Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2016

Project Title: **Making Cars Talk: An Investigation into the Use of the IEEE 802.11p Standard when Applied to Vehicular Ad-Hoc Networks**

Student: **Sanjana George**

CID: **00731650**

Course: **Electrical and Electronic Engineering with Management Year 4**

Project Supervisor: **Professor Kin Leung**

Second Marker: **Dr Tania Stathaki**

Contents

Abbreviations List

AC = Access Category
AIFS = Arbitrary Interframe Space
CCH = Control Channel
CSMA/CA = Carrier Sense Multiple Access/ Collision Avoidance
DCF = Distributed Coordination Function
DSRC = Dedicated Short Range Communication
EDCA = Enhanced Distributed Channel Access
EDCAF = Enhanced Distributed Channel Access Function
FCC = Federal Communications Commission
HCCA = Hybrid Coordination Function Controlled Channel Access
HCF = Hybrid Coordination Function
IEEE = Institute of Electrical and Electronics Engineers
ITS = Intelligent Transportation Systems
IVC = Inter-Vehicular Communication
MANET = Mobile Ah-Hoc Network
MCCA = Mesh Coordination Function Controlled Channel Access
MCF = Mesh Coordination Function
OBU = On-Board Units
PC = Point Coordinator
PCF = Point Coordination Function
PHY = Physical Layer
QoS = Quality Of Service
RVC = Road-Vehicle Communication
RSU = Road-Side Units
SCH = Service Channel
STA = Station
TA = Timing Advertisement
TXOP = Transmission Opportunity
UP = User Priority
UTC = Coordinated Universal Time
VANET = Vehicular Ad-Hoc Network
V2R = Vehicle to Road
V2V = Vehicle to Vehicle
WAVE = Wireless Access Vehicular Environment
WBSS = WAVE Basic Service Set
WLAN = Wireless Local Area Network
WM = Wireless Medium
WSA = WAVE Service Advertisements

1 Acknowledgements

I would primarily like to thank Professor Kin Leung for his incredibly helpful guidance and advice throughout the duration of this project. Additionally, a special mention goes out to my friends and family for their unwavering support and faith in me.

2 Abstract

This project is an investigation into the Medium Access Control (MAC) sub-layer of the IEEE 802.11p protocol that is used as the standard in Vehicle-To-Vehicle communication networks. Vehicular Ad-Hoc Networks (VANETs) can improve road safety, limit accidents and reduce congestion by providing additional, timely information to moving vehicles.

VANETs have several unique characteristics when compared to the regular class of Mobile Ad-Hoc Networks and as such, their MAC layer is not optimised. This project uses MATLAB to model two standard network protocols and then aims to improve a specific control function of the IEEE 802.11p MAC layer for use in VANET systems. The performance of each system perspective is evaluated by measuring throughput, latency and packet loss probabilities, to quantify relative effectiveness.

The control mechanism improvement, studies the timing split when data is received from a Control Channel (CCH) and then one of several Service Channels (SCHs). Despite traffic level variations on the channel streams, a fixed access time exists for each, which is not ideal. This study shows that by dynamically adjusting the time boundary of the CCH and SCH intervals based on traffic load, the performance of the MAC sub-layer of VANET systems can be improved.

3 Introduction

Vehicular Ad-hoc Networks (VANETs), fall under the broad umbrella of Mobile Ad-hoc Networks (MANETs). VANETs however, have several unique characteristics that make them a worthwhile area to research. They can be used to improve road safety for drivers, limit accidents and reduce congestion, hence lowering air pollution levels.

The project specification details the key aim to be an investigation into network protocols and control functions that are specific for use in VANETs. However, within this, it is important to clearly define the scope and aims for the project as this allows for a clear path of interest, amidst a field in which a vast amount of research exists. A set of protocols and standards exists that is designed for Vehicle-to-Vehicle (V2V) communication. These use Dedicated Short-Range Communication (DSRC) channels for transmission. Within this, the IEEE 802.11p protocol is the standard used for the Medium Access Control (MAC) layer. [1]

Yet from research, it can be argued, that the 802.11p Medium Access Control protocol has room for improvement. For example, a high enough level of quality of service is extremely important, particularly when applied to vehicle safety. [2] This project aims to investigate and analyse the efficiency of the 802.11p MAC protocol when used in VANETs.

This will primarily be carried out using MATLAB. This software is sufficient to simulate the first order effects of the real environment. Through simulating the protocols and their impact on basic networks any enhancements made to the 802.11p protocol can be assessed. The measure of success will be evaluated by testing protocol versions against performance criteria such as throughput, latency and packet loss probability. The methods are compared using graph plots and quantitative relative success probabilities.

This report splits the background research section into two parts. The first studies gives a general overview of VANET systems and their applications, whilst the second focuses on the MAC sub-layer of the 802.11p protocol, taking a more refined look at the system. Possible protocol improvement characteristics are then analysed and any existing improvement techniques are briefly summarised, before the project motivation, aims and objectives are defined.

The analysis for modelling VANET systems is then carried out and five models are designed. These include two protocol implementations that are exercised using two different methods and one control mechanism to attempt to improve system functionality. The improved resource allocation mechanism proved to be most challenging on implementation since several parameters had to be guesstimated using reasonable values. This was overcome, by taking this into account when evaluating the model.

After the overall testing aims are defined, each of the five models is implemented in MATLAB, tested against performance criteria and analysed using graphical results. The modelling work and overall project results are then summarised in the evaluation, further work and conclusion sections at the end of the report.

As a final note, this project has been carried out in such a way that if a third party wishes to carry on the research they should be able to without difficulty.

4 An Overview of Vehicular Ad-Hoc Network Systems

4.1 The Growth of Wireless Networks

With the growth of technology over the past few decades, there has been a huge increase in the demand for wireless communication technologies. Features such as internet access, media downloads and other interactive services can now all be carried out via remote access. This is possible due to the presence of Mobile Ad-Hoc Networks (MANETs). The increasing societal dependence on data transmission such as voice and video interactions, means the field is rapidly evolving and incredibly dynamic. Previous wireless networks are constantly being improved upon and integrated into new systems whilst there is pressure on new wireless standards to aid smooth roaming and interoperability across networks. [3]

The Federal Communications Commission (FCC) has now allocated 75MHz of the Dedicated Short-Range Communications spectrum at 5.9GHz to be used specifically for Vehicle-To-Vehicle (V2V) and Vehicle-To-Road (V2R) purposes. [4] This will be looked at in more detail in Section 5.1, which discusses current standards.

4.2 An Introduction to Vehicular Ad-Hoc Networks

The growth in wireless communications, as discussed in Section 4.1, has meant that Inter-Vehicular Communication (IVC) and Road-Vehicle Communication (RVC) systems have been given more attention. [5] This is in combination with the evidence in Fig. 1, which indicates a huge amount of growth forecasted in the automotive industry from 85.5 million car sales in 2013 to approximately 100 million car sales expected in 2018. [6] This has led to Vehicular Ad-Hoc Networks (VANETs) being created as a special self-organised wireless subset of Mobile Ad-Hoc Networks. [7][4] VANETs are interconnected without the need for any centralised infrastructure. [3] They have an incredibly dynamic topology, as they are general-purpose networks, which are influenced by their surrounding environment and vehicle movement characteristics. [3][8]

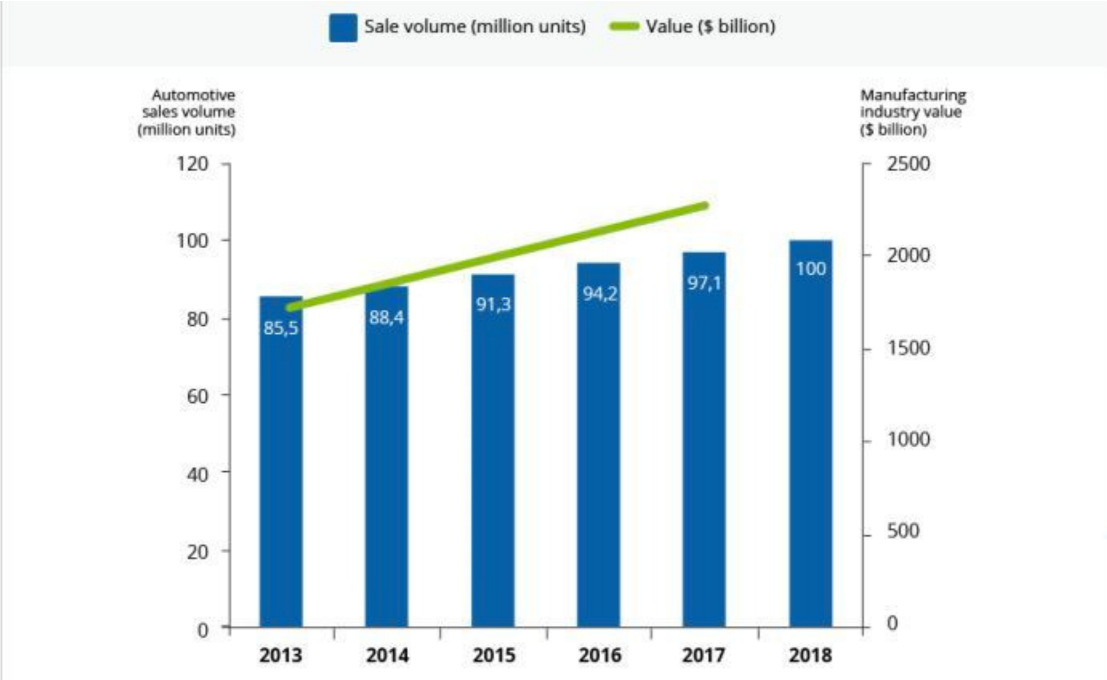


Figure 1: Global Forecasted Growth in the Automotive Industry from 2013 to 2018.[6]

VANETs allow drivers to have an enhanced experience, giving them information that is otherwise not in human eyesight. This could be related to other vehicles or their road ahead. [7] Road-Side Units (RSUs) are placed by stop signs, traffic lights or intersections and can communicate with On-Board Units (OBUs) that are built into cars. These OBUs can also allow cars to communicate with each other. [4] The mobile networks that are automatically established can receive information on the velocity or position of nearby cars. [7] This results in several safety applications for VANETs that will be discussed further in Section 4.3. The concepts explained so far are shown more clearly pictorially in Fig. 2 below. Note that in this project, the focus will not be on RVC development since ultimately, from a network perspective, road infrastructure can be treated as a stationary car.

Although the field of VANETs is relatively new, as is much of the routing and quality of service research, there is large market growth expected in the future. [3][8] Many national

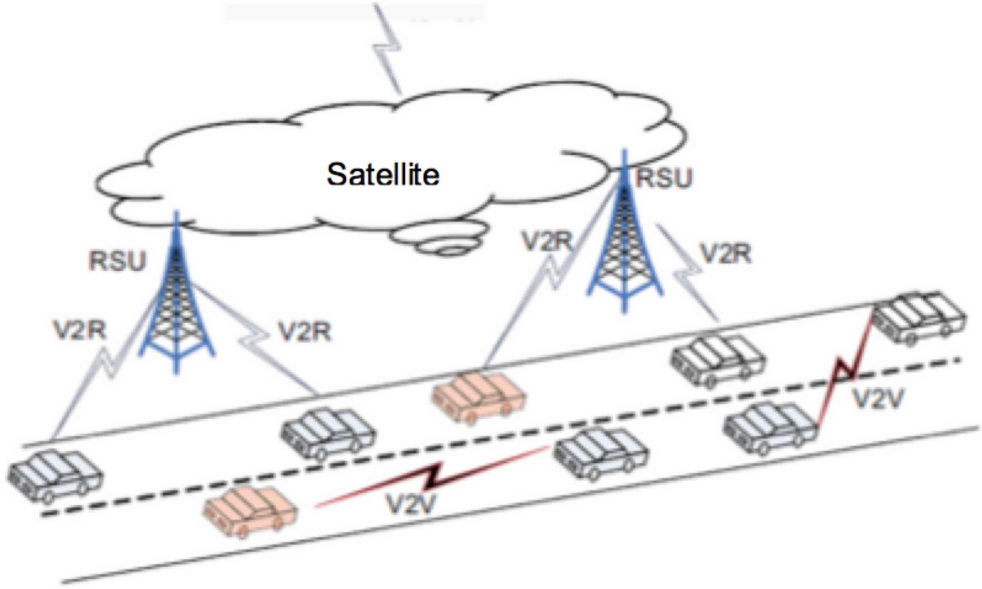


Figure 2: Conceptual diagram showing V2V and V2R communications including the RSUs that connect to a satellite. [9]

plans look to renew highway information systems through the implementation of IVC systems. [4] The field of IVCs is therefore being acknowledged both by the academic community and also by government organisations that are looking into schemes such as Advanced Driver Assistance Systems. In fact, car manufacturers such as BMW and Toyota are also developing projects in the field. [5]

4.3 The Application of VANETs in Society

VANETs allow access to a vast range of applications and uses as they form a part of Intelligent Transportation Systems (ITS). Their most relevant and dominating use is in the improvement of road safety, through making new forms of IVC possible. [3] This is of vital importance since road injuries are the leading cause of death around the world, killing 1.3 million people each year and injuring over 50 million. They mostly affect 5-29 year olds; with more than 1000 people under 25 killed every day and this figure expected to double in poorer countries by 2020. [10] The list below gives more detail on additional practical uses for VANETs:

- **Road Safety:** The operation of active safety systems improves upon proactive systems and gives information to drivers at an early stage, where a change can still be made. This is a life-critical application that could greatly improve public safety. Accidents can be avoided and passenger safety improved through the exchange of warning messages. Cars can be steered and aided at tricky parts of the road such as motorway entrances and cross roads, thus avoiding possible collisions. [4][5]

- **Road Congestion:** Road information provided by VANETs can improve vehicle traffic flow and control. This leads to a reduction in congestion, thus limiting fuel and time wastage. [4][5][10] Arguably, this results in environmental benefits.

- **Road Comfort:** Details can be displayed regarding weather forecasts, restaurant addresses and gas station locations. [5]

- **Lower Costs Relative to RVC Systems:** RVC system infrastructures can be incredibly expensive to install in rural areas, where vehicles are not as common as in metropolitan cities. The use of OBUs and VANETs could greatly improve this. [10]

- **Additional Uses:** Group communications, electronic toll collections, the sharing of multimedia information. When acting as a terminal network and accessing the Internet, media downloads can be performed. [4][5]

As an aside, in order for VANETs to be functional, the DSRC technology is crucial. How well they perform is largely determined by the routing in their network and thus the underlying protocols. These protocols are the focus of this project. [3][4]

4.4 VANET System Challenges and How They Differ From MANETs

VANET networks pose several challenges in their design and structure. These include security issues, the efficiency of the MAC protocols, Quality of Service (QoS), multi-hop routing and scalability. [7]

Network security is a particularly relevant research area. Lives are at stake when driving and so it is important that the network cannot be hacked and tampered with. This ties into the design of the MAC protocols. High reliability is important, such that all messages can be disseminated through the network in an efficient and timely manner. [4] This should also take into account the priority of the messages, allowing more important messages to be sent first. In other words, there should be fair packet transmission. The MAC protocols should also fulfil QoS standards for each application of use. [4]

As previously mentioned, VANETs are a subset of MANETs and so share certain features such as their constantly changing topologies and their fast moving nodes. [5] However, there are a multitude of features that are unique to VANETs, or areas where the standard routing solutions used for MANETs cannot be applied to VANETs. This may be because they do not take advantage of unique VANET features or are not as effective as they could be. The following points indicate aspects of MANET solutions that are not ideal when implemented in VANETs.

- **Saving Unique Information:** As discussed earlier, the OBUs in cars can collect information on velocities, road maps and expected motions. This can be useful to improve VANET protocols, but are not applicable in the case of MANETs. [3]
- **Scalability and Storage:** There is a limit on the number of mobile nodes that MANET protocols can support and this can be expensive to implement on large VANET networks. Additionally, the proactive MANET protocols that have routing tables storing routes to other nodes in the network are incredibly impractical for VANETs. [3]
- **Mobility:** In terms of motion, VANET nodes follow the roads and are limited by speed limits, traffic jams or road features such as traffic lights. [5] MANETs disregard this potentially useful information, as they assume arbitrary motion. [3]
- **The Use of Flooding:** In reactive routing protocols, the data sources find a route to the destination by flooding the network. In the proactive case, nodes send

periodic messages to a confined neighbourhood or the entire network. This puts a strain on resources such as bandwidth and is not feasible for large networks. MANETs use flooding, however this is unreasonable for VANETs, where the large number of nodes means that if flooding was used, it should be limited to a set area radius. [3]

- **Localized Routing:** In reactive protocols, all the nodes flood the network, whereas in proactive solutions all the nodes form routing tables. In VANETs, if the destination of the data is known then localised solutions work better, since nodes can gather the required information from their neighbourhood. This improves scalability and the ability to adapt to different network environments. [3]

4.5 VANET Mobility Model Characteristics

There are several features that VANET protocols can factor in when they are being designed. These include vehicle motion characteristics such as direction and speed of motion (greater than in MANETs), as well as the direction of the information flow.

Features such as, limited bandwidth, variations in channel capacity and channel errors caused by location, restrict wireless channels and so a realistic mobility model is necessary for designing and evaluating routing protocols. [3][7]

A high quality model can take into account the street or city environment, the velocity and density of vehicles and the position of obstructions such as buildings. [7] If prediction is used when designing routing protocols, certain notes on vehicle behaviour should be taken into account [11]:

- Certain vehicles, such as buses have fixed routes.
- Drivers always know where they are going and the route they intend to take.
- There are statistical regularities in their driving behaviours.

- Traffic statistics can provide an insight into the details of certain routes.
- The road layouts change very little over long periods of time. [11]

These facts could potentially be used to improve routing protocols in VANETs.

The aim of the IEEE 802.11p MAC protocol is to give the minimum set of specifications, allowing for the constantly changing VANET environment and given that data exchanges have a limited time frame which is much shorter than for Wireless Local Area Networks (WLAN). This set of instructions tries to guarantee interoperability between various wireless devices. [4] However it is important to note that this project will not be creating or looking to improve existing mobility models, instead it looks to enhance the MAC protocol used in conjunction with any mobility model.

4.6 The External VANET Development Environment

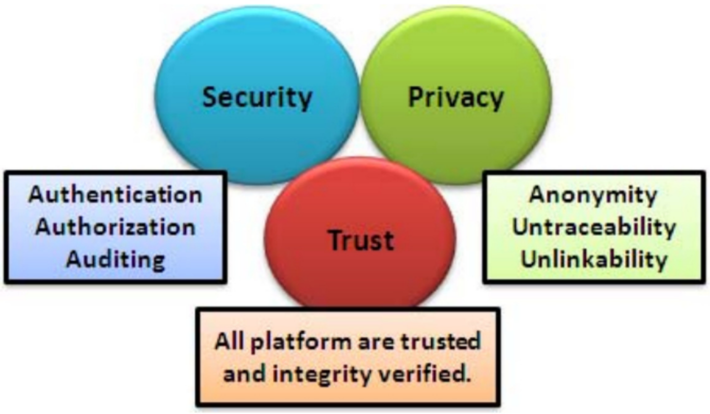


Figure 3: The Main Public Concerns Regarding the Implementation of VANETS [4]

There are external entry barriers that VANET development must overcome in order for successful implementation in automobiles. These are mostly related to security features since, as mentioned in Section 4.4, network hacking would be disastrous. Although academics, government bodies, car manufacturers and end users all play a part in the widespread adoption of VANETs. [10]

The main factors that must be considered, are given in Fig. 3, which highlights the main public concerns and scepticism that VANETs may face as they begin to be integrated into society. Currently, extra applications and private services are being allowed to use VANETs, in order to lower installation costs and encourage market adoption of the new technology. [4]

5 A Study of the MAC Sublayer of the 802.11p Protocol

5.1 The Development of IEEE Standards in the Field of VANETs

With the growth of research in the field of wireless networks and communications, a range of new standards was developed. In this case, the foundation of Wireless Access in the Vehicular Environment (WAVE) architecture is the IEEE 802.11 standard. [12]

Fig. 4 shows the 7 10-MHz wide channels that are present in the DSRC spectrum. The Control Channel (CCH) is reserved purely for safety communications and is channel 178, whilst the two edge channels are for any accident avoidance applications or public safety communications that are developed in the future. The rest are standard service channels (SCH). [4]

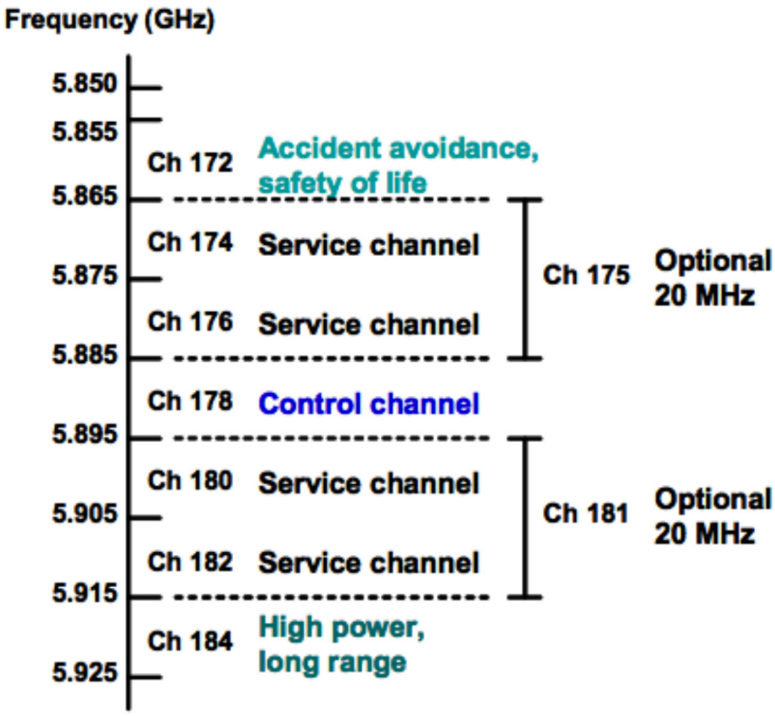


Figure 4: DSRC Channel Assignment [4]

The IEEE vehicular network standards P1609.1, P1609.2, P1609.3, and P1609.4 were all released for trial use. Their uses are briefly described below:

- **P1609.1:** The standard WAVE Resource Manager that defines interfaces, message data formats and services of the application. It acts as a gateway, providing applications access to other architectures. [4]

- **P1609.2:** Security focussed, carries out the formatting and processing of secure message exchanges. Tries to protect WAVE messages from being infiltrated or faked. [4]

- **P1609.3:** Defines the management information base for the protocol stack, and manages routing and transport facilities. [4]

- **P1609.4:** Handles the specification of all of the other channels in the DSRC standard, including the Control and Service Channels. [4][13]

The IEEE then worked on the 802.11p MAC standard, which improved upon the 802.11e and 802.11a standards, specifically to support the unique aspects of VANETs. [4][12][14] This allowed for developments to reduce transmission interferences and also for routing protocols that maximised signal-to-interference levels. [12] The IEEE 802.11p MAC layer protocol uses Carrier-Sense Multiple Access/ Collision Avoidance (CSMA/CA) to access links. [4][15] The second part of the 802.11p standard is the physical (PHY) layer, which works in the 5.850 - 5.925 GHz DSRC spectrum. However, this aspect will not be the main focus of this project. [4]

Despite many available communication channels, the 802.11p protocol is not ideal for certain VANET applications, such as safety. [12] This is because safety applications require a high QoS, which IEEE 802.11p cannot reliably deliver, particularly in heavy traffic congestion. [14] This idea will be explored further in the upcoming sections.

5.2 An Introduction to the IEEE 802.11p MAC Protocol

The IEEE 802.11p standard defines a PHY layer and a MAC sub-layer, which is closely linked to the IEEE 1609.4 WAVE standard protocol stack. Together the IEEE 1609.4 standard and the IEEE 802.11p protocol form the basis of the multichannel operation MAC sub-layer. This is shown diagrammatically in Fig. 5 below. The key features of the sub-layer are detailed below and these will be investigated more thoroughly in the following sections.

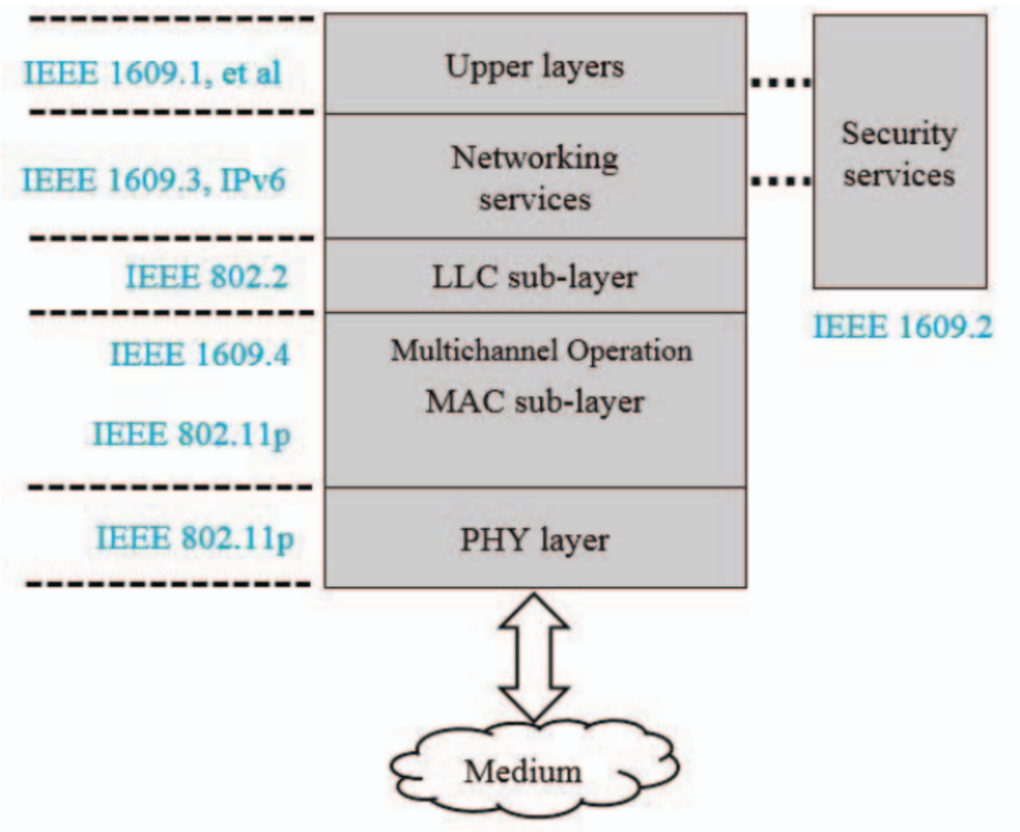


Figure 5: The WAVE Protocol Stack and the IEEE 1609.x Standards [16]

The basic medium access format for the IEEE 802.11p is via the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol and the set of rules it details allows efficient access for packages to share the communication channel. [17] This will be studied and explained in more detail over the course of this project since the CS-

MA/CA protocol forms an integral part of the Distributed Coordination Function (DCF) scheme. [18] Within the IEEE 802.11p framework, Timing Advertisement (TA) frames, give information about the time at which a message was sent. [17]

A packet priority scheme is used to maintain a good quality of service at each end of the channel called the Enhanced Distributed Coordination Access (EDCA) protocol, which uses CSMA/CA. [18][16] EDCA is a contention-based channel access scheme and the protocol makes use of a Hybrid Coordination Function (HCF). [18] It subdivides packets into one of four Access Categories (ACs), based on their importance, such that greater channel access is given to classes with more important information. This will be analysed further in Section 5.3.

