2022

QUALITY OF EXPERIENCE ASSURANCE IN WIRELESS NETWORKS FOR MPEG-4 TRANSMISSION

MD. KHAMBARI, MOHD NAJWAN

http://hdl.handle.net/10026.1/19206

http://dx.doi.org/10.24382/1029
University of Plymouth

COPYRIGHT STATEMENT

UNIVERSITY OF PLYMOUTH

QUALITY OF EXPERIENCE ASSURANCE IN WIRELESS NETWORKS FOR MPEG-4 TRANSMISSION

by

MOHD NAJWAN BIN MD KHAMBARI

A thesis submitted to the University of Plymouth

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Mathematics and Engineering

March 2022

ACKNOWLEDGEMENTS

First and foremost, I express my sincere gratitude to the Almighty God for showering me with so many blessings.

This work would not have been possible without the support and dedication of my supervision team. I would like to thank my Director of Study, Associate Professor Dr Bogdan Ghita for his continuous support and excellent guidance in completion of this thesis. I would like to further extend my gratitude to my co-supervisors Dr David Lancaster and Dr Lingfen Sun for their excellent professional input and commitment in the process of conducting this research. Without their invaluable support, this work would not have been possible. It has been a privilege working together and I hope this collaboration can continue in the future.

I am grateful to the Centre for Security, Communications and Network Research (CSCAN) of Plymouth University for giving me the opportunity to further my PhD studies. Special thanks to Universiti Teknikal Malaysia Melaka (UTeM) for their scholarship that helped me to achieve my goals.

Finally, my biggest gratitude to my parents Mr. Khambari and Mrs. Norhuda for their inspiration, my wife Mrs. Nurulhuda and my children Iman, Ihsan and Ilham for their love, encouragement, patience, sacrifice and understanding in completing this thesis. I could not wish for a better family.

AUTHOR'S DECLARATION

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee. Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

Relevant scientific seminars and conferences were regularly attended at which work was often presented:

Publications:

Khambari, N., Ghita, B., Lancaster, D., Sun, L., "QoE Enhancements in IEEE 802.11 e EDCA for Video Transmission through Selective Queueing.", Proceedings of the *11th International Network Conference (INC),* 2016. Frankfurt, Germany, pp. 85-90. ISBN: 978-1-84102-410-3

Khambari, N., Ghita, B., Sun, L., "QoE-driven Video Enhancements in Wireless Networks through Predictive Packet Drops", Proceedings of the *IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* 2017. Rome, Italy. DOI: 10.1109/WiMOB.2017.8115811

Khambari, N., Ghita, B., "QoE Enhancements for Video Traffic in Wireless Networks through Selective Packet Drops", Proceedings of the *5th International Conference on Computational Science and Technology (ICCST 2018),* 2018, Kota Kinabalu, Malaysia. DOI: 10.1007/978-981-13-2622-6_29

Word count for the main body of this thesis: **31, 837**

Signature:

Date: 14 March 2022

ABSTRACT

QUALITY OF EXPERIENCE ASSURANCE IN WIRELESS NETWORKS FOR MPEG-4 TRANSMISSION

MOHD NAJWAN BIN MD KHAMBARI

The evolution of IEEE 802.11 wireless local area network (WLAN) has emerged as the most preferred choice for organizations as well as end users and consumers to access the network, especially the Internet. Today, WLAN technology has becoming more mature and slowly taking over the job for information sharing which was once borne by wired networks. WLAN has not only been used to transfer data packets, but also multimedia packets that are more time sensitive. Given the nature of WLAN which is prone to interference, several challenges need to be encountered for WLAN to keep abreast with the ability of wired network, especially in terms of performance and user experience. Among the challenges of WLAN is the support of Quality of Experience (QoE) to stream video which demands special attention and immediate solution. Failing to address this demand in a timely and effective manner will eventually limit the success of WLAN.

This thesis presents a novel QoE provisioning technique in the IEEE 802.11e networks. With these techniques, video traffic can be prioritized so that the end users will have better video quality especially during wireless network congestion. This is done by looking at the types of videos frames and prioritises the more important ones (I-Frame). These techniques use two different algorithms, namely congestion-triggered based and PSNR-triggered based. Within the former method, I-Frames are prioritized when the system detects that the network is congesting. Less important video frames (B-Frames) will be dropped from the queue whenever an I-Frame enters the MAC queue. Meanwhile in PSNR-triggered based, I-Frames will be given priority over B-Frames when the end user's predicted PSNR level is below a certain threshold. Simulations results done using NS-3 showed that improvements on the proposed schemes, both in objective and

subjective evaluation. On implementing the proposed techniques, better QoE can be achieved where better PSNR and MOS were recorded. Finally, a brief discussion on the constraints during experiment implementations and future directions of QoE in WLAN conclude this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

TABLE OF NOMENCLATURE

WLAN	Wireless Local Area Network
QoS	Quality of Service
QoE	Quality of Experience
OSI	Open System Interconnection
VBR	Variable Bit Rate
MPEG4	Moving Picture Expert Group 4
HCF	Hybrid Coordination Function
EDCA	Enhanced Distributed Channel Access
HCCA	HCF Controlled Channel Access
PSNR	Peak Signal to Noise Ratio
GoP	Group of Pictures
PktI	Packet that carries I-Frame information
PktP	Packet that carries P-Frame information
PktB	Packet that carries B-Frame information
BE	Best Effort
GS	Guaranteed Service
DS	Differentiated Service
IntServ	Integrated Service
RSVP	Resource Reservation Protocol

MOS	Mean Opinion Score
CP	Contention Period
CFP	Contention Free Period
HC	Hybrid Controller
QAP	QoS Access Point
QSTA	QoS Station
TXOP	Transmit Opportunity
TSPEC	Transmission Specification
IFS	Interframe Space
AIFS	Arbitrary IFS
CW	Contention Window
RTS/ CTS	Request to Send/ Clear to Send
QF	Queue Full
PP	Predicted PSNR
Q-ROPB	Queue Full - event-based trigger and Remove Own PktB
Q-RAPB	Queue Full - event-based trigger and Remove Any PktB
P-ROPB	Predicted PSNR – event-based trigger, Remove Own PktB
P-RAPB	Predicted PSNR – event-based trigger, Remove Any PktB
α	Dependency value of P-Frames and B-Frames towards I-Frame
I_n, P_n and B_n	Number of packets needed to encode I Frame, P-Frame and B-Frame within a GoP
FFMPEG	Fast Forward MPEG
VO	Voice Traffic
VI	Video Traffic
BE	Best Effort Traffic
BK	Background Traffic

ET/ ETMP4	Evaluate Trace for MP4 Video
FV	Fix Video tool

CHAPTER 1: INTRODUCTION

1 Introduction

The strong and growing demand for WLANs in both consumer markets such as residential networks, and industrial markets such as retail, education, and healthcare has been documented repeatedly in business, industry and education (Braden et al., 1994; IEEE Std 802.1D, 2004; Ohrtman, 2004; Khambari et al., 2007; Khan et al., 2008; Arabi et al., 2010; Debnath, 2012). As the network world becomes more popular, the network load has become a critical issue. The wireless LAN (WLAN), which was originally designed to carry data traffic such as file transfer, e-mail and Internet browsing), is now being used to carry real-time and multimedia traffic that requires fast yet reliable transmission.

In recent years, a lot of research has focused on providing Quality of Service (QoS) by evaluating the quality of the network performance parameters based on the lower layers of the Open Systems Interconnection (OSI). These parameters include packet loss, throughput, delay, jitter, and bandwidth utilization. On the other hand, IEEE has also amended the IEEE 802.11 standard to support QoS, which is known as the IEEE 802.11e standard (IEEE Std. 802.11e-2005, 2005). Besides that, there is also a dedicated amendment on the wireless LAN standard for multicast video transmission. This amendment is named the IEEE 802.11aa (IEEE Std. 802.11aa - 2012, 2012).

Since the mid-2000s, high demands for video services over the Internet has been observed (Vassiliou et al., 2006). On the same time, the popularity of the Internet led to integrating video communications into the best-effort packet networks. In networks, Variable Bit Rate (VBR) is a term used that relates to the bitrate used in video encoding where it defines the data output variation per time segment. Transporting video through best-effort packet network is complicated because different video has different bit rates and thus the network cannot predict the best mechanism to transmit the video from the sender to the receiver.

On the other hand, wireless LAN represents the preferred network access and option and is being widely used (Kosek-Szott et al., 2013). This is due to deployment costs, variety of features, and ease of setup, providing an excellent platform for generic data transfer. However, transmitting video traffic over dynamic environment such as wireless networks remain a challenging issue in spite of the progress made through the IEEE 802.11e framework and the associated research. This is because of the nature of the wireless networks itself where the bandwidth, delay, and loss are not known in advance and are unbounded.

Since WLAN is now used to stream multimedia traffic, relying on QoS parameters is no longer sufficient. To evaluate the user-perceived application quality, looking to parameters beyond QoS is now vital. Thus, this is where the term Quality of Experience (QoE) has been introduced. With QoE, a measurement of end-to-end performance at the services level from the users' perspective can be assessed and evaluated. New approaches are required to define performance measurement while considering the subjective nature of the users (Jain, 2003).

By considering of MPEG-4 videos transmission across wireless networks, this research presented in this thesis aims to improve the performance of video streams (VBR traffic) in terms of the Quality of Experience. This will be achieved by considering the video structure and protecting the most important frames to maintain the video quality. Furthermore, this work will also try to investigate sharing the network resources fairly in an environment where CBR and generic data transfers exist as well.

1.1 Demands of Video Streaming in Mobile Networks

In recent years, the growing demand of video content has spurred the research on video streaming over wireless network. Cisco in their annual networking index report have predicted that wireless networks and video streaming will represent most traffic types on the Internet. For example, based on the network trends (Cisco, 2017) predicts that 63% of the total IP traffic will be from wireless traffic by 2021. In the same report, it is stated that in 2016, 73% of the total traffic are video-based, and the percentage will increase to 82% by 2021.

The predictions of wireless and video traffic to increase are not something new. In 2010 (Stanley, 2010) projected that 69% of the mobile traffic are accounted for video traffic by 2014. This is based on a massive trend that consumers are looking forward to watch video content through streaming. The prediction is actually not far off where in 2014, Cisco reported that 64% of the global consumer Internet traffic is consisted of video traffic (Cisco, 2015). The statistics was reflected in a report by (Ramachandran, 2014) in The Wall Street Journal that the number of paid cable TV subscribers declined over the years as "video streaming services lure customers away from traditional pay TV". Meanwhile, Nielsen (2014) also confirms the predictions where it showed that there is an obvious increase of people watching video through the Internet while a decline is also observed on subscribers watching the traditional TV in the US.

In 2015, (Cisco, 2015) issued a Visual Networking Index (VNI) report and again, predicted an increase in video traffic across the Internet and an increase of traffic from wireless and mobile devices. By 2019, video traffic will be 80% of all consumer Internet traffic while Wi-Fi and mobile devices will account 66% of the IP traffic.

Cisco in their annual network forecasting report (Cisco, 2011, 2012, 2013, 2014, 2015, 2016, 2017), the Cisco Visual Networking Index (VNI) has recorded annual statistics of the types of traffic and the methods used by end users to access the Internet globally. From 2010, significant growth of both wireless and video traffic has been recorded. Figure 1.1 below shows the percentage of the wireless and video traffic in six years since 2010.

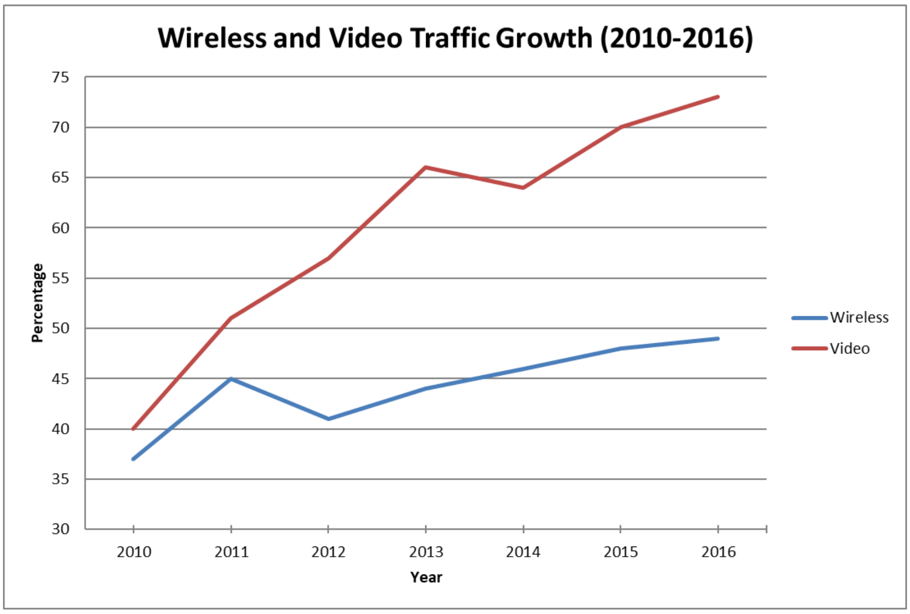


Figure 1.1: Wireless and Video Traffic Percentage between 2010 and 2016.

1.2 Aims and Objectives

In recent years, a lot of research on designing a robust and adaptive mechanism for video streaming across wireless network has been carried out. However, transporting video traffic across wireless medium remains a challenge.

This thesis focuses on enhancing the QoE of video streaming in the IEEE 802.11e network. This is done by optimizing the shared bandwidth and adaptively queue the video frames based on the saturation level of the network resources i.e., queue buffer and bandwidth. Important video frames will be protected while the least important frames will be selectively dropped while preserving the quality of the video.

The objectives of this thesis are as follows:

- To assess the state of the art of video streaming over IEEE 802.11e wireless networks and the background literature within the infrastructure mode that implements the Enhanced Distributed Channel Access (EDCA) in terms of maintaining video quality as perceived by end users. These include prior work within the research community that implements various techniques and solutions to optimally utilize the available bandwidth and queue buffer to prevent packet loss while prioritizing video frames ahead of other non-media packets thus enhancing video streaming experience within the limited network resources.
- To identify the weaknesses of the current wireless networks in handling video traffic transmission efficiently. Though IEEE 802.11 is being implemented widely throughout the globe, streaming video remains a challenging issue. Although EDCA introduced new methods for streaming multimedia files, there is still room for improvement to enhance video streaming for a better Quality of Experience (QoE).

- To propose innovative algorithm to address the issue of video packets being dropped within the MAC layer. During network congestion, the MAC layer can no longer buffer video packets and thus packets are being dropped. Protecting important video frames is a very challenging task as to preserve the quality of the video. Thus, selectively dropping less important video frames is crucial.
- To extend the functionality of the above proposed scheme to support QoE based threshold to selectively drop less important video frames. In this thesis, this extended functionality would be able to predict the PSNR level of the receiver by tracking the packets that has been dropped at the sender. Less important video frames will selectively be dropped once the PSNR level drops to a certain threshold.
- To evaluate the performance of the proposed algorithms and compare them with the previous IEEE standards and research. The main parameter would be Peak Noise to Signal Ratio (PSNR), where PSNR will be used as the benchmark of triggering the proposed algorithm as well as doing video quality predictions to adjust video packet prioritization. Further, a frame-by-frame analysis to compare video quality between the default algorithm and the proposed solutions are made to evaluate the video quality subjectively, as experienced by end users.

1.3 Achievements of the Thesis

The research findings in this thesis presented significant contributions to offer better QoE for video transmission across wireless connection. Preserving the quality of a video remains a challenging task especially in network-congested scenarios, hence this thesis proposed methods to overcome those challenges. Specifically, the contributions of this thesis can be described as follows:

- **Introduced novel algorithms of handling high priority packets for better QoE**: In this thesis, new algorithms have been introduced to prioritise packets that will likely affect the quality of the video at the receiving end. These algorithms are based on two types of triggers: congestion-based and PSNR-based triggers.

    - Congestion-based trigger: On the event of a network congestion, packets cannot be sent by the sender and thus making the queue on the sender congested. The queue buffer will eventually be full and resulted to packets, including the high priority packet types to be dropped from the queue.

    - PSNR-based trigger: This proposed algorithm does not depend on network congestion to be enabled, but the PSNR level/ threshold that the sender wants to guarantee. To do this, a method to predict the PSNR level at the receiver was presented and verified in this thesis.

  Upon these triggers, a better management of packet prioritisation are applied to allow important packet types to be queued first and avoid them from being dropped. Congestion-based trigger algorithms introduced in this thesis are the **Q-RAPB** and **Q-ROPB** while the PSNR-based trigger algorithms are **P-RAPB** and **P-ROPB**. From the analyses done in Chapter 5, it can be concluded that the proposed algorithms have been proven to offer better QoE compared to the default algorithms that is being used in the IEEE 802.11e.

- **Relay GoP frame type information to MAC layer**: On the Application layer, the video that is being planned to be transmitted will be converted into GoP, which consists of I-Frame, P-Frame and B-Frame as being discussed in Section 2.1.1. These frames affect the quality of the video differently. As the frames are being sent to the MAC layer to be ready for transmission, these frames are being fragmented into packets and therefore, losing the information of the type of frame. This makes the MAC layer unaware of that vital information. To overcome this challenge, this thesis through Evalvid simulations, has made the frame type information available until the MAC layer, making the MAC Layer aware of the frame type. This cross-layer information passing method has enabled the proposed algorithm to prioritize more important packets by queueing and selectively dropping less important video packets.

- **Method to identify video types**: Different types of video (slow, moderate and rapid) requires different treatments on how they are being transmitted across the network. For example, a rapid-moving video cannot afford to lose a lot of I-Frame loss as every video frame is significantly different from the other, as opposed to a slow-moving video. In Section 3.4.1, this thesis has introduced a method on identifying the video type by analysing the ratio between the different types of packet frames in a GoP. A pattern between these ratios has been identified and presented, where it can be used to differentiate between video types. This is essential especially in queueing and to selectively dropping packets as these different types of video has different tolerance towards packet loss.

- **Method to predict PSNR level on the receiver's side**: For a sender to maintain a certain level of QoE, it is essential to predict the level of the perceived video quality especially in real time. Since Evalvid is streaming the video using RTP/ UDP, there is no feedback from the receiver about the current quality of the video that is received. It is rightly so, as video transmission is time sensitive and therefore transmitting packets using UDP has less delay. However, this poses a great challenge as to know the current QoE level of the video on the receiver's side. In this thesis, a method on how to predict the QoE level of the perceived video based on the PSNR level has been introduced in Section 3.4.2. This is based on thousands of experiment samples of packet drops to see the cause and effect of losing packets from a lower to higher percentage. Based on the observations, a method has been developed to calculate the predicted PSNR so that the proposed algorithms of P-RAPB and P-ROPB can be applied efficiently.

- **Introduced new entities within NS-3/ Evalvid codes**: In order for the experiments to apply the proposed algorithms successfully, a lot of programming are involved especially on the simulation side within NS-3 and Evalvid. New packet tags have been introduced, such as Application ID (AppId), Sequence Id (SeqId) and NodeId (NId) Tags. These tags make the information-passing across different layers, possible. Besides that, new library files are also introduced making the proposed system able to predict the perceived PSNR, based on the calculation of packet losses.

1.4 Thesis Structure

The research detailed in this thesis is divided into six chapters. An overview of each of the remaining chapters is provided as follows.

Chapter 2 begins with an introduction to MPEG-4 video frames, QoS and QoE. It also describes briefly on how Quality of Experience can be evaluated through objective and subjective methods. Then, the IEEE Standards and amendments regarding QoS and video streaming in wireless networks are discussed before looking into previous research which had been done regarding this scope of research.

Chapter 3 presents the novel technique to protect the structure of the video for better QoS. To benchmark the proposed method, the chapter also introduces a simulation testbed that allows a comprehensive evaluation of its strengths and limitations. Finally, the challenge faced to setup the testbed and the preliminary results are presented.

Chapter 4 presents the simulation implementation of the experiments. The tools and metrices are outlined and the techniques on how the experiments are being conduct are later presented.

Chapter 5 discusses the results of the experiment. The efficiency of the proposed algorithm is discussed and analysed. Based on the outcome, it can be determined whether the proposed technique will offer better QoE for end users.

Chapter 6 concludes the thesis. A conclusion to the research is discussed as well as the limitation of this research. Finally, future work is presented to further enhance the performance of the wireless network to provide better QoE for the end users.

CHAPTER 2: VIDEO QUALITY ASSESSMENT AND PREVIOUS WORKS

2 Introduction

This chapter discusses the structure of MPEG-4 video streams and the aspects of QoS and QoE in the context of video communication. A mobile video streaming consists of three main components which are the video compression, the wireless media and the human visual system (HVS) (Su et al., 2016). This chapter starts with an overview of the first component on how a video is constructed and establish the understandings of the structure of MPEG-4. This is the basis on giving priority to important video frames especially in the critical cases where the network resources such as bandwidth and queue buffer are saturated.

Next, the HVS is examined where QoS and QoE are briefly discussed to identify the parameters involved in determining the performance of the video traffic in wireless networks. Finally, IEEE standards which are the wireless media that involve QoS, and multimedia streaming are also discussed before looking into prior research that has been done in this research scope.

2.1 MPEG-4 Video

MPEG-4 has the capability to encode multiple objects within one frame and was developed mainly for storing and delivering multimedia content over the Internet (Watkinson, 2001).

Therefore, in this work, MPEG-4 structure will be used in experiments as the video standard to simulate video transmission in the IEEE 802.11e wireless networks.

The MPEG-4 video structure is made up of three types of video frames: I-Frame, P-Frame, and B-frame.

I-Frame: Intra coded frame. This frame is encoded independently and therefore has no relationships with any other successive or previous frames. I-Frame is the reference frame where the other type of frames will refer to be decoded. I-Frame is the most important frame because other frames cannot be decoded without it.

P-Frame: Predictive coded frame. This frame is predictively coded with the reference of previous I-Frame or the other P-Frames. Therefore, to be decoded, P-frames need to have the information of the previous I-Frame or other P-Frame information. Without the information, P-frames cannot be decoded. P-Frames will be the reference frame for other P-frames as well.

B-Frame: Bidirectional coded frame. This frame is coded based on the information of successive and previous P-Frames and B-Frames. B-Frame is the least important frame because no other frames depend on B-Frames to decode. In other words, if there is an error on the B-Frame, it will not affect the other frames.

2.1.1 Group of Pictures (GoP)

The sequence of the MPEG-4 frames consists of a header and a Group of Pictures (GoP). A GoP determines the number of frames that are grouped together to form a chunk of video and therefore a full length of a video will consist of multiple GoPs. Each GOP contains the I, P and B frames. The pattern of GOP (M, N) is characterized by the distance between anchor pictures (which is I-Frame or P-Frame) denoted as M and the distance between two full images (which is I-Frame) denoted as N. N also refers to the size of the GoP where it is the number of frames counting from the I-Frame to the next I-Frame. The size of a GoP is determined by how the video is being encoded. A GOP (3,15) will have the pattern as in Figure 2.1below:

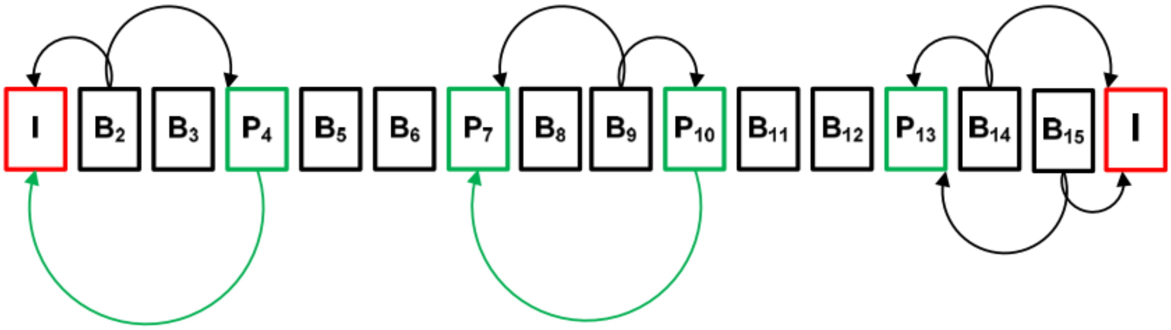


Figure 2.1: A structure of GOP (3,15) and its dependency (Debnath, 2012)

The GoP size of a video affects the size of the video. As the GoP size increase, so as the number of P and B-Frames. Since these frames only show the difference of the previous image, the sizes are small comparatively to I-Frame which contributes a smaller size of a video.

Each frame has a different role and importance; thus, they have different sizes and priority. While in a normal transport-based prioritisation scheme frames would be dealt with equally, their different importance can be exploited to propose a more efficient approach of video transmission for the IEEE 802.11e wireless networks.

GoP settings can be made intelligent. This means that a sender can be configured to adjust the GoP according to the type of the video whether there is high or slow movement. For example, if the stream of the video consists of both high and slow movement, the GoP can be adjusted automatically where at the high movement part, the distance between I-Frames would be shorter (to maintain high quality) while at the slow movement part, the distance between I-Frames can be made longer (to save memory and bandwidth). Since this thesis is focusing on the queuing of the MAC Layer, having an intelligent GoP settings is beyond the scope of the thesis.

I, P and B-Frames are made up by several packets. Throughout this thesis, packets that carry I, P and B-Frames will be addressed as **PktI**, **PktP** and **PktB** respectively.

Using MPEG-4 for multimedia content over the Internet has its own advantages. It does not only save the storage, but also saves bandwidth. It all depends on the structure and size of the GoP

An I-Frame itself is independent and contains a full image of a frame of a video. Meanwhile, P-Frame and B-Frame only contain the information that is different from the I-Frame. Thus, a P-Frame or a B-Frame have a significantly smaller size than an I-Frame. In a video structure where less I-Frame are involved, video size will be smaller and thus, less storage and bandwidth. Longer GoP (distance between two I-Frames) means more storage and bandwidth can be saved while decreasing the risk of being affected by network noise.

Take an example of a video. If that video is being encoded with a small GoP, it is likely to have more I-Frames throughout the whole length of the video. Not only the video has a big size and takes more storage, but it is also prone to quality degradation. This is because if a transmission problem happens, it will be more likely to hit an I-Frame. This is a different if the GoP size is longer where more P-Frame and B-Frames are being used. Transmission noise will likely hit a P-Frame of B-Frame. The more P-Frames and B-Frames being used, the lesser storage and bandwidth are being consumed.

2.2 Quality of Service (QoS)

The term Quality of Service (QoS) was introduced by the International Communication Union in relation to provide a general guide that contribute collectively to the quality of service as perceived by the user and gauge users' satisfaction degree (International Communication Union, 1994). From a network performance point of view, QoS has been defined in several ways, but the meaning still stems from the purpose. For example, Pattara-Atikom (2005) defined QoS as is the ability to provide a level of assurance for data delivery over the network where traffic with different requirements will receive different levels of QoS assurance. Meanwhile (Acharya et al., 2010) defined QoS as is the ability of network element to provide some level of assurance for consistent network data delivery. It can be measured by looking and analysing the parameters of a WLAN that will reflect directly to the definition of performance in a wireless network.

A network that supports QoS must be able to deliver a service needed by a specific network application from end-to end with some level of control over delay, loss, jitter, or bandwidth. Network QoS can be categorized into three levels of service:

Best Effort Service (BE): Basic connectivity with no guarantees and no priority applied to it. It refers to a typical networking queue based on a First In First Out (FIFO) mechanism. Most of the time, it is an easy but expensive to provide QoS. This is because it involves overprovisioning the network by over purchasing bandwidths to accommodate various types of traffic. By doing this, network traffic does not need to compete to use the available bandwidth

Guaranteed Service (GS): A reservation of network resources to ensure that specific traffic gets a specific level of service it requires. It is also popular with the term Integrated Service (IntServ). Using the Resource Reservation Protocol (RSVP), every network device from end to end will reserve a certain amount of bandwidth for the incoming traffic. While the traffic is guaranteed, it consumes a lot of bandwidth. Calls need to be made by network devices to its neighbouring intermediate devices to reserve bandwidth for the incoming traffic. Therefore, it involves a lot of overhead. IntServ also suffers from the scalability issue as all devices need to be RSVP capable

Differentiated Service (DS): Expedited handling for specific classes of traffic. It is one of the complicated methods to provide QoS. This is because a lot of components are involved such as traffic classification, congestion control, collision and congestion avoidance, traffic shaping and traffic policing. However, it is better than IntServ because it the traffic is treated on a "by hop" basis rather than resource reservation throughout the network.

By looking and analysing the network performance parameters, it is possible to measure the performance of a wireless network. Previous research evaluated wireless performance by measuring the QoS parameters. However, each research had their own point of view of determining QoS, thus not all of the parameters were being used. (Anastasi and Lenzini, 2000) focused on throughput and access delay but with some addition in average queuing delay and average MAC delay. (Lindgren et al., 2003) also mentioned that the main metrics to be monitored are throughput, medium utilization, collision rate, access delay and cumulative delay distribution. (Fallah and Alnuweiri, 2005) who conducted their own research also stressed on throughput and delay monitoring.

2.3 Quality of Experience (QoE)

Quality of Service (QoS) has been the standard parameter to measure the performance of a network. Parameters such as throughput, packet loss, delay and jitter has been the statistics of analysing the reliability of a network. However, recent studies showed that QoS might need a complementary perspective, the Quality of Experience (QoE) to meet end demands and requirements.

Quality of Experience is becoming one of the most important topics concerned by the service providers (Vassiliou et al., 2006), as QoS provisioning does not necessarily reflect the satisfaction of end users. This is because today, humans are the quality meters and their expectations to a particular product, service or application carry a great value (Laghari et al., 2012). QoS guarantees technical demands of throughput, delay, and jitter, but only up to the third of the Open System Interconnection (OSI) layer. QoE on the other hand, can evaluate subjective video perception better (Bhamidipati and Kilari, 2010) by assessing and providing another perspective of network performance, from end to end up to the Application layer.

Since the term QoE is being used in the research community, various definitions have been brought up. QoE is defined as the overall acceptability of an application or service, as perceived subjectively by the end-user (Vassiliou et al., 2006). Meanwhile, ITU-T issued a working paper where it defines QoE as the overall acceptability of an application or service as perceived subjectively by end users (ITU-T, 2008). It is basically a subjective measurement of end-to-end performance at the services level, from the point of view of the users. Comparing QoS with Qoe, earlier works of researchers (Noh et al., 2008) (Bhamidipati and Kilari, 2010) has agreed that QoE is on a higher, abstract layer. This is considered to be a perceptual pseudo-layer. QoE is based on subjective parameters where it examines the interaction between contents and the how the users perceived them. In terms of multimedia QoE, it usually refers to voice and video contents. However, this thesis will be focusing on the video quality exclusively. Some of the evaluation of the quality of the videos are the colour, light intensity or failure of some pixel (Maia et al., 2015). Though there are multiple definitions of QoE, the definition of QoE in terms of video quality is user-oriented and can be sum as how a user experience a video in terms of quality perception, feelings and thoughts (Baraković and Skorin-Kapov, 2013).

In a lot of cases, users still complaint on having difficulties to access multimedia contents although the throughput is at an acceptable level (Kuo and Wu, 2011). Meanwhile, different multimedia content needs different requirements. Different types of video require different type of network treatment and prioritization. One of the parameters of evaluating a quality of a video is the number of frames it contains in a second, or frame per second (FPS). A fast-moving video that consists of distinct frames in a second requires sensitive attention compared to slow moving video. For example, in a fast-moving video like sports, each frame in a second is different and distinct. This type of video needs aggressive video prioritisation compared to video like news telecast where movement is very minimal. Meanwhile, a music video for example, needs to ensure audio is rendered at a high quality where prioritisation of audio is higher than video.

In the previous years, subjective psychological issues and human cognitive aspects are typically not considered to determine a video quality. Ironically, these aspects are the ones that are directly determining the Quality of Experience (Laghari et al., 2011). These factors finally need to be considered to successfully evaluate how the end users perceive the multimedia files through network. With a successful QoE guarantee, end-users will benefit and match their expectations of experiencing a video while from a business perspective; happy customers will provide better benefit for the network provider.

2.3.1 Evaluating Video QoE

In the previous section, it has been established that providing a certain amount of QoE is crucial to guarantee users' satisfaction. A video quality is evaluated by examining a video that is passed through a certain transmission system with the perceived degraded quality being compared to the original video. Generally, video quality can be evaluated through the subjective and objective method (Khan et al., 2009), as shown in Figure 2.2 below.

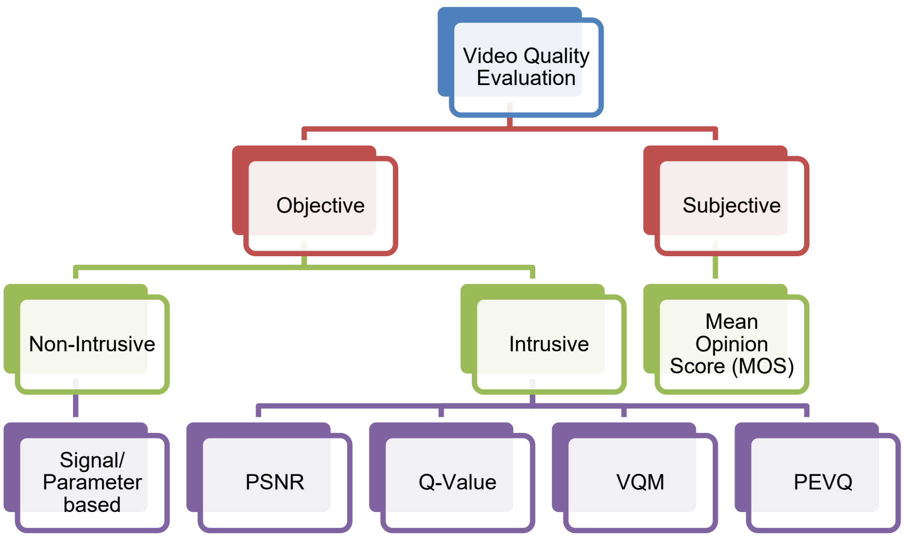


Figure 2.2: Video Quality Evaluation Methods

Objective evaluation estimates the perceived video quality through a mathematical model. It can be categorized in two sub methods – Intrusive, which is based on a signal and parameter-based measurement, and non-Intrusive, which involves comparing the original and the impaired versions of the video. The tools and parameters usually used to determine video quality are Peak Signal to Noise Ratio (PSNR), Q-Value, Video Quality Metric (VQM) and Perceptual Evaluation of Video Quality (PEVQ).

On the other hand, subjective evaluation involves the end users to watch the video as rendered at the receiving end and evaluate the quality of experience through Mean Opinion Score (MOS) Though it is the most reliable method, it is time consuming and involves high cost. MOS is a five-level scale on which subjects will rate a video. The scale can be shown as in Table 2.1 below.

Table 2.1: Mean Opinion Scale (MOS) Rate Mapping to Quality Standard

Rate	Quality
1	Bad
2	Poor
3	Fair
4	Good
5	Excellent

Subjective QoE evaluations are time consuming and are done manually. It does not allow automatic quality ratings (Zinner et al., 2010) and thus, defeats the purpose of developing an intelligent system to be adaptive towards the quality of the video. Objective evaluations on the other hand, can be done real-time and automatically. This is important for an intelligent system so that the network parameters can be adapted in real-time in response to the current QoE of a video.

