Cognitive Radio Enabled Vehicular Ad Hoc Networks

Rajith Chaminda Abeywardana

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering,
The University of Auckland, New Zealand.

July 2019

Abstract

Vehicular communication is fast gaining popularity to facilitate safer, smarter, efficient and sustainable transportation through intelligent transport systems. Vehicular Ad Hoc Networks (VANETs) are decentralised wireless networks enabling direct communication between vehicles on the fly. VANETs can be used to exchange safety-related information as well as provide entertainment and convenience to travellers. These applications require high data throughputs, especially for the latter, making the dedicated bandwidth available for VANETs insufficient. Cognitive radio technology, which utilises existing under-utilised spectrum resources through opportunistic access, satisfies these throughput demands of VANETs.

A simulation framework for application service management in VANETs was designed and developed to bridge the gap between existing simulators. This framework also supports modelling of multi-radio multi-channel communications and radio channel access mechanisms as per the IEEE WAVE standards. An additional simulation framework was developed for cognitive radio enabled VANETs to simulate vehicular nodes, primary user transmissions, spectrum sensing, and spectrum hand-offs.

A three-state sensing model that uses a carrier sensing mechanism in addition to energy detection was developed to distinguish between the primary and secondary users. A higher probability of detection was observed with a sensing interval much shorter than the transmission signal duration of the primary user. Furthermore, the highest probability of detection was recorded with a sensing interval of 10% of the primary user signal duration. A dynamic sensing technique was employed as an attempt to reduce the system overheads, which inevitably increase with the reduced sensing interval, without compromising the accuracy of sensing. With this technique, the probability of detection remained unchanged, but a significant reduction of up to 90% was observed in the system overheads.

An algorithm was developed to continuously evaluate the data congestion of all dedicated channels available for vehicular communication and select the least congested channel for cognitive radio signalling. As the framework addresses all aspects with attention to finer details, it is a compelling platform for all firmware and application developers in industrial and research communities alike.

To my loving family Dulsha, Nethuli and Mineli…

Acknowledgements

First, I would like to thank the University of Auckland, for offering me the University of Auckland Doctoral Scholarship, which made a big difference in facing this tedious and testing journey of studying towards a PhD.

I would like to thank my supervisor Professor Kevin Sowerby for his invaluable support, guidance and motivation throughout my PhD. His expertise in the area of telecommunications has been a pillar of strength in advancing my research. His guidance and support through this PhD journey have been invaluable.

I would also like to extend my gratitude to my second supervisor Associate Professor Stevan Berber, for his guidance, support and encouragement towards completing this research. In addition, I would like to thank the technical staff in the ECE department, especially Mr. Wai Yeung for his technical support and laboratory assistance.

I would also like to thank my friends at the electrical department Dr. Mostafa Mehrabi, Ms. Maryam Hemmati, Mr. Hamidreza Rahanmaee, Ms. Yuan Liu, Mr. Yuan Song, Ms. Nadhiya Mohammad, Ms. Hanie Mehdinezhad, Ms. Latha Murugesan, Ms. Suchetha Sehgal, Ms. Thiaza Thasthakeer, Mr. Yuan Goh, Mr. Sujith Ramachandran, Dr. Maja Krivokuća, Dr. Vedrana Krivokuća, Mr. Benjamin Tan, Mr. Arash Tayebi, Dr. Liang Huang, Dr. Saranga Weerasinghe, Mr. Joseph Tsoi and other peers for their, guidance, support and companionship during this journey.

My sincere gratitude goes out to Chanika, Nadun, Dhanu, Keshav, Aunty Manel, Uncle Dayajith, Aunty Jayantha, Uncle Dayananda, Aunty Upendra, Uncle Kumar, and all others who helped me in numerous ways. Especially the babysitting, meals and moral support at difficult times meant a lot to me.

I would like to extend my gratitude to our families, my parents Abey and Ranjani, sister Ruwandhi, wife's parents, Nihal and Priyani, sister Malsha and her husband Kasun for all their support. My father has always been a pillar of strength, a telecommunications engineer and entrepreneur who was my first inspiration towards studying telecommunication engineering. I would also like to especially thank my parents who are halfway across the world but have extended their financial and moral support. I would also like to mention my mother in law's dedication towards keeping our fridge loaded with cooked food every week, delivered straight from her kitchen in Hamilton.

Thank you to my little girls Nethuli and Mineli, although a PhD may not mean much to you, you have put a smile on my face at the end of the day with your little stories and pretty drawings. This has been a very long and tiring journey for our whole family, but I hope I have inspired you to dream big and work hard to make your dreams come true.

Last but not least, my heartfelt gratitude goes out to my loving wife Dulsha, who inspired me to venture on this PhD journey and supported me at each step, in every possible way to achieve this milestone in my life! It was nice to have someone to share the endless late nights, wake me up in the morning, help me with your artistic skills, share the load of parenting and have someone to discuss my research with. You have always been there to inspire me on this long tedious journey.

This thesis would not have been possible without each and every one of you, so thank you!

Rajith Abeywardana

1st October 2018

Table of Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

List of Figures

List of Tables

Nomenclature

AC	-	Access Category
AIFS	-	Arbitration Inter-Frame Space
AIFSN	-	Arbitration Interframe Space Number
AP	-	Access Point
API	-	Application Programming Interface
ARIB	-	The Association of Radio Industries and Businesses
BER	-	Bit Error Rate
BSS	-	Basic Service Set
CBR	-	Channel Busy Ratio
CCA	-	Clear Channel Assessment
CCC	-	Common Control Channel
CCH	-	Control Channel
CFM	-	Car-Following Model
CR	-	Cognitive Radio
CR-VANET	-	Cognitive Radio Enabled Vehicular Ad Hoc Network
CSMA/CA	-	Carrier Sense Multiple Access/Collision Avoidance
CW	-	Contention Window
CWmin	-	Minimum Contention Window
CWmax	-	Maximum Contention Window
DSA	-	Dynamic Spectrum Access
DSRC	-	Dedicated Short Range Communications
ECC	-	Electronic Communications Committee
EDCA	-	Enhanced Distributed Channel Access
EDCAF	-	Enhanced Distributed Channel Access Function
ETSI	-	European Telecommunications Standards Institute
FCC	-	Federal Communications Commission
GPL	-	General Public License
GPS	-	Global Positioning System
GUI	-	Graphical User Interface
IDM	-	Intelligent-Driver Model

IEEE	-	Institute of Electrical and Electronics Engineers
IP	-	Internet Protocol
IPv6	-	Internet Protocol Version 6
ISM	-	Industrial, Scientific and Medical
ITS	-	Intelligent Transportation Systems
LAN	-	Local Area Network
MAC	-	Medium Access Control
MiXiM	-	Mixed Simulator
NED	-	Network Description
NIST	-	National Institute of Standards and Technology
OSA	-	Opportunistic Spectrum Access
PRR	-	Packet Reception Ratio
PSID	-	Provider Service Identifier
PU	-	Primary User
QoS	-	Quality of Service
RSU	-	Roadside Unit
RWM	-	Random Waypoint Mobility
SCH	-	Service Channel
SDR	-	Software-Defined Radios
SIFS	-	Short Inter-Frame Space
SNIR	-	Signal to Noise plus Interference Ratio
SNR	-	Signal-to-Noise Ratio
SP	-	Service Provider
SPEC	-	Standard Performance Evaluation Corporation
SU	-	Service User
SUMO	-	Simulation of Urban Mobility
TraCI	-	Traffic Control Interface
TVWS	-	Television White Spaces
UTC	-	Coordinated Universal Time
V2I	-	Vehicle-to-Infrastructure
V2V	-	Vehicle-to-Vehicle
VANET	-	Vehicular Ad Hoc Network

Veins - Vehicles in Network Simulation

WAVE - Wireless Access in Vehicular Environments

WSA - WAVE Service Advertisement

WSM - WAVE Short Message

WSMP - WAVE Short Message Protocol

THE UNIVERSITY OF AUCKLAND
Te Whare Wananga o Tāmaki Makaurau
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

"Empowering infotainment applications: A multi-channel service management framework for cognitive radio enabled vehicular Ad Hoc networks," IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 2018, pp. 1-5.

Work from this publication appears in Chapters 3 and 6.

Nature of contribution by PhD candidate	Formulating ideas, system design and analysis, writing paper
Extent of contribution by PhD candidate (%)	80%

CO-AUTHORS

Name	Nature of Contribution
Kevin W. Sowerby	Advice and guidance on the work described. Editing, proof reading and finalising paper.
Stevan M. Berber	Advice and guidance on the work described. Editing, proof reading and finalising paper.

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Kevin W. Sowerby	Kevin Sowerby	24 September 2018
Stevan M. Berber	Berber	21 September 2018

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work**. Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

"SimuCRV: A Simulation Framework for Cognitive Radio Enabled Vehicular Ad Hoc Networks," IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, Australia, 2017, pp. 1-5.

Work from this publication appears in Chapters 2, 3 and 5

Nature of contribution by PhD candidate	Formulating ideas, conducting literature review, system design and analysis, writing paper
Extent of contribution by PhD candidate (%)	80%

CO-AUTHORS

Name	**Nature of Contribution**
Kevin W. Sowerby	Advice and guidance on the work described. Editing, proof reading and finalising paper.
Stevan M. Berber	Advice and guidance on the work described. Editing, proof reading and finalising paper.

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	**Signature**	**Date**
Kevin W. Sowerby	Kevin Sowerby	24 September 2018
Stevan M. Berber	Berber	21 September 2018

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work**. Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

"Spectrum Sensing in Cognitive Radio Enabled Vehicular Ad Hoc Networks: A Review," 7th IEEE Conference on Information and Automation for Sustainability, Colombo, Sri Lanka, 2014, pp. 1-6.

Work from this publication appears in Chapters 1 and 3.

Nature of contribution by PhD candidate	Formulating ideas, conducting literature review, writing paper
Extent of contribution by PhD candidate (%)	80%

CO-AUTHORS

Name	Nature of Contribution
Kevin W. Sowerby	Advice and guidance on the work described. Editing, proof reading and finalising paper.
Stevan M. Berber	Advice and guidance on the work described. Editing, proof reading and finalising paper.

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Kevin W. Sowerby		24 September 2018
Stevan M. Berber	Berber	21 September 2018

CHAPTER 1

Introduction

1.1 Vehicular Ad Hoc Networks

Vehicular communications have gained much attention and popularity among the industries and academia in recent years and constitute a hot topic for research. It is a key enabling technology for Intelligent Transportation Systems (often referred to as ITS) that endeavour to deliver safer, smarter, more efficient and environmentally sustainable transport through the use of innovative information and communication technologies [1], [2]. ITS intends to create network connectivity among vehicles and infrastructure comprised of traffic lights, variable message signs, roadside sensors, etc. to support travellers and operators in making better-informed decisions [3]. These networks that comprise vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications are called Vehicular Ad Hoc Networks (VANETs). VANETs are decentralised wireless networks that enable direct communication between vehicles on the fly without relying on infrastructures, such as access points or base stations. Figure 1.1 illustrates an example of such a vehicular communication network.

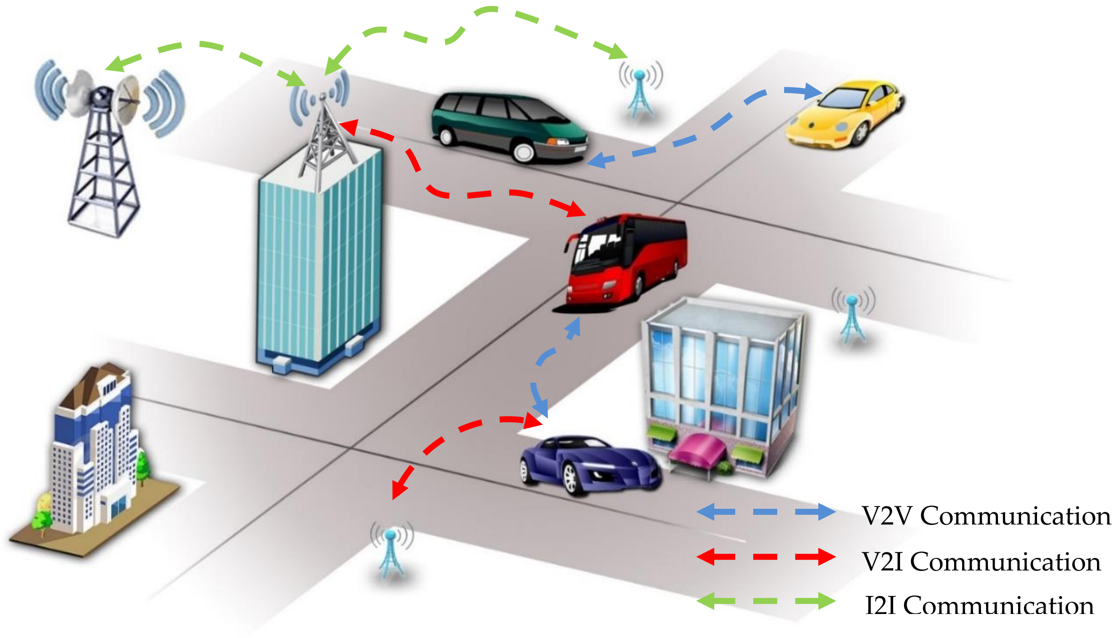


Figure 1.1: VANET application model

1.2 Applications of VANETs

VANETs have attracted significant investments from industries and research institutions in countries like United States, Japan, Singapore and South Korea, enabling promising applications of VANETs concerning road safety, efficiency and convenience [3]. Motor vehicle accidents are among the top 10 leading causes of mortality, worldwide [4]. The number of deaths in 2012 due to motor vehicle accidents in New Zealand is estimated at 7.4 per 100,000 population [5]. Analyses by the U.S. Department of Transportation's National Highway Traffic Safety Administration shows that approximately 80 percent of the crash scenarios, including rear-end crashes, lane departures, lane change or merge crashes, curve speed or excessive speed crashes, and stop sign violations involving non-impaired drivers could potentially be addressed with V2V and V2I communication systems [6].

VANETs enable information exchange between vehicles to increase situational awareness of drivers and assist drivers in avoiding crashes. For example, drivers may be able to take an evasive action in response to a system alert of imminent crash situations communicated through V2V systems, such as when a vehicle that is two or more cars ahead in the same lane brakes suddenly or when a vehicle is positioned in the driver's blind side during a lane change. Communication with infrastructure allows detection of roadway conditions and hazards, such as potholes, road works or weather-related conditions, assists drivers in keeping within a posted speed limit or informs when an upcoming traffic light is about to change [3], [7].

Traffic congestion wastes billions of hours of valuable time and produces tons of carbon dioxide in the atmosphere. According to the Texas Transportation Institute's urban mobility report, traffic congestion has caused urban Americans to travel extra 5.5 billion hours, accounting for a congestion cost of $121 billion in 2011 [8]. In the case of Auckland, the costs of congestion are approximately $1.25 billion per annum at 2010 prices [9]. Therefore, provision of real-time information to motorists such as delays due to accidents, inclement weather or roadworks, assists them to avoid congested routes and take alternative routes. These choices account for increased mobility, fuel efficiency and environmental benefits such as reduced air and noise pollution on highways [10].

The idea of letting vehicles exchange information with each other has given rise to a vast number of novel applications that provide value-added services to travellers [11], [12]. These applications can provide road users with information, advertisements, and entertainment to make their journeys convenient and enjoyable. For example, drivers would be interested in knowing vehicle parking spaces or nearby stations having cheaper fuel. On the other hand, business entities such as restaurants and local stores may wish to advertise their services to commuters. Passengers of nearby vehicles might be keen on playing video games or exchanging videos among them.

1.3 Worldwide Spectrum Allocations for VANETs

The applications discussed in the previous section have various communication requirements. The safety-critical applications, in general, have stringent delay requirements (latencies of less than 100 milliseconds) [13], high reliability and low data volume per message. Although the safety-critical message transmissions rely on little data volume per message, during peak periods of vehicular traffic with high densities of vehicles within the communication range of each other, this may translate to rather high data rates. The situation is further aggravated due to periodic rebroadcasts of safety messages, typically at a frequency of 10Hz [13], requiring the network to handle thousands of messages per second at significant data rates.

On the other hand, the demand for vehicular communication applications continues to grow in the direction of convenience and entertainment. However, these applications may require the transmission of larger message volumes and more throughput than conventional safety applications. Following the high expectations of vehicular communications to improve traveller safety, reduce traffic congestions and promote convenience and entertainment applications, the regulatory bodies in the USA, Europe, and Japan have successfully allocated spectrum for vehicular communications.

In 1999, the Federal Communications Commission (FCC), an independent agency of the government that regulates radio communications in the USA, has allocated 75MHz of spectrum at 5.850-5.925 GHz having seven 10MHz channels to support Dedicated Short-Range Communications (DSRC) for ITS services requiring high-speed wireless communications over short distances (generally less than 1000 meters) [14]. The frequency band, 5.850 GHz to 5.855 GHz, was reserved for future ITS applications. In the U.S., the equipment deployed in the DSRC frequency band is currently expected to comply with the IEEE (Institute of Electrical and Electronics Engineers) standard called Wireless Access in Vehicular Environments (WAVE) [15]. The WAVE standard is discussed in detail in Chapter 3.

In the European Union, spectrum allocations in its member states and associated countries are governed by the European Commission [16]. Following the FCC rulings, in 2008, the European Commission in its decision 2008/671/EC mandated the use of the frequency band 5.875 GHz to 5.905 GHz for safety-related ITS applications throughout the member states of the European Union [17]. Furthermore, the Electronic Communications Committee (ECC) decision (08)01 considers the designation of the frequency band 5.905 GHz to 5.925 GHz for an extension of ITS spectrum [18], while the ECC recommendation (08)01 addresses the usage of the band 5.855 GHz to 5.875 GHz for non-safety applications of ITS [19]. The ECC is responsible for developing regulatory policies in electronic communications for Europe and harmonising the efficient use of the radio spectrum [20].

In Japan, 80 MHz of spectrum at 5.770-5.850 GHz has been allocated for DSRC under ARIB (The Association of Radio Industries and Businesses) standard [21]. Figure 1.2 illustrates the DSRC frequency band specifications in Europe, USA and Japan.

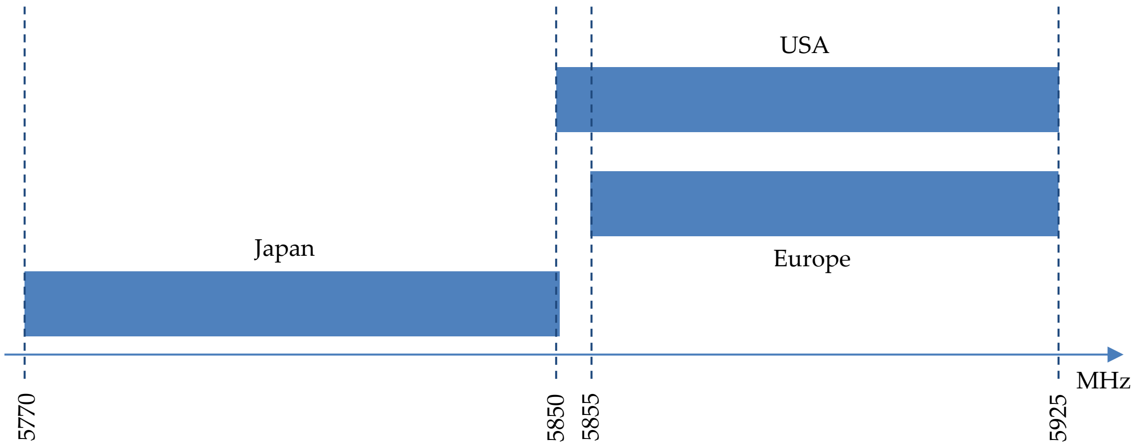


Figure 1.2: DSRC frequency band specifications in Europe, USA and Japan

1.4 Spectrum Scarcity in VANETs

As discussed in the previous section, 70-75MHZ of bandwidth has been allocated for DSRC communications in the USA and Europe. Although non-safety applications are permitted within this frequency band, the primary purpose of DSRC is to enable safety related vehicular communications. However, as per recent studies carried out by researchers [22]-[25], and also based on simulations carried out during this research, these dedicated spectrum resources are barely adequate to sustain the throughput demands of vehicular safety applications alone, especially during peak periods of vehicular traffic. In these conditions, the activities of non-safety applications that require high data throughput may become severely restricted. Even though the advances in technology create the potential for radio systems to use the spectrum more efficiently than in the past, it is challenging to keep up with ever-increasing throughput requirements of data-intensive vehicular communication applications merely by efficient usage of the allocated spectrum. Hence, more spectrum resources are required to cater to the high throughput demands of these applications (both safety and non-safety), while providing the required Quality of Service (QoS).

1.5 Opportunistic Spectrum Access

The radio spectrum is one of the most tightly regulated resources of all-time. The spectrum allocation policies usually divide the radio spectrum into well-defined blocks and statically grant exclusive rights on a long-term basis to licensees or services over large geographical areas. Most frequency blocks are generally auctioned by regulatory authorities, such as the FCC in the USA, and sold to the highest bidders to maximise government revenues, while some portions of the radio spectrum are freely available for low power operations, such as the 2.4GHz and 5GHz Industrial, Scientific and Medical (ISM) bands. This fixed spectrum allocation policy causes unlicensed radio bands to be overutilized, and on the other hand, licensed frequency blocks to be intermittently utilised. For example, as per spectrum occupancy measurements [26], [27], even premium frequencies below 3GHz in densely populated urban areas have appeared to be quiet most of the time.

These findings stressed the need to introduce a new spectrum management paradigm that creates the potential for radio systems to use the spectrum more intensively. This paradigm, known as opportunistic spectrum access (OSA) or dynamic spectrum access (DSA), allows unlicensed radios to temporarily access specific underutilised frequency bands when licensed wireless services do not occupy such bands.

1.6 Cognitive Radio Technology

The cognitive radio (CR) technology goes hand in hand with the opportunistic spectrum access paradigm. It is a technology that enables radios to achieve self-configuration capabilities required for OSA. The CRs are commonly realised through software in software-defined radios (SDRs). Traditionally, the radios were implemented with hardware that executed specific radio functions and had no or limited software programmable capabilities. With the advancements of microprocessors and programmable electronic devices, many of those radio functions are accomplished by software modules running on microprocessors. The operational characteristics, such as the frequency range, modulation type or maximum output power, of such a software-defined radio may be changed at will, by only modifying or changing the software. However, an SDR is not capable of autonomously adopting new characteristics based on a reasoned assessment of the environment in which it operates. The CRs are the next generation of SDRs wherein the transmitter parameters of the radios are reconfigured into the most effective form based on changing requirements and conditions of the operating environment and through a process of cognition [28].

The term cognitive radio was initially coined by Joseph Mitola III, where he envisioned the CR as a software-defined radio with model-based reasoning and learning capabilities and introduced a cognitive cycle through which it can decide and adapt itself autonomously based on its perceived information [29]-[31]. It follows that being aware of the radio environment is an essential factor of CR technology. Information on the radio environment may be collected through passive spectrum sensing or accessing other data sources, such as databases containing information about available frequencies. Spectrum sensing is necessary to detect spectrum holes by measuring the RF energy levels and characteristics of received signals.

The CR operations are permitted in some countries in specific frequency bands on a secondary basis at locations where that spectrum is open. For example, in the US, the FCC has allowed the operation of unlicensed CR devices in the VHF/UHF broadcast television spectrum between 54–862 MHz, often referred to as the television white spaces (TVWS) [32], [33].

1.7 Cognitive Radio Enabled VANETs

In recent years, CR technology has been considered as a potential solution to increase the available bandwidth for vehicular communications, and thereby improve the achievable throughput. The communication requirements of safety-critical applications are mainly being taken care of by the DSRC spectrum. Hence, the objective of using CRs in VANETs is assisting high throughput requirements of non-safety applications such as infotainment applications.

Since the frequencies of dynamic spectrum access bands such as the television white spaces (54–862 MHz) are much lower than the frequency bands allocated for vehicular communication in both USA (5.850-5.925GHz) and Europe (5.855-5.925GHz), communications in the TVWS profit from much better propagation conditions, especially in urban environments, where buildings are the main factor limiting the range of vehicular communications.

Application of CR technology in vehicular environments is comparatively a new research area and numerous problems still require investigation. This research intends to address several partially investigated issues that are important for the effective deployment of CR enabled VANETs (CR-VANETs). These research issues associated with CR-VANETs are described in detail, in Chapter 2. In the following section, an overview of these issues is presented.

1.8 Research Issues in CR Enabled VANETs

Performance evaluation is an important aspect of vehicular communication research. While data obtained through field experiments provide immediate feedback of the vehicular communication system being utilised, deploying real testbeds usually involves high costs and presents substantial implementation complexities since a large number of testbeds are needed to relate the experimental results to a full-scale vehicular communication system. Due to these reasons, studies on inter-vehicular communication systems are generally carried out through simulations. Simulations allow researchers to study and evaluate every part of the communication system without the hassles and expenses of real testbeds. In the situation of CR enabled VANETs, this requires an integrated simulator that provides modelling of both CR and VANET environments. Nevertheless, the evaluation of CR-VANETs constitutes a critical issue due to the lack of such simulators.

Spectrum sensing is an essential function of the cognitive cycle. It is performed to identify the transmitting activities of licensed radio users in dynamic spectrum access networks and to be aware of the spectrum usage by other cognitive radio users. The two types of transmissions (licensed and secondary user) must be distinguished by CR users, as the radio channel in which a CR operates must only be vacated upon detecting a licensed user. This area of study has not been thoroughly investigated in CR-VANETs, where there are no base stations to control the transmitting activities of CRs (vehicles).

In CR networks, a channel is required to exchange control information between neighbouring CR nodes. Such a channel may be employed to discover services offered by other CR devices, establish and maintain peer-to-peer connections, exchange spectrum sensing information with one another or switch between various dynamic spectrum access bands. Hence, establishing a reliable channel for control signal transmission is essential for the proper operation of a CR network. While this functionality is well addressed in CR networks with stationary nodes, limited attention has been received on this problem in CR enabled VANETs and remains an important research issue.

1.9 Research Objectives and Scope

This research aims to utilise cognitive radio techniques in vehicular ad hoc networks effectively. The following novel approaches are developed in this thesis, conforming to the IEEE WAVE standards, to overcome the limitations described in the previous section.

- A simulation framework for application service management in VANETs
- A framework for simulating CR enabled VANETs
- A sensing technique to distinctively identify licensed users and secondary network users
- An algorithm to continuously evaluate the channel congestion in the DSRC spectrum for effective utilisation of these channels in the transmission of CR related control signals
- Extending the CR-VANET simulation framework to model the above multi-channel control signal transmission algorithm

The main reason for adhering to these standards is for future users of the proposed work, to be able to have a common ground for the ease of hardware implementation. When there is little or no deviation from the standards, there is a much higher probability of this work being put to good use in the future.

The scope of the simulation framework for application service management includes:

- Modelling multi-radio multi-channel communication
- Modelling application service advertisement and discovery
- Modelling data transfers over distinct radios channels
- Modelling simultaneous application service operation
- Modelling channel access mechanisms
- Modelling radio signal propagation
- Modelling vehicular mobility

The simulation framework for CR-VANETs has the following functions:

- Modelling mobile (vehicles) and stationary CR nodes

- Modelling licensed user transmissions
- Modelling CR based application service advertisement and discovery
- Modelling spectrum sensing
- Distinctively identifying licensed and secondary user transmissions
- Modelling spectrum hand-off

The multi-channel CR related control signal transmission algorithm includes:

- Continuously evaluating the congestion of DSRC channels
- Selecting a suitable channel to transmit control signals
- Spectrum hand-off

1.10 Thesis Contributions

As a whole, this thesis contributes to a complete simulation framework which facilitates modelling and analysis of cognitive radio enabled vehicular ad hoc networks. The unique feature is that it includes a mechanism for application service management in dynamic spectrum access allowed bands. The novel spectrum sensing mechanism developed uses a combination of carrier sensing and energy detection for effectively distinguishing between primary and secondary users. A multi-channel signalling framework for cognitive radio enabled VANETs was also designed and implemented.

The main contributions of this thesis are:

- A simulation framework for application service management in VANETs which simulates the two WAVE device roles, the service provider and service user roles, application service advertisement and discovery, simultaneous application service operation and routing application data over distinct radio channels. It also simulates the two channel access methods, alternating and continuous channel access modes, and changing operating radio channels while accurately handling packet reception and transmission at the time of channel switching.
- A framework for simulating CR enabled VANETs, which models primary user transmissions, reception of primary user signals (by the secondary users),

secondary user data transmissions on dynamic spectrum access enabled bands (such as television white spaces), advertisement and discovery of secondary user application services on radio bands allocated for VANETs (such as the DSRC band), spectrum sensing and radio channel hand-off.

- An algorithm to continuously evaluate the channel congestion in the DSRC spectrum and employ any channel in the DSRC band for transmitting CR related control signals.

The simulation framework for cognitive radio enabled vehicular ad hoc networks was developed extending the '*Veins*' vehicular network simulation framework. Table 1.1 compares the functionalities of the developed simulation framework and the *Veins* framework.

Table 1.1: Comparison of the functionalities of the developed and 'Veins' simulation frameworks

Functions	**Developed framework**	**Veins framework**
Modelling wireless connectivity	✓	✓
Channel models	✓	✓
Signal transmission and reception	✓	✓
Radio signal propagation	✓	✓
Modelling interference and noise	✓	✓
Quality of service enabled channel access	✓	✓
Vehicular Mobility	✓	✓
WAVE short message generation	✓	✓
Alternating channel access between the control channel and a service channel	✓	✓
Continuous control channel access	✓	✓
Alternating channel access between any two channels	✓	✗
Continuous service channel access	✓	✗
Multi-radio multi-channel communications	✓	✗
Application service provider and user roles	✓	✗
Application service advertisement and discovery	✓	✗
Simultaneous application service operation	✓	✗
Simultaneous data transfers over distinct radio channels	✓	✗
Licensed user transmissions	✓	✗
CR based application service advertisement and discovery	✓	✗
Spectrum sensing	✓	✗
Identify licensed and unlicensed user transmissions distinctively	✓	✗
Spectrum hand-off	✓	✗

1.11 Publications

The work described in this thesis has been published as follows:

1. (2018) Abeywardana, R. C., Sowerby, K. W., and Berber, S. M.; "Empowering infotainment applications: A multi-channel service management framework for cognitive radio enabled vehicular ad hoc networks," Proceedings of IEEE 87th Vehicular Technology Conference 2018 (VTC 2018), 3rd – 6th June 2018, Porto, Portugal.

2. (2017) Abeywardana, R. C., Sowerby, K. W., and Berber, S. M.; "SimuCRV: A simulation framework for cognitive radio enabled vehicular ad hoc networks," Proceedings of IEEE 85th Vehicular Technology Conference 2017 (VTC 2017), 4th – 7th June 2017, Sydney, Australia.

3. (2014) Abeywardana, R. C., Sowerby, K. W., and Berber, S. M.; "Spectrum sensing in cognitive radio enabled vehicular ad hoc networks: A review," Proceedings of 7th IEEE Conference on Information and Automation for Sustainability, (ICIAfS 2014), 22nd – 24th December 2014, Colombo, Sri Lanka.

1.12 Thesis Organisation

The remaining chapters are organised as follows. Chapter 2 reviews existing literature in cognitive radio cycle, CR standards, spectrum sensing techniques, spectrum sensing in CR enabled VANETs, rendezvous in CR networks, simulating CR networks and VANETs. Limitations in applying CR techniques to VANETs are then highlighted. The limited availability of simulation frameworks for such networks is also discussed.

Chapter 3 presents the underlying standards of VANETs to lay the foundation on which the proposed work is carried out. The primary standard discussed here is the

IEEE WAVE standard. The network simulator and the road traffic simulator chosen for implementation are also introduced in detail. The chapter highlights the need for simulations and the types of simulation models that will be used in this system.

Chapter 4 discusses the implementation details of the simulation framework for vehicular application service management. The work carried out by the different layers of this framework is presented in detail. Simulation results obtained from this framework are also presented and discussed.

Chapter 5 presents the implementation details of the simulation framework for cognitive radio enabled vehicular ad hoc networks. Work carried out to differentiate licensed and unlicensed users is elaborated upon in this chapter. The spectrum handoff process implemented in this simulation module is also discussed. The performance of the spectrum sensing algorithm is also evaluated.

Chapter 6 discusses the details of the algorithm developed to effectively employ all available channels in the DSRC spectrum in CR related control signal transmission. Details of implementing this algorithm on the developed simulation framework are discussed in this chapter. The performance of this algorithm is then evaluated.

Chapter 7 draws conclusions and makes recommendations for future work to improve the proposed CR enabled VANET frameworks.

Figure 1.3 shows the relationship between the thesis chapters and their contents.

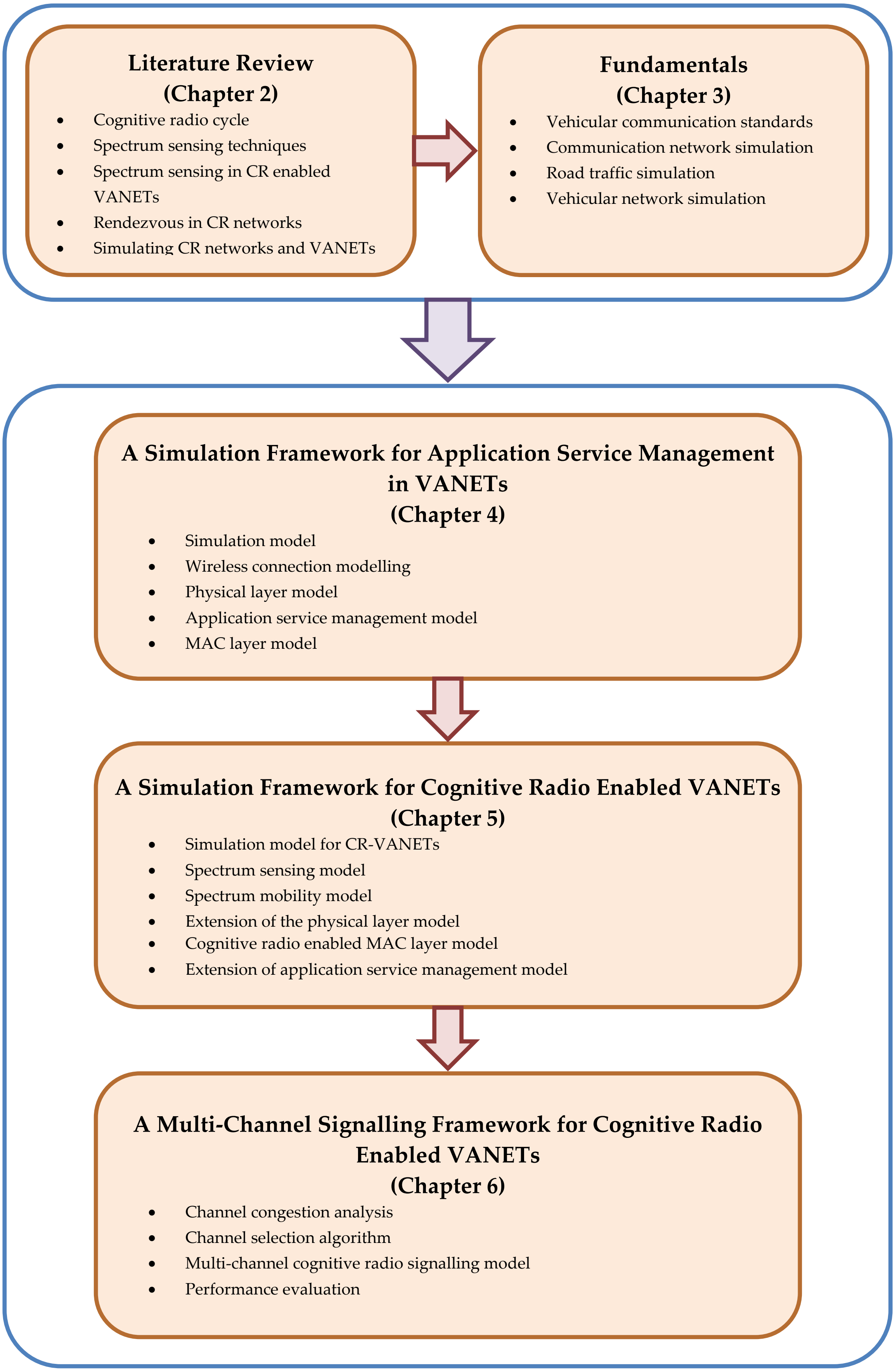


Figure 1.3: The content flow of chapters

CHAPTER 2

Literature Review

2.1 Cognitive Radio Technology

The cognitive radio (CR) technology empowers wireless terminals to assess and access frequency bands dynamically. It brings intelligence to radios to learn from the environment and adapt accordingly. While there are many definitions presented in the literature and by telecommunication regulatory bodies around the world for cognitive radios, currently no formal definition has been adopted. Some of the known definitions of CRs are summarised below to give a feel for the capabilities and functions that are expected of CRs.

- Joseph Mitola [30]: "A radio that employs model-based reasoning to achieve a specified level of competence in radio-related domains."
- Simon Haykin [34]: "An intelligent wireless communication system that is aware of its surrounding environment and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters."
- Federal Communications Commission (FCC) [28]: A radio that changes its transmitter parameters based on interaction with the environment in which it operates, while adhering to FCC rules.
- National Telecommunications and Information Administration of the U.S.A. [35]: “A radio or system that senses its operational electromagnetic environment and can dynamically and autonomously adjust its radio operating parameters to modify system operation.”
- IEEE (Dynamic Spectrum Access Networks Standards Committee) [36]: “A radio system employing technology that allows the system to obtain knowledge of its operational and geographical environment, established policies and its internal state to dynamically and autonomously adjust its operational

> parameters and protocols according to its obtained knowledge in order to achieve predefined objectives and to learn from the results obtained.”

From these definitions, it can be inferred that a CR is a system that possesses cognition capability and reconfigurability through which it distinguishes itself from traditional radios. The cognition capability enables a CR to sense the dynamically changing surrounding radio environment, analyse gathered information and decide on the best course of action about the spectrum band to be employed and the transmission strategy to be adopted. In cognitive radio terminology, the primary users (PUs) are the licensed operators of a specific shared spectrum segment that have higher priority on the usage of the band, whereas the secondary users or the CR users, as the name suggests, have lower priority and may exploit this spectrum without causing interference to primary users. Hence, the secondary users should be able to sense the radio spectrum that is unused by the primary network, quickly seize opportunities to transmit and vacate when a primary user reoccupies the spectrum.

The way in which a cognitive radio conceptually interacts with the radio environment, commonly referred to as the cognitive cycle, is illustrated in Figure 2.1. The cognition cycle is executed continuously by CRs to detect spectrum opportunities, analyse, decide and act to explore the best opportunities [37].

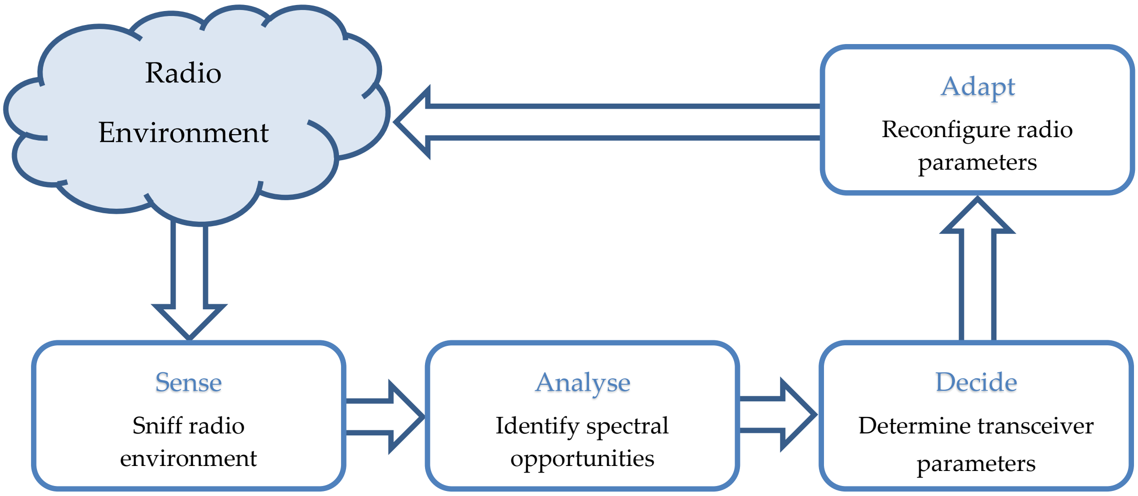


Figure 2.1: The cognitive cycle

Spectrum sensing is one of the most critical components of the cognitive cycle and refers to the process of sniffing the radio environment using an antenna to capture electromagnetic activities that take place over various spectrum bands due to radio transmissions. CRs make real-time judgements on when and how long to sense radio channels based on spectrum regulations and collect adequate spectrum-related information to speculate about the radio environment.

The decision-making process involved in drawing conclusions about the spectrum is referred to as spectrum analysis. In most literature, the two processes, spectrum sensing and analysis, are commonly mentioned as "spectrum sensing", where it is assumed that analysis is part of the process. Data collected by sniffing the spectrum is analysed to learn the surrounding radio environment, for example, to identify the vacant radio channels in dynamic spectrum access networks and be aware of the spectrum usage by other secondary users [38].

Once a CR gathers information about spectral opportunities, the next step is to decide the set of transmission actions to be taken while conforming to local rules and regulations. These include selecting a suitable radio channel, transmission power, data rates, modulation etc. for future transmissions. When an appropriate radio channel is chosen and transmission parameters are configured, CR communication can take place on this band of spectrum.

The operations over various spectrum bands are facilitated by CR's ability to reconfigure its transceiver parameters on the fly. A CR continuously senses and assesses its radio environment, and if it detects a primary user or experiences a deterioration of channel conditions in its currently used channel, it should switch to a different radio channel. The process by which a CR device alters its frequency of operation is referred to as spectrum mobility. Hence, it must be associated with a spectrum handoff mechanism that allows a seamless transition between the current and new frequency, without making disruptions to any ongoing communications between CR devices.

2.1.1 Spectrum Sensing Techniques

As mentioned in the previous section, spectrum sensing is essential to learn the radio environment. There are various spectrum sensing techniques (i.e. sensing and analysis) available in the literature for identifying the presence of the primary user signal transmissions, such as energy detector based sensing, cyclostationarity based sensing, matched filter based sensing, etc. [39]. The energy detection based spectrum sensing is the most common method of spectrum sensing due to its simplicity and low computational and implementation complexities [39], [40]. This method of sensing, which was proposed by Harry Urkowitz [41], does not require prior knowledge of the signal being transmitted. Under energy detection, the power of the received signal is compared with a threshold, and a binary hypothesis test is performed to identify the presence or absence of transmitted signals in a channel. Some of the challenges with energy detector based sensing are choosing a threshold for detecting transmitted signals and poor performance under low signal-to-noise ratio (SNR) values [42].

When some characteristics of the primary user signal are known, for example, the modulation type, symbol rates, etc., the cyclostationarity based sensing technique may be used to identify the transmitted signal. This technique does not require a threshold to distinguish between a primary signal and noise. However, this advantage over the energy detection method comes at a price, as a higher sampling rate is needed to get a sufficient number of samples, which leads to increased computational complexity [38].

