Intelligent and Bandwidth-Efficient Medium Access Control Protocols for IEEE 802.11p-based Vehicular Ad Hoc Networks

Andreas Pressas

Submitted for the degree of Doctor of Philosophy

University of Sussex

September 2019

Declaration

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature:

Andreas Pressas

List of Acronyms

ACK Acknowledgement

AC Access Category

AIFS Arbitration Inter-Frame Spacing

BEB Binary Exponential Backoff

BSS Basic Service Set

C-V2X Cellular Vehicle-to-Anything

CACC Cooperative Adaptive Cruise Control

CAM Cooperative Awareness Message

CAN Controller Area Network

CCA Clear Channel Assessment

CCE Collective Contention Estimation

CCH Control Channel

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CW Contention Window

DCC Distributed Congestion Control

DCF Distributed Coordination Function

DENM Decentralised Environmental Notification Message

DIFS DCF Inter-Frame Space

DSRC Dedicated Short Range Communication

DoS Denial-of-Service

EDCA Enhanced Distributed Channel Access

FDMA Frequency-Division Multiple Access

GPRS General Packet Radio Service

GSM Global System for Mobile Communication

HS Hot Spot

IPv6 Internet Protocol Version 6

ITS Intelligent Transport Systems

k-NN k-Nearest Neighbors

L1 Layer 1 (PHY)

L2 Layer 2 (MAC)

LIN Local Interconnect Network

LTE Long-Term Evolution

MAC Medium Access Control

MDP Markov Decision Process

NIC Network Interface Controller/Card

OBU On-Board Unit

OCB Outside the Context of a BSS

OSI Open Systems Interconnection

PCA Principle Component Analysis

PCI Peripheral Component Interconnect

PDR Packet Delivery Ratio

PHY Physical Layer

PSID Provider Service ID

QoS Quality of Service

RL Reinforcement Learning

RSU Roadside Unit

RTT Round-Trip Time

RoI Region of Interest

SCH Service Channel

SIFS Short Interframe Space

SVM Support Vector Machine

TCP Transmission Control Protocol

TDMA Time-Division Multiple Access

TPC Transmit Power Control

TXOP Transmit Opportunity

UDP User Datagram Protocol

V2B Vehicle-to-Broadband Cloud

V2D Vehicle-to-Device

V2I Vehicle-to-Infrastructure

V2P Vehicle-to-Pedestrian

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Anything

VANETs Vehicular Ad Hoc Networks

WAVE Wireless Access in Vehicular Environments

WLANs Wireless Local Area Networks

WSMP WAVE Short Message Protocol

WSN Wireless Sensor Network

Abstract

Vehicle-to-Vehicle (V2V) technology aims to enable safer and more sophisticated transportation via the spontaneous formation of Vehicular Ad hoc Networks (VANETs). This type of wireless networks allows the exchange of kinematic and other data among vehicles, for the primary purpose of safer and more efficient driving, as well as efficient traffic management and other third-party services. Their infrastructure-less, unbounded nature allows the formation of dense networks that present a channel sharing issue, which is harder to tackle than in conventional WLANs.

This thesis focuses on optimising channel access strategies, which is important for the efficient usage of the available wireless bandwidth and the successful deployment of VANETs. To start with, the default channel access control method for V2V is evaluated hardware via modifying the appropriate wireless interface Linux driver to enable finer on-the-fly control of IEEE 802.11p access control layer parameters. More complex channel sharing scenarios are evaluated via simulations and findings on the behaviour of the access control mechanism are presented. A complete channel sharing efficiency assessment is conducted, including throughput, fairness and latency measurements. A new IEEE 802.11p-compatible Q-Learning-based access control approach that improves upon the studied protocol is presented. The stations feature algorithms that "learn" how to act optimally in VANETs in order to maximise their achieved packet delivery and minimise bandwidth wastage. The feasibility of Q-Learning to be used as the base of self-learning protocols for IEEE 802.11p-based V2V communication access control in dense environments is investigated in terms of parameter tuning, necessary time of exploration, achieving latency requirements, scaling, multi-hop and accommodation of simultaneous applications. Additionally, the novel Collection Contention Estimation (CCE) mechanism for Q-Learning-based access control is presented. By embedding it on the Q-Learning agents, faster convergence, higher throughput, better service separation and short-term fairness are achieved in simulated network deployments.

The acquired new insights on the network performance of the proposed algorithms can provide precise guidelines for efficient designs of practical, reliable, fair and ultra-low latency V2V communication systems for dense topologies. These results can potentially have an impact across a range of related areas, including various types of wireless networks and resource allocation for these, network protocol and transceiver design as well as Q-Learning applicability and considerations for correct use.

Acknowledgements

Foremost, I would like to express my gratitude to my advisor Dr. Zhengguo Sheng for his patience, continuous support and guidance during my Ph.D, as well as teaching me how to conduct research in a way that matters and can be propagated to the world. His creativity and attention to detail have been greatly inspiring to me. I would also like to express my gratitude to my second advisor, Dr. Falah Ali, for his support and encouragement in conducting research in a field that is very interesting to me, as well as conveying his vast experience in the area. His curiosity and ability to always ask the right questions have shaped the direction of my research. I also want to thank my family and friends, Stavros, Konstantina, Evangelia, Dimitris, Georgia to name some, for their understanding and continuous support throughout my journey in academia. They have been a constant source of confidence and encouragement for me. Finally, I would like to thank my fellow Ph.D students and friends for the interesting exchanges, which always helped in keeping things in perspective.

Contents

List of Tables

List of Figures

List of Publications

1. **Andreas Pressas**, Zhengguo Sheng, Falah H. Ali, Daxin Tian. "A Q-Learning Approach with Collective Contention Estimation for Bandwidth-efficient and Fair Access Control in IEEE 802.11p Vehicular Networks". IEEE Transactions in Vehicular Technology (IEEE TVT), Volume: 68, Issue: 9, Pages: 9136 - 9150, Sept. 2019

2. **Andreas Pressas**, Zhengguo Sheng, Falah H. Ali. "Applications of Machine Learning in Wireless Communications", IET, invited book chapter, ISBN: 9781785616570, 2019

3. Zhengguo Sheng, **Andreas Pressas**, Victor Ocheri, Falah H. Ali, Richard Rudd and Maziar Nekovee. "Intelligent 5G Vehicular Networks: An Integration of DSRC and mmWave Communications". In Proc. of International Conference on Information and Communication Technology Convergence (ICTC) - Emerging Solutions for 5G and Beyond 5G Workshop 2018, Korea

4. **Andreas Pressas**, Zhengguo Sheng, Falah H. Ali, Daxin Tian, Maziar Nekovee. "Contention-based learning MAC protocol for broadcast vehicle-to-vehicle communication". In Proc. of IEEE Vehicular Networking Conference (VNC) 2017, Italy

5. **Andreas Pressas**, Mümin Özpolat, Zhengguo Shen, Falah H. Ali. "Performance evaluation of networking protocols for connected vehicles". International Journal of Vehicular Telematics and Infotainment Systems (IJVTIS), 1 (1). pp. 1-14.

6. **Andreas Pressas**, Zhengguo Sheng, Peter Fussey, David Lund. "Connected vehicles in smart cities: interworking from inside vehicles to outside". In Proc. of IEEE International Conference on Sensing, Communication and Networking (SECON), Poster/Demo session, 2016, June 28, London (received Demo Paper Award)

Chapter 1

Introduction

1.1 Overview

Vehicle-to-Vehicle (V2V) technology aims to enable safer and more sophisticated transportation starting with minor, inexpensive additions of communication equipment on conventional vehicles and moving towards network-assisted fully autonomous driving. It is a fundamental component of the Intelligent Transportation Systems (ITS) [8] and Internet of Things (IoT) [109]. This technology allows for the formation of Vehicular Ad Hoc Networks (VANETs), a new type of network which allows the exchange of kinematic data among vehicles, for the primary purpose of safer and more efficient driving, as well as efficient traffic management and other third-party services. VANETs can help minimise road accidents and randomness in driving with on-time alerts, enable (semi)-autonomous vehicle applications, as well as enhance the whole travelling experience with new infotainment systems which allow acquiring navigation maps and other information from peers.

V2V communications enable the wireless ad hoc networking of moving vehicles within a Region of Interest (RoI), for safety message exchanges and other purposes. The key enabling technology, specifying the physical (PHY) and medium access control (MAC) layers of the V2V stack is IEEE 802.11p, which enables communications Outside the Context of a Basic service set (OCB) via the Dedicated Short Range Communication (DSRC) frequencies at 5.9 GHz. With DSRC specifying a 1-hop range of up to 1 km Line-of-Sight (LoS), wireless vehicular networks will have to accommodate many transmitting vehicle-stations within the range of each other. Additionally, with the Internet of Vehicles proposing an ever increasing amount of promising applications, novel protocols are needed to meet challenging demands not addressed by the conventional standard, since IEEE 802.11p belongs in the IEEE 802.11 family of protocols originally designed to be used in Wireless Local

Area Networks (WLANs). The DSRC PHY and MAC must be scalable and it is expected that the stack often will have to manage 50-100 interconnected stations in an immediate communication zone.

A MAC protocol defines the rules of how multiple network stations access the shared channel to avoid packet collisions. The de-facto MAC layer used in IEEE 802.11p-based networks is implemented as a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm, which is a distributed, contention-based protocol. For VANETS, CSMA is preferable to centralised solutions such as Time-Division Multiple Access (TDMA) or Frequency-Division Multiple Access (FDMA) [78] [106], since these would require synchronisation among stations which is very difficult to achieve in such mobile, infrastructure-less networks. But there is still space for improvement, especially when it comes to wireless vehicular networks which are unbounded, ad hoc networks with long one-hop transmission range, that allows them to become quite dense and congested in urban environments, leading to packet collisions. Every vehicle must maintain a relative standard of transmission rate or else the rest of the vehicles in near proximity would not be aware of its existence. A vehicle-station's packets colliding and being dropped effectively means that it is disconnected from the wireless vehicular network for the period of time that these packets are dropped, which may pose safety concerns. Furthermore, with the majority of radio stations that form VANETs being moving vehicles, the latency requirements of some ITS applications can be very strict.

As a solution, this thesis studies a novel self-learning channel sharing control method that can be biased towards satisfying various V2V applications, for both unicast and broadcast V2V exchanges via DSRC links. It allows to directly interconnect a large number of vehicles and stationary units via IEEE 802.11p wireless interfaces, by employing a Reinforcement Learning (RL) algorithm to perform CW adaptation. This technique allows the designers to improve networking performance via self-learning channel access controllers, without having to make major modifications to existing hardware. Moreover, the real-time learning and control requirements of the algorithm are considered.

1.2 Challenges and Research Questions

1.2.1 Motivation

A MAC protocol is part of the data link layer (L2) of the OSI model and defines the rules of how the various network stations share access to the channel. The IEEE 802.11p stack for V2V employs the CSMA/CA MAC, the decentralized contention-based access control algorithm which has been extensively tested in WLANs and Wireless Sensor Networks (WSNs). The primarily one-to-many nature of transmissions for VANETs presents some problems for the IEEE 802.11-inherited MAC layer which is not designed to accommodate dense broadcast traffic. Additionally, due to the safety nature of the packets exchanged via DSRC and their short temporal validity, the Contention Window (CW) parameter defined by CSMA/CA for the purpose of randomising the time of access to the channel among the various stations to avoid collisions, is kept small according to the default IEEE 802.11p specification. While keeping the CW value small lowers the end-to-end latency of transmissions, studies [99] [30] have shown that this is a primary cause of packet collisions in DSRC-based networks, which cannot be eliminated by the IEEE 802.11p MAC as it is due to its inability to adapt the parameter. Additionally, the IEEE 802.11-based MAC presents an intrinsic (short-term) fairness problem whereby stations cannot gain access to the wireless medium with equal probability under heavy traffic conditions [98], which can often be the case in wireless vehicular networks. This could impair the reliability of applications such as information collection from vehicular sensors, safety-related real time traffic, and TCP applications. Furthermore, the fairness problem may seriously affect the quality of service (QoS) support for DSRC-based networks, meaning that the desirable QoS for some uses may not be satisfied due to unfair access opportunities. The CW L2 parameter is definitive to the network performance and its correct adaptation could largely improve the performance of suggested applications.

The data to be exchanged via VANETs can divided into technical (i.e., vehicular, proximity sensors, radars), crowd-sourced (i.e., maps, environment,traffic,parking) and personal (i.e., VoIP, Internet radio, routes) applications. We believe that a significant part of this data will be exchanged through V2V or Vehicle-to-Everything (V2X) links, making system scalability a critical issue to address. There is a need for an efficient MAC protocol for V2V communication purposes, that adapts to the density of vehicular traffic and types of traffic (data rates etc.), since network conditions and topology are not known a-priori. Applications for VANETs vary a lot, as do their communication requirements. Pre-crash sensing or (semi) autonomous applications such as Cooperative Adaptive Cruise Control (CACC) [70] rely on ultra-low latency exchanges (< 20 ms) for warnings or directly driving vehicle control systems. Others are oriented towards more assistive, road safety and traffic efficiency uses such as lane-changing and emergency braking, with strict but more easily met latency requirements (< 100 ms) [108]. Finally there are also convenience and infotainment uses where delay is not as critical in comparison but the transferred data volume can be much larger.

1.2.2 Aims and Objectives

The studied problem is the sharing of wireless DSRC frequencies for dense urban, highway or smart-city scenarios where many vehicles would need to communicate with each other as well as other connected elements of the environment such as Road-side Units. The higher-level objective of this work is to develop a DSRC-compatible MAC layer capable of self-improving over time, that can meet key requirements for various VANET applications, such as reliability and bandwidth efficiency and low latency as well as enhancing short-term fairness and handling of service separation. The objectives of this thesis are presented in detail below.

- Gaining a better understanding of the IEEE 802.11p MAC contention resolution capability in various VANET scenarios featuring multiple transmitters. The protocol's performance would have to be evaluated in a real hardware implementation, as well as computer simulations that allow modelling of dense networks, various application layers or multi-hop.
- Investigation of the applicability of Reinforcement Learning in the distributed access control problem in VANETs. The outcome of this would be a novel adaptive MAC layer protocol based on RL agents, tailored for wireless vehicular networks. The protocol would have to the ability to "learn" how to optimise the MAC layer performance in various networking scenarios. The protocol's performance would have to be evaluated against a set of networking metrics. It would have to support broadcast (OCB) mode which is the primary mode of operation for V2X.
- Design of detailed reward functions that allow the RL agents to strive towards achieving multiple MAC layer goals simultaneously. A MAC protocol for VANETs has to achieve high packet delivery, low latencies and fairness of bandwidth allocation among vehicles. The reward function would make RL agents "aware" of these objectives, and make them take actions to reach them.

1.3 Contributions

We design and present an IEEE 802.11p-compliant MAC algorithm based on Q-Learning. It simultaneously targets reliable packet delivery and throughput-fairness, while being latency-aware. It features the proposed CCE reward method for Q-Learning, designed to tackle the inherent fairness problem appearing in CSMA-based IEEE 802.11p networks, to achieve more efficient channel sharing in terms of providing (near) equal transmission opportunities and improved transmission reliability for all stations. A summary of contributions is listed as follows:

- Chapter 3: The investigation of the default DSRC MAC Layer and its contention resolution capability is the first contribution of this work. More specifically, we examine the effect of the Contention Window parameter on communications under heavy channel contention, with a multitude of vehicles attempting to exchange data packets with each other. The DSRC protocol's capability in channel sharing is tested in a real-world implementation based on commodity hardware. The hardware platform used to evaluate the DSRC MAC provides access to the Link Layer software so that the studied parameters can manually set. This chapter presents a unique study of the effect that the DSRC CW parameter has on communication performance, based on real hardware test-bed built of of-the-shelf components.

  Furthermore, a more extensive, simulation-based study of the effect of the parameter in denser VANETs was conducted in OMNeT++. A MAC protocol evaluation framework around packet delivery ratio, latency and fairness is presented. The effect of different IEEE 802.11p-compatible values of the parameter are examined, applied symmetrically or asymmetrically to the network, to promote either fairness and overall network-wide reliability or favour high-priority stations respectively.

- Chapter 4: After investigating the capability of the baseline DSRC protocol in handling communications in congested VANETs, improvements based on Machine (Reinforcement) Learning are employed to increase the available bandwidth utilisation and achieve more efficient communication, in terms of achieved data transfer rates as well as transmission latency. A novel Q-Learning-based MAC protocol for both unicast and broadcast DSRC systems, featuring CW adaptation, is presented. The protocol is effective when transmitting in classic, unicast mode, as well as broadcast (OCB) mode which is the default for V2X. It also supports both single-hop and multi-hop information dissemination via retransmissions. Real-time effectiveness and learning performance is considered and evaluated.

- Chapter 5: Given the criticality of packet exchanges among vehicles in DSRC networks, an investigation of fair sharing of the bandwidth among multiple nodes is also of concern. A new Q-Learning-based MAC protocol is developed that features a Collective Contention Estimation (CCE) algorithm, for enhanced fairness and throughput. The capability of the Q-Learning-based MAC and the CCE algorithm to accommodate different classes of data going through the network simultaneously is also examined. Additionally, a way of combining multiple objectives that the Q-Learning can strive towards is proposed and evaluated.

1.4 Structure of thesis

The remainder of the thesis is outlined as follows: Chapter 2 reviews the existing technical background on V2X communications and the Link Layer technology, as well as related research work in the problems and adaptive MAC solutions. Chapter 3 reviews the IEEE 802.11p MAC protocol for broadcast communication. Chapter 4 presents the development of a modification of the IEEE 802.11p MAC based on Reinforcement Learning, as well as evaluation under various network conditions. An enhanced version of the RL-based MAC protocol based on a novel Collective Contention Estimation reward function is developed in Chapter 5. Finally in Chapter 6, we conclude our findings and suggest topics for future research on the matter.

Chapter 2

Background

2.1 Introduction to Vehicular Networking

The Intelligent Transportation Systems (ITS) network architecture comprises of three domains: the in-vehicle, the ad hoc and the infrastructure domain, as seen in [51]. The in-vehicle domain is composed of an on-board communication unit (OBU) and multiple control units. The connections between them are usually wired, utilising protocols such as Controller Area Network (CAN), Local Interconnect Network (LIN) or Ethernet etc. and sometimes wireless. The ad hoc domain is composed of vehicles equipped with such OBUs and roadside units (RSUs) [58]. The OBUs can be seen as the mobile nodes of a wireless ad hoc network and likewise RSUs are static nodes. Additionally, RSUs can be connected to existing infrastructure and the Internet via gateways, as well as communicate with each other directly or via multi-hop. There are two types of infrastructure domain access, RSUs and hot spots (HSs). These provide OBUs access to the Internet. In the absence of RSUs and HSs, OBUs could also use cellular radio networks (GSM,GPRS,LTE) [58] for the same purpose. The various networking domains and their respective components can be seen in Fig 2.1.

2.1.1 Types of Communication

In-vehicle communication refers to a car's various electronic controllers communicating within the in-vehicle domain. The in-vehicle communication system can detect the vehicle's performance regarding the internal systems (electrical and mechanical) as well as driver's fatigue and drowsiness [5], which is critical for driver and public safety. In the ad hoc domain, Vehicle-to-Vehicle (V2V) communication can provide a data exchange platform for the drivers to share information and warning messages, so as to expand driver assistance

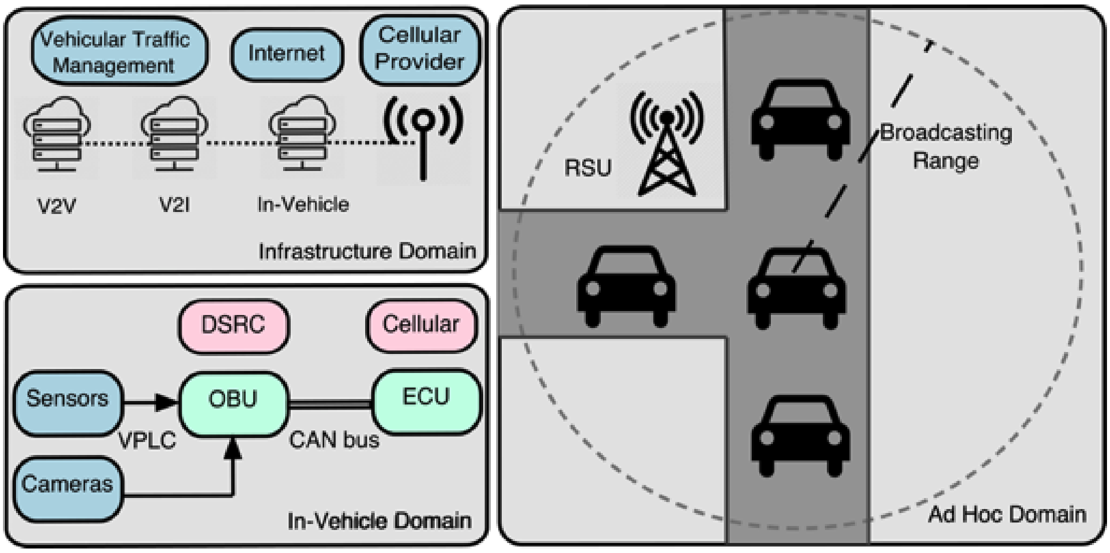


Figure 2.1: Networking Domains for VANETs

and prevent road accidents. Vehicle-to-road Infrastructure (V2I) communication enabled by VANETs allows real-time traffic updates for drivers, a sophisticated and efficient traffic light system as well, as could provide environmental sensing and monitoring. V2I can extend to Vehicle-to-Broadband Cloud (V2B) communication means that vehicles may communicate via wireless broadband mechanisms such as 4G/5G (infrastructure domain). As the broadband cloud includes more traffic information and monitoring data as well as infotainment, this type of communication will be useful for active driver assistance and vehicle tracking as well as other infotainment services [33]. The V2V and V2I communication types can be collectively referred to as Vehicle-to-Everything (V2X). More types of V2X communication are examined by researchers, e.g., Vehicle-to-Pedestrian (V2P) and Vehicle-to-Device (V2D) intended more for connectivity among cars and bicycles as well as other devices.

2.1.2 V2X communication technologies

What has been the design under debate in all parts of the world is the Ad Hoc Domain, or V2X communication. There are two dominant types of V2X communication technology depending on the underlying technology being used: WLAN-based, and cellular-based. Various proposals on the V2V radio technology around the world, such as the EU Commission's Delegated Act on ITS [29] define a hybrid approach, endorsing the ITS-G5 standard, also known as IEEE 802.11p or DSRC, as the baseline technology for direct V2V communication since it is a mature, well-researched physical layer solution. Complementary 4G or 5G Cellular V2X (C-V2X) technology can be used for longer range communication to infrastructure and cloud services.

Work in [37] shows that IEEE 802.11p exhibits lower latency and higher delivery ratio than LTE in scenarios fewer than 50 vehicles. More specifically, for smaller network densities, the standard allows end-to-end delays less than 100 ms and throughput of 10 kbps which satisfies the requirements set by active road safety applications and few of the lightweight cooperative traffic awareness applications. However, as the number of vehicles increases, the standard is unable to accommodate the increased network traffic and support performance requirements for more demanding applications.

A white paper from NXP Semiconductors [93] finds IEEE 802.11p (ITS-G5 in Europe) to be the superior technology to allow deploying V2X communications right now. The comparison done by NXP reveals the much better coverage by ITS-G5, using the state-of-the-art modems, with minimal packet loss recorded at 400 meters of range (>93% of transmission success), with LTE-V2V achieving just 40% for the same range. The theoretical specification for IEEE 802.11p PHY defines upper TX power enough to reach a theoretical LoS range of 1 km. These ranges also indicate that V2V networks can become particularly dense i.e., in urban or highway scenarios, with multiple vehicular or road-side transmitters exchanging kinematic, traffic and other information.

There has been newly proposed C-V2X technology, based on the Release 14 of the LTE standard, which includes two modes for V2V communications: Mode 3 (base-station-scheduled) and Mode 4 (autonomously-scheduled). C-V2X Mode 3 is not comparable with IEEE 802.11p since it does not support the formation of ad hoc (infrastructure-less) networks, rather makes use of cellular infrastructure support for resource allocation. This means that only C-V2X Mode 4 is relevant since V2V safety applications cannot depend on the availability of infrastructure-based cellular coverage. It is specifically designed for V2X communications (using the PC5 sidelink interface) and allows vehicle-stations to autonomously select and manage their radio resources.

The IEEE 802.11p standard is a mature V2X technology which is suitable for deployment right now having been tested in field trials. The same is not true for C-V2X Mode 4, which is new technology with modems not yet widely available, and there has been limited, mostly analytical and simulation-based research into its performance and suitability for V2X use. Also, different studies have proved possible inefficiencies of the autonomous resource scheduling of C-V2X Mode 4 when the transmissions are not periodic [36]. Work in [96] shows that IEEE 802.11p outperforms C-V2X Mode 4 in terms of packet delivery when tested in a platooning (CACC) scenario.

2.2 Dedicated Short Range Communication (DSRC)

The primary functionality that VANETs will contribute towards the Smart City environment is advanced active road safety. A vehicular safety communication network is ad hoc, highly mobile with a large number of contending nodes. The safety messages are very short as it is their useful lifetime-relevance, and must be received with high probability [106]. The key enabling technology, specifying Layer 1 and 2 of the protocol stack used in V2X (ad hoc domain), is Dedicated Short Range Communication (DSRC). The DSRC radio technology is essentially IEEE 802.11a adjusted for low overhead operations in the DSRC spectrum (30 MHz in the 5.9 GHz band for Europe). It is being standardized as IEEE 802.11p [2].

2.2.1 IEEE 802.11p

In the architecture of classic IEEE 802.11 networks, there are three basic modes of operation:

- A Basic Service Set (BSS), which includes an access point (AP) node that behaves as the controller/master station (STA).
- The (Independent Basic Service Set) IBSS, which is formed by STAs without infrastructure (AP/s). Networks formed like this are called ad-hoc networks.
- The (Extended Service Set), which is the union of two or more BSSs connected by a distribution system [111].

The most suitable architecture for a VANET would be the IBSS. A STA (node) within an IBSS acts as the AP and periodically broadcasts the Service Set ID (SSID) and other information. The rest of the nodes receive these packets and synchronize their time and frequency accordingly. Communication can only be established as long as the STAs belong in the same Service Set (SS).

The IEEE 802.11p amendment defines a mode called “Outside the context of BSS" (OCB) in its Medium Access Layer, that enables exchanging data without the need for the node to belong in a Service Set (BSS), and thus, without the overhead required for these association and authentication procedures with an Access Point before exchanging data.

DSRC defines 7 licenced channels, each of 10 MHz bandwidth (as seen in Fig. 2.2): 6 service channels (SCH) and 1 control channel (CCH). All safety messages, whether transmitted by vehicles or RSUs, are to be sent in the control channel, which has to be regularly

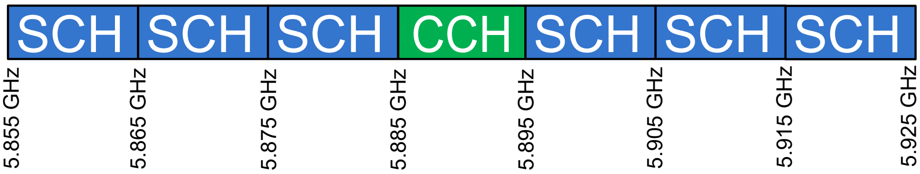


Figure 2.2: The channels available for 802.11p

monitored by all vehicles. The control channel could be also used by RSUs to inform approaching vehicles of their services, then use the service channel to exchange data with interested vehicles.

The explicit multi-channel nature of DSRC necessitates a concurrent multi-channel operational scheme for safety and non-safety applications [43]. This need is facilitated with a MAC protocol extension by the IEEE 1609 working group, which deals with the standardization of the DSRC communication stack between the link layer and applications. The IEEE 802.11p and IEEE 1609.x protocols combined are called Wireless Access in Vehicular Environments (WAVE) since they aim to enable wireless communication between vehicles. The entire protocol stack based on DSRC L1 and L2, suggested for V2X use is depicted in Fig. 2.3.

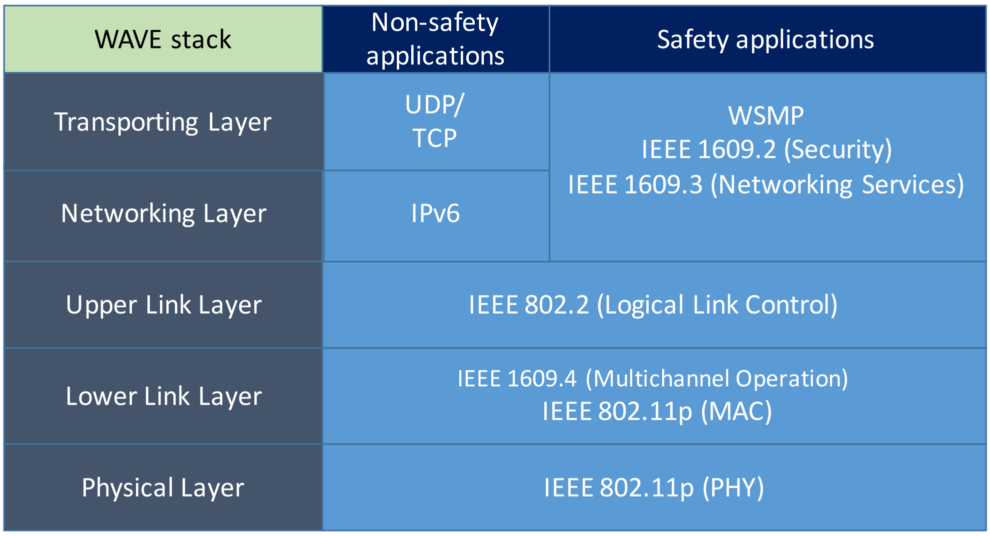


Figure 2.3: The DSRC/WAVE Protocol Stack and Associated Standards

There has been work on why the multi-channel operation of DSRC in the way it is currently designed poses some issues, with the most significant being bandwidth wastage [27]. This is because the time frame that a car with a single antenna system would have to be tuned to each channel is fixed (50 ms), no matter if the actual data exchange can be completed in less time than that. This also sets a hard latency requirement of 50 ms, meaning that a vehicle should get its information across within this time since this is the period of a station scanning the CCH. For these reasons, this work focuses on optimising single-channel operation, but can be used as-is to resolve contention in multiple channels if the IEEE 1609.4 switching layer is in operation.

2.2.2 Wave Short Message Protocol

There are two stacks supported by WAVE, one being the classic Internet Protocol version 6 (IPv6), and another one being WAVE Short Message Protocol (WSMP). The reason for having two variations in the upper layers is to distinguish the messages as high-priority/time sensitive or more latency-tolerant and feature-rich packet transmissions such as UDP transactions.

These are intended for applications such as collision avoidance, that do not necessarily require very large datagram lengths or complex packets to be transmitted, rather than high probability of reception and low latency. The format of a WSMP packet is depicted in Fig. 2.4. The overhead is 11 bytes, when a typical UDP-IPv6 packet has a minimum overhead of 52 bytes [56]. WSMP enables sending short messages while directly manipulating the Physical Layer Characteristics such as the transmission power and data rate so that nearby vehicles have a high probability of reception within a set time frame. A Provider Service ID (PSID) field is similar to a port number in TCP/UDP, which acts as an identity and answers which application is a specific WSMP heading towards. To improve the latency, a WSMP exchange does not necessitate the formation of a BSS, which is a requirement for Service Channel exchanges.

WSM version 1 byte	Security Type 1 byte	Channel Number 1 byte	Data Rate 1 byte	TX Power 1 byte	PSID 4 bytes	Length 2 bytes	DATA Variable

Figure 2.4: The Format of a WSMP packet

Apart from safety message exchange, connected cars can provide extra functionality and enable driving assisting and infotainment systems, such as downloading city map content from RSUs, exchanging video for extended driver vision or even uploading traffic information to the cloud towards an efficient traffic light system. However, WSMP is not able to support these, or the classic Internet applications or exchange of multimedia, and it does not need to since such applications are more tolerant to delay or fluctuations in network performance. By supporting the IPv6 stack, which is also open and already widely deployed, such services are easily and inexpensively deployable in a vehicular environment.

2.2.3 Message Types

Two types of WSMP messages are sent through the control channel by every DSRC-enabled vehicle:

Periodic safety messages: These are broadcast status messages (beacons) containing information of the dynamics such as direction, velocity etc. of the transmitting vehicle. These messages are meaningful for a short period of time, so that the receivers can be approximately aware of the movement of the transmitter, and quickly become irrelevant. RSUs also utilize these beacons for traffic light status etc. The beaconing interval is usually 100 ms or less ($F_{beacon} > 10\,\text{Hz}$). These packets are also referred to as Cooperative Awareness Messages (CAMs).

Event-triggered messages: Changes in the vehicle dynamics (hard breaking) or RSU status activate the broadcasting of emergency messages with safety information (i.e., road accident warning, unexpected breaking ahead, slippery road). These packets are also referred to as Decentralized Environmental Notification Messages (DENMs).

There are also **non-safety communications**, that can happen for file transfers (local map updates, infotainment) or transactions (toll collection) and others. These can take place in the service channels but are advertised through WSA messages in the control channel, in which every DSRC-enabled vehicle is tuned in by default.

2.2.4 Summary of relevant DSRC communication properties

The control channel is the one to facilitate safety communications through the exchange of safety-related or CACC packets. Many of the safety applications are based on single hop communication since they are very localized. The basic DSRC communication design proposals do not feature any networking (packet routing) capabilities. Although safety communications are often single-hop, the system is unbounded and supports 1 km range in LoS, which means that V2V communication can stretch to great distances, unlike a bounded system (cells in mobile telephony) [43].

There are, though, scenarios where the message needs to be disseminated to the vehicles beyond the immediate transmission range of a sender. In these cases, multi-hop communication is used. Such cases can again be safety related, like on-time warnings for an accident/hazard along a highway), information about the traffic in an extended area or other geo-significant information. The work in [17] focuses on safety-related applications via multi-hop communication, thus disseminating information such as warning messages (e.g., accident, blocked street, traffic congestion etc.) to a greater RoI. Additionally, [54] presents various routing techniques and considerations for large-scale VANETs. Rebroadcasting schemes for enhancing the multi-hop performance or reliability are seen in [107]. Safety applications made possible through VANETs require a low end-to-end delay and high packet delivery probability. Additionally, since the exchanged information should be shared with all vehicles in an RoI, VANETs will be the first large-scale networks where communication is based mainly on broadcast rather than on unicast transmissions, which means that it is targeted at vehicles depending on where they are (within a relevant RoI) rather than some form of identification. The choice of an IEEE 802.11 based technology for this kind of network raises some issues [88]. The MAC protocol in this family of standards is well known for its inability to cope with large scale broadcast communications, since it was designed for a different use-case and it clearly favours unicast [71] communication.

Furthermore, channel access is not centrally managed in DSRC, since vehicular communication networks should be spontaneously formed without the need for infrastructure support, which translates to a fully distributed architecture. A major concern for DSRC is that since all DSRC-enabled vehicles and infrastructure continuously broadcast beacon messages as well as event-triggered safety messages, such a system would require special design so that it can work reliably and efficiently in a large scale. Originally the CCH was proposed to facilitate the exchange of safety messages, complying with the WSMP. Occasionally, it would be used for advertising non-safety applications (by RSUs) which take place in one of the service channels. These are called WAVE Service Advertisement (WSA) messages. The receiving node would get informed of the existence of such applications, and tune in the appropriate channel if it needs to make use of these. These advertisements are generally lightweight and their effect to the control channel's load is insignificant [43]. But more advanced applications have been suggested that could benefit from the low latencies CCH would provide, i.e., (semi) autonomous driving applications like CACC.

Consequently, the focus of the CCH performance characteristics initially was towards low latency and high delivery probability rather than high transfer rates. Nevertheless, given that the IEEE 1609.4 multi-channel system is not being actively implemented and used, and given the performance problems that can arise from it, DSRC networks as of now would operate at a single frequency at a time, which means that the CCH would have to support higher amounts of traffic if possible. The lower MAC layer would essentially remain unchanged in all cases (single-channel or multi-channel) and must have the capability to adapt and accommodate all different types of traffic irrespectively of the channel or use.

2.3 The IEEE 802.11p Medium Access Control

The MAC protocol is responsible for transferring data reliably when there is more than one station attempting to access the same channel simultaneously. An efficient MAC will strive for maximum channel utilization with minimum collisions. The Distributed Coordination Function (DCF) is the fundamental MAC technique of the IEEE 802.11-based standards. DCF employs the CSMA/CA algorithm for sharing access to the common medium among multiple peers in a distributed manner.

Optimising the CSMA/CA MAC layer essentially translates to appropriate tuning of four related parameters, namely: minimum and maximum Contention Window (CW_{min} and CW_{max}), the Arbitrary Interframe Spacing ($AIFS$) and lengths of packet bursts or transmission opportunity limit ($TXOP$ limit). A fifth parameter representing the $backoff$ window multiplier was studied during the standardization process, but was eventually abandoned due to doubts about its effectiveness and replaced with a fixed multiplier of 2 [105].

2.3.1 The CSMA/CA algorithm

We start with the basic principle of the medium access operation for networks based on the IEEE 802.11 family of protocols, which works as follows:

- Once a packet is ready for transmission, the station is required to sense the state of the wireless medium before transmitting (listen before talk) to determine whether another station is transmitting or not. This is done by performing a Clear Channel Assessment (CCA) (listening for transmissions at the PHY) which includes comparing against some threshold to determine whether the channel is idle while accounting for noise.

- If the station finds that the medium is continuously idle for a DCF Interframe Space ($DIFS$) period (or a variable $AIFS$ period for separating different classes of data), the station is given permission to transmit after it goes through an additional time period called $backoff$, defined by the CW parameter. The purpose of the $backoff$ is to introduce some asynchronisation which helps the case in which two station's DIFS expire simultaneously. When the $backoff$ counter reaches 0, the packet is transmitted immediately.

- If the channel turns busy before the DIFS interval expiration, the station again defers from transmission until the medium is again idle for the duration of a DIFS interval.

- When a unicast packet has been received correctly, the destination station waits for a Short Interframe Space ($SIFS$) interval to give priority to an ACK packet transmission, sent back to the source node to indicate successful reception.

- Transmit Opportunity or $TXOP$ is the amount of time a station can send frames when it has won contention for the wireless medium. A $TXOP$ of 0 means that the station can only send 1 packet during the $TXOP$ period. If a station with $TXOP > 0$ obtains the channel, it will be permitted to transmit a sequence of data packets in the time duration defined by the parameter. Once the packet/s are sent, it must contend for the wireless medium again with the CCA, $AIFS$ and CW.

- If an ACK is not received by the source station in due time, the transmission is considered as failed and a retransmission of the packet is arranged (unless the maximum number of retransmissions has been reached). The CW value is set by the Binary Exponential Backoff algorithm prior to every retransmission.

2.3.2 Binary Exponential Backoff Mechanism

The range of the generated random $backoff$ timer is bounded by the Contention Window (CW). More especially, the node randomly draws an integer $backoff$ from the uniform distribution over the interval $[0, CW]$, where the initial CW value equals CW_{min}, and counts down for $backoff$ time slot intervals before attempting to transmit. The $backoff$ value will be reduced only when the channel is free, or else the counter freezes until the medium turns idle again.

The mechanism of CW adaptation for unicast packets is the Binary Exponential Backoff (BEB) algorithm. The station uniformly selects a random value for its $backoff$ counter within $[0, CW_i]$, where CW_i is the current CW size and i is the number of failed attempts to transmit this single packet. The default BEB adaptation mechanism can be described as follows;

$$CW_i = 2^i \times CW_{min} \quad \text{for} \quad i \in [0, m], \tag{2.1}$$

where the number of the $backoff$ stages m is given by:

$$m = log_2(CW_{max}/CW_{min}), \tag{2.2}$$

At the first transmission attempt for a packet,

$$CW_0 = CW_{min}. \tag{2.3}$$

If a unicast packet encounters a collision (meaning no ACK was received for a set time frame), then

$$CW_1 = 2 \times CW, \tag{2.4}$$

CW_i is doubled every time a collision happens, until it reaches

$$CW_m = CW_{max} = 2^m \times CW_{min}. \tag{2.5}$$

When $CW_i = CW_{max}$, it maintains this value until a successful transmission is achieved (ACK received). Then CW_i will be reset to CW_{min}, and the process will start again for the next unicast packet. In essence, in the classic IEEE 802.11-based unicast networks, the CW parameter adapts to a value between CW_{min} and CW_{max}, depending on the delivery outcome of the transmitted packets. If a packet transmission fails (ACK not received), the CW parameter is doubled. If the following transmission fails, the CW is doubled again and so goes on until either it reaches CW_{max} or it successfully transmits a packet and resets CW to CW_{min}. By using this mechanism it is less probable that two or more nodes pick the same $backoff$ value and transmit simultaneously.