In this thesis, one of the metrics that is being evaluated is the **Peak Signal to Noise Ratio (PSNR).** PSNR is a type of objective evaluation that represents the ratio between the maximum value of a signal and the power of the corrupting noise that affects the said signal. In terms of video quality evaluation, PSNR value is the difference between the original and distorted video.

It can be defined mathematically from Mean Squared Error (MSE) of two images in a matrix form. MSE is defined as:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

(2.1)

Where

m x n : image size

I : original image

K : distorted image

Meanwhile PSNR is expressed in decibels (dB), and can defined as:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

(2.2)

Where:

MAX_I : Maximum possible absolute value of an image

In this thesis, PSNR is the main metric that is being used to evaluate the quality of the video. This is because PSNR measures the absolute difference between videos and importantly, there are studies that has shown to map between PSNR (objective evaluation) and Mean Opinion

Score (subjective evaluation) metric. This is important as later in this thesis, the proposed algorithm will predict the level of the perceived video quality at the receiver by monitoring PSNR. A video is of good quality if the value is at least 30 dB shown in Table 2.2 below.

Table 2.2: PSNR to MOS Conversion (Padle and Mendre, 2014)

PSNR (dB)	MOS
>37	5 (Excellent)
31-37	4 (Good)
25-31	3 (Fair)
20-25	2 (Poor)
<20	1 (Bad)

ITU-T issued a standard method to assess video quality (ITU-T, 2016). Focusing exclusively on video quality evaluation, three assessment guides has been recommended. The methods are as below:

Absolute Category Rating (ACR): It is also known as single stimulus[1] method as test stimuli are presented one at a time. The stimuli are then rated independently by the subjects (without any reference[2] to compare) on a category scale, which is similar to the MOS scale.

Degradation Category Rating (DCR): Also known as Double Stimulus Method (DMS). This is because the stimulus is presented in pairs, where the first stimuli is always the reference. The second stimulus is the one under test. Subjects are then required to rate the second stimuli in relation to the reference stimuli on a five-level scale, which is also similar to MOS.

Comparison Category Rating (CCR): Like DCR, CCR stimuli are also represented in pairs. However, the reference stimulus is presented in a randomized order where the reference stimuli is presented first 50% of the time and presented as the second stimuli on the other time. It can be used to compare between a reference and a processed video, or to compare two impaired

[1] Sample to be rated. It can be a reference, or processed sample.
[2] A reference sample, which is used by subjects to compare a stimulus as a yardstick.

videos. While ACR and DCR are rated through a 5-level scale MOS, CCR uses seven where the worst scale is at -3 (much worse) to +3 (much better).

In this thesis, both objective and subjective evaluation are done. From the subjective perspective, DCR method is used for video quality evaluation as the subjects are required to compare stimuli that are untreated and treated.

2.4 IEEE 802.11 Standards on QoS

Throughout the years, a lot of effort had been made to enhance the performance of the legacy IEEE 802.11 network. In spite of its benefits, the original IEEE 802.11 standard specification did include a number of inherent challenges. A number of subsequent studies by IEEE themselves focused on upgrading of bandwidth capabilities of the wireless media from IEEE 802.11a (IEEE Std 802.11a - 1999 (R2003), 2003), 802.11b (IEEE Std 802.11b - 1999 (R2003), 2003), 802.11g (IEEE Std 802.11g - 2003, 2003) and the MIMO 802.11n (IEEE Std 802.11n - 2009, 2009) through various techniques which involve both the contention based component (Distributed Coordination Function, DCF) and the non-contention based component (Point Coordination Function, PCF). However a specific amendment to address the QoS issue in WLANs, the IEEE 802.11e (IEEE Std. 802.11e-2005, 2005) had been published by IEEE as well as IEEE 802.11aa (IEEE Std. 802.11aa - 2012, 2012) for multimedia content in multicast transmission.

2.4.1 Challenges on Streaming Video in Wireless Network

Wireless network in its nature is half-duplex which means transmission and reception of traffic cannot be done at the same time. This already affects the performance of it compared to its wired counterpart. Besides that, there are also a few factors that make transmission especially video traffic in wireless network more challenging.

Bit Error Rates: Wireless network depends on radio frequency to carry information. Radio frequency in nature is susceptible to reflection, refraction, diffraction, scattering, and adsorption. These leads to packet loss that will degrade the quality of the video transmitted. Since wireless network (wifi) is an unlicensed band, the wireless signal is also prone to interference from nearby signals. It is not a matter of whether packets are loss, but how to manage packet loss that leads to bit error rates.

Variable bandwidth across distance: One of the characteristics of a radio signal is that it attenuates over distance. This resulted to variability of speed and bandwidth of the channel. In an open space, a station which is nearer to an Access Point has better connection speed than stations which are farther. This means in a scenario where stations are not stationary, it is impossible to attain constant speed to transmit video traffic.

Transmission delay: Signal attenuation, interference as well as multi-path fading contributes to delays and variation of delays which is known as jitter. Since video streaming is time sensitive, the timing of packet arrival at the receiver must be consistent. Therefore, packet delay and jitter will introduce jerkiness towards the quality of the video and thus affects the user experience.

Realising the challenges, IEEE released an amendment to address these issues to prioritize multimedia traffic that is time sensitive that supports Quality of Service (QoE) – IEEE 802.11e.

2.4.2 IEEE 802.11e

The IEEE 802.11e amendment aims to provide QoS especially for multimedia traffic by giving a certain amount of priority for specific hosts. The transmission priority of a host is determined through the type of traffic it carries. HCF starts with a beacon frame and comprises two main phases. They are the Contention Period (CP) and Contention Free Period (CFP).

In CP, high priority stations are assigned low waiting time before being allowed to use the channel. This mechanism is called the Enhance Distributed Channel Access (EDCA). Meanwhile in CFP, the mechanism reserves uncontended amount of bandwidth for multimedia traffic. This mechanism is called the HCF Controlled Channel Access (HCCA). These two components made up the Hybrid Coordination Function (HCF) which only operates in QoS network configurations. While EDCA is used only during contention period, HCCA can be used in both contention and contention free period. HCCA and EDCA will be discussed later in Section 2.5.1 and 2.5.2 respectively.

HCF, which is also known as a superframe, is made up of beacon frame, EDCA (the contention period) and HCCA (the contention free period) as shown below in Figure 2.3.

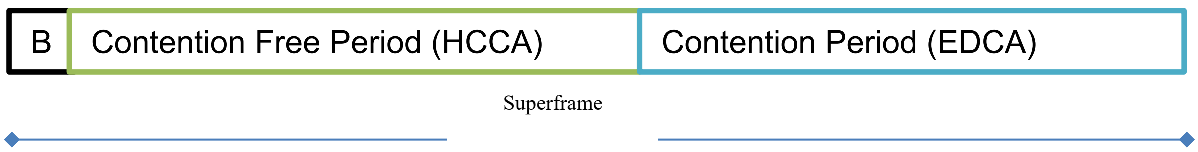


Figure 2.3: The structure of Superframe

The beacon frame signals the start of the EDCA-HCCA cycle while the total period between a beacon frame and the next beacon frame cycle is the beacon interval; the duration of HCCA and EDCA are not specified in the IEEE 802.11e standard.

Within the QoS enabled wireless network environment, the Access Point (AP), Stations and Basic Service Sets (BSS) that operate under the IEEE 802.11e mode are referred as QoS Access Point (QAP), QoS Stations (QSTA) and QoS BSS (QBSS) respectively. It is also worth to note that CP and EDCA can be used interchangeably while CFP and HCCA can be used interchangeably as well.

2.4.2.1 HCF Controlled Channel Access (HCCA)

The HCCA scheduler is a centralized access medium controlled by the Hybrid controller (HC) which resides in the QAP. It acts as a bandwidth management controller. Instead of contenting for the bandwidth, QSTAs wait to be polled by the HC to access the medium in HCCA (infrastructure mode). The HC maintains a list of QSTA to be polled in the CFP.

If a QSTA with data to transmit is being polled, the QAP will evaluate whether to allow the QSTA to use the bandwidth. If a QSTA is granted permission to transmit, the QAP will give a maximum duration of how long it can do the transmission. The duration of the transmission allowed is called the transmit opportunity (TXOP). Each QSTA that is being polled will be allocated an amount of TXOP for it to transfer data as much as it can until the TXOP expires. The period of a station being polled on the first round to the second round is the Service Interval (SI). The concept of SI can be depicted as in Figure 2.4 below.

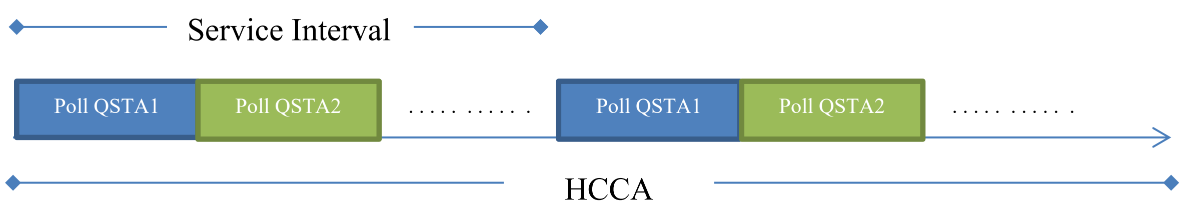


Figure 2.4: Service Interval in an HCCA period (CFP)

HC can take control whenever it needs to allocate TXOP to QSTA. Although there is a dedicated period in the superframe for CFP medium access, HC can still activate CFP in the EDCA period, if necessary, as it has the highest priority amongst all stations. The CFP that is activated in the CP is referred to as the Controlled Access Phase (CAP). HC is not allowed to interrupt ongoing transmission, but can take control of the medium to enable CAP whenever it detects that the medium is idle for certain period in CP.

Assuming a $QSTA_i$ has traffic flow to transmit, it will send a TXOP request packet to the HC. The TXOP request (also known as add traffic stream request, ADD-TS-Request) packet contains information regarding the requirements of the traffic that the QSTA carries. This information is known as Transmission Specification (TSPEC). The TSPEC parameters are shown as in Table 2.3 below.

Table 2.3: TSPEC parameters in HCCA (Ruscelli et al., 2012)

TSPEC parameters (units)	Symbol
Mean data rate (b/s)	R
Nominal SDU size (B)	L
Minimum PHY rate (b/s)	Γ
Delay bound (s)	D
Maximum Service Interval (s)	MSI
Other parameters (units)	**Symbol**
Peak data rate (b/s)	Π
Peak frame rate (frames/s)	Φ
Burstiness factor	B
Frame size (b)	L
Frame Interarrival time (s)	τ
Interarrival time upper tolerance (s)	δ_u
Interarrival time lower tolerance (s)	δ_l
Service Interval (s)	SI
Transmission Opportunity interval (s)	$TXOP$
Number of transmitted SDU	N
Amount of transmitted data (b)	χ

These parameters will then be calculated by the HC through a series of equations to determine the appropriate TXOP amount for the particular QSTA.

Upon receiving the ADD-TS-Request, the HC examines the content of the packet. The HC then calculates the SI for the network to accommodate both the new QSTA and the existing QSTA (if the HCCA is already running with active QSTA). Maximum SI (MSI) is the maximum waiting time a STA can wait before the next poll (polling interval) and is provided by each QSTA in TSPEC to the HC. The MSI to be used in the HCCA period must be a number that satisfies conditions as below:

- Minimum of the maximum SI of all QSTA
- A number submultiple to the beacon interval

The next step for the HC is to calculate the number of the MAC Service Data Unit (MSDU) of the flow of the requesting $QSTA_i$. We name this flow as $flow_i$. This can be done through (2.3) below:

$$N_i = \left\lceil \frac{SI \times \rho_i}{L_i} \right\rceil \tag{2.3}$$

Where:

N_i	:	Number of MSDU of $flow_i$ that arrives at the mean data rate during an SI
SI	:	Service Interval
ρ_i	:	Mean Data Rate of flow i
L_i	:	Nominal MSDU size

The value of N_i will then be used to calculate the appropriate TXOP for the $flow_i$. To calculate the TXOP for $flow_i$, $TXOP_i$, where ((2.4) will be used as shown below:

$$TXOP_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O\right) \tag{2.4}$$

Where:

O	:	Overhead of the packets
R_{data}	:	Physical data rates

However, the requesting QSTA will not be given the TXOP straightaway. There is an admission control decision on whether the QSTA will be included into the list of QSTA transmitting in the CFP. The HC must determine whether there is enough available bandwidth to accommodate the requesting QSTA. The admission control can be done through a test using (2.5) below.

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=1}^{k} \frac{TXOP_i}{SI} \leq \frac{T - T_{CP}}{T} \tag{2.5}$$

Where:

$TXOP_{k+1}$	:	The TXOP of the requesting QSTA
k	:	Number of existing streams
T	:	The duration of the Superframe (Beacon Interval)
T_{CP}	:	Duration of Contention Period

From this test, the HC will determine whether to grant the TXOP to the requesting QSTA. If the equation is satisfied, the HC admits flow k+1 into its polling list and allocates TXOP to the flow based on the HCCA scheduling scheme. The HC will then reply to the requesting QSTA, containing the information regarding the acceptance of the request to be added into the polling list.

2.4.2.1.1 Shortcomings of HCCA

Although IEEE 802.11e brings a new dimension of providing a QoS enabled wireless network, encompassing better services to vary multimedia traffic, it also includes several drawbacks. Firstly, IEEE 802.11e is designed to protect and carry Constant Bit Rate (CBR) traffic. This is apparent from the calculation and assignment of the TXOP to the traffic streams, as the TXOP is calculated based on the average value of the traffic requirement parameters. Given TXOP is fixed, it cannot take fluctuations of data rates and packet sizes into account. Similar to multimedia transmissions, the traffic pattern varies whereby the data rates, packet sizes as well as several other parameters will be fluctuating. Numerous studies highlighted that, although HCCA demonstrates good performance for CBR, it performs poorly in the case of multimedia traffic (Ju and Chung, 2013).

Secondly, there is no mechanism for retransmission provided through this amendment. It is assumed that the channel is error free and that, all the duration of time assigned to the stations are practically to transmit the packets successfully without a single consideration of packet drop. This situation is not ideal especially when it comes to real world network traffic, particularly in wireless environments. Therefore, for an actual network implementation with interference and packet loss, HCCA does not handle it very well and cannot utilize the bandwidth efficiently (Kar et al., 2012).

Finally, several studies showed that the calculation for TXOP allocation in the IEEE 802.11e amendment is not precise (Siris & Courcoubetis, 2006), (Noh et al., 2010). More specifically, the equation to estimate the number of packets for each SI is inaccurate. This leads to an insufficient amount of TXOP being allocated to QSTA which ends with packets being dropped.

2.4.2.2 Enhanced Distributed Channel Access (EDCA)

DCF, which is the underlying mechanism in the legacy IEEE 802.11 network and EDCA are both contention-based medium access methods. However, while both DCF and EDCA are contention based, EDCA introduced four new enhancements from DCF:

- Multiple Access Categories (AC)
- Multiple AIFS
- Multiple independent backoff (Contention Window)
- EDCA Transmit Opportunity (EDCA-TXOP).

Each new enhancement will be discussed in the next paragraphs.

MULTIPLE ACCESS CATEGORIES: EDCA introduced four different Access Categories (AC 0 to AC 3) that determine the priority of the traffic. The priorities are differentiated based on the traffic types:

- AC 0 is reserved for the highest traffic type which is the **voice** stream. This Access Category is also known as **AC_VO**.
- AC 1 is reserved for the second highest priority which is **video**. This Access Category is also named as **AC_VI**.
- AC 2 is the second lowest priority. Also known as **AC_BE**, it is dedicated for **best effort** stream.
- AC 3 is the lowest priority Access Category which is dedicated for **background** traffic. It is also named as the **AC_BK**.

An architectural view of the four queues is presented below in Figure 2.5.

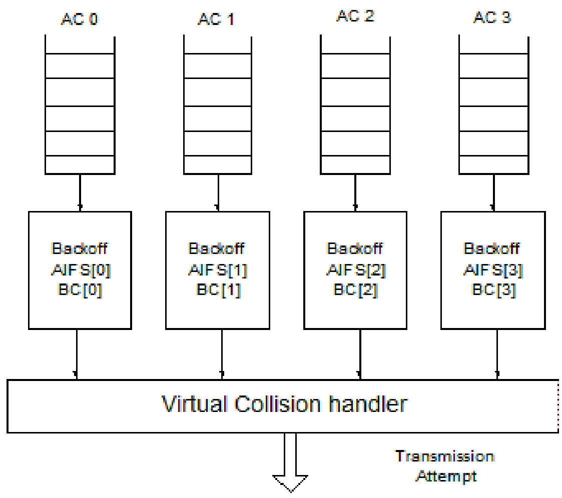


Figure 2.5: Different queues in a single STA

Since there are four different queues in a single QSTA, each AC competes to access the wireless media and there is a high possibility that the backoff counter of the queues may reach zero at the same time. This will lead to internal collision where queues within the same STA collide. To address this issue, IEEE 802.11e incorporated a Virtual Collision Handler (as in Figure 2.5): when internal collision happens, the Virtual Collision Handler (VCH) will compare the colliding queues and give way to the higher priority traffic to access the wireless medium while lower priority traffic will backoff and double the CW value before contending the wireless medium again.

MULTIPLE AIFS: In IEEE 802.11e, a new type of Interframe Space (IFS) is introduced, named the Arbitrary IFS (AIFS). It is equivalent to DIFS in DCF. The value of AIFS varies according to the priority of the ACs. With the priority value, each AC will have different waiting time.

The highest priority, AC_VO will have the shortest waiting time while the lowest priority, AC_BK will have the longest waiting time. The mechanism can be shown in Figure 2.6 below.

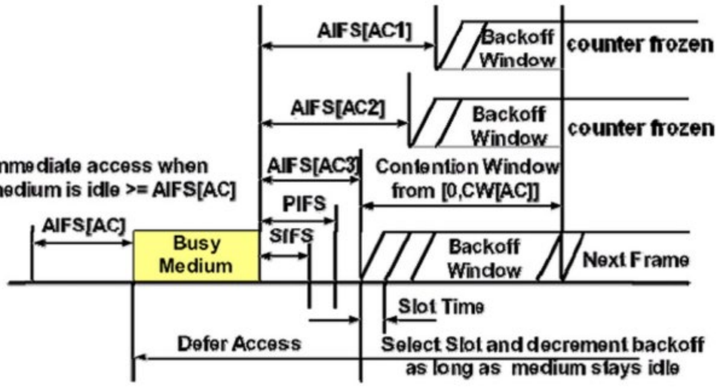


Figure 2.6: Different AIFS for each Access Category (Sadek et al., 2015)

The value of AIFS is configurable and differs among IEEE 802.11a, 802.11b, 802.11g and 802.11n. In IEEE 802.11g and 802.11n, the value of AIFS can differ for backward compatibility with other 2.4GHz based wireless networks. Default values for QSTA are defined in IEEE 802.11e amendment section 7.3.2.27(IEEE Std. 802.11e-2005, 2005). The summary of the default values can be shown as in Table 2.4 below.

Table 2.4: Summary of the values of AIFS among different Access Categories

AC	AIFSN	802.11b AIFS[AC]	802.11g AIFS[AC]	802.11a AIFS[AC]	802.11n 2.4GHz AIFS[AC]	802.11n 5GHz AIFS[AC]
SIFS Time	---	10µs	10µs	16µs	10µs	16µs
Slot Time	---	20µs	Long = 20µs Short = 9µs	9µs	Long = 20µs Short = 9µs	9µs
AC_VO	2	50µs	Long = 50µs Short = 28µs	34µs	Long = 50µs Short = 28µs	34µs
AC_VI	2	50µs	Long = 50µs Short = 28µs	34µs	Long = 50µs Short = 28µs	34µs
AC_BE	3	70µs	Long = 70µs Short = 37µs	43µs	Long = 70µs Short = 37µs	43µs
AC_BK	7	150µs	Long = 150µs Short = 73µs	79µs	Long = 150µs Short = 73µs	79µs

MULTIPLE Contention Window Range: Under the same intention as in multiple AIFS, multiple Contention Window (CW) range also aims to give shorter waiting time to high priority traffic.

After sensing the medium is idle for AIFS, each AC will start their own backoff counter. By having a shorter range of CW, high priority AC are more likely to transmit ahead of the lower priority ACs.

EDCA-TXOP: TXOP is the time interval during which QSTA has the right to deliver packets. TXOP controls the station channel utilization by maintaining that the QSTA that was given the opportunity to transmit must not utilize radio resources for duration longer than the specified limit. During the TXOP, the transmitting QSTA can send as many packets as possible.

Comparing the IEEE 802.11e EDCA with the legacy IEEE 802.11 DCF, there are four major attribute differences:

Firstly, DCF has only one access category where it queues all the packets regardless the types and priority of them. In EDCA, there are four main categories, and the packets are queued based on these four priorities: Voice, Video, Best Effort and Background traffic.

Secondly in DCF, a station is only permitted to send 1 single MSDU when it has the opportunity to transmit. In EDCA, TXOP was introduced where if the TXOP is not expired, the station that was given permission to transmit can transmit as many packets as it can.

Thirdly, all traffic in DCF will have to wait the same amount of time, which is DIFS after sensing that the medium is idle before starting the CW countdown. In EDCA, different priority will have different waiting time which is also known as the AIFS.

Finally, the CW range for DCF is the same for all station where the range is [0-1023]. Meanwhile EDCA introduced different range of CW based on the Access Categories. A voice traffic will have the range of [3,7] while video traffic will have the range of [7-15]. Both Best Effort and Background traffic will have the CW range of [15-1023].

The four enhancements in IEEE 802.11e lead to a QSTA having different waiting time to access the medium shown in Figure 2.7 below.

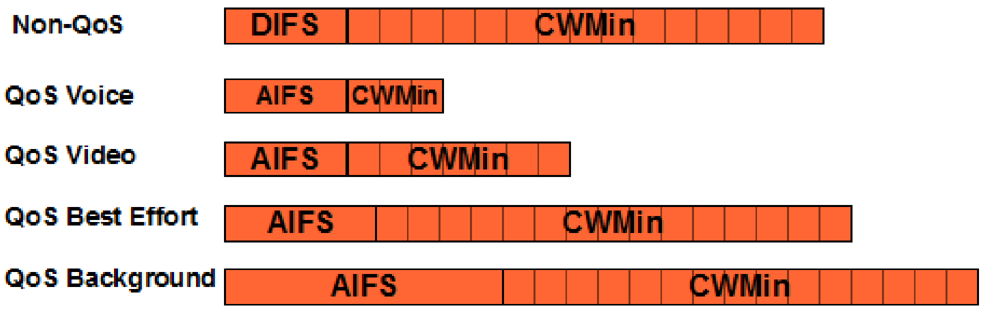


Figure 2.7: Different waiting time for different Access Category in DCF and EDCA

While the highest AC have the shortest waiting time before accessing the medium, the lowest priority AC will have to wait longer before accessing the wireless medium.

2.4.2.2.1 Shortcomings of EDCA

Virtual Collision: The basis of service differentiation in EDCA is that each traffic type will have a different waiting time before it can access the wireless media. Four access categories (AC) were introduced in every EDCA QSTA where every AC has their own queue for the packets to be stacked before being released to the media. What happens now is that every queue will now compete for the media. This will lead to virtual (or sometimes known as internal) collision which can further increase the waiting time of the streams involved.

Lack of VBR support: Though the IEEE 802.11e claimed that the new amendment supports QoS, it does not preserve priority differentiation for Variable Bit Rate (VBR) traffic which is the traffic pattern of video stream in nature. The use of static parameter cannot adapt to fluctuating (variable) network conditions and therefore not appropriate for sudden packet loss or increase.

Fixed or Non-Adaptive TXOP: If transmission of a video frame is longer than the TXOP limit, sending a long burst by splitting it among multiple TXOP can cause unwanted delay and jitter. This is important especially if the frame type is the I-frame, which has the largest frame size and is the most crucial in a Group of Pictures (GoP) of a video stream.

2.4.2.3 Conclusions on HCCA and EDCA

Given their features, HCCA is the better choice to transmit QoS flows (Mangold et al., 2003) based on several factors compared to EDCA. This is because EDCA cannot guarantee any throughput or delay bounds, but only performance differentiation among the access categories (Kosek-Szott et al., 2013). Moreover, the nature of HCCA is to protect the bandwidth and allocate a finite amount of them to a certain network flow.

However, the algorithm for scheduling polling times and periods in HCCA is derived from different applications and it involves high cost and complexity for practical implementation. These include the beacon frames and other control frames that work in the HCCA mechanism. On top of that, most of commercial WLAN devices use EDCA and that, Wi-Fi Multimedia (WMM) only supports prioritized media access according to EDCA (Kuo, 2008). Therefore, the rest of this report will be focusing on EDCA.

2.4.3 IEEE 802.11aa

Supporting multimedia traffic in wireless networks has become more crucial, requiring the release of a new amendment to cater for these types of traffic.

In 2012, the IEEE 802.11aa amendment was published to address QoS issues in wireless network from different angles. Four of the issues to be handled that had been identified by this amendment are multicasts, refining EDCA channel granularity, overlapping BSS and compatibility with the audio video bridge. These issues will be discussed in the next subsections.

2.4.3.1 Issue 1: Multicast Mechanism in IEEE 802.11

The multicast mechanism provided by the legacy IEEE 802.11 is unreliable and cannot offer the necessary QoS. Theoretically, multicast has its own advantage on reducing the usage of the bandwidth capacity by avoiding unnecessary flow replication. However, implementing multicast transmission in a wireless environment poses a totally different challenge, due to the nature of the wireless channel itself that is prone to interference.

Receiving acknowledge (ACK) packets from all receivers is inefficient due to several factors such as large overhead, scheduling, and synchronization issues. Therefore, the MAC layer itself specifies that multicast must be transmitted using basic access scheme where a Request to Send/ Clear to Send (RTS/CTS) mechanism is not allowed. Moreover, the legacy IEEE 802.11 also specifies that multicast must be transmitted using the lowest bit rate supported by all stations. A station with low bit rate slows down the entire network.

Several enhancements have been suggested in the IEEE 802.11aa amendment. These include introducing IEEE 802.11v Directed Multicast Service (DMS), Groupcast with Retries - with Unsolicited Retries (GCR-UR) and GCR Block ACK. These enhancements will be discussed in the next paragraphs.

DMS converts multicast streams to unicast frames directed to each group. While this method is most reliable, it has a big overhead and does not scale well. Moreover, the overhead greatly increases with retransmissions due to lost frames and therefore is more suited with small number of receivers (de la Oliva et al., 2013).

GCR-UR enforces the retransmission of the multicast frames (without the request of receiver and not because of unacknowledged packet) to be repeated a number of times to increase the probability of the frame reach the receiver. This is done by the AP on purpose, without receiving any retry request (Shin et al., 2013). However, the number of repeated transmission is not specified by the amendment and depends on implementation (Maraslis et al., 2012). The performance of this method depends on the number of retries, not the number of receivers. Though it does not introduce significant overheads, the main weakness is it requires proper tuning on the number of retires to guarantee an optimal performance. While insufficient number of retries might end up receivers do not receive the packet, too much retries will introduce a lot of unnecessary packets in the network which may saturate the bandwidth.

GCR-Block ACK extends the Block ACK in the IEEE 802.11n amendment (IEEE Std 802.11n - 2009, 2009). In this method, the AP sends a number of multicasts frames and asks one or more stations to ACK the transmitted frames. Frames not received by one or more STA can be retransmitted. AP needs to have a copy of all frames if they are not acknowledged (AP keeps track of the packets that are not acknowledged). However, similar to DMS the selection of the station to acknowledge on behalf of other stations is left for implementation. On the other hand, this method can apply a lot of the leader-based method and algorithms.

The summary of DMS, GCR-UR and GCR-Block ACK can be depicted as in Figure 2.8 below.

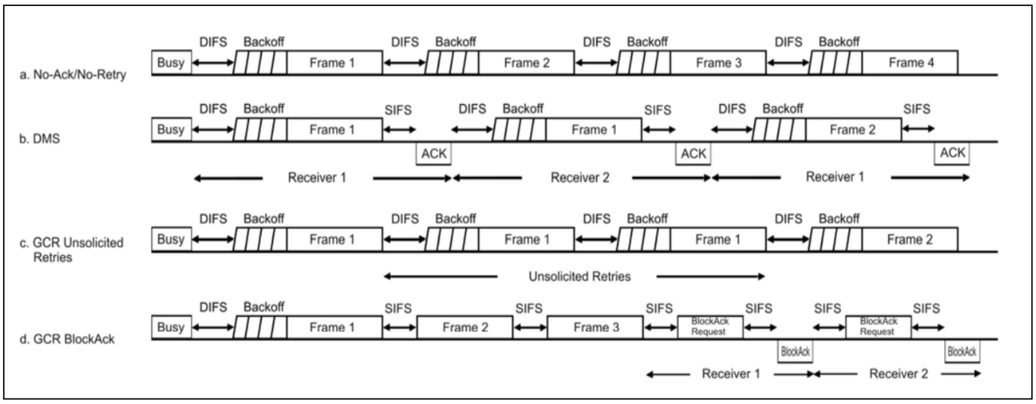


Figure 2.8: Multicast mechanisms in the IEEE 802.11aa (Maraslis et al., 2012)

All the above methods can be grouped under the Group Addressed Transmission Service (GATS) introduced by the IEEE 802.11aa.

2.4.3.2 Issue 2: Refining EDCA Channel Granularity

The IEEE 802.11e EDCA introduced different Access Categories (AC) to handle different types of traffic. However, there are no mechanisms to differentiate flows within the same categories. This is important specially to prioritize real time versus non-real time stream. Moreover, smaller Contention Window (CW) for video AC means higher collision probability (Maraslis et al., 2012). This will lead to retransmission and throughput degradation.

IEEE 802.11aa addresses the prioritization issue by increasing the EDCA granularity. This can be done by introducing Intra-access category prioritization (Charfi et al., 2013) by specifying alternate queues for the two highest priorities AC, which are the video and voice AC. Drop Eligibility Indicator (DEI) is also introduced to handle the collision probability. When DEI is activated, the stream will have smaller maximum number of retransmissions for long and short retry counter. Therefore, if there is bandwidth shortage, stream with DEI activated will likely reach the maximum number of retransmission and discarded. DEI can be set in the TSPEC.

2.4.3.3 Issue 3: Overlapping BSS in Infrastructure Mode

Overlapping Wireless LANs (BSS) create will have a high probability of creating QoS problems especially the channels themselves are overlapping. This leads to traffic competing between networks which reduce the bandwidth. It can be worse and more severe when QoE is enabled. This is because the QBSS cannot support performance demands for multimedia streams because of the interference of other QBSS using the same frequency bandwidth and channel. HCCA may solve the problem for a network by protecting the bandwidth of its own network. However, if two or more QBSSs are using HCCA on the same channel and bandwidth, it needs to obey other AP's TXOP.

To address this issue, the IEEE 802.11aa introduced a decentralized mechanism for neighbouring AP to exchange information about the QoS depended on traffic load in each QBSS. This information can be used for more efficient channel selection because it enables QBSS to share the channel while enabling APs to cooperate and work as a bigger QoS-aware network sharing the wireless medium fairly. To exchange information between APs, QoS Load (QLoad) is introduced. It is the information used by an AP to inform its neighbouring AP about its QoS traffic load. E.g.: number of QBSS it overlaps, QoS traffic load for itself and the QoS traffic load for the QBSS it overlaps. IEEE 802.11aa defined a set of control frames that can be used by AP using HCCA function to inform the neighbouring AP of their TXOP allocations. Therefore, other AP can use this information to schedule their TXOP and to ensure enough bandwidth for new stream admission.

2.4.3.4 Issue 4: Compatibility with the Audio Video Bridge

To be able to accommodate multimedia traffic, a network must be compatible with the IEEE 802.1 audio video bridge (AVB). However, this is not the case with the legacy IEEE 802.11. The IEEE 802.1 audio video bridge aims to provide the specifications that will allow time-synchronized low latency streaming services through networks (IEEE Std 802.1q - 2011 (Revision of IEEE Std 802.1q-2005), 2011). This standard consists of the following components:

- IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications (gPTP),
- IEEE 802.1Qat: Stream Reservation Protocol (SRP)
- IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams (FQTSS)
- IEEE 802.1BA: Audio Video Bridging Systems

IEEE 802.11aa is required to work closely with the AVB task group so that it will be compatible with Stream Reservation Protocol (SRP). It does not require major changes but in need of some addition the control frames to allow AP to perform reservation procedures.

2.5 Previous Research

In this section, an overview of the previous works is provided to study the current solution to provide QoE especially in wireless transmissions. From the studies, the weaknesses of the available infrastructure are identified whilst recent works by previous research are also being studied as a platform to build up the thesis.

The emerging of both video streaming demand has opened challenges on how to transport them across wireless medium. It is very crucial for time-sensitive data to arrive at the receiver with very minimal delay for good Quality of Experience. A lot of previous work has been done to overcome the shortcomings of the IEEE 802.11e. From the perspective of HCF Controlled Channel Access (HCCA), previous research can be categorized into two different categories. The first one involves modifying the calculation of Transmission Opportunity (TXOP) assigned to a station. This includes considering fluctuations of data rates, types of packets and packet sizes. Besides that, a retransmission mechanism has also been considered and introduced for transmission suffering packet drops and packet errors. This is crucial especially transmission of multimedia traffic in noisy channels. Second category is the enhancement of the HCCA polling mechanism itself. While a polling mechanism is capable to protect a certain amount of bandwidth for a multimedia station, it is also apparent that it introduces significant control and management frames, which may saturate the network. This includes the beacon and polling frame that may, at one stage, compete with the multimedia traffic to withhold the wireless channel.

On the other hand, significant effort has also been put into enhancing the efficiency of Enhanced Distributed Channel Access (EDCA). Contention based wireless network has always been the preferred setup due to the nature of ad-hoc wireless networks which are easier to setup and less complex to implement. It is also practical in real environment as well as in simulations. Efforts to enhance EDCA involve making the parameters in the scheduling to be adaptive and accommodate Variable Bit Rate (VBR) video, reducing packet drops by manipulating the four different AC queues in the QoS Stations (QSTA), as well as allocating extra bandwidth or TXOP for real time traffic.

The next subsections will discuss previous works based on both perspectives mentioned from the previous paragraph regarding HCCA before the related works regarding EDCA is discussed.

2.5.1 HCCA Enhancements to Support QoS/ QoE

In (Kim et al., 2006), the authors aimed to determine the optimal TXOP duration for VBR traffic by assigning different SI values for high priority traffic. The idea is to try to meet the delay bound of the packet of each station in the HCCA polling list. By doing this, high priority stations are allowed to transmit with the duration beyond of its own TXOP as long as the low priority packets are waiting before their delay bound expires. Through experiments, it is proved by the authors that the packet loss of this proposed method is very low compared to the IEEE 802.11e standard scheme.

The idea to extend TXOP was also followed by (Jansang and Phonphoem, 2011) where they proposed a new dynamic TXOP assignment for video stations, called Adjustable TXOP Mechanism for Video (ATMV), with two different levels (ATMV1 and ATMV2) to support different type of videos, based on feedback information. The experiment results showed that videos can be well supported and carried through ATMV1 (slow movement video) and ATMV2 (gentle and rapid movement video).