In the upper MAC layer, there is an alternating access scheme where channel time is split evenly between the CCH, for safety issues and each of the six SCHs for media and entertainment purposes. [19] This feature is an important part of IEEE 1609.4. It will be shown further on in this project that the central control channel interval length has a significant impact on the delivery of safety messages. [18] The even split of transmission between the CCH and SCHs, is not necessarily ideal and this resource allocation control mechanism will be investigated as the main method by which the efficiency of the MAC sub-layer can be improved.

For example in a system where position messages are of great importance, particularly in safety applications, the IEEE 802.11p standard is flawed. [20] Priority messages can contain data on vehicle position and velocity and when broadcast periodically with a heavy network load, the unpredictability of delivery in the IEEE 802.11p MAC scheme is undesirable. [20] This shall be looked at in more detail in Section 5.6.

Fig. 6 shows the MAC sub-layer architecture in more detail and indicates the co-operation of the distributed coordination function (DCF), the point coordination func-

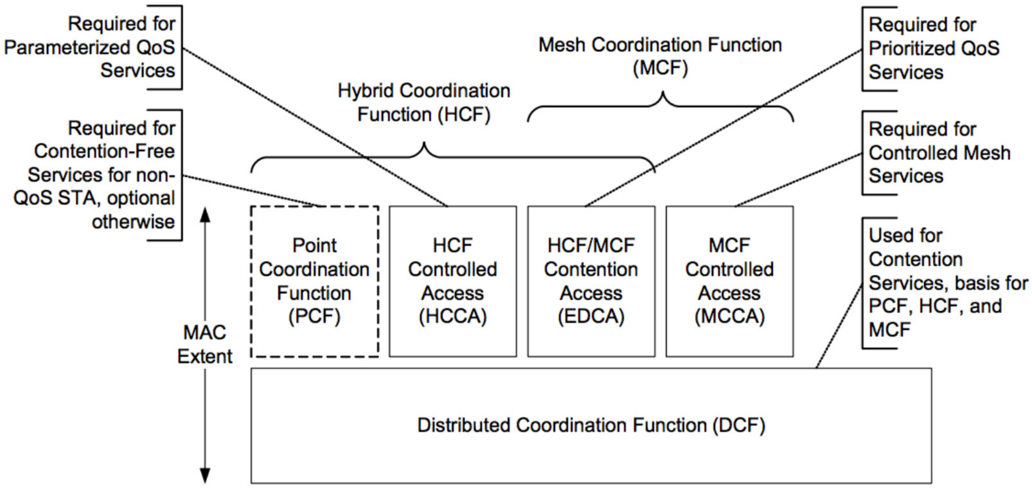


Figure 6: The MAC Layer Architecture [21]

tion (PCF), the hybrid coordination function (HCF) and the Mesh Coordination Function (MCF). In a station implementation where quality of service is relevant both DCF and HCF are present and these will form the basis of research within this project. PCF is optional in all stations and is less integral to the upcoming analysis. [21]

- **DCF:** The distributed coordination function is the fundamental and crucial access method of the IEEE 802.11 MAC utilising CSMA/CA. It is implemented in all stations and considered successful once an Acknowledgement (ACK) frame is received from the station. [21][22] This protocol will be investigated, however less emphasis will be placed on the receipt of the acknowledgment frames. Detailed assumptions will be made for this before models are designed and tested.

- **PCF:** The point coordination function is an optional access method that is only utilised on infrastructure network configurations and thus will not be analysed in this project. When determining which station has the right to transmit it uses polling, where the point coordinator (PC) is the polling master. All frame transmissions also have a shorter inter-frame space, and therefore wait time that those in using the DCF mechanism. [21][22][23]

- **HCF:** The hybrid coordination function combines elements of the DCF and PCF and has two main channel access schemes. The first has been mentioned previously and is the contention-based channel access method known as the enhanced distributed channel access (EDCA) mechanism. The second provides controlled channel access and allows for contention-free transfers, it is referred to as the HCF controlled channel access (HCCA) mechanism. [21][22][24] This project focuses on the EDCA mechanism as it forms the basis for priority classes and builds directly on the solid analytical foundation provided by the DCF protocol.

- **MCF:** Certain stations implement only the Mesh Coordination Function, which has both a contention-free channel access mechanism and a contention-based channel access mechanism. Here the contention-based protocol is EDCA however, the contention-free mechanism is slightly different to HCCA and is called the MCF controlled channel access method (MCCA). This further justifies investing into EDCA, since it is widely used within the field of VANETS. [21][22]

5.3 Access Categories Explained

The EDCA mechanism of the IEEE 802.11p protocol classes information into four different Access Categories (ACs), which allows for eight different User Priorities (UPs) to be accounted for. [21] The four access categories of information are: Background (AC_BK), Best Effort (AC_BE), Video (AC_VI) and Voice (AC_VO). Each of these is then further subdivided into two priority levels. This is shown clearly in Fig. 7.

By categorising information by user priority, the EDCA function allows for varying and distributed access to the wireless medium based on the AC of the data packet. This supports efficient packet delivery, increasing the likelihood that the most important packets are transmitted quickly. [21]

An enhanced and more technically specific version of the DCF exists for each access class and this is called an enhanced distributed channel access function (EDCAF). [21] Essentially, each AC operates similarly to an independent DCF station with its own

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
Highest	7	NC	AC_VO	Voice

Figure 7: User Priority to Access Category Mappings [21]

EDCAF. It then contends for Transmission Opportunities (TXOPs) on the wireless medium. [25] Transmission opportunities may be obtained using either or both of the EDCA or HCCA channel access mechanisms, so can either be classed as an EDCA TXOP or an HCCA TXOP. [21] Further detail on the EDCA mechanism can be found in Section 5.8.

5.4 A Quick Look At The CCH and SCH Channels

The DSRC band contains the WAVE frequency that operates over a range of 75 MHz. Within this there exists one Control Channel and up to six Service channels that are present in the alternating access scheme. The CCH is responsible for the transmission of traffic safety messages and system control exchanges, whilst the SCHs are used for various non safety applications such as entertainment data. [26][27]

In order for the channels to be coordinated between vehicles, a global time reference is used and this is done via a global navigation satellite and referred to as the Coordinated Universal Time (UTC). [27] Vehicles alternate between the CCH and the SCHs during sync intervals that have a total fixed length of 100ms. The time spent in each channel is split evenly at 50ms each. [26][27]. This is shown more clearly in Fig. 8.

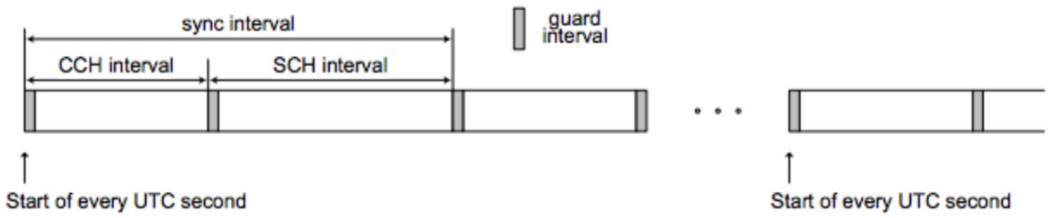


Figure 8: Timing of the CCH and SCH channels [26]

During the CCH interval, all vehicles that are communicating on the network, must tune into the CCH frequency so that they can exchange safety-related information. They can then choose to switch to one of the SCH frequencies once the SCH interval begins. [27] Additionally, during the CCH interval, all activities on the SCHs are paused, likewise during the SCH interval, all activities on the CCH are paused. [26] Fig. 8 also shows a guard interval, which lasts 4ms and occurs before every CCH or SCH interval. This takes into account any timing inaccuracies, switching delays or necessary catch ups on frame transmissions that were started close to the end of the previous interval. [26][27] New transmissions cannot be started whilst guard intervals are being carried out. [26] Also, both the CCH and the SCHs can support the four Access Categories and channel coordination can make this difficult as both are carrying real time traffic. [26]

There are two types of messages that are transmitted over the CCH, the first are Beacons. These short messages declare the presence, location, speed and direction of the vehicle to the cars surrounding it. This is very useful for applications such as accident avoidance or driver assistance, which involve two or more cars. [27] The second message type are WAVE Service Advertisements (WSAs). These are sent to advertise when a WAVE-Basic Service Set (WBSS) is being set-up and give information on the relevant parameters. The set-up of a WBSS provides connectivity and transport of non-safety data during the SCH interval. [27]

From this analysis it is clear that allowing the CCH more time allocation will aid the transmission of safety related messages. However, a higher volume of traffic is expected on the SCHs and since this is often related to media, it can bring in revenues for companies. This creates appeal for the SCH interval to be longer, leading to a trade-off between the two sides of the argument, such that there are optimal channel time allocations to give enough time to the CCH but also handle large traffic volumes on the SCHs. [26] This suggests that the fixed-length subintervals of 50ms are not ideal since the traffic load volumes are uncontrollable. Therefore dynamically adjusting the time boundary at which the transmission switches from the CCH to the SCH or vice versa, can help to improve channel utilisation, which is an important idea that is studied and developed as a control mechanism during this project. Only a few ideas for this have recently been developed in the field and it is a relatively new area of research. [28]

5.5 System Timing Model and Explanation

The timing diagram in Fig.9 is useful to explain the process of gaining channel access and will be useful for reference in the upcoming sections, when explaining how the protocols operate. Access Categories (ACs) in the 802.11p protocol try to prioritise messages regardless of the density of vehicles, with arguably limited success. Messages are queued in the ACs based on their priorities. If packets from different queues both want channel access, the message with higher priority has a greater probability of accessing the channel. [18] This will be explained further in Section 5.8.

The timing interval between each frame transmission is known as the Interframe Space (IFS). Each station determines whether a medium is idle or not by using the carrier sense function throughout this interval. The carrier sense principal will be explained in the next section. There are six different interframe spaces that are defined so that there can be different priority levels of data, when accessing the wireless medium. [21] The six IFS types are as follows:

- RIFS = Reduced Interframe Space

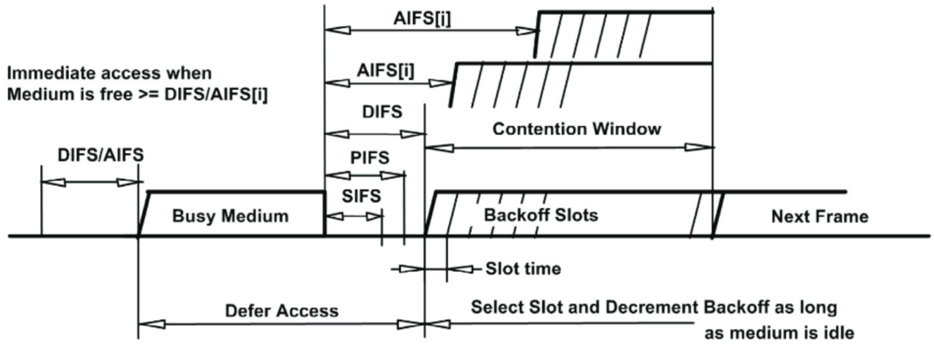


Figure 9: A Basic EDCA Access Method [18]

- SIFS = Short Interframe Space
- PIFS = PCF Interframe Space
- DIFS = DCF Interframe Space
- AIFS = Arbitration Interframe Space
- EIFS = Extended Interframe Space

They are each defined as time gaps on the medium and are independent of the station bit rate. They are all of a fixed length, except for the AIFS which will be studied in more detail in Sections 5.7 and 5.8. [21]

5.6 How the Current MAC Mechanism Operates

The operation of the current MAC mechanism that is used will now be mathematically analysed with respect to the efficiency of message delivery and message reliability.
Let the following assumptions exist [19]:

- There are N nodes (vehicles) that have a packet that needs to be transmitted at the start of each CCH interval.
- At the start of the CCH interval, all transmissions are delayed by a random back-off time. This is due to the existence of the guard interval that indicates the wireless

channel as busy so that multiple devices do not attempt transmission when they switch network.

- The lifetime of a safety related packet is one CCH interval since it is life-critical.
- If the packet cannot be transmitted, it would become void since newer messages would exist, so let a new packet replace the old unsent one.
- A collision will occur if two or more nodes complete their back-off processes simultaneously. [19]

MAC protocols try to enforce reliability and fulfil real-time requirements in VANET systems, however as mentioned earlier, the CSMA based protocol has certain drawbacks. It has trouble avoiding collisions if more than one vehicle has their back-off counter set to zero simultaneously and this worsens with an increase in traffic. [19] It also has problems with fairness and predictability at times. This is especially noticeable when periodic positioning messages are used. [20]

The IEEE 802.11p protocol uses the Carrier Sense Multiple Access/ Collision Avoidance method. It is decentralised, requires no serious synchronisation, supports various packet sizes and is therefore quite low complexity. [20] The simplest process is carried out as follows:

1. The node that has a data packet or frame to send, senses the wireless channel.
2. If the channel is available, the node starts to transmit the frame.
3. If the channel is unavailable, the node carries out a random back-off, by waiting an arbitrary number of time slots before retransmission. This countdown begins when the medium becomes idle and is interrupted in any non-idle interval. It then resumes when the medium returns to idle.
4. The sending node waits for an acknowledgement (ACK) frame from the recipient. If there is no ACK frame within a certain time period, it retransmits the frame

after another random back-off. Note that frames sent to group addresses are not acknowledged and only sent once. [17]

5.7 The DCF Aspect of the 802.11p Protocol Using CSMA/CA

Looking at the back-off procedure now in more detail, for the full DCF perspective that uses CSMA/CA, the process occurs in the following manner:

1. The node that has a data packet or frame to send, senses the wireless channel.
2. If the channel is available, the node starts to transmit the frame.
3. If the channel is unavailable, the node carries out a random back-off. An integer is chosen from a uniform distribution of [0, w], where w is the contention window of slots.
4. This integer is multiplied by the slot time, T (which is usually 13μs in IEEE 802.11p) and set as the back-off value.
5. The back-off value is decreased by one slot time, each time the channel is sensed as free.
6. When it hits zero, the data is sent immediately. [20]
7. If the channel is busy however, another random back-off time is chosen and the process is repeated.

The listening or sensing of the channel period is known as the Arbitration Interframe Space (AIFS). [20] It is also important to understand how the back-off time is generated. It is known as a truncated binary exponential back-off time. The set of contention window (CW) values increases in ascending integer powers of 2 minus 1, beginning with the set minimum CW value and increasing on each failed reattempt until it reaches the set maximum CW value. [21] These set values are explained in more detail in the following section.

So in mathematical terms, after c collisions the number of available back off slots in the contention window is given by $[0, 1, 2, ..., N]$ where $N = 2^c - 1$. This range of $[0, 1, 2, ..., N]$ is then treated as a uniform distribution and the back-off integer value is chosen.

5.8 The EDCA Aspect of the 802.11p Protocol Using CSMA/CA

In the EDCA access method, a node that wants to transmit will sense the channel. If the channel is clear for longer than an AIFS Access Category (AC) period, then the node will transmit. If the channel becomes busy, the node delays transmission, again by selecting a random back-off time. [18] In this case the back-off is performed as follows:

1. An integer is chosen from a uniform distribution of [0, w[AC]], where the initial w[AC] is w_{min}[AC].
2. The interval size doubles if the next transmission attempt fails, until w[AC] equals w_{max}[AC].
3. If no medium activity is sensed for one AIFS[AC] period then the wait time decrements by one slot time. The same laws of suspending back-off if the channel is deemed busy, apply in this case. The medium must then be idle for AIFS[AC] before the back-off resumes.
4. Finally, if the back-off reaches zero, the data is sent immediately.
5. If the medium is still busy, the node carries out the back-off again, however now w[AC] is left unchanged, once it has reached the maximum window size. This attempts to guarantee that important safety messages can be exchanged reliably and in a reasonable time. [18]

The EDCA mechanism is used for contention-based priority quality of service support, as previously discussed, and the four Access Categories give support for four priorities of data traffic. [29][30] As shown in Fig. 10, each AC queue contends for Transmission Opportunities on the channel by using its own unique EDCAF. [25] The EDCA parameters used in this function will be discussed now.

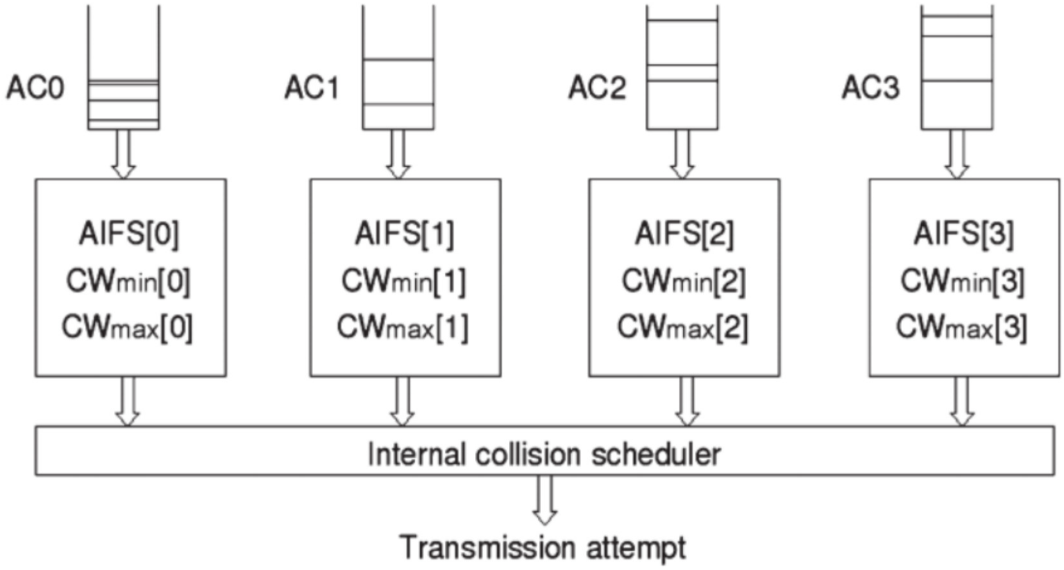


Figure 10: The Prioritisation Mechanism Inside a Single Station [25]

So there are four transmission queues and four independent EDCAFs for each AC. Now each AC has an Arbitration interframe space and this is denoted by AIFS[AC]. They also each have a different minimum and maximum for the Contention Window (CW) size and this is shown in Table 1. [25][31][32]

AC	**CWmin**	**CWmax**	**AIFSN**
AC_BK	aCWmin	aCWmax	9
AC_BE	aCWmin	aCWmax	6
AC_VI	[(aCWmin+1)/2] - 1	aCWmin	3
AC_VO	[(aCWmin+1)/4] - 1	[(aCWmin + 1)/2] - 1	2

Table 1: Default EDCA Parameters for the Different Access Category Operations [31]

The AIFS is essentially an addition to the backoff procedure described in Section 5.7 for the DCF procedure. [25][33]. The duration of the AIFS[AC] is found from the AIFSN[AC] number by using the following relation: [34]

$$AIFS[AC] = AIFSN[AC] \times aSlotTime + aSIFSTime \tag{1}$$

The AIFSN[AC] number is given as a default EDCA parameter and is set by each MAC protocol, as shown in Table 1. Different ACs have different AIFSNs, based on their priority levels. aSlotTime and aSIFSTime refer to the duration of a slot time and the length of the SIFS. These will both be estimated as parameters in testing shortly. Also of note, each channel access timer has a back-off function timer that contains the number of back-off slots. [25]

Finally, we can draw from this that the AC with a shorter AIFS has higher priority to access the channel. Furthermore, assigning a shorter CW size to an AC, means that said AC has a greater chance of accessing the channel than an AC with a longer CW. [25] This makes sense, since the former AC would be faster when reattempting transmission on the channel. Note lastly, that Transmission Opportunities and the role that they play will be described more closely when designing and testing the MATLAB models in Sections 10-14.

5.9 The Average Maximum Time to Complete CCH Transmissions

As additional research, a brief mathematical description of CCH transmission, is given here. *P(l,n,w,k)* in Equation 2 [19], gives the probability that n vehicles in the network, select back-offs from a contention window containing w slots. During this process there are *(l-1)* empty slots that pass, before there is an attempt at transmission and then k number of vehicles try to transmit in the lth slot. [19]

$$P(l,n,w,k) = (1 - \frac{l-1}{w})^n . \begin{pmatrix} n \\ k \end{pmatrix} (\frac{1}{w-l+1})^k (1 - \frac{1}{w-l+1})^{n-k} \tag{2}$$

The average maximum time for the CCH transmission to be completed in a set interval for all vehicles can then be calculated using Equation 3. [19] This assumes that there is a max limit of w slots left by the back-off counter when n nodes have not yet transmitted. Here T_s is the length of time taken up by a successful transmission and T_c is the length of time taken up by a collision. [19] The first term of the summation accounts for the successful transmission when only one node transmits in the lth slot, whilst the second term is when there is a collision between two or more nodes.

$$T(w,n) = \sum_{l=1}^{w} [P(l,n,w,1)[T_s + T(w-l,n-1)] + \sum_{k=2}^{n} P(l,n,w,k)[T_c + T(w-l,n-k)]] \quad (3)$$

However, if this result is analysed in detail numerically, it can be found that often the CCH interval is either wasted or conversely, too many safety related packages are dropped. [19] This point is further enforced in Section 6.2. Thus there is motivation for a channel-switching scheme that guarantees reliability and makes better use of the network.

5.10 The Probability of Packet Losses on CCH

Following on from the previous point, let *L(w,n)* be the average number of safety packages that are dropped due to collision. This leads to Equation 4. [19]

$$L(w,n) = \sum_{l=1}^{w} [P(l,n,w,1) * L(w-l,n-1) + \sum_{k=2}^{n} P(l,n,w,k)[k + L(w-l,n-k)]] \quad (4)$$

From this, using a recursive calculation, the probability of packet loss can be computed. This is shown in Equation 5. [19]

$$P(W,N) = \frac{L(W,N)}{N} \quad (5)$$

Substituting in numerical values, results have shown that sometimes more than half of the sent packets are dropped because of collisions. [19] This result would only worsen with more cars attempting to access the network. Lengthening the contention window could help this, but on the other hand this may increase delays. Since CSMA is not free from collisions, higher traffic levels worsen the mechanism. [19] Again, this provides further motivation for developing an improved MAC protocol.

6 Protocol Improvement Characteristics and Existing Techniques

6.1 A Summary of Current Protocol Improvement Techniques

There is currently a large amount of networking research that looks to improve the IEEE 802.11p MAC protocol. A selection of the ideas in these have been summarised in this section.

- **Extending the CCH interval:** Results show that this can develop the protocol, although high collision rates mean that reliability is still challenging. [19] However if this is extended appropriately when the vehicle density is less than a certain threshold, this improves collision rates, reliability, throughput and delay. [18]

- **Adapting the CCH and SCH intervals:** Conversely, reducing the CCH interval may improve the SCH service however, this does not tackle the high collision rate challenge mentioned above. [19]

- **SOFT MAC:** A Space-Orthogonal Frequency-Time MAC protocol has been proposed that theoretically can provide sufficient QoS levels. By using a random access period and allocating guaranteed transmission slots through reservation, it achieves a higher saturation (maximum) throughput than 802.11p. [14]

- **STDMA:** Self-Organising Time Division Multiple Access results are seen to outperform CSMA of 802.11p, by always allowing nodes access to the channel with an upper bound on access delay. This seems to work better even when the channel is not saturated. However, despite being fair with a predictable delay, strict synchronisation is necessary via a global navigation satellite system and self-organising needs periodic position messages to be present in the system. [20] These factors are not ideal.

- **Combining CSMA with TDMA:** Again it is argued that adaptively adjusting the length of the CCH and SCH intervals, improves the use of resources. Using

TDMA is said to improve both reliability and resource utilisation, with simulations showing that it outperforms the existing IEEE 802.11p protocol. [19]

- **Other TDMA based MAC Protocols:** These include ADHOC, a Cluster-based Medium Access Scheme, using collision-free time slots for data transmission and a Deterministic Medium Access scheme called VDA, where vehicles can access the medium with lower contention than in the IEEE 802.11p EDCA or DCF schemes. [19][35][36]

6.2 Protocol Characteristics with the Potential for Improvement

Before trying to improve the IEEE 802.11p protocol, it is important to take a look at current research test results. In this section results taken from a paper indicating general trends in data will be shown.

The 802.11p MAC protocol efficiency is impacted by several factors including, the message generation function, packet size of important safety messages, node density and the distance between nodes. This evaluation briefly discusses the protocol performance in terms of collisions, reliability, delay and throughput. [18] The figures were obtained by researchers using the OMNeT++ tool, and whilst experimenting with the CCH interval, but they give a good idea of overall data trends. [18]

Fig. 11a, shows the average collision probability as the vehicle density increases. It is clear that as the vehicle density increases, there is a much greater probability of a collision occurring since more vehicles are trying to transmit during the CCH interval. Decreasing the interval itself also increases competition. [18]

Fig. 11b shows the average MAC delay, which has a direct impact on end-to-end delays, again for different vehicle densities. The delay, similarly to the collision probability, increases as the vehicle density increases or if the CCH interval decreases. Both these factors would increase the back-off time in the protocol, which mainly causes the delay. The delay always appears to be less than 100ms. [18]

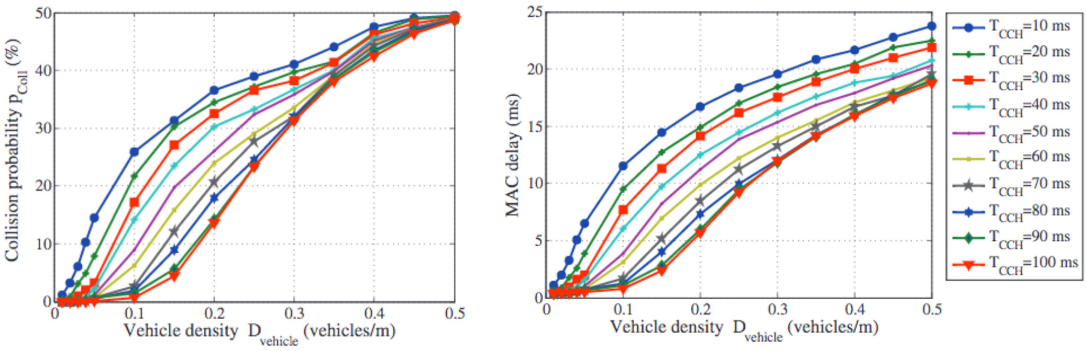


Figure 11: (a) Collision Probability and (b) MAC Delay against Vehicle Density [18]

Fig. 12a shows the delivery reliability of safety messages for varying vehicle densities. Here reliability decreases if there is an increase in vehicle density or a reduction in the CCH interval. Since reliability is closely linked to collision probability, this is to be expected. [18] Fig. 12b shows the network throughput for varying vehicle densities. [18] It can be seen that as the vehicle density increases, the network can start to saturate, however this process is slower for lower CCH intervals.