Two problems appear with the BEB mechanism when trying to establish communication among many highly mobile nodes. Firstly, in dense wireless networks such as VANETs there is higher probability that multiple nodes choose the same CW value, resulting to collisions. Secondly, every time a collision occurs, the CW size is doubled to avoid more collisions. But given that the network density for a VANET can vary a lot over short time periods because of high mobility, a node using a large CW value (because of previous failed transmissions) will wait more than it needs to before transmitting under lighter network conditions. This will result in unnecessary delay.

2.3.3 Enhanced Distributed Channel Access

When just the basic DCF scheme is employed, all nodes contend for access to the medium using the CSMA/CA algorithm with the same parameters. But there are cases where the transmitted data packets are different regarding content and purpose. In such cases, priority of transmission should be handled differently depending on the tolerance of each class of data regarding latency, since the Quality of Service (QoS) for all ideally should be guaranteed [104]. For example, real-time traffic information and collision warning messages have strict delay requirements, while applications such as map data downloading and Internet browsing are more time-tolerant. In order to meet the different QoS requirements such as end-to-end delay and throughput, traffic should be differentiated depending on these. The way of doing this service separation is by setting different contention parameters for different classes of data.

The IEEE 802.11p stack is compatible with the Enhanced Distributed Channel Access (EDCA) from IEEE 802.11e in order to improve the QoS, as an alternative or a complimentary technique to the multi-channel operation defined by IEEE 1609. It offers traffic classification through 4 priority queues, or Access Categories (ACs). When packets have different ACs, they contend internally and the winner will participate in external contention [76]. Data generated by a station's application layers, depending on their class go through a different AC. Every AC has a different value of Arbitrary Inter Frame Space ($AIFS$), which defines a period a wireless node has to wait before it is allowed to transmit its next frame, which replaces $DIFS$ in EDCA-enabled stations. The Contention Window limits CW_{min} and CW_{max}, from which the additionally random $backoff$ waiting time is computed are also variable depending on the AC. The highest the priority, the lowest the value of its $AIFS$ and the limits of its CW, so that internal contention is more likely to be won by the data going through it. The different ACs and the parameter values assigned to each one are shown in Table 2.1. The duration $AIFS(AC)$ is derived from the value $AIFSN(AC)$ by the relation:

$$AIFS(AC) = AIFSN(AC) \times SlotTime. \tag{2.6}$$

As shown in Table 2.1, highly important messages (safety broadcasts) fall in AC3 which has the lowest Arbitrary Inter-Frame Space ($AIFS$) and CW size, so they are more likely to win the internal contention and keep the transmission delay as low as possible. The QoS requirements for various vehicular networking applications can be found in [106].

AC (Priority)	Data Class	CW_{min}	CW_{max}	AIFSN
3 (High)	Safety Related	3	7	2
2	Voice	7	15	3
1	Best Effort	15	1023	6
0 (Low)	Background Traffic	15	1023	9

Table 2.1: Contention Parameters for different Access Categories in 802.11p

2.3.4 Issues of the IEEE 802.11p MAC for Vehicular Ad Hoc Networks

Therefore, this thesis focuses on studying and improving the DCF, which is the default contention-based protocol used for channel sharing in IEEE 802.11-based wireless networks, and consequently IEEE 802.11p VANETs. It employs the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm to manage access to the medium among stations in a distributed way. The DCF's purpose is to avoid collisions of packets by utilizing both the CSMA scheme and the BEB algorithm. When the network carries more data than it handle (network congestion), the Quality of Service (QoS) is negatively impacted. A more efficient MAC layer can better control access to the medium and resolve some of that congestion, improving the QoS.

The BEB algorithm, when enabled, adjusts the CW parameter based on the number of consecutive collisions detected by lack of incoming ACK packets. When it comes to the IEEE 802.11p amendment for V2V communication, the BEB part of the DCF can be considered harmful since it relies on these explicit ACK packets to adjust the $backoff$ parameter depending on whether a transmission was successful or not. This can cause increased delays and unreliability because the non-reception of ACK packets is blocking other urgent transmissions, as seen in [47].

Additionally, implementation of neither the BEB nor ACKs is done for broadcast (OCB) transmissions because they will cause the ACK implosion phenomenon [41] which can lead to service disruption, since there can be many recipients that will all return an ACK upon reception, causing more collisions and packet drops than actually help resolve network traffic congestion. This means that broadcast communication in DSRC has no acknowledgement feature and the choice of $backoff$ values is always limited within $[0, CW_{min}]$. In broadcast transmissions, though, which is the primary way of exchanging information in IEEE 802.11p-based networks, there is no reaction to increases in network load by enlarging the CW parameter or BEB. The reason for this is that original packets are not acknowledged to avoid the acknowledgement storm problem, because every recipient would invoke a $SIFS$ interval and try to send back an ACK, which would cause interference and lead to collisions. Consequently, for the broadcasting case, the $backoff$ counter reinitialises to a uniformly distributed value within $[0, CW_{min}]$ no matter the outcome of the attempted transmission. The operation of CSMA/CA for both unicast and broadcast transmissions can be seen in Fig. 2.5.

A small CW_{min} value means that the stations will not have to wait for many time slots before they can transmit when the channel is sensed to be idle. This is preferable in sparse

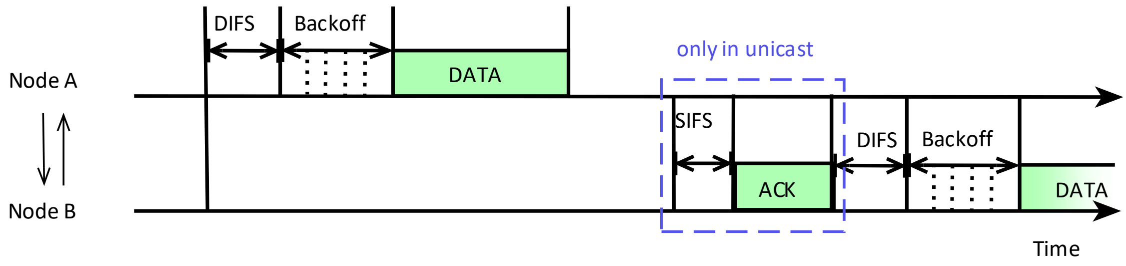


Figure 2.5: A CSMA/CA cycle of operation, managing channel access among transmitting nodes A and B, for both unicast and broadcast transmissions.

networks since it keeps the total transmission delay low and helps not miss transmission opportunities because of waiting longer than needed. But in an urban environment where multiple vehicle-stations continuously transmit using a small CW_{min}, the probability of two or more stations drawing the same $backoff$ after both finding the channel idle and attempting to transmit simultaneously will unavoidably increase, which leads to packet collisions and bandwidth wastage.

Furthermore, the BEB mechanism presents an intrinsic fairness problem, because each station relies on its own direct experience to estimate congestion, which often leads to asymmetric views. Consequently, when the mechanism is utilised under high traffic loads, some stations achieve significantly larger throughput than others, as shown in some studies in literature [98] [55]. The problem occurs due to the fact that BEB resets the CW of a successful sender to CW_{min}, while other stations could continue to maintain larger CW sizes, thus reducing their chances of capturing the channel and resulting in continuous channel domination by the successful station. But even with the BEB mechanism disabled, the large number of collisions in a congested wireless vehicular network can result in unfairness in the system. Consequently an efficient $backoff$ adaptation algorithm replacement that adjusts the CW parameter as needed to tackle the described packet drop and fairness problems could be of great use in such environments.

2.4 Contention in VANETs and Control Approaches

The network traffic congestion in VANETs has a devastating impact on the performance of ITS applications. Given the large number of contending vehicle-stations, especially in an urban environment, it has been found [67] that the default CSMA/CA-based access control layer is not reliable enough due to high collision rates. This means channel congestion control and broadcast performance improvements of the 802.11p MAC are of particular concern and need to be addressed [43] in order to meet the QoS requirements of DSRC applications. A significant reason for this, to be addressed through this research, is the non-adaptation or sub-optimal adaptation of the minimum Contention Window (CW_{min}) size.

2.4.1 Typical Congestion Control Methods

The node density in a typical VANET scenario can vary from very sparse connectivity to more than 100 cars interconnected with each other [88], so VANETs have to (up/down)scale really well [49]. The modifications brought by the IEEE 802.11p amendment focused on the physical layer, while the classic 802.11 MAC layer was enhanced for transmission of data outside BSS context which will contribute towards the scalability goal by removing the association and authentication overheads. But IEEE 802.11 was designed for unicast applications in mind, so it comes as no surprise that the control channel operating under 802.11p can be saturated solely by beaconing, even for medium vehicular or network densities [88].

One idea on how to treat degrading performance on increasing vehicle density that has been around for a long time is limiting the number of contending nodes, which can be done by using mechanisms for transmission power control. When access to the medium becomes difficult, reducing the transmission power of a station reduces the interference area [91]. There are, however, some limitations on the minimum area that safety messages should reach. Another solution, sometimes combined with power control, is controlling the transmission time of a beacon. Since the packet's size are determined by the application, only the data rate can be adjusted. Higher data rate translates into higher transmission probability [66], but also requires higher SNR at the destination of the message, so the coverage area is reduced. This solution suffers from the same limitation as power control.

The way to operate on maximum coverage area and still avoid collisions and prevent performance degradation is optimising the MAC layer employed by the stations in the network. TDMA is found [106] not to be appropriate to resolve the MAC issues presented as it was designed for centralised systems, and would not be applicable in IEEE 802.11p-based networks. There have been TDMA protocol implementations operating in distributed manner but they still are not immune to the contention problem, are more difficult to implement and can accommodate a limited number of vehicle-stations [78], especially with the periodic broadcasting of CAM packets. Hence, the ideal solution to efficient channel allocation among multiple vehicle-stations would be based on the default IEEE 802.11p MAC protocol (CSMA/CA), but with an appropriate $backoff$ (CW) adaptation mechanism relying on the experienced network traffic density. With a high number of transmitting nodes, a large CW size is needed to avoid unnecessary collisions. On the other hand, when the traffic load of the network is low, a small CW size is needed so that potential senders can access the wireless medium with a short delay [102] [23] [103], thus make more efficient use of channel bandwidth. Additionally, the time the channel is idle because of nodes being in the $backoff$ stage could be minimised. In an ideal situation, there would be zero idle time (which is essentially lost and is a synonym of bandwidth wastage) between messages in congested networks with the exception of the $DIFS$ period [88]. The ITS-G5 specification suggests [29] what is defined as the Distributed Congestion Control (DCC) mechanism to control the network load and avoid unstable behaviour. It incorporates the CSMA protocol as the MAC layer featuring CW adaptation via the BEB mechanism (just for unicast transmissions) and Transmit Power Control (TPC) in PHY.

2.4.2 MAC-level Congestion Mitigation in IEEE 802.11p

Work in [57] studies the sensitivity of throughput, latency and fairness to changes of the CW_{min}, CW_{max} parameters of the DCF in IEEE 802.11-based networks with many contending stations. Modifications to the IEEE 802.11 DCF have been proposed regarding mitigating the inherent fairness problem of the DCF, such as the solutions presented in [98] and [16] which both use a $backoff$ copying scheme to achieve fairer bandwidth allocation among stations. However, traditional IEEE 802.11-based networks require that stations are interconnected via an Access Point, and are designed for unicast exchanges. Consequently the protocol cannot be used as-is for V2V communications, which has to be infrastructure-less and accommodate geo-significant transmissions to be received by all peers within a RoI.

The IEEE 802.11p (DSRC) amendment is proposed to tackle peer-to-peer (ad hoc) networking for vehicles. The MAC layer of the protocol adopts the DCF and includes the new OCB mode of operation which allows vehicles to form ad hoc networks among them and enable broadcast transmissions as the primary form of communication. The poor performance of the DSRC MAC in supporting safety applications mainly due to the high collision probability of the broadcasted packets is identified as a key issue in the MAC layer of vehicular networks in [62]. Campolo, *et al.* in [25] show that packet delivery probability, modelled as a function of CW and the number of vehicles, is negatively affected as the nodes increase. Then in [24] they suggest that increasing the CW size reduces the frame loss probability in a similar IEEE 802.11p broadcasting scenario. The work in [48] suggests that a larger CW favours packet delivery for status-message broadcasting which is more delay-tolerant. The impact that vehicular density and increased traffic have on transmission reliability, in terms of packet delivery rates, is also shown in [40]. Additionally, it proposes a new MAC protocol that trades increased packet delay, which still remains below the required threshold for most safety applications, for decreased packet loss by introducing retransmissions. These findings contradict the analysis presented in [77] which suggests that large CW values will increase delay to the point that they can invalidate the proper function of some V2V applications. Work in [108] shows that some proposed safety applications such as Pre-Crash Sensing / Cooperative Collision Mitigation cannot tolerate more than 20 ms of packet delivery latency. A swarming approach for CW adaptation, towards optimising the one-hop delay in inter-platoon V2V communications is presented in [103] . Furthermore, work in [39] proves that for V2I exchanges in sparser networks, a smaller CW will benefit the packet delivery performance of the faster-moving vehicles, allowing them to content fairly with the slower ones. We conclude that there cannot be a value of CW that is suitable for all circumstances, and that can be a problem in broadcast IEEE 802.11p where by default the size of the parameter is not adapted to network traffic.

2.5 Computational Intelligence in Networking

There has been research, as the one presented in [87], which uses fuzzy logic to hard-code existing knowledge regarding the relation of network density with the CW value defining the $backoff$ of vehicle-stations. Fuzzy Logic requires expert knowledge of the system and how the controlled parameter should affect the output. This and other hard-coded or heuristic-based solutions from literature presented so far can be insufficient since the same level of CW can affect the performance in completely different ways given the uncertainty regarding network traffic properties and dynamics of conditions. Not all situations and respective solutions can be known a-priori, so a system controlled this way can have sub-par performance in scenarios that are not predicted at the time of the controller design. Lately there has been emerging work on "smart" communication networks, that employ Machine Learning (ML) algorithms on various levels of the networking stack towards improving their efficiency and enabling new applications [6] [80] [97]. ML was defined in 1959 by Arthur Samuel as "the field of study that gives computers the ability to learn without being explicitly programmed". ML algorithms typically belong in one of three categories:

Supervised, Unsupervised and Reinforcement Learning (RL).

Algorithms belonging in the supervised learning category are provided with “labelled” data sets, which are used to approximate a system model from the relationship between input and output data, in terms of some examined features. Supervised learning algorithms aim to map the target output to the input features as best as possible, so that when given new input data, they can predict the output data based on the built mapping function. The Support Vector Machine (SVM) classification algorithm is used for the purpose of resource allocation for CSMA/CA in [3]. It is also tested as a solution for localization in wireless networks, as seen in [28] or antenna selection in [44]. In that work, a classifier built on the k-nearest neighbors (k-NN) algorithm is also examined as a solution for the same purpose. Neural Networks are also examined for the use of localisation [85] [85] and routing [113] in wireless networks, as well as other purposes.

Unsupervised learning algorithms are provided with unlabelled input data, so no corresponding output is given at the training stage. The goal of algorithms belonging in this ML category is to classify the input data into different groups by investigating their similarities, by discovering previously unknown patterns in the data. When it comes to networking and communications applications, unsupervised learning algorithms are naturally used in problems such as node clustering and data aggregation. Work in [81] explores clustering of nodes in WSNs based on the k-means unsupervised learning algorithm. The same algorithm is used in [38] for periodic data aggregation in WSNs. Work presented in [46] examines the use of the Principal Component Analysis (PCA) algorithm to assess the QoS in VANETs.

RL is a class of ML algorithms fit for problems of sequential decision making and control. It can be used as a parameter-perturbation/adaptive-control method for Markov Decision Processes (MDPs) [13], a discrete time, stochastic control formulation. RL is based on the idea that if an action is followed by a satisfactory state of affairs, or by an improvement in the state of affairs, then the agent’s tendency to produce that action is strengthened, i.e., reinforced. In contrast to the two categories of algorithms mentioned already, RL is not a data-driven approach, meaning that it does not require existing training datasets, labelled or unlabelled, a-priori for the purpose of building a system model. In RL there is an agent that interacts with the external world, and instead of being taught by an example dataset, it learns by exploring the environment and exploiting the knowledge it acquires. The actions the agent takes are rewarded (reinforced) or penalized. The agent uses this feedback from the environment to learn the best sequence of actions or “policy” to optimize a cumulative reward. Networking areas that RL has traditionally been applied on is packet routing [21] [4], resource allocation in wireless networks [34] [82] and other decision-making problems for which collecting a batch of samples for all possible settings and environments is difficult or impossible.

Research on ML solutions towards resource allocation problems, such as the channel sharing - access control problem for VANETs studied in this thesis, is focused on developing RL-based methods [97]. This is because RL is capable of dealing with decision making problems without requiring a detailed dataset collected a-priori as an input. Such complete data sets which accurately represent the examined situations are difficult to collect for resource allocation problems, since the number of uncertainties in a network in terms of density and characteristics of transmitters, data traffic properties for each etc. is very large. Consequently, data-driven approaches such as supervised or unsupervised learning algorithms partially examine such problems, since training sets can only contain a sub-set of possible networking scenarios and combinations of allocating resources.

2.6 Reinforcement Learning in Markovian Environments

2.6.1 Markov Decision Processes

In RL, the learning agents can be studied mathematically by adopting the MDP formalism. An MDP is defined as a (S, A, P, R) tuple, where S stands for the set of possible states, A_s is the set of possible actions from state $s \in S$, $P_a(s, s')$ is the probability to transit from a state $s \in S$ to $s' \in S$ by performing an action $a \in A$. $R_a(s, s')$ is the reinforcement (or immediate reward), result of the transition from state s to state s' because of an action a, as seen in Fig. 2.6. The decision policy π maps the state set to the action set, $\pi : S \to A$. Therefore, the MDP can be solved by discovering the optimal policy that decides the action $\pi(s) \in A$ that the agent will make when in state $s \in S$.

2.6.2 Q-Learning

There are, though, many practical scenarios, such as the channel access control problem studied in this work, for which the transition probability $P_{\pi(s)}(s, s')$ or the reward function $R_{\pi(s)}(s, s')$ are unknown, which makes it difficult to evaluate the policy π. Q-learning [100] [101] is an effective and popular algorithm for learning from delayed reinforcement to determine an optimal policy π in absence of the transition probability. It is a form of model-free reinforcement learning which provides agents the ability to learn how to act

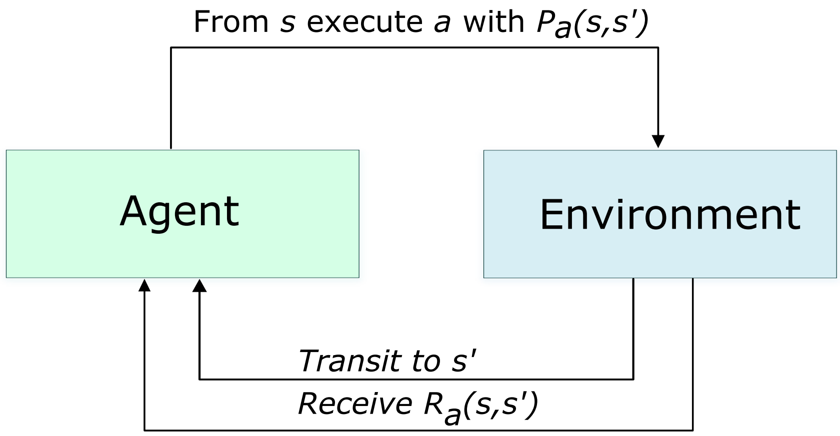


Figure 2.6: Absract MDP model

optimally in Markovian domains by experiencing the consequences of their actions, without requiring maps of these domains.

In Q-learning, the agent maintains a table of $Q[S, A]$, where S is the set of states and A is the set of actions. At each discrete time step $t = 1, 2, \ldots, \infty$, the agent observes the state $s_t \in S$ of the MDP, selects an action $a_t \in A$, receives the resultant reward r_t and observes the resulting next state $s_{t+1} \in S$. This experience (s_t, a_t, r_t, s_{t+1}) updates the Q-function at the observed state-action pair, thus provides the updated $Q(s_t, a_t)$. The algorithm, therefore, is defined by (2.7) which calculates the quantity of a state-action (s, a) combination. The goal of the agent is to maximise its cumulative reward. The core of the algorithm is a value iteration update. It assumes the current value and makes a correction based on the newly acquired information, as shown below.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times [r_t + \gamma \times \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \ - Q(s_t, a_t)] \tag{2.7}$$

where the discount factor γ models the importance of future rewards. A factor of $\gamma = 0$ will make the agent "myopic" or short-sighted by only considering current rewards, while a factor close to $\gamma = 1$ will make it strive for a high long-term reward. The learning rate α quantifies to what extent the newly acquired information will override the old information. An agent with $\alpha = 0$ will not learn anything, while with $\alpha = 1$ it would consider only the most recent information. The $\max_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1})$ quantity is the maximum Q value among possible actions in the next state.

2.6.3 RL for Channel Sharing in Wireless Networks

There has been significant work in academic and industrial research focusing on ML solutions or other intelligent algorithms which are targeted specifically on wireless MAC layer issues. Current approaches are inadequate to cope with the growth of autonomous network elements in various IoT environments, including V2V and V2I communication. An overview of the convergence of machine learning and communications, focusing on applications in wireless networking, is presented in [80]. The study indicates that the performance of mobile networks is strongly influenced by radio resource management i.e., medium access control parameters and suggests that ML techniques can be utilised to augment the MAC layer.

There has also been emerging work specifically focusing on employing the MDP formulation and RL algorithms towards optimising the channel access control layer in wireless networks. In [84] the Markov Decision Process (MDP) formulation is used to design a MAC layer with deterministic $backoff$ for virtualized IEEE 802.11 WLANs. For V2V exchanges, the work presented in [92] examines the IEEE 802.11p MAC performance under the condition of channel contention using the Markov model from [18] and proposes a passive contention estimation technique by observing the count of idle inter-frame slots. The problem of optimizing the IEEE 802.11 $backoff$ mechanism as an MDP is formulated in [7], and Reinforcement Learning algorithms are proposed as a solution. Work in [60] examines adopting Reinforcement Learning as an energy-efficient channel sharing technique for wireless sensor networks. A Q-Learning based MAC protocol for unicast, delay-sensitive VANET exchanges is proposed in [102]. We found that this work does not consider the broadcast nature of VANETs, or the learning algorithm convergence and real-time requirements set by such vehicular use-cases. Additionally there is a potential to further improve the performance regarding packet delivery for various latency requirements and fairness.

2.7 Summary

The vehicular ad-hoc networking domain of the ITS is of particular interest. It is wireless and favours low-latency exchanges, but could be improved to accommodate higher network traffic and denser topologies. The CSMA MAC is the de-facto channel sharing protocol for V2V communications under DSRC frequencies. In the quest for car-to-car communication for intelligent transportation networking applications, improvements on the access control methods could be enablers for applying these in large networks.

The most critical CSMA parameter affecting the channel sharing efficiency and therefore transmission performance in VANETs consisting of multiple stations is CW. According to the literature, it should be set in different levels depending on various factors such as the number of contending stations, or others affecting the network traffic such as the transmitted packet sizes, required latency by applications etc. It also can determine the fairness of the system, thus ensuring or not whether all vehicle-stations in a VANET acquire the appropriate portion of the bandwidth at all times to accommodate their transmissions.

Given the criticality of the applications that have to be enabled via V2X links, these objectives should be satisfied in networks of formed of many vehicles, thus become an objective of the MAC layer. Traditional techniques of sharing the wireless medium are not enough for all but the simplest of applications in such dense networks. Thus a lot of research focus is given into intelligent algorithms for networking use. An investigation into existing intelligent MAC agents is conducted, as well as proposed solutions from the general field. The Markov Decision Process and Reinforcement Learning frameworks are presented, to be used as an enabler of intelligent MAC solutions in the next chapters.

Chapter 3

Performance Assessment of the IEEE 802.11p MAC Layer

3.1 Hardware-based evaluation of the DSRC access control layer

3.1.1 Introduction

The CSMA/CA CW parameter is definitive to the Link Layer performance in VANETs. Specifically CW_{min} parameter defines the entire range of $backoff$ values a station in a VANET can use for broadcast transmissions and can have a great effect regarding channel sharing efficiency and communication performance. To help us understand the true effect the parameter has on communications, testing on real networking hardware was necessary. To the best of our knowledge, there has been no work in evaluating the IEEE 802.11p MAC and especially the effect the CW parameter has on communications, via real hardware implementations. A real-world testing platform that is completely open and allows realistic evaluations of the IEEE 802.11p MAC protocol in hardware became an objective of the study, since no such commercial solution that satisfies this requirement exists to our knowledge. The platform would need to be built from commodity hardware and the software stack would need to rely on open technologies such as the Linux operating system and open wireless drivers that can be modified. Extensibility is also a requirement so that more experiments could be designed in the future.

A real hardware system of interconnected stations was designed in order to observe the effect that CW adaptation has on a station's performance in a congested network. The experiment is based on the Linux wireless subsystem and drivers, which is the most realistic

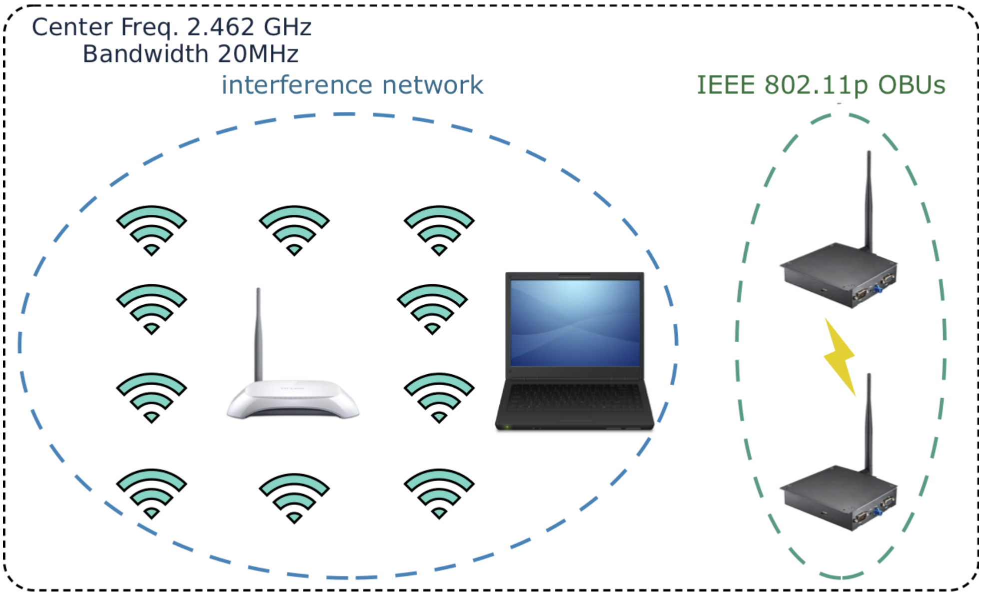


Figure 3.1: Testbed schematic with 2 x IEEE 802.11p and 10 x IEEE 802.11n stations tuned in the same frequency.

way of evaluation since the Linux IEEE 802.11 implementation is open and modifiable, and a lot of existing commercial networking equipment is built around it.

3.1.2 System Architecture

Two IEEE 802.11p OBUs - stations were implemented, one used as transmitter and one as a receiver, for the purpose of assessing the transmitters performance. These stations operate according to the IEEE 802.11p specification so there is no need for an Access Point in between them, and communication is done in an ad-hoc manner. A second network, tuned in the same frequency as the two V2V stations was implemented as to emulate channel contention. This consists of 10 constantly-transmitting WLAN transmitters and a PC receiving all the traffic and evaluating their, all connected to a WiFi access point. A schematic of the testbed is presented at Fig. 3.1.

3.1.3 The IEEE 802.11p stations

Linux Networking Sub-system

The wireless network interface card (NICs) is connected through a PCI-e interface. In most cases, NIC hardware and firmware running on the NICs microcontroller, or just firmware would handle MAC and lower layer functionality. These implementations do not allow freedom of development since developers would not have access to the firmware code. The generation of wireless NICs used for this experiment use a software MAC (SoftMAC),

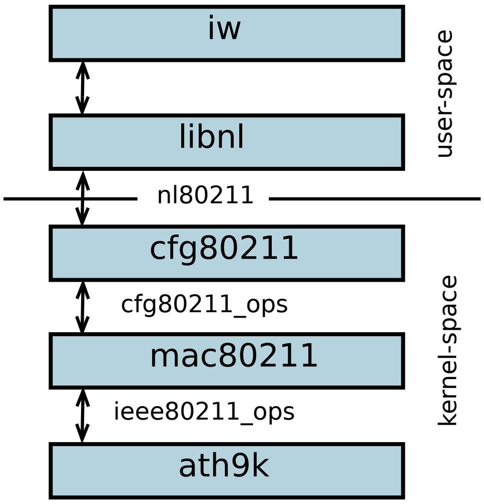


Figure 3.2: Linux IEEE 802.11 stack

loaded as a Linux Kernel module.

The main building blocks of the IEEE 802.11 Linux implementation regarding the MAC layer (and lower-layer) functionality can be seen in Fig. 3.2. Drivers for SoftMAC devices are built on a framework named `mac80211`, and offload functionality that would traditionally be on the hardware do be implemented and controlled in software. The `cfg80211` framework is responsible for the configuration of SoftMAC devices in Linux. The `mac80211` API depends on it for registration to the networking Linux subsystem and configuration, as well as applying regulatory restrictions. Finally, `ath9k` is the driver the kernel loads to interface with the card. The `ieee80211_ops` and `cfg80211_ops` define the callbacks between these APIs. The `nl80211` subsystem, based on the Netlink protocol, acts as a bridge between the `cfg80211` API and the user-space configuration tools. The user-space configuration tools like `iw` are also based on Netlink.

With the networking stack, the drivers and the configuration tools being open source, end-developers can read and modify a large part of the wireless adapter's functionality and parameters. The IEEE 802.11p specification has already been ported to the Linux Kernel, as seen in [59]. The Outside the Context of a BSS (OCB) mode has been enabled in the MAC layer, allowing the NICs to transmit packets without being associated with an access point and even supports the DSRC frequencies at 5.9 GHz. The `iw` utility has been modified accordingly to include new commands for using OCB mode (e.g., join a DSRC channel, leave a DSRC channel).

So this version of the kernel was deployed on 2 APU boards, one to be used as a transmitter and one that would act as a receiver, for performance benchmarking. The addition that was needed regarding the operation of the `Ath9k` driver was a way to be able to change the CW_{min} parameter so that we could test its effect on transmissions in a congested environment. The `AR_DLCL_IFS` register is responsible for setting the CW_{min}, CW_{max} and $AIFSN$ parameters. But modifying the value of CW_{min} directly through that register would require a kernel recompilation for every new value which is a lengthy process. This could make the experiment less accurate since the networking conditions would slightly vary depending on external interference. So the goal was to be able to change the CW_{min} parameter through the user-space quickly so many experiment runs could be conducted in the same environment.

Hardware

The OBUs are implemented with APU2C4 single-board computers, running a Debian Linux derivative (Voyage Linux) with a modified Kernel for IEEE 802.11p support. The board, seen in Fig. 3.3 features an quad-core AMD x86 processor, 4 GBs of RAM, 1 m-SATA port where the SSD with the OS is connected and two mini PCI-E ports, of which one is used to connect the wireless adapter. It also features 1 serial port, used for communicating with the system before the OS is installed or loaded, and 3 Ethernet ports, of which one is used as a high-speed interface with the system via the SSH protocol, after the Linux OS boots.

A Wireless NIC supporting the `Ath9k` driver (SoftMAC) would be needed so that the CW parameter could be adapted from software. For this testbed, a Compex WLE200NX miniPCI-e card is used, which is based on the Atheros AR9280 chipset, which is Ath9k-compatible. These cards support IEEE 802.11n connectivity, with default frequency ranges of 2.412 - 2.472 GHz and 5.180 - 5.825 GHz which is extended to DSRC frequencies via the Kernel (although not used for this experiment).

Contention Window Adaptation

The `iw` Linux user-space utility was adapted to send new CW values to `nl80211`, which on its own turn adapts the parameter used by `mac80211`. A piece of code that was found on legacy versions of `iw` utility allowed manually setting the currently employed Access Category of the EDCA, among the 4 options. That code was re-added at the `phy.c` collection of lower layer functions of the IEEE 802.11p-modified `iw`. The limitation is 4 ACs (set by the kernel driver), so just 4 CW values could be tested at a time, and then the modified `iw` program had to be recompiled with the rest of the values. 7 IEEE 802.11p-compatible CW values were tested (3-255), with $TXOP = 0$ and $AIFS = 2$.

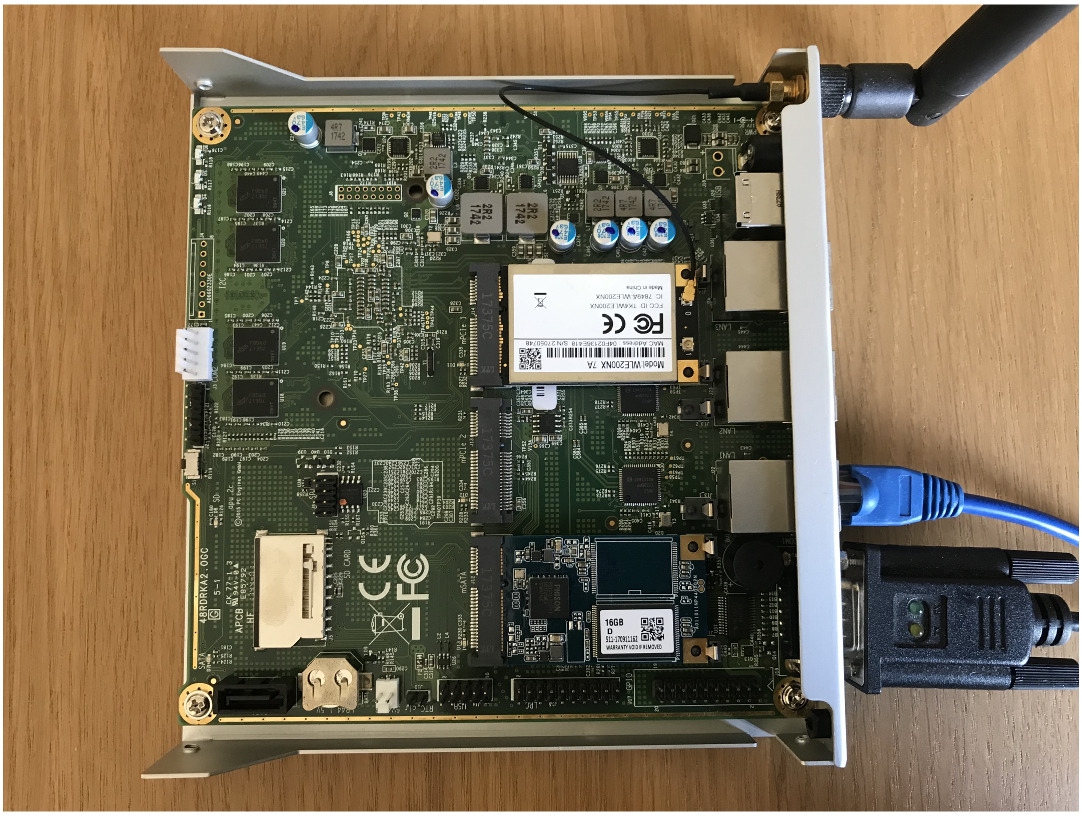


Figure 3.3: APU2C4, the single-board computer used for building DSRC OBUs, equipped with a Compex WLE200NX miniPCI-e IEEE 802.11n wireless module.

Both CW_{min} and CW_{max} would have to be set, as defined at the driver level, although only the minimum parameter would be used in OCB transmissions. In this implementation $CW_{min} = CW_{max}$.

Listing 3.1: C code for passing parameters from userspace to `nl80211` via `libnl`

```
NLA_PUT_U8(msg, NL80211_TXQ_ATTR_QUEUE, queue);
NLA_PUT_U16(msg, NL80211_TXQ_ATTR_TXOP, txop);
NLA_PUT_U16(msg, NL80211_TXQ_ATTR_CWMIN, cwmin);
NLA_PUT_U16(msg, NL80211_TXQ_ATTR_CWMAX, cwmax);
NLA_PUT_U8(msg, NL80211_TXQ_ATTR_AIFS, aifs);
```

Upon using `iw` to adapt the above parameters, the `nl80211` layer would automatically parse these, thanks to the `parse_txq_params` function, as seen below.

Listing 3.2: C code for passing parameters from userspace to `nl80211` via `libnl`

```
txq_params->ac = nla_get_u8(tb[NL80211_TXQ_ATTR_AC]);
txq_params->txop = nla_get_u16(tb[NL80211_TXQ_ATTR_TXOP]);
txq_params->cwmin = nla_get_u16(tb[NL80211_TXQ_ATTR_CWMIN]);
txq_params->cwmax = nla_get_u16(tb[NL80211_TXQ_ATTR_CWMAX]);
txq_params->aifs = nla_get_u8(tb[NL80211_TXQ_ATTR_AIFS]);
```

The `nl80211_set_wiphy` function, which sets physical and other lower layer properties of the IEEE 802.11 driver in its own turn calls the `parse_txq_params`. A necessary condition in order for the function to be called is that the NIC is set in Access Point or Peer-to-Peer GO mode (also known as WiFi Direct). A necessary addition to the code so that `nl80211` could parse and forward the new settings would be support for the OCB mode, done as seen below.

Listing 3.3: C code for enabling CW adaptation via `nl80211` in OCB mode

```
if (netdev->ieee80211_ptr->iftype != NL80211_IFTYPE_AP &&
    netdev->ieee80211_ptr->iftype != NL80211_IFTYPE_P2P_GO &&
    netdev->ieee80211_ptr->iftype != NL80211_IFTYPE_OCB)
return -EINVAL;
```

By performing these changes, we avoid recompiling the kernel every time we need to change the CW_{min} parameter to directly enter in into the `AR_DLCL_IFS` register. This way we can quickly and fairly collect measurements, without large time gaps between using different CW values, and avoiding a change in nearby network conditions.

Listing 3.4: C code for passing the *CW* parameters to the NIC firmware located in `ath9k/mac.c`

```
REG_WRITE(ah, AR_DLCL_IFS(q),
          SM(cwMin, AR_D_LCL_IFS_CWMIN) |
          SM(qi->tqi_cwmax, AR_D_LCL_IFS_CWMAX) |
          SM(qi->tqi_aifs, AR_D_LCL_IFS_AIFS));
```

3.1.4 Contending Nodes Design

Needing to simulate channel contention, and do it an a way that would be simple and inexpensive compare to the method of building IEEE 802.11p stations, it was decided that the experiment would be conducted in WiFi frequencies so that more common wireless modules could be utilised as contending stations.

The ESP-8266 is a popular WiFi module for its Internet of Things applications. It features an IEEE 802.11b/g/n transceiver at the 2.4 GHz band, as well as an on-board 32-bit microcontroller with an operating frequency of 80 MHz and 1 MB of built-in flash memory. These hardware capabilities enable full TCP/IP stack support. The ESP-8266 chips were programmed as UDP clients that connected to the AP, and generated and attempted to transmit 200 packets/s with the server's IP as a destination address. The packets were 400 bytes long so that significant contention could be emulated even using a small number of contending stations (10). The setup is seen in 3.4.