Another idea to extend the TXOP is by reclaiming unspent or unused TXOP in the case of a VBR traffic. This was highlighted by (Cecchetti et al., 2012) by proposing Immediate Dynamic TXOP HCCA (IDTH) bandwidth reclaiming. Since the IEEE 802.11e standard is unable to efficiently manage VBR due to the same period polling and constant TXOP duration, unspent TXOP is lost when the data rate drops because the assigned TXOP is now smaller than the one assigned at the beginning of the polling process. IDTH recovers the portion of the lost TXOP to further provide capacity for the next VBR stream. This improves the resource assignment by 75% of waiting packet and reduced the mean delay and the time packet remains in queue length.

Meanwhile, (Bin Muhamad Noh et al., 2009; Bin Muhamad Noh et al., 2010) suggested on using a different method to calculate the MSDU to be sent by a station. It is claimed that the standard IEEE 802.11e miscalculate the number of MSDUs and thus results to a wrong calculation of TXOP. The new technique calculates the MSDU using the inter-arrival time which is referred as inter_MU. Besides that, additional TXOP can be allocated in an SI if the HC received corrupted MAC in the previous SI. It is evident that the performance of the proposed technique is better than the standard IEEE 802.11e during transmissions in a noisy environment. In another work, (Bin Muhamad Noh et al., 2010) looked into a cross-layer approach and proposed additional TXOP based on the queue length of the stations. A station with high queue length due to the burstiness of the VBR traffic will be assigned additional TXOP as long as the total SI is not occupied. Besides that, the authors also proposed video frame skipping which is important especially while transmitting video frames. If the station fails to send the I-frames of a video stream, all the subsequent P and B frames are all dropped. Through simulation, the performance of the proposed technique outperforms the standard IEEE 802.11e.

Some researchers addressed the challenge to prioritize video traffic by integrating EDCA and HCCA. For example, (Ruscelli et al., 2012) claimed that very few works have considered the possibility to integrate the service provided by HCCA with the resources available in EDCA. Their work proposed a new technique which allows the network node to use both HCCA and EDCA to transmit the same traffic stream. This new approach, called Overboost uses HCCA to negotiate a minimum bandwidth that deals with traffic stream that require more bandwidth than the negotiated one and redirects the excess bandwidth to EDCA. Overboost reacts to bursty traffic (VBR) where the efficiency, throughput, queue length and delay analysis shows significant improvements. However, CBR traffic e.g., VoIP are not affected by Overboost.

From another perspective, (Ali et al., 2013) claimed that in most cases, papers are trying to provide better performance for video traffic but at the same time, neglect and overlook the reality that the network consists of a mixture of different traffic types. In practical networks, the real-time traffic is mixed with non-real-time traffic. Therefore, the authors proposed a new algorithm, Selectivity Function Scheduler (SFS) with the ability to accommodate multiple streams with different levels of QoS requirements concurrently running on the same station. SFS incorporate several functionalities including accommodating different type of traffic (CBR and VBR) and provide diverse QoS assurance, consider channel conditions and QoS requirements of different traffic and taking the fairness of traffic to balance tradeoff between maximizing throughput for users without starving other users into account. An analytical evaluation has been conducted and the results showed that SFS outperforms the standard IEEE 802.11e with higher throughput and lower packet drop in VBR traffic. However, the average delay for voice and video for SFS is high but is still below delay bound.

In another experiment with mixed traffic types, (Ju and Chung, 2013) proposed a variable TXOP for VBR traffic in HCCA by introducing a variability factor. Variable length of TXOP is assigned to different types of traffic based on numbers of packets to be transmitted. Through a series of simulations, the authors show that the proposed technique outperforms the standard IEEE 802.11e because it allocates dynamic TXOP while the IEEE 802.11e maintains a fixed order of traffic stream. It is also evident that the enhancement of performance is contributed by the fact that the proposed scheme considers VBR traffic pattern and allocates optimal TXOP time for the stations, besides improving the network bandwidth utilization.

Significant effort has also been put towards improving the polling mechanism in HCCA. The main idea is to try to eliminate as much management overhead as possible to accommodate more high priority packets into the channel. (Murray et al., 2009) estimated that ACK packets in IEEE 802.11 themselves consume as much as 22% of the bandwidth. However, proposing a new polling mechanism can be tricky because eliminating management and control frames will also lead to desynchronization of stations from HC and will affect the establishment of Network Allocation Vector (NAV) period and error control mechanism.

(Viegas et al., 2013) proposed a new polling technique which introduces the Virtual Token Passing (VTP) to eliminate the traditional scheduling algorithm. They introduced Group Sequential Communication (GSC) based on a Publish-Subscribe approach. This technique aims to reduce the protocol in the standard IEEE 802.11e amendment and propose an error recovery mechanism based on Block ACK. The research proved that management frames, such as beacon, null and ACK frames, contribute to large utilization of network bandwidth. To elaborate, GSC starts when the HC sends a Beacon frame which sets the NAV of all STA. At his point, all the STA will just be in the standby mode and listen to the network. Then the HC grants the medium access to a group of real time (RT) stations, named the GSC stations (GSC STA). The first GSC STA will hold the VTP and transmits packets if the STA ID is the same as the token number. If a GSC STA has nothing to transmit, it will just pass the token to the next GSC STA by increasing the assigned number on the token. At the end of the transmission, verification is done by the HC using Block ACK. If there are any error messages, another procedure, called Second Chance, is activated. The retransmission method will be the same of the previous round of transmission except it only involves GSC STA with packet errors only. Through simulations, it is evident that GSC reduces the utilization of bandwidth by eliminating the management frames. However, this simulation is done using CBR traffic only.

2.5.2 EDCA Enhancements to Support QoS/ QoE

Early research papers have been comparing the performance of Distributed Channel Access (DCF) and EDCA on carrying multimedia traffic through the wireless media. This is because of the introduction of four different Access Categories (AC)s in EDCA that leads to media contention within the QSTA itself. Abu-khadrah et al., (2014) for example, compared DCF to EDCA and design network scenarios using OPNET simulator and compares the QoS values.

Meanwhile, Anitha & Jayakumari (2014) also compared the performance of DCF and EDCA under different network load conditions, with some adjustment of the EDCA media access parameters. The performances of WLAN under variable number of nodes are analyzed using adjustable parameters of EDCA which are the TXOP limit, Contention Window (CW) values which involves CW_{min} and CW_{max}, and Arbitrary Interframe Space (AIFS). By varying the number of nodes between 10 and 50, the work concluded that EDCA leads to better results when compared to DCF. Additionally, throughput analysis suggested that TXOP adjustments are best suited for high number of nodes (between 45 to 50 nodes). Minimum CW values are best suited for minimum number of nodes, while high CW values are best suited for medium number of nodes. On the other hand, AIFS values gave fair results for all types of networks. Delay analysis found that increase in TXOP limit is best suited for small, medium and large number of nodes. Therefore, increasing the TXOP limit give a very low drop and delay compared to DCF and EDCA. The increase in TXOP limit gives high performance improvement than the other access method parameters. The work suggested that to improve throughput, rate adaptation scheme can be applied to the high TXOP limit of the EDCA access method.

Throughout the years, research community has been actively doing experiments and finding ways on how to enhance the wireless network and support QoE to stream video. The solutions implemented can be categorized into several general techniques.

Internal collision issue: EDCA has introduced four different AC for different types of traffic. However, they still share the same physical layer and thus these different traffic types will compete to use the available bandwidth.

Tune MAC parameters: Secondly, there are a lot of traditional and classic solutions of manipulating the IEEE 802.11e parameters. The fine tuning of the parameters which will be bias towards video traffic will eventually give a higher probability of the traffic to be transmitted ahead of other types of traffic.

Cross-layer approach: A more dynamic solution, the cross-layer approach is being introduced and implemented to achieve better results in terms of video traffic performance. Cross-layer allows information to be shared across the different layers of OSI and thus enables the possibility of developing an adaptive solution to cater video traffic based on different types of scenarios.

Early research has been doing the fine-tuning of the IEEE 802.11 independent parameters such as Contention Window (CW), Transmission Opportunity (TXOP) and the interframe spaces, which has been the classic approach. These solutions give the high priority traffic shorter waiting time to access the wireless media, but do not exploit the significance of specific traffic type such as video that needs an adaptive treatment due to the nature of its variability in data rates. For example, (Son et al., 2013) has been trying to enhance the QoS level in EDCA to fit into the medical field. The challenging task is to guarantee the required medical-grade QoS of various medical applications since medical grade QoS is directly connected to the patient condition. By considering two metrics (packet delay and the ratio of late and received packets), the AIFS were adjusted to improve network performance. It is shown that through this proposed method, more QSTA are allowed to be connected before the network becomes saturated compared to the original EDCA. However, this does not really address video transmission in wireless networks. Though positive results have been recorded, most of the solutions has been non-adaptive and is not content-aware in terms of the video contents as well as the types of the video packets.

Managing internal collision is stems on having a right balance between high and low priority traffic. Since the introduction of four different ACs in EDCA, research community has been focusing on the Virtual Collision handler (VCH) algorithm for managing internal colliding traffic streams. In the typical wireless LAN environment, internal collision of packets will lead to the ACs with lower priority backs off and double the Contention Window. After retrying several times, the packets will be dropped due to the maximum number of retransmissions or full buffer. This issue is important especially when video traffic collide with voice traffic where AC_VO will have the higher precedence. While AC_VO can continue to transmit, AC_VI will back off and possibly end up packets being dropped. Within this domain, several researchers tried to address this issue which will be discussed in the next paragraphs.

(Tariq and Perveen, 2010) tune the EDCA Contention Window to address internal collision. The work found out that though assigning a shorter Contention Window (CW) range for high priority traffic is good, it affects the low priority traffic badly. This is because when high and low priority traffic collides within the same QoS Station (QSTA), the high priority traffic will always get the preference of the Virtual Collision handler (VCH) to transmit. Low priority traffic will back off. Tuning the low priority traffic to a lower range of CW also does not help much as the VCH prefers the high priority traffic especially when the high priority traffic flow is high.

Meanwhile, (Prakash and Thangaraj, 2012) compared the performance of both EDCA and DCF under non-saturated network condition to study the effects of service differentiation through CW and AIFS differentiation among different access categories. The main concern in their work was to see the effects of collision among the flows that are competing for access to the wireless medium. In their work, performance of both DCF and EDCA are analysed on different number of stations, with different parameters of CW and AIFS and the results show that the throughput performance of EDCA is better. However as expected, as the number of stations grows, EDCA throughput decreases as well as DCF due to the increased number of collisions. In EDCA, the throughput of voice and video increased. The throughput increases for high priority traffic while it is evidenced that throughput for low priority traffic is relatively low.

Still within the same domain of managing internal collision, (Liu et al., 2013) introduced a Mechanism for Internal Collision (MIC) to reduce the internal collision rate and provide fairness for lower priority traffic by decreasing the backoff time of AC with lower priority. When collision occurs, instead of doubling the CW, low priority traffic will not enter backoff straightaway. Low priority AC acts according to the successful transmission rate of the high priority AC. If high priority traffic AC transmitted the packets successfully, the low priority traffic will transmit after SIFS. Else, it will enter backoff and doubles the Contention Window. Compared to EDCA, MIC can improve the performance of low priority AC and at the same time, does not worsen high priority AC delay. With the increase of nodes, delay of MIC and EDCA increases gradually but delay of MIC is still lower than EDCA. Moreover, the method effectively solves the internal and external collision problem where the delay of lower priority traffic declined at almost 40%.

On the other hand, cross-layer approach has been the popular approach in recent years. This solution enables information to be shared across layers which allows solution mechanism to be selective and adaptive. For example, the queuing system at the MAC Layer will be able to identify the types of the multimedia packets which were being "told" by the upper layer and thus give appropriate preference of priority without wasting the bandwidth and at the same time offer moderate fairness. Within the cross-layer domain, video traffic can be identified and treated more accurately because the system knows more specifically the type of video frame a packet is carrying.

Through the idea of cross-layer, (Ksentini et al., 2006) proposed a new mapping algorithm to fit VBR video traffic into the EDCA mechanism. A static mapping algorithm was used to remap the IEEE 802.11e EDCA where the I-frame will always be mapped to AC[2] while P-frame and B-frame will be mapped to AC[1] and AC[0] respectively. However, if the AC[2] queue is empty (traffic load is light), mapping P and B-frame to AC[1] and AC[0] will just pose unnecessary delay and jitter. To make things worse, there will also be a high number of video packet loss if both AC[1] and AC[0] queues are full.

Meanwhile, (Lin et al., 2009) tries to accommodate VBR traffic in to IEEE 802.11e EDCA by introducing a cross-layer mapping algorithm. The work done tries to overcome the shortcomings of static mapping where it dynamically maps video traffic to the appropriate Access category (AC) based on two parameters: significance of video data and the network traffic load. In 802.11e, the video will be channelled to the AC[2] queue. Problem arises when there is a sudden burst of video traffic which makes the buffer queue to be suddenly full. The buffer will be overflown, and any incoming packets will be dropped. The authors suggested that if the AC[2] buffer is full, any incoming video frames will be channelled to other queues (AC[0 or 1]) depending on the parameters described previously. This process is called demotion. However, the video traffic will never interfere the highest priority queue of AC[3]. On doing this, they introduced threshold high and threshold low to monitor the traffic load. This is to map and change the priority of the incoming packet. Through simulations, it is evident that the I-Frame loss of the video traffic had been reduced from a total of 14 frames to 4. Though this method is more flexible, it was not mentioned in this work on how the demotion probability is derived.

Work done by (Rodrigues et al., 2008) controls the quality level of video by providing a mechanism to discard packets during network congestion. They proposed QoS/ QoE Adaptive Video Control (2QAV) which has two operational modes. In the basic mode, 2QAV adapts video sessions to the current network conditions. Frames are only dropped according to their importance to keep the system simple. Meanwhile in the enhanced mode, 2QAV adapts the video quality level by considering the dependency of the video session. This means 2QAV would drop non-video packets, as well as dropping packets based on the dependency of set of frames. If an I-Frame cannot be admitted in a network, B-Frames that depends on the said I-Frame would be dropped as well. 2QAV showed a significant improvement in protecting P-Frames where it reduces P-Frame loss by 60% and 23% in best-effort and Diffserv respectively. However, since all traffic class (video and non-video) shares the same queue, (no dedicated queue for each traffic class as in the IEEE 802.11e), the process of discarding non-video traffic and best-effort traffic would mean more slots for video traffic. This helps a lot in saving the video packets from being dropped. In comparisons to IEEE 802.11e where video and non-video traffic (voice, best effort and background) traffic has their own dedicated queue, dropping non-video traffic would not give a significant improvement in terms of video quality so long as the video queue is still congested.

The relationship between adapting a video bitrate, the content of a video and limitation of network resources is relatively less researched. To optimise and harmonise the said entities, (Khan et al., 2010) introduced a QoE-driven model to optimize content provisioning and network resource utilization. The optimisation of content provisioning is done by determining the initial content quality as being requested by users' QoE requirement, before adapting the video sender bit rate (SBR). The QoE requirement and prediction is based on the authors' previous work in (Khan et al., 2009). Based on a user's QoE requirement, an appropriate SBR is calculated by the content provider and optimised resources are provided by the network operators. Results showed that at the Application level, the proposed solution successfully maximised the content provisioning by adapting the SBR while the proposed solution at the network layer successfully determined a trade-off between Application Level QoS (AQoS) and Network Level QoS (QoS).

In a cross-layer approach that involves the APP, MAC and PHY layer, (Abdel Khalek et al., 2012) proposes a design to ensure better QoE for H.264 videos that are coded with Scalable Video Coding (SVC). Based on the ACK history, online QoS-QoE mapping was proposed. APP and PHY layer are made aware of each other's condition. PHY layer would adapt to provide unequal error protection for each video layer based on the proposed online QoS-QoE mapping. Meanwhile, the APP layer is made to be aware of the buffer starvation on the channel and adjust its rates accordingly. This rate adaptation is done by selecting a set of temporal and quality layers without incurring playback buffer starvation, based on the aggregated channel statistics. Using the proposed mechanism, it is claimed that the architecture avoided buffer starvation while handling channel fluctuations and regulates the buffer very well. It also achieved 30% increase in video capacity in comparison with the throughput optimal link adaptation. However, although this study implements the proposed mechanism in a wireless environment, it is not stated that which wireless standard were used. Besides that, only SVC videos were considered and QoE mapping is based on the QoS metrics and not MOS as perceived by end users.

Meanwhile (Yao et al., 2014) tries to enhance the dynamic mapping of by introducing Adaptive Mapping Mechanism (AMM) based on work done by (Lin et al., 2009). The main difference between AMM and the previous work is that AMM checks for other queues before assigning packet frames to the other queues. Besides that, the highest priority a video frame (I-frame) can be assigned is to AC[3]. P-frame will have the probability to be assigned to AC[3] or AC[2] while B-frame will be assigned to AC[2] or AC[1] depending on the mapping control module. It is confirmed through simulations that AMM reduces packet loss ratio from 20.18% (Dynamic Mapping) and 33.99% (EDCA) to 10.62%. Putting I-Frame packets in AC[3] helps to improve video traffic components. However, how this solution will affect voice traffic (which is more time sensitive than video) was not discussed.

In recent developments, wifi is usually considered as an add-on to mobile data networks for Long Term Evolution (LTE), 3G or 4G. These types of topologies are known as the heterogeneous network. For example, a holistic framework approach called Multimedia Transport for Mobile Video Applications (MEDIEVAL) was introduced by Bo Fu et al., (2013) to achieve optimal QoE in a heterogeneous network environment. There are four main entities that makes up MEDIEVAL which are the Video Service (VS), Transport Optimization (TO), Mobility Management and Wireless Access (WA). All four entities span from link layer to network layer. Meanwhile, traffic management is based on QoE-based optimization and traffic shaping and using VQM to estimate the perceptual QoE, the SVC videos are adapted by dropping the SVC layers. Results showed that the framework is capable to double the MOS readings especially during traffic congestions from below 2.0 to as high as 4.6 in the MOS scale. However, this approach is complicated where it involves a lot of interfacing between layers, and it only uses wifi for traffic off-loading.

Meanwhile, some research focuses more on maintaining high level of QoE while handover. Stations move from one AP to the other and involve handover between those AP. However, the IEEE 802.11 did not specify any recommended strategy. If the wireless handover is to refer to the traditional RSS-based handover, it will cause long time delay. To address this issue, (Zhang et al., 2015) proposed a new method to reduce the delay while handover. This involves network controller to collect the neighbour list to the APs in the network. When a mobile station moves, the station itself will predict the time a handover is going to happen, based on the strength of the Received Strength Signal Indicator (RSSI) of an AP and request the neighbouring list the station actively detects. The station then actively detects the channels until connected to the next AP. Because RSSI is never consistent, the authors introduced handover filter to mitigate signal fluctuations to eliminate additional overhead. Results showed that their proposed scheme had reduced the handover delay effectively with lesser service interruption times.

2.6 Discussion and Conclusion

This chapter started with a discussion on the structure of the MPEG-4 video, as well as the definition and methods of evaluating QoS and QoE. The subjective techniques for QoE evaluation involve watching the received video, give scores based on the quality of the video, which are then mapped using a Mean Opinion Score scale; the objective techniques involve using certain signal measurements to evaluate the perceived video, typically using PSNR and PEVQ.

Secondly the IEEE 802.11e standard is discussed. The mechanism of the standard consists of two channel access alternatives: HCCA and EDCA. HCCA includes AP as the controller to do the polling while EDCA provides priority-based channel access. Both the advantages and disadvantages of the methods have also been discussed. Then, the IEEE 802.11aa amendment which is tailored to address multimedia transmission in multicast environment has also been investigated.

Enhancements to support QoS which was introduced in HCCA and EDCA has also been discussed. These include introducing adaptive TXOP, applying retransmission mechanism and implementing the cross-layer approach. Within the HCCA domain, most of the research aims to allocate more TXOP for video flows or redesign the polling system in favour of the stations that carry video traffic. Meanwhile within the EDCA domain, allocating more TXOP is also the main idea as well as redesigns the queueing mapping or implementing a cross-layer approach to prioritize the video frames.

Maintaining an acceptable QoE level remains a challenging task. Most research involving better video streaming in IEEE 802.11e seeks to identify methods which are sender-centric; to enhance the effectiveness of the sender. However, only few studies include consciously knowing or at least predicting the current QoE level as being perceived by the receiver in real time, and those that have, mostly needs literal feedback from the receiver to report the QoE status which consumes more bandwidth. Therefore, future investigations on predicting current QoE level of the receiver especially without feedbacks from receiver would be helpful to develop a method to be adaptive, based on the current QoE state of the receiver. Prior studies have also used a lot of different video formats in the experiments, MPEG-4 and SVC among other things. Although different encodings have been used, the principle of transmitting the video through networks is still the same: the video must be converted into packets, queued at the MAC layer, and finally being transmitted. Whichever encodings are used, there are rankings of priority within the video frames themselves. For example, MPEG-4 are made of I, P and B-Frames with different priority levels while SVC makes up a video through different rankings of layer priority as well. Meanwhile, although most studies try to deliver all the packets without fail, they discount the studies of the significance of packets being dropped towards QoE. Since most resources in networks are limited (e.g., bandwidth, queue buffer), it may therefore be advantageous to also investigate whether some video packets that are in the queue, to be dropped selectively without jeopardizing the QoE level, to allocate space for more important video packets.

The next chapter will start with the proposed techniques to further enhance the EDCA mechanism to support video traffic beyond the current ability of IEEE 802.11e. The proof of concept and techniques will be introduced before experiments are being conducted.

CHAPTER 3: IMPROVING THE QOS/ QOE IN WLAN

3 Introduction

In the previous chapter, it has been discussed that most techniques in the previous works are sender-centric, disregarding trying to predict the real-time QoE level of the receiver. Besides that, there are very little study on how different packet type that carry different video frame (I, P and B-Frame) drops can affect the QoE level. This chapter brings together both disregarded area and presents a novel QoE technique to prioritize video traffic. The technique is based on the cross-layer method where every video packet is tagged according to the video frame type it carries at the Application Layer. The tags are carried by the packets throughout all the layers so that it can be recognized, therefore each layer can interact with the packets and be adaptive to assign priority levels. Works in this chapter were published in Khambari et al., (2016, 2017, 2018).

Most of the papers focusing on improving the performance of video traffic in the IEEE 802.11e do not identify the frame type of the video, and the type of the video itself (Rapid, Moderate or Slow movement). Although it is acceptable to just prioritize video traffic based on the QoS tag, it is best to really prioritize the video based on frame type, and the type of the video as well. This is because both techniques will refine the prioritization granularity. With prioritization of frame type, there is another service differentiation within the video frame itself so that prioritization can be done selectively in a worst-case scenario (e.g.: during the saturation of bandwidth or queue). Meanwhile, the ability of the system to detect the type of the video (slow, moderate, and rapid movement) will give another dimension of video traffic prioritization to match the quality expectation of the end users.

3.1 Aims of the Proposed Techniques

In this thesis, the proposed work is focused on Media Access Control (MAC) layer. The MAC layer is one of the sub layers that makes up the Data Link Layer of the OSI Model. This sub layer is responsible to allocate medium access of a device, moving data packets from a network interface to another, and vice versa. Packets that are waiting to be transmitted from the interface are queued in this layer. In the IEEE 802.11e standards, there are four different queues that resides in this mac layer, where each queue caters different types of data traffic; voice, video, best-effort, and background which has been discussed in Chapter 2. This thesis emphasized on studying the queue for video traffic on the MAC layer, where different types of video packets reside. In this thesis, MPEG-4 encoding is used and therefore the video is processed to be encoded as PktI[3], PktP[4] PktB[5] and these types of packets make up a complete video frame. In the proposed techniques, the types of the video packets are identified, prioritized, and rearranged in the queue, so that a Quality of Experience of a video traffic can be assured.

[3] Packet that carries I-Frame
[4] Packet that carries P-Frame
[5] Packet that carries B-Frame

The whole idea of the techniques is to prioritize the video frame and protect the I-Frame so that more video traffic can be pushed beyond the ability of the IEEE 802.11e but at the same time, having a better performance in terms of QoE value. One of the techniques to achieve this is to minimize the number of I-Frame packet drops.

Taking Rodrigues et. al (2008) work into perspective, the proposed mechanism might be similar but nevertheless is also different. The approach to queueing in Rodrigues' work is of wired network whilst the proposed mechanism runs on a wireless environment, more robust where less calculation needs to be done. Rodrigues' work protects all the I-Frames by discarding non video packets and therefore this method affects more traffic. Comparatively, the proposed mechanism is less effective, where protecting the I-frame 100% might not be achievable due to the nature of wireless networks, where packet loss is not necessarily due to congestion. The proposed mechanism also incorporates a PSNR estimation mechanism to estimate the receiver's PSNR without getting the feedback from the receiver which will be discussed later in this chapter.

the environment which is wireless, and the method is affecting more traffic than the other. The proposed method is likely to be more robust where less calculation needs to be done. What I'm assuming of the results. My method is less affective but also less aggressive.

The default IEEE 802.11e does not have any prioritization within the queue that can examine the types of the video packet. In a normal video packet transmission, packets (PktI, PktP PktB) might get dropped at MAC Layer of the sender due to several reasons. Two main factors are from a **full queue buffer** and secondly, due to the **packet timeout** in the queue. The queue buffer at the MAC layer is a limited space where in can only store a limited number of packets at a time. During network congestion, packets are unable to be sent by the sender and thus, packets congestion will build up in the queue. Once the queue is full, incoming packets towards the beginning of the queue will be dropped, due to that there are no space left in the queue to allocate the packet. Similarly, when a packet is being stored in the queue for a certain amount of time (possibly due to network congestion too), the packet will also be dropped from the queue. This is to make sure that there are no stale packets in the queue that will prevent new, updated packets to be enqueued in the queue. To summarize, network congestion will cause the queue to be full, and will lead to packet drops due to a full queue and packet timeout.

3.2 Event Triggers for the Proposed Algorithm

In the previous section, network congestion has been identified as a factor that leads to queue congestion within the sender. When a queue is congested, incoming packets towards the queue will get dropped. In a worst-case scenario, a PktI might get dropped if it is the incoming packet towards the full queue. This will therefore lead to a high video quality degradation. Based upon the above factors on how a packet could possibly be dropped, the proposed techniques will prioritize PktI at the MAC layer level to protect it from being dismissed from the queue. If the queue is being detected as full, the proposed algorithm will be activated. This means, the event of the queue being full **(Queue Full, QF)** is the threshold or trigger of the proposed algorithm to be activated.

In addition to this, another event that is being proposed in this thesis is the **Predicted PSNR, PP**. PSNR (Peak Signal to Noise Ratio) is a parameter to gauge a video's QoE level. To build upon this, it has been discussed in the previous chapter that most of the previous work are only sender centric, which means the real-time QoE level of the receiver is often discounted. This leads the system to be rigid and thus cannot be adaptive towards network traffic changes. To address this issue, it is therefore in this proposed technique, the QoE level of the receiver will be considered. To do this, the PSNR level of the receiver will be predicted in real-time. The proposed technique will also be activated if it is being detected that the PSNR of the receiver falls below a certain threshold. Therefore, **Predicted PSNR (PP)** is also an event, besides **QF**, for the proposed technique to be activated during video packet transmission. Predicting the perceived PSNR of the receiver involves a preliminary experiment, which is discussed later in section 3.4.2.

3.3 Proposed Techniques to Improve Video Transmission in IEEE 802.11e

The focus of this section is to elaborate on the techniques to enhance the video performance in EDCA of the IEEE 802.11e. The significance of focusing on the EDCA area over HCCA has been discussed previously in Section 2.5.2. The techniques in this section are proposed to support relative QoE for video traffic in the IEEE 802.11e wireless LAN.

In the previous section, it has been discussed that there are two events that will trigger the proposed algorithm to be activated. Firstly, is network congestion. Network congestion will lead to **QF** where packets will drop if there are no slots available in the MAC queue. Crucially, it is most desirable to protect PktI from being the packets to be dropped. However, since queue size is limited, trade-off must be made to accommodate an incoming PktI into the already-full queue. Similarly, if another event, **PP** falls below a threshold, it signals that the perceived QoE level of the receiver is deteriorating and therefore, something must be done to stop the PSNR readings from falling and increase it back above the desired threshold.

The basis of the proposed algorithm is to do a trade-off by sacrificing (dropping from the queue) the least important packet which is PktB to accommodate PktI in those two events that prompt to trigger the proposed algorithm. If PktI of a certain video flow comes to the beginning of a full queue and faces one of these two events, (the triggers), the proposed algorithm will have to remove PktB from the queue.

In a queue that holds video packets, it can be assumed that there will only be one of these two cases; the queue only contains video packets from the same video flow, or secondly it contains video packets from multiple video flows. In the first case, removing PktB from the same flow as PktI is straightforward because there is only one video flow.

However, in the second scenario, the proposed algorithm will have two options to remove a PktB; to remove PktB from the same video flow as the incoming PktI, or to remove any PktB i.e remove any PktB regardless of the source of video flow. In this thesis, removing PktB from the same video flow as the incoming PktI is named as "removing PktI's own PktB" or Remove Own PktB (ROPB). This is shown as in Figure 3.1 below.

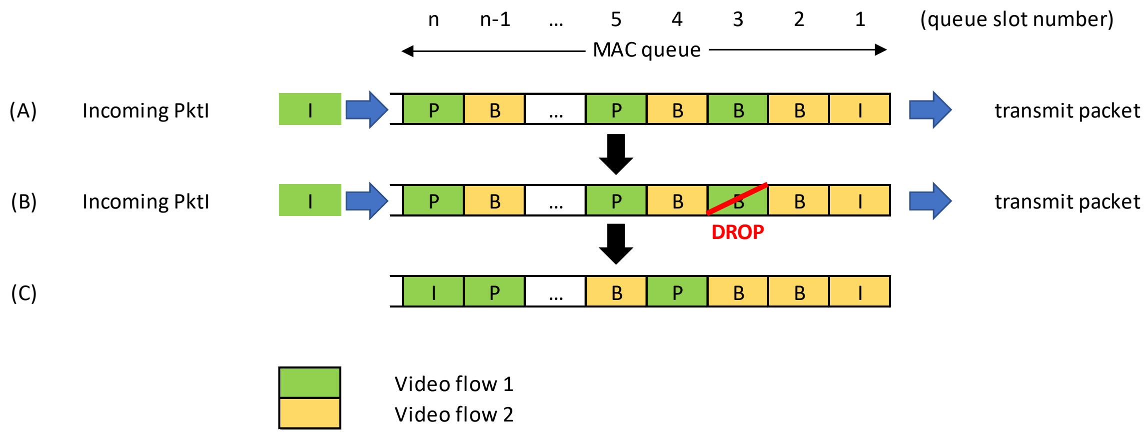


Figure 3.1: Remove Own PktB from the MAC Queue of the Sender

Referring to the above figure, an incoming PktI from video flow 1enters a queue that is already full (A). Instead of dropping the incoming PktI, the algorithm will remove a PktB that has been in the queue the longest time, which is the nearest to the end of the queue. In this case, it is PktB from video flow 2. However, since the incoming PktI is of video flow 1, the algorithm will look for PktB that is of video flow 1 and drops it from the queue (B). This enables the other packets to shift further into the queue, which gives a space for the incoming PktI to slot in (C). The algorithm removed a PktB which is of the same flow as the incoming PktI.

On the other hand, removing PktB from any video flow is named as "removing any PktB" or Remove Any PktB (RAPB). RAPB can be pictured as in Figure 3.2 below.

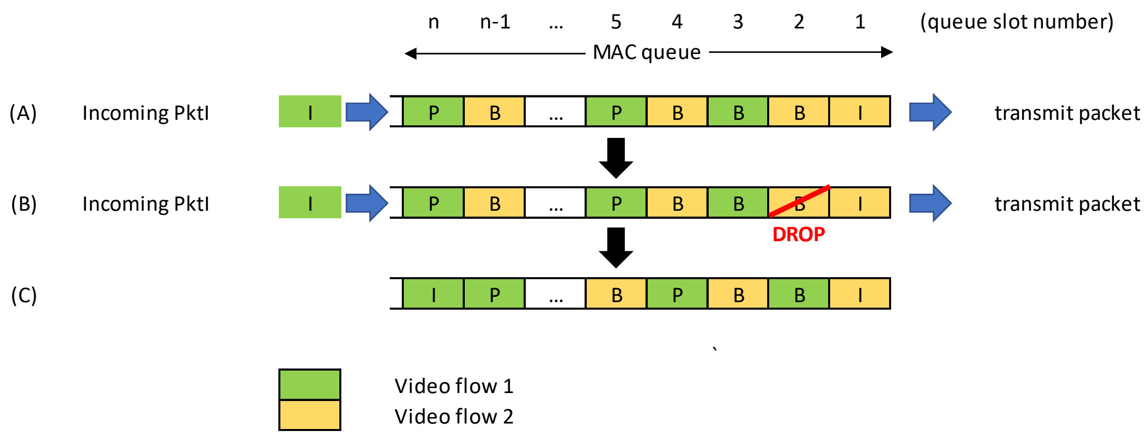


Figure 3.2: Remove Any PktB from the MAC Queue of the Sender

In the above figure, an incoming PktI from video flow 1 enters the MAC queue that is already full (A) but instead of being dropped, the algorithm will find a space by dropping a PktB regardless of the video source of PktI (B). This will pave way for the other packets to shift in, and thus making a space available for the incoming PktI to slot in (C).

Regardless of ROPB and RAPB, only one PktB will be removed for a PktI, and the one being removed is the one nearest to the end of the queue (has the longest time dwelt in the queue).

3.3.1 Remove PktB Based on Queue Trigger

In the previous section, it has been discussed that QF is an event trigger to activate the process of PktB removal, and there are two ways to remove a PktB from the queue. Bringing these two elements together, the algorithm of using queue congestion as a trigger is divided into two methods. The first method is the removal of any PktB that exists in the queue while the second method is removing PktB that comes from the same flow of the incoming Pkt_I, at the event of QF. In both methods, the PktB that is being removed is the one closest to the end of the queue. It is also the PktB that has been staying in the queue the longest. To differentiate both PktB removal methods throughout this thesis, naming conventions of removing PktB based on QF trigger is as follows:

Q-ROPB: Queue Full - event-based trigger and Remove Own PktB (Remove PktB nearest to the end of the MAC queue, which has the same flow as in the incoming PktI). Packets are checked whether they come from the same flow by looking up the Application Id (AppId)[6].

Q-RAPB: Queue Full - event-based trigger and Remove Any PktB (Remove Any PktB nearest to the end of the queue, regardless whether it is from the same flow as the incoming PktI).

Q-ROPB flowchart is shown as in Figure 3.3.

[6] AppId: the marker used to identify event flows in a simulated scenario. Each video flow is generated from different applications, and every application has a unique Id.