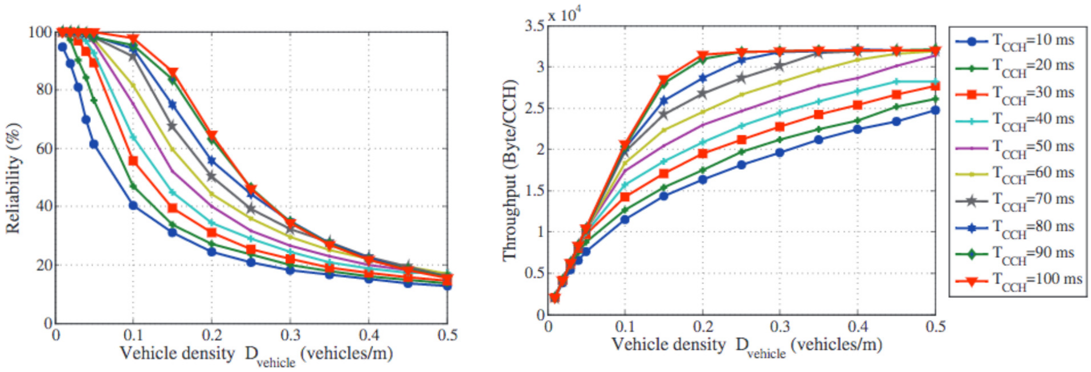


Figure 12: (a) Reliability and (b) Throughput against Vehicle Density [18]

In VANETs today, expired routing information is a large problem, given the high mobility of nodes. As mentioned earlier, reducing this end-to-end delay is of vital importance in emergencies. [3][7]

Applications	Priority Category	Allowable Latency (ms)	Network Traffic Type	Message Range (m)
Life-Critical Safety	1	100	Event	300
Safety Warning	2	100	Periodic	50-300
Electronic Toll Collection	3	50	Event	15
Internet Access	4	500	Event	300
Group Communications	4	500	Event	300
Roadside Service Finder	4	500	Event	300

Table 2: Example VANET Applications and their Allowable Latencies [4]

Table 2 details the allowable latencies given the VANET application, as well as the priority of the application and the message range in metres. [4] This delay is the length of time until a packet arrives at the correct destination after being sent. Queuing, propagation, channel access delay and retransmission or rebroadcasting can influence it. [16]

Low latency and high reliability requirements have already been emphasised as crucial for safety applications but results seen in Fig.11b suggest that the 802.11p MAC protocol can actually satisfy the requirements of less than 100ms, although reliability can vary. [18]

In further analysis, the impact on the system of changing the CCH interval settings will be investigated and throughput, latency and packet loss probabilities, will be looked at. This is important since it has been found that, given a set CCH, if safety messages are sent when there is a high vehicle density many packets will be lost to a high collision rate. At the other extreme, with very few vehicles the interval is too long and the time is wasted, as shown in Fig.11a. [18] To summarise, all of the criteria discussed above are essential to efficient protocols and will be investigated in more detail over the course of this project.

Interestingly, a class of messages exists, known as comfort messages that are relatively tolerant of delay. They only use a small proportion of the network bandwidth, which is generally reserved for emergency information. Research currently encourages the development of a Geocast routing protocol using these messages, however this area specifically will not be focussed on in this project. [3]

7 Project Motivation and Requirements Capture

7.1 Project Motivation

As previously mentioned in Section 4, there has been a huge amount of growth in demand in both the automotive and wireless communications industries in recent years. VANET systems have incredibly unique properties when compared to MANETs and are highly dynamic in topology. The network itself is also constantly adapting to the surrounding environments as the vehicles are in motion.

The efficient operation of VANET systems can lead to improvements in road safety by helping to prevent car accidents, potentially saving lives. Thus the investments by car companies in Advanced Driver Assistance Schemes are unsurprising. They can lessen road congestion and also increase driver comfort by providing media or application services.

The MAC sub-layer of the IEEE 802.11p standard is crucial to the effective operation of VANETs as it determines the method by which data packets are distributed through the network.

If packets are not transmitted fast enough then they may be dropped as the data becomes out of date compared to newly generated messages. This is a problem, particularly if the majority of those packets are safety messages. This issue only worsens as the number of sources or cars increases and so message priority needs to be taken into account when implementing the protocol. This is done via the EDCA mechanism and can be improved further by analysing the CCH and SCH channels and the impact that they have on the system.

Therefore it is incredibly important to gain a full understanding of how network protocols and control mechanisms can affect the latency, throughput and packet loss probabilities within the system, which is the focus of this project.

7.2 Project Aims and Objectives

This section summarises the aims and objectives that have been kept in mind throughout this project, over the course of the year. These have been listed below:

- Study the field of VANETs to a reasonable level of detail and understanding.
- Appreciate that VANETs have several unique features when compared to MANETs and that these should be exploited or investigated from the perspective of the MAC protocol.
- Understand how the IEEE 802.11p MAC protocol works and the main parameters that can affect the efficiency of its operation.
- Identify methods that could help improve the protocol for use specifically in VANET systems.
- Develop methods to model the IEEE 802.11p MAC protocol and implement these ideas using MATLAB.
- These methods include attempting to model the DCF CSMA/CA protocol and also the EDCA contention based channel access scheme to account for different priority levels of data.
- The MAC sub-layer protocol improvement focus will be to model and analyse the impact of dynamically adjusting the timing of the CCH and SCH intervals based on traffic load. This will then be compared to the current method of using fixed time intervals for each channel, to evaluate its success.
- Study the models with regards to latency, throughput and the probability of packets being dropped from the system, using a variety of perspectives to study the operation of the MAC protocol.

- Gain a deeper awareness of how the total system works.
- Regularly evaluate progress with regards to time frames and the Gantt chart and meet with the project supervisor at regular intervals throughout the year.

7.3 Structure of Work Throughout Year

The Gantt chart for the project has been given in Appendix A. This lists out the deliverables that were carried out each term. Project work was consistently scheduled into weekly work plans and so took into account heavy workload in other courses. The Gantt chart has been split according to the three university terms Autumn, Spring and Summer. The tables for each term show the start and end dates for specific tasks, as well as their duration. Each timeline is shown diagrammatically below the table for each term. The overarching tasks that were completed each term are given below:

- Autumn Term: Projects allocated, researched project topics, developed ideas on how to model the MAC protocol and how it could be improved.
- Spring Term: Focussed in on key project aims, implemented ideas in MATLAB and Simulink.
- Summer Term: Tested simulations and protocols. Continued collecting data and carrying out tests. Wrote up final project report and prepared for project presentation.

7.3.1 Amendments From Original Project Plan

The primary difficulty was in using the NS-2 protocol simulation software efficiently. Its heavy reliance on command line inputs, after a lengthy set up procedure, made it inaccessible and not ideal when creating system models. This risk was mitigated since all work was developed in MATLAB in any case. NS-2 was ultimately abandoned due to these factors and MATLAB became the focus for all following work.

Once the initially broad scope of the project had been narrowed down and clear objectives were set, the choice was made not to use the VISSIM traffic network simulator. This would have not directly contributed to the aim of the project and was also excluded for time constraint reasons. However it may be an interesting area to explore as further work.

Lastly, the majority of the testing was carried out in the Summer term, which was acceptable since adequate time margins were provided for this in the initial plan.

8 Analysis and Design

8.1 Modelling VANET Behaviour

Since 1999, when the Federal Communications Commission allocated the 75-MHz DSRC band at 5.9 GHz for vehicular communications, there has been a vast amount of research on the modelling and analytical evaluation of the IEEE 802.11p protocol. With cars as the highly mobile transceivers in the system, there are two main methods of modelling. [37]

The first method involves 802.11p models that are based on simple vehicle traffic assumptions such as even car spacing, but leave out features such as road environments or traffic lights. Thus the models do not encompass the overall variations in network performance along a road in a real-life setting. [37]

The second method uses simulations to profile road traffic and network communication traffic for performance evaluation, such as MATLAB and VISSIM. These can be computationally time consuming and so unsuitable for fast predictions, however they are reliable as performance indicators. [37] This is the category into which much of the analysis for this project will come under. An overview of the methodology behind VANET modelling is shown in Fig. 13.

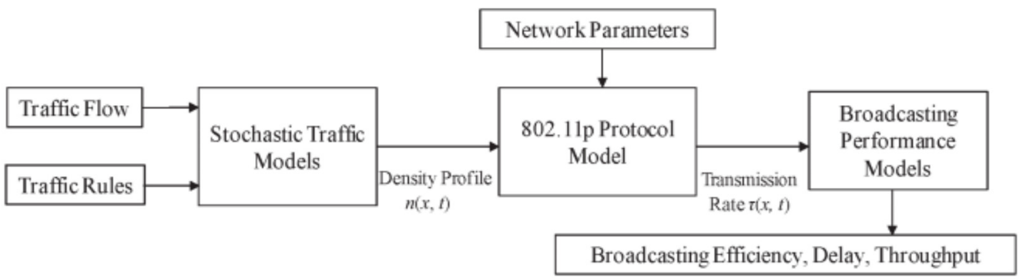


Figure 13: Overview of the Methodology for 802.11p VANET Broadcasting Performance Modelling [37]

Simulating VANETs accurately from start to finish requires two features: a road traffic simulator to model the cars as nodes in the network and a network simulator to analyse the behaviour of the wireless network. [3] Realistic models can include lanes and directions, traffic, weather conditions, roadblocks and driver behaviours. [3]

Models are crucial in network analysis as they provide data on any form of communication and at any network level. Results are usually fast and cheap, so that work can be rapidly improved. Although it may be said that simulations lack realism and should not be depended on, they allow scenarios to be analysed with far less effort than by using real equipment. [38]

It is impractical to immediately set-up and test VANETs, so simulations are in fact necessary to test protocols in the WAVE stack. Many of the most accurate VANET simulation tools for the IEEE 802.11p protocols are not free or available publicly. NS-2 Network Simulator is free and is a widely accepted software that can simulate DSRC/WAVE protocols, although it is not necessarily the strongest tool when modelling the multichannel feature of the IEEE 802.11p protocol. [16] For the purposes of this project, MATLAB modelling will suffice.

Usually packet simulators use simple transmission models that are related to the appropriate distribution functions, to give the probability of successful packet reception. However for VANETs this approach is more complicated since often, reliable model results depend upon accurate simulations of the PHY network level. [38] Assuming high reliability and low bit error rates for packet transmission, removes the need to account for this feature in great detail.

8.2 Types of Simulators for Modelling VANETs

Over the years, three types of VANET simulators have been developed:

- **Mobility Simulators:** Software environments that create vehicle movement models.

- **Network Simulators:** These can test network protocols in terms of performance.

- **Integrated Simulators:** These are a combination of both the mobility and network simulators. [3]

This project will use MATLAB as a network simulator to model the key network protocols. MATLAB is a high level language that has widespread use across all areas of academic research. It is a scientific computing tool that is useful for system design and graphing. It is highly compatible and can be interfaced or integrated with work in other programming languages or Microsoft office applications. In this project it is used to model two protocols and a resource allocation control mechanism. [39]

Aside from MATLAB, there are also a whole host of alternative network simulators, which should be mentioned as an aside. These are as follows:

- **NS-2:** Object-oriented network simulator for wireless networks that is widely used by academics as it is a free-open source software. [40][41] Users write TCL scripts to develop the simulation environment. [3][11]

- **OPNET:** Commercial simulator for wired and wireless networks. [3]

- **OMNeT++:** Supports network and mobility models and is an object-oriented modular discrete event simulator. It has a component-based design so that new features and protocols can be added via modules. [3] The PHY layer simulation works similarly to the way it does in NS-2. [38]

- **SUMO:** Simulation of Urban Mobility can accurately simulate vehicle behaviour, including road features and traffic details. It is a professional software that is often used with a tool called Vanet-sim. [11]

- **TraNS:** Traffic and Network Simulation Environment is a bridge between SUMO and NS-2 and provides a more realistic setting for simulating VANETs. Like NS-2 it is open source software. [11]

- **Other Professional VANET Simulators:** NCTUns, GrooveNet [16]
- **Other Network Simulators:** QualNet, GloMoSim, NS-3 [16]

Note that although NCTUns and NS-3 may be able to better model the IEEE 802.11p protocol, MATLAB will be sufficient to achieve the aims of this project. [16]

8.3 Generalised Model of the Overall System

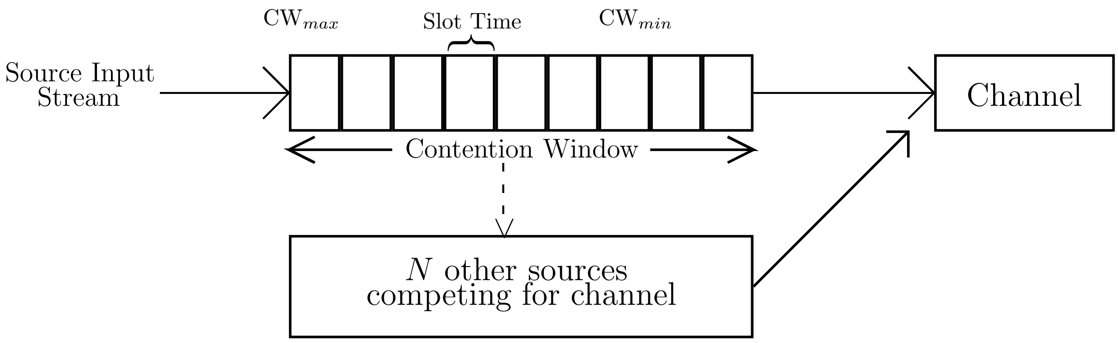


Figure 14: Simplistic Model of Contention for Channel Access

A generalised model of the system allows for, a better overall understanding of the processes occurring, albeit from a simplistic perspective. Fig. 14 shows the base model structure of how sources compete for access of the channel and will be used for the majority of the upcoming modelling. An example location for CWmin has been given, to indicate that the contention window size can vary based on how many collisions have occurred, as is true according to the exponential back-off calculation. So for an incoming source, the channel is either sensed to be busy or idle and so either a back-off slot can be chosen or transmission can occur straightaway.

The designed MATLAB model, attempts to simulate the CSMA/CA aspect of the DCF protocol and then further this model to incorporate the Access Categories for user data priority that are found in the EDCA mechanism. The channel is modelled using two methods, the first simply allocates a probability to the channel being busy and the second uses an sample array of values for a channel model showing it to be in either a busy or idle state at any given time.

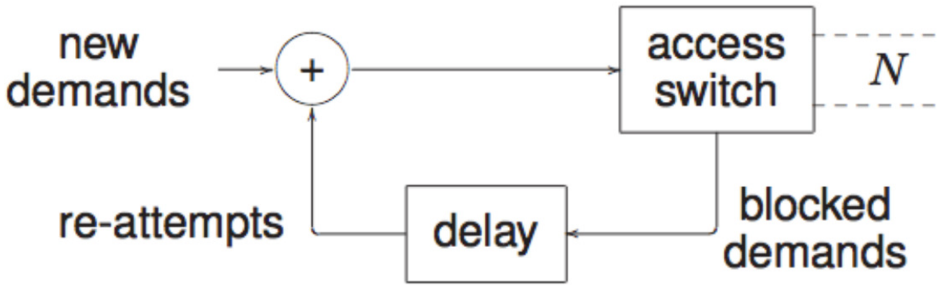


Figure 15: Model for Reattempts of Packet Transmission [42]

The models are based on assumptions regarding parameters such as slot time, or time between arriving packets. A particularly important assumption that is made for all of the models is the reattempt model used for the system. In most general traffic theory models, the rate of new demands or packet arrivals to the system can be treated as a Poisson Distribution with a mean arrival rate of λ, which is used in the upcoming models. However, if re-submission of blocked attempts were considered, as in Fig. 15, then this would result in a non-Poisson arrival stream. This is highly likely to occur under heavy traffic conditions. By making the crucial assumption that re-submissions do occur but in a regular manner, then the total offered load to the system can simply be increased and each individual reattempt does not need to be considered. [42]

8.4 Design and Structure of the Models

The five models that were designed and implemented in MATLAB will now be summarised below, before they are each analysed in further detail. In order to have a starting point for a couple of the models, base parameters were chosen as given in Table 3. These are parameters that were used in a research paper using NS-2 to model the IEEE

802.11p protocol. They formed a good starting point from which ideas were developed and parameters varied. It also ensured that any variations were kept to a reasonable proportion and were close to realistic values. Note that the maximum and minimum values for the contention window and the AIFSN numbers in the EDCA mechanism are fixed for each Access Category and these are described further on in this section.

Parameter	**Value**
SIFS	$10\mu s$
Slot Time	$20\mu s$
CW Minimum	31
CW Maximum	1023
Retry Limit	5
Data Rate	1MB/s

Table 3: Example NS-2 IEEE 802.11p Simulation Parameters [4]

Model 1: The CSMA/CA Protocol Used in the DCF

This model is the most fundamental and important of all when modelling the MAC layer. The CSMA/CA protocol forms the basis of the DCF and the algorithm for this was implemented as shown in Fig. 16.

In this code, the likelihood of the channel being busy was defined as a probability, rather than by having separate busy and idle states for it. Active modelling of the channel was left aside until model 2. The model then studies the latency and throughput for one packet as the channel busy probability is increased. The number of packets lost is also studied.

Model 2: An Improved Version of the CSMA/CA Protocol Used in the DCF

This model was developed as an improvement on the basic structure of model 1. It uses three different models for the channel by using vectors that are 1000 values in length.

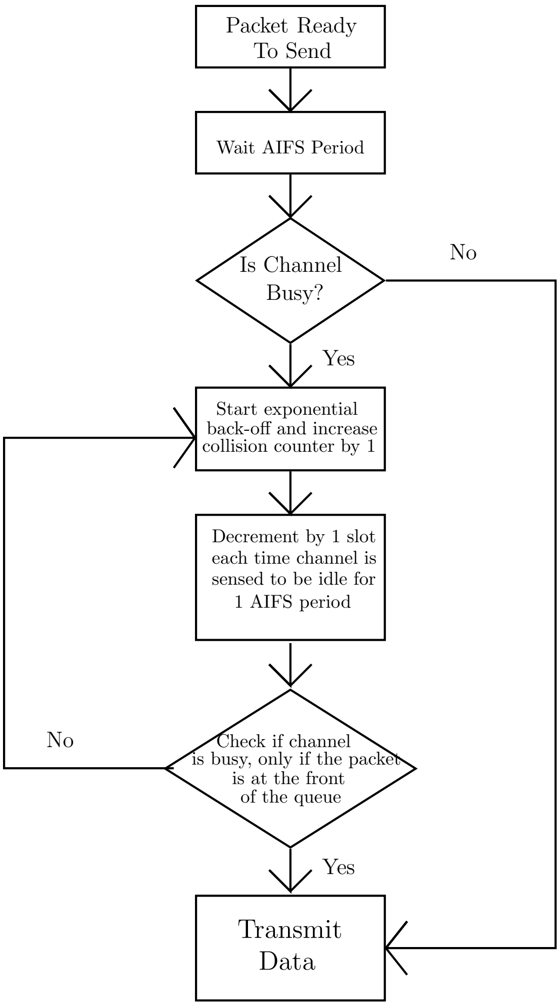


Figure 16: Algorithm Used for CSMA/CA Implementation

Each cell is either busy or idle and the three channels have the probability of the channel being busy set to 0.25, 0.5 and 0.75 respectively. Each integer is modelled as a slot time frame of $20\mu s$ so that the model can have an element of actual timing rather than simply having a probability. This model looks at the latency for each packet number and the cumulative latency over a set of packet transmissions. It thus studies how long it takes one packet to go through the system.

Model 3: The EDCA Mechanism Used in the HCF

It was decided to model the EDCA mechanism since this is widely used in the field of VANETs and forms the basis for user priorities of data packets and is implemented in both the HCF and MCF. The system now looks at the four Access Categories and takes data priority into account. The minimum and maximum contention window sizes can be calculated for each class using the formula given in Section 5.8 of this report and this gives the results in Table 4.

AC	**CWmin**	**CWmax**	**AIFSN**	**Max TXOP (ms)**
AC_BK	15	1023	7	0
AC_BE	15	1023	3	0
AC_VI	7	15	2	3
AC_VO	3	7	2	1.5

Table 4: Calculated EDCA Parameters for Each Access Category [34]

This builds on models 1 and 2 in the sense that it incorporates each unique AIFS time for each Access Category into the DCF model. The probability of the channel being busy is measured as a probability like in model 1 and in this case the packets lost are observed. There is also a look into how much the transmission opportunity time allocation varies with channel busy probability. The mean throughput also varies with transmission opportunity time allocation and this will be investigated.

Model 4: An Improved Version of the EDCA Mechanism Used in the HCF

Model 4 was developed as an improved version of model 3 where the improvement is similar in structure to the improvement made to the CSMA/CA protocol between models 1 and 2. Again, there are three different models for the channel created by using vectors with same channel busy probabilities of 0.25, 0.5 and 0.75. The time frames are also still 20μs. This model also looks at the latency for each packet number and the cumulative latency over a set of packet transmissions, therefore studying how long it takes one packet to be transmitted.

Model 5: A Dynamic Time Boundary for the CCH and SCH Subintervals

This model was chosen as the main route to investigate an improvement to the MAC sub-layer control mechanisms. The resource allocation model adaptively adjusts the time boundary of the CCH and SCH channels based on the previous time interval. The results of this are compared to the standard case with no dynamic adjustment. Estimates are decided on for the length of time taken by a packet to be processed and also the maximum packet size.

The model of the channel is split into two components, the traffic on the CCH and the SCH channels respectively. The two channels are designed to have a specific volume of traffic on each of them, which is described in more detail and shown on a graph in Section 14. The model chooses the time boundary primarily based on the safety channel statistics and then chooses the service channel subinterval length after this. For this model the quantity of packets that are dropped, over 220 allocated time intervals of channel transmission, is investigated.

9 Overall Testing Aims

Each of the two protocol models and the resource allocation control mechanism will be tested when they are analysed. This allows for comparison between the three scenarios and a thorough evaluation in which a further understanding of the relationships between the systems can be developed. Any assumptions made will be explained for each of the models in the upcoming sections, before they are implemented and tested.

Each system perspective will be assessed with the view to explore certain criteria. The three that will be focussed on are given below:

- **Low Latency:** The time delay in the receiving of data packets after they have been sent should ideally be as low as possible. There are certain minimum allowable latencies depending on the VANET application and this has been discussed in Section 6.2.
- **Low Packet Loss Probabilities:** This ties into the point above, since if packets take too long to be transmitted, they most likely contain outdated information compared to newer data contending for channel access. After reaching a retry limit the older packets will be dropped from the channel. The likelihood of this occurring should be minimised.
- **High Throughput:** Trying to maximise the throughput of the system is essential to improving the service provided by the MAC sub-layer. Additionally, the greater the vehicle density can be before reaching network saturation, the better.

Note that the relative importance of these factors will vary depending on the scenario. If messages are being sent regarding vehicle safety then there may not be a large amount of data being sent, however is it crucial that there is low latency and high reliability of transmission. [4] It must be added that certain cases will not be taken into account and model limitations will be detailed for each perspective. Additionally there were factors such as unpredictable driver behaviour or exact location based road tests that were not looked into, although will be discussed in possible further work.

For the purposes of this project and to limit the complexity of the models to a reasonable level, the transmission reliability of the channel will be assumed to be sufficiently high. This is a reasonable assumption to make since an adaptive modulation technique that was proposed for the IEEE 802.11n standard selects the most ideal modulation or coding scheme once it has evaluated the channel conditions. It also chooses the modulation scheme based on the performance required by the MAC layer for different applications. [43] This proves promising to increase the transmission reliability of the channel.

Graphs of results for the factors such as throughput and latency will be compared. This gives a clear framework for evaluating the success of each method and allows for a deeper analysis of each model. In addition to this analysis, it will be necessary to assess where model limitations exist if they were to be implemented in reality.

10 Model 1: The CSMA/CA Protocol Used in the DCF

10.1 Implementation

As previously discussed, Model 1 implements the CSMA/CA Algorithm that is the foundation of the Distributed Coordination Function. The main algorithm that was implemented in MATLAB is shown in Fig. 16. The model analyses the perspective of one source and assumes there is an infinite number of other sources attempting to transmit on the channel. The probability of the channel being busy is varied from 0.01 to 0.99 in 0.01 increments and latency, throughput and the number of dropped packets on each reattempt are then measured. The full code for this model can be found in Appendix B, however certain important snippets have been explained below.

```
idle = rand(1);
        while (idle < p_channel_busy(i) && c<retry_limit)
            c = c + 1;
```

The 'idle' variable chooses a random variable from a uniform distribution between 0 and 1 and then compares this to the probability of the channel being busy and checks whether the retry limit of 10 has been reached. The number of collisions, 'c' is then incremented by 1.

```
    if (2^c - 1) < min_window
        backoff = min_window;
    elseif (2^c-1)>max_window
        backoff = max_window;
    else
        backoff = 2^c - 1;
    end
slot_choice = round(backoff*rand(1));
```

The size of the Contention window is then found using the collision number. If it is less that the minimum size, it is set to the minimum window size, likewise if it is greater than the maximum window size then it is set to the maximum window size. Otherwise it is left alone. The slot choice was then found by multiplying this back off by a random number from a uniform distribution, before rounding it to a whole number. This code was later made into a separate function that was used in later models. This function code can be found in Appendix C.

```
totalLatency(n,i) = totalLatency(n,i) + (c+1)*58;


meanTotalLatency    = mean(totalLatency);
meanTotalThroughput = packetSize./meanTotalLatency;
```

The total latency was updated on every iteration of the 'for' loop, it was found by taking the slot choice and multiplying this by the slot time. It is recalculated outside of the loop to take into the AIFS time. In this case 58μs has been used as an arbitrary reasonable value. The mean of the latency was then calculated, along with the throughput mean. This was found by dividing the packet size by mean total latency.

```
if c==retry_limit
            packetsLost(i) = packetsLost(i)+1;
        end
```

Finally if the number of collisions reached the retry limit then the number of packets dropped was incremented by 1.

10.2 Testing

There are several assumptions made in this model and these are detailed as follows:

- The slot time was set to be 20μs. This was chosen from research to be a reasonable value.
- A constant packet size of 300 bytes was assumed. Again this was chosen from research and kept constant to keep the model at a reasonable level of complexity.

- A channel link size of 1 was chosen, which means that competing sources are fighting for access of 1 channel.
- The transmission opportunity time was naturally set to 0, since TXOPs are not present in the DCF protocol.
- It was decided not to overly complicate the model by incorporating the Request-To-Send/Clear-To-Send mechanism. This is because packets are dropped anyway if they are not sent fast enough and additionally, we are assuming that there is a sufficiently low packet bit error rate. It also slows down the entire process.
- The retry limit was set to 10, so that there was a finite number of reattempts.
- A trial repetition number of 1000 was used.
- Minimum and maximum CW sizes of 31 and 1023 respectively were used, which stays true to the values used for the DCF protocol.
- The probability of the channel being busy was varied from 0.01 to 0.99, using increment values of 0.01.

The average latency and the average throughput were found over the range of channel busy probabilities and plotted as seen in the Results section. Furthermore, the throughput is the throughput of the input source and not of the channel. Since we are assuming infinite other sources, the channel throughput is related to the busy probability.

10.3 Results

Fig. 17 shows that the log of the latency increases almost linearly as the probability of the channel being busy increases. This is to be expected since the busier the channel, the longer packets will spend reattempting to access the channel. The log of the total latency has been taken since otherwise there is such an exponential increase in the delay sizes for channel busy probabilities between 0.9 and 1, that the rest of the delay values become so small in comparison that the graph plot becomes void of meaning.