UDP was preferred over TCP to be used as the Transport Layer for the purpose of assessing just the Link Layer performance of the vehicle OBU transmitter, as it is connectionless, and does not feature error-checking and delivery guarantees (i.e., re-transmissions). A PC running Linux, connected to the same WLAN was acting as a UDP server for the ESP nodes to connect to, for the purpose of collecting measurements and validating correct system operation. A benchmarking utility was written and executed at the server computer so as to record the contending nodes' performance, as seen in Fig. 3.5. It features multi-threading, so that every new client could be assessed individually based on an ID contained in the transmitted packets, and it can be expandable to more clients without changes, by automatically adding a new thread collecting measurements every time a new ID is detected in the incoming packets. The ID of the transmitting station was contained in the first 2 bytes of every packet, leaving a payload of 398 bytes. A common WiFi router was used as an Access Point and the appropriate channel was set through its user interface.

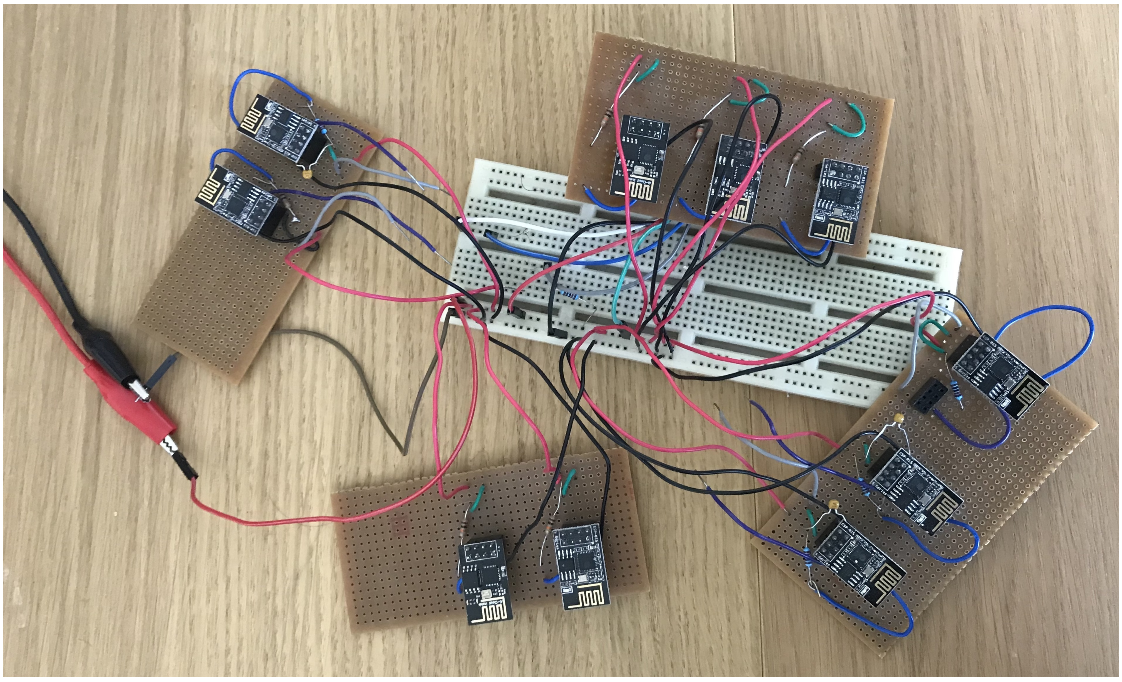


Figure 3.4: 10 IEEE 802.11n ESP-8266 stations were used to emulate contending stations.

```
ID: 0; No of RX: 3865379 packets; IAT: 0.006448 sec; Freq: 155.08; 398 bytes Through: 0.494 Mbit/s
ID: 2; No of RX: 3828326 packets; IAT: 0.006511 sec; Freq: 153.59; 398 bytes Through: 0.489 Mbit/s
ID: 9; No of RX: 3988932 packets; IAT: 0.006248 sec; Freq: 160.04; 398 bytes Through: 0.510 Mbit/s
ID: 7; No of RX: 3790219 packets; IAT: 0.006575 sec; Freq: 152.08; 398 bytes Through: 0.484 Mbit/s
ID: 4; No of RX: 3891416 packets; IAT: 0.006404 sec; Freq: 156.15; 398 bytes Through: 0.497 Mbit/s
ID: 6; No of RX: 3937279 packets; IAT: 0.006329 sec; Freq: 157.99; 398 bytes Through: 0.503 Mbit/s
ID: 3; No of RX: 3883911 packets; IAT: 0.006416 sec; Freq: 155.85; 398 bytes Through: 0.496 Mbit/s
ID: 5; No of RX: 3906018 packets; IAT: 0.006380 sec; Freq: 156.75; 398 bytes Through: 0.499 Mbit/s
ID: 8; No of RX: 3875355 packets; IAT: 0.006429 sec; Freq: 155.54; 398 bytes Through: 0.495 Mbit/s
ID: 1; No of RX: 4025650 packets; IAT: 0.006189 sec; Freq: 161.57; 398 bytes Through: 0.514 Mbit/s
No of Clients 10
```

Figure 3.5: The output of the benchmarking command-line tool running at the UDP server, collecting statistics for 10 constantly transmitting ESP-8266 stations.

3.1.5 System Integration

The system had to be easily expandable while being inexpensive, so that a situation of observable channel contention because of multiple transmitters could be reached and studied. For this reason, commodity WiFi chipsets where used as contending stations and measurements had to be collected at the 2.4 GHz band of frequencies. The whole system would ideally be housed in a room clear from other radio signals or a large metallic enclosure, to avoid external interference. For practical reasons this could not be done, so the measurements were taken in WiFi channels where and when there was no activity. Software was used to track an unused WiFi channel, as seen in Figure 3.6. No Bluetooth or other devices operating at ISM frequencies where placed near the testbed.

The WiFi router on which the ESP-8266 stations were connected was tuned in WiFi channel 6 (at 2437 MHz) since it was not used from other neighbouring WLANs at the area the testbed was deployed, as seen in Figure 3.6. The OBUs were also tuned in the same frequency with a 20 MHz-wide channel (for fairness in measurements, although IEEE 802.11p proposes using 10 MHz-wide channels - and the driver is capable of doing so), by using the `ip` and `iw` command line tools with the following parameters.

Listing 3.5: reset NIC and connect via OCB in Linux bash

```
$ip link set wlan0 down
$iw dev wlan0 set type ocb
$ip link set wlan0 up
$iw dev wlan0 ocb join 2437 20MHz
```

3.1.6 Evaluation

Every measurement consists of 200 packet copies transmitted back-to-back, with $L_p = 400$ bytes. The inter-arrival time (IAT) between the packets is calculated at the receiver side and averaged over the 200 transmissions. Measuring small bursts of traffic ensured that the samples were not as likely to be affected by interference. The packet reception frequency f_{RX} can be obtained from IAT via (3.1). Performance gain of enforcing different CW_{min} values is calculated over the worse-performing CW_{min} value. Collected results can be seen at Table 5.1.

$$f_{RX} = \frac{1}{IAT} \tag{3.1}$$

Figure 3.6: The WiFi explorer software, used to monitor the WiFi channels and assist with selecting the appropriate, free from interference, wireless channel.

CW	**3**	**7**	**15**	**31**	**63**	**127**	**255**
$\overline{IAT}$ **(s)**	0.01196	0.01254	0.01323	0.01356	0.016426	0.017991	0.01907
f_{RX} **(Hz)**	83.6302	79.7304	75.59142	73.7583	60.8779	55.58407	52.43747
Gain %	59.48561	52.04853	44.15535	40.65949	16.0961	6.00067	0

Table 3.1: The effect of using different CW values on DSRC exchanges in terms of mean packet inter-arrival time.

During the experiment procedure, the communication blackout phenomenon occurs occasionally, where transmissions get blocked, as documented in other works which describe hardware experimentation using the Linux networking stack [61]. The IAT measurements affected by the phenomenon are not considered so as not to influence the accuracy of the final results. It is observed that even with a few congesting stations, significant performance gain can be obtained by enforcing a smaller CW_{min} value. This happens because the smaller $backoff$ time gives the OBU an advantage compared to its peers, when competing with them for access to the medium. The ESP-8266 stations feature classic IEEE 802.11n chipset with CSMA/CA, and consequently they compete with each other for access to the medium with a CW of at least 15, that can reach up to 1024 if multiple collisions occur. The DSRC station in this environment has an unfair advantage when competing with the ESP-8266 stations since the $backoff$ value drawn from the interval $[0, CW]$ will be smaller. Since the OBU operates in OCB mode, the CW quantity is always fixed at CW_{min}, which gives an additional edge to the station since its employed $backoff$ value will likely be the smallest among the contenders more often than not. By using the following equation, the rate of transmission can be translated to throughput R, visualised in Fig. 3.7.

$$R = f_{RX} \times L_p \times 8bits \tag{3.2}$$

The measured IAT of packets from a DSRC node to the other without any active contending stations is 0.0083405-0.007884 s (min-max), translating to an achieved transfer rate in the range of 383.67-405.885 Kbit/s. Once the contending stations are activated, it can be seen that there is a significant loss in the packet delivery performance of the DSRC transmitter, which increases as its employed CW increases. It becomes obvious in Fig. 3.7 that there is a breakpoint below $CW = 31$, meaning that reducing the CW used by the DSRC station below this value seems to significantly help its transmissions, in the sense that it wins contention more often. This correlates with the CW used by the IEEE 802.11n stations implemented with the ESP-8266 (which typically ranges from 15-1024 depending on the number of consecutive collisions - $backoff$ stages). This result agrees with work such as [22] that indicate that the CW size parameter of a station can be reduced below the one used by competing peers in order to gain a larger bandwidth share. Through this experiment it becomes apparent that this behaviour can be exploited in hardware by performing the presented Linux kernel modifications. This can be especially problematic for the studied IEEE 802.11p-networks which primary target vehicular safety use, since this method can be exploited to implement (multiple) malicious actors that consume bandwidth disproportionately to other vehicles or RSUs and can effectively jam a large portion of their transmissions. This raises concerns about the protocol design, since most of the V2X communication will be broadcast, and consequently not require the transmission of ACK packets upon reception of a message from a receiver. This means that IEEE 802.11p OCB transmissions are inherently unreliable and the transmitter cannot be sure whether its transmissions are received correctly, so effectively a disconnection of any duration from the network can go unnoticed from the transmitter.

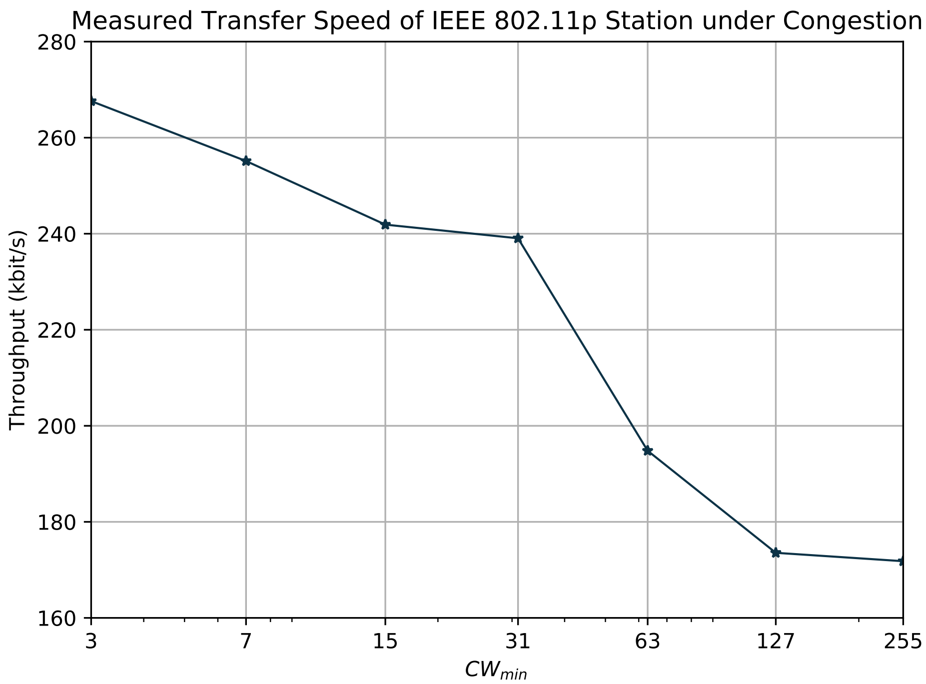


Figure 3.7: Achieved throughput of a IEEE 802.11p station found in a congested WiFi channel in the 2.4 GHz band, for different values of CW_{min}.

3.2 V2V communication and access control performance evaluation under dense simulated networks

The hardware implementation and evaluation of the DSRC protocol stack is a useful but restrictive and rather economically infeasible process. In the presented experiment, we evaluated the IEEE 802.11p MAC and the effect the *CW* parameter has on it, but did it while operating in WiFi frequencies, so that we can easily have a few neighbouring nodes competing for channel access while avoiding the cost and complexity of having many stations operating at DSRC frequencies instead.

In a realistic urban scenario through, the network densities will be much higher. As mentioned, the DSRC L1 and L2 are expected to be able to manage 50-100 contending stations, with some studies [106] indicating ever more. Additionally, the PHY and propagation characteristics are slightly different from IEEE 802.11n (frequency of operation, channel width, data rates and TX range etc.). Consequently, a simulated environment is necessary so that studies for dense V2V networks can be conducted.

3.2.1 Simulation Modeling

A Vehicular Ad Hoc Network simulation has two main components; a **network component** as described above, which must have the capability to simulate the behavior of communication networks as well as a **vehicular traffic component** which provides accurate enough mobility patterns for the nodes of such a network (vehicles/cars).

Network Simulator

There are a few software environments for simulating a wireless network [53], of which OMNeT++ is chosen for its available models, maturity, clear and flexible organisation and code structure as well as advanced GUI capabilities. OMNeT++ [95] is a simulation platform written in C++ with a component-based, modular and extensible architecture.

The basic entities in OMNeT++ are simple modules implemented in C++. Compound modules can be built of simple modules as well as compound modules. These modules can be hosts, routers, switches or any other networking devices. Modules communicate with each other via message passing through gates. The connections from one gate to another can have various channel characteristics such as error/data rate or propagation delay.

An important reason for choosing OMNeT++ to conduct simulation experiments is the availability of third party libraries containing many protocol implementations for wireless networks. The VEINS 4.4 (Vehicles in Network Simulation) framework is used for its DSRC/IEEE 802.11p implementation and its ability to bind a network simulation with a live mobility simulation conducted by SUMO v0.25.

A benefit of using OMNeT++ is the availability of high level APIs from the C++ standard library (STL). These can be of great assistance when performing mathematical operations (power, probabilities, exponential etc.) and provide data structures for efficient storage, searching and manipulation of data (i.e., vectors, queues.).

Mobility Simulator

Since vehicular traffic flow is very complex to model, researchers try to predict road traffic using simulations. A traffic simulator introduces models of transportation systems such as freeway junctions, arterial routes, roundabouts to the system under study. Simulation of Urban Mobility (SUMO) [12] is an open source microscopic and continuous road traffic simulation package which enables us to simulate the car flow in a large road network. Microscopic traffic flow models, in contrast to macroscopic, simulate single vehicle units, taking under consideration properties such as position and velocity of individual vehicles.

The Krauss mobility model is the default vehicle mobility model used in SUMO. It is a microscopic, space-continuous, car following model based on the safe speed paradigm. The driver tries to stay away from the vehicle ahead (leading) at a distance and a safe speed that allows him to adapt to the leading vehicle's deceleration if needed [50]. The safe speed can be calculated as follows

$$v_{safe} = v_{lead}(t) + \frac{g(t) - v_{lead}(t) \times \tau}{\frac{v_{lead}(t)+v_f(t)}{2b} + \tau}, \tag{3.3}$$

where v_{lead} represents the speed of the leading vehicle in time t, g_t is the gap to the leading vehicle in time t, τ is the drivers reaction time (usually 1 s) and b is the maximum deceleration ability of the vehicle. But v_{safe} can exceed the maximum allowed speed on the road or the vehicle's capability. Consequently, the desired speed is calculated from the following,

$$v_{desired}(t) = min(v_{max}, v(t-1) + a, v_{safe}(t)), \tag{3.4}$$

where v_{max} is the maximum velocity of the vehicle and a is the acceleration capability of the vehicle. But since the driver cannot realistically drive always perfectly (with the desired velocity), to get the vehicle's speed the following equation is used

$$v(t) = max(0, random[v_{desired}(t) - \sigma a, v_{desired}(t)]). \tag{3.5}$$

where σ is between 0 and 1 and models the driver's imperfection.

Simulating a virtual VANET scenario

The simulation environment on which novel Medium Access algorithms are to be evaluated uses SUMO and open data to reproduce accurate car mobility. The map can be extracted off OpenStreetMap and converted to an XML file to define the road network or can be hard-coded for simpler topologies (grids, highways etc.). Then random trips are generated from this road network file, and finally these trips are converted to routes and traffic flow. The resulting files are used in SUMO for live traffic simulation as depicted in Fig. 3.8.

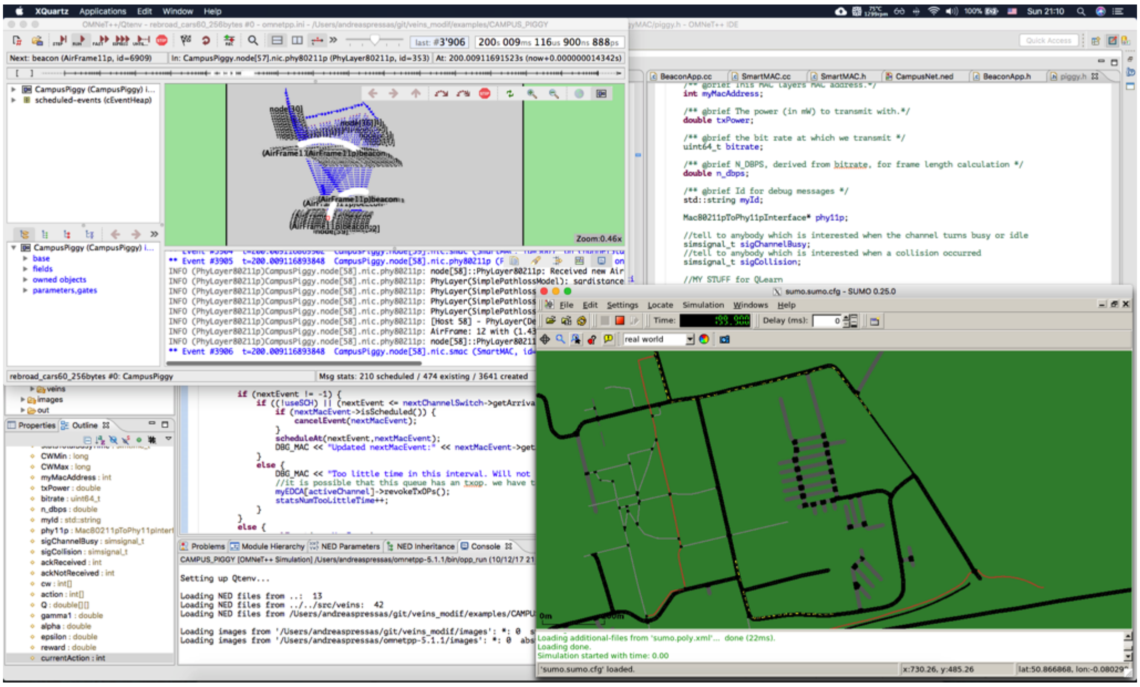


Figure 3.8: IEEE 802.11p/WAVE & Mobility simulation with OMNeT++ & SUMO & Veins.

Each node within OMNeT++, either mobile (car) or static (Roadside unit) consists of a network interface that uses the IEEE 802.11p PHY and MAC and the application layer that describes a basic safety message exchange and a mobility module. A car, chosen in random fashion, broadcasts a periodic safety message, much like the ones specified in the WAVE Short Message Protocol (WSMP). Fig. 3.9 shows examples of V2V connectivity, where cars broadcast a safety message to neighbouring cars within range.

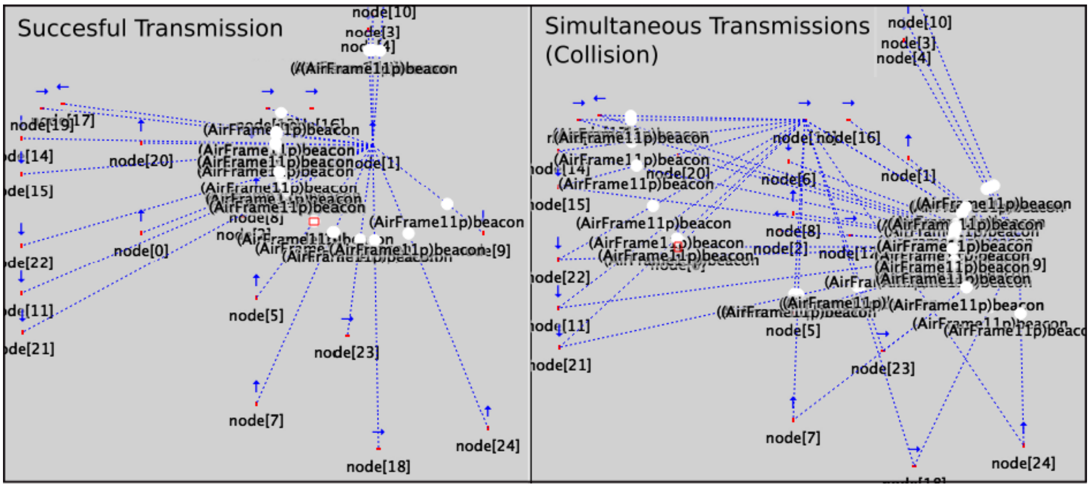


Figure 3.9: A station broadcasts a packet in OMNeT++ (Left). Two stations simultaneously attempt to broadcast their packets, leading to collision and lost transmissions (Right).

Examined Network Densities

With a theoretical maximum LoS range of $r_{tx} = 1\,km$, a realistic density of 50 transmitting vehicles is defined. As mentioned already, the proposals for V2V communication indicate that MAC layer should handle the beaconing and additional applications for 50-100 stations in the communication zone of each other without a major collapse in performance, which gives a maximum vehicle density of:

$$D_{network} = \frac{N_{vehicles}}{\pi \times r_{tx}^2} = \frac{100}{3.14} \approx 31.84\,vehicles/km^2, \tag{3.6}$$

If the deployed stations are tuned to reach half that range and the area of coverage of a vehicle is reduced, we get the density by limiting the maximum range to $r_{tx} = 500\,\text{m}$:

$$D_{network} = \frac{N_{vehicles}}{\pi \times r_{tx}^2} = \frac{100}{0.76} \approx 131.57\,vehicles/km^2, \tag{3.7}$$

These values can be characterised as low-to-medium density and high density respectively. These densities are within the ranges examined by other publications, such as [72] which studies information dissemination for $D_{network} = [20, 300]\,vehicles/km^2$. The $vehicles/m$ density metric (for highways), or absolute number of vehicles are also used.

Evaluating Throughput

We use receiver-centric metrics to evaluate the performance of the system regarding standard's and suggested approaches' performance, since they better represent the level of awareness every vehicle has of its surrounding vehicles. Raw throughput in terms of intact packets received over time is measured at all receivers and then a moving average filter is applied so that we can collect a system-wide reading over time. That way the real-time effect of the learning algorithm onto network performance can be evaluated.

Achieved throughput can also be expressed in terms of system-wide Packet Delivery Ratio (PDR). For homogeneous scenarios, we measure all packets received per station and find the mean number of received packets $N_{packets_{RX}}$. The total number of packets to be transmitted, successfully or not, during a time period is known a priori from broadcasting frequency, since for a single station $N_{packets_{TX}} = f_b(Hz) \times t_{period}(s)$, and network-wide $\sum N_{packets_{TX}} = N_{packets_{TX}} \times N_{vehicles}$. Therefore, to get the PDR,

$$\overline{PDR} = \frac{N_{packets_{RX}}}{\sum N_{packets_{TX}}} = \frac{Th_{measured}}{Th_{max}}. \tag{3.8}$$

This evaluation of system performance based on $\sum N_{packets_{TX}}$ works well for homogeneous, single-hop scenarios, but less so for multi-hop, since this quantity is not known at all times and even if it was, it is inefficient to use it to calculate PDR. The reason for that is that every vehicle experiences a different environment in terms of nearby transmitters (within range), so PDR cannot be simply found using this method. Additionally, because in multi-hop we care about the reachability of protocols in terms of distance they cover, PDR is measured in terms of unique copies of packets received. So instead we find $N_{original\_packets_{TX}} = f_{gen} \times t_{period}$, and PDR is found in terms of unique copies of packets received from anywhere in the network.

Evaluating Latency

On the other hand, end-to-end latency of received transmissions is measured at a single station placed in the middle of the network. Each generated packet contains the time it was created at the application layer, and is subtracted by the time of reception by the receiving node's application layer.

$$\overline{latency} = \frac{\sum (t_{App_{RX}} - t_{App_{TX}})}{N_{packets_{RX}}}. \tag{3.9}$$

If we consider the case of beaconing (periodic broadcast transmission of kinematic information about the vehicle such as position or velocity), then we find the following limitations to occur regarding end-to-end communication latencies presented in Table 3.2. The table shows packet relevance time and displacement per packet for various packet transmission rates for a low relative vehicle velocity of 50 km/h = 13.9 m/s (which can reach up to more than 200 km/h).

f_b	**Time of relevance**	**Displacement**
20 Hz	50 ms	0.695 m/packet
30 Hz	33 ms	0.463 m/packet
50 Hz	20 ms	0.278 m/packet

Table 3.2: Time of relevance and displacement among vehicle-stations per transmission by vehicles for various packet transmission rates f_b.

If a high accuracy is required by an ITS application, the displacement per packet exchanged between a transmitter and receiver should be kept low. It is then natural that some work dealing with applications like CACC [70] model the control system considering communication latency of just 10 ms.

Evaluating Fairness

The fairness objective can be characterized in two different manners: long-term and short-term. Long-term fairness is measured over long time periods, corresponding to the transmission of many packets by a station, i.e., 1000 or more. A MAC protocol is considered to be long-term fair if the probability of successful channel access observed over a long period of time (many packets transmitted) converges to $1/N$ for N competing hosts. But a MAC protocol should also provide equal opportunity for access to the medium over short time periods as well, i.e., lasting a few seconds or tens of packets transmitted per station. A MAC protocol can be long term fair but short-term unfair, meaning that one host may continuously capture the channel over short time intervals. Vehicles transmit safety-related, irreplaceable packets with a short time of relevance. All cars should be given equal transmission opportunities, not only in the long term but in the short term as well (i.e., 2-4 s).

$$J(x_1, x_2, ..., x_n) = \frac{(\Sigma_{i=1}^{n} x_i)^2}{n \times \Sigma_{i=1}^{n} x_i^2}. \tag{3.10}$$

We conduct short-term fairness analysis using Jain's fairness index [42] shown in (3.10), which is a popular metric for measuring the unfairness of an allocation vector. We adopt it for analysing the fairness of achieved throughput among wireless vehicular stations. The index value equals unity corresponds to the fairest allocation in which all stations achieve the same throughput. We set the fairness criterion to be $J = 95\%$, according to other works dealing with fairness in IEEE 802.11-based systems, such as [14][15]. The number of received packets from all transmitters are measured at a single vehicle and J is calculated over a sampling window of 1 to 10 s (short-term to long-term) with a step of 0.5 s. This result is averaged over equally spaced starting points with $\varepsilon = \alpha = 0.05$, to increase the accuracy of measurement.

3.2.2 Simulation Parameters

A goal of this chapter is to study the the intrinsic properties of the IEEE 802.11p channel access method in terms of throughput, latency and fairness, so we concentrate on a homogeneous singe-hop scenario in which all stations experience similar transmission conditions, meaning that no station is disadvantaged by its signal quality, traffic pattern, or spatial position, or other asymmetries. We collect our results within a specific RoI of $\approx 600\,m \times 500\,m$ within the University of Sussex campus, and we set the transmission power of the stations to a high enough level within the DSRC limit (30 dBm), so that they all can reach each other and the MAC evaluation is not influenced by border effects (hidden/exposed stations). The artificial campus map used for simulations can be seen in Fig. 3.10. Since we are interested in evaluating just the baseline MAC layer networking performance, the effect of mobility is minimised by enforcing low speeds to the vehicles ($\approx 5km/h$).

Figure 3.10: Campus map used by SUMO for vehicular traffic co-simulation.

In these simulations, all cars in the network are broadcasting packets such as CAMs or

Parameter	Value
Channel Frequency	5.89 GHz
Channel Bandwidth	10 MHz
Transmission rate R	6, 9 Mbit/s
Transmission power	30 dBm
Network density	50 vehicles
Packet size L_p	256 bytes
Backoff slot time	13 μs
Broadcasting Frequency f_b	30, 50 Hz
Packet Generation Offset	0.005 s

Table 3.3: Simulation Parameters for IEEE 802.11p MAC evaluation.

DENMs. Most proposed V2X applications need a packet transmission rate of at least 10 Hz [19], while some need even up to 50 Hz [52]. A packet transmission rate of 10 Hz is adequate when considering just periodic status message broadcasting. If additional warnings, CACC or other systems are implemented, the packet generation and transmitting frequency for every station naturally goes up, while the packet length also varies depending on the use. For simplicity, we evaluate V2V systems where all stations attempt to transmit 30 packets/s, with packet length $L_p = 256\,bytes$. Some asynchronisation is introduced to transmissions by adding a randomised offset time that can reach a maximum of 0.005 s. The bit rate R is set from the allowed DSRC rates at 6 Mbit/s and frequency of operation is set at 5.89 GHz.

3.2.3 IEEE 802.11p MAC evaluation in a symmetrical network

The effect of CW on PDR can be seen in Fig. 3.11. In a network formed of a lot of contending stations, enforcing a larger CW_{min} value on all of them improves the network-wide performance regarding delivery of intact packets via successful collision avoidance. This happens because there is lower probability of more than one station drawing the same $backoff$ value from the interval $[0, CW_{min}]$ when they find the medium being busy, which results into lower probability of collisions occurring. Using a small CW_{min} level results in fewer options in terms of $backoff$ values for stations and consequently higher collision probability in the network, since two or more stations drawing the same $backoff$ results into simultaneous transmissions and subsequently their failure.

The effect of CW on end-to-end TX latency is represented with a Cumulative Dis-

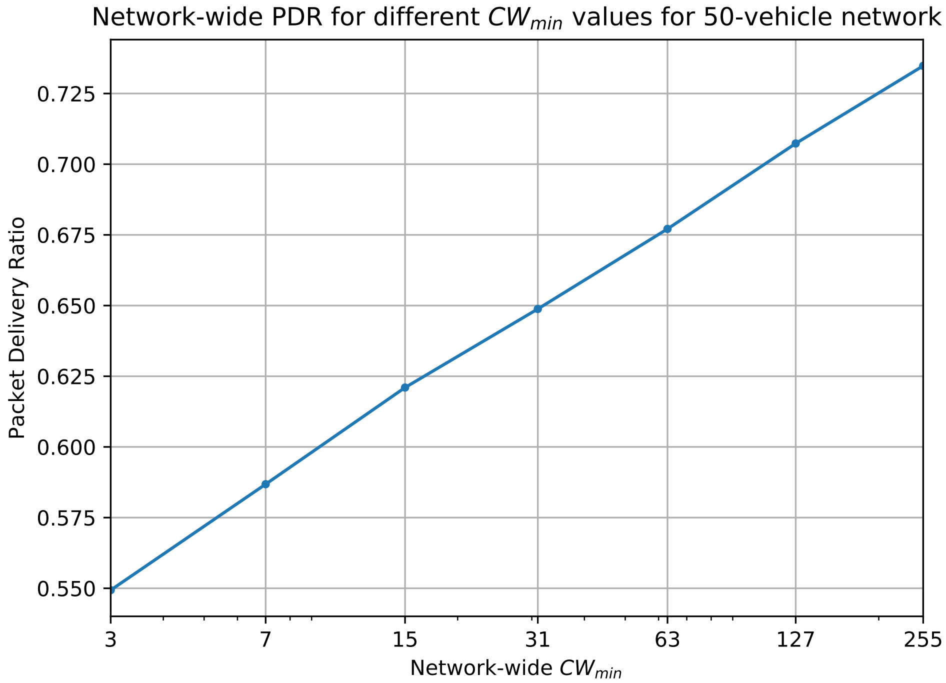


Figure 3.11: PDR in a network of 50 contending IEEE 802.11p stations for different CW_{min} values applied on all stations.

tribution Function (end-to-end latency over percentage of received packets), as seen in Fig. 3.12. Naturally, a smaller CW_{min} value results in smaller $backoff$ values being used by stations and consequently smaller end-to-end transmission delays. This is because the $backoff$ parameter is the number of time-slots waiting and adds to end-to-end transmission latency. So every time the wireless medium is found to be busy, $T_{backoff} = backoff \times T_{slot}$ is added to the transmission delay.

So $T_{backoff}$ of up to $max(backoff) \times T_{slot} = 255 \times 13\,\mu s = 3.315\,ms$ can be added to the transmission time. But the added latency to a transmission because of the $backoff$ time will be much more in a network with multiple contenders. As mentioned already, if the channel turns busy during the $backoff$ countdown then it freezes, and starts again from the value it left of, after the channel is found to be idle for a $DIFS$ period. This means that the transmission process by a station can be paused multiple times (until the $backoff$ timer expires). Consequently, the longer the $backoff$ time is, the probability of missing a transmission opportunity increases, and the total added end-to-end latency to the transmission can be impacted heavily. For a network of many transmitters, naturally the $backoff$ countdown process can freeze multiple times if the $backoff$ parameter is large.

The previous figure presents raw latencies of transmissions over percentage of these

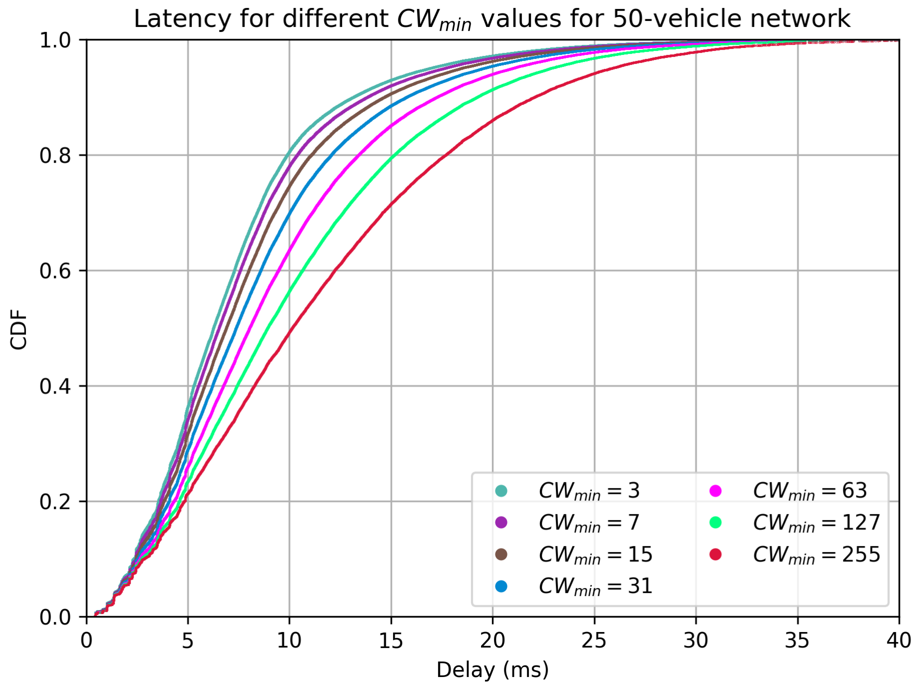


Figure 3.12: Latency versus percentage of successful transmissions in an IEEE 802.11p-based VANET of 50 stations for different network-wide CW_{min} values.

successful transmissions, disregarding the percentage of successful transmissions over attempted transmissions (PDR). Normalising the result by multiplying with PDR, reveals the achieved packet delivery ratio, over latency of these transmissions, recorded in Fig. 3.13. From this figure it becomes clear that even when setting $CW_{min} = 3$ for all stations does not necessarily mean that the system will better accommodate latency-sensitive exchanges. We see that for a medium-density network of 50 vehicle-stations, a network-wide setting of $CW_{min} = 7/15$ will achieve the most successful 10 ms-exchanges, while a setting of $CW_{min} = 127$ will achieve the most successful exchanges with a requirement of 20 ms end-to-end. For latencies above 24 ms, using the higher CW_{min} will result in lower collisions and higher PDR. We try a maximum value of $CW_{min} = 255$, since even this value can negatively affect the delivery of transmissions with a sub-20 ms end-to-end latency target.

In Fig. 3.14 is depicted the transmission fairness of the network over intervals of 0.5 to 10 s with a step of 0.5 s. Although the same value is used network wide, there is still unfairness due to a high amount of collisions. For a CW_{min} value of 63 and above the network becomes short-term fair within 2 seconds or 60 transmissions/station.

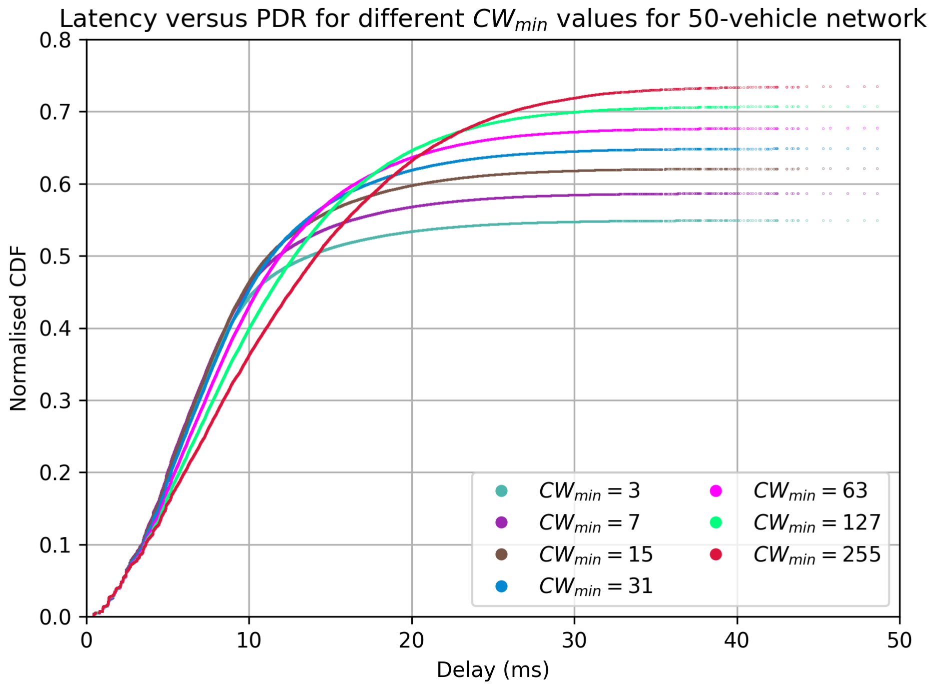


Figure 3.13: Latency versus packet delivery fraction in an IEEE 802.11p-based VANET of 50 stations in OMNeT++.

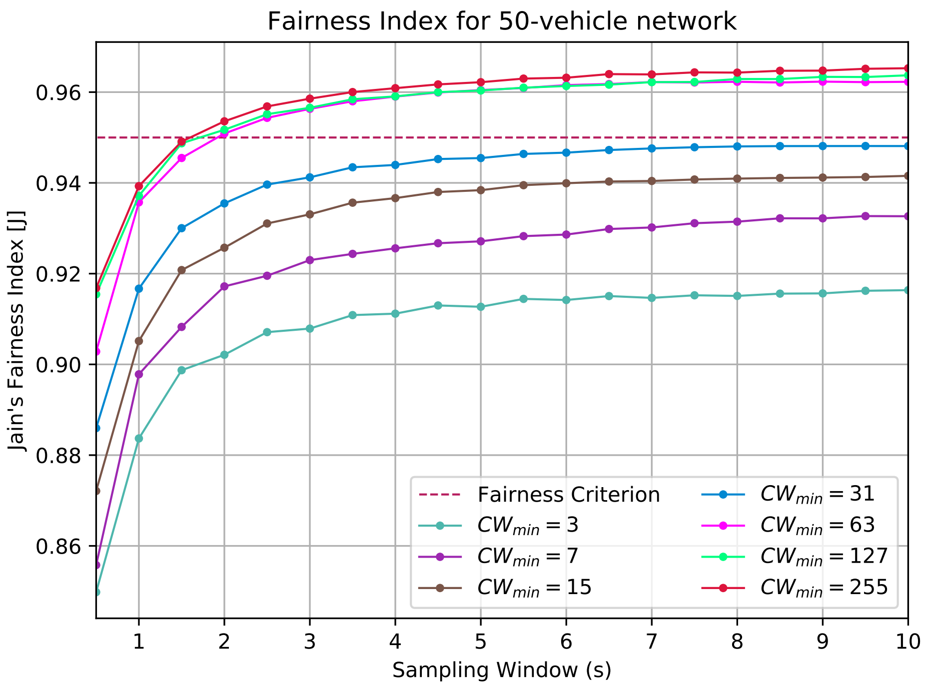


Figure 3.14: Achieved fairness among vehicle-stations employing the same CW_{min} for communications.