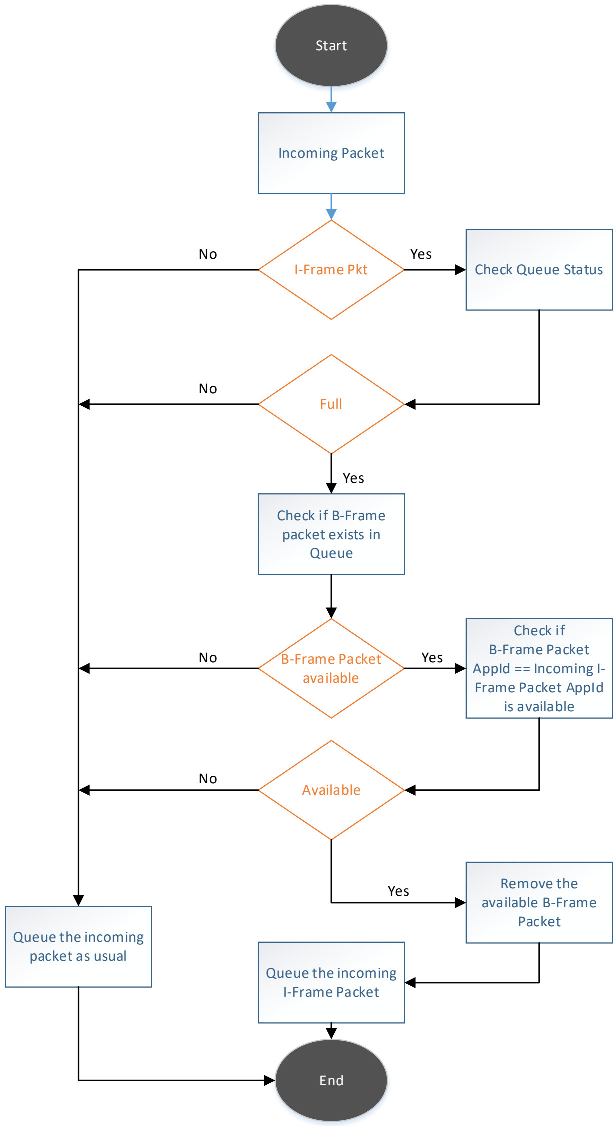


Figure 3.3: **Q-ROPB** Flowchart

In **Q-ROPB**, it is expected that a video flow will only affect its own QoE and does not interfere with the other flows' performance.

Meanwhile, **Q-RAPB** can be depicted in a flowchart as below:

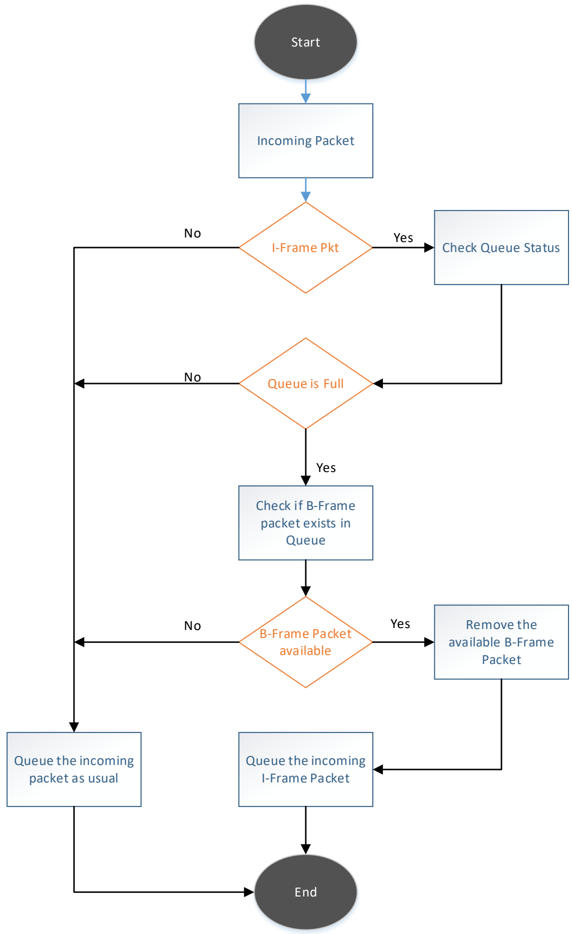


Figure 3.4: **Q-RAPB** Flowchart

All incoming packets of PktP and PktB will be enqueued on the default mechanism as the IEEE 802.11e standard, regardless the status of the queue congestion. However, Pkt_I will be enqueued differently at the event of the queue getting congested. If the queue is congested where the incoming Pkt_I will have to be dropped, the proposed method will try to make room for the Pkt_I (instead of dropping it). This will be done by removing any single PktB from the queue. The emptied slot will then be allocated to the incoming Pkt_I. By applying this method, PktB is dropped instead of PktI.

3.3.2 Remove PktB Based on PSNR Trigger

In the previous sections, it has been discussed that Predicted PSNR (PP) is another trigger that will activate the proposed algorithm. The objective of the PP-based algorithms in this section is to make sure the perceived PSNR of the sender is maintained at an acceptable threshold. In this thesis, the PSNR threshold that will activate the algorithm was set to 30, which is the border between a fair and good quality video, according to the PSNR and MOS mapping (Table 2.2).

In Figure 3.5 below, at the event of the predicted PSNR falls below the threshold while PktI is about to enter the queue, the algorithm will remove any existing PktB from the queue to be replaced by the incoming PktI. By doing this, it is expected that the PSNR threshold would be achieved faster because less important frames are being removed to allow important frames to be queued in as early as possible.

The process of removing PktB from the sender's MAC queue is similar to QF-based method. The only difference is this technique is a PP-based trigger. To bring it further, naming conventions are also set for removing PktB based on the events of PP. The naming are as follows:

P-ROPB: Predicted PSNR – event-based trigger, Remove Own PktB (Remove PktB nearest to the end of the MAC queue, which has the same flow as in the incoming PktI). Packets can be checked whether they are coming from the same flow by looking up the Application Id (AppId).

P-RAPB: Predicted PSNR – event-based trigger, Remove Any PktB (Remove Any PktB nearest to the end of the queue, regardless of whether it is from the same flow as the incoming PktI).

Figure 3.5 shows the flow of **P-ROPB** mechanism. In this method, only PktB of the same source of the incoming PktI will be removed. It is expected that if a video flow's predicted PSNR is not met within the threshold, it will not affect other flow's PSNR because other video's PktB are not removed.

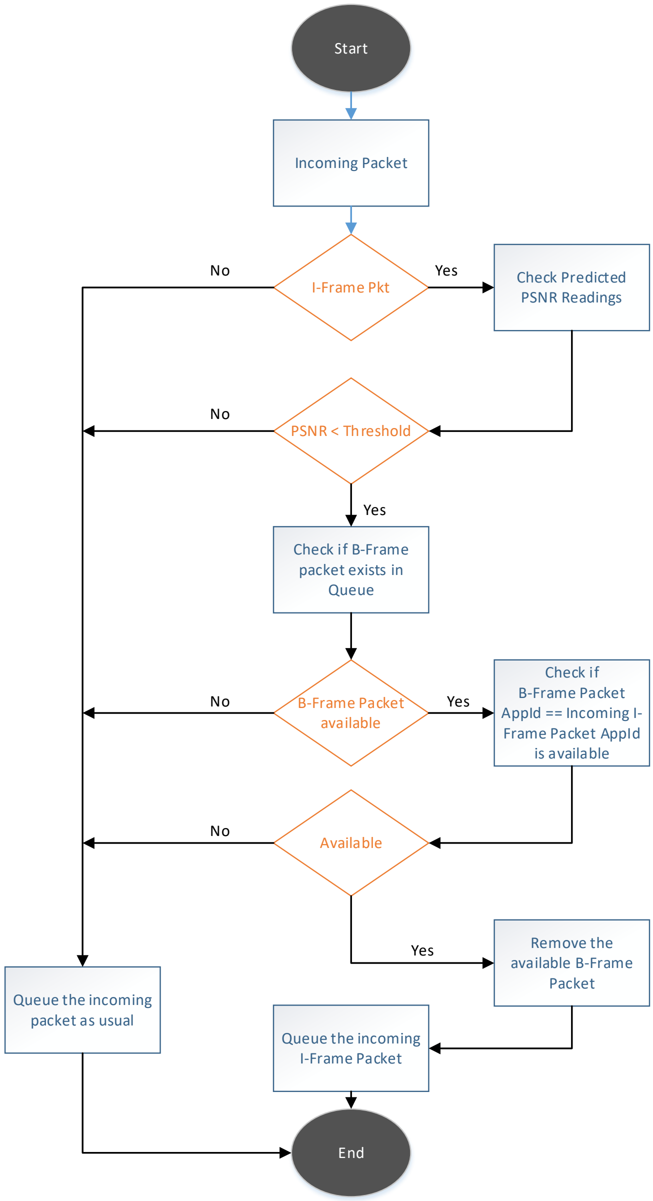


Figure 3.5: **P-ROPB** Flowchart

In the **P-RAPB** method (Figure 3.6), a more lenient approached was adopted as any PktB from any video source can be removed.

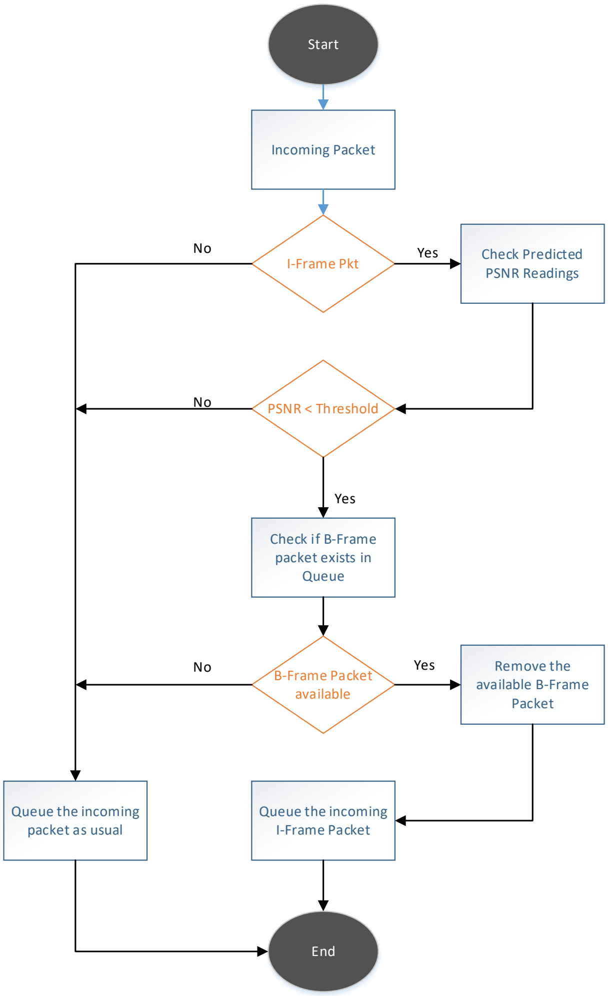


Figure 3.6: **P-RAPB** Flowchart

The main objective of both PP-based proposed algorithms above is to ensure that video flows will meet a certain PSNR threshold. PSNR predictions and queue adjustments are done on real-time so that the perceived PSNR can always be ensured at an acceptable level.

In both methods proposed above, the MAC Layer will keep track of the packets that have been dropped due to queue overflow or queue timeout. The algorithm will then check the MAC layer of the packets that have been dropped, and then predict the current perceived PSNR.

3.4 Preliminary Experiment

In this thesis, preliminary experiments are done as a proof of concept of the proposed techniques. In this research, the preliminary experiment involves two experiments. The first experiment is to predict the contents type of the video by looking into the relationship between I, P and B-Frames. This is important for the proposed technique to determine the content type of the video the sender is streaming. The second preliminary experiment is done to derive a model on how predictions of the perceived video PSNR can be done on the sender’s side.

In the preliminary experiments, MP4 files are used as the video format. Seven videos were used which represents three video type categories: Slow, Moderate and Rapid movement. Categorization of a video were done by (Khan et al., 2009). The seven videos used are Akiyo, Grandma, Suzie, Carphone, Foreman, Football and Stefan. All the video has CIF resolution. These experiments are described in the next two subsections.

3.4.1 Preliminary Experiment 1: Identifying Video Type Experiment

Prior research did not lead to a viable technique for IEEE 802.11e to provide better QoE that can detect the video type automatically, and to be adaptive towards network scenarios and conditions especially at the lower layer where packet types cannot be determine (whether they carry multimedia data). In this thesis, this first preliminary experiment is done to find if there is a pattern between slow, moderate, and rapid movement video. This detection should be doable and should be possible to be implemented in the actual simulation environment of the proposed techniques.

Firstly, to follow the logical sequence of converting a video to packets, the steps are as follows:

i. Video are converted to frames, which consists of the I, P and B-Frames
ii. Each frame will be fragmented to smaller packets so that it can be transmitted through the network.

Building upon step (ii), bigger sized frame will make up bigger number of packets. This leads to the idea of the relationship between the I, P and B-Frames. In a slow movement video, P and B-Frames will not be significantly different in terms of data to be compared to I-Frames. Therefore, P and B-Frames only carry a small amount of data, which only need small-sized packets. In contrast, rapid movement requires P and B-Frame to have data that have a bigger difference than I-Frame. Therefore, these packets carry a lot of data and require a lot of packets to transport them across the network.

However, this thesis is not using the looking into determining the video type by looking at the I, P and B frame ratio and indeed developing a mechanism to determine the type of the video would be useful from a research perspective. However, it does not improve on the objectives of the thesis.

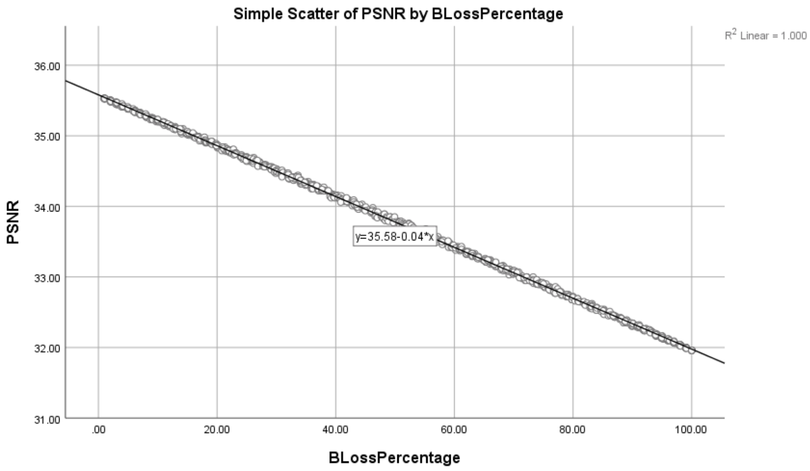
3.4.2 Preliminary Experiment 2: Deriving PSNR Estimation.

In the previous section, it has been discussed that estimating the QoE level of the receiver has been neglected by most of the previous studies. In this thesis, PSNR level at the receiver will be estimated and hence, a technique to estimate the receiver's PSNR at the sender's side should be developed.

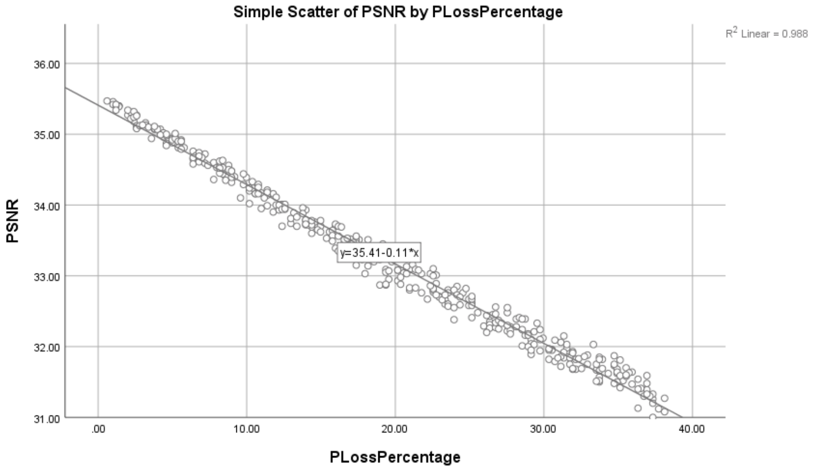
The main objective of this preliminary experiment is to develop an understanding on how the I, P and B-Frames affect the video quality. This is done by looking at the correlation between the percentage of dropped packets, as well as the type of the packets being dropped and how it affects the PSNR readings. By tracking and logging the number as well as the types of packets being dropped at the MAC level of a sender, the PSNR level can be estimated. However, this has the assumption that the remaining packets transmitted from the queue are received by the receiver.

The correlation between dropped packets and how it affects PSNR readings are observed through running simulations. Three sets of simulations ($Test_I$, $Test_P$ and $Test_B$) were done based on the packet types of I, P and B where packets are being dropped from 1% to 100%. Each packet drop percentage scenario is iterated 10 times and hence, each test set are run 1000 times.

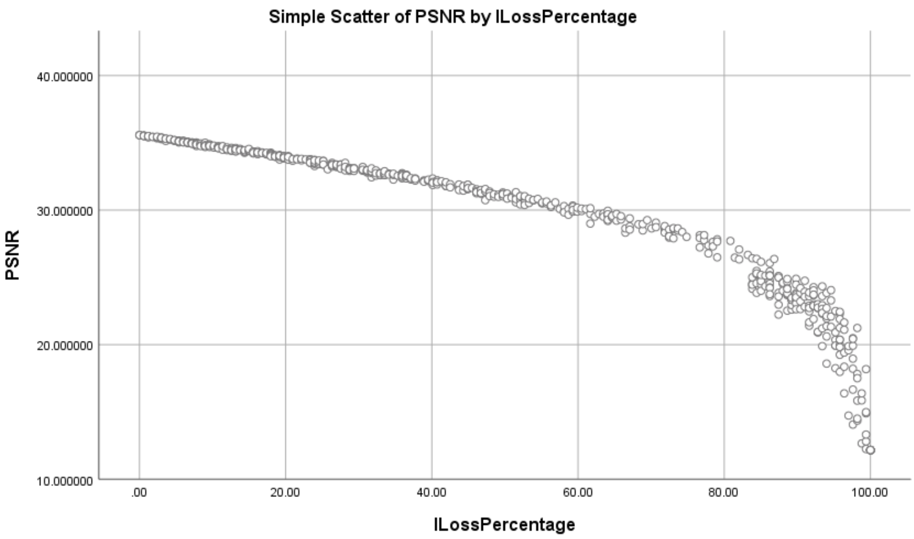
From these simulations, the data were then mapped into a scatterplot to observe the correlation between the dropped packets and how it affects the PSNR. Simulations were done separately based on the packet types as shown in Figure 3.7 below.



(a) $Test_B$: PSNR and Packet B Loss Percentage



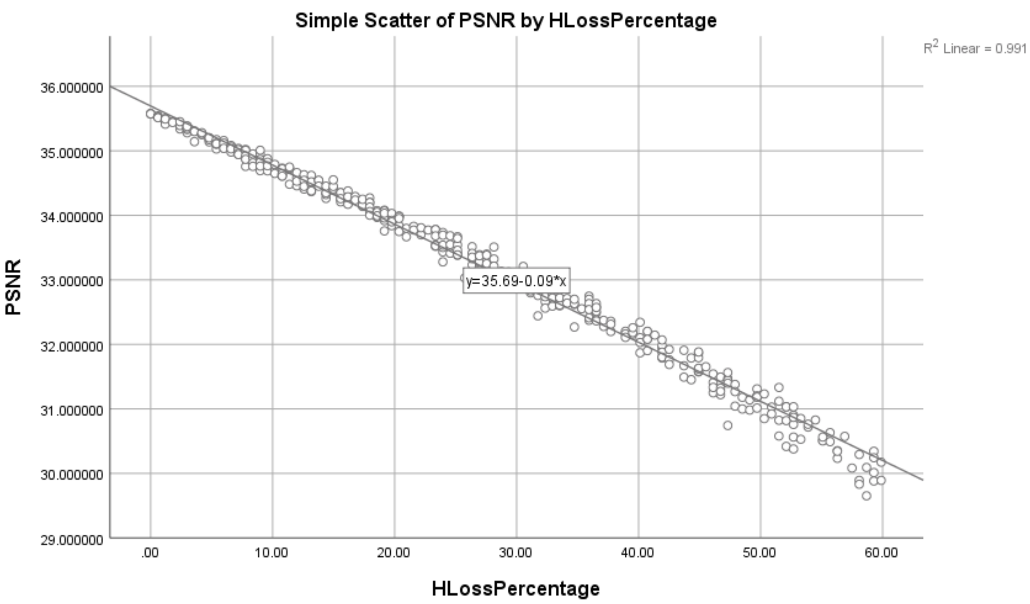
(b) Test$_P$: PSNR and Packet P Loss Percentage



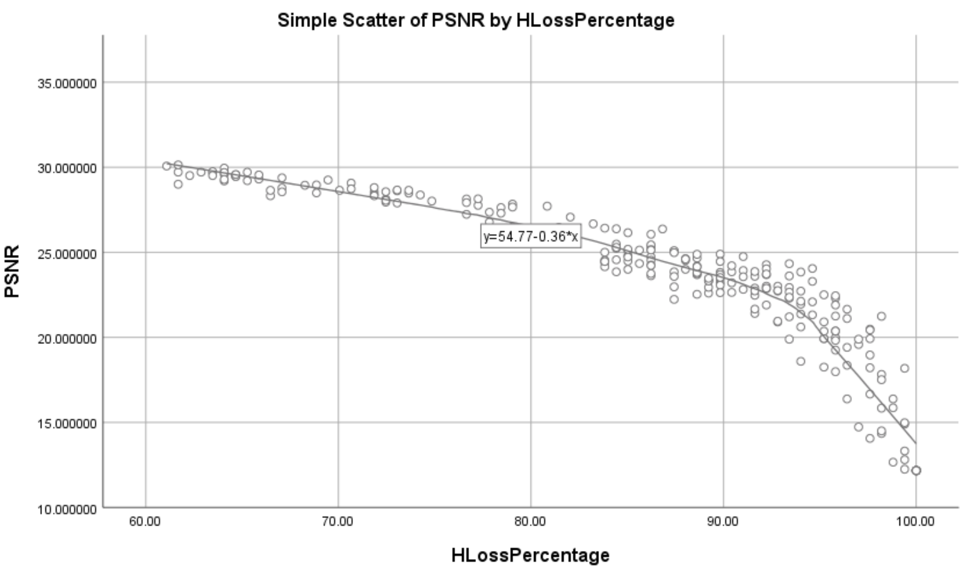
(c) Test$_I$: PSNR and packet I Loss Percentage

Figure 3.7 (a,b,c) Correlation between the losses (%) of different packet types and PSNR

Figure 3.7 above shows the correlation between the losses of different types of frames and PSNR. It is observed that P and B Frames both have a direct linear correlation toward PSNR as shown from the scatterplot. However, I-Frame have an indirect correlation where the trend consists of two patterns. For the overall I Losss percentage test, the PSNR declines in a linear trend between 0% and 60% but declines abruptly towards 100% I-Loss. To examine it further, the scatter graph is divided into two parts where I loss percentage is below 60% and above 60%, as shown in Figure 3.8



(a) PSNR and Packet I Loss Percentage (below 60)



(b) PSNR and Packet I Loss Percentage (above 60)

Figure 3.8 (a,b): Effects of I packet loss towards PSNR below and above 60%

Based on the scatterplots above, the correlation between PSNR and packet drops for each packet type are observed to be linear for both B-Frames and P-Frames and hence linear regression is applied for both scatter plots. However, I-Frames are observed to have different correlation depending on the rate of the packet loss. In these observations of scatterplots, equations are derived. All B, P and I loss below 60% fits a linear regression while I loss above 60% fits the Loess method. Based on the fittings, the equations derived are as follow:

$$\text{Test}_{B}\text{: B-Loss PSNR} = 35.58 - 0.04x \tag{3.1}$$

$$\text{Test}_{P}\text{: P-Loss PSNR} = 35.41-0.11x \tag{3.2}$$

$$\text{Test}_{IA}\text{: I-Loss PSNR (below 60\%)} = 35.69 - 0.09x \tag{3.3}$$

$$\text{Test}_{IB}\text{: I-Loss PSNR (above 60\%)} = 54.77-0.36x \tag{3.4}$$

Previously, it has been discussed that the proposed system will always track and log the dropped packets from the MAC queue at the sender. Therefore the sender will always have the information of I, P and B packet losses. Combined with the equation obtained above, the sender will be able to estimate the current PSNR of the receiver while the video is still being streamed. Overall_PSNR, which is the PSNR estimation calculation based on the tracking log is the reference that the proposed system will always refer to estimate the PSNR. Based on the log, number of packets that has been dropped will be input to Overall_PSNR and hence, the PSNR can be estimated. Estimation is done so that the proposed algorithms will be activated when the video quality degrades to a certain PSNR level. The sender would have not known the PSNR readings, had the dropped packets are not tracked and logged. Upon this, the algorithm will adjust the arrangements of packets in the queue by protecting PktI from being dropped. PSNR estimation mechanism is implemented in both **P-RAPB** and **P-ROPB**.

The derivation of estimated PSNR starts with the proposed mechanism to look at the loss percentage of the most important packet that is the I-Frame packets. This will then follow by examining the loss percentage of P and B-Frame packets. Applying the equations obtained from the previous experiment, the PSNR can be estimated as the following:

$$\text{PSNR Estimation} = \text{MaxPSNR} - (M_I * X_I) - (M_P * X_P) - (M_B - X_B)$$

Where

MaxPSNR = Maximum achievable PSNR based on the experiment done

M_I, M_P, and M_B = Slope of the I, P and B loss percentage respectively

X_I, X_P, and X_B = I, P and B packet loss percentage respectively.

(3.5)

However, since I-Frame losses gives different results based on the percentage of losses, PSNR estimation can be derived based on whether I-Frame losses are below or above 60%. Therefore, by applying the equations derived by the scatter plots, estimation of PSNR can be shown in the equations below.

PSNR estimation (I Loss < 60%)

- PSNR Estimation = 35.69 - 0.09(I Loss Percentage) - 0.11(P Loss Percentage) - 0.04(B Loss Percentage)

PSNR estimation (I Loss >60%)

- PSNR Estimation = 54.77 - 0.36(I Loss Percentage) - 0.11(P Loss Percentage) - 0.04(B Loss Percentage)

To test whether the model developed is viable; simulation scenarios were run and both the estimated and actual PSNR were calculated and compared.

Results showed that the estimated PSNR readings follow closely to the actual PSNR. Results can be shown as in Figure 3.9 below.

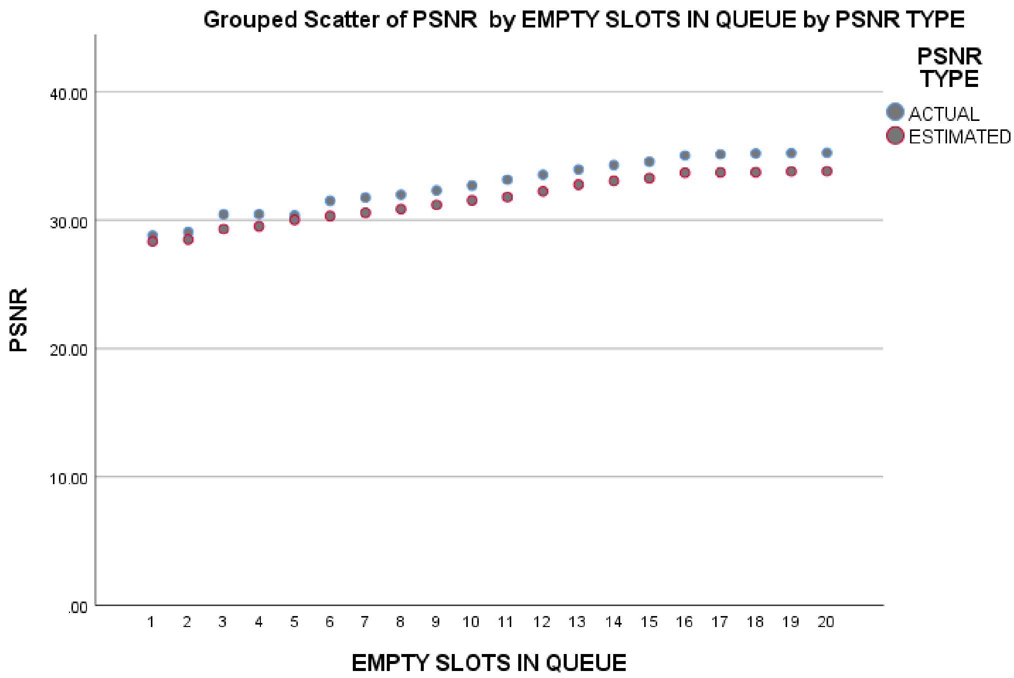


Figure 3.9: Relation of Estimated and Perceived PSNR in the Estimation Model

Estimated PSNR is the PSNR value calculated at the sender while Actual PSNR is the PSNR value calculated at the receiver.

The above figure is an example of how the PSNR Plot Reference is being able to be used to estimate the perceived PSNR closely.

It is important to note that the model that has been developed excludes the scenarios where the first packet of the video flow (packet SeqId=1[7]) is dropped. In this case, the model would be different as this affects the PSNR heavily. From this preliminary experiment, two major findings have been discovered:

Finding 1: The quality of the perceived video is badly affected if the first packet of the first GoP of the video is lost. The packet, which is the first PktI in the first GoP of a video stream, can cause the PSNR reading to drop as high as **19.88%** if it is failed to be received by the receiver. However, based on observations, the probability of these scenarios to occur is very low (0.017). From the total of 2780 scenarios simulated, it only occurred 48 times, especially when the video transmission starts at a time where network traffic is very high with the queue is very congested.

Finding 2: The pattern of PktI loss does not determine the overall PSNR value. This means, PSNR will be the same in a scenario regardless whether PktI is being dropped successively within a GoP or PktI is dropped scattered across GoP. PSNR value will be the same if the number of PktI drop is the same. This does not include special cases as per in **Finding 1** where PSNR will be highly negatively affected had the PktI being dropped is the one with SeqId=1.

[7] The first PktI in the first GoP of a video stream

3.5 Discussion and Conclusion

This chapter proposed a solution to enhance QoE of video transmissions in wireless environments. The main idea is to remove a PktB to give priority to PktI in scenarios where PktI is incoming towards the queue and the network resources have become depleted. In this thesis, the resources refer to the bandwidth and queue buffer on the MAC layer of the sender. Two main factors have been proposed as the trigger to execute the action of removing PktB from the queue. The triggers are Queue Full (QF) and Predicted PSNR (PP) where QF refers to an event where the queue has become congested while PP refers the minimum PSNR level that the sender would like to guarantee to the receiver.

The purpose of using queue congestion as the trigger of removing PktB is to allow the queue being utilized to the maximum. In the other algorithm where PSNR is the trigger, the removal of PktB is purely based on PSNR threshold. In these two algorithms that are being discussed, QF trigger allows the queue to build up packets which means that the queue can be fully utilized while the other trigger, while PP makes sure that the video quality of a video flow is maintained at a desired level.

In this chapter, it has been discussed that predicting the video type of a video is important so that proposed methods can be adaptive towards video content type and network condition. To predict the level of PSNR, preliminary experiments were done in Section 3.4.2 where simulations with random packet drops have been conducted to observe any pattern where the drops of the video packet type have any relation towards the end users' PSNR. Through simulations, an equation has been constructed using regression analysis. This equation has been the core of the proposed algorithm to predict the users' PSNR without having to have feedback from receivers themselves. When the algorithm senses that the PSNR is below the minimum level, the algorithm of removing PktB will be activated, giving priority to PktI.

Other preliminary experiments were also done and discussed in Section 3.4.1, to predict the type of the video that is being streamed by the sender. By examining the GoP of a video, the content type of the video is possible to be identified. This is done by looking at the sizes of P-Frames and B-Frames in relation to the I-Frames. A fast-moving video will not have big differences in terms of size between I, P and B-Frames while slow moving video have the opposite characteristics.

In the next chapter, a detailed information on how the network simulation is done will be discussed.

CHAPTER 4: RESEARCH METHODOLOGY

4 Introduction

In this thesis, computer network simulation has been selected as the method to be used to gain understanding of the research problem as well as to establish the cause and effect of the new proposed method. In this setup, the proposed method will be tested by generating and processing the packets based on the proposed method. Simulation is one of the crucial elements to test the proposed algorithm. It gives an overview whether the newly introduced concept can be adopted or improved. There are several network simulators that has been used in the research field throughout the years, and it has been the favourable method to run experiments. This is due to the advantages of simulations of being scalable, reproducible, and not being constrained by hardware. In this thesis, simulations were done in the NS-3 environment.

4.1 Rationale of the Selected Method

NS-3 has gained popularity in recent years as the codes are more transparent and easier to be debugged, besides being maintained, and validated through a large opensource community. Moreover, NS-3 is built on one language (C++) and therefore, easier to debug. Compared to other network simulators such as NS-2, NS-3 made latest protocols available to be simulated and have the needed components to run simulations needed for this thesis.

Since this research work is based on investigating the performance of IEEE 802.11e, it is crucial that the tool being selected does not only support WiFi, but QoS as well. Within NS-3, an object called *NetDevice* models the different types of network interfaces. One of the interfaces that maps and supports Wifi networks that conforms to the IEEE 802.11 standard is the *WifiNetDevice* model. Besides supporting infrastructure and adhoc modes, *WifiNetDevice* supports different wireless physical layer models (e.g.: IEEE 802.11a/b/g/n/ac) withn different frequency bands (2.4 GHz and 5.0 GHz). Most importantly, it is possible to simulate QoS-based EDCA of queue extensions in IEEE 802.11e, which is crucial regarding to this thesis. The design architecture of *WifiNetDevice* is shown below:

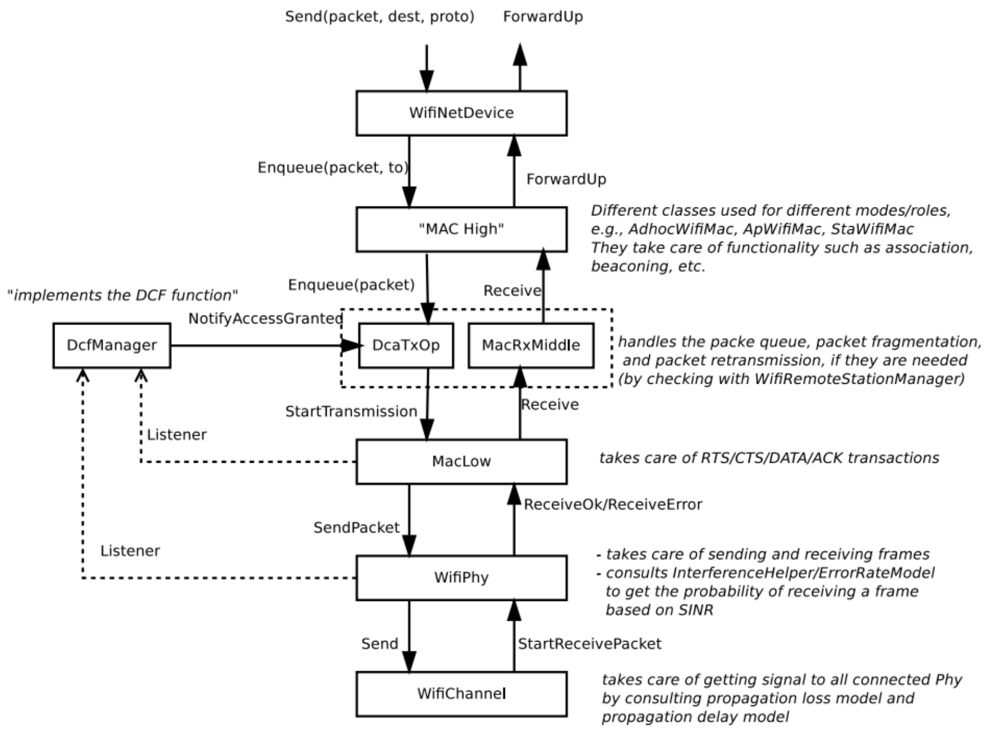


Figure 4.1: Architectural design of *WifiNetDevice* in NS-3 (NS3 Project, 2016)

Three sublayer models below *WifiNetDevice* are *MAC High*, *MacLow* and *WifiPhy*. While MAC High handles non-time-critical processes in Wifi such as the MAC-level beacon generation, probing, and association state machines, *MacLow* model functions like medium access and rate adaptation while *WifiPhy* handles the process of sending and receiving frames. In depth information of each model can be found in the NS-3 Wifi Model documentation (NS3 Project, 2016).