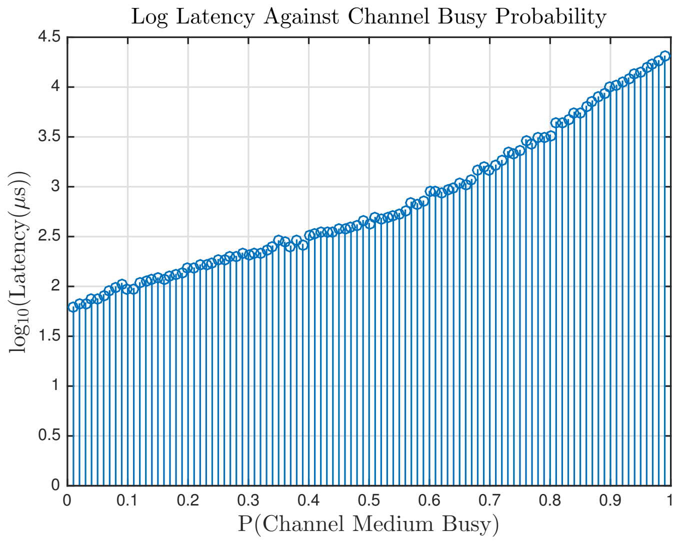


Figure 17: Plot of the Log Latency Against Channel Busy Probability

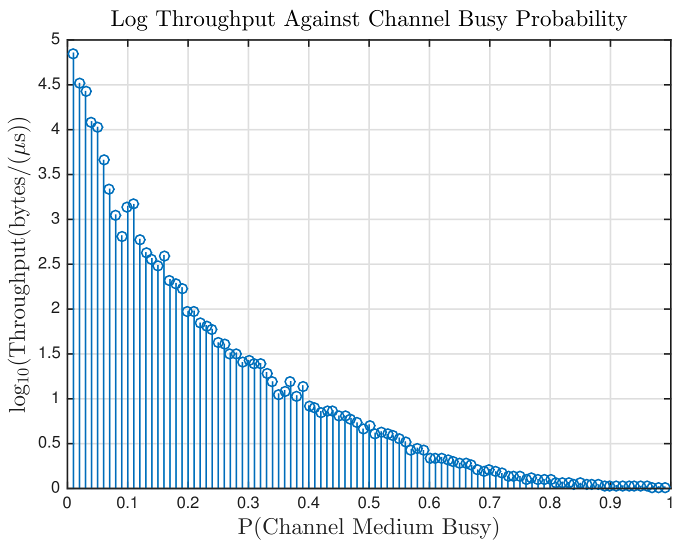


Figure 18: Plot of the Log Throughput Against Channel Busy Probability

Fig. 18 shows the log of the throughput as the probability of the channel being busy increases. It is important to note here that since a constant packet length of 300 bytes was chosen, the throughput is only dependent on the latency, according to the formula described in the Implementation section. As the packet length does not determine the graph trend, the throughput is seen to be inversely proportional to the latency. This is again to be expected.

Fig. 19 shows the number of packets lost against the probability of the channel being busy. This is not a log plot, as each value is a proportion of the 1000 trials that were attempted on each run of the model. The graph shows a steep increase in the number of packets lost as the channel becomes busy and this follows the theory described. It appears to be almost an exponential increase, with over 900 packets lost when the channel has a 0.99 probability of being busy, compared to less than 100 packets lost for channel probabilities of 0.75 or lower.

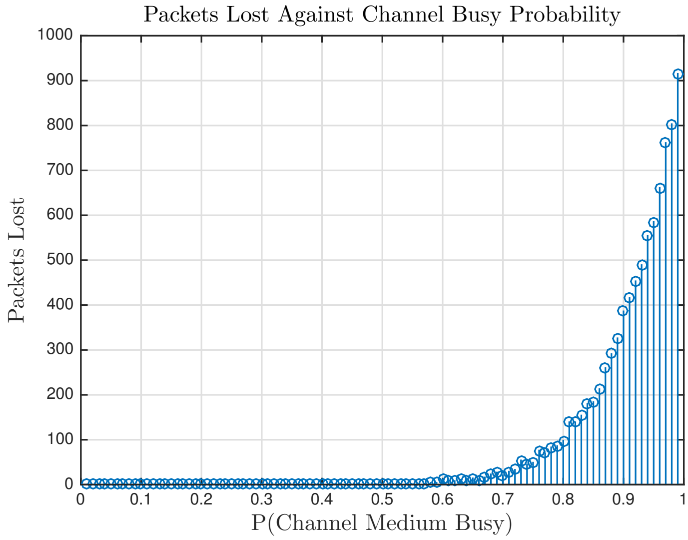


Figure 19: Plot of the Number of Packets Lost Against Channel Busy Probability

11 Model 2: An Improved Version of the CSMA/CA Protocol Used in the DCF

11.1 Implementation

As previously discussed, Model 2 implements an improved version of the CSMA/CA algorithm for three different levels of channel busy probability. The model now takes into account only decrementing the slot time when the channel is free to give a scenario that is closer to the reality and the algorithm shown in Fig. 16. This again views the perspective of one source and assumes that other sources cause the channel to have the busy-idle pattern that it has. In reality a shuffle function in MATLAB is used to create the channel pattern of ones and zeros for any given probability level.
The basic structure of the code is as follows:

- Check if the channel is busy, if it is busy then:
  - If nothing is in the CW, then put the data in the CW
  - If something is in the CW then leave it where it is
- If the channel is free then:
  - If nothing is in the channel then transmit the data
  - If there is a packet in the CW, then decrement the slot time

In this code, graphs of the latency for each of 1000 trials and then the cumulative latency for each of the channel probability levels of 0.25, 0.5 and 0.75 have been plotted. The full code for this model can be found in Appendix D, however certain important snippets have been explained below.

```
if busyFlag == 1
    if isCWFree == 0
        c = c + 1;
        backoff = ChooseCWDelay(min_window, max_window, c);
```

```
            isCWFree = 1;
        end
```

If the channel is busy and there is nothing in the contention window, then increment the number of collisions, choose a random back-off time and place the data in the contention window. Then set the 'isCWfree' flag to 1, to acknowledge that data is present in the contention window.

```
    elseif busyFlag == 0
        if ReachedFrontQueue == 1
            isCWFree = 0;
            packetTransmitted = 1;
        end
```

Alternatively, if the channel is free and a packet has reached the front of the queue then acknowledge that a packet has been transmitted successfully and reset the CW to being empty.

```
        if isCWFree == 0
            packetTransmitted = 1;
        elseif isCWFree == 1
            backoff = backoff - 1;
            if backoff == 0
                ReachedFrontQueue = 1;
            end
```

If the channel is not busy and the contention window is free, then transmit the data. If the channel is not busy, but there is data in the contention window then decrement the slot time of the data. If the slot time back-off now reaches zero, then the packet has reached the front of the queue.

```
totalLatency(j) = (i-1)*20;
```

The total latency calculation is seen here and was worked out as the number of slots of the modelled channel used, multiplied by the length of the slot time of 20μs. This can be done since each data value of the channel has been chosen to last the length of one slot time, to simply the process whilst still maintaining a realistic model.

```
if removePacketsDropped == 1
        totalLatency(totalLatency > 4000) = 0;
    end
```

Importantly, there is a flag called 'removePacketsDropped' present in the model. When the simulation is run with this set to zero, it will be seen in the results section that those packets that fail to be transmitted and have incredibly high latencies skew the data. Setting this to 1, allows the dropped packets to be removed from the graph data and stop it from skewing.

11.2 Testing

There are several assumptions made in this model and these are detailed as follows:

- The slot time was set to be 20μs. This was chosen from research to be a reasonable value.
- A constant packet size of 300 bytes was assumed. Again this was chosen from research and kept constant to keep the model at a reasonable level of complexity.
- A channel link size of 1 was chosen, which means that competing sources are fighting for access of 1 channel.
- The transmission opportunity time was naturally set to 0, since TXOPs are not present in the DCF protocol.
- It was decided not to overly complicate the model by incorporating the Request-To-Send/Clear-To-Send mechanism. This is because packets are dropped anyway

if they are not sent fast enough and additionally, we are assuming that there is a sufficiently low packet bit error rate. It also slows down the entire process.

- The retry limit was set to 10, so that there was a finite number of reattempts.
- A trial repetition number of 1000 was used.
- Minimum and maximum CW sizes of 31 and 1023 respectively were used, which stays true to the values used for the DCF protocol.
- The first value of each channel model was set to 1, to assume that the packet saw the channel as busy on arrival and was not sent through immediately, since the channel is not refreshed on each run of the loop. Importantly, this likely leads to a slight overestimation of the latencies at the start of the procedure.
- Each data value of the channel has been chosen to last the length of one slot time, for a realistic scenario whilst still keeping the model at a reasonable level of complexity.

The latency for each of the 1000 trials and also the cumulative latency for each of the channel probability levels of 0.25, 0.5 and 0.75 will now be shown in the results section.

11.3 Results

For all of the following graphs, the colour key is given below:

Black = Channel Busy Probability of 0.25
Blue = Channel Busy Probability of 0.5
Red = Channel Busy Probability of 0.75

Fig. 20 shows the latency against packet number when dropped packets are not removed. It is clear here, the extent to which the latency of dropped packets has an impact on system efficiency, even when the retry limit is set to 10. This is the reason the condition

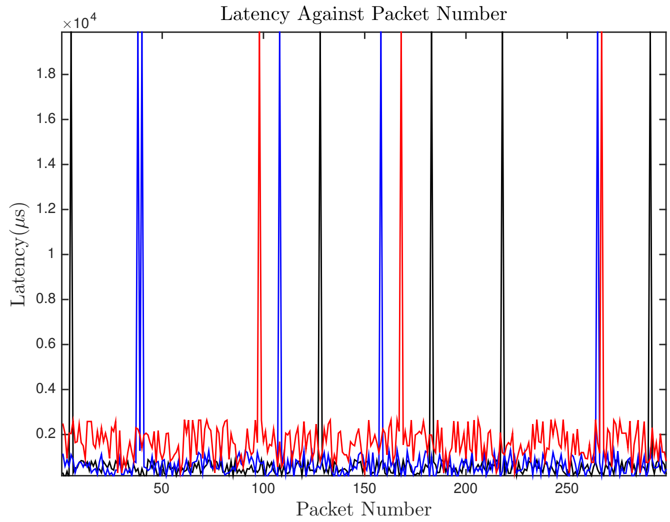


Figure 20: Plot of Latency Against Packet Number When Dropped Packets are not Removed

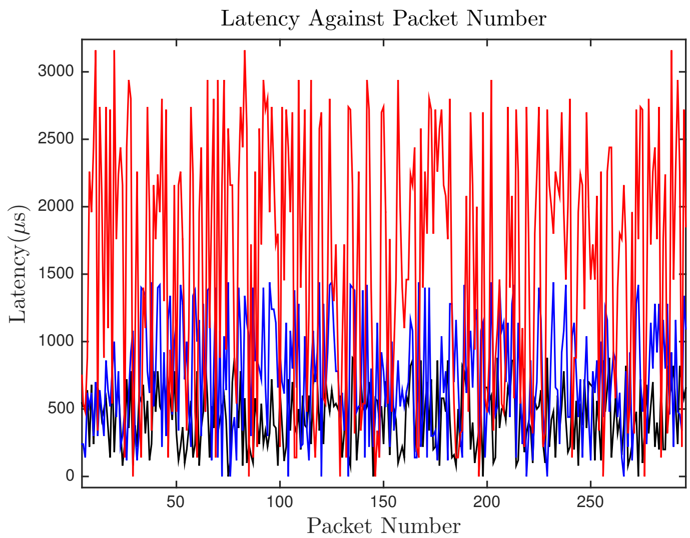


Figure 21: Plot of Latency Against Packet Number When Dropped Packets are Removed

was placed in the code to remove these spikes from the data to see the trend more clearly.

This new version of the graph plot can be seen in Fig. 21. Now it is clear that the general latency of the packets being sent increases as the channel busy probability increases.

This is to be expected and we can quantify estimates of these results. When the channel busy probability is 0.25, the average latency is approximately 250μs, for 0.5 it is 750μs and for 0.75 it increases to 1500μs. This shows that when the probability is doubled from 0.25, the average latency triples and when the probability is tripled, the average latency is six times that of the 0.25 level. Average latency does not increase linearly as channel busy probability increases.

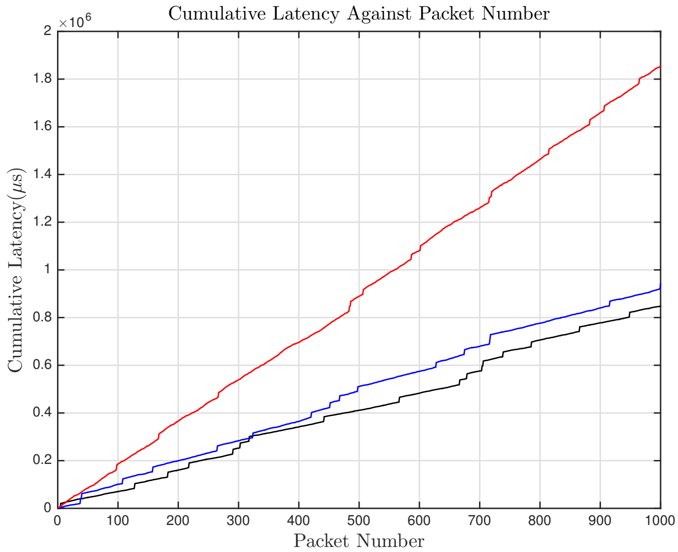


Figure 22: Plot of Cumulative Latency Against Packet Number When Dropped Packets are not Removed

Fig. 22 shows the cumulative latency against packet number when dropped packets are not removed. Here we can see that the lower the channel probability the lower the cumulative latency for all 1000 trials, as expected from theory. The lowest line is a probability of 0.25, followed by 0.5, then the highest line of 0.75, which has a cumulative latency of over 1.8x$10^6$$\mu$s. The spacing between the lines, which is far greater between 0.75 and 0.5 than it is between 0.5 and 0.25, also reinforces that latency does not increase linearly with probability.

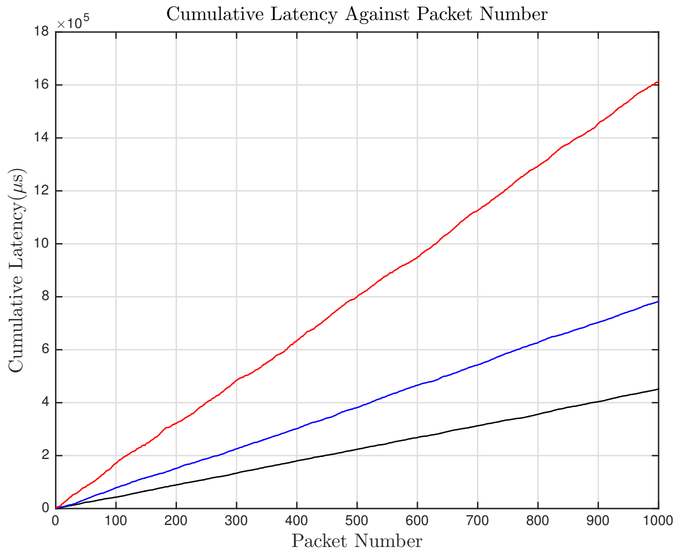


Figure 23: Plot of Cumulative Latency Against Packet Number When Dropped Packets are Removed

This graph trend does not change when the dropped packets are removed from the data, as shown in Fig. 23. However, it can be seen that the cumulative latencies are now lower, with a probability of 0.75 giving 1.6x$10^6$$\mu$s, with similar drops in the 0.5 and 0.25 lines. This shows how much of an impact removing the dropped packets has on the numerical data, rather than on the trend of the graph itself.

12 Model 3: The EDCA Mechanism Used in the HCF

12.1 Implementation

Model 3 implements the EDCA protocol that plays a crucial role in the Hybrid Coordination Function by allowing for User Priority by using the four Access Categories. These are referred to in the upcoming sections as follows:

1st AC = Background
2nd AC = Best Effort
3rd AC = Video
4th AC = Voice

These can be referred back to Table 4 to see the minimum and maximum contention window sizes for each AC, as well as the AIFSN and maximum TXOP lengths for them. When the AC obtains a Transmission Opportunity, it has complete access to the channel for that length of time. Note that ACs 1 and 2 do not have any TXOP time, as they are low priority and therefore can be treated quite separately to the modelling of ACs 3 and 4.

Here, the time allocated to TXOPs was modelled as varying with the channel busy probability according to the adapted cubic curve, which is shown in the results section. This means that for each fixed channel busy probability there is a fixed probability of a TXOP occurring. The variable 'TXOP_Flag' was then chosen as a random number between 0 and 1 and compared to this TXOP probability. If 'TXOP_Flag' was lower than the TXOP probability, then a TXOP occurred for that AC.

The model for ACs 1 and 2 are the same as the code used for model 1, in Section 10 and will therefore not be re-explained. The only difference between AC1 and AC2 is that they have different AIFS times and we shall see how that impacts the results.

Model 3 varies the probability of the channel being busy from 0.01 to 0.99 in 0.01 increments and latency, throughput and the number of dropped packets on each trial are then measured. The full code for this model can be found in Appendix E, however certain important snippets, that are related to the codes for managing the TXOP aspect in ACs 3 and 4, have been explained below.

```
TXOP_Prob = (p_channel_busy-0.5).^3 + 0.5;
TXOP_Prob = k*[TXOP_Prob(floor(N/2)+1:end),TXOP_Prob(1:floor(N/2))+0.2352];
TXOP_Prob = TXOP_Prob./max(TXOP_Prob);
Time_Unit = 200;
```

Here the graph plot defining the way in which the TXOP occurrence probability varies with the channel busy probability has been defined. It was created by plotting a cubic curve and then splitting it part way. The split part was then attached onto the other side of the curve. The final result can be seen in Fig. 26.

```
ScaleToTime_Prob = min (( totalLatency(n,i)-
LatencyWhereItOccurs )/Time_Unit, 1)*TXOP_Prob(i);

TXOP_Flag = rand(1);

if TXOP_Flag < ScaleToTime_Prob
    LatencyWhereItOccurs = totalLatency(n,i);
    TXOP_TimeLeft = MaxTXOP(j)*1000 - slot_choice*slot_time;
    packetCount(n,i) = packetCount(n,i) + dataRate*(TXOP_TimeLeft);
    TXOP_Time(n,i) = TXOP_Time(n,i) + MaxTXOP(j)*1000;
end
```

In this section of code, 'ScaleToTime_Prob' shows the probability that a TXOP occurs in the time that has passed since the last TXOP. 'TXOP_Flag' is then set to a uniformly distributed random value, as previously mentioned. Then if the 'TXOP_Flag' is less than the 'ScaleToTime_Prob' then a TXOP is begun. This ensures that at low probabilities of the channel being busy, a TXOP is less likely to occur. Then the model updates the time for when the last TXOP has occurred and stores this in the variable 'LatencyWhereItOccurs'. Next it calculates the real time spent in the TXOP, which is the maximum time length of the TXOP minus the time the packet in the contention window takes to reach the front of the queue. The following line updates the packet count, which is proportional to the incoming data rate. Finally it updates the total time spent in the TXOP, which is measured in microseconds.

```
TotalTime = mean(TXOP_Time + totalLatency);
maxRate = dataRate*ones(1,length(p_channel_busy));
```

In this last code snippet, the average total time is found, so that the experimental throughput can be calculated and plotted. The throughput is found as the mean packet count divided by the total time. This is then compared to the maximum possible throughput, which is calculated in the second line using the maximum data rate.

12.2 Testing

There are several assumptions made in this model and these are detailed as follows:

- The slot time was set to be 20μs. This was chosen from research to be a reasonable value.
- A constant packet size of 300 bytes was assumed. Again this was chosen from research and kept constant to keep the model at a reasonable level of complexity.
- A channel link size of 1 was chosen, which means that each AC is fighting with other sources for access of 1 channel.
- It was decided not to overly complicate the model by incorporating the Request-To-Send/Clear-To-Send mechanism.
- The retry limit was set to 10, so that there was a finite number of reattempts.

- A trial repetition number of 1000 was used.
- The CW minimum and maximum values, the AIFSN values and the TXOP values for all ACs were kept as in Table 4.
- The SIFS time was set to be 10μs as research found this to be a reasonable value and the maximum data rate was set to 5 bytes/μs.
- The AIFS values were calculated as: AIFS = (Slot time x AIFSN) + SIFS
- The probability of the channel being busy was varied from 0.01 to 0.99, using increment values of 0.01.

12.3 Results

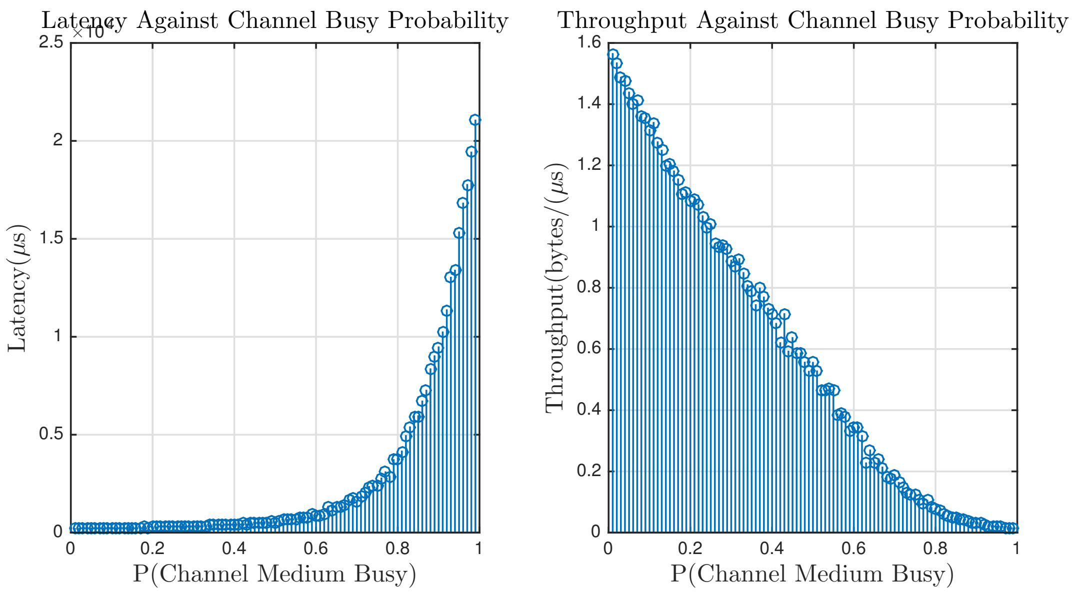


Figure 24: Plots of the Latency and Throughput Against Channel Busy Probability for the 1st Access Category

Fig. 24 shows plots of the latency and throughput against the probability of the channel being busy for the 1st Access Category. The latency shows an exponential increase as the channel busy probability increases, which is to be expected since ACs 1 and 2 act so similarly to model 1, as they both have a TXOP of zero. At a channel busy probability of 0.99, the latency is as high as 2 x $10^4\mu$s. Conversely the throughput decreases steadily from approximately 1.6 bytes/μs, which again is unsurprising.

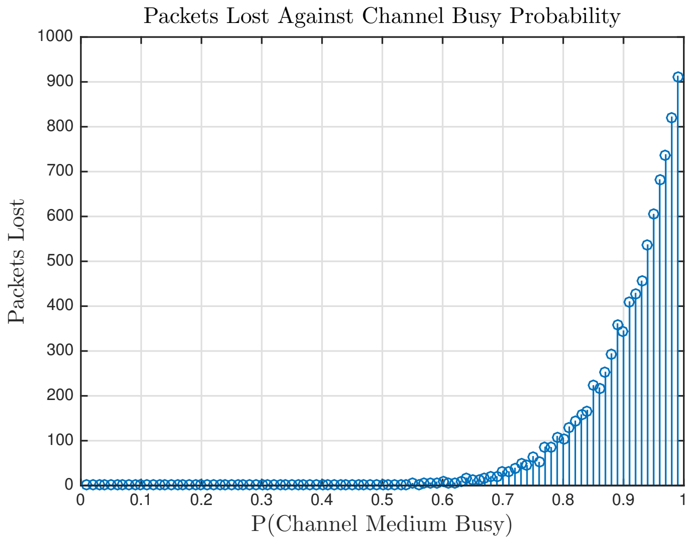


Figure 25: Plot of the Packets Lost Against the Channel Busy Probability for the 1st Access Category

Fig. 25 shows the number of packets lost as the channel busy probability increases, again for the 1st Access Category. This increases exponentially and reaches over 900 packets lost when the channel has a 0.99 probability of being busy. The graph plots for AC2 follow the same trend and have been given in Appendix F. This is natural since it has the same CW minimum and maximum of 15 and 1023 respectively, as AC1 and neither have a TXOP. However the AIFSN of AC1 is 9, whilst the AIFSN of AC2 is 6. The maximum latency is at a similar peak value of 2 x $10^4\mu$s, but the throughput decreases from a higher value of approximately 2.25 bytes/μs. This makes sense from theory, since a lower AIFSN means that the AIFS time is lower and so less time is spent waiting before a transmission occurs. It may seem strange then that the packets lost reach peak levels of 1800, however, on further reflection, the shorter AIFS time means that packets are quicker to retry accessing the channel and so hit the retry limit faster. This causes a higher number of packets to be lost as the channel busy probability increases.

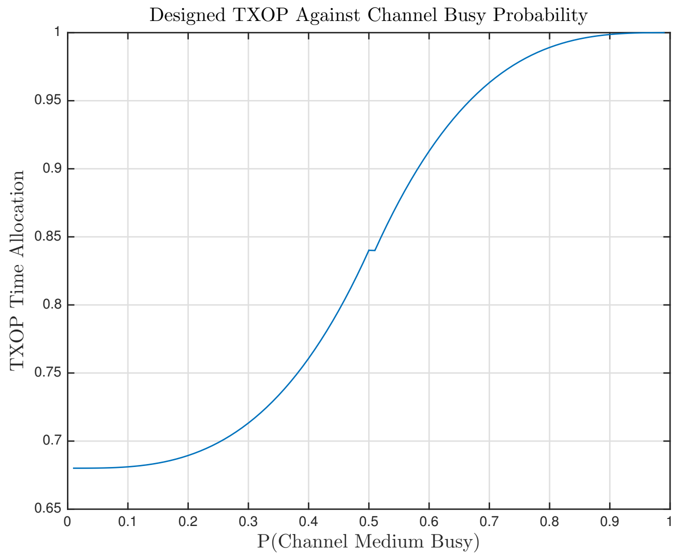


Figure 26: Plot of the Designed TXOP Time Allocation Against Channel Busy Probability

Fig. 26 shows the designed TXOP time allocation against channel busy probability. This shape was chosen in order to have a low probability of TXOP occurring at low probabilities of the channel being busy, since when the channel is mostly idle there is less need for prioritisation. At high levels of channel busy probability, there are greater chances of a TXOP occurring, since it is now more important that packets with a higher priority can have faster access to the channel.

Fig. 27 plots the mean throughput against channel busy probability for the 3rd AC when the data rate is at 5 bytes/μs. The black dashed line shows the maximum data rate and throughput which is constant at the given value of 5 bytes/μs. It may seem unusual that the experimental throughput manages to remain so constant at approximately 4.5 bytes/μs, despite variations in the channel busy probability. This is where the designed TXOP allocation graph shape in Fig. 26 really comes into play, as it allocates TXOP transmission time according to how busy the channel is.

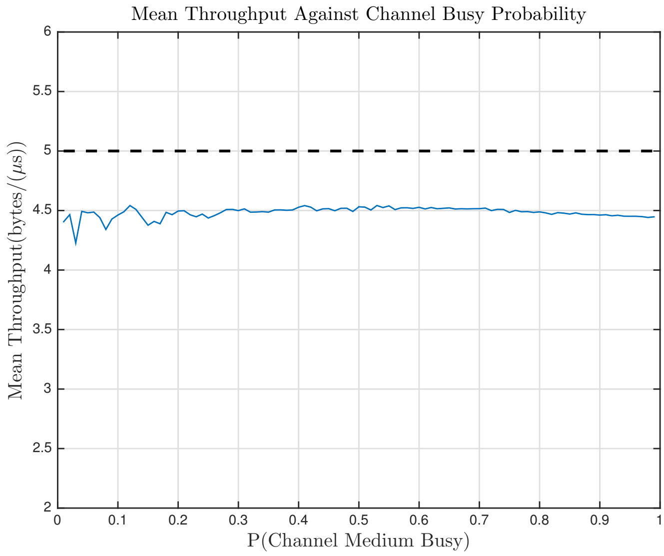


Figure 27: Plot of the Mean Throughput Against Channel Busy Probability for the 3rd Access Category using a Data Rate of 5 bytes/μs

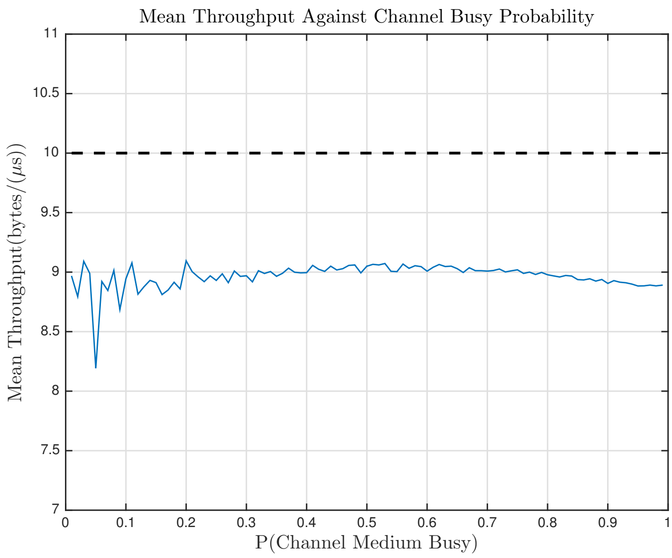


Figure 28: Plot of the Mean Throughput Against Channel Busy Probability for the 3rd Access Category using a Data Rate of 10 bytes/μs

Fig. 28 plots the mean throughput against the channel busy probability for the third

AC but with the maximum data rate and thus throughput placed at 10 bytes/μs. Now it can be seen that the experimental values are always bounded by the data rate, here at approximately 9 bytes/μs, which reinforces the validity of the model and is to be expected.