When the primary user signal is fully known (such as the modulation format, carrier frequency, data rate, pulse shape, etc.), a CR may use the matched filter. A matched filter needs to perform the entire receiver operations to detect a signal. Furthermore, it requires perfect timing and synchronisation, resulting in high computational complexity. Hence, these factors may prevent the use of a matched filter for opportunistic spectrum access [42].

In highly mobile networks such as vehicular networks, spectrum sensing must be fast in order to track primary user signals. The energy detection based sensing is a viable solution for identifying primary user transmissions in CR-VANETs due to fast signal detection, simple design and low computational and implementation complexities. Although the performance of energy detection deteriorates under low SNR values, this does not apply to vehicular environments. Low SNR values are usually observed when a vehicle is at the border of a primary user's signal coverage area. As vehicles are mobile, they may either move out of a primary user's interference range or move into its coverage area (high SNR values may be observed under these conditions) and detect primary user transmissions.

2.1.2 Spectrum Sensing in Cognitive Radio Enabled VANETs

The VANETs are decentralised networks that allow direct communication between vehicles without the aid of base stations. Vehicles (WAVE devices) share the available wireless channels and compete for channel access with a carrier sense multiple access/collision avoidance (CSMA/CA) scheme. When vehicles operate in radio channels where dynamic spectrum access is allowed (such as TVWS), they need to detect the presence of both primary and secondary users. Since base stations are not available to manage secondary user (vehicle) transmissions in those channels, any vehicle can access them when they sense a channel is idle. However, when vehicles detect a signal transmission on a channel, it is essential to determine whether a PU or a secondary user is using the channel, as the channel should be vacated if a PU uses it.

Energy detection has been the preferred method of spectrum sensing in CR-VANETs by authors of recent research due to its advantages discussed in the previous section [43]-[49]. One of the drawbacks of energy detection based sensing is that its inability to distinguish between a secondary user transmission and a primary user transmission, as this technique only measures the energy associated with a received signal. Hence, to differentiate between a PU and a secondary user transmission, a periodic channel silence phase is required, shutting down all secondary user transmissions, in order to sense possible PU transmissions [47]-[49]. However, a periodic channel silence phase degrades the performance of secondary user networks. Furthermore, it is impractical to implement strict sensing schedules in VANETs, as there are no centralised base stations to coordinate them. Another drawback is that when multiple secondary user networks operate in tandem (for example, several IEEE802.11 networks), it is impossible to force coordinated spectrum sensing phases between different networks, even if base stations are controlling each network.

Few studies have been conducted to perform energy detection based spectrum sensing to distinguish PUs and secondary users without having scheduled sensing periods. Authors of [46] have considered utilising smart antennas to track the direction of arrival of received signals with prior knowledge of PU location and identify a PU transmission. However, there was no mention of how this method could be realised. In [50]-[52], CR networks with IEEE802.11 based stationary secondary users have been studied. The IEEE802.11 devices compete for channel access with CSMA/CA mechanism, similar to the WAVE devices. Out of these, the studies [51], [52] measure IEEE802.11 physical layer error counts and compare against a threshold value to detect primary users. Threshold values are experimentally selected and depends on the wireless environment. Hence, this method is not suitable in VANETs where wireless environment varies dynamically. In [50], the secondary users determine that a PU has occupied a channel if the channel is found to be busy for a period longer than the transmission duration of data and its acknowledgement. This work assumes that there will be no collision among secondary user data transmissions, and collisions only occur between secondary users and a PU when a PU occupy the spectrum. Nevertheless, this assumption does not apply to CR-VANETs where packet collisions between vehicles are inevitable. Under these conditions, secondary users may falsely assume that a channel is occupied by a PU as collisions will keep a channel busy for a period longer than the transmission duration of a data packet and its acknowledgement.

Although few studies have been carried out to distinctively identify PU and secondary user signal transmissions using energy detection in CR-VANETs with no forced quiet sensing periods, this area of study has not been thoroughly investigated.

2.1.3 Cognitive Radio Standards

The CR based operations are permitted in the VHF/UHF television spectrum between 54–862 MHz, referred to as the television white spaces (TVWS). Several standardisation activities are governing the operation of CRs in the TVWS such as IEEE802.22-2011 [33] and IEEE802.11af-2013 [53]. IEEE 802.22 is the first standard that allowed cognitive radio operations in the TVWS. The Wireless Regional Area Networks (WRANs) for which this standard has been developed focuses on constructing point-to-multipoint networks comprised of a base station and user terminals denoted as consumer premise equipment (CPE). The Medium Access Control (MAC) layer regulates downstream medium access by Time Division Multiplexing (TDM), while the upstream is managed collectively by a Demand Assigned Multiple Access (DAMA) and an Orthogonal Frequency Division Multiple Access (OFDMA) system.

The IEEE802.11af standard is an amendment to the IEEE802.11 standard that enables IEEE802.11 devices to access TVWS with the aid of an access point opportunistically. This standard adopts the CSMA/CA based channel access scheme at the MAC layer. Both these standards are required to obtain TVWS channel availability information from a geolocation database.

2.2 Rendezvous in Cognitive Radio Enabled VANETs

In ad hoc networking environments, wherein there are no centralised servers or base stations to manage connections between nodes, the CR devices show their existence by broadcasting control messages on a radio channel common to each other known as the Common Control Channel (CCC). The neighbouring CRs make use of these broadcasts to establish and maintain peer-to-peer connections as well as to discover services offered by other CR devices [54]. The process of two or more radios meeting and establishing a communication link between them on a common channel is referred to as *rendezvous*. A CCC may also be employed by CRs to negotiate channel access in the shared spectrum or exchange spectrum sensing information with one another [47], [48], [55]. Moreover, a CCC is necessary to support spectrum mobility (discussed in Section 2.1), which enables CRs to make seamless transitions between frequencies.

Hence, establishing a reliable CCC is essential for the proper operation of a CR network, as the secondary user communications that take place on a shared licensed band is dependent on the network coordination facility provided by a CCC. While this functionality is well addressed in CR networks with stationary nodes, minuscule attention has been received on the problem of maintaining a CCC in CR enabled VANETs. Moreover, the CCC schemes designed for static CR networks may not be directly applied to vehicular networks, which are characterised by highly mobile nodes, intermittent connections and dynamic network topologies. Several CCC designs proposed in the literature are reviewed in the sections to follow.

2.2.1 Dedicated Control Channel Designs

Allocating a dedicated channel to exchange control information simplifies the network coordination among secondary users. Hence, it is often considered that a dedicated CCC is allocated in a band licensed to CR networks [56]-[60]. However, in addition to hefty licensing charges involved in this method, a network-wide CCC is prone to saturation. In CR enabled VANETs research, the channel reserved for safety and management related communications has been considered as a possible CCC by authors in recent studies [47], [48], [55]. At first glance, utilising a channel that has network-wide coverage may seem an attractive solution in establishing connections among vehicles. However, this channel may incur data congestion solely with safety-related communications [23] (hence, the reason for using the CR technology). When the same channel is accessed for CR operations, especially during high-density vehicular traffic, the channel may become even more congested with high packet collisions making it unavailable for both safety and CR communications and would present a single point of failure.

On the other hand, in several CR studies, dedicated CCCs have been allocated in the unlicensed ISM bands [61]. However, since other unlicensed radios of heterogeneous networks access these channels, CRs may be subjected to the interference from these radios. A CCC channel may perhaps be allocated in the shared licensed spectrum. Nonetheless, due to the dynamic nature of licensed user activities, employing a dedicated channel free of incumbent licensed user activities across the whole network is rather impractical.

2.2.2 Group-Based Common Control Channel Designs

The CR users in a neighbourhood generally detect similar spectrum opportunities in the shared licensed band. Hence, the interests of researchers have drawn towards selecting a channel commonly available to a group or cluster of CRs in proximity and using it as a CCC. In distributed CR networks, the process of forming groups usually involves discovering nearby CRs, creating links between them and obtaining lists of channels that each CR has identified as vacant. These operations may be performed by each CR user or a group/cluster manager in a neighbourhood. In the former case, every CR device is required to broadcast beacons on all available channels to indicate its availability, as well as to scan channels to receive broadcasts from other CR devices [62]-[64]. After initial discovery, each CR user identifies its neighbours and their available channels and based on this information a channel that provides the largest connectivity is selected. The group based CCC design also has several drawbacks. Firstly, the process of discovering neighbouring CRs and allocating a CCC takes a long time to complete and involves a considerable amount of broadcast overheads. Secondly, as the number of CR users in a group increases, the control channel may become congested similar to the dedicated control channel designs.

In the case of cluster manager based CCC designs, the CRs first scan through all the channels to identify beacons of nearby CRs before commencing their broadcast. If no such broadcast is detected, a CR node may initiate the beacon broadcast to form a cluster and become the controller or the cluster manager [65], [66]. This solution has been adopted to select a CCC in CR enabled VANETs by the authors in [55]. However, due to the motion of vehicles, the network topologies, as well as the vacant channels observed by vehicles, may frequently change especially in urban areas, and forming clusters and establishing CCCs may not be possible. However, this technique may potentially be applied to groups of vehicles travelling together (convoys) in suburban areas or motorways, as illustrated in Figure 2.2.

Figure 2.2: A vehicle forming a cluster to select a common control channel

2.2.3 Sequence-Based Common Control Channel Designs

Both dedicated and group-based CCC designs have the drawback of control channel saturation when a large number of CR nodes are occupying the same channel. Therefore, to diversify the control channel allocation over the shared spectrum, the sequence based CCC selection schemes have made use of random or predetermined channel hopping sequences, wherein the CR pairs visit the available channels in an orderly manner at defined time slots and achieve rendezvous when they hop to the same channel. This technique has enabled CR pairs to communicate on different control channels, due to the variety of hopping sequences being utilised. There are several ways of constructing sequences, namely, permutation-based [67], pseudo-random based [68], adaptive pseudo-random based [69] and quorum based [70].

However, since the sequence-based CCC designs establish pairwise connectivity between neighbouring CR nodes, setting up links with multiple CR nodes may incur high overheads and delays. Furthermore, CRs need to have prior knowledge of neighbouring CRs that they intend to form connections with, as no neighbour discovery process or control signal broadcast takes place. Although the sequence-based CCC designs are suitable for CR networks with fixed nodes, they are not appropriate for vehicular networks in which the neighbouring nodes change dynamically, and communications with multiple vehicles take place.

2.3 Simulating Cognitive Radio Enabled VANETs

In communication networks, as large as vehicular networks, studies on new technologies are usually carried out through simulations, due to high costs and implementation complexities involved with deploying real testbeds. Nonetheless, these simulations that model the operation of a real-world communication system must be able to accurately represent the characteristics and functions of the physical system of interest to a degree of fidelity [71]. In the context of CR enabled VANETs, this requires an integrated simulator that provides modelling of both CR and VANET environments. Such a simulator must be able to model radio signal propagation, CR functions, vehicular mobility and vehicular communication protocols (for example, IEEE WAVE). Nevertheless, integrated CR-VANET simulators that characterise all these functionalities have not been made available to the research community. Conversely, a limited number of simulators are available for modelling VANETs as well as stationary CR networks when these networks are considered individually.

2.3.1 VANET Simulators

Network simulators are usually employed in communication network modelling to understand the interaction between different network entities, to analyse the behaviour of protocols at different levels of the communication stack and to model the traffic that nodes carry. Most network simulators model systems wherein the state variables change at discrete points in time and the models of interest are analysed numerically [71]. These types of simulators are referred to as discrete event simulators (discussed in detail in Chapter 3). Several well-known discrete event simulation platforms are commonly employed to build simulation frameworks or models for communication networks, more popular being the open source tools such as OMNeT++ and ns-3, and the commercial simulators, for instance, the OPNET Modeler (Riverbed Modeler) and QualNet.

While all these simulators are quite similar to each other, the main difference remains in the number of available models. In the situation of VANETs, the commercial network simulators (OPNET and QualNet) possess only the standard communication models that may be extensible, but not the simulation models explicitly developed for VANETs [72], [73]. OMNeT++ and ns-3, on the other hand, are composed of communication protocols that add specialised functionalities of VANETs. ns-3 is a free software publicly available under the GNU General Public License (GPL) version 2 license for research, development, and use. Although ns-3 includes an intrinsic vehicular communication model (WAVE), it currently has the limitation of not involving any vehicular mobility [74]. Furthermore, ns-3 does not contain a built-in graphical user interface (GUI) to visualise the network being simulated.

OMNeT++ is also a simulation platform distributed free of charge for any non-commercial use, such as for teaching or research purposes at academic institutions or non-profit research organisations [75]. OMNeT++ has gained popularity as a network simulation platform among both research and industrial communities due to its extensible and modular component architecture for simulation models, and GUI support. OMNeT++ was also selected as part of the Standard Performance Evaluation Corporation (SPEC) CPU 2006 benchmark suite (SPEC is a non-profit corporation that establishes, maintains and endorses standardised benchmarks). More importantly, OMNeT++ offers a vehicular network simulation framework through an independently developed project called '*Veins*' that integrates both vehicular communications and vehicular mobility. For these reasons, OMNeT++ was selected as the platform to be used in VANET simulations in this research, and the *Veins* simulation framework was employed as a base model. More details on the *Veins* framework are presented in Chapter 3.

2.3.2 Cognitive Radio Network Simulators

Research on CR networks has been in existence over the last one and a half decades. Nevertheless, the availability of general purpose CR simulators (with or without support for mobility) to model CR networks is still limited, and studies have been mostly carried out using simulators that only characterise the physical layer communications, for example, MATLAB. The functionalities of the upper layers are not commonly modelled in those CR simulations.

The *crSimulator* is a simulation model for CR networks in OMNeT++ [76]. It includes physical and medium access control (MAC) layers of CR nodes, as well as a spectrum sensing module to detect the activities of licensed operators in a radio channel.

However, this work does not present a mechanism to differentiate the licensed users from the users operating on these channels on a secondary basis. Furthermore, *crSimulator* framework does not support node mobility and makes static connections between nodes. In vehicular networks where distances between nodes change as they move, making dynamic connections is essential. Hence, this simulation framework is not suitable for modelling dynamic spectrum access in vehicular ad hoc networks.

A CR framework for OMNeT++ that supports CR nodes with multiple radios was introduced in [77]. A radio operating on a fixed frequency was intended to provide access to a common control channel, while the other radios having the capability to switch between channels were utilised to make use of the spectrum sharing enabled channels. The radio implementations are based on IEEE802.11, wherein access to channels are provided through CSMA/CA. The scanner module carries out spectrum sensing, and the CR engine module manages scanning requests and channel selections. Nevertheless, the way the spectrum sensing is performed (for example, through energy detection) has not been mentioned in this work. Similar to the *crSimulator* framework, supporting node mobility and distinguishing licensed operators from secondary users are not part of this framework. An ongoing effort towards the development of a CR simulation framework for OMNeT++ was presented in [78].

A simulation framework based on ns-2 for CR networks was proposed in [79]. In this research, the CR users first enter a mandatory sensing phase pausing all transmission activities of secondary users to sense any PU activities and select a free channel to enter an operating phase, wherein the CR users contend for channel access through CSMA/CA. At the end of the contention period, the operating channel is sensed again, and the cycle repeats. A drawback of this design is that the periodic sensing phase, which is required to differentiate between a PU and a secondary user transmission, degrades the performance of secondary user networks. Moreover, as all the channels available for dynamic spectrum access are sensed during the sensing phase, a channel is sensed for a limited amount of time when a large number of channels are available. In [80], authors implement a CR simulation model for ns-3 that estimates PU activity using a static database. The PU database, which is a text file loaded at the start of the simulation, specifies the number of PUs, the occupied channel, transmit power, and a list of PU activity times. Nevertheless, the simulation framework does not provide a spectrum sensing facility which is required to track real-time PU activities not included or updated in the database.

The CR simulation models mentioned above lack other aspects of CR networks such as node mobility, distinctively identifying primary and secondary users (without employing periodic silence phases), discovering nearby CRs and spectrum handoff mechanisms that allow seamless transitions between frequencies without breaking secondary user transmissions. Furthermore, none of these proposed models except the *crSimulator* has been made available to the research community. Hence, modelling of CR networks continues to be a research challenge with the above functionalities yet to be implemented.

CHAPTER 3

Fundamentals

3.1 Vehicular Communication Standards

Standardisation plays an important role in ensuring the safety and reliability of the procedures, products and services used every day. Standards assure compatibility and interoperability of devices and open up market access. This encourages the development and implementation of new technologies that could transform the future. In the context of ITS, standardisation and regulation are required to facilitate communication between vehicles of various automobile manufacturers. The standardisation of vehicular communications is a complex and slow process. Currently, there is no single standard that enables a globally interoperable ITS system.

In the USA, the FCC collaborates with the IEEE Standards Association, an organisation within the Institute of Electrical and Electronics Engineers (IEEE), to create standards. The IEEE standard related to dedicated short-range vehicular communications is called WAVE (Wireless Access in Vehicular Environments) [15]. This chapter discusses in detail the elements of this standard. In Europe, the European Telecommunications Standards Institute (ETSI), an independent, not-for-profit organisation, has standardised a different protocol stack for inter-vehicular communications called ETSI ITS-G5 technology [81], [82]. The Japanese standard for vehicular communications is called ARIB (The Association of Radio Industries and Businesses) [21]. However, the details of both the ETSI and the ARIB standards are not covered in this thesis.

The IEEE WAVE standard provides a communication protocol stack optimised for the vehicular environment, as illustrated in Figure 3.1. The physical layer and lower medium access control (MAC) sublayer of the IEEE WAVE standard are defined by the IEEE 802.11p protocol [83], which is now incorporated in the IEEE standard 802.11-2012 [84], while the upper layers are represented by the family of IEEE 1609 standards [15]. The IEEE standard 802.11p provides extensions to IEEE standard 802.11-2012 to facilitate communications taking place in rapidly varying vehicular communication environments over very short durations. The IEEE standard 1609.4 [85] specifies the operations over multiple radio channels, while the networking services are defined in the IEEE standard 1609.3 [86].

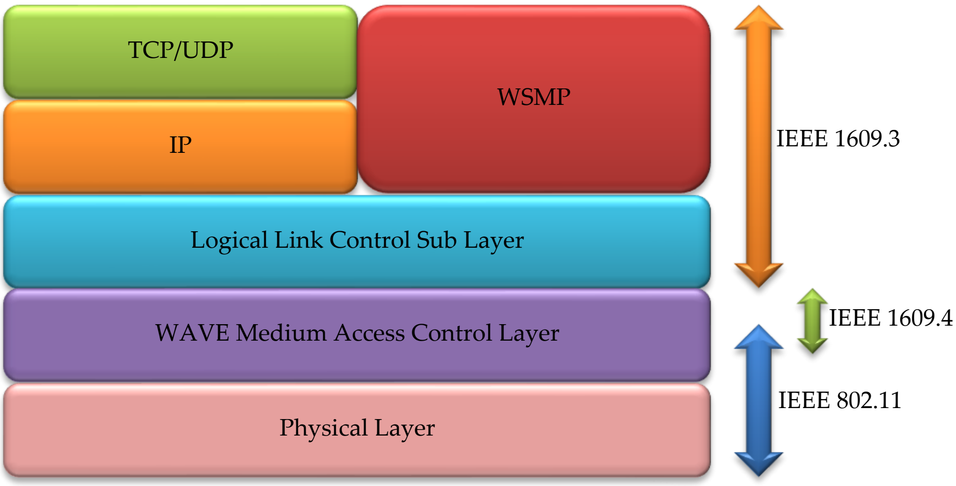


Figure 3.1: IEEE WAVE communication protocol stack [15]

3.1.1 IEEE Standard 802.11p

The IEEE standard 802.11p provides extensions to the IEEE standard 802.11-2012 for operating outside the context of a basic service set (BSS). The BSS is the basic building block of an IEEE 802.11 wireless local area network (WLAN) that facilitates communication between wireless devices. A central device called an Access Point (AP) forms a BSS and centralises access and control over a group of wireless devices. To gain access to the services provided by a BSS, the devices must become associated or members of a BSS. All communications between wireless devices in a BSS traverse the AP, rather than communicating directly with each other.

The extensions provided by the IEEE 802.11p protocol allows wireless devices that are not associated with a BSS (outside the context of a BSS) to transmit data frames. When a device is not a member of a BSS, it does not use the IEEE 802.11 services such as authentication, association, or data confidentiality. Hence, this facilitates immediate communication with an individual or a group of wireless devices, while avoiding the latencies associated with establishing a BSS. Such form of communication is particularly suitable for use in rapidly varying communication environments such as VANETS. A device may transmit a data frame outside the context of a BSS only if the *dot11OCBActivated* parameter is 'true'.

3.1.2 IEEE Standard 1609.4

In the US, 70MHz of the spectrum at 5.855-5.925 GHz has been currently allocated for vehicular communications under the IEEE WAVE standard. There are two classes of radio channels defined in the WAVE standard, namely control channel (CCH) and service channel (SCH). The 70 MHz bandwidth is divided into one CCH and six SCHs allocating 10 MHz for each channel, as shown in Figure 3.2. The IEEE channel 178 is considered as the CCH, while the channels 172, 174, 176, 180, 182 and 184 are occupied as SCHs. The CCH is frequently utilised as a common channel for signalling, whereas the SCHs are intended for general-purpose application data transfers.

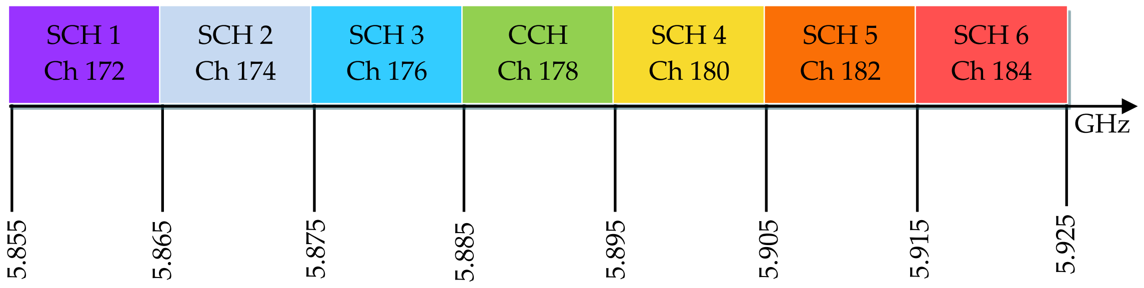


Figure 3.2: DSRC channel allocations [15]

Since the spectrum allocated for vehicular communications consists of seven channels, the WAVE devices are required to perform channel coordination to operate over multiple wireless channels while operating outside the context of a BSS. In support of multi-channel operations, the IEEE1609.4 standard specifies channel coordination features and management functions as extensions to the IEEE802.11 MAC sublayer. The standard divides the channel access time into two 50ms repetitive intervals to facilitate single-physical layer devices to access multiple channels alternatively, for example, a device may operate on the CCH during the initial time interval (time slot 0) and switch to a designated SCH at the start of the second channel interval (time slot 1), as shown in

Figure 3.3. The WAVE devices switching channels on time slot boundaries synchronise to the Coordinated Universal Time (UTC), modulo 1 second, common time reference, which may be obtained from a local source such as Global Positioning System (GPS) equipment or derived from timing information received over the air from other WAVE devices.

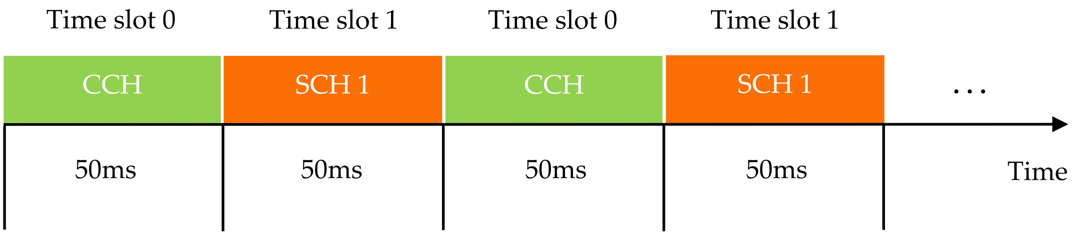


Figure 3.3: A channel access example [85]

The wireless channel access mechanism used by IEEE 802.11 based devices is carrier sense multiple access/collision avoidance (CSMA/CA), in which nodes listen to the channel before transmission to identify whether another node is transmitting and shall attempt to transmit, if the channel is determined to be idle for a specified listening period. However, if the channel is deemed busy, a node should defer until the end of the current transmission and perform a back-off procedure (after determining the medium is idle for the specified listening period) before attempting to transmit [84].

The back-off procedure draws a random integer from a uniform distribution over the interval [0, CW] and decreases it while the channel is idle, at the rate of one decrement per slot time, which is 13 μs for a 10 MHz channel, as per IEEE standard 802.11-2012 [84]. CW refers to the current value of the contention window, which will be doubled from its initial value (minimum contention window - CWmin) until it reaches a maximum value (CWmax), with every failed transmission attempt of a specific packet (as implied by the absence of an acknowledgement frame). However, for broadcast packets, the CW value is always set to CWmin, due to the lack of acknowledgements in broadcast transmissions.

If the channel is determined to be busy during the back-off, a node suspends the countdown until the channel becomes idle and resumes after the medium is deemed idle for the specified listening period. A node shall transmit when the value of the back-off reaches zero. After successful transmission of a packet, the CW value will be set to its minimum value, CWmin.

The IEEE WAVE devices add quality-of-service (QoS) attributes to the CSMA/CA mechanism to transmit data packets with specific priorities. This method of channel access is called Enhanced Distributed Channel Access (EDCA), in which high-priority data has increased probability of receiving access to the channel than low-priority data. The levels of priorities in EDCA are called access categories (ACs), and four access categories (ACs) are defined in the IEEE standard 802.11-2012 to support differentiating user priorities [84]. In EDCA, nodes maintain queues for each access category with a standard set of parameters defined for each AC. Differentiation between these queues (ACs) is achieved by varying the following parameters:

- Amount of time a device senses a channel to be idle (listening period) before transmission or back-off, called the Arbitration Inter-Frame Space (AIFS).
- The minimum and the maximum values of the contention window (CWmin and CWmax) to be used for the back-off.

AIFS is derived from the formula (3.1), where Short Inter-Frame Space (SIFS) is the amount of time required to receive the last symbol of a frame at the air interface and respond with the first symbol on the air interface of a response frame. The Arbitration Interframe Space Number (AIFSN) is also dependent on the AC, and it indicates the number of slots after a SIFS duration a device should wait before either starting a transmission or invoking a back-off. The default EDCA parameter set used in IEEE802.11p (with the *dot11OCBActivated* parameter set to ‘true’) is tabulated in Table 3.1, while the physical layer characteristics are given in Table 3.2.

$$AIFS[AC] = AIFSN[AC] \times Slot\ Time + SIFS \tag{3.1}$$

Table 3.1: Default EDCA parameter set used in IEEE802.11p as per IEEE standard 802.11-2012 [84]

Priority	Access category (AC) number	Access category (AC) description	Minimum contention window (CWmin)	Maximum contention window (CWmax)	AIFSN
Lowest	0	AC_BK (Background)	aCWmin	aCWmax	9
↓	1	AC_BE (Best effort)	aCWmin	aCWmax	6
	2	AC_VI (Video)	[(aCWmin+1)/2]–1	aCWmin	3
Highest	3	AC_VO (Voice)	[(aCWmin+1)/4]–1	[(aCWmin+1)/2]–1	2

Table 3.2: The IEEE802.11p physical layer characteristics as per IEEE standard 802.11-2012 [84]

Characteristics	Value
Slot Time	13μs
SIFS	32 μs
aCWmin	15
aCWmax	1023

3.1.3 IEEE Standard 1609.3

The IEEE1609.3 standard specifies networking services and the associated management functions required for WAVE operation. It also introduces a protocol optimised for the wireless vehicular environment called the WAVE Short Message Protocol (WSMP). The WSMP is suitable for message-based applications, and environments that cause to experience intermittent radio connectivity. The messages based on WSMP, called WAVE Short Messages (WSMs), may be transmitted on both CCH and SCHs as they consume minimal channel capacity. The WSMP messages are comprised of a WSMP header and a WAVE Short Message (WSM) data (Figure 3.4). The IEEE1609.3 standard also supports the generic Internet Protocol version 6 (IPv6). However, Internet Protocol (IP) traffic is allowed only on SCHs to keep the CCH free of high-volume traffic.

Figure 3.4: WAVE short message format [86]

The IEEE1609.3 standard defines two WAVE device roles, namely provider role and user role. WAVE devices offering data services on one or more channels are considered as service providers (SPs). The SPs indicate their availability for data exchange through the broadcasting of WAVE Service Advertisements (WSAs). The service users (SUs) interested in obtaining services from SPs monitor for received WSAs and may assign SCHs indicated in WSAs. WSMP is utilised to send WSAs, where WSAs are encapsulated in WSM data. A WSA may be sent secured or unsecured. The WSA packet format is illustrated in Figure 3.5. The service information segment provides information relating to services being offered by a WAVE device broadcasting WSAs, while the channel information segment offers information about channels utilised by these application services.

If an SP offers IPv6 based application services, information about infrastructure internetwork connectivity, such as the IPv6 subnet prefix of the link and the default gateway, are provided in the WAVE Routing Advertisement of the WSA. This allows receiving devices to be configured to partake on the advertised IPv6 network.

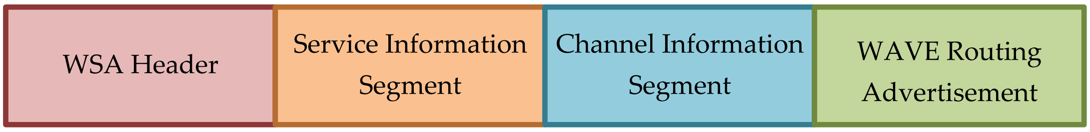


Figure 3.5: WAVE service advertisement format [86]

3.2 Simulating Vehicular Communication Systems

Performance evaluation is an important aspect of vehicular communication research. There are several ways to evaluate the performance, in other words, assess the benefits and drawbacks, of vehicular communication systems. Gathering performance data through field experiments gives immediate feedback regarding the components employed in a vehicular communication system. Nevertheless, to relate these experimental results to a full-scale vehicular communication system, a large number of testbeds needs to be deployed. This usually involves high costs and presents substantial implementation complexities. Another approach is to perform an analytical evaluation of the vehicular communication system under consideration [87]. However, evaluation of the entire vehicular communication system through this method is rather difficult, often requiring to make simplified assumptions or examine isolated parts of the system.

Due to these reasons, studies on inter-vehicular communication systems is generally carried out through simulations. Simulations facilitate researchers to study and evaluate every part of the communication system, yet without the hassles and excessive expenses of real testbeds and the simplifications of analytical models. Simulations allow systems to be evaluated during design stages before such systems are built and deployed. Hence, simulations serve as a design tool to predict the performance of new systems under varying conditions, as well as an analysis tool to foresee the outcomes of modifications made to an existing system.

A simulation imitates the operation of a real-world system and draws inferences regarding its operating characteristics [88]. Many real-world systems, such as inter-vehicular communications systems, are extremely complex that numerical, computer-based simulations are required to imitate their behaviour. Simulation models are developed to study the operation of a system as it progresses over time. These models are formed of a set of assumptions, expressed mathematically or logically, relating the objects of interest of the system. Once the simulation models are developed and validated, data is collected from the simulation, as if a real system is being observed, to evaluate the performance of a system [71].

3.2.1 Types of Simulation Models

Simulation models are commonly categorised as being discrete or continuous. While the states of discrete models change only at distinct points in time, the states change continuously over time in a continuous model. Although there is no hard and fast rule to use a discrete or continuous model to represent a system, the choice depends on the characteristics of the system and the objective of the study. For example, in vehicular communications, if the characteristics and movement of individual data packets over a communication channel are important, the channel could be modelled discretely. On the other hand, if the flow of packets in aggregate over a channel is of significance, a continuous model may be used to describe the system appropriately. A simulation model may also be classified as dynamic or static (depending on whether they represent a system that changes over time or at a particular point in time), and as stochastic or deterministic (depending on whether they contain random variables as input or not).

In terms of inter-vehicular communications, the behaviour of the system will change over time and effects on the system will be random. Furthermore, there will be separate points in time at which the state of the system may change, or in other words, events may occur. The modelling of such a system is called discrete event simulation, and it can be considered the most relevant method of representing the inter-vehicular communication systems.

3.2.2 Communication Network Simulation

There are several discrete event simulation platforms, such as OMNeT++, ns-3, OPNET Modeler (Riverbed Modeler), QualNet, etc., that are generally employed nowadays to build simulation models for communication networks. As discussed in Chapter 2, out of these simulation platforms the free and open source OMNeT++ platform was chosen to develop the VANET and CR related simulation models required for this research due to its extensible and modular component architecture, GUI support and the availability of partially developed VANET simulation models through third parties. In this section, the fundamental components of OMNeT++ will be discussed in detail.

OMNeT++ is a general-purpose simulator that provides the necessary infrastructure and tools for building simulation models. It follows an object-oriented, hierarchical approach to modelling where the models are assembled from reusable components called *modules* that communicate by passing messages to each other. The OMNeT++ simulations models are commonly known as *networks*. The fundamental components of OMNeT++ are described below.

- *System module* – Sits at the highest level of the module hierarchy and is comprised of submodules that may also encompass submodules themselves. The depth of module nesting is not limited in OMNeT++.

- *Compound modules* – Modules that hold submodules.
- *Simple modules* – The active components containing the algorithms of the OMNeT++ model that form the lowest level of the module hierarchy.
- *Gates* – The input and output interfaces of modules.
- *Connections* – The links created between the gates within a single level of the module hierarchy. For example, in a compound module encompassing multiple submodules, gates of two submodules, or a gate of a submodule and a gate of the compound module may be linked. Connections may be assigned with properties such as propagation delay, data rate and bit error rate.
- *Channels* – Reusable connections that encapsulate specific properties of a connection.
- *Messages* – Arbitrary data exchanged between modules. For example, in communication network simulations, these messages may represent frames or packets. Messages are transmitted through output gates and received through input gates.

Figure 3.6 shows the hierarchical modelling concept of OMNeT++, where a compound module containing two simple modules connect to another simple module via gates.

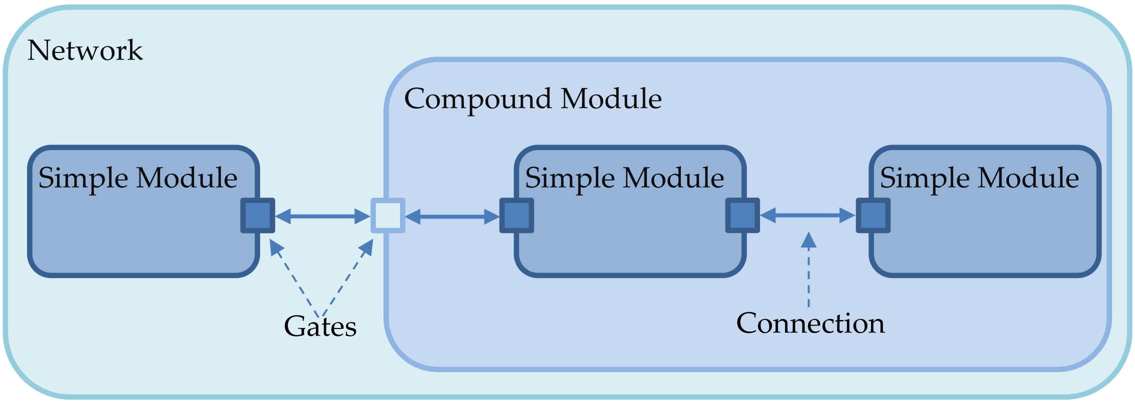


Figure 3.6: Hierarchical modelling concept of OMNeT++

The algorithms that handle the behaviour of a simple module such as sending and receiving messages are written in C++ linking to the OMNeT++ kernel (see Appendix A.3). However, the code that describes the structure of a simulation model (for example, modules and channels) are programmed through a high-level language called NED (Network Description). NED enables declaration of simple modules and connecting them in compound modules (see Appendix A.1). Furthermore, the codes that declare the message structures are stored in plain-text message definition files (see Appendix A.2), while the parameters passing configuration data to simple modules during runtime are stored in initialisation files.

3.2.3 Road Traffic Simulation

In order to accurately represent the characteristics and functions of VANETs, modelling only in the context of communication technology may not be sufficient. A realistic VANET simulation should also model the motions of vehicles since communications and vehicular movements go hand in hand [89]. Most network simulators such as ns-3 or OMNeT++ have integrated support for node mobility to a degree of sophistication [90], [91]. The synthetic models included in these simulators employ mathematical models for describing the behaviour of mobile nodes. They are able to model deterministic movements of nodes along linear, polygonal or circular paths, as well as nodes that move randomly without any restrictions. Examples of random movement models include Random Waypoint Mobility (RWM), Random Walk Mobility, Gauss Markov Mobility and Mass Mobility.

The RWM model is commonly employed in VANET simulations due to its ease of use [92]. The waypoints in this model are uniformly distributed over a given area, and the nodes will travel between the waypoints on a straight line at random speeds chosen from a uniform distribution [93]. However, the model may produce erroneous results due to the inability of the model to reach a steady average node speed [94]. In VANET simulations, inaccurate representation of vehicular traffic may eventually lead to wrong conclusions, even when the communication level modelling is flawless [95].

A mobility model intending to produce an accurate and realistic vehicular motion pattern should include road topologies, obstacles, characteristics of individual vehicles such as acceleration, deceleration and speed capabilities (microscopic description), driver behaviours, etc. [92]. Due to the complexity of modelling vehicular mobility, the general synthetic mobility models are not adequate to generate realistic vehicular motion patterns. Traffic simulators, on the other hand, include refined synthetic models which have been validated based on real mobility traces or behaviour surveys. Mostly developed for urban traffic engineering, fine grain commercial traffic simulators such as Aimsun [96], Paramics Microsimulation [97], PTV Vissim [98], etc. or SUMO are capable of modelling individual vehicle traffic at a microscopic level. However, only some of these simulators facilitate interconnection with a network simulator.

On the other hand, the results of scientific work are comparable only when the simulators used to evaluate them have similar features, models or implementation details. Hence, using a readily available free and open source software simulator, which is used by many researchers for their applications would enable comparability of research. In this section, the most popular of those, the Simulation of Urban Mobility (SUMO) is briefly introduced.

SUMO is a microscopic road traffic simulation package developed by the Centre for Applied Informatics and the Institute for Transportation Research at the German Aerospace Centre [99]. Its first implementation started in the year 2001, and it has been developed continuously since then. SUMO is a free and open source software available under the GNU General Public License (GPL) version 3. It is designed to simulate large road networks, and it enables modelling of multimodal traffic systems including road vehicles, public transport and pedestrians. It also provides external control through an open Application Programming Interface (API) called TraCI, short for Traffic Control Interface. It is for these reasons that SUMO has achieved its popularity among the VANET research community.

SUMO uses a modified version of the time-discrete and space-continuous single lane Krauß Car-Following Model (CFM) [100] as its default. The CFM adapts a following car’s mobility as per a set of rules without colliding with the leading vehicle. However, the need to change to a particular lane or the navigational aspect of choosing a lane, in order to be able to continue a route is not regarded in this model [101]. More recently, SUMO has also adopted the same complex Gibbs model [102] for lane changing called the Intelligent-Driver Model (IDM) from Treiber *et al* [103].

SUMO road traffic simulations support multiple lanes that can be restricted to certain classes of vehicles. Each lane can be interconnected with junctions and the traffic at intersections can be regulated by either right-of-way rules or by traffic lights. SUMO supports the importation of complex road networks and demand data from various sources such as OpenStreetMap [104], or generation of the same manually. OpenStreetMap provides free geographic data such as street maps, contributed by a community of mappers, under the Open Data Commons Open Database License (ODbL). It also includes openly-licenced data from national mapping agencies in some countries, for example, Australia, Austria, Canada, Finland, France, Netherlands, New Zealand, Slovenia, South Africa, the United Kingdom and the United States.

3.2.4 Veins Integrated Simulation Framework

'*Veins*' (short for Vehicles in Network Simulation) [105] is a vehicular network simulation framework (Figure 3.7) that executes on OMNeT++ while integrating vehicular mobility through SUMO. Each *Veins* simulation is performed by running both OMNeT++ and SUMO in parallel. During simulation runs, the two simulators exchange commands and vehicular mobility data via a TCP connection established through the Traffic Control Interface provided by SUMO. The *Veins* framework reflects movements of vehicles in SUMO as movements of nodes in an OMNeT++ simulation. It relies on the functionalities provided by MiXiM (**mix**ed s**im**ulator) [106] for accurate modelling of physical layer effects such as modulation and coding, and channel models. MiXiM incorporates several channel models that simulate path loss, log-normal shadowing, Jakes fading, etc. The physical layer is responsible for sending and receiving frames, applying various channel effects and bit error calculations [107].

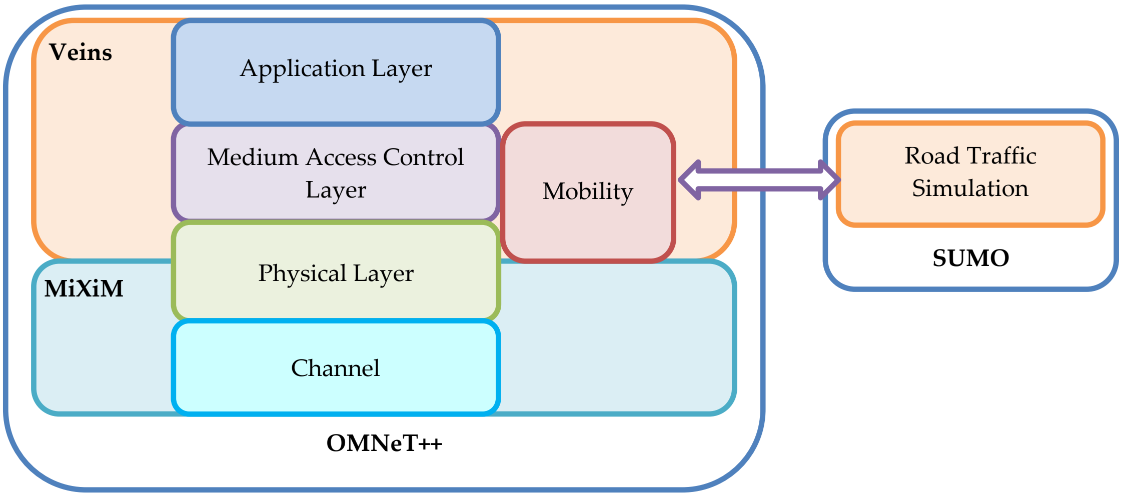


Figure 3.7: The Veins simulation framework [105]

The *Veins* application layer generates WSMs and passes down to the MAC layer where they are placed in an Enhanced Distributed Channel Access (EDCA) queueing subsystem, as shown in Figure 3.8. EDCA is used by WAVE devices when contending for channel access to transmit packets with specific user priorities. The EDCA mechanism defined in the IEEE standard 802.11-2012 includes four access categories (ACs) or queues to facilitate differentiating user priorities [84]. The operation of EDCA is explained in detail in Section 3.1.2. The *Veins* MAC layer supports alternating channel switching between the CCH and an SCH defined at the start of the simulation. Hence, the *Veins* framework contains two EDCA subsystems (one for each channel type, CCH and SCH) with four queues in each subsystem. Each queue has an independent EDCA function (EDCAF) that determines when a packet in a queue is permitted to be transmitted via the wireless medium. It operates the back-off counter for each queue and initiates the transmission of a packet in a queue. If packets from two access categories are ready to be transmitted at the same time (i.e. internal contention), the packet from the higher priority AC receives the opportunity to transmit, while the contention window size is doubled and the back-off procedure is invoked on the lower priority AC [108].