4.2 Method of Data Collection

To simulate real video traffic, an extended module of NS-3 is used which is the Evalvid framework (Klaue et al., 2003). Evalvid simulates the video packet being generated in the network environment.

Evalvid is the main component to run simulation experiments where it is being bound to run as an additional module to NS-3. It is a framework and tool set for video quality evaluation. It can reveal the quality of the video perceived by the receiver through several QoS and QoE parameters which are packet losses, delay, jitter, PSNR and MOS.

Evalvid consist of several components and tools required for video simulation in NS-3. These include the Video Encoder, Video Sender (VS), Evaluate Trace (ET), Fix Video (FV) Peak Signal to Noise Ratio (PSNR) and Mean Opinion Score (MOS):

Source: The source of the raw video file in YUV format. It is the reference video to compare the perceived video at the receiver's side. The experiments described in this project used a CIF (Common Intermediate Format or Common Interchange Format) size YUV video file.

Video Encoder: Video encoder receives video source and converts the YUV files to a streamable format. In this thesis, MPEG-4 format is used. This video stream will be read by the next component which is the Video Sender (VS).

Video Sender (VS): VS is a tool that generates trace files that contains information about the MPEG-4 files, which is given by the Video Encoder. This trace file will be the guideline for the Evalvid tool to generate packets.

Evalvid API: This is the core of the simulation where the tracefiles are being sent through the network simulator. The API will convert the video tracefile to a sender tracefile where each video frame will be segmented to several packets (according to the information from the video tracefile) before being sent. Meanwhile at the receiver, the list of the packets being received will be converted to a receiver's tracefile. These tracefiles from the sender and receiver will then be compared and a receiver's perceived video will be constructed using the next tool, ET/ ETMP4.

ET/ ETMP4: Evaluate Trace (ET) or Evaluate Trace MP4 (ETMP4) is a tool to reconstruct the received video. By comparing the sender trace file and the receiver trace file, ET/ ETMP4 reconstructs the video at the receiver's end.

FV: Fix Video (FV) is used to do error concealment at the receiver's end. In this work, the Fast Forward MPEG (FFMPEG) codec uses this technique to conceal lost frames with the last successful decoded frame so that the numbers of frames being sent and received are equal. This is important especially in the later stage where the perceived video quality is being evaluated by the PSNR tool.

Playout Buffer: This is the buffer where all the packets are being placed before being replay. Longer playout buffer duration will give more time for the receiver to reconstruct and replay the video in case there are miss orderings of received packet frames. However, a long period of playout buffer is not ideal because it does not reflect a realistic environment.

PSNR: This tool is used to measure the video quality objectively. It compares the original YUV file with the one reconstructed at the receiver's end and computes the score of the quality of the video. The quality of the video in terms of PSNR has been discussed earlier in the previous chapters.

The workflow between the components, tools and files involved in the Evalvid framework can be shown in Figure 4.2 below.

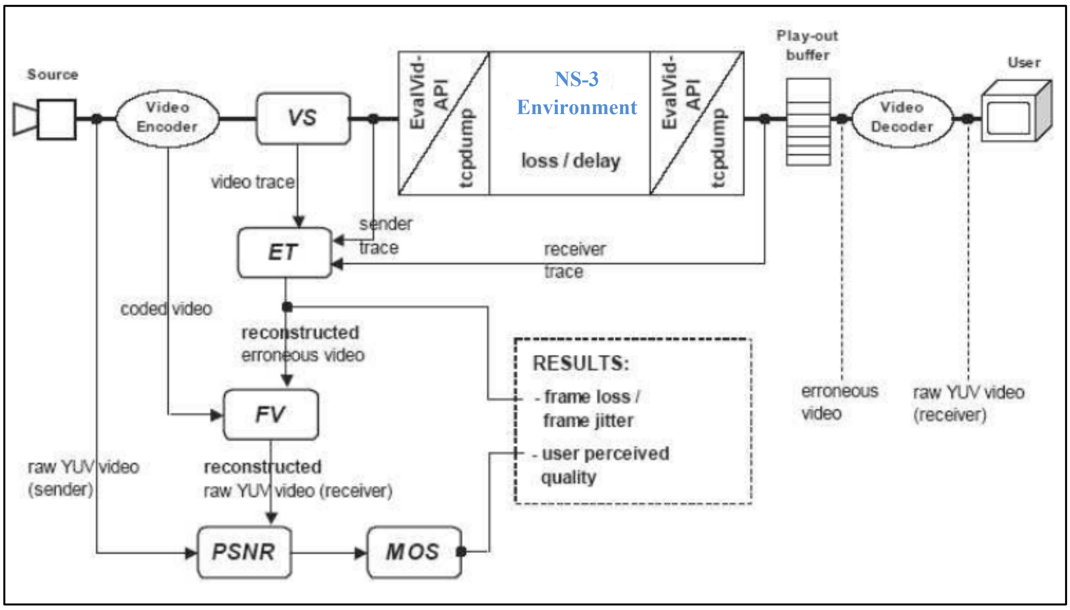


Figure 4.2: Relationship between tools within the Evalvid Framework

Evalvid uses YUV files as the raw video file. In this research work, simulations steps are done as follows:

YUV file is converted to MPEG4 before being fragmented to packets to be prepared to stream in the simulator.

During the simulation where packets travel from sender to receiver, the proposed methods are applied within the simulator to observe the cause and effect of the proposed method towards video quality transmission.

The receiver received the fragmented packets and reconstructed them to MPEG4 and finally to YUV file again. The received YUV file is then being compared to the receiver's YUV file for the perceived video quality to be evaluated. \

4.3 Experiment and Simulation Setup

In this section, the technique on how the simulation scenario setup is determined are discussed before the real simulation itself can be done and executed. Basically, the experiment is grouped categorically according to the proposed algorithm on how PktB is removed.

In the previous sections, it has been discussed that two techniques of PktB removal are proposed, which are Remove Own PktB (ROPB) and Remove Any PktB (RAPB). These two techniques are then run based on two different state-scenario triggers which are Queue Full (QF) and Predicted PSNR (PP). These triggers activate the respected proposed techniques of ROPB and RAPB.

The groupings of experiment can be depicted as in in Figure 4.3 below.

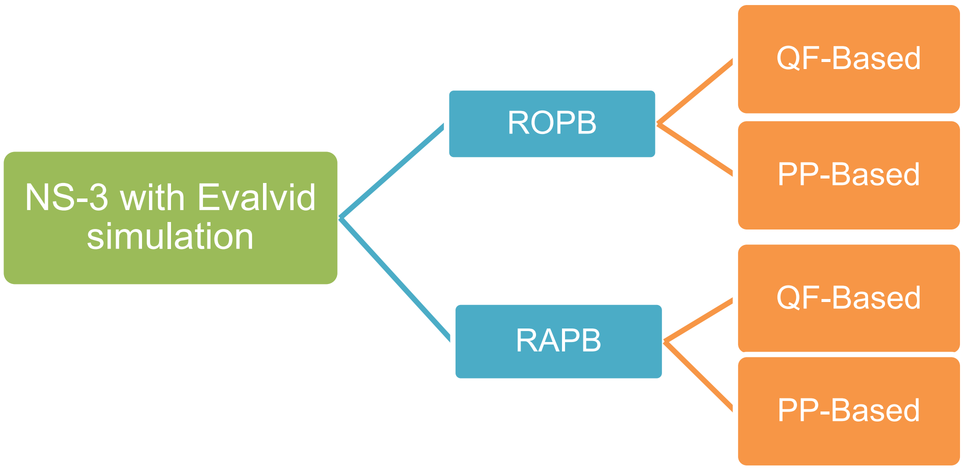


Figure 4.3: Groupings of experiments

ROPB and ROPB are the main groups while QF and PP-Based are the subgroups of the experiments.

4.3.1 Network Topology and Scenario Setup

The wireless network simulation involved in this experiment is infrastructure mode where all the wireless stations will be connected to an Access Point (AP). To minimize unidentified factors that affect the network performance, below are the settings of the scenario that has been implemented in the simulation experiment:

Environment: The network environment being setup is in accordance with the IEEE 802.11e with the QoS enabled. This activates the Enhanced Distributed Channel Coordination (EDCA), where the features of this type of network is being deployed. This includes the multiple queues based on the four traffic types, which are the Background, Best Effort, Video and Voice.

Radio: All simulation conforms to the IEEE 802.11g standard. Bandwidths used are from 1Mbps to 11Mbps, depending on the scenario.

Network Structure: Network is simulated in wireless infrastructure mode. The simulations included one sender, one access point, and three mobile stations acting as receivers

Coverage: All of the stations were in the coverage of each other, therefore the Request to Send/ Clear to Send (RTS/CTS) was not essential and being disabled. This is also to minimize management packets in the network and use the bandwidth mainly for multimedia transmission. Each station can listen to each other to eliminate the hidden terminal factor.

Mobility: Except for the Access Point (AP), all the mobile stations applied a mobility model from a library from NS-3 which is the “RandomWalk2dMobilityModel”. With this mobility model, each station moves at a random direction within a specified boundary. Movements are still within each mobile stations’ and AP wireless coverage to avoid hidden node scenario. AP was in stationary mode.

Addressing: All the stations will communicate using IPv4 addresses within the same subdomain, to eliminate routing issues.

Traffic Generator: Evalvid is used to generate video traffic. The video sender is known as the Evalvid Server while the video receiver is known as Evalvid Client. The underlying transport protocol is User Datagram Protocol (UDP). For voice, background and best effort traffic, the OnOff Application within the NS-3 are used.

OnOff Application: A traffic generator which is built in NS-3. The name of the application is due to the nature of the application that alternates between On and Off states once the application function has been called. During the On state, Constant Bit Rate (CBR) traffic is being generated within a specified rate and packet size.

Video: There are several videos that are usually used in QoS and QoE experiments. These videos are usually categorized based on the nature of the video itself: Slow movement (Akiyo, Grandma, Suzie video), moderate movement (Carphone, Foreman, tennis video) and rapid movement (Football, Stefan, highway). These videos are publicly available for download here (http://www2.tkn.tu-berlin.de/research/evalvid/qcif.html. Last accessed 2016/2/15).

In this research work, the *highway* video is being used. It is a rapid movement video and thus would be more sensitive towards the network performance degradation where the P and B-Frames are significantly different than the parent I-Frame.

The topology of the network simulation can be shown as in Figure 4.4 below.

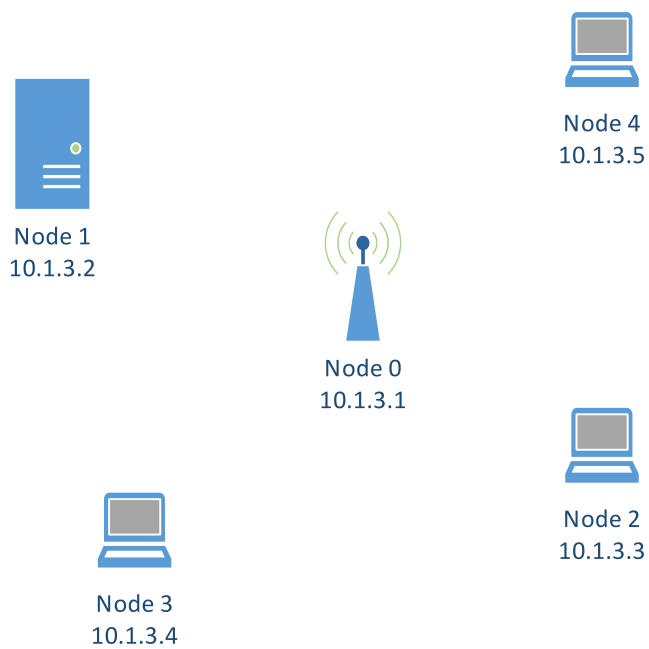


Figure 4.4: Network simulation topology

In the above figure, there is one sender with multiple receivers. Depending on the scenario, the number of receivers will be between 5 and 10. Node 1 (N1) is the sender where it streams the video traffic to the receivers of Node 2 to Node 4 (N2 to N4).

In the next sections, both proposed algorithms of ROPB (Remove Own PktB) and RAPB (Remove Any PktB) are discussed.

4.3.2 Simulation Scenario Experiment

From a bigger perspective, there are two scenario categories involved. The first category is the scenario that used all the default algorithm than conforms to the IEEE 802.11e standard. This scenario is the controlled experiment, or the yardstick for the second category, which is the scenario that applies the proposed algorithm, to compare with. In this thesis, the IEEE 802.11e default algorithm will be referred as **Default11e**. For the purpose of this experiment, there are five flows in each scenario. The scenario involved for the IEEE 802.11e algorithm can be shown as in table below.

Table 4.1: Naming Conventions for Default11e Scenario

Algorithm	Num. of Video Flows	Scenario name
Default11e	5	SC_DEF5

Scenarios that applied the proposed algorithm, which are the **ROPB** and **RAPB**-based experiments are done based on two triggers, which are **Queue Full (QF)** and **Predicted PSNR (PP)**. For both experiments, the setup is similar in terms of the video types being used and number of iterations of the experiment ran. The proposed algorithms are also implemented in scenarios with 5 flows. Based on the elements mentioned, several scenarios are set up and summarized in table below.

Table 4.2: Naming Conventions for ROPB and RAPB-based Scenario

Algorithm	Num. of Video Flows	Trigger	
		QF	PP
ROPB	5	SC_ROPB5QF	SC_ROPB5PP
RAPB	5	SC_RAPB5QF	SC_RAPB5PP

From the table above, there will be 4 different scenarios where every proposed algorithm will have six different scenarios based on the number of video flows and triggers. Each scenario is run 10 times where each video flows will have a random starting time between the scenarios.

Regarding video flows, each video flow are named as **Vi[video flow number]**. For example, video flow number one is named as Vi1 while the second video flow is named as Vi2. To standardize the simulation, the first video flow (Vi1) will always apply the proposed algorithm and therefore the performance of the first video flow will be monitored. The starting time each video flow is randomized to simulate a realistic environment.

4.3.3 Data Capturing and Statistics Retrieval

Since NS-3 is an open-source network simulation and the process to setup the network is 100% code based, careful steps need to be taken to ensure the scenario is correctly designed. In NS-3, an option called *visualize* and a module called *NetAnim* can be enabled so that the scenario can be seen graphically. The visualize module displays the information of each station, including the IPv4 address, MAC address, port numbers involved for each application. It gives a bird's eye view on how the nodes are being positioned besides showing the traffic routes and speed in real-time of the simulation. Below in Figure 4.5 is shown an example of a simulation with the visualize option enabled.

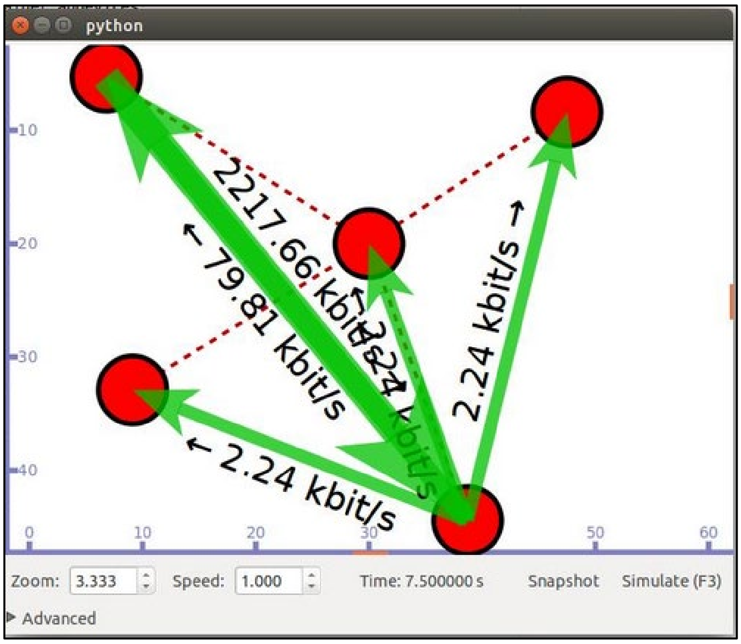


Figure 4.5: Simulation in NS-3 with visualize option enabled

By using the animation tool, it is possible to re-check whether the network has been setup and designed correctly. This ensures that the network has been setup clearly, the data transfer between stations are correctly being sent and received and that the data being captured are valid.

Meanwhile, a combination of NS-3 and Evalvid offers two types of data statistics retrieval. They are QoS and QoE parameters based. These types of data retrieval are discussed in the next subsections.

4.3.3.1 QoS-based Evaluation Tools

NS-3 incorporates a tool which is called Flowmon (Carneiro et al., 2009). It simplifies the process of extracting performance metrics from the simulation events. This is done by introducing flow probes within each traffic flow which can be shown as in Figure 4.6 below.

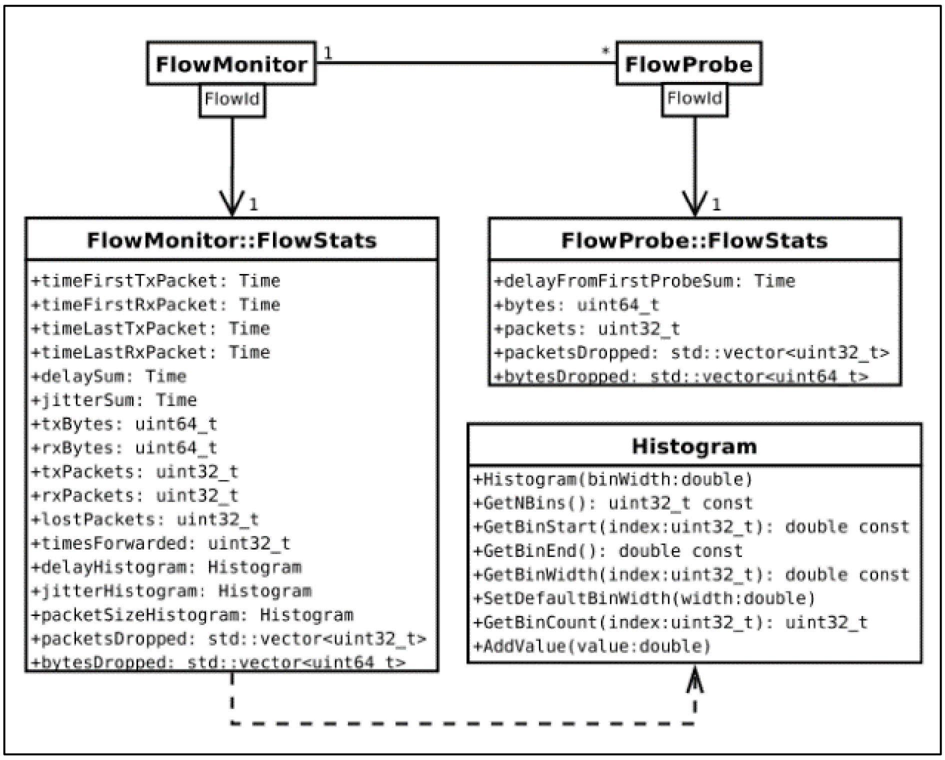


Figure 4.6: Flowmon Data Collection through probes (Carneiro et al., 2009)

Through these flow probes, packet movements are tracked and after the end of simulation, Flowmon calculates the parameters such as throughput, number of packets transmitted and received, flow id, source and destination of addresses and ports.

Besides Flowmon, another tool which is video-specific is also used which is ET/ ETMP4. Besides reconstructing the perceived video file, the ET/ ETMP4 tool also calculates the loss (in percentage), delay and rate of the video packets in the simulation.

4.3.3.2 QoE-based Evaluation Tools

The Evalvid module comes with objective-based QoE evaluation tool, PSNR. The PSNR tool is an executable script that compares the original YUV file of the source with the YUV file constructed on the receiver. This is done by comparing each frame of the YUV files and the difference between the frames are calculated as Peak Signal to Noise Ratio (PSNR). An average value of PSNR with a standard deviation is then presented after the calculation had been done.

Besides that, MSU Video Quality Measurement Tool which is Windows based software is also used. It does not only calculate the PSNR but is also able to generate graphs to compare the PSNR between the sender's and receiver's file.

4.3.3.3 Trace Files

By default, all successful NS-3 simulations will generate a tcpdump file. This is helpful to see whether the network setup and settings are done in the correct and expected way. For example, a lot of customized network tags that was not on the default NS-3 configuration (e.g.: QoS, AppId, NodeId Tags) are introduced in this thesis. Besides checking whether these tags are correctly applied through the simulation codes, these can also be checked through tcpdump files using software like WireShark to check whether the packets are being correctly tagged.

In this thesis, several customized log files were also introduced and created to trace the movement of the packets throughout the network. For example, a log file named Log_PacketDrop.out was created to keep track the packets that have been dropped and it is the source of predicting the PSNR through the proposed model.

4.3.4 Video traffic Evaluation

Video traffic evaluation is done using two main methods: objective and subjective. While objective evaluation is based on PSNR, the subjective method relies on Mean Opinion Score (MOS) that requires real users to literally view the perceived video and give scores.

4.3.4.1 QoE Objective Evaluation

Objective evaluation involves tools being used to evaluate the quality of the perceived video. In the objective evaluation, PSNR scale is being used to determine the quality of the video as perceived by the end user. To get the PSNR readings, several processes must be done to finally reconstruct the packets to form the received video.

After the simulation by Evalvid had ended, two files were generated. Firstly, is the sender trace file which contains the information about the packets and frame type the packets carry. In most cases, the file is named as *sd_a01*. Secondly is the trace file that was generated at the receiver that contains similar information as the sender's trace file, but maybe with some packet loss due to network condition. By default, Evalvid named this file as *rd_a01*. Using these two files, the transmitted video can be reconstructed as seen by the receiver by going through several steps shown below.

Step 1: Reconstruct the MP4 video at the receiver's end. This can be done using the ETMP4 tool from the terminal.

```
$etmp4 -f -0 sd_a01 rd_a01 <input - st_file> <input - original_mp4> <output - reconstructed_mp4>
```

This generates a possibly corrupt video file in which the lost frames are loss.

Step 2: Convert the reconstructed MP4 file to a YUV format. This can be done using FFMPEG:

```
$ffmpeg -i <input - reconstructed_mp4> <output - reconstructed_yuv>
```

This is important because through YUV, a frame-by-frame analysis can be done by PSNR to evaluate the quality of the video.

Step 3: Calculate PSNR. Using the PSNR tool, the quality of the video can be evaluated.

```
$psnr 352 288 420 <input - original_yuv> <input - reconstructed_yuv>
```

The PSNR tool will then make a frame-by-frame comparison between the original and reconstructed YUV video. Three types of readings will be output by the tool which is the frame by frame PSNR, the average PSNR for the video and the standard deviation of the video PSNR.

All the above process from preparing the video to be simulated from a sender to being sent to the receiver as well as being evaluated can be illustrated as in Figure 4.7 below, using the *highway.yuv* video that is being used throughout this research work.

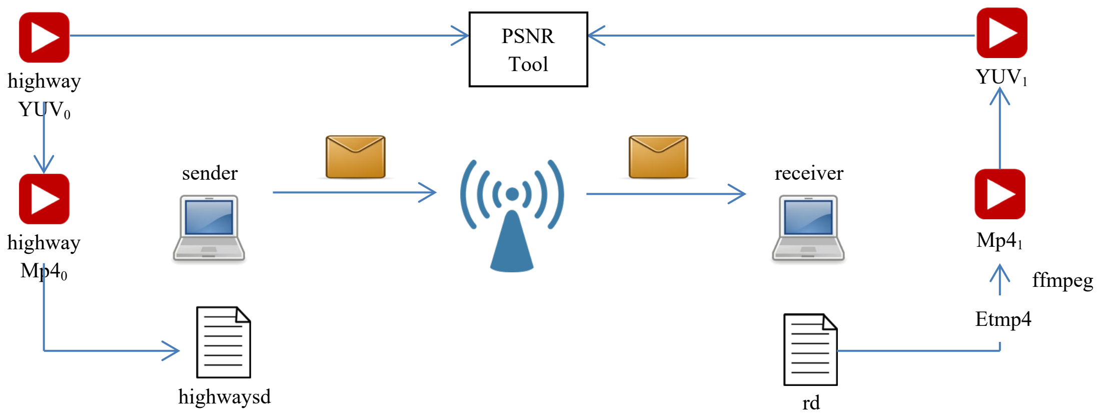


Figure 4.7: PSNR Evaluation Process

The above figure can be broken up into these steps:

Step 1: The original YUV file (YUV_0) from the sender, *highway.yuv* is converted to $MP4_0$ using the ffmpeg tool.

Step 2: The $MP4_0$ is now being prepared to be simulated through the network simulator. The sender on the Evalvid framework does not send the $MP4_0$ literally, but the $MP4_0$ information through a packet generator. This information such as the number of packets required to send an I-Frame is extracted from $MP4_0$ using MP4Trace tool. This information is called the sender tracefile (*sd*). The simulator traffic generator will then generate packets based on the information on the *sd* file and are sent through the network.

Step 3: Upon receiving the packets, the receiver will then extract information based on the packets and create a received tracefile (*rd*).

Step 4: Using ETMP4 tool, *sd* and *rd* files are compared by the Evalvid framework to reconstruct the receiver's Mp4 file, named $MP4_1$.

Step 5: Using FFMPEG tool, the received *highway* $MP4_1$ is converted to *highway* YUV_1. To get the PSNR readings, *highway* YUV_0 and *highway* YUV_1 are compared using the PSNR tool and the PSNR readings are obtained.

The PSNR tool works by comparing the frames between the original and output YUV files. Besides giving output for the overall video PSNR, it also calculates each frame's PSNR for a better PSNR comparison and accuracy.

4.3.4.2 Subjective Evaluation

Subjective evaluation involves examining the actual video as being perceived by the receiver. To do that, several tools are being used. One of the essential tools is ETMP4 which reconstructs the packets being received at the receiver to a viewable video. During the simulation, the Evalvid API will keep track of the packets received by the receiver. The list of the packets receive is called the receiver tracefile. Upon the completion of the simulation, tracefiles from both sender and receiver will be compared and taking the original MP4 as the benchmark, a new MP4 which is as being perceived by the receiver constructed. The MP4 is viewable as being perceived by the receiver. However, to compare the video frame by frame, the MP4 needs to be converted to the raw YUV format. This can be done using the ffmpeg tool.

Another tool that is being used for the subjective evaluation is MSU Video Quality Measurement Tool (MSU Graphics and Media Lab, 2015). This tool allows comparative analysis to be done where the video can be played and compared frame by frame. On the side note, this tool also supports evaluation of other QoE parameters such as SSIM and VQM.

4.4 Discussion and Conclusion

In this chapter, the simulation setup has been discussed. The simulation uses NS-3 as the main platform that supports the IEEE 802.11e infrastructure. On top of that, Evalvid API has been imported into NS-3 to support video streaming using real video. The components that are involved in the simulation are also discussed which include FV, VS and ETMP4 among others. Besides that, the topology of the network has also been discussed together with the scope of the wireless network. The network scenario includes several mobile stations between 5 and 10 depending on the scenario that has been designed.

Besides that, a sub-topic on how data are gathered and captured is also discussed. This is important to see the differences in terms of video performance to compare between the **Default11e** and proposed solution. To make sure the settings are correct, NS-3 allows users to visualize the network topology together with the attributes of the mobile station and the network through a feature called visualize. Finally, after the simulation had been run, data collection is made and analyse through objective and subjective method.

In the next chapter, evaluation of the video streaming performance is discussed and the objective, as well as subjective analyses is compared.

CHAPTER 5: EVALUATION OF THE PROPOSED ALGORITHM

5 Introduction

It is a challenging task to compare proposed algorithms with prior solutions from the research community. This is due to limited access to the previously implemented solutions and a vast difference in terms of environments, scope, and limitations where the solutions have been presented. Therefore, throughout this chapter, the proposed algorithms in this thesis will be compared and evaluated against the standard IEEE 802.11e's performance.

The analysis and evaluation start with the algorithm that is queue based, where the algorithm is enabled when the queue is full. Then, discussions are focused on the algorithm where a certain level of PSNR is to be achieved. Finally, a combination of the previous algorithms will also be discussed to find a balance between minimizing the PktI drop and ensuring that PSNR is maintained above a specific threshold.

5.1 Objective Results of Removing PktB on Queue Full (QF) Event

In the event of Queue Full (QF), two methods of removing PktB are implemented, which has been discussed previously in Chapter 3. Both methods are Remove Any PktB (RAPB) and

Remove Own PktB (ROPB). Several scenarios were simulated with different application starting times and number of stations which has been discussed previously in Section 4.3.2. However, for the simplicity of discussion, the scenarios with **5 video flows (5ViFlow)** are being discussed. This includes scenario SC_DEF5, SC_ROPB5QF, SC_ROPB5PP, SC_RAPB5QF, SC_RAPB5PP. More variation of scenario to verify the efficiency of the algorithm will be discussed later in this chapter.

On the overall scale, every video flow in the simulation scenario has shown increase in PSNR readings in comparison with the **Default11e** queueing proposed in the IEEE 802.11e. This can be shown in a boxplot as in Figure 5.1 below.

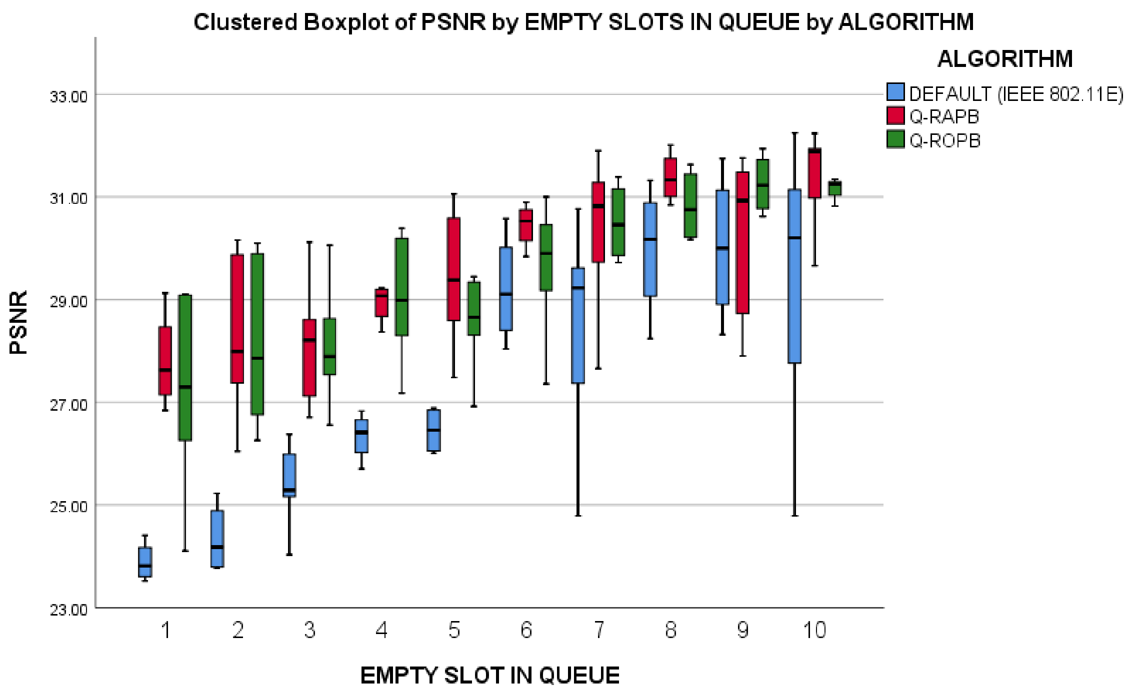


Figure 5.1: Overview of the PSNR performance of the queue-based algorithm

The above graph shows the PSNR on different number of empty slots in queue. The lesser empty slots in a queue, the more effective the proposed algorithm is. This shows that the proposed algorithm works best when the resource is very scarce. Meanwhile as the number of empty slots increases, the algorithm became less effective and the readings of the PSNR became close to each other. This is because the queue size is big enough to make it less likely to be full, and low probability of having the chance where PktI enters the queue while the queue is full (less likely to trigger the algorithm). On the overview, both Q-RAPB and Q-ROPB improved the PSNR readings. Meanwhile, Table 5.1 below shows the mean and standard deviation of the boxplot data of Figure 5.1.

Table 5.1: Standard deviation and mean comparisons between Default, Q-RAPB and Q-ROPB.

Empty Slots	Algo	Mean	Std Dev	Empty Slots	Algo	Mean	Std Dev
1	Default	23.30	1.35	6	Default	28.33	2.17
	Q-RAPB	27.09	1.80		Q-RAPB	29.86	1.38
	Q-ROPB	27.17	2.10		Q-ROPB	29.58	1.38
2	Default	23.72	1.51	7	Default	28.36	2.34
	Q-RAPB	28.29	1.73		Q-RAPB	30.28	1.66
	Q-ROPB	28.17	1.76		Q-ROPB	29.93	1.45
3	Default	25.37	0.58	8	Default	28.87	2.72
	Q-RAPB	28.15	1.34		Q-RAPB	30.63	1.72
	Q-ROPB	28.14	1.31		Q-ROPB	30.12	1.70
4	Default	25.91	1.04	9	Default	28.93	2.75
	Q-RAPB	28.58	0.87		Q-RAPB	30.16	1.74
	Q-ROPB	29.01	1.34		Q-ROPB	30.59	1.57
5	Default	25.98	1.13	10	Default	29.23	2.98
	Q-RAPB	29.42	1.46		Q-RAPB	31.34	1.05
	Q-ROPB	28.54	1.02		Q-ROPB	30.77	1.72

Both Q-RAPB and Q-ROPB showed better average PSNR value of 29.31 and 29.03, respectively. With a standard deviation that are relatively lower than Default (2.42), it is observed that both Q-RAPB (1.23) and Q-ROPB (1.22) showed more consistency and the resultant PSNR data does not vary as much as Default.

In this section, the detailed analyses of both the proposed methods within the same algorithm are discussed together. Both pseudo-code of **Q-RAPB** and **Q-ROPB** is shown below.