The graph for the 4th AC is given in Appendix F for a data rate of 5 bytes/μs. It is almost identical to graph in Fig. 27, even though there are variations in their maximum and minimum CW sizes and their respective AIFSN values.

In all graph cases there does seem to be an optimum, albeit gentle, value of 0.5 for the channel busy probability, at which point the curves do seem to have a slightly higher throughput. This is where they are closest to the theoretical boundary value of the maximum data rate. They also all have greater variations at their start when channel busy probabilities are low. At these values the throughput is much more reliant on how much data is actually being transmitted, which is variable, rather than the impact that the TXOP time allocation has on the throughput value of the AC, which keeps a more consistent value.

13 Model 4: An Improved Version of the EDCA Mechanism Used in the HCF

13.1 Implementation

Model 4 implements an improved version of the EDCA model described in Model 3, for three different levels of channel busy probability. Similarly to Model 2, these levels were chosen to be 0.25, 0.5 and 0.75 respectively with length 1000 and the model now takes into account only decrementing the slot time when the channel is free. This again uses a shuffle function in MATLAB to create the channel pattern of ones and zeros for any given probability level.

The ACs can again be referred back to Table 4 to see the minimum and maximum contention window sizes for each AC, as well as the AIFSN and maximum TXOP lengths for them. Since ACs 1 and 2 do not have any TXOP time, their model was developed separately to the models of ACs 3 and 4. Thus the written code for this was very similar to that of model 2 and can be found in full in Appendix G. The code will not be explained again, but the graphed results will be analysed in the results section, since AC1 and AC2 do have different AIFS times.

AC3 and AC4 for Video and Voice packets were then looked at in more detail using two codes, which can be found in full in Appendices H and I. Since using a real channel model allows for a real timing analysis in microseconds, instead of using the TXOP time allocation graph as in Model 3, it can now be decided in terms of slot time when the TXOP occurs. This is shown in the first code snippet below. The first code, in Appendix H, results in a graph showing the total number of packets sent against each repetition of the loop. The second, in Appendix I, runs the channel for busy probabilities ranging from 0.01 to 0.99 in 0.01 increments and shows the extent to which using TXOP improves transmission as a function of how busy the channel is. Important code snippets that are related to managing the TXOP aspect in ACs 3 and 4 have been explained below.

```
TXOPOccurence(125) = 1; TXOPOccurence(375) = 1;
TXOPOccurence(625) = 1; TXOPOccurence(875) = 1;

assert(length(TXOPOccurence) == size(t,2));
```

This line controls whether or not the TXOP is enabled, that is to say that commenting out the line of code can disable it. It also specifies the slot times at which a TXOP should occur. The second line simply makes sure that the TXOP vector and the timing vector are the same length.

```
TXOPCounter = 0;
TXOPActive = 0;
TimeLastPacketSent = 0;
numPacketsSent = 0;
```

These four variables can each be explained very simply. The 'TXOPCounter' tracks how long the TXOP has lasted for so far in each loop. 'TXOPActive' flags whether or not the TXOP period is active. 'TimeLastPacketSent' is the time at which the last packet was sent. 'numPacketsSent' is the total number of packets sent for a certain repetition.

```
if ((i-TimeLastPacketSent))>=5
    numPacketsSent = numPacketsSent + 1;
    TimeLastPacketSent = i;
end
```

This condition states that since the packets arrive at a rate of 1 every 100μs, a packet can only be transmitted if it has arrived fully. When this occurs, the number of packets sent is incremented and the time at which the last packet was sent is updated.

```
elseif (TXOPFlag == 1 || TXOPActive == 1)
    TXOPActive = 1;
    TXOPCounter = TXOPCounter + 1;
    numPacketsSent = numPacketsSent + 0.2;
```

```
    if TXOPCounter == MaxTXOPSlot(ac)
        TXOPActive = 0;
        TXOPCounter = 0;
    end
end
```

This excerpt demonstrates what takes places when the TXOP period is active. Firstly the 'TXOPActive' flag is set to 1 and then the 'TXOPCounter' counts the number of slot times during the TXOP. The number of packets sent during each TXOP period is 0.2, since Slot time/Arrival rate = 20μs/100μs = 0.2. This is the value that is used to increment 'numPacketsSent'. If the time limit of the TXOP has been reached, then the counter is reset, and the 'TXOPActive' flag is set to 0, since the TXOP is no longer occurring.

```
timeVector = [ones(1,round(percBusy(z)*N)),zeros(1,round((1-percBusy(z))*N))];
```

This line of code is used in the last version of this model, found in Appendix I, when the proportion of channel busyness is varied. It is used to generate a model of the channel of a specific proportion of busyness that is defined in the vector 'percBusy'.

13.2 Testing

There are several assumptions made in this model and these are detailed as follows:

- The slot time was set to be 20μs. This was chosen from research to be a reasonable value.
- A constant packet size of 300 bytes was assumed. Again this was chosen from research and kept constant to keep the model at a reasonable level of complexity.
- The packet arrival rate was set to be 3MB/s, which means that a 1 complete packet arrives every 100μs. [44]
- A channel link size of 1 was chosen, which means that each AC is fighting with other sources for access of 1 channel.

- It was decided not to overly complicate the model by incorporating the Request-To-Send/Clear-To-Send mechanism.
- The retry limit was set to 10, so that there was a finite number of reattempts.
- A trial repetition number of 1000 was used.
- The CW minimum and maximum values, the AIFSN values and the TXOP values for all ACs were kept as in Table 4.
- The SIFS time was set to be 10μs as research found this to be a reasonable value and the maximum data rate was set to 5 bytes/μs.
- The AIFS values were calculated as: AIFS = (Slot time x AIFSN) + SIFS
- The first value of each channel model was set to 1, to assume that the packet saw the channel as busy on arrival and was not sent through immediately, since the channel is not refreshed on each run of the loop. Importantly, this likely leads to a slight overestimation of the latencies at the start of the procedure.
- Each data value of the channel has been chosen to last the length of one slot time, for a realistic scenario whilst still keeping the model at a reasonable level of complexity.
- In the third version, in which the benefit of TXOP is investigated, the probability of the channel being busy was varied from 0.01 to 0.99, using increment values of 0.01.

13.3 Results

Fig. 29 shows a plot of the latency against packet number for the 1st AC. The graph plots use black for a channel busy probability of 0.25, blue for 0.5 and red for 0.75.

It can be seen in the figure that the black line trend is generally lower than that of blue and the red line is the highest. This is to be expected and additionally as the

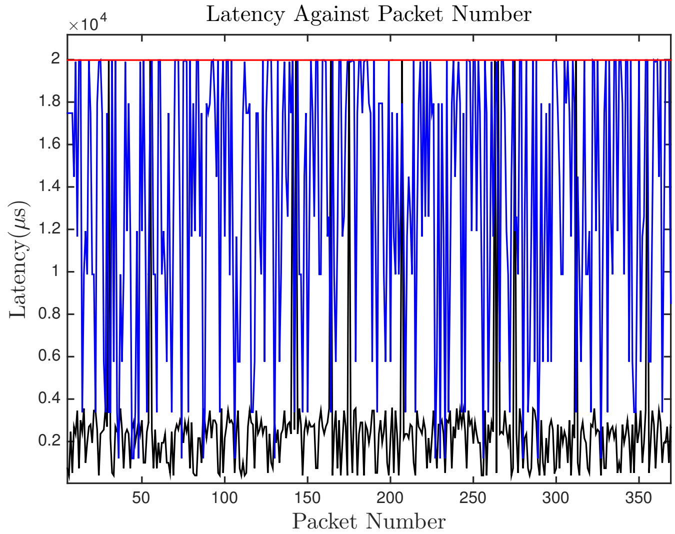


Figure 29: Plot of Latency Against Packet Number for the 1st Access Category

probability increases the variation also increases along with the average latency which is approximately 2000μs for 0.25 and 14000μs for 0.5, at which point the latencies start to hit the limit of 2x$10^4\mu$s. Interestingly the red line is a constant value at the maximum latency of 2x$10^4\mu$s. This is because in this case the channel busy probability is high at 0.75 and the AIFS time of AC1 is 100μs. This means the AC must have $100\mu s/5\mu s = 5$ free slot times before it can transmit data, which never occurs at a busyness level of 0.75.

Fig. 30 is a plot of the cumulative latency against packet number for the 1st AC. This graph is very similar to that of model 2 with the black data line having lower cumulative latency values than the highest data line, which is red. The red plot reaches a cumulative latency of 2x$10^7\mu$s, followed by blue which reaches approximately 1.3x$10^7\mu$s and finally black is much lower at just over 0.2x$10^7\mu$s. This is natural, as previously discussed and all the trends appear to be fairly linear.

The graph trends for AC2 are very similar to this and are given in Appendix J. AC2 has a lower AIFS time of 60μs and so the red data trend does often not hit the maximum latency limit. Interestingly, but unsurprisingly, all of the average latencies are far

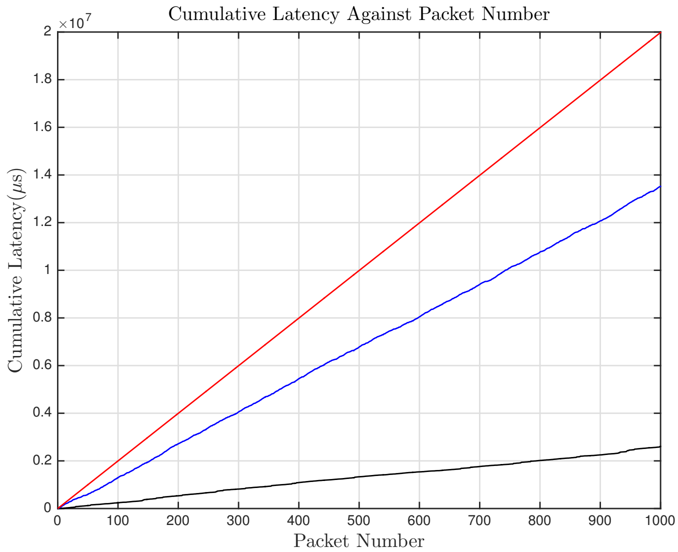


Figure 30: Plot of Cumulative Latency Against Packet Number for the 1st Access Category

lower. The black data averages approximately 1000μs, the blue data averages 2000μs and finally the highest probability of red, averages 12000μs. The red and blue trends do still occasionally hit the latency limit but far less frequently that for AC1. The same reduction is thus reflected in the cumulative latency graph, where the red data line only reaches 1.2x10$^7\mu$s followed by blue at 0.3x10$^7\mu$s and then the black line which reaches just under 0.2x10$^7\mu$s. Since the difference between the two black lines of AC1 and AC2 is so small but the difference between the two red lines is much larger, we can deduce that the less busy the channel is, the less of an impact the AIFS time has on latency.

We can now analyse both ACs 3 and 4, which are higher priority data packets and do have transmission opportunities. For these two graphs, the blue data is a channel busy probability of 0.25, red is 0.5 and yellow is 0.75.

Fig. 31 looks at TXOPs in general and plots the number of packets transmitted for every trial number for the 3 levels of channel busy probability when TXOP is disabled.

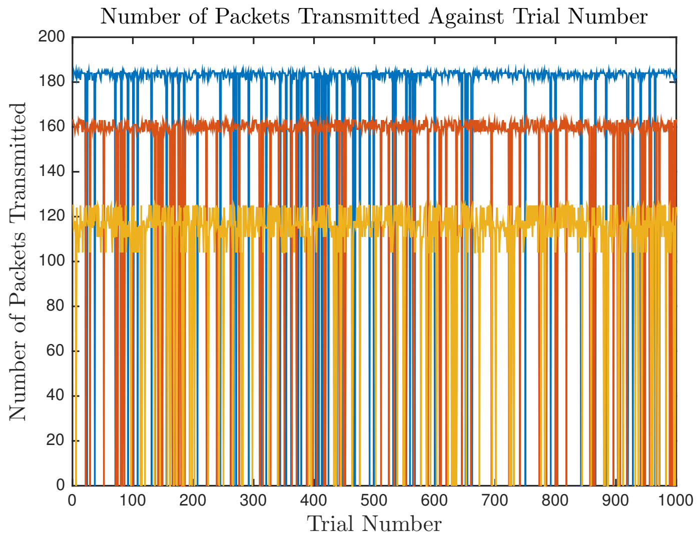


Figure 31: Plot of the Number of Packets Transmitted Against Trial Number for 3 Channel Busy Probabilities when TXOP is Disabled

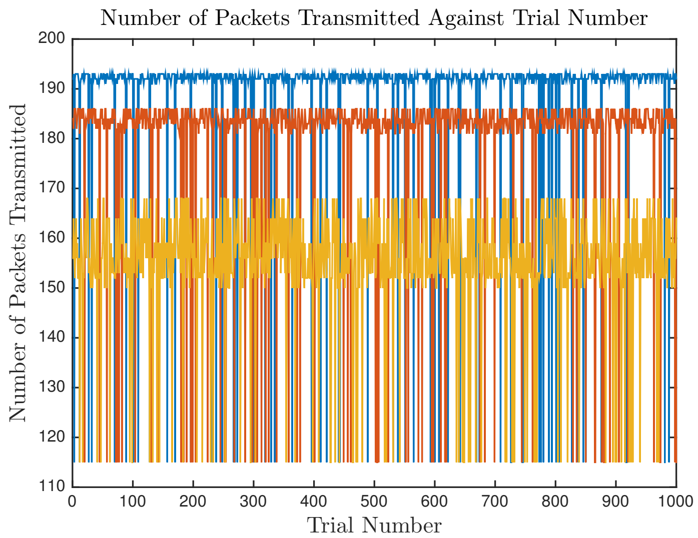


Figure 32: Plot of the Number of Packets Transmitted Against Trial Number for 3 Channel Busy Probabilities when TXOP is Enabled

Looking at the number of packets transmitted on average, it can be seen that for the highest busyness level of 0.75, approximately 120 packets go through, for 0.5 we have 160 packets in general and finally at 0.25 approximately 180 packets are transmitted.

Downwards spikes occur when no packets are transmitted and this is most likely to occur when the retry limit is reached and the packet is dropped.

These values can now be compared to Fig. 32 which is the same plot, but this time with the TXOP periods enabled. Now looking again at the average numbers of packets that are transmitted, for the highest busyness level of 0.75, approximately 160 packets go through, for 0.5 there is an increase to 185 packets and at 0.25 approximately 190 packets are now transmitted. This is a significant increase for each level, but what is more fascinating is that at 0.75 the increase in the number of packets that can be transmitted is 40, at 0.5 it is 25 and at 0.25 it is only 10. This strongly implies that having active TXOP periods is more effective at higher levels of channel busyness. This is further investigated in the upcoming graphs, when the channel busyness is varied to further study the effect of having TXOPs.

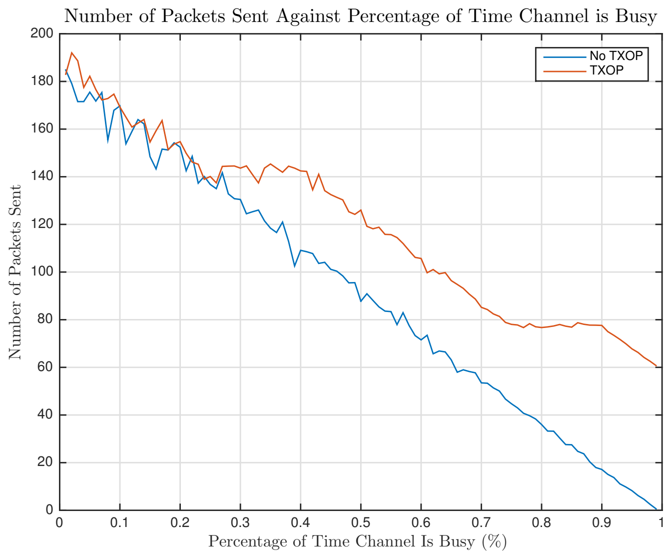


Figure 33: Plot of the Number of Packets Sent Against the Percentage of Time the Channel is Busy for the 3rd Access Category

Fig. 33 plots the number of packets that are transmitted against the percentage of time for which the channel is set to be busy for the 3rd Access Category. The red line indi-

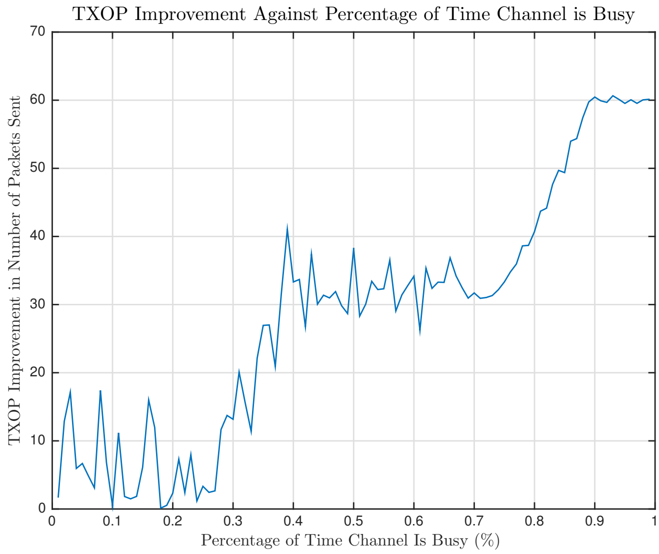


Figure 34: Plot of the TXOP Improvement Against the Percentage of Time the Channel is Busy for the 3rd Access Category

cates the results when TXOPs are enabled, whilst the blue line shows the results when TXOP is disabled.

It is evident that the red line is mostly higher than the blue. Up till approximately a channel busy proportion of 25% the red and blue lines are quite close together, then there is a much larger distance between them up until around 75%. In this second region when TXOP is enabled, approximately 40 more packets can be sent. Finally the gap between the lines is largest beyond 75%, at which point around 60 extra packets can be sent by the AC when TXOP is enabled. This confirms the implication made in the previous section, stating that TXOP being enabled is more useful at higher levels of channel busyness.

Fig. 34 then shows the improvement allowed by TXOP against the percentage of the time for which the channel is busy for the 3rd Access Category. This is essentially the red line minus the blue line and thus shows how much of an impact enabling TXOP actually has. The three regions that were described can now be seen very clearly. There are quite large variations in the data points up until 70%, this is the point at which the TXOP advantage really comes into play because the channel is so busy. At lower levels, results are more dependent on how much data the channel is actually transmitting.

The two graphs are now repeated for AC4. It is important to note that AC3 has a minimum and maximum contention window of 7 and 15 respectively, an AIFS of $40\mu s$ and a maximum transmission slot of $150 = 3ms/20\mu s$. In contrast, AC4 has a minimum and maximum contention window of 3 and 7 respectively, an AIFS of $40\mu s$ and a maximum transmission slot of $75 = 1.5ms/20\mu s$. For this reason, when modelled, the TXOP for AC3 was set to occur twice, wherease the TXOP for AC4 was set to run 4 times. This was so that the TXOP occurred for a total of 6ms in both cases.

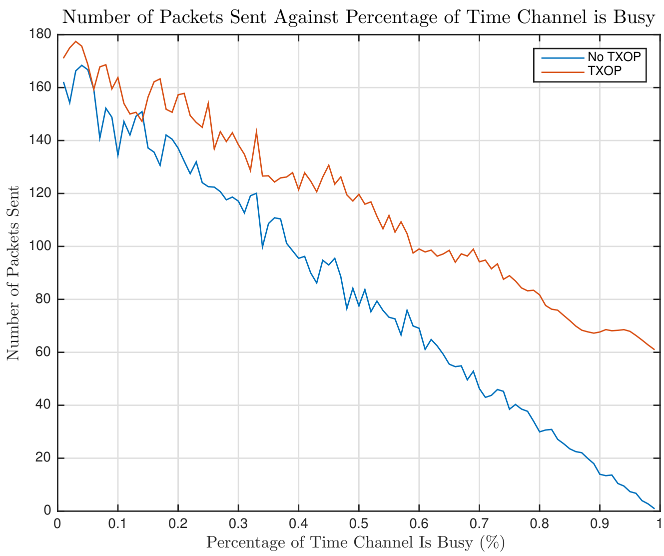


Figure 35: Plot of the Number of Packets Sent Against the Percentage of Time the Channel is Busy for the 4th Access Category

Now in Fig. 35, the TXOP shows a gradual but steady increase in the amount of benefit it provides, compared to the case of AC3, which has 3 distinct regions. It still ends up

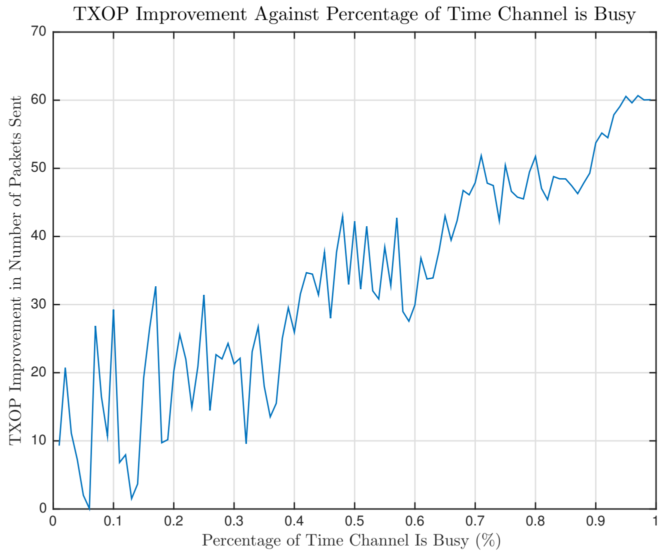


Figure 36: Plot of the TXOP Improvement Against the Percentage of Time the Channel is Busy for the 4th Access Category

with a difference of approximately 60 extra packets at the maximum channel busyness, but Fig. 36 shows a much steadier, almost linear increase in the TXOP benefit, ignoring data fluctuations. This still confirms that TXOP is more useful at higher levels of channel busyness and the difference in the graph shape from AC3, is most likely due to the difference in the timing distributions of the TXOPs for AC3 and AC4 when the model simulation is run.

14 Model 5: A Dynamic Time Boundary for the CCH and SCH Subintervals

14.1 Implementation

Model 5 is a resource allocation control mechanism that implements a dynamic time boundary for the CCH and SCH subintervals. It assesses a possible improvement to the MAC sub-layer of the 802.11p protocol. The model adaptively adjusts the time boundary of the CCH and SCH channels based on the previous time interval. The results of doing this are compared to the case of a static time boundary in terms of the number of packets that would be dropped from each channel.

The model of the channel is split into two components, the traffic on the CCH and the SCH channels respectively. These are initially modelled to follow a specific pattern of traffic, the graph plot for which is shown in the results section. The model chooses the boundary primarily based on the safety channel statistics and then chooses the service channel subinterval length after this. This is done by creating an 'alpha' variable, set to 0.6 that gives a slightly higher weighted priority to those messages on the control channel. The model runs over 22 seconds on the time axis. The full code for this model can be found in Appendix K, however certain relevant snippets have been explained below.

```
y = linspace(50,150,20);
inputStream1=[50*ones(1,40), y, 150*ones(1,140-60),flip(y),50*ones(1,40),y];
inputStream2 = -inputStream1+150;
inputStream = [inputStream1',inputStream2']';
```

Here the input stream vector is defined for the CCH and the SCH. The top row of the vector is set to the CCH values and the lower row is set to the SCH values. The graph for this can be found in the results section.

```
maxPacketsFixed = 50/msPerPacket;
refPackets = (50/msPerPacket)*ones(2,N);
streamDropped = (inputStream - refPackets).';
streamDropped(streamDropped < 0) = 0;
```

In the first two lines the maximum number of packets that can be transmitted on either channel is calculated and then a vector is created for this. In the second two lines, the formula for the number of packets dropped is set and a minimum bound is placed at zero so that a negative value does not appear when no packets are dropped.

```
ControlDropped(i)  = max(currentPacketsControl-maxPossibleControl,0);
ServiceDropped(i)  = max(currentPacketsService-maxPossibleService,0);
```

This formula simply calculates the number of packets dropped as the number of packets that arrive minus the maximum possible number of messages that can be transmitted. This is calculated for both the CCH and the SCH.

```
scaledPacketsControl = alpha*currentPacketsControl;
scaledPacketsService = (1-alpha)*currentPacketsService;
```

This excerpt weights the total number of packets that are being transmitted on each channel according to the given priority level. In this case, as mentioned, 'alpha' was set to 0.6, giving a higher weight to the control channel and the safety related messages that it carries.

```
CCH(i+1) = (scaledPacketsControl/(scaledPacketsControl+scaledPacketsService))
*100;
    if CCH(i+1) > maxBound
        CCH(i+1) = maxBound;
    elseif CCH(i+1)<minBound
        CCH(i+1) = minBound;
    end
    SCH(i+1) = 100-CCH(i+1);
```

Finally, here the CCH subinterval is updated based on the ratio of the previous 100ms time interval. The minimum and maximum bound conditions of 20ms and 80ms respectively are also implemented to help control the system and ensure that neither channel is ever neglected entirely. After this has been done for the CCH, the SCH sub interval is updated accordingly.

14.2 Testing

There are several assumptions made in this model and these are detailed as follows:

- The maximum size of each packet was set as a constant value of 100 bytes. Data packets can vary in length and this was chosen as a reasonable size to use for the purposes of a sensible model of the system.
- Each packet of data was estimated to take a maximum of 1ms to be transmitted. This means that the data rate is 100 kB/s which is a conservative value compared to the 1MB/s rate used in the earlier models.
- The length of the input stream was set to be 220 intervals, where each interval slot of the channel transmission was 100ms.
- Every 100ms channel synchronisation occurs, so every 100ms all cars synchronise to the safety control channel.
- There was negligible switching time between the CCH and the SCH to account for. This is not strictly true, as in reality there is a very short guard interval of 4ms before each CCH or SCH interval.
- The minimum length of time for transmission regardless of how few packets are being transmitted on the channel, was set to 20ms. This therefore gave a maximum bound of 100ms - 20ms = 80ms.
- The weighting proportion 'alpha' was set to 0.6 to place more emphasis on the packets coming from the CCH. This was seen to be a reasonable level to show the importance of the safety messages, without completely biasing the system.

14.3 Results

Several different trends were tested for the traffic levels on the SCH and CCH channels including random, linear and exponentially decreasing. These lines of the MATLAB code have been commented out in Appendix K. However, none of these seemed to be a realistic scenario of what may occur over time.

