering, W. P. Kilareski, and S. S. Washburn, *Principles of Highway Engineering and Traffic Analysis*. John Wiley & Sons, Inc., 2004.

[47] S. Tsugawa, S. Kato, K. Tokuda, T. Matsui, and H. Fujii, “An architecture for cooperative driving of automated vehicles,” pp. 422–427, 2000.

[48] I. Wilmink, G. Klunder, and B. van Arem, “Traffic flow effects of integrated full-range speed assistance (IRSA),” *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 1204–1210, 13-15 June 2007.

[49] C. van Driel and B. van Arem, “Traffic flow impacts of a congestion assistant,” *Proc. TRB 87th Annual Meeting, Washington DC, USA*, Jan 2008.

[50] N. Ghouti, A. Serrarens, D. Van Sambeek, and J. Ploeg, “Connect&Drive C-ACC for Reducing Congestion Dynamics,” *16th ITS World Congres*, Oct 2009.

[51] W. Klein Wolterink, G. J. Heijenk, and G. Karagiannis, “Automated Merging in a Cooperative Adaptive Cruise Control (CACC) System,” in *Fifth ERCIM Workshop on eMobility, Vilanova i la Geltrú, Catalonia, Spain*. Barcelona: Technical University of Catalonia, June 2011, pp. 23–26.

[52] W. Klein Wolterink, G. J. Heijenk, and G. Karagiannis, “Constrained geocast to support cooperative adaptive cruise control (cacc) merging,” in *Vehicular Networking Conference (VNC), 2010 IEEE*, dec. 2010, pp. 41 –48.

[53] ——, “Dissemination protocols to support cooperative adaptive cruise control (cacc) merging,” in *ITS Telecommunications (ITST), 2011 11th International Conference on*, aug. 2011, pp. 15 –20.

[54] O. Baijer, J. de Jongh, A. van Neerbos, B. Netten, and H. Wedemeijer, “GCDC interaction protocol v2.4,” Strategic Platform for Intelligent Traffic Systems (SPITS) and Grand Cooperative Driving Challenge (GCDC), Tech. Rep., April 2011. [Online]. Available: http://www.gcdc.net/mainmenu/Home/downloads/Technology

[55] “Intelligent Transport Systems (ITS); Communications Architecture,” European Telecommunications Standards Institute: Technical Committee Intelligent Transport System (ITS), Tech. Rep., 9 2010.

[56] G. Karagiannis, O. Altintas, E. Ekici, G. J. Heijenk, B. Jarupan, K. Lin, and T. Weil, “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions,” *Communications Surveys Tutorials, IEEE*, vol. 13, no. 4, pp. 584 –616, quarter 2011.

[57] S. Tsugawa, “Issues and recent trends in vehicle safety communication systems,” *IATSS Res (Int Assoc Traffic Saf Sci)*, vol. 29, no. 1, pp. 7–15, 2005.

[58] ——, “Inter-vehicle communications and their applications to intelligent vehicles: an overview,” *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, pp. 564–569 vol.2, 17-21 June 2002.

[59] “Car-2-car communication consortium website,” http://www.car-2-car.org, last checked: 2013-03-06.

[60] "C2C-CC Manifesto: Overview of the C2C-CC system version 1.1," Car 2 Car Communication Consortium, Tech. Rep., Aug 2007.

[61] CVIS Project, http://www.cvisproject.org, last checked: 2013-03-07.

[62] COMeSafety, "D31 European ITS Communication Architecture," Information Society Technologies, Tech. Rep., 2008.

[63] Safespot Project, http://www.safespot-eu.org, last checked: 2013-03-07.

[64] "DRAFT INTERNATIONAL STANDARD ISO/DIS 21215 - Intelligent transport systems - Communications access for land mobiles (CALM) - CALM M5," ISO/TC 204, Tech. Rep., april 2009.

[65] " Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band ," European Telecommunications Standards Institute: Technical Committee Intelligent Transport System (ITS), Tech. Rep., 7 2013.

[66] T. Weil, "Wireless Access in Vehicular Environments (WAVE) - the emerging IEEE 1609 standard," Nov 2007.

[67] J. Zhu and S. Roy, "MAC for dedicated short range communications in intelligent transport system," *Communications Magazine, IEEE*, vol. 41, no. 12, pp. 60–67, Dec. 2003.

[68] C. Campolo, A. Molinaro, and A. Vinel, "Understanding the performance of short-lived control broadcast packets in 802.11p/WAVE vehicular networks," in *Vehicular Networking Conference (VNC), 2011 IEEE*, nov. 2011, pp. 102 –108.

[69] "IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Multi-channel Operation," *IEEE Std 1609.4-2010 (Revision of IEEE Std 1609.4-2006)*, pp. 1 –89, 7 2011.

[70] N. Probert, "SAE J2735 Whitepaper - DRAFT Dedicated Short Range Communications (DSRC) Heartbeat Messages: Big vs. Small," Nissan Technical Center North America, Tech. Rep., Aug 2007.

[71] Information Society Technologies, "D31 european its communication architecture overall framework proof of concept implementation," Information Society Technologies, Tech. Rep., March 2009.

[72] COMeSafety, "D31 European ITS Communication Architecture - annex 10," Information Society Technologies, Tech. Rep., 2008.

[73] W. Klein Wolterink and Michel, "Comments to the connect&drive message format v1.4.1," University of Twente, Tech. Rep., October 2009.

[74] W. Klein Wolterink, "Comments to the connect&drive message format v1.4.1," University of Twente, Tech. Rep., October 2009.

[75] K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, "On the ability of the 802.11p MAC method and STDMA to support real-time vehicle-to-vehicle communication," *EURASIP J. Wirel. Commun. Netw.*, pp. 1–13, 2009.

[76] K. Sjöberg, E. Uhlemann, and E. Ström, "How severe is the hidden terminal problem in VANETs when using CSMA and STDMA?" in *Vehicular Technology Conference (VTC Fall)*, sept. 2011, pp. 1 –5.

[77] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," *IEEE Std 802.3, 1998 Edition*, p. i, 1998.

[78] "Broadband Radio Access Networks (BRAN); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional specification," *ETSI EN 300 652 V1.2.1* , Jul 1998.

[79] "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer," *ETSI TS 101 475 v1.2.2*, Feb 2001.

[80] "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 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, June 12 2007.

[81] "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 amendment 6: Wireless access in vehicular environments," *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1 –51, 15 2010.

[82] "Supplement to 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: High-Speed Physical Layer in the 5 GHz Band," *IEEE Std 802.11a-1999*, p. i, 1999.

[83] Task Group E, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications - amendment 8: Medium access control (MAC) qos enhancements," *IEEE 802.11e 2005*, Nov 2005.

[84] W. P. D'Amico and M. H. Lauss, "Wireless local area network flight demonstration for high doppler conditions," *Johns Hopkins APL digest, volume 24, number 4*, 2004.