Q-RAPB

```
//PktI:  Packets that carry I-Frame fragment
//PktB:  Packets that carry B-Frame fragment

If incoming packet is PktI AND Queue is Full
       Then remove any PktB from the queue
       Enqueue the incoming PktI
```

Q-ROPB

```
//PktI:  Packets that carry I-Frame fragment
//PktB:  Packets that carry B-Frame fragment

If incoming packet is PktI AND Queue is Full
       Then remove PktB from the queue which has the same flow as PktI
       Enqueue the incoming PktI
```

The main difference between the two pseudo-code above is that the first one removes any PktB in the queue while the second one only removes PktB with the same Application Id as the incoming PktI. This means the proposed algorithm for a video flow will not affect or interfere with the other video flows in the same queue.

Table 5.2 below shows readings of how the PSNR readings reacts toward queue size. Both **Q-RAPB** and **Q-ROPB** show a significant improvement in PSNR when compared to the **Default11e** processing, but there is only a marginal difference (less than 1%) between the **Q-RAPB** and **Q-ROPB** performance.

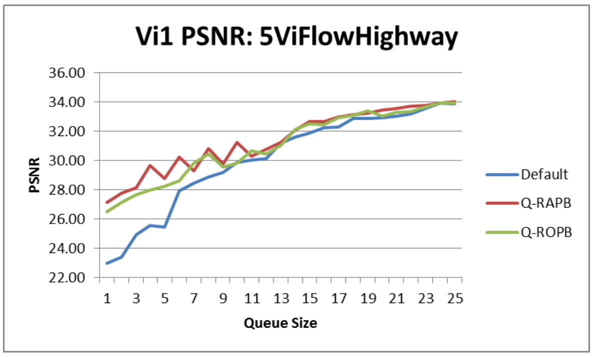
Table 5.2: Average PSNR of Video Flows in Queue Congestion-Based Algorithm

PSNR			
QUEUE SIZE	**Default11e**	**Q-RAPB**	**Q-ROPB**
1	23.27	27.16	26.93
2	23.90	27.87	27.57
3	25.31	28.21	28.01
4	25.81	28.91	28.43
5	26.16	29.10	28.84
6	28.48	29.77	29.41
7	29.00	29.86	29.95
8	29.43	30.64	30.17
9	29.78	30.41	30.34
10	30.23	31.16	30.71

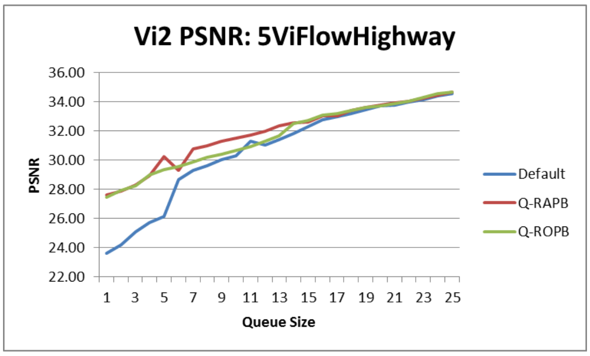
The algorithm works most efficiently where queue size is very small, where the probability of the queue being full is high. At an extreme end where queue size is only 1, the PSNR in **Q-RAPB** and **Q-ROPB** increases by **16.71%** and **15.72%** respectively. The **Default11e** queueing led to a PSNR of **23.27**, which falls into the Poor categorization of MOS evaluation of video quality. As the proposed algorithm of **Q-RAPB** and **Q-ROPB** being applied, the PSNR increased to **27.16** and **26.93** respectively, which increased the MOS to a Fair category in the scale of video quality, which can be referred in Table 2.2.

However, as the queue size increases, the efficiency of the proposed algorithm decreases, which can be seen with a scenario where the queue size is 10. The PSNR difference between the default, **Q-RAPB** and **Q-ROPB** is less than **1%**. The algorithm that has the capability of removing only PktB that has the same id as an incoming PktI has a slightly lower performance. This is because the probability of PktI having its own PktB in queue is lower than the probability of having any PktB in the queue. This means even though there are PktB (least important packet) in the queue, the algorithm still cannot remove the packet unless the PktB belongs to the same id as the incoming PktI when the queue is full. The idea behind removing a PktB that has the same App Id as an incoming PktI is that a video should not interfere with packets in the queue belonging to another video. However, after running several experiments, this algorithm is less practical because there is no point keeping a PktB in a queue regardless of the App Id, to compromise a PktI.

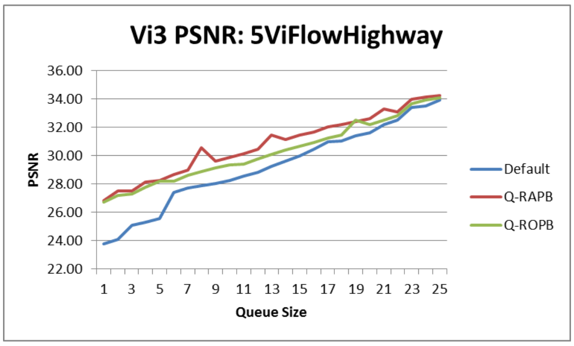
Meanwhile, Figure 5.2 below breaks down the average PSNR into individual graph of each video flow in the 5ViFlow scenario.



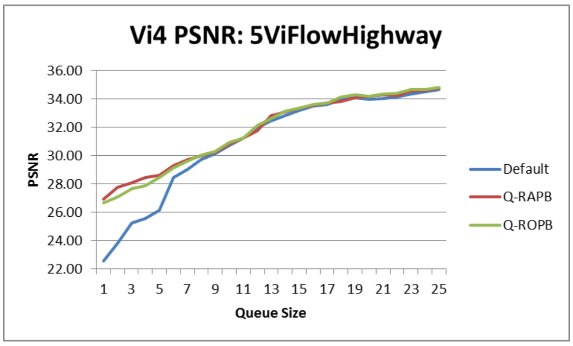
(a)



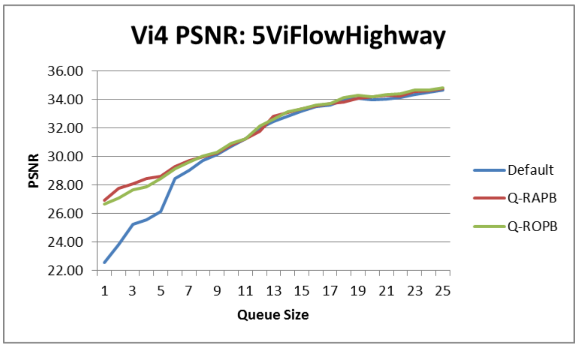
(b)



(c)



(d)



(e)

Figure 5.2: PSNR Breakdown for every flow in scenario 5ViFlow (Default11e, **Q-RAPB** and **Q-ROPB**)

From the figure above, all the flows experience better PSNR values. an increased number of PSNR reading. This is to show that the algorithm has successfully increased all individual flows, which was not shown previously in the average PSNR increase.

The significance of the improvement is statistically proven through a statistical test, depending on the normality of the data distribution. Wilcoxon Rank-Sum tests with Bonferroni corrections is done where **queue size is 5**. Wilcoxon Rank-Sum tests was chosen as the resultant distributions do not seem normal, based on the normality test done through SPSS. The normality test can be shown as in Table 5.3 below.

Table 5.3: Normality Test for Default, Q-RAPB and Q-ROPB

Tests of Normality						
	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	Df	Sig.	Statistic	df	Sig.
Default_PSNR	.138	20	.200*	.950	20	.371
QRAPM_PSNR	.099	20	.200*	.965	20	.644
QROPB_PSNR	.219	20	.013	.743	20	.000
*. This is a lower bound of the true significance.						
a. Lilliefors Significance Correction						

In a normality test, H_0 assumes that data is normally distributed. In a Shapiro-Wilk test, H_0 will be rejected if the significant value is < 0.05. From the table above, the p-value for Q-ROPB less than 0.05 and thus H_0 is rejected, which means the data is not normally distributed.

Tables below show the results of Wilcoxon Rank-Sum tests.

Table 5.4: Wilcoxon Rank-Sum Tests Output Tables

Descriptive Statistics					
	N	Mean	Std. Deviation	Minimum	Maximum
Default_PSNR	20	25.5605	.94389	24.09	27.22
QRAPB_PSNR	20	28.9855	.94471	27.39	30.74
QROPB_PSNR	20	28.4615	2.22824	20.66	30.39

(a): Descriptive analysis Output

Ranks				
		N	Mean Rank	Sum of Ranks
QRAPB_PSNR - Default_PSNR	Negative Ranks	0[a]	.00	.00
	Positive Ranks	20[b]	10.50	210.00
	Ties	0[c]		
	Total	20		
QROPB_PSNR - Default_PSNR	Negative Ranks	2[d]	9.00	18.00
	Positive Ranks	18[e]	10.67	192.00
	Ties	0[f]		
	Total	20		

a. QRAPB_PSNR < Default_PSNR

b. QRAPB_PSNR > Default_PSNR

c. QRAPB_PSNR = Default_PSNR

d. QROPB_PSNR < Default_PSNR

e. QROPB_PSNR > Default_PSNR

f. QROPB_PSNR = Default_PSNR

(b) Signed Ranks Test

Test Statistics[a]		
	QRAPB_PSNR - Default_PSNR	QROPB_PSNR - Default_PSNR
Z	-3.920[b]	-3.248[b]
Asymp. Sig. (2-tailed)	.000	.001

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

(c) Test statistics

In Table 5.4(a), it can already be seen that the direction of the proposed algorithms are giving positive direction where the Means are bigger than the Default algorithm. Observations in Table 5.4 (b) showed a 100% of improvements in QRAPB where all the PSNR readings are positive ranks. QROPB showed the improvements as well, with 80% of the readings are positively ranked. It was expected QRAPB to be better than QROPB as QRAPB will discard more B-Frames which gave higher chances for I-Frame to be allocated in the MAC queue. Finally test statistics in Table 5.4 (c) it can be observed that the significant value of both QRAPB and QROPB to be .000 and .001 respectively. Since this is a multiple comparison test, Bonferroni adjustment is applied where the significant value is divided with the number of tests. Since the initial significant value is 0.05 and there are two tests being compared, the new significant value would be 0.05/2 = 0.025. Both significant values for Q-RAPB and Q-ROPB are < 0.025 which means the proposed algorithms are proven statistically to give significant improvements toward the video PSNR.

5.2 Visual Results of Removing PktB with Queue Full (QF) Event

In the previous section, objective method was used to evaluate the quality of the video. From the statistical view of PSNR readings, it can be clearly proven that the proposed method of **Q-RAPB** and **Q-ROPB** can improve the quality of the video. However, the evaluation lacks the feedback on how the users perceive the quality of the video. Thus, visual comparison is done to see the improvements of the video quality after applying the **Q-RAPB** and **Q-ROPB** algorithms.

The Evalvid module that was installed within NS-3 has the capability to reconstruct the packets that has been received by the receiver to a viewable video. The reconstructed video is then being compared to the original video frame by frame so that visual comparisons can be observed.

Before viewing the perceived video, it is crucial to distinguish the points where the perceived video is affected the most on the **Default11e** algorithm. It is important to note that the original video from the sender contains a total number of 2000 of frames. However, the perceived video contains less than 2000 frames due to the packets being damaged or loss. This leads to several frames failed to be reconstructed. For the Vi1 Flow, the **Default11e** algorithm successfully reconstructed 1990 frames while **Q-RAPB** and **Q-ROPB** reconstructed 1997 frames

In the visual comparison evaluation and analysis, Vi1 (Video Flow 1) is taken as a sample to view the quality of the video. Figure 5.3 below shows the PSNR of each frame throughout the video between the **Default11e** and **Q-RAPB** algorithm.

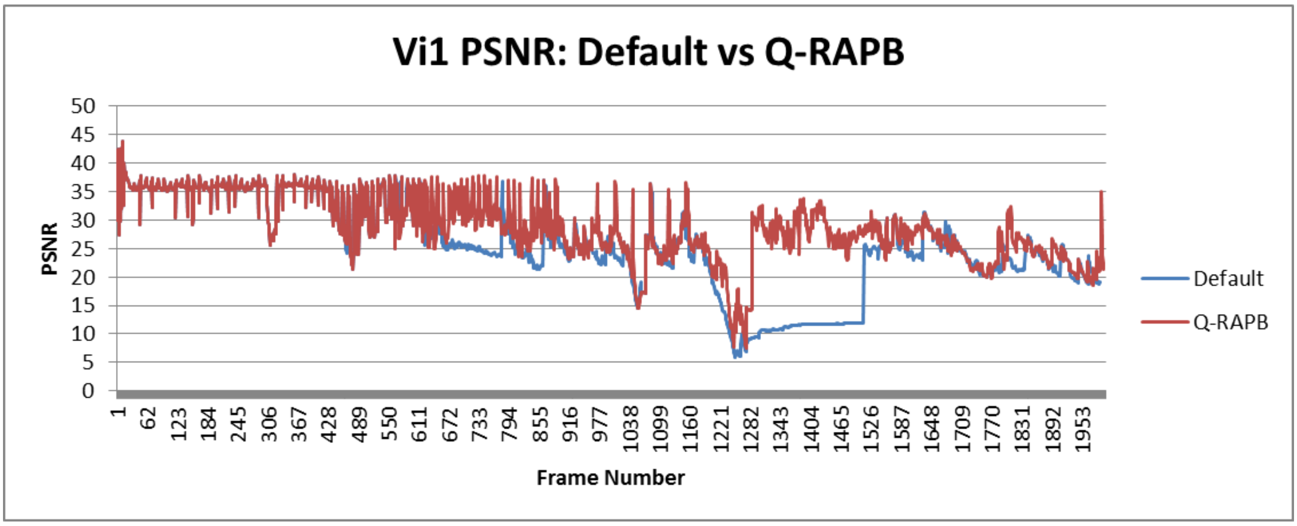


Figure 5.3: PSNR comparisons per frame between the Default and **Q-RAPB**

The overall PSNR in Vi1 for **Default11e** and **Q-RAPB** is **25.49** and **28.76** respectively. From Figure 5.3 above, there were several significant points where **Q-RAPB** had better PSNR readings than **Default11e**. For example, between Frame 659 and 753, the video PSNR that applies **Default11e** algorithm maintains at about 25.00 while **Q-RAPB** between 30.00 and

35.00. Meanwhile at Frame 1317 and 1505, the PSNR of the video on **Default11e** algorithm drops to between 10.00 and 15.00 while **Q-RAPB** rectifies that by maintaining the PSNR between 25.00 and 30.00 by rearranging the packets to push out PktB while giving priority to PktI. The **Default11e** PSNR readings behaved in that manner due to loss of packets where it does not have any method to save important packets when the queue is full.

Meanwhile the overall PSNR value for **Q-ROPB** is **28.22**, which is not very much different from **Q-RAPB** at **28.76**. However, comparing **Q-ROPB** to the **Default11e** algorithm, improvements can be seen which can be shown in Figure 5.4 below.

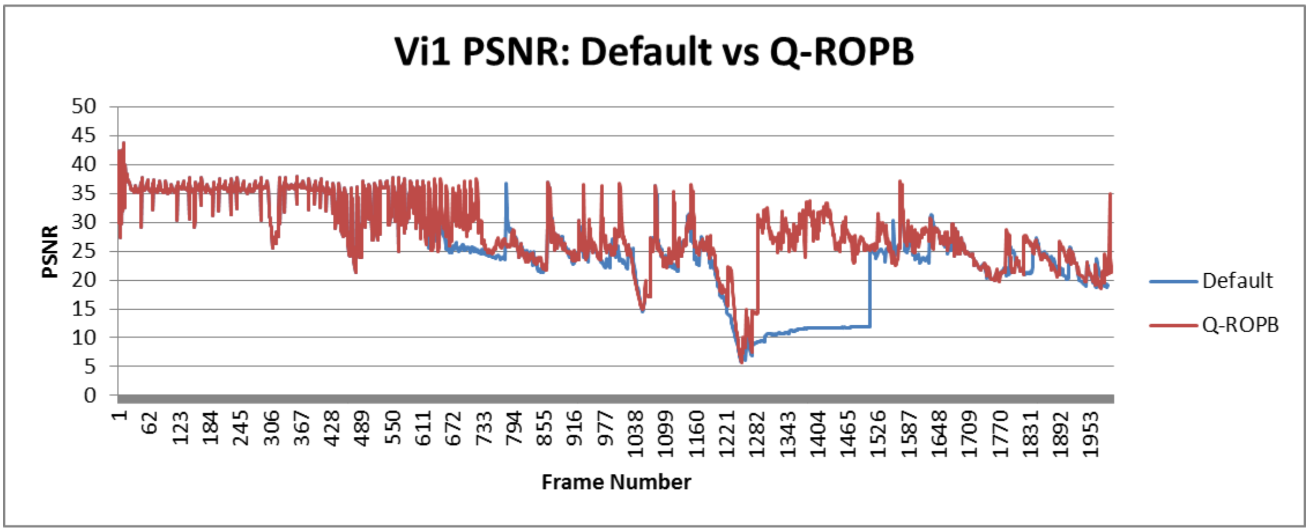
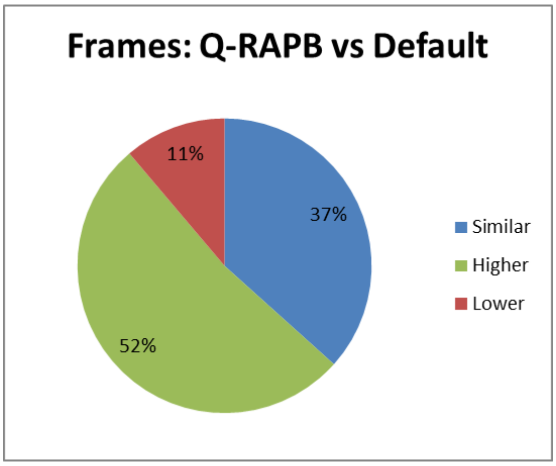


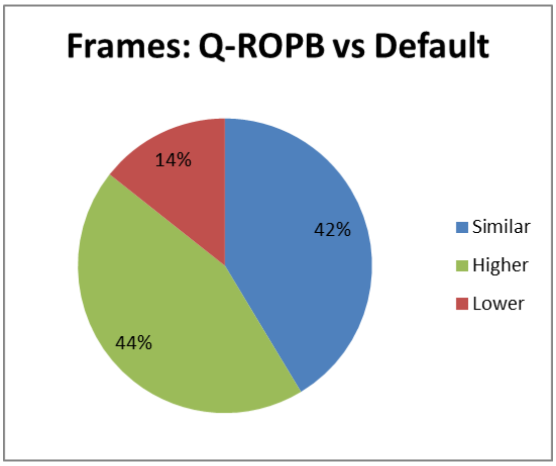
Figure 5.4: PSNR comparisons per frame between the Default and **Q-ROPB**

There are several parts where **Q-ROPB** had successfully maintained high PSNR readings. Throughout Frame 611 to 1529, differences in PSNR readings can be seen where **Default11e** recorded PSNR of 23.03 while **Q-ROPB** has a PSNR of 29.20. Although there are several spikes of higher PSNR recorded by the **Default11e** algorithm, most of the PSNR readings in that period are higher in **Q-ROPB**.

To get a clearer picture, a ratio of between higher, similar and lower PSNR value in comparison between **Default11e** and **Q-RAPB**, and **Default11e** and **Q-ROPB** can be shown as in Figure 5.5 (a) and Figure 5.5 (b) respectively.



(a)



(b)

Figure 5.5: Comparison of Frame PSNR Level of **Q-RAPB** and **Q-ROPB** in relative to **Default11e** algorithm within the Vi1 Flow

In terms of frame-to-frame comparisons, **Q-RAPB** has 1036 frames (52%) with higher PSNR than **Default11e** while 223 frames (11%) of **Q-RAPB** have lower PSNR than **Default11e**. Meanwhile for **Q-ROPB**, 287 frames (15%) of the frames in **Default11e** algorithm have higher PSNR values whereas 886 frames (44%) of the frames are higher in **Q-ROPB**. 824 of the frames (41%) have similar PSNR values.

Figure 5.6 below gives an overview of how the damaged frames from the **Default11e** algorithm look like in comparisons to the same frame where **Q-RAPB** and **Q-ROPB** were able to maintain the PSNR at a higher level. The **Default11e**, **Q-RAPB** and **Q-ROPB** algorithm are denoted as (a), (b) and (c) respectively.

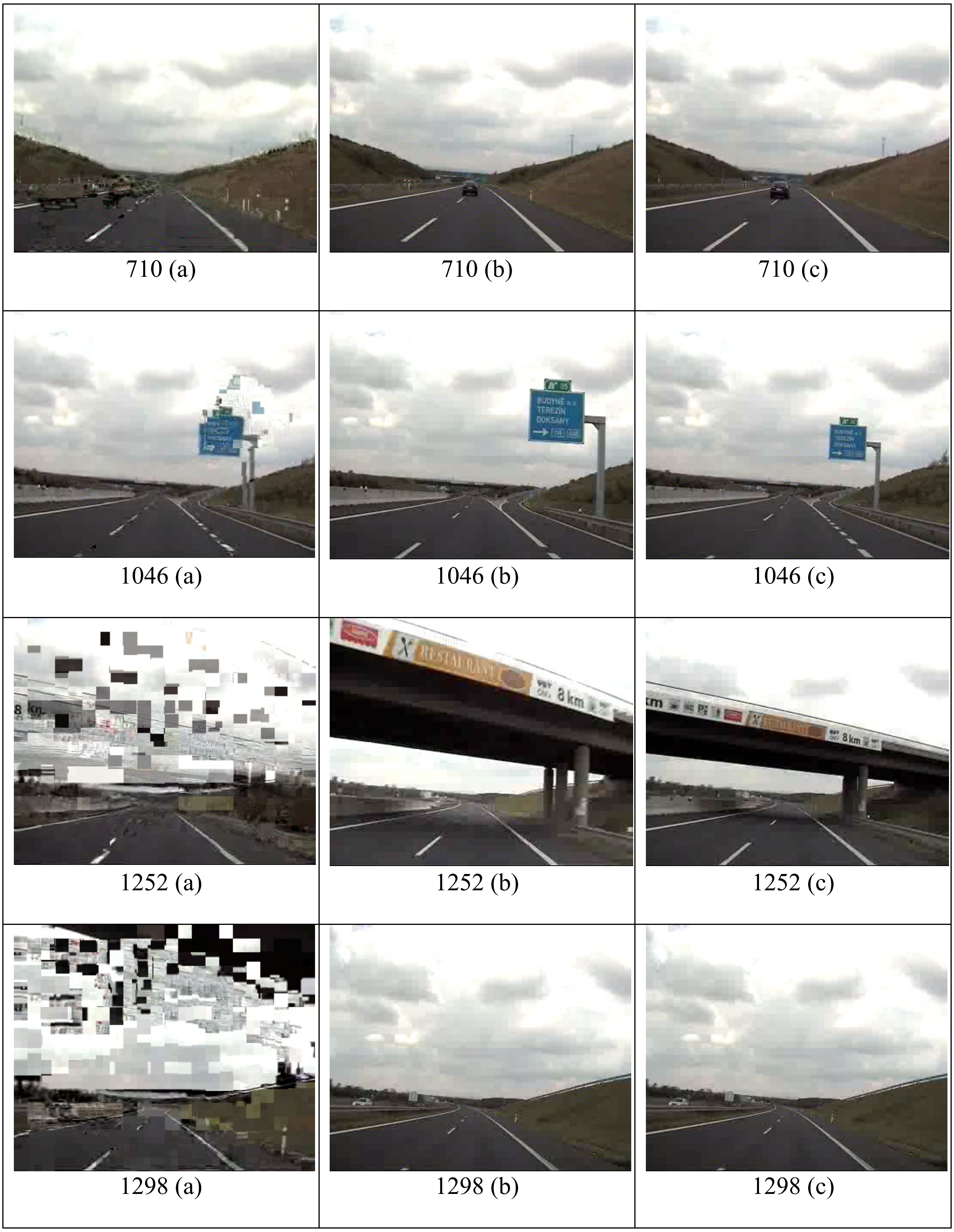


Figure 5.6: Frame comparisons between Default11e (a) and Q-RAPB (b) and Q-ROPB (c) queueing algorithm

In the figure above, 4 frames were taken as samples to compare the video quality subjectively. The frame samples were taken where the PSNR readings between the **Default11e**, **Q-RAPB** and **Q-ROPB** has significant differences. The frames involved are Frame 710, 1046, 1252 and 1298. In all four frames, a significant difference in terms of video quality can be observed. While the **Default11e** algorithm produced pixelated frames that cannot be viewed, both **Q-RAPB** and **Q-ROPB** offered better frame quality which is very much better.

5.3 Subjective Evaluation of Removing PktB on Queue Full Event

As the users are the final perceivers of video flows in multimedia communication, it is important to evaluate the QoE of video at the user level. To view the video as being perceived by the receivers, the Vi1 video flow in the above figure is reconstructed using the ETMP4 tool. This enables the video to be prepared for subjective evaluation. From the user-level QoE assessment, it is found that **Default11e** video deteriorates when the queue is becoming full.

The subjective assessment was conducted in a lab where 57 people (subjects) were involved. Each subject is shown the original video that is being streamed by the sender and to be compared with the reconstructed Default11e, Q-ROPB and Q-RAPB output video at the receiver. Subjects are asked to evaluate the video by classifying them into different categories. Five categories of video quality are used in the rating scale: “Bad” is assigned to integer 1. “Poor” to integer 2, “Fair” to integer 3, “Good” to integer 4 and finally “Excellent” to integer 5.

The method used to execute the subjective analysis is based on Degradation Category Rating (DCR) which was recommended in (ITU-T, 2016). This method has also been discussed previously in Section 2.3.1 where the treated video samples are compared to the untreated video samples. In this experiment, the untreated video is the original video that is being sent at the sender while the treated ones are those at the receiver after undergoing the algorithms of Default11e, Q-RAPB and Q-ROPB. From the evaluation, the results can be shown as in Table 5.5 below.

Table 5.5: Subjective Analysis Score of the Original, Default11e, Q-RAPB and Q-ROPB video

Original					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	27	24	3	3	0
Percentage	47.37	42.11	5.26	5.26	0.00
Avg Score	4.32				

(a)

Default11e					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	3	3	9	18	24
Percentage	5.26	5.26	15.79	31.58	42.11
Avg Score	2.00				

(b)

Q-RAPB					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	6	24	18	9	0
Percentage	10.53	42.11	31.58	15.79	0.00
Avg Score	3.47				

(c)

Q-ROPB					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	5	23	19	10	0
Percentage	8.77	40.35	33.33	17.54	0.00
Avg Score	3.40				

(d)

Table 5.5 (a), (b), (c) and (d) represents the score of the untreated, Default11e, Q-RAPB and Q-ROPB videos respectfully. From the above observations, it can be said that video that is being perceive by the receiver that undergone the Default11e algorithm is being categorised as having "Poor" in quality. Meanwhile Q-RAPB and Q-ROPB has scored better, which is an increase of 73.4% and 70% respectfully, against Default11e that brings the video quality between "Fair" and "Good".

The reason behind the scores above is because the proposed algorithms of Q-RAPB and Q-ROPB has rearranged the queue so that important packets of PktI is not being dropped when the queue at the sender is full, by dropping unimportant packets of PktB. Q-ROPB scored lower than Q-RAPB due to the algorithm can only remove a PktB that has the same AppId as the PktI that is being prioritized. Therefore, there will be moments where some of the PktI will be dropped even though there are PktB in the queue when the queue is congested. Those PktB cannot be removed in place of the PktI because they do not belong to the same AppId.

5.4 Discussions

In the previous sections, the results of **Q-RAPB** and **Q-ROPB** in comparison with **Default11e** are discussed. It is evident that the results of both Q-RAPB and Q-ROPB are quite similar although they are implementing a slightly different method which can be shown as in Figure 5.7, between **Q-RAPB** and **Q-ROPB** PSNR per frame for Vi1 in a 5ViFlowHighway scenario.

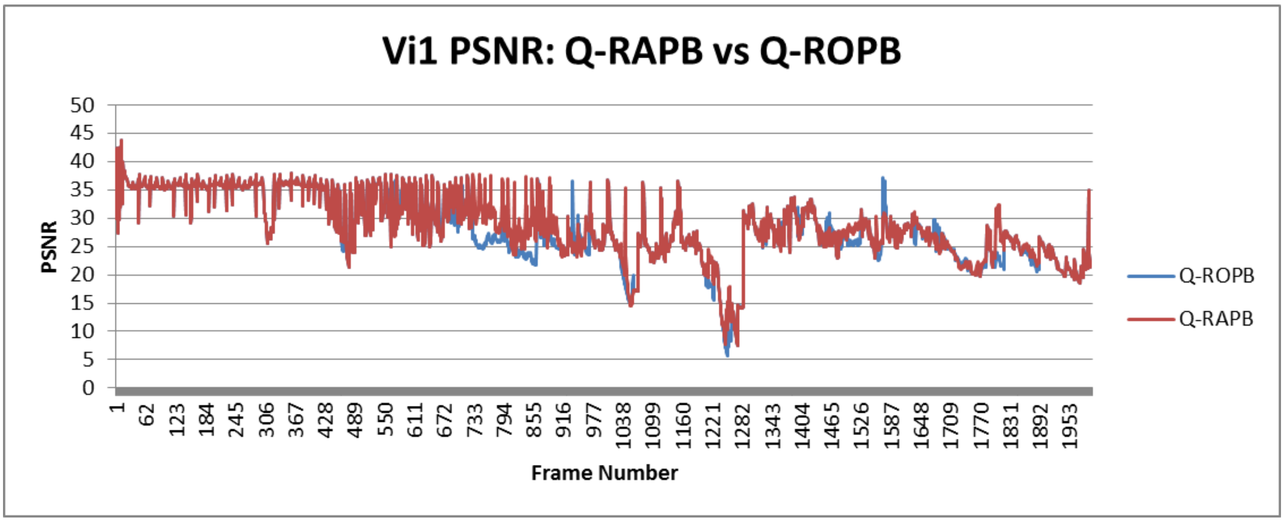


Figure 5.7: Per frame PSNR between Q-RAPB and Q-ROPB

The similarity in PSNR readings is due to the queue size. Since the queue size is relatively small, it affects the removal of PktB from the queue, where there is a very high possibility that there are no PktB in the queue to be removed. As the size of queue increases, the occurrences of packets being dropped due to overflow became lesser, thus the proposed algorithms were not activated. The longer queue acts as a buffer, so the arrival rate is higher than the bottleneck link, thus the loss rate would remain the same. This is evident for the video intervals where the PSNR readings for the proposed algorithm are similar.

Meanwhile, it is also expected that the **Q-ROPB** will have a slightly lower PSNR than **Q-RAPB**. This is because **Q-ROPB** needs a specific type of PktB to be removed whereas **Q-RAPB** can remove any PktB in the queue. This means that, in **Q-ROPB**, there is a chance that a PktI will be dropped even though there is a PktB where the PktB is not from the same AppId as PktI. A PktI will be dropped and a PktB from the queue will be kept, although PktI have a higher precedence than PktB.

5.5 Objective Results of Removing PktB with PSNR Trigger

The main objective of this thesis is to maintain an acceptable level of QoE for the video flows in the network. In the previous section, PktB is removed from the queue to accommodate PktI in case the queue is full. This is done regardless of the PSNR readings. Although it is effective, both Q-RAPB and Q-ROPB do not have a level of PSNR to maintain and there is no point of removing PktB if all the video in the flows has already reached an acceptable level of PSNR.

To make the algorithm more effective, a new feature is introduced. Instead of waiting for the queue to be full before the algorithm is being applied, the new proposed method will remove PktB whenever the PSNR falls below a certain threshold. These new algorithms are named P-

ROPB and P-RAPB which has been discussed in detail previously in Section 3.3.2. In order to do this, the proposed algorithm will need to predict the perceived PSNR before it can start removing PktB from the queue.

Relating to P-RAPB and P-ROPB, one of the preliminary experiments done, discussed in Section 3.4.2 is to develop a method to predict the perceived PSNR readings at the sender. The results of the predicted PSNR was then tested against the actual PSNR at the receiver's end and results showed that the predicted PSNR follows closely to that of the receiver's actual PSNR with a standard deviation of **2.38**. In terms of the PSNR scale, it only deviates about **4.76%** from the actual perceived PSNR which means, the predicted PSNR does not vary much from the actual PSNR. This method is then used to predict the actual PSNR in P-RAPB and P-ROPB to do PSNR prediction.

In this section, both **P-RAPB** and **P-ROPB** are discussed together, as they are both related to be analysed. Both of them are PSNR threshold-based, but the latter only removes PktB packets that belong to the same flow as the PktI. Both algorithms are configured to have a threshold of **PSNR=30**, meaning that the algorithm will keep running while the predicted PSNR is below 30.

Implementing **P-RAPB** in the NS-3 simulator has the pseudo code as below:

```
//PktI:  Packets that carry I-Frame fragment
//PktB:  Packets that carry B-Frame fragment
//Threshold is configured to a level deemed acceptable by end users

If incoming packet is PktI AND Predicted_PSNR < Threshold
       Then remove any PktB from the queue
       Enqueue the incoming PktI
```

Meanwhile, implementing **P-ROPB** has the following pseudocode as shown below:

```
//PktI:  Packets that carry I-Frame fragment
//PktB:  Packets that carry B-Frame fragment
//Threshold is configured to a level deemed acceptable by end users

If incoming packet is PktI AND Predicted_PSNR < Threshold
       Then remove PktB from the queue which has the same flow as PktI
       Enqueue the incoming PktI
```

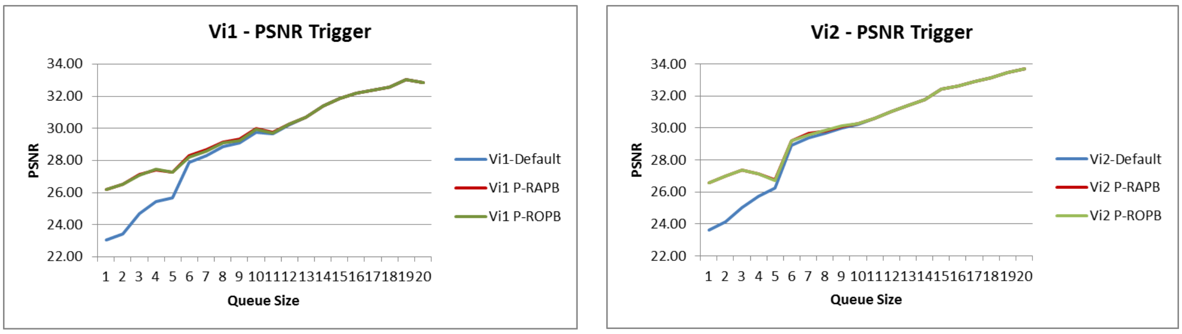
When testing both variants of the algorithm, the queue size was configured from 1 to 50 to determine the impact that congestion and queue length have on the resulting performance. After the experiment, a significant increase of collective PSNR between the flows can be observed when compared against the Default11e queueing. All video flows that implement the **P-RAPB** and **P-ROPB** showed an increase of PSNR in readings especially when the queue is small (less than 10 slots) and congested. Table 5.6 below is an example of the average PSNR improvements within different queueing mechanism where queue size is between 1 and 10. The average PSNR is calculated based on the PSNR readings for every flow in each queueing algorithm.