3.2.4 Greedy stations in saturated networks

The hardware finding inspired us to do a larger scale simulation using OMNeT++ for the purpose of confirming the finding of the CW parameter affecting the symmetry of bandwidth allocation among stations and benefiting a portion of them in the network. This time, 50 IEEE 802.11p stations are deployed in total. All of them attempt to transmit 512-byte packets every 20 ms. Consequently, the effective broadcasting (transmitting) frequency is f_b=50 Hz, resulting to an attempted transfer rate of

$$Th_{station} = f_b \times L_{packet} \times 8\,bits \tag{3.11}$$

$$Th_{station} = 50\,Hz \times 512\,bytes \times 8\,bits = 204.8\,Kbit/s, \tag{3.12}$$

generated per station. The channel bitrate R is set at 9 Mbit/s to help accommodate the increased traffic. Among these stations, 40 use a fixed CW_{min} of 255 (so for broadcast IEEE 802.11p the max $backoff$ period can be 255 timeslots). The CW_{min} is varied for the remaining 10 of these stations, in an attempt to see if it would similarly affect the communications in such dense, high-traffic networks.

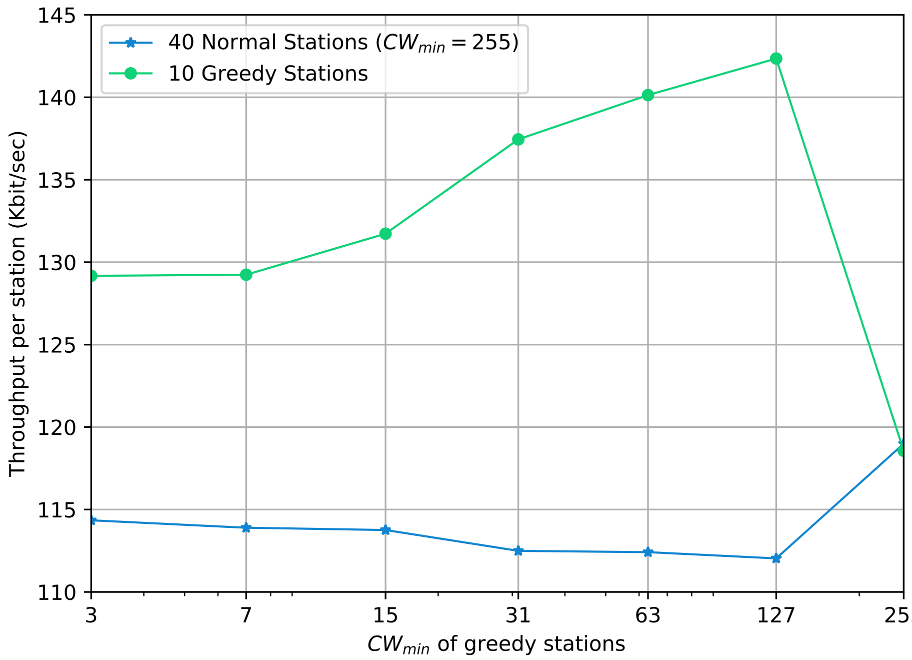


Figure 3.15: IEEE 802.11p stations gain a significant advantage in communications over peers by using a smaller CW_{min}.

Results are presented in Fig. 3.15. This time we can ensure symmetrical stations in terms of MAC layer settings, PHY implementation and differences in hardware. Naturally,

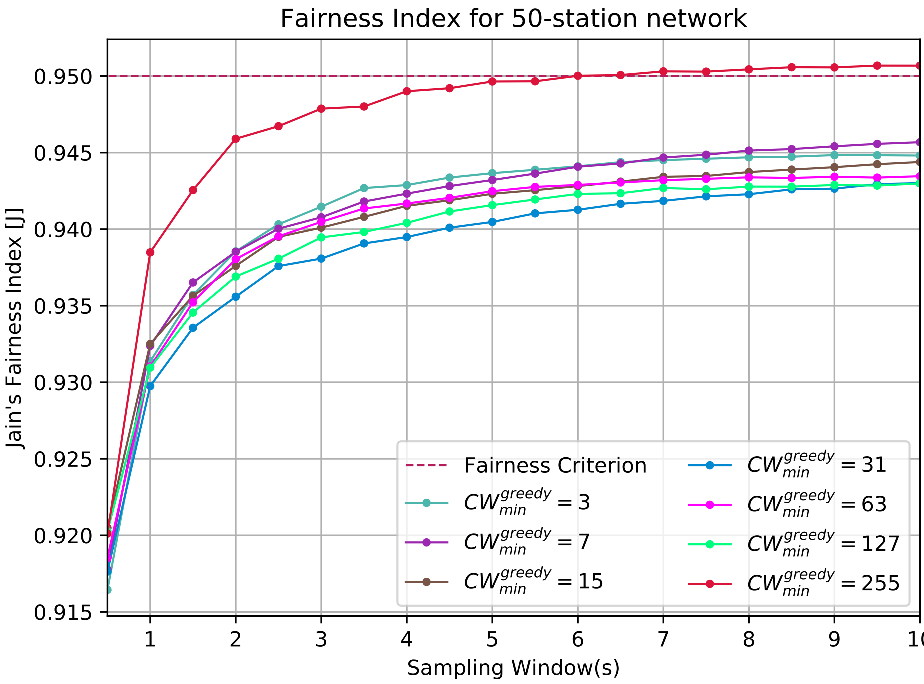


Figure 3.16: Network-wide fairness, affected by a minority of stations using different CW values than the rest.

using the same $CW = 255$ for all stations yields similar results in terms of throughput collected in OMNeT++. Using any lower CW value for the 10 remaining stations labelled as "greedy" will yield higher throughput result for said nodes, in expense of some of the packet deliveries performed by the "normal" stations (with CW=255). Again, we find this natural since this gives higher chances of the "greedy stations" competing with smaller $backoffs$ than "normal" IEEE 802.11p stations. But it can also be observed that the higher CW values (below 255) employed by greedy stations (63, 127) still yield the highest performance, because of increased collisions when lowering the CW. This is again normal for this case of heavy contention in the simulator, since longer $backoff$ times result in smaller probability of collisions. On the other hand, since the largest percentage of stations (4/5) uses a large CW value, the studied (greedy) stations will outperform these even for the smallest $CW = 3$.

This asymmetry in the VANET in terms of the CW_{min} value of the stations affects the overall network performance, reflected in measured throughput fairness when examined within either short or longer intervals. In a VANET undergoing significant contention, even a minority ($1/5_{th}$) of stations exploiting their contending priority via CW adaptation negatively impacts the fairness performance, as seen in Fig. 3.16.

3.3 Summary

After performing a study on the IEEE 802.11p MAC internals via both hardware and our simulator stack using OMNeT++ and Veins, it was found that OCB DSRC systems performance in dense networks is greatly affected by the CW_{min} parameter. A higher value enforced for all stations gives less probability of two or more stations drawing the same $backoff$ value post-$DIFS$, and transmitting simultaneously after the $backoff$ countdown ends. This translates in higher achieved throughput, for the network and the stations individually. On the other hand, increasing the CW value also gives a higher probability of stations drawing larger $backoff$ times, which in dense networks will increase the end-to-end transmission latency and cannot be tolerated by some V2V applications. Additionally, if some stations use a smaller CW parameter than the rest, there is higher probability they will win access to the medium when contending. It is found that for all stations transmitting with the same frequency f_b and packet size L_p, using the same CW value for all promotes symmetry in the network regarding $backoff$ times and translates to higher short/long-term network-wide fairness (J). Consequently, the optimal CW value for a station is very dependent on the existing network traffic, and can be exploited to favour transmissions of some nodes over others in the network. Although larger CW values reduce collisions when employed by many/all stations in a dense network, it does not necessarily translate to best delivery performance for all stations and types of applications.

Chapter 4

Q-Learning-based IEEE 802.11p-compatible Access Control

4.1 Introduction

The *CW* parameter is found to significantly affect the performance of IEEE 802.11p stations when found in a network where a large amount of traffic is exchanged. Correct adaptation of the parameter can offer significant performance benefits to the network and the stations individually. We find that the optimal value of the parameter depends on multiple factors, such as the network density in terms of active transmitters and amount of traffic exchanged at a time as well as required throughput and latency tolerance of the application or prioritisation of some exchanges over others. There is a clear trade-off when selecting the *CW* size for stations in a network, since it should be large enough to be able to accommodate the network traffic as much as possible without collisions, but not unnecessarily large so that it increases packet transmission latency and stations miss opportunities to transmit because of waiting too long and the channel turning busy. In other scenarios, the *CW* of a specific data class should be lower to gain a larger portion of the available bandwidth for prioritisation purposes.

The primary objective of the study is a protocol that learns how to control the *CW* parameter in VANETs to enhance packet delivery performance in congested networks, thus make more efficient use of the available bandwidth. The Q-Learning machine learning algorithm is employed because of its ability to discover good solutions over time by learning via trial-end-error interactions with the environment, without requiring any knowledge of the environment. For these reasons we employ Q-Learning to adapt the *CW* as needed. This algorithm requires insignificant computational capability from the MAC controller and has minimal networking overhead, apart from some form of reception acknowledgement that is typically standard in unicast wireless networks for reliability purposes and is utilised by most applied contention-based MAC protocols for the purpose of feedback.

When combined with the IEEE 802.11p MAC, the Q-Learning algorithm can detect network contention and adapt a station's CW value as needed to resolve it as much as possible, without any information about the network or the application layer itself known a-priori. By each station acting toward its own interest regarding CW selection, the entire network can achieve more bandwidth-efficient channel sharing. In the following sections we present employing (2.7) as a learning, self-improving, control method for managing channel access among IEEE 802.11p stations.

4.2 Q-Learning MAC Protocol Design

The adaptive backoff problem fits into the MDP formulation. RL is used to design a MAC protocol that selects the appropriate CW value based on gained experience from its interactions with the environment within an immediate communication zone. The proposed MAC protocol, features an RL-based algorithm that adjusts the CW size based on feedback given from probabilistic rebroadcasts in order to avoid packet collisions. In the remaining of this section we present employing Q-Learning to design a learning, self-improving, control protocol for sharing the wireless medium among multiple IEEE 802.11p stations. With the state space S being the available CW values, (2.7) is adapted and used as follows;

$$Q(CW_t, a_t) \leftarrow Q(CW_t, a_t) + \alpha \times [r_t + \gamma \times \max_{a_{t+1}} Q(CW_{t+1}, a_{t+1}) \ - Q(CW_t, a_t)]. \tag{4.1}$$

The protocol works as follows: a station transmits a packet and then gets feedback r_t depending on the outcome of this transmission, determined by the reception or not of a packet containing an ACK within an acceptable Round-Trip Time (RTT). The RTT is defined as the time needed for the original transmission to be completed and an ACK for that transmission to be received by the transmitter's application layer. The acceptable RTT depends on the transmission latency requirements, and in this ACK implementation is set to be less than the packet generation period for simplicity. The Q-Learning agent then adapts the station's CW value accordingly before sending the next packet, and then the process is repeated. The Q-Learning MAC protocol's operation is depicted in Fig. 4.1.

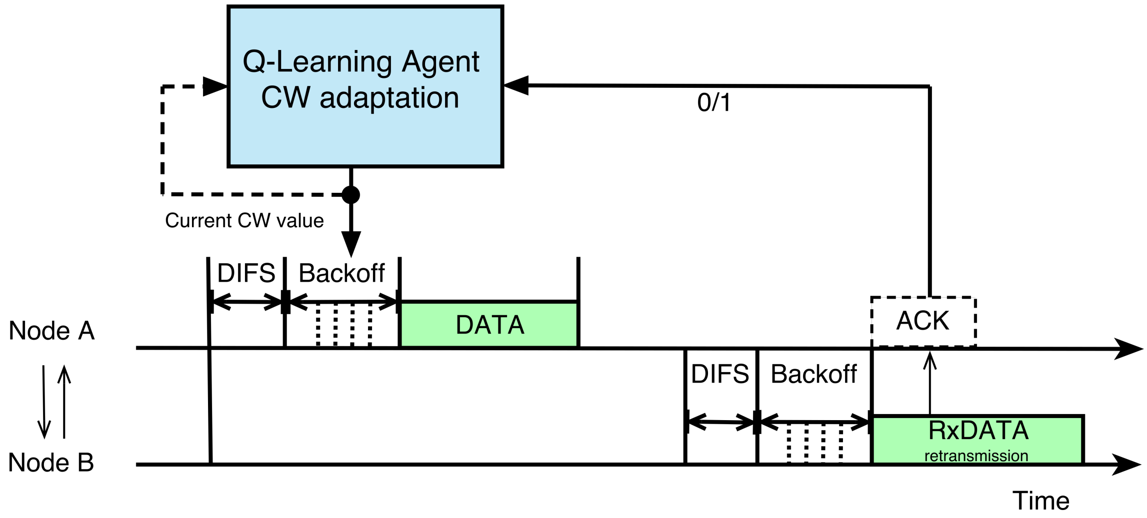


Figure 4.1: Q-Learning-based MAC protocol schematic of operation

4.2.1 The Exploration-Exploitation (Action Selection) Dilemma

The Q-Learning MAC protocol's primary purpose in this application is to converge to a (near) optimum output, in terms of packet delivery ratio. It achieves this by transitioning to different CW values (states S) by performing actions $a \in A$, transmitting packets and then getting experience from these transmissions using said CW values, via feedback in the form of overheard retransmissions. The operation of the proposed self-learning channel access control mechanism is summarised in Algorithm 1.

Watkins and Dayan [101] proved that Q-Learning converges to the optimum (s, a) pair/s with probability 1 as long as all actions are repeatedly sampled in all states s and the (s, a) pairs are represented discretely. To meet the second convergence criterion, the state space S for this channel access control contains 7 discrete IEEE 802.11p-compliant CW values ranging from $CW_{min} = 3$ to $CW_{max} = 255$. The CW is adapted according to (2), prior to every packet transmission attempt. The action space A contains the 3 following actions a, which are the same the BEB MAC mechanism uses to adapt the CW upon transmission failure.

$$CW_{t+1} \xleftarrow{a \in \{CW_t\ -\ 1/2, CW_t, CW_t \times 2 - 1\}} CW_t \tag{4.2}$$

RL algorithms differ from supervised learning [45] ones in that correct input-output pairs are never presented, and sub-optimal actions are not explicitly corrected. In addition, there is a focus on on-line performance, which necessitates finding a balance between exploration of uncharted territory and exploitation of already acquired knowledge. This in practice translates as a trade-off in how the learning agent in this protocol selects its next action for every algorithm iteration. It can either explore by randomly picking an action from (4.2) so that the algorithm can transit to a different (s, a) pair and get experience (reward) from it, or follow a greedy strategy that exploits its so-far gained experience, and choose the action a which yields the highest Q-value for the state s it is currently in, given by

$$\pi(s) = \arg\max_{a} Q(s, a)\,. \tag{4.3}$$

4.2.2 Accelerated Learning with Decaying ε-greedy

The RL algorithm's purpose is to converge to a (near) optimum output, in terms of CW value selection. The greedy policy with respect to the Q-values tries to exploit what is known to work continuously, however, since it does not explore all (s, a) pairs properly, it fails satisfying the first criterion. At the other extreme, a fully random policy continuously explores all (s, a) pairs, but it will behave sub-optimally as a controller. An interesting compromise between the two extremes is the ε-greedy policy [89], which executes the greedy policy with probability $1 - \varepsilon$. This balancing between exploitation and exploration can guarantee convergence and often good performance.

The proposed protocol uses ε-greedy strategy to focus the algorithm's exploration on the most promising CW trajectories. Specifically, it guarantees the first convergence criterion by forcing the agent to sample all (s, a) pairs over time with probability ε. Consequently, the proposed algorithmic implementation satisfies both convergence criteria, but further optimisation is needed regarding convergence speed and applicability of the system. In practice the Q-Learning algorithm converges under different factors depending on the application and complexity. When deployed in a new environment, the agent should mostly explore and value immediate rewards, and then progressively show its preference for the discovered (near) optimal actions $\pi(s)$ as it is becoming more sure of its Q estimates. This can be achieved via the decay function shown below,

$$\varepsilon = \alpha = 1 - \frac{N_{tx}}{N_{decay}} \quad \text{for} \quad 0 \leq N_{tx} \leq N_{decay}, \tag{4.4}$$

where N_{tx} is the time since starting expressed as the number of transmitted packets in that period, and N_{decay} is a pre-set number of packets that sets the decay period.

This decay function is necessary to guarantee convergence towards the last known optimum policy in probabilistic systems such as the proposed contention-based MAC, since there is no known optimum final state. By reducing the values of ε and α over time via (4.4), the agent is forced to progressively focus on exploitation of gained experience and strive for a high long term reward. This way, when approaching the end of the decay period the found (near) optimal states (CW values) are revealed.

4.2.3 Initialising and Training the Controller

The strategy presented above can also be used to get instant performance benefits, starting from the first transmission. This can be done by pre-loading approximate controllers, pre-trained for different transmitted bit rates and number of neighbours using (4.4), to the agent's memory. These controllers define an initial policy that positively biases the search and accelerates the learning process. The agent's objective in this phase is to quickly populate its Q-table with values (explore all the state-action pairs multiple times) and form an initial impression of the environment. The lookup table (Q-table) is produced by encoding this knowledge (Q-values) for a set period of N_{decay} a priori and can be used as an initial approximate controller which yields an instant performance benefit since the system is deployed.

Q-Learning is an iterative algorithm so it implicitly assumes an initial condition before the first update occurs. Zero initial conditions are used the very first time the algorithm is trained on a set environment, except from some forbidden state-action pairs with large negative values, so it does not waste iterations in which it would try to increase/decrease the CW level when it is already set on the upper/lower limit. The algorithm is also explicitly programmed to avoid performing these actions on exploration. The un-trained, initial Q-table is set as in (4.5), where the rows represent the possible states - CW sizes and columns stand for the action space.

$$
\mathbf{Q_0[7][3]} = \begin{array}{c}
\mathbf{CW} \\ \mathbf{3} \\ \mathbf{7} \\ \mathbf{15} \\ \mathbf{31} \\ \mathbf{63} \\ \mathbf{127} \\ \mathbf{255}
\end{array}
\begin{array}{c}
\begin{array}{ccc} {}^{(CW-1)}/_{2} & CW & CW \times 2 + 1 \end{array} \\
\begin{pmatrix}
-100 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & -100
\end{pmatrix}
\end{array}
\tag{4.5}
$$

For even faster adaptation to environment changes, every agent can train and employ different controllers for every sensed density and received bit rate combination. The station has the ability to sense the number of one-hop neighbours since they all transmit heartbeat, status packets periodically. It also does not have the memory constraints that typical sensor networks have. An example of a controller's table at the end of the ε decay period as in (4.4) can be seen in (4.6). The controller was trained with $\gamma = 0.7$ and a decay period lasting for 180 s in a 60-car network. A trajectory leading to optimum/near-optimum CW/s is being formed (depending on past experience) by choosing the maximum Q-value for every CW-state, seen in bold font. The controller in (4.6) oscillates between the values 31 and 63 when exploiting the Q-table to find the optimum CW.

$$\mathbf{Q}^{\pi}[\mathbf{7}][\mathbf{3}] \approx \begin{array}{c c} \mathbf{CW} & \begin{array}{ccc} {}^{(CW-1)}/_{2} & CW & CW\times 2+1 \end{array} \\ \begin{array}{c} \mathbf{3} \\ \mathbf{7} \\ \mathbf{15} \\ \mathbf{31} \\ \mathbf{63} \\ \mathbf{127} \\ \mathbf{255} \end{array} & \begin{pmatrix} -100 & -0.0721847 & \mathbf{0.238827} \\ -0.0754744 & -0.0325265 & \mathbf{0.67485} \\ 0.19809 & 0.28012 & \mathbf{0.816807} \\ 0.289607 & 0.298519 & \mathbf{0.491713} \\ \mathbf{0.494494} & 0.101152 & 0.283855 \\ \mathbf{0.204304} & -0.0551011 & -0.0217622 \\ \mathbf{0.174506} & -0.867557 & -100 \end{pmatrix} \end{array} \tag{4.6}$$

4.2.4 Online Controller Augmentation

While a pre-trained, approximate controller is useful for speeding up the learning process as well as getting an instant performance benefit, its drawback is that it is less useful for adapting to change in the environment while on-line. The on-line efficiency of the Q-Learning controller depends on finding the right balance between exploitation of the station's current knowledge, and exploration for gathering new information. This means that the algorithm must sometimes perform actions other than dictated by the current policy, to update and augment that controller with new information.

While the station is online, exploratory action selection should be less frequent (i.e., $\varepsilon = 0.05$) than in a-priori learning (ε starts from 1), and is there primarily to compensate for modelling errors in the approximate controller. This means that the controller in its online operation uses the optimum Q-value 95% of the time, and makes exploratory CW perturbations 5% of the time in order to gain new experience. In this way the agent still has the opportunity to correct its behaviour based on new interactions with the VANET and corresponding rewards.

If sudden changes occur to the networking environment experienced by a station, which can be judged by incoming traffic., then it can invoke the ε-decay function (4.4)) again. Vehicle-stations entering an new unexplored VANET environment could also request approximate MAC controller Q-tables by their peers.

4.3 Implementation Details

4.3.1 Reward Function and Algorithm

In RL, the only positive or negative reinforcement an agent receives upon acting so that it can learn to behave correctly in its environment, comes in a form of a scalar reward signal. Taking advantage of the link capacity for maximum packet delivery (throughput) was of primary concern for this design, aiming to satisfy the requirements of V2V traffic (frequent broadcasting of kinematic and multimedia information). For this purpose, the reward function is based on the success of these transmissions. Reward r can be either 1 or -1 for successful (ACK) and failed transmissions (no ACK) correspondingly. A successful transmission from the same consecutive state - CW is not given any reward. The following pseudo-code summarizes our proposed protocol.

Algorithm 1 Q-Learning MAC for IEEE 802.11p

Initialize $Q_0[CW, A]$ at $t_0 = 0$

$CW_0 = CW_{min} = 3$ at $t_0 = 0$

if $N_{tx} < N_{train}$ **then**

  $\varepsilon, \alpha \leftarrow decay\,function$ ▷ according to rule (4.4)

else

  $\varepsilon, \alpha \leftarrow constant$

end if

procedure ACTION_SELECTION(CW_t) ▷ ε-greedy

  if $p_\varepsilon \leq \varepsilon$ **then**

    $a_{t+1} \leftarrow random({}^{(CW_t - 1)}/_{2}, CW_t, CW_t \times 2 - 1)$

  else if $p_\varepsilon \geq 1 - \varepsilon$ **then**

    $a_{t+1} \leftarrow a_\pi$

  end if

  $CW_{t+1} \leftarrow CW^{a_{t+1}}$

end procedure

```
procedure SEND(TxPacket, SeqID, CW_{t+1})
    TxPacket.OriginId ← SeqID
    RTT ← 0 s
    Content(CW_{t+1})
    CW_{t-1} ← CW_t
    CW_t ← CW_{t+1}
end procedure

procedure FEEDBACK(CW_t, CW_{t-1},RxPacket)
    if RxPacket.OriginId=TxPacket.SeqId AND RTT =< 0.1 s then
        if a_t ≠ (CW_t ⟵ CW_{t-1}) then
            r_t ← 1
        end if
    else if RTT > 0.1 s then
        r_t ← −1
    end if
    update Q(CW_{t+1}, a_{t+1})                ▷ according to rule (4.1)
    Action_selection(CW_t)
end procedure
```

4.3.2 Implicit ACKs

To add reliability in OCB transmissions, we employ implicit ACKs originating from the application layer and piggybacked in normal packets. This is a proven technique explored in various publications [35], [90] [75] [63] to add reliability to broadcast (OCB) transmissions in VANETs. Additionally, because these ACK packets originate from the application layer, they do not block other transmissions in the queue of the original transmitter until they are received or a timer expires, as it happens on the classic unicast MAC-level implementation. This feedback technique is more appropriate for the safety-critical and less delay-tolerant exchanges happening in VANETs, as seen in [47].

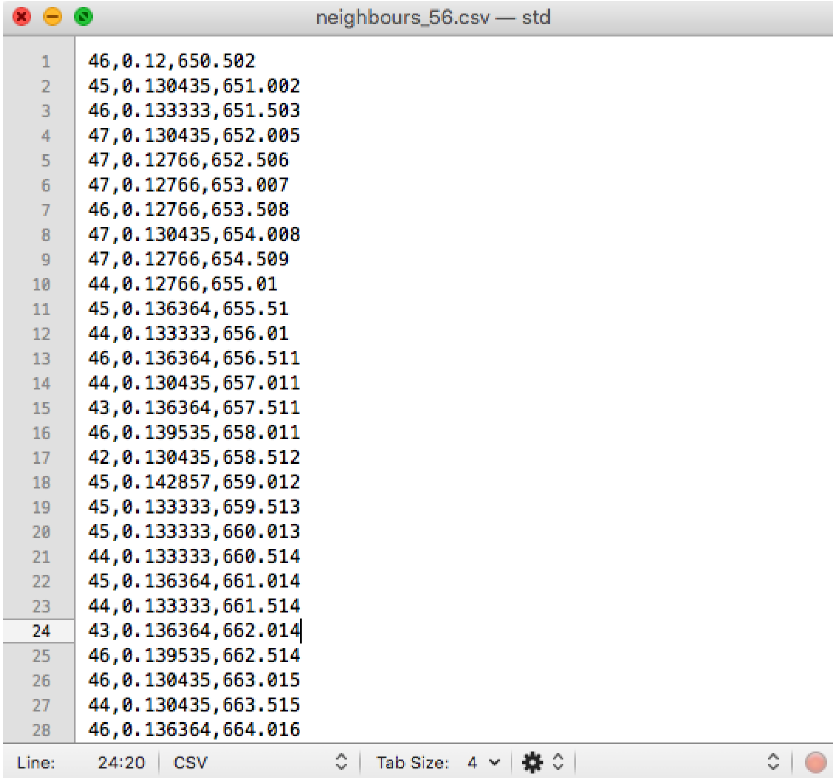
We also exploit their usefulness as feedback to apply MAC techniques that can adapt the *CW* parameter appropriately to satisfy different kinds of V2V traffic as efficiently as possible. In our simulations, to keep the implementation practical and fair we use said forwarded packets as ACKs, in both single-hop and multi-hop systems. Since VANETs are to enable exchanges of up-to-date kinematics-related data [32], the observation (feedback) delay, or the RTT (time between transmission of a packet with set CW value and ACK reception for that packet) should be below the packet generation period (i.e., 100 ms [86]), or else it is considered as out-of-date.

4.3.3 Broadcast Storm and Dissemination Mechanism

In order to increase the reliability of V2V broadcast communications, some vehicles need to serve as relays and rebroadcast messages that are received so that other stations that are not within range of the original transmitter can get the messages. The modified intelligent MAC layer designs under study can be used with any forwarding protocol [112] [69], with the simplest method being a constant forwarding probability i.e., $P_{fwd} = 10\%$. This would mean that a station will rebroadcast $P_{fwd} \times N_{received}$ additional packets, and although can ensure the packet is forwarded towards all directions, could impose a larger load than needed on the network and the MAC Layer. If the probability becomes too small, then forwarding could be unreliable, or limited coverage would be achieved.

However, the broadcast storm phenomenon will occur if there are excessive rebroadcast messages in the vehicular network. To remedy this problem, a method of rebroadcasting received messages should be designed in a way that reduces redundant retransmissions as much as possible, while the information reaches all the vehicles that it should. In the presented simulations, a more efficient way to have restricted probabilistic rebroadcasting is employed (implementation shown in Appendix 1). In order to restrict the number of forwarded packets for the reasons mentioned above (avoid network saturation, collect fair measurements), we use (4.7), which assigns each vehicle-transmitter a forwarding probability depending on the number of potential forwarding candidates (nearby IEEE 802.11p equipped vehicles) and the number of retransmissions/packet required, as seen in 4.2.

Three main ideas on a vehicle-station realistically and accurately estimating the number of other neighbouring vehicles (which are within communication range), are found in literature. To start with, the network density that is experienced by a vehicle can be found via beacon packets [11] [83]. Each vehicle delivers its speed and position to other vehicles. Thus, a vehicle can have a sense of how many neighbours are within communication range from packets containing some sort of identification. Another method is found in [9], where the network density is estimated based on the number and length of stops the vehicle makes. The more often the car stops, and the longer it stands after, the greater the density. This method can work in a transportation system where all vehicles would have DSRC capability, so that the network traffic would be proportional to nearby road



neighbours_56.csv — std

```
46,0.12,650.502
45,0.130435,651.002
46,0.133333,651.503
47,0.130435,652.005
47,0.12766,652.506
47,0.12766,653.007
46,0.12766,653.508
47,0.130435,654.008
47,0.12766,654.509
44,0.12766,655.01
45,0.136364,655.51
44,0.133333,656.01
46,0.136364,656.511
44,0.130435,657.011
43,0.136364,657.511
46,0.139535,658.011
42,0.130435,658.512
45,0.142857,659.012
45,0.133333,659.513
45,0.133333,660.013
44,0.133333,660.514
45,0.136364,661.014
44,0.133333,661.514
43,0.136364,662.014
46,0.139535,662.514
46,0.130435,663.015
44,0.130435,663.515
46,0.136364,664.016
```


Figure 4.2: A station (with Id=56) perceives the number of one-hop neighbours (Column 1) in a multi-hop topology of 100 stations, and calculates P_{fwd} on the fly (Column 2) every 500 ms (as seen in Column 3).

traffic. The third method is estimation of the experienced network density based on the time a DSRC unit finds the wireless channel to be occupied by some other transmission. The correlation between the channel busy ratio perceived by a station and the number of its neighbouring stations has been studied in [10].

For this implementation, each vehicle periodically keeps track of the numbers of other transmitting vehicles nearby, via different IDs in incoming messages. The refresh rate for this does not need to be very small, as the estimation of forwarding candidates could be accurate by updating it every 0.5 s. Even if we assume that two vehicles move at opposite directions with 100 km/h, the relative velocity among them is 55.56 m/s, and with a beaconing period of 0.1 s the displacement among them is a bit less than 6 meters for every packet they both exchange. So even in the worst-case scenario of high speeds, with 1 km theoretical max TX range, every station should be aware of the number of its relevant neighbours (within range) with a high accuracy every i.e., 0.5 s. An exact number is not needed, but the more accurate the estimation, the lower the redundancy of forwarded packets.

By employing probabilistic retransmissions, we can also have feedback (ACK) regarding the outcome of transmissions in a broadcast environment. When the retransmissions are overheard from the original transmitter, they act as non-blocking ACKs originating from the application layer, which should be the de-facto ACK method in VANETs as suggested in [47]. This means that the proposed MAC protocols can be applied for purely unicast transmissions but they can also comply with the IEEE 802.11p specification which primarily operates in OCB mode to allow one-to-many information exchanges.

So considering a network with density $N_{vehicles}$ with each station generating data packets with constant rate f_{gen}, the receivers can calculate the forwarding (ACK) probability P_{fwd} in real time according to

$$P_{fwd} = \frac{N_{fwd}}{N_{vehicles}} \tag{4.7}$$

by detecting the number of relevant nearby active transmitters via the incoming packets containing the node IDs and the number of hops, so as to consider only immediate neighbours and disregard packets received from multi-hop paths (retransmissions). That way the effective beaconing frequency can be calculated as follows

$$f_{beacon} = f_{gen} \times (1 + N_{fwd}). \tag{4.8}$$

Consequently we can get the maximum theoretical network-wide throughput as seen below

$$T_h = N_{vehicles} \times f_{beacon} \times L_{packet} \times 8\,bit, \tag{4.9}$$

which gives us 3.072 Mbit/s for 50 transmitting stations sending $L_p = 256\,$byte packets with $N_{fwd} = N_{ACK} = 2$, which is chosen so that an ACK will be received with higher confidence since the packet delivery probability in studied systems is less than 1.

4.4 Performance Evaluation of the Q-Learning-based MAC protocol

4.4.1 Simulation Setup

The achieved improvement on link-level contention was of primary concern, so a multitude of tests were run for a single hop scenario, with every node being within the range of the others. A multi hop scenario is also presented, which makes the hidden station effect apparent in the performance of the network. Additionally, by setting an infinite queue size, packet losses from collisions can be accurately measured.

We set $R = 12\,$Mbit/s so that the channel does not bottleneck even the denser network scenario it terms of data traffic evaluated in the presented simulations [94]. Every

Parameter	Value
Simulation time	300 s
Training period N_{train}	1800 packets
Channel Frequency	5.89 GHz
Channel Bandwidth	10 MHz
Transmission rate R	9, 12 Mbit/s
Transmission power	Single-hop: 30 dBm Multi-hop: 17 dBm
Backoff slot time	13 μs
Packet Generation Frequency f_{gen}	10 Hz
ACKs per packet N_{fwd}	Single-hop: 2 Multi-hop: 6
Effective Broadcasting Frequency f_b	(4.8)
Packet Generation Offset	0.005 s

Table 4.1: Simulation Parameters for Q-Learning-based MAC protocol evaluation.

station generates 10 packets/s, and also retransmits original packets received from others with a variable probability P_{fwd}, found every 0.5 ms by each station by (4.7) depending on the number of potential receivers. The retransmitted packets carry ACK packets that are needed for reliability purposes as well as feedback for the CW adaptation mechanisms, as explained in Section 4.3.2. In practice, an acknowledgement can be carried by any broadcasted packet, since most of the payload would still be utilised to enable other applications. In our implementation, they are just replicas of messages, so that we can collect fair measurements when approaching channel saturation. They are also used for forwarding purposes in multi-hop deployments. Additionally, the Veins 4.4 IEEE 802.11p implementation does not support unicast transmissions by default, which can be emulated with these probabilistic retransmissions.

4.4.2 Emulating the BEB algorithm for OCB transmissions

Veins focuses on the broadcast, OCB, IEEE 802.11p protocol stack which does not feature ACKs. Consequently, the IEEE 802.11p Veins implementation does not feature the BEB part of the DCF, since it relies on explicit ACK packets to adjust the $backoff$ parameter depending on whether a transmission was successful or not. For the purpose of comparison, we implemented a Pseudo-BEB CW adaptation mechanism based on feedback from non-blocking, application-layer ACKs on top of the IEEE 802.11p Veins implementation. The CW adaptation by a station in a 50-vehicle network using the Pseudo-BEB MAC layer can be seen in (4.3). The same for 100 vehicles can be seen in (4.4). As seen, the CW employed by a vehicle reaches higher levels because of the increased presence of collisions.

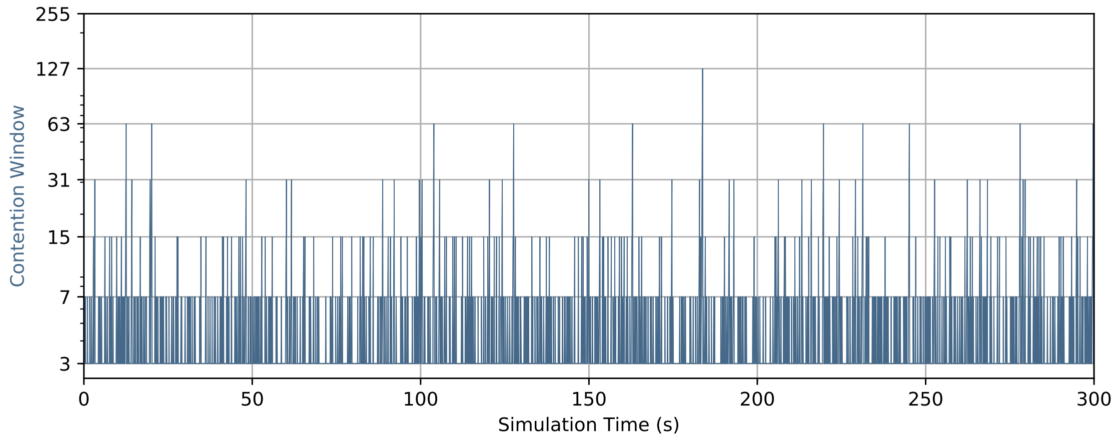


Figure 4.3: CW adaptation over time performed by our Pseudo-BEB implementation, deployed in a broadcast environment of 50 vehicles. For every detected collision, CW is doubled until it reaches a CW_{max} value.

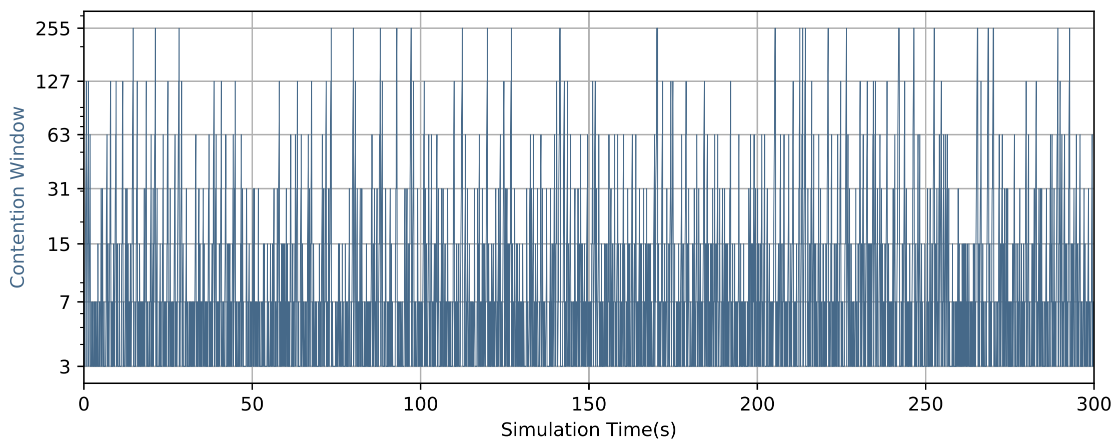


Figure 4.4: The Pseudo-BEB algorithm deployed in a network of 100 vehicles will set the CW on higher levels on average, because of the detected increased contention.

4.4.3 Optimising the RL algorithm's performance

Exploration rate ε

We evaluate the performance of different exploration policies regarding achieved packet delivery (throughput), since this is the optimization goal of them, as shown in Fig. 4.5.