[85] S. Sibecas, C. A. Corral, S. Emami, and G. Stratis, "On the suitability of 802.11a/RA for high-mobility DSRC," *VTC*, 2002.

[86] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, 11-14 2008, pp. 2036 –2040.

[87] U. M. Khaldoun Albarazi and N. Al-holou, "Doppler shift impact on vehicular ad-hoc networks," *Canadian Journal on Multimedia and Wireless Networks*, vol. 2, no. 3, pp. 46–56, August 2011.

[88] R. Jain, *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation and modeling*. John Wiley & Sons, Inc., 1991.

[89] B. R. Haverkort, *Performance of Computer Communication Systems - A Model-Based Approach*. John Wiley & Sons, Ltd., 1998.

[90] "OMNeT++ community website," http://www.omnetpp.org/.

[91] G. Pongor, "Omnet: Objective modular network testbed," in *MASCOTS '93: Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems*. San Diego, CA, USA: The Society for Computer Simulation, International, 1993, pp. 323–326.

[92] MiXiM - Simulation Framework for Wireless and Mobile Networks, http://mixim.sourceforge.net, last checked: 2013-03-07.

[93] "Mobility framework for omnet++," http://mobility-fw.sourceforge.net/.

[94] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in omnet++ the mixim vision," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 71:1–71:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=1416222.1416302

[95] C. E. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing - RFC 3561," https://tools.ietf.org/html/rfc3561, July 2003, last checked: 2013-03-07.

[96] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, Oct. 1994. [Online]. Available: http://doi.acm.org/10.1145/190809.190336

[97] Clausen, T. and Jacquet, P., "Optimized link state routing protocol (olsr) - rfc 3626," http://tools.ietf.org/html/rfc3626, October 2003, last checked: 2013-03-07.

[98] R. S. Schwartz, K. Das, H. Scholten, and P. Havinga, "Exploiting beacons for scalable broadcast data dissemination in vanets," in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*, ser. VANET '12. New York, NY, USA: ACM, 2012, pp. 53–62. [Online]. Available: http://doi.acm.org/10.1145/2307888.2307899

[99] "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization," European Telecommunications Standards Institute: Technical Committee Intelligent Transport System (ITS), Tech. Rep., 6 2011.

[100] P. Szczurek, B. Xu, O. Wolfson, and J. Lin, "Estimating relevance for the emergency electronic brake light application," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –19, 2012.

[101] F. Basma, Y. Tachwali, and H. Refai, "Intersection collision avoidance system using infrastructure communication," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, oct. 2011, pp. 422 –427.

[102] M. D. Hill, "What is scalability?" *SIGARCH Comput. Archit. News*, vol. 18, no. 4, pp. 18–21, Dec. 1990. [Online]. Available: http://doi.acm.org/10.1145/121973.121975

[103] A. B. Bondi, "Characteristics of scalability and their impact on performance," in *Proceedings of the 2nd international workshop on Software and performance*, ser. WOSP '00. New York, NY, USA: ACM, 2000, pp. 195–203. [Online]. Available: http://doi.acm.org/10.1145/350391.350432

[104] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 153–167, 2002.

[105] L. Cheng, B. Henty, D. Stancil, F. Bai, and P. Mudalige, "Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frequency Band," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 8, pp. 1501–1516, 2007.

[106] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 11–33, Jan.-March.

[107] T. Lippmann, “Connect & drive - wireless communication architecture v0.2,” Ti-WMC, Tech. Rep., 2010.

[108] A. Vinel, Y. Koucheryavy, S. Andreev, and D. Staehle, “Estimation of a successful beacon reception probability in vehicular ad-hoc networks,” pp. 416–420, 2009. [Online]. Available: http://doi.acm.org/10.1145/1582379.1582470

[109] L. Yang, J. Guo, and Y. Wu, “Channel Adaptive One Hop Broadcasting for VANETs,” in *IEEE ITSC 2008*, Oct. 2008, pp. 369–374.

[110] M. Torrent-Moreno, “Inter-vehicle communications: Achieving safety in a distributed wireless environment: Challenges, systems and protocols,” Ph.D. dissertation, Universität Fridericiana zu Karlsruhe, 2007.

[111] A. Athanasopoulos, E. Topalis, C. Antonopoulos, and S. Koubias, “Evaluation Analysis of the Performance of IEEE 802.11b and IEEE 802.11g Standards,” in *ICN/ICONS/MCL*, April 2006, pp. 141–141.

[112] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi, “An experimental study on the capture effect in 802.11a networks,” in *WinTECH '07*. ACM, 2007, pp. 19–26.

[113] Ti-WMC, “Flame, forwading layer for meshing,” http://www.ti-wmc.nl/images/stories/downloads/flame-wp-1.0.pdf, Tech. Rep., Dec 2005.

[114] N. Balon and J. Guo, “Increasing broadcast reliability in vehicular ad hoc networks,” in *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*. New York, NY, USA: ACM, 2006, pp. 104–105.

[115] UT-DACS/Ti-WMC, “Connect & drive - communication logging list v0.5,” UT / Ti-WMC, Tech. Rep., April 2010.

[116] Connect&Drive project, “Connect and drive demo 2011,” http://youtu.be/OoRuE7OqFEs, March 2011, last checked: 2013-03-12.

[117] J. Ploeg, “Connect & Drive Prius Commissioning Report (internal report),” TNO, Connect&Drive project, Tech. Rep., Mar 2011.

[118] M. Oh, “An experimental comparison of packet delivery schemes in a linear mesh topology,” in *Information Science and Applications (ICISA), 2012 International Conference on*, may 2012, pp. 1 –4.

[119] J. de Jongh, “GCDC communications stack - building and installing the CALM FAST router,” SPITS - Strategic Platform for Intelligent Traffic Systems, Tech. Rep., Nov 2010.

[120] E. Koenders, “CALM FAST router, design, v0.1, SPITS Deliverable,” Tech. Rep., Mar 2010.

[121] “Simulation of Urban MObility (SUMO),” http://sumo.sourceforge.net/, last checked: 2013-03-07.

[122] “Simulink introduction in mathworks official website (visited in may 2011).” [Online]. Available: http://www.mathworks.com/products/simulink/description1.html

[123] C. Sommer, Z. Yao, R. German, and F. Dressler, “On the need for bidirectional coupling of road traffic microsimulation and network simulation,” in *the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc 2008): 1st ACM SIGMOBILE International Workshop on Mobility Models for Networking Research (Mobility-Models'08), Hong Kong, China*, 2008.

[124] 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, pp. 3–15, 2011.

[125] C. Lei, “Cooperative adaptive cruise control model study based on traffic & network simulation,” Tech. Rep., 2011. [Online]. Available: http://www.utwente.nl/ewi/dacs/assignments/completed/bachelor/reports/2011-lei_chenxi.pdf

[126] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “Sumo - simulation of urban mobility: An overview,” in *SIMUL 2011, The Third International Conference on Advances in System Simulation*, Barcelona, Spain, 2011.

