Quality Queue Management for Future Wireless Networks

Gianluigi Pibiri

A Dissertation submitted to the University of Dublin, Trinity College

in fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

September 2017

Declaration

I declare that this work has not been submitted as an exercise for a degree at this or any other University and it is entirely my own work.

Gianluigi Pibiri

Dated: September 14th, 2017

Permission to Lend and/or Copy

I agree to deposit this thesis in the University's open access institutional repository or allow the librarian to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Gianluigi Pibiri

Dated: September 14th, 2017

Acknowledgements

I will be forever thankful to my supervisor, Dr. Meriel Huggard, for her support and patience with me me over the years. Her suggestions and advice have been most helpful; not only throughout the Ph.D. process, but also in my personal life.

I would like to thank Dr. Ciarán Mc Goldrick for his willingness to provide me with advice and guidance. I have benefited greatly from his insightful technical observations throughout my work on this dissertation. He was a second supervisor to me and was always happy to suggest possible research directions for me to explore when I ran into challenges along the way.

I thank my colleagues Ricardo, Alessandro, Ritesh and Waqas for their support and friendship – I enjoyed drinking copious cups of coffee and tea with you all over the past few years.

I enjoyed drinking copious cups of coffee and tea with you all over the past few years.

I would like to thank my family, and in particular my parents, for their belief in me and their on-going support. I would not have completed this work without the knowledge that they were there for me no matter what.

I express my gratitude and thanks to all those who have supported me in my work on this thesis.

Gianluigi Pibiri

University of Dublin, Trinity College

September 2017

Abstract

Wireless networking protocols and mobile devices are key contributors to the ever-growing demand for realtime services. These large throughput services are often prioritised and managed using specialised traffic policies. Today's wireless networks will be replaced by their newer, faster, less expensive counterparts in the very near future. These Very High Throughput networks will deliver a significant increase in throughput and a reduction in effective cost.

Realtime services are sensitive to packet loss and delay, and their successful delivery demands a high standard of Quality of Service (QoS). The quality perceived by the final user is measured as the Quality of Experience (QoE). Hence, Telephone Companies (TELCOs) aim to provide realtime services with the best QoE possible.

This thesis aims to extend the provision of QoE to realtime services on future wireless networks.

The first contribution of this work is a novel metric, the expected Quality of Service (eQoS), that estimates QoE on future wireless networks using combinatorics. eQoS provides an almost instantaneous estimate of the QoE at the node and is inferred from the traffic flows crossing the node. It provides a statistical estimate of the end user perception of the network's quality.

The second contribution is a theoretical model to capture the probability of successful and unsuccessful channel access on a future wireless network. This theoretical model uses combinatorics to calculate the probabilities that (i) a collision occurs, (ii) a packet is transmitted and (iii) the channel is idle.

The third contribution is to converge eQoS and the theoretical model into the Quality Queue

Management (QQM) system for future wireless networks. QQM is designed using fuzzy logic controllers, it manages wireless resources with the goal of providing real time traffic flows with the best QoE possible in the network. QQM incorporates eQoS and is a robust, efficient, scalable means of managing realtime services on future wireless networks. The novelty of the QQM system is to use eQoS to manage network parameters.

The QQM system optimises the throughput between the flows, manages the queues and provides the maximum number of realtime services with the best quality possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Almost all global information exchanges take place across the Internet and it is continuously growing and evolving to meet the needs of its ever-expanding multitude of users. It allows for the exchange of information via a wide-range of protocols, from simple file transfers and more complex database related web services through to the delivery of key realtime services like phone calls and video streaming.

The demand for continuity of access to information, regardless of the user's location, has encouraged the development and refinement of existing wireless technologies and protocols. The volume of network traffic generated by laptops, tablets and mobile phones is continuously increasing [1] [2] [3] and these are now key instruments for information exchange.

The information being transmitted is, itself, changing in nature. The focus is now on realtime data delivery as these services are one of the major drivers of network traffic growth in terms of volume and the number of data flows [1] [2] [3]. Consequently, those developing new wireless network technologies must consider customer satisfaction as a key aspect of their provisioning policy. This is mainly due by economic factors: customers pay for these services and Telephone Companies (TELCOs) compete for business by offering their users low cost, high quality services.

1.1 Context

Mobile communications are a core element of the services provided by modern TELCOs. Mobile devices are a practical, economic mechanism for everyday communications in the home and office. ISPs are interested in wireless technologies and their evolution, as they are seen as a key element of their future service offerings. They are constantly seeking to develop and expand the number and variety of wireless services they offer so as to increase the number of sources of revenue available to them.

There are two main technologies that mobile, wireless devices use to access internet. The first of these is cellular networking; for example, Fourth Generation (4G) phone networks [4] or Worldwide Interoperability for Microwave Access (WiMax) [5] networks that use existing protocols such as the Internet Protocol (IP) [5]. The second access method that is commonly used employs wireless networking technologies based on the IEEE802.11 [5] family of protocols.

Wireless networks can form Local Area Networks (LAN) using High Throughput (HT) wireless protocols, while some lower throughput wireless protocols form smaller Personal Area Networks (PAN). Wireless networking technologies are not limited to these settings and it should be noted that it is equally possible to use wireless technologies to form the backbone or backhaul of an access wireless network.

Future wireless networks, as heralded by the standardisation of Very High Throughput (VHT) [6] in December 2013, are able to provide real time services that rival those on offer on more expensive cellular networks. The widespread deployment of Quality of Service (QoS) provision mechanisms and traffic management algorithms within these networks allow them to provide mobile devices with access to the Internet at an almost negligible cost.

VHT [7] refers to the physical characteristics of the protocol with a larger channel than High Throughput (HT) and, consequently, the ability to transmit data in the wireless channel at a higher speed than traditional HT.

1.1.1 Quality of Service

The prevalence of wireless networks, and the mobile devices that use them, have forced TELCOs to explore ways to provide realtime services to their customers with some form of quality assurance. In particular, TELCOs are seeking to improve the quality of the services provided, the QoS, and the quality as assessed by the end-users of those services, the Quality of Experience (QoE). Users may make decisions on whether to use a service, or not, based on their perceived QoE [8].

QoE [8] is a well-established means of determining customer satisfaction. It indicates if the service is provided with "good" quality from the end-users perspective. Hence, TELCOs are interested in monitoring QoE and in managing the traffic on their network so as to guarantee their customers a high level of QoE.

TELCOs can easily measure the QoS through the physical network parameters. Deterioration of the network parameters indicates that the service is provided with low QoE. The challenge is to find an easily calculated, instantaneous mechanism that captures the relationship between QoE and the physical network parameters and a mechanism that uses this to adjust the network parameters and improve the quality of the service delivered to the end user and, consequently, the QoE.

1.1.2 Traffic Management

This thesis explores ways to manage the traffic in a wireless network effectively and efficiently. The first mechanism employed to do this will be the regulation of channel access through traffic prioritisation, the second will use Active Queue Management (AQM) [9] algorithms.

AQM [9] algorithms were introduced over twenty years ago as a congestion avoidance mechanism for use in router buffers. A key objective of most AQM schemes is to avoid the queue becoming full by dropping selected packets as they arrive at the queue. Traditional buffering systems simply drop arriving packets when the queue is full as there is no place to store then. These systems will then continue to drop packets as long as the queue remains congested, regardless of the prevailing network conditions.

The key goal of all AQM schemes is to drop packets before the queue overflows [9]. This signals traffic sources about the existence of network congestion and they respond by reducing the packet arrival rate at the queue. AQM schemes can handle Transmission Control Protocol (TCP) traffic, because the traffic throughput is regulated via acknowledgement packets sent to the traffic source. On the other hand, User Datagram Protocol (UDP) traffic cannot be managed in this way as, by definition, it does not respond to acknowledgements or packet drops [10].

These schemes have mainly been designed and implemented on wired networks and there has been only limited exploration of how they may be successfully implemented on wireless networks. Their use on wireless networks is complicated by the widely varying characteristics of the traffic mix and the typically low traffic throughput observed on these networks [11] [12].

The benefits of AQM schemes are that they prevent and delay congestion, they can increase fairness amongst traffic flows and they can control the throughput of traffic. From the point of view of quality, they can help to manage physical network parameters; for example, they can maximize throughput and reduce queueing delays. These improvements do not necessarily have a knock-on effect on the delivery of QoE as network parameter improvements only have an indirect impact on the QoE.

Traffic management schemes that are designed to provide specific QoE guarantees are attractive for a number of reasons. For example, by improving the QoE it follows that the network functionality will also improve; moreover, the network is protected from undesirable traffic and any unfairness between flows is eliminated. In addition, QoE can provide network elements with a form of feedback for unresponsive flows [13].

1.2 Motivation

This dissertation is motivated by three main technical and economic concerns: the provision of high quality internet access that truly delivers on the performance promised by future wireless networks, the need to provide an effective open resource sharing network infrastructure that offers an alternative to cellular systems and the financial desire to offer satisfactory, low cost services to network users.

The evolution of network access methods, devices used and services required is well documented [1] [2] [3]. Wireless technologies are the most popular method of internet access, and almost everyone has one or more mobile devices, such as a cell phone, tablet or laptop. Future wireless protocols will soon become widely available on network devices for public use in the home, office and wider environment. As inferred from [1] [2] [3], real-time communications like phone calls, video calls and conferencing and more general audio and video streaming are currently, and likely to remain, the dominant methods of communication.

Future wireless networks have the potential to offer the same services as cell phone networks but on a network where local access points provide high quality internet connectivity. Such networks could then provide services with a quality that is comparable to that of more expensive, sophisticated cellular networks in terms of the available throughput, the cost and the geographic spread of availability. This is most feasible in urban areas where the concentration of future wireless network access points is likely to be most dense.

End users require the wireless services they are provided with to be of satisfactory quality and, ideally, at minimal cost. If future wireless networks are to provide these services then it is necessary to enhance the mechanisms for quality provision associated with them. In particular, network traffic management systems need to extend beyond the provision of basic QoS to include enhanced guarantees of end-user QoE.

The discussion above motivates the need to develop a system to optimise future wireless in order to deliver their services with the highest quality possible and was the inspiration for the work presented in this thesis.

1.3 Contribution

This thesis addresses the challenge of designing a system that is suitable for use in a typical wireless access point and capable of delivering guaranteed QoE to realtime services. The design of this system relies on two core contributions of this thesis: the first is a metric to provide an instantaneous estimate of the QoE at a node; the second is a theoretical model to predict some key behaviours of future wireless networks.

The first contribution is the expected Quality of Service (eQoS) metric. It provides an instantaneous estimate of customer satisfaction. It is inferred at the node via per flow sampling. The novelty is in the capability of eQoS to instantaneously capture the relationship between physical network parameters and customer satisfaction.

The second contribution is a new practical, scalable theoretical model that captures packet transmissions and collisions. It is a function of the future wireless network's environmental variables. The novelty of the theoretical model is that it uses combinatorics to infer the probability that, at a given node, the channel is idle, that a collision occurs or that a packet is transmitted.

Finally, the new traffic management algorithm, Quality Queue Management (QQM) exploits eQoS and the theoretical model to control and maintain the quality of each flow in the system. QQM has been designed for future wireless networks, but it is equally applicable to other high throughput wireless networking systems. QQM's novelty is in the way in which it is designed to manage and exploit the interactions between the quality management features of the basic priority queueing system and the channel access mechanisms deployed in multi-queue based systems.

Four publications support the work developed in this thesis. The first two of these relate to how existing AQM schemes can be successfully deployed in a wireless environment. In particular, the first article [12] provides a clear articulation of how traffic prioritisation impacts on wireless access point performance; while the second article sets out how existing queue management algorithms need to be re-engineered for use in a wireless environment [11]. The third publication [14] is of most relevance to this thesis as it is the first articulation of the eQoS metric that forms one of the pillars on which the QQM algorithm is built. The fourth publication [15] completes the contribution of this thesis including the theoretical model and the QQM algorithm.

1.4 Outcomes

This thesis describes the design of an innovative traffic management system that draws on novel elements presented in this work to provide its key functionalities.

The eQoS metric provides instantaneous estimates of the QoE. It will be shown that these estimates are comparable to those obtained using popular automated perceived quality metrics.

The theoretical model will be shown to efficiently estimate the probabilities that the channel is idle, that a packet is transmitted or that a collision occurs when the Enhanced Distributed Channel Access (EDCA) method is used in the wireless network.

The QQM traffic management system can be implemented in all nodes that have access to a wireless network. It is designed to be used in a future wireless networks; however, it can also be applied in all wireless networking environments that use priority systems with multiple queues and a scheduler to manage access to the channel. The QQM traffic management system is structured so that it does not interfere with, or alter, the wireless networking protocol used but rather it works to improve the protocol's performance and enhance its delivery of a quality service to the end user.

1.5 Structure of the Thesis

This section provides a detailed description of the structure of this thesis and of the topics discussed in the remaining chapters.

Chapter two details current wireless network technologies and future wireless networking protocols [6] [16] [17]. The chapter also introduces channel access methods and real time services, such as voice and video, on wireless networks.

The third chapter provides a detailed description of the state of the art, describing the essential underpinnings of this dissertation. The chapter is divided into three parts. The first provides the necessary definitions and sets out the state of the art in relation to QoS [8]. Particular attention is paid to QoE [8] and its evaluation in wireless networks. The second part of the chapter describes wireless network architectures and the mathematical models used to describe wireless network traffic and determine channel access contention. The last part of the chapter is a thorough description of active queue management and its application in wireless networks.

Chapter four sets out the novel quality metric, eQoS. The chapter describes the combinatorial methods used to infer the eQoS for standard realtime services: Voice over IP (VoIP) [18], audio and video traffic [19]. The chapter concludes with a discussion of possible application scenarios for the formulas derived.

QQM and the theoretical model used to determine channel access priorities are detailed in chapter five. This model, together with the eQoS metric, lies at the heart of the design of the QQM algorithm.

Chapter six provides an evaluation of the algorithms and formulas detailed in chapters four and five. It also includes a detailed description of the traffic mix, metrics and software platform used for these evaluations. The results of simulations used to evaluate the eQoS metric are presented and possible application scenarios for the theoretical model are explored. In addition, the chapter presents an extensive exploration by simulation of a practical implementation of QQM algorithm and its components. These results are compared with those obtained from a traditional network that does not make use of QQM.

Chapter seven concludes this thesis, providing a summary of its contributions and suggestions for future work.

1.6 Summary

This chapter provided a brief introduction to the central elements of this thesis. It discussed the wireless network environments and the basics concepts underpinning Quality of Service and Active Queue Management. The initial introduction was followed by a detailed description of the key contributions of this work. Finally, a thesis road map was presented.

1.7 Publications

- PIBIRI, G., MC GOLDRICK, C. AND HUGGARD, M. Ensuring quality services on WiFi networks for offloaded cellular traffic. In *Wireless On-demand Network Systems and Services (WONS)*, 2017 13th Annual Conference on. IEEE, pp. 136-143.

- PIBIRI, G., MC GOLDRICK, C., AND HUGGARD, M. Expected Quality of Service (eQoS) a Network Metric for Capturing End-user Experience. In *Wireless Days (WD)*, 2012 IFIP (2012), IEEE, pp. 1–6.

- PIBIRI, G., MC GOLDRICK, C., AND HUGGARD, M. Enhancing AQM performance on Wireless Networks. In*Wireless Days (WD)*, 2012 IFIP (2012), IEEE, pp. 1–3.

- PIBIRI, G., MC GOLDRICK, C., AND HUGGARD, M. Using Active Queue Management to Enhance Performance in IEEE802.11. In *Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks* (2009), ACM, pp. 70–77.

Chapter 2

Wireless Networking and Real Time Services

The increasing popularity, and growing capabilities, of mobile wireless communications devices provides a strong motivation for the development of future wireless telecommunication systems. These devices have rapidly overtaken desktop computers, home telephones and other communication devices that make use of wired connections [1] [2] [3].

The typical services provided by mobile wireless devices include telephony, video conferencing, media streaming and, of course, non real-time data. Telephone calls are usually provided using VoIP [18] services, while audio and video are used for other services.

Mobile devices typically connect to the internet using Wide Area Network (WAN) and Wireless Local Area Network (WLAN) technologies. WAN services may be delivered using Third Generation (3G) [5] and 4G [4] mobile phone networks, or by WiMax [5] which is included as one of the 4G standards in release 2 [4]. WLAN services are most often delivered by the IEEE 802.11 standards that are collectively branded as WIreless FIdelity (WiFi) [5].

The newer WiFi protocols offer comparable performance per user to that of 4G technologies [20] [21]. They form part of the future businesses model for TELCOs and it is anticipated that they will dramatically reduce the costs for both TELCOs and their customers. This observation motivates the key contribution of this thesis, namely the enhancement of the quality of service and experience that will be offered by future WiFi protocols for realtime and critical services like VoIP and video streaming.

A brief description of wireless internet access methods is provided below. This is followed by a more detailed description of the IEEE802.11 protocol family and its evolution pathways. Real time services used for everyday, everywhere communications are then detailed in the concluding section of this chapter.

2.1 Wireless Mobile Networks

Today's mobile devices typically connect to the internet using cellular and WiFi networks. Mobile phone networks are now in their fourth generation, 4G. These are capable of achieving download speeds of up to 1 Gbit/s. Current WiFi networks achieve a throughput performance of up to 600 Mbit/s but in the next few years it is envisioned that they will rival 4G technologies in terms of throughput and diffusion.

Mobile phone networks are a mix of legacy 2G and 3G technologies alongside the newer 4G ones. 3G mobile devices mainly use Universal Mobile Telecommunications System (UMTS) [5]. The other popular 3G standard is CDMA2000 [5] which operates in a similar way to UMTS. High Speed Packet Access (HSPA) or HSPA+ (HSDPA advanced) are used in conjunction with multiple access schemes, such as Wideband Code Division Multiple Access (W-CDMA), to access the network. Communications between the mobile device or User Equipment (UE) and the Base Station (BS), are generally duplexed using Frequency Division Duplex (FDD) [5]. 3G systems typically operate around the 2.0 GHz frequency band, but this varies from country to country depending on licensing agreements. Four QoS classes are provided: VoIP and audio streaming are given the highest priority, then video streaming; while services that require a lower priority, such as non-real time data transfer, are managed by the final two classes [5]. Of the two lower priority classes, one is used for best effort traffic and the other for background traffic [5].

By 2020 it is anticipated that most internet traffic will be generated by 4G devices [1]. 4G traffic is IP traffic only [22], thus TELCOs will offer IP services and effectively become Internet Service Providers (ISPs). 4G networks offer similar service capabilities to those of existing WiFi networks but they achieve higher performance in terms of quality assurance and the security provided. Many telecommunications companies are repurposing their existing 3G architecture to deliver 4G to their customers.

The technologies currently used for 4G are WiMax and Long Term Evolution Advanced (LTE-A) [4]. WiMax is based upon the IEEE802.16 protocol [5]. IEEE802.16m is known as WiMax Profile 2.0 and fits the 4G requirements [4]. It works in a frequency range from 450MHz to 3.6 GHz [23] and communications are duplexed using Frequency Division Duplex (FDD) and Time Division Duplex (TDD) over the digital encoding Orthogonal Frequency-Division Multiplexing (OFDM). OFDM is also used for multi-user channel access. WiMax uses a time slot based scheduler to manage channel access. Time slot allocations are managed by the base station (BS) and used to control the QoS provided to the clients; hence it is necessary for the mobile device to maintain an almost constant connection to the BS.

WiMax can be used in rural or low density population zones as a replacement for a wired backhaul. IEEE802.16 [23] defines five QoS classes: a very high priority QoS class and the four QoS classes specified for UMTS. WiMax achieves its best transfer rate when using 64 Quadrature Amplitude Modulation (QAM) with a channel width of 20MHz [24].

LTE Advanced [25] [22] is one of the few defined and tested wireless telecommunications standards that meets 4G requirements. LTE Advanced introduces some new structural features such as the use of Multiple Input Multiple Output (MIMO) [21] [26]. Section 2.1.4.1 of this dissertation provides a more detailed description of MIMO. Communications are duplexed using FDD and a mix of OFDMA and scalable FDMA for the uplink. LTE advanced includes other features; for example, cognitive radio is used to optimise the radio communication parameters and an Evolved Node B reduces the mobile node architecture complexity [25] [22].