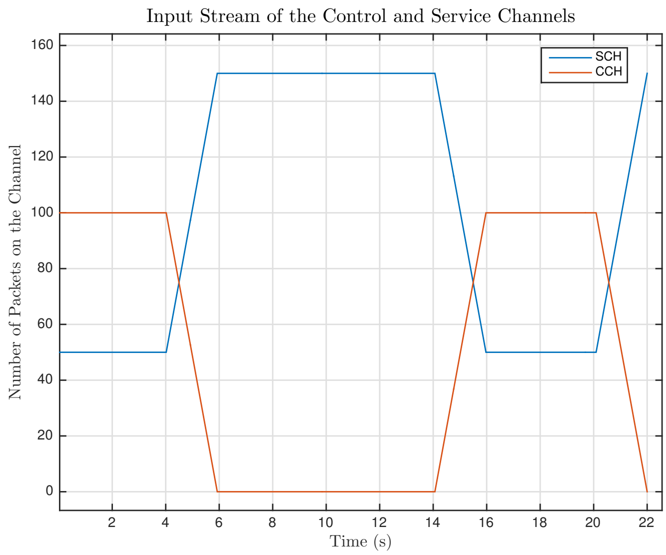


Figure 37: Plot of the Traffic Input Stream Modelled for the CCH and SCH

Fig. 37 shows the traffic input stream that was decided on as this fulfils the fact that generally more packets need to be transmitted on the SCH than on the CCH, due to revenues generated from media services. It also shows high and low periods of traffic on the CCH compared to the SCH and transition stages between the high and low traffic periods. The stable levels are given at 0, 50, 100 and 150 packets attempting transmission, with 2s allotted for each transition period.

Fig. 38 shows the number of packets that would be dropped from the CCH and SCH channel if there was static time boundary. That is to say, if a fixed subinterval of 50ms was allocated to the CCH followed by a fixed subinterval of 50ms for the SCH for every 100ms. It is clear that far more SCH packets are dropped than those on the CCH, which is to be expected since there is more traffic on the SCH in the model design. In the stable periods, only 50 of the packets can actually be transmitted, which is not ideal or particularly efficient.

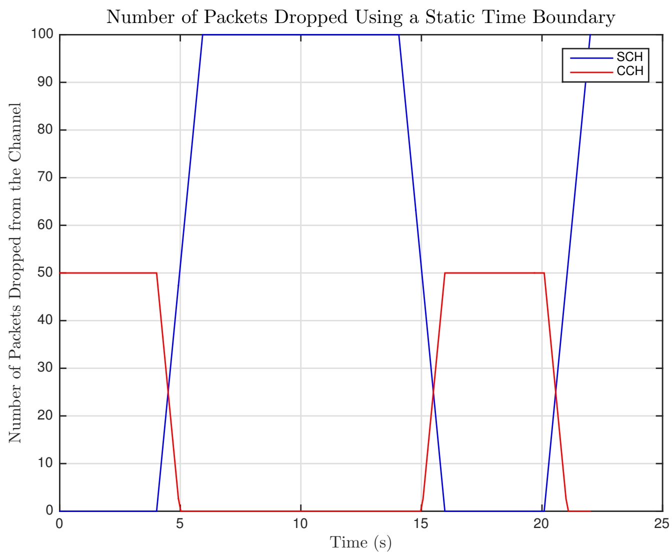


Figure 38: Plot of the Number of Dropped Packets from the CCH and SCH Using a Static Time Boundary

Fig. 39 then compares the previous graph to the packet dropped results when the dynamic boundary is used. The dynamic boundary, as mentioned uses the ratio of channel traffic from the previous 100ms time interval and then adjusts the boundary position, with the CCH: SCH packet priorities weighted as 0.6:0.4. The dashed lines show the packets that would be dropped when using the dynamic boundary technique. In the high traffic portion of the SCH, approximately 30 more SCH packets can now be transmitted, which is a 30% improvement from the static case. However, the region where there is relatively low SCH traffic and but high CCH traffic, forms a trade-off between the SCH and CCH channels. The number of dropped packets on the CCH does indeed drop by approximately 8 packets, however the SCH values increase by a similar value.

Thus to assess the overall impact of using the dynamic time boundary, Fig. 40 was

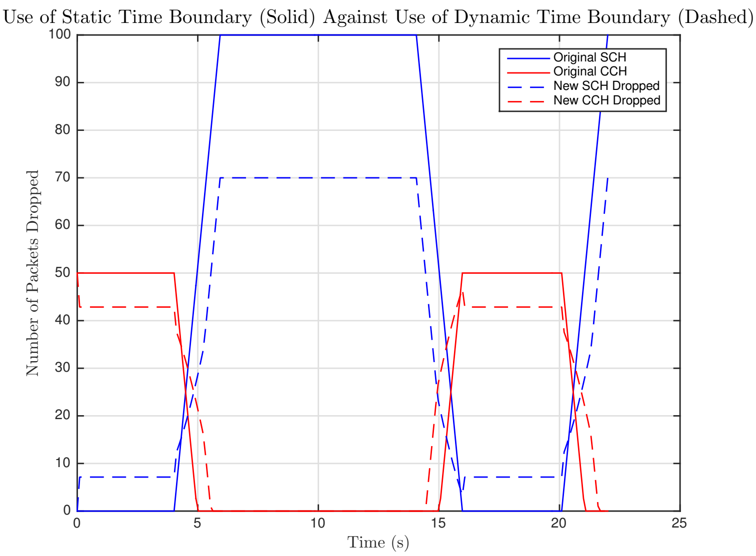


Figure 39: Plot Comparing the Number of Dropped Packets from the CCH and SCH channels when Using a Static Time Boundary, Compared to a Dynamic Time Boundary

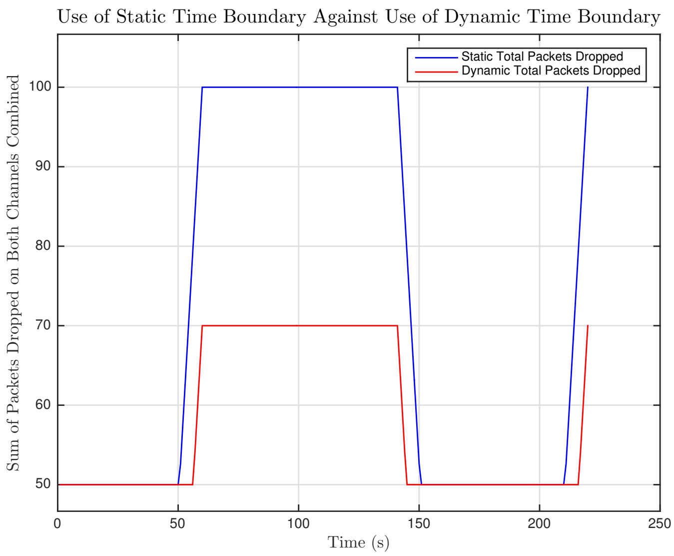


Figure 40: Plot of the Sum of Packets Dropped on the CCH and SCH Channels when Using a Static Time Boundary, Compared to a Dynamic Time Boundary

plotted, showing the total number of packets dropped on both the CCH and SCH combined for both the static and dynamic boundary cases. Now it is evident that although there are the lower traffic portions of the plot where there is no improvement from using the dynamic boundary due to the trade-off seen in the previous graph plot, there is a significant benefit at times when there is a large imbalance in the amount of traffic being transmitted on the two channels. Importantly, it was also found that the choice of 'alpha' was crucial, if it was increased from 0.6 the performance of the dynamic boundary worsened, but if it was lower at for example 0.4, then the overall performance remained the same, but a far greater trade-off occurred in the region where the CCH has high traffic. This is shown below in Fig. 41.

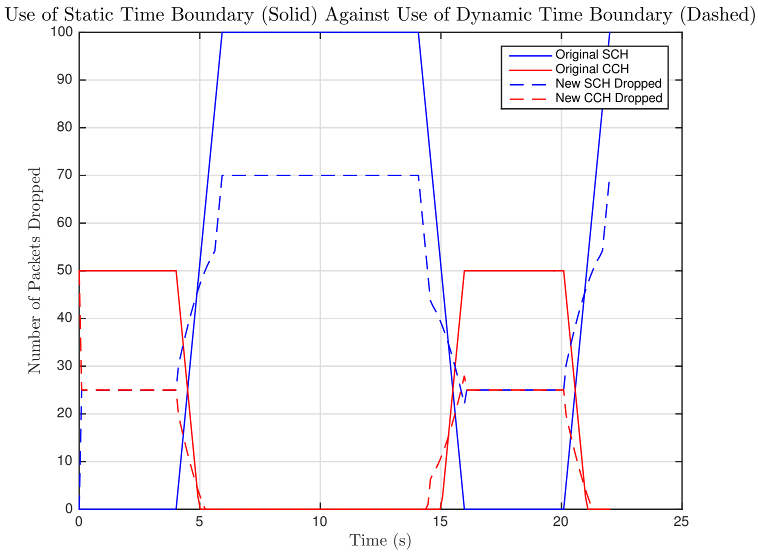


Figure 41: Plot Comparing the Number of Dropped Packets from the CCH and SCH channels when Using a Static Time Boundary, Compared to a Dynamic Time Boundary

All of this evidence therefore suggests that adjusting the time boundary dynamically has far greater benefits when there is a large enough difference between the traffic amounts on the two channels, than if the traffic levels are balanced. This is logical, since the more balanced the channel traffic levels are, the more optimum a 50ms:50ms channel time split becomes and the less impact dynamic adjustment would have.

15 Evaluation

15.1 Evaluation of the Models

By collating all of the discoveries made throughout the modelling process, we can attempt to evaluate the system as a whole and gather together experimental conclusions.

The log of the latency was found to increase almost linearly as the probability of the channel being busy increased. This was because the busier the channel, the longer the time packets spend reattempting access to the channel. Both models 1 and 2 confirmed this theory. Since a constant packet size was assumed, the throughput was seen to be inversely proportional to the latency. There was also a steep, exponential increase in the number of packets lost as the channel busyness increased from 0.1 to 0.99.

Over a period of 1000 trials, the less busy the channel was, the lower the cumulative latency was found to be. Significant decreases in these cumulative latencies also occurred when dropped packets were discarded from the data set. Thus quantifying how much of an impact the dropped packets had on the numerical data, without affecting the overall trend.

When studying the 1st Access Category, the lack of a TXOP period meant that the latency, cumulative latency, throughput and packet loss graphs followed the same trends, as in models 1 and 2. The latency also had lower variations in data as the channel busy probability increased. The 2nd Access Category however, despite following the same trends, was greatly impacted by its shorter AIFS time. The short AIFS time meant that packets were quicker to reattempt channel access, quickly hit the retry limit and were then dropped, leading to a greater number of packet losses. It was also discovered that the less busy the channel was, the less of an impact the AIFS time had on latency.

For the 3rd and 4th Access Categories, the experimental throughput values were found to be bounded by the maximum throughput that was calculated using the data rate. This occurred when the TXOP time allocation was allotted such that at high levels of channel busy probability there was a greater chance of a TXOP occurring. This meant that higher priority packets would get quicker access to the channel. A possible optimum value of 0.5 was found for the channel busy probability to achieve a maximum throughput, although the slope of the curve was incredibly gentle.

It was then found that having active TXOP periods was more effective at higher levels of channel busyness. This was confirmed to be the case for the 3rd and 4th Access Categories. Variation in the data up to a channel busyness proportion of 0.7 were most likely caused by results being more dependent in early stages, on how much the channel is actually transmitting. At high levels of busyness, enabling TXOP provides a much larger advantage.

The traffic allocation control mechanism for the CCH and SCH showed a 30% improvement using a dynamic time boundary compared to a static time boundary during periods of a far higher proportion of SCH traffic compared to CCH traffic. There was less of an improvement using dynamic adjustment at balanced traffic levels, when a static 50ms:50ms time split would naturally work well. This occurred when the priority weighting was set to be 60% based on the traffic on the CCH, as safety messages were deemed more urgent. All in all, the improvement to the MAC sub-layer was shown to be feasible in helping to improve efficiency and adjust time slot allocation between the Control Channel and the Service Channel.

15.2 Limitations of the Models

There are several limitations associated with the models that have been developed and which could also be seen as drawbacks of the project. It is vital that these are acknowledged so that the analysis carried out is viewed from a realistic and critical perspective. Factors that the models fail to acknowledge are detailed below:

- The model does not take into account the Acknowledgement packets needing to be received by the sender. This is not necessary when group messages are sent, but otherwise would be a part of the system. This could have an impact on latency accuracy, as waiting for an Acknowledgement from the receiver takes time.

- The Request-To-Send/Clear-To-Send protocol was neglected. This is a protocol where the transmitter effectively checks with the receiver whether it is ready for a packet to be transmitted. It is not necessarily implemented in all VANET systems, since it slows down the whole transmission procedure. However if by chance it were chosen to be included, then latency measurements in the models would be an underestimate.

- The research does not take into account practical issues that may occur in reality when implementing a VANET system. These could include issues such as the changing distance between two cars or whether they are able to synchronise with the global satellite when travelling through a tunnel. Time delays may also occur in reality that may be due to problems with a steady connection being established with other cars. This may also include the 'Hidden Terminal' problem where two mobile nodes may not be able to communicate if one is unable to be accessed. [45]

- The model could be approached from a different angle, looking at a finite number of sources, perhaps ranging from 2 upwards, and carrying out an exact analysis using real patterns of communication data.

- The parameters used in the models may be location dependant or scenario dependant, for example the volume of traffic on the CCH or SCH channels. This may impact the data and results, however the extent of this impact is unknown.

- Incorporating the 4ms guard interval when switching between the CCH and SCH should not affect the benefits of the dynamic time boundary, but may have a slight time delay impact in terms of how many packets can be transmitted overall. Then again, in reality packet sizes would vary and this is a parameter that could be further explored with respect to latency and throughput.

- Finally, it is difficult to compare the results of the models to graphical results shown in literature. This is because the majority of papers are either theoretical or use the NS-2 software to simulate networks using a more on-the-road perspective, analysing factors such as vehicular node distance or density.

15.3 Comparison To Aims and Objectives

Evaluating the project from an overview perspective and reflecting on the aims and objectives that were initially laid out in Section 7.2, the following tasks have been achieved:

- The field of VANETs has been studied to a reasonable level of detail and understanding.
- The unique features of VANETs compared to MANETs were investigated from the perspective of the MAC protocol.
- The operation of the IEEE 802.11p MAC protocol was understood as well as the main parameters that can impact its efficiency.
- Techniques that could help improve the MAC protocol for use in VANET systems were summarised.
- Four methods to model the IEEE 802.11p MAC protocol were successfully implemented in MATLAB. These included modelling the DCF CSMA/CA protocol and also the EDCA contention-based channel access scheme to account for different priority levels of data.
- The models were studied with respect to latency, throughput and packet loss probabilities, using a variety of perspectives.
- The MAC sub-layer protocol improvement modelled and analysed the impact of dynamically adjusting the timing of the CCH and SCH intervals based on traffic

load. This was compared to the current method of using fixed time intervals for each channel, to evaluate its success.

- A far deeper awareness and understanding of the system was developed.
- Progress was regularly evaluated with regards to the Gantt chart and the project supervisor was met at regular intervals throughout the year.

Overall all of the key objectives of the project that were developed have been fulfilled. This is aside from the refinements made to the project aim, the details for which can be found in Section 7.3.1.

16 Further Work

There are several possible areas that can build on and further explore the work carried out in this report. There are also interesting alternative routes and tangents that the work could have taken that were not chosen as the main topic of this report. The most significant possible expansion points are detailed below:

- The NS-2 network simulation software, that is widely accepted, can be used to visually simulate routing or control and multicast protocols, such as the DSRC/WAVE protocols for wireless networks. It can also demonstrate the MAC routing protocols in practise for the IEEE 802.11p layer. It is commonly used by academics in the field of network research and the simulation environment is developed via TCL scripts. [3][11]
- This can be extended to the use of VISSIM, which is a traffic network simulation software that can profile road traffic and network communication traffic. [37]
- Throughout this work, it was assumed that the error rate for packet data transmission was negligible and that if a packet was delivered it was uncorrupted. However, this is not always the case. The model can be increased in complexity to take a varying bit error rate into account and thus have an increased number of retransmissions. A low packet error rate is necessary from end to end in the network and thus improving protocol reliability and decreasing the number of packet retransmissions is vital to increasing network efficiency.
- There are also plenty of tests that can be carried out in differing scenarios, when the environment in which the vehicle is transmitting is changed. A few of these are listed below:
  - An increased or decreased distance between the vehicles.
  - Different positioning of the vehicles with respect to each other.
  - Altering network parameters.

- Accounting for unpredictable driver behaviour.
- Experimenting with the maximum number of cars that can successfully transmit and receive messages on a set distance of road.
- Carrying out real life testing and data collection using vehicles on a testing track.

17 Conclusion

Bringing together all the work that has been carried out over the course of the year, this project began with an overview of VANET systems. Their possible impacts on society were studied and their unique features when compared to MANET systems were investigated. The project aim was then refined to focus on the MAC sub-layer of the 802.11p protocol, which is used as the standard for inter-vehicular communication. The Distributed Coordination Function and Enhanced Distributed Channel Access mechanisms were analysed and explained in detail and a potential improvement for resource allocation on the Control and Service Channels was developed.

The analysis of designing VANET systems was then discussed and five key models were chosen to form the main investigative part of the research study. The first four models implemented the DCF and EDCA protocols using two different methods, the EDCA modelling being a more complex build on the DCF model, incorporating user packet priority into the system. These were successfully developed using MATLAB and studied against the performance criteria of latency, throughput and packet loss. The fifth model implemented a dynamic time boundary control mechanism for traffic allocation on the CCH and SCH, concluding that it is indeed feasible to improve the system efficiency by using a dynamic, rather than static time boundary.

User Guide

The MATLAB software needs to be installed in order to run the codes found in Appendices B to K of this project. MATLAB version R2015a was used in this case and this is easily downloaded using the following link:

http://uk.mathworks.com/products/new_products/release2015a.html?s_tid=gn_loc_drop
The MATLAB scripts themselves are simple to run once the software has been installed.

Bibliography

[1] Yuan Yao, Lei Rao, and Xue Liu. Performance and reliability analysis of ieee 802.11p safety communication in a highway environment. *IEEE Transactions on Vehicular Technology*, 62(9):4198–212, 11 2013.

[2] H. Hartenstein and K. P. Laberteaux. A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, 46(6):164–71, 06 2008.

[3] E. Spaho, L. Barolli, G. Mino, F. Xhafa, and V. Kolici. Vanet simulators: A survey on mobility and routing protocols. In *2011 International Conference on Broadband, Wireless Computing, Communication and Applications*, pages 1–10, Los Alamitos, CA, USA, 26-28 Oct. 2011 2011. Grad. Sch. of Eng., Fukuoka Inst. of Technol. (FIT), Fukuoka, Japan, IEEE Computer Society. T3: 2011 International Conference on Broadband, Wireless Computing, Communication and Applications;.

[4] Yi Qian, Kejie Lu, and N. Moayeri. A secure vanet mac protocol for dsrc applications. In *2008 IEEE Global Telecommunications Conference*, page 5 pp., Piscataway, NJ, USA, 30 Nov.-4 Dec. 2008 2008. Nat. Inst. of Stand. Technol., Gaithersburg, MD, United States, IEEE. T3: 2008 IEEE Global Telecommunications Conference;.

[5] Xi Sun and Li Xia-miao. Study of the feasibility of vanet and its routing protocols. In *2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, October 12, 2008 - October 14*, pages Wuhan University, China; Dalian University of Technology, China; IEEE Antennas and Propagation Society; Scientific Research Publishing, USA; IEEE Communications Society, Dalian, China, 2008 2008. College of Traffic and Transport Engineering, Central South University, Changsha 410075, China, Inst. of Elec. and Elec. Eng. Computer Society. Compilation and indexing terms, Copyright 2016 Elsevier Inc.; T3: 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008.

[6] ReportLinker. World automobile sector: Top market reports, http://www.reportlinker.com/ci02294/automotive.html, accessed:14/06/2016.

[7] Lv Peng, Zheng Bo, and Zhou Zhongyong. Simulation of vanet in a more realistic scenario. In *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, page 3 pp., Piscataway, NJ, USA, 23-25 Sept. 2011 2011. 95007 Troops, Commander Dept., PLA, Guangzhou, China, IEEE. T3: 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM);.

[8] G. Youssef, N. Idboufker, K. Elbaamrani, and R. Elassali. Exploration and suitability of numerous routing protocol in vanet. In *2013 International Conference on Industrial Engineering and Systems Management (IESM)*, page 1 pp., Piscataway, NJ, USA, 28-30 Oct. 2013 2013. IEEE. T3: 2013 International Conference on Industrial Engineering and Systems Management (IESM). Proceedings.

[9] Kyung Hee University Networking Lab. Ad-hoc networks introduction, http://networking.khu.ac.kr/layouts/net/research/res33.htm, accessed: 04/06/2016.

[10] I. A. Sumra, H. Hasbullah, and J. A. Manan. Vanet security research and development ecosystem. In *Energy & Sustainability: Exploring the Innovative Minds*, 2011 National Postgraduate Conference (NPC 2011), page 4 pp., Los Alamitos, CA, USA, 19-20 Sept. 2011 2011. Comput. Inf. Sci. Dept., Univ. Teknol. PETRONAS, Tronoh, Malaysia, IEEE Computer Society. T3: 2011 National Postgraduate Conference (NPC 2011). Energy Sustainability: Exploring the Innovative Minds.

[11] Liming Zheng, Wanlei Li, and Bo Xie. Research on communications over vanet under different scenes and implementation of vehicle terminal. In *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2012)*, page 7 pp., Piscataway, NJ, USA, 21-23 Sept. 2012 2012. Dept. of Electron. Eng., Jinan Univ., Guangzhou, China, IEEE. T3: 2012 8th Interna-

tional Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2012).

[12] P. Fazio, F. De Rango, and A. Lupia. A new application for enhancing vanet services in emergency situations using the wave/802.11p standard. In *2013 IFIP Wireless Days (WD 2013)*, page 3 pp., Piscataway, NJ, USA, 13-15 Nov. 2013 2013. DIMES Dept., Univ. of Calabria, Cosenza, Italy, IEEE. T3: 2013 IFIP Wireless Days (WD 2013).

[13] Dennis Bodson. Ieee standards revisions and approvals, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5770989, accessed: 14/06/2016.

[14] G. M. Abdalla, M. Abu-Rgheff, and S. M Senouci. Space-orthogonal frequency-time medium access control (soft mac) for vanet. In *2009 Global Information Infrastructure Symposium (GIIS)*, page 8 pp., Piscataway, NJ, USA, 23-26 June 2009 2009. Univ. of Plymouth, Plymouth, United Kingdom, IEEE. T3: 2009 Global Information Infrastructure Symposium (GIIS).

[15] K. Bilstrup, E. Uhlemann, E. G. Strom, and U. Bilstrup. Evaluation of the ieee 802.11p mac method for vehicle-to-vehicle communication. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5, 2008. ID: 1.

[16] Nasser Torabi and Behrouz Shahgholi Ghahfarokhi. Implementation of the ieee 802.11p/1609.4 dsrc/wave in ns-2. In *4th International Conference on Computer and Knowledge Engineering, ICCKE 2014, October 29, 2014 - October 30*, pages 519–524, Azadi Square, Mashhad, Iran, 2014 2014. Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, Institute of Electrical and Electronics Engineers Inc. Compilation and indexing terms, Copyright 2016 Elsevier Inc.; T3: Proceedings of the 4th International Conference on Computer and Knowledge Engineering, ICCKE 2014.

[17] J. B. Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–82, 07 2011.

[18] Lusheng Miao, K. Djouani, B. J. Van Wyk, and Y. Hamam. Performance evaluation of ieee 802.11p mac protocol in vanets safety applications. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1663–8, Piscataway, NJ, USA, 7-10 April 2013 2013. French South African Inst. of Technol. (F'SATI), Tshwane Univ. of Technol., Tshwane, South Africa, IEEE. T3: 2013 IEEE Wireless Communications and Networking Conference (WCNC).

[19] Jinjie Guo, Yiding Huo, Chang Hu, Tianning Liang, Yu Liu, and Lin Zhang. An adaptive and reliable mac mechanism for ieee 1609.4 and 802.11p vanets. In *2012 15th International Symposium on Wireless Personal Multimedia Communications, WPMC 2012, September 24, 2012 - September 27*, pages 55–59, Taipei, Taiwan, 2012 2012. Key Lab of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, China, IEEE Computer Society. Compilation and indexing terms, Copyright 2016 Elsevier Inc.; T3: International Symposium on Wireless Personal Multimedia Communications, WPMC.

[20] K. S. Bilstrup, E. Uhlemann, and E. G. Strom. Scalability issues of the mac methods stdma and csma of ieee 802.11p when used in vanets. In *2010 International Conference On Communications Workshops*, page 5 pp., Piscataway, NJ, USA, 23-27 May 2010 2010. Centre for Res. on Embedded Syst., Halmstad Univ., Halmstad, Sweden, IEEE. T3: 2010 International Conference On Communications Workshops.

[21] Ieee standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirementspart 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, 29th March 2012.

[22] Vaishali D. Khairnar and Ketan Kotecha. Performance of vehicle-to-vehicle communication using ieee 802.11p in vehicular ad-hoc network environment. *CoRR*, abs/1304.3357, 2013.

[23] B. Sikdar. An analytic model for the delay in ieee 802.11 pcf mac-based wireless networks. *IEEE Transactions on Wireless Communications*, 6(4):1542–1550, 2007. ID: 1.

[24] I. Inan, F. Keceli, and E. Ayanoglu. Analysis of the 802.11e enhanced distributed channel access function. *IEEE Transactions on Communications*, 57(6):1753–1764, 2009. ID: 1.

[25] C. Han, M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen. Analytical study of the ieee 802.11p mac sublayer in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):873–886, 2012. ID: 1.

[26] J. Misic, G. Badawy, S. Rashwand, and V. B. Misic. Tradeoff issues for cch/sch duty cycle for ieee 802.11p single channel devices. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6, 2010. ID: 1.

[27] C. Campolo, A. Vinel, A. Molinaro, and Y. Koucheryavy. Modeling broadcasting in ieee 802.11p/wave vehicular networks. *IEEE Communications Letters*, 15(2):199–201, 2011. ID: 1.

[28] Hongseok Yoo and Dongkyun Kim. Dynamic channel coordination schemes for ieee 802.11p/1609 vehicular networks: A survey. *International Journal of Distributed Sensor Networks*, (827317), 2013.

[29] C. l. Huang and W. Liao. Throughput and delay performance of ieee 802.11e enhanced distributed channel access (edca) under saturation condition. *IEEE Transactions on Wireless Communications*, 6(1):136–145, 2007. ID: 1.

[30] A. Dogman, R. Saatchi, and S. Al-Khayatt. Improving quality of service in ieee 802.11e enhanced distributed channel access protocol. In *Communication Systems,*

Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on, pages 1–6, 2012. ID: 1.

[31] 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, 2010. ID: 1.

[32] Yi Wang, A. Ahmed, B. Krishnamachari, and K. Psounis. Ieee 802.11p performance evaluation and protocol enhancement. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 317–322, 2008. ID: 1.

[33] H. Wu, X. Wang, Q. Zhang, and X. Shen. Ieee 802.11e enhanced distributed channel access (edca) throughput analysis. In *2006 IEEE International Conference on Communications*, volume 1, pages 223–228, 2006. ID: 1.

[34] 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 8: Medium access control (mac) quality of service enhancements, 2005. ID: 1.

[35] A. S. K. Mammu, U. Hernandez-Jayo, and N. Sainz. Cluster-based mac in vanets for safety applications. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, pages 1424–1429, 2013. ID: 1.

[36] J. Rezgui, S. Cherkaoui, and O. Chakroun. Deterministic access for dsrc/802.11p vehicular safety communication. In *2011 7th International Wireless Communications and Mobile Computing Conference*, pages 595–600, 2011. ID: 1.