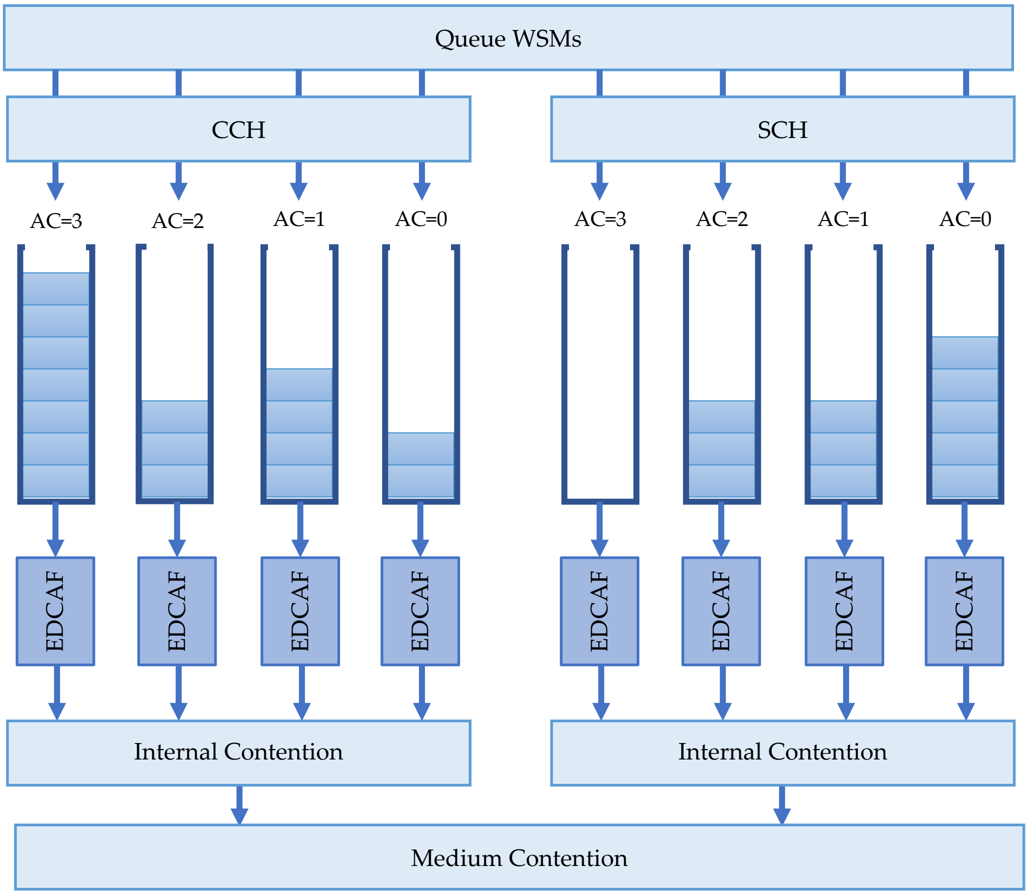


Figure 3.8: Veins EDCA implementation

CHAPTER 4

A Simulation Framework for Application Service Management in VANETs

4.1 Introduction

The IEEE WAVE family of standards provides the foundation for a broad range of vehicular communication applications. While traditional applications of VANETs enhance road safety and reduce traffic congestion, novel applications provide travellers with ubiquitous wireless communication services such as information, advertisements, and entertainment to make their journeys convenient and enjoyable. Since there are numerous types of applications, a proper mechanism is required to manage and deliver these services. Simulators are often required to model these complex application scenarios of VANET, but none of the currently available simulators supports WAVE application service management. The primary goal of this work was, therefore, to develop a simulation framework for application service management in VANETs as per the IEEE WAVE standards.

The simulation framework was developed on '*OMNeT++*' [75] extending the '*Veins*' (short for Vehicles in Network Simulation) [105] vehicular network simulation framework described in Chapter 3. The *Veins* framework currently supports vehicular mobility, radio signal propagation, and Quality of Service (QoS) enabled channel access. During this research, the *Veins* framework was extended to model the following functions.

- Multi-radio multi-channel communications
- Application service provider and user roles
- Application service advertisement and discovery
- Simultaneous application service operation

- Data transfers over distinct radio channels
- Continuous channel access, as well as alternating channel access between any two channels on a radio

4.2 General Structure

4.2.1 Simulation Modules

As discussed in Chapter 3, *OMNeT++* is a discrete event simulator, which has gained widespread popularity as a network simulation platform across a broad scientific and industrial user community. The simulation models of *OMNeT++* are developed using extensible and modular components. The algorithms of the *OMNeT++* simulation models (programmed in C++) are contained within simple modules, while compound modules encompass these simple modules. The *Veins* simulation framework (which executes on *OMNeT++*) was developed using '*MiXiM*' (**mix**ed s**im**ulator) [106] that includes models of the environment, wireless connectivity, wireless channel, signal transmission and reception. The *Veins* framework offers additional models to simulate vehicular mobility, WAVE short message (WSM) generation, alternating channel access between two fixed channels (CCH and a SCH) and Enhanced Distributed Channel Access (EDCA) that adds quality-of-service (QoS) attributes to the carrier sense multiple access/collision avoidance (CSMA/CA) mechanism [108].

In *Veins*/*MiXiM* simulation frameworks, the network simulations are carried out on a limited two or three-dimensional space in *OMNeT++* called '*playground*'. The '*world utility*' simple module is responsible for collecting input parameters such as the dimensions of the network, number of dimensions and if the *playground* is modelled as a torus (details are provided in Section 4.3.2). A *playground* may include nodes such as roadside units (RSUs) and vehicles, as well as obstacles. Nodes are compound modules that represent wireless communication devices with protocol stacks (particulars are given in Section 4.2.2). They are modelled as dimensionless entities having positions on the *playground* that can be either fixed or mobile. Obstacles, on the other hand, are non-communicating entities with physical dimensions that locate on the *playground* such as buildings. Wireless signals that propagate through these obstacles will experience signal attenuation and the '*Obstacle Control*' simple module models these obstacles that block radio signal transmissions. The *Connection Manager* module is responsible for dynamically managing the connections between nodes [107]. The details of this module are discussed in Section 4.3.1. Figure 4.1 shows an example of a vehicular communication network, which comprises of RSUs, vehicles, obstacles (buildings and houses) and management modules.

Figure 4.1: An example vehicular communication network

In the *Veins* simulation framework, realistic vehicular motion patterns were modelled through the '*SUMO*' road traffic simulation platform. *SUMO* provides external control through an open Application Programming Interface (API) called '*TraCI*', short for Traffic Control Interface [99]. During simulation runs, both *OMNeT++* and SUMO are executed in parallel where the two simulators exchange commands and vehicular mobility data via a TCP connection established through the *SUMO* Traffic Control Interface. The '*Traci Scenario Manager*' module of *Veins* simulation framework reflects movements of vehicles in *SUMO* as movements of nodes in *OMNeT++* [105]. Since *OMNeT++* is an event-based simulator, mobility is handled by scheduling node movements at regular time intervals. These distinct node movements in OMNeT++ go hand in hand with *SUMO*, which also advances simulation time in discrete time steps [109].

The *Traci Scenario Manager* module of *Veins* framework and SUMO buffer the commands arriving in-between the simulators' time steps to assure the synchronous execution at defined time intervals. At each time step, *Veins* sends buffered commands to *SUMO* to execute the corresponding time step of the road traffic simulation. After completing the time step, SUMO sends mobility data (for example, position, speed, etc.) of all vehicles back to *Veins*. The *Veins* network simulator responds to received vehicular mobility data by introducing new nodes in *Veins*, moving the already introduced nodes and removing nodes that had reached their destination [109].

4.2.2 Node Modules

The vehicle and RSU nodes are modelled as a compound module that encompasses various sub-modules. Figure 4.2 shows the modules that make up the compound module of a node in the developed simulation framework. The underlying functionalities of each module in an RSU node are identical to that of a vehicular node, except mobility. The mobility module of an RSU sets its position (no movement) within the simulation area (*playground*), whereas the mobility module related to a vehicular node ('*Traci Mobility*') handles the motion of an existing vehicle in the network simulation in conjunction with the simulation module, *Traci Scenario Manager*. The *Traci Mobility* and the *Traci Scenario Manager* modules are part of the *Veins* simulation framework. The operations and algorithms pertaining to the mobility of vehicles are beyond the scope of this thesis.

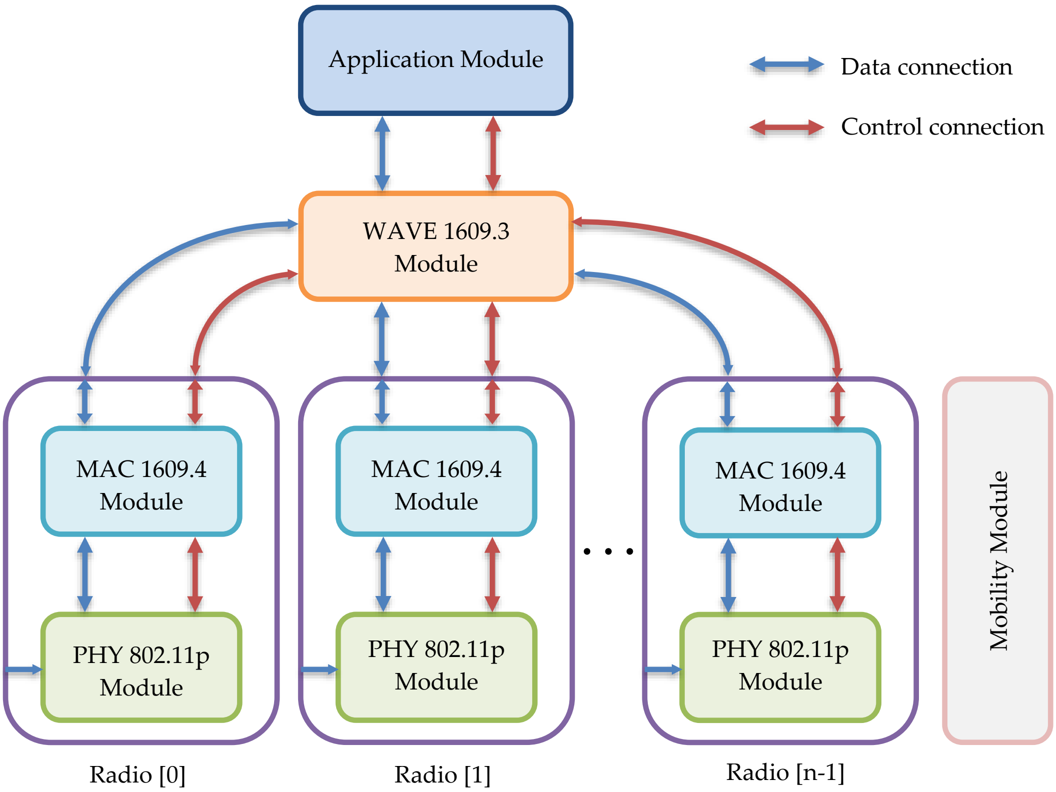


Figure 4.2: A simulation model of a node

The application module, the WAVE 1609.3 module and an array of radio (transceiver) modules constitute the developed communication model of a node. A radio is a compound module consisting of a physical layer (modelled by the PHY 802.11p module) and a medium access control layer (modelled via the MAC 1609.4 module). The application services offered by WAVE devices are implemented in the application module, while the WAVE 1609.3 module integrates the WAVE application service management framework and channel coordination functions required to operate over multiple wireless channels. The WAVE 1609.3 module also generates WAVE service advertisements and WAVE short messages as requested by the application module (details are given in Section 4.3.4). The WAVE 1609.3 module can be connected to any number of radio modules, where the number is configured in the *OMNeT++* '*Initialization File*'. The '*Initialization File*' comprises of settings that control how the simulation is executed and values for model parameters. The MAC 1609.4 module incorporates radio channel switching models and the QoS enabled CSMA/CA model (discussed in Section 4.3.5), while the PHY 802.11p module simulates the transmission and reception of signals (details are provided in Section 4.3.3). The data and control messages are passed between PHY 802.11p, MAC 1609.4, WAVE 1609.3 and application modules using two types of connections. The data messages such as WSMs are passed from one module to the other via 'data connections', whereas the control messages (for example, channel change requests) are exchanged through the 'control connections'.

As mentioned in Chapter 3, the WAVE devices that perform channel switching are required to synchronise to a common time reference (as specified in the IEEE1609.4 standard) to support channel coordination in a multi-channel environment. Hence, in this design, the simulation time is used as the common time base, and the timing information is made available to both MAC 1609.4 and WAVE 1609.3 modules.

4.3 Model Implementation

4.3.1 Wireless Connection Modelling

In wireless simulations, connectivity modelling is a challenging task as compared to wired simulations. While two nodes linked by wires can be easily modelled in *OMNeT++* by '*connections*', the wireless channel between two nodes cannot be easily represented by a single *connection*, as it is a broadcast medium. A signal transmitted by a node affects all other nodes operating in the same frequency range as the transmitter, unless the received power at nodes that are far away from the transmitter is significantly low after signal attenuation, that it goes undetected.

The '*Connection Manager*' module in the *Veins/MiXiM* simulation framework is responsible for maintaining and updating wireless connectivity information between nodes, or more specifically, radio interfaces. It evaluates the distances between radio interfaces at discrete time intervals and establishes connections between them only if they are within the maximal interference distance (4.1) of each other [107], [110]. This technique reduces the computational complexities of wireless communication simulations.

$$Maximal\ interference\ distance = \left[\frac{P_t\lambda^2}{(4\pi)^2 10^{R_s/10}}\right]^{1/n} \tag{4.1}$$

P_t - maximum transmission power used for this network [mW]

R_s - the sensitivity of the physical layer [dBm]

n - path loss coefficient

λ - wavelength

In order to be considered for making connections, radio interfaces need to be registered with a *Connection Manager*. A *Connection Manager* maintains node connection information related to a particular frequency (for example, the centre frequency of the DSRC band). In the *Veins* simulation framework, the number of radio interfaces on a node that can be allocated to a *Connection Manager* is limited to one. This limitation prevents modelling scenarios that require simultaneous data transfers on multiple radio interfaces, for example, modelling an application service that needs to broadcast management messages (such as service advertisements) on one channel and transmit application data on another channel at the same time.

Hence, the *Veins* simulation framework was extended during this research to accommodate multiple radio interfaces on a node. In the extended framework, the multiple radio interfaces on a node can be registered with one or more *Connection Manager* modules. When several radios on a node are allocated to the same *Connection Manager* (i.e. the radios operate in the same frequency band), the radios get wirelessly connected and may interfere with each other, if they switch to the same radio channel at the same time (note that the physical layer can switch to different radio channels). However, the developed WAVE 1609.3 module together with the MAC 1609.4 module manage access to radio channels and prevent radios from switching into the same channel during the same time interval. This situation is discussed in detail in Section 4.3.4.1.

In the simulation framework, connections are created among the radio interfaces that are registered under a particular *Connection Manager*, and no connections are made between radio interfaces that are assigned to various *Connection Manager*s (i.e. different spectrum ranges). Hence, when radios on a node are registered with different *Connection Manager* modules, the spectrum bands supported by different *Connection Managers* must be sufficiently separated, since interferences between various *Connection*

Managers are not considered. For example, two *Connection Managers* can be employed to maintain radios operating in the 5.9 GHz DSRC and 2.4 GHz ISM bands.

4.3.2 Environmental Model

When simulating vehicular communication scenarios, the limited area of the *playground* may produce undesired border effects, which may affect the simulation results. For example, Figure 4.3 illustrates a 1km highway in a rectangular *playground*. In this example, the vehicles can establish connections between them if they are within the maximum interference distance of 500m. The vehicles in the middle of the road segment, hence, will receive interference from approximately all the vehicles on the highway. On the other hand, vehicles that are closer to the left border will only interfere with vehicles which are moving between the left border and the middle of the road segment, and vice versa.

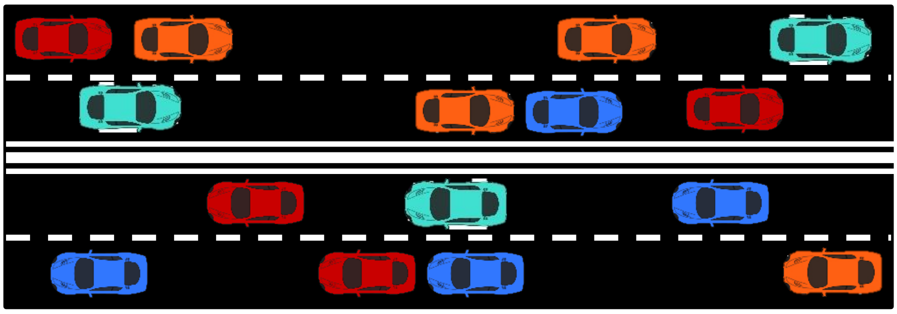


Figure 4.3: Vehicular movements in a rectangular space in OMNeT++

In order to perform simulations avoiding such border effects, the edges of the rectangular *playground* may be connected together (the top edge with the bottom edge and the left edge with the right edge) to model the simulation area as the surface of a torus with toroidal distances between nodes (as shown in Figure 4.4). On a torus, if the distance between two points on an axis is longer than half the size of the axis, a shorter path exists over the border on this axis. Therefore, the simulation models the network topology in a way that nodes nearer to a border are considered as being closer to vehicles at the opposite border and are allowed to establish links. Hence, on a torus, vehicles travelling closer to the borders of the *playground* are subjected to similar interference conditions as vehicles in the middle of the *playground*.

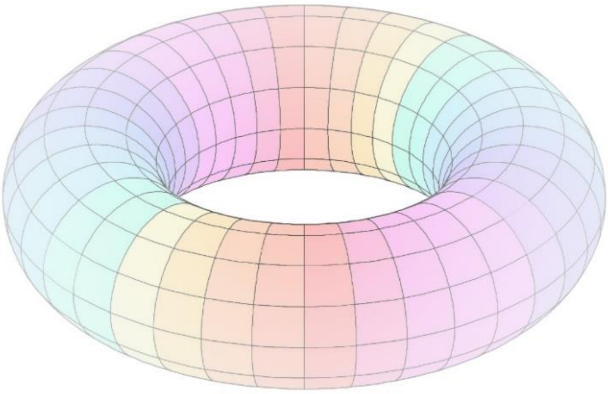


Figure 4.4: Simulation area modelled as a torus

4.3.3 Physical Layer Model

The physical layer modelled by the PHY 802.11p module is responsible for sending and receiving messages, filtering signals, collision detection, bit error calculation and applying the channel models used in the simulation. The *Veins/MiXiM* physical layer is divided into three parts, '*Base Physical Layer*', '*Analogue Models*' and '*Decider*' [107], as illustrated in Figure 4.5. The *Base Physical Layer* of a node provides interfaces to its MAC layer and handles sending and receiving signals over the wireless medium, while the *Analogue Models* simulate attenuation (such as path loss, shadowing and fading) of a received signal. The *Decider* is responsible for evaluating the received messages and classifying them as noise or signal. It also performs demodulation (bit error calculation) of the received signal.

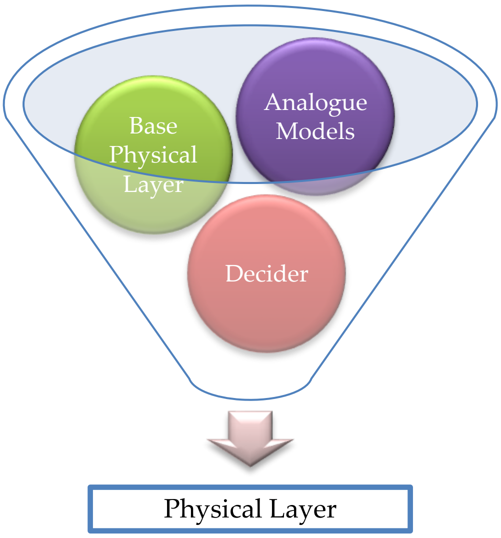


Figure 4.5: Modules and classes that make up the physical layer

4.3.3.1 Signal Transmission

The strength of a signal transmitted from one node to another is influenced by the environment it travels through. These effects are modelled with signal attenuation factors caused by path loss, shadowing and fading. In order to simulate such real-world wireless communication signals each transmitted message is attached with a '*Signal*' model representing sending power, attenuation, and bit-rate in the three dimensions time, frequency, and space [107]. A transmitting node specifies the sending power, frequency and bit-rate whereas the receiving node adds attenuation to a *Signal*. A *Signal* is encompassed in an '*Air Frame*' message, which is then exchanged between physical layers of different nodes.

Figure 4.6 shows the chain of events that occur when transmitting a message. At the MAC layer, when a WAVE Short Message (WSM) gets selected for transmission after contending for channel access, a MAC packet encompassing the selected WSM is created and passed down to the *Base Physical Layer* together with information (power, frequency and bit-rate) on how the packet should be transmitted. The *Base Physical Layer* makes use of this information to create a 'Signal' that gets encompassed in an *Air Frame*. Subsequently, the MAC packet handed down to the *Base Physical Layer* is encapsulated in the *Air Frame*. When a MAC packet is encapsulated in an *Air Frame*, its packet length gets increased by the length of the encapsulated *Air Frame*. Next, the *Base Physical Layer* queries the *Connection Manager* for a list of radio interfaces connected to the transmitting radio and calculates signal propagation delay for each connected radio. Finally, it delays the transmission by the calculated propagation delay and delivers the *Air Frame* to the physical layers of linked nodes.

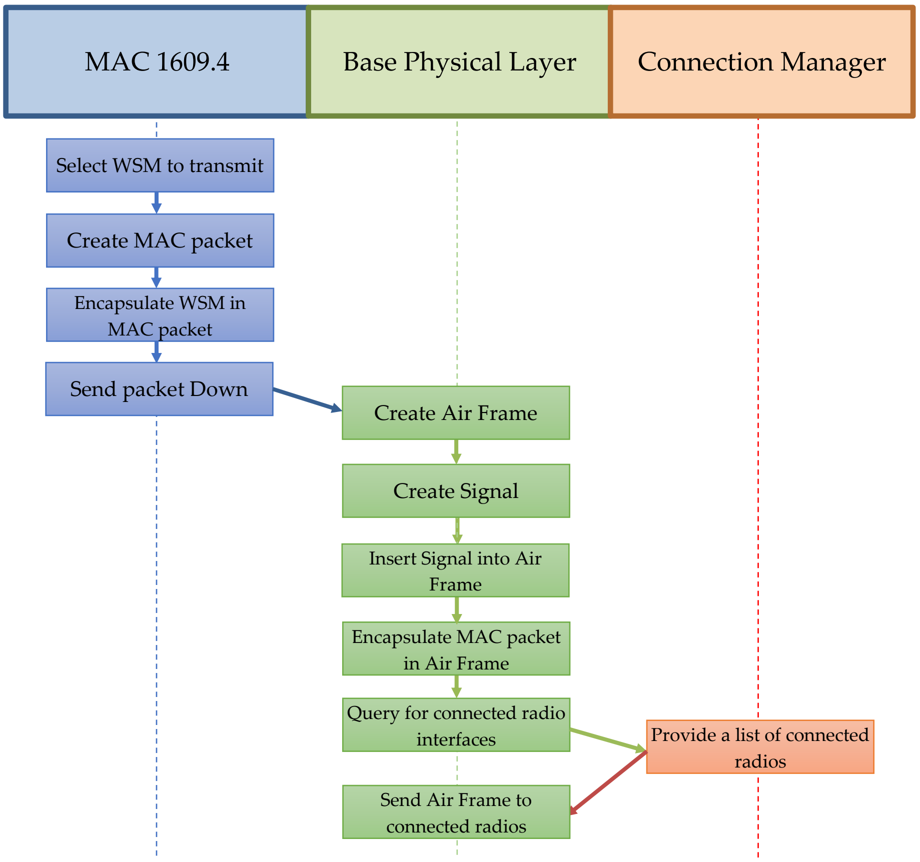


Figure 4.6: Events that occur at different modules when transmitting a message

4.3.3.2 Multi-Channel Physical Layer

The physical layer of a node is allowed to switch into multiple channels, and access to these channels are managed through the MAC 1609.4 module. For example, a physical layer operating in the DSRC frequency band can change to any of the 10MHz channels given in Table 4.1 as governed by the MAC layer. However, it should be noted that when establishing wireless connections between radio interfaces, the *Connection Manager* utilises only the centre frequency of a spectrum band (for example, 5.89 GHz is used for the DSRC band) and not the centre frequencies of individual channels. The radio interfaces allocated to a *Connection Manager* will receive signals from other radios registered in the same *Connection Manager* despite the channel they operate in if they are within the maximal interference distance of each other. It is the responsibility of the receiving *Decider* module to filter out all signals but the ones that are being received on the channel that the physical layer is tuned in.

However, it was identified during this research that the *Decider* module in the *Veins* framework does not correctly filter the received signals as required. It takes into account the power associated with signals in the entire spectrum band without discarding the signals that fall outside the channel the physical layer is tuned into when detecting energy in the operating channel. This falsely indicates the operating channel is busy when it is in fact idle. Therefore, in this research, the *Decider* module was further developed to accurately filter the signals received on the operating channel (details are provided in Section 4.3.3.3).

Table 4.1: The DSRC Channels

Channel	Centre Frequency (GHz)	Channel Classification
172	5.86	SCH1
174	5.87	SCH2
176	5.88	SCH3
178	5.89	CCH
180	5.90	SCH4
182	5.91	SCH5
184	5.92	SCH6

4.3.3.3 Signal Reception

The reception of an *Air Frame* is a complicated process when compared to its transmission. The received *Air Frames* are initially handled by the *Base Physical Layer* and registered in a module called '*Channel Info*'. The *Channel Info* keeps track of received *Air Frames* and provides details of all *Air Frames* intersecting within a given time interval. These details are then used by the *Decider* to calculate the Signal to Noise plus Interference Ratio (SNIR) of an *Air Frame* being received, for which all other *Air Frames* on the wireless medium are interference [111]. The *Channel Info* holds received *Air Frames* regardless of the channel that the physical layer is tuned into at the time of reception of the signal. It must be noted that a radio receives all *Air Frames* transmitted by another radio if the two radios are registered under the same *Connection Manager* and are positioned within the maximal interference distance (4.1) of each other. An *Air*

Frame is discarded by *Channel Info* when all other time-intersecting *Air Frames* with the considered *Air Frame* are completely received. An example of receiving *Air Frames* by a radio over a period is shown in Figure 4.7.

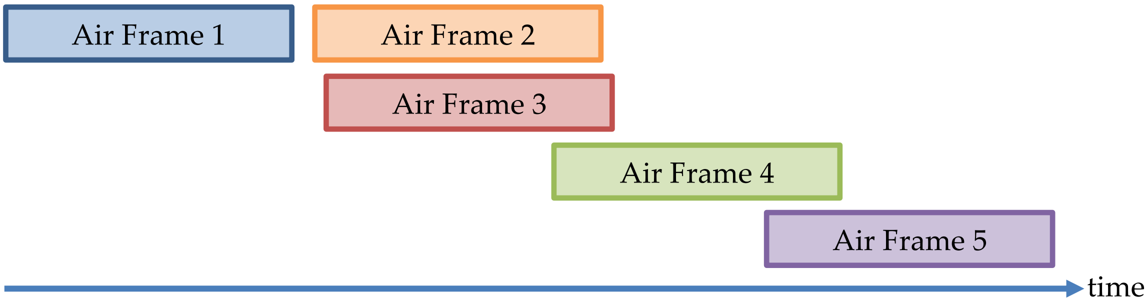


Figure 4.7: An example of receiving Air Frames by a radio over a period

Once a received signal is added to *Channel Info*, the *Base Physical Layer* passes the received *Air Frame* to the *Analogue Model* to calculate the attenuation of the signal. The attenuation of a signal is calculated by implementing path-loss, shadowing and fading models. Any number of *Analogue Models* can be applied to the signals received by the *Base Physical Layer*, and the final attenuation is calculated by adding up the attenuation of each analogue model. Next, the received signal is handed over to the *Decider* to determine whether the signal can be correctly received.

In this implementation, the signal is processed twice by the *Decider*; once when the reception starts and again when it ends. At the start of the signal reception, the *Decider* checks if the received signal has the same centre frequency as the channel that the physical layer is tuned into. Also, it analyses the power level of the signal at the start time to determine whether the signal is even strong enough to be received. If the signal is too weak or if the signal is on a channel different to the current physical layer channel, the channel is considered idle, and the signal will not be attempted to decode at the end. However, superposition of low power signals may make the channel busy, and this situation will be evaluated through the Clear Channel Assessment (CCA) process (see Appendix A.3.2).

In the *Veins* simulation framework, the CCA process senses the power associated with all signals in the wireless medium (in the spectrum band) at the time of assessment, without filtering out the frequencies that fall outside the channel that the physical layer is tuned into. For this reason, when signals are being received on other channels, the CCA will falsely indicate that the currently tuned channel is busy even if it is idle. Hence, in this research, the problem of not filtering out the signals being received on channels other than the tuned channel by the CCA process was rectified. During CCA, all the signals stored in *Channel Info* which have the same frequency as the physical layer frequency are selected, and their receiving power levels are summed up. Moreover, the thermal noise will also be added in if it is to be simulated. The resulting net power level (P_r) is compared against a chosen CCA threshold (P_γ) to make the decision D between H_1 or H_0, where the channel is idle under the null hypothesis $\mathcal{H}_0$ and busy under $\mathcal{H}_1$ (4.2).

$$D = \begin{cases} \mathcal{H}_0\colon\ P_r \leq P_\gamma \\ \mathcal{H}_1\colon\ P_r > P_\gamma \end{cases} \tag{4.2}$$

If the signal is strong enough to receive, it will be analysed again by the *Decider* at the end of the signal when the reception of the signal has been completed. At this point, the *Decider* calculates the Signal to Noise plus Interference Ratio (SNIR), as given in the equation (4.3), where S represents the power level of the signal analysed, N denotes the thermal noise floor, and I represents the sum of the power levels of all the other signals received on the same radio channel overlapping the examined signal. The value of the thermal noise is configured in the *OMNeT++ Initialization File*.

$$SNIR = \frac{S}{N + I} \tag{4.3}$$

From the SNIR function, the *Decider* then derives the Bit Error Rate (BER) and calculates the probability that the packet is received with any errors, P_{err}, according to the National Institute of Standards and Technology (NIST) error rate model presented in [112], in order to decide if the packet could be successfully received. Next, a random number in the range [0,1) is generated and compared against P_{err}. If the random number is larger than P_{err}, the packet is assumed to be successfully received, and it is sent to the MAC layer. Otherwise, it is reported to the MAC layer as an erroneous reception through a control message. At the end of the signal reception process, the *Decider* determines whether the channel is idle or busy through the CCA and sets channel status accordingly.

4.3.3.4 Radio Channel Switching

The physical layer tunes into different radio channels as governed by the MAC layer. In the developed simulation framework, the physical layer accepts '*change PHY channel*' requests sent by the MAC layer to switch to a specified radio channel. The process of handling a '*change PHY channel*' request by the physical layer is shown in Figure 4.8. When a '*change PHY channel*' request is received to switch to another radio channel, the physical layer first checks whether it is currently transmitting a signal since it cannot tune into another channel while there is an ongoing transmitting activity. If there is no such activity, the physical layer immediately switches to the requested radio channel. However, if the physical layer was decoding a received signal on the previous channel before the switch, the frame is marked as non-received since it should not be processed any more. The physical layer then performs a clear channel assessment on the new channel to determine whether the channel is idle or busy and sets the channel status accordingly. Next, it sends a '*channel idle*' message to the MAC layer if it is idle or a '*channel busy*' message if it is busy.

On the other hand, if there is an ongoing transmitting activity, channel switching will be performed when the transmission is completed. Therefore, the physical layer sets the value of the '*pending channel switch*' variable to '1' to indicate that radio switching needs to be carried out at the end of the transmitting activity. Furthermore, it records the radio channel to be switched to in the '*next radio channel*' variable. When the ongoing transmitting activity is completed, the physical layer evaluates the value of the '*pending channel switch*' variable and if it is '1', it switches to the radio channel indicated by the '*next radio channel*' variable.

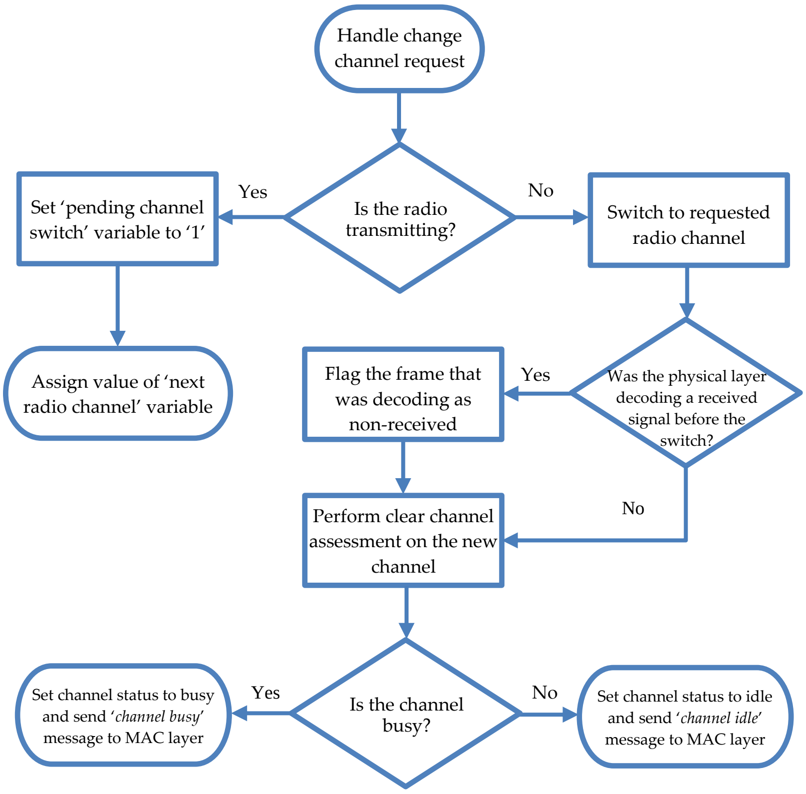


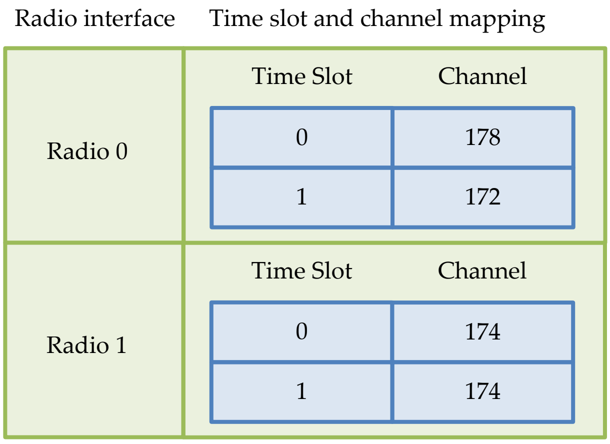
Figure 4.8: The physical layer handling channel switching requests

4.3.4 WAVE Application Service Management Model

The developed simulation framework allows application services to access several channels simultaneously on different radio interfaces and exchange data among near-by nodes on channels agreed upon. When there are multiple radios available on a node, application data need to be routed through channels on different radios during the exact time slot specified by the application service. The WAVE 1609.3 module was designed to handle various coordination tasks and manage application services on a node (for example, keeping track of application services and their radio channel utilisation information). It also accepts data from the application layer for transmission using the WAVE Short Message Protocol (WSMP) and delivers received WSM data from the MAC 1609.4 layer up to the application layer.

The application module is connected to the WAVE 1609.3 module through two input and two output gates, wherein one set of input and output gates is used for exchanging data messages between the modules and the other for control messages. These are single gates which support a single connection between two gates. On the other hand, the WAVE 1609.3 module is interfaced with several radio modules, as shown in Figure 4.2. Hence, gate vectors that support multiple connections were employed in the WAVE 1609.3 module to connect with the single gates in the radio modules (see Appendix A.1). The total number of radio modules available on a node can be configured in the 'Initialization File' before the start of a simulation, and the WAVE 1609.3 module forms the gate vectors of size equal to the number specified.

The MAC layer can be configured to switch to different radio channels in each time slot during initialisation of the MAC module (these parameters can be assigned in the Initialization File). The WAVE 1609.3 module queries MAC sub-modules of each connected radio and creates a mapping ('*radio interface to time slot and channel*' mapping) to keep track of the connected radio interfaces and their initial radio channel assignments (the channels that the MAC layer accesses during each time slot). An example '*radio interface to time slot and channel*' mapping of a dual radio node is shown in Figure 4.9. This mapping is considered as the node's current '*radio interface to time slot and channel*' mapping since it will be updated each time an application service is started or changed (see Appendix A.3.3).



Radio interface	Time slot and channel mapping	
	Time Slot	Channel
Radio 0	0	178
	1	172
	Time Slot	Channel
Radio 1	0	174
	1	174

Figure 4.9: An example 'radio interface to time slot and channel' mapping

As per the IEEE standard 1609.3, two WAVE device roles are implemented on this module. A node offering application services on a service channel is considered as a service provider (SP), whereas a node interested in obtaining services is regarded as a service user (SU). The SPs indicate their availability for data exchange through the broadcasting of WAVE Service Advertisements (WSAs), while SUs monitor for received

WSAs with the potential of participating in data exchange. The WSAs are usually broadcast on a predefined channel such as the CCH in order for SPs to rendezvous (defined in Section 2.2) with SUs. The application services are offered on SCHs, and the application module seeks access to these SCHs to start an application service or to partake in an ongoing data transmission. The application module sends several types of service request messages via the control connections to the WAVE 1609.3 module in order to gain necessary access to channels. These service request messages are, namely,

- provider service request
- user service request
- WAVE short message request

The provider service requests are sent by the application layer of an SP to indicate the WAVE 1609.3 module that the application layer intends to have WSAs generated on its behalf and channel access provided. On the other hand, user service requests are made by the application layer of an SU to indicate the WAVE 1609.3 module that it is interested in participating in application services and obtain access to radio channels when these services become available.

4.3.4.1 The Service Provider Model

The service provider model was developed within the WAVE 1609.3 module to offer application services on specified radio channels (usually SCHs). In order to obtain access to the SCH required, the application module first sends a provider service request message to the WAVE 1609.3 module. The WAVE 1609.3 module accepts these messages from the application module to start an application service on the specified SCH. The service requests cause the WAVE 1609.3 module to allow access to the specified SCH and generate WSAs. The provider service requests may also be used to stop or change an existing WSA transmission. The format of a provider service request message is shown in Figure 4.10, and the details of the parameters are presented in Table 4.2. These parameters must be configured in the ‘Initialization Files’ before the start of a simulation run.

action	Provider service identifier	Advertiser identifier	Service channel	Channel access	Service radio interface	WSA channel	WSA time slot	WSA radio interface	Repeat rate

Figure 4.10: Format of the provider service request

Table 4.2: Provider service request parameters

Field	Description
Action	Start, change or stop a WSA transmission. Values: 1 (start), 2 (change), 3 (stop)
Provider service identifier	A number to identify the application service
Advertiser identifier	An identifier (string value) to be sent in the WSA to recognise the provider of the service
Service channel	The channel on which the application service is operated
Channel access	Time slot during which the service is provided. Values: 0 (time slot 0), 1 (time slot 1), 2 (time slots 1 and 2)
Service radio interface	Radio interface to be used for the service
WSA channel	Channel on which the WSAs are broadcast. By default, this is set to CCH
WSA time slot	Time slot in which the WSA broadcast takes place. Values: 0 (time slot 0), 1 (time slot 1)
WSA radio interface	Radio interface to be used for the WSA broadcast
Repeat rate	The number of WSAs to be repeatedly broadcast per second

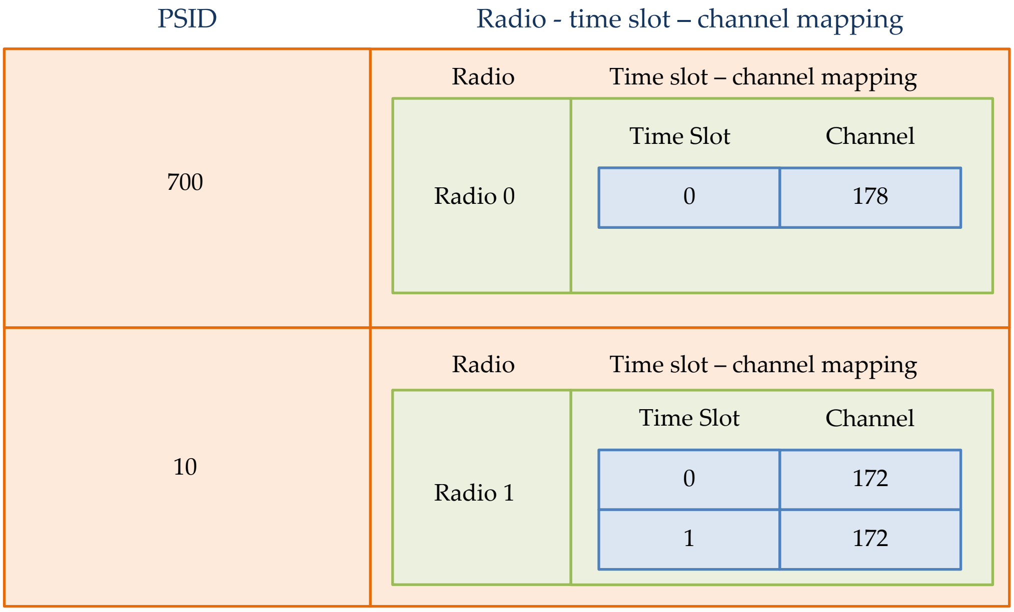
4.3.4.1.1 Starting an Application Service

Upon reception of a provider service request with value of the ‘action’ field set to ‘start’, the WAVE 1609.3 module stores the provider service identifier (PSID) of the application service and the requested channel assignment information such as the service channel number, channel access and the radio interface in a mapping (‘*PSID to radio time slot channel*’ mapping). The WAVE 1609.3 module considers WSA generation also as a service and allocates a PSID number in the format of ‘7XX’ (700-799) to record WSA transmission information. The PSID numbers allocated for WSAs start from 700 and with each provider service request (with the value of the ‘action’ field set to ‘start’), the

PSID number is incremented by one. For example, a provider may wish to offer an application service that delivers weather information (having the PSID value 10). The service is offered through 'radio 1' on SCH 172 during both time slots, and it is advertised through 'radio 0' on CCH. When a provider service request is received with the information provided in Table 4.3, the '*PSID to radio time slot channel*' mapping shown in Figure 4.11 will be formed. Each time a provider service request is received, the PSID number related to the WAVE service advertisement will be incremented by one and recorded in the mapping.