2.1.1 WiFi Networks

The mobile phone technologies listed above provide network access for mobile devices. An alternate means of network access is provided by WiFi. This has some advantages over cellular networks. First of all it is much cheaper to deploy. At a minimum it requires a Digital Subscriber Line (DSL) and a small router with a wireless interface as an Access Point (AP). It does not require the installation of any pylons or antennas.

Secondly, WiFi can provided network access for each Small Office/Home Office (SOHO) and can also be used for local or personal area network communications. Users are managed locally and do not depend directly on TELCOs. This makes the network much easier to configure and the associated costs are much lower than for mobile phone networks. This can be considered a disadvantage for the TELCOs, as they lose direct commercial control over their customers. On the other hand customers do not have access to direct support from the TELCOs.

WiFi uses the IEEE802.11 [5] family of protocols. They are backward compatible and this gives the user the opportunity to choose between protocols that are best suited to their needs. WiFi has many of the same features for quality assurance as a cellular phone network. The work detailed in this thesis seeks to expand the quality assurance features of future wireless networks.

One architecture used to provide internet access for mobile devices is an infrastructure based wireless network [5]. This is the most popular architecture and is used for both WiFi and cellular networks and is illustrated in figure 2.1. In an infrastructure based wireless network mobile nodes communicate directly with the AP or BS, and the AP acts as gateway between the wired backhaul network and the wireless network. By contrast, in an ad hoc network architecture all the mobile devices can communicate with each other without the need for a gateway node. This is used for many mesh and sensor networks [5]. The nodes on the left hand side of figure 2.1 are wired nodes, and act as both traffic sources and destinations. The nodes on the right hand side of figure 2.1 are mobile devices and these are also potential traffic destinations or sources. For example, during a voice or video call a mobile phone will be both a source and a destination node for traffic; while a tablet downloading a file from the network is a mobile destination node.

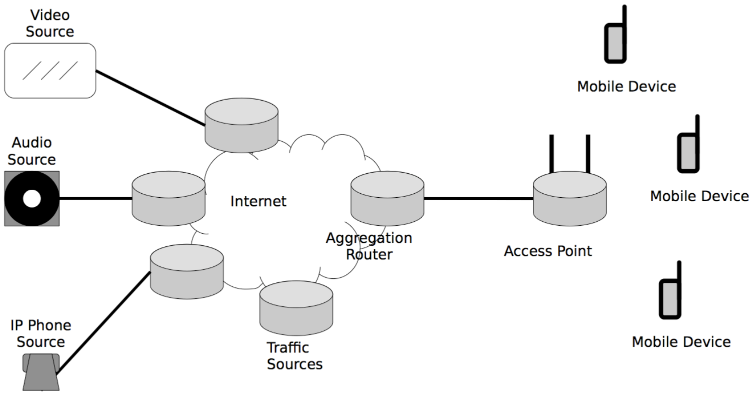

Fig. 2.1: Infrastructure Wireless Network

The typical traffic sources are shown in figure 2.1. These include IP-based phone calls, audio sources such as Internet radio and video.

The AP in an infrastructure wireless network has the potential to be a network bottleneck. This occurs when the rate at which traffic arrives at the AP along the wired connection exceeds the channel throughput available at the AP. This causes the queue length at the AP to grow until it reaches its maximum value.

The theoretical maximum throughput [27] achieved by future wireless network protocols will be comparable to the capacity of the wired link connecting the AP to the rest of the network. However, this may not be sufficient to prevent congestion due to buffer overflow. This is because the wireless channel is shared amongst the mobile stations and the available throughput is reduced due to the use of collision avoidance mechanisms, see section 2.1.2, the number of mobile devices connected to the network and the services provided by the network. From this it can be concluded that the WiFi AP is a key network node where the traffic has to be carefully managed.

An ad hoc wireless network without any communication link to other networks is an Independent Basic Service Set (IBSS) [5]. When the wireless network can communicate with other wireless networks it is known as a Basic Service Set (BSS). The backhaul of a wireless network, together with the backbone network is the Distribution System (DS). Through the backhaul the wired traffic is transported to and from the AP [5]. Multiple wireless networks can be joined together, with the DS, creating an Extended Service Set (ESS) [5].

Wired links are typically full duplex and operate in both directions at the same time using different physical cables; by contrast, mobile devices on a cellular network communicate in both directions using FDD channels [5], while WiFi uses TDD for communication between the mobile stations and the AP. Even when technologies like MIMO or MU-MIMO are used, see sections 2.1.4.1 and 2.3.1, the communication is not full duplex.

In the following section a more detailed description of the protocols used in WiFi networks is provided.

2.1.2 IEEE802.11 Protocols

IEEE802.11 [5] was standardised in 1997 and defines OSI layers 1 and 2 for wireless communication over the 2.4GHz frequency band. Its initial throughput of 1 Mbps was later extended to 2 Mbps and today's variants achieve almost 7 Gbps.

IEEE 802.11 also specifies the following services: Station Services (SS) associated with a mobile device and Distribution System Services (DSS) associated with an AP [5]. SS are services such as authentication, deauthentication, privacy and MAC Service Data Unit (MSDU) delivery [5]. DSS are similar to SS as, for example, they allow for association, disassociation, distribution, integration, and re-association [5]. All these services are either provided with special packets called beacons or inferred from normal communications via packet headers.

The most popular channel access method in IEEE802.11 is the Distributed Control Function (DCF) [5] [28] [29]. DCF uses Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) to distribute traffic and avoid collisions between data packets. Before sending a packet each station, including the AP, has to listen to the channel for a time called a Distributed Inter Frame Spacing (DIFS). Each station that finds the channel idle after a DIFS has to wait a random backoff time before starting to transmit. This backoff time depends on the size of the contention window. The contention window is a random value between 0 and CW. The stations start decrementing their backoff timers when the channel is idle. If one of the stations starts transmitting, all the other stations stop decrementing their backoff timers and wait until the transmission is completed to continue decrementing their backoff timers. The channel is sensed by each station every DIFS time. A the station can transmit when its backoff timer is zero and the channel is idle. A collision occurs when two stations transmit simultaneously. In this case the CWs of the colliding stations are doubled [5] [28] [29].

The alternative access method for IEEE802.11 uses a Point Coordination Function (PCF) [30]. With PCF the AP is the coordinator and decides the time and the order in which every station can transmit over the channel. As discussed in [30], PCF is not popular and it is usually not implemented at the AP. A variety of enhancements to PCF have been developed to provide quality assurance [31], but these are quite complex in nature and have not led to a rise in its popularity.

Collision Avoidance [5] can be achieved by sending a Request To Send (RTS) packet and then waiting for the destination node to respond with a Clear To Send (CTS) packet before the transmission of a data packet. Each RTS and CTS packet is sent after an interval known as a Short Inter Frame Spacing (SIFS). After the CTS packet is received, the station sends the data packet. Data packet reception is confirmed to the source by the destination node using Acknowledgement (ACK) packets. Once a data packet is received the destination waits a SIFS interval before sending an ACK packet back to the source.

For backward compatibility control packets like RTS, CTS, ACK or ARP are sent at a lower speed, usually 1Mbps, but their speed of transmission depends on the IEEE802.11 protocol extension being used in the network.

In the following subsection the three of the earliest and most popular IEEE802.11 protocol extensions are considered in more detail.

2.1.3 IEEE802.11a, b and g

The three protocols IEEE802.11a [32], IEEE802.11b [33] and IEEE802.11g [34] are extensions of IEEE802.11 [5] [26]. Together with IEEE802.11n [21] these are the most widely used WLAN protocols in mobile devices; both because they are supported by the equipment available on the market and because they are backward compatible with each other.

Other IEEE802.11 [5] extensions exist, but these often have a specific purpose such as the provision of quality guarantees e.g. IEEE802.11e, or security e.g. IEEE802.11i, or because they can work with special node architectures e.g. IEEE802.11s.

IEEE802.11a, b and g include CSMA for collision avoidance. This feature is optional, but it is to be recommended if collisions between data packets are to be avoided.

IEEE802.11a [5] uses OFDM to transmit data on a 20 MHz channel. IEEE802.11a uses 48 subcarriers for data and 4 pilot carriers, giving a total of 52 subcarriers per channel on the 5 GHz frequency band. On each carrier, it uses Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 16-QAM or 64-QAM modulation. With 64-QAM it can use a 5/6 rate encoder and a guard interval of 800 nanoseconds between symbols. IEEE802.11a has a spectral efficiency of 2.7 bits per second per Hertz at the maximum speed of 54 Mbps on a 20 MHz channel [26]. IEEE802.11a offers transmission at rates that vary between 6 Mbps and 54 Mbps.

IEEE802.11b [5] uses Complementary Code Keying (CCK) modulation on a 22 MHz channel at the frequency of 2.4 GHz. Its spectral efficiency is 0.5 bits per second per Hertz at the maximum speed of 11Mbps [26]. IEEE802.11b achieves data rates from 5.5 Mbps to 11 Mbps.

IEEE802.11g [5] uses a 20MHz channel at the frequency of 2.4 GHz. Its spectral efficiency is 2.7 bits per second per Hertz at a top speed of 54 Mbps [26]. IEEE 802.11g achieves the same throughput as IEEE802.11a when OFDM is used. The data rates for IEEE802.11g vary between 1 Mbps to 54 Mbps. Rates of 22 Mbps and 33 Mbps can be achieved using 8 Phase-Shift Keying (PSK) modulation.

These IEEE802.11 protocols have been superseded by IEEE802.11n and this is considered in the following subsection.

2.1.4 IEEE802.11n

The final version of IEEE802.11n [21] [26] was released in September 2009. It not only increases the available channel throughput but also includes features that improve efficiency and quality of service. It introduces three major features: an increased radio channel size, the use of

MIMO and frame aggregation.

IEEE802.11n includes improvements to the physical layer. Like IEEE802.11a and g, it uses a 20 MHz channel but it also introduces a 40 MHz channel [26]. A single 40 MHz channel offers greater potential when compared to the simple aggregation of two 20 MHz channels. By using a single 40 MHz channel, the stopband between the two aggregated 20 MHz channels can be used for additional OFDM subcarriers to increase spectral efficiency [26]. 20 MHz channels are used at 2.4 GHz and 5 GHz, while 40 MHz channels are usually used only at 5 GHz.

The method used to transmit data is OFDM. There are 52 subcarriers in a 20 MHz channel and 108 subcarriers in a 40 MHz channel [26]. The modulation used to reach the maximum throughput is 64 QAM with a 5/6 encoder rate. The maximum throughput is 65 Mbps in a 20 MHz channel and 135 Mbps in a 40 MHz channel. The throughput can be multiplied by the number of transmitters, up to a maximum of four, using space division multiple access. This gives the maximum throughput on a 20 MHz channel as 260 Mbps and on a 40 MHz channel as 540 Mbps [26]. Another important IEEE802.11n feature to increase the data transmission rate is the reduction of the guard interval and symbol times to 400 nanoseconds and 3.6 microseconds respectively.

The second innovative feature of IEEE802.11n is that it optimises the throughput by aggregating frames [26]. MSDU or MAC Sublayer Protocol Data Unit (MPDU) frames can be joined together and transmitted as a single large frame. A large frame resulting from the aggregation of MSDUs has a single MAC header and tailer. A large frame resulting from the aggregation of MPDUs has one MAC header and one trailer for each MPDU. All large frames have a single Layer 1 header. For both frame aggregation methods a block acknowledgement feature is available [26]. This returns a single acknowledgement frame for each aggregate frame, avoiding the need for a single acknowledgement per frame.

There are two disadvantages associated with frame aggregation [26]. First of all, MSDU frames can only be aggregated if they have the same destination; this is not the case for MPDU frame aggregation. If the number of mobile stations sharing the network is large then it is unlikely that MSDUs will have the same destination. The second disadvantage of frame aggregation is the large number of frames lost if a single large frame is damaged or not received at the destination. This disadvantage is more evident when the collision avoidance feature is not used as a whole frame is lost if a collision occurs.

Both frame aggregation and block acknowledgement are not considered in this thesis for two reasons. First of all, the system proposed in this work does not include features to improve the throughput introduced by the protocol. Secondly, the number of mobile stations involved in the network simulations is large and so it is unlikely that any of the queues contain packets to be sent to the same destination in sequence. When packets to the same destination are not in sequence in the queue the frame aggregation and block acknowledgement features are inefficient and effectively useless.

IEEE802.11n also includes a power saving feature that is useful for constrained mobile devices. The feature switches the radio frequency transmitters on and off, either dynamically or statically, reducing the power used at the transmitter [26].

Like the other members of this protocol family, IEEE802.11n is backward compatible. IEEE802.11n permits different modulations for each spatial stream on the 2.4 GHz frequency band, the legacy preamble and header are managed so as to avoid reception by mobile devices that use older standards. Backward compatibility is not needed if the frequency band used is 5 GHz, and so IEEE802.11n operates on this band using a 40 MHz channel [26].

The increased throughput of IEEE802.11n does not mean that the bottleneck between wired and wireless networks at the AP no longer exists. Using wireless standards with high, or very high throughput, for example IEEE802.11ac [6], the bottleneck is likely to exist because the wired backhaul capacity is almost always greater than the throughput available on the channel. The wired throughput, the huge number of mobile devices connected to the access point and services available on wireless networks all combine to create a bottleneck at the AP.

The other advantage of IEEE802.11n is that it uses MIMO. As this technology is used by other members of the IEEE802.11 family of protocols it is considered separately below.

2.1.4.1 MIMO

Multiple Input Multiple Output (MIMO) [21] [26] technology incorporates multiple transmitters and receivers in the same device. The goal of MIMO is to increase the Signal-to-Noise Ratio (SNR), so it is particularly suited for environments where radio communications are difficult; for example, due to the presence of radio noise. MIMO was designed to overcome problems associated with multipath propagation and reflections, thereby increasing the available throughput on the wireless network.

MIMO achieves different objectives depending on the number of transmitters and receivers used and their configurations. When the number of transmitters is greater than the number of receivers, MIMO can be configured in four modes. The antenna selection configuration chooses the best antenna for transmission based on packet error measurements. Transmit beamforming adjusts the transmitted signals giving rise to improvements in the SNR. Cyclic delay diversity configures the transmitter to send a copy of the same signal on each antenna using different subcarrier frequencies, but the SNR improvements achieved are not that significant. Space time block coding is when blocks of data are sent by different transmitter antennas but in different orders [26].

Using multiple transmitters and receivers, MIMO can be configured to achieve space division multiplexing. This gives rise to a significant increase in the throughput by using a different spatial stream for each antenna [26].

MIMO can be configured as an equaliser when the number of antennas at the receiver exceeds the number of transmitters. In this mode the receiver combines multiple signals to improve the SNR [26].

MIMO provides improvements in throughput, radio signal reception and bit errors. This dissertation focuses on channel access improvements and the management of packet scheduling on a theoretical error free channel, therefore MIMO is not considered. On the other hand the improvements introduced by MIMO are likely to be beneficial to the system proposed in this thesis.

In the next section the quality improvements achieved by the IEEE802.11e protocol are detailed.

2.2 IEEE802.11e

IEEE802.11e [35] [36] is an amendment to the standard that redesigned the OSI Layer 2 of the IEEE802.11 protocol to enhance QoS.

The IEEE802.11a, b and n protocols considered above, all manage traffic using a single queue. In particular, they do not split the traffic into separate macro-categories and they do not use multiple queues or priorities. IEEE802.11e improves upon the two coordination functions DCF and PCF [5] by using Class of Service (CoS) [37] to manage traffic priorities. CoS is a technique developed to split the traffic into separate macro-categories, each of which is associated with a specific queue and a specific priority.

A new coordination function, called the Hybrid Coordination Function (HCF) [35] [38] is used to define two new channel access schemes: EDCA and HCF Controlled Channel Access (HCCA). The traffic is split into Traffic Categories (TC) and each TC is managed in a queue that has an associated priority for access to the channel. EDCA is an improvement of DCF and it manages four traffic CoS. HCCA [35] [38] evaluates the services and sessions associated with each mobile device in order to coordinate channel access. It also considers QoS parameters when managing channel access. HCCA is more complex than EDCA and is a very efficient algorithm in terms of the QoS achieved.

EDCA utilises four Access Categories [35]: AC_0, AC_1, AC_2 and AC_3. Each AC is defined as a collection of parameters that characterise a queue and regulate the channel access priority. Each AC has an associated queue: Q_0, Q_1, Q_2 and Q_3. These queues store packets prior to their transmission on the wireless channel.

The first Access Category (AC) [35], AC_0, also called AC_{VO}, has highest priority; it is used for voice and audio traffic. Q_0 is the queue associated with AC_0. The second AC is AC_1, also known as AC_{VI}. It has a lower priority than AC_{VO} and it is reserved for managing video traffic.

Access Category	CW min	CW max	TXOP	$AIFSN$	Type of traffic
AC_0 (AC_VO)	3	7	1.504ms	2	voice and audio
AC_1 (AC_VI)	7	15	3.008ms	2	video
AC_2 (AC_BE)	15	1023	0	3	best effort
AC_3 (AC_BK)	15	1023	0	7	background

Table 2.1: Summary of AC parameters [35]

Q_1 is the queue associated with AC_1. The third AC is AC_2, also called AC_{BE}. It has a lower priority than AC_1. It is used for traffic that requires best effort; for example, telnet sessions or database access. Q_2 is the queue associated with AC_2. The last AC is AC_3. AC_3 manages the background traffic and is also known as AC_{BK}. It has the lowest priority. It handless traffic such as that generated by file transfers and web browsers. The queue associated with AC_3 is Q_3.

For each AC queue EDCA controls access to the channel using a backoff time [35], a CW that is set to be between a minimum and maximum value and an Arbitration Inter-Frame Space (AIFS) Number (AIFSN). The backoff time is a randomly chosen number of time slots between 0 and the minimal contention window, CW_{min}. Every AC has a different CW_{min} as shown in table 2.1. The CW is doubled every time a collision occurs between transmitted packets. The CW can increase up to the value of CW_{max}.

Each AC has a specific Arbitration Inter-Frame Space (AIFS) [35]. $AIFS(i)$ is the AIFS associated with access category AC_i where, for example, $i = 1, 2, 3, 4$. The AIFS is the interval between frame transmissions, it replaces the DIFS used in IEEE802.11 and varies depending on priority. AIFS is calculated by the formula [35]:

$$AIFS[AC] = AIFSN[AC] \times SlotTime + SIFS. \tag{2.1}$$

The AC with highest priority has the lowest CW and AIFS Number, therefore it is more likely to access the channel before any other AC.

Another feature that characterises AC_0 and AC_1 is the Transmission Opportunity (TXOP).

User Priority	Access Category	Type of traffic
1	AC_BK	Background
2	AC_BK	Background (Spare)
0	AC_BE	Best Effort
3	AC_BE	Best Effort (Excellent Effort)
4	AC_VI	Video (Controlled Load)
5	AC_VI	Video
6	AC_VO	Voice
7	AC_VO	Voice (Network Control)

Table 2.2: Summary of AC map between 7 and 4 ACS [35] [39]

When an AC has access to the channel, transmission is not limited to a single packet, rather packets are transmitted for an interval of length TXOP, separated by a SIFS. Table 2.1 summarises the most commonly defined parameters for each AC [1]. While table 2.2 shows how the four ACs can be remapped to the seven CoS defined in 802.1d [35] [39] and vice versa. These parameters are re-purposed for the IEEE802.11ac [6] protocol.

IEEE802.11e also allows for Automatic Power Save Delivery (APSD) [40] [35] to reduce energy consumption. After the end of a service interval a station automatically switches into sleep mode until the start of the next service period. Service intervals may follow a predetermined schedule or may be triggered by the AP. The block acknowledgement, defined in IEEE802.11n, is also defined in IEEE802.11e to turn off acknowledgements. It is also possible to save time and throughput if the service does not require feedback and packet retransmission.

In the next section the main features of the two IEEE802.11 protocols that are likely to dominate future wireless networks are detailed.

[1] http://www.ieee802.org/1/files/public/docs2008/avb-gs-802-11-qos-tutorial-1108.pdf

2.3 IEEE802.11ac

IEEE802.11ac [6] was approved in December 2013. This very powerful standard significantly increases throughput on WiFi networks and, in conjunction with the existing EDCA standard, is able to provide high quality performance. The actual IEEE802.11ac release is draft 3.0.