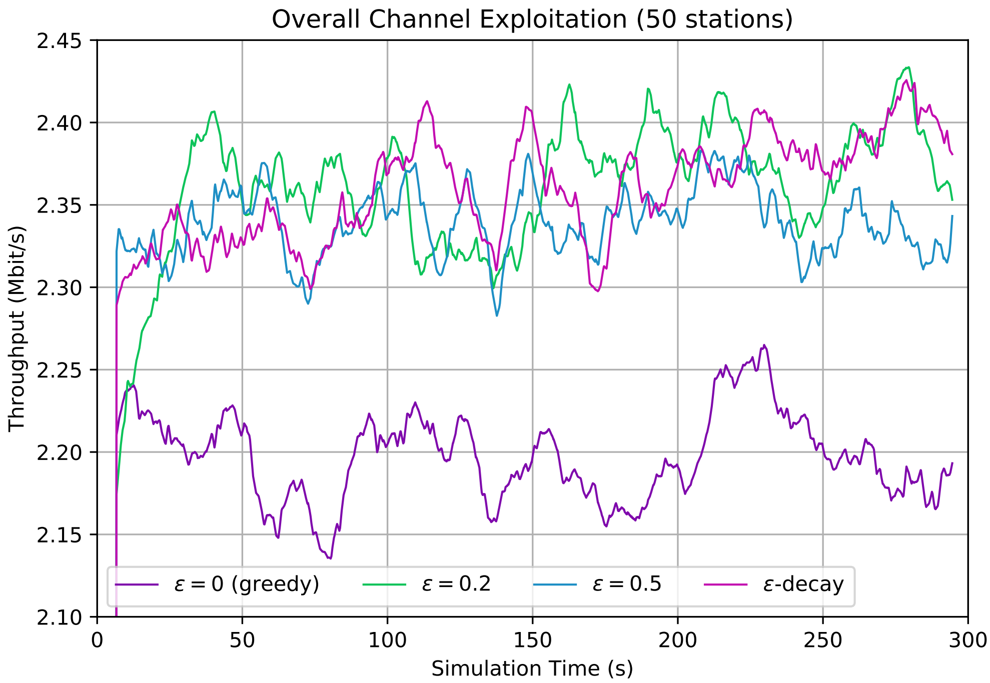


Figure 4.5: Q-Learning MAC protocol performance for different exploration-exploitation policies, with $\alpha = 0.5$

We first tested vanilla Q-Learning (greedy), which basically always uses the action found to be optimal at every state, not enforcing any exploration policy at all ($\varepsilon = 0$). This algorithm as seen can be stuck at a local maximum since it does not fairly explore all available action states (regarding the reward they return). The employed CW over time reveals the reason for this, since the algorithm does not explore all (s, a) pairs at the vanilla case, as seen in Fig. 4.6. Enforcing a maximum exploration rate $\varepsilon = 1$ has the opposite effect on the employed CW size, as observed in Fig. 4.7.

Then we tested a balanced exploration-exploitation ε-greedy policy, where the agent would take an exploratory (random) move 50% of the time and exploit its best-known option for 50% of the time ($\varepsilon = 0.5$). Although it raises the system throughput, we observe wide oscillations because of the increased exploration strategy. Limiting the agents' exploration to just 20% of the time ($\varepsilon = 0.2$) delays getting a performance gain but also

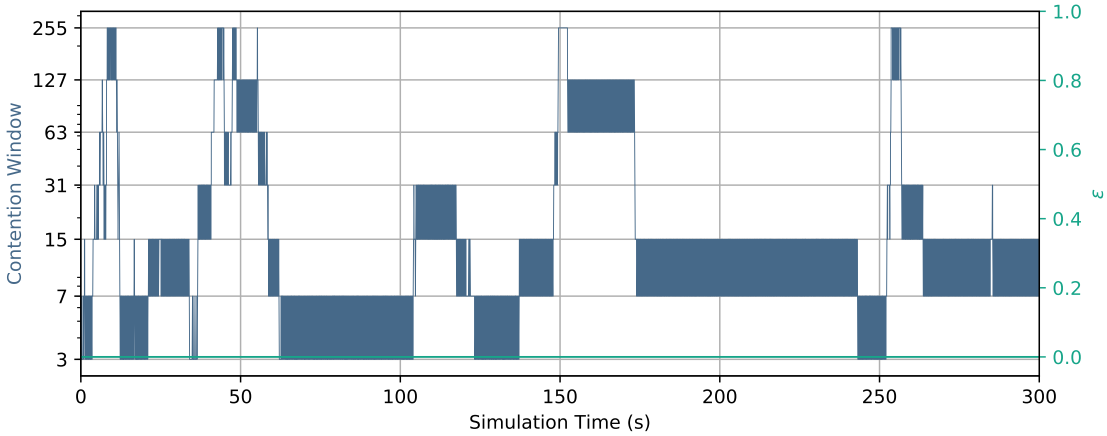


Figure 4.6: Trace of CW over time for a station for vanilla Q-Learning (greedy policy).

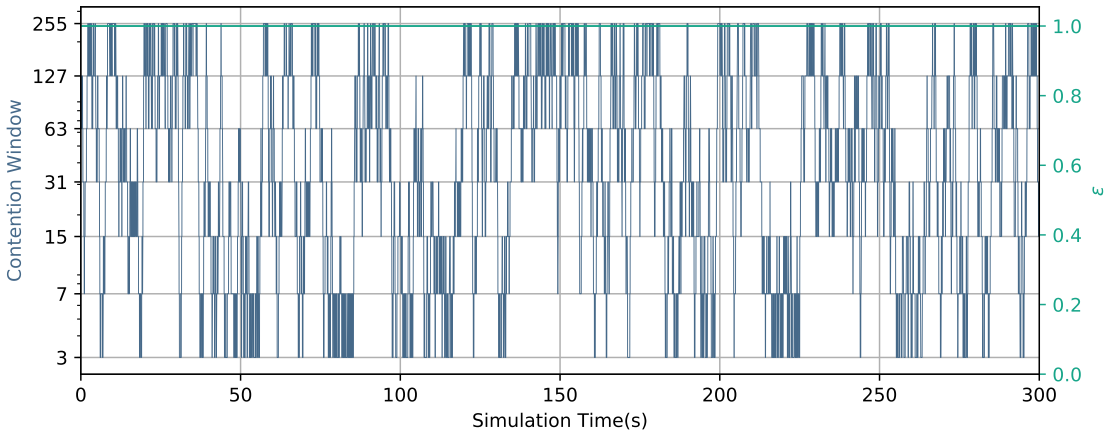


Figure 4.7: Trace of CW over time for a station. If constant $\varepsilon = 1$ is enforced, the Q-Learning algorithm can be used as a purely search algorithm, as it continuously explores all (s, a) pairs, but does not behave correctly as a controller.

mitigates the oscillation problem to some level and improves throughput performance after some training has happened. We observe a maximum difference of 100 kbit/s for a 50-vehicle scenario, but the difference will be more significant in denser scenarios with a higher system throughput. Additionally, although the mean throughput slowly increases, the oscillation is still significant enough after a period of 300 s. Given limited time for the Q-Learning algorithm to converge to a good solution, exploration policies with constant exploration ratios ε perform suboptimally, but reduced exploration ratios definitely assist convergence. An issue is that if we reduce ε too much, fair exploration of all (s, a) pairs is not guaranteed. The employed CW over time for the algorithm with $\varepsilon = 0.2$ can be seen in Fig. 4.8. The learning agent tends to use the upper CW levels after 150 s, and mostly uses $CW = 31/63$ after 270 s. Nevertheless, convergence time is long and many unnecessary exploratory (thus possibly suboptimal) actions are taken.

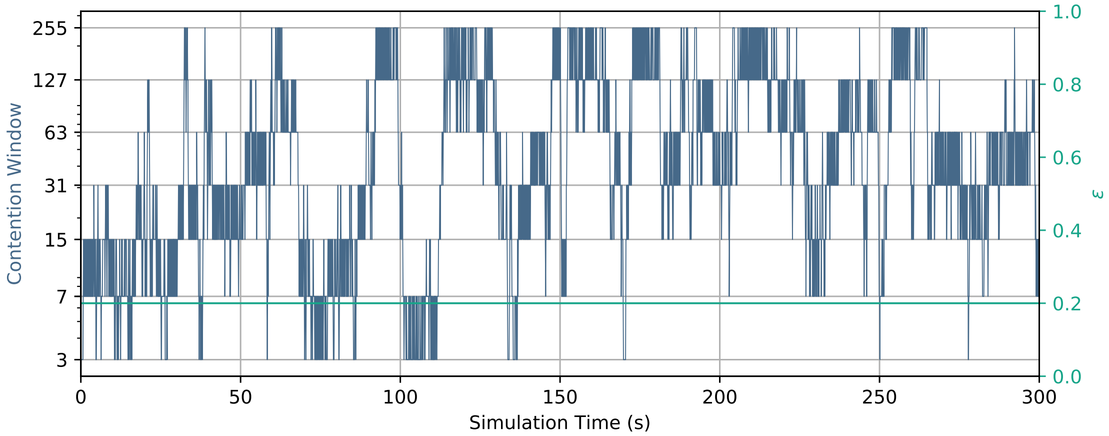


Figure 4.8: Trace of CW over time for a station with constant $\varepsilon = 0.2$. It constantly explores random (s, a) pairs 20% of the time, which delays convergence.

Finally we evaluate our suggestion to the convergence problem, which is the decaying-ε-greedy strategy presented in Section 4.2.2 that forces a lot of exploration early on and quickly restricts ε to small values to force convergence to the best known (s, a), while continuing to correct the agent's behaviour. As seen in Fig. 4.5 after 50 s, with ε being already below 50%, the algorithm keeps raising mean throughput. The oscillation after 150 s is much narrower than other strategies since ε is kept small at 10%, and throughput keeps increasing until the end of the simulation. The employed CW over time for the algorithm with a decaying ε can be seen in Fig. 4.9.

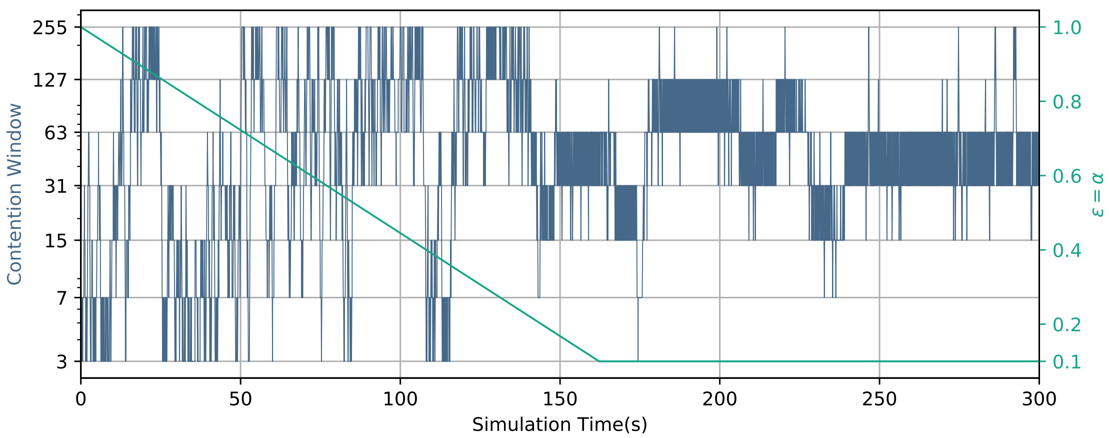


Figure 4.9: Trace of CW over time for a station using the proposed $\varepsilon - decay$ solution.

Learning rate α

After we have decided on the ε-decay technique, we proceed to an investigation of the effect the α parameter has on the algorithm's performance. The discount rate γ is set to 0.7. Results of said investigation are depicted in Fig. 4.10. It can be observed that reducing both ε and α quantities simultaneously yields the best performance regarding maximum achieved throughput, as well as far fewer oscillations after $\approx$140 s. It makes sense that the α parameter should decrease as well as time passes, since this way the algorithm is forced to have more "confidence" in its so-far acquired knowledge, and avoiding overriding information as often (just 10% of the time in this experiment).

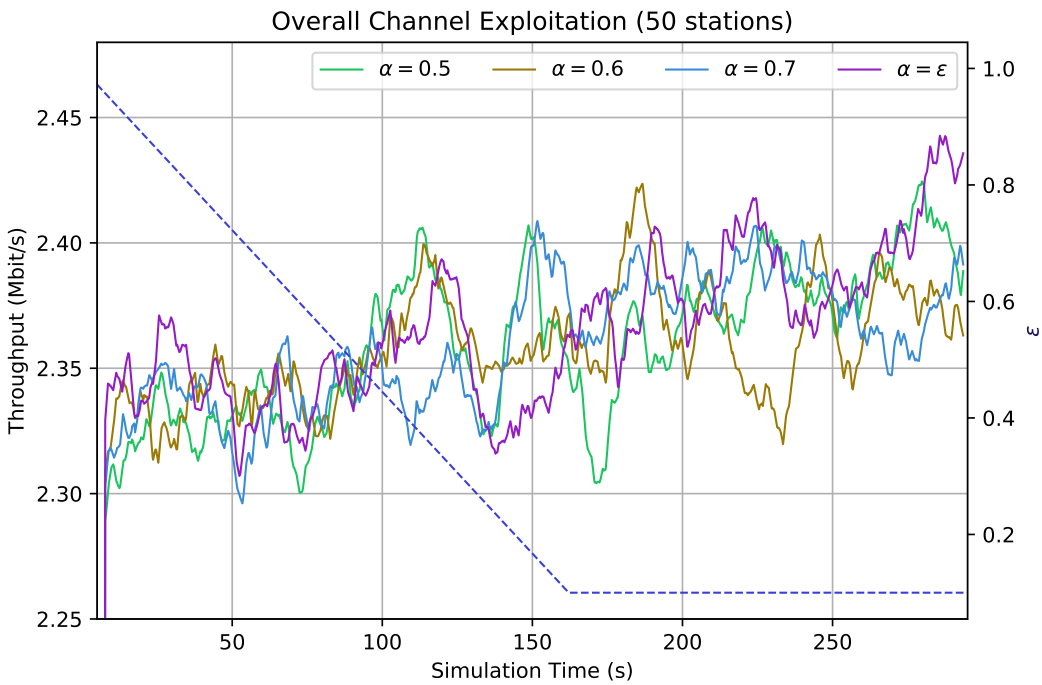


Figure 4.10: Q-Learning MAC protocol performance for different values of α

Reducing the value of α over time via the decay function (4.4) (also used for ε decay over time), essentially forces the agent to progressively limit the rate of overriding the existing experience by newly acquired rewards. This way, the so-far found (near) optimal states-CW/s are revealed as the agent becomes more confident in its so-far gained experience as time progresses, and behaves better as a controller avoiding large oscillations around the optimal CW value.

Discount rate γ

Most algorithms for solving Markov Decision Processes rely on a discount factor γ, which ensures their convergence. It is generally assumed that using an artificially low discount factor will improve the convergence rate, while sacrificing the solution quality. But this is not necessarily true in all cases [73], and typically this parameter is treated as part of the problem, and needs to be tuned for a given problem by trial and observation (or appropriate heuristics if available or obvious to the designer). The appropriate value of γ for a given problem correlates with the reward function, the size of the search (s, a) space as well as the exploration policy (in this case the ε-decay period). The values typically used for this parameter range from 0.6 to 0.99 [31].

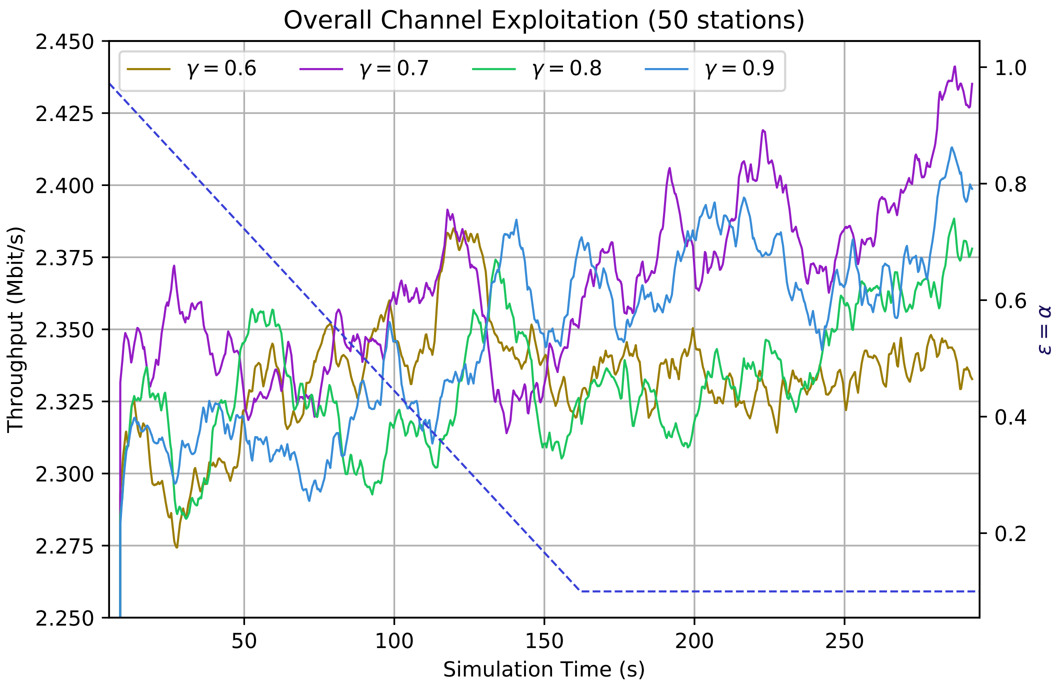


Figure 4.11: Q-Learning MAC protocol performance for different values of γ

We proceed to an investigation of the effect the γ parameter has on the algorithm's performance, as shown in Fig. 4.11, using the ε and α-decay accelerated exploration technique present previously. In the presented scenario, it can be observed that a discount factor of $\gamma = 0.7$ yields a better result sooner and finds the best solution over time among examined values.

Time of exploration N_{train}

The required time of training $T_{train} = \frac{N_{train}}{f_g}$ (where f_g is the packet generation frequency as shown previously), or period of intense exploration, was also of concern. For often-changing, safety-oriented networks such as VANETs, T_{train} should be as small as possible, as long as it does not significantly compromise the algorithm's convergence (converging around a sub-optimal CW size because of insufficient exploration). The algorithm's performance regarding achieved throughput is evaluated for different training times.

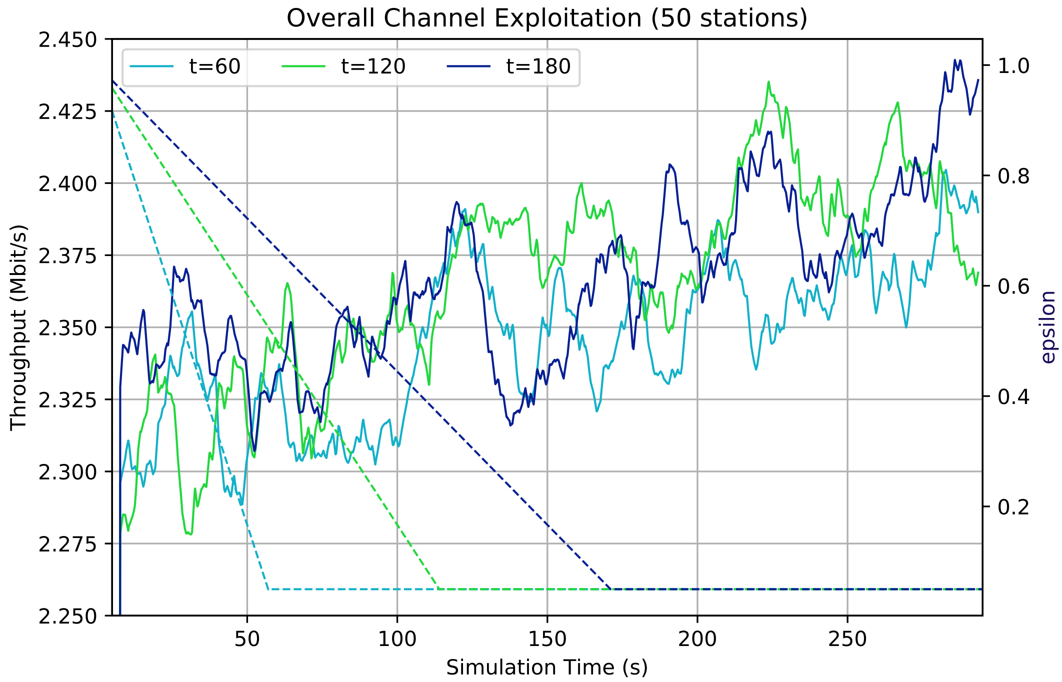


Figure 4.12: Q-Learning MAC protocol performance for different training times T_{train}

In Fig. 4.12 it can be observed that increasing the "training", or exploration time can give system designers larger confidence in the result. As made apparent in results for a 50-vehicle scenario, if needed the algorithm can "train" for shorter times and still improve achieved performance, with the downside of sacrificing the quality of solution.

4.4.4 Evaluation for different types of network traffic

In order to evaluate the performance of our novel proposed protocol in comparison to the existing IEEE 802.11p MAC method, simulations are carried out using the same setup as before. We evaluate the Q-Learning MAC protocol against different types of congestion. Packets lost are not recovered since we are concerned with the performance of the Link Layer.

The discount rate γ is set at 0.7. We use the α, ϵ-decay method for training the MAC agents since it forces them to explore all action-state pairs faster early-on, and then focus on the most promising trajectory. The simulation run time for the proposed MAC protocol consists of two stages, as seen previously in Fig. 4.9. First is the controller intense exploration stage, which lasts for $N_{decay} = 1800$ generated packets (or 180 s with $f_{gen} = 10\,\mathrm{Hz}$), during which the ε, α parameters keep decreasing until they reach $\varepsilon_{min} = \alpha_{min} = 0.05$. Then follows the evaluation or online period which lasts for 120 s, in which the agent acts according its acquired knowledge for 95% of the time. During this time, we benchmark the effect of the trained controllers regarding network performance as well as keep "learning" (5% of the time) for controller augmentation. For IEEE 802.11p simulations, only the evaluation stage is needed, which lasts for the same time.

Benchmarked Protocols

- **IEEE 802.11p:** It is the baseline protocol operating in OCB (broadcast) mode with fixed $CW = CW_{min} = 3$ [highest priority-AC3] or $CW = CW_{min} = 15$ [AC0 and AC1], since these enable lower-latency transfers. It has no CW adaptation capability.

- **Pseudo-BEB:** The addition of retransmissions originating from the receivers' application layer allowed us to emulate the WiFi Binary Exponential Backoff algorithm for the IEEE 802.11p MAC and compare the novel Q-Learning protocols against it in a fully broadcast, OCB system, as well as emulate unicast transmissions.

- **Q_MAC:** The proposed MAC protocol featuring an intelligent CW adaptation solution based on the Q-Learning algorithm.

Effect of Increased Network Density

In VANETs, the network density changes depending on location and time of the day. We test the performance of the novel MAC against the standard IEEE 802.11p protocol for different number of cars. We evaluate the scalability of the MAC protocol by using it on vehicles travelling in the simulated map described previously. Large enough transmission power enables a simulated scenario without Hidden Terminals. The packet size L_p is 256 bytes, and the packet generation frequency f_{gen} is set at 10 Hz. Fig. 4.14 shows the increase of successfully delivered packets when using our novel MAC protocol. When using the standard IEEE 802.11p, PDR decreases in denser networks due to the increased collisions between data packets. The proposed MAC is designed to adjust the size of CW as needed to achieve maximum packet delivery.

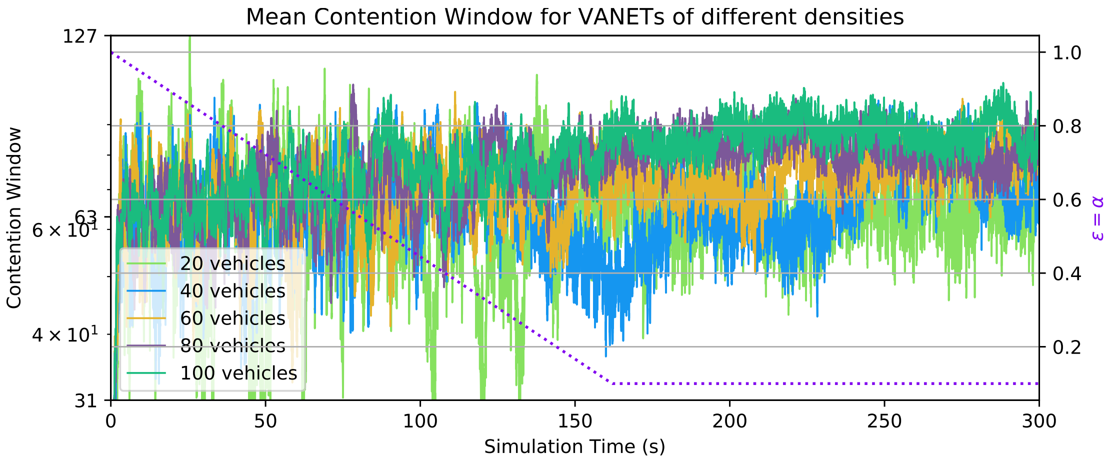


Figure 4.13: Average network-wide CW for VANETs of different node densities

The IEEE 802.11p MAC performance assessment found already that in dense VANETs, where network congestion and consequently packet collisions are increased, a higher CW level used by all agents can mitigate the packet drop problem and improve the agents' transmission success ratio. The behaviour of the intelligent protocol proposed in this chapter regarding CW adaptation for different network densities can be observed in Fig. 4.13. It can be seen that in this symmetrical VANET case, the protocol detects that larger CW values should be used by all stations to accommodate traffic in denser networks.

Network-wide PDR for the proposed Q-Learning MAC is measured after the initial, more exploratory phase with $\varepsilon > 0.05$ (after the agent has gained some experience), as seen in Fig. 4.14. The agent still explores random actions 5% of the time ($\varepsilon = 0.05$), for better adaptability and augmentation of the built controller (Q-table). In sparse networks, the proposed algorithm still outperforms the rest of the protocols, but the difference from the minimum CW is not as significant as it can be when the density increases. When the network density exceeds 40 cars, the proposed learning MAC performs much better regarding successful deliveries. A $\approx$ 40.5% increase in performance (packets delivered) over the fastest AC of the DSRC MAC is observed in a network formed of 80 cars when

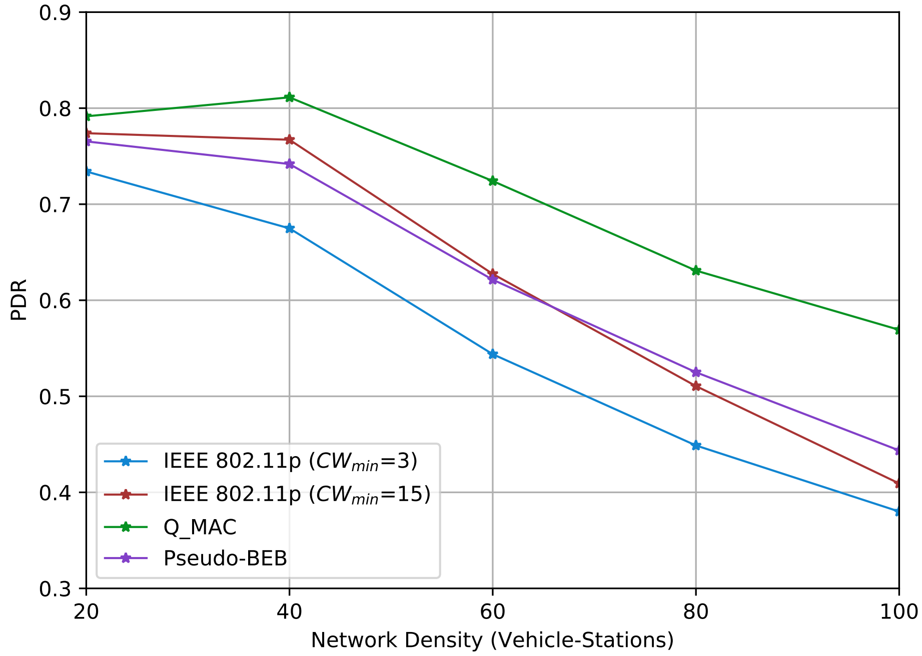


Figure 4.14: PDR versus network density for periodic broadcasting of 256-byte packets.

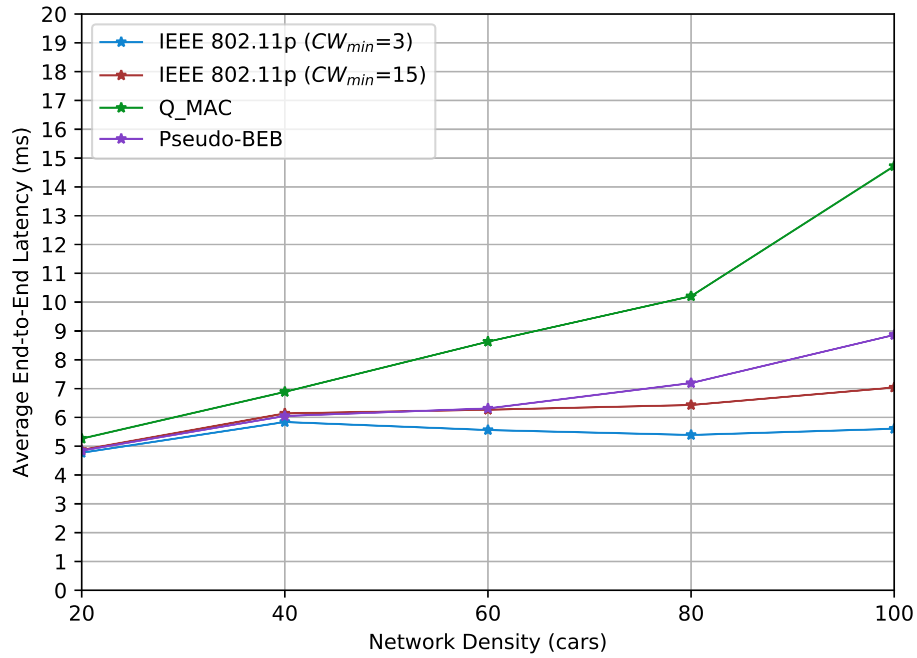


Figure 4.15: Mean end-to-end latency for successful transmissions versus network density for broadcasting of 256-byte packets.

using the proposed modified DSRC MAC with the self-learning $backoff$ mechanism. For a network of 100 cars this increase in PDR reaches $\approx 51\%$.

The recorded end-to-end delay for successful transmissions, shown in Fig. 4.15 indicates that the protocol can raise latency, especially for denser scenarios. This is expected as the increased CW enforced by the learning MAC method when traffic is increased adds to the channel access time and consequently the overall transmission time. The worst case scenario simulated is for 100 simultaneous transceivers within the immediate range of each other, in which the mean transmission latency is increased by almost 9 ms, raising the total measured mean latency to 14.8 ms, when using the Q-Learning MAC instead of the fastest AC of the default DSRC MAC.

Effect of Data Rate

We examine the performance of both the standard and enhanced protocol for different data rates. PDR is measured for a network of 50 nodes without hidden terminals. The broadcasting frequency is set at $f_b = 10\,\text{Hz}$, and the packet size L_p is varied from 64 bytes to 512 bytes. It can be observed that for 512-byte packets the mean PDR achieved by the stations featuring the Q-Learning MAC protocol increases by 66.47% over using the default VANET MAC. The gain increases when transmitting 1024-byte packets. It is clear that for larger packet transmissions the Q-Learning based protocol will be much more reliable, as seen in Fig. 4.16.

Effect of Multi-hop

In a network without a fixed topology which has to cover a large area, the most common way to disseminate information is to forward packets across the network. In VANETs, DSRC-enabled vehicles can cooperate to deliver data messages via multi-hop paths, without the need of centralized administration. Some vehicles upon reception of an original message will operate as relays, using some forwarding method such as the probabilistic retransmission mechanism we employ in this study. In this scenario we test the performance of the proposed protocol when attempting to reach stations that are located up to two hops away. We can evaluate performance for two-hop transmissions by reducing the transmission power to 17 dBm so that not all vehicles in a simulation can reach each other. As observed in Fig. 4.17, when the network density increases, the proposed MAC still offers a valid delivery benefit for vehicle-stations contending for access on the same DSRC channel.

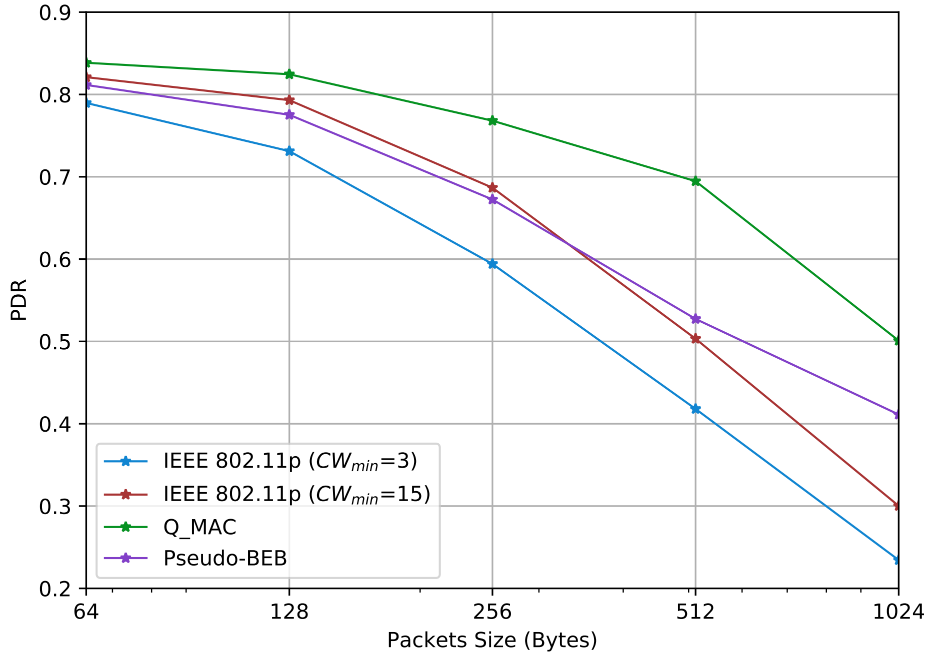


Figure 4.16: PDR versus packet size for 50 vehicles broadcasting with $f_b = 10\,\text{Hz}$

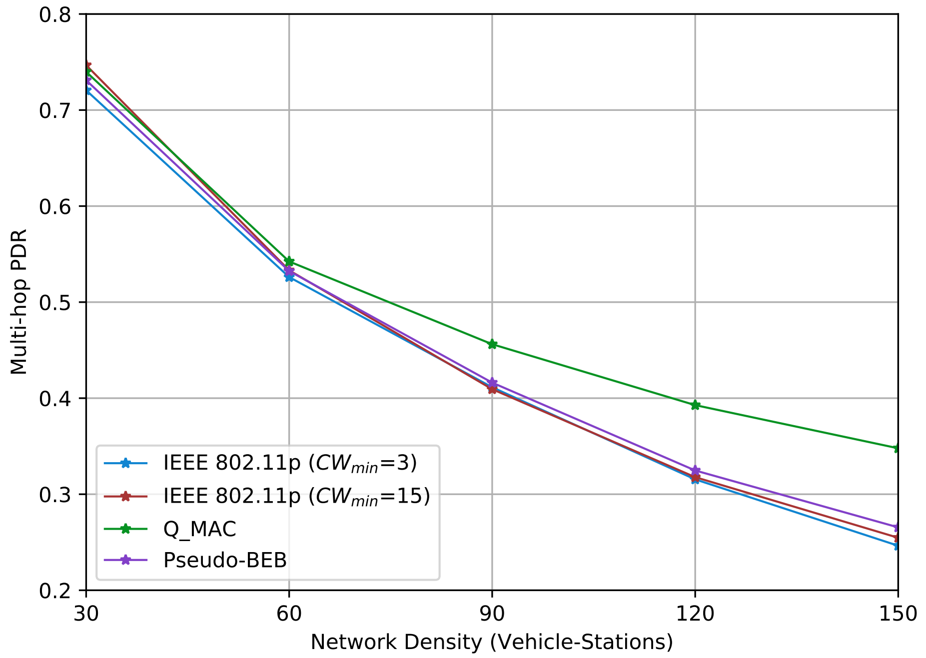


Figure 4.17: PDR versus network density for broadcasting and relaying 256-byte packets in two-hop network

4.5 Q-Learning MAC agents in saturated networks

The proposed Q-Learning MAC algorithm behaves greedily, meaning that every station tries to optimise its own throughput performance on the fly, by adapting the CW value as needed. As already seen in this chapter, in dense VANETs where the learning algorithm is utilised by all vehicles, the vehicles increase the size of CW enough so that the probability of collisions by simultaneous transmissions is reduced. This aligns the findings from Section 3.2.3, which confirm that all stations using the same, large CW value increases the total system throughput and promotes symmetry in the network regarding packet deliveries from all stations.

On the other hand, in a network where the majority of stations use a large CW size under high network traffic, the rest of vehicle-stations can exploit the way CSMA operates to win contention over them by reducing their CW size, as seen in Section 3.2.3. We reproduce a saturated VANET traffic scenario. The channel data rate is set at 9 Mbit/s, as previously. This leaves an available bandwidth of 180 Kbit/sec per station, for 50 stations if bandwidth was equally allocated. As before, all $N_{vehicles} = 50$ stations are generating $L_{packet} = 512$-byte packets with $f_{gen} = 50/3\,\text{Hz}$ (beaconing interval is $I_{beacon} = 0.06\,s$ in OMNeT++), translating to an effective beaconing frequency of $f_b = 50\,\text{Hz}$ for $N_{fwd} = 2$, from (4.8). Also $4/5 \times N_{vehicles}$ use a constant $CW_{min} = CW$ value of 255. The purpose of this experiment is to examine whether the Q-Learning MAC algorithm can validate the findings of Section 3.2.3. We deploy the adaptive learning MAC protocol on the rest of the stations ($N_{learning} = 10$), and observe their mean CW over time, seen in Fig. 4.18. As expected, the algorithm correctly detects the effect of CW in such a scenario, with stations settling at $CW_{mean} \approx [63, 127] < 255$.

This means that the learning MAC protocol employs smaller CW sizes to improve packet delivery performance. This is different to the scenarios presented previously, in which the algorithm attempts to maximise the stations' communication performance by increasing the CW parameter. It is worth noting that, as found in Section 3.2.4, in such a high traffic scenario the greedy stations will still tackle more collisions if they use a CW that is large enough, but below the one employed by their competitors (<255). This is validated when using the Q-Learning MAC algorithm for the greedy stations, as seen below in Fig. 4.19.

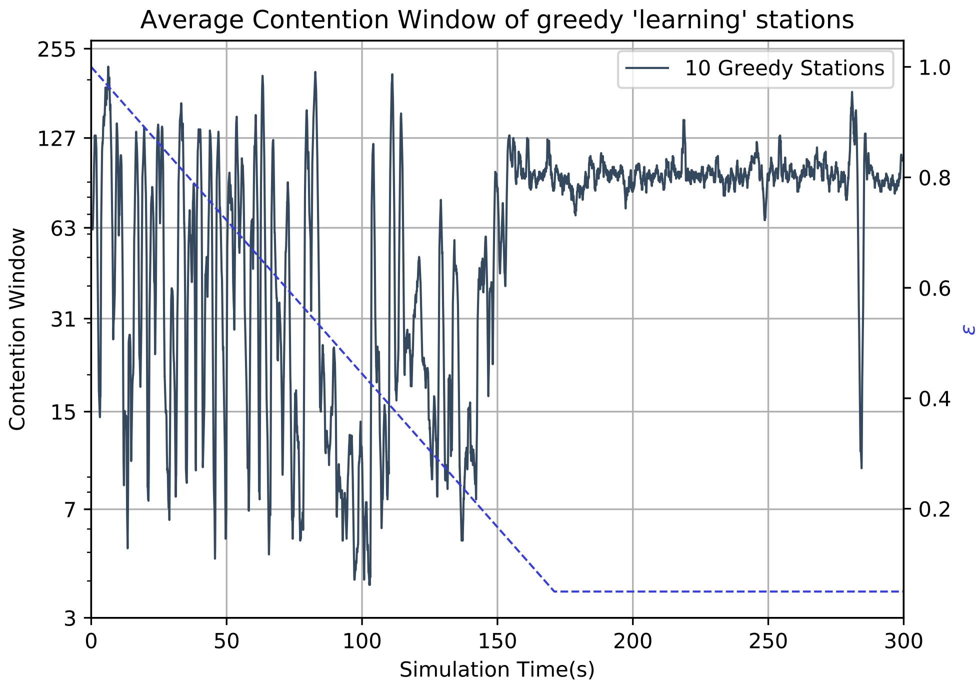


Figure 4.18: Average CW over time for the 10 stations featuring the Q-Learning-based MAC, deployed in a saturated network of 40 existing nodes using the IEEE 802.11p MAC with fixed $CW = 255$

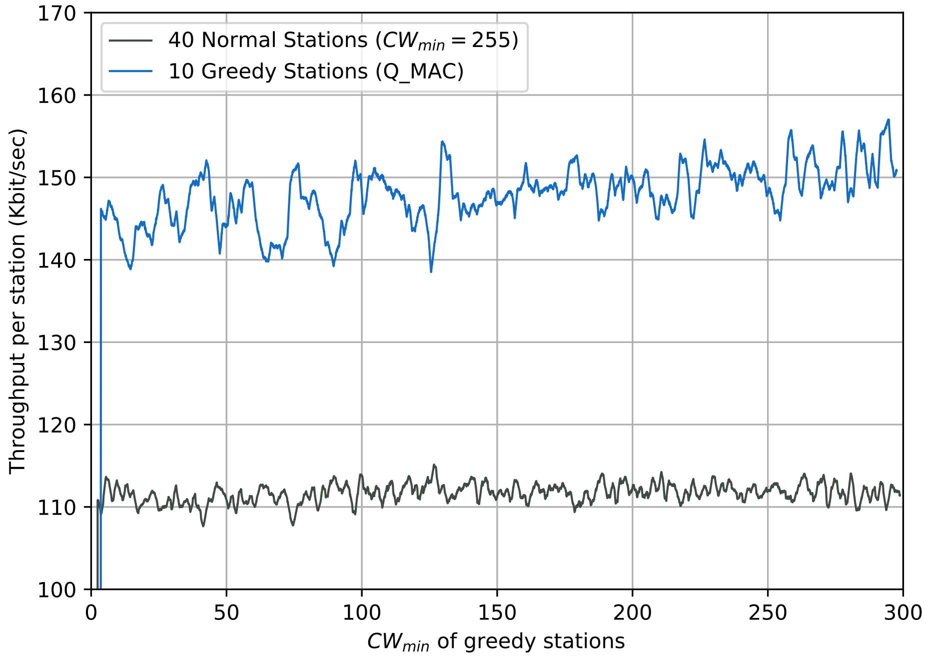


Figure 4.19: Mean throughput of stations in a saturated network, depending on their employed MAC layer.