Table 4.3: An example provider service request

Field	Value
Action	Start
Provider service identifier	10
Advertiser identifier	uoa
Service channel	172
Channel access	2
Service radio interface	1
WSA channel	178
WSA time slot	0
WSA radio interface	0
Repeat rate	10



PSID	Radio - time slot – channel mapping		
	Radio	Time slot – channel mapping	
		Time Slot	Channel
700	Radio 0	0	178
	Radio	Time slot – channel mapping	
		Time Slot	Channel
10	Radio 1	0	172
		1	172

Figure 4.11: An example application (PSID) to service information (radio-time slot-channel) mapping

It should be noted that it is the responsibility of the application layer to request appropriate channel access for its application service. Hence, there are a set of rules to be followed when such a request is made.

1. Applications should avoid using the same channel on different radio interfaces during a time slot in order to mitigate self-interference.
2. Different channels on a radio interface shall not be employed by various applications during the same time slot, as it is physically impossible for a radio interface to switch into multiple channels at the same time. However, different applications may employ the same channel on a radio interface in the same time interval.

When a provider service request is received, the WAVE 1609.3 module checks whether the radio channels requested for an application service (i.e. '*PSID to radio time slot channel*' mapping) complies with the channel allocation rules given above and generates an error if any of the conditions are violated. Next, these channel allocations will be compared against the node's current '*radio interface to time slot and channel*' mapping shown in Figure 4.9 (which were configured during the initialisation stages), and if no conflicts are found, it will be updated with the new application channel assignments.

However, in the example given, when the two mappings (‘*PSID to radio time slot channel*’ and ‘*radio interface to time slot and channel*’) shown in Figure 4.9 and Figure 4.11 are compared, it can be noticed that both radios have been assigned to tune into channel 172 during time slot 1. In such situations, the WAVE 1609.3 module will alter the initially assigned channel to one of the other nonconflicting DSRC channel. Essentially, the ‘*PSID to radio-time slot-channel*’ mapping holds channel switching information pertinent to active application services, while the ‘*radio interface to time slot and channel*’ mapping keeps a record of a node’s all current channel switching details. Hence, in this case, the channel 172 allocated to radio 0 during time slot 1 (in the ‘*radio interface to time slot and channel*’ mapping) will be changed to 176. Based on the values stored in the channel allocation mapping, the physical layer channel switching will be performed in two steps. Initially, a control message (‘*change MAC channel*’ request) will be sent to the MAC layer in each radio interface (listed in the ‘*radio interface to time slot and channel*’ mapping) indicating the radio channel allocations for the time slots 0 and 1 (format shown in Figure 4.12). Based on PSID to radio-time slot-channel mapping, the ‘*change MAC channel*’ request will also indicate whether an application requires continuous channel access (if an application needs continuous access, the value of the ‘alternating channel access’ field is set to ‘0’; otherwise it is set to ‘1’.) Subsequently, the MAC layer performs physical layer channel switching by sending a ‘*change PHY channel*’ request to the physical layer.

Radio interface	Time slot 0 channel	Time slot 1 channel	Alternating channel access

Figure 4.12: Format of the change MAC channel request

In the example given, two change MAC channel requests will be sent to the MAC layers in the two radios. The request sent to ‘radio 0’ (with the value of the ‘alternating channel access’ field set to 1) will have channel assignments 178 and 176 during time slot 0 and 1, respectively. On the other hand, ‘radio 1’ will receive a change MAC channel request with channel 172 allocated for both time slots and the value of the ‘alternating channel access’ field set to 0. The MAC layers retain received channel allocations under the variables, ‘*next TS0 channel*’ and ‘*next TS1 channel*’, and performs necessary channel switching as described in Section 4.3.5 under the MAC layer model. The message flows related to the provider service request given in the example are illustrated in Figure 4.13.

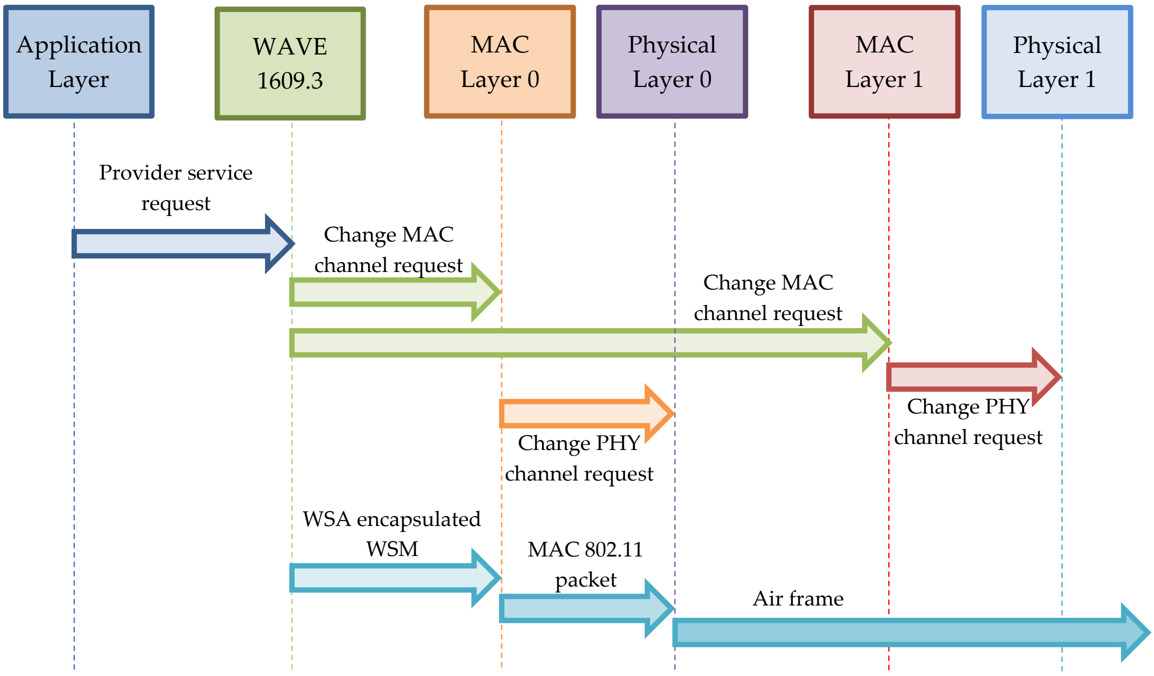
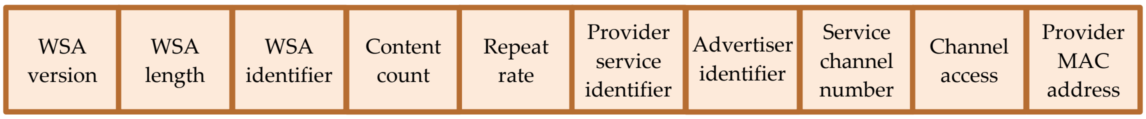


Figure 4.13: An example of message flow for a provider service request

Once the necessary radio channel access is granted, the WAVE 1609.3 module generates a WSA of the format shown in Figure 4.14 based on the details given in the provider service request. The descriptions of the parameters are presented in Table 4.4.



WSA version	WSA length	WSA identifier	Content count	Repeat rate	Provider service identifier	Advertiser identifier	Service channel number	Channel access	Provider MAC address

Figure 4.14: Format of the WAVE service advertisement

Table 4.4: WAVE service advertisement parameters

Field	Description
WSA version	The value of the WSA version shall be 3 according to the IEEE Std. 1609.3-2016 [86]
WSA Length	WSA packet length in bits
WSA identifier	Identifies a unique WSA
Content count	Used by the recipient to determine whether a WSA is a repeat of the previous one from the same source. Value of content count is incremented by one, each time the content of WSA is changed.
Repeat rate	Obtained from the provider service request.
Provider service identifier	Obtained from the provider service request.
Advertiser identifier	Obtained from the provider service request.
Service channel number	Obtained from the provider service request.
Channel access	Obtained from the provider service request.
Provider MAC address	MAC address of the radio interface on which the service is provided. The format of the MAC address is "xx:xx:xx:xx:xx:xx".

Next, the generated WSA is encapsulated in a WSM (as shown in Figure 4.15). When encapsulating a WSA in a WSM, a WAVE short message protocol (WSMP) header is attached to the WSA, and the length of the WSA packet is added to the length of the WSMP header. The provider service id (PSID) is used to identify the type of data included in the WSM, and in the case of WSAs, the PSID value is set to 7. The format of the WSMP header used in the simulations is shown in Figure 4.16, and the details of each parameter are presented in Table 4.5.

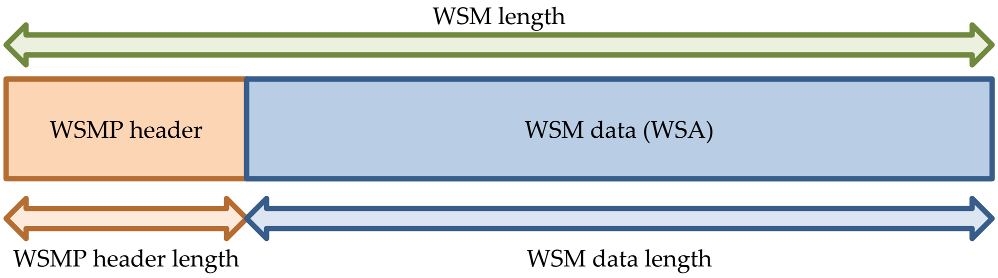


Figure 4.15: The WAVE short message

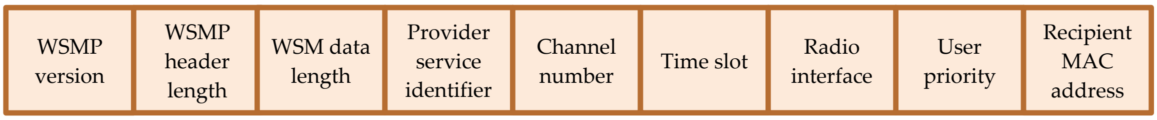


Figure 4.16: Format of the WAVE short message protocol header

Table 4.5: WAVE short message protocol header parameters

Field	Description
WSMP version	The value of the WSM protocol version shall be 3 according to the IEEE Std. 1609.3-2016 [86]
WSMP header length	WSM header length in bits
WSM data length	Values for WSM data length: 1 to *maximum WSM length* minus *WSM header length*. *Maximum WSM length* is 18,416 bits (2302 bytes). Default *WSM length* is 11,200 bits (1400 bytes). When a WSA is sent within WSM data, *WSM data length* value indicates the length of the WSA.
Channel number	The channel on which the WSM is sent. Extracted from the provider service request (*WSA channel*).
Time slot	Time slot in which the WSM transmission is to occur. Extracted from the provider service request (*WSA time slot*).
Access category	Access category. Values range from 0-3. This is set to 3 for WSA messages, by default.
Provider service id (PSID)	A unique number to identify the service. The PSID value used to identify WSA messages in the simulations is 7.
Recipient MAC address	MAC address of the recipient. This is set to broadcast address "ff:ff:ff:ff:ff:ff" for WSA messages.

Next, the generated WSM is passed down to the MAC layer to be broadcast on the required channel. Furthermore, the WSAs (encapsulated in WSMs) will be scheduled to recurrently broadcast at the repeat rate indicated in the provider service request. Each WSA will have the same WSA identifier and the content count number to indicate that it is a repeat broadcast of the original WSA. Figure 4.17 shows the service provider role of the WAVE 1609.3 module.

The SUs will be on the lookout for received WSAs with the potential of partaking in data exchange. The service user role of this process is explained in Section 4.3.4.2.

Figure 4.17: The service provider role of WAVE 1609.3 module

4.3.4.1.2 Changing and Stopping an Application Service

The WAVE 1609.3 module can make changes to the contents of an ongoing WSA transmission if required. When a provider service request is received with the 'action' field set to 'change', along with the other parameters of an existing application service (such as the service channel, channel access or radio interface values) altered, the corresponding '*PSID to radio time slot channel*' mapping will be updated with the new channel assignments. Next, the new channel assignments are checked against the channel allocations of other application services for any conflicts and an error is generated if there are any violations (as discussed in Section 4.3.4.1.1). The new channel allocations will also be compared against the node's current '*radio interface to time slot and channel*' mapping, and if no conflicts are found, it will be updated with the new values. However, if there is a conflicting channel allocation, the WAVE 1609.3 module will change the node's currently assigned channel to another nonconflicting channel.

Next, the WAVE 1609.3 module sends a set of '*change MAC channel*' requests to the MAC layers on connected radio modules indicating the new time slot and radio channel assignments. Subsequently, to inform the SUs about the changes made to the existing application service, the WSA parameters are updated to the new values along with the current value of 'content count', which is incremented by one. However, before generating the WSAs with the new 'content count' value, the currently scheduled WSA transmission is cancelled.

The WAVE 1609.3 module was also designed to end an existing application service. Upon receiving a provider service request with 'action' field set to 'stop', the corresponding application-service information is removed from the '*PSID to radio time slot channel*' mapping, and the scheduled WSA transmissions of the related application service are cancelled. However, no modification will be performed to a node's current '*radio to time slot and channel*' mapping information. The radio channels will tune into the same channels that were assigned previously (while the application service was active) until another application service starts or make changes to an existing service.

4.3.4.2 The Service User Model

The service user model within the WAVE 1609.3 module was developed to enable nodes to partake in application services offered by the service providers. The application layer lets the WAVE 1609.3 know of its intention to participate in an application service (as indicated by the provider service identifier and the advertiser identifier) through a control message (‘user service request’) and seeks access to necessary radio channels. The user service request has the message format shown in Figure 4.18, and the description of each parameter is listed in Table 4.6.

Figure 4.18: Format of the user service request

Table 4.6: User service request parameters

Field	Description
Action	Add or remove interest in participating in an application service. Values: 1 (add), 2 (remove)
Provider service identifier (PSID)	The application service the user is interested in
Advertiser identifier	The Advertiser Identifier string sent in the WSA to identify the provider of the service
WSA channel	The radio channel to tune into to receive WSA broadcast. By default, this is set to CCH
WSA time slot	The time slot during which the WSA broadcast will receive. Values: 0 (time slot 0), 1 (time slot 1). The default time slot is 0.
WSA radio interface	Radio interface to be used to monitor for received WSAs
Service radio interface	Radio interface to be used for the application service

A user service request specifies the radio interface and the channel to tune into during a particular time slot to receive WAVE service advertisements and the radio interface to be employed for the application service (the channel and the time slot/s on which the service is available are obtained from WSAs). Upon reception of a user service request with the 'action' field set to 'add', the WAVE 1609.3 module populates the '*PSID to radio time slot channel*' mapping with WSA reception channel, time slot and radio interface, in a similar manner as it does when a provider service request is received. It allocates a PSID number in the format of '7XX' (700-799) to record WSA reception service and allocates the number 700 for the first user service request, and with each similar request, the PSID number is incremented by one. The WAVE 1609.3 module evaluates if the radio channel allocated for WSA reception violates the channel assignment conditions set out in Section 4.3.4.1.1 and generates an error if it detects any noncomplying radio channel assignments.

If no errors are found, it records the provider service identifier and the advertiser identifier of the anticipated service in the '*PSID to advertiser id*' mapping, and the provider service identifier and the service radio interface in the '*PSID to service radio interface*' mapping. Subsequently, it updates the node's current '*radio to time slot and channel*' mapping and sends '*change MAC channel*' requests to the MAC layers in each connected radio interface. The MAC layers will then send '*change PHY channel*' requests to the physical layers to tune into necessary radio channels and receive WSAs broadcast by service providers. The parameters of the user service request sent by the application layer to participate in the example application service provided in Section 4.3.4.1.1 is given in Table 4.7. The process of handling a user service request is illustrated in Figure 4.19.

Table 4.7: An example user service request

Field	Value
Action	1
Provider service identifier	10
Advertiser identifier	uoa
WSA channel	178
WSA time slot	0
WSA radio interface	0

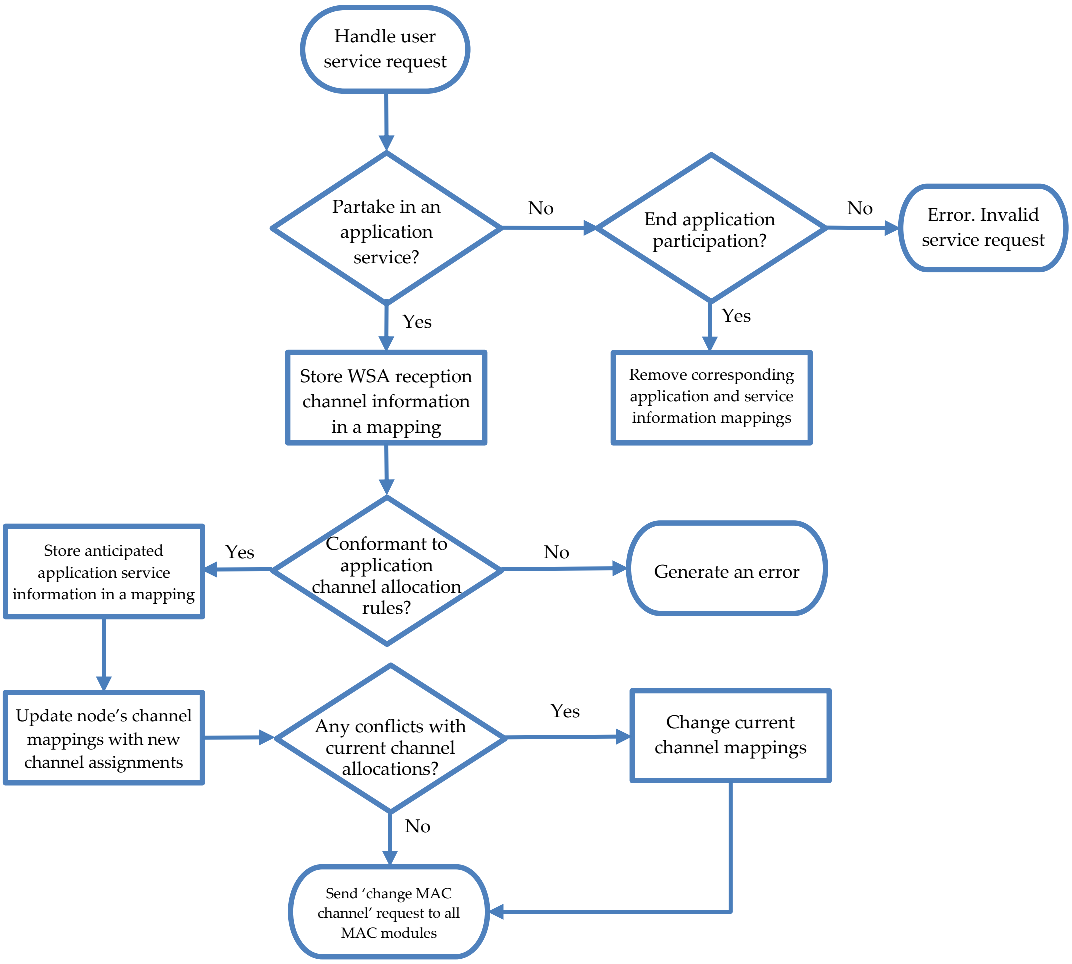


Figure 4.19: The process of handling a user service request

4.3.4.2.1 *Receiving WAVE Service Advertisements*

A service user monitoring the 'WSA channel' receives *Air Frames* comprised of WSA messages, which are then sent up to the WAVE 1609.3 module. The PSID value ('7') stored in a WSM is used to distinguish a WSA encapsulated WSM from a general WSM, and upon receiving a WSM possessing a WSA, the WSMP header is removed (decapsulated) to obtain the WSA. Then, the provider service identifier and the advertiser identifier of a received WSA are compared against the expected values stored in the '*PSID to advertiser id*' mapping, and if the PSID or the advertiser identifier is different from the one in the mapping, the WSA is discarded, as it is not related to an application service that the SU intends to take part in. However, if it is a valid WSA, the next step is to check whether the PSID of the received WSA is available in the '*PSID to WSA identifier and content count*' mapping (shown in Figure 4.20), and if it is not present, populate the mapping with the PSID, WSA identifier and the content count values. An already existing PSID value in the mapping means that the node has been receiving WSAs related to the PSID and the last received WSA may be a repeat of a previous or a WSA with its contents changed. The content count of a repeated WSA is the same as the one before, whereas it is incremented in an altered WSA. Subsequently, the content count value in the '*PSID to WSA identifier and content count*' mapping will be updated if it is a WSA with changed contents. A repeated WSA broadcast is discarded by the WAVE 1609.3 module, whereas a new or an altered WSA is further processed.

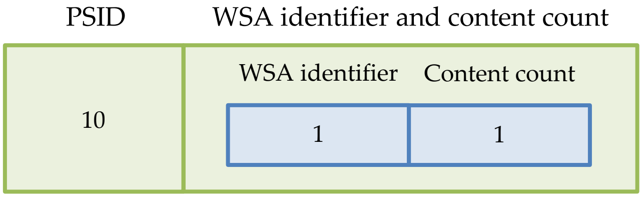


Figure 4.20: An example PSID to WSA identifier and content count mapping

The WAVE 1609.3 module obtains the radio interface to be used for the application service from the '*PSID to service radio interface*' mapping and adds radio channel and time slot information related to the application service in the '*PSID to radio time slot channel*' mapping. If channel information for a particular PSID already exists in the '*PSID to radio time slot channel*' mapping, it will be updated. It also examines if there are any radio channel assignments related to the advertised application service that violate the conditions laid out in Section 4.3.4.1.1 and creates an error if it identifies any noncomplying radio channel allocations. Next, the node's current '*radio to time slot and channel*' mapping is updated and '*change MAC channel*' requests are sent to the MAC layers in each connected radio interface to tune into the requested channels and participate in the application service. Finally, a control message ('WSA indicator') is sent up to the application layer to indicate that an SP is offering an application service that the SU is interested in and the required channel access has been granted. The WSA indicator has the message format in Figure 4.21. Upon receiving a WSA indicator, the application layer may initiate a WAVE short message exchange with the SP on the assigned service radio channel. The process of receiving a WSA is shown in Figure 4.22, while Figure 4.23 visualises the flow of messages taking place between different layers of an SU.

Provider service identifier	Service channel	Channel access	Service radio interface	Provider MAC address

Figure 4.21: Format of the WSA indicator

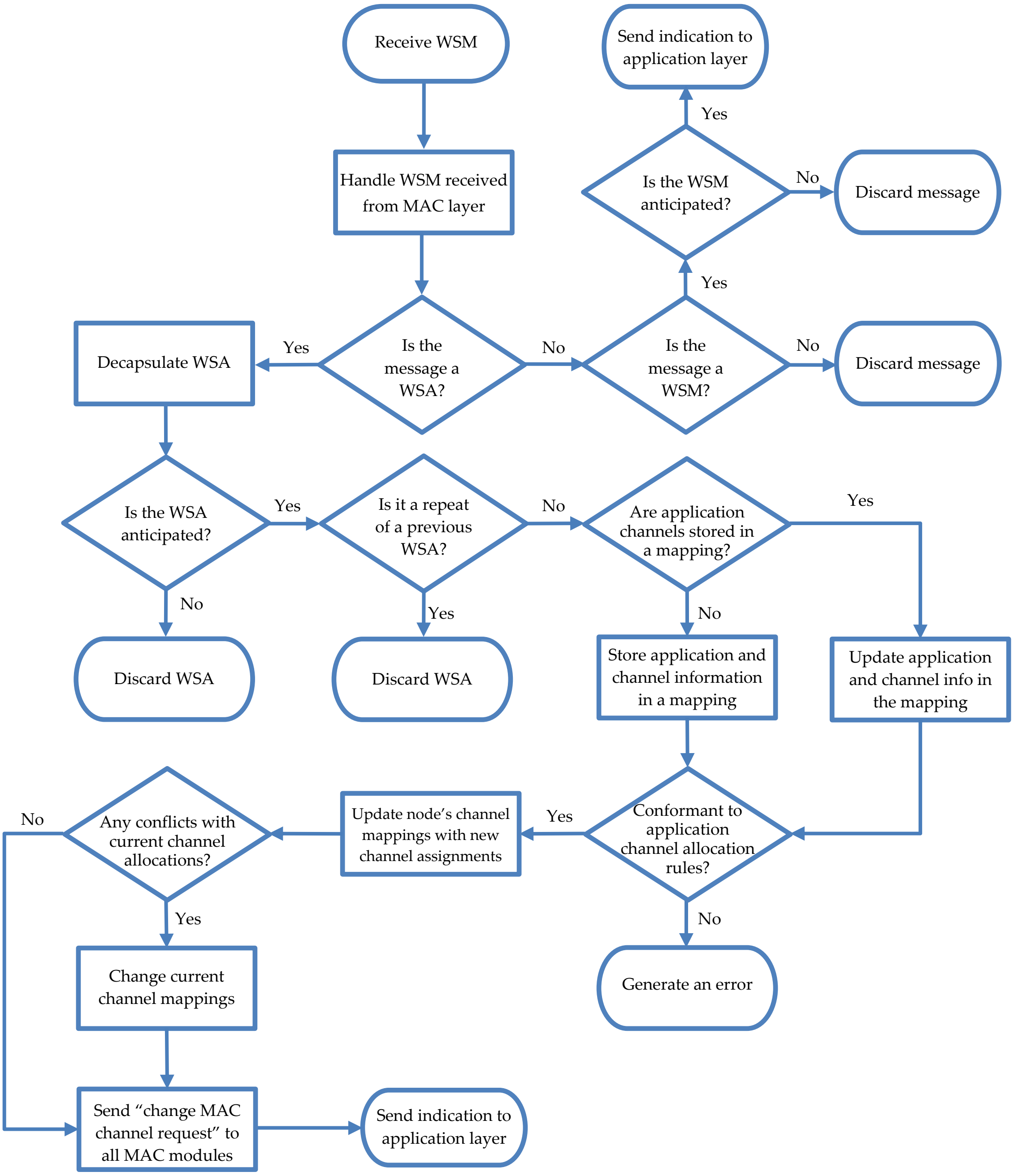


Figure 4.22: Receiving WAVE service advertisements

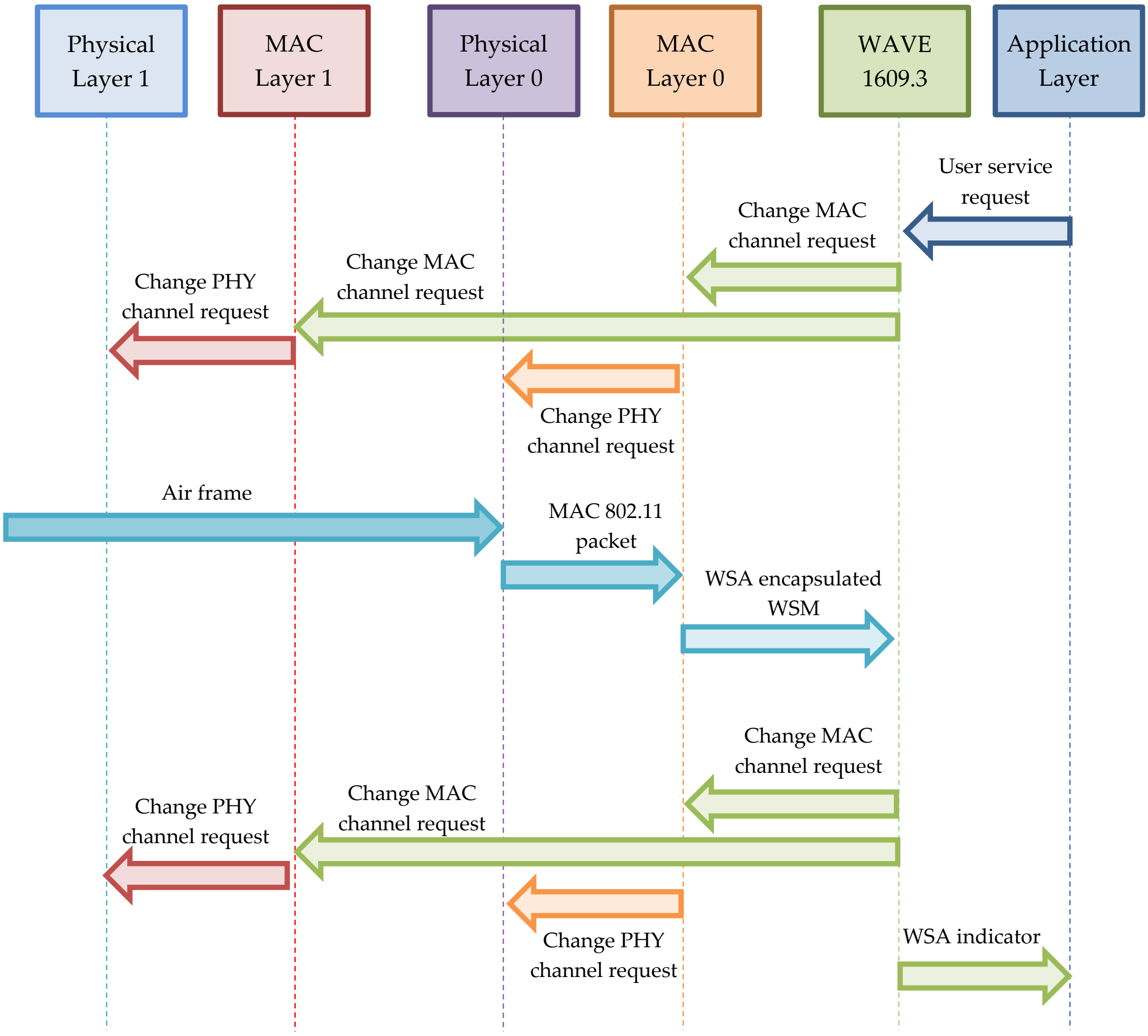


Figure 4.23: Example message flows between different layers of a service user

4.3.4.2.2 *Terminating Participation in an Application Service*

The user service requests were developed to terminate participation in an application service. These requests are sent with 'action' field set to 'remove' and upon receiving them the WAVE 1609.3 module removes the application service information from the mappings '*PSID to advertiser id*', '*PSID to service radio interface*' and '*PSID to radio time slot channel*'. Nevertheless, no changes shall be done to a node's current '*radio to time slot and channel*' mapping information.

4.3.4.3 Transmission and reception of WAVE short messages

The WAVE 1609.3 module was designed to send and receive WAVE short messages as requested by an application service running on the application layer. Radio channel access to send WSMs is initially granted through a provider service request or a user service request. The process of generating a WSM starts at the application layer where a ‘WSM request’ (Figure 4.24) is sent from the application layer to the WAVE 1609.3 module. Upon receiving a WSM request, the WAVE 1609.3 module inspects the ‘*PSID to radio time slot channel*’ mapping and obtains the channel assignment information related to the PSID in the WSM request. If the PSID value is not found in the mapping, an error will be generated. Otherwise, the module generates a WSM with data supplied in the WSM request and channel access information obtained from the ‘*PSID to radio time slot channel*’ mapping, and passes it down to the MAC layer to be sent to the ‘recipient MAC address’.

Provider service identifier	Recipient MAC address	User priority	WSM data length	WSM data

Figure 4.24: Format of the WSM request

The WAVE 1609.3 module of the recipient, upon receiving a WSM checks whether the PSID of the received message is available in the ‘*PSID to advertiser id*’ mapping to identify whether it is associated with an application service that the user is participating (as shown in Figure 4.22). If the PSID is not found in the ‘*PSID to advertiser id*’ mapping, the received WSM is discarded, whereas a valid WSM generates a ‘WSM indicator’ message (format shown in Figure 4.25) with data obtained from the received WSM. The WSM indicator is then sent up to the application layer.

Provider service identifier	Sender MAC address	WSM data

Figure 4.25: Format of the WSM indicator

4.3.5 MAC Layer Model

The IEEE1609.4 standard of the WAVE protocol is modelled by the MAC 1609.4 module of the simulation framework. The MAC 1609.4 module has initially been a part of the *Veins* framework, which incorporated radio channel switching models, as well as the QoS enabled CSMA/CA model. The *Veins* MAC 1609.4 module allows continuous CCH access and performs alternating channel switching between the CCH and an SCH defined at the start of a simulation run. However, it does not permit continuous access to any SCH or alternating switching between two SCHs. Moreover, when simulating alternating channel access, *Veins* lets CCH access only during the time slot 0 and an SCH access during time slot 1. In the *Veins* framework, the higher layers are unable to dynamically change access to channels and transmit WSMs on the desired channel during the intended time slot.

Hence, during this research, the MAC layer of the *Veins* simulation framework was extended to support alternating and continuous channel access schemes, which can be managed by the upper layers of the framework. In the developed framework, the alternating channel access scheme allows a radio to access any two channels at alternate time intervals. For example, a radio may repeatedly access the CCH during time slot 0 and the SCH 1 during time slot 1, as illustrated in Figure 4.26(a). It may even alternate between two SCHs, such as SCH 1 and SCH 2, as shown in Figure 4.26(b). As the name suggests, the continuous channel access mode provides uninterrupted access to the same channel without considering the time slot boundaries as in Figure 4.26(c).

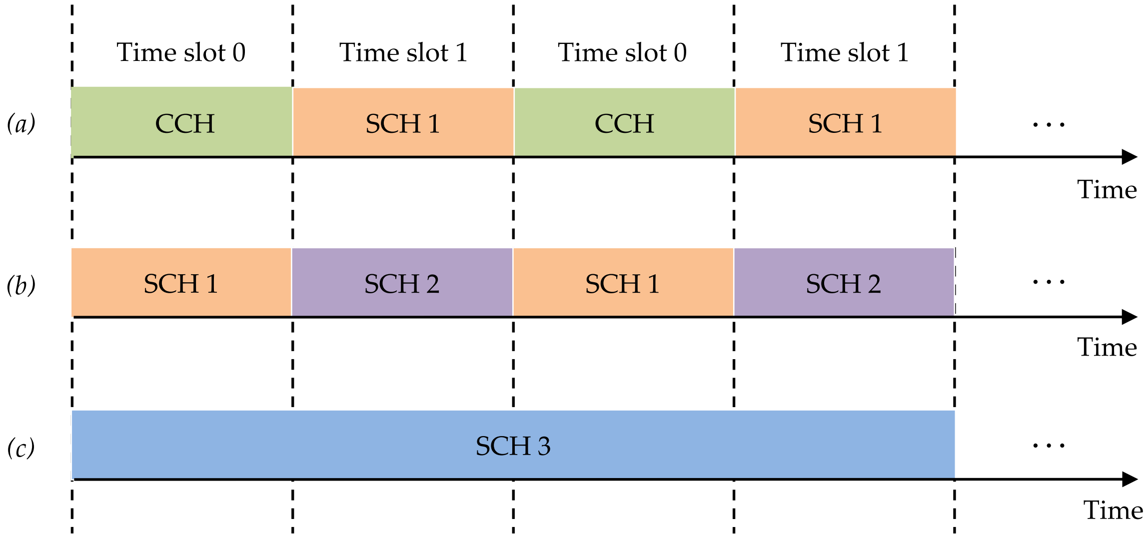


Figure 4.26: Channel access examples (a) alternating access between CCH and SCH1 (b) alternating access between SCH1 and SCH2 (c) continuous access

As described in Chapter 3 (Section 3.2.4), the *Veins* platform contains two Enhanced Distributed Channel Access (EDCA) packet queuing systems. One EDCA system (of type CCH) queues WSMs requiring access to the CCH during time slot 0, while the other (of type SCH) does the same for WSMs that need to be transmitted on an SCH (during time slot 1) defined at the start of a simulation run. However, a channel type based (CCH or SCH) queuing system prevents the MAC layer from transmitting packets that require alternating access between two SCHs (for example, between SCH1 and SCH2 during time slot 0 and 1, respectively).

Hence, in this research, to improve the flexibility of packet transmission, a ‘time slot’ based EDCA queuing system was introduced to replace the existing ‘channel type’ based system. The developed MAC 1609.4 module contains three ‘time slots’ based EDCA queuing systems (‘time slot 0’, ‘time slot 1’ and ‘continuous’), as shown in Figure 4.27. The WSMs arriving from the WAVE 1609.3 module are routed to the appropriate EDCA subsystem based on the transmission time slot information contained in the WSMP header. The WSMs with time slot data set to ‘0’ and ‘1’ are routed to ‘time slot 0’ and ‘time slot 1’ EDCA subsystems, respectively, while WSMs requiring continuous access to a channel (time slot set to ‘2’) are routed to the ‘continuous’ EDCA subsystem. Moreover, the MAC 1609.4 module only accepts WSMs relevant to the current channel access mode (i.e. if the module is operating in the alternating channel access mode, the WSMs with the time slot parameter set to ‘0’ or ‘1’ are accepted, while WSMs having the same parameter configured to ‘2’ are only accepted when the MAC 1609.4 module is in the continuous channel access mode).

Each EDCA subsystem contains four queues (see Appendix A.3.4), which prioritise the received WSMs according to the access category (AC). The EDCA function (EDCAF) in each of the queues determines when a WSM in a queue is allowed to be sent over the wireless medium. It provides different contention and transmission parameters for WSMs with different priorities. When the MAC 1609.4 module is operating in alternating channel access mode, WSMs from either ‘time slot 0’ or ‘time slot 1’ EDCA subsystem are chosen to transmit depending on the time slot the module is currently in (for example, WSMs queued in ‘time slot 0’ EDCA subsystem are selected for transmission during time slot 0). If the MAC 1609.4 module is accessing a channel continuously, WSMs are chosen from ‘continuous’ EDCA subsystem to transmit over the current operating channel. The functionality of an EDCA subsystem is covered in Section 3.2.4.

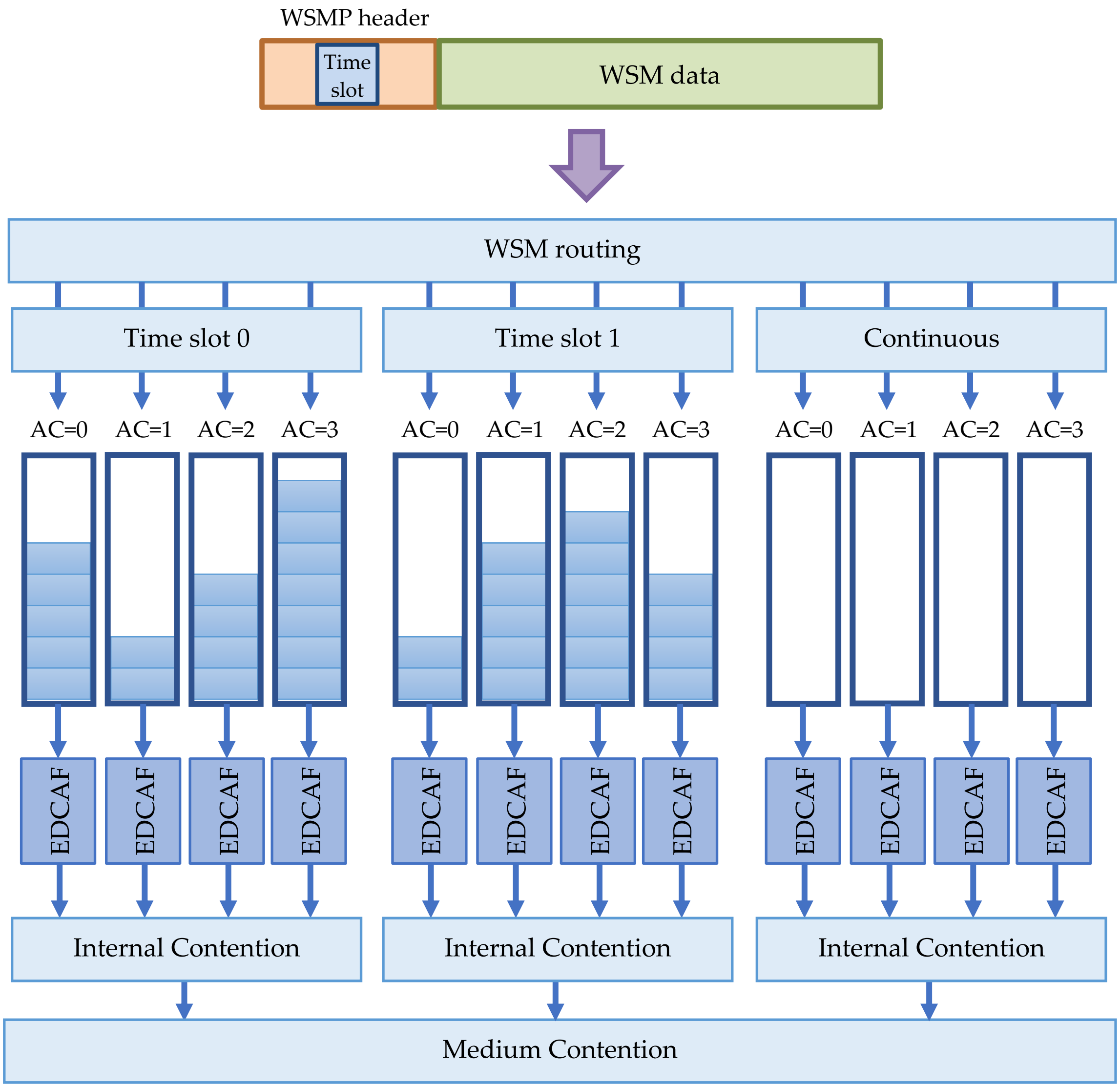


Figure 4.27: The Enhanced Distributed Channel Access (EDCA) implementation

The WSM that wins the contention is handed down to the physical layer to transmit over the wireless channel that the physical layer is currently tuned in. However, switching the physical layer to the required radio channel is the responsibility of the WAVE 1609.3 module. The MAC 1609.4 module stores the channel access information under several variables. While the variables '*current TS0 channel*' and '*current TS1 channel*' hold the channel numbers that are accessed during each time slot, the '*alternating channel access state*' variable identifies if the MAC layer is operating in the alternating channel access mode. The MAC 1609.4 module discards any WSM from the WAVE 1609.3 module that does not have the same channel access information as the MAC 1609.4 module (i.e. the 'channel number' and the 'time slot' parameters of the WSMP header must match the relevant '*current TS0 channel*' or '*current TS1 channel*' values). Hence, before passing down the WSMs, the WAVE 1609.3 module sends channel switching requests to the MAC 1609.4 module in the form of control messages to obtain necessary channel access. The MAC 1609.4 module accepts the channel switching requests ('*change MAC channel*' requests) sent by the WAVE 1609.3 module and provides access to radio channels during the required time slots as shown in Figure 4.28. The format of the '*change MAC channel*' request is illustrated in Figure 4.12.