IEEE802.11ac is backward compatible with the existing popular standards IEEE802.11a, b, g and n. It operates in the 5 GHz frequency band, and will include most of the important features of its predecessors.

In contrast to IEEE802.11n, IEEE802.11ac increases the available channel sizes to 80MHz and 160MHz. These 80MHz and 160MHz channels use a single spatial stream, while more spatial streams are optional. The modulation used is 256-QAM, 3/4 and 5/6; whereas IEEE802.11n uses 64-QAM, 5/6. Binary Convolutional Coding (BCC) is used to correct the errors at the receiver and Space-Time Block Coding (STBC) is used to improve transmission by sending multiple copies of the frame for each antenna. The throughput achieved goes from 780 Mbit/s for a 160 MHz channel and single antenna up to a maximum of around 7 Gbit/s [7] when multiple antennas are used.

IEEE802.11ac includes MIMO and also introduces the concept of Multi User - Multi Input Multi Output (MU-MIMO). As MU-MIMO is not specific to IEEE802.11ac it will be considered separately in section 2.3.1 below.

Most of the Layer 2 features of IEEE802.11n are included in IEEE802.11ac. These include TXOP, aggregation of MSDUs and MPDUs together with a substantial increase in the maximum frame size and CSMA/CA to avoid packet collision. The high speed data transmission achieved by IEEE802.11ac contrasts with the low speed of the PPDU headers, as shown in [6] and [7], making RTS and CTS control packets extremely onerous and costly in terms of throughput. This is because RTS and CTS packets are backward compatible and so their transmission time is comparable to that needed for much larger data packets. For this reason, this thesis does not use RTS and CTS packets. In addition, the algorithm proposed in this dissertation makes RTS and CTS packets redundant as it reduces the likelihood of collisions.

For this dissertation the 160MHz channel is considered to have a throughput of 780 Mbit/s for packet transmission using Single Input and Single Output (SISO). As discussed in section 2.1.4, Block Acknowledgment (BA) can be used for a sequence of frames with the same destination. However, such sequences are unlikely if the number of mobile stations sharing the network is large and so the BA feature will not be implemented and used in this work. In the wireless configuration considered in this thesis all mobile nodes and access points use the IEEE802.11ac standard. The technical specifications of the IEEE802.11ac protocol, including VHT, packet structure, etc., are given in [7] and [6].

As detailed in section 2.2, QoS enhancements for wireless networks are specified in IEEE802.11e [35]. However, this discussion would not be complete without mention of two other amendments that focus on improving quality: IEEE802.11aa [17] and IEEE802.11ae [16].

IEEE802.11aa [41] [42] [17] introduces specific features to improve the delivery of video and audio streams over a wireless network.

One of these features is Groupcast with Retries (GCR). This feature mainly concerns broadcast traffic and makes it possible to deliver one media streaming to multiple recipients [41]. The intra-access category prioritisation feature introduces two more ACs for video and audio: the Alternate Voice (A_VO) and Alternate Video (A_VI) classes [41]. A Stream Classification Service (SCS) acts to arbitrarily map the media to these new ACs [41]. While these features are beyond the scope of this work, they could be used to extend the functionality of the system proposed in this dissertation.

Overlapping Basic Service Set (OBSS) management [41] is another IEEE 802.11aa feature, but it is not relevant for this thesis as it is mainly concerned with the physical layer. The other key feature to mention is the Stream Reservation Protocol (SRP) [41]. This assigns network resources to deliver the media stream with better quality. This feature requires advance configuration and, unlike the system proposed in this thesis, it does not actively follow changes in the traffic.

IEEE802.11ae [16] [41] uses the QoS Management Frame (QMF) procedure to prioritise management frames. It also defines the interactions between the stations; in particular, it uses beacon or frame exchanges to prioritise frames. In this thesis control packets will be prioritised for all mobile stations and the AP. This is in agreement with the IEEE802.11ae protocol and so this amendment does not affect the results presented in this work.

In order to conclude this overview of the IEEE802.11 family of protocols it is necessary to discuss the functionality and key features of the extended version of MIMO used by IEEE802.11ac, i.e. MU-MIMO. This is included in the subsection below.

2.3.1 MU-MIMO

IEEE802.11ac makes use of Multi User - MIMO (MU-MIMO) [43] [7]. To be more specific, IEEE802.11ac implements DownLink MU-MIMO (DL MU-MIMO). DL MU-MIMO uses different spatial streams to transmit to different mobile stations at the same time. By contrast, the mobile stations cannot communicate simultaneously with the AP, rather they transmit one at a time.

In Layer 1, MU-MIMO [44] mixes beamforming and energy optimisation, known as Null Steering, to optimise transmission to a specific mobile station and multiple communications with a number of mobile stations.

One of the key features of MU-MIMO is at layer 2 where this thesis focusses on. MU-MIMO applies TXOP at Layer 2. TXOP transmission of multiple frames is permitted only between mobile devices with the same AC. A mechanism for improving TXOP between different ACs has been proposed [43]: the AC that has the opportunity to use the TXOP can share the TXOP simultaneously with other ACs.

Even though MU-MIMO provides significant improvements for multiple transmission, it was not made available in the early releases of IEEE802.11ac compatible devices [44].

2.4 Wireless Network Services

Everyday communications involving mobile devices typically make use of a wide variety of services. Real time services are the most popular and these have been experiencing sustained and rapid growth in recent years [1] [2] [3]. They include not only phone, conference and video calls, but also all forms of audio and video streaming. These services are sensitive to packet delay and packet loss [45]. They are also constrained by the available throughput on the link. For these reasons they are generally knows as critical services because packet delay and packet loss are critical for the proper delivery of these services. Other internet traffic services such as email, online chat and social networks are not considered critical services as they are not as time sensitive as real-time services.

The components of a real time service are described in detail in the following sections.

2.4.1 Streaming

The word streaming [19] is used to indicate the transmission of media events over the network. These consist of constant or variable rate flows of packets containing the media data. The streaming events considered in this work are audio and video streams: audio streaming may be a VoIP call or digital audio transmitted over the Internet, video streaming may be a video call or digital video events transmitted on internet. By definition VoIP [18] is a streaming service, but in this work it is considered as a separate method of providing a real time service. This is because it is a low throughput real time service that may sometimes be transmitted on a switched network and therefore it is managed with high priority by the IEEE802.11e and IEEE802.11ac protocols.

A streaming event is live if it is transmitted to the final user instantaneously and so requires minimal buffering. In this case the only possible sources of delay arise from the encoding/decoding and transmission of the data [19]. The stream is not live if it is an event that has already been encoded and completely buffered, or where transmission is deferred until a later time. In this case, packet retransmissions are possible if the buffer at the destination is large enough.

The simplest network protocols used for streaming are UDP and TCP [46]. TCP regulates throughput at the source via adjustments in the Congestion Window size and analysis of the acknowledgements (ACKs) received [47] [48] [49]. The Congestion Window is the number of TCP packets transmitted per Round Trip Time (RTT). This is the time for a TCP packet to go from the source to the destination and for the corresponding TCP acknowledgement to travel from the destination to the source. Every packet that is marked or dropped causes TCP to halve the Congestion Window size.

Unlike TCP, the UDP [10] protocol is not affected by packet dropping or marking events at the network layer; for this reason UDP flows are called unresponsive flows [10]. However, they may have an indirect response to dropping or marking events when feedback to the source is provided by the application layer. For example, the Real-time Transport Protocol (RTP) uses the Real-time Transport Control Protocol (RTCP) [50] [51] to provide feedback to the source about unreceived packets. For completeness, it should be noted that there are two other categories of unresponsive flows [10]: intermittent UDP and TCP flows. These are very difficult to manage because they only exist for an extremely short interval of time.

RTP [50] and the Real-time Transport Control Protocol (RTCP) [50] can also be used with UDP. Both protocols operate at the Application Layer. RTP provides streaming over the network and includes some features to improve quality, like the use of packet sequence numbers. RTCP works in conjunction with RTP providing statistical feedback between the source and destination.

Streams may be generated by the Hypertext Transfer Protocol (HTTP) [52]; for example, when using HTTP Live Streaming (HLS) [2] [53] or MPEG-Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [54]. HLS uses the HTTP protocol and provides the streaming service as a HTTP download, MPEG-DASH is similar to HLS with the additional ability to use selective streaming speeds. Protocols at the application layer are not considered in this thesis, rather the focus of this work is on lower layer packet flows.

The main streaming protocol considered in this work is the UDP protocol. There are many reasons for this decision. First of all, the design and implementation of the work proposed in this thesis uses the simple UDP protocol at the transport layer, automatically neglecting all protocols at the application layer. This allows the user to implement any protocol they wish on top of UDP. Secondly, the features of protocols such as RTP are not required; for example, the use of packet sequence numbers. The system proposed in this thesis also provides some quality features

[2] `http://tools.ietf.org/html/draft-pantos-http-live-streaming-13`

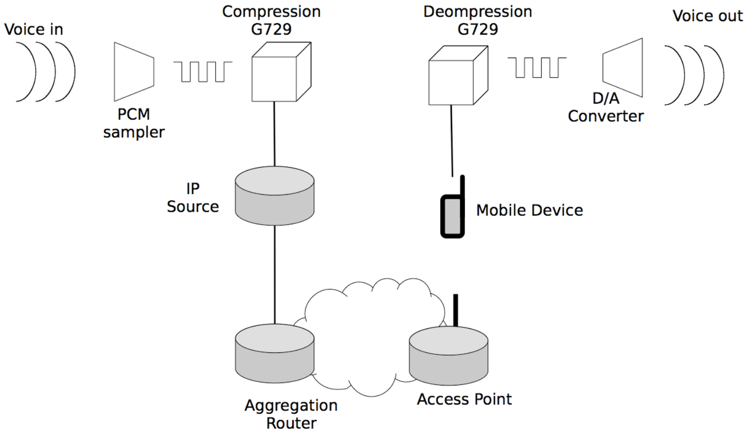

Fig. 2.2: VoIP protocol with compression G.729

of its own. Thirdly, realtime services transmitted live do not require packet retransmissions or protocols like RTCP for the provision of quality feedback; rather they only need a simple datagram protocol. As it is the goal of this thesis to propose a system to improve quality in the most general way possible, protocols other than UDP can only improve the efficiency of the system proposed in this work.

In the next three subsections the most popular encoding standards for VoIP, audio and video streaming are discussed in greater detail.

2.4.2 Voice over IP

VoIP [18] is the service used to replace traditional switched telephony networks for the transport of voice data from the call source to the call destination and vice versa. In today's world, switched networks are no longer needed as VoIP can be used with a simple internet connection.

Figure 2.2 shows the path followed by a voice call from a wired node to a wireless node. The user voice is sampled on the wired terminal and transformed into a digital signal using Pulse Code Modulation (PCM) [55]. The sampling frequency for a voice call in a TDM switched network is 8 KHz [18]. This is twice the frequency of 4 KHz needed for effective reconstruction of the voice signal.

The sample is compressed to reduce the throughput needed for transport from the source to the destination. Examples of popular voice compression algorithms are G.711 [56] and G.729 [57]; for more details on these and other algorithms see [3] [58].

A G.729 VoIP packet is composed of 2 blocks of length 10 ms requiring a speed of 8 Kbps. Some compression algorithms, for example G.711, require higher data rates of the order of 64 Kbps, while other algorithms, like G.729 AnnexD, require less speed, e.g. 6.3 Kbps. G.729 is the most common compression algorithm and it produces data at a constant bit rate data. This makes it an ideal choice for use in the evaluation of the system proposed in this thesis.

The compression process naturally gives rise to delays in the transport of voice data. After compression, the voice data is packetised and inserted into the network to be routed to the receiver's terminal. The process is repeated in the opposite direction for the voice traffic generated on the mobile terminal. Further detail on VoIP and its associated delay are given in [18].

Acceptable voice delay limits are set out by the International Telecommunication Union (ITU) in Recommendation G.114 [59]: for optimal quality the packet delay cannot exceed 150ms. Delays over 150ms and less than 400ms are acceptable but, as a consequence, the voice transmission may experience periods where the quality drops to an unacceptable level. Delays in excess of 400ms are not acceptable as the voice communication quality will not be tolerable [59].

The sources of transmission delay can be divided into two classes: intrinsic and extrinsic. Intrinsic sources of delay are defined as those associated with sampling, compression or coding delays, and also with delays due to the packetisation of data. Extrinsic delays are generated by network nodes; for example, they are queueing, transmission, switching or routing delays.

Intrinsic sources of delay can only be reduced through improvements in the capabilities of the coding algorithm used and in the CPU performance of the node. By contrast, extrinsic sources of delay can be mitigated by the use of algorithms at the node to manage delay [18].

In the next section the encoding method used for audio and video will be detailed.

[3] http://www.cisco.com/c/en/us/support/docs/voice/h323/14069-codec-complexity.html

2.4.3 Audio encoding

In this dissertation the audio considered for streaming is high quality stereo, also known as CD quality; that is 44KHz stereo with 16 bits per sample. This kind of audio can be encoded in different formats, for example MP3 [60] [61], AAC [62] or AC-3 [63].

The audio encoding used in this work is Advanced Audio Coding (AAC) [62] [64], because it has two main characteristics: a constant packet rate and a variable packet size. Using these two characteristics it will be shown that the system proposed in this thesis can be applied to a large range of traffic typologies. AAC is also representative of generic audio encoding and the use of another audio encoding scheme will not change the validity of the results in this thesis.

AAC is used on popular video sharing websites, e.g. YouTube[4]. As an encoding method it is more efficient in terms of quality than MP3 encoding. AAC also includes a large number of scalable encoding features to provide high or low audio quality. Audio encoded in high quality is used for services like digital radio or digital television; while audio encoded in low quality can be used for streaming services like voice messaging in a web chat.

AAC encoding is performed in three stages [62]: in the first step the audio signal is subsampled and the result is transformed from the time domain to the frequency domain. In the second step, a filtering procedure is applied to eliminate non perceptual components; while in the final step quantisation and encoding procedures are applied. This encoding process means that there is a loss of quality associated with the use of AAC.

While AAC is usually configured to provide CD quality audio, it can also be configured to use a large number of channels, each sampled at a high frequency. The frame is the smallest block used for encoded audio, with each frame being decoded independently [62].

Figure 2.3 gives an example of audio encoding; in particular, it shows how frame sizes can vary over time. This sample is taken from the trailer for the movie WarGames[5]. In this figure, the x-axis is the time in seconds and the y-axis is the audio frame size as measured in bytes. Each tiny vertical bar represents a single audio frame. The audio in AAC format is extracted

[4] http://www.youtube.com

[5] https://www.youtube.com/watch?v=hbqMuvnx5MU

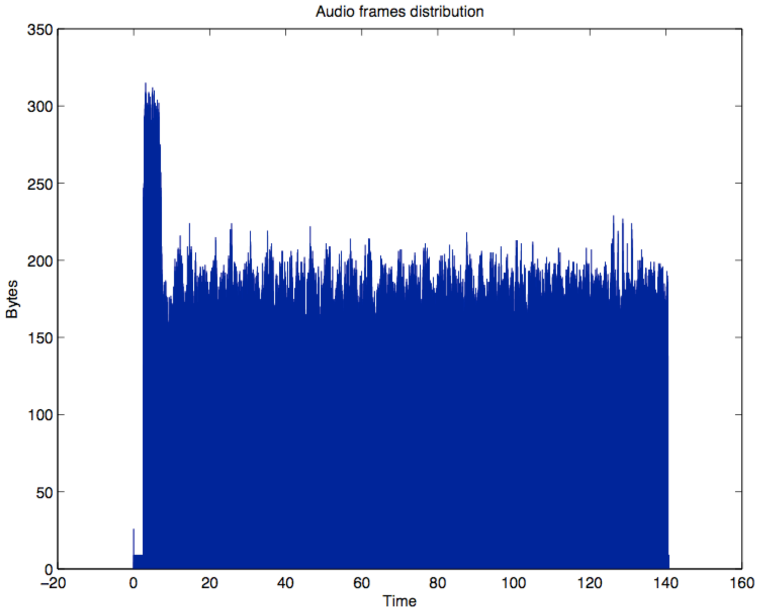

Fig. 2.3: Audio Frames size showing variations due to the nature of the data.

from a video that contains simultaneous dialogue and music. In the sample shown it can be seen that at the beginning of the sample the audio is silent and so the frame size is only a few bytes, then for a short interval of time the sound samples cover a much larger spectrum and the packet size is over 300 bytes, finally the frame size stabilises and is practically constant.

In the case of audio, the packets coincide with the frames, therefore the number of frames per second is the information unit used for quality calculations. It should be noted that while the number of frames per second is constant, the packet size is not.

In the next section the most popular video encoding standard is discussed.

2.4.4 Video encoding

Video coding schemes like H.262 (MPEG-2 Part 2) [65] or H.264 [66] can be used to compress video data. The video coding format H.264 is chosen because it has two key characteristics: a variable packet rate and a variable packet size. These two characteristics are directly related to the traffic typologies: a constant packet rate and a constant packet size are used for VoIP traffic, a constant packet rate and a variable packet size is used for audio while a variable packet rate and variable packet size is employed for video. H.264 is used to represent generic video encoding and can be replaced by another video encoding scheme without altering the validity of this work.

MPEG-4 [66] is one of the common methods used to encode audio and video to be streamed. It is supported by the majority of mobile devices. MPEG-4 uses a file (.mp4) structure like a box, where the encoded audio and video are separated. Video can be encoded using MPEG-4 Part 10 [66], also known like as Advanced Video Coding (AVC) or H.264. It was formulated by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) [66] [19] [67].

Audio can be encoded and structured using AAC [62]. These are the methods used in this thesis. The encoding and compression processes reduce the size of each stream and so increase the number of streams that can be transmitted at the same time on the wireless link. In a similar manner to the way the audio encoding process drops any frequencies that are imperceptible to the human ear [62], the video encoding process drops any redundancies in the frames that are imperceptible to the human eye [67]. A video photogram sequence is sampled and encoded using three kind or frames: I frames, P frames, and B frames [67].

Each I frame [68] [67] is coded and decoded as a single, independent image and is not related in any way to other frames. P frames are not independent, they are encoded and decoded using information from the last I frame. P frames [68] [67] are encoded using the Motion Compensator Algorithm. P frames are used when differences between the current frame and the last I frame are small; for example, if the camera is in motion or when an object is moving. Bidirectional (B) frames [68] [67] are coded using information from the last and the next I and P frames [68].

Figure 2.4 shows a Group of Pictures which is composed of a sequence of I, P and B frames. It is denoted by GoP(N, M) [67], where N indicates the number of frames between two I frames, and M indicates the number of frames between two I or P frames. In figure 2.4 $N = 12$ and $M = 5$.