[127] T.-S. Ho and K.-C. Chen, “Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol,” in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC’96., Seventh IEEE International Symposium on*, vol. 2, oct 1996, pp. 407 –411 vol.2.

[128] G. Bianchi, L. Fratta, and M. Oliveri, “Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs,” in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC’96., Seventh IEEE International Symposium on*, vol. 2, oct 1996, pp. 392 –396 vol.2.

[129] H. Chhaya and S. Gupta, “Performance of asynchronous data transfer methods of IEEE 802.11 MAC protocol,” *Personal Communications, IEEE*, vol. 3, no. 5, pp. 8 –15, oct 1996.

[130] J. Hui and M. Devetsikiotis, “Performance analysis of IEEE 802.11 e EDCA by a unified model,” in *Global Telecommunications Conference, 2004. GLOBECOM’04. IEEE*, vol. 2. IEEE, 2004, pp. 754–759.

[131] F. Cali, M. Conti, and E. Gregori, “Ieee 802.11 wireless lan: capacity analysis and protocol enhancement,” in *INFOCOM ’98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, 1998, pp. 142–149 vol.1.

[132] Y. Wang, A. Ahmed, B. Krishnamachari, and K. Psounis, “IEEE 802.11p Performance Evaluation and Protocol Enhancement,” *IEEE intl. conf. on Vehicular Electronics and safety*, Sept 2008.

[133] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, “Performance of reliable transport protocol over IEEE 802.11 wireless LAN: analysis and enhancement,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2002, pp. 599 – 607 vol.2.

[134] A. Banchs and L. Vollero, “Throughput analysis and optimal configuration of 802.11 e edca,” *Computer Networks*, vol. 50, no. 11, pp. 1749–1768, 2006.

[135] P. Chatzimisios, A. Boucouvalas, and V. Vitsas, “Ieee 802.11 packet delay-a finite retry limit analysis,” in *Global Telecommunications Conference, 2003. GLOBECOM ’03. IEEE*, vol. 2, 2003, pp. 950–954 Vol.2.

[136] N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai, and V. Sadekar, “Broadcast storm mitigation techniques in vehicular ad hoc networks,” *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 84–94, December 2007.

[137] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *MobiHoc ’02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2002, pp. 194–205.

[138] G. Bianchi, “IEEE 802.11-saturation throughput analysis,” *Communications Letters, IEEE*, vol. 2, no. 12, pp. 318–320, dec 1998.

[139] ——, “Performance analysis of the IEEE 802.11 Distributed Coordination Function,” *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, Mar 2000.

[140] E. Ziouva and T. Antonakopoulos, “CSMA/CA performance under high traffic conditions: throughput and delay analysis,” *Computer Communications*, vol. 25, no. 3, pp. 313 – 321, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366401003693

[141] G. Wang, Y. Shu, L. Zhang, and O. Yang, “Delay analysis of the IEEE 802.11 DCF,” in *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, vol. 2, sept. 2003, pp. 1737 – 1741 vol.2.

[142] L. Zhang, Y. Shu, O. Yang, and G. Wang, “Study of Medium Access Delay in IEEE 802.11 Wireless Networks,” *IEICE TRANSACTIONS on Communications Vol.E89-B*, no. 4, pp. 1284–1293, April 2006.

[143] A. Abdrabou and W. Zhuang, “Service Time Approximation in IEEE 802.11 Single-Hop Ad-hoc Networks,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 2346–2350.

[144] I. Tinnirello, G. Bianchi, and Y. Xiao, “Refinements on IEEE 802.11 Distributed Coordination Function Modeling Approaches,” *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 3, pp. 1055–1067, March.

[145] Y. Xiao, “Performance analysis of IEEE 802.11e EDCF under saturation condition,” in *Communications, 2004 IEEE International Conference on*, vol. 1, june 2004, pp. 170 – 174 Vol.1.

[146] J. Hui and M. Devetsikiotis, “A unified model for the performance analysis of IEEE 802.11e EDCA,” *Communications, IEEE Transactions on*, vol. 53, no. 9, pp. 1498 – 1510, sept. 2005.

[147] Y. C. Tay and K. C. Chua, “A capacity analysis for the IEEE 802.11 MAC protocol,” *Wirel. Netw.*, vol. 7, no. 2, pp. 159–171, Mar. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1016637622896

[148] L. Xiong and G. Mao, “An Analysis of the Coexistence of IEEE 802.11 DCF and IEEE 802.11e EDCA,” in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, march 2007, pp. 2264 –2269.

[149] N. Taher, Y. Doudane, and B. El Hassan, “A complete and accurate analytical model for 802.11e edca under saturation conditions,” in *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, May 2009, pp. 800–807.

[150] I. Tinnirello and G. Bianchi, “Rethinking the IEEE 802.11e EDCA Performance Modeling Methodology,” *Networking, IEEE/ACM Transactions on*, vol. 18, no. 2, April 2010.

[151] X. Yang, R. P. Liu, and M. Hedley, “A channel access cycle based model for IEEE 802.11e EDCA in unsaturated traffic conditions,” in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, April, pp. 1496–1501.

[152] E. Felemban and E. Ekici, “Single Hop IEEE 802.11 DCF Analysis Revisited: Accurate Modeling of Channel Access Delay and Throughput for Saturated and Unsaturated Traffic Cases,” *Wireless Communications, IEEE Transactions on*, vol. 10, no. 10, pp. 3256–3266, 2011.

[153] D. Malone, K. Duffy, and D. Leith, “Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions,” *Networking, IEEE/ACM Transactions on*, vol. 15, no. 1, pp. 159–172, Feb. 2007.

[154] N. Dao and R. Malaney, "A new markov model for non-saturated 802.11 networks," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, Jan. 2008, pp. 420–424.

[155] R. P. Liu, G. Sutton, and I. Collings, "A New Queueing Model for QoS Analysis of IEEE 802.11 DCF with Finite Buffer and Load," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 8, pp. 2664–2675, 2010.

[156] P. E. Engelstad and O. N. Østerbø, "Non-saturation and saturation analysis of IEEE 802.11e EDCA with starvation prediction," in *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2005, pp. 224–233.

[157] ——, "An analytical model of the virtual collision handler of 802.11e," in *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '05. New York, NY, USA: ACM, 2005, pp. 255–259. [Online]. Available: http://doi.acm.org/10.1145/1089444.1089489

[158] ——, "Delay and Throughput Analysis of IEEE 802.11e EDCA with Starvation Prediction," in *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, nov. 2005, pp. 647 –655.

[159] ——, "Queueing Delay Analysis of IEEE 802.11e EDCA," in *WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services.*, 2006.

[160] ——, "The delay distribution of IEEE 802.11e EDCA and 802.11 DCF," in *Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International*, april 2006, pp. 10 pp. –96.

[161] ——, "Analysis of the Total Delay of IEEE 802.11e EDCA and 802.11 DCF," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 2, june 2006, pp. 552 –559.