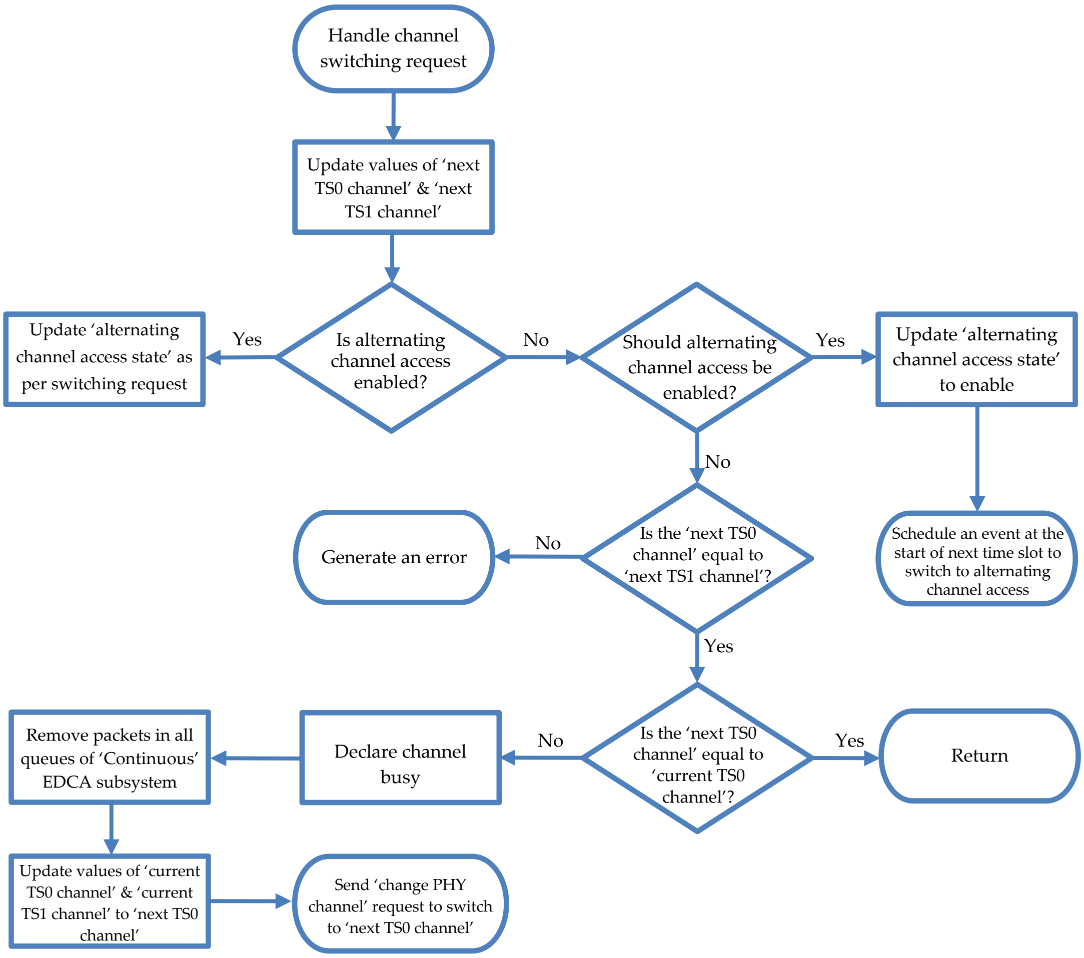


Figure 4.28: The MAC layer handling channel switching requests

When a '*change MAC channel*' request is received, the MAC 1609.4 module first obtains the numbers of the channels that it needs to access during the upcoming time slots from the request message and stores them in the variables '*next TS0 channel*' and '*next TS1* *channel*'. The MAC 1609.4 module allows a radio to switch to either alternating or continuous channel access modes as per the '*change MAC channel*' request. If the module was operating in the alternating channel access mode by the time the request message is received, the module updates the '*alternating channel access state*'. While the MAC 1609.4 module is in alternating channel access mode, an event is scheduled at the start of every time slot to perform alternating channel access, during which the values of '*alternating channel access state*', '*next TS0 channel*' and '*next TS1 channel*' are assessed. The process of handling alternating channel access at the start of each time slot is shown in Figure 4.29. If the MAC 1609.4 module was operating in the continuous channel access mode at the time '*change MAC channel*' request is received, the module handles the request in two ways. If alternating channel access mode is requested, the module updates the value of '*alternating channel access state*' to 'enable' and schedules an event at the start of the next time slot to perform alternating channel access.

On the other hand, if the '*change MAC channel*' request only requires the MAC 1609.4 module to switch channels while remaining in the continuous channel access mode, the module immediately proceeds with the actions required to alter the currently occupying channel. The MAC 1609.4 module first declares the channel busy while switching takes place, in order to suspend the back-off countdowns on 'continuous' EDCA subsystem queues. Next, the WSMs in all the queues of the 'continuous' EDCA subsystem are removed, since these WSMs are queued to be transmitted on the '*current TS0 channel*' (or '*current TS1 channel*'). Note that both '*current TS0 channel*' and '*current TS1 channel*' contain the same value when the MAC layer is operating in the continuous channel access mode. The values of '*current TS0 channel*' and '*current TS1 channel*' are then updated to the value of '*next TS0 channel*' (or '*next TS1 channel*') and a '*change PHY channel*' request is sent to the physical layer to change the operating radio channel to the '*next TS0 channel*'. Once the radio channel switching is completed, the physical layer will send a '*channel idle*' message back to the MAC layer if the switched channel is idle (as discussed in Section 4.3.3.4). The MAC layer declares the channel as idle when it receives a '*channel idle*' message to resume the back-off countdowns and start contention on the 'continuous' EDCA subsystem queues.

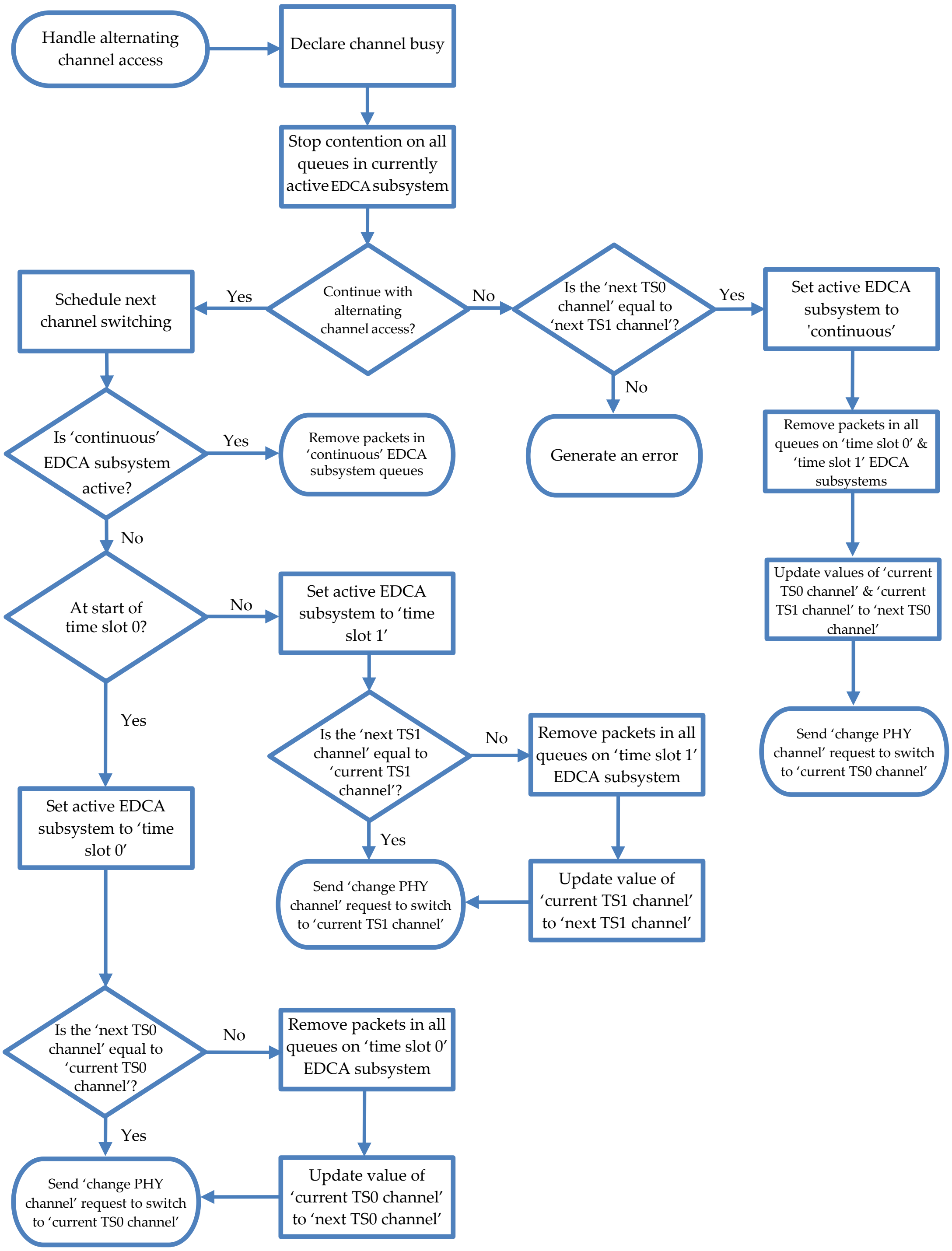


Figure 4.29: The MAC layer handling alternating channel access

If an event is scheduled at the start of a time slot to handle alternating channel access, the MAC 1609.4 module executes the process shown in Figure 4.29. An event to handle alternating channel access may be scheduled, when the module is in alternating channel access mode and wishes to continue in the same mode or switch to continuous channel access mode. It may also be scheduled when the module intends to change to alternating channel access mode while it is in continuous channel access mode. At the start of the alternating channel access process, the MAC 1609.4 module declares the channel busy to halt the back-off countdowns and stop contention on all the queues in the currently active EDCA subsystem.

Next, it decides whether to continue with alternating channel access mode by evaluating the value of '*alternating channel access state*'. If the value was set to 'enable', the module schedules an event to handle the alternating channel access process again at the start of the next time slot. Moreover, if the 'continuous' EDCA subsystem is currently active, the WSMs queued in this subsystem are removed since WSMs from the 'continuous' EDCA subsystem are not chosen to transmit when operating in the alternating channel access mode. Subsequently, either 'time slot 0' or 'time slot 1' EDCA subsystem is chosen as the active queuing system depending on the time slot the MAC 1609.4 module is currently in (WSMs are chosen from the active EDCA subsystem to transmit over the wireless medium).

The module then evaluates if a change of channel is required for the time slot the module is in by comparing the channel currently allocated for the time slot against the channel that it needs to be assigned with. In other words, the module compares the value of '*current TS0 channel*' against '*next TS0 channel*' at the start of time slot 0 and the value of '*current TS1 channel*' against '*next TS1 channel*' at the start of time slot 1. When a change in the alternating channel is required, the WSMs queued in the active EDCA subsystem are removed, as they were supposed to be transmitted on the channel previously associated with the current time slot. Furthermore, the channel linked to the current time slot is updated to the next channel. Afterwards, a '*change PHY channel*' request is sent to the physical layer to switch to the channel associated with the current time slot. The physical layer then completes channel switching and sends a '*channel idle*' message back to the MAC layer if the switched channel is idle. The IEEE standard 1609.4 defines a guard interval of 4ms during which the nodes using alternating channel access do not transmit [86]. Hence, in this simulation framework, if a '*channel idle*' message is received during the guard interval, the MAC 1609.4 module waits until the end of the guard interval and declares the channel idle to resume back-off countdown and start contention on the active EDCA subsystem queues.

While deciding on the channel access mode, if the value of '*alternating channel access state*' was set to 'disable', the 'continuous' EDCA subsystem is chosen as the active queuing system. Furthermore, the WSMs queued in both 'time slot 0' and 'time slot 1' EDCA subsystems are removed since WSMs from these queues are not chosen to transmit when operating in continuous channel access mode. Next, the values of the '*current TS0 channel*' and the '*current TS1 channel*' are updated to the '*next TS0 channel*' (or '*next TS1 channel*', as they are both equal in continuous channel access mode) and a '*change PHY channel*' request is sent to the physical layer to switch to the '*current TS0 channel*'. When a '*change PHY channel*' request is received, the physical layer performs channel switching and sends a '*channel idle*' message back to the MAC layer if the switched channel is idle.

4.3.6 Simulations

The example scenario provided in Section 4.3.4, wherein a dual radio node is operating as a service provider, was simulated using the developed framework. Initially, the node's first radio was set up to work in the alternating channel access mode while switching between the CCH (channel 178) and the SCH 1 (channel 172) during time slot 0 and time slot 1, respectively, as illustrated in Figure 4.30. Here, the radio's operating channel number is recorded twice at each time slot boundary, of which, once at the start of the alternating channel switching process and another at the end of the switching process. On the other hand, the node's second radio was set up to continuously access the SCH 2 (channel 174), as shown in Figure 4.31. The operating channel number of this radio was logged at every 50ms.

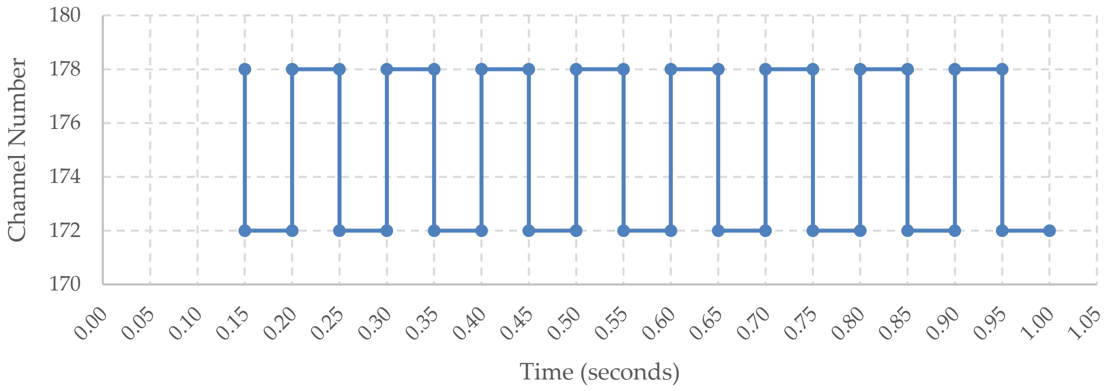


Figure 4.30: The radio operating in the alternating channel access mode

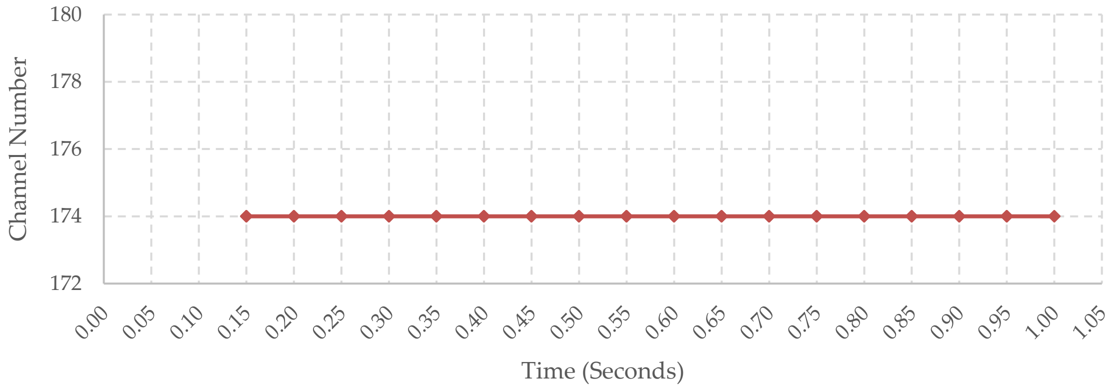


Figure 4.31: The radio operating in the continuous channel access mode

Subsequently, the application layer sends a provider service request to the WAVE 1609.3 module to gain access to SCH 1 (channel 172) on the second radio to start an application service. The WAVE 1609.3 module in return sends a '*change MAC channel request*' to the MAC layer of the second radio to change the channel to SCH 1. It also sends another '*change MAC channel request*' to the MAC layer of the first radio to request alternating channel access between the CCH and SCH 3 (channel 176) during time slot 0 and 1, to avoid switching to the same channel (SCH 1) during time slot 1. The simulation was performed for two different situations, where the provider service request was sent at different times. In the first instance, the request was sent during time slot 0 (at 1.33 seconds), while it was sent during time slot 1 (at 1.38 seconds) in the second situation. Figure 4.32 and Figure 4.34 show the channel switching that takes place on the second radio, which operates in the continuous channel access mode, at times 1.33 seconds and 1.38 seconds, respectively. It was observed here that the radio operating in the continuous channel access mode changed the channel at the same time it received the '*change MAC channel*' request. However, it can be observed in Figure 4.33 and Figure 4.35 that in the first radio, which operates in the alternating channel mode, the channel did not change immediately when its MAC layer received a 'change MAC channel' request but changed to SCH 3 at the start of the next time slot 1. The observed result was as expected since the channel switching algorithm changes the channel being accessed during time slot 1 only at the start of the same time slot.

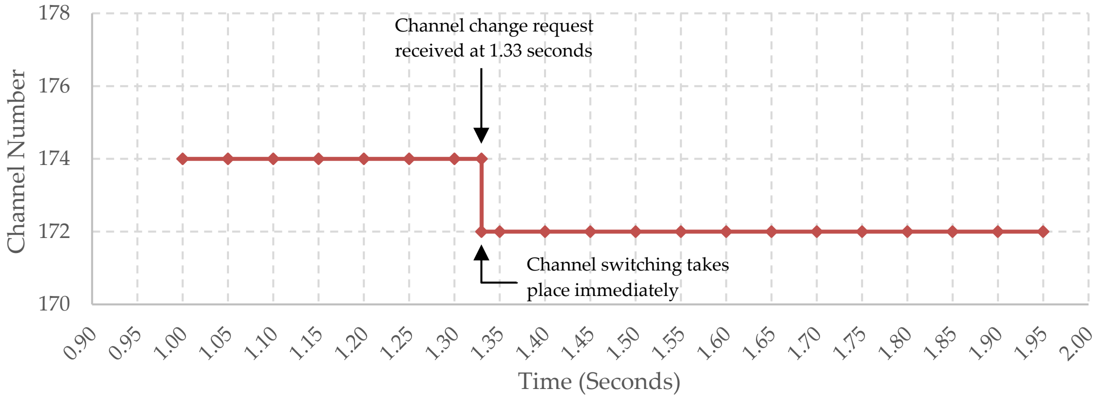


Figure 4.32: Changing channels in the continuous channel access mode after receiving a change channel request at 1.33 seconds

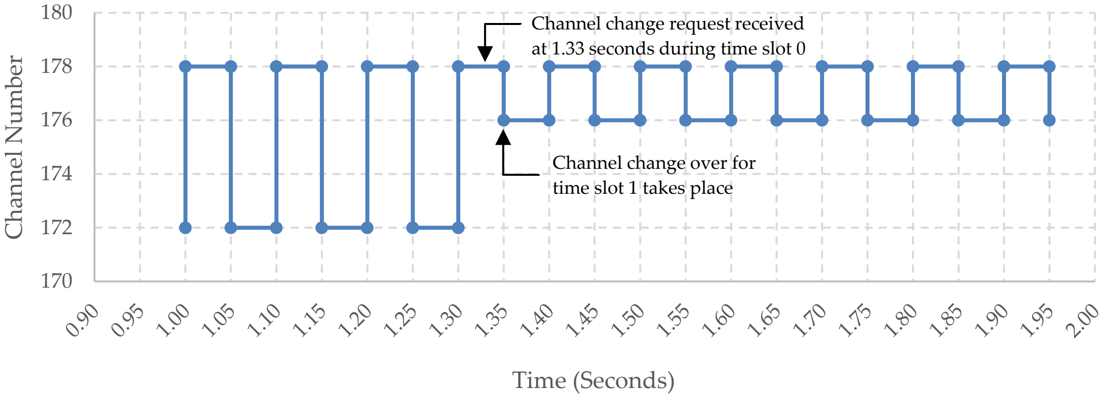


Figure 4.33: Changing channels in the alternating channel access mode after receiving a change channel request at 1.33 seconds

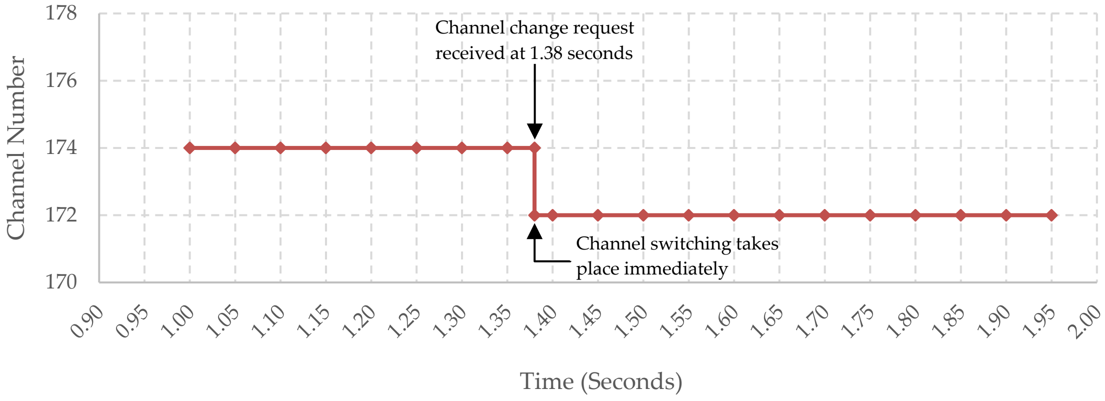


Figure 4.34: Changing channels in the continuous channel access mode after receiving a change channel request at 1.38 seconds

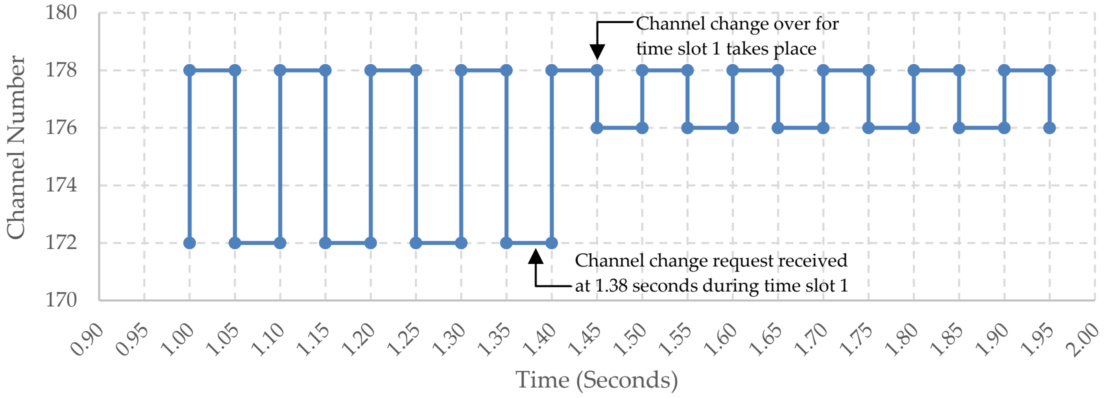


Figure 4.35: Changing channels in the alternating channel access mode after receiving a change channel request at 1.38 seconds

In the *Veins* simulation framework, the CCA process incorrectly senses the power associated with received signals and makes a channel busy even if it is idle. This issue was corrected in the developed simulation framework. In order to compare the operation of the two frameworks, a node operating in the alternating channel access mode was simulated, where it broadcast 20 WSMs on the CCH during time slot 0 intervals and 20 WSMs on the SCH1 during time slot 1 intervals (the payload length of a transmitted packet was 1400 bytes, and the data rate was 6 Mbps). A second node was tuned into CCH, and the channel utilisation (amount of time the channel was considered busy relative to the duration of the simulation) was observed for 10 seconds using the two frameworks. A third node was tuned into SCH1, and the channel utilisation was again observed for 10 seconds using only the developed simulation framework since the Veins framework cannot continuously access a channel other than the CCH. Figure 4.36 shows the channel utilisations observed by the two simulation frameworks. At 6 Mbps data rate, each packet takes approximately 1.9ms to broadcast and 20 packets takes around 38ms. Hence, ideally, a node tuned into either CCH or SCH1 should observe a busy channel for approximately 3.8 seconds during the 10 seconds simulation time (i.e. 38%). It can be seen in Figure 4.36 that the developed simulation framework correctly observes the channel utilisation, while the *Veins* framework is unable to do the same.

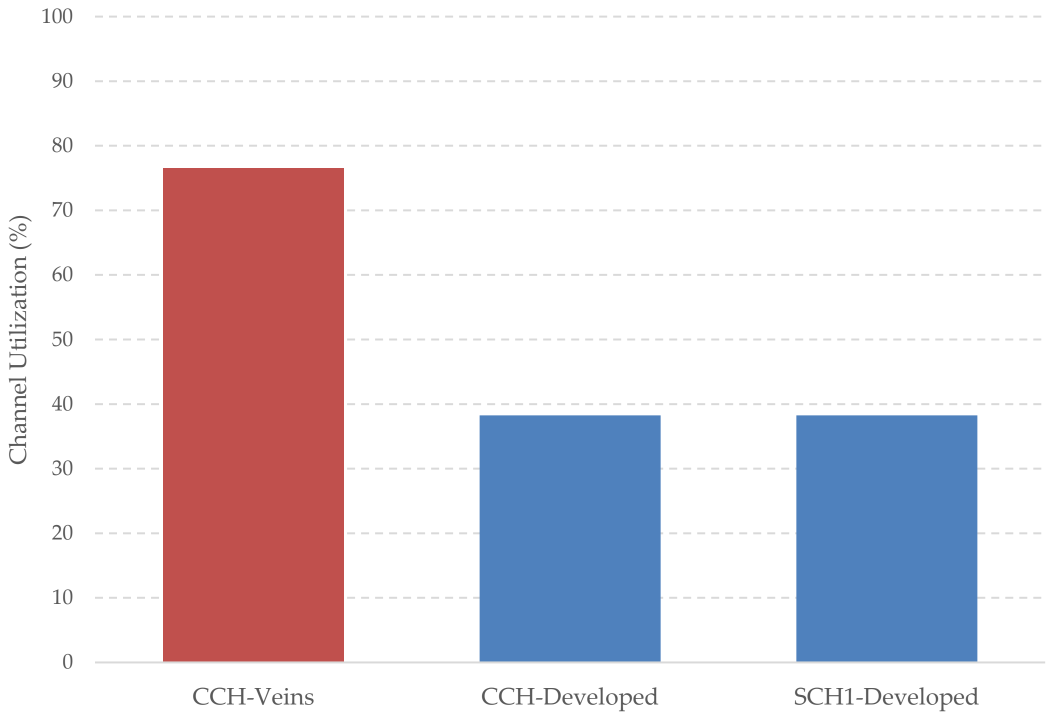


Figure 4.36: Observed radio channel utilisations by the developed framework and Veins

CHAPTER 5

A Simulation Framework for Cognitive Radio Enabled VANETs

5.1 Introduction

The modelling of cognitive radio (CR) networks remains a significant research challenge due to limited availability of CR network simulators. As reviewed in Chapter 2, the only openly available simulator (*crSimulator*) lacks several vital functionalities required to simulate CR enabled VANETs, such as node mobility, dynamic connection management, identifying primary user (PU) transmissions in a distributed network and spectrum hand-off mechanisms to allow seamless transitions between radio channels. Hence, during this research, a simulation framework for cognitive radio enabled VANETs was developed to perform the following functions.

- Simulate mobile (vehicles) and stationary CR nodes
- Simulate PU transmissions
- Simulate CR based application service advertisement and discovery
- Simulate spectrum sensing
- Identify primary and secondary user transmissions distinctively
- Simulate spectrum hand-off

5.2 Simulation Model

In ad hoc networking environments such as cognitive radio enabled VANETs (CR-VANETs), a proper mechanism is required to offer application services on channels in which dynamic spectrum access (DSA) is allowed, and also to advertise and discover them. The service management framework described in Chapter 4 laid the foundation for building a multi-radio multi-channel radio environment required for modelling CR-VANETs. This work was further extended to enable simulation scenarios that explore open spectrum bands for vehicular communications using cognitive functions. In this simulation model, the nodes in the CR-VANET employ a DSA allowed band such as the Television White Spaces (TVWS) band (54-862 MHz) for application data transmissions, while the channel 178 (CCH) in the DSRC spectrum is used for broadcasting announcements related to these services (in this chapter, the TVWS is used as the DSA allowed band). The TVWS channel on which the application services are offered is advertised through WAVE service advertisements (WSAs) (the details on WSAs are provided in Chapter 4).

Since two spectrum bands are employed in CR-VANETs, the nodes in this simulation framework need to be composed of at least two transceivers, of which one operates in the DSRC band while the other in the TVWS band. However, the TVWS channels used for CR communications need to be sensed frequently for primary user detection. Therefore, sensing and transmitting schedules need to be implemented if only two transceivers are utilised. However, the realisation of such schedules was beyond the scope of this research. Hence, in this simulation framework, an additional transceiver operating in the TVWS band ('TVWS radio 2') was employed primarily for incumbent detection, as shown in Figure 5.1. The 'TVWS radio 1' is used for application data transmission, while the DSRC radio is utilised for broadcasting service announcements.

It should be noted that all the transceivers in the CR-VANET are based on the IEEE standard 802.11p (incorporated in the IEEE standard 802.11-2012) and employ the CSMA/CA wireless channel access mechanism. While there exist several secondary user networks of different standards (IEEE802.22, IEEE802.11af, IEEE802.19.1, etc.), only the IEEE802.11 standard based secondary user networks were considered in this research. Hence, it was assumed that all transceivers in the secondary user networks are based on IEEE802.11 standard, and the IEEE802.11p transceivers on vehicles are able to decode the header of any IEEE802.11 signal.

The CR-MAC 1609.4 module in the model was created extending the MAC 1609.4 module described in Section 4.3.5 to carry out cognitive functions such as spectrum sensing. The PHY 802.11p module was also extended to support the CR tasks performed by the CR-MAC 1609.4 modules. The two WAVE service models, the service provider (SP) model and the service user (SU) model described in Section 4.3.4, are implemented in the WAVE 1609.3 module. This module was also extended to perform various coordination tasks required for CR operations.

In this setup, the primary users operate in the TVWS band while sharing their spectrum resources with the unlicensed secondary users. Moreover, M non-overlapping channels $\{C_i, i = 1, 2, .., M\}$ available for CR operations, having centre frequencies at $\{f_c^i\}_{i=1}^{M}$ and bandwidth of 10MHz, can be defined at the start of a simulation run.

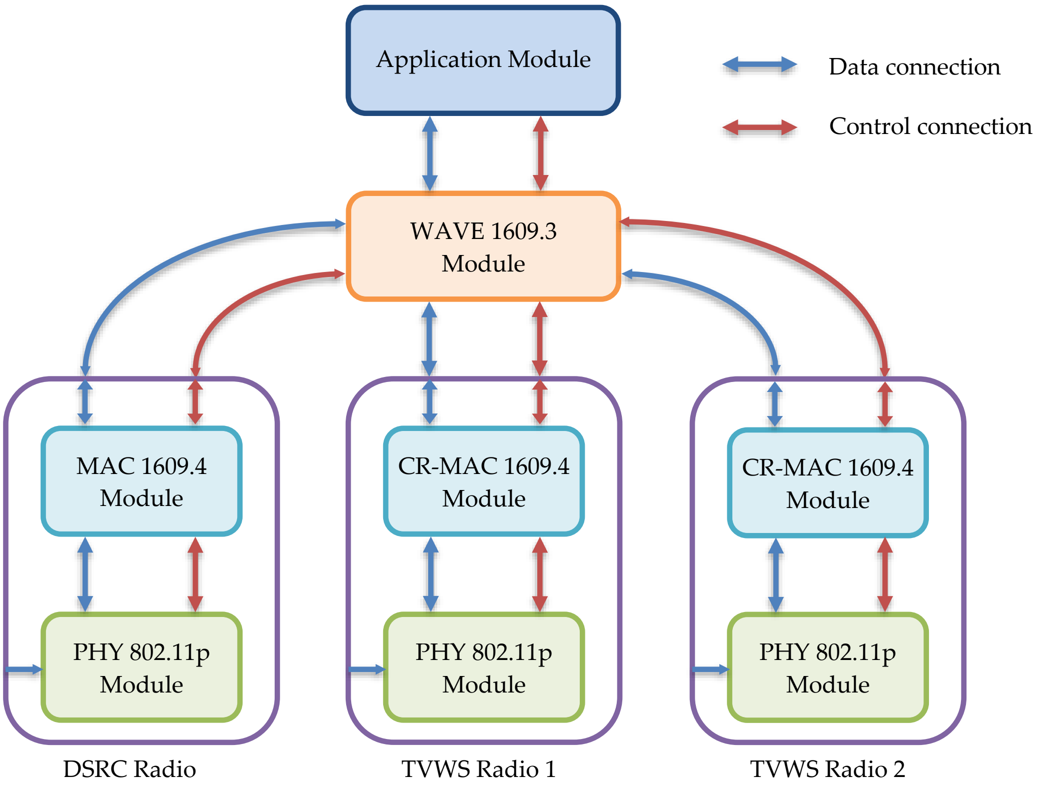


Figure 5.1: A simulation model of a cognitive radio node

5.2.1 Spectrum Sensing Model

Spectrum sensing is a critical function in the cognitive cycle in learning the radio environment. It ensures that access to a shared channel is provided only when a PU does not employ it. As discussed in Chapter 2 (literature review), most spectrum sensing algorithms intend to detect the existence of a radio signal on a channel through energy detection. While sensing through energy detection is fast and suitable for detecting any signal (known or unknown), it is only capable of classifying a channel into either busy or idle state. In this method, a CR performs a hypothesis test on the presence or absence of signals in a channel, where the channel is idle under the null hypothesis $\mathcal{H}_0$ and busy under $\mathcal{H}_1$ (5.1).

$$\mathcal{H}_0\ (idle) \quad vs \quad \mathcal{H}_1\ (busy) \tag{5.1}$$

A signal received $r(t)$ by a CR may include any radio signal in the environment, as well as noise $n(t)$. When the idle scenario is considered, the received signal is essentially the noise in the radio environment, and under the busy scenario, the received signal may consist of a signal s(t) transmitted from a radio present in the surrounding environment and noise (5.2).

$$r(t) = \begin{cases} n(t);\ \mathcal{H}_0 \\ s(t) + n(t);\ \mathcal{H}_1 \end{cases} \tag{5.2}$$

A CR then compares the received signal power P_r with a selected threshold P_γ to make the decision D between $\mathcal{H}_1$ or $\mathcal{H}_0$ (5.3).

$$D = \begin{cases} \mathcal{H}_0:\ P_r \leq P_\gamma \\ \mathcal{H}_1:\ P_r > P_\gamma \end{cases} \tag{5.3}$$

According to this two-state sensing model, any captured signal is considered as a PU signal. Hence, for this model to accurately identify PUs, mandatory channel sensing phases are periodically enforced, pausing all secondary user transmissions. This type of spectrum sensing is possible when only one infrastructure-based CR network exists, wherein a base station can inform the secondary users of its sensing schedules or halt sensing processes in the network if a secondary user is transmitting data.

Nevertheless, the two-state sensing model is not appropriate for ad hoc networks such as VANETs since there is no central entity to coordinate sensing schedules. Furthermore, it is not suitable for environments where multiple CR networks operate on the same shared channel, as implementing sensing schedules across all secondary user networks is not practical. When periodic sensing phases are not coordinated or employed, secondary users may perceive a shared channel as being occupied by a PU each time it is accessed by another secondary user and may refrain from accessing the channel. Hence, it is vital to identify whether a PU or a secondary user is employing a shared channel in networking environments where periodic channel silence phases are not enforced.

In order to address this issue, a three-state spectrum sensing model was developed during this research. In this method, a CR categorises a channel into $\mathcal{H}_0\ (idle)$, $\mathcal{H}_1\ (occupied\ by\ a\ PU)$ or $\mathcal{H}_2\ (occupied\ by\ a$ secondary user), based on the radio signal it receives. As in the two-state sensing model, the received signal may consist of just noise under the null hypothesis. While the received signal may consist of a signal $s_p(t)$ transmitted from a PU radio and noise $n(t)$ under $\mathcal{H}_1$, it may comprise of a secondary user signal $s_s(t)$ and noise under $\mathcal{H}_2$ (5.4).

$$r(t) = \begin{cases} n(t);\ \mathcal{H}_0 \\ s_p(t) + n(t);\ \mathcal{H}_1 \\ s_s(t) + n(t);\ \mathcal{H}_2 \end{cases} \tag{5.4}$$

The fundamental challenge in the above three-state model is accurately differentiating between the channel states $\mathcal{H}_1$ and $\mathcal{H}_2$. Hence, in order to distinguish a PU from a secondary user signal, a carrier sensing mechanism was employed in addition to energy detection. As mentioned earlier, the radios decode the header of any variant of IEEE802.11 signal, and a carrier is sensed if the physical layer can successfully receive the header of such a signal. In this three-state spectrum sensing model, a radio channel is monitored for a period of T_s (‘*continuous sensing interval*’) for any carriers received. At the end of this period, a clear channel assessment (CCA), described in Section 4.3.3.3, is carried out on the channel to sense the power level associated with any signals on the radio channel (at the time of assessment). Subsequently, a decision is made on the occupancy of the channel based on the result of the CCA process and the detection of a carrier. The period T_s must be sufficiently large to capture several secondary user packets, but needs to be smaller than PU signal transmission durations. It is assumed that the transmission durations of PU signals are much longer than the transmission durations of secondary user signals. The process of identifying channel occupancy (idle or occupied by a PU or secondary users) is shown in Figure 5.2. At the end of the radio channel monitoring period T_s, if the CCA process indicates that the channel is busy, the channel is sensed for another period of $T_{sa} (< T_s)$ defined as the '*adaptive sensing interval*'. This process is continued for '*maximum number of sensing intervals*' (N_s) if the channel is busy at the end of each interval. As shown in Figure 5.3, there are several situations where the CCA process may indicate a busy channel at the time of making the decision.

1. When a secondary user signal is being received — Figure 5.3 (a).
2. When a collision has occurred among secondary user transmissions — Figure 5.3 (b).
3. When a primary user is transmitting — Figure 5.3 (c).

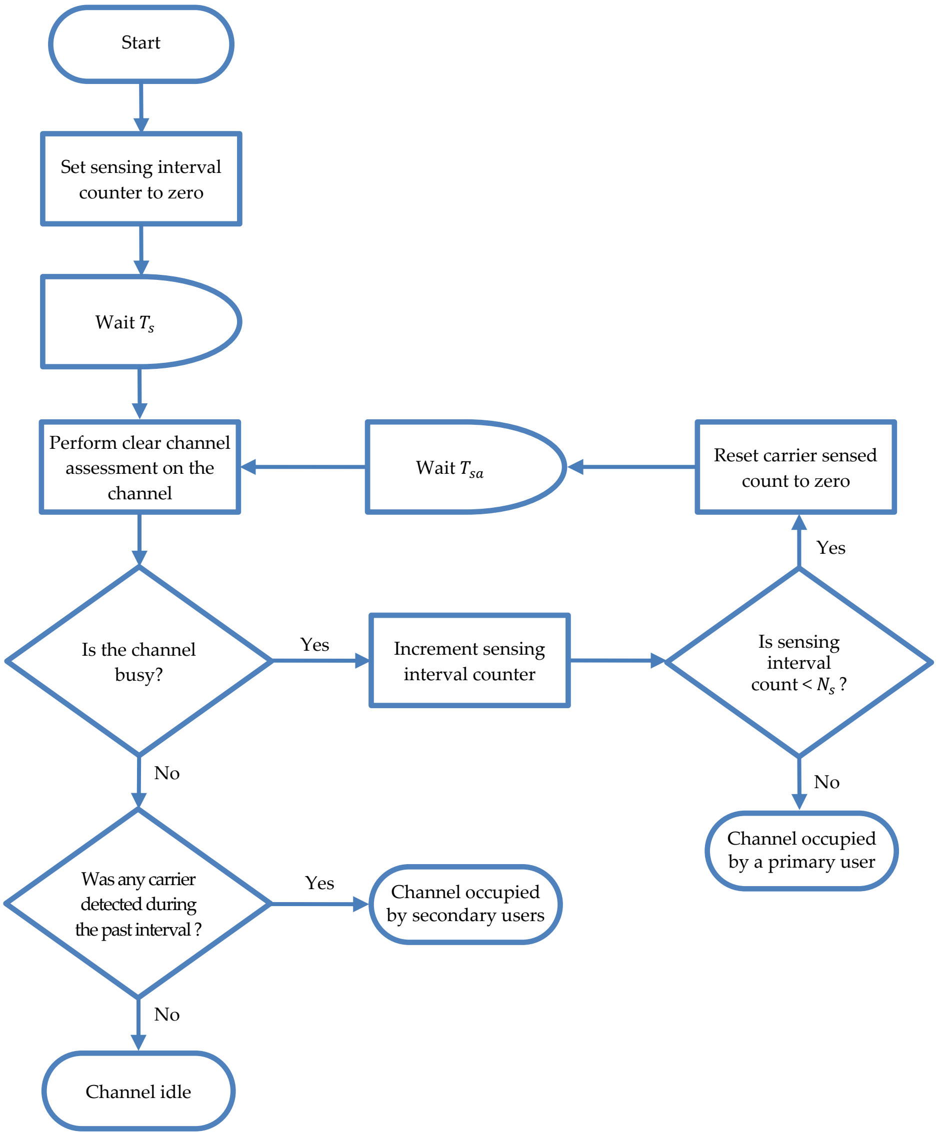


Figure 5.2: The process of identifying channel occupancy

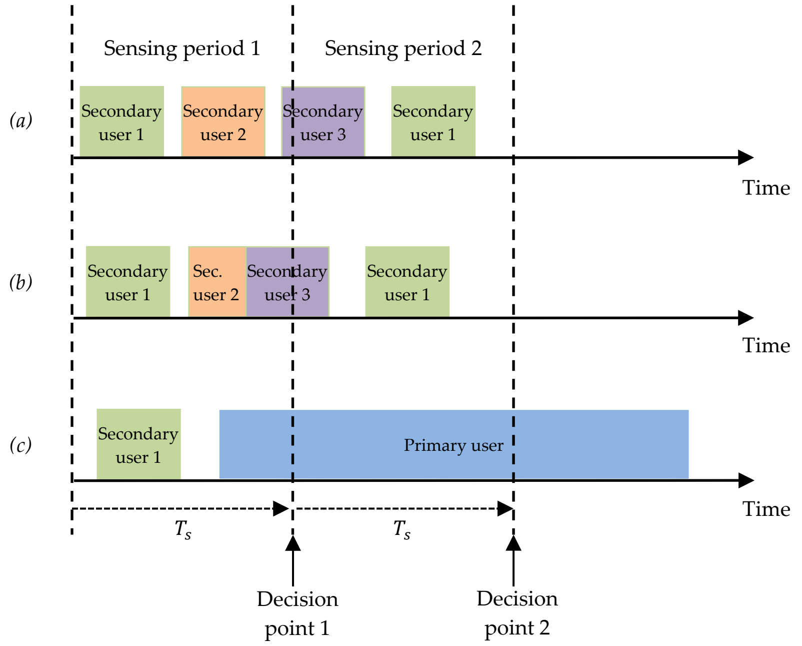


Figure 5.3: Example situations where the channel may be busy at decision points (a) when a secondary user signal is being received (b) when a collision has occurred among secondary user transmissions (c) when a primary user is transmitting

If a primary user is not transmitting, there is a high probability that the CCA process may indicate that a channel (which was previously busy) is idle when it is monitored for several periods. Therefore, if the CCA process indicates that a channel is busy after several sensing periods, it is decided that a PU has occupied the channel. However, the total time that a channel is monitored for must be shorter than the transmission duration of a primary user but needs to be sufficiently long to avoid detecting power associated with any secondary user signal collisions and the number of sensing periods must be chosen accordingly. On the other hand, if a carrier was sensed during a monitoring period, but the CCA indicates an idle channel at the end of the period, it is determined that secondary users are accessing the channel. If no carrier was detected during the last monitored period and the CCA process indicates an idle channel, the channel is deemed idle.