4.6 Summary

We have introduced a contention-based MAC protocol for V2V broadcast transmissions that relies on Q-Learning to maximise throughput via CW size adjustments by continuously interacting with the network. We developed simulations to demonstrate the effectiveness of the algorithm when utilised to implement a wireless MAC protocol. First we demonstrated the effect that the learning rate α, discount rate γ and exploration rate ϵ have on the effectiveness of the algorithm in terms of network performance over time.

We found that compared to the base IEEE 802.11p CSMA and BEB protocols, the Q-Learning MAC protocol can largely mitigate the collision and packet drop problem in congested VANETs by discovering the appropriate CW value to be used for transmissions. Results prove that the proposed protocol, with all parameters set appropriately, allows the network to scale better to increased network density and accommodate higher data rates compared to the IEEE 802.11p standard. It also enables more reliable packet delivery and higher system throughput, while maintaining acceptable delay levels. By deploying some stations featuring the learning MAC in an existing saturated VANET, it is confirmed that the suggested MAC can automatically discover solutions that benefits stations with higher-priority traffic. A consideration that arises from these experiments is the latency performance of the studied Q-Learning-based MAC layer.

Chapter 5

Collective Contention Estimation Reward Mechanism for the Q-Learning-based MAC

5.1 Introduction

The Q-Learning-based MAC protocol as presented so far behaves greedily, since every station tries to maximise its own individual throughput performance. By each station selfishly optimising its throughput, the network-wide throughput will be increased. But by deploying the proposed Q-Learning MAC protocol in this way, throughput fairness among stations cannot be guaranteed, in the sense that some stations will perform better than others, which also affects the bandwidth exploitation capability of the system. Since all stations greedily try to optimise their individual performance, it is possible that some will perform better in the expense of others, depending on the individually collected experience of every station, undermining the network's fairness. With the DSRC communication relying on ad-hoc networking without a central management entity to enforce the fair usage of the channel among multiple vehicles, a decentralised technique should be employed to mitigate the fairness problems in such networks. Based on this information, the Collective Contention Estimation (CCE) reward mechanism for the Q-Learning-based MAC was designed, towards an effort to improve on fairness in dense VANETs.

By combining the logic behind $backoff$ copying [98] [16] with an internal critic provides goal-specific "advice" [65] in the form of state-dependent rewards, the protocol is enhanced towards more efficient and fair bandwidth allocation among stations. Based on the fact that the CW level represents the contending priority for a station, fairness can be promoted among transmissions of multiple vehicles with similar information exchange needs by ensuring symmetry regarding their CW levels. The resultant algorithm exploits this information and allows collaboration in finding the appropriate CW level to accommodate the network traffic. The CCE algorithm is not directly enforcing any CW solution rather than provides suggestions to the Q-Learning-based MAC controller, which ultimately judges the quality of solutions based on the success of transmissions. This way, the RL agent's reward function is enhanced with clues that speed up convergence and help accommodate a wider spectrum applications in terms of fairness and throughput requirements. Furthermore, a technique of combining two sub-goals within the Q-Learning-based MAC algorithm's reward function is proposed and evaluated, so that the CCE algorithm can be used in conjunction with other optimisation goals. This way different reward functions can be developed and utilised depending on whether we strive for high reliability (packet delivery) and fairness or low latency or even balancing both.

5.2 Enhancing the Reward Function

An RL agent receives positive or negative reinforcement in the form of a scalar reward signal upon acting so that it can learn to behave correctly in its environment. Taking advantage of the full channel capacity and achieving maximum packet delivery (throughput) is of primary concern for this system, aiming to satisfy the reliability requirement of V2V traffic. This can be accomplished by employing a simplistic binary reward function according to which the agent is rewarded with 1 in case of successful transmission (ACK received) and -1 in the case of a failed transmission, as presented and evaluated in the previous chapter.

In this design the the agent has a harder problem to solve, compared to using a more detailed, granular reward function. Specifying a more detailed reward function can help the algorithm converge faster, since more clues are provided. Evaluative feedback from internal critics associated with specific goals can be employed to make a function which returns a different reward depending on the CW that was used for every transmission, leading to faster convergence as well as better networking performance. Essentially we can bias the Q-Learning agent to prefer some CW values instead of others, depending not only on the success of transmission but also on (a set of) sub-goals which optimise some other performance-related objectives.

Based on this logic, we present a gradient-based reward function designed for the needs of urban vehicular networks where bandwidth efficiency and fairness regarding channel occupation among stations are of utmost importance. It is based on copying the CW

sizes used by neighbouring transmitting stations and comparing them with the CW the on-board Q-Learning agent suggests. The reward is based both on the success of the packet and the result of that comparison. This addition can be utilised when having many vehicles with similar network presence (i.e. data rate, number of transmitting neighbouring stations) and helps to collectively find the optimum CW that accurately reflects the level of contention. We also validate the delay-sensitive scheme found in [102] and propose a function that combines both objectives.

5.3 The CCE Algorithm

Inspired by [98] [16], we adapt and introduce the $backoff$ copying idea to the Q-Learning agent, in which the receiving stations copy the CW size from overheard data frames coming from nearby stations that experience similar network conditions. This technique can be used as a way to bias the reward function so that agent-stations collectively estimate the network congestion level, as well as compete more fairly for the channel, since all of them content with fairly similar CW sizes.

Our mechanism starts with a piggybacking routine in which the employed CW value for each transmission is piggybacked onto the packet to be transmitted. Receiving stations invoke a CW copying routine, which adds the CW value to a $\boldsymbol{\Sigma}CW[]$ vector. The size of the vector depends on the number of receipt transmissions and a set $PacketsWindow$ parameter. Once the vector fills up, for every new added CW value the last one is removed (FIFO). That way every agent utilising this algorithm considers only the latest receipt CW values, which helps estimate the network-wide congestion level for as long as the window dictates (1 second in this case to keep up with increased mobility and changing topology of vehicular networks).

We use the term "popular" for a CW size, by meaning that the receiving station notices that other transmitters often achieve successful transmissions when using it. A CW size is the most popular system-wise when used for the majority of (successful) overheard transmissions from stations that experience a similar environment. When the receivers become transmitters themselves and eventually get acknowledgement for a successful transmission, a reward calculation routine based on this idea is invoked. Transmitting stations scan the $\boldsymbol{\Sigma}CW[PacketsWindow]$ vector, calculate the frequencies (popularity) of CW values appearing there, by $\frac{counter_{CW}}{length(\boldsymbol{\Sigma}CW[])}$ and store the results in a vector $Frequencies_{CW}[]$ which has a size dictated by the different possible CW values. This vector then gets sorted in descending order, while the algorithm keeps track of what index (CW value) corresponds to which frequency. The agent gets rewarded depending on the order the CW size it used for that transmission has in that vector.

The Q-Learning agent rewards itself more for using CW values that are placed first in order on that vector (are often used to successfully transmit a packet), and less for CW values that are near the end of the vector (are rarely used), by employing equally distanced rewards. This way, the reward function just considers the order of CW levels by their popularity, but not the popularity itself ($\frac{counter_{CW}}{length(\boldsymbol{\Sigma} CW[])}$) so that it is more fair and the Q-Learning agent does not get biased early on and fixed on a potentially wrong CW trajectory. Specifically, when the transmitting station succeeds (and gets an ACK) using the most commonly successful (popular) CW size within its first hop neighbours with same transmission properties (no exploratory packets, similar data rate), its embedded Q-Learning agent is given the maximum possible reward. Every other CW placing below that in order of popularity will get its acquired reward reduced by 1/7th at a time (since we consider 7 CW levels). i.e. in the case of the least popular CW (with the least successful transmissions in the near network), the reward multiplier will be 1/7. The mechanism's operation is summarised in Algorithm 2.

The CCE reward function is expected to improve fairness and reduce the convergence time of the Q-Learning algorithm, thus give a larger performance benefit, earlier. It is also quite efficient regarding networking overhead since it costs just 3 bits per packet to represent the 7 CW levels which can be easily absorbed by the IEEE 802.11p standard. It could also be adapted for prioritisation among different classes of data since many proposed techniques use different CW sizes for the same purpose.

Algorithm 2 Collective Contention Estimation

```
ΣCW = []
CW_levels[7] = [3, 7, 15, 31, 63, 127, 255]
Reward_CW[7] = [1/7, 2/7, 3/7, 4/7, 5/7, 1]

procedure CW_Copy(RxPacket)
    if RxPacket.AppType = This.AppType
     AND RxPacket.Explore = 0 then
        PacketsWindow++                                  ▷ Resets to 0 every 1s
        if length(ΣCW) > PacketsWindow then
            ΣCW[].remove(ΣCW[0])
        end if
        ΣCW[].add(Packet.GetCW)
    end if
end procedure
```

procedure R_{CCE}(RxPacket)
  for $i \leftarrow 0; i < length(CW_{levels[]}); i++$ **do**
    if $RxPacket.\mathbf{CW} = CW_{levels}[i]$ **then**
      $index_{CW} \leftarrow i$ ▷ Find CW index
    end if
  end for
  $counter_{CW} \leftarrow 0$
  for $i \leftarrow 0, i < length(\Sigma CW[]), i++$ **do**
    if $\boldsymbol{\Sigma} CW[i] = RxPacket.\mathbf{CW}$ **then**
      $counter_{CW}++$
    end if
  end for
  $Frequencies_{CW}[index_{CW}] \leftarrow \frac{counter_{CW}}{length(\boldsymbol{\Sigma} CW[])}$
  $SortedFrequencies_{CW}[] \leftarrow Frequencies_{CW}[]$
  sort($SortedFrequencies_{CW}[]$)
  for $i = 0; i < length(Frequencies_{CW}[]); i++$ **do**
    if $Frequencies_{CW}[index_{CW}] =$
    $SortedFrequencies_{CW}[i]$ **then**
      $index_{reward} \leftarrow i$
    end if
  end for
  return $Reward_{CW}[index_{reward}]$
end procedure

5.4 Combination of two sub-goals

Similar logic regarding reward assignments can be applied to introduce delay awareness to the protocol, as seen in [102]. As mentioned, the CW parameter is defined as the number of timeslots the station has to weight prior to transmitting, so the smaller this parameter is, the better in terms of total latency. The smaller CW values can be given higher reward. The larger the CW size, the lower the reward given.

Additionally we can further optimise performance, by combining the two objectives (fairness and low latency). This can be achieved by specifying even more detailed reward function, featuring 49 discrete reward levels (equally distanced from each other) if the proposed fairness-aware, CCE reward function is used in conjunction with a delay-aware reward function. This would also focus the agent on a trajectory even faster than using just 2 or 7 reward levels as shown before.

$$R_{func}(CW) = R_{CCE}(CW) \times R_{delay}(CW), \tag{5.1}$$

We found the approach in 5.1 to be more efficient when it comes to minimising latency than a "softer" reward approach of combining rewards like $R_{func}(CW) = \frac{R_{CCE}(CW)+R_{delay}(CW)}{2}$, via which the agent can receive relatively high rewards without necessarily achieving a high reward from both the delay-aware and CCE functions. So i.e., the reward would be $r_t \leftarrow r_{CCE} \times r_{delay} = 1/7 \times 4/7 = 4/49$ for using the CW value which is the least common found in receipt packets, but is averagely favourable for delay intolerant applications. Effectively, using the product of the result of the two functions as a reward, makes the one act as a filter to the other. This way, the agent is less punished when it simultaneously achieves both sub-goals (low latency, fairness) in a single transmission. If the designer of a system needs to add bias towards one optimisation factor compared to the other, a weighted product function can be used as in the following,

$$R_{func}(CW) = R_{CCE}^{k_{CCE}}(CW) \times R_{delay}^{k_{delay}}(CW), \tag{5.2}$$

where the weights, $k_{CCE} + k_{delay} = 2$ and $0 < k_{CCE}, k_{delay} < 2$. The neutral case in (5.1) can be obtained for $k_{CCE} = k_{delay} = 1$. A schematic of the protocol's operation utilising both enhancements (fairness and delay awareness) is seen below in Fig. 5.1.

Plotting the reward function equation, also expressed as (5.3), for increasing values (from the lower part of the figure towards the upper) of R_2 in the range of [1/7,1] with a step of 1/7 (them being the stages of the rewards for 7 different CW values) reveals

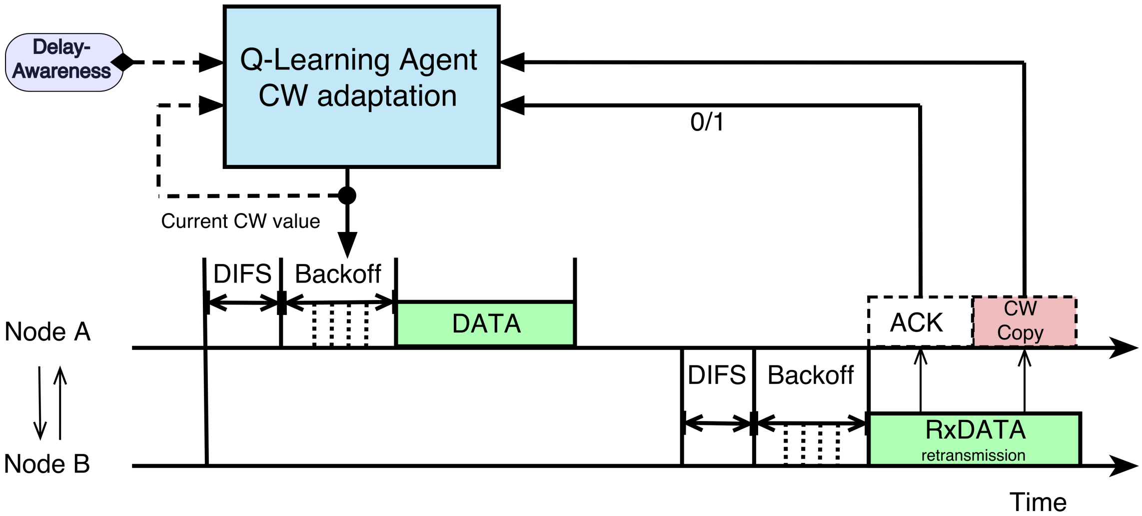


Figure 5.1: Q-Learning MAC with fairness and latency optimisations

the total reward R_{total} acquired by the Q-Learning agent for all possible combinations of individual rewards given by R_{CCE} and R_{delay} and biases k_{CCE}, k_{delay}. The resulting figure (Fig. 5.2) can be especially useful when defining the value of biases, since the Q-Learning algorithm's convergence relies on the temporal difference among reward values, and this difference should be significant enough over time for the various (s, a) pairs so that the algorithm can decide on a (near) optimal value more easily (within less iterations).

$$R_{total} = R_1^k \times R_2^{2-k} \tag{5.3}$$

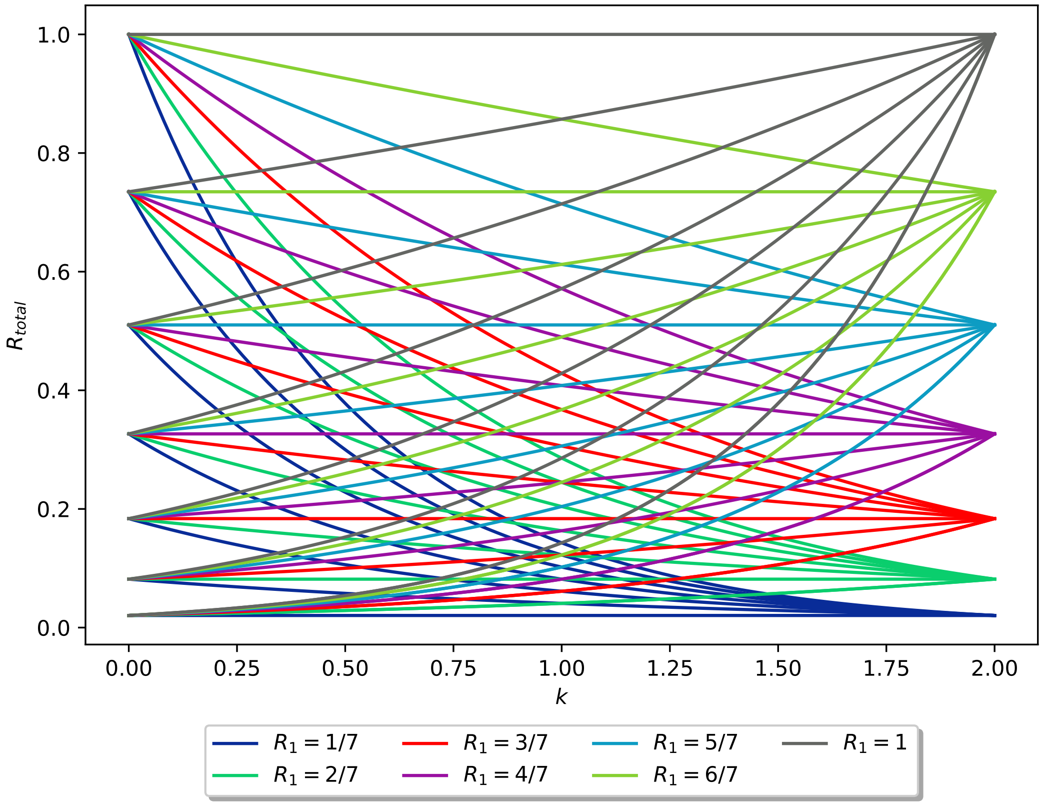


Figure 5.2: R_{total} for different combinations of R_1, R_2, k

5.5 Performance Evaluation

A MAC protocol should achieve three main objectives when the wireless medium is shared among multiple vehicle stations: bandwidth efficiency, low latency, and fairness. Consequently, we evaluate our designs against that criteria.

5.5.1 Experiment Setup

Simulation Parameters

All the cars within the area content for access to medium when trying to transmit a packet or rebroadcast a copy of one. We perform our tests in a simulated vehicular environment with moving IEEE 802.11p stations implemented with OMNeT++ 5 and the Veins framework. The SUMO mobility co-simulator takes care of the vehicle mobility aspect. This time all vehicles are placed on a 3-lane highway and travel with a maximum velocity of 15 m/s so that the maximum distance travelled is 4.5 km in 300 s. The Krauss mobility model is used with default parameters as seen in Table 5.1, and the maximum distance among them reaches up to no more than 1 km as the simulation progresses. A snapshot of the formation of vehicles at 100 s of simulation time can be seen in Fig. 5.3.

The scenarios envisaged in this work consider $N_{vehicles} = 50$ or 100 stations; each station generates data packets with constant rate $f_{gen} = 10\,\mathrm{Hz}$ by employing a bit rate, R, which would depend on the experienced channel quality. The receivers can calculate the forwarding (ACK) probability P_{fwd} in real time according to (4.7). All packets have a common header which is similar to CAMs or DENMs, but is modified to include Node ID, application type, whether a packet is original or a retransmission, the employed CW and whether that CW was used due to exploration or exploitation.

Regarding Q-Learning training and evaluation, the discount factor γ is in the range of 0.7 to 0.9. Both ϵ and α quantities undergo exponential decay, as seen in 5.4 rather than linear decay as in the previously chapter, since it forces the system to use gained experience and limits randomness much faster, which is especially useful for mobile environments such as vehicular networks. A larger decay constant λ will make ε and α vanish more rapidly, which enforces exploitation sooner but may negatively affect learning so there is a trade-off to be made. This technique, in combination with the CCE technique will allow the Q-Learning algorithm to provide greater performance benefit, earlier. This can be achieved via the function shown below,

$$\varepsilon = \alpha = e^{-\lambda \times \frac{N_{tx}}{N_{train}}} \quad \text{for} \quad 0 \leq N_{tx} \leq N_{train}, \tag{5.4}$$

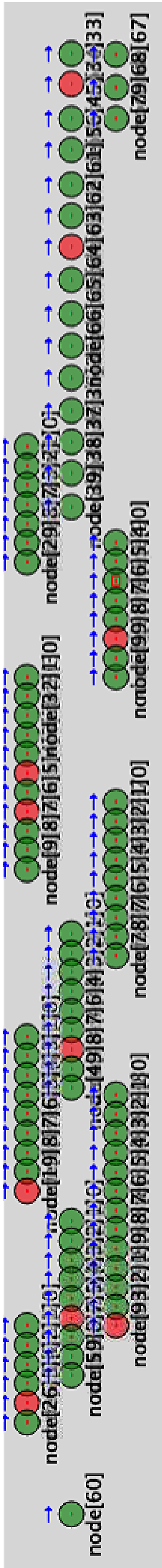


Figure 5.3: The 3-lane highway scenario used in network simulations. Green/red colours of vehicles identify successful/failed transmission (ACK/NACK received) of their latest packets respectively.

Parameter	Value
Simulation time	300 s
Training period N_{train}	1800 packets
Channel Frequency	5.89 GHz
Channel Bandwidth	10 MHz
Transmission rate R	9 Mbit/s
Transmission power	Single-hop: 30 dBm Multi-hop: 17 dBm
Packet size L_p	256 bytes
Backoff slot time	13 μs
Packet Generation Frequency f_{gen}	10 Hz
ACKs per packet N_{fwd}	Single-hop: 2 Multi-hop: 6
Effective Broadcasting Frequency f_b	(4.8)
Packet Generation Offset	0.005 s
Discount rate γ	[0.7, 0.9]
Mobility Model	Krauss model with default parameters ($\sigma = 0.5$, $\tau = 1$)
Maximum Vehicle Velocity	15 m/s
Vehicle Acceleration Ability	2.6 m/s^2
Vehicle Deceleration Ability	4.5 m/s^2

Table 5.1: Simulation Parameters for Q-Learning-based MAC with CCE evaluation

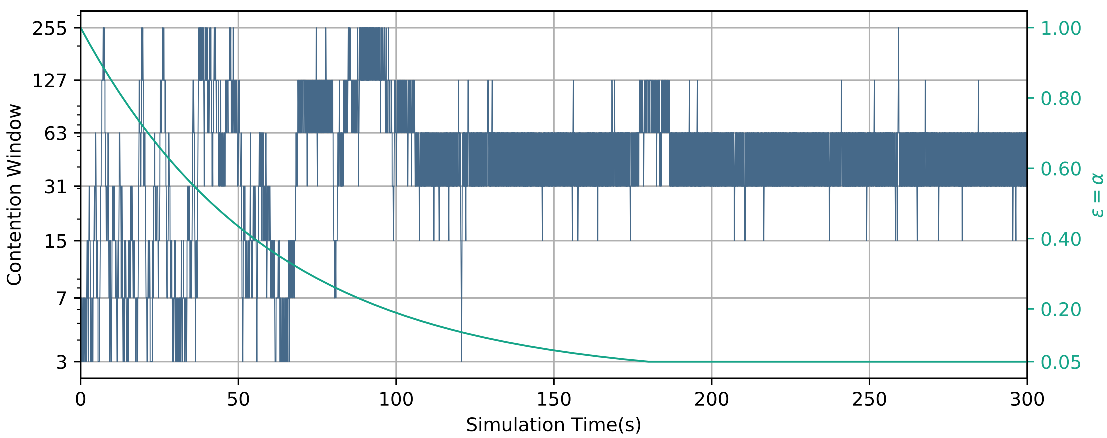


Figure 5.4: CW adaptation by a single station utilising the Q-Learning based MAC protocol with exponentially-decaying ε-greedy exploration policy

The learning rate α and ε-decay function lasts for 180 s or $N_{train} = 1800(5400)$ original (total) packets, with $\lambda = 3$. The two training methods, linear and exponential, and their effect in throughput over time can be seen in Fig. 5.5. It can be observed that the exponential ε-decay method yields a relatively larger performance benefit, faster, with a small impact in the quality of the solution. Throughout this section, we present 5-minute snapshots of the agent's behaviour under various configurations and metrics, that combine both the training (intense exploration) and evaluation (ε=0.05) stages, revealing how MAC Q-Learning agents would perform in a system if deployed with initialised Q-Tables.

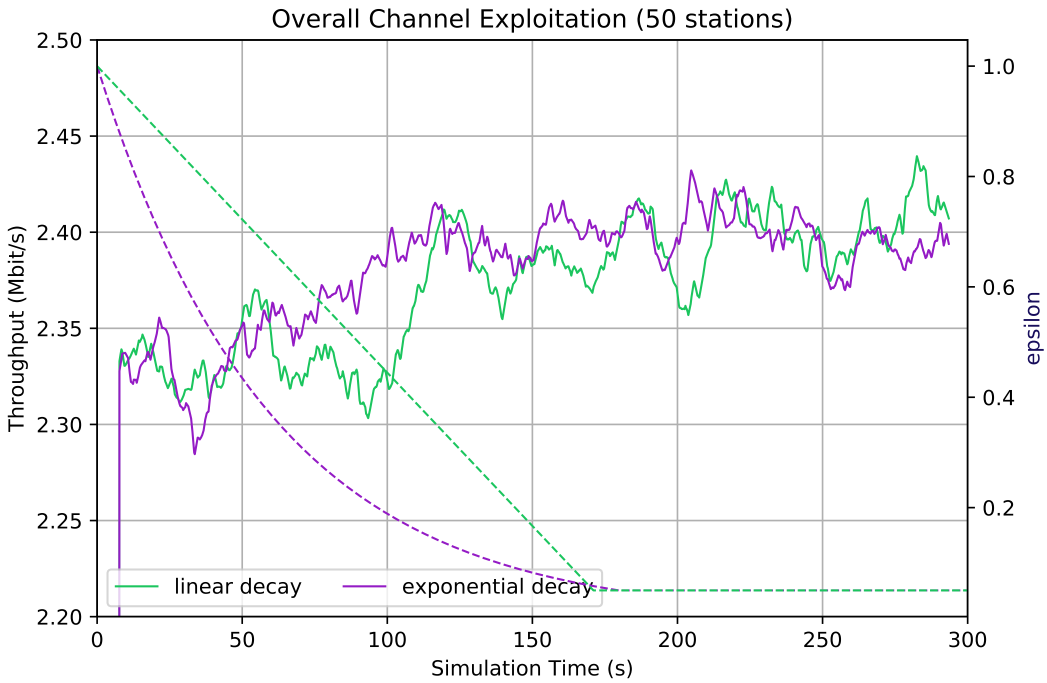


Figure 5.5: CW adaptation by a single station utilising the Q-Learning based MAC protocol with different decaying ε-greedy exploration policies

Benchmarked Protocols

- **IEEE 802.11p:** It is the baseline protocol operating in OCB (broadcast) mode with fixed $CW = CW_{min} = 3$, as defined in the standard for the fastest AC (lowest transmission latency). It has no CW adaptation capability.

- **Pseudo-BEB:** The emulated BEB CA algorithm adapted for the IEEE 802.11p MAC based on feedback from with implicit ACK packets via retransmissions.

- **Q_MAC:** Our original protocol first presented in Chapter 4 with a binary reward function. Its operation is depicted in Fig. 4.1.

- **Q_MAC+CCE:** The novel protocol introduced in this Chapter based on Q-Learning in conjunction with the CCE reward algorithm where $R_{func} = R_{CCE}$
- **Q_MAC+Delay:** It is the Q-Learning agent using the delay-aware reward function from [102] where $R_{func} = R_{delay}$.
- **Q_MAC+Delay+CCE:** The novel protocol which targets satisfying both sub-goals, utilising 5.2.

Applying a moving average filter to the CW recordings over time reveals the mean system-wide CW over time. From these CW dynamics, we can make interesting observations about the significance of this parameter in dense IEEE 802.11p networks, as well as evaluate the collective behaviour of the Q-Learning agents over time using various reward functions. It can be seen in Fig. 5.6 that all the proposed solutions try to minimize the medium congestion level by enforcing different CW values on communications.

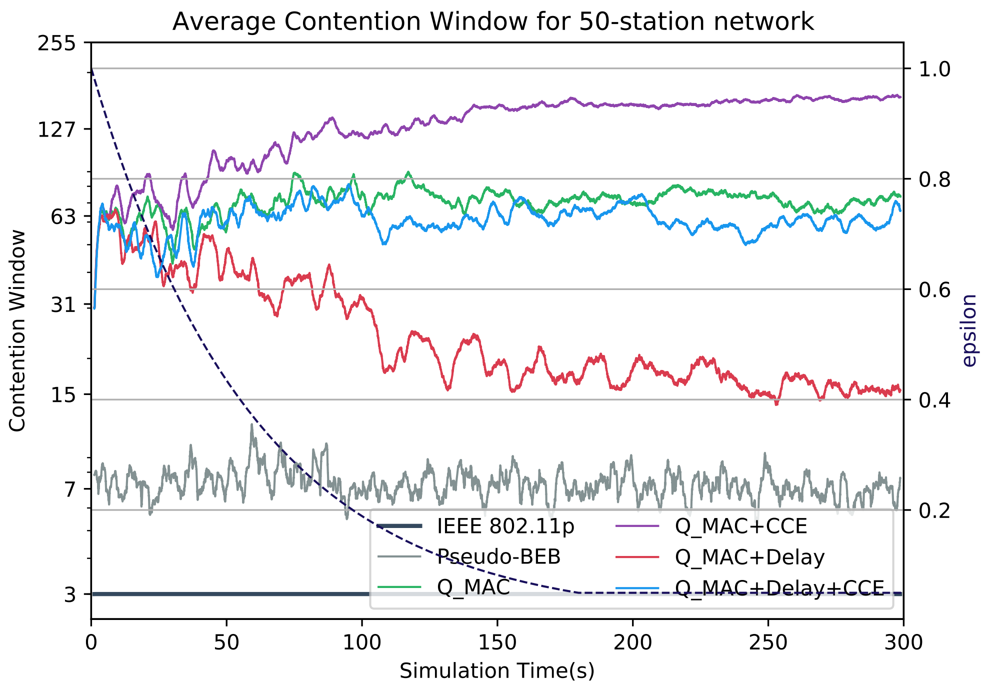


Figure 5.6: Network-wide CW dynamics for different collision avoidance mechanisms

The original Q-Learning MAC protocol strives for maximum transmission reliability, and the one with the CCE reward function strives for both reliability and fairness regarding contention. The delay-aware function tries to use a CW as small as possible while achieving acceptable reliability. When combining both reward functions, as in (5.1), the mean system-wide CW is quite higher since the agent strives for reliability and fairness, but still lower than the other two Q-Learning based solutions.

In the following, we show our findings regarding throughput, fairness and latency in four different V2V scenarios: Medium Traffic, High Traffic, Two Simultaneous Applications and Multi-hop.

5.5.2 Medium Traffic Environment

We first evaluate the proposed and existing MAC mechanisms against each other when deployed in a sparser network environment of 50 vehicles transmitting 256-byte packets. In Fig. 5.7, it can be seen how each protocol utilises the channel, since efficient bandwidth usage is their primary objective. The protocols' performance is evaluated against the maximum achievable throughput (Throughput Optimal), which is found via performing an exhaustive search among the possible CW values applied symmetrically to the network, as shown in Appendix 2. The CCE reward function (Q_MAC+CCE) clearly performs better regarding achieved throughput, since the agents collectively estimating congestion do a better job than every one acting completely independently. Also the use of similar CW sizes is enforced, which leads to less collisions. The other Q-Learning based solutions also perform quite better than the baseline OCB IEEE 802.11p with $CW = CW_{min}$. Our BEB implementation on the other hand is not yielding a great increase in throughput compared to the original protocol. The poor performance of BEB is due to the increase of collisions under increased network traffic load, since the mechanism is collision-triggered and resets a station's CW to CW_{min} after every successful transmission. On the other hand, the proposed solutions update the CW around a value that resolves as many collisions as possible and keep it there.

Furthermore, the achieved transmission latency is examined. The Normalised CDF is produced for each protocol, shown in Fig. 5.8. An interesting observation from Fig. 5.8 is that each solution shows different performance limits on delay and packet deliver ratio. With a more relaxed delay deadline, non-delay sensitive solutions show better packet delivery ratio, e.g., achieving optimal throughput can translates to almost 79% of packet delivery ratio but with a latency of up to 40 ms. Q_MAC+CCE is very close at 77%, and outperforms the optimal throughput solution for latency requirements below 30 ms.

Additionally, given latency requirements of 12 ms to 20 ms [1], our Q_MAC+Delay+CCE performs better than the rest of the protocols achieving the highest transmission reliab-

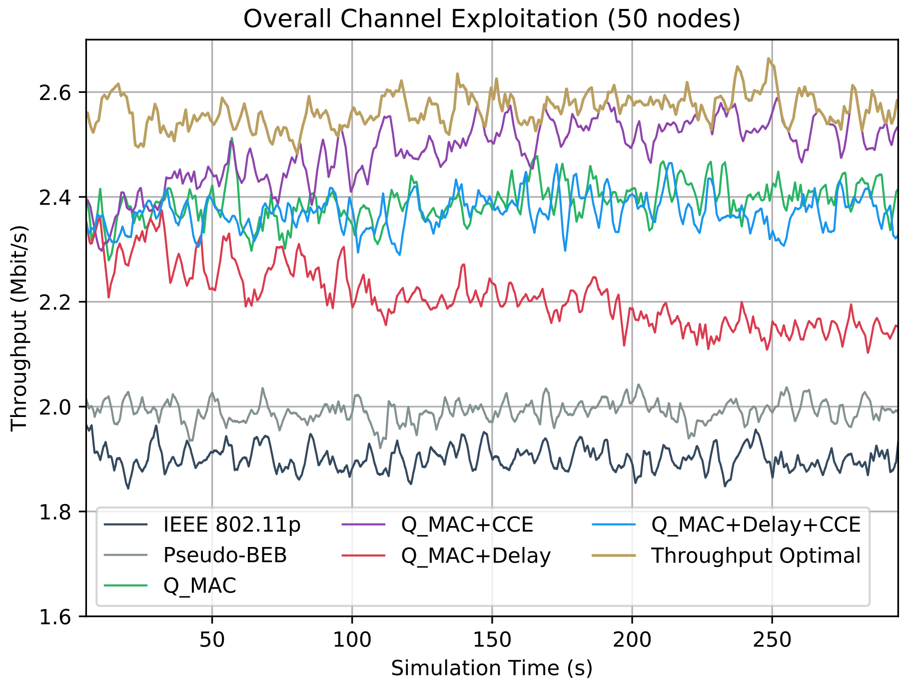


Figure 5.7: Mean network-wide throughput for 50 stations

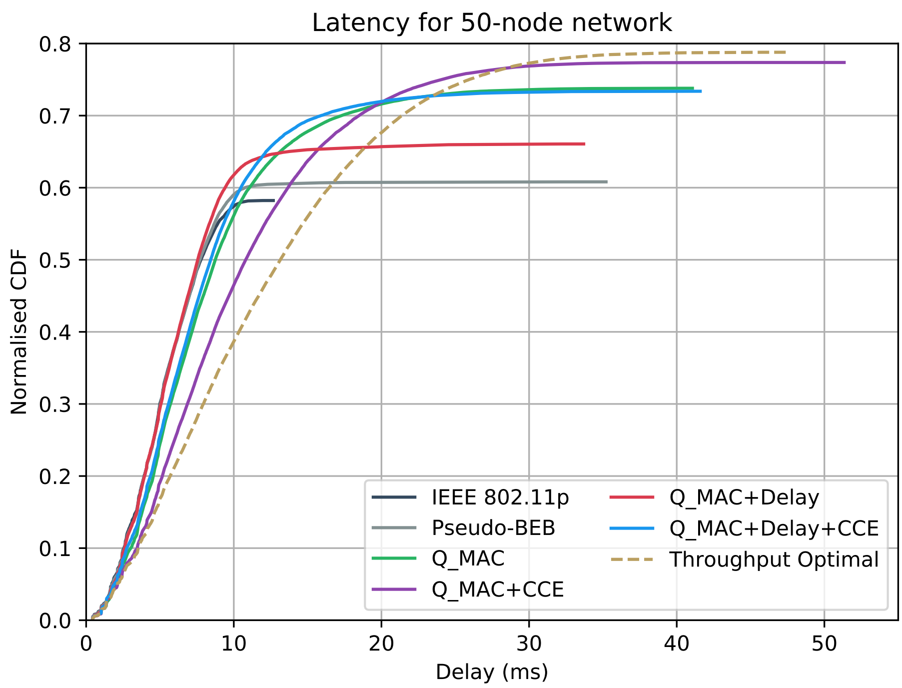


Figure 5.8: End-to-end transmission latency versus PDR in 50-station network

ility, i.e., a packet delivery ratio of 72% shown on the Y-axis. Q_MAC+Delay is the best solution if latencies lower than 12 ms are needed. So we conclude that with appropriate tuning (balancing the trade-off between delay awareness and CCE with (5.2)) the Q-Learning MAC protocol can better satisfy even the most stringent delay requirements for the medium-traffic environment. Fig. 5.8 can be used as a guideline to select a suitable access strategy given an application requirement.

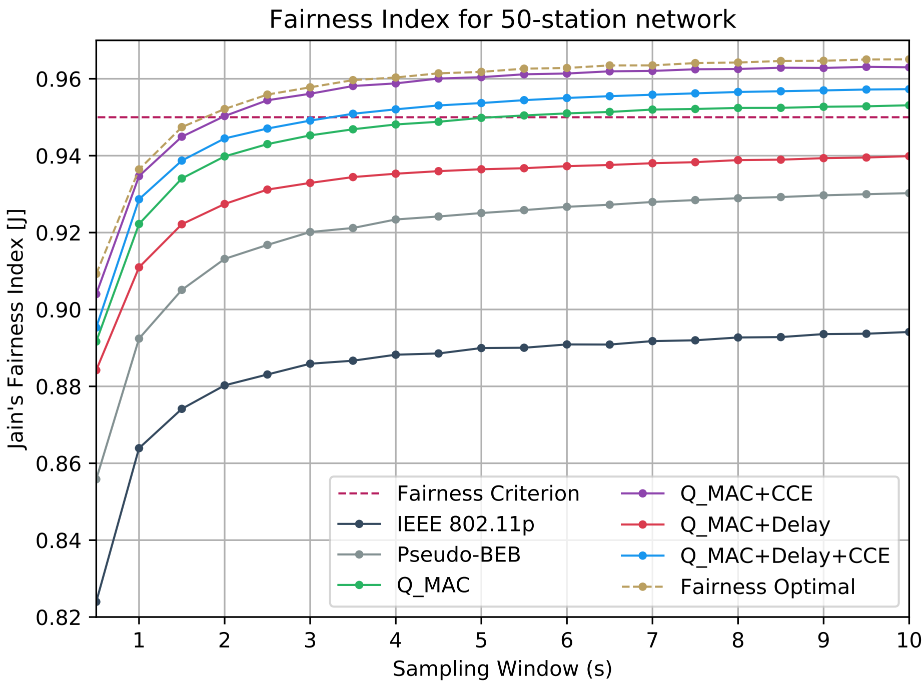


Figure 5.9: Recorded fairness in 50-station network over different sampling windows increasing with a step of 0.5 s

When it comes to fairness measurements, shown in Fig. 5.9, the CCE enhanced Q-Learning agents perform better than the simpler protocols they are based on (Q_MAC and Q_Delay), as expected. Specifically, the Q_MAC+CCE protocol meets our strict fairness criterion even within 2 s or 60 packet transmission attempts per station, (compared to 4.5-5 s for Q_MAC) which is great for critical exchanges. If the CCE reward function is combined with delay-awareness (Q_MAC+Delay+CCE) it takes 3 s for the same level of packet fairness to be achieved. The delay-focused reward function without CCE performs quite worse in that regard, since it does not achieve optimal fairness even in a 10 s sample - or 300 transmitted packets per station. We conclude that for this sparser scenario, using the proposed CCE reward function makes a significant difference regarding fair bandwidth allocation among vehicles.