[162] N. Wisitpongphan and O. Tonguz, "Scalable broadcast strategies for ad hoc routing protocols," *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 6 pp.–, Jan. 2006.

[163] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pp. 3866–3875, Jan. 2002.

[164] P. Rani, N. Sharma, and P. Singh, "Performance comparison of vanet routing protocols," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, 2011, pp. 1–4.

[165] X. Chen, H. Refai, and X. Ma, "Saturation Performance of IEEE 802.11 Broadcast Scheme in Ad Hoc Wireless LANs," in *IEEE VTC*, 30 2007-Oct. 3 2007, pp. 1897–1901.

[166] X. Ma and X. Chen, "Delay and broadcast reception rates of highway safety applications in vehicular ad hoc networks," in *Mobile Networking for Vehicular Environments*, 2007, pp. 85–90.

[167] A. Rao, A. Kherani, and A. Mahanti, "Performance evaluation of 802.11 broadcasts for a single cell network with unsaturated nodes," in *Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet*, ser. NETWORKING'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 836–847. [Online]. Available: http://dl.acm.org/citation.cfm?id=1792514.1792606

[168] A. Vinel, V. Vishnevsky, and Y. Koucheryavy, "A simple analytical model for the periodic broadcasting in vehicular ad-hoc networks," in *GLOBECOM Workshops, 2008 IEEE*, 30 2008-Dec. 4 2008, pp. 1–5.

[169] C. Campolo, A. Vinel, A. Molinaro, and Y. Koucheryavy, "Modeling broadcasting in IEEE 802.11p/WAVE vehicular networks," *Communications Letters, IEEE*, vol. 15, no. 2, pp. 199 –201, february 2011.

[170] J. J. Blum and A. Eskandarian, "Avoiding Timeslot Boundary Synchronization for Multihop Message Broadcast in Vehicular Networks," pp. 1 –5, april 2009.

[171] C. H. Foh and J. Tantra, "Comments on IEEE 802.11 saturation throughput analysis with freezing of backoff counters," *Communications Letters, IEEE*, vol. 9, no. 2, pp. 130–132, 2005.

[172] M. Benaïm and J.-Y. Le Boudec, "A class of mean field interaction models for computer and communication systems," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, April, pp. 589–590.

[173] C. Bordenave, D. McDonald, and A. Proutiére, "Random multi-access algorithms - a mean field analysis," Tech. Rep. 5632, 2005.

[174] M. Garetto and C.-F. Chiasserini, "Performance analysis of the 802.11 distributed coordination function under sporadic traffic," pp. 1343–1347, 2005. [Online]. Available: http://dx.doi.org/10.1007/11422778_111

[175] V. Ramaiyan, A. Kumar, and E. Altman, "Fixed Point Analysis of Single Cell IEEE 802.11e WLANs: Uniqueness and Multistability," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 5, pp. 1080–1093, Oct. 2008.

[176] R. Pueboobpaphan, F. Liu, and B. Van Arem, "The impacts of a communication based merging assistant on traffic flows of manual and equipped vehicles at an on-ramp using traffic flow simulation," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010, pp. 1468–1473.

[177] M. Killat and H. Hartenstein, "Vehicular ad hoc networks: How to show the impact on traffic safety?" in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, 2007, pp. 659–663.

[178] H. T. Friis, "A note on a simple transmission formula," *Proceedings of IRE*, vol. 34, pp. 254–256, May 1946.

[179] F. Baccelli, F. Karpelevich, M. Kelbert, A. Puhalskii, A. Rybko, and Y. Suhov, "A mean-field limit for a class of queueing networks," *Journal of Statistical Physics*, vol. 66, pp. 803–825, 1992. [Online]. Available: http://dx.doi.org/10.1007/BF01055703

[180] G. Bianchi, I. Tinnirello, and L. Scalia, "Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations," *Network, IEEE*, vol. 19, no. 4, pp. 28 – 34, july-aug. 2005.

[181] P. van Wijngaarden, "Frame capture in IEEE 802.11p vehicular networks - a simulation-based approach," 2011, master Thesis.

[182] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, pp. 1805–1824, Aug 2000.

[183] E. M. van Eenennaam, "Vehicle-to-vehicle communication for driver support systems," December 2008, master Thesis.

[184] "Quadstone Paramics traffic microsimulation software," http://www.paramics-online.com/, last checked: 2013-03-07.

[185] R. S. Schwartz, "Towards an over-the-horizon awareness to driver support systems in highway real-world scenarios," August 2009, master Thesis.

[186] R. S. Schwartz, E. M. van Eenennaam, G. Karagiannis, G. J. Heijenk, W. Klein Wolterink, and H. Scholten, "Using v2v communication to create over-the-horizon awareness in multiple-lane highway scenarios," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, june 2010, pp. 998 –1005.

[187] C. Sommer, "Traffic control interface (traci) modules for omnet++," http://www7.informatik.uni-erlangen.de/ sommer/omnet/traci/.

[188] S. Robinson and A. Ioannu, "The problem of the initial transient: Techniques for estimating the warm-up period for discrete-event simulation models," *Warwick Business School, University of Warwick, Coventry, UK*, 2007.

[189] " Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," European Telecommunications Standards Institute: Technical Committee Intelligent Transport System (ITS), Tech. Rep., 5 2011.

[190] L. Stibor, Y. Zang, G. R. Hiertz, S. Max, and H.-J. Reumerman, "Congestion control in IEEE 802.11p - IEEE 802.11-06/1534r1," Tech. Rep., 2006.

[191] W. Zhang, A. Festag, R. Baldessari, and L. Le, "Congestion control for safety messages in VANETs: Concepts and framework," in *ITST*, Oct. 2008, pp. 199–203.

[192] S. Sharafkandi, G. Bansal, J. Kenney, and D. Du, "Using edca to improve vehicle safety messaging," in *Vehicular Networking Conference (VNC), 2012 IEEE*, Nov., pp. 70–77.

[193] M. Torrent-Moreno, P. Santi, and H. Hartenstein, "Distributed fair transmit power adjustment for vehicular ad hoc networks," in *SECON '06*, vol. 2, Sept. 2006, pp. 479–488.

[194] R. K. Schmidt, T. Leinmüller, E. Schoch, F. Kargl, and G. Schäfer, "Exploration of adaptive beaconing for efficient intervehicle safety communication," *Netwrk. Mag. of Global Internetwkg.*, vol. 24, pp. 14–19, January 2010. [Online]. Available: http://dx.doi.org/10.1109/MNET.2010.5395778

[195] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, june 2011, pp. 350 –358.

[196] S. Öncü, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Tradeoffs between control and network specifications," *IEEE ITSC 2011*, October 2011.

[197] K. H. Suleiman, T. Javidi, M. Liu, and S. Kittipiyakul, "The Impact of MAC Buffer Size on the Throughput Performance of IEEE 802.11," *43rd Annual Allerton Conference on Communication, Control and Computing*, 2005.