[37] H. J. F. Qiu, I. W. H Ho, C. K. Tse, and Yu Xie. A methodology for studying 802.11p vanet broadcasting performance with practical vehicle distribution. *IEEE Transactions on Vehicular Technology*, 64(10):4756–69, 10 2015.

[38] M. Segata and R. L. Cigno. Simulation of 802.11 phy/mac: The quest for accuracy and efficiency. In *2012 9th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 99–106, Piscataway, NJ, USA, 9-11 Jan. 2012 2012. DISI, Univ. of Trento, Trento, Italy, IEEE. T3: 2012 9th Annual Conference on Wireless On-demand Network Systems and Services (WONS).

[39] The MathWorks. Matlab key features, http://uk.mathworks.com/products/matlab/features.html, accessed: 04/06/2016.

[40] The network simulator - ns-2, http://www.isi.edu/nsnam/ns/, accessed: 14/06/2016.

[41] D. Jiang M. Torrent-Moreno L. Delgrossi Q. Chen, F. Schmidt-Eisenlohr and H. Hartenstein. Overhaul of ieee 802.11 modeling andsimulation in ns-2, in proceedings of the 10th acm symposium on modeling, analysis, and simulation of wireless and mobile systems,mswim. *07, 22-26 Oct. 2007.*

[42] J. A. Barria. Traffic theory & queueing systems course notes, 06/01/2012.

[43] F. Peng, J. Zhang, and W. E. Ryan. Adaptive modulation and coding for ieee 802.11n. In *2007 IEEE Wireless Communications and Networking Conference*, pages 656–661, 2007. ID: 1.

[44] M. Boban, T. T. V. Vinhoza, M. Ferreira, J. Barros, and O. K. Tonguz. Impact of vehicles as obstacles in vehicular ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 29(1):15–28, 2011. ID: 1.

[45] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000. ID: 1.

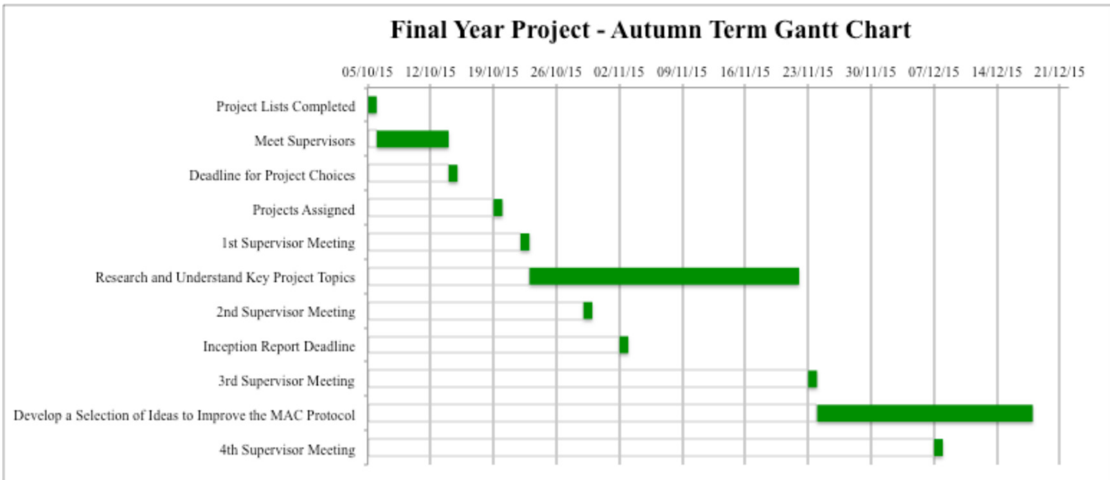
Cover Page Photo Credit: Sanjana George

Appendix A: Gantt Charts

Final Year Project - Autumn Term Gantt Chart

Title:	Make Cars Talk		
Supervisor:	Professor K. K. Leung	**Name:**	Sanjana George
PID:	1500109	**CID:**	00731650

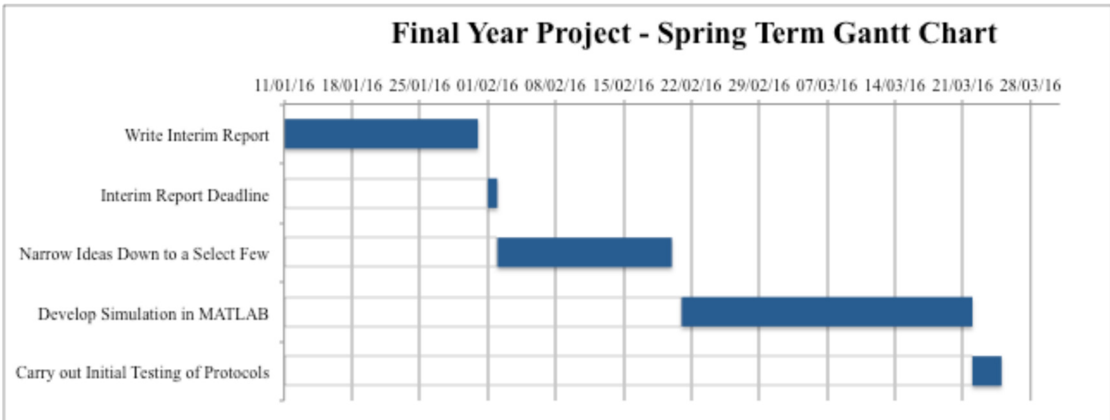
	Task Name	Start	End	Duration (days)
Autumn Term	Project Lists Completed	05/10/15	05/10/15	1
	Meet Supervisors	06/10/15	14/10/15	8
	Deadline for Project Choices	14/10/15	14/10/15	1
	Projects Assigned	19/10/15	19/10/15	1
	1st Supervisor Meeting	22/10/15	22/10/15	1
	Research and Understand Key Project Topics	23/10/15	22/11/15	30
	2nd Supervisor Meeting	29/10/15	29/10/15	1
	Inception Report Deadline	02/11/15	02/11/15	1
	3rd Supervisor Meeting	23/11/15	23/11/15	1
	Develop a Selection of Ideas to Improve the MAC Protocol	24/11/15	18/12/15	24
	4th Supervisor Meeting	07/12/15	07/12/15	1



Final Year Project - Spring Term Gantt Chart

Title: Make Cars Talk
Supervisor: Professor K. K. Leung
PID: 1500109
Name: Sanjana George
CID: 00731650

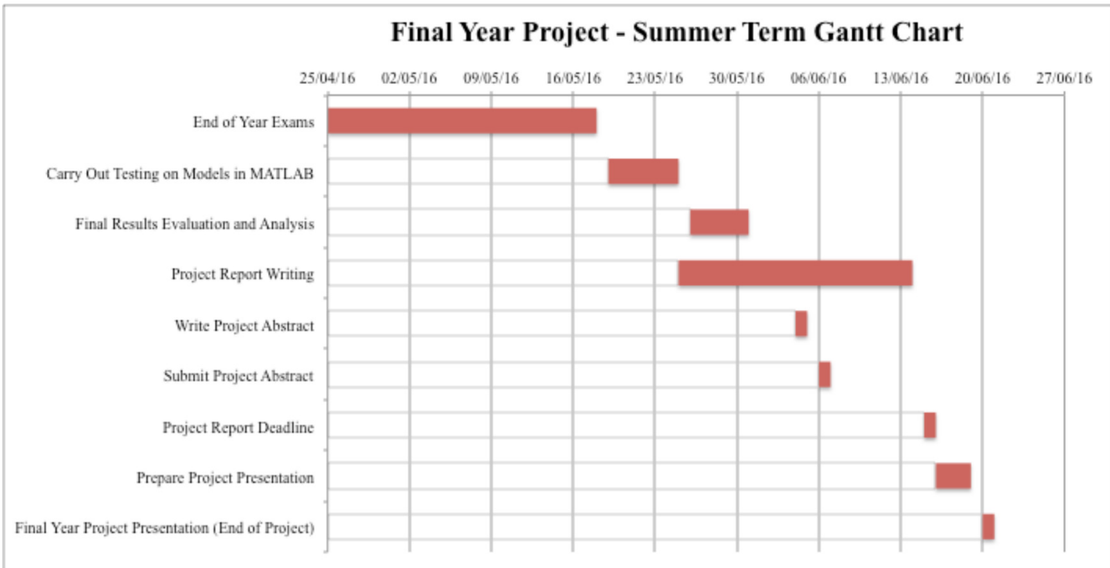
	Task Name	Start	End	Duration (days)
Spring Term	Write Interim Report	11/01/16	31/01/16	20
	Interim Report Deadline	01/02/16	01/02/16	1
	Narrow Ideas Down to a Select Few	02/02/16	20/02/16	18
	Develop Simulation in MATLAB	21/02/16	22/03/16	30
	Carry out Initial Testing of Protocols	22/03/16	25/03/16	3



Final Year Project - Summer Term Gantt Chart

Title: Make Cars Talk
Supervisor: Professor K. K. Leung   **Name:** Sanjana George
PID: 1500109   **CID:** 00731650

	Task Name	Start	End	Duration (days)
Summer Term	End of Year Exams	25/04/16	18/05/16	23
	Carry Out Testing on Models in MATLAB	19/05/16	25/05/16	6
	Final Results Evaluation and Analysis	26/05/16	31/05/16	5
	Project Report Writing	25/05/16	14/06/16	20
	Write Project Abstract	04/06/16	05/06/16	1
	Submit Project Abstract	06/06/16	06/06/16	1
	Project Report Deadline	15/06/16	15/06/16	1
	Prepare Project Presentation	16/06/16	19/06/16	3
	Final Year Project Presentation (End of Project)	20/06/16	20/06/16	1



Appendix B: Model 1 MATLAB Code

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Model 1: The CSMA/CA Protocol Used in the DCF

clear variables;
close all;
clc;

slot_time = 20;     %microseconds
c = 0;              %number of collisions
retry_limit = 10;   %finite number of reattempts

rep = 1000;         %number of trial repetitions
packetSize = 300;   %packet size of 300 bytes
p_channel_busy      = 0.01:0.01:0.99;   %probability of a busy channel
totalLatency        = zeros(rep,length(p_channel_busy));
totalThroughput     = zeros(size(totalLatency));

min_window = 31;    %min CW size
max_window = 1023;  %max CW size

packetsLost = zeros(1, length(p_channel_busy));

for n = 1:rep

    for i = 1:length(p_channel_busy)

        %Generate uniformly distributed random number
        idle = rand(1);
```

```
%When channel busy
while (idle < p_channel_busy(i) && c<retry_limit)

    %Increment no. of collisions.
    c = c + 1;

    %Find uniform distribution limit for back-off
    %CW min and max act as bounds
    if (2^c - 1) < min_window
        backoff = min_window;
    elseif (2^c-1)>max_window
        backoff = max_window;
    else
        backoff = 2^c - 1;
    end

    %Find exponential back-off slot
    slot_choice = round(backoff*rand(1));

    %Update total Latency as an ongoing sum
    totalLatency(n,i) = totalLatency(n,i) + slot_choice*slot_time;

    %Reset idle probability
    idle = rand(1);
end

%Pack is lost if retry limit is reached
if c==retry_limit
    packetsLost(i) = packetsLost(i)+1;
end

%Calculate final total latency for each trial, where 58 is an
%example AIFS time
totalLatency(n,i) = totalLatency(n,i) + (c+1)*58;
%reset number of collisions to zero
c = 0;
```

```
    end
end

%Caulcuate mean latencies and mean throughput
meanTotalLatency    = mean(totalLatency);
meanTotalThroughput = packetSize./meanTotalLatency;

%graph plots for throughput, latency and packets lost
f1 = figure;
stem(p_channel_busy,packetsLost);
grid on;
xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
ylabel('Packets Lost','interpreter','latex','FontSize',14);
title('Packets Lost Against Channel Busy Probability','interpreter','latex',
'FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);

f1=figure;
stem(p_channel_busy,log10(meanTotalLatency));
grid on;
xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
ylabel('log$_{10}$(Latency($\mu$s))','interpreter','latex','FontSize',14);
title('Log Latency Against Channel Busy Probability','interpreter','latex',
'FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);

f2=figure;
stem(p_channel_busy,(meanTotalThroughput));
grid on;
xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
ylabel('log$_{10}$(Throughput(bytes/($\mu$s))','interpreter','latex','FontSize',14);
title('Log Throughput Against Channel Busy Probability','interpreter','latex',
'FontSize',14);
set(f2, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);
```

Appendix C: Function to Define Slot Choice After a Collision

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Code Title: Function to Define Slot Choice After a Collision

function slot_choice = ChooseCWDelay(min_window, max_window, c)

    %Find uniform distribution limit for back-off
    %CW min and max act as bounds
    if (2^c - 1) < min_window
        index = min_window;
    elseif (2^c-1)>max_window
        index = max_window;
    else
        index = 2^c - 1;
    end

    %Find exponential back-off slot
    slot_choice = round(index*rand(1));
end
```

Appendix D: Model 2 MATLAB Code

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Model 2: An Improved Version of the CSMA/CA Protocol Used in the DCF

clear variables;
close all;
clc;

slot_time = 20;     %microseconds
c = 0;              %number of collisions
retry_limit = 10;   %finite number of reattempts

rep = 1000;         %number of trial repetitions
packetSize = 300;   %packet size of 300 bytes
min_window = 31;    %min CW size
max_window = 1023;  %max CW size

%let one integer be the same duration as slot time
%create three different channels with busy states of probability of 0.25
%0.5 and 0.75 respectively and length 1000
%set the first value of each channel to 1, so it starts busy
N = 1000;
percBusy = [0.25;0.50;0.75];
t = zeros(length(percBusy),N);
t1 = [ones(1,percBusy(1)*N),zeros(1,(1-percBusy(1)).*N)];
t1 = t1(randperm(length(t1)));
t2 = [ones(1,percBusy(2)*N),zeros(1,(1-percBusy(2)).*N)];
t2 = t2(randperm(length(t2)));
t3 = [ones(1,percBusy(3)*N),zeros(1,(1-percBusy(3)).*N)];
t3 = t3(randperm(length(t3)));
```

```
t(1,:) = t1; t(2,:) = t2; t(3,:) = t3;
t(1,1) = 1;  t(2,1) = 1;  t(3,1) = 1;
i = 1;
k = 1;

%set variables for whether a packet is at the front of the queue
%whether the CW is free and the number of collisions
ReachedFrontQueue = 0;
isCWFree = 0;
c = 0;

totalLatency = zeros(1,rep);
TotalLatencyChannel = zeros(size(t,1),rep);
rep = 1000;

%set to 0 to include the dropped packets in the graph data
%set to 1 to remove the dropped packets from the graph data
removePacketsDropped = 1;

for n = 1:size(t,1)
    for j = 1:rep
        i = 1;

        %initialise all variables at the start of the loop
        packetTransmitted = 0;
        c = 0;
        ReachedFrontQueue = 0;
        isCWFree = 0;

        while (i < N) && (packetTransmitted ~= 1)

            busyFlag = t(n,i);

            %if the channel is busy
            if busyFlag == 1
                %and nothing is in the contention window
```

```
        if isCWFree == 0
            %allocate uniformly distributed delay
            %increment the number of collisions
            %acknowledge data is now present in the CW
            c = c + 1;
            backoff = ChooseCWDelay(min_window, max_window, c);
            isCWFree = 1;
        end

    %if the channel is free
    elseif busyFlag == 0
        %and the packet is at the front of the queue
        if ReachedFrontQueue == 1
            %reset the CW to being free
            %acknowledge that the packet has been transmitted
            isCWFree = 0;
            packetTransmitted = 1;
        end

        %and the CW is free
        if isCWFree == 0
            %transmit the packet with no added latency
            packetTransmitted = 1;

        %and the CW contains a packet
        elseif isCWFree == 1
            %decrement the back off slot
            backoff = backoff - 1;

            %if the back off slot has decremented to zero
            %the packet has reached the front of the queue
            if backoff == 0
                ReachedFrontQueue = 1;
            end
        end
    end
```

```
            %increment the position in the modelled channel vector
            i = i + 1;
        end

        %total latency calculation
        totalLatency(j) = (i-1)*20;
    end

    %if the removePacketsDropped flag is set to 1
    %remove all dropped packets from the data set
    %this shows the data without it being skewed by packet with huge
    %latencies that are caused by never being transmitted
    if removePacketsDropped == 1
        totalLatency(totalLatency > 4000) = 0;
    end

    %fill the TotalLatencyChannel vector
    TotalLatencyChannel(n,:) = totalLatency;
end

%set colours for each channel busy probability
%black = 0.25
%blue = 0.5
%red = 0.75
%plot graphs for latency and cumulative latency against packet number
colors = ['k';'b';'r'];
f1 = figure;
for i = 1:size(t,1)
    plot(TotalLatencyChannel(i,:),colors(i));
    hold on;
end
xlabel('Packet Number','interpreter','latex','FontSize',14);
ylabel('Latency($\mu$s)','interpreter','latex','FontSize',14);
title('Latency Against Packet Number','interpreter','latex','FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);
```

```
f2 = figure;
for i = 1:size(t,1)
    plot(cumsum(TotalLatencyChannel(i,:)),colors(i));
    hold on;
end
grid on;
xlabel('Packet Number','interpreter','latex','FontSize',14);
ylabel('Cumulative Latency($\mu$s)','interpreter','latex','FontSize',14);
title('Cumulative Latency Against Packet Number','interpreter','latex',
'FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);
```

Appendix E: Model 3 MATLAB Code

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Model 3: The EDCA Mechanism Used in the HCF

clear variables;
close all;
clc;

slot_time = 20;     %microseconds
c = 0;              %number of collisions
retry_limit = 10;   %finite number of reattempts

%the CW Min & Max Sizes for each of the priority levels.
%the SIFS time in microseconds
CWMin    = [15;15;7;3];
CWMax    = [1023;1023;15;7];
SIFS     = 10;
AIFSN    = [9;6;3;2];

%formula to calculate AIFS from AIFSN
AIFS     = slot_time.*AIFSN + SIFS;

%the max TXOP time for each AC in ms
MaxTXOP = [0;0;3.008;1.504];

rep = 1000;         %number of trial repetitions
packetSize = 300;   %packet size of 300 bytes
p_channel_busy      = 0.01:0.01:0.99;   %probability of a busy channel

totalLatency        = zeros(rep,length(p_channel_busy));
```

```
totalThroughput      = zeros(size(totalLatency));
packetsLost          = zeros(1  , length(p_channel_busy));
%packets lost for each busy probability
%% Background and Best Effort Base Modelling : Without TXOP
% first 2 elements for low-priority when TXOP = 0ms.

 for j = 1:2            %CW min and max for the first two ACs
    totalLatency        = zeros(rep,length(p_channel_busy));
    totalThroughput     = zeros(size(totalLatency));
    for n = 1:rep       %for 1000 repetitions
        min_window = CWMin(j);
        max_window = CWMax(j);

        for i = 1:length(p_channel_busy)

            %Generate uniformly distributed random number
            idle = rand(1);

            %When channel busy
            while (idle < p_channel_busy(i) && c<retry_limit)

                %Increment no. of collisions.
                c = c + 1;

                %Find uniform distribution limit for back-off
                %CW min and max act as bounds
                if (2^c - 1) < min_window
                    backoff = min_window;
                elseif (2^c-1)>max_window
                    backoff = max_window;
                else
                    backoff = 2^c - 1;
                end

                %Find exponential back-off slot
                slot_choice = round(backoff*rand(1));
```

```
            %Update total Latency as an ongoing sum
            totalLatency(n,i) = totalLatency(n,i) + slot_choice*slot_time;

            %Reset idle probability
            idle = rand(1);
        end

        %Pack is lost if retry limit is reached
        if c == retry_limit
            packetsLost(i) = packetsLost(i) + 1;
        end

        %Calculate final total latency for each trial using
        %the AIFS time
        totalLatency(n,i) = totalLatency(n,i) + (c+1)*AIFS(j);
        %reset number of collisions to zero
        c = 0;
    end
end

%graph plots for throughput, latency and packets lost
f1 = figure;
stem(p_channel_busy, packetsLost);
grid on;
xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
ylabel('Packets Lost','interpreter','latex','FontSize',14);
title('Packet Loss Vs. Channel Busy Probability','interpreter','latex',
'FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);
f1 = figure;
subplot(1,2,1);
stem(p_channel_busy, log10(mean(totalLatency)));
grid on;
xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
ylabel('log$_{10}$(Latency($\mu$s))','interpreter','latex','FontSize',14);
```

```
    title('Log Latency Against Channel Busy Probability','interpreter','latex',
    'FontSize',14);
    subplot(1,2,2);
    stem(p_channel_busy, log10(1*packetSize./mean(totalLatency)));
    grid on;
    xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
    ylabel('log$_{10}$(Throughput(bytes/($\mu$s))','interpreter','latex',
    'FontSize',14);
    title('Log Throughput Against Channel Busy Probability','interpreter','latex',
    'FontSize',14);
    set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);
 end

%% Video & Voice Base Modelling: TXOP

%the TXOP will be varied according to the channel busy probability
k = 1;
N = length(p_channel_busy);

%probability that a TXOP occurs in a set time period.
TXOP_Prob = (p_channel_busy-0.5).^3 + 0.5;
TXOP_Prob = k*[TXOP_Prob(floor(N/2)+1:end),TXOP_Prob(1:floor(N/2))+0.2352];
TXOP_Prob = TXOP_Prob./max(TXOP_Prob);
Time_Unit = 200;

%plot of the TXOP variation with channel busy probability
f1 = figure;
plot(p_channel_busy,TXOP_Prob);
grid on;
xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
ylabel('TXOP Time Allocation','interpreter','latex','FontSize',14);
title('Designed TXOP Against Channel Busy Probability','interpreter','latex',
'FontSize',14);

%number of collisions starts as zero
c = 0;
```

```
%j is set to be the value of the relevant AC
j = 3;
min_window = CWMin(3);
max_window = CWMax(3);
dataRate   = 5;          %Byte/us
packetCount = zeros(rep, length(p_channel_busy));
%total number of transmitted packets
TXOP_Time   = zeros(rep, length(p_channel_busy));
%total amount of time spent in TXOP
packetsLost = zeros(1  , length(p_channel_busy));
%packets lost for each busy probability
LatencyWhereItOccurs = 0;
%stores slot time when last TXOP period occured

for n = 1:1000
    for i = 1:length(p_channel_busy)

        %Generate uniformly distributed random number
        idle = rand(1);

        %When channel busy
        while (idle < p_channel_busy(i) && c<retry_limit)

            %Increment no. of collisions
            c = c+1;

            %Find uniform distribution limit for back-off
            %CW min and max act as bounds
            if (2^c - 1) < min_window
                backoff = min_window;
            elseif (2^c-1)>max_window
                backoff = max_window;
            else
                backoff = 2^c - 1;
            end
```

```
        %Find exponential back-off slot
        slot_choice = round(backoff*rand(1));

        %Update total Latency as an ongoing sum
        totalLatency(n,i) = totalLatency(n,i) + slot_choice*slot_time;

        %Probability that TXOP occurs in the time that has passed
        %since the last TXOP
        ScaleToTime_Prob = min (( totalLatency(n,i)- LatencyWhereItOccurs )
        /Time_Unit , 1)*TXOP_Prob(i);

        %Generate uniformly distributed random number
        TXOP_Flag = rand(1);

        %If this is less than the TXOP probability then begin TXOP
        if TXOP_Flag < ScaleToTime_Prob

            %Update the time for when the last TXOP has occured
            LatencyWhereItOccurs = totalLatency(n,i);

            %Calculate the real time spent in TXOP
            TXOP_TimeLeft = MaxTXOP(j)*1000 - slot_choice*slot_time;

            %Update the packet count
            packetCount(n,i) = packetCount(n,i) + dataRate*(TXOP_TimeLeft);

            %Update the time spent in TXOP
            TXOP_Time(n,i) = TXOP_Time(n,i) + MaxTXOP(j)*1000;
        end

        %Reset idle probability
        idle = rand(1);
    end

    %resets slot time when last TXOP period occured
```

```
        LatencyWhereItOccurs = 0;

        %Packet is lost if retry limit is reached
        if c == retry_limit
            packetsLost(i) = packetsLost(i) + 1;
        end

        %reset number of collisions to zero
        c = 0;
    end
end

%calculate the average total time so that Throughput can be calculated
TotalTime = mean(TXOP_Time + totalLatency);

%calculate the maximum possible Throughput using the max data rate
maxRate = dataRate*ones(1,length(p_channel_busy));

%graph plot of Mean Throughput against Channel Busy Probability
%compared to Line of Maximum Throughput
f1 = figure;
plot(p_channel_busy,mean(packetCount)./TotalTime);
hold on;
plot(p_channel_busy,maxRate,'k--','LineWidth',2);
grid on;
xlabel('P(Channel Medium Busy)','interpreter','latex','FontSize',14);
ylabel('Mean Throughput(bytes/($\mu$s))','interpreter','latex',
'FontSize',14);
title('Mean Throughput Against Channel Busy Probability','interpreter',
'latex','FontSize',14);
ylim([dataRate-3 dataRate+1]);
```

Appendix F: Model 3 Additional Graphs

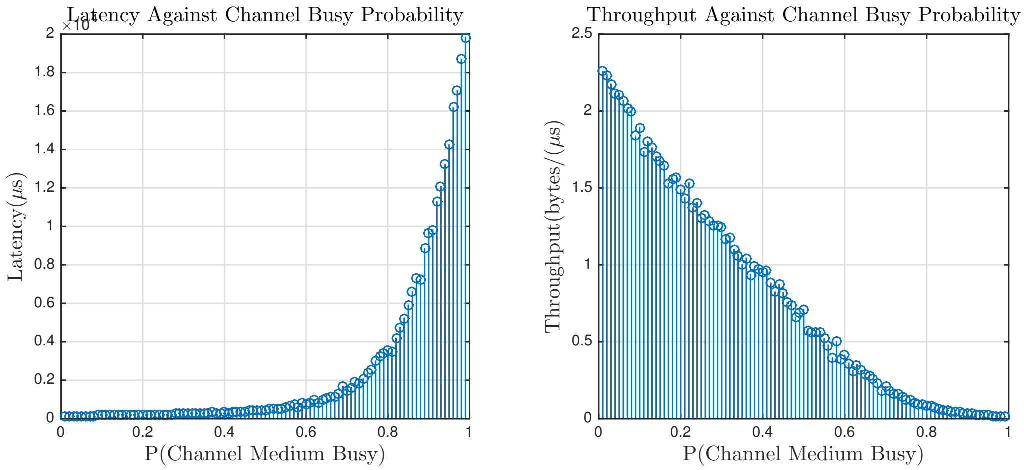


Figure 42: Plots of the Latency and Throughput Against Channel Busy Probability for the 2nd Access Category

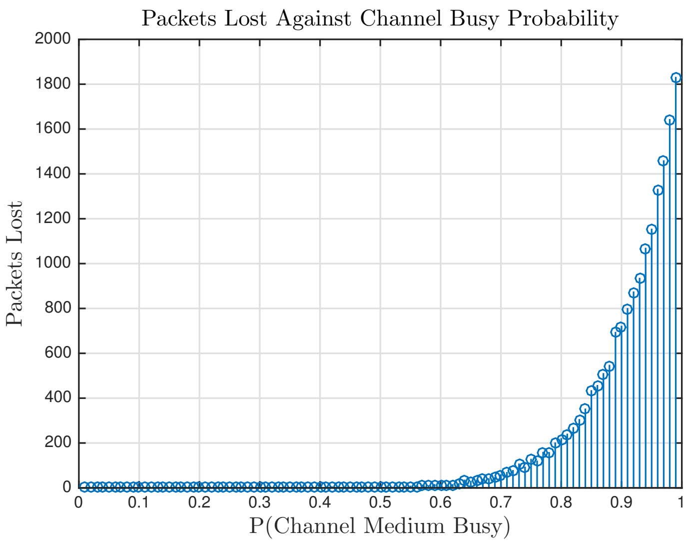


Figure 43: Plot of the Packets Lost Against the Channel Busy Probability for the 2nd Access Category