The spectrum sensing model employs a dynamically adaptive sensing interval to monitor a channel. The duration that a channel is monitored for when CCA indicates a busy channel (T_{sa}) is selected differently to the *continuous sensing intervals* (T_s) to make the channel sensing more efficient. While CCA indicates an idle channel, the channel is monitored in long T_s *continuous sensing intervals* to reduce system overheads and switches to short *adaptive sensing intervals* T_{sa} $(< T_s)$ when CCA indicates a busy channel to improve the accuracy of channel sensing.

The service providing nodes in the CR-VANET performs sensing on all the M channels available for CR operations regularly. The service providers tune into a radio channel and decide on the occupancy through sensing and record the type of channel occupancy. Once a decision has been made, they move onto the next channel to carry out the same task. This process is continued while spectrum sensing is enabled on a service providing node. When choosing a channel for data transmission, service providers give priority to idle TVWS channels over secondary user occupied channels. If a service provider is unable to find an idle channel, it selects a channel occupied by secondary users and competes with them for channel access. Once a radio channel has been selected for transmission, it will be removed from the spectrum sensing channel list.

5.2.2 Spectrum Mobility Model

The CR channel (on TVWS radio 1) that a service provider (SP) selects to offer its application services must also be monitored for primary users. When a PU occupies a CR channel, a secondary user is not permitted to transmit and must immediately vacate the channel through a radio channel hand-off mechanism. Each time a signal is received, the secondary users start monitoring the time elapsed while the channel is busy. It is assumed that the transmission times of PUs are much longer than the transmission times of secondary users. Hence, if the secondary users detect that a channel is busy for an extended period, they switch to another CR channel which is not occupied by a PU since the channel might have been busy due to the presence of a PU. After detecting a busy channel, the time that a secondary user waits before continuing with the channel switch (T_D) can be specified in the *initialisation file* at the start of a simulation run.

In order to carry out seamless transitions between radio channels, the SPs make use of the WAVE service advertisements (WSAs) discussed in Chapter 4. The radio channels available for CR operations are being sensed continuously by the SPs using the 'TVWS radio 2' to identify radio channels that are not occupied by the PUs and use them as backup channels. In the event that the operating channel becomes unavailable for CR operations, the SPs will switch to one of the backup channels. The radio channel that was used before the switch is then added back to the spectrum sensing channel list, while the backup channel is removed from the list.

The backup channels that SPs intend to use are advertised using WSAs along with the current operating channel. In order to accommodate the advertisement of the backup channel, the format of the WSA given in Section 4.3.4.1 was modified, as shown in Figure 5.4. The service users (SUs) upon receiving a WSA will tune one of the TVWS radios to the 'service channel number' while the other TVWS radio to the 'backup service channel number'.

WSA version	WSA length	WSA identifier	Content count	Repeat rate	Provider service identifier	Advertiser identifier	Service channel number	Backup service channel number	Channel access	Provider MAC address

Figure 5.4: Format of the WAVE service advertisement

5.2.3 Primary User Model

In the developed simulation framework, the primary user's activity on licensed radio channels is modelled as alternating ON and OFF periods. A primary user generates a signal, which holds for an exponentially distributed time with a mean T_{ON}, and turns off for an exponentially distributed time with a mean T_{OFF}. The T_{ON} and T_{OFF} periods, as well as the primary user's operating centre frequency, transmission power and position, can be configured at the start of a simulation run using the *Initialization Files*.

5.3 Model Implementation

5.3.1 Extending the Physical Layer Model

The physical layer model represented by the PHY 802.11p module (described in Section 4.3.3) was extended to implement the three-state spectrum sensing model described in Section 5.2.1 and to receive non-IEEE802.11 signals of primary users. As described in Section 4.3.3, the physical layer is divided into three parts, '*Base Physical Layer*', '*Analogue Models*' and '*Decider*'. One of the main tasks of the '*Base Physical Layer*' is the processing of received *Air Frames*. The *Base Physical Layer* registers all received *Air Frames* in the module called '*Channel Info*' that keeps track of the *Air Frames* and provides details of all intersecting *Air Frames* within a given time interval. Next, the *Base Physical Layer* passes the received *Air Frame* to the *Analogue Model* to calculate the attenuation of the signal. Subsequently, the received signal is handed over to the *Decider* to determine whether the signal can be correctly received.

In this implementation, the signal is processed three times by the *Decider*.

1. At the start of the *Air Frame* reception
2. When the header is received
3. At the end of the *Air Frame* reception

At the start of the signal reception, the *Decider* evaluates if the signal is received on the channel that the physical layer is tuned in. Moreover, it analyses the power level of the signal to determine if the signal is strong enough to be received. If the signal is too weak or if the signal is on a channel different to the current physical layer channel, the channel is considered idle, and the signal will not be attempted to decode at the end. Next, the Clear Channel Assessment (CCA) process described in Section 4.3.3.3 analyses whether the superposition of low power signals may make the channel busy. If the CCA decides that the channel is busy, a '*channel busy*' control message is sent to the MAC layer.

On the other hand, if the signal is strong enough to be detected by the receiver, a '*channel busy*' control message is sent to the MAC layer, and an event will be scheduled to re-evaluate the signal by the *Decider* when the header is received. The next time the *Decider* gets the signal, it decides whether the header can be correctly received by calculating the Signal to Noise plus Interference Ratio (SNIR) and Bit Error Rate (BER), as described in Section 4.3.3.3, up to the current time (end of the header). If the header can be decoded, a '*carrier sensed*' control message is sent to the MAC layer to indicate that the header of an IEEE802.11 signal has successfully been received. Next, an event will be scheduled to re-evaluate the signal by the *Decider* at the end of the *Air Frame* reception. If the signal header has been received correctly, the *Decider* evaluates the payload for bit errors when the *Air Frame* is received completely. If both the header and the payload can be received correctly, the packet is sent to the MAC layer. Otherwise, it is reported to the MAC layer as an erroneous reception through a control message. At the end of the signal reception process, the *Decider* determines whether the channel is idle or busy through the CCA and sends a '*channel idle*' control message to the MAC layer if it decides that the channel is idle.

5.3.2 The Cognitive Radio Enabled MAC Layer Model

The CR-MAC 1609.4 module in the simulation model was created extending the MAC 1609.4 module described in Section 4.3.5 to carry out cognitive functions such as spectrum sensing. The spectrum sensing functionality should be enabled on one of the TVWS radios through the MAC layer settings on the '*Initialisation file*' before the start of a simulation run. Furthermore, the radio channels that need to be sensed for PU activity should also be entered into the '*sensing channel list*'. The CR-MAC 1609.4 module selected for spectrum sensing does not accept any messages from the upper layers to be transmitted, as well as '*change MAC channel*' requests. If spectrum sensing is enabled on a TVWS radio, the CR-MAC 1609.4 module executes the spectrum sensing process shown in Figure 5.5.

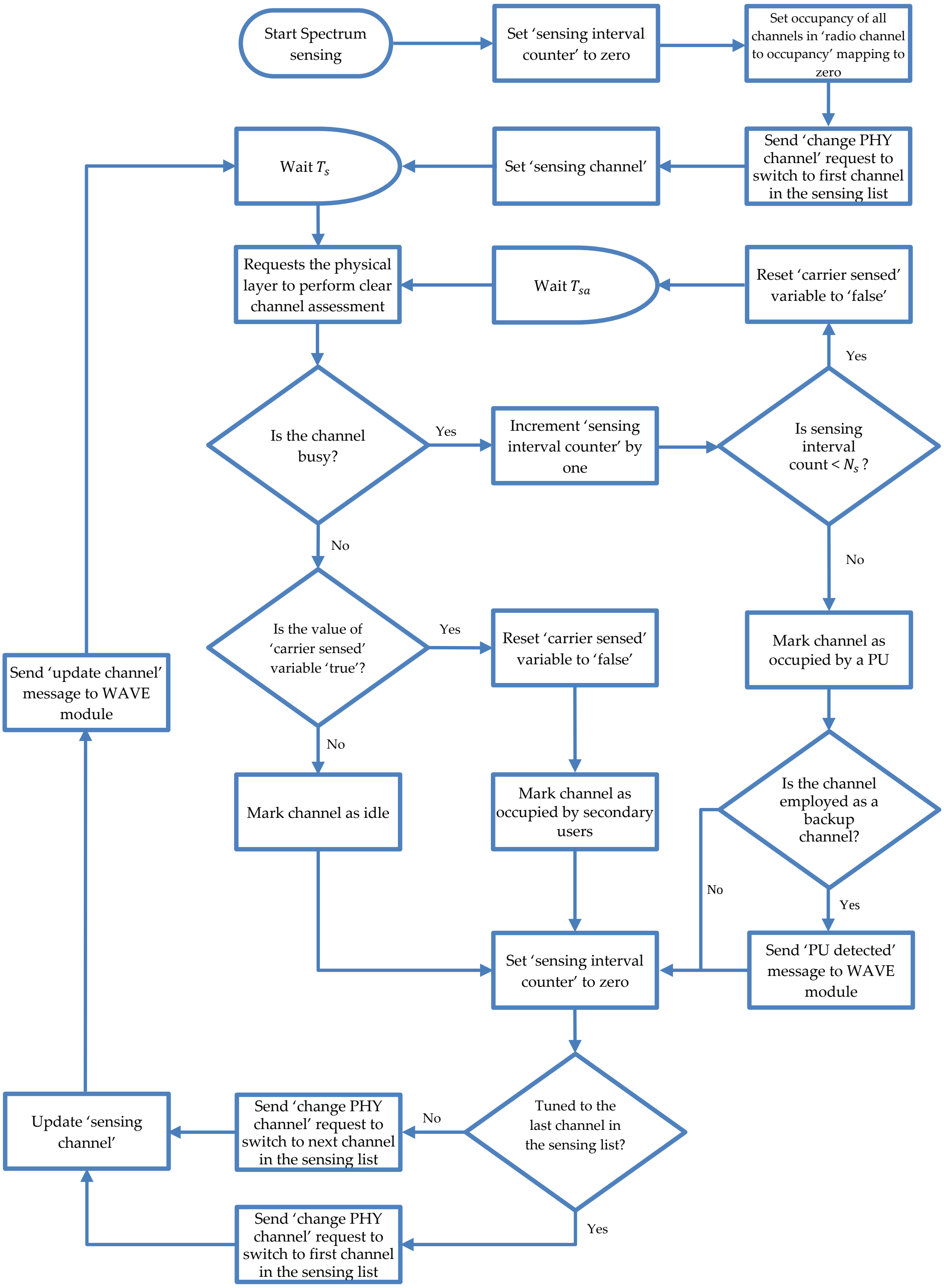


Figure 5.5: Spectrum sensing process

The spectrum sensing enabled CR-MAC 1609.4 module tunes into each channel in the '*sensing channel list*' and determines the type of occupancy of a channel (idle, occupied by secondary users or a PU) after monitoring the channel for a period of T_s ('*continuous sensing interval*') or T_{sa} ('*adaptive sensing interval*'), as discussed in Section 5.2.1. It records the occupancy of each channel in the '*radio channel to occupancy*' mapping. Initially, the occupancy of each channel in the '*radio channel to occupancy*' mapping is set to 'zero' to indicate that it is not yet sensed. If the CR-MAC 1609.4 module determines a channel as being occupied by a PU, the value of the occupancy is set to 'one'. It assigns the value 'two' if it decides that a channel is occupied by secondary users and sets the value 'three' when a channel is idle. An application service that operates on a DSA allowed band utilises two idle or secondary user occupied channels, of which one channel is employed as the primary channel for application data transfers while the other is used as a backup channel. The CR-MAC 1609.4 module records the backup channel that will be used for application data transfers in the variable '*backup channel*'.

The CR-MAC 1609.4 module assigns the value 'true' to the variable '*carrier sensed*' when it receives a '*carrier sensed*' control message from the physical layer. At the end of a '*continuous sensing interval*', the CR-MAC 1609.4 module requests the physical layer to perform CCA on the channel, and if the CCA process indicates that the channel is busy, the CR-MAC 1609.4 module monitors the channel for another '*adaptive sensing interval*' (T_{sa}) and the '*sensing interval counter*' is incremented by one. The module will monitor the same channel for N_s '*maximum number of sensing intervals*' if the CCA process indicates a busy channel at the end of each monitoring period. If CCA indicates a busy channel after N_s intervals the channel will be flagged as PU occupied. Moreover, if the channel is employed as a backup channel, a control message ('*PU indicator*') is sent to the WAVE 1609.3 module to notify about the PU occupancy. The format of the '*PU indicator*' message is shown in Figure 5.6, where the field 'channel number' provides the channel which the PU has been detected and the 'channel status' indicates whether the channel is a backup or a primary channel (the value 'zero' indicates a backup channel, while 'one' indicates a primary channel). At the end of a channel monitoring period, if CCA indicates an idle channel and the value of the '*carrier sensed*' variable is 'true', the CR-MAC 1609.4 module will mark the channel as secondary user occupied. On the other hand, it will flag the channel as idle when '*carrier sensed*' variable is 'false' and the CCA indicates an idle channel.

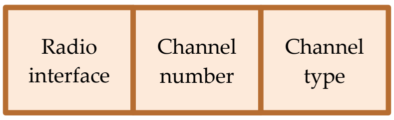


Figure 5.6: Format of the PU indicator

The CR-MAC 1609.4 module of the TVWS radio 1 used for application data transfers is also on the lookout for PU signals on the primary channel. As described in Section 5.2.2, if the module detects that the primary channel is busy for a period of T_D it switches to the backup channel. When a '*channel busy*' control message is received from the physical layer, it assigns the value 'true' to the variable '*energy detected*' and additionally schedules an event to evaluate the status of '*energy detected*' after a T_D period. The CR-MAC 1609.4 module sends a '*PU indicator*' message to the WAVE 1609.3 module, with the 'channel status' field set to 'one', if the value of '*energy detected*' is 'true' after the T_D period.

When a '*PU indicator*' message is received with value 'one' assigned to the 'channel status' field, the WAVE 1609.3 module starts the process of tuning into the backup channel. It first sends a '*release backup channel*' control message to the MAC layer of the spectrum sensing enabled TVWS radio to gain access to the backup channel. Upon receiving a '*release backup channel*' message, the MAC layer removes the backup channel from the '*sensing channel list*', since the backup channel will be used by the non-spectrum sensing enabled TVWS radio. Moreover, if the MAC layer of the spectrum sensing enabled TVWS radio is tuned to the backup channel to sense the channel at the time of receiving a '*release backup channel*' message, it will immediately tune to next channel in the '*sensing channel list*' and reset the value of the '*carrier sensed*' variable to 'false'. Finally, the MAC layer will send a '*backup channel released*' message back to the WAVE 1609.3 module to confirm that the backup channel is removed from sensing and it is free to use.

5.3.3 Extending the WAVE Service Management Model

The WAVE 1609.3 module manages the application services on a node, as discussed in Chapter 4. The service provider model within the WAVE 1609.3 module was developed to offer application services on a given channel. On the other hand, the service user model was developed to enable nodes to participate in application services offered by the service providers. However, the WAVE 1609.3 module was required to be extended to deliver these application services on a TVWS channel.

5.3.3.1 The Service Provider Model

The service provider model of the WAVE 1609.3 module accepts provider service requests from the application module to have WSAs generated and access provided to a channel to transmit application service data, as described in Section 4.3.4.1. When the WAVE 1609.3 module receives a provider service request, it evaluates the ‘service channel’ field in the request, and if it indicates a channel number, access to the channel is provided through the radio specified in the ‘service radio interface’ field, as discussed in Section 4.3.4.1.1. However, if access to a specific channel is not requested, the WAVE 1609.3 module grants access to a TVWS channel (not occupied by a PU) through the stated TVWS radio interface. The WAVE 1609.3 module first queries the MAC modules of each connected radio to find the radio that performs spectrum sensing. The WAVE 1609.3 module then queries for an idle channel recorded in the ‘*radio channel to occupancy*’ mapping of the spectrum sensing enabled MAC module, and if no idle channel is found, a secondary user occupied channel is selected. However, if there are no idle or secondary user occupied channels available, an error will be generated. Furthermore, when choosing a channel, the ‘*sensing channel*’ is excluded from the selection as the MAC module is currently sensing it. Since the selected channel is used by the non-spectrum sensing TVWS radio for application data transfers, it is removed from the ‘*sensing channel list*’ to avoid switching into the same channel by the spectrum sensing enabled TVWS radio.

Next, the WAVE 1609.3 module queries for another idle or secondary user occupied channel to be used as a backup channel if the primary channel becomes unavailable due to PU activity. The '*sensing channel*' as well as the primary channel selected for application data transfers are omitted when selecting a backup channel, and the chosen channel is then recorded in the variable '*backup channel*'. Next, the WAVE 1609.3 module provides access to the selected primary TVWS channel through the non-spectrum sensing enabled TVWS radio as described in Section 4.3.4.1.1. The module then advertises the primary channel on which the application service is offered along with the backup channel using the WSA messages shown in Figure 5.4. The secondary users monitoring for received WSAs will tune one of their TVWS radios to primary channel while the other to the backup channel (details are provided in Section 5.3.3.2).

The spectrum sensing enabled MAC module senses the channels in the '*sensing channel list*' repeatedly, and if it detects a PU on the backup channel, it sends a '*PU indicator*' control message to the WAVE 1609.3 module. Upon reception of a '*PU indicator*' message with 'channel status' field set to 'zero' (indicates a backup channel), the WAVE 1609.3 module queries for another backup channel which is idle or secondary user occupied, and updates the '*backup channel*' variable. Next, to inform service users about the changes made to the application service, the module increments the current value of the 'content count' by one and generates WSAs with the new backup channel. Before generating the WSAs with the new 'content count' value, the scheduled WSA transmissions are cancelled.

The MAC layer of the TVWS radio used for application data transfers (TVWS radio 1) is also on alert for PU signals and sends a '*PU indicator*' control message with its 'channel status' field set to 'one' (indicates a primary channel) to the WAVE 1609.3 module upon detecting a PU on the primary channel. When a '*PU indicator*' message is received, the WAVE 1609.3 module initially sends a '*release backup channel*' control message to the MAC layer of the spectrum sensing enabled TVWS radio. When the backup channel release is completed (described in Section 5.3.3.1), the MAC layer sends a '*backup channel released*' message to the WAVE 1609.3 modules.

Upon receiving a '*backup channel released*' message, the WAVE 1609.3 module sends a '*change MAC channel*' request to the MAC layer of the non-spectrum sensing enabled TVWS radio to switch to the backup channel. The '*change MAC channel*' request described in Chapter 4 removes packets queued in the EDCA subsystem when the operating channel is changed. Since the packets queued should still be delivered to the recipient through the backup channel, these packets need to be retained in the queue. Hence, the '*change MAC channel*' request was extended, and an additional field ('retain packets') was added as shown in Figure 5.7 to notify the MAC layer not to remove the existing packets from the queue.

Radio interface	Time slot 0 channel	Time slot 1 channel	Alternating channel access	Retain packets

Figure 5.7: Format of the change MAC channel request

Next, the primary channel that was used before the switch is added back to the '*sensing channel* list' by sending an '*add sensing channel*' message to the MAC layer of the spectrum sensing enabled TVWS radio indicating the PU detected channel. Furthermore, the occupancy of the PU detected channel in the '*radio channel to occupancy*' mapping is set to zero to indicate that the channel is not yet sensed. Subsequently, the WAVE 1609.3 module queries the MAC layer for another backup channel and advertises the new backup channel along with the new primary channel (previous backup channel) using WSAs by incrementing the current value of the 'content count' by one. In order to provide a seamless transition between the radio channels, the secondary users listen on the backup channel in addition to the primary channel (details are provided in Section 5.3.3.2).

5.3.3.2 The Service User Model

Under the service user (SU) model of the WAVE 1609.3 module, the SUs monitor for received WSAs. The SU model described in Section 4.3.4.2.1 was extended to accept the WSAs (format shown in Figure 5.4) with a 'backup service channel number'. The application layer initially sends a user service request to obtain access to radio channels to receive WSAs and also to provide information related to the application service that the SU intends to participate. The user service request described in Section 4.3.4.2.1 was extended (the 'backup service radio interface' field was added) to notify the

WAVE 1609.3 module regarding the service radio interface to be used for the backup channel, if the application service employs a backup channel. The format of the user service request is shown in Figure 5.8. Upon reception of a user service request with the 'action' field set to 'add', the WAVE 1609.3 module records the provider service identifier (PSID) and the service radio interface in the '*PSID to service radio interface*' mapping. Furthermore, it records the service radio interface for the backup channel by adding the suffix '00' to the PSID. Subsequently, it sends a '*change MAC channel*' request to the MAC layer of the 'WSA radio interface' to switch to the 'WSA channel' to receive WSAs.

Action	Provider service identifier	Advertiser identifier	WSA channel	WSA time slot	WSA radio interface	Service radio interface	Backup service radio interface

Figure 5.8: Format of the user service request

Upon receiving a WSA, the WAVE 1609.3 module obtains the radio interface to be employed for the application service from the '*PSID to service radio interface*' mapping and adds radio channel information ('service channel number') related to the application service in the '*PSID to radio time slot channel*' mapping. If the received WSA contains a 'backup service channel number', the module adds the suffix '00' to the PSID and evaluates whether a radio interface has been assigned for the backup channel in the '*PSID to service radio interface*' mapping. If a radio interface is allocated for the backup channel, the WAVE 1609.3 module adds the backup channel information also in the '*PSID to radio time slot channel*' mapping. Subsequently, it sends '*change MAC channel*' requests to the MAC layers of the 'service radio interface' and 'backup service radio interface' (if available) to tune to the 'service channel' and 'backup service channel', respectively, and participate in the application service.

When a service provider detects a PU in its operating channel (primary channel), it switches to the advertised backup channel as described in Section 5.3.3.1. If a service user has a radio interface allocated for the backup channel and it has tuned in to the backup channel at the time the service provider is switching to the backup channel, the service user will receive the application service data without any interruption. Hence, this channel handover mechanism provides a seamless transition between the radio channels. However, even if a service user does not have a radio interface tuned to the backup channel, it will still be able to participate in the application service after the service provider's channel switching activity since the service provider advertises the new service channel using the WSAs. Nevertheless, the service user may observe a short interruption to the application service until it switches to the new service channel.

Upon reception of a WSA with the new service channel and the backup service channel, the WAVE 1609.3 module compares the new service channel with the previous backup channel. If they are the same, the module swaps the radio interfaces assigned for the application service (PSID) and its backup in the '*PSID to service radio interface*' mapping (i.e. the radio interface specified for the application service will be assigned to its backup, while radio interface specified for the backup will be allocated to the application service). Furthermore, the entries in the '*PSID to radio time slot channel*' mapping related application service and its backup are also swapped in a similar manner. Subsequently, the '*PSID to radio time slot channel*' mapping is updated with the new backup channel, and a '*change MAC channel*' request is sent to the MAC layer of the radio interface assigned for the backup channel.

5.4 Performance Evaluation

In order to investigate the performance of the three-state spectrum sensing model, a 1km highway was set up on the SUMO road traffic simulator, and two vehicular traffic flows which begin at each end of the road were configured. The 1km road segment was constructed with four lanes in each direction to accommodate more vehicles. Also, to perform simulations avoiding border effects (as described in Section 4.3.2), the simulation area (playground) was modelled as the surface of a torus. The performance of the sensing model was evaluated through the following metrics.

Probability of detection – Defined as the probability that a vehicle detects the occupancy of the channel accurately. This includes detecting the presence of a PU when the channel is occupied by a PU, sensing the presence of secondary users when the channel is occupied by secondary users and detecting an idle channel when the channel is not occupied by either primary or secondary users.

Probability of false alarm – Defined as the probability that a vehicle detects the presence of a PU when in fact the channel is not occupied by a PU.

Probability of missed detection – Defined as the probability that a vehicle fails to indicate the presence of a PU when the channel is occupied by a PU.

The performance of the spectrum sensing model was evaluated under three different communication scenarios. In the first scenario, a primary user transmission tower was positioned by the side of the road, as shown in Figure 5.9. The primary user generates a signal that has exponentially distributed ON and OFF periods with mean T_{ON} and T_{OFF} (as described in Section 5.4). The primary user signal transmission takes place on a radio channel (centre frequency at 812 MHz and bandwidth 10 MHz) in the TVWS band, and the vehicles (operating as service providers) sense the channel, as described in Section 5.2.1. The *maximum number of sensing intervals* N_s used in this scenario was two. The occupancy of the radio channel was recorded by vehicles at the end of each *continuous sensing interval* T_s. The *adaptive sensing interval* T_{sa} was also made equal to T_s. The probability of detection and the probability of missed detection were then calculated for different T_s values, and T_{ON} and T_{OFF} values. The simulation was repeated 1000 times for each instance. The parameters used for the simulation of the first scenario are given in Table 5.1.

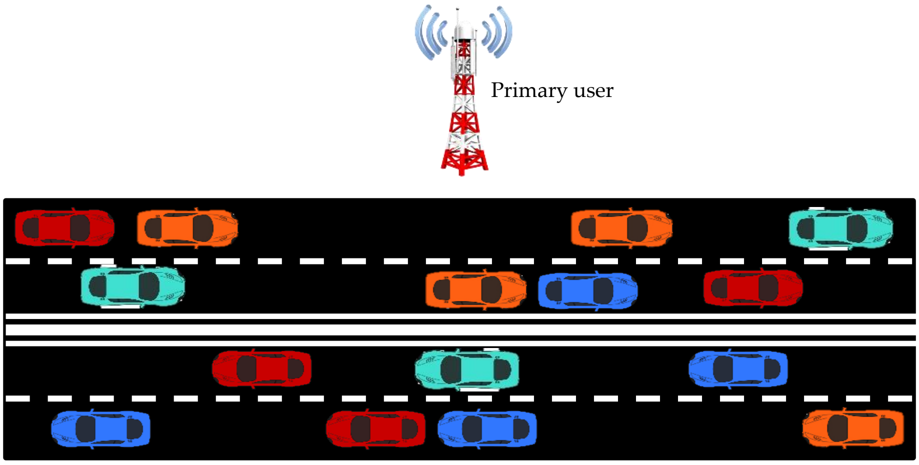


Figure 5.9: The simulation scenario with a primary user by the side of the road

Table 5.1: Simulation parameters of scenario one

Simulation Parameters	Values
Propagation model	Path loss
Radio transmit power	13.01dBm (20mW)
Radio sensitivity	-89dBm
Thermal noise	-110 dBm
MAC bit rate	6Mbps
WSM packet size	1400bytes
Max. number of sensing intervals (N_s)	2

The data collected during the first scenarios is represented in Figure 5.10 and Figure 5.11. According to the spectrum sensing algorithm, the sensing interval should ideally be shorter than the transmission lengths of the primary user. Hence, the simulations were performed for sensing intervals up to 100ms. It can be observed in Figure 5.10 that the probability of detection was higher than 0.995 across the entire range of sensing intervals. The highest probability of detection was recorded when the channel was sensed at 10ms intervals indicating shorter sensing intervals lead to a higher probability of detection. However, when the sensing intervals are shortened the overheads of the system will inevitably increase. Since the system was achieving a probability of detection of approximately 0.9997 at 10ms sensing intervals, the simulations for sensing intervals less than 10ms were not carried out as the objective of the developed spectrum sensing mechanism is to lower the system's sensing overheads.

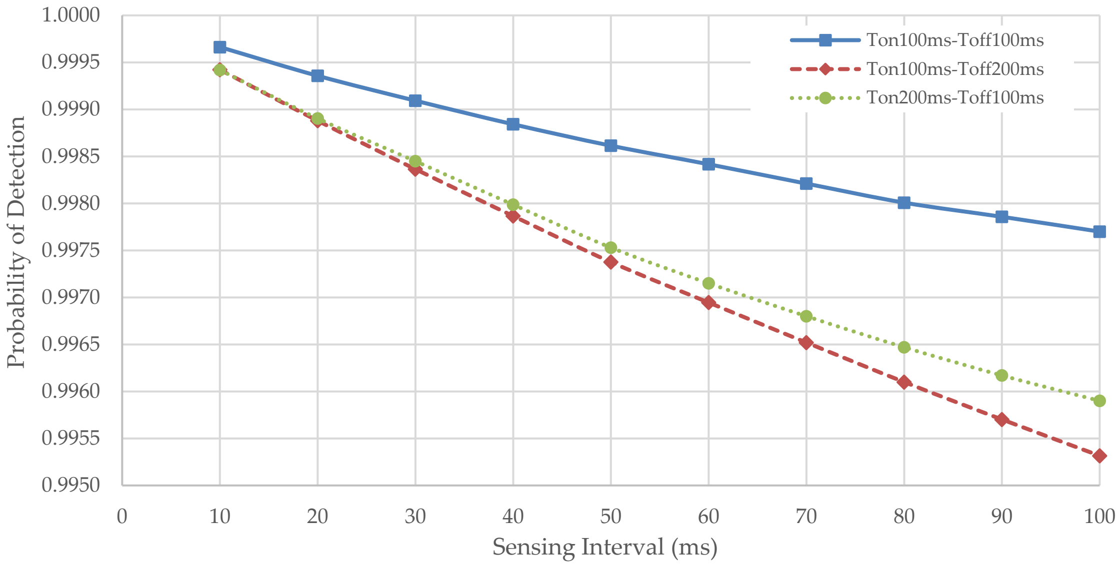


Figure 5.10: Probability of detection against sensing interval for different primary user transmission durations

The sensing algorithm decides that a channel is not occupied by a PU, if the clear channel assessment (CCA) indicates an idle channel at the time the sensing decision is made. However, if a PU was transmitting during some part of the sensing interval but was turned off when the sensing decision was made, the sensing algorithm fails to identify the presence of the PU. Hence, when the sensing interval is much smaller than the transmission duration of a PU, the probability of missing the detection of a PU is smaller. In Figure 5.11, the lowest probability of missed detection was observed when the sensing interval was as low as 10ms. Moreover, the probability of missed detection was always lower than 0.005 for the sensing intervals simulated.

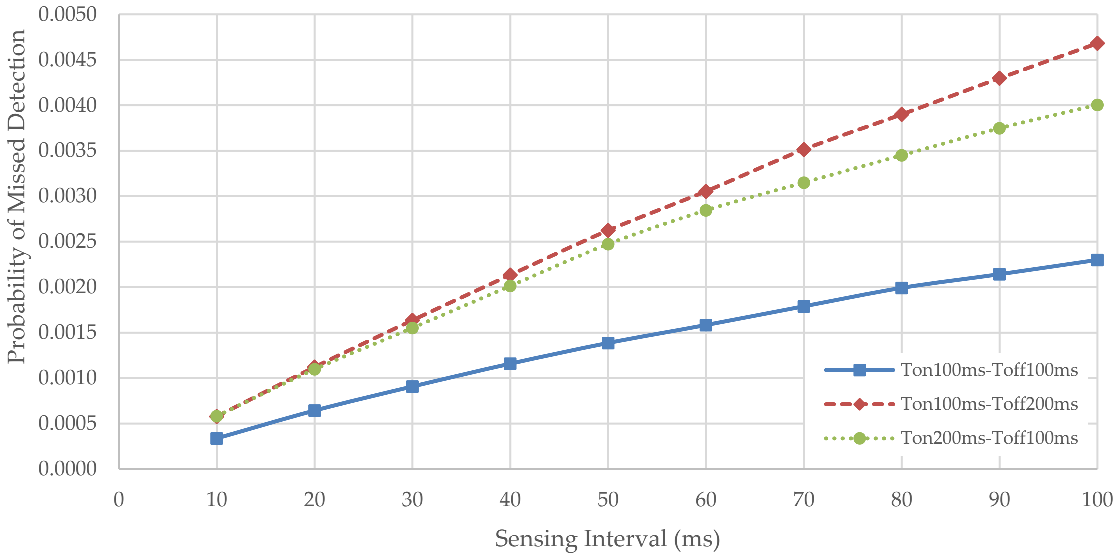


Figure 5.11: Probability of missed detection against sensing interval for different primary user transmission durations

In the second scenario, a number of vehicles were configured to broadcast a WAVE short message (WSM) repeatedly on the same radio channel as before (centre frequency at 812 MHz). The sending interval between the generated WSMs are exponentially distributed with mean (T_{send}) 100 ms. Furthermore, the channel hand-off mechanism of the transmitting vehicles was disabled to keep the transmission on the same channel. The vehicles that were not configured to transmit were set up to sense and record the occupancy of the radio channel. The probability of detection and the probability of false alarm were calculated for different numbers of transmitting vehicles and N_s (*maximum* *number of sensing intervals*). The simulation was repeated 1000 times for each instance. The parameter set used for the simulation of the second scenario is given in Table 5.2.

Table 5.2: Simulation parameters of scenario two

Simulation Parameters	Values
Propagation model	Path loss
MAC bit rate	6Mbps
WSM packet size	1400bytes
Radio transmit power	13.01dBm (20mW)
Radio sensitivity	-89dBm
Thermal noise	-110 dBm
T_s	10ms
T_{sa}	10ms
T_{send}	100ms

Figure 5.12 and Figure 5.13 show the simulation results obtained during the second scenario. It can be observed that the probability of detection deteriorates with the rise in the number of transmitting vehicles. On the other hand, the probability of false alarm rises with the increase in the number of transmitting vehicles. When a large number of vehicles compete for radio channel access through the CSMA/CA mechanism, the probability of packet collisions increases. The channel may be perceived as being occupied by a PU when these collisions occur among secondary user transmissions at the time of deciding the occupancy of a channel. Hence, the increase in the probability of false alarms. However, to improve the accuracy of differentiating PUs from secondary users, the developed algorithm repeatedly senses a channel over several periods when the CCA indicates a busy channel, before deciding on the occupancy of the channel. It can be clearly seen from Figure 5.12 and Figure 5.13 that the probability of detection improves and the probability of false alarm reduces significantly when the *maximum number of sensing intervals* N_s was increased. For example, by increasing the number of sensing intervals from 2 to 10, an approximately 76% improvement in the probability of detection was achieved when 40 vehicles were accessing the channel.

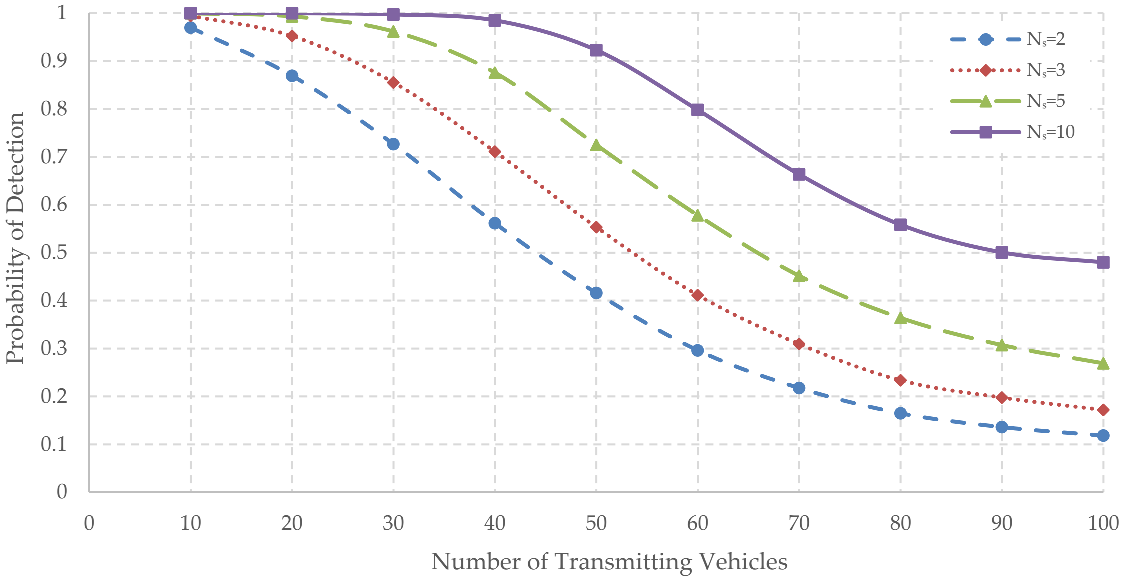


Figure 5.12: Probability of detection against number of transmitting vehicles

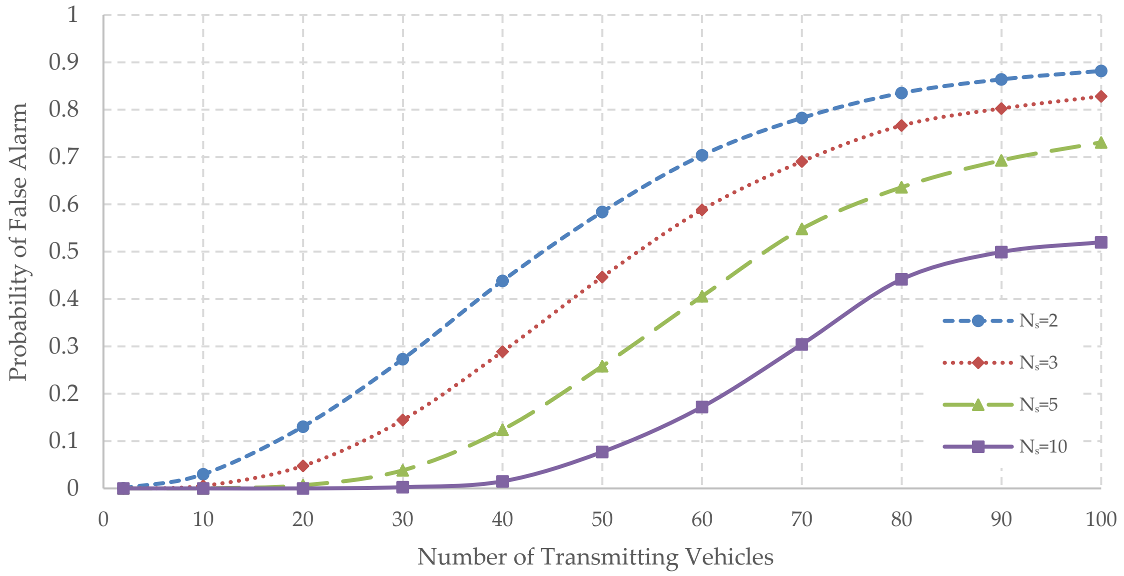


Figure 5.13: Probability of false alarm against number of transmitting vehicles

In the third scenario, several vehicles were again configured to broadcast a WSM repeatedly on the same radio channel with an exponentially distributed mean (T_{send}) of 100ms between the generated WSMs. The non-transmitting vehicles were configured to sense and record the occupancy of the radio channel. The sensing of the channel was performed with and without dynamic sensing. The sensing overhead, which is the number of times a channel is sensed per second, was quantified using the following metric *(5.5)*.

$$Sensing\ overhead = \frac{Number\ of\ times\ a\ channel\ is\ sensed}{Total\ simulation\ time} \tag{5.5}$$

The *continuous sensing interval* (T_s) and the *adaptive sensing interval* (T_{sa}) used for the non-dynamic sensing scenario were 10ms. For the dynamic sensing scenario, the T_s was maintained at 100ms and upon sensing a busy channel it is dropped down to 10ms which was defined as the *adaptive sensing interval* (T_{sa}). Although T_s of 100ms reduces the probability of detection, it would significantly reduce the sensing overheads. To improve the accuracy of sensing while reducing the sensing overheads, T_{sa} was chosen to be 10ms. The *maximum number of sensing intervals* N_s was maintained at 10 for improved accuracy of sensing. The simulation was repeated 1000 times for each instance, and the parameter set used for the simulation of the third scenario is tabulated in Table 5.3. The probability of detection and the sensing overhead were calculated for different numbers of transmitting vehicles, as shown in Figure 5.14 and Figure 5.15.

Table 5.3: Simulation parameters of scenario three

Simulation Parameters	**Values**
Propagation model	Path loss
MAC bit rate	6Mbps
WSM packet size	1400bytes
Radio transmit power	13.01dBm (20mW)
Radio sensitivity	-89dBm
Thermal noise	-110 dBm
T_s for non-dynamic sensing	10ms
T_{sa} for non-dynamic sensing	10ms
T_s for dynamic sensing	100ms
T_{sa} for dynamic sensing	10ms
T_{send}	100ms

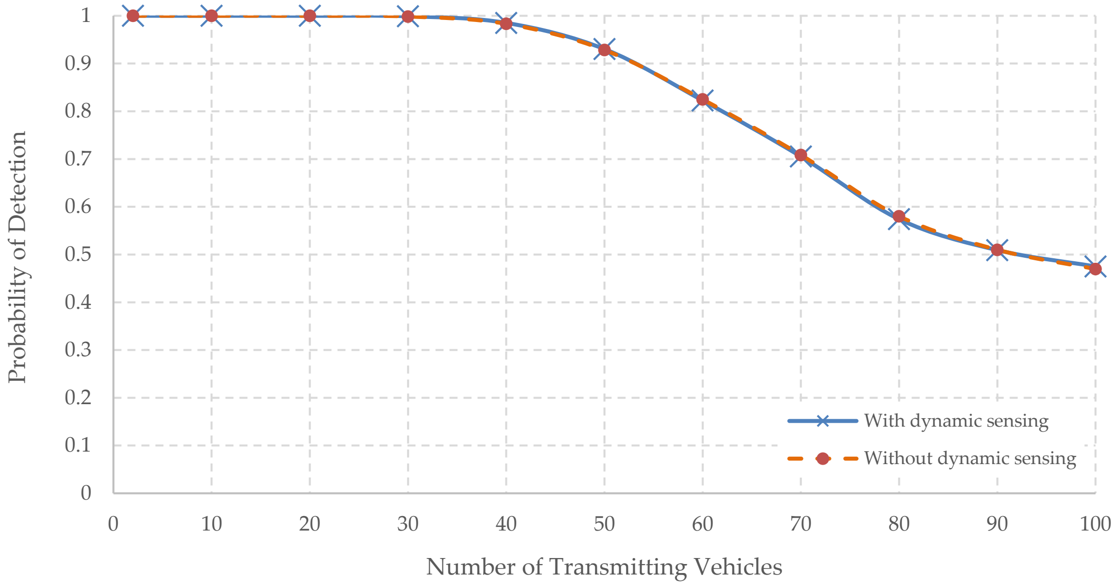


Figure 5.14: Probability of detection against number of transmitting vehicles

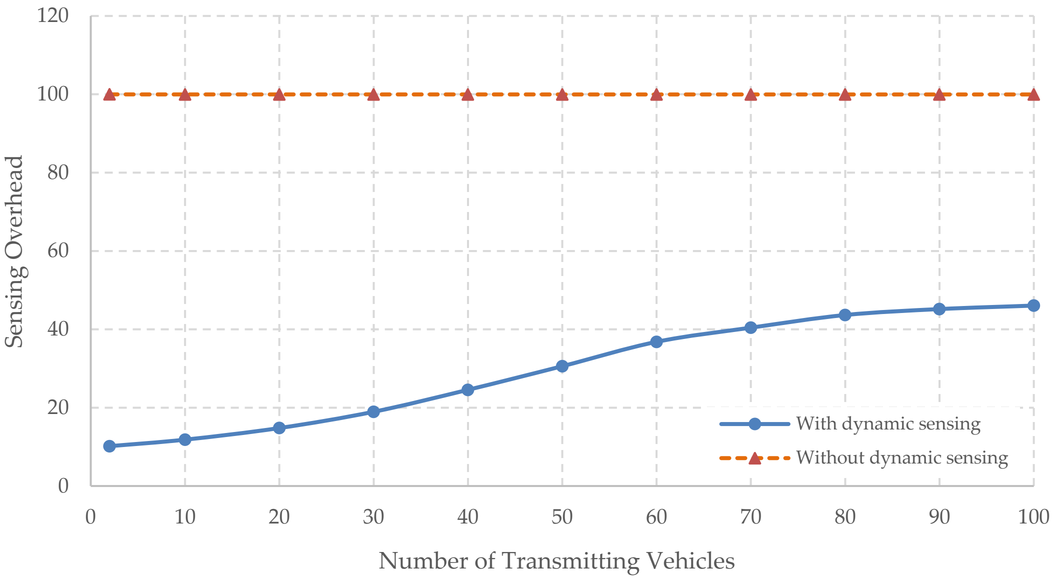


Figure 5.15: Sensing overhead against number of transmitting vehicles

The probability of detection observed in Figure 5.14 was identical for both scenarios with and without dynamic sensing. However, it was observed in Figure 5.15 that the dynamic sensing significantly reduced the system's overheads between 54% and 90%, indicating that less system resources would be needed with this novel technique.