Table 5.6: Video PSNR of **Default11e**, **P-RAPB** and **P-ROPB** where queue is between 1 and 10, and Threshold equal to 30

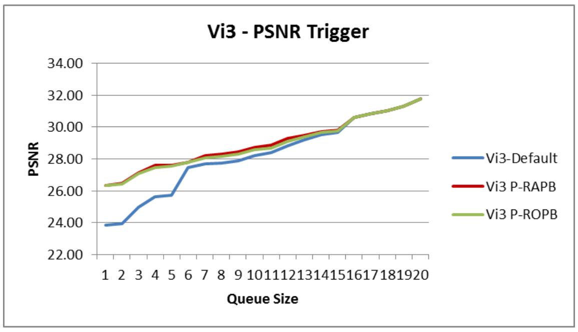
PSNR			
Queue Size	**Default11e**	**P-RAPB**	**P-ROPB**
1	23.27	26.31	26.30
2	23.90	26.77	26.76
3	25.31	27.33	27.32
4	25.81	27.34	27.32
5	26.16	27.26	27.24
6	28.48	28.68	28.66
7	29.00	29.21	29.19
8	29.43	29.62	29.63
9	29.78	29.94	29.92
10	30.23	30.35	30.36

Evidently, the proposed algorithm worked significantly on increasing the PSNR readings for both **P-RAPB** and **P-ROPB** up to a certain level. **P-RAPB** and **P-ROPB** work best when the queue size is small (queue congestion is high). On an extreme level, at queue size 1, the improvement of PSNR can be seen to increase **12.42%** in scenario applying the **P-RAPB** and **P-ROPB** in comparison with the **Default11e** algorithm. Certain scenarios which may lead to congestion similar to this case, where the queue have only one slot left to accommodate a packet. This brings the quality of the video from "Poor" to "Fair" on the video quality scale as in Table 2.2. However, as the queue size increases, the PSNR readings between these three algorithms become quite similar as the queue size approaches 10. This is because, more slots are available in the queue to accommodate packets thus become less congested.

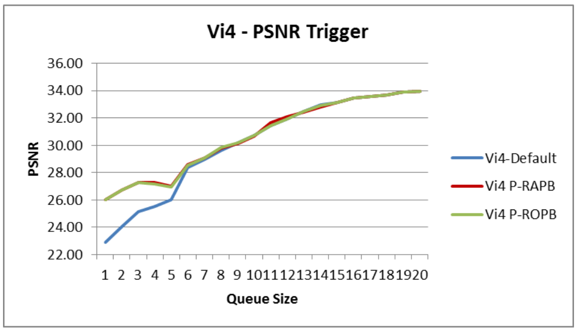
To look closely on how the algorithm affects each flow in the simulation scenario, a graph for each video flow is drawn. This involves plotting the **Default11e** and two of the proposed algorithms, **P-RAPB** and **P-ROPB** in the same graph to see how the PSNR improves. This can be depicted as in Figure 5.8 below.



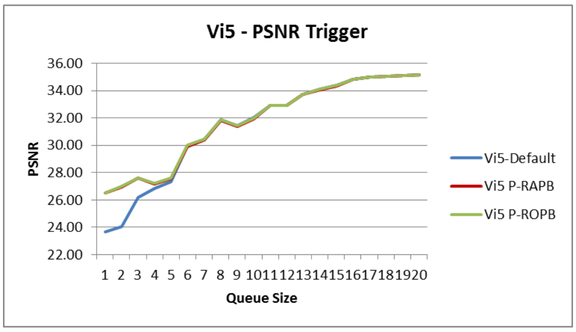
(a) (b)



(c)



(d)



(e)

Figure 5.8: Breakdown of each PSNR for every flow in scenario 5ViFlow (Default11e, P-RAPB and P-ROPB)

The above figure is an example of a flow Vi1 to Vi5 being simulated with both methods of algorithm where PSNR threshold is the trigger to activate the algorithm. In the simulation, queue size is changed from as low as 1 to 50. This is to see how queue congestion will affect the packet queueing and thus affect the predicted and perceived PSNR. However, for queue sizes beyond 10, no significant different of PSNR can be observed. This is because the size of the queue is now big enough to prevent packet loss hence, the algorithm is not executed.

Though there are differences in the PSNR readings between the algorithms, it is important to measure the significance of the proposed algorithms. On a surface level, the improvements can be seen to improve the PSNR. Similar to Q-RAPB and Q-ROPB, both P-RAPB and P-

ROPB are being analysed through both the normality (to check whether data is normally distributed) and significance test.

Table 5.7: Normality test for Default, P-RAPB and P-ROPB

Tests of Normality						
	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Default_PSNR	.138	20	.200*	.950	20	.371
PRAPB_PSNR	.231	20	.006	.826	20	.002
PROPB_PSNR	.245	20	.003	.795	20	.001
*. This is a lower bound of the true significance.						
a. Lilliefors Significance Correction						

The normality test above shows that both P-RAPB and P-ROPB to have a significant value of < 0.05 on the Shapiro-Wilk table. This means both sets of readings do not have a normally distributed data. Wilcoxon Rank-Sum significance test is used to test the significance of the proposed algorithm of P-RAPB and P-ROPB.

Table 5.8: Wilcoxon Rank-Sum Tests Output Tables

Descriptive Statistics					
	N	Mean	Std. Deviation	Minimum	Maximum
Default_PSNR	20	25.5605	.94389	24.09	27.22
PRAPB_PSNR	20	27.1065	1.06366	23.57	28.74
PROPB_PSNR	20	26.9640	1.09485	24.67	28.81

(a) Descriptive analysis output

Ranks				
		N	Mean Rank	Sum of Ranks
PRAPB_PSNR - Default_PSNR	Negative Ranks	2[a]	7.00	14.00
	Positive Ranks	18[b]	10.89	196.00
	Ties	0[c]		
	Total	20		
PROPB_PSNR - Default_PSNR	Negative Ranks	2[d]	8.50	17.00
	Positive Ranks	18[e]	10.72	193.00
	Ties	0[f]		
	Total	20		

a. PRAPB_PSNR < Default_PSNR

b. PRAPB_PSNR > Default_PSNR

c. PRAPB_PSNR = Default_PSNR

d. PROPB_PSNR < Default_PSNR

e. PROPB_PSNR > Default_PSNR

f. PROPB_PSNR = Default_PSNR

(b) Signed Ranks Test

Test Statistics[a]		
	PRAPB_PSNR - Default_PSNR	PROPB_PSNR - Default_PSNR
Z	-3.397[b]	-3.285[b]
Asymp. Sig. (2-tailed)	.001	.001

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

(c) Test statistics

Table 5.8 (b) showed that 18 of the readings are positive ranks for both P-RAPB and P-ROPB. This means 80% of the proposed algorithm gave better results compared to the readings without them. This is further proven through the test statistics in Table 5.8 (c). Since this is a multiple comparison, Bonferroni adjustment is applied to the test statistics. Two tests were conducted and therefore the significance level is divided with the number of test (2) where the new significance level is 0.025. Significant value for both P-RAPB and P-ROPB are 0.01

respectively which is < 0.025 showed that the proposed algorithms have significant improvements in comparisons toward Default.

5.6 Visual Results of Removing PktB with Predicted PSNR (PP) Event

In the previous section, data analysis from a statistical point of view has been observed and analysed. It is it is evident that the proposed algorithms had given significant improvements which were tested through the t-Test.

In this section, the subjective video quality for each frame which applies the **Default11e**, **P-RAPB** and **P-ROPB** algorithm is viewed and analysed. In the example of Vi1 in Figure 5.9 below, comparisons between the **Default11e** and **P-RAPB** is discussed.

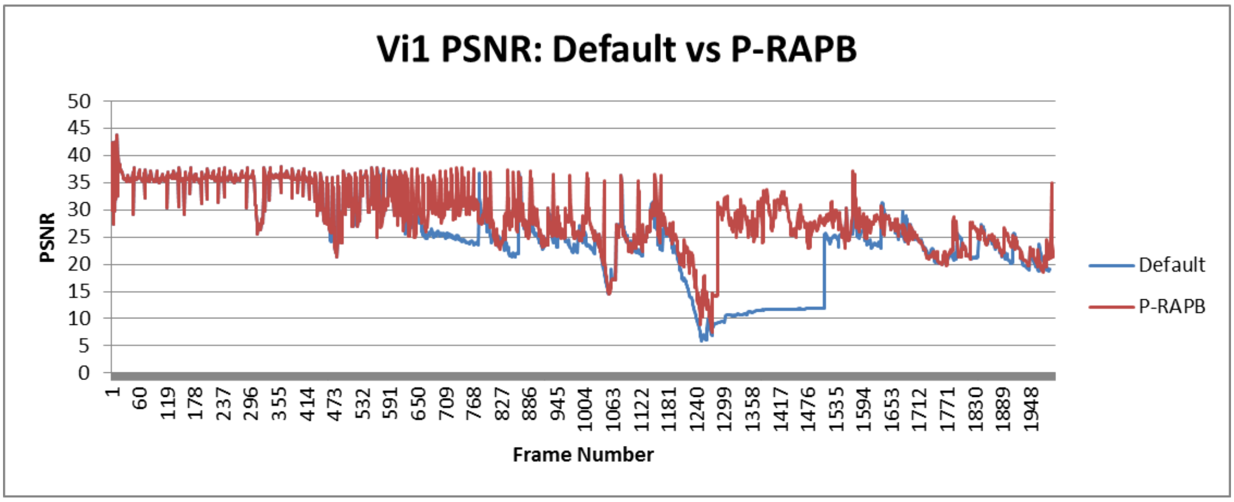


Figure 5.9: PSNR comparisons per frame between Default and P-RAPB

From the figure above, the PSNR for **Default11e** and **P-RAPB** is **25.49** and **27.27** respectively. Both algorithms seem to have a similar PSNR. However, in **Default11e** algorithm, the low

PSNR readings continue to run until frame 1529 while **P-RAPB** recovers from the PSNR dip at frame 1343.

Meanwhile Figure 5.10 below is the comparison of PSNR frame between **Default11e** and **P-ROPB**. Though difference can be seen, it is actually very similar to **P-RAPB**, with a PSNR of **27.24**.

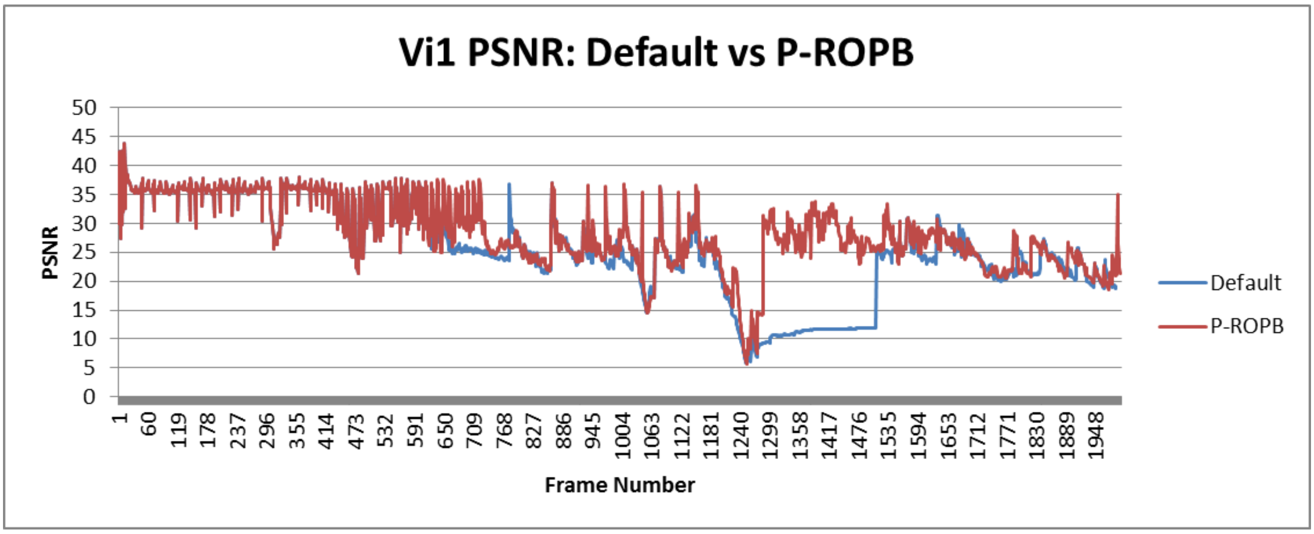


Figure 5.10: PSNR comparisons per frame between Default and **P-ROPB**

Figure 5.11 below gives an overview of how the damaged frames from the **Default11e** algorithm look like in comparisons to the same frame where **P-RAPB** and **P-ROPB** were able to maintain the PSNR at a higher level. The **Default11e**, P-**RAPB** and **P-ROPB** algorithm are denoted as (a), (b) and (c) respectively.

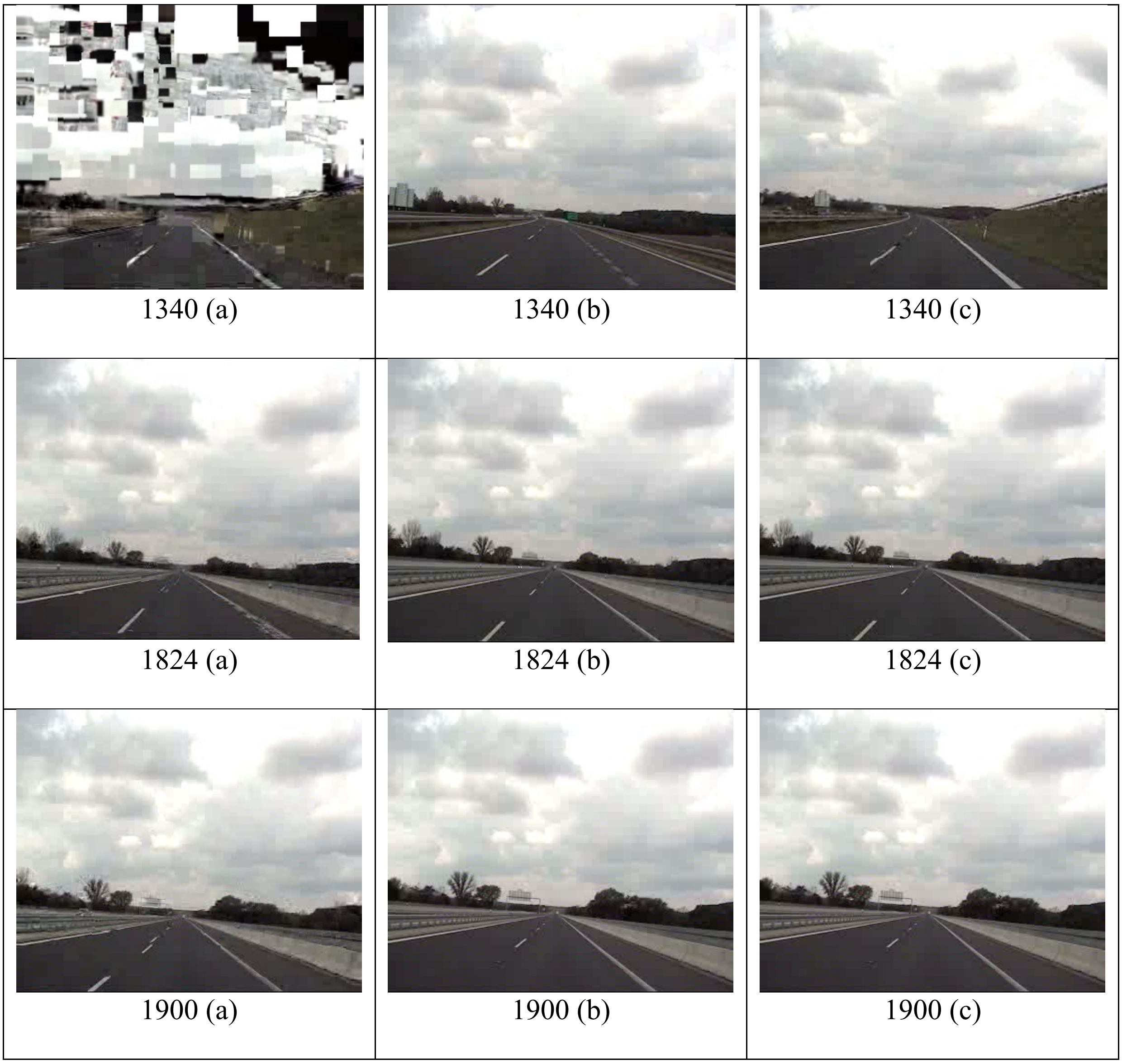


Figure 5.11: Frame comparisons between Default11e (a) and P-RAPB (b) and P-ROPB (c) queueing algorithm

On the overall view, P-RAPB and P-ROPB have the same results, although they both use a slightly different algorithm. This is mainly due to the low probability of removing a PktB from the queue due to the main factor of the queue size. With a queue size of 5, there is a high chance that there are no PktB in the queue to be removed or in the case of P-ROPB, there are no PktB of its own flow to be removed.

In a PSNR reading, the maximum score for a PSNR is 50, which is the highest level of video quality. Throughout the experiments where PSNR is set as the trigger to remove PktB, the thresholds are set to 30 where the PSNR is partially compromised. It is worth to note that the threshold is flexible where it is adjustable to suit the quality that a sender wanted to guarantee. However, the full working scale of the algorithm is hard to be seen when the video quality is set to be partly compromised i.e. in this case, threshold is set to 30. In a scenario where the algorithm is always enabled (threshold = 50), the algorithm can clearly be seen to work its way to achieve the highest PSNR possible. P-RAPB (50) refers to the same P-RAPB algorithm but with a threshold of 50.

It is important to note that the highest achievable PSNR of the video used in P-RAPB(50) is approximately 35. This is due to the encoding process which has been done prior to streaming the video and introduced noise to it. Referring to Figure 5.12 above, it can be seen that P-RAPB(50) is constantly trying to achieve the maximum achievable PSNR and as a result, achieved the highest possible PSNR possible in the experiments where P-RAPB(50) gives a PSNR of 29.23. While, the PSNR threshold is set to the maximum unachievable value in P-RAPB(50), the algorithm always runs. In comparison to Q-RAPB and Q-ROPB, the algorithm is not triggered by queue size, therefore it does not need to wait for the queue size to be full before the algorithm is being triggered.

In comparison of all three algorithm, it is apparent that P-RAPB(50) has a more consistent pattern of guaranteeing the highest PSNR. The Default11e, P-RAPB and P-RAPB(50) resulted to PSNR readings of 25.49, 28.74 and 29.23 respectively, resulting in improvements of 12.75% and 14.67% for P-RAPB and P-RAPB(50) respectively.

5.7 Subjective Evaluation of Removing PktB on Predicted PSNR (PP) Event

Subjective evaluation were also conducted for P-RAPB and P-ROPB using the same setup and method as Q-RAPB and Q-ROPB in Section 5.3. 57 people (subjects) are asked to compare the video quality as being perceived by the receiver, between video from Default11e, P-RAPB and P-ROPB based on the DCR method recommended by (ITU-T, 2016). Subjects evaluated the quality of the perceived video based from a scale of 1 (Poor) to 5 (Excellent). Findings of the subjective evaluation can be shown as in Table 5.9 below.

Table 5.9: Subjective Analysis Score of the Original, Default11e, P-RAPB and P-ROPB video

Original					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	27	24	3	3	0
Percentage	47.37	42.11	5.26	5.26	0.00
Avg Score	4.32				

(a)

Default11e					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	3	3	9	18	24
Percentage	5.26	5.26	15.79	31.58	42.11
Avg Score	2.00				

(b)

P-RAPB					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	8	21	19	9	0
Percentage	14.04	36.84	33.33	15.79	0.00
Avg Score	3.49				

(c)

P-ROPB					
Class	5 (Excellent)	4 (Good)	3 (Fair)	2 (Poor)	1 (Bad)
Count	7	20	20	10	0
Percentage	12.28	35.09	35.09	17.54	0.00
Avg Score	3.42				

(d)

Table 5.9 (a), (b), (c) and (d) represents the score of the untreated, Default11e, P-RAPB and P-ROPB videos respectfully. As with the QF-based trigger algorithms, PP-based trigger algorithm has also shown significant improvements. The data showed that the video quality score increased approximately 74.5% and 71% for P-RAPB and P-ROPB respectfully. P-RAPB showed a slightly better reading than P-ROPB as the former algorithm is free to remove any PktB, should the PSNR level drop below the desired threshold. Thus, it has more PktB to remove, thus accommodating more important packet, PktI.

5.8 Discussions and conclusion

In this chapter, several algorithms have been analysed. This chapter evaluated the proposed algorithms to offer better QoE using comprehensive set of simulations based on scenarios described in 4.3.2. Algorithms that are congestion-triggered, or which has been referred to as Queue Full (QF) event and PSNR-triggered, referred as Predicted PSNR (PP) event algorithm has been compared against the **Default11e** algorithm, which is the standard queueing process done in the IEEE 802.11e.

With the algorithm that is queue congestion-triggered (**Q-RAPB** and **Q-ROPB**), they allow the queue to be fully utilised before removing any PktB to enhance the PSNR at critical situations. This avoids video from being jumpy because when the packets are being allowed to stack up in the queue, there is a higher chance that there are lots of PktB to be removed. PktBs in the queue are removed one by one which means some of the PktB are allowed to be in the queue and actually being transferred to the receiver. This allows some PktB to fill in the gaps between PktI and PktP and make the video smoother.

In comparison with a PSNR-triggered algorithm (**P-RAPB** and **P-ROPB**), it is strictly based on the level of guarantee that has been set by the sender. The prioritisation of traffic relies on the perceived quality, regardless of the queue status. Thus, this algorithm does not wait for the queue to be full. On a side note, this also means it will have less PktB in queue to be removed when the algorithm is triggered. However, since the PSNR threshold can be configured, it is more flexible depending on the video quality the sender wants to guarantee.

To take the work done by Rodrigues et. Al (2008) in perspective, the I-frame loss rate is better due to the expense of non-video traffic. This is because non-video packets are being discarded in favour of video packets since the queue is being shared between both video and non-video packets. In this thesis, the queue for different types of traffic in EDCA are separate by default and the focus is only on the VI (video) queue. Rodrigues et. al (2008) also ran the simulation in a wired network and therefore it has more control and protection of the packets. In contrast, the proposed mechanism runs on wireless network that is subject to packet losses that are not necessarily due to congestion. The method applied in IEEE 802.11e is not available on Rodrigues's work and therefore the concept is the same, but the environment is different. The proposed mechanism is also more robust and therefore simpler, though it might not catch all the intricacies. It also does not touch other application traffic (non-video) and therefore the amount of fixing is limited. In addition, the proposed mechanism also keeps on recalculating the PSNR throughout the process of video streaming to ensure better video quality.

Though the proposed mechanism is subject to wireless stations only and the impact of the environment and may require successive update and changes, nevertheless it provides a level of estimate for the resulting performance and improved the overall video quality.

CHAPTER 6: CONCLUSION AND FUTURE DEVELOPMENT

6 Introduction

This thesis aims to improve the Quality of Experience of video streaming in an infrastructure-based wireless network of IEEE 802.11e by proposing novel solutions to improving the transmission of video in congested network environments. The resources include queue size and bandwidth where being limited, they resulted to packets being dropped which include the important video packets. To overcome these challenges, the proposed solutions tried to utilize the available resources as efficient as possible by looking at the frame type to positively identify and differentiate between packets and prioritise the traffic, which will affect the quality of the video positively at the receiving end. This will help to maintain a high level of QoE. The proposed solutions are triggered either by video quality (PSNR) or network status (congestion).

This thesis has presented novel schemes on selectively dropping less important video packets. In MPEG-4 transmission, video packets that are less important are the ones that carry the B-Frame information. This thesis proposed two methods of removing the B-Frame packets; 1) removing B-Frame packets that belongs to a video flow that is affected by the network congestion (ROPB) and 2) removing any B-Frame regardless of the video flow (RAPB).

Dropping the B-Frames is done based on two different scenario triggers. The first trigger is Queue Full (QF). In this scenario, the proposed algorithm monitors the queue status. Once the algorithm detects that the queue is full, the removal of B-Frame packets is done to avoid I-Frame packets being dropped due to a full queue. Meanwhile the second trigger where B-Frame packets are selectively dropped is when the algorithm detects that the PSNR level of the receiver drops to a threshold below the desired level. In this thesis, the desired PSNR threshold is configured to 30dB. This is because PSNR levels below 30dB is not considered as of "good" quality (Padle and Mendre, 2014).

6.1 Conclusions

This thesis aims to enhance the EDCA queueing in IEEE 802.11e by proposing two main solutions to support better video quality in a wireless network scenario with limited resources. Two main solutions were proposed where the first solution is based on queue congestion and the second one is QoE driven where PSNR is used as the main trigger to guarantee a certain level of PSNR. Results from the analysis from t-Tests showed that both solutions gave significant improvements. Moreover, the results from PSNR readings and actual video frame visualisation also showed significant improvements where the actual perceived video on the receiver's side is better and more viewable.

The purpose of this thesis is to address the issue of important video packets being dropped within the MAC layer and propose mechanism to avoid these packets being dropped especially during network congestion. Based on the analysis and results, it can be shown that the mechanisms proposed in the thesis improved the transmission of video in wireless networks without impacting the performance of the other applications. The concept of removing packets until quality is stabilized around a predefined threshold allows ensuring the quality of the received video and at the same time minimizing the amount of traffic that is running through the network.

To facilitate the proposed mechanism to ensure a predefine video quality threshold, a mechanism to estimate receiver's PSNR was developed. Previous studies have lacked the studies on estimating the PSNR on the receiver's side without getting direct feedback from the receiver. In this thesis, the PSNR estimation technique was developed and incorporated in the proposed mechanism. With this estimation calculation, the sender will have the capability to get the video quality status that was currently being streamed as perceived by the receiver. The estimation mechanism proved to be crucial to ensure the proposed mechanism would to be activated at the correct moment where the PSNR starts to degrade. Significant improvements of PSNR were demonstrated as shown in Section 5.5.

6.2 Challenges and Limitations

The proposed mechanism to support better QoE in wireless networks encountered several challenges as described below:

1. **TXOP**: TXOP is currently not being supported in NS-3 as of the current version. Currently, it is assumed that the stations that successfully get the access to the channel will be given only 1 packet to transmit, and not based on time duration as in the IEEE 802.11e standard. Therefore, results may differ from simulations to real networks.

2. **Video content type**: In this research, a technique was used in the preliminary experiment to detect and differentiate the video content types of video. Several distinctive patterns have been discovered but further work needs to be done to integrate the findings in the proposed algorithm. Identifying a video content type is important where it will determine on

how extreme a video can tolerate PktI drops. This is important especially in a network with a mixed of different types of video content.

3. **PktI location**: Determine whether PktI is in the SeqId=1 position. This is important as according to the findings of the experiment, PktI with SeqId=1 has a high weightage in determining the PSNR of a video.

4. **Packet timeout in MAC Queue**: In the experiments conducted, packet timeout in MAC queue is not considered. This is due to limiting the parameters so that the experiment can be controlled and the results on observing the parameters being tested can be recorded accurately.

5. **Video types**: In this thesis, only MP4 video has been considered. However, there are other types of video and technologies that are being implemented in today's network such as Scalable Video Coding (SVC) and Dynamic Adaptive Streaming over HTTP (DASH). These video also are made up of different layers of video priority. For example, SVC is made up from a Base Layer and Scalable Extension layer. Theoretically the same solution can be applied when the network resources become limited where the extension layer should be eliminated to prioritize the base layer.

6.3 Future Work

During the course of completing this thesis, experiments have been done to provide better QoE for the receiving end of the video stream. However, there are findings that are worth to be investigated further as it will contribute towards offering better QoE in video streaming. The issues are as follows:

I-Frame placement: One of the findings in this thesis is that the location of packet drops determines the quality of the video. For example, the very first packet (SeqId=1) that carries the I-Frame of a video flow is very crucial that if this packet is dropped, the PSNR can drop as high as **19.88%**. Secondly, first packet that carries information of the first I-Frame of a GoP also gives an impact towards video quality. Thirdly, the weight of an I-Frame towards affecting the PSNR also depends on the scene of the video. For example, an I-Frame drop in a slow-moving scene does not give a big impact as in a slow-moving scene. To summarize, the location of the packet drop within a GoP determines how a video quality is affected. To address this issue, a detailed model on determining whether the packets belong to the same GoP should be developed. This can be shown as in Figure 6.1 below.

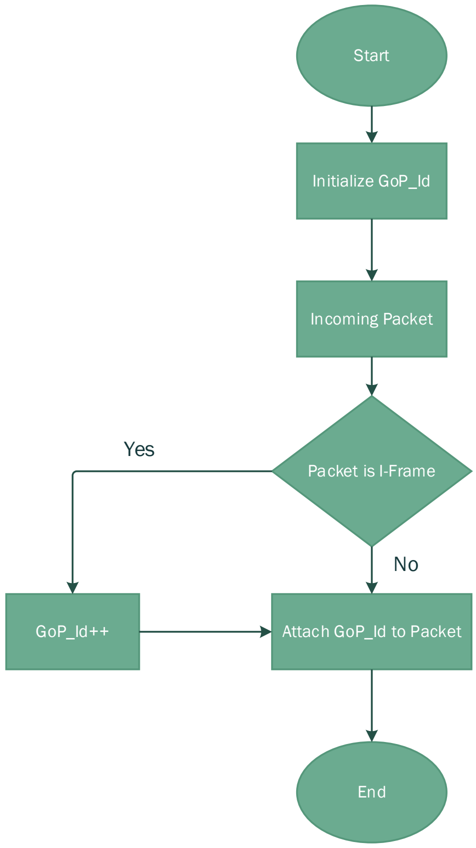


Figure 6.1: Proposed algorithm on tracking the packet's GoP Id

From the flowchart above, the first PktI that comes after a non PktI packet is the marker of a new GoP. However, there will be a problem if several packet losses that spans between two different GoPs occurs, especially when the second GoP lost all its PktI in the process. This can be shown as in Figure 6.2 below.

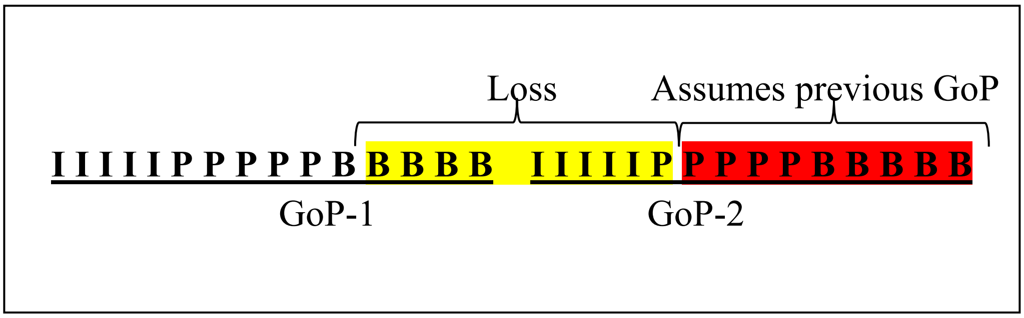


Figure 6.2: Lost packets between two GoP leads to GoP numbering confusion

From the above figure, the packet loss is highlighted in yellow where it spans between two different GoPs. Since the loss also involves all the I-Frames of GoP-2 (no PktI to mark a new GoP), the above algorithm cannot distinguish that the following P-Frames in the GoP-2 belongs to a different GoP than GoP-1. To avoid this problem, the algorithm should be implemented at the APP level where loss is very minimal.

PktI loss aware algorithm: Another future work that can be developed is for the sender to be aware of the percentage of PktI that has been dropped at the queue and being able to respond upon it. At a certain threshold, a sequence of dropped PktI in a GoP will affect the whole GoP where the frame of the video is not possible to be viewed or reconstructed. If that certain threshold is reached, it is a better idea to drop the subsequent PktP and PktB that comes after the PktI because PktP and PktB are dependent to PktI which are already lost. Queueing the said PktP and PktB just take up queue space and bandwidth as there are no point of transmitting them. Even if PktP and PktB were successfully transmitted over the network, the video frames are still cannot be reconstructed due to the PktI loss. Although the proposed algorithm will try to protect PktI from being dropped, it is impossible to do so at all times. For example, due to queue congestion, it is inevitable that PktI may be dropped from the queue.

Queue Timeout (TO) issue: In the previous experiments done in this research, the timeout (TO) of a packet in a queue is neglected. This means there is no expiry time for a packet to be in a queue. TO is omitted in the experiment scenario to control the experiment from external factors that cause the packets in the queue to drop other than the queue being full. However, in a realistic network that is not the case. A technique that can be implemented to avoid PktI from being dropped due to queue timeout is to put PktI to the front of the queue if the TO is near. By doing this, theoretically it gives a high chance of preventing PktI from be dropped. This is important especially in a very low bandwidth wireless network where in these scenarios TO contribute to most of the packet drop. There is no use of removing PktB to insert PktI where in the end the PktI that was inserted will be dropped due to TO. This idea is tested but the improvement of PSNR is quite low. However, in terms of the percentage of PktI being dropped, this technique has successfully prevented PktI from being dropped from the queue. In one of the side experiments done, this technique has been tested in 10ViFlow scenario in conjunction with **P-RAPB** and **P-ROPB**, where the algorithms are named **P-RAPB**-I and **P-ROPB**-I. To be consistent, Vi1 has been taken into example again where in **P-RAPB** and **P-ROPB**, the PSNR are charted at **27.94** and **27.90** respectively while on the other hand, **P-RAPB**-I and **P-ROPB**-I marked **28.24** and **28.56** of PSNR respectively.

Content aware algorithm: In Section 3.4.1, one of the preliminary experiments done is the idea of the possibility of the system to automatically be aware of the video content, whether it is a rapid, moderate, or slow-moving video. Although the experiments have proved that it is possible to do so based on the ratio of the I, P and B-Frames, more studies are needed until the self-aware element can really be incorporated in a novel system. This includes the ability of the system to be aware of a video that can contain multiple types of video in a single stream.

REFERENCES

Abdel Khalek, A., Caramanis, C. and Heath, R.W. 2012. A Cross-Layer Design for Perceptual Optimization Of H.264/SVC with Unequal Error Protection. *IEEE Journal on Selected Areas in Communications*. [Online]. **30**(7), pp.1157–1171.

Abu-khadrah, A., Zakaria, Z., Othman, M. and Teknikal, U. 2014. Evaluate QOS parameters for VOIP using IEEE 802.11 (DCF) and IEEE 802.11e (EDCA). *Australian Journal of Basic and Applied Sciences*. **8**(January), pp.265–272.

Acharya, P.R., Vityanathan, V. and Chellaih, P.R. 2010. WLAN QoS Issues and IEEE 802 11e QoS Enhancement. *International Journal of Computer Theory and Engineering*. **2**(1), pp.143–149.

Ali, N.A., Bourawy, A. and Hassanein, H. 2013. Selectivity function scheduler for IEEE 802.11e HCCA access mode. *Wireless Communications and Mobile Computing*. **13**(2), pp.95–110.

Anastasi, G. and Lenzini, L. 2000. QoS Provided by the IEEE 802.11 Wireless LAN to Advanced Data Applications : A Simulation Analysis. *Wireless Networks*. **6**, pp.99–108.