There are a variety of profiles available for use in the video encoding process. Each profile determines the framing process: a high quality profile has more detail and provides a high level

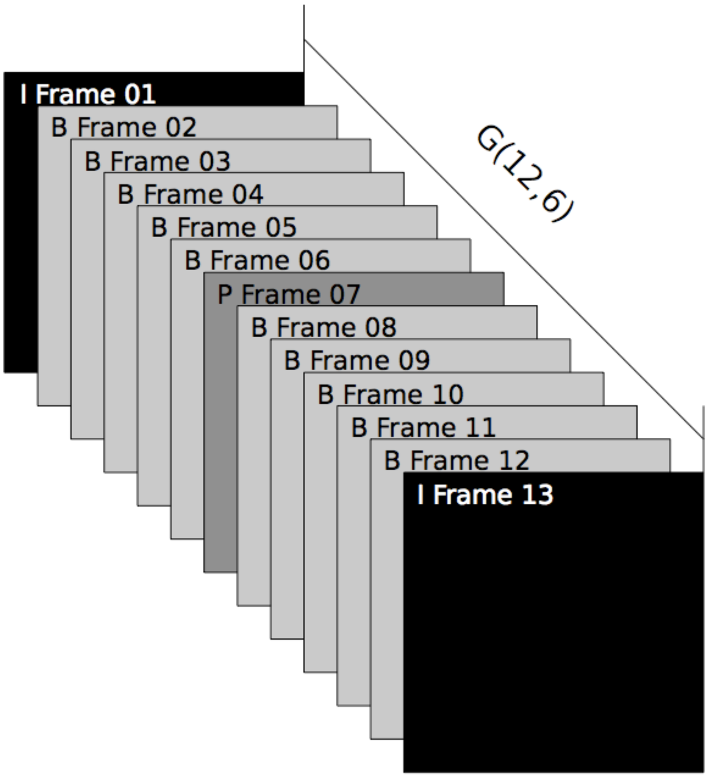

Fig. 2.4: Structure of the MPEG Group Of Pictures [67]. I Frames are in black, B Frames are in light grey and P Frames are in light grey

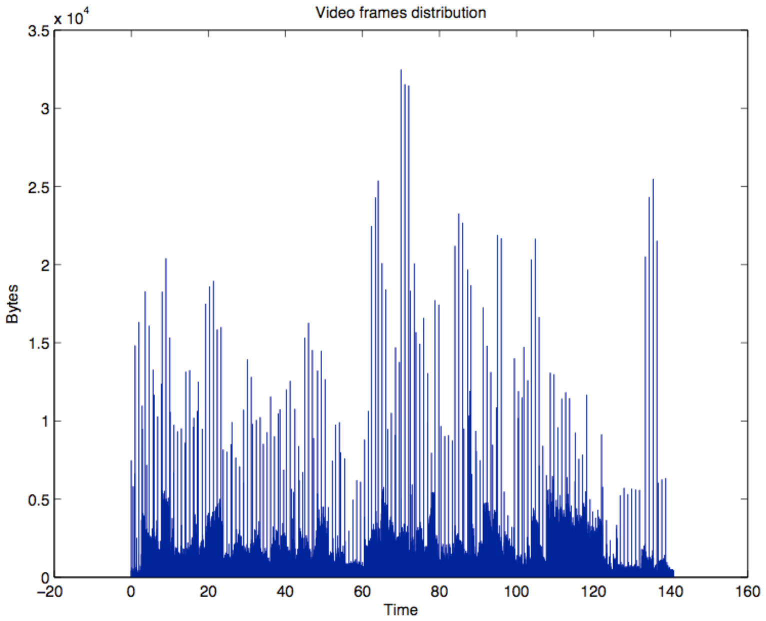

Fig. 2.5: Example of Video Frames showing variations in Frame Size.

of quality; while the baseline profile captures less detail and does not include any B frames [68].

On the network side, video is sent by default as a Variable Bit Rate (VBR) stream because the frame sizes are not constant. While it is possible to force video to be sent with a constant bit rate [69], for this dissertation the worst case scenario of transmission using VBR is considered.

I frames are larger than B and P frames and so they are more likely to be fragmented into a large number of packets; on the other hand fewer packets are required for transmission of P and B frames. Figure 2.5 gives an example of the frame size for a video stream. This sample is taken from the trailer for the movie WarGames[6]. In this figure the x-axis is the time in seconds, the y-axis it the audio frame sizes measured in bytes and each tiny vertical bar represents a video frame. The H.264 format video sample is 140 seconds long and contains several fast scene changes; for example, from outdoors and indoors. In this case the video encoding uses 30 frames per second, with one I frame for every 28 P frames. The figure also shows how video frame sizes are, in general, much more variable in size than those used for audio streaming.

[6] https://www.youtube.com/watch?v=hbqMuvnx5MU

2.5 Summary

This chapter described the wireless protocols and service standards that are used for everyday communication between mobile devices. Real time services form an essential, and growing, element of these communications. The 3G and the newer 4G cellular technologies offer performance and quality assurance mechanisms that exceed those of future WiFi technologies. The first implementation of WiFi was the IEEE802.11 standard. This was improved and extended through changes to Layer 1 in the basic extensions IEEE802.11a, IEEE802.11b and IEEE802.11g. These were superseded by the High Throughput protocol 802.11n which not only included improvements to Layer 1, but also new Layer 2 features and the introduction of MIMO. MIMO technology configures multiple transmitters and receivers at the node to enhance the SNR, optimise the channel and improve the throughput.

IEEE802.11e introduced significant quality improvements at Layer 2. These improvements include quality features to manage the traffic when High Throughput protocols are implemented in the network. They consist of a multi queue architecture and policies for network access for these queues. These features of IEEE802.11e formed the main framework for the definition of quality features in future wireless networking protocols.

Quality assurance for future WiFi technologies is defined in IEEE802.11ae at Layer 2, as an extension of IEEE802.11e. The IEEE802.11ac protocol uses MultiUser-MIMO. MU-MIMO not only improves on MIMO but also specifies procedures and features to allow multiple devices contemporaneous access to the channel, thus enhancing the throughput.

This chapter concluded with a discussion on the provision of the main real time services used for communications: VoIP, audio and video streaming.

In the following chapter the focus shifts to a review of the literature on the provision of Quality of Service and, in particular, Active Queue Management. A description of suitable mathematical models of the behaviour of wireless networks is also provided.

Chapter 3

A Review of Network Quality Provision Mechanisms

The chapter provides a detailed description of previous research related to the central contribution of this thesis. The focus is on three macro areas: Quality of Service, congestion control in wireless networks and Active Queue Management. Most of the work reviewed touches on more than one of these areas; where this is the case a classification is made according to the main goals and objectives of the work.

Quality of Service (QoS) [8] is a collection of several network metrics, taken directly at the physical layer or derived from it, used to measure and improve the quality of the services provided on a network. In the first part of this chapter QoS levels are defined and classified. Of key interest for this dissertation is the QoS metric known as Quality of Experience (QoE) [8]. It is a subjective measure of customer satisfaction that gives an indication of the end-user's perspective of the quality of the service being provided and the effectiveness of the network's operation. It is influenced by customer experience in the use of the service and is of increasing interest to TELCOs.

Realtime services are some of the most challenging services for wireless networks to handle because they are very sensitive to packet delay and packet loss. A description of automatic methods used to estimate QoE for realtime services is provided. Automatic QoE estimation methods have been standardised by the International Telecommunication Union (ITU). They do not depend on user feedback; rather they estimate measurements of QoE using mathematical algorithms that compare the data streamed by the source with that received at the destination. This chapter will describe these in more detail and discuss their limitations.

The next topic discussed in this chapter is that of wireless networking. Existing research related to real-world wireless networks is reviewed and analysed. This forms the background for an overview of the next generation of wireless networks, called future wireless networks. The research findings of relevance to this dissertation are those related to the development of a theoretical model to describe and predict the behaviour of wireless networks. Mathematical models have been developed for single and multi queue systems, and they have been used to design dynamic algorithms which optimise channel access.

In order to satisfy QoS guarantees it is not sufficient to simply improve the performance of the wireless network and enhance existing wireless protocols. It is also necessary to manage the traffic in single queue or multi queue systems. Active Queue Management (AQM) [9] [70] provides an efficient set of algorithms for avoiding network congestion. These algorithms can also be designed to simultaneously guarantee a high QoE in wired networks. This thesis will also show that they are important tools for guaranteeing a high QoE in infrastructure based wireless networks.

The most popular and important AQM algorithms are detailed in terms of both their basic functionality and their efficiency. Other AQM algorithms that adopt techniques of relevance to the design section of this dissertation are also discussed; in particular, AQM schemes that use fuzzy logic and fuzzy control systems. The implementation of AQM in wireless networks forms the starting point for the development of the new AQM algorithms for use in future wireless networks detailed in this thesis.

The chapter is organised as follows: in the next section a detailed description of QoS is provided. This includes both QoE and possible QoS improvements for use in wireless networks. In section 3.2 theoretical models used to capture the behaviour of wireless networks are discussed.

Finally, AQM is defined and a classification of the most important AQM algorithms is provided. This section also details some particular AQM algorithms that are of direct relevance to the work contained in this thesis.

3.1 Quality of Service

Quality of Service [8] is usually defined as a metric that captures how well a service is provided using a standard set of procedures and methodologies. The measurement of QoS has been divided into three layers [8]: Intrinsic QoS, Perceived QoS and Assessed QoS; however, the boundaries between these three layers are neither well defined nor clear-cut. The definitions of these three layers are provided by the ITU [71] [72] [73] and IETF [74].

Intrinsic QoS (IQoS) [8] is commonly known as just QoS. It is an evaluation of quality at the network performance level. The appropriate metrics for quality evaluation at the Intrinsic QoS level are the network parameters. These are useful for determining if a network is performing correctly and if a service is provided satisfactorily from the point of view of the network parameters, but they do not provide a direct measurement of the QoS. If the QoS parameters are good then the service is likely to be provided with high quality and vice versa [8].

The quality of the service provided can generally be empirically deduced from measurements of IQoS. The network parameters provide little direct information related to the service. For example, if the number of packets lost is high, then it can be anticipated that the service is provided with poor quality. On the other hand if the number of packet lost is low, the service is more likely to be provided with good quality. IQoS does not specify an exact percentage of packets lost that is to be considered high or low and consequently the quality level for the end user cannot be easily defined as good, acceptable, bad, etc.

Perceived QoS (PQoS) is a measure of the human perception of the quality of the service provided [8]. PQoS is inferred from the network parameters and it is often used by TELCOs to monitor the quality of the service they provide. PQoS is divided into four classes [73] [75], each of which focuses on an aspect of the QoS from the customer's or TELCO's point of view: QoS offered and achieved from the provider's point of view, and QoS required and perceived from the customer's point of view. An evaluation of components to estimate the four perceived QoS classes in future networks is proposed in [75].

The Assessed QoS (AQoS) [8] is a high level measure of customer satisfaction; for example, when the user decides if they want to continue to use a service or not [8]. Like PQoS, it is a subjective metric as it is based on the user's opinion of the service. Quality of Experience (QoE) [76] is a practical quantification of AQoS, even if QoE quantifies aspects of the PQoS from the user's point of view [8]. PQoS and AQoS may initially appear to be similar; however, they are different as AQoS is not related to the the network parameters but only to customer satisfaction. QoE is measured at the application level; while PQoS, from the TELCOs point of view, is inferred from the network's performance. The metrics used to evaluate QoE depend only on the human, end user opinion and experience of the network quality.

3.1.1 QoE

TELCOs are interested in determining customer satisfaction in order to improve their business proposition and revenue streams; allowing them to bill customers based on the QoE received and also as a means of ranking the service they provide. Therefore TELCOs have been considering measurements of PQoS and QoE as key enablers in the development and planning of new services.

As stated above, PQoS measures the quality perceived by end-users, while QoE measures customer satisfaction. QoE captures end user opinion using a scale called the Mean Opinion Score (MOS) [76]. MOS is a quantisation of the customer satisfaction into five distinct levels [77], these are summarised in table 3.1 [77] [8].

It would obviously be very helpful to determine a direct relationship between the physical network parameters and the end-user opinion. The most significant work in this direction [78] established an exponential interdependency between QoE and QoS, known as IQX. This links together the number of packets dropped and the QoE.

IQX and the QoE estimation method proposed in this thesis differ in three key ways: first of

QoE	**MOS**	**Impairment**
Bad	1	Very annoying
Poor	2	Annoying
Fair	3	Slightly annoying
Good	4	Perceptible but not annoying
Excellent	5	Imperceptible

Table 3.1: Summary of MOS levels [77] [8]

all the method proposed in this work estimates the QoE instantaneously at the node. Secondly, it is inferred mathematically from the network performance parameters. Thirdly, it can be used for all real time services, regardless of the protocol used [14].

The central idea behind IQX is that QoE degrades exponentially with the number of packets dropped. IQX is summarised by the exponential formula [78] [79]:

$$\frac{\partial QoE}{\partial QoS} \cong -(QoE - \gamma) \tag{3.1}$$

The solution of equation 3.1 is [78] [79]:

$$QoE = \alpha \cdot e^{-\beta \cdot QoS} + \gamma \tag{3.2}$$

Where α, β and γ are proportionality factors introduced in the solution of the differential equation. IQX has been used for the voice codecs iLBC and G.711 [78] and captures an exponential relationship between the number of dropped packets and the QoE.

In [79] the relationship between QoE and QoS was analysed for web services. Two possible relationships between the metrics are explored: a logarithmic and an exponential one. The logarithmic nature of the relationship between the QoS and QoE metrics was also noted in [80], where the Weber-Fechner Law (WFL) was used as the basis for determining the relationship between QoE and QoS.

When looking at QoE in a wireless network, the main focus of research has been on its measurement and improvement [81]. This dissertation concentrates on the measurement of QoE at the Access Point (AP) as it is considered a critical point for traffic management. An AP is often a bottleneck for traffic arriving from the wired network. It is the interface between the wired network, where packet loss is negligible, and the wireless network, where packet loss can have a significant impact on service provisioning [12].

There are two types of QoE measurement, those requiring human feedback and those using automatic computational algorithms. Human feedback can be gathered using a scored opinion after a service is provided; for example, by requesting a quality score of a phone call or video streaming using the MOS [79]. The information collected then needs to be statistically analysed to summarise the user feedback received.

The computational algorithms proposed by the ITU for the estimation of QoE can be classified for each type of realtime service that they are used for. The most popular realtime services are phone calls, audio streaming and video streaming. Realtime services were discussed in more detail in section 2.4.

Another way to classify the computational algorithms is as Full Reference (FR), No Reference (NR) and Reduced Reference (RR) algorithms [82] [83]. FR algorithms compare the stream at the source with the stream at the destination; RR [83] algorithms send some parameters to predict the QoE without the need for access to the orginal stream, NR algorithms use only the stream at the destination to estimate the MOS [79].

The QoE of a phone call over the internet may be automatically estimated using the Perceptual Evaluation of Speech Quality (PESQ) [84] algorithm. PESQ is designed to automatically estimate the QoE of speech streaming without the need to ascertain the end-user's opinion. The algorithm simulates the subjective perception of the end-user through objective mathematical approximations. It is a comparative or FR method that requires full information on the speech at the source and at the destination. The source and the destination are compared using the following main stages [85] [86]: alignment of the power levels, filtering and alignment in time, removal of inaudible parts, disturbance calculations and perception estimation through a cognitive model.

Virtual Speech Quality Objective Listener (ViSQOL) [87] is another method that can be used to evaluate speech quality. It is based on the NSIM (Neurogram Similarity Index Measure) [88], a metric to predict speech intelligibility [88].

In 2011, PESQ was replaced by the Perceptual Objective Listening Quality Assessment (POLQA) [89] score. POLQA is a more precise estimate of the MOS as it uses a larger band of the audio signal than PESQ. It can be used to estimate the MOS for high quality audio. Like PESQ, POLQA [90] is a comparative method or FR algorithm. It operates in the frequency domain and it captures distortions between the original source audio transmitted and the audio received at the destination that are not perceived by the PESQ algorithm. POLQA was designed to enhance the performance of PESQ [90] and to obtain a more precise MOS estimation. The increased sampling frequencies introduced in POLQA are comparable with CD audio quality.

A NR method to evaluate speech quality is proposed in ITU-T recommendation P.563: a "Single-ended method for objective speech quality assessment in narrow-band telephony applications" [91]. This does not use the source data as a reference to estimate the MOS. The disadvantage of NR methods is that the distortions have to be known in advance and not inferred through comparisons with the source signal [92]. For this reason NR methods are not considered in this thesis. As this dissertation uses speech sampled at low frequencies, methods like POLQA are not utilised in this work. However, high quality audio at high frequency sampling will be considered in this thesis.

Comparative methods are more precise than non comparative ones, but they have two main limitations. The first limitation is the need to have access to the speech source; this means the evaluation method cannot be used to obtain an instantaneous estimate of the MOS. The second limitation is the algorithm's complexity. The algorithm requires not only the time needed to capture a stream of data but also the time needed to carry out the comparison with the source data.

When an evaluation of high resolution sound quality is required, an automatic MOS score estimate can be obtained using the Perceptual Evaluation of Audio Quality (PEAQ) [93] algorithm. Like PESQ, PEAQ is a comparative algorithm and it compares the audio streaming source data with that received at the destination. It works on audio data that is sampled at a frequency above the 8KHz used for phone calls: the conformance tests are conducted with samples at 48 kHz, 16-bit PCM [93]. As the audio quality is higher than that used with PESQ, PEAQ uses a more sophisticated algorithm and is sensitive to small variations in the streaming samples. The audio analysis results are classified using Objective Difference Grade (ODG) [93] [94] values. ODG scores are in the range from 0 to -4 where 0 corresponds to a MOS score of 5 and -4 corresponds to a MOS score of 1.

In the case of video streaming, the MOS score estimate can be obtained using the Perceptual Evaluation of Video Quality (PEVQ) [95]. PEVQ is a comparative algorithm for video streaming as it needs access to the source video data stream for comparison with that received at the destination. QoE of video streams may also be measured using the Peak Signal-to-Noise Ratio (PSNR) [96]. This standard is based on image comparison and is assumed to have a direct relationship with the subjective end-user opinion. PSNR compares the maximum power of the signal to the noise. A nonlinear relationship between PSNR and MOS was proposed in [97]. PSNR is known to have some limitations as it does not always provide an accurate estimate of the MOS; however, it is easy to calculate when compared to PEVQ and PSNR and is one of the most common QoE metrics for video streams [97]. Objective quality metrics are detailed in [98].

PSNR is not the only alternative mechanism to PEVQ for estimating the MOS metric for video streaming. There is also the Structural Similarity (SSIM) metric [99] and the Media Delivery Index (MDI) [100]. The first of these exploits similarities between images, while the second focuses on measures which affect the quality of video streaming. Compared to PSNR the SSIM metric is mathematically more complex [101]; however, it does show a fairly strong correlation with subjective methods where PSNR has a poor correlation [101].

While [96] details the key limitations of PSNR, it also acknowledges that PSNR is a valid metric for comparison of video with the same content at the same rate and that it is easy to implement. Its measurement precision is sufficiently accurate for the evaluation work required in this thesis.

Another way to measure the assessed quality of a video stream is using the Video Quality

Metric (VQM) [101] [102]. VQM is a comparative method with the following steps: calibration, estimation of features, comparison of features with the original video, calculation of the VQM measure. The VQM method is mathematically very complex [101], but it has been shown to correlate well with subjective metrics [101].

The Moving Pictures Quality Metric (MPQM) [101] [103] is a comparative measure that considers two aspects of human perception. The first one is sensitivity of the human eye to the perception of signals, the second one is the human eye's sensitivity to the combination of multiple signals. MPQM is not always reliable when compared with other subjective methods [101]. MPQM is a mathematically complex method [101] and it has a varying [101] correlation with subjective methods.

Algorithms like PESQ, PEAQ [93] and PEVQ [95] aim to reproduce human perception, so their goal is to provide a QoE estimate using the MOS scale. In addition to their need for access to the original source data, they are computationally intense and therefore they are onerous for a network node to calculate and cannot be used to provide an instantaneous QoE estimate.

IQoS parameter thresholds that affect the QoE evaluation of voice, audio, video and text of the end-user are summarised in the ITU-T G.1010 [104] categories. A specific description of methodologies to evaluate the quality of television pictures from the end-user point of view are given in ITU-R BT.500 [105]. ITU-T P.911 [106], and its correction ITU-T P.911C1 [107], describes methods to evaluate the quality of unidirectional multimedia, audio and video streams. Finally, ITU-T P.913 [108] proposes methods to subjectively evaluate internet television quality. This recommendation can be used to compare multiple audiovisual streaming devices and the performance of an audiovisual streaming device in multiple environments.

The discussion above shows a key limitation of existing QoE estimation methods: there is no mechanism for instantaneous QoE estimation. If such a metric existed it would not only provide feedback for improved network management, but would also permit a network node to manage traffic and provide QoS guarantees.

3.1.2 QoS in Wireless Networks

In this section the discussion shifts to the provision of QoS in wireless networks and the focus moves from the challenge of instantaneous QoE estimation to the estimation of QoE in a wireless environment.

Three approaches to improve QoS are considered: prioritisation, combining features across multiple layers and error recovery. The provision of QoS guarantees in a wireless network has mainly focused on backhaul and MAC layer improvements. Some approaches look to guarantee, and improve, the QoS by prioritising traffic. In [109] the in-frame packets for video streaming are prioritised. The scheme is proposed in a EDCA environment and it reorganises the video transmission over the various Access Categories (ACs). The scheme improves the PSNR and, consequently, the QoE.