[198] C. Brandauer, G. Iannaccone, C. Diot, T. Ziegler, S. Fdida, and M. May, "Comparison of tail drop and active queue management performance for bulk-data and web-like internet traffic," *Computers and Communications, IEEE Symposium on*, vol. 0, p. 0122, 2001.

[199] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *Networking, IEEE/ACM Transactions on*, vol. 1, no. 4, pp. 397 –413, aug 1993.

[200] S. Eichler, "Performance evaluation of the IEEE 802.11p WAVE communication standard," *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 2199–2203, 30 2007-Oct. 3 2007.

[201] C.-M. Huang, C.-C. Yang, and H.-D. Huang, "An Effective Channel Utilization Scheme for IEEE 1609.4 Protocol," in *Ubiquitous Information Technologies Applications, 2009. ICUT '09. Proceedings of the 4th International Conference on*, dec. 2009, pp. 1 –6.

[202] L. Zhang, Y. Liu, Z. Wang, J. Guo, Y. Huo, Y. Yao, C. Hu, and Y. Sun, "Evaluating and improving the performance of IEEE 802.11p/1609.4 networks with single channel devices," in *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*, sept. 2011, pp. 877 –881.

[203] S.-Y. Wang, H.-L. Chao, K.-C. Liu, T.-W. He, C.-C. Lin, and C.-L. Chou, "Evaluating and improving the TCP/UDP performances of IEEE 802.11(p)/1609 networks," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, july 2008, pp. 163 –168.

[204] A. Ahmad, M. Doughan, I. Mougharbel, and M. Marot, "A new adapted back-off scheme for broadcasting on IEEE 1609.4 control channel in VANET," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, 2012, pp. 9–15.

[205] R. Lasowski, C. Scheuermann, F. Gschwandtner, and C. Linnhoff-Popien, "Evaluation of adjacent channel interference in single radio vehicular ad-hoc networks," *Intelligent Vehicular Communications System workshop, co-located with CCNC 2011, Las Vegas, USA*, pp. 267–271, 2011.

[206] "Intelligent Transport Systems (ITS); European Profile Standard for the Physical and Medium Access Control Layer of Intelligent Transport Systems Operating in the 5 GHz Frequency Band," European Telecommunications Standards Instititute, Tech. Rep., 2010.

[207] "SIM$^{\text{TD}}$: Safe and Intelligent Mobility - Test Field Germany - integrated car-to-x communication system," http://www.simtd.de, SIM$^{\text{TD}}$ Consortium, Oct 2011, last checked: 2013-08-19.

[208] "SCORE@F: Système Coopératif Routier Expérimental @ France," https://project.inria.fr/scoref/en, SCORE@F Consortium, last checked: 2013-08-19.

[209] " Freeband Awareness: context AWARE mobile NEtworks and ServiceS ," http://www.utwente.nl/ewi/telemedicine/Projects/freebandawareness.doc/, Freeband consortium, last checked: 2013-08-31.

[210] "C,mm,n: Open source mobility," http://www.cmmn.org/, C,MM,N project, last checked: 2013-08-31.

List of acronyms

ABS Anti-lock Breaking System. 1, 11
AC Access Category. 41–43, 59, 116, 134, 153, 159, 180
ACC Adaptive Cruise Control. 14, 15, 18, 56, 63, 88, 97, 99, 100, 102, 108
ADAS Advanced Driver Assistance System. 2
AIFS Arbitration Interframe Space. 42, 116, 159
AODV Ad hoc On-demand Distance Vector. 55, 85
AP Access Point. 28, 29, 37, 39, 43, 51, 54, 116
ARIB Association of Radio Industries and Businesses. 20
AU Application Unit. 22, 23

bc backoff counter. 30, 32, 33, 43, 49, 68, 111, 113–115, 119, 123, 134, 135, 152, 160, 178, 179
BEB Binary Exponential Backoff. 33
BER Bit Error Rate. 47, 50, 51, 72, 138
BO backoff. 122
BPSK Binary Phase Shift Keying. 46
BSM Basic Safety Messages. 24
BSS Basic Service Set. 39, 40

C2C-CC Car 2 Car Communication Consortium. 20, 85
CA Collision Avoidance. 30, 116
CACC Cooperative Adaptive Cruise Control. 2, 5, 6, 12, 15, 17–19, 51, 53, 54, 56, 57, 63, 76, 84, 88, 91, 95, 98, 99, 102, 108, 140, 173, 180, 181
CALM Communications Access for Land Mobiles. 20
CAM Cooperative Awareness Message. 26
CAN Controller Area Network. 21
CC Cruise Control. 18
CCC Cooperative Cruise Control. 18
CCH Control Channel. 24, 94, 118, 193, 194, 196, 198
COTS Commercial-off-the-Shelf. 96
CRC Cyclic Redundancy Check. 32, 33, 38, 65
CS Carrier Sense. 30, 31, 36, 72, 74
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. 9, 30, 49, 51, 58, 59, 73, 81, 109, 111, 113, 120, 155, 184
CSMA/CD Carrier Sense Multiple Access with Collision Detection. 58
CTS Clear-To-Send. 34
CW Contention Window. 30–33, 35, 41–43, 49, 68, 115, 118, 122, 139, 159, 169–171, 173, 178, 179

DCF Distributed Coordination Function. 29, 30, 41, 50, 74, 113, 116, 133, 134, 138, 153, 158, 159, 166, 183, 187
DIFS DCF Interframe Space. 36, 42, 159
DSDV Destination-Sequenced Distance Vector. 55
DSRC Dedicated Short Range Communication. 23
DSSS Direct Sequence Spread Spectrum. 43, 44, 46
DTMC Discrete Time Markov Chain. 5, 113, 121, 122, 125, 138

EDCA Enhanced Distributed Channel Access. 29, 41, 51, 112, 114, 116, 117, 133, 138, 153, 158, 166, 170, 183

EIFS Extended Interframe Space. 36, 42, 43, 114, 143, 153, 158
EIRP Equivalent Isotropically Radiated Power. 44
EIVP European ITS VANET Protocol. 25, 57, 85, 88
ESP Electronic Stability Program. 1
ETC Electronic Toll Collection. 23
ETSI European Telecommunications Standards Institute. 20, 45

FC Frame Capture. 138
FCC Federal Communications Commission. 45
FCS Frame Check Sequence. 38, 40
FEC Forward Error Correction. 47, 50, 65
FIFO First In, First Out. 121, 139, 179, 181, 182, 185, 191
FLAME Forwarding Layer for Meshing. 85

GCDC Grand Cooperative Driving Challenge. 19, 79
GPRS General Packet Radio Services. 22
GPS Global Positioning System. 85, 97
GSM Global System for Mobile Communications. 22