CHAPTER 6

A Multi-Channel Signalling Framework for Cognitive Radio Enabled VANETs

6.1 Introduction

In ad hoc networking environments such as cognitive radio enabled VANETs (CR-VANETs), where no centralized servers or base stations are available to manage connections between nodes, a proper mechanism is required to discover application services offered on channels in which dynamic spectrum access (DSA) is allowed, such as the Television White Spaces (TVWS). It is also essential to establish and maintain connections between a provider and a user of an application service. In Chapter 5, such a system was developed to manage these application services by making use of WAVE service advertisements (WSAs). However, this necessitated a communication channel common to all the nodes in a vehicular network to broadcast WSAs, wherein the CCH (radio channel 178) was employed. In VANETs, vehicles often tune into the CCH (at least during time slot 0) to receive vehicular safety and management messages.

In recent CR-VANET related studies, the CCH has been utilised as a common channel by many other authors for cognitive radio (CR) related signalling purposes and exchanging spectrum sensing information [47], [48], [55]. While employing a dedicated common control channel in the DSRC band seems rather attractive, it also presents a single point of failure when the channel becomes congested with high packet collisions. It was observed during this research as well as in some recent studies [23] that when a large number of vehicles access the CCH for safety communications, solely, the channel may become congested with high packet collisions. Hence, when the CCH is employed for CR operations in addition to safety-related communications, especially during high-density vehicular traffic, the channel may become more congested, making it unavailable for both safety and CR communications. In this research, the primary purpose of using the CR technology is to improve the bandwidth available for non-safety applications of VANETs that require high data throughputs. These applications that use DSA bands for data transfers is dependent on CR signalling that takes place on the CCH. However, when the CCH becomes congested, the CR control message transmissions get disrupted, and as a result, the non-safety CR communications on DSA bands may be interrupted or failed altogether.

Hence, in this research, multiple channels in the DSRC spectrum were employed for CR related control information exchange, instead of a single channel. An algorithm was developed to continuously evaluate the congestion of all the channels in the DSRC spectrum and use the channel with the least amount of data congestion for CR signalling. Furthermore, a complete simulation framework was developed to model the designed algorithm on OMNeT++.

6.2 Multi-Channel Cognitive Radio Signalling Model

During the research, a multi-channel CR signalling model was developed extending the CR simulation framework described in Chapter 5. The nodes in this framework are comprised of three radio modules, as shown in Figure 5.1. One of the radios operates in the 5.8 GHz DSRC band, while the other two radios operate in a DSA allowed band such as the TVWS band (54-862 MHz). Spectrum sensing is carried out by one of the TVWS radios, whereas the other offers application services on PU vacant TVWS channels. The DSRC radio, on the other hand, broadcasts announcements related to these services on DSRC channels. Furthermore, the two WAVE service models, the service provider (SP) model and the service user (SU) model described in Section 5.3.3, were extended to facilitate the multi-channel CR signalling model.

6.2.1 Service Provider Model

The service provider model developed within the WAVE 1609.3 module (explained in Section 4.3.4.1) provides necessary channel access to start an application service on a radio channel specified by the application module and generates WSAs. This model was further extended to facilitate the multi-channel CR signalling model, which continuously evaluates the data congestion of all seven DSRC channels and identifies the best channel to broadcast WSAs. In CR environments, the WAVE 1609.3 module employs the WSAs in various ways.

- To advertise the TVWS channels on which the application services are offered.
- To move application services to another vacant channel in the TVWS band when the currently utilised channel gets occupied by a primary user.
- To evaluate the quality of the radio channel.

In the developed algorithm, the DSRC radio operates in the alternating channel access mode, while switching between the CCH and an SCH during time slot 0 and 1, respectively. At the end of every time interval, the SP evaluates the data congestion on the channel that it was tuned into. In order to measure the channel congestion, the metric called the channel busy ratio (CBR) was created. It is calculated as the fraction of the channel access time (50ms) a node considers a channel busy (6.1). A channel is deemed busy when a node itself transmits and when the clear channel assessment (CCA) process indicates that the channel is busy (described in Section 4.3.3.3). A higher CBR value means that the channel has experienced higher data congestion during the past time slot.

$$Channel\ busy\ ratio = \frac{channel\ busy\ period}{channel\ access\ time} \times 100\% \tag{6.1}$$

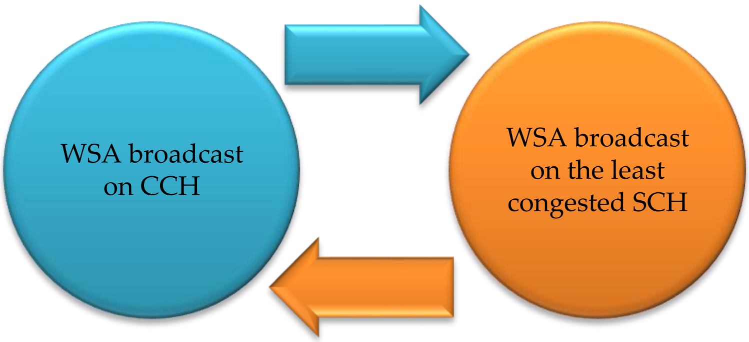
An SP is required to switch into every DSRC channel and calculate the CBR value relevant to the channel to identify the data congestion on DSRC channels. In order to carry out this process, the WAVE 1609.3 module of an SP starts off its WSA transmissions on the CCH during time slot 0 and calculates the CBR value of the CCH (CBR_{CCH}). While broadcasting of WSA messages takes place in the CCH, the SP hops to different SCHs in a round robin fashion during time slot 1 intervals to evaluate data congestion of SCHs. The order in which an SP hops to various SCH is irrelevant, provided that an SP switches to every SCH. An example channel hopping sequence is shown in Figure 6.1. In order to obtain the CBR values of all the SCHs, an SP requires a minimum of 600ms (i.e. 100ms x 6).

Figure 6.1: An example channel hopping sequence

When an SP has cycled through all the SCHs at least once and obtained channel congestion data, it decides the channel on which the WSAs should be transmitted at the beginning of each time slot. At the start of time slot 1, an SP evaluates the recorded CBR values of SCHs ($CBR_{SCH\,i}, 1 \leq i \leq 6$) and selects the SCH having the least data congestion. Then the CBR value of the least congested SCH is compared against the last obtained CBR value of the CCH. If the CBR of the chosen SCH is less than half the CBR of CCH, the SP terminates its channel hopping process and tunes into the selected SCH to broadcast WSAs. Otherwise, it will continue the channel hopping process and evaluate channel congestions. If an SP decides to switch the WSA transmissions to an SCH, it will broadcast WSAs on both the CCH (during time slot 0) and the selected SCH (during time slot 1) for a period of 600ms (the time required to complete a channel hopping round) in order to avoid disruptions to any ongoing WSA transmissions. In this algorithm, the priority was given to the CCH over an SCH for broadcasting WSA messages, since any SU can obtain a WSA transmission on a CCH. A WSA broadcast on an SCH can only be detected by SUs that has implemented the channel hopping algorithm, described in Section 6.2.2. Hence the reason for setting the threshold value to switch the transmission from the CCH to SCH at half the CBR value of the CCH.

At the start of time slot 0, if an SCH was used to broadcast WSAs, its channel congestion will be compared with that of CCH, obtained during time slot 0, to decide whether to move WSA transmissions back to the CCH. An SP switches its WSA broadcast back to CCH if the CBR of CCH is less than 75% of the CBR value of the currently utilised SCH (for the same reason described in the previous paragraph). In this case, the channel hopping will also be restarted to rediscover a better SCH for WSA transmissions. An SP needs to tune into all the SCHs at least once and obtain CBR values of each channel before deciding on the best SCH to broadcast WSAs. The channel states of a service provider are shown in Figure 6.2.

$$if\ min\{CBR_{SCH\ i}\ , 1 \leq i \leq 6\} < 0.5\ CBR_{CCH}$$



$$if\ CBR_{CCH} < 0.75\ CBR_{SCH\ current}$$

Figure 6.2: The service provider channel states

6.2.2 Service User Model

The service user model was developed to enable nodes to take part in application services offered by the service providers (discussed in Section 4.3.4.2). This model was extended to facilitate the reception of WSAs on multiple DSRC channels. Since the SPs operate on alternating channel access mode and send WSAs on either the CCH or an SCH depending on channel congestion, the SUs are also required to operate in the same mode and monitor both the CCH and SCHs during time slot 0 and time slot 1, respectively. The SUs continuously tune into different SCHs (order irrelevant) during time slot 1 intervals until a WSA transmission taking place in an SCH is discovered. An example channel hopping sequence is shown in Figure 6.1. While no order needs to be followed when switching into SCHs, the SUs are required to tune into every SCH within 600ms, since the SPs only broadcast WSAs for 600ms in both the CCH and an SCH when moving a WSA broadcast from the CCH to an SCH.

When an SU detects a WSA transmission in an SCH, the SCH channel hopping process is ceased, and the SUs may switch between the CCH and the SCH receiving the WSA transmission. In this instance, the SUs may receive the WSAs on both the CCH and the currently tuned SCH (the SUs keep track of the WSAs received on the SCH). In order to identify the channel on which the WSAs have received, the SUs make use of the 'channel number' embedded in the WSA encapsulated WSM. Next, if an SP switches WSA transmissions back to the CCH, the SUs will receive the WSAs on the CCH and none on the currently switched SCH. Hence, when the SUs receive no WSAs on the tuned SCH for 600ms (the minimum time taken by the SPs to scan the SCHs), the SCH hopping process will be restarted to detect any WSA transmissions on an SCH.

In the developed framework, the repeat rate of the WSA (format is shown in Figure 5.4) is utilised by SUs to evaluate the quality of the channel that the WSAs are received. The repeat rate indicates the number of times a WSA is repeatedly broadcast by an SP per second (for example, 10 WSAs may be broadcast within a second). In order to measure the quality of the channel, the metric called the packet reception ratio (PRR) was created. It is defined as the ratio between the number of WSAs received per second (WSA reception rate) and the WSA repeat rate.

$$Packet\ reception\ ratio = \frac{WSA\ reception\ rate}{WSA\ repeat\ rate} \times 100\% \qquad (6.2)$$

6.3 Model Implementation

6.3.1 Extending the WAVE Service Management Model

The service provider model within the WAVE1609.3 module accepts '*provider service request*' messages (described in Section 4.3.4.1) from the application module and generates WSAs to advertise the application services, which will be operating on TVWS channels. During this research, the '*provider service requests*' were further modified to allow the application services to request the WAVE1609.3 module to assess data congestion on DSRC channels and employ the channel with the least amount of data congestion for broadcasting WSA messages. The format of the '*provider service request*' is shown in Figure 6.3, wherein the parameter 'congestion analysis' indicate the WAVE 1609.3 module to enable or disable the evaluation of data congestion on DSRC channels for broadcasting WSAs. The 'WSA channel' and 'WSA time slot' parameters specify the channel and time slot on which the WSA transmission should take place initially. The SPs generally start off WSA broadcast during time slot 0 on channel 178 (CCH) since the SUs usually listen on the CCH during time slot 0. If no value is assigned for the 'service channel' field in the '*provider service request*', the WAVE 1609.3 module allocates a primary user vacant channel in the TVWS band to allow the application layer to exchange data. The process of identifying a channel not occupied by the primary users is explained in detail in Chapter 5. The WAVE 1609.3 module sends '*change MAC channel*' requests to the MAC layers to gain access to the necessary radio channels.

action	Provider service identifier	Service channel	Channel access	Service radio interface	WSA channel	WSA time slot	WSA radio interface	Congestion analysis	Repeat rate

Figure 6.3: Format of the provider service request

In the extended WAVE 1609.3 model, the format of the '*change MAC channel*' request is modified (shown in Figure 6.4) by adding the additional fields 'device role', 'congestion analysis' and 'channel hopping'. The 'device role' specifies whether a node is operating as an SP or SU. If a '*change MAC channel*' request is generated as a result of receiving a '*provider service request*', the 'device role' field is set to '1' to indicate the MAC layer that the service provider role is active. When preparing a '*change MAC channel*' request, the value of the 'congestion analysis' field is obtained from the '*provider service request*'. On the other hand, if a user service request has caused a '*change MAC channel*' request to be generated, the 'device role' field is set to '0' to indicate that the node is operating as an SU.

The '*change MAC channel*' request related to a service provider role updates the MAC layer variable '*congestion analysis*', in addition to the channel allocation variables such as the '*alternating channel access state*', '*next TS0 channel*' and '*next TS1 channel*'. A request with 'congestion analysis' enabled makes the MAC layer tune into different SCHs sequentially during time slot 1 intervals. The MAC layer holds the SCH numbers in an array and upon reception of a '*change MAC channel*' request to enable the data congestion analysis, the MAC layer activates SCH hopping starting from the SCH 1. The process of handling a '*change MAC* channel' request by the MAC layer is described in Section 6.3.2.

Radio interface	Time slot 0 channel	Time slot 1 channel	Alternating channel access	Retain packets	Device role	Congestion analysis	Channel hopping

Figure 6.4: Format of the change MAC channel request

If the MAC layer discovers a less congested channel for WSA transmissions, it sends a control message ('*best channel*' indicator) to inform the WAVE 1609.3 module regarding the channel. The format of the '*best channel*' indicator message is shown in Figure 6.5. Upon receiving a '*best channel'* indicator message, the WAVE 1609.3 module cancels currently scheduled WSA transmissions and updates WSA broadcast channel accordingly. The module then re-schedules WSAs to broadcast on the '*best channel*' recurrently. If the '*best channel*' is an SCH, the WSAs will be broadcast on the CCH in addition to the SCH for the first 600ms.

Radio interface	Best channel	Time slot

Figure 6.5: Format of the best channel indicator message

The service user model within the WAVE 1609.3 module enables nodes to partake in application services offered by the service providers. The application layer lets the WAVE 1609.3 module know of its intent to participate in an application service through a '*user service request*' and seeks access to necessary radio channels (details are provided in Section 4.3.4.2). A '*user service request*' enables a radio to tune into a particular channel (usually the CCH) to receive WSAs broadcast by the service providers. During this research, the '*user service requests*' were modified to allow the application services to request the WAVE1609.3 module to tune to different DSRC channels instead of a single channel to receive WSAs.

The format of the '*user service request*' is shown in Figure 6.6, wherein the parameter 'channel hopping' indicates the WAVE1609.3 module to enable or disable switching into different SCHs sequentially during time slot 1 to receive WSAs. The 'WSA channel' and 'WSA time slot' parameters specify the channel and time slot on which an SU should listen on, generally. Hence, these values are generally set to channel 178 (CCH) and time slot 0, respectively. Upon receiving a '*user service request*', the WAVE 1609.3 module executes the process shown in Figure 4.18 (explained in detail in Section 4.3.4.2) and sends '*change MAC channel*' requests (shown in Figure 6.4) to the MAC layers. As mentioned earlier in this section, the 'device role' field in the '*change MAC channel*' request is set to '0' to indicate the MAC layer that the service user role is active. Furthermore, the 'channel hopping' value is obtained from the received '*user service request*' message.

Action	Provider service identifier	Advertiser identifier	WSA channel	WSA time slot	WSA radio interface	Channel hopping	Service radio interface	Backup service radio interface

Figure 6.6: Format of the user service request

6.3.2 Extending the MAC Layer Model

The functionality of the MAC 1609.4 module was extended to support the multi-channel CR signalling model. According to the developed algorithm, the DSRC radio that broadcasts WSAs should operate in the alternating channel access mode, while switching between the CCH and an SCH during time slot 0 and 1, respectively. The MAC 1609.4 module accepts the channel switching requests ('*change MAC channel*' requests) sent by the WAVE 1609.3 module and provides access to radio channels during the required time slots as shown in Figure 4.28. In order to gain access to the CCH and an SCH alternately, the '*change MAC channel*' requests are sent with 'alternating channel access' enabled. These requests make the MAC layer to update the variables, '*alternating channel access state*', '*next TS0 channel*', '*next TS1 channel*', '*device role*', '*congestion analysis*' and '*channel hopping*', and schedule an event to perform alternating channel switching at the start of the next time slot. While the variables ‘*congestion analysis*’ and ‘*channel hopping*’ are enabled when the service provider role is active, only ‘*channel hopping*’ is enabled for the service user role.

The MAC 1609.4 module executes the process shown in Figure 6.7 at the next time slot boundary if an event is scheduled to achieve alternating channel access. It initially declares the channel busy to halt the back-off countdowns and stop contention on all the queues in the currently active EDCA subsystem. Then, if the value of the ‘*alternating channel access state*’ variable is set to disable, the MAC layer changes to the continuous channel access mode. The process of switching to the continuous channel access mode is described in detail in Section 4.3.5 (Figure 4.28). On the other hand, if the ‘*alternating channel access state*’ variable is set to enable, the MAC layer schedules an event at the start of the next time slot to perform alternating channel switching. Furthermore, if the ‘continuous’ EDCA subsystem is active, the packets in all its queues are removed.

Next, the MAC layer assesses if the service provider role is active and performs different sets of actions depending on the time slot the node is operating in. At the start of time slot 1, the MAC layer calculates the CBR of the channel that the node was tuned into during time slot 0 (usually the CCH) and makes the ‘time slot 1’ EDCA subsystem active. While the variables ‘*congestion analysis*’ and ‘*channel hopping*’ are enabled, the MAC layer tunes into different SCHs sequentially during time slot 1 intervals. If it has tuned into all six SCHs during previous time slot 1 intervals, the SCH having the smallest CBR value is selected, and its CBR is compared against the CBR of the CCH obtained during the previous time slot. If the CBR of the least congested SCH (channel having the smallest CBR) is less than half the CBR of the CCH, the SCH is chosen as the best channel to broadcast WSAs, and a control message (‘*best channel’* indicator) is sent up to the MAC layer to indicate the selected SCH. The MAC 1609.4 module then tunes into the chosen SCH, and the SCH hopping process is ceased. However, if the MAC layer is unable to find an SCH having less than half the CBR of SCH, the sequential SCH hopping procedure is continued.

At the start of the time slot 0, if the MAC layer is operating as a service provider, it calculates the CBR of the channel (usually an SCH) that it was tuned into during time slot 1. The multi-channel CR signalling algorithm is always in the search for a less congested channel to broadcast WSAs. Hence, while an SCH is occupied with broadcasting WSAs, the MAC layer evaluates the data congestion of the CCH to determine if it can move WSA broadcast back to the CCH since the SUs can quickly discover a WSA transmission on the CCH. When the CBR of the CCH becomes 75% of the CBR value of the current SCH, the MAC layer selects the CCH as the best channel to broadcast WSAs and indicates its decision to the WAVE 1609.3 module via a control message (‘*best channel’* indicator). Under these circumstances, the MAC layer enables SCH hopping to rediscover an SCH that is less congested than the CCH.

The MAC layer executes the process shown in Figure 6.8 if the service user role is active. At the start of time slot 1, the MAC layer checks the value of the channel hopping variable and tunes into SCHs sequentially during time slot 1 intervals while the channel hopping setting is enabled. The MAC 1609.4 module records the number of WSAs received during time slot 1 and evaluates the WSA count at the start of the next time slot 0. While channel hopping is active, if any WSAs are received during the preceding time slot 1, the channel hopping is deactivated, and the MAC layer locks on to the SCH currently tuned in (‘*current TS1 channel*’) by setting the variable ‘*receiving WSAs on SCH*’ to ‘true’. Hence, the DSRC radio will alternate between the CCH and the current SCH thereafter. Furthermore, the received WSA count is also reset to zero.

If an SP moves WSA transmission from an SCH back to the CCH, the SUs will no longer receive the WSAs on the SCH that they are locked on. Under these circumstances, the SUs monitor the SCH for six ‘time slot 1’ intervals (the minimum time taken by the SPs to evaluate data congestion on all six SCHs), and if no WSAs are received during that period, SCH hopping is re-enabled and the SCH lock is released by setting the variable ‘*receiving WSAs on SCH*’ to ‘false’.

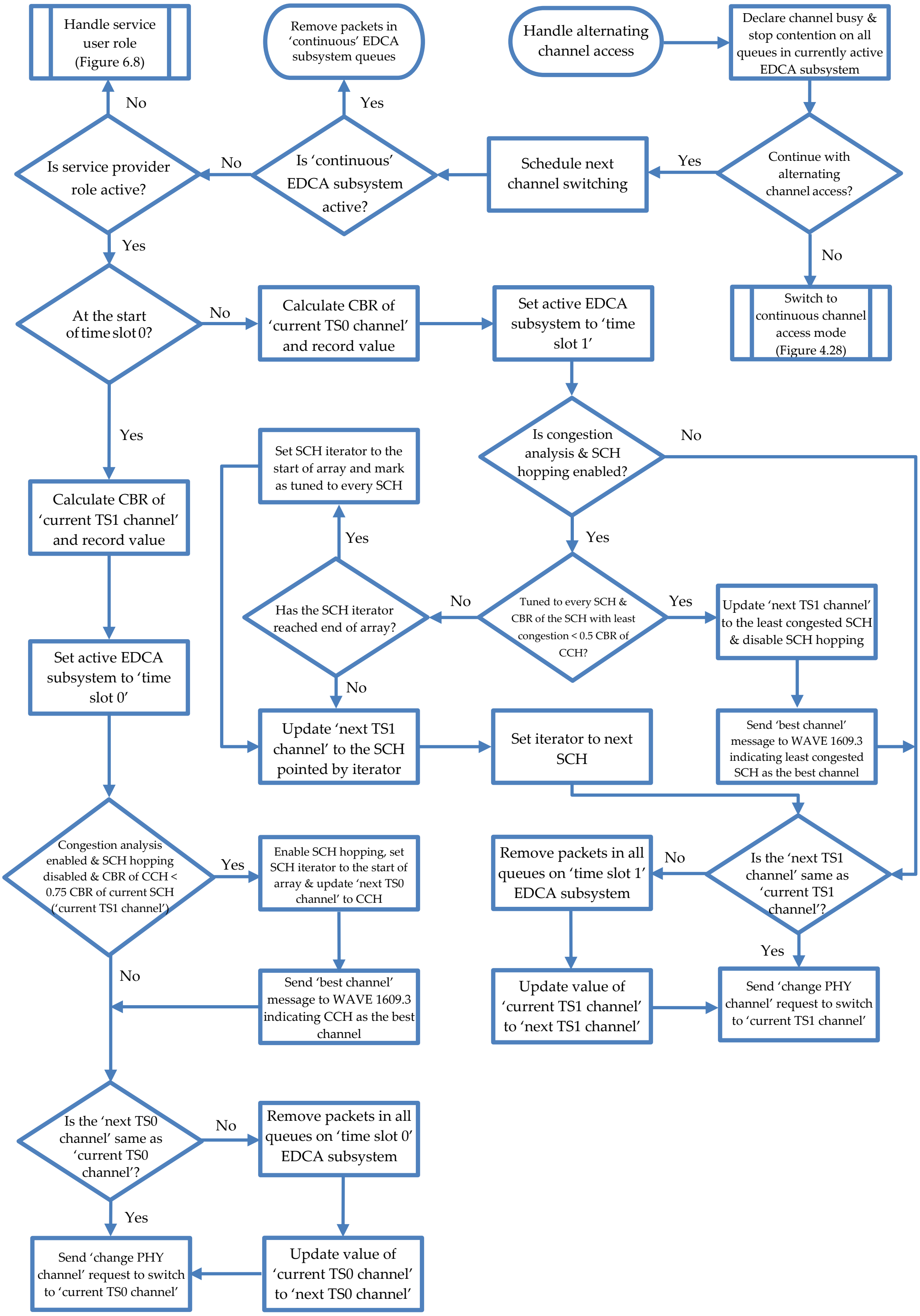


Figure 6.7: The MAC layer handling alternating channel access

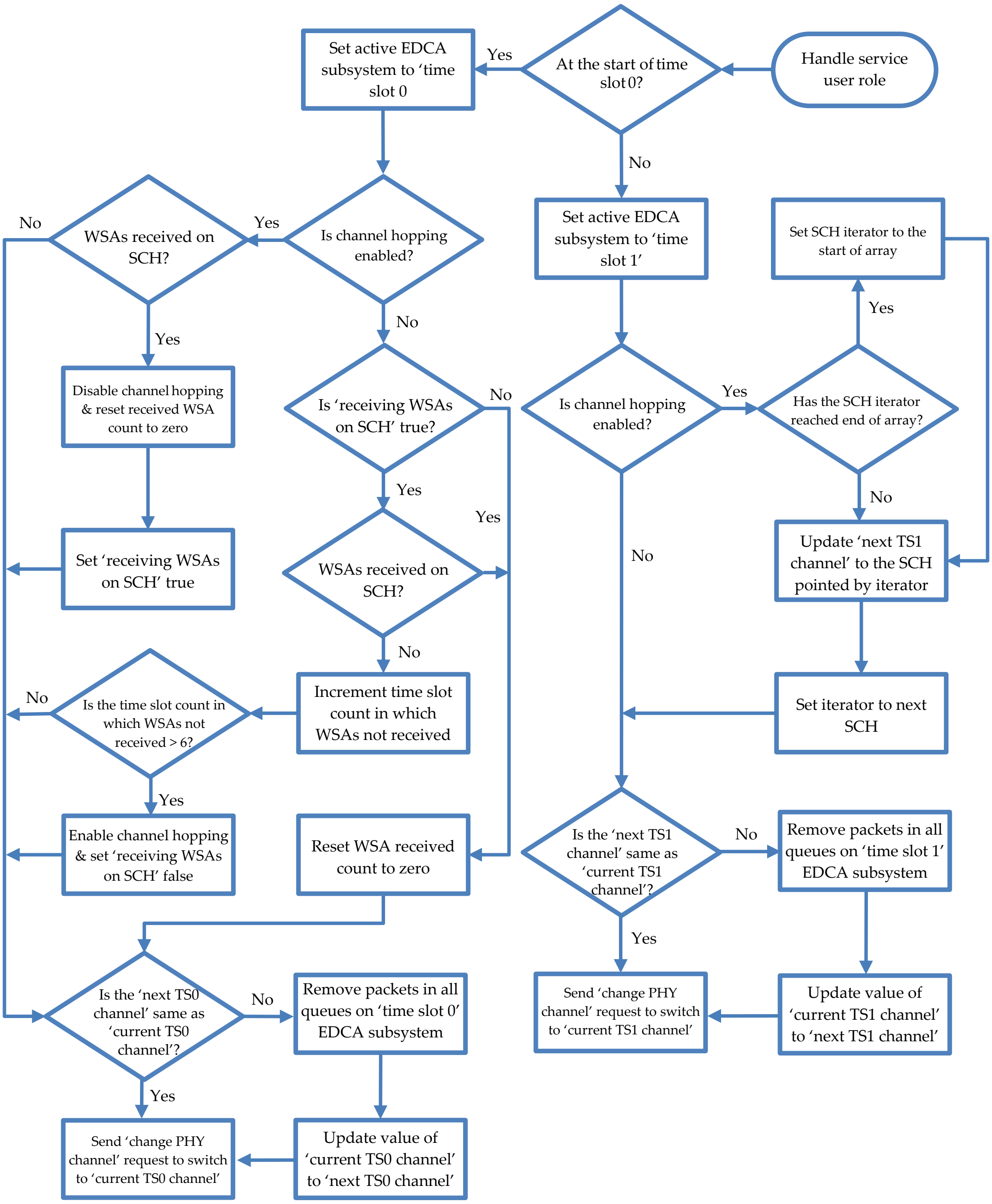


Figure 6.8: The MAC layer handling the service user role

6.4 Performance Evaluation

In order to investigate the performance of the multi-channel CR signalling framework, a 1km highway was prepared on the SUMO road traffic simulator with two vehicular traffic flows configured starting at each end of the road. The 1km road segment was constructed with four lanes in each direction to accommodate more vehicles. Also, in order to perform simulations avoiding border effects (as described in Section 4.3.2), the simulation area (playground) was modelled as the surface of a torus. Moreover, to compare the performance of the developed multi-channel CR signalling framework against the single-channel CR signalling framework (used in Chapter 5), two different vehicular communication scenarios were prepared.

In the first scenario, the single-channel CR signalling framework was evaluated. In this scenario, vehicles were configured to broadcast a WSM on the CCH during time slot 0 intervals. Additionally, one of the vehicles was set up to be an SP that broadcasts WSAs on the CCH at a 'repeat rate' of 10 WSAs per second indicating the availability of an application service offered on a channel in the TVWS. The vehicles configured to broadcast WSMs were also set up to be the SUs that receive WSAs. Moreover, the channel busy ratios (CBRs) of the CCH were obtained from the service providing vehicle at the end of time slot 0 intervals and the packet reception ratios (PRRs) were obtained from the SU vehicles. The number of vehicles broadcasting WSMs were varied, and the simulation was repeated 1000 times for each instance. Subsequently, the average CBR of the CCH and the average PRR were calculated for different numbers of vehicles. The data collected during the first scenario is represented in Figure 6.9, Figure 6.10 and Figure 6.11. It can be clearly seen from Figure 6.11 that in the single channel CR signalling framework, a significant packet loss incurs when the CCH is utilised for more than half of the time slot 0 interval (i.e. average $CBR_{CCH} > 50\%$).

In order to evaluate the multi-channel CR signalling framework, in the second scenario, the vehicles were again configured to broadcast a WSM on the CCH during time slot 0 intervals. Next, thirty RSUs having six radio interfaces on each were placed along the edges of the road (15 RSUs on each side of the road). Each radio interface on an RSU operates in the alternating channel access mode accessing SCHs as shown in Table 6.1.

All thirty RSUs were configured to broadcast a WSM during time slot 1 intervals on SCHs 2 to 6 to make the average CBR of each channel as high as approximately 80% (according to Figure 6.9) and prevent the channel selection algorithm from choosing an SCH other than SCH 1 for broadcasting WSAs. Hence, the reason for placing a high number of RSUs along the edges of the road. The WSAs in this situation are also broadcast on the CCH at the same repeat rate as the first scenario by the service providing vehicle. The developed channel selection algorithm was then applied to identify a better channel for WSA broadcast automatically. Next, the average CBR of the CCH was varied for specific average CBR values of SCH 1 by varying the number of vehicles communicating on the CCH and the number of RSUs communicating on the SCH 1. At each instance, the simulation was repeated for 1000 times to calculate the average PRR. The parameters pertinent to the simulation are summarised in Table 6.2.

Table 6.1: Alternating channel access of each radio interface on RSUs

Radio Interface	Time Slot 0	Time Slot 1
1	SCH 1	SCH 6
2	SCH 2	SCH 5
3	SCH 3	SCH 4
4	SCH 4	SCH 3
5	SCH 5	SCH 2
6	SCH 6	SCH 1

Table 6.2: Simulation parameters

Simulation Parameters	Values
Propagation model	Path loss
MAC bit rate	6Mbps
WSM packet size	1400bytes
Radio transmit power	20mW
Radio sensitivity	-89dBm
Thermal noise	-110 dBm
WSA repeat rate	10WSA/second

Figure 6.12 shows the simulation results obtained using the multi-channel CR signalling framework. A significant improvement in PRR, which is very abrupt, is observed when an SCH can be selected where the congestion is at least 50% less than the CCH. The broadcasting is then moved to the chosen SCH (which is SCH 1), and hence the improvement in the PRR. For example, when the CCH congestion is 80%, an SCH with 40% or less congestion is chosen (SCH 1) and used for the WSA broadcast instead of the CCH. At this point, the PRR jumps from 50% to 95.5%, as seen in Figure 6.12, which is a 91% improvement.

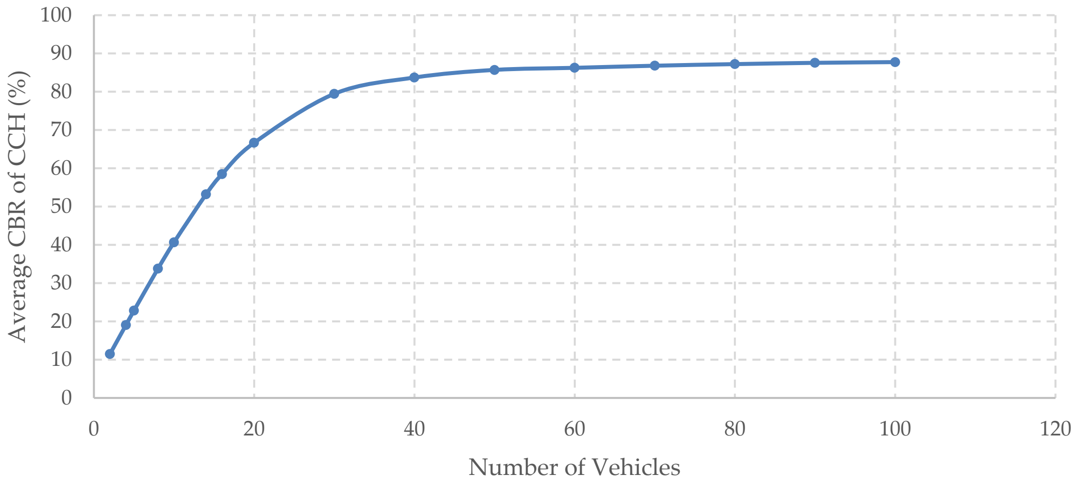


Figure 6.9: The average channel busy ratio of CCH against the number of vehicles

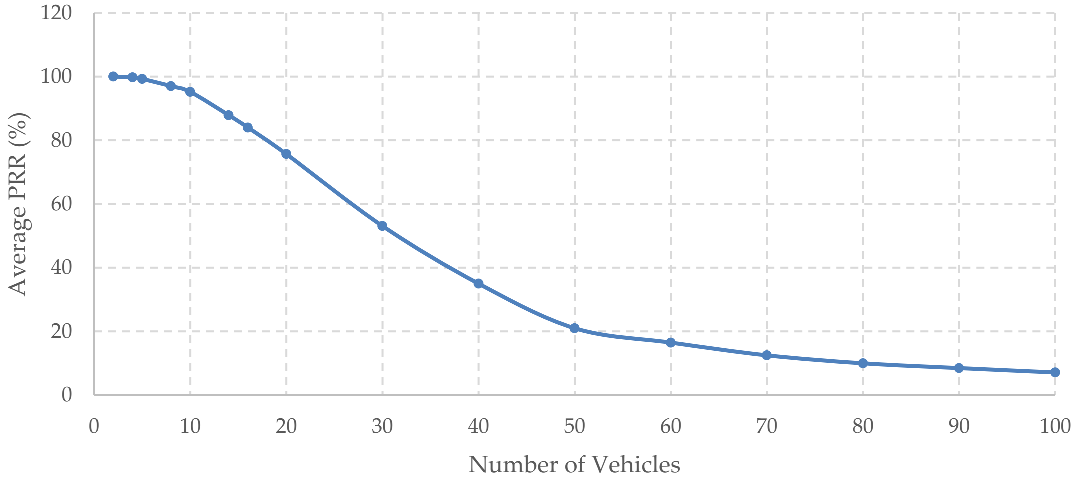


Figure 6.10: The average packet reception ratio against the number of vehicles

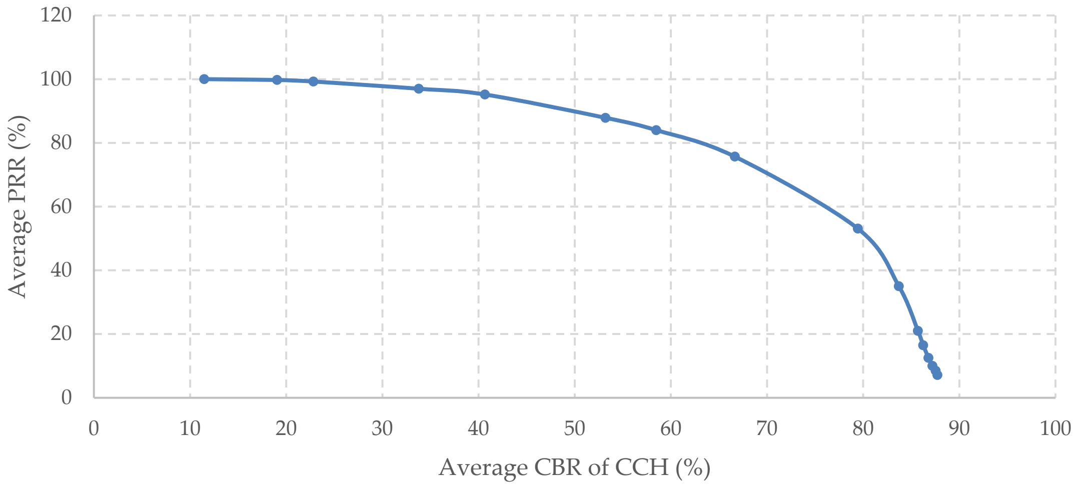


Figure 6.11: The average packet reception ratio against the average channel busy ratio of CCH

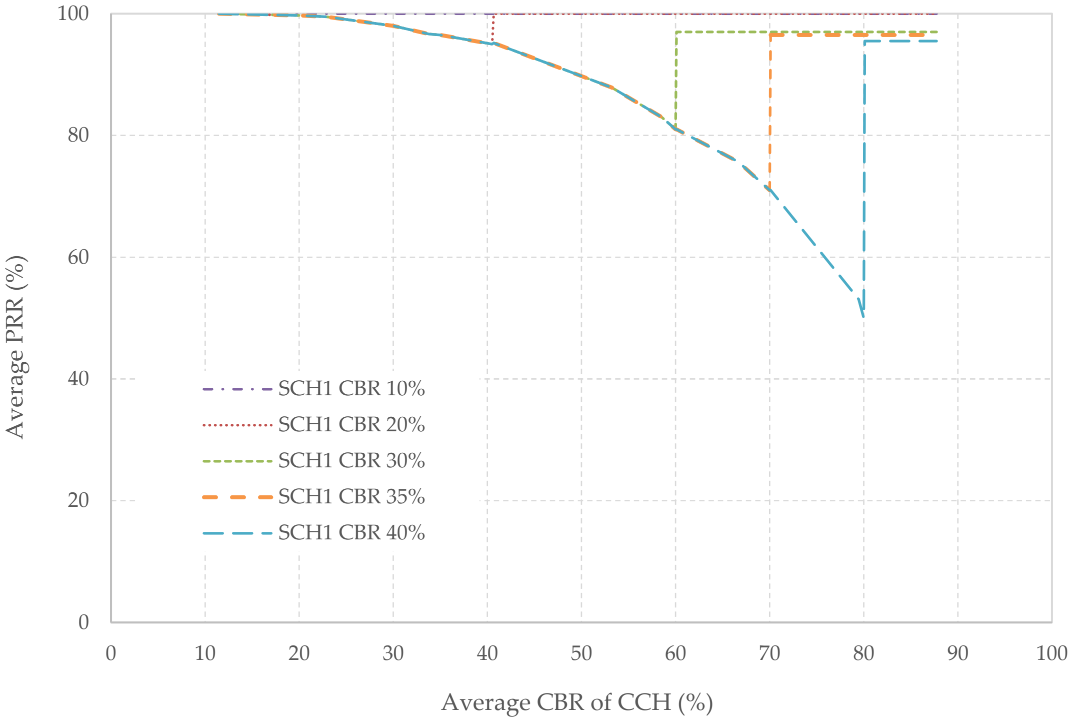


Figure 6.12: The average packet reception ratio against the average channel busy ratio of CCH when the best channel selection algorithm is applied

CHAPTER 7

Conclusions

7.1 General Conclusions

This thesis contains research and development work conducted on the effective utilisation of cognitive radio techniques in vehicular ad hoc networks while conforming to the IEEE WAVE standards. The focus of this work was on delivering the following features as the major outcomes of this PhD project:

- Development of a simulation framework for application service management in VANETs
- Development of a framework for simulating CR enabled VANETs
- Development of a sensing technique to distinctively identify licensed users and secondary network users
- Development of an algorithm to continuously evaluate the channel congestion in the DSRC spectrum for effective utilisation of these channels in the transmission of CR related control signals
- Extending the CR-VANET simulation framework to model the above multi-channel control signal transmission algorithm

Chapter 1 discussed the need for using the CR technology in VANETs. Various applications of VANETs require high data throughputs, especially for non-safety applications such as convenience and entertainment for travellers. Although dedicated bandwidth has been made available for VANETs, spectrum resources are barely adequate for vehicular safety applications alone. The need for more spectrum resources to cater to the high throughput demands is evident. The CR technology makes use of the under-utilised existing spectrum resources through opportunistic spectrum access.

An initial review of the literature presented in Chapter 2 investigated definitions of cognitive radios to understand the nature of capabilities and functionalities expected. Literature on cognitive radio cycle, spectrum sensing techniques and extending these techniques to CR enabled VANETs, rendezvous in CR networks, simulating CR networks and VANETs, and CR standards were reviewed. It was concluded that there are limitations in applying CR techniques to VANETs. In highly mobile vehicular networks, spectrum sensing needs to be fast to track primary user signals. Energy detection based sensing is a viable solution for identifying primary user transmissions in CR-VANETs due to fast signal detection, simple design and low computational and implementation complexities. However, most of the existing energy detection implementations require a forced quiet sensing period to identify a primary user transmission distinctively. A few studies have identified primary and secondary user signal transmissions using energy detection in CR-VANETs without needing this forced quiet sensing period, but this area of study has not been thoroughly investigated. Various common control channel designs to achieve rendezvous in a CR network have been discussed. However, most of these are better suited for cognitive radio networks with stationary nodes and not for vehicular networks with mobile nodes.

Integrated CR-VANET simulators characterising radio signal propagation, CR functions, vehicular mobility and vehicular communication protocols are not available to the research community. However, a limited number of simulators model VANETs as well as stationary CR networks when these networks are considered individually. A limited number of models have been proposed for CR networks, but only the implementation of *crSimulator* is publicly available, yet it is not suited for VANETs due to many limitations. Therefore, a need for a simulation framework which can model characteristics of both CR networks and VANETs was identified.