5.5.3 High Traffic Environment

We then test 100 contending stations transmitting 256-byte packets. Aggregate throughput measurements over time for the system are shown in Fig. 5.10. When compared to each other, the protocols perform as in the previous scenario regarding achieved throughput.

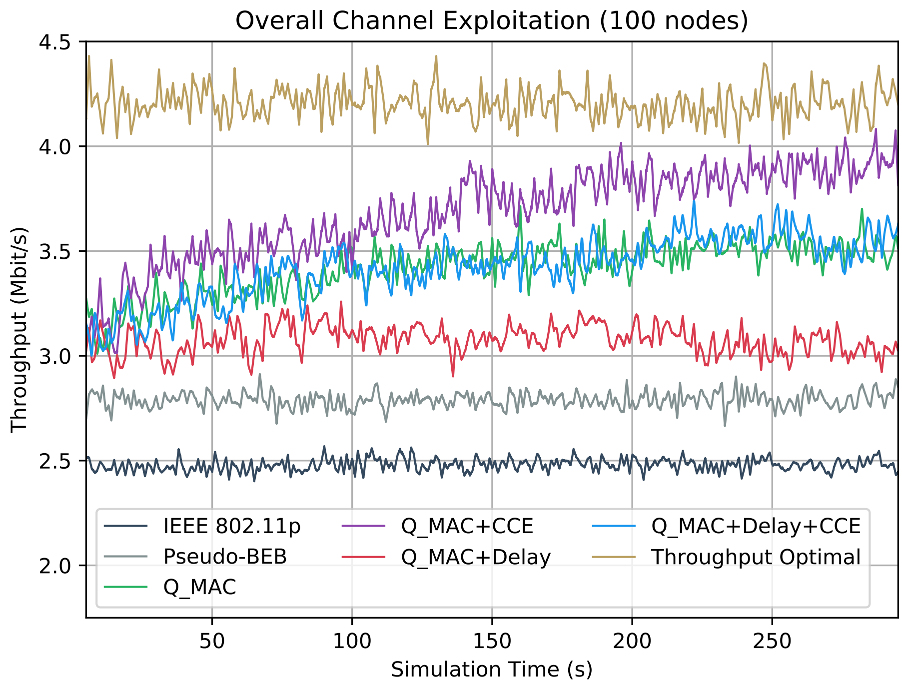


Figure 5.10: Mean network-wide throughput for 100 stations

Although the performance gap between the proposed CCE reward function and optimal is slightly wider, i.e., 6.33% during 300 s of simulation time, given more time, the protocol can achieve optimal throughput. In terms of the practical requirements on short-term performance and applicability in VANETs, the algorithm can yield the presented gain over time or be pre-trained and activated in dense environments where there is large quantity of information to be exchanged among vehicles tuned in the same DSRC channel.

But when comes to transmission latency, shown in Fig. 5.11, the learning MAC with joint CCE and delay-awareness outperforms all MAC solutions in terms of packet delivery for latency requirements among 22 to 33 ms. Q_MAC, which cannot be further controlled performs quite closely. The Q_MAC+Delay protocol, which defines what is possible when focusing on low latency exchanges, outperforms the rest for 13.5 to 22 ms. Given a delay requirement of 100 ms which is typical for V2V applications, Q_MAC+CCE is more preferable in practice since it achieves the highest delivery ratio.

When it comes to transmission fairness, shown in Fig. 5.12 the results are quite similar

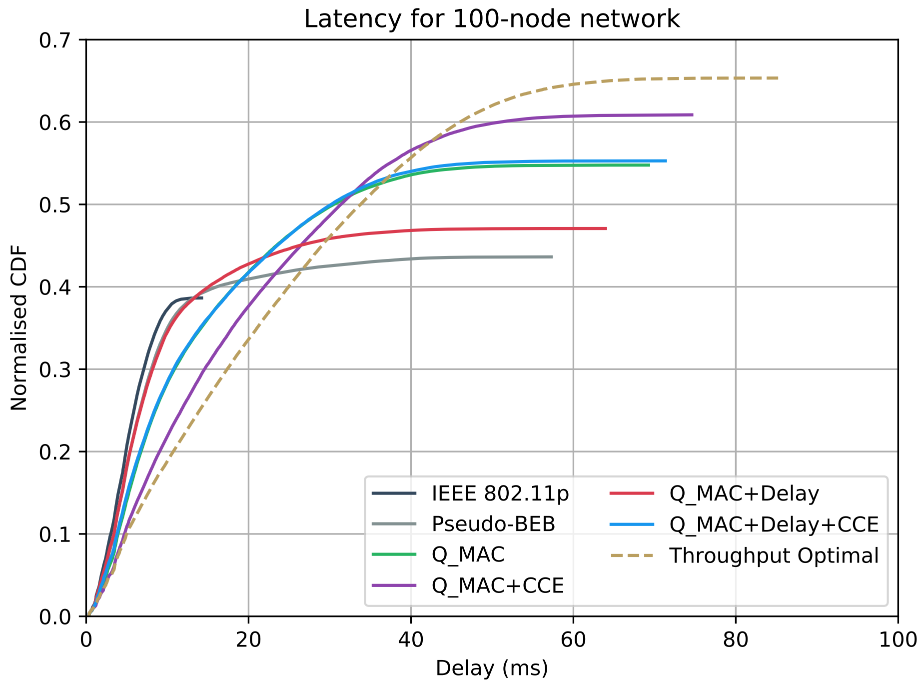


Figure 5.11: End-to-end transmission latency versus PDR in 100-station network

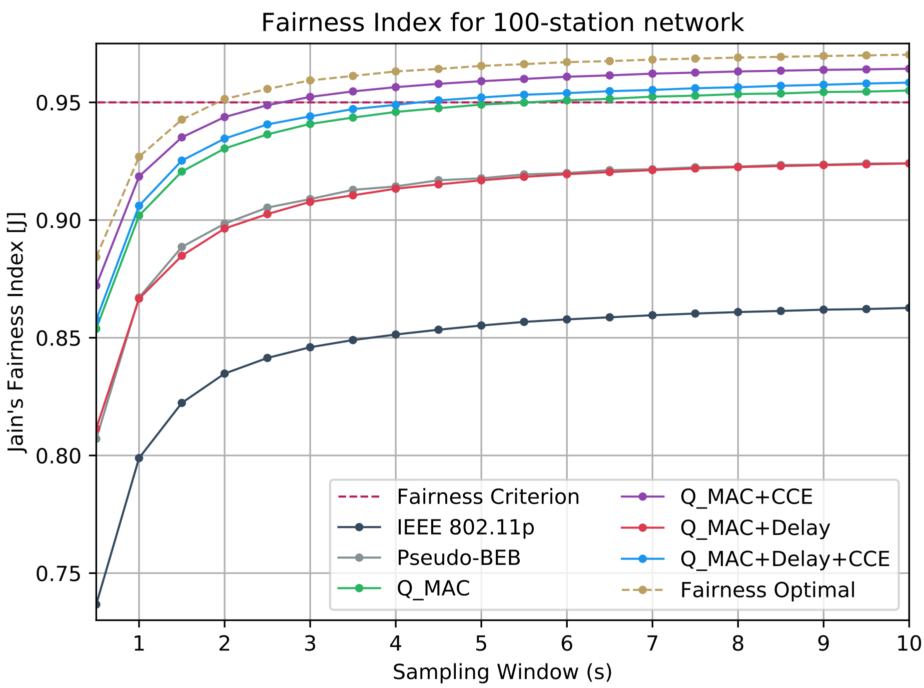


Figure 5.12: Recorded fairness in 100-station network over different sampling windows increasing with a step of 0.5 s.

to the first scenario. Both CCE-enhanced Q-Learning protocols are throughput-fair within shorter time than their non-CCE counterparts. The BEB and Q_MAC+Delay are not fair in the short term or long term when evaluated against our criterion. The baseline DSRC MAC also cannot handle 100 cars regarding neither long-term nor short-term fairness.

5.5.4 Two simultaneous services

The same mechanism for improving fairness on a network level can be employed by the protocol to better accommodate multiple simultaneous applications, by the same (EDCA-like priorities) or different stations. We enforce application separation regarding CW by making the CCE algorithm check the application type field which is contained in the packets, meaning that only CW values from packets of the same application get copied and affect the Q-Learning reward function. Additionally, only stations running the same application retransmit each other's packets so that we can collect fair measurements.

We simulate stations of two types, running different application layers. To make a fair comparison regarding raw network-wide throughput, we set 80% of the vehicles to transmit 256-byte packets and 20% of the vehicles to transmit 1024-byte packets. Consequently in the scenario of 50 vehicles presented below, 40 cars run the first application and 10 cars run the second one. Assuming no packet losses, the throughput of the two applications should be equal to each other ($\frac{T_{hB}}{T_{hA}} = 1$). Only stations running the same application collectively estimate the optimum application-wide CW, instead of all stations trying to find the optimum system-wide CW. The recorded application-wide throughput for all protocols can be seen in Fig. 5.13 and Fig. 5.14 for applications A and B respectively.

It can be observed that there is significant increase in throughput (approaching the optimal solution) when our novel learning technique is applied to the DSRC MAC. Although the throughput of the two applications would be equal should there be no contention, in practice larger packet transmissions are more prone to collisions, and if losses occur the throughput is also affected much more, because of the larger packet size. This is reflected in the collected results, as expected. But if we evaluate application-wide fairness expressed as a ratio of throughput of application B (1024 byte packets) over application A (256 byte packets), the proposed learning technique shows significant improvement over the DSRC stack. The Q_MAC+CCE protocol achieves a ratio of up to $\frac{T_{hB}}{T_{hA}} \approx 0.74$ for throughput of application B over application A, compared to 0.658 for the baseline IEEE 802.11p solution, while yielding the highest overall throughput as well, within 6.5% of the optimal solution.

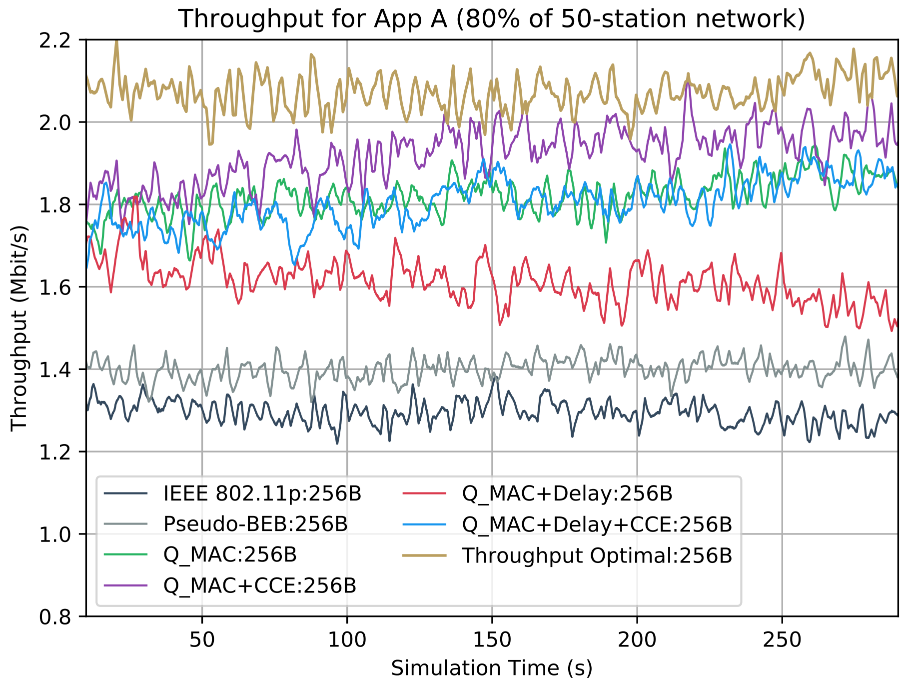


Figure 5.13: Total throughput achieved by stations transmitting 256-byte packets

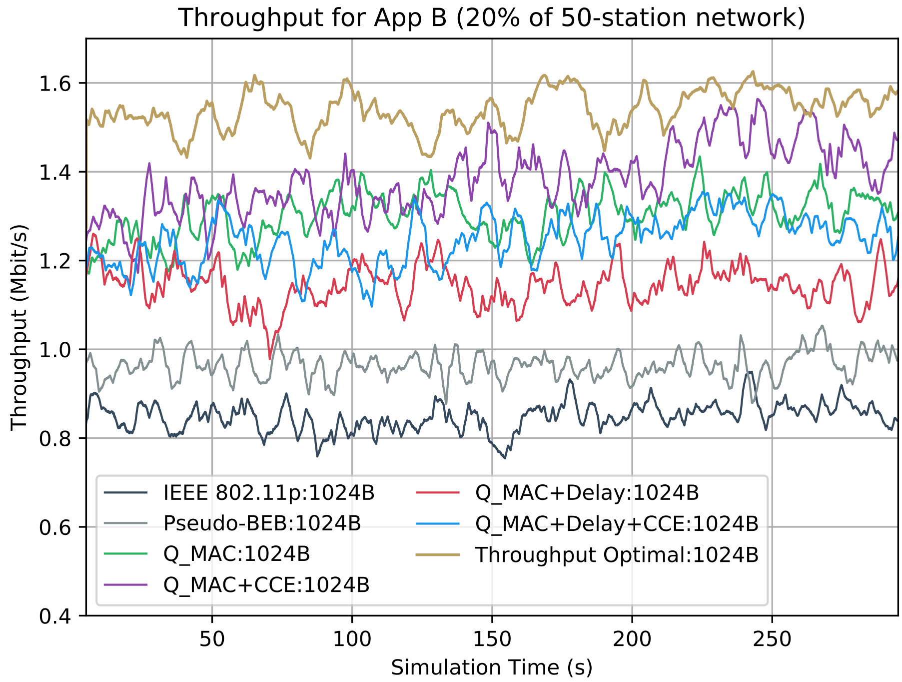


Figure 5.14: Total throughput achieved by stations transmitting 1024-byte packets

Regarding end-to-end transmission latency, depicted for all successful packets in Fig. 5.15, we again observe that using R_{func} with both sub-goals combined (Q_MAC+Delay+CCE), again favours low latency exchanges, with the protocol achieving the higher delivery ratio for latencies below 28 ms all the way down to 14 ms end-to-end. Again, given a delay requirement of 100 ms which is typical for many V2V applications, the protocol with the highest raw throughput Q_MAC+CCE performs better.

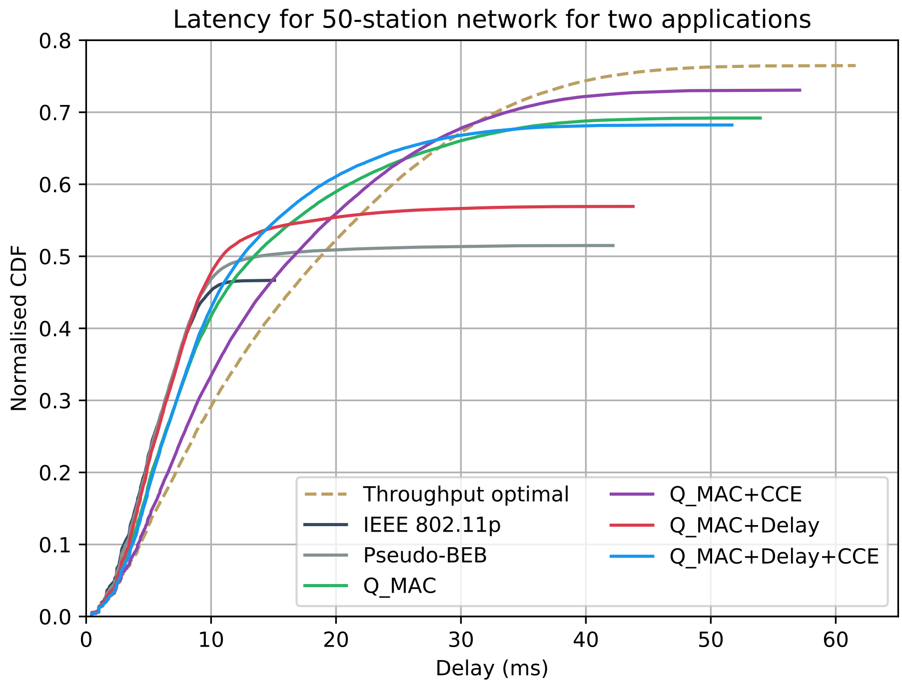


Figure 5.15: End-to-end transmission latency of transmissions in 50-station network versus PDR for both concurrent applications

Network-wide packet-based fairness for all stations, no matter the application they are running, can be seen in Fig. 5.16. IEEE 802.11p CSMA with or without the BEB exhibits a more severe fairness problem under these multi-rate conditions, which can be tackled using the learning-based methods. We can again confirm that resetting to CW_{min} is not good regarding delivery or fairness in sustained high traffic. Again, better performance can be achieved when the proposed CCE method is utilised in conjunction with the Q-Learning MAC mechanism, with or without delay awareness. The fairness aware learning protocol Q_MAC+CCE achieves $J = 95\%$ within a window of 2-2.5 s or 60-75 transmitted packets per station, while Q_MAC achieves the same of fairness within about 6.5-7 s or 195-210 packets. The Q_MAC+Delay+CCE protocol can reach the set criterion within 3.5 s for this simulation scenario, while the other latency optimised protocol without considering the CCE function (Q_MAC+Delay) cannot reach the fairness criterion at all.

An interesting observation is how the baseline Q_MAC and the CCE-enhanced proto-

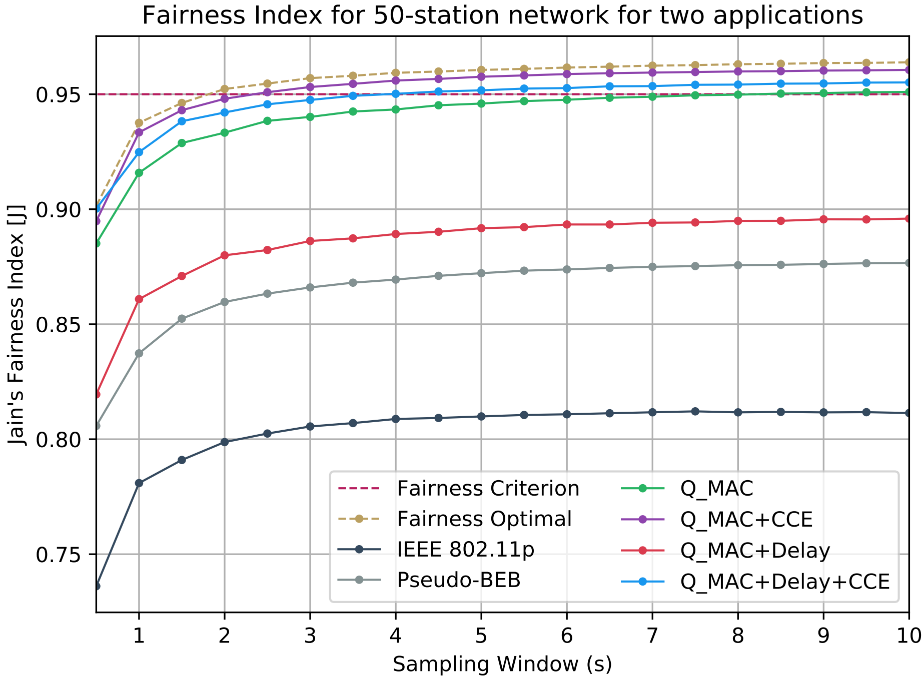


Figure 5.16: Recorded fairness in 50-station network for two concurrent applications

cols handle service separation by adapting the CW of the station's MAC layers, depicted in Figure 5.17. Both the baseline learning and the CCE-enhanced mechanism favour transmission of larger packets by reducing the CW value used to contend for the channel by the stations transmitting them. This happens since larger transmissions are more prone to collisions, which the Q-Learning MAC perceives and tries to mitigate. Larger packets have a lower probability of successful transmission, since they need more time for their transmission to be completed. Naturally, as seen in the previous figures, this prioritisation results in both higher overall (network-wide) and application-wide throughput. So it can be observed that stations transmitting smaller packets use a larger CW value on average when compared to the ones transmitting much larger packets. This result also correlates with the findings from the hardware and simulation experiment of DSRC-based asymmetrical network. In the presence of heavy contention, a few stations can gain significantly more transmission opportunities if their $backoff$ time is smaller than their contending peers.

When looking for the optimum system value, we cross validated the Q-Learning result by using a smaller CW value for the stations transmitting larger packets, so that they could win more often when contending for channel access. Indeed, both Normalised CDF and fairness show that packet delivery disregarding latency of transmissions is very similar

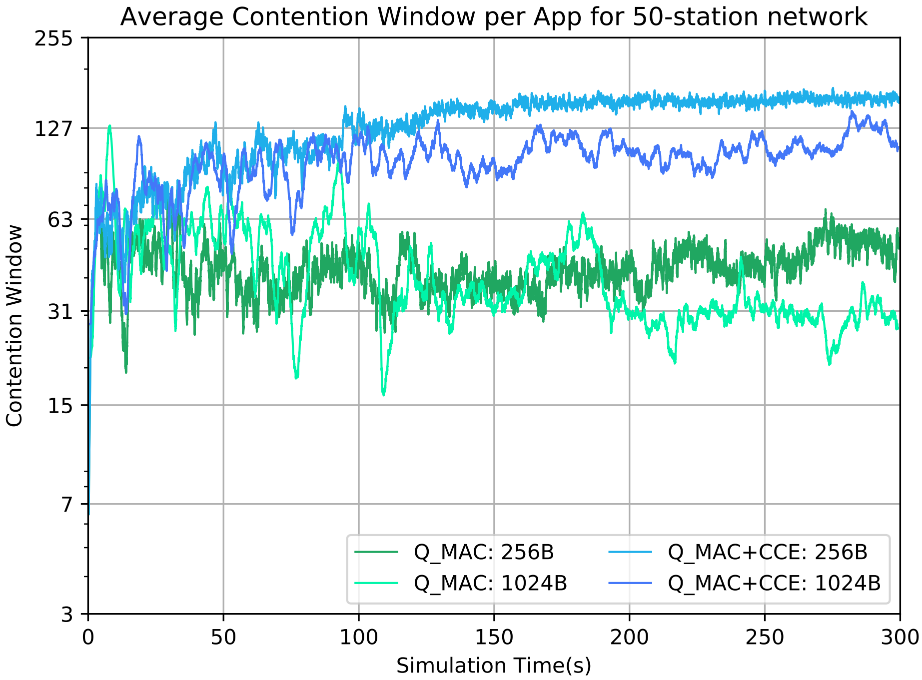


Figure 5.17: Application-wide CW dynamics for different Q-Learning MAC implementations

among the $CW_{256} = CW_{1024} = 255$ and $CW_{256} = 255, CW_{1024} = 127$ cases, as seen in Fig. 5.18 . They also perform quite closely regarding fairness, as seen in Fig. 5.19. The latter performs optimally among solutions for transmissions that need to be $< 33\,ms$, and for fairness recorded for 7.5-9 s, seen in 5.19.

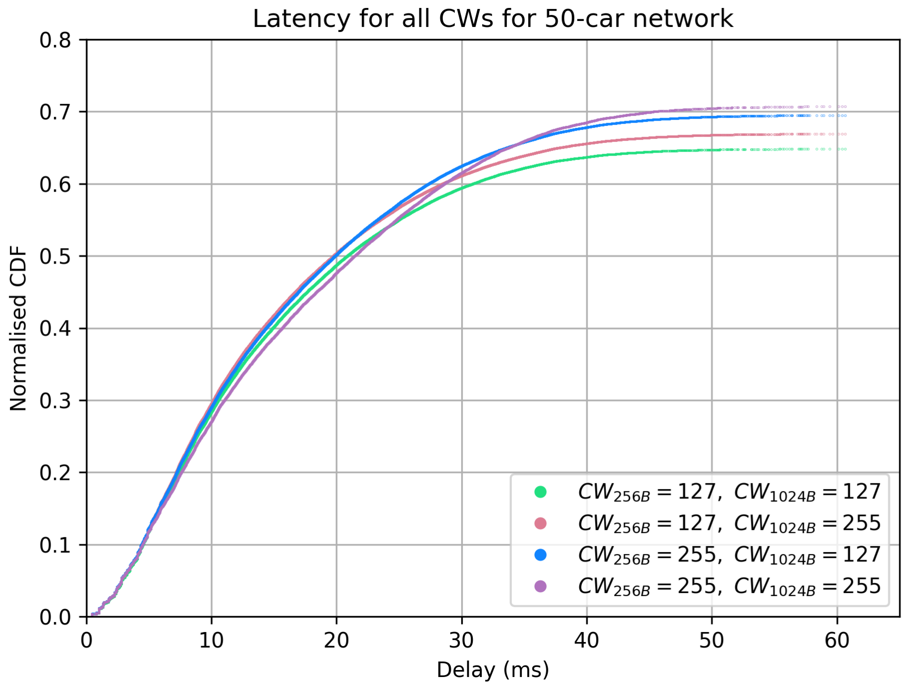


Figure 5.18: Latency for 50 stations running 2 applications - examining different combinations of CW per app.

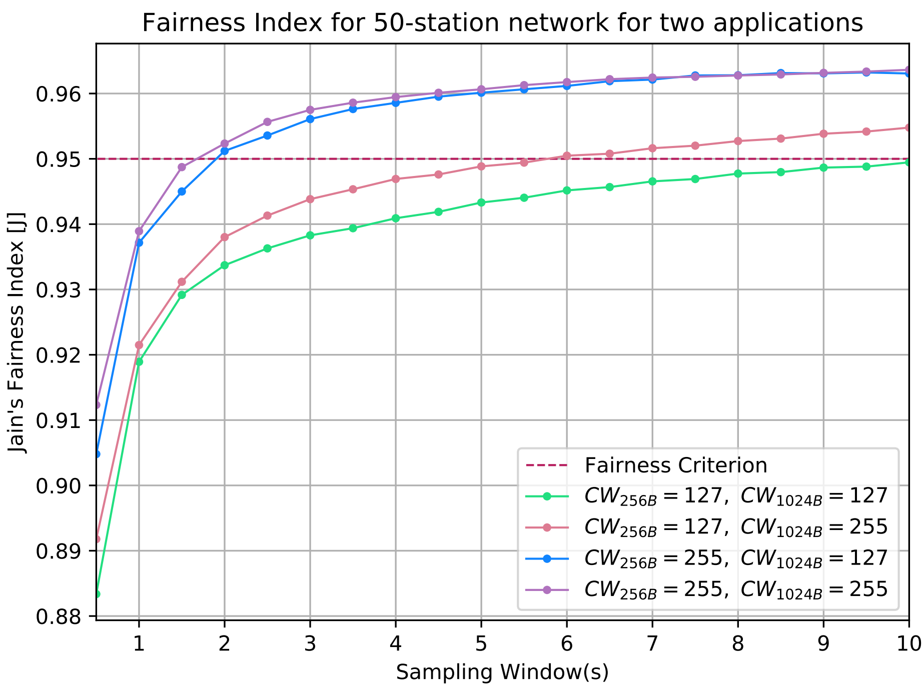


Figure 5.19: Fairness for 50 stations running 2 applications - examining different combinations among CW values per app.

5.5.5 Multi-Hop Environment

The performance of the Q-Learning MAC has also been studied under a dense multi-hop network environment of 100 stations, which are placed at most 2 hops away from each other. Every vehicle periodically calculates its packet forwarding probability P_{fwd} depending on the number of its one-hop neighbours via (4.7), by setting $N_{fwd} = N_{ACK} = 6$ to ensure coverage for the given RoI, even in the increased presence of collisions because of hidden nodes. Each vehicle forwards a copy of a received packet at most once to limit redundancy. Again the Q_MAC+CCE protocol yields the highest raw throughput among the protocols, as seen in Fig. 5.20. It can also be observed that it learns how to increase performance faster than the rest of the protocols.

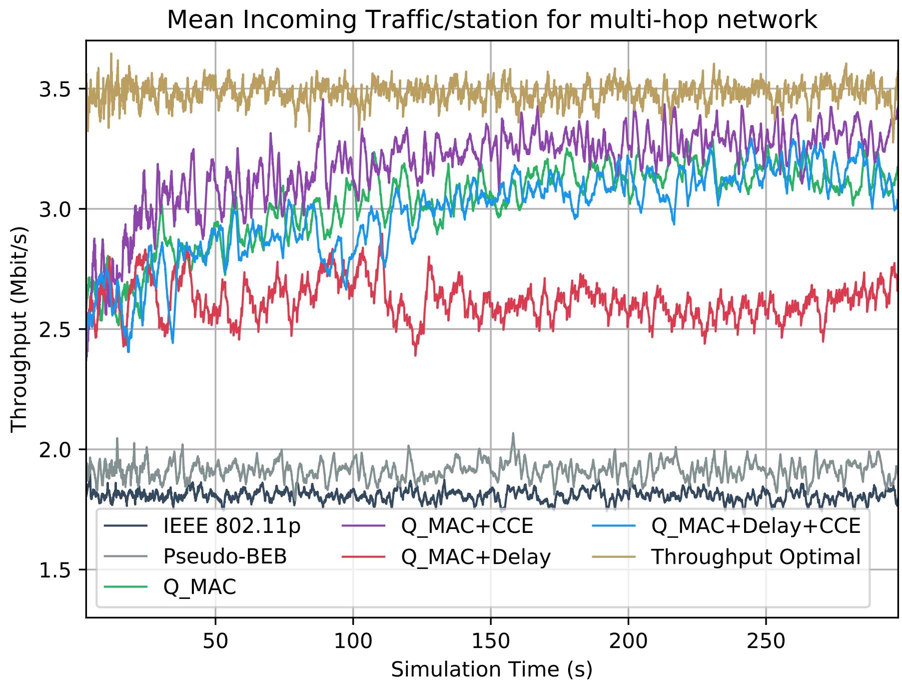


Figure 5.20: Experienced incoming traffic in multi-hop network.

When it comes to latency performance in this scenario, depicted in Fig. 5.21, only unique copies of packets are considered, whether they come from single-hop or two-hop paths, since this reveals more about the performance of the system. Combining CCE and delay awareness in the reward function (Q_MAC+CCE+Delay) with equal bias yields better performance for requirements among 34.5 ms to 47.5 ms, very close to that of Q_MAC which cannot be further controlled. As always, biasing the protocol towards delay with $k_{delay} > k_{CCE}$ in (5.2) can yield even higher delivery rates for latency-sensitive transmissions. Focusing entirely on delay (Q_MAC+Delay) will make the Q-Learning algorithm

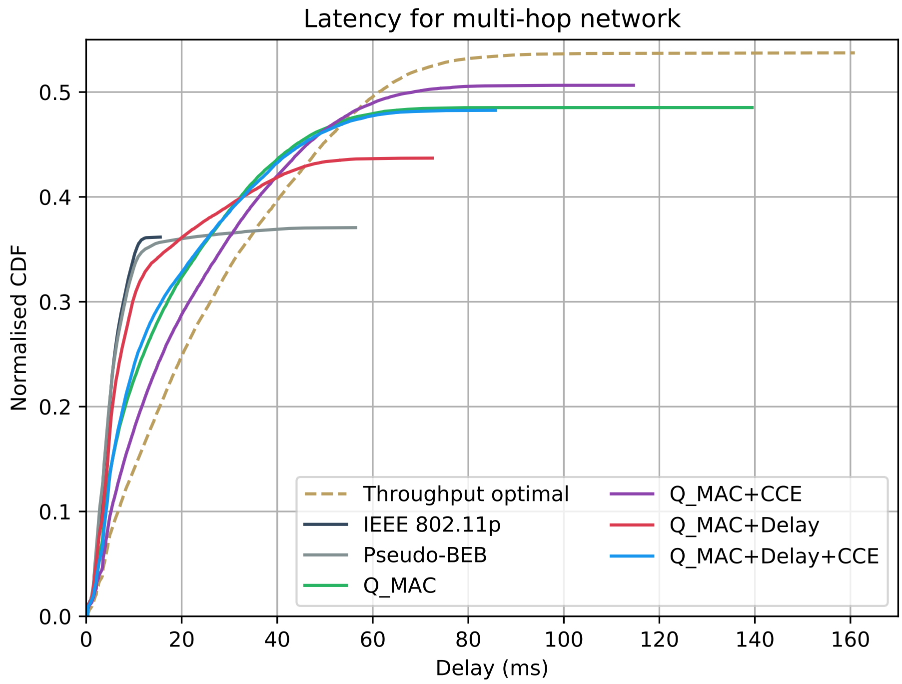


Figure 5.21: End-to-end transmission latency versus PDR in multi-hop network.

outperform all the rest for latencies down to 19 ms in a multi-hop setting. Given that these requirements are not common in multi-hop transmissions, for a requirement of 100 ms the Q_MAC+CCE would again yield the highest performance.

Regarding fairness, again we evaluate it regarding flows of unique packets considering their origin (the vehicle that generated the packet) and disregarding whether they arrive via single-hop (the vehicle that generated the information) or multi-hop (forwarding vehicle) paths. This way we can assess the performance of the multi-hop network regarding its capability to fairly carry information among all vehicles in the RoI, whether they are immediate (single-hop) neighbours of the receiver or not. Results are depicted in Fig. 5.22. Achieved multi-hop fairness is naturally lower, but CCE-enhanced protocols continue to vastly improve on the simpler Q-Learning protocols they are based on, with the best performing just 5% below the optimal fairness found for the system. Q_MAC+CCE can reach $J = 79.4\%$, compared to the simpler Q_MAC with the binary reward function which goes up to $J = 74.6\%$. Similarly, Q_MAC+Delay+CCE goes up to $J = 76.2\%$, while Q_MAC+Delay can reach a maximum of $J = 72.45\%$ within 10 s or 100 original packets transmitted per vehicle.

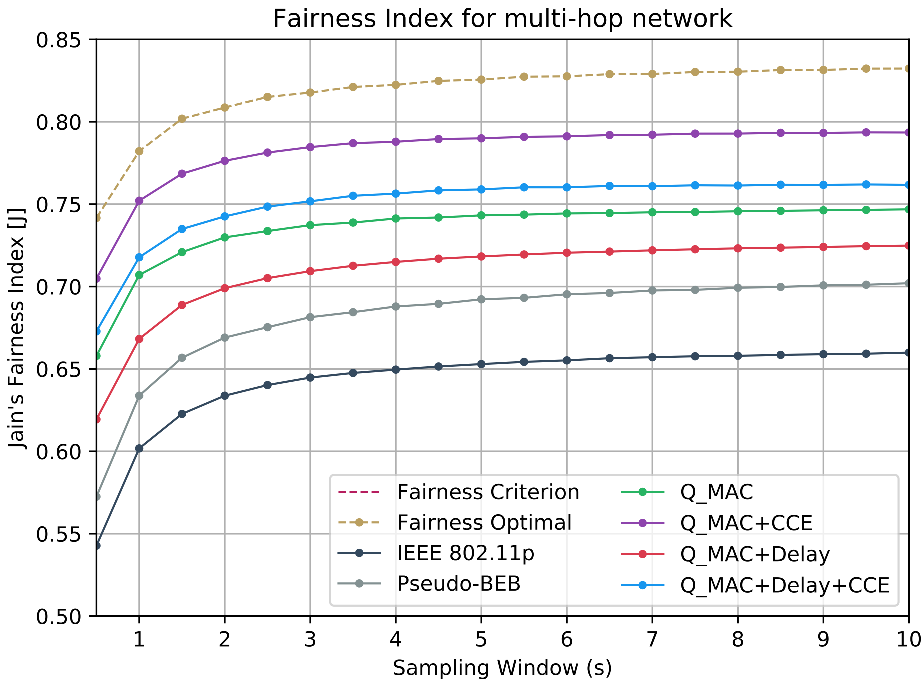


Figure 5.22: Recorded fairness 100 flows in multi-hop network over different sampling windows.

5.5.6 Using different weights for R_{delay}, R_{CCE}

The results presented so far use equal bias $k_{delay} = k_{CCE} = 1$ in (5.2) when examining the effect that Q_MAC+Delay+CCE has on communications. The Q_MAC+Delay implementation favours low latency exchanges but has a cost on total throughput and fairness. A reward system that introduces CCE to a lesser degree is examined, so that we can bias the system towards favouring low-latency exchanges while improving fairness and throughput to some degree.

Fig. 5.23 shows the achieved network wide throughput for the Q-Learning-based MAC protocol when the reward function in (5.2) is deployed with $k_{delay} = 1.35$ and $k_{CCE} = 0.65$, against $k_{delay} = k_{CCE} = 1$ which is the default Q_MAC+Delay+CCE version presented so far, and Q_MAC+Delay ($k_{delay} = 2$, $k_{CCE} = 0$). Measurements are collected for the sparser, 50-vehicle scenario.

Fig. 5.24 reveals the achieved latency for the same Q_MAC implementations. The $k = 1.35$ weight biases the protocol more towards delay with less weight towards a collectively found optimum value. Naturally, it should perform better in lower latencies than Q_MAC+Delay+CCE with $k = 1$, and it does, outperforming all other for latencies of 9.9 ms to 13.5 ms.

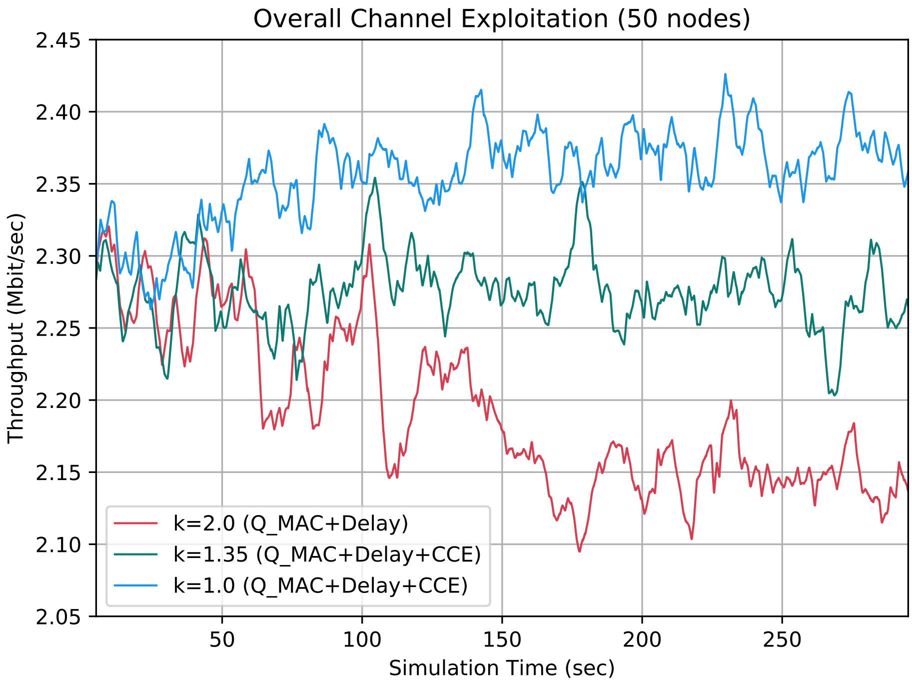


Figure 5.23: Network-wide throughput for different values of k

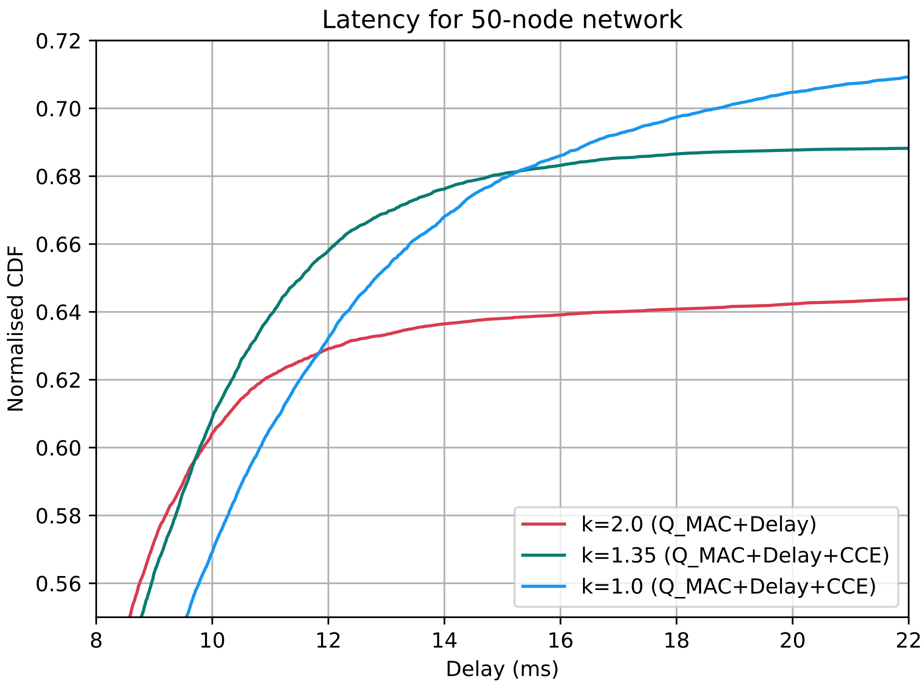


Figure 5.24: Latency for different k values

Fig. 5.25 show how the three protocols stand among each other regarding measured fairness performance. As expected, focusing exclusively on minimising latency has a negative impact on the network's fairness, and can be minimising by also using the CCE function. These results have been collected for the medium density scenario studied in this chapter, which features 50 vehicles transmitting 256-byte packets with $f_b = 30\,Hz$, but the benefits can be more significant as the network contention increases and the need for traffic differentiation via achieved latency is larger.