Other approaches look to combine features across various network layers. In [110] a 3G architecture is developed to provide services with better QoS network parameters. This approach is based on the allocation of resources and QoS functionalities at different network layers. Scalable Streaming Video Protocol (SSVP), as proposed in [111], operates over UDP and is designed to provide video streaming. It focuses on optimising network QoS parameters.

Alternate approaches have focused on network error recovery. [112] reviews Forward Error Correction (FEC) in terms of a new protocol and architecture to use for providing improved multimedia services. It also describes how the perceived QoS for video services can be improved when packets are lost in bursts. Bursty packet loss is a key consideration for the model developed later in this thesis.

The use of prioritisation and the combination of features across multiple layers are approaches that are adopted in this thesis. They are discussed in greater depth in the following section. An exhaustive and complete description of link sharing for various classes of traffic and QoS optimisation for real time services is provided in [37]. It also considers prioritisation of traffic and hierarchical link sharing. This may be considered as a practical application of CoS [37] provision. This has been defined in a number of ways [8], and plays an important role

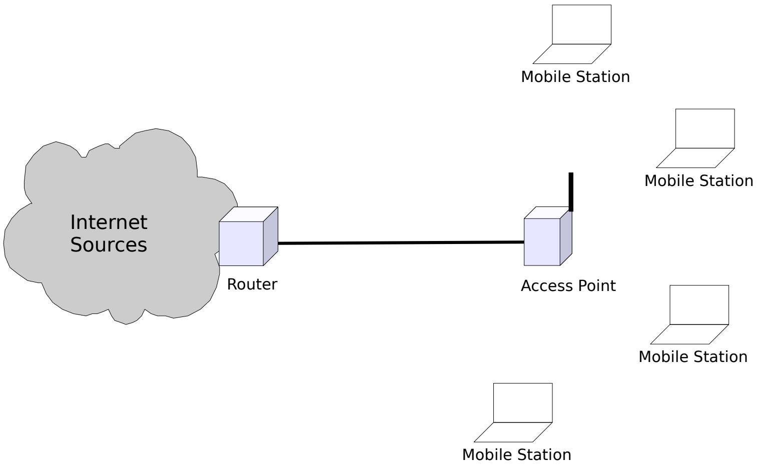

Fig. 3.1: Wireless infrastructure architecture

in wireless network QoS provision. CoS will be discussed in more detail below.

In this section QoS in both wired and wireless networks has been considered. In the following section the focus shifts to wireless networks and their associated protocols.

3.2 Wireless Network Architectures and Protocols

There are many wireless communication architectures and protocols; however, this research focuses on infrastructure wireless networks that utilise the ever-evolving IEEE802.11 [5] suite of protocols. The contribution of this work is not restricted to this family of protocols and may be easily extended to other network environments; for example, cellular networks and WiMax [5]. Future wireless networks were discussed in greater detail in chapter 2 as they are an essential target for the work contained in this thesis.

There are two main wireless architectures: ad hoc wireless network and infrastructure wireless networks [5]. An infrastructure wireless network is shown in figure 3.1. Two aspects of the infrastructure network architecture affect the QoS [12]. The first of these is the wireless channel. The channel is the only link between the Access Point (AP) and all the mobile stations and has to be shared between them. The second issue is the bottleneck at the Access Point. It is anticipated that most of the traffic crossing the AP is in the downlink direction; that is from the AP to the mobile stations. Downlink traffic arrives from the wired network where the link typically has a much greater capacity than the wireless link, and can also be used bidirectionally.

Both these challenges are considered in [12] and [11] and are explored in greater depth in section 3.3. In the following, the channel sharing problem is analysed in more detail.

Ad hoc wireless networks, also known as peer-to-peer networks, [5] do not have a defined AP; therefore, the nodes can communicate with each other directly if they are in communication range or indirectly through intermediate nodes. There is no fixed AP and the routing protocol dynamically changes the path packets take depending on various aspects of the network and the resources available. Some ad hoc wireless networks have a coordinator node; these are typically not access networks, rather they are used as the architecture for Personal Area Network (PAN) or sensor networks and employ protocols other than the IEEE802.11 suite [5] or they work in conjunction with an infrastructure network to form a wireless mesh network.

Even when technologies like MIMO or MU-MIMO, see sections 2.1.4.1 and 2.3.1, are used in a wireless infrastructure network; the wireless channel will be shared in at least one direction, such as in MU-MIMO. In these cases the available channel throughput is shared between the mobile stations; moreover, the channel capacity is drastically reduced due to the use of MAC and physical layer headers and preambles.

In [27] the Theoretical Maximum Throughput (TMT) is calculated for the a, b and g variants of the IEEE802.11 protocol. The TMT is the maximum throughput available, in theory, on a wireless network with averaged backoff time when there are no errors or collisions. TMT is defined as the ratio between the MAC layer Service Data Units (MSDUs) and the time needed to transmit them, thus TMT depends mainly on packet size [27]:

$$TMT = \frac{Size_{(MSDU)}}{Delay_{(MSDU)}} \tag{3.3}$$

It should be noted that it is not correct to consider the nominal throughput available in a wireless network as the effective throughput available at the transport layer. This is because headers and delays introduced by lower layers substantially increase the time needed to send a packet over the network.

The TMT is an overestimation of the effective throughput achievable on a wireless network [12], mainly due to the TCP protocol [49]: for example, TCP acknowledgements do not affect the data transmitted at the transport layer, rather they reduce the time available to transmit TCP data packets. The throughput is also reduced by unavoidable collisions when the number of mobile stations accessing the network is large and by the necessary transmission of control and routing packets, even if the number of these packets is small [27].

Another important consideration is the variety of IEEE802.11 protocols accessing the network simultaneously. It is not unusual to find various IEEE802.11 protocols with different relative speeds accessing the network at the same time. Thus making it impossible to accurately estimate the TMT. The coexistence of mobile stations with different bit-rates is addressed by changing the initial contention window [113].

Improvements in wireless link utilisation is another issue to consider. Improvements to the link utilisation in a network implementing the IEEE802.11e [5] protocol have been proposed [114]. These employ a dynamic Contention Window (CW) adaptive technique to increase throughput. IEEE802.11e, see section 2.2, is an existing enhancement of the IEEE802.11 protocol suite that uses a multi queue system to provide acceptable QoS guarantees. It achieves this by managing the CW using the information available on the wireless channel status.

An important method to provide and guarantee high levels of QoS is Differentiated Services (DiffServ) [115] [116]. DiffServ defines router behaviour for different classes of network traffic. The DiffServ Request for Comments (RFC) considers traffic management [117], the definition of different CoS [118] and the provision of QoS improvements [119].

CoS [37] is method to combine together and manage similar types of traffic with differing priorities; for example, audio, video, e-mail and web traffic. The methods used to combine different traffic types, depend on the nature of the traffic and the network technologies used [8].

Traffic that has ben classified according to CoS is then managed by services like DiffServ [115] [116].

DiffServ [115] [116] is designed to improve and guarantee QoS. It works by splitting the traffic into classes and then managing each class according to a predefined set of rules. DiffServ implements the well known Per Hop Behavior (PHB) configuration: a set of rules which defines the priority and traffic policies to apply to each traffic class crossing the node. Traffic classes at the node are recommended and specified so that DiffServ can be used across multiple routers. The DiffServ [115] [116] definition itself allows for the use of 6 bits in the IP header to configure a traffic class. A class of traffic can be defined based on traffic type or on physical IP fields; for example, by source or destination addresses. While DiffServ has the advantage of being able to manage different traffic types, it is not designed to manage the classes of service themselves when the traffic mix changes. This is a big disadvantage of DiffServ.

Another disadvantage is related to link capacity: low link capacity typically gives rise to an increase in packet drops and hence to a reduction in QoS. This has a significant impact in wireless networks, where the AP can often be a network bottleneck.

Packet loss in the wireless environment [120] will now be considered in more detail. As wireless channels can be affected by fading, packets can be lost during transmission between the AP and the mobile stations and vice versa.

Two situations can occur after a packet is transmitted [120]: the packet is successfully received by the destination either without error or with errors corrected, or the packet is not successfully received by the destination either because it has been lost or because it contains too many errors. The latter situation can create problems for traffic management systems and the behaviour of TCP.

In the next section the focus turns to mathematical models used to describe the behaviour of wireless networks.

3.2.1 Mathematical Models of Wireless Networks

One of the most interesting aspects of research on wireless networks is the use of mathematical models of channel access and sharing amongst the nodes. The mathematical description of traffic behaviour plays a very important role in the management of traffic crossing the AP and consequently in the provision of guaranteed QoS levels. In this section mathematical models that describe the traffic mix in a wireless network are explored in more detail.

The DCF method discussed in 2.1.2 is used to coordinate the channel access for transmission between the mobile stations in a wireless network. The Bianchi model [123] is an analytical model used to calculate the throughput available in a wireless network when the Distributed Coordination Function (DCF) and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) are used. The model is built using a Markov chain to estimate the backoff window size and, consequently, the probability of channel access and collisions [123].

A mathematical approach to model EDCA is presented in [124]. The probabilistic model proposed achieves results that are comparable with those proposed in this thesis. The model is focused on the calculation of the probability that a station wins access to the channel. However, the mathematical complexity of the model limits it applicability.

An analytical model to evaluate the throughput of an EDCA environment in saturation conditions is proposed in [125]. It uses mean value analysis [125], a technique that is computationally less complex than traditional Markov chain methods.

A Markov chain model of EDCA is proposed in [126]. This extends the backoff estimation methodology through the inclusion of appropriate AIFS and Contention Windows (CW). The model is then used to estimate throughput, delay and packet loss. Unfortunately, the framework only captures saturation conditions and, moreover, the analysis includes the use of a Markov chain, therefore is not immediate to be used at the node.

The EDCA method, discussed in 2.1.2, is an evolution of DCF where the mobile stations and the AP implement multi queue systems. The coexistence of mobile stations that use both DCF and EDCA methods in the same wireless network is considered in [127]. A controller to manage the Contention Windows (CW) of the stations that implement the DCF method to access the channel is considered. The technique is called Dynamic ACK Skipping (DACKS) and a Proportional Controller is applied to avoid growth of the CW due to missing control packet acknowledgements at the MAC layer.

An alternate extension to capture wireless network behaviour where the EDCA method is used to access the channel is proposed in [128]. The analysis focuses on estimating the throughput and delay but adopts a different, more complex approach than in [126]. Like [126], the focus is on EDCA in saturation, and to fully incorporate the AIFS the model is complex and not easy to calculate instantaneously.

Another solution to describe, and then manage, traffic in a wireless environment focuses on the CW. In [129] the General Contention window Adaptation (GCA) algorithm is proposed. This algorithm manages the CW size and it is shown to be efficient, to optimise throughput and to guarantee fairness. The GCA algorithm is complex when compared with that proposed in this thesis and, in addition, the algorithm proposed in this thesis is part of larger scheme to manage wireless network traffic.

Models of wireless sensor networks that use a probabilistic approach involving combinatorics rather than Markov chains are of particular relevance to this thesis. In [130] the probability of a collision in an ad hoc wireless environment where the DCF method is used to access the channel is estimated using a binomial like formula. In [131] the binomial theorem and probabilistic methods are used in the design of a protocol for saving power at a node in a wireless ad hoc network.

A key advantage of this methodology is that the calculations involved are significantly less complex than those required for a Markov chain model. This lack of complexity offers a distinct advantage as it is necessary to recalculate probabilities each time the protocol and network parameters change. The computations required can be easily carried out at the node. The main disadvantage of this method is in a reduction in the precision of the results obtained relative to those achieved using a Markov model.

The model proposed in this dissertation is entirely based on combinatorics and address the need for a precise, simple model to describe the traffic and channel access within a future wireless network. It differs from the previous approach because it is designed for a general wireless network where the EDCA method is used to access the channel with different priorities.

3.2.1.1 Other Mathematical Concepts

Before this section concludes it is necessary to discuss three other mathematical concepts of relevance to this dissertation. The first, the German Tank Problem [132] [90], is an empirical method that can be used for calculating the average CW size. The second concept is the Bernoulli distribution [133]. This can be used to determine probabilities for a two state system. The final concept is that of combinatorics [134]. This will be used for determining probabilities for packet loss in chapter 4.

The German Tank Problem was designed during WWII and was used to estimate the number of tanks in existence by sampling their production serial numbers [132] [90]. These values were then used to estimate the maximum of a uniform distribution. Periodic sampling of the CW allows for its size to be estimated using the German Tank problem formula [132]:

$$CW_{max} = (CW_{MaxSampled} - 1) + \frac{CW_{MaxSampled}}{NumOfSamples} \tag{3.4}$$

Where:

CW_{max} is the maximum contention window size;

$CW_{MaxSampled}$ is the maximum CW in the samples;

$NumOfSamples$ is the number of samples taken.

In order to apply this model, duplicated CW values in the range $[0, 2 \times CW_{min}]$ have to be excluded, as do large CW values that exceed $2 \times CW_{min}$. This method works best over a large sampling time interval and with a large value of CW_{min}.

The second mathematical concept to discuss is the Bernoulli distribution [133] [134]. This is a binomial distribution with two possible states, 0 and 1. If the probability of success, i.e. being in state 1, is p; then the probability of being unsuccessful, i.e. being in state 0, is $q = 1 - p$. This is useful for determining probabilities for a two state system when the probability of being in one of the two possible states is known.

	Mathematical Symbol	Formula
Permutation	$P_{n,r}$	$\frac{n!}{(n-r)!}$
Permutation with repetition	$P'_{n,r}$	n^r
Combinations	$C_{n,r}$	$\binom{n}{r} = \frac{n!}{r!(n-r)!}$

Table 3.2: Summary of the Notation used for Permutations and Combinations

The final mathematical topic to consider is combinatorics. Permutations and combinations [134] can be extremely useful for determining probabilities. Before giving their more formal mathematical definitions, it may be helpful to look at a simple example involving the digits 1, 2 and 3 and the number of possible arrangements of these digits. If it is assumed that the digits cannot be repeated then these can be arranged as 123, 132, 213, 231, 312 and 321. If these are read as decimal numbers then the order in which the digits occur matters, as $123 \neq 132$. This is a permutation without repetition of the digits 1, 2 and 3. It can be seen that there are 6 such arrangements. This concept can be further extended to that of a permutation with repetition where the repetition of digits is allowed. In this case the possible permutations of the digits 1,2 and 3 are 111, 112, 121, 122, etc. In contrast to a permutation, a combination is an arrangement of items where order does not matter i.e. the combination 123 is viewed as being the same as 132, 213, 231, 312 and 321.

More formally, a permutation of n objects taken r at a time without repetition is $\frac{n!}{(n-r)!}$. This is indicated by the symbol $P_{n,r}$. A permutation with repetition of n objects taken r at a time is n^r and is indicated by the symbol $P'_{n,r}$. A combination of r objects taken from a set of n objects is $\binom{n}{r} = \frac{n!}{r!(n-r)!}$ and it is indicated by $C_{n,r}$ [134]. These are summarised in table 5.2

Having considered quality of service and wireless networks this chapter now turns to a discussion of Active Queue Management schemes.

3.3 Active Queue Management

Active Queue Management (AQM) schemes form a category of algorithms designed to act on a router queue to reduce, and ideally eliminate, queue overflows [9] [70]. The main goal of AQM algorithms is to drop packets before the queue overflows. Without AQM the router uses the drop-tail mechanism where arriving packets are dropped when the queue is already full.

The main action of AQM algorithms is to drop packets before the queue is full with the intent of inducing a reduction in the speed of traffic being sent by a source [9] [70]. Another way to achieve this goal is to set the Explicit Congestion Notification (ECN) [122] bit in the packet header. Both solutions work well with the TCP protocol but they do not have any direct impact on the sending rate of sources utilising the UDP protocol. They work indirectly as the effect is achieved through feedback from the application layer at the destination node to the data source.

AQM algorithms not only reduce the queue length, they also reduce the probability of network congestion. Many AQM algorithms have been designed to improve some aspects of QoS throughout packet marking or dropping processes. These can reduce the delay experienced by packets in the queue, and also optimise throughput and network performance by reducing the traffic speed so that it matches the network's traffic capacity.

In addition to reducing the queue size and avoiding network congestion, AQM algorithms are usually designed with a particular aim. The aim can be, for example, optimising traffic in order to guarantee fairness between flows or reducing the packet delay at the queue. The particular aim of an algorithm is often used as a method for classifying AQM algorithms.

The main AQM algorithms can be divided into categories using a variety of classification schemes; for example, by objective or by approach [135]. Some AQM schemes try to reduce packet delays at the queue, e.g. REM [70] and AVQ [136]. Other algorithms, like BLUE [137], maximise link utilisation. Another class of algorithms aims to restrict the queue length so that it is bounded between minimum and maximum threshold values, e.g. RED [9] and its derivatives [138] [139] [140] [141] [142] [143] [144] [145] [146].

A detailed AQM classification procedure is presented in [135] and is based on the algorithm's design philosophy. Some AQM algorithms are designed heuristically, e.g. RED [9], AVQ [136] and BLUE [137], while others are designed using more formal mathematical methods such as Control Theory [147] [148], e.g. Proportional Integral control (PI) [149] or Predictive Functional Controller (PFC) [150]. Hybrid AQM algorithms, like PD-RED [138] and FIPD [151], act so as to not only manage the queue size but also to optimise the throughput. Other AQM algorithms try to optimise the calculation of the dropping probability by utilising two competing metrics [135].

In the following section the AQM algorithms that are of most relevance to this dissertation are detailed. All these AQM algorithms are generally designed for, and applied on wired networks. AQM applications in a wireless environment will be considered later in this section.

3.3.1 Random Early Detection (RED) and its Derivatives

RED [9] is probably the most popular and well-known AQM algorithm. It is capable of managing a large range of traffic mixtures and forms the foundation for the development of many other AQM schemes.

RED divides the queue into three sections using two distinct thresholds: the minimum and the maximum threshold. If the average queue size is less than the minimum threshold, the arriving packet is put into the queue. If the average queue size is between the minimum and maximum thresholds, the packet may, or may not, be marked.

This is determined using two probabilities p_a and p_b. p_b is given by [9] :

$$p_b = max_p \times \frac{avg - min_{th}}{max_{th} - min_{th}} \tag{3.5}$$

and varies between 0 and max_p. Where:

min_{th} and max_{th} are the minimum and the maximum threshold;

avg is the average queue length calculated using an Exponential Weighted Moving Average (EWMA) formula [9];

max_p is the max probability used to mark a packet;

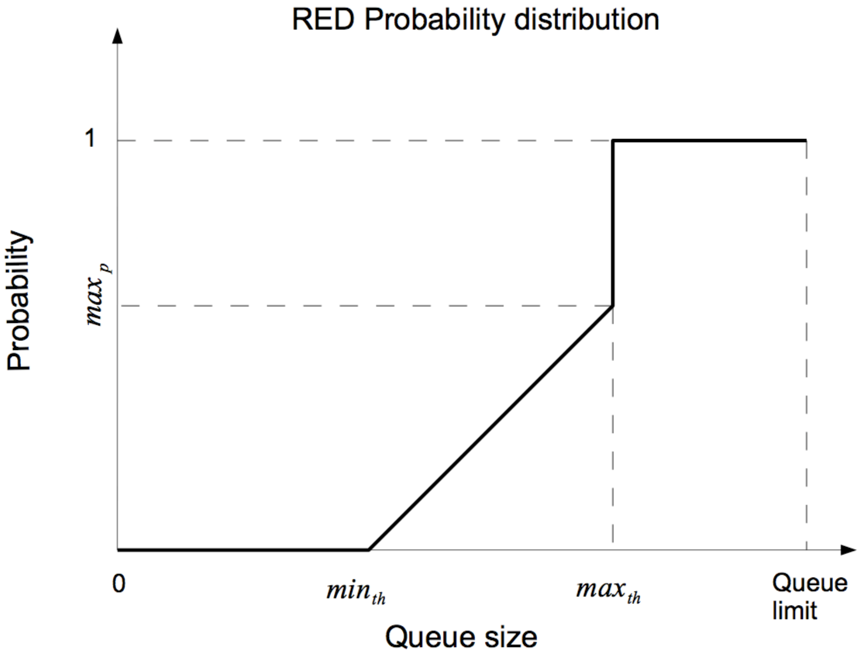

Fig. 3.2: RED probability distribution [121]

p_a is given by [9] :

$$p_a = \frac{p_b}{1 - count \times p_b} \tag{3.6}$$

where $count$ is the previous number of unmarked packets.