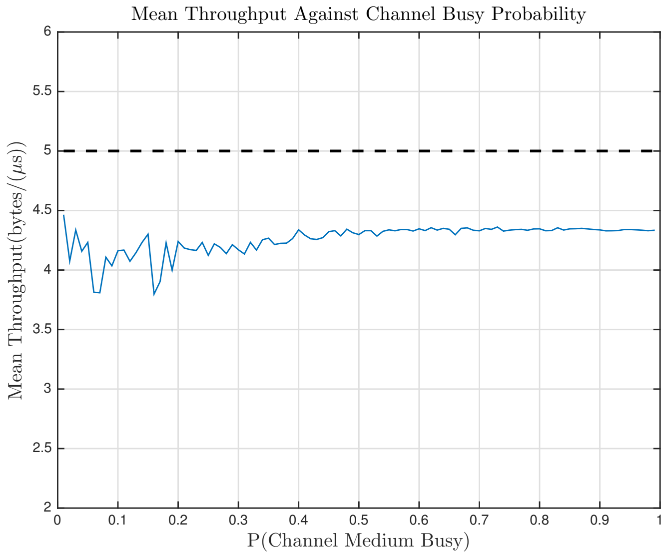


Figure 44: Plot of the Mean Throughput Against Channel Busy Probability for the 4th Access Category Using a Data Rate of 5 bytes/μs

Appendix G: Model 4 MATLAB Code Part 1/3

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Model 4a: An Improved Version of the EDCA Mechanism Used in the HCF for
           %Background and Best Effort

clear variables;
close all;
clc;

slot_time = 20;      %microseconds
c = 0;               %number of collisions
retry_limit = 10;    %finite number of reattempts
```

```
rep = 1000;           %number of trial repetitions
packetSize = 300;     %packet size of 300 bytes

%the CW Min & Max Sizes and AIFS times for each of the priority levels
CWMin    = [15;15];
CWMax    = [1023;1023];
AIFS = [100;60];
AIFS_Slot = AIFS./slot_time; %How many slots the AIFS takes up (5 & 3)
MaxTXOP = [0;0];              %No TXOP is present

%let one integer be the same duration as slot time
%create three different channels with busy states of probability of 0.25
%0.5 and 0.75 respectively and length 1000
%set the first value of each channel to 1, so it starts busy
N = 1000;
percBusy = [0.25;0.50;0.75];
t = zeros(length(percBusy),N);
t1 = [ones(1,percBusy(1)*N),zeros(1,(1-percBusy(1)).*N)];
t1 = t1(randperm(length(t1)));
t2 = [ones(1,percBusy(2)*N),zeros(1,(1-percBusy(2)).*N)];
t2 = t2(randperm(length(t2)));
t3 = [ones(1,percBusy(3)*N),zeros(1,(1-percBusy(3)).*N)];
t3 = t3(randperm(length(t3)));
t(1,:) = t1; t(2,:) = t2; t(3,:) = t3;
t(1,1) = 1;  t(2,1) = 1; t(3,1)  = 1;

i = 1;  %Loop counter
k = 1;  %How busy the channel is, the row number of 't'
ac = 2; %The AC number

totalLatency = zeros(1,rep);
TotalLatencyChannel = zeros(size(t,1),rep);

AIFS_Count = 0;
packetTransmitted = 0;
ReachedFrontQueue = 0;
```

```
isCWFree = 0;

for n = 1:size(t,1)
    for j = 1:rep

        %initialise all variables at the start of the loop
        i = 1;
        packetTransmitted = 0;
        c = 0;
        ReachedFrontQueue = 0;
        isCWFree = 0;
        AIFS_Count = 0;


        while (i < N) && (packetTransmitted ~= 1)

            busyFlag = t(n,i);

            %if the channel is busy
            if busyFlag == 1
                %set AIFS count to zero
                AIFS_Count = 0;
                %and nothing is in the contention window
                if isCWFree == 0
                    %allocate uniformly distributed delay
                    %increment the number of collisions
                    %acknowledge data is now present in the CW
                    c = c + 1;
                    backoff = ChooseCWDelay(CWMin(ac), CWMax(ac), c);
                    isCWFree = 1;
                end

            %if the channel is free
            elseif busyFlag == 0
                %increment the AIFS_Count by 1
                AIFS_Count = AIFS_Count + 1;
```

```
            %and the packet is at the front of the queue
            if ReachedFrontQueue == 1
                %reset the CW to being free
                %acknowledge that the packet has been transmitted
                isCWFree = 0;
                packetTransmitted = 1;
            end

            %and the CW is free
            if isCWFree == 0
                    %transmit the packet with no added latency
                    packetTransmitted = 1;

            %and the CW contains a packet and the AIFS_Count has
            %reached the AIFS number of slots for that AC
            elseif isCWFree == 1 && AIFS_Count == AIFS_Slot(ac)
                %decrement the back off slot
                %reset the AIFS_Count to 0
                backoff = backoff - 1;
                AIFS_Count = 0;

                %if the back off slot has decremented to zero
                %the packet has reached the front of the queue
                if backoff == 0
                    ReachedFrontQueue = 1;
                end
            end
        end

        %increment the position in the modelled channel vector
        i = i + 1;
    end

    %total latency calculation
    totalLatency(j) = (i-1)*20;
end
```

```
    %fill the TotalLatencyChannel vector
    TotalLatencyChannel(n,:) = totalLatency;
end

%set colours for each channel busy probability
%black = 0.25
%blue = 0.5
%red = 0.75
%plot graphs for latency and cumulative latency against packet number
colors = ['k';'b';'r'];
f1 = figure;
for i = 1:size(t,1)
    plot(TotalLatencyChannel(i,:),colors(i));
    hold on;
end
xlabel('Packet Number','interpreter','latex','FontSize',14);
ylabel('Latency($\mu$s)','interpreter','latex','FontSize',14);
title('Latency Against Packet Number','interpreter','latex',
'FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);

f2 = figure;
for i = 1:size(t,1)
    plot(cumsum(TotalLatencyChannel(i,:)),colors(i));
    hold on;
end
grid on;
xlabel('Packet Number','interpreter','latex','FontSize',14);
ylabel('Cumulative Latency($\mu$s)','interpreter','latex','FontSize',14);
title('Cumulative Latency Against Packet Number','interpreter','latex',
'FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);
```

Appendix H: Model 4 MATLAB Code Part 2/3

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Model 4b: An Improved Version of the EDCA Mechanism Used in the HCF for
           %Video and Voice

clear variables;
close all;
clc;

slot_time = 20;     %microseconds
c = 0;              %number of collisions
retry_limit = 10;   %finite number of reattempts

rep = 1000;         %number of trial repetitions
packetSize = 300;   %packet size of 300 bytes
ArrivalRate = 100;  %packet arrival rate in microseconds

%the CW Min & Max Sizes and AIFS times for each of the priority levels
CWMin   = [7;3];
CWMax   = [15;7];
AIFS = [40;40];
AIFS_Slot = AIFS./slot_time; %How many slots the AIFS takes up 2 & 2
MaxTXOPSlot = [150;75];      %Length of the TXOP for each AC

%let one integer be the same duration as slot time
%create three different channels with busy states of probability of 0.25
%0.5 and 0.75 respectively and length 1000
%set the first value of each channel to 1, so it starts busy
N = 1000;
percBusy = [0.25;0.50;0.75];
```

```
t = zeros(length(percBusy),N);
t1 = [ones(1,percBusy(1)*N),zeros(1,(1-percBusy(1))*N)];
t1 = t1(randperm(length(t1)));
t2 = [ones(1,percBusy(2)*N),zeros(1,(1-percBusy(2))*N)];
t2 = t2(randperm(length(t2)));
t3 = [ones(1,percBusy(3)*N),zeros(1,(1-percBusy(3))*N)];
t3 = t3(randperm(length(t3)));
t(1,:) = t1; t(2,:) = t2; t(3,:) = t3;
t(1,1) = 1;  t(2,1) = 1; t(3,1)  = 1;

TXOPOccurence = zeros(1,N);

%Controls whether TXOP is enabled or not
TXOPOccurence(125) = 1; TXOPOccurence(375) = 1;
TXOPOccurence(625) = 1; TXOPOccurence(875) = 1;

%Makes sure TXOP Vector and time vector are the same length.
assert(length(TXOPOccurence) == size(t,2));

i = 1;  %Loop counter
k = 1;  %How busy the channel is, the row number of 't'
ac = 1; %The AC number

AIFS_Count = 0;
packetTransmitted = 0;
ReachedFrontQueue = 0;
isCWFree = 0;

TXOPCounter = 0;         %tracks how long TXOP has lasted so far
TXOPActive = 0;          %flag for whether TXOP is active
TimeLastPacketSent = 0; %the time the last packet was sent
numPacketsSent = 0;      %total number of packets sent for a certain repetition

numPacketsTotal = zeros(size(t,1), rep);
totalLatency = zeros(1,rep);
TotalLatencyChannel = zeros(size(t,1),rep);
```

```
for n = 1:3              %loop through the 3 channel busy probabilities
    for j = 1:rep

        %initialise all variables at the start of the loop
        i = 1;
        c = 0;
        AIFS_Count = 0;
        packetTransmitted = 0;
        ReachedFrontQueue = 0;
        isCWFree = 0;
        TXOPCounter = 0;
        TXOPActive = 0;
        TimeLastPacketSent = 0;
        numPacketsSent = 0;

        %move forward in time in terms of slot time spaces
        while (i < N)

            %check if channel is busy
            busyFlag = t(n,i);
            % Check if TXOP has started
            TXOPFlag = TXOPOccurence(i);

            %% TXOP Not active: Run Channel as usual
            if TXOPFlag == 0 && TXOPActive == 0

                %if the channel is busy
                if busyFlag == 1
                    %set AIFS count to zero
                    AIFS_Count = 0;
                    %and nothing is in the contention window
                    if isCWFree == 0
                        %allocate uniformly distributed delay
                        %increment the number of collisions
                        %acknowledge data is now present in the CW
```

```
            c = c + 1;
            backoff = ChooseCWDelay(CWMin(ac), CWMax(ac), c);
            isCWFree = 1;
        end

    %if the channel is free
    elseif busyFlag == 0
        %increment the AIFS_Count by 1
        AIFS_Count = AIFS_Count + 1;

        %and the packet is at the front of the queue
        if ReachedFrontQueue == 1
            %reset the CW to being free
            isCWFree = 0;

            %the packets arrive at rate of 100microseconds
            %a packet can only be transmitted if it has arrived
            %fully
            if ((i-TimeLastPacketSent))>=5
                %increment number of packets sent
                numPacketsSent = numPacketsSent + 1;
                %update the time the last packet was sent
                TimeLastPacketSent = i;
            end

        %and the CW is free
        elseif isCWFree == 0

            %the packets arrive at rate of 100microseconds
            %a packet can only be transmitted if it has arrived
            %fully
            if ((i-TimeLastPacketSent))>=5
                %increment number of packets sent
                numPacketsSent = numPacketsSent + 1;
                %update the time the last packet was sent
                TimeLastPacketSent = i;
```

```
                end

            %and the CW contains a packet and the AIFS_Count has
            %reached the AIFS number of slots for that AC
            elseif isCWFree == 1 && AIFS_Count == AIFS_Slot(ac)
                %decrement the back off slot
                %reset the AIFS_Count to 0
                backoff = backoff - 1;
                AIFS_Count = 0;

                %if the back off slot has decremented to zero
                %the packet has reached the front of the queue
                %and reset the CW to be free
                if backoff == 0
                    ReachedFrontQueue = 1;
                    isCWFree = 0;
                end
            end
        end

    %% TXOP Active: Run Channel including TXOP
    elseif (TXOPFlag == 1 || TXOPActive == 1)

        %TXOP is active, so set flag to 1
        TXOPActive = 1;

        %Count the number of slot times during TXOP
        TXOPCounter = TXOPCounter + 1;

        %the number of packets sent is 0.2 per TXOP period since
        %20us/100us = 0.2
        numPacketsSent = numPacketsSent + 0.2;

        %if we have reached the TXOP time limit
        %reset the counter
        %and set the active flag to 0
```

```
                if TXOPCounter == MaxTXOPSlot(ac)
                    TXOPActive = 0;
                    TXOPCounter = 0;
                end
            end

            %increment the position in the modelled channel vector
            i=i+1;

        end

        %total latency calculation
        totalLatency(j) = (i-1)*slot_time;
        numPacketsTotal(n,j) = numPacketsSent;
    end
    TotalLatencyChannel(n,:) = totalLatency;
end

%When TXOP is enabled: Minimum value of 60 =75*4*0.2
%When TXOP is not enabled: Minimum value of 0
f1=figure;
plot(numPacketsTotal.');
xlabel('Trial Number','interpreter','latex','FontSize',14);
ylabel('Number of Packets Transmitted','interpreter','latex','FontSize',14);
title('Number of Packets Transmitted Against Trial Number','interpreter',
'latex','FontSize',14);
set(f1, 'Units','centimeters', 'Position',[0 0 0.8*21 0.4*29.7]);
```

Appendix I: Model 4 MATLAB Code Part 3/3

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Model 4b: An Improved Version of the EDCA Mechanism Used in the HCF for
           %Video and Voice

clear variables;
close all;
clc;

slot_time = 20;     %microseconds
c = 0;              %number of collisions
retry_limit = 10;   %finite number of reattempts

rep = 100;         %number of trial repetitions
packetSize = 300;   %packet size of 300 bytes
ArrivalRate = 100;  %packet arrival rate

%the CW Min & Max Sizes and AIFS times for each of the priority levels
CWMin   = [7;3];
CWMax   = [15;7];
AIFS = [40;40];
AIFS_Slot = AIFS./slot_time; %How many slots the AIFS takes up 2 & 2
MaxTXOPSlot = [150;75];      %Length of the TXOP for each AC

ac = 1; %VIDEO: ac = 1, AUDIO: ac = 2;
N = 1000; % Number of slot times to run

percBusy = 0.01:0.01:0.99;
Z = length(percBusy);
numPacketsTotal = zeros(Z, rep);
```

```
TotalLatencyChannel = zeros(Z, rep);
totalLatency = zeros(1, rep);
numPacketsTXOPTotal = [];

for T = 1:2

    if T == 1
        %TXOP does not occur
        TXOPOccurence = zeros(1,N);
    else
        %TXOP occurs
        TXOPOccurence = zeros(1,N);
        %For ac = 1, TXOP = 3ms so let it only occur twice
        %For ac = 2, TXOP = 1.5ms so let it occur four times
        if ac == 1
            TXOPOccurence(250) = 1; TXOPOccurence(750) = 1;
        else
            TXOPOccurence(125) = 1; TXOPOccurence(375) = 1;
            TXOPOccurence(625) = 1; TXOPOccurence(875) = 1;
        end
    end

    %loop through all channel busy probability settings
    for z = 1:Z

        %Generate a busy channel of a certain proportion.
        timeVector = [ones(1,round(percBusy(z)*N)),zeros(1,round
        ((1-percBusy(z))*N))];

        for j = 1:rep

            %initialise all variables at the start of the loop
            i = 1;
            AIFS_Count = 0;
            packetTransmitted = 0;
            ReachedFrontQueue = 0;
```

```
        isCWFree = 0;
        TXOPCounter = 0;
        TXOPActive = 0;
        c = 0;
        TimeLastPacketSent = 0;
        numPacketsSent = 0;

        %move forward in time in terms of slot time spaces
        while (i < N)

            %check if channel is busy
            busyFlag = timeVector(i);
            %check if TXOP has started
            TXOPFlag = TXOPOccurence(i);

            %% TXOP Not active: Run Channel as usual.
            if TXOPFlag == 0 && TXOPActive == 0

                %if the channel is busy
                if busyFlag == 1
                    %set AIFS count to zero
                    AIFS_Count = 0;
                    %and nothing is in the contention window
                    if isCWFree == 0
                        %allocate uniformly distributed delay
                        %increment the number of collisions
                        %acknowledge data is now present in the CW
                        c = c + 1;
                        backoff = ChooseCWDelay(CWMin(ac), CWMax(ac), c);
                        isCWFree = 1;
                    end

                %if the channel is free
                elseif busyFlag == 0
                    %increment the AIFS_Count by 1
                    AIFS_Count = AIFS_Count + 1;
```

```
%and the packet is at the front of the queue
if ReachedFrontQueue == 1
    %reset the CW to being free
    isCWFree = 0;

    %the packets arrive at rate of 100microseconds
    %a packet can only be transmitted if it has arrived
    %fully
    if ((i-TimeLastPacketSent))>=5
        %increment number of packets sent
        numPacketsSent = numPacketsSent + 1;
        %update the time the last packet was sent
        TimeLastPacketSent = i;
    end

%update the time the last packet was sent
elseif isCWFree == 0

   %the packets arrive at rate of 100microseconds
   %a packet can only be transmitted if it has arrived
   %fully
   if ((i-TimeLastPacketSent))>=5
       %increment number of packets sent
        numPacketsSent = numPacketsSent + 1;
        %update the time the last packet was sent
        TimeLastPacketSent = i;
   end

%and the CW contains a packet and the AIFS_Count has
%reached the AIFS number of slots for that AC
elseif isCWFree == 1 && AIFS_Count == AIFS_Slot(ac)
    %decrement the back off slot
    %reset the AIFS_Count to 0
    backoff = backoff - 1;
    AIFS_Count = 0;
```

```
                %if the back off slot has decremented to zero
                %the packet has reached the front of the queue
                if backoff == 0
                    ReachedFrontQueue = 1;
                end
            end
        end

    %% TXOP Active: Run Channel Including TXOP
    elseif (TXOPFlag == 1 || TXOPActive == 1)

        %TXOP is active, so set flag to 1
        TXOPActive = 1;

        %Count the number of slot times during TXOP
        TXOPCounter = TXOPCounter + 1;

        %the number of packets sent is 0.2 per TXOP period since
        %20us/100us = 0.2
        numPacketsSent = numPacketsSent + 0.2;

        %if we have reached the TXOP time limit
        %reset the counter
        %and set the active flag to 0
        if TXOPCounter == MaxTXOPSlot(ac)
            TXOPActive = 0;
            TXOPCounter = 0;
        end
    end

    %increment the position in the modelled channel vector
    i=i+1;
end

%total latency calculation
```

```
            totalLatency(j) = (i-1)*slot_time;
            numPacketsTotal(z,j) = numPacketsSent;

        end
        TotalLatencyChannel(z,:) = totalLatency;
    end
    numPacketsTXOPTotal = [numPacketsTXOPTotal;numPacketsTotal];
end



meanPackets = mean(numPacketsTXOPTotal, 2);
meanPackets = reshape(meanPackets, [], 2);
figure;
plot(percBusy, meanPackets);
xlabel('Percentage of Time Channel Is Busy (\%) ','interpreter','latex'
,'FontSize',12);
ylabel('Number of Packets Sent','interpreter','latex','FontSize',12);
title('Number of Packets Sent Against Percentage of Time Channel is Busy',
'interpreter','latex','FontSize',14);
legend('No TXOP','TXOP');
grid on;

figure;
plot(percBusy, abs(meanPackets(:,2)-meanPackets(:,1)));
grid on;
xlabel('Percentage of Time Channel Is Busy (\%) ','interpreter',
'latex','FontSize',12);
ylabel('TXOP Improvement in Number of Packets Sent','interpreter',
'latex','FontSize',12);
title('TXOP Improvement Against Percentage of Time Channel is Busy',
'interpreter','latex','FontSize',14);
```

Appendix J: Model 4 Additional Graphs

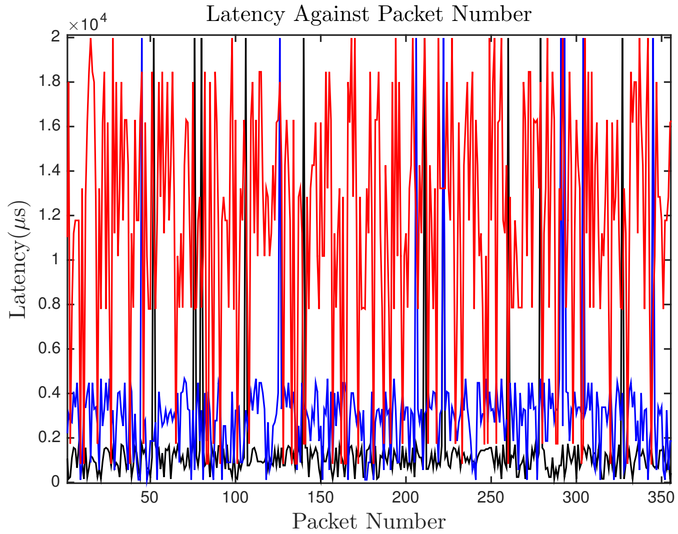


Figure 45: Plot of Latency Against Packet Number for the 2nd Access Category

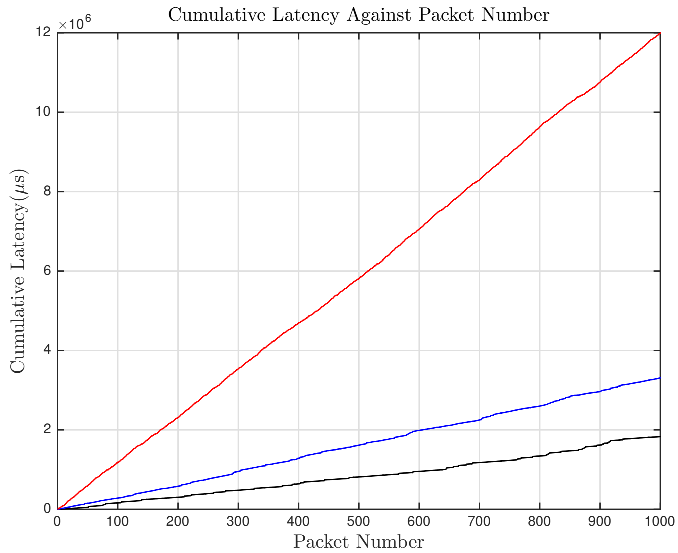


Figure 46: Plot of Cumulative Latency Against Packet Number for the 2nd Access Category

Appendix K: Model 5 MATLAB Code

```
%Name: Sanjana George
%CID: 00731650
%Project Name: Making Cars Talk
%Project Supervisor: Professor K. K. Leung
%Model 5: A Dynamic Time Boundary for the CCH and SCH Subintervals

clear variables;
close all;
clc;

msPerPacket = 1;          %let each packet take 1ms to transmit
maxPacketSize = 100;      %let each packet have size 100bytes
N = 220;                  %length of the input stream
SyncTime = 100;           %every 100ms synchronisation occurs
t = linspace(0,N*100/1000,N); %scales the x-axis time to be correct in seconds

%INPUT STREAMS THAT WERE TRIALLED
% %Random input stream.
%inputStream   = round(maxPacketSize*rand(2, N));

% %Linearly increasing input stream.
%inputStream   = round(maxPacketSize*[linspace(0,0.4,N)+0.02*rand(1,N);
%linspace(0.3,0.7,N)+0.02*rand(1,N)]);

% %Exponentially decreasing input streams.
%inputStream   = round(maxPacketSize*[ 0.5*exp(-linspace(0,3,N))+20 ;
%exp(-linspace(-0.5,2.5,N))+20]);

%define the input stream vector for the CCH and SCH
%top line of vector = CCH
%bottom line of vector = SCH
y = linspace(50,150,20);
```

```
inputStream1=[50*ones(1,40), y, 150*ones(1,140-60),flip(y),50*ones(1,40),y];
inputStream2 = -inputStream1+150;
inputStream = [inputStream1',inputStream2']';

%initialise the time boundary to be 50ms:50ms
CCH = zeros(1,N+1); CCH(1) = 50;
SCH = zeros(1,N+1); SCH(1) = 50;
ServiceDropped = zeros(1, N);
ControlDropped = zeros(1, N);

%calculate the maximum number of packets that can be transmitted on either
%channel and create vector for it
maxPacketsFixed = 50/msPerPacket;
refPackets = (50/msPerPacket)*ones(2,N);

%set the formula for the number of packets dropped
%place a minimum bound at 0 so that results are not negative
streamDropped = (inputStream - refPackets).';
streamDropped(streamDropped < 0) = 0;

%choose intervals based on safety message statistics rather than both
%choose service period after choosing control period
%CCH weighting
alpha = 0.6;

%set minimum subinterval window time for either channel to be 20ms
%thus making the maximum bound for a channel 80ms
minBound = 20;
maxBound = 100-minBound;

for i = 1:N

    %split the input stream into the CCH and SCH data
    currentPacketsControl = inputStream(1,i);
    currentPacketsService = inputStream(2,i);
```

```
    %calculate the maximum possible packet transmission for each stream
    maxPossibleService = SCH(i)/msPerPacket;
    maxPossibleControl = CCH(i)/msPerPacket;

    %packets dropped on each channel
    %is equal to the number of packets that come in
    %minus the maximum possible number of messages that can be transmitted
    ControlDropped(i)  = max(currentPacketsControl-maxPossibleControl,0);
    ServiceDropped(i)  = max(currentPacketsService-maxPossibleService,0);

    %weight the total number of packets depending on priority
    scaledPacketsControl = alpha*currentPacketsControl;
    scaledPacketsService = (1-alpha)*currentPacketsService;

    %update CCH subinterval based on ratios of previous 100ms time interval
    CCH(i+1) = (scaledPacketsControl/(scaledPacketsControl+scaledPacketsService))
    *100;
    if CCH(i+1) > maxBound
        CCH(i+1) = maxBound;
    elseif CCH(i+1)<minBound
        CCH(i+1) = minBound;
    end

    %update SCH subinterval accordingly
    SCH(i+1) = 100-CCH(i+1);
end

figure;
plot(t,inputStream');
ylim([0 200]);
grid on;
xlabel('Time (s)','interpreter','latex','FontSize',12);
ylabel('Number of Packets on the Channel','interpreter','latex','FontSize',12);
title('Input Stream of the Control and Service Channels','interpreter',
'latex','FontSize',14);
legend('SCH','CCH');
```

```
figure;
plot(t, streamDropped(:,1),'b');
hold on;
plot(t, streamDropped(:,2),'r');
grid on;
legend('SCH','CCH');
xlabel('Time (s)','interpreter','latex','FontSize',12);
ylabel('Number of Packets Dropped from the Channel','interpreter','latex',
'FontSize',12);
title('Number of Packets Dropped Using a Static Time Boundary','interpreter',
'latex','FontSize',14);

figure
plot(t, streamDropped(:,1),'b');
hold on;
plot(t, streamDropped(:,2),'r');
hold on;
plot(t, ControlDropped,'b--');
hold on;
plot(t, ServiceDropped,'r--');
grid on;
legend('Original SCH','Original CCH','New SCH Dropped','New CCH Dropped');
xlabel('Time (s)','interpreter','latex','FontSize',12);
ylabel('Number of Packets Dropped','interpreter','latex','FontSize',12);
title('Use of Static Time Boundary (Solid) Against Use of Dynamic Time Boundary
(Dashed)','interpreter','latex','FontSize',14);

figure;
plot(sum(streamDropped'),'b');
hold on;
plot(ControlDropped+ServiceDropped,'r');
grid on;
legend('Static Total Packets Dropped','Dynamic Total Packets Dropped');
xlabel('Time (s)','interpreter','latex','FontSize',12);
ylabel('Sum of Packets Dropped on Both Channels Combined','interpreter',
```

```
'latex','FontSize',12);
title('Use of Static Time Boundary Against Use of Dynamic Time Boundary',
'interpreter','latex','FontSize',14);
```