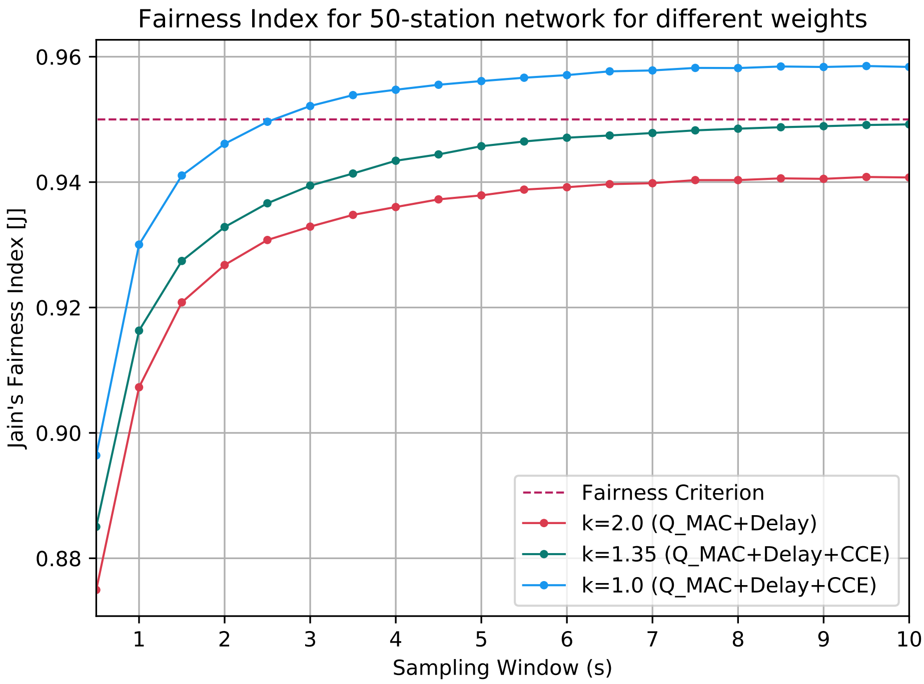


Figure 5.25: Fairness among 50 stations

5.6 Summary

The CCE function for the Q-Learning MAC protocol offers great benefits on real VANET deployment. It accelerates the convergence of the Q-Learning MAC, yielding greater throughput (packet delivery) and fairness performance results.

The reward function presented in this work can be used to trade raw throughput and fairness for lowering transmission latency or the opposite. Additionally, CCE enhanced Q-Learning MAC protocols consistently outperform the protocols they are based on, in terms of fairness and raw throughput. It can be observed that when combining both CCE and delay-awareness mechanisms, a designer can bias the Q-Learning agent towards either high delivery for delay-sensitive traffic or strive for maximum data rates for large exchanges. So there is a clear trade-off when biasing the learning agent: it can strive towards maximum raw throughput and fairness or reliable low-latency transmissions, or a combination of the two, depending on requirements of given application.

Finally, results from evaluating two applications (different packet lengths) simultaneously reveal that the CCE function enhances the ability of the Q-Learning MAC protocol to learn how to operate similarly to the EDCA function. It can enhance service separation by contention priority via CW adaptation, without being explicitly programmed to do so, but depending on the application it tries to accommodate.

Chapter 6

Conclusions and Future Work

6.1 Contributions and Conclusions

We have introduced a contention-based protocol for V2V transmissions that relies on Q-Learning to increase access control efficiency by continuously interacting with the network. Simulations were developed to demonstrate the effectiveness of the MAC protocol. Results prove that the proposed protocol allows the network to scale better to increased network density and achieve higher transfer rates and fairness compared to the IEEE 802.11p standard, while able to maintain a tolerable level of latency. In the following subsections, we briefly highlight the important contributions and conclusions of this thesis.

6.1.1 Evaluation of the DSRC MAC protocol

The main contribution of Chapter 3 is to provide an understanding on the effect the CW parameter has on network performance when there are multiple stations sharing a DSRC channel. Initially, a hardware-based experiment indicates that given high data rates, even a few stations can cause significant channel contention. The CW parameter of the CSMA-based MAC layer is varied, and it can be exploited by a station to gain an advantage in communication over competing peers, and increase its total throughput. Then a preliminary simulation study is presented, that shows the effect of the CW parameter in symmetrical systems. We find that the choice of CW_{min} parameter depends on the need of an application regarding latency and throughput. The effect the parameter has on fairness of bandwidth allocation among stations is also examined. Finally an attempt to replicate and validate the hardware study findings via a more complex simulated network is presented. The acquired insights on the network performance related to CW_{min} also provided a precise guideline for the efficient designs of practical and reliable vehicular communications systems presented in the next chapters of this study.

The detailed contributions include:

- A hardware testbed featuring a novel Linux kernel modification with appropriate user-space software, that allows the adaptation of CW_{min} and other properties of the Linux IEEE 802.11p MAC implementation, supporting the new OCB mode.

- Investigation into the effect the CW_{min} value has on broadcast transmissions of a single vehicle OBU when found in a congested network, enabled by the hardware testbed.

- A complete framework of assessing MAC layer performance incorporating throughput, raw latency, latency versus packet delivery and fairness measurements.

- Investigation of the effect of the CW parameter regarding fair bandwidth allocation. Simulation results using the presented framework regarding the default DSRC MAC for a range of CW values simultaneously applied to all stations for symmetry, indicate a benefit to the overall experience of all users in the network.

- Further investigation into MAC-level fairness. Simulation results using the same framework regarding the default DSRC MAC for a range of CW values in asymmetrical scenarios, indicate that users can greedily acquire a larger portion of the bandwidth if they content for channel access with lower CW values from the peers.

6.1.2 Investigation of the ability of Reinforcement Learning to be used for Channel Access Control

In Chapter 4 we present details regarding the implementation of a MAC protocol based the Q-Learning algorithm. The state-action space and a simple reward function is defined. The exploration-exploitation problem was studied so that a feasible control technique for mobile vehicular networks based on Q-Learning could be designed. The Q-Learning algorithm would have to converge to a (near) optimal solution in relatively short time. To decide on a solution, we compare different exploration-exploitation policies, and end up on a ε-decay solution that forces exploration early on the station's deployment in a VANET and exploits the acquired knowledge as soon as possible. Evaluation regarding Q-Learning parameters and how they affect network performance over time is also presented. A compatible message forwarding protocol is applied at the higher layers. Finally, we vary the network traffic properties such as number of stations and transmitted message length. Both single-hop and multi-hop topologies are tested. The Q-Learning-based MAC protocol for DSRC finds a good solution for all scenarios and outperforms baseline DSRC in terms of achieved packet delivery performance. As a downside, we notice that the algorithm leads to increases in transmission latency as a result of the CW adaptation.

The detailed contributions include:

- A Q-Learning based framework for channel access control compatible with broadcast DSRC transmissions, targeting throughput optimisation.
- Investigation into the exploration-exploitation dilemma and the effect the ε parameter has on the algorithm's convergence time and output regarding correct CW selection. The proposal of the decaying ε-greedy algorithm as a solution towards quickly building access controllers via Q-Learning.
- Investigation of how the RL parameters α and γ as well as the time of exploration affect the achieved network throughput.
- Evaluation of the Q-Learning-based MAC protocol with networking criteria. An assessment of the algorithm's performance regarding achieved packet delivery and latency when compared with existing solutions for different network densities, packet lengths and multi-hop scenarios.

6.1.3 Reward Function Enhancements

Chapter 5 presents the novel Collective Contention Estimation (CCE) function for Q-Learning and how it can enable accommodating different kinds of traffic. Trying to optimize the Q-Learning-based MAC protocol, the CCE function can be used to increase the achieved delivery rate - throughput even more, bringing it very close to the maximum achievable throughput for the various studied scenarios. It also significantly accelerates convergence, yielding performance benefits earlier than the baseline reward mechanism. Throughput fairness among vehicle-stations is also significantly improved. The mechanism is evaluated for different network densities, different simultaneous services and both single-hop and multi-hop topologies. The service separation capability of the algorithm is also evaluated for two simultaneous applications in a network. Additionally, by providing a way to combine multiple goals, the protocol can become latency-aware so that higher delivery rates and fairness can be achieved even for delay-sensitive communications. We evaluate the ability of the mechanism to satisfy lower-latency applications

The detailed contributions include:

- A novel reward mechanism named Collective Contention Estimation (CCE), which is compatible with the Q-Learning-based MAC protocol is presented. By incorporating the $backoff$ copying technique, it promotes symmetry in the network regarding the employed CW by relevant vehicle-stations. This way, throughput gains are accelerated and larger. Throughput fairness is also greatly enhanced in the network, regarding both the short and long-term.

- An investigation of how the CCE mechanism enables the Q-Learning-based MAC protocol to better handle different service separation regarding contending priority of stations depending on application.

- A method for the Q-Learning algorithm to target multiple networking objectives so that applications with various latency requirements can be better accommodated by the protocol is suggested. The method can also be used to bias the algorithm towards focusing to a greater degree towards an objective rather than another.

6.2 Future Research Subjects

6.2.1 EDCA-like priorities via Q-Learning for Vehicular Data Traffic

The Q-Learning-based MAC protocols introduced in this thesis have been proven to be able to tackle contention among multiple stations sharing the common wireless medium. According to the experimental results collected in this thesis, they can be adapted to handle internal and external contention among different applications considering their urgency.

Further work could focus on enhanced bandwidth allocation among transmissions different kinds of traffic. Q-Learning has already been examined as a solution to resource provisioning and QoS enhancement for vehicular cloud [79]. Enhanced bandwidth allocation among different services could be achieved by combining the proposed protocol with sliding window techniques which restrict the number of CW levels per application depending on its priority. The QoS for different applications could be further enhanced by having multiple Q-Learning MAC agents in a station, that are trained based on the type and priority of traffic they have to accommodate. The Q-Learning MAC protocol could also be trained in scenarios of fast moving vehicles where the time window of opportunity for data exchange with other vehicles or RSUs is smaller than usual. A mobility-aware EDCA function could be designed based on Q-Learning, to separate traffic not only based on application requirements but the contact time vehicles have with each other or some RSU.

6.2.2 VANET MAC Layer design based on Deep Q-Learning

The vanilla Q-Learning algorithm can work effectively when the (s, a) space is small enough that can be explored in sensible time and consequently just a look-up table has to be maintained for the update of the Q-value. However, it is impossible to apply the Q-Learning with look-up tables when the (s, a) space becomes very large, as in a joint resource management problem. Realistically, a large number of (s, a) pairs may be rarely visited, thus the corresponding Q-values would not be updated as frequently as needed, leading to a much longer required time for convergence. To solve this problem, it is common to use a function approximator to estimate the Q-value function as $Q(s, a, \theta) \approx Q(s, a)$. It can either be a linear function approximator, or a non-linear function approximator such as a neural network, referred to as a Q-network [68]. Once θ is determined, the Q-values $Q(s_t, a_t)$ will be the outputs of the neural network.

Q-Networks have been studied for their potential application towards solving networking problems, such as network resources allocation for data centres [26], and naturally research has extended towards the emerging vehicular networks. Work in [110] presents using deep reinforcement learning to handle resource allocation and the broadcast scheduling jointly for C-V2X. For the VANET congestion control problem studied, it could be examined whether the neural network can address complex mappings between the controlled stations parameters or network data and the desired output based on a large amount of training data, which would be used to determine the Q-values. The Q-network has to be trained with a large amount of simulated data, which are generated from interactions of agents and an environment simulator. Coupling OMNeT++ with Google's Tensorflow API would allow researchers to train and evaluate various neural network-based learning algorithms in terms of their networking performance, without the need to implement and train the neural networks themselves in OMNeT++, thus allowing further research into applying the Q-Learning algorithm in large (s, a) spaces.

6.2.3 Denial-of-Service Mitigation in VANETs

An interesting future research direction is vulnerability mitigation, especially of Denial-of-Service (DoS) attacks which could be particularly dangerous in safety-oriented networks such as VANETs. Malicious actors can generate artificial contention to jam useful transmissions of their peers. By employing techniques to estimate the $backoff$ period of some station or RSU, multiple stations can be placed near and RSU to synchronise their transmissions (Distributed DoS attack), as seen in [20]. According to this study, this becomes easier in VANETs, since IEEE 802.11p by default uses small CW sizes, especially for high priority, safety-oriented traffic (that goes through faster ACs). They suggest that by using larger CW sizes this kind of attacks becomes harder to perform, since a lot of time and malicious stations would be needed.

There has been existing work focusing on DoS attack detection and mitigation using machine learning techniques, as well as some focusing on RL. Such work is presented in [74], which studies the detection of a stealthy DoS attack in a Software Defined Network using Q-Learning. Work in [64] explores the possibility of DDoS Response using RL in large-scale network topologies. The proposed Q-Learning-based protocols could be used as a base for intelligent mitigation or avoidance of MAC-layer DoS attacks by synchronisation. More research could be focused on this matter, as more sophisticated security techniques would be needed in real deployments.

6.2.4 Hardware deployment of the Q-Learning-based MAC

We have showed that adaptation of the CW parameter in the IEEE 802.11p Link Layer is feasible in the Linux Kernel. Furthermore, it can be done from userspace applications communicating with system calls with the Kernel networking stack. With the CW adaptation having an observable and measurable effect on communication performance of the IEEE 802.11p stations, a next step would be incorporating smart adaptation algorithms, such as the ones presented in this work into hardware deployments. The Q-Learning-based CW adaptation protocol itself could be implemented as a userspace application for simplicity. A larger VANET of multiple transmitters would have to be deployed to replicate the simulation setups presented in the thesis.

The current system evaluates IAT at the receiving IEEE 802.11p station. To maintain that measurement technique for producing feedback for the Q-Learning-based MAC, the IAT measurements of transmissions or $\overline{IAT}$ over multiple transmissions at a time can be compared to some pre-defined or adaptive threshold that indicates the quality of communications. Then populate each controller's Q-table (s, a) element (where s is a CW level), the feedback (reward r) would have to be transmitted back to the original sender, in the same form as presented throughout the thesis. The reward can either be binary (1 if the achieved IAT is below the threshold or -1 if it exceeds it), or a percentage indicating how far of from the desired value is the achieved IAT.

References

[1] 5g; service requirements for enhanced v2x scenarios (3gpp ts 22.186 version 15.3.0 release 15). etsi, 2018. 97

[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 amendment 6: Wireless access in vehicular environments," ieee, pp. 1–51, 2010. 11

[3] N. Aihara and K. Adachi. Orthogonal resource allocation using svm for csma/ca. In *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 779–784, Nov 2018. 25

[4] H. A. A. Al-Rawi, M. A. Ng, and K.-L. A. Yau. Application of reinforcement learning to routing in distributed wireless networks: a review. *Artificial Intelligence Review*, 43(3):381–416, Mar 2015. 26

[5] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan. A comprehensive survey on vehicular Ad Hoc network. *Journal of Network and Computer Applications*, vol. 37:pp. 380–392, 2014. 8

[6] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys Tutorials*, 16(4):1996–2018, Fourthquarter 2014. 24

[7] S. Amuru, Y. Xiao, M. van der Schaar, and R. M. Buehrer. To send or not to send - learning mac contention. In *Proc. of IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2015. 28

[8] F. Arena and G. Pau. An overview of vehicular communications. *Future Internet*, 11:27, 01 2019. 2

[9] M. M. Artimy, W. Robertson, and W. J. Phillips. Assignment of dynamic transmission range based on estimation of vehicle density. In *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '05, pages 40–48, New York, NY, USA, 2005. 64

[10] A. Autolitano, M. Reineri, R. M. Scopigno, C. Campolo, and A. Molinaro. Understanding the channel busy ratio metrics for decentralized congestion control in vanets. In *Proc. of International Conference on Connected Vehicles and Expo (ICCVE)*, pages 717–722, Nov 2014. 65

[11] B. Bako, I. Rikanovic, F. Kargl, and El. Schoch. Adaptive topology based gossiping in vanets using position information. In *Proc. of Mobile Ad-Hoc and Sensor Networks*, pages 66–78, Berlin, Heidelberg, 2007. 64

[12] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility - an Overview. Number c, pages 63–68, 2011. 43

[13] R. Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. 25

[14] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau. Fairness and its impact on delay in 802.11 networks. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, volume 5, pages 2967–2973, Nov 2004. 48

[15] G. Berger-Sabbatel, A. Duda, M. Heusse, and F. Rousseau. Short-term fairness of 802.11 networks with several hosts. In *Proc. of Mobile and Wireless Communication Networks*, pages 263–274, Boston, MA, 2005. 48

[16] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Macaw: A media access protocol for wireless lan's. *SIGCOMM Computer Communication Review*, 24(4):212–225, October 1994. 23, 83, 85

[17] Yuanguo Bi, H. Zhou, W. Zhuang, and H. Zhao. *Overview of Safety Message Broadcast in Vehicular Networks*, pages 11–24. Springer International Publishing, Cham, 2017. 14

[18] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000. 28

[19] B.E. Bilgin and V.C. Gungor. Performance comparison of ieee 802.11 p and ieee 802.11 b for vehicle-to-vehicle communications in highway, rural, and urban areas. *International Journal of Vehicular Technology*, 2013. 49

[20] S. Biswas, J. Mišić, and V. Mišić. Ddos attack on wave-enabled vanet through synchronization. In *Proc. of IEEE Global Communications Conference (GLOBECOM)*, pages 1079–1084, Dec 2012. 118

[21] J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Proc. of Advances in neural information processing systems*, pages 671–678, 1994. 26

[22] M. Cagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux. On selfish behavior in csma/ca networks. In *Proc. of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 4, pages 2513–2524 vol. 4, March 2005. 40

[23] A. Cailean, B. Cagneau, L. Chassagne, V. Popa, and M. Dimian. A survey on the usage of dsrc and vlc in communication-based vehicle safety applications. In *Proc. of IEEE 21st Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, pages 69–74, Nov 2014. 23

[24] C. Campolo, A. Molinaro, A. Vinel, and Y. Zhang. Modeling prioritized broadcasting in multichannel vehicular networks. *IEEE Transactions on Vehicular Technology*, 61(2):687–701, Feb 2012. 24

[25] C. Campolo, A. Vinel, A. Molinaro, and Y. Koucheryavy. Modeling broadcasting in ieee 802.11p/wave vehicular networks. *IEEE Communications Letters*, 15(2):199–201, February 2011. 23

[26] L. Chen, J. Lingys, K. Chen, and F. Liu. Auto: Scaling deep reinforcement learning for datacenter-scale automatic traffic optimization. In *Proc. of the 2018 Conference of the ACM Special Interest Group on Data Communication(SIGCOMM)*, pages 191–205, New York, NY, USA, 2018. ACM. 118

[27] Q. Chen, D. Jiang, and L. Delgrossi. Ieee 1609.4 dsrc multi-channel operations and its implications on vehicle safety communications. In *Proc. of IEEE Vehicular Networking Conference (VNC)*, pages 1–8, Oct 2009. 12

[28] A. Chriki, H. Touati, and H. Snoussi. Svm-based indoor localization in wireless

sensor networks. In *Proc. of 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1144–1149, June 2017. 25

[29] European Commission. Commission delegated regulation (eu): supplementing directive 2010/40/eu of the european parliament and of the council with regard to the deployment and operational use of cooperative intelligent transport systems. March 2019. 9, 23

[30] S. Eichler. Performance evaluation of the ieee 802.11p wave communication standard. In *Proc. of IEEE 66th Vehicular Technology Conference (VTC)*, pages 2199–2203, 2007. 4

[31] E. Even-Dar and Y. Mansour. Learning rates for q-learning. *J. Mach. Learn. Res.*, 5:1–25, December 2004. 73

[32] E. C. Eze, S.-J. Zhang, E.-J. Liu, and J. C. Eze. Advances in vehicular ad-hoc networks (vanets): Challenges and road-map for future development. 13(1), 2016. 63

[33] M. Faezipour, M. Nourani, A. Saeed, and S. Addepalli. Progress and challenges in intelligent vehicle area networks. *Commun. ACM*, 55(2):90–100, feb 2012. 9

[34] F. Fu and M. van der Schaar. Learning to compete for resources in wireless stochastic games. *IEEE Transactions on Vehicular Technology*, 58(4):1904–1919, May 2009. 26

[35] D. Ghose and F. Y. Li. Enabling retransmissions for achieving reliable multicast communications in wsns. In *Proc. of IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2016. 63

[36] M. Gonzalez-Martin, M. Sepulcre, R. Molina-Masegosa, and J. Gozalvez. Analytical models of the performance of c-v2x mode 4 vehicular communications, 10 2018. 10

[37] Z. Hameed Mir and F. Filali. Lte and ieee 802.11p for vehicular networking: a performance evaluation. *EURASIP Journal on Wireless Comm. and Netw.*, 2014(1):89, 2014. 10

[38] H. Harb, A. Makhoul, D. Laiymani, A. Jaber, and R. Tawil. K-means based clustering approach for data aggregation in periodic sensor networks. In *Proc. of IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 434–441, Oct 2014. 25

[39] V. P. Harigovindan, A. V. Babu, and L. Jacob. Ensuring fair access in ieee 802.11p-based vehicle-to-infrastructure networks. *EURASIP Journal on Wireless Communications and Networking*, (1):168, May 2012. 24

[40] M. I. Hassan, H. L. Vu, and T. Sakurai. Performance analysis of the ieee 802.11 mac protocol for dsrc safety applications. *IEEE transactions on vehicular technology*, 60(8):3882–3896, 2011. 24

[41] M. Impett, M. S. Corson, and V. Park. A receiver-oriented approach to reliable broadcast in ad hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference. Conference Record*, volume 1, pages 117–122 vol.1, Sept 2000. 20

[42] R. Jain, D.-M. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *CoRR*, cs.NI/9809099, 1998. 47

[43] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich. Design of 5.9GHz DSRC-based Vehicular Safety Communication. *IEEE Wireless Communications*, 13(5):36 – 43, 2006. 12, 14, 15, 22

[44] J. Joung. Machine learning-based antenna selection in wireless communications. *IEEE Communications Letters*, 20(11):2241–2244, Nov 2016. 25

[45] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996. 58

[46] A.I. Khan, C. Goh, C. Lim, and S. Sharma. Quality of service assessment in connected vehicles. In *Proc. of the Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR 2017)*, 2017. 25

[47] F. Klingler, F. Dressler, and C. Sommer. Ieee 802.11p unicast considered harmful. In *Proc. of IEEE Vehicular Networking Conference (VNC)*, pages 76–83, Dec 2015. 20, 63, 66

[48] B. Kloiber, J. Härri, T. Strang, and S. Sand. Bigger is better — combining contention window adaptation with geo-based backoff generation in dsrc networks. In *Proc. of International Conference on Connected Vehicles and Expo (ICCVE)*, pages 227–233, 2014. 24

[49] T. Kosch, C. J. Adler, S. Eichler, C. Schroth, and M. Strassberger. The scalability problem of vehicular ad hoc networks and how to solve it. *IEEE Wireless Communications*, 13(5):22–28, October 2006. 22

[50] D. Krajzewicz, M. Hartinger, G. Hertkorn, P. Mieth, C. Rössel, J. Zimmer, and P. Wagner. Using the road traffic simulation "sumo" for educational purposes. In *Proc. of Traffic and Granular Flow*, pages 217–222, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 43

[51] L. Le, A. Festag, R. Baldessari, and W. Zhang. CAR-2-X Communication in Europe. In *Vehicular Networks*, Chapman & Hall/CRC Computer & Information Science Series, pages 1–32. Chapman and Hall/CRC, mar 2009. 8

[52] J.-M. Lee, M.-S. Woo, and S.-G. Min. Performance analysis of wave control channels for public safety services in vanets. *International Journal of Computer and Communication Engineering*, 2(5):563, 2013. 49

[53] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. Comparative study of wireless network simulators. In *Proc. of Seventh International Conference on Networking (ICN)*, pages 517–523. IEEE, 2008. 42

[54] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2(2):12–22, June 2007. 14

[55] T. Li, T. Tang, and C. Chang. A new backoff algorithm for ieee 802.11 distributed coordination function. In *Proc. of Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 3, pages 455–459, Aug 2009. 21

[56] Y. J. Li. An Overview of the DSRC / WAVE Technology. *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, pages 544–558, 2012. 13

[57] Z. Li, A. Das, A. K. Gupta, and S. Nandi. Performance analysis of ieee 802.11 dcf: Throughput, delay, and fairness. *Unpublished. Available on 16th Mar*, 2011. 23

[58] W. Liang, Z. Li, H. Zhang, Y. Sun, and R. Bie. Vehicular Ad Hoc Networks: Architectures, Research Issues, Challenges and Trends. *International Journal of Distributed Sensor Networks*, 2015:102–113, 2014. 8

[59] R. Lisovỳ, M. Sojka, and Z. Hanzálek. Ieee 802.11 p linux kernel implementation. 2014. 32

[60] Z. Liu and I. Elhanany. Rl-mac: A qos-aware reinforcement learning based mac protocol for wireless sensor networks. In *Proc. of IEEE International Conference on Networking, Sensing and Control*, pages 768–773, 2006. 28

[61] Z. Liu, Z. Liu, Z. Meng, X. Yang, L. Pu, and L. Zhang. Implementation and performance measurement of a v2x communication system for vehicle and pedestrian safety. *International Journal of Distributed Sensor Networks*, 12, 09 2016. 40

[62] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark. Connected vehicles: Solutions and challenges. *IEEE Internet of Things Journal*, 1(4):289–299, Aug 2014. 23

[63] M. A. Mahmood and W. K. G. Seah. Event reliability in wireless sensor networks. In *Proc. of Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 377–382, Dec 2011. 63

[64] K. Malialis and D. Kudenko. Large-scale ddos response using cooperative reinforcement learning. In *Proc. of 11th European Workshop on Multi-Agent Systems (EUMAS)*, 2013. 119

[65] M. J. Mataric. Reward functions for accelerated learning. In *Machine Learning Proceedings 1994*, pages 181–189. Elsevier, 1994. 83

[66] Y. Mertens, M. Wellens, and P. Mahonen. Simulation-based performance evaluation of enhanced broadcast schemes for IEEE 802.11-based vehicular networks. *IEEE Vehicular Technology Conference*, pages 3042–3046, 2008. 22

[67] L. Miao, K. Djouani, B. J. V. Wyk, and Y. Hamam. Performance Evaluation of IEEE 802.11p MAC Protocol in VANETs Safety Applications. In *Proc. of Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, number i, pages 1663–1668, 2013. 21

[68] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. 118

[69] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar. Broadcast storm mitigation techniques in vehicular ad hoc networks. *IEEE Wireless Communications*, 14(6):84–94, December 2007. 64

[70] G. Naus, R. Vugts, J. Ploeg, R. v. d. Molengraft, and M. Steinbuch. Cooperative adaptive cruise control, design and experiments. In *Proc. of the 2010 American Control Conference*, pages 6145–6150, June 2010. 4, 47

[71] R. Oliveira, L Bernardo, and P. Pinto. Performance Analysis of the IEEE 802.11 Distributed Coordination Function with Unicast and Broadcast Traffic. In *Proc. of the 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2006. 15

[72] C. I. Paredes, A. M. Mezher, M. Aguilar-Igartua, and J. Forné. Game-theoretical design of an adaptive distributed dissemination protocol for vanets. In *Sensors*, 2018. 45

[73] M. Petrik and B. Scherrer. Biasing approximate dynamic programming with a lower discount factor. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1265–1272. Curran Associates, Inc., 2009. 73

[74] T. Phan, T. Gias, S. T. Islam, T. Huong, N. Thanh, and T. Bauschert. Q-mind: Defeating stealthy dos attacks in sdn with a machine-learning based defense framework, 07 2019. 119

[75] K. C. K. Priya and S. Terence. Retp: Reliable event transmission protocol in a wireless sensor network. In *Proc. of IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, pages 181–188, March 2013. 63

[76] H. J. F. Qiu, I. W. H. Ho, C. K. Tse, and Y. Xie. A Methodology for Studying 802.11p VANET Broadcasting Performance with Practical Vehicle Distribution. *IEEE Transactions on Vehicular Technology*, 64(10):4756–4769, 2015. 19

[77] R. Reinders, M. van Eenennaam, G. Karagiannis, and G. Heijenk. Contention window analysis for beaconing in vanets. In *Proc. of 7th International Wireless Communications and Mobile Computing Conference*, pages 1481–1487, July 2011. 24

[78] G. V. Rossi and K. K. Leung. Optimised csma/ca protocol for safety messages in vehicular ad-hoc networks. In *Proc. of IEEE Symposium on Computers and Communications (ISCC)*, pages 689–696, July 2017. 3, 22

[79] M. Salahuddin, A. Al-Fuqaha, and M. Guizani. Reinforcement learning for resource provisioning in the vehicular cloud. *IEEE Wireless Communications*, 23:128–135, 08 2016. 117

[80] W. Samek, S. Stanczak, and T. Wiegand. The convergence of machine learning and communications. *CoRR*, abs/1708.08299, 2017. 24, 28

[81] P. Sasikumar and S. Khara. K-means clustering in wireless sensor networks. In *Proc. of Fourth International Conference on Computational Intelligence and Communication Networks*, pages 140–144, Nov 2012. 25

[82] K. Shah and M. Kumar. Distributed independent reinforcement learning (dirl) approach to resource management in wireless sensor networks. In *Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–9, Oct 2007. 26

[83] P. Shankar, T. Nadeem, J. Rosca, and L. Iftode. Cars: Context-aware rate selection for vehicular networks. In *Proc. of 16th IEEE International Conference on Network Protocols (ICNP)*, pages 1–12, 12 2008. 64

[84] A. Dalili Shoaei, M. Derakhshani, S. Parsaeifard, and T. Le-Ngoc. Mdp-based mac design with deterministic backoffs in virtualized 802.11 wlans. *IEEE Transactions on Vehicular Technology*, 65(9):7754–7759, Sept 2016. 28

[85] P. Singh and S. Agrawal. Tdoa based node localization in wsn using neural networks. In *Proc. of International Conference on Communication Systems and Network Technologies*, pages 400–404, April 2013. 25

[86] S. Son and K.-J. Park. Beat: Beacon inter-reception time ensured adaptive transmission for vehicle-to-vehicle safety communication. *Sensors*, 19:3061, 07 2019. 64

[87] A. B. Souza, A. L. B. Barros, P. Barros, A. Sérgio de S. Vieira, F. Maciel, and J. Celestino Jr. An adaptive mechanism for access control in vanets. 03 2019. 24

[88] R. Stanica, E. Chaput, and A. L. Beylot. Enhancements of IEEE 802.11p protocol for access control on a VANET control channel. *IEEE International Conference on Communications*, 2011. 15, 22, 23

[89] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. 59

[90] K. Tharatipayakul, S. Gordon, and K. Kaemarungsi. iack: Implicit acknowledgements to improve multicast reliability in wireless sensor networks. In *Proc. of 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6, May 2014. 63

[91] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein. Vehicle-to-vehicle communication: Fair transmit power control for safety-critical information. *IEEE Transactions on Vehicular Technology*, 58(7):3684–3703, 2009. 22

[92] Q. Tse, W. Si, and J. Taheri. Estimating contention of ieee 802.11 broadcasts based on inter-frame idle slots. In *Proc. of IEEE Conference on Local Computer Networks - Workshops*, pages 120–127, Oct 2013. 28

[93] A. Turley, K. Moerman, A. Filippi, and V. Martinez. C-ITS: Three observations on LTE-V2X and ETSI ITS-G5—A comparison. Technical report, NXP, 2018. 10

[94] US National Public Safety Telecommunications Council. 5.9 ghz dsrc operational concept introduction. 66

[95] A. Varga and R. Hornig. An overview of the omnet++ simulation environment. In *Proc. of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Simutools, pages 60:1–60:10, 2008. 42

[96] V. Vukadinovic, K. Bakowski, P. Marsch, I. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, L. David, and I. Thibault. 3gpp c-v2x and ieee 802.11p for vehicle-to-vehicle communications in highway platooning scenarios. *Ad Hoc Networks*, 74, 03 2018. 10

[97] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang. Machine learning for networking: Workflow, advances and opportunities. *IEEE Network*, 32(2):92–99, March 2018. 24, 26

[98] X. Wang and G. B. Giannakis. Csma/cca: A modified csma/ca protocol mitigating the fairness problem for ieee 802.11 dcf. *EURASIP Journal on Wireless Communications and Networking*, 2006(1):039604, Mar 2006. 4, 21, 23, 83, 85

[99] Y. Wang, A. Ahmed, B. Krishnamachari, and K. Psounis. Ieee 802.11p performance evaluation and protocol enhancement. In *Proc. of IEEE International Conference on Vehicular Electronics and Safety*, pages 317–322, 2008. 4

[100] C. J. C. H. Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, University of Cambridge, Cambridge, UK, May 1989. 26

[101] C. J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992. 26, 58

[102] C. Wu, S. Ohzahata, Y. Ji, and T. Kato. A mac protocol for delay-sensitive vanet applications with self-learning contention scheme. In *Proc. of IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pages 438–443, Jan 2014. 23, 28, 85, 89, 96

[103] Q. Wu, S. Nie, Pi. Fan, Z. Li, and C. Zhang. A swarming approach to optimize the one-hop delay in smart driving inter-platoon communications. *Sensors*, 18:3307, Sept. 2018. 23, 24

[104] X. Xia and Z. Niu. Enhanced DCF MAC scheme for providing differentiated QoS in ITS. In *Proc .of the 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, pages 280–285, 2004. 18

[105] D. Xu, T. Sakurai, and H. L. Vu. An access delay model for ieee 802.11e edca. *IEEE Transactions on Mobile Computing*, 8(2):261–275, Feb 2009. 16

[106] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Vehicle-to-vehicle safety messaging in dsrc. In *Proc. of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET, pages 19–28, New York, NY, USA, 2004. ACM. 3, 11, 19, 22, 42

[107] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Medium Access Control Protocol Design for Vehicle – Vehicle Safety Messages. *IEEE Transactions on Vehicular Technology*, 56(2):499–518, Mar. 2007. 15

[108] Z. Xu, X. Li, X. Zhao, M. H Zhang, and Z. Wang. Dsrc versus 4g-lte for connected vehicle applications: A study on field experiments of vehicular communication performance. *Journal of advanced transportation*, 435, 08 2017. 4, 24

[109] L. Yang, X. Cheng, M. Ghogho, E. Ayanoglu, T. Huang, and N. Zheng. Guest editorial special issue on iot on the move: Enabling technologies and driving applications for internet of intelligent vehicles (ioiv). *IEEE Internet of Things Journal*, 6(1):1–5, Feb 2019. 2

[110] H. Ye, G. Y. Li, and B.-H. F. Juang. Deep reinforcement learning based resource allocation for v2v communications. *arXiv preprint arXiv:1805.07222*, 2018. 118

[111] X. Zhang and D. Qiao. *Quality, Reliability, Security and Robustness in Heterogeneous Networks: 7th International Conference on Heterogeneous Networking for Quality.* Springer Publishing Company, Incorporated, 2012. 11

[112] X. M. Zhang, E. B. Wang, J. J. Xia, and D. K. Sung. A neighbor coverage-based probabilistic rebroadcast for reducing routing overhead in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 12(3):424–433, March 2013. 64

[113] W. Zhao, D. Liu, and Y. Jiang. Distributed neural network routing algorithm based on global information of wireless sensor network. In *Proc. of WRI International Conference on Communications and Mobile Computing*, volume 1, pages 552–555, Jan 2009. 25

Appendix A

OMNeT++ extension for reliable broadcasting via implicit ACKs

Listing A.1: Simulation extension for probabilistic retransmissions, to support reliable broadcasting (ACKs) and multi-hop.

```
//A collection of unique vehicle IDs.
std::vector<int> Vvehicles;

if(wsm->getHops()==0)
{
        if(std::find(Vvehicles.begin(),Vvehicles.end(),
        wsm->getSenderAddress()) !=Vvehicles.end())
        {
                /*Do nothing more, this is a packet coming
                from an already known vehicle.*/
        }
        else
        {
                //Add vehicleID to collection.
                Vvehicles.push_back(wsm->getSenderAddress());
        }
}

/* Refresh number of neighbours measurement
        every 500 ms */
if(simTime().dbl()>rec_timeVehicles+0.5)
{
```

```
        neighbours = Vvehicles.size();
        Vvehicles.clear(); //reset vector holding vehicleIDs
        P_ack = N_ack/neighbours;

        //time reset
        rec_timeVehicles = simTime().dbl();

        //write log file
        std::ofstream NeighboursLog;
        NeighboursLog.open (directory + "neighbours_"
        + std::to_string(myId) +".csv",std::ios_base::app);
        NeighboursLog<<Vvehicles.size()<<","<<P_ack<<
        ","<<rec_timeVehicles<<std::endl;
        NeighboursLog.close();
}

//Checks whether a packet should be retransmitted or not.
if (wsm->getHops()<MAX_HOPS && wsm->getRateType()==thisRateType)
{
        // Forward Packet
        if(uniform(0,1)<P_ack)
        {
                WaveShortMessage *copy = wsm->dup();
                copy->setHops(copy->getHops()+1);
                //copy->setRateType(rateType);
                sendWSM(copy);
        }
}
```

Appendix B

Search for CSMA performance upper bound

Algorithm 3 Sequential Search for Maximum Throughput and Fairness

$CW[7] = [3, 7, 15, 31, 63, 127, 255]$
for $CW_{level} \leftarrow 0; CW_{level} < 6; CW_{level} ++$ **do**
    for $i_{vehicles} \leftarrow 0; i_{vehicles} < N_{vehicles}; i_{vehicles} ++$ **do**
        $Vehicle[i_{vehicles}]$.**CW** $\leftarrow CW[CW_{level}]$
        $Throughput[CW_{level}] \leftarrow MeasureThroughput$
        $JainFairness[CW_{level}] \leftarrow MeasureFairness$
    end for
end for
$Throughput_{MAX} = 0$
$JainFairness_{MAX} = 0$
for $CW_{level} \leftarrow 0; CW_{level} < 6; CW_{level} ++$ **do**
    if $Throughput[CW_{level}] > Throughput_{MAX}$ **then**
        $Throughput_{MAX} \leftarrow Throughput[CW_{level}]$
    end if
    if $JainFairness[CW_{level}] > JainFairness_{MAX}$ **then**
        $JainFairness_{MAX} \leftarrow JainFairness[CW_{level}]$
    end if
end for
Return $Throughput_{MAX}, JainFairness_{MAX}$