For each arriving packet p_a is compared to a randomly generated probability. If the random probability exceeds p_a the packet is dropped or marked, otherwise the packet is unmarked and placed in the queue.

If the average queue size exceeds the maximum threshold then all arriving packets are dropped or marked. Figure 3.2 shows the probability distribution for the RED algorithm. This can be modified by implementing the $gentle$[1] [152] feature. This replaces the step in probability value at th_{max} with a growing, $gentle$ probability between max_{th} and $2 \times max_{th}$ or the maximum queue size, which ever is smaller.

[1] `http://www.icir.org/floyd/red/gentle.html`

The RED algorithm is analysed in detail in [153], where its advantages for use in both wired and wireless scenarios are evaluated. [153] also highlights some difficulties that arise when applying the RED algorithm in wireless networks.

WRED [145] [146] is a version of the RED algorithm implemented in some commercial routers. It was designed to improve QoS and is mainly applied in multi queue systems [37]. WRED changes the RED algorithm parameters for different classes of traffic. In multi queue systems each queue manages a particular category of traffic with a specified priority and the RED thresholds can be adjusted for each queue in the system. It is of note that in some implementations WRED is only applied to TCP traffic. This is an advantage in some situations; for example, when important UDP traffic is passing through the queue. In other situations it can result in an excessive reduction in TCP traffic throughput.

An alternative to WRED is the application of multiple RED algorithms on the same queue. This is the case with the RIO algorithm [154]. RIO is a double application of RED to two different classes of tagged traffic: $input$ and $output$. These refer to the input and output traffic at the router. The router checks if the arriving packet is marked as input or output and consequently it applies the RED algorithm with the appropriate parameters. This means that, in practice, it uses two RED algorithms on the same queue and both algorithms have the same average queue size.

Many AQM algorithms have been derived from the RED algorithm [138] [139] [140] [141] [142] [143] [144]. One RED-derived algorithm of particular interest is RED-PD [138] .

RED-PD [138] aims to improve the fairness between the flows at the queue. It identifies the flow with high throughput using the history of the packet dropped by the RED algorithm. Preferential Dropping (PD) is then applied to these higher throughput flows. RED-PD uses a multiple list identification mechanism to identify the rate of each flow. Flows which have an arrival rate that exceeds the target throughput have an increased dropping probability. Flows which have an arrival rate below the target throughput have their dropping probability decreased.

Adaptive RED (ARED) [139] seeks to improve two aspects of the original RED algorithm: the first is the average queue length which is related to the incoming traffic and to the algorithm parameters; the second is the number of packets dropped. RED usually drops too many packets and so leads to an excessive reduction in throughput. ARED dynamically modifies the maximum probability in order to stabilise the average queue length between the minimum and maximum RED thresholds. The Additive Increase Multiplicative Decrease (AIMD) algorithm modulates the maximum probability following changes in the traffic.

Another RED improvement is Hybrid RED (HRED) [140]. It incorporates key features of other AQM algorithms into RED. HRED is designed to provide queue length stability and it can be configured to meet various QoS parameters specification. HRED uses an adjustment algorithm to identify and follow the traffic changes.

LRU-RED [144] is a combination of the RED algorithm with a Least Recently Used (LRU) cache. The LRU cache stores information about the flows passing through the queue by analysing packets. The cache is used to identify high throughput flows. Different RED configurations are then used to substantially reduce the rate of the high throughput flows relative to other flows passing through the queue.

High throughput flows are a feature of another AQM algorithm, CHOKe [141]. This was designed to establish fairness between flows. Each packet arriving at the queue is compared with an existing randomly selected packet from the queue. If the packets are from the same flow it is more probable that the flow has an high throughput, and so both packets are dropped. On the other hand, if the packets are from different flows the RED algorithm is applied to the arriving packet. CHOKe's main goal is to equalise the throughput across all flows. From a statistical perspective, when the traffic sources are unresponsive flows then CHOKe responds more quickly to bursty traffic than LRU-RED [141] [10].

Two more features are introduced in Self Adaptive CHOKe (SAC) [142]. The first of these manages TCP in a different way to UDP traffic, the second is a self-tuning mechanism for the algorithm's parameters. These are achieved through modifications of the original CHOKe algorithm. UDP traffic is always managed using CHOKe, whereas TCP traffic is only managed if the average queue length exceeds min_{th}. Two probabilities are associated with UDP traffic in order to improve the management of unresponsive flows [10]; these are the probability that the traffic arriving at the queue is UDP, and the probability that the packet selected is from the same UDP flow as the last packet to arrive at the queue. CHOKe performance can also be improved by dividing the queue into regions, where each flow is assigned to a specific region. The algorithm then drops the first packet from the tail of the region that corresponds to the flow of a randomly selected packet.

Channel fairness is a feature of Temporal Fair RED (TFRED) [143]. It aims to provide a fair share of time on the wireless link between the flows by monitoring their throughput. This AQM algorithm also provides a means of congestion avoidance using the traditional RED mechanism.

3.3.2 Random Exponential Marking (REM)

The Random Exponential Marking (REM) [70] algorithm is one of the most popular AQM schemes. REM's main goals are to limit the queue length to a minimum value and to guarantee an acceptable balance between the rate of traffic joining the queue and the link capacity. REM uses a pricing mechanism to calculate the dropping or marking probability [70]:

$$p_l(t+1) = [p_l(t) + \gamma(\alpha_l(b_l(t) - b_l^*) + x_l(t) - c_l(t))]^+ \tag{3.7}$$

Where:

$p_l(t)$ is the price;

γ and α_l are two small, positive constants;

$b_l(t)$ is is the actual queue size;

b_l^* is the desired queue size;

$x_l(t)$ is input rate;

$c_l(t)$ is link or channel capacity;

$[z]^+$ is the $max(z, 0)$.

The price is calculated using the queue input rate and link capacity. The price increases when the input rate increases relative to the link capacity; similarly the price is reduced when the input rate is less than the link capacity. The price is periodically updated and the dropping or marking probability is then calculated using [70]:

$$p(t) = 1 - \phi^{-p_l(t)} \tag{3.8}$$

Where:

$p(t)$ is the marking or dropping probability;

ϕ is a constant greater than 1.

Through this mechanism REM achieves shorter queue lengths with a high link utilisation.

3.3.3 Adaptive Virtual Queue (AVQ)

Adaptive Virtual Queue (AVQ) [136] marks or drops arriving packets based on a comparison of the lengths of two queues that the system maintains: the first one is the real queue, C, and the other is an ad hoc virtual queue, $\widetilde{C}$. The virtual queue length is continuously updated to guarantee the desired link or channel utilisation and its capacity is calculated using the formula [136]:

$$\widetilde{C} = \alpha(\gamma C - \lambda) \tag{3.9}$$

Where:

α is a smoothing parameter [136];

γ is the desired link or channel utilisation;

λ is the traffic rate of the link or the channel.

It is implicit that the virtual queue size is less than, or at most equal to, the real queue length. The AVQ algorithm has been shown to maintain a stable queue length together with the desired link utilisation.

3.3.4 BLUE

Unlike the RED [9] algorithm, which calculates the dropping or marking probability based on the queue length, the BLUE [137] algorithm calculates this probability by considering both the packets dropped at the queue and the link utilisation.

BLUE [137] marks packets that cause the queue length to exceed a queue threshold L with a probability p_m. Time intervals of length $freeze\_time$ are used. Probability p_m is updated every $freeze\_time$ and if the packet is dropped then p_m is increased by a quantity δ_1. On the other hand, if after a $freeze\_time$ interval the link is idle, the probability p_m is decreased by a quantity δ_2. BLUE outperforms RED for some traffic configurations, but only when the parameters L, $freeze\_time$, δ_1 and δ_2 are adjusted to suit the traffic characteristics. The values of δ_1 and δ_2 are chosen together with the $freeze\_time$ to set the probability p_m to a value between 0 and 1 according to variations in the traffic.

One refinement of BLUE is the Stochastic Fair BLUE (SFB) algorithm [137] [155]. SFB uses a Bloom Filter [137] and a matrix of N bins and L levels to identify unresponsive flows [10] and to limit their throughput. The probability p_m is increased or decreased depending on the flow information contained in the associated bin.

The use of the names RED and BLUE caused others to name their AQM schemes after colours; two such algorithms are YELLOW [156] and GREEN [157]. YELLOW [156] monitors the link and its associated queue. A key feature of this algorithm is that it manages the dropping probability based on the observed link load. The GREEN [157] algorithm works in the opposite way from YELLOW. It checks the rate that packets arrive at the queue and it then increases or decreases the dropping, or marking, probability by comparing the arrival rate with the link capacity. If the arrival rate exceeds the link capacity then the dropping, or marking, probability is incremented; while if the rate is below the link capacity then the dropping, or marking, probability is decremented.

3.3.5 AQM using Control Theory

In this section AQM algorithms designed using control theory [148] are considered. The algorithms follow a similar design philosophy to RED in that they have queue thresholds and associated probabilities of dropping, or marking, packets. These algorithms use control theory to determine how this probability should be calculated. The behaviour of TCP traffic can be described through the evolution of its Congestion Window [147] [121]. This is captured in the following differential equations [147] [121]:

$$\dot{W(t)} = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t)) \tag{3.10}$$

$$\dot{Q(t)} = \frac{W(t)}{R(t)}N(t) - C \tag{3.11}$$

$$R(t) = T_p + \frac{Q(t)}{C(t)} \tag{3.12}$$

Where:

$W(t)$ is the TCP window size;

$N(t)$ is the number of TCP flows;

$R(t)$ is the Round Trip Time (RTT);

T_p is the propagation delay;

$Q(t)$ is the queue length;

$C(t)$ is the link capacity;

$p(t)$ is the packet drop probability [121].

The RTT is the time needed for a data packet to go from the source to the destination and for the related acknowledgement packet to go from the destination back to the source [49] [121].

Equation 3.10 describes the TCP Congestion Window evolution: it increases by one unit for each round trip time and is halved when a packet is dropped. $p(t)$ is the dropping probability generated at the queue by the AQM algorithm. Equation 3.11 describes how the queue length evolves. The queue length is calculated as the difference between the number of arriving packets and the channel capacity. Equation 3.12 calculates the Round Trip Time; that is, the time needed for a TCP packet to go from the source to the destination and for the related TCP acknowledgement to be sent from the destination back to the source.

In the language of control theory, the AQM algorithm is the feedback controller in figure 3.3. This figure is a general description of a closed loop system. It includes the desired dropping probability, p_0, and the desired queue length, q_0. It also includes the unresponsive flows [10], u, which are added to the long lived TCP flows, θ. In [10] unresponsive flows are added to the controller loop, these include: short lived TCP flows, UDP flows and intermittent UDP flows.

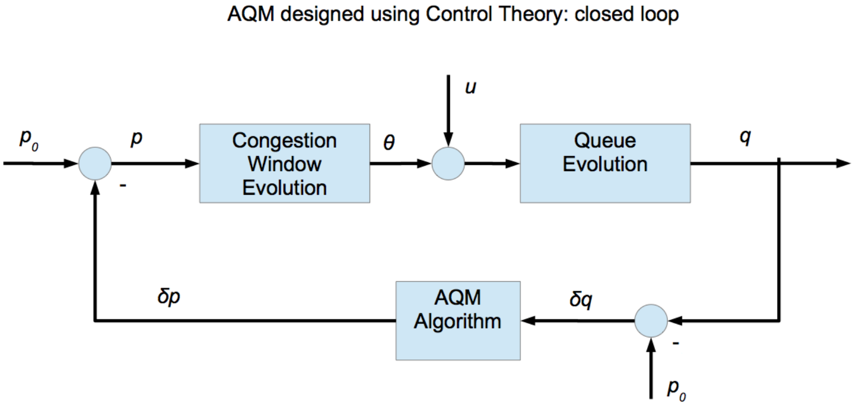

Fig. 3.3: Closed loop AQM controller [10] [149]

AQM algorithms can be designed using traditional controllers, the most popular are the Proportional Intergral (PI) controller [149], the Proportional Derivative (PD) controller [158] and the Proportional Integral Derivative (PID) controller [159]. There are a number of AQM schemes that make use of PID; for example [160] combines a PID controller with a neural network.

The controllers used to design AQM schemes are not restricted to traditional PI, PD, and PID controllers. For example, Predictive Functional Control (PFC) has been used in the design of an AQM scheme [150]. This controller has low complexity and was designed to manage delay in a wired, high speed network. When compared to traditional AQM algorithms and AQM algorithms designed using control theory, PFC performs well in terms of link utilisation, fairness and robustness.

Dynamic RED (DRED), presented in [161], also uses control theory. It uses a simple integral controller to stabilise the queue length and is independent of the number of TCP connections passing through the node.

3.4 Fuzzy Logic and Fuzzy Control Systems

An alternative to traditional control theory is fuzzy logic. This section provides a discussion of fuzzy logic and fuzzy controllers [162] [163] [164] as these play an important role in the overall contribution of this thesis.

Fuzzy logic was introduced in 1965 by Professor Lotfi Zadeh [162] [163] [164]. The term fuzzy relates to the fact that rather than dealing with logical values that must be either true or false, the system allows for the concept of values that are partially true or partially false. In mathematical terms, if true is represented by 1 and false is represented by 0, then fuzzy logic allows for values on a continuous scale between 0 and 1.

A fuzzy system consists of a fuzzy set and fuzzy rules. A fuzzy set is specified by giving a suitable domain, called the universe of discourse, and a membership function to determine elements of the set [162] [163] [164]. For any domain of discourse X, a membership function on X is a function from X to the interval $[0, 1]$ on the real number line, $\mathbb{R}$. For example, if $\mu_A(x)$ represents a membership function for a fuzzy set A where $x \in X$ [163] then $\mu_A(x) = 0$ if the element x is not in the set A, $\mu_A(x) = 1$ if the element x is in the set A, while values between 0 and 1 characterise elements that only partially belong to the fuzzy set A. The value of $\mu_A(x)$ quantifies the grade of membership of the element x of the fuzzy set. The membership function for a fuzzy set may represent the specific, crisp inputs with well known curves, such as triangular, trapezoidal or bell curves [163] [164].

The fuzzification process transform the crisp inputs to a fuzzy controller into elements of fuzzy sets [163]. Fuzzy rules then operate on the fuzzified inputs. The defuzzification process consists of a sequence of operations that map the fuzzy set to a set of specific, crisp output values that can be used outside the fuzzy domain. There many ways to perform defuzzification; of these the three key methods are: Mean of Maximum (MOM), Center of Area (COA) and the Height Method (HM) [163] [164].

The Mean of Maximum (MOM) method estimates the crisp value as the average of the outputs with the highest degree. The Center of Area (COA), or Center of Gravity (COG), method

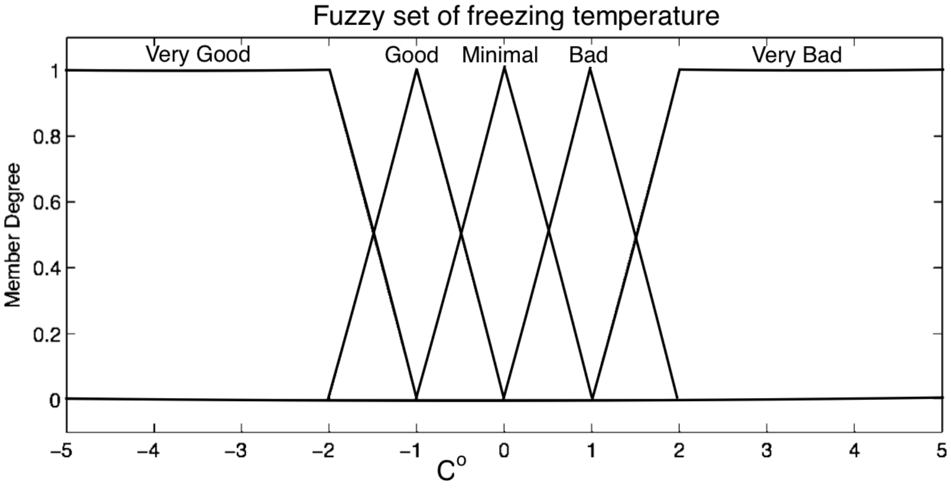

Fig. 3.4: Example of Fuzzy set [163]

estimates the crisp value as the center of gravity of the area identified by the fuzzy set outputs. The Height Method (HM) calculates the crisp value as the weighted average of the maximum of each output [163] [164].

Figure 3.4 shows a temperature set, based on that given in [163]. The Good, Minimal and Bad values are fuzzified by triangular curves, the values Very Good and Very Bad are half-triangular with shoulders indicating the physical limits of the temperature system. Fuzzy rules are of the form [163]:

```
If <condition> THEN <consequence>
```

where there can be more than one `<condition>` and `<consequence>`. The rules can be related to an expression in a natural language, such as English. Linguistic variables can be used to make the rules easier to comprehend. For example, in figure 3.4 the variable name of VERY GOOD, for temperatures below -2^oC might be more easily understood by using the more natural description COLD [163].

One useful application of Fuzzy logic is in the design of controllers. Figure 3.5 shows a diagram of a general closed loop architecture of a fuzzy controller.

The process is the system controlled by the fuzzy controller [164]. The loop input is the one

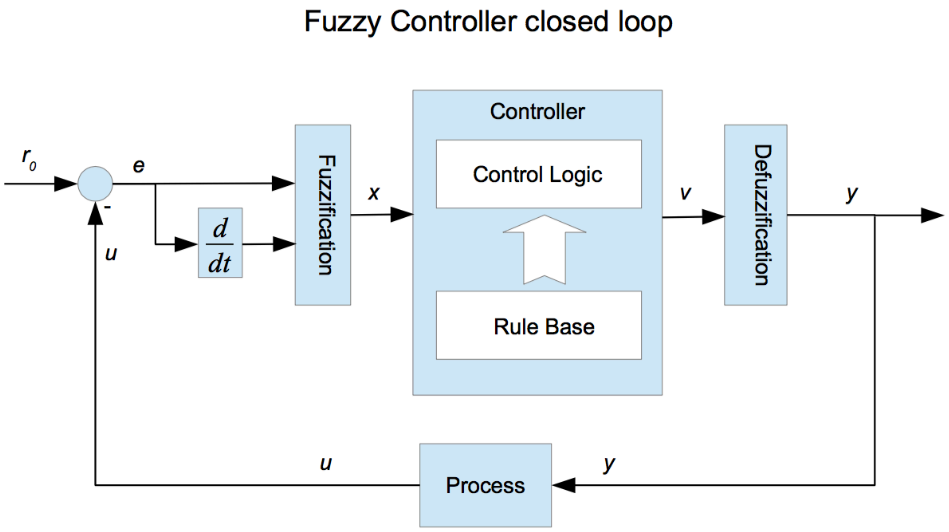

Fig. 3.5: Closed loop Fuzzy Controller [162] [163] [164]

or more reference signals, r_0, directed to the fuzzy controller. The error, e, is calculated as the difference between the reference signal r_0 and an output from the process. The inputs to the fuzzification block are usually this error and the change in the error over the time, known as the error rate. Before the input is fuzzified, gain blocks can be used to re-scale the inputs. If the gain blocks are not needed, they can be bypassed by setting the gain to 1 [164].

The fuzzification block converts the crisp input values into members of a fuzzy set to be used by the fuzzy controller. The reverse operation is performed by the defuzzification block, which converts members of the fuzzy set into crisp output values. The fuzzy set input to the fuzzy controller is represented by x. The fuzzy set output from the fuzzy controller is labelled v. Where necessary, scaling operations are performed in the fuzzification and defuzzification blocks [164].

The fuzzy controller consists of two elements: the rule base and the control logic. The rule base contains all the rules needed to manage the system, while the control logic decides which rules to use to achieve the appropriate input [164].

In the next subsection applications of fuzzy controllers to the design of AQM algorithms are discussed.

3.4.1 AQM using Fuzzy Logic and Fuzzy Control Systems

The use of fuzzy logic and fuzzy control systems in AQM has been explored in [165]. One clear advantage of fuzzy control systems is that they are less complex and more robust when compared to traditional control systems. They are particularly efficacious when a quick reaction is required [166].