HCCA HCF Controlled Channel Access. 29
HTAS High Tech Automotive Systems. 18

ICI Inter Carrier Interference. 47, 67
IDM Intelligent Driver Model. 140
IFS Interframe Space. 30–32, 35, 40, 42, 61, 126, 129, 142, 154, 158, 159
IR Infra Red. 29
IRSA Integrated full-Range Speed Assistant. 17
ISI Inter Symbol Interference. 47, 67
ISM Industrial, Scientific and Medical. 44
ISO International Organisation for Standardisation. 20
ITS Intelligent Transportation Systems. 10, 12, 20, 55, 173, 175, 180
IVC Inter-Vehicle Communication. 22

LAN Local Area Network. 28, 43, 55, 181, 209
LDM Local Dynamic Map. 56
LIFO Last In, First Out. 181, 182, 185, 191
LIN Local Interconnect Network. 21
LLC Logical Link Control. 28, 54
LoS Line-of-Sight. 102
LTE Long Term Evolution. 22

MAC Medium Access Control. 4, 7, 28, 29, 31, 59, 68, 69, 111, 113, 116, 117, 120, 138, 140, 144, 169, 179, 182, 191
MCCA MCF Controlled Channel Access. 29
MCF Mesh Coordination Function. 29
MIMO Multiple Input, Multiple Output. 29

NAV Network Allocation Vector. 33, 34, 37, 158
NPD Newest Packet Drop. 139, 182, 185, 186, 192

OBU On-Board Unit. 22, 84, 95
OFDM Orthogonal Frequency Division Multiplexing. 44–46, 153
OLSR Optimised Link State Routing. 55, 85
OPD Oldest Packet Drop. 59, 182, 185, 186, 193
OSI Open System Interconnection. 138

PBO post-backoff. 122
PCB Printed Circuit Board. 96
PCF Point Coordination Function. 29
PDU Protocol Data Unit. 26
PLCP Physical Layer Convergence Protocol. 37, 45
PMD Physical Medium Dependence. 29
PSDU Protocol Service Data Unit. 46

QAM Quadrature Amplitude Modulation. 46
QoS Quality of Service. 40, 41, 122
QPSK Quaternary Phase Shift Keying. 46

RED Random Early Detection. 181, 192
RF Radio Frequency. 29, 65
RFID Radio-Frequency Identification. 23
RSU Road-Side Unit. 22, 24, 40, 57, 84
RTS Request-To-Send. 34
RTS/CTS Request-To-Send Clear-To-Send. 34, 35, 113

SCH Service Channel. 24, 193, 194, 198
SIFS Short Interframe Space. 32, 36, 42
SNR Signal-to-noise Ratio. 50, 138, 140
SUMO Simulation of Urban Mobility. 99, 100, 140

TDMA Time-Division Multiple Access. 73
TPDU Transport Protocol Data Unit. 26
TXOP Transmission Opportunity. 42

UDP User Datagram Protocol. 85, 88
UMTS Universal Mobile Telecommunications System. 22
UTC Coordinated Universal Time. 24, 25, 57

V2I Vehicle-to-Infrastructure. 22, 28, 45
V2V Vehicle-to-Vehicle. 22, 28, 45, 57, 63, 85, 91
VCH Virtual Collision Handler. 41, 59, 116, 134, 139
VCS Virtual Carrier Sensing. 33, 34, 37
VoIP Voice over IP. 41, 112

WAVE Wireless Access in Vehicular Environments. 23
WLAN Wireless Local Area Network. 39–41, 45, 111–113, 116, 119, 122, 138, 160, 165, 180, 181
WSA WAVE Service Advertisement. 24, 118
WSM WAVE Short Message. 24
WSMP WAVE Short Message Protocol. 24

Index

About the author

Emiel Martijn van Eenennaam was born in Nijmegen, The Netherlands, in 1982. He has obtained the B.Sc. degree in Telematics in 2007 for research performed at the Architecture and Services of Networked Applications (ASNA) group at the University of Twente, The Netherlands. The topic of this research was context-aware service provisioning within the Freeband AWARENESS project [209]. More concretely, the goal was to develop a mobile proxy for communication with a Bluetooth-capable camera using the Basic Imaging Profile (BIP) and Object Exchange Profile (OBEX).

Shortly after, in 2008 the M.Sc. degree followed, also in Telematics from University of Twente. The focus of this project was on wireless multi-hop dissemination of traffic information on highways to enable congestion assistance systems. This work was performed at the Design and Analysis of Communication Systems (DACS) group. This work resulted in Martijn's first publication at IEEE V2VCOM 2008 in Eindhoven [16]. During the Master project Martijn also participated in the C,MM,N project [210], developing a driving simulator in which the future of driving could be experienced. This driving simulator was present at the AutoRAI in 2009, where over a thousand visitors were able to experience the future of driving. The setup subsequently toured various events in The Netherlands.

Later, from 2009 to 2013 the pursuit of the Ph.D. degree also took place within the DACS group at the University of Twente. The Ph.D. research went in parallel with the Connect&Drive project from 2009 to 2010, and in 2010-2011 with the Grand Cooperative Driving Challenge (GCDC). Besides the official publications, Martijn was also actively involved in promotional activities for the UT and the DACS research group, such as by making promotional videos or acting as invited speaker at various events. During this period, Martijn supervised several bachelor and master students.

Presently, Martijn works at Ovis Telematics, a high-tech company started by University of Twente alumni, where his professional interests in vehicles and communication continue. Martijn can be reached at `martijn@freeminded.org`.

List of publications in reverse-chronological order:

- Heijenk, G.J. and van Eenennaam, E.M. and Remke, A.K.I *Performance Comparison of IEEE 802.11 DCF and EDCA for Beaconing in Vehicular Networks*. IEEE Vehicular Networking Conference, VNC 2013, 16-18 Dec 2013, Boston, U.S.A. (submitted for publication).
- Klaassen, B. and van Eenennaam, E.M. and Schaake, L. and Buurke, J. *Gait feature detection based on markerless motion tracking*, 4th IEEE Dutch Bio-Medical Engineering Conference, 24-25 Jan 2013, Egmond aan Zee, The Netherlands.
- Klaassen, B. and van Eenennaam, E.M. and Schaake, L. and Buurke, J. *An automatic camera following system for clinical gait analysis*, 4th IEEE Dutch Bio-Medical Engineering Conference, 24-25 Jan 2013, Egmond aan Zee, The Netherlands.
- van Eenennaam, E.M. and Remke, A.K.I. and Heijenk, G.J. (2012) *An Analytical Model for Beaconing in VANETs*, IEEE Vehicular Networking Conference, VNC 2012, 14-16 Nov, Seoul, South Korea.
- van Eenennaam, E.M. and van de Venis, A. and Karagiannis, G (2012) *Impact of IEEE 1609.4 channel switching on the IEEE 802.11p beaconing performance*, IFIP Wireless Days 2012, 21-23 Nov, Dublin, Ireland.
- Lei, C. and van Eenennaam, E.M. and Klein Wolterink, W. and Ploeg, J. and Karagiannis, G. and Heijenk, G.J. (2012) *Evaluation of CACC String Stability using SUMO, Simulink, and OMNeT++*. EURASIP Journal on Wireless Communications and Networking, 2012. ISSN 1687-1499
- van Eenennaam, E.M. and Hendriks, L. and Karagiannis, G. and Heijenk, G.J. (2011) *Oldest Packet Drop (OPD): a Buffering Mechanism for Beaconing in IEEE 802.11p VANETs*. In: Third IEEE Vehicular Networking Conference, VNC 2011, 14-16 Nov 2011, Amsterdam, The Netherlands. IEEE Communications Society.
- Lei, C. and van Eenennaam, E.M. and Klein Wolterink, W. and Karagiannis, G. and Heijenk, G.J. and Ploeg, J. *Impact of Packet Loss on CACC String Stability Performance*. Proceedings of the Eleventh International Workshop on ITS Telecommunications 24 August 2011.
- van Eenennaam, E.M. and Heijenk, G.J. and Karagiannis, G. and van Arem, B. (2011) *Over-the-Horizon Awareness for Advanced Driver Assistance Systems: the TrafficFilter and microSlotted 1-Persistence Flooding*. Technical Report TR-CTIT-11-21, Centre for Telematics and Information Technology, University of Twente, Enschede. ISSN 1381-3625
- Reinders, R. and van Eenennaam, E.M. And Karagiannis, G. and Heijenk, G.J. *Contention Window Analysis for Beaconing in VANETs*. In: Seventh IEEE International Wireless Communications and Mobile Computing conference, IWCMC 2011, 5-8 Jul 2011, Istanbul, Turkey. pp. 1481-1487. IEEE Computer Society. ISBN 978-1-4244-9537-5

- Ghafoor, K.Z. and Bakar, K.A. and van Eenennaam, E.M., and Khokhar, R.H. and Gonzalez, A.J. (2011) *A Fuzzy Logic Approach to Beaconing for Vehicular Ad hoc Networks*. Telecommunication Systems Journal, Springer Verlag, DOI 10.1007/s11235-011-9466-8, ISSN 1018-4864.

- Schwartz, R.S. and van Eenennaam, E.M. and Karagiannis, G. and Heijenk, G.J. and Klein Wolterink, W. and Scholten, J. *Using V2V communication to create Over-the-horizon Awareness in multiple-lane highway scenarios* In: IEEE Intelligent Vehicles Symposium (IV) 2010, 21 June 2010

- van Wijngaarden, P and van Eenennaam, E.M. *Frame Capture in IEEE 802.11p Vehicular Networks*. In: Welcome to the Wireless World: Second Workshop on Pervasive Application of Wireless Technologies, University of Twente, 28 September 2010.

- van Eenennaam, E.M. and Karagiannis, G. and Heijenk, G.J (2010) *Towards Scalable Beaconing in VANETs*. In: Fourth ERCIM workshop on eMobility, Luleå, Sweden, 31 May 2010, pp. 103-108. Luleå University of Technology, Luleå, Sweden. ISBN 978-91-7439-103-9

- van Eenennaam, E.M. and Klein Wolterink, W. and Karagiannis, G. and Heijenk, G.J. (2009) *Exploring the Solution Space of Beaconing in VANETs*. In: First IEEE Vehicular Networking Conference, VNC2009, 28-30 Oct 2009, Tokyo, Japan. IEEE Communications Society. ISBN 978-1-4244-5687-1

- van Eenennaam, E.M. (2009) *microSlotted 1-Persistent Flooding in VANETs*, In: Welcome to the Wireless World: Workshop on Pervasive Application of Wireless Technologies, University of Twente, 13 November 2009
  Best paper award

- van Eenennaam, E.M. and Heijenk, G.J. (2008) *Providing over-the-horizon awareness to driver support systems*. In: The Fourth International Workshop on Vehicle-to-Vehicle Communications, V2VCOM 2008, 3 Jun 2008, Eindhoven, The Netherlands. pp. 19-25. University of Twente. ISBN 978-90-365-2693-7

Acknowledgments

Lectori salutem. I have never attended a single course in Latin, but one picks up fragments every now and then. Although officially only four years go into a Ph.D. project, a lifetime of experience culminated in the work before you, in some way or another. One collects bits and pieces over the years. *"Though this be madness, yet there is method in't"*, as Polonius in Shakespeare's Hamlet put it so fittingly.

In Dutch, a Ph.D. student is called an A.I.O. (Assistent in Opleiding). But with respect to the past few years, I think the U.S. Marine Corps' definition is more fitting: *Adapt, Improvise, and Overcome*. And there sure was a lot to adapt, to improvise, and to overcome.

Nearly five years went by in a blast and I acquired many skills. Did many interesting things, traveled the world and met interesting people. I would not have gotten this far without help, support, challenges and (occasionally) due resistance from many people. My teacher in elementary school, Meester Albert, thought it best I'd go do something practical (MAVO) despite achieving the highest score of the class in a national test (C¿to). Later, a few years into high school, a 'profession orientation test' concluded I should work in food preparation. What'd you expect from a 14-year old boy, whose world revolves to a great extend around the acquisition and consumption of food. Against certain "words of advise", I then pursued all the technical (and supposedly difficult) subjects in high school, and then went on to university to study Telematics.

I would like to thank my parents for raising me the way they did, instilling the values and mindset which were undoubtedly instrumental to getting where I am now, and not giving up halfway. Anything can be learned and achieved if you are willing to work hard. Also, a word of thanks goes out to the rest of my family. The commitment of pursuing a Ph.D. may sometimes cause a person to behave like a hermit, but it is always comforting to know you have a family you can fall back to, unconditionally. Suzanne, Justine, Marjolein jullie zijn geweldige zusjes!

Debbie I know you a few years by now, and you became a special person in my life during the past year. Although at times it seemed that my research and everything related to it was the only thing I could talk about, I am very glad you were there to listen and encourage me to keep going.

And then there's my second family: Stephan, Jochem, Jurriaan, Way Yee, Helena, and Gerwin—you put up with me during the time I was living on the UT Campus and somehow still are among my closest friends. Wat hebben we samen veel meegemaakt! Also, thanks goes out to Birgit for showing an excellent example, providing helpful support, being a great friend, and one of my paranymphs together with Jurriaan. Zonder jullie hulp zou ik waarschijnlijk te druk zijn geweest met stressen om nog iets gedaan te krijgen.

There seems to be a balance in my life between mental and physical labour. The more I work with my head, the more I need to work with my hands. Therefore

I am grateful for the many people who voluntered their house for demolition (or reconstruction) over the past years, allowing me to take my mind off of the research for a while. Words of thanks go out to Sander & Marieke ("The terrorists are behind that wall!") who never seemed to live in one place for too long, Stephan & Desiree, Jurriaan & Birgit, Bart's parents, Jochem & Marjan, Wieneke & Roelof, and probably some others I forgot. In some way a car fanatic, I survived university by building a Burton sportscar in the weekends, and survived a Ph.D. by restoring a Citroen 2CV.