Chapter 3 detailed the different layers of the IEEE WAVE communication protocol stack on which the proposed work was carried out. The network simulator OMNeT++ was chosen to develop the VANET and CR simulation model. The *Veins* vehicular network simulation framework was employed as a base model which works in conjunction with the road traffic simulator SUMO.

Chapter 4 described the design and development of the simulation framework for application service management in VANETs. The existing simulators do not support simulation of complex application scenarios in VANETs. The developed simulation framework supports modelling of application service management in VANETs as per the IEEE WAVE standards. This simulation framework developed on '*OMNeT++*' extended the '*Veins*' vehicular network simulation framework described in Chapter 3. The *Veins* framework was extended to model multi-radio multi-channel communications, application service roles, application service advertisement and discovery, simultaneous application service operation, data transfers over distinct radios channels and channel access mechanisms. Simulations were carried out using a vehicle serving as a service provider with dual radios, of which, one operates in alternating channel access mode and the other in continuous channel access mode. When a channel change request is made, the radio operating in continuous channel access mode changes over immediately. The other radio responds in the next immediate time slot relevant to the channel that should change. These results generated by the developed simulation framework were exactly as expected.

Chapter 5 described the details of the developed simulation framework for cognitive radio enabled VANETs. This framework simulates mobile (vehicles) and stationary CR nodes, primary user transmissions, CR based application service advertisement and discovery, spectrum sensing and spectrum hand-offs, and also identifies primary and secondary user transmissions distinctively. The multi-radio multi-channel radio environment required for modelling CR-VANETs was based on the foundation laid by the service management framework described in Chapter 4.

Existing spectrum sensing algorithms intend to detect the existence of a radio signal on a channel through energy detection, using a two-state sensing model, which requires periodic channel silence phases to identify a primary user transmission. This two-state sensing model is not applicable for ad-hoc networks such as VANETs as no central entity is available to coordinate sensing schedules. Hence, a three-state spectrum sensing model was developed during this research to distinguish between the primary and secondary user signals without employing mandatory channel silence phases. The three-state sensing model uses a carrier sensing mechanism in addition to energy detection.

When only a primary user was present, the simulation results showed that a higher probability of detection can be obtained with a sensing interval shorter than the transmission signal duration of the primary user. Furthermore, the highest probability of detection was recorded when the channel was sensed at 10ms intervals. When secondary users were only present the probability of detection deteriorates with the increasing number of transmitting vehicles. When secondary users compete for channel access, collisions occur among transmissions, leading to falsely perceiving these as occupied by a primary user. Upon detecting energy, the sensing model attempts to confirm that it is not due to a primary user, by repeatedly sensing the channel. For example, the probability of detection improved by 76% when the number of maximum sensing intervals was increased from 2 to 10 with 40 vehicles accessing the channel. Although reducing the sensing interval improves the probability of detection it adds to system overheads. Hence, a dynamic sensing technique was employed as an attempt to reduce the system overheads without compromising the accuracy of sensing. The results indicated that the probability of detection remained unchanged when dynamic sensing was adopted, but significantly reduced the system's overheads up to 90%.

A channel is required to exchange control information between neighbouring CR nodes. These channels are employed to discover services offered by other CR devices, establish and maintain peer-to-peer connections, exchange spectrum sensing information or switch between various dynamic spectrum access bands. Recent CR-VANET related studies have utilised the CCH as a common channel for cognitive radio related signalling purposes. Although a dedicated common control channel in the DSRC is highly desirable, this can also be the single point of failure when the channel becomes congested with high packet collisions. Therefore, in Chapter 6, a system was presented where multiple channels in the DSRC spectrum are employed for CR related control information exchange. The developed algorithm continuously evaluates the data congestion of all channels in the DSRC spectrum to choose the least congested channel for CR signalling. A complete simulation framework was also developed on OMNeT++ to model the designed algorithm. The simulation results obtained using the multi-channel CR signalling framework showed a significant improvement in the packet reception ratio. This was observed when an SCH was able to be selected with at least 50% less congestion than the CCH, and the control information broadcast was moved to the chosen SCH. The packet reception ratio increased up to 95.5% showing improvements as high as 91% in comparison to using a single channel.

7.2 Engineering Implications

The system developed in this research is a complete solution dedicated to increasing data throughputs for vehicular communication by making use of under-utilised spectrum resources through cognitive radio techniques. This system mainly caters towards communicating non-safety applications such as providing entertainment and convenience to travellers. This may be in the form of advertising, sharing information, and entertainment applications between travellers among a group of vehicles. Nevertheless, such applications use up spectrum resources allocated for safety communications. The developed system makes bandwidth available for non-safety applications through opportunistic spectrum access effectively and alleviates data congestion on channels dedicated to vehicular safety communications.

This system allows developers to use it as a V2V communication platform which conforms with the IEEE WAVE standard, for a variety of different applications:

- Informing drivers about nearby parking spots and parking rates
- Promotional offers from nearby businesses, such as restaurants, supermarkets, pop-up food vendors and fuel stations
- General information such as up to date weather and traffic updates
- Instant messaging and voice platform for a convoy of vehicles in the absence of mobile network
- Running entertainment applications among a convoy of vehicles such as playing games, sharing music and movies

The key difference here is that it does not load the public mobile network by using a vehicular ad hoc network. This is especially important at locations where the mobile networks are not present. The conformance with the IEEE WAVE standard makes it more desirable for hardware and software developers as it has a common ground for all applications. Such applications can be marketed to vendors, service providers and mobile device users.

An established V2V communication network inspires equipment manufacturers to use this technology for new developments. It can be adopted as a new technology on mobile devices, such as 'V2V communication enabled', which can be a highly marketable aspect. If the utilisation of this system is increased, the running cost of it will be reduced significantly, making V2V communication devices more available and affordable. With the increased use of the system, software developers will inevitably be encouraged to create more applications to promote V2V.

The developed simulation frameworks provide an excellent tool for the research and industrial communities alike. The lack of such a platform has set back the research community wanting to progress in this aspect. These frameworks can be used to advance research and development in the utilisation of cognitive radio technology in VANETs. They encourage the use of vehicular communications for non-safety purposes. Service providers are also motivated to use vehicular communication as a form of advertising with a well established vehicular communication system.

Another attractive attribute of the developed system is that it can be directly implemented on hardware without the need for modification. As the framework has addressed all aspects with attention to finer details, it is a very user-friendly platform for all firmware and application developers. This is an excellent platform for equipment developers to speed up the process of implementing hardware for vehicular communication applications.

The developed system significantly improves the efficiency of local spectrum resources through cognitive radio technology. However, cognitive radio technologies introduce extra overheads on hardware and dedicated vehicular spectrum resources. This system significantly reduces these overheads on hardware through dynamic spectrum sensing, reducing the need for more powerful hardware. Therefore, the system is 'ready to use' on existing hardware making it highly desirable by device developers, and leading to cheaper and faster implementation of this technology. The efficient use of dedicated vehicular spectrum resources for cognitive radio technology makes it highly acceptable by spectrum resource allocation authorities, as it implies that dedicated spectrum resources are not overloaded while a new communication platform is being introduced.

7.3 Recommendations for Future Work

The developed simulation frameworks provide a highly powerful tool for the research and industrial communities alike, to conduct research and development in vehicular communication and cognitive radio technologies. Several future works can be recommended to enhance and utilise the frameworks to their full potential.

1. Implement the higher layers of the IEEE WAVE communication protocol stack
   In the current implementation, the TCP/IP layers have not been implemented. However, when simulating real-world vehicular communication applications, the TCP/IP layers will be required to a great extent. These additional layers will help to accurately model and analyse the performance of such applications. Future researchers working on these frameworks can look into implementing the TCP/IP layers to enhance the developed framework.

2. Implement the functions to enable and disable radio interfaces as desired
   Each node using multiple radio interfaces for application services currently have each radio interface active throughout the operation. This setup is not very energy efficient and to mitigate this problem the radio interfaces need to be activated on demand. The ability to turn off radio interfaces that are not needed at the time would also be greatly beneficial to save energy, especially with electric vehicles gaining popularity.

3. Automatically assign radio interfaces for application services by service users
   Currently, when a service user receives a WAVE service advertisement, a radio channel is allocated for application services on a radio interface defined at the start of a simulation run. This is not ideal for a real-life scenario, and it would be

a lot more practical to have the radio interface assigned by the service user automatically depending on the availability of radio channel resources to the service user at the time.

4. Investigate methods to mitigate undetected interference issues

   Currently, the '*Connection Manager*' module in the simulation framework maintains and updates wireless connectivity information between radio interfaces on various nodes. Based on the distances between radio interfaces at discrete time intervals, connections are established when they are within the maximal interference distance. Although this technique reduces the computational complexities of wireless communication simulations, this can potentially lead to interference issues. When a nearby node, just outside the communication range of another transmitting node, enters the communication range while transmitting on the same channel unknowingly, interference will be experienced by the other node. It is recommended that future researchers analyse and investigate methods to resolve this issue to model the situation more accurately.

5. Implementing an adaptive threshold level in switching between control and service channels for cognitive radio control signal transmissions

   At present, the decision is made to switch from the control channel to the service channel if the service channel is 50% less congested than the control channel. This works well when the control channel is experiencing high levels of congestion such as 80%. However, this method is not ideal when the congestion levels of the control and service channels are at about 10% and 3% respectively. The control channel itself has very little congestion, and switching is not realistically needed. Therefore, an adaptive channel switching algorithm which can make more sensible choices when alternating between channels is highly desirable and recommended as future work.

Appendix A

A.1 Network Description Coding Examples

The structure of a simulation model is programmed through the language called NED (Network Description). The NED codes for the compound modules, vehicle and RSU, are given below.

```
//The compound module for a vehicle node
module Car
{
    parameters:
       string applType; //type of the application layer
       string veinsmobilityType; //type of the mobility module
       int countRadio = default(1); //Number of radios

    submodules:
       appl: <applType> like simuCRV.base.modules.IBaseApplLayer
       wave: Wave1609_3

       //submodule type name defined via typename pattern assignments
       radio[countRadio]: <default("DSRCRadio")> like  simuCRV.modules.radio.IRadio

       veinsmobility: <veinsmobilityType> like simuCRV.base.modules.IMobility

    connections:
       for i=0..countRadio-1 {
            radio[i].upperLayerOut --> wave.lowerLayerInVector++;
            radio[i].upperLayerIn <-- wave.lowerLayerOutVector++;
            radio[i].upperControlOut --> wave.lowerControlInVector++;
            radio[i].upperControlIn <-- wave.lowerControlOutVector++;
        }

            wave.upperLayerOut --> appl.lowerLayerIn;
            wave.upperLayerIn <-- appl.lowerLayerOut;
            wave.upperControlOut --> appl.lowerControlIn;
            wave.upperControlIn <-- appl.lowerControlOut;
}

//The compound module for an RSU node
module RSU
{
    parameters:
       string applType; //type of the application layer
       int countRadio = default(1); //Number of radios

    submodules:
       appl: <applType> like simuCRV.base.modules.IBaseApplLayer
       wave: Wave1609_3

       //submodule type name defined via typename pattern assignments
       radio[countRadio]: <default("DSRCRadio")> like simuCRV.modules.radio.IRadio
       mobility: BaseMobility

    connections:
        for i = 0..countRadio-1 {
            radio[i].upperLayerOut --> wave.lowerLayerInVector++;
            radio[i].upperLayerIn <-- wave.lowerLayerOutVector++;
            radio[i].upperControlOut --> wave.lowerControlInVector++;
```

```
            radio[i].upperControlIn <-- wave.lowerControlOutVector++;
     }

            wave.upperLayerOut --> appl.lowerLayerIn;
            wave.upperLayerIn <-- appl.lowerLayerOut;
            wave.upperControlOut --> appl.lowerControlIn;
            wave.upperControlIn <-- appl.lowerControlOut;
}
```

A radio module (compound) consists of a physical layer and a medium access control layer. Two types of radio modules were created during this research to operate in the DSRC and TVWS bands. The corresponding NED codes for the DSRC and TVWS radios are given below.

```
//The compound module for a DSRC radio
module DSRCRadio like IRadio
{
    parameters:
       string connectionManagerName = default("DSRCChannel");

    gates:
       input upperLayerIn;
       output upperLayerOut;
       output upperControlOut;
       input upperControlIn;
       input radioIn @directIn;

    submodules:
       phy80211p: Phy80211p
       mac1609_4: Mac1609_4

    connections:

       radioIn --> phy80211p.radioIn;

       mac1609_4.lowerControlOut --> phy80211p.upperControlIn;
       mac1609_4.lowerLayerOut --> phy80211p.upperLayerIn;
       phy80211p.upperLayerOut --> mac1609_4.lowerLayerIn;
       phy80211p.upperControlOut --> mac1609_4.lowerControlIn;

       mac1609_4.upperControlIn <-- upperControlIn;
       mac1609_4.upperLayerIn <-- upperLayerIn;
       mac1609_4.upperLayerOut --> upperLayerOut;
       mac1609_4.upperControlOut --> upperControlOut;
}

//The compound module for a TVWS radio
module TVWSRadio like IRadio
{
    parameters:
       string connectionManagerName=default("TVWSChannel");

    gates:
       input upperLayerIn;
       output upperLayerOut;
       output upperControlOut;
       input upperControlIn;
       input radioIn @directIn;

    submodules:
       phy80211p: Phy80211p
       CR_Mac1609_4: CR_Mac1609_4
```

```
    connections:
        radioIn --> phy80211p.radioIn;

        CR_Mac1609_4.lowerControlOut --> phy80211p.upperControlIn;
        CR_Mac1609_4.lowerLayerOut --> phy80211p.upperLayerIn;
        phy80211p.upperLayerOut --> CR_Mac1609_4.lowerLayerIn;
        phy80211p.upperControlOut --> CR_Mac1609_4.lowerControlIn;

        CR_Mac1609_4.upperControlIn <-- upperControlIn;
        CR_Mac1609_4.upperLayerIn <-- upperLayerIn;
        CR_Mac1609_4.upperLayerOut --> upperLayerOut;
        CR_Mac1609_4.upperControlOut --> upperControlOut;
}
```

A.2 Message Definition Examples

The codes declaring the message structures are written in plain-text message definition files called msg. The message definition file for the provider service request is given below.

```
enum ProviderServiceRequestMessageKinds {
        //Indicates the action requested by the provider service request
        PROVIDER_START  = 1;
        PROVIDER_CHANGE = 2;
        PROVIDER_STOP   = 3;
        };

message ProviderServiceRequestMessage {
        int providerServiceID = 10;
        string advertiserID = "uoa";
        int serviceChannelNumber = 174;
        int channelAccess = 1;
        int serviceRadioInterface = 0;
        int wsaChannel = 178;
        int wsaTimeSlot = 0;
        int wsaRadioInterface = 0;
        int repeatRate = 50;
        }
```

The message definition file for the user service request is as follows.

```
enum UserServiceRequestMessageKinds {

        //Indicates the action requested by the user service request
        USER_ADD   = 1;
        USER_REMOVE = 2;
        };

message UserServiceRequestMessage {

        int providerServiceID = 10;
        string advertiserID = "uoa";
        int wsaChannel = 178;
        int wsaTimeSlot = 0;
        int wsaRadioInterface = 0;
        int serviceRadioInterface = 0;
        int backupServiceRadioInterface = 1;
}
```

A.3 C++ Programming Examples

The simple modules in the simulation framework are implemented using the C++ programming language. Several C++ codes written for the application service management framework described in Chapter 4 are presented here below.

A.3.1 Identifying Gate Vectors

The gate vectors supporting multiple connections are used in the WAVE 1609.3 module to connect with the gates in the radio modules. The following C++ code obtains the gate identification numbers of all connected gates.

```
// Gate IDs of vector gates.
int *lowerLayerInVector;
int *lowerLayerOutVector;
int *lowerControlInVector;
int *lowerControlOutVector;

lowerLayerInVector = new int[gateSize("lowerLayerInVector")];
lowerLayerOutVector = new int[gateSize("lowerLayerOutVector")];
lowerControlInVector = new int[gateSize("lowerControlInVector")];
lowerControlOutVector = new int[gateSize("lowerControlOutVector")];

/* Module gates are represented by cGate objects. cGate objects know to which other
gates they are connected. */

//Enumerating gates

for (cModule::GateIterator i(this); !i.end(); i++) {

        cGate *gate = i();
        if (!gate->isVector()){
                ev << "scalar gate, " << endl;
        }

        //A vector gate

        else if      (std::string(gate->getName())== "lowerLayerInVector") {
                lowerLayerInVector[gate->getIndex()] = gate->getId();
        }
        else if (std::string(gate->getName())== "lowerLayerOutVector") {
                lowerLayerOutVector[gate->getIndex()] = gate->getId();
        }
        else if (std::string(gate->getName())== "lowerControlInVector") {
                lowerControlInVector[gate->getIndex()] = gate->getId();
        }
        else if    (std::string(gate->getName())== "lowerControlOutVector") {
                lowerControlOutVector[gate->getIndex()] = gate->getId();
        }
        else { opp_error("The gate %s does not match any of the configured gates on the
        application module",gate->getName());
        }
}
```

A.3.2 Clear Channel Assessment

```
bool Decider::cca(simtime_t_cref time, AirFrame* exclude) {
       AirFrameVector airFrames;

       //Get all Air Frames intersecting within the time interval [start, end]
       getChannelInfo(time, time, airFrames);

       Mapping* resultMap = MappingUtils::createMapping(Argument::MappedZero,
                      DimensionSet::timeDomain);

       /* Select all signals having the same frequency as the physical layer frequency
          and sum up their receiving power levels */

       for (AirFrameVector::const_iterator it = airFrames.begin();
                                                   it != airFrames.end(); ++it) {

       /* Skip the Air Frame if iterator points to exclude (including the default-case
       'exclude == 0') */

           if (*it == exclude) {
                      continue;
           }
           // Iterator should not point to 0
           assert(*it != 0);

           // Obtain the signal
           Signal& signal = (*it)->getSignal();

           /* Get the receiving power of the Signal at start-time and centre frequency
           (the frequency that the physical layer is tuned into) */

           Argument start(DimensionSet::timeFreqDomain);
           start.setTime(signal.getReceptionStart());
           start.setArgValue(Dimension::frequency_static(), centreFrequency);
           double recvPower = signal.getReceivingPower()->getValue(start);

           /* If the Signal is being received on the channel the physical layer is
           tuned into, add its power mapping to the resultMap */

           if (recvPower > 0) {

               const ConstMapping* const recvPowerMap = signal.getReceivingPower();
               assert(recvPowerMap);

               Mapping* resultMapNew = MappingUtils::add(*recvPowerMap, *resultMap,
                                                    Argument::MappedZero);
               // discard old mapping
               delete resultMap;
               resultMap = resultMapNew;
               resultMapNew = 0;

           }
       }

       // Add in thermal noise if it is to be simulated
       ConstMapping* thermalNoise = phy->getThermalNoise(time, time);

       if (thermalNoise) {
               Mapping* tmp = resultMap;
               resultMap = MappingUtils::add(*resultMap, *thermalNoise);
               delete tmp;
       }
```

```
        Argument min(DimensionSet::timeFreqDomain);
        min.setTime(time);

        min.setArgValue(Dimension::frequency_static(), centerFrequency);

        *The resulting power level is compared against a chosen CCA threshold. Channel
        is idle if the power level is lower than the threshold. */

        bool isChannelIdle = MappingUtils::findMin(*resultMap,min,min) < ccaThreshold;

        delete resultMap;
        return isChannelIdle;

}
```

A.3.3 Querying MAC Sub-Modules

At the start of every simulation, the `findMacModules(const cModule *const)` function in the WAVE 1609.3 module queries the MAC sub-modules of each connected radio and creates a mapping (*'radio interface to time slot and channel'* mapping) to keep track of the connected radio interfaces and their initial radio channel assignments.

```
Wave1609_3ToMac1609_4Interface* Wave1609_3::findMacModules(const cModule *const top)
{
  int interfaceNumber = 0;

    for (cModule::SubmoduleIterator i(top); !i.end(); i++) {

        cModule *const sub = i();
        Wave1609_3ToMac1609_4Interface * castMac = dynamic_cast
                                             <Wave1609_3ToMac1609_4Interface *> (sub);
        if (castMac != NULL)       {

           interfaceToTimeSlotChannelMapping.insert(std::pair<int,std::map<int,int>
           >(interfaceNumber,castMac->getTimeSlotToChannelMapping()));

           interfaceNumber++;
         }

        findMacModules(sub);

    }

    return NULL;
}
```

A.3.4 Creating EDCA Queuing System

The developed MAC 1609.4 module contains three ‘time slots’ based EDCA queuing systems (‘time slot 0’, ‘time slot 1’ and ‘continuous’). The C++ code for creating these queues is given below.

```cpp
//Create timeslot 0 EDCA subsystem
myEDCA[type_TS0] = new EDCA(type_TS0,par("queueSize").longValue());
myEDCA[type_TS0]->createQueue(2,(((CWMIN+1)/4)-1),(((CWMIN +1)/2)-1),AC_VO);
myEDCA[type_TS0]->createQueue(3,(((CWMIN+1)/2)-1),CWMIN,AC_VI);
myEDCA[type_TS0]->createQueue(6,CWMIN,CWMAX,AC_BE);
myEDCA[type_TS0]->createQueue(9,CWMIN,CWMAX,AC_BK);

//Create timeslot 1 EDCA subsystem
myEDCA[type_TS1] = new EDCA(type_TS1,par("queueSize").longValue());
myEDCA[type_TS1]->createQueue(2,(((CWMIN+1)/4)-1),(((CWMIN +1)/2)-1),AC_VO);
myEDCA[type_TS1]->createQueue(3,(((CWMIN+1)/2)-1),CWMIN,AC_VI);
myEDCA[type_TS1]->createQueue(6,CWMIN,CWMAX,AC_BE);
myEDCA[type_TS1]->createQueue(9,CWMIN,CWMAX,AC_BK);

//Create continuous EDCA subsystem
myEDCA[type_Continuous] = new EDCA(type_Continuous,par("queueSize").longValue());
myEDCA[type_Continuous]->createQueue(2,(((CWMIN+1)/4)-1),(((CWMIN +1)/2)-1),AC_VO);
myEDCA[type_Continuous]->createQueue(3,(((CWMIN+1)/2)-1),CWMIN,AC_VI);
myEDCA[type_Continuous]->createQueue(6,CWMIN,CWMAX,AC_BE);
myEDCA[type_Continuous]->createQueue(9,CWMIN,CWMAX,AC_BK);
```

References

[1] Intelligent Transport Systems Australia. (2014). *About | ITS Australia*. Available: https://www.its-australia.com.au/about/.

[2] ITS United Kingdom. (2014). *About Us | ITS-UK*. Available: http://www.its-uk.org.uk/about/aboutus.

[3] S. Ezell, "Explaining international IT application leadership: Intelligent transportation systems," The Information Technology & Innovation Foundation, Jan. 2010.

[4] World Health Organization. (2013, July). *The top 10 causes of death*. Available: http://www.who.int/mediacentre/factsheets/fs310/en/.

[5] Ministry of Health. Mortality and demographic data 2012. Ministry of Health. Wellington, New Zealand. 2015.

[6] W. G. Najm *et al*, "Frequency of target crashes for IntelliDrive safety systems," National Highway Traffic Safety Administration, Washington, DC, Tech. Rep. DOT HS 811381, Oct. 2010.

[7] T. Green *et al*, "Safety pilot fact sheet: Improving safety and mobility through connected vehicle technology," National Highway Traffic Safety Administration, 2012.

[8] D. Schrank, B. Eisele and T. Lomax, "TTI's 2012 urban mobility report," Texas A&M Transportation Institute, Dec. 2012.

[9] I. P. Wallis and D. R. Lupton, "The costs of congestion reappraised," NZ Transport Agency, Wellington, New Zealand, Tech. Rep. 489, Feb. 2013.

[10] Department of Transportation, "Transforming transportation through connectivity: ITS strategic research plan, 2010–2014 (progress update, 2012)," U.S. Department of Transportation, Washington, DC, Tech. Rep. FHWA-JPO-12-019, Oct. 2012.

[11] H. Hartenstein and K. Laberteaux, *VANET : Vehicular Applications and Inter-Networking Technologies.* Chichester, U.K.: Chichester, U.K. : Wiley 2010, 2010.

[12] T. L. Willke, P. Tientrakool and N. F. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *Communications Surveys & Tutorials, IEEE,* vol. 11, *(2),* pp. 3-20, 2009.

[13] V. CAMP, "Vehicle safety communications project task 3 final report identify intelligent vehicle safety applications enabled by DSRC," U. S. Department of Transportation, Washington, DC, Tech. Rep. DOT HS 809 859, Mar. 2005.

[14] Federal Communications Commission, "Report and order in the matter of amendment of parts 2 and 90 of the commission's rules to allocate the 5.850–5.925 GHz band to mobile service for dedicated short range communications of intelligent transportation services," Washington, DC, Tech. Rep. FCC 99-305, Oct. 1999.

[15] IEEE, "IEEE Guide for Wireless Access in Vehicular Environments (WAVE) — Architecture," *IEEE Std 1609. 0-2013,* pp. 1-78, Mar, 2014.

[16] The European Parliament, "Decision No 676/2002/EC of the European parliament and of the council of 7 March 2002 on a regulatory framework for radio spectrum policy in the European Community (Radio

Spectrum Decision)," *Official Journal of the European Communities, (L108),* pp. 1-6, 2002. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002D0676&from=EN.

[17] European Commission, "2008/671/EC: Commission Decision of 5 August 2008 on the harmonised use of radio spectrum in the 5875 - 5905 MHz frequency band for safety-related applications of Intelligent Transport Systems (ITS)," *Official Journal of the European Union, (L220),* pp. 24-26, 2008. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32008D0671&from=EN.

[18] Electronic Communications Committee. The harmonised use of the 5875-5925 MHz frequency band for intelligent transport systems (ITS). 2008 Available: https://www.ecodocdb.dk/download/b470d271-048b/ECCDEC0801.PDF.

[19] Electronic Communications Committee. Use of the band 5855-5875 MHz for intelligent transport systems (ITS). 2008 Available: https://www.ecodocdb.dk/download/798c1836-20c6/REC0801.pdf.

[20] The European Conference of Postal and Telecommunications Administrations, "CEPT all about our organisation," 2018. Available: https://cept.org/files/1047/CEPT%20Leaflet_June%202018.pdf.

[21] Association of Radio Industries and Businesses. Dedicated short-range communication systems. 2001 Available: https://www.arib.or.jp/english/html/overview/doc/5-STD-T75v1_0-E2.pdf.

[22] Z. Wang and M. Hassan, "How much of dsrc is available for non-safety use?" in *Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking (VANET '08),* NY, 2008, pp. 23-29.

[23] K. Fawaz *et al*, "Improving reliability of safety applications in vehicle ad hoc networks through the implementation of a cognitive network," in *Telecommunications (ICT), 2010 IEEE 17th International Conference On,* 2010, pp. 798-805.

[24] S. Eichler, "Performance evaluation of the IEEE 802.11p WAVE communication standard," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th,* 2007, pp. 2199-2203.

[25] C. Campolo, A. Molinaro and A. Vinel, "Understanding the performance of short-lived control broadcast packets in 802.11p/WAVE vehicular networks," in *Vehicular Networking Conference (VNC), 2011 IEEE,* 2011, pp. 102-108.

[26] Shared Spectrum Company. *Spectrum Reports 2004-2009*. Available: http://www.sharedspectrum.com/papers/spectrum-reports/.

[27] R. I. C. Chiang, G. B. Rowe and K. W. Sowerby, "A quantitative analysis of spectral occupancy measurements for cognitive radio," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th,* 2007, pp. 3016-3020.

[28] Federal Communications Commission, "Report and order in the matter of facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies," Washington, D.C., Tech. Rep. FCC 05-57, Mar. 2005.

[29] J. Mitola and G. Q. Maguire Jr., "Cognitive radio: making software radios more personal," *Personal Communications, IEEE,* vol. 6, *(4),* pp. 13-18, 1999.

[30] J. Mitola, "Cognitive radio for flexible mobile multimedia communications," in *Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop On,* 1999, pp. 3-10.

[31] J. Mitola, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio." , Royal Institute of Technology (KTH), Stockhom,Sweden, 2000.

[32] Federal Communications Commission, "Second report and order and memorandum and order in the matter of unlicensed operation in the TV broadcast bands, additional spectrum for unlicensed devices below 900 MHz and in the 3 GHz band," Washington, D.C., Tech. Rep. FCC 08-260, Nov. 2008.

[33] IEEE Computer Society, "IEEE Standard for Information technology — Local and metropolitan area networks — Specific requirements — Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and procedures for operation in the TV Bands," *IEEE Std 802. 22-2011,* pp. 1-680, 2011.

[34] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications,* vol. 23, *(2),* pp. 201-220, 2005.

[35] Federal Communications Commission, "Comments of the national telecommunications and information administration in the matter of facilitating opportunities for flexible, efficient and reliable spectrum use employing cognitive radio technologies," Washington, D.C., Tech. Rep. ET Docket No. 03-108, Feb. 2005.

[36] IEEE Communications Society, "IEEE Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management Amendment 1: Addition of New Terms and Associated Definitions," *IEEE Std 1900. 1a-2012 (Amendment to IEEE Std 1900. 1-2008),* pp. 1-21, 2013.

[37] A. Khattab, D. Perkins and M. Bayoumi, *Cognitive Radio Networks: From Theory to Practice.* New York, NY: Springer New York, 2013.

[38] K. Sithamparanathan and A. Giorgetti, *Cognitive Radio Techniques: Spectrum Sensing, Interference Mitigation, and Localization.* Boston, MA: Artech House, 2012.

[39] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials,* vol. 11, *(1),* pp. 116-130, 2009.

[40] A. M. Wyglinski, M. Nekovee and Y. T. Hou, *Cognitive Radio Communications and Networks:Principles and Practice.* Burlington, MA: Academic Press, 2010.

[41] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE,* vol. 55, *(4),* pp. 523-531, 1967.

[42] S. Atapattu, C. Tellambura and H. Jiang, *Energy Detection for Spectrum Sensing in Cognitive Radio.* New York, NY: Springer, 2014.

[43] H. Rasheed, F. Haroon and N. Rajatheva, "Energy detection with diversity combining over KG fading for cognitive VANET," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall),* 2016, pp. 1-5.

[44] X. Qian and L. Hao, "Spectrum sensing with energy detection in cognitive vehicular ad hoc networks," in *2014 IEEE 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014),* 2014, pp. 1-5.

[45] I. Souid, H. B. Chikha and R. Attia, "Blind spectrum sensing in cognitive vehicular ad hoc networks over nakagami-m fading channels," in *2014 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM),* 2014, pp. 1-5.

[46] D. B. Rawat *et al,* "CRAVE: Cognitive radio enabled vehicular communications in heterogeneous networks," in *2013 IEEE Radio and Wireless Symposium,* 2013, pp. 190-192.

[47] M. Di Felice, K. R. Chowdhury and L. Bononi, "Cooperative spectrum management in cognitive vehicular ad hoc networks," in *Vehicular Networking Conference (VNC), 2011 IEEE,* 2011, pp. 47-54.

[48] M. Di Felice, K. R. Chowdhury and L. Bononi, "Analyzing the potential of cooperative cognitive radio technology on inter-vehicle communication," in *Wireless Days (WD), 2010 IFIP,* 2010, pp. 1-6.

[49] Husheng Li and D. K. Irick, "Collaborative spectrum sensing in cognitive radio vehicular ad hoc networks: Belief propagation on highway," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st,* 2010, pp. 1-5.

[50] B. Dappuri and T. G. Venkatesh, "Design and Performance Analysis of Cognitive WLAN MAC Protocol," *IEEE Systems Journal,* vol. PP, *(99),* pp. 1-12, 2017.

[51] K. Hong, S. Sengupta and R. Chandramouli, "SpiderRadio: A Cognitive Radio Implementation Using IEEE 802.11 Components," *IEEE Transactions on Mobile Computing,* vol. 12, *(11),* pp. 2105-2118, 2013.

[52] K. G. Shin *et al*, "An experimental approach to spectrum sensing in cognitive radio networks with off-the-shelf IEEE 802.11 devices," in *2007 4th IEEE Consumer Communications and Networking Conference,* 2007, pp. 1154-1158.

[53] "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 5: Television White Spaces (TVWS) Operation," *IEEE Std 802. 11af-2013 (Amendment to IEEE Std 802. 11-2012, as Amended by IEEE Std 802. 11ae-2012, IEEE Std 802. 11aa-2012, IEEE Std 802. 11ad-2012, and IEEE Std 802. 11ac-2013),* pp. 1-198, 2014.

[54] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Physical Communication,* vol. 4, *(1),* pp. 26-39, 3, 2011.

[55] Xiao Yu Wang and Pin-Han Ho, "A Novel Sensing Coordination Framework for CR-VANETs," *Vehicular Technology, IEEE Transactions On,* vol. 59, *(4),* pp. 1936-1948, 2010.

[56] P. Hu and M. Ibnkahla, "CM-MAC: A cognitive MAC protocol with mobility support in cognitive radio ad hoc networks," in *2012 IEEE International Conference on Communications (ICC),* 2012, pp. 430-434.

[57] H. Su and X. Zhang, "Cross-Layer Based Opportunistic MAC Protocols for QoS Provisionings Over Cognitive Radio Wireless Networks," *IEEE Journal on Selected Areas in Communications,* vol. 26, *(1),* pp. 118-129, 2008.

[58] L. Le and E. Hossain, "A MAC protocol for opportunistic spectrum access in cognitive radio networks," in *2008 IEEE Wireless Communications and Networking Conference,* 2008, pp. 1426-1430.

[59] P. Ren, Y. Wang and Q. Du, "CAD-MAC: A Channel-Aggregation Diversity Based MAC Protocol for Spectrum and Energy Efficient Cognitive Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications,* vol. 32, *(2),* pp. 237-250, 2014.

[60] S. Debroy, S. De and M. Chatterjee, "Contention Based Multichannel MAC Protocol for Distributed Cognitive Radio Networks," *IEEE Transactions on Mobile Computing,* vol. 13, *(12),* pp. 2749-2762, 2014.

[61] J. Jia, Q. Zhang and X. S. Shen, "HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management," *IEEE Journal on Selected Areas in Communications,* vol. 26, *(1),* pp. 106-117, 2008.

[62] Jun Zhao, Haitao Zheng and Guang-Hua Yang, "Distributed coordination in dynamic spectrum allocation networks," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.* 2005, pp. 259-268.

[63] C. Cordeiro and K. Challapali, "C-MAC: A cognitive MAC protocol for multi-channel wireless networks," in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks,* 2007, pp. 147-157.

[64] B. F. Lo, I. F. Akyildiz and A. M. Al-Dhelaan, "Efficient Recovery Control Channel Design in Cognitive Radio Ad Hoc Networks," *IEEE Transactions on Vehicular Technology,* vol. 59, *(9),* pp. 4513-4526, 2010.

[65] T. Chen *et al*, "CogMesh: A cluster-based cognitive radio network," in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks,* 2007, pp. 168-178.

[66] L. Lazos, S. Liu and M. Krunz, "Spectrum opportunity-based control channel assignment in cognitive radio networks," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks,* 2009, pp. 1-9.

[67] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium On,* 2008, pp. 1-7.

[68] N. Baldo, A. Asterjadhi and M. Zorzi, "Dynamic Spectrum Access Using a Network Coded Cognitive Control Channel," *IEEE Transactions on Wireless Communications,* vol. 9, *(8),* pp. 2575-2587, 2010.

[69] C. Cormio and K. R. Chowdhury, "An adaptive multiple rendezvous control channel for cognitive radio wireless ad hoc networks," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops),* 2010, pp. 346-351.

[70] K. Bian, J. M. Park and R. Chen, "Control Channel Establishment in Cognitive Radio Networks using Channel Hopping," *IEEE Journal on Selected Areas in Communications,* vol. 29, *(4),* pp. 689-703, 2011.

[71] J. Banks *et al*, *Discrete-Event System Simulation.* (5th edition ed.) Prentice Hall, 2009.

[72] Riverbed Technology, "Riverbed Modeler for defence organizations, network equipment manufacturers and research organizations," vol. Riverbed Modeler datasheet, 2014. Available: https://www.riverbed.com/sg/document/fpo/9306_Riverbed_Modeler_DS_101314KC-2.pdf.

[73] SCALABLE Network Technologies. (2017). *Network Models* [QualNet]. Available: http://web.scalable-networks.com/network-models.

[74] (2017, June 19). *ns-3: WAVE models*. Available: https://www.nsnam.org/docs/models/html/wave.html.

[75] A. Varga. (2016). *OMNeT++ Discrete Event Simulator - Introduction*. Available: https://omnetpp.org/intro.

[76] S. N. Khan, M. A. Kalil and A. Mitschele-Thiel, "crSimulator: A discrete simulation model for cognitive radio ad hoc networks in OMNeT ++," in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP,* 2013, pp. 1-7.

[77] J. Marinho and E. Monteiro, "Cognitive radio simulation based on OMNeT++/MiXiM," in *11ª Conferencia Sobre Redes De Computadores (CRC),* Portugal, 2011.

[78] G. Caso, L. De Nardis and O. Holland, "Simulation of cognitive radio networks in OMNeT++," in *Cognitive Radio and Networking for Heterogeneous Wireless Networks: Recent Advances and Visions for the Future*, M. Di Benedetto *et al*, Ed. Cham: Springer International Publishing, 2015, pp. 291-313.

[79] V. Esmaeelzadeh *et al*, "CogNS: A Simulation Framework for Cognitive Radio Networks," *Wireless Personal Communications,* vol. 72, *(4),* pp. 2849-2865, 11, 2013.

[80] D. Yang, B. Li and C. Cui, "Research of Cognitive Radio Network Simulation Platform Based on NS3," *Hans Journal of Wireless Communications,* vol. 7, *(1),* pp. 23-30, Feb., 2017.

[81] European Telecommunications Standards Institute. Intelligent transport systems (ITS) - radiocommunications equipment operating in the 5855 MHz to 5925 MHz frequency band - harmonised standard covering the essential requirements of article 3.2 of directive 2014/53/EU. France. 2017Available: https://www.etsi.org/deliver/etsi_en/302500_302599/302571/02.01.01_60/en_302571v020101p.pdf.

[82] European Telecommunications Standards Institute. Intelligent transport systems (ITS) - access layer specification for intelligent transport systems operating in the 5 GHz frequency band. France. 2013 Available: https://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.02.01_60/en_302663v010201p.pdf.

[83] IEEE, "IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802. 11p-2010 (Amendment to IEEE Std 802. 11-2007 as Amended by IEEE Std 802. 11k-2008, IEEE Std 802. 11r-2008, IEEE Std 802. 11y-2008, IEEE Std 802. 11n-2009, and IEEE Std 802. 11w-2009),* pp. 1-51, 2010.

[84] IEEE, "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802. 11-2012 (Revision of IEEE Std 802. 11-2007),* pp. 1-2793, 2012.

[85] IEEE Vehicular Technology Society, "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) — Multi-Channel Operation," *IEEE Std 1609. 4-2016 (Revision of IEEE Std 1609. 4-2010),* pp. 1-94, 2016.

[86] IEEE Vehicular Technology Society, "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) — Networking Services," *IEEE Std 1609. 3-2016 (Revision of IEEE Std 1609. 3-2010),* pp. 1-160, 2016.

[87] M. Asefi, J. W. Mark and X. Shen, "A cross-layer path selection scheme for video streaming over vehicular ad-hoc networks," in *2010 IEEE 72nd Vehicular Technology Conference - Fall,* 2010, pp. 1-5.

[88] A. M. Law, *Simulation Modeling and Analysis.* (Fifth ed.) New York, NY: McGraw-Hill Education. 2015.

[89] W. Viriyasitavat, F. Bai and O. K. Tonguz, "Dynamics of Network Connectivity in Urban Vehicular Networks," *IEEE Journal on Selected Areas in Communications,* vol. 29, *(3),* pp. 515-533, 2011.

[90] (2017, Aug 21). *ns-3: Mobility*. Available: https://www.nsnam.org/docs/models/html/mobility.html.

[91] (2016, Jan 22). *INET Framework for OMNeT++: Manual*. Available: https://omnetpp.org/doc/inet/api-current/inet-manual-draft.pdf.

[92] J. Harri, F. Filali and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Communications Surveys & Tutorials,* vol. 11, *(4),* pp. 19-41, 2009.

[93] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing,* T. Imielinski and H. F. Korth, Eds. Boston, MA: Springer US, 1996, pp. 153-181.

[94] J. Yoon, M. Liu and B. Noble, "Random waypoint considered harmful," in *IEEE INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. no.03CH37428),* 2003, pp. 1312-1321 vol.2.

[95] S. Uppoor, M. Fiore and J. Härri, "Synthetic mobility traces for vehicular networking," in *Vehicular Networks*, John Wiley & Sons, Inc., 2013, pp. 209-245.

[96] Transport Simulation Systems. (2017). *Aimsun traffic modelling software*. Available: https://www.aimsun.com/aimsun.

[97] SYSTRA Limited. (2016). *Paramics Microsimulation*. Available: http://www.sias.com/2013/sp/sparamicshome.htm.

[98] (2017). *PTV Vissim*. Available: http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/.

[99] Institute of Transportation Systems. *DLR - Institute of Transportation Systems - SUMO – Simulation of Urban MObility*. Available: http://www.sumo.dlr.de.

[100] S. Krauß, P. Wagner and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E,* vol. 55, *(5),* pp. 5597, 1997.

[101] D. Krajzewicz, "Traffic simulation with SUMO-simulation of urban mobility," in *Fundamentals of Traffic Simulation*, J. Barceló, Ed. New York, NY: Springer New York, 2010, pp. 269-293.

[102] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transportation Research Part B: Methodological,* vol. 20, *(5),* pp. 403-414, 1986.

[103] M. Treiber, A. Hennecke and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E,* vol. 62, *(2),* pp. 1805, 2000.

[104] OpenStreetMap Foundation. (2017). *OpenStreetMap*. Available: https://www.openstreetmap.org/about.

[105] C. Sommer. (2016, April 07). *Veins-The open source vehicular network simulation framework*. Available: http://veins.car2x.org/documentation/.

[106] (2011). *MiXiM simulator for wireless and mobile networks using OMNeT++.* Available: http://mixim.sourceforge.net.

[107] A. Köpke *et al*, "Simulating wireless and mobile networks in OMNeT++ – the MiXiM vision," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (Simutools '08),* Marseille, France, 2008, pp. 1-8.

[108] D. Eckhoff and C. Sommer, "A multi-channel IEEE 1609.4 and 802.11p EDCA model for the veins framework," in *5th ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2012),* 2012.

[109] C. Sommer, "Car-to-X Communication in Heterogeneous Environments." PhD, Department of Computer Science, University of Erlangen, 2011.

[110] T. S. Rappaport, *Wireless Communications: Principles and Practice.* (2nd Edition ed.) Upper Saddle River, N.J.: Prentice Hall PTR, 2002.

[111] K. Wessel *et al*, "MiXiM – the physical layer - an architecture overview," in *Proceeding of the 2nd International Workshop on OMNeT++,* 2009.

[112] G. Pei and T. R. Henderson, "Validation of OFDM error rate model in ns-3," The Boeing Company, 2010.