A Fast and Autonomic Fuzzy Controller (FAFC) has been proposed in [135]. It is based on fuzzy logic and incorporates mechanisms for self configuration. When evaluated on a wired network, its performance is comparable to that of traditional AQM algorithms and AQM algorithms designed using control theory. The advantages of FAFC are mainly related to network performance; in particular, it improves queue length control and QoS parameters such as delay and latency. FAFC was not designed to manage QoE.

The calculations needed to determine the traditional RED dropping probability involve a significant overhead. The Fuzzy Logic Controller (FLC) presented in [151] replaces this with a Fuzzy-based Intelligent Packet Drop (FIPD). This reduces the calculation overhead by using a decision table to determine when to drop packets.

Fuzzy logic controllers are not only advantageous for queue management, they also allow for dynamic management of CW [167]. Contention window management reduces collisions and optimises throughput. The Contention Window Fuzzy Controller (CWFC) [167] is an example of a fuzzy logic controller used for contention window management on a wireless network where the EDCA method is implemented. CWFC manages the CW size based on two inputs: throughput information, as extrapolated from control packets, and the number of packet retransmissions. It should also be noted that, unlike most AQM schemes, it is an AQM algorithm specifically designed for use in wireless networks.

In the following section AQM algorithm behaviour and design in a wireless network environment will be considered in greater detail.

3.5 AQM in Wireless Networks

The AQM algorithms detailed above were generally designed and implemented in wired networks. A different approach is needed for the design and implementation of AQM algorithms in wireless networks.

Previous work by the author of this thesis has considered the approach needed for AQM implementation in infrastructure wireless networks [12] [11]. This work sought to maintain the original AQM algorithms' characteristics, while adapting them for the wireless environment.

Two main challenges have to be overcome when implementing AQM in wireless networks. The first challenge is that of variations in the traffic mix. In comparison to backbone routers, a wireless Access Point (AP) manages a relatively small amount of traffic and so variations in traffic can be considered to be discrete.

The second challenge relates to the wireless channel. Even when using technologies like MIMO [21] [26] and MU-MIMO [43] [7], the wireless channel is completely or partially shared between the mobile stations and the access point. Consequently the access point is often a traffic bottleneck; moreover, the wired link to the access point often has a much greater capacity than that of the wireless channel itself.

Wireless networks use a single link, the channel, to carry communications between a distribution node, the AP, and the mobile stations. In addition to their usual workload AQM algorithms that operate on wireless networks must also manage control or routing packets; for example, ARP messages that pass through the AP. These packets are essential for the operation of a wireless network. In order to ensure their safe delivery, [12] proposed to put control and routing packets at the head of the queue when the queue is managed by an AQM algorithm. The incorporation of this into five existing AQM algorithms for use on wireless networks was considered. The algorithms used were RED [9], REM [70], AVQ [136] , BLUE [137] and RIO [154].

The RED algorithm was adapted for use on wireless networks by adding a module to move routing and control packets to the head of the queue [12]. Some settings of the algorithm parameters to adapt the RED algorithm to the prevailing traffic mix on a wireless network were also suggested. As the RIO algorithm is simply a double application of the RED algorithm it can be modified in a similar way.

When the REM [70] algorithm is used in an access point, routing and control packets can be moved to the head of the queue in the same way as for RED. Unfortunately this prioritisation process for control and routing packets creates a discontinuity in the price calculation and in the calculation of the dropping and marking probability. A way to overcome this challenge is presented in [12].

AVQ [136] can be modified to introduce control and routing packet prioritisation [12]. This makes use of the Theoretical Maximum Throughput [27] as an estimate of the link capacity, as used in the original AVQ algorithm [12].

BLUE [137], like RED, requires an adjustment of its parameters in order for the algorithm to function correctly in a wireless network that prioritises control and routing packets [12].

A general discussion of AQM algorithms and their adaptation for use in a wireless environment is presented in [11]. In particular, changes needed for the RED [9], REM [70] and BLUE [137] algorithms are presented.

Analysis of RED [9] suggests the use of the Hold-Winters procedure [168] to determine the average queue length at a wireless access point. The Hold-Winters procedure was originally proposed in [168] for the RED algorithm. In [11] the procedure is adapted and repurposed for use in a wireless network: the average queue length is calculated using a forecast formula instead of the forecast average formula. In particular, two small free constants, α and γ, are used to estimate the level and trend of the forecast. This modified version of RED performs better than the original RED in a wireless network [11].

The REM [70] algorithm requires two substantial formula changes in order to operate correctly in a wireless environment. The first change is to set the channel link capacity, $c_l(t)$, to be the Theoretical Maximum Throughput (TMT) [27] of the link. In order to estimate the TMT it is necessary to consider how the wireless channel is shared between the Access Point and mobile stations. The second change is to modify the calculation of the input rate, $x_l(t)$. An

Exponentially Weighted Moving Average (EWMA) [9] is used to filter the bursty traffic and transform $x_l(t)$ into a continuous variable.

The BLUE [137] algorithm has two parameters, δ_1 and δ_2, used to increase and decrease the dropping probability. When the BLUE algorithm is applied at an access point, it becomes necessary to dynamically change the δ_1 and δ_2 parameters. [11] proposes a function to adjust these parameters according to variations in the traffic and in accordance with the design of the original BLUE algorithm [137] .

In contrast to RED, REM and BLUE, some algorithms have been designed to operate in wireless networks in order to solve specific wireless networking problems. For example [169] considers the problem of fairness between upload and download traffic. The algorithm operates in an infrastructure wireless network when the Distributed Coordination Function is used. It solves the fairness problem by implementing a PI controller to dynamically change the contention windows.

VQ-RED [170] looks at the fairness between flows. It uses multiple virtual queues, one for each flow. Each virtual queue is managed by a RED algorithm. The arriving packets at the AP are distributed between the queues by a classifier. Fairness is achieved by seeking to analyse and manage the queue lengths. The algorithm not only improves the fairness but also reduces packet delay.

Issues related to fairness for TCP protocols can be addressed using two virtual queues [171]: one virtual queue is maintained for TCP data packets and the other for TCP acknowledgements. Each queue is managed by a PI algorithm. The algorithm also acts on the contention window by implementing an AIMD algorithm.

Other AQM algorithms specifically consider congestion or channel status. In the case of Channel-Aware Active Queue Management (CA-AQM) [172], a per flow price is calculated using the queue length and the bit rate. This results in a stable queue and a fair sharing of the channel access time.

The AQM algorithm presented in [173] is based on multi class RED. Each traffic class has a different loss priority and so the buffer is divided using multiple thresholds. The algorithm manages priorities using a separate RED algorithm for each class.

An alternate approach to the calculation of the dropping or marking probability in RED-like algorithms is presented in [174]. Instead of the queue length, the time taken for a packet to access the channel together with the number of RTS packets sent by the station are used as indicators of channel congestion in an ad hoc network.

The proxy-RED Algorithm [175] is designed to address the issue of differing capacities between a wired and wireless network. The RED, or ARED, algorithm is modified to fit the network characteristics, and applied to the wired network before the AP, via a proxy function. The difference in capacity between the wired network and the wireless channel is smoothed by proxy-RED, improving the efficiency of the wireless network and reducing congestion at the AP.

To complete this overview of AQM in a wireless environment, three other AQM schemes must be mentioned. The first of these is Ad hoc Hazard RED [176] (AHRED). This algorithm is designed to sit on top of the RED algorithm. It calculates a dropping, or marking, probability using a hazard function. This function makes AHRED more aggressive as the volume of traffic grows and the queue length nears capacity and less aggressive when traffic levels are low and the queue is nearly empty.

The second is EF-AQM [177]. It uses a controller at the intersection between heterogeneous networks to avoid congestion. It is a simple application of control theory on wireless networks.

The final AQM scheme to be considered also uses control theory [120]. It is an application of the H_∞ controller [120] in a wireless network environment. It uses modified versions of equations 3.10 and 3.11 to capture variations in the Congestion Window. [120] considers the fading effects on wireless networks.

The discussion above illustrates the limitations of AQM that motivate this work. First of all it is important to note that AQM algorithms are not specifically designed to improve QoE, rather they are designed for congestion avoidance. However, as has been shown above, they can contribute to aspects of QoS such as fairness or the refinement of IQoS parameters.

Secondly, if AQM is to be used on a wireless network then design modifications are needed; not only because of the nature of the wireless environment but also due to the use of channel access methods like EDCA. AQM schemes need to be able to manage different kinds of traffic with particular characteristics in each AC to fit CoS requirements.

Finally, wireless AQM schemes are not sufficient for the provision of services with the best QoE possible. This is because they only act on the queue. A more holistic approach that includes wireless AQM schemes as part of a larger traffic management system is needed.

3.6 Summary

The provision of quality of service guarantees on a wireless network is a challenging problem. Existing techniques and algorithms used for QoS provision at the MAC level are not well-equipped to cope with a wide variety of traffic mixes and protocols used on wireless networks.

This chapter presented a detailed literature review that covered three main areas: Quality of Service, the mathematical modeling of congestion on wireless networks and Active Queue Management schemes.

The discussion in this chapter identified gaps in the existing body of knowledge. The first of these is the lack of a technique to instantaneously estimate Quality of Experience (QoE), a metric that is of particular interest to Internet Service Providers. The second is the need for a wireless network traffic management system capable of providing near-optimal QoE.

In the following chapter a new quality metric for wireless networks is presented.

Chapter 4

eQoS

Efforts to capture Quality of Service (QoS) have given rise to a large collection of metrics, measures, methods and procedures for the evaluation of network performance and service provisioning. As a consequence, the specification of QoS [71] has evolved and been extended since it was originally defined by the ITU in 1994.

QoS metrics can evaluate service provisioning at a low level, where network parameters are measured; or at a high level, where end user satisfaction is measured using Quality of Experience (QoE) [8]. QoS metrics are useful as they not only allow for management of the network and control of services, but they also help in the improvement of service provisioning and in the development of new network protocols and architectures.

TELCOSs have a vested interest in QoS metrics. On the one hand, they need to measure network performance and plan future network development; on the other hand they need to have good estimates of QoE to enable them to develop new services for their users and provide Service Level Agreement (SLA) guarantees.

The research community has been looking for robust ways to automate QoE measurement or estimation. The International Telecommunication Union (ITU) has designed computer algorithms to automatically evaluate voice, audio and video services; providing estimates of the human perception of these services. QoE can be automatically calculated via a comparison between services at the source and at the destination. This methodology does not depend on end user feedback. QoE assumes an important role in QoS management when applied to critical nodes for quality management purposes. When considering quality issues, an Access Point (AP) is the most critical node in an infrastructure wireless network.

Two key characteristics are essential for capturing QoE: quick estimation and simple calculation. To achieve these a new metric to estimate the quality of a service provided is required. This need is addressed in this chapter where a new metric, eQoS, is defined [14].

The chapter is structured as follows: in the first two sections the definition of eQoS and the motivations behind it are presented. The three most important realtime services are then considered as the application of eQoS to Voice over IP, audio and video streams is detailed. Finally, some practical applications of eQoS are provided.

4.1 The Definition of eQoS

The perceived quality [8] of a network is measured through feedback obtained from the end user or estimated using computer algorithms. These computer algorithms compare the service at the source with that at the destination and estimate the human perception of the network quality. Their complexity means they are not suitable for use in all networks for the estimation of QoE at the network node. A new metric that is compatible with the definition of QoE is [14]:

Definition 1. *eQoS is a metric that captures the perceived QoS through an almost instantaneous measurement of loss in network quality.*

eQoS is expressed as a proportion between 0 and 1 or as a percentage and is a dimensionless quantity. It is a near instantaneous expression of the customer satisfaction. eQoS provides a mechanism for calculating the perceived quality at a node for critical services. Critical services are usually provided in real time and are highly sensitive to packet loss and delay. eQoS has been designed to include some important characteristics: it is a near instantaneous measure, calculated by sampling some physical network parameters over a suitable, short interval of time; it is recalculated every sampling time interval and varies depending on the service being provided.

Traditional QoE metrics require audio or video to be measured over an extended period of time by the end user; for example, traditional algorithms typically require more than 10 seconds of audio or video traffic in order to provide an objective evaluation [93].

The eQoS sampling time is not a fixed value; it is set according to the time needed to encode the service. To simplify the calculation of eQoS and to allow for its application across a wide variety of services, a one second sampling time interval is recommended. Use of different eQoS sampling times does not affect the validity of the metric; what is essential is that the number of packets transmitted per second by each service has to be sufficient to obtain a reasonable eQoS estimate. In other words the granularity of the sampling interval must be such that the samples are statistically acceptable and valid. For services provided with a particular protocol, like TCP, the eQoS sampling intervals may vary. This is because the number of samples per second will depend on the Round Trip Time (RTT) [121].

eQoS depends on a small number of QoS network parameters and so its calculation is very simple. It measures the perceived QoS and, at the same time, provides an estimate of QoE. Existing computer algorithms that seek to replicate the complex human perception of quality use numerous variables and their complexity means that considerable CPU effort is needed for their calculation.

eQoS does not follow the traditional algorithm methodology of comparing the service at the destination with the original service provided at the source: It is inferred statistically using a small number of QoS network parameters that are estimated at the node.

eQoS is designed to be calculated per flow and to provide a near instantaneous evaluation of service quality at the node. It can be applied at an AP or where single or multi queue systems are present and it can be used to inform the operation of queue management algorithms.

4.1.1 Description and Motivation

The definition of eQoS was motivated by the need for a simple mechanism to calculate the perceived QoS at an AP. Traffic management is a critical issue for nodes like APs, where the traffic is switched from a wired link to a wireless link and vice versa. The novel idea is to find an almost instantaneous estimate of a perceived QoS measure at the AP and immediately use it as a parameter for traffic management.

The first step towards a complete eQoS definition is the identification of the key physical network parameters that affect the perceived QoS. Throughput measurements do not give direct information on the quality lost at the AP. This is because throughput is measured over time rather than instantaneously. This makes it unsuitable for use in the calculation of eQoS. On the other hand, the number of packets dropped or lost at the AP is easy to calculate and plays a significant role in the QoS perceived by the end user. Each packet dropped is a small service interruption that may be perceived by the human senses.

Packet delivery delay is another factor affecting the perceived quality. If the packet latency at the AP exceeds the maximum delay allowed, then the packet cannot be used to reproduce the service at the destination and has to be dropped. This is the case, for example, for an audio packet that is delayed for too long in the AP and cannot be reproduced at the destination. Similar considerations apply when the packet delay variation exceeds the maximum delay permitted. In both these cases the packet should be dropped at the AP to prevent unnecessary transmissions on the wireless link and the consequential waste of network time and bandwidth.

For VoIP services, the packet delay variation, or jitter, should not exceed 30ms [45], while delay should be kept below 150ms [45] to provide the service with the best quality [45]. Thresholds for jitter and delay are not strictly fixed, they can vary depending on the kind of service provided and the network configuration. Jitter is defined as the transmission delay between two consecutive packets from the same flow [178] but in the practice it is often considered as the packet delay variation [45].

eQoS is calculated by considering packet loss at the AP; this includes packets that are dropped at the node and packets that are delayed beyond the maximum limit permitted. The number of lost packets per sampling time interval contributes to the loss in quality experienced by the end user, and is expressed as a percentage of the total number of packets received in the sampling time interval for the given service.

The number of packets lost is not sufficient to quantise the perceived loss of quality during

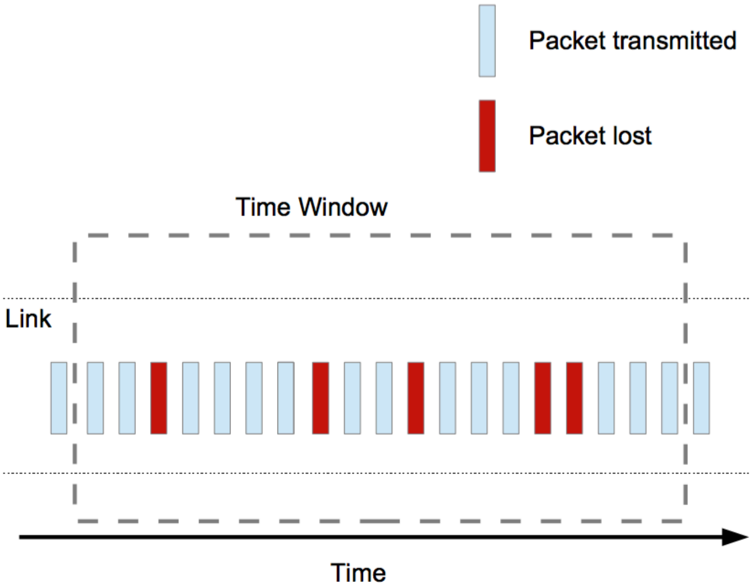

Fig. 4.1: Illustration of packets transmitted and lost.

the sampling time interval. If two packets are lost and 98 packets are transmitted, this might not be perceptually relevant for the final user. Rather it is the adjacencies between dropped pockets that matter: the closer the loss or dropping events are in time, the more likely it is that the event will be perceived by the end user. For example, when receiving an audio stream, if two packets are lost, this may not be relevant for the audio quality; however if the two packets dropped are consecutive the loss in quality may be significant, e.g. this could be the loss of an important word in a conversation. Figure 4.1 illustrates a stream of packets flowing during a given time window. It indicates where some of the packets are lost and it can be seen that two of the packet loss events occur in sequence. When packet loss events occur in sequence or are temporally close to each other then this may lead to a perceived loss of quality by the end user. Therefore, there are two factors that contribute to our definition of eQoS: the ratio of packets lost to packets transmitted and the probability of adjacencies between packet loss events. Both factors can be determined numerically, the first one as the percentage of packets lost and the second through a probabilistic estimate that packet losses occur in sequence. This numerical approach results in a metric that is simpler and quicker to calculate than a complex method involving subjective human perception and recognition methodologies. It is also a more efficient way of estimating QoE when a precise QoE value is not required.

eQoS values vary between 0 and 1 and indicate the level of customer satisfaction. In order to make the metric easier to interpret, eQoS can be expressed as a percentage. eQoS can be used to determine QoE by the use of threshold eQoS values. These thresholds mark the decision point at which a user will decide to discontinue use of a service. The relationship between eQoS and MOS is slightly more challenging to determine as eQoS provides a near instantaneous estimate of quality, while a MOS value is an average computed over time.

As explained above eQoS is the sum of two contributions: one due to packet loss and the other due to the loss of adjacent packets. It is expressed as:

$$eQoS = \alpha F(P_D, P_d) + \beta G(P_D, P_d) \tag{4.1}$$

Where:

$F(P_D, P_d)$ is the loss contribution,

$G(P_D, P_d)$ is the adjacency contribution,

P_D is the number of dropped packets and

P_d is the number of packets dropped because they have experienced excessive delay.

α and β are two parameters with $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ and $\alpha + \beta = 1$. These are discussed in more detail in section 4.1.2.

The first contribution is the loss contribution $F(P_D, P_d)$. It was defined above as the ratio between packets lost at the AP in one second and the number of packets generated at the source. It is expressed by the formula:

$$F(P_D, P_d) = \frac{P_D + P_d}{P_T} \tag{4.2}$$

where P_T is the number of packets per second generated at the source.

The second contribution to eQoS is $G(P_D, P_d)$. It is the contribution due to the loss of adjacent packets. It can be viewed as a measure of the occurrence of an inter-packet loss distance of zero [179]. It can be calculated in two ways. The first method is to continuously check for adjacencies of packet loss events. This method is complex and not practical for use when a near instantaneous estimate of eQoS is needed. It requires a data structure to store the status of each packet, as dropped or transmitted, and two algorithms: one to update the data structure for each packet arriving at the AP and one algorithm to parse the data structure and count the adjacencies between the lost or dropped packets. The second method is to create a statistical model to calculate the probability that packet losses are sequential. This second method is simpler and only requires measurements of the number of packets lost and transmitted; it then uses combinatorics to obtain the required probability.

The use of combinatorics relies on the assumption that the packet loss events are independent. Wireless network conditions may mean that when a packet is lost some adjacent packets in the sequence are lost as well. This is likely to happen for a large number of packets in a row because of network congestion. If a few packets are lost then these can be considered as independent events while if more packets are lost these should not be considered as independent events. For the purposes of this work packet losses are considered independent because quality degrades rapidly when just a few packets are lost.