The same gratefulness goes out to Ultimate Bart for being a buddy in the various Ultimate Sports activities and Epic Movie productions we have undertaken over the past years (867-5309) together with JakeOfOzz, Sophie, Almer, Nick, Marjolein, Anja, and various other people. Also, I would like to thank my gang of Thursday-nighters for the many epic adventures which were an excellent blend of fun and games, food and drink, fighting fierce damsels and saving many dragons in distress. On the same note, I want to thank by buddies from Sagittarius.

During highschool, I spent a year living in Canada, meeting great people while having a good time and picking up a useful skill or two. I am certain this experience has to a great extent contributed to the vast collection of English words (and their diversity) on these pages. I would like to use this opportunity to thank, among others, Chris (throughout this project I often wondered "What would Dolph do?"), Lindsey (I'm a doctor now, but not the kind that helps people), Shauna (Mom2), Grandma Doreen (Young Einstein finally got his Ph.D.), Serena, the Creightons, the Reimerts, the Butlers, and the Bradshaws.

In the past five years I have been working with various inspiring people in several interesting and challenging projects. I have worked on the C,MM,N project with some great people, doing a lot of fun stuff such as engineering, braggelen and maggelen, applying tjak and joegel and trucking the resulting driving simulator all over the Netherlands. Or, to put it in a more formal way, as part of a great team we were thinking out-of-the box about the future, towards sustainable and innovative mobility. The Cooperative Adaptive Cruise Control was a central part of the driving experience we implemented in that simulator, in which we counted over 1000 visitors during the AutoRAI 2009.

Also, big thanks to people whom I worked with in Connect&Drive. We had many fruitful discussions in a multi-disciplinary project—implementing a real-world CACC system. After a virtual implementation in C,MM,N, the real-world C&D system was like a dream come true. Lessons learned there trickle through this dissertation. And then there was an opportunity to work in a small, international team on a vehicle to participate in the Grand Cooperative Driving Challenge. As part of Team FUTURUM I worked together with great people from the UT, Fontys Automotive in Helmond, the Netherlands, and University of Waterloo, Canada. It will come as no surprise that CACC was a central theme in that project as well.

Although the A.I.O. may sometimes behave like a hermit, a Ph.D. is a journey on which you encounter many companions. I want to thank my students who, either directly through joint research or discussions, also contributed to this work. Ramon S. Schwartz, René Reinders, Pieter van Wijngaarden, Kristina Monova, Filip

Slanchev, Chenxi Lei, Mark te Brake, Peter Oostewechel, Fieke Hillerström, Pim Wijnands, Dennis Sluijsmans, Mark Wierbosch, Luuk Hendriks, Anne van der Venis, Bart Klaassen, and Martin Westra. I hope our collaboration has been as inspiring for you as it was for me.

Furthermore, I would like to thank colleagues and students working in the DACS Wireless Networking lab for the vast amounts of coffee which contributes to a large extent to the meaningful black blobs on this collection of paper—the covers of which are, according to some, too far apart. Not the brown liquid from the vending machines which does not deserve to be called coffee, but the brown gold refined using our own renegade coffee machine. As I did in my Bachelor's thesis back in 2005, I also want to thank my good friend Douwe Egberts for his continued support.

A warm thank-you goes out to my DACS friends. We had a lot of fun together and I learned a lot from our many intriguing rappa-room conversations. Wouter was a silent presence in our office who remarked witfully that "The only way out, is right through!" when it comes to finishing your dissertation, and how true this is. Sarwar joined the Wireless Networking lab, always smiling and cheerful (even when he found out black coffee can be quite bitter :). Thanks goes out to Silvia, and later Jeanette, for taking care of the many small "regeldingetjes" an absent-minded scientist tends to forget.

I also want to thank my "coaches" Boudewijn, Geert, and Georgios for meticulous proof-reading and helpful input during the entire process. Also, a word of thanks goes out to my committee members. I am delighted to have Bart van Arem as part of my committee. He was the promotor of Cornelie van Driel, whose dissertation was about driver support systems and traffic flow impacts, on which my Master research was based and which is also the context of this work. It is very fitting he is also involved in my Ph.D. research.

Parts of chapter 4 are the result of work done together with my students Kristina Monova and Chenxi Lei. Parts of chapter 6 resulted from work done together with René Reinders, Luuk Hendriks and Anne van der Venis. Also, I would like to express my gratitude to my colleagues Anne Remke and Daniël Reijsbergen for helpful input in chapter 5. I would also like to thank fellow researchers I met at conferences, some of whom I have had very interesting email correspondence with. Also, respect goes to the anonymous peer reviewers of my publications over the past years. Having done various reviews myself I can say this work is often *pro bono* and goes unmentioned.

Another word of thanks goes out to Enno Oosterhuis, Axel Belinfante and Stefan Blom for their support for using the Twickel cluster. Without access to a compute cluster, the large-scale simulation studies in this work would not have been possible.

I would also like to thank developers and supporters of Free and Open Source software around the world for their unselfish and continued effort in giving the world free—and often superior—tools for computing. I know that without open source software, this dissertation, and the entire research on which it is founded, would not have come to be. Research, by definition, is doing new things. Quite often, available software is not sufficiently capable of doing what a scientist needs to get done. Proprietary software is in many cases too restrictive, which is why open source software is embraced so widely by the research community. Personally, I am very grateful I did not write this document in MS Word.

The powerful scripting environment and commandline tools available on the GNU/Linux platform in particular have been fundamental to completion of the various arduous procedures involving many steps which, fortunately, could easily be scripted. These tools were supplied in the OpenSuSe, Ubuntu or Scientific Linux distributions put together by great people from all over the world.

Large parts of this work were edited using either Vim or Kile, the KDE Integrated LaTeXEnvironment. Speaking of LaTeX, this system is foundational to scientific publishing and it is in that sense under-appreciated in our Computer Science program at the University of Twente (or any of the programs, for that matter). Many figures in this dissertation are generated using PGF/TikZ, producing vector graphics from a geometric/algebraic description. These tools quickly become second nature to anyone familiar with thinking in code.

All simulation studies were performed using the Open Source simulation tool OMNeT++ and the MiXiM framework. Over the course of this research, various bugs were found and reported; and sometimes fixed within hours. For that I would like to thank the developers of OMNeT++ and MiXiM, their work has been quite vital.

Going on about free stuff available on the Internet... I would also like to thank Soma.fm, Slayradio.org and RadioParadise.com for their commercial-free, listener-supported internet radio they kept on streaming my way throughout the past years.

Though far apart, united by heart

Machinae Supremacy – Rise of a Digital Nation

– This page is intentionally left blank –

– This page is intentionally left blank –

– This page is intentionally left blank –

– This page is intentionally left blank –

(you can stop reading now)