Anitha, A. and Jayakumari, J. 2014. Performance Analysis of WLAN Under Variable Number of Nodes Using the Adjustable Parameters in EDCA. *Journal of Theoretical and Applied Information Technology*. **60**(2), pp.351–357.

Arabi, M., Ghita, B. and Wang, X. 2010. Improving Fairness in Ad Hoc Networks through Collision Rate Control *In*: *International Network Conference*., pp.51–59.

Baraković, S. and Skorin-Kapov, L. 2013. Survey and challenges of qoe management issues in wireless networks. *Journal of Computer Networks and Communications*. **2013**.

Bhamidipati, V.D. and Kilari, S. 2010. *Effect of Delay/ Delay Variable on QoE in Video Streaming*. Blekinge Institute of technology.

Bo Fu, Kunzmann, G., Corujo, D., Wetterwald, M. and Costa, R. 2013. QoE-aware traffic management for mobile video delivery *In*: *2013 IEEE International Conference on Communications Workshops (ICC)*. IEEE, pp.652–656.

Braden, R., Clark, D. and Shenker, S.. 1994. RFC1633: Integrated Services in the Internet Architecture: an Overview. *IETF RFC 1633, July*, pp.1–28.

Carneiro, G., Fortuna, P. and Ricardo, M. 2009. FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3). *Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools*. p.10.

Cecchetti, G., Ruscelli, A.L., Mastropaolo, A. and Lipari, G. 2012. Providing variable TXOP for IEEE 802.11e HCCA real-time networks *In*: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, pp.1508–1513.

Charfi, E., Chaari, L. and Kamoun, L. 2013. PHY/MAC enhancements and qos mechanisms for very high throughput WLANs: A survey. *IEEE Communications Surveys and Tutorials*. **15**(4), pp.1714–1735.

Cisco 2011. *Cisco Visual Networking Index: Forecast and Methodology, 2010-2015*

Cisco 2012. *Cisco Visual Networking Index: Forecast and Methodology, 2011-2016*

Cisco 2013. *Cisco Visual Networking Index: Forecast and Methodology, 2012 - 2017*.

Cisco 2014. *Cisco Visual Networking Index: Forecast and Methodology, 2013 - 2018*.

Cisco 2015. *Cisco Visual Networking Index: Forecast and Methodology, 2014 - 2019*.

Cisco 2016. *Cisco Visual Networking Index: Forecast and Methodology, 2015 - 2020*.

Cisco 2017. *Cisco Visual Networking Index: Forecast and Methodology, 2016 - 2021*.

Debnath, T. 2012. *A Novel QoS-Aware MPEG-4 Video Delivery Algorithm over the Lossy IEEE 802 . 11 WLANs to Improve the Video Quality*. Dublin Institute of technology.

Fallah, Y.P. and Alnuweiri, H. 2005. A controlled-access scheduling mechanism for QoS provisioning in IEEE 802.11e wireless LANs *In*: *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*

- *Q2SWinet '05*. New York, USA: ACM Press, p.122.

IEEE Std. 802.11aa - 2012 2012. Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2 : MAC Enhancements for Robust Audio Video Streaming.

IEEE Std. 802.11e-2005 2005. Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications, amendment 8: medium access control (MAC) quality of service enhancements (2005). pp.1–211.

IEEE Std 802.11a - 1999 (R2003) 2003. Supplement to IEEE Standard for Information technology. Telecommunications and information exchange between systems. Local and metropolitan area networks. Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Sp.

IEEE Std 802.11b - 1999 (R2003) 2003. Supplement to IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY). **1999**.

IEEE Std 802.11g - 2003 2003. IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification.

IEEE Std 802.11n - 2009 2009. IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements.

IEEE Std 802.1D 2004. IEEE Standard for Local and Metropolitan Area networks: Media Access Control (MAC) Bridges. **2004**.

IEEE Std 802.1q - 2011 (Revision of IEEE Std 802.1q-2005) 2011. IEEE Standard for Local and Metropolitan Area Networks – Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks.

International Communication Union 1994. *Terms and Definitions Related to Quality of Service and Network Performance Including Dependability*.

ITU-T 2016. *Methods for the Subjective Assessment of Video Quality, Audio Quality and*

Audiovisual Quality of Internet Video and Distribution Quality Television in any Environment.

ITU-T 2008. *Vocabulary for performance and quality of service. Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10/G.100*.

Jain, R. 2003. Media vision - Experience isn't only screen deep. *IEEE Multimedia*. **10**(1), pp.80–81.

Jansang, A. and Phonphoem, A. 2011. Adjustable TXOP mechanism for supporting video transmission in IEEE 802.11e HCCA. *Eurasip Journal on Wireless Communications and Networking*. **2011**, pp.1–16.

Ju, K. and Chung, K. 2013. Dynamic TXOP allocation for multimedia QoS providing over wireless networks *In*: *The International Conference on Information Networking 2013 (ICOIN)*. IEEE, pp.397–401.

Kar, R., Madhar Saheb, S., Dharmasa, P. and Bhattacharjee, A.K. 2012. Enhanced hybrid coordination function controlled channel access-based adaptive scheduler for delay sensitive traffic in IEEE 802.11e networks. *IET Networks*. pp.281–288.

Khambari, M.N.M., Sahib, S., Fairuz, M. and Othman, I. 2007. Improving IEEE 802 . 11 to Support Quality of Service in Wireless Networks *In*: *Regional Conference on Computational Science and Technologies*. Kota Kinabalu, Malaysia.

Khambari, N., Ghita, B., Lancaster, D., Sun, L., "QoE Enhancements in IEEE 802.11 e EDCA for Video Transmission through Selective Queueing.", Proceedings of the *11th International Network Conference (INC),* 2016. Frankfurt, Germany.

Khambari, N., Ghita, B., Sun, L., "QoE-driven Video Enhancements in Wireless Networks through Predictive Packet Drops", Proceedings of the *IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* 2017. Rome, Italy.

Khambari, N., Ghita, B., "QoE Enhancements for Video Traffic in Wireless Networks through Selective Packet Drops", Proceedings of the *5th International Conference on Computational Science and Technology (ICCST 2018),* 2018, Kota Kinabalu, Malaysia.

Khan, A., Sun, L. and Ifeachor, E. 2009. Content Clustering Based Video Quality Prediction Model for MPEG4 Video Streaming over Wireless Networks. *Journal of Multimedia*.

pp.228–239.

Khan, A., Sun, L., Jammeh, E. and Ifeachor, E. 2010. Quality of experience-driven adaptation scheme for video applications over wireless networks. *IET Communications*. **4**(11), p.1337.

Khan, K.U.R., Zaman, R.U. and Reddy, a. V. 2008. Performance Comparison of On-Demand and Table Driven Ad Hoc Routing Protocols Using NCTUns. *Tenth International Conference on Computer Modeling and Simulation (uksim 2008)*. pp.336–341.

Kim, T.O., Chang, Y., Kim, Y. and Choi, B.D. 2006. *Management of Convergence Networks and Services* (Y.-T. Kim & M. Takano, eds.). Berlin, Heidelberg: Springer Berlin Heidelberg.

Klaue, J., Rathke, B. and Wolisz, A. 2003. EvalVid - A Framework for Video Transmission and Quality Evaluation *In*: *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*., pp.255–272.

Kosek-Szott, K., Natkaniec, M., Szott, S., Krasilov, A., Lyakhov, A., Safonov, A. and Tinnirello, I. 2013. What's new for QoS in IEEE 802.11? *IEEE Network*. **27**(6), pp.95–104.

Ksentini, A., Naimi, M. and Gueroui, A. 2006. Toward an Improvement of H.264 Video Transmission over IEEE 802.11e Through a Cross-Layer Architecture. *IEEE Communications Magazine*. pp.107–114.

Kuo, W.-K. and Wu, K.-W. 2011. Traffic prediction and QoS transmission of real-time live VBR videos in WLANs. *ACM Transactions on Multimedia Computing, Communications, and Applications*. pp.1–21.

Kuo, W. 2008. Supporting multimedia streaming and best-effort data transmission over IEEE 802.11e EDCA. *International Journal of Network Management*. **18**(3), pp.209–228.

de la Oliva, A., Serrano, P., Salvador, P. and Banchs, A. 2013. Performance evaluation of the IEEE 802.11aa multicast mechanisms for video streaming *In*: *2013 IEEE 14th International Symposium on 'A World of Wireless, Mobile and Multimedia Networks' (WoWMoM)* IEEE, pp.1–9.

Laghari, K.U.R., Crespi, N., Molina, B. and Palau, C.E. 2011. QoE aware service delivery in distributed environment. *Proceedings - 25th IEEE International Conference on Advanced*

Information Networking and Applications Workshops, WAINA 2011., pp.837–842.

Laghari, R., Crespi, N., Telecom, I. and Sudparis, T. 2012. Toward Total Quality of Experience: A QoE Model in a Communication Ecosystem. *IEEE Communications Magazine*. (April), pp.58–65.

Lin, C.-H., Shieh, C.-K., Ke, C.-H., Chilamkurti, N.K. and Zeadally, S. 2009. An Adaptive Cross-Layer Mapping Algorithm for MPEG-4 Video Transmission over IEEE 802.11e WLAN. *Telecommunication Systems*. **42**(3–4), pp.223–234.

Lindgren, A., Almquist, A. and Schelén, O. 2003. Quality of Service Schemes for IEEE 802 . 11 Wireless LANs – An Evaluation. *Mobile Networks and Applications*. **8**(3), pp.223–235.

Liu, H., Li, C., Hao, S., Cai, X. and Li, J. 2013. A novel internal collision managing mechanism of IEEE 802.11e EDCA *In*: *2013 19th Asia-Pacific Conference on Communications (APCC)*. Bali, Indonesia: IEEE, pp.473–478.

Maia, O.B., Yehia, H.C. and Errico, L. De 2015. A concise review of the quality of experience assessment for video streaming. *Computer Communications*. pp.1–12.

Mangold, S., Hiertz, G.R., Klein, O. and Walke, B. 2003. Analysis of IEEE 802.11e for QoS support in wireless LANs. *IEEE Wireless Communications*. **10**(6), pp.40–50.

Maraslis, K., Chatzimisios, P. and Boucouvalas, A. 2012. IEEE 802.11aa: Improvements on video transmission over wireless LANs *In*: *2012 IEEE International Conference on Communications (ICC)*. IEEE, pp.115–119.

MSU Graphics and Media Lab 2015. MSU Video Quality Measurement Tool. [Accessed 25 April 2016]. Available from: http://compression.ru/index_en.htm.

Bin Muhamad Noh, Z.A., Suzuki, T. and Tasaka, S. 2009. Effect of the Service Interval on Application-Level QoS of Audio-Video Transmission by IEEE 802.11e HCCA *In*: *IEICE General Conference*.

Bin Muhamad Noh, Z.A., Suzuki, T. and Tasaka, S. 2010. QoE Estimation from Application-Level QoS in Audio-Video Transmission by IEEE 802 . 11e HCCA *In*: *IEICE General Conference*., pp.2–3.

Murray, D., Koziniec, T. and Dixon, M. 2009. Solving Ack inefficiencies in 802.11 networks

In: *2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*. IEEE, pp.1–6.

Nielsen 2014. *The Total Audience Report*.

Noh, Z.A.B.M., Suzuki, T. and Tasaka, S. 2008. Guaranteeing QoE in Audio-Video Transmission by IEEE 802.11e HCCA *In*: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, pp.1551–1561.

Noh, Z.A.M., Takahiro, S. and Shuji, T. 2010. Application-Level QoS and QoE Assessment of a Cross-Layer Packet Scheduling Scheme for Audio-Video Transmission over Error-Prone IEEE 802.11e HCCA Wireless LANs. *IEICE Transactions on Communications*. **E93-B**(6), pp.1384–1394.

NS3 Project 2016. *The NS-3 Wi-Fi Module Documentation* [Online]. Available from: https://www.nsnam.org/workshops/wns3-2010/dot11s.pdf.

Ohrtman, F. 2004. *Voice Over 802.11*. Artech House.

Padle, S.M. and Mendre, V. 2014. Cross Layer (Application and Medium Access Control) Design and Optimization of Wireless Network for. . **5**(1), pp.1–7.

Pattara-Atikom, W. 2005. *Quality of Service Support in IEEE 802.11 Wireless LAN*. University of Pittsburgh.

Prakash, G. and Thangaraj, P. 2012. Performance Comparison of IEEE 802.11e EDCA and 802.11b DCF Under Non-Saturation Condition using Network Simulator. *Research Journal of Applied Sciences, Engineering and Technology*. **4**(22), pp.4748–4754.

Ramachandran, S. 2014. Pay-TV 'Cord Cutting' Accelerates. *The Wall Street Journal*.

Rodrigues, D., Silva, D., Cerqueira, E. and Monteiro, E. 2008. Quality of Service and Quality of Experience in Video Streaming. *In Proc. of the International Workshop on Traffic Management and Traffic Engineering for the Future Internet (FITraMEn2008), EuroNF NoE, Porto, Portugal,*., pp.1–8.

Ruscelli, A.L., Cecchetti, G., Alifano, A. and Lipari, G. 2012. Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks. *Ad Hoc Networks*. **10**(2), pp.147–161.

Sadek, R., Youssif, A. and Elaraby, A. 2015. MPEG-4 Video Transmission Over IEEE 802.11e

Wireless Mesh Networks Using Dynamic-Cross-Layer Approach. *National Academy Science Letters*.

Shin, Y., Choi, M., Koo, J. and Choi, S. 2013. Video multicast over WLANs: Power saving and reliability perspectives. *IEEE Network*. **27**(2), pp.40–46.

Siris, V.A. and Courcoubetis, C. 2006. Resource Control for the EDCA and HCCA Mechanisms in IEEE 802.11e Networks *In*: *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*. IEEE, pp.1–6.

Son, S., Park, K. and Park, E. 2013. Adaptive tuning of IEEE 802.11e EDCA for medical-grade QoS *In*: *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)* Da nang, Vietnam: IEEE, pp.650–651.

Stanley, M. 2010. *Internet Trends*.

Su, G.-M., Su, X., Bai, Y., Wang, M., Vasilakos, A. V. and Wang, H. 2016. QoE in video streaming over wireless networks: perspectives and research challenges. *Wireless Networks*. **22**(5), pp.1571–1593.

Tariq, A.S.M. and Perveen, K. 2010. Analysis of Internal Collision and Dropping Packets Characteristics of EDCA IEEE802.11e Using NS-2.34 Simulator *In*: *World Congress on Engineering and Computer Science*., pp.20–23.

Vassiliou, V., Antoniou, P., Giannakou, I. and Pitsillides, A. 2006. Requirements for the Transmission of Streaming Video in Mobile Wireless Networks *In*: *International Conference on Artificial Neural Network*., pp.528–537.

Viegas, R., Guedes, L.A., Vasques, F., Portugal, P. and Moraes, R. 2013. A new MAC scheme specifically suited for real-time industrial communication based on IEEE 802.11e. *Computers & Electrical Engineering*. **39**(6), pp.1684–1704.

Watkinson, J. 2001. *The MPEG Handbook : MPEG-1, MPEG-2, MPEG-4* [Online]. IEEE. Available from: http://ieeexplore.ieee.org/document/6341273/.

Yao, X.X.-W., Wang, W., Yang, S., Cen, Y., Yao, X.X.-W., Pan, T. and Camp, J. 2014. IPB-frame Adaptive Mapping Mechanism for Video Transmission over IEEE 802.11e WLANs. *ACM SIGCOMM Computer Communication Review*. **44**(2), pp.5–12.

Zhang, H., Lu, Z., Wen, X. and Hu, Z. 2015. QoE-Based Reduction of Handover Delay for

Multimedia Application in IEEE 802.11 Networks. *IEEE Communications Letters*. **19**(11), pp.1873–1876.

Zinner, T., Hohlfeld, O., Abboud, O. and Hossfeld, T. 2010. Impact of frame rate and resolution on objective QoE metrics *In*: *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, pp.29–34.

Appendix A

Code snippets are shown here. More codes are available at:

`https://github.com/naj1k/ns-3-Plymouth/tree/master`

Main simulation code snippets

```
// Set qos-tag for packets
void SetTagTid (uint8_t mytid, Ptr<const Packet> pkt ){
  QosTag  qosTag ;
  qosTag.SetTid (mytid);
  pkt->AddPacketTag (qosTag);
  //NS_LOG_UNCOND ("Sent Packet with Tag: " << (int)mytid << "\n");
}

int main (int argc, char *argv[]){

  //Enable logging for EvalvidClient and Evalvid Server
  LogComponentEnable ("EvalvidClient", LOG_LEVEL_INFO);
  LogComponentEnable ("EvalvidServer", LOG_LEVEL_INFO);

  bool verbose = true;
  uint32_t nWifi = 4;
  std::string rtslimit = "3000";
  uint32_t qcase = 1;                              //Queue selection
  uint32_t selectapp = -1;                         //App selection
  uint32_t maxpktnum = 400;                        //Default value for max pkt num in a queue
  std::string stfile = "st_highway_cif.st";  //video to use

  uint32_t psnrthreshold = 30;
  uint32_t Vi1Time = (rand() % 10) + 1;
  uint32_t Vi2Time = (rand() % 10) + 1;
  uint32_t Vi3Time = (rand() % 10) + 1;
  uint32_t Vi4Time = (rand() % 10) + 1;
  uint32_t Vi5Time = (rand() % 10) + 1;

  CommandLine cmd;
  cmd.AddValue ("nWifi", "Number of wifi STA devices", nWifi);
  cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);
  cmd.AddValue ("stfile", "Tell echo applications to log if true", stfile);
  cmd.AddValue ("maxpktnum", "maximum number of packet in queue", maxpktnum);
  cmd.AddValue ("qcase", "which algorithm enqueue to use in sta-wifi-mac", qcase);
  cmd.AddValue ("selectapp", "which algorithm enqueue to use in sta-wifi-mac", selectapp);
  cmd.AddValue ("Vi1Time", "additional start/ stop time", Vi1Time);
  cmd.AddValue ("Vi2Time", "additional start/ stop time", Vi2Time);
  cmd.AddValue ("Vi3Time", "additional start/ stop time", Vi3Time);
  cmd.AddValue ("Vi4Time", "additional start/ stop time", Vi4Time);
  cmd.AddValue ("Vi5Time", "additional start/ stop time", Vi5Time);
  cmd.AddValue ("psnrthreshold", "change the psnr threhold", psnrthreshold);

  cmd.Parse (argc,argv);

  NodeContainer wifiApNode;            //Create AP nodes (Node0) wifiApNode1
  wifiApNode.Create(1);
```

```
  NodeContainer wifiStaNodes;         //Create wifi nodes (n1, n2, n3 and n4) wifiStaNodes0,
                                      wifiStaNodes1, wifiStaNodes2, wifiStaNodes3
  wifiStaNodes.Create (nWifi);

  // Create & setup wifi channel
  YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
  YansWifiPhyHelper phy = YansWifiPhyHelper::Default();
  phy.SetChannel (channel.Create ());

  // Install wireless devices
  WifiHelper wifi = WifiHelper::Default ();
  wifi.SetStandard (WIFI_PHY_STANDARD_80211b);
  wifi.SetRemoteStationManager ("ns3::AarfWifiManager");

  QosWifiMacHelper mac = QosWifiMacHelper::Default ();

  Ssid ssid = Ssid ("ns-3-ssid");

  mac.SetType ("ns3::StaWifiMac",
               "Ssid", SsidValue (ssid),
               "qcase", UintegerValue(qcase),
               "psnrthreshold", UintegerValue(psnrthreshold),
               "selectapp", UintegerValue(selectapp),
               "ActiveProbing", BooleanValue (false),
               "QosSupported", BooleanValue(true));

  NetDeviceContainer staDevices;
  staDevices = wifi.Install (phy, mac, wifiStaNodes);

  mac.SetType ("ns3::ApWifiMac",
               "Ssid", SsidValue (ssid),
               "QosSupported", BooleanValue(true));

  NetDeviceContainer apDevices;
  apDevices = wifi.Install (phy, mac, wifiApNode);

  InternetStackHelper stack;
  stack.Install (wifiApNode);
  stack.Install (wifiStaNodes);

  Ipv4AddressHelper address;

  address.SetBase ("10.1.3.0", "255.255.255.0");
  Ipv4InterfaceContainer wifiInterfaces, wifiApInterface;
  wifiApInterface = address.Assign (apDevices);      // 10.1.3.1 / 24
  wifiInterfaces = address.Assign (staDevices); // 10.1.3.x / 24


//Same code for each pair
// PAIR X
// =======
// VI connection from n1 to n2
    port+=1;
// EvalvidClient (RECEIVER)
// ************************

EvalvidClientHelper vi1_rx (wifiInterfaces.GetAddress(0),port);     //insert sender's
                                                     address, sender's port# in the bracket
vi1_rx.SetAttribute ("ReceiverDumpFilename", StringValue("rd_a01_vi1_rx_highway"));
ApplicationContainer vi1_rx_app;
vi1_rx_app = vi1_rx.Install (wifiStaNodes.Get(1));
//install rx app on the receiver's node
vi1_rx_app.Start (Seconds (10.0 + Vi1Time));
vi1_rx_app.Stop (Seconds (100.0 + Vi1Time));

 // EvalvidServer (SENDER)
 // **********************
 EvalvidServerHelper vi1_tx (port);
 vi1_tx.SetAttribute ("SenderTraceFilename", StringValue(stfile));
 vi1_tx.SetAttribute ("SenderDumpFilename", StringValue("sd_a01_vi1_tx_highway"));
 vi1_tx.SetAttribute ("PacketPayload", UintegerValue(1014));

 ApplicationContainer vi1_tx_app;
 vi1_tx_app = vi1_tx.Install (wifiStaNodes.Get(0));
 vi1_tx_app.Start (Seconds (5.0 + Vi1Time));  //application START
 vi1_tx_app.Stop (Seconds (200.0 + Vi1Time));  //application STOP

 Simulator::Stop (Seconds (500.0));
 Simulator::Run ();
```

LIBRARY CODE: STA-WIFI-MAC.CC

Code Snippet

```
//header files added by Najwan
#include "ns3/node-id-tag.h"          //identify node id tag in CheckPacketInfo()
#include "ns3/frame-type-tag.h"       //identify frame type tag in CheckPacketInfo()
#include "ns3/app-id-tag.h"           //identify application id tag in CheckPacketInfo()
#include "ns3/dev-id-tag.h"           //identify device id tag in CheckPacketInfo()
#include "ns3/ipv4-header.h"          //identify source and destination IP in CheckPacketInfo()
#include "ns3/seq-ts-header.h"        //identify packet sequence in CheckPacketInfo()
#include "ns3/udp-header.h"           //needed for identifying source port in CheckPacketInfo()
#include "ns3/evalvid-server.h"       //use in conjunction with EvalvidServer pointer
#include "ns3/node-list.h"            //use in conjuction with the NodeList
#include "wifi-net-device.h"          //use in conjuction with WifiNetDevice pointer line 494
#include "wifi-mac-queue.h"           //use in conjuction with accessing the WifiMacQueue
                                        pointer and the BK,BE,VI,VO Accessor
#include <fstream>                    //To use with std::ofstream
#include <iostream>
#include <string>

TypeId
StaWifiMac::GetTypeId (void)
{
  static TypeId tid = TypeId ("ns3::StaWifiMac")
    .SetParent<RegularWifiMac> ()
    .SetGroupName ("Wifi")
    .AddConstructor<StaWifiMac> ()

    //Attributes added by Najwan

    .AddAttribute ("qcase",
                   "Choose which algorithm to use "
                   "for enqueueing.",
                   UintegerValue (1),
                   MakeUintegerAccessor (&StaWifiMac::m_qcase),
                   MakeUintegerChecker<uint32_t> ())
    .AddAttribute ("selectapp",
                   "Choose which app "
                   "has the proposed algorithm.",
                   UintegerValue (-1),
                   MakeUintegerAccessor (&StaWifiMac::m_selectapp),
                   MakeUintegerChecker<uint32_t> ())
    .AddAttribute ("psnrthreshold",
                   "Choose the value of "
                   "psnr threshold.",
                   UintegerValue (30),
                   MakeUintegerAccessor (&StaWifiMac::m_psnrthreshold),
                   MakeUintegerChecker<uint32_t> ())

//Added by Najwan to check packet info
//Check all the packet info by sending to CheckPacketInfo()
//=========================================================

Ptr<Packet> packet_copy = packet->Copy();    //NK: Copy packet to be sent to CheckPacketInfo()
CheckPacketInfo(packet_copy);                //NK: Send packet_copy to CheckPacketInfo()

//Get the packet information
Mac48Address sender = m_low->GetAddress ();
Mac48Address receiver = to;
Ipv4Address SourceIP = GetSourceIP();
Ipv4Address DestinationIP = GetDestinationIP();
uint32_t SourcePort = GetSourcePort();
uint32_t SeqId = GetSeqId();
int FrameTypeValue = GetFrameTypeValue();
std::string FrameTypeString = GetFrameTypeString();
//uint8_t Tid = GetQosTid();
uint32_t NodeId = GetNodeId();
uint32_t DevId = GetDevId();
uint32_t AppId = GetAppId();



//switch cases for different algorithms
switch (m_qcase) {

//Default queueing
```

```
case 1:{
		QueueDefault(tid, packet, hdr, MacQueueNVi, SourcePort, AppId, VidContentType,
		TotalI, TotalP, TotalB);
} break;


//Remove Any B-Frame to acommodate Own I-Frame
case 5:{
	std::cout << "RemoveAnyBFrameForOwnIFrameIfViQueueFull" << std::endl;

	if (FrameTypeValue==1){
	       RemoveAnyBFrameForOwnIFrameIfViQueueFull(tid, packet, hdr, MacQueueNVi, receiver,
		EvalServ, SourcePort, FrameTypeValue, SeqId, AppId);

	} else {
		QueueDefault(tid, packet, hdr, MacQueueNVi, SourcePort, AppId, VidContentType,
		TotalI, TotalP, TotalB);
	}
} break;


//Remove Own B-Frame to acommodate Own I-Frame
case 6:{
	if (FrameTypeValue==1){
		std::cout << "RemoveOwnBFrameForOwnIFrameIfViQueueFull" << std::endl;

		RemoveOwnBFrameForOwnIFrameIfViQueueFull(tid, packet, hdr, MacQueueNVi, receiver,
		EvalServ, SourcePort, FrameTypeValue, SeqId, AppId);

	} else {
		QueueDefault(tid, packet, hdr, MacQueueNVi, SourcePort, AppId, VidContentType,
		TotalI, TotalP, TotalB);
	}
} break;


//RemoveAnyBFrameWhileQoEAbove30 RAPB if QoE < 30
case 8:{
	if (FrameTypeValue==1){
	std::cout << "RemoveAnyBFrameIfQoEBelow30" << std::endl;
	RemoveAnyBFrameIfQoEBelow30(tid, packet, hdr, MacQueueNVi, receiver, EvalServ,
	SourcePort, FrameTypeValue, TotalI, TotalP, TotalB, VidContentType, SeqId, AppId);

	} else {
		QueueDefault(tid, packet, hdr, MacQueueNVi, SourcePort, AppId, VidContentType,
		TotalI, TotalP, TotalB);
	}
} break;


//RemoveOwnBFrameWhileQoEAbove30 ROPB if QoE < 30
case 9:{
	if (FrameTypeValue==1){
		RemoveOwnBFrameIfQoEBelow30 (tid, packet, hdr, MacQueueNVi, receiver, EvalServ,
		SourcePort, FrameTypeValue, TotalI, TotalP, TotalB, VidContentType, SeqId,
		AppId);

	} else {
		QueueDefault(tid, packet, hdr, MacQueueNVi, SourcePort, AppId, VidContentType,
		TotalI, TotalP, TotalB);
	}
} break;

//Checking packet info

void StaWifiMac::CheckPacketInfo (Ptr<Packet> packet_copy){
	/* Check packet info based on metadata
	 * 1. SourceIP		(IPv4 Header)
	 * 2. DestinationIP	(IPv4 Header)
	 * 3. Source Port	(UDP Header)
	 * 4. SeqId			(SeqTs Header)
	 */

	Ipv4Address SourceIP, DestinationIP;
	uint32_t SourcePort = -1, SeqId = -1;

	packet_copy->EnablePrinting();

	PacketMetadata::ItemIterator metadataIterator = packet_copy->BeginItem();
```

```
PacketMetadata::Item item;

while (metadataIterator.HasNext()){
        item = metadataIterator.Next();

        //CheckIpv4Header
        if(item.tid.GetName()=="ns3::Ipv4Header"){
                //std::cout << "ns3::Ipv4Header" << std::endl;
                Callback<ObjectBase *> constr = item.tid.GetConstructor();
                NS_ASSERT(!constr.IsNull());

                ObjectBase *instance = constr();
                NS_ASSERT(instance != 0);

                Ipv4Header* ipv4Header = dynamic_cast<Ipv4Header*> (instance);
                NS_ASSERT(ipv4Header != 0);
                ipv4Header->Deserialize(item.current);

                // The tcp sequence can now obtain the source of the packet
                SourceIP                = ipv4Header->GetSource();
                DestinationIP  = ipv4Header->GetDestination();

                //Set Source / Destination Address***
                SetSourceIP(SourceIP);
                SetDestinationIP(DestinationIP);
                delete ipv4Header;
        }

        //CheckUdpHeader
        if(item.tid.GetName()=="ns3::UdpHeader"){
                //std::cout << "ns3::UdpHeader" << std::endl;
                Callback<ObjectBase *> constr = item.tid.GetConstructor();
                NS_ASSERT(!constr.IsNull());

                ObjectBase *instance = constr();
                NS_ASSERT(instance != 0);

                UdpHeader* udpHeader = dynamic_cast<UdpHeader*> (instance);
                NS_ASSERT(udpHeader != 0);
                udpHeader->Deserialize(item.current);

                // The source port can now obtain the source of the packet
                SourcePort = udpHeader->GetSourcePort();

                //Set Source Port Number***
                SetSourcePort(SourcePort);
                delete udpHeader;
        }

        //Check SeqTsHeader
        if(item.tid.GetName()=="ns3::SeqTsHeader"){
                //std::cout << "ns3::SeqTsHeader" << std::endl;
                Callback<ObjectBase *> constr = item.tid.GetConstructor();
                NS_ASSERT(!constr.IsNull());

                ObjectBase *instance = constr();
                NS_ASSERT(instance != 0);

                SeqTsHeader* seqTsHeader = dynamic_cast<SeqTsHeader*> (instance);
                NS_ASSERT(seqTsHeader != 0);
                seqTsHeader->Deserialize(item.current);

                // The tcp sequence can now obtain the source of the packet
                SeqId = seqTsHeader->GetSeq();

                //Set SeqId***
                SetSeqId(SeqId);
                delete seqTsHeader;
                break;
        }
}

/* Check packet info based on frame tag
 * 1. Frame Tag Value  (Frame Type Tag)
 * 2. Frame Type              (Frame Type Tag)
 * 3. Qos Tid                 (QoS Tag)
 */

//Check Frame Type Tag
```

```
        FrameTypeTag frameTag;
        uint32_t FrameTypeValue;
        std::string FrameTypeString = "";

        if (packet_copy->PeekPacketTag(frameTag)){

                FrameTypeValue = frameTag.GetFrameType();
                SetFrameTypeValue(FrameTypeValue);

                if (FrameTypeValue == 1){
                        FrameTypeString = "I";
                        SetFrameTypeString(FrameTypeString);

                } else if (FrameTypeValue == 2){
                        FrameTypeString = "P";
                        SetFrameTypeString(FrameTypeString);

                } else if (FrameTypeValue == 3){
                        FrameTypeString = "B";
                        SetFrameTypeString(FrameTypeString);

                } else {
                        SetFrameTypeString("UNKNOWN FRAMETAG NUMBER" );
                }
        }

        //Check QoS Tag
        QosTag qosTag;
        uint8_t qosTid;

        if (packet_copy->PeekPacketTag(qosTag)){
                qosTid = qosTag.GetTid();
                SetQosTid(qosTid);
        }

        //Check NodeId Tag
        NodeIdTag nodeIdTag;
        uint32_t NodeId=-1;

        if (packet_copy->PeekPacketTag(nodeIdTag)){
                NodeId = nodeIdTag.GetNodeId();
                SetNodeId(NodeId);
        }

        //Check DevId Tag
        DevIdTag devIdTag;
        uint32_t DevId=-1;

        if (packet_copy->PeekPacketTag(devIdTag)){
                DevId = devIdTag.GetDevId();
                SetDevId(DevId);
        }

        //Check AppId Tag
        AppIdTag appIdTag;
        uint32_t AppId=-1;

        if (packet_copy->PeekPacketTag(appIdTag)){
                AppId = appIdTag.GetAppId();
                SetAppId(AppId);
        }
}

//Setter Getter Start

//SourceIP Setter/ Getter
void StaWifiMac::SetSourceIP(Ipv4Address SourceIP){
        m_SourceIP = SourceIP;
}

Ipv4Address StaWifiMac::GetSourceIP(){
        return m_SourceIP;
}

//DestinationIP Setter/ Getter
void StaWifiMac::SetDestinationIP(Ipv4Address DestinationIP){
        m_DestinationIP = DestinationIP;
}
```

```
Ipv4Address StaWifiMac::GetDestinationIP(){
        return m_DestinationIP;
}

//SourcePort Setter/ Getter
void StaWifiMac::SetSourcePort(uint32_t SourcePort){
        m_SourcePort = SourcePort;
}

uint32_t StaWifiMac::GetSourcePort(){
        return m_SourcePort;
}

//SeqId Setter/ Getter
void StaWifiMac::SetSeqId(uint32_t SeqId){
        m_SeqId = SeqId;
}

uint32_t StaWifiMac::GetSeqId(){
        return m_SeqId;
}

//FrameTypeValue Setter/ Getter
void StaWifiMac::SetFrameTypeValue(uint32_t FrameTypeValue){
        m_FrameTypeValue = FrameTypeValue;
}

uint32_t StaWifiMac::GetFrameTypeValue(){
        return m_FrameTypeValue;
}

//FrameType Setter/ Getter
void StaWifiMac::SetFrameTypeString(std::string FrameTypeString){
        m_FrameTypeString = FrameTypeString;
}

std::string StaWifiMac::GetFrameTypeString(){
        return m_FrameTypeString;
}

//QoSTid Setter/ Getter
void StaWifiMac::SetQosTid(uint8_t QosTid){
        m_QosTid = QosTid;
}

uint8_t StaWifiMac::GetQosTid(){
        return m_QosTid;
}

//NodeId Setter/ Getter
void StaWifiMac::SetNodeId(uint32_t NodeId){
        m_NodeId = NodeId;
}

uint32_t StaWifiMac::GetNodeId(){
        return m_NodeId;
}

//DeviceId Setter/ Getter
void StaWifiMac::SetDevId(uint32_t DevId){
        m_DevId = DevId;
}

uint32_t StaWifiMac::GetDevId(){
        return m_DevId;
}

//AppId Setter/ Getter
void StaWifiMac::SetAppId(uint32_t AppId){
        m_AppId = AppId;
}

uint32_t StaWifiMac::GetAppId(){
        return m_AppId;
}

//Setter Getter End
//Added by Najwan [-END-]
```