In this section we show how combinatorics can be used to obtain an estimate of the adjacency loss contribution to eQoS.

Considering a set of n numbers, the number of ways that k numbers can be taken from this set without replacement is [180]:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{4.3}$$

This is known as a combination. This formula can be used to find the number of way that k packets can be dropped when n packets are transmitted.

The number of ways that k numbers can be drawn from a set of n numbers without any of the k numbers being in sequence is:

$$\binom{n-k+1}{k} = \frac{(n-k+1)!}{k!(n-2k+1)!} \tag{4.4}$$

This formula, inferred by basic combinatorics [180], can be used to find the number of ways that k non-sequential packets can be dropped when n packets are transmitted. Using subtraction the number of ways that k numbers can be drawn from a set of n numbers, with at least 2 of the k numbers being in sequence, is given by:

$$\begin{pmatrix} n \\ k \end{pmatrix} - \begin{pmatrix} n-k+1 \\ k \end{pmatrix} = \frac{n!}{k!(n-k)!} - \frac{(n-k+1)!}{k!(n-2k+1)!} \tag{4.5}$$

When $k \geq 2$ this formula can be used to estimate when at least two sequential packets are dropped.

The calculations above can be used to determine the probability that at least two packet drops are in sequence when k packets are dropped from a set of n transmitted packets. This can be expressed as:

$$\mathbb{P}(n,k) = \frac{\begin{pmatrix} n \\ k \end{pmatrix} - \begin{pmatrix} n-k+1 \\ k \end{pmatrix}}{\begin{pmatrix} n \\ k \end{pmatrix}} \tag{4.6}$$

The adjacency contribution, $G(P_D, P_d)$, in equation 4.1 for eQoS can now be found using equation 4.6 as:

$$G(P_D, P_d) = \mathbb{P}(P_T, (P_D + P_d)) \tag{4.7}$$

Where:

P_D is the number of dropped packets,

P_d is the number of packets dropped because they have experienced excessive delay and

P_T is the number of packets per second generated at the source.

It should be noted that $\mathbb{P}(n,k)$ is not related to a specific number of packets dropped in sequence; it is the probability that at least two packets are dropped in sequence. Packet dropping events are perceived by the end user regardless of the number of packet drops that occur. It easy to see that if more than 25 packets are dropped from a set of 50 packets arriving at the queue, then the probability that some packets are dropped in sequence is 1. The higher the probability of packets being sequentially dropped, the higher the probability that the number of packets dropped in sequence is large. Long sequences of packet drops are more likely to be perceived by the end user.

Before proceeding to the calculation of eQoS for a VoIP service the parameters α and β are considered in more detail.

4.1.2 Setting the Parameters α and β

Parameters α and β play a very important role in the definition of eQoS. eQoS is given by:

$$eQoS = \alpha F(P_D, P_d) + \beta G(P_D, P_d). \tag{4.8}$$

eQoS values vary between 0 and 1. The two weights α and β are such that:

$$\alpha \geq 0; \beta \geq 0. \tag{4.9}$$

They also satisfy the following equation:

$$\alpha + \beta = 1. \tag{4.10}$$

If the eQoS is close to 0, the service is expected to be provided with almost no loss of quality. If the eQoS is close to 1, then the service provided is poor or is not provided at all. For practical reasons, eQoS can also be expressed as a percentage.

Greater weight should be given to the contribution made to eQoS by the loss of adjacent packets relative to that made by loss. This is captured by an additional constraint on α and β:

$$\beta > \alpha. \tag{4.11}$$

This imposes a logarithmic aspect to the eQoS curve as a function of the dropped packets. A similar curve was observed in [78] where a relationship between dropped packets and QoE was proposed. The logarithmic nature of this kind of event was also discussed in [80].

The challenge is to find the most appropriate values for α and β in order to capture the eQoS for the service being considered. eQoS is a perceived QoS, therefore α and β have to be determined in order to make eQoS conformant with the nature of the QoS being measured; that is, the achieved QoS, the perceived QoS, the offered QoS and the required QoS [8].

Two approaches are needed in order to determine α and β. The first is to evaluate the Mean Opinion Sore (MOS) [71] for a given service when different numbers of packets are dropped. The results are then used to inform the choice of the most appropriate eQoS curve to use. The second approach is to determine the desired eQoS curve for the service and estimate the parameters α and β from it. As quality decreases rapidly when a few packets are lost, precise estimates of α and β are not required. The approach adopted will depend on the service provider and will be dictated by business needs and by drawing on their experience of providing the service.

MOS and eQoS are two completely different measures. MOS is a metric that takes time to evaluate as it is an average value. It depends upon feedback from the final user or a computer algorithm. By contrast, eQoS gives a near instantaneous estimate for each sampling time interval. It indicates that the end user is perceiving a loss of quality and may cancel the service if no action is taken to improve its quality. This is in line with the definition of QoE and so eQoS can be viewed as a realtime estimate of QoE. It should be noted that eQoS is valid only for the sampling time interval for which it has been calculated. If in a given sampling time interval 50% of packets are dropped and in the following sampling time no packets are dropped, then it is not the case that the eQoS over the two intervals is the average value of 25%.

In the next section a practical application of QoS for VoIP is described.

4.2 eQoS for VoIP Services

In this section the practical calculation of eQoS for Voice Over IP (VoIP) streaming services is considered. The calculation is performed using the network architecture shown in figure 2.2. It replicates a phone call between a wired source and a destination that is a mobile device. As a phone call is bidirectional a flow in the opposite direction needs to be considered also.

The VoIP packets generated at the source are transmitted to the Access Point (AP) via a wired network, the last hop between the AP and the destination is a wireless link. The goal of eQoS is to evaluate the quality at the AP, taking into account packets dropped due to excessive delay and packets that are lost. It is assumed that packets arrive at the AP queue from the source at regular intervals, the delay is close to the theoretical minimum value and that jitter is close to zero. These are reasonable assumptions for a wired link. If these assumptions are not true, problems are already present in the backhaul or in the backbone and the quality of the service provided is already compromised. In this case the problem will be detected at the AP, but cannot be fixed by it. While beyond the scope of this work, it is theoretically possible for the system to be designed to provide meaningful feedback to the backhaul or the backbone in this situation.

The AP is a critical point in the network. It is anticipated that it will be a bottleneck because even if the rated throughput of the wired and wireless networks is comparable, the Theoretical Maximum Throughput (TMT) available on a wireless network is lower than on the wired link. Therefore, the AP is a bottleneck even when Very High Throughput (VHT) protocols are used in the wireless channel. The wireless network channel is unique and operates bidirectionally between the AP and the mobile devices. If the number of mobile devices is too high then packets may be dropped due to delays at the AP queue or when the AP queue is full.

In this example G.729 [57] encoding is used, it is a PCM [55] that uses 8KHz sampling for voice calls. The sampling time is $1s$. The G.729 VoIP encoder transmits 50 packets per second and each packet transports two $10ms$ code blocks of speech. If a packet is dropped, both the blocks are lost and $20ms$ of speech cannot be heard at the destination. If two packets are dropped in sequence the amount of speech lost will be $40ms$ long and the probability that the loss will be perceived by the human ear increases.

As described in the previous section, combinatorics can be used to estimate the probability that k packets in a series of n packets are dropped in sequence. In order to provide some insight into the equations given above, a heuristic derivation is considered below. In this example $n = 50$ as 50 packets are transmitted per second. Let $\{a, b\}$ indicate that packets a and b are dropped, then the possible ways in which 2 packets from the 50 can be dropped are:

$$\{1,2\},\{1,3\},\{1,4\},...,\{48,49\},\{48,50\},\{49,50\} \tag{4.12}$$

From equation 4.3 there are 1225 ways this can occur.

The possible ways in which 2 packets from the 50 can be dropped in sequence are are:

$$\{1,2\}, \{2,3\}, \{3,4\}, ..., \{47,48\}, \{48,49\}, \{49,50\} \tag{4.13}$$

It can be seen that there are 49 such events.

The possible ways in which 2 packets can be dropped without being in sequence is 1176 i.e 1225 - 49. Therefore, there are 1225 ways in which two packets can be dropped from the 50 packets transmitted. These heuristic calculations agree with the equations given above. From equation 4.4 it can be seen that the number of combinations where two packets from the fifty transmitted are not dropped in sequence is 1176. From equation 4.6 the probability of dropping two packets is sequence from the 50 transmitted is found to be $\mathbb{P}(50, 2) = 0.04$.

Figure 4.2 shows the adjacency contribution function $G(P_D + P_d)$. The horizontal axis represents the number of dropped packets and the vertical axis represents the probability that packets are dropped in sequence. The number of packets transmitted per second is 50. It can be seen that the probability grows rapidly and after 15 packets are dropped the probability that packets are dropped in sequence is very close to 1. This shows how the service quality is significantly affected when just a few packets are lost.

The probability values in figure 4.2 are discrete, but they can be fitted with a sigmoidal curve [181]. This curve is also shown in figure 4.2. The equation for this curve is:

$$f(x) = b + \frac{a - b}{1 + e^{(-\frac{x-c}{d})}} \tag{4.14}$$

The probabilities shown were calculated using Matlab [182], the parameters a, b, c and d that fit the curve were obtained through a sequence of automatic approximations using gnuplot [183]. They are: $a \cong 1$, $b \cong -0.1$, $c \cong 5.8$, and $d \cong 2$. a, b, are the bottom and the top of the curve respectively, c is the point halfway between the bottom and the top and d is the slope [184] [185]. The curve is a graphical continuous approximation of the discrete probabilities, it is valid for $x \geq 2$ and $x \leq 13$, that is when between 2 and 13 packets are dropped. For $x > 13$ the probability is close to 1 and the VoIP stream cannot be decoded and reproduced at the destination.

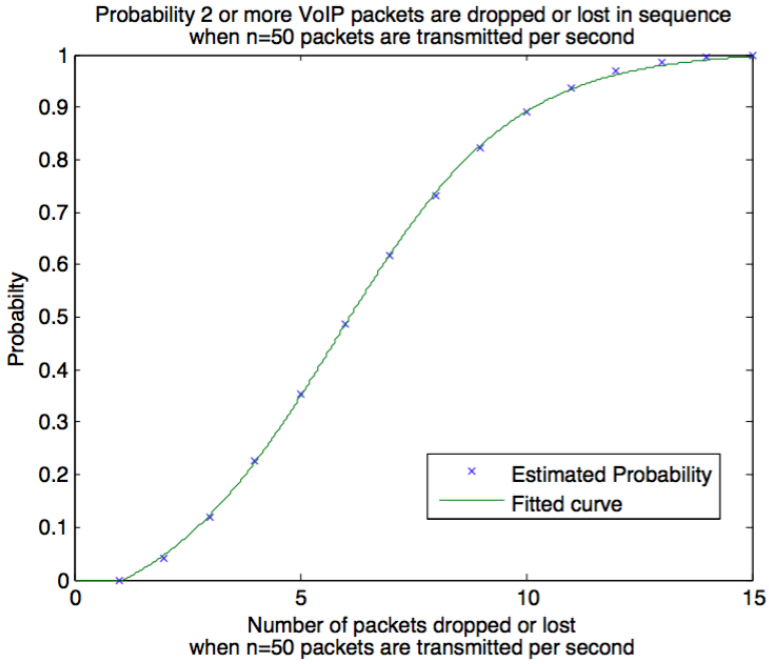

Fig. 4.2: Estimated probabilities and the fitted sigmodal curve for VoIP traffic.

The above discussion leads to the following equation for the calculation of the expected Quality of Service (eQoS) for VoIP:

$$eQoS = \alpha \frac{P_D + P_d}{P_T} + \beta \left(-0.1 + \frac{1.1}{1 + e^{(-\frac{P_D + P_d - 5.8}{2})}} \right) \tag{4.15}$$

The family of eQoS curves obtained by varying α and β in equation 4.15 are shown in figure 4.3. This was obtained using Matlab [182]. If α is equal to 1 and β is equal to 0, then the eQoS will be the line consisting of the $F(P_D, P_d)$ contribution. Similarly, if β is equal to 1 and α is equal to 0, eQoS will be the sigmodal curve obtained from the $G(P_D, P_d)$ contribution alone.

The α and β parameters capture the contribution made to eQoS by the loss of packets and due to loss of adjacent packets. In the example considered in Chapter 6 below, α is set to 0.25 and β to 0.75 for VoIP encoded with G.729.

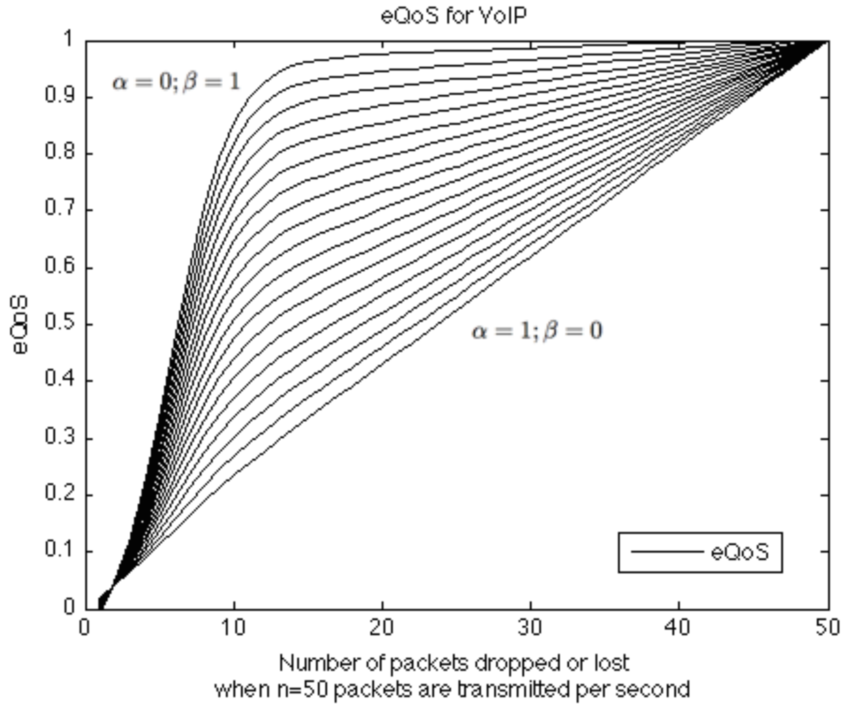

Fig. 4.3: eQoS curves for VoIP traffic with varying values of α and β.

4.2.1 eQoS for Audio Streaming

In the above the focus was on estimating eQoS for VoIP services, in this section the focus shifts to the calculation of this quality metric for an audio streaming service.

As mentioned in section 2.4.3, audio streaming can be considered as a constant flow of information and the eQoS formulas derived above for VoIP traffic can be reused with only a few modifications.

As described in section 2.4.1 a streaming service can be either live or deferred. The streaming is defined as live when it is transmitted from a live event or with a minimum buffer at the destination so that the only delay permitted is due to encoding and transmission. An example of live streaming is a live sports event, when audio and video are transmitted with only an encoding delay and minimum buffering. Live streaming also occurs when the stream is stored at the source and is provided upon request using the maximum throughput available. For example, the streaming of Video On Demand (VOD) from a web site is considered live streaming.

A deferred stream is partially stored at the destination and the transmission time and the play time are not related. In this case packet transmission is deferred, thus packet retransmissions at the application layer and throughput reductions do not impact on the quality of the stream.

eQoS has to be calculated for live, critical services, where the traffic is more sensitive to delays and retransmissions are not possible. Non real time services are not critical and can be considered and managed as a data transfer.

Audio streaming was described in section 2.4.3. For the calculation below it is encoded and streamed using a Variable Bit Rate (VBR) service. This contrasts with VoIP which is a Constant Bit Rate (CBR) service, with constant frame size and constant packet size. For design purposes, audio streaming is considered as a constant flow of information with different packet and frame sizes.

As for the VoIP service, the eQoS calculation for an audio streaming service is explained using an example. Using AAC encoding the stream is divided into frames, where each frame contains 1024 samples and is encapsulated in a packet [62]. There is a direct relationship between the packet and its audio information, that is the 1024 samples the packet contains. The number of frames, and hence the number of packets transmitted per second, is constant at about 43 per second, but for AAC coding the packet size and sample duration varies. Each frame reproduces about $23ms$ of audio. Figure 2.3 provides an example of audio streaming, showing the frames and packets transmitted.

The packet sizes vary but, as mentioned above, the information per packet is constant. The frames containing no information, i.e. no sound or silence, are an exception. These frames have no information and if they are lost the quality is not affected. Unfortunately the contents of a packet can only be analyzed at the application level, therefore these kinds of packets cannot be detected and managed at a node. In this discussion they are considered like any other packet in the system.

As the differences between audio streaming and VoIP are almost negligible, eQoS for audio streaming is calculated using the same formula as for VoIP:

$$eQoS = \alpha F(f_D, f_d) + \beta G(f_D, f_d). \tag{4.16}$$

Where:

$F(f_D, f_d)$ is the loss contribution,

$G(f_D, f_d)$ is the adjacency contribution,

f_D is the number of frames lost through dropping and

f_d is the number of frames lost due to excessive delays.

Both the loss and adjacency contribution are the same as for VoIP. However, the formula is slightly different to the one used for VoIP as the adjacency probabilities are fitted by a sigmoidal [186] curve with different coefficients due to the variation in the total number of packets transmitted per second. Using these coefficients, the eQoS for an audio stream is given by:

$$eQoS = \alpha \frac{f_D + f_d}{f_T} + \beta \left(-0.1 + \frac{0.9}{1 + \exp(-\frac{f_D + f_d - 5.4}{1.8})} \right). \tag{4.17}$$

The formula is calculated using frames instead of packets as the information unit. Each frame is contained in a packet, therefore the changes are only in notation rather than in substance. In the next section the α and β parameters for an audio streaming service are considered.

4.2.2 Setting α and β for Audio Streaming

In this section the guidelines for setting the parameters α and β for audio streaming are explored. Figure 4.5, obtained using Matlab [182], shows the family of curves that arise when the parameters α and β are varied. As described for the VoIP service, the line is the $F(f_D, f_d)$ contribution when β is equal to zero; the purely sigmoidal function is the $G(f_D, f_d)$ contribution when α is equal to zero.

Audio streaming differs from VoIP in both the mode of transmission and the media contents as an audio stream can transmit voice, sound and music. eQoS components are more important than for VoIP, where the sampling process is at a low frequency, therefore more careful consideration is needed when determining α and β.

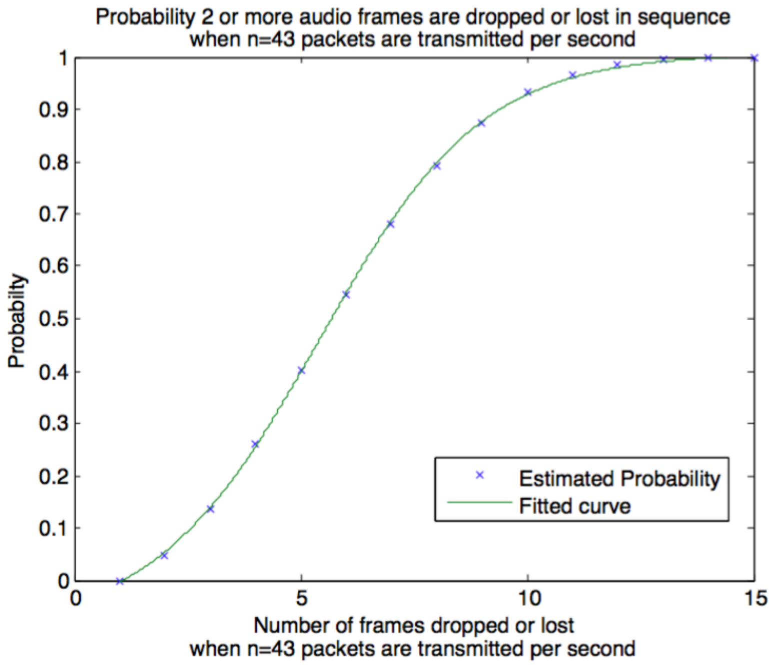

Fig. 4.4: Estimated probabilites and the fitted sigmodal curve for audio traffic.

The best solution is to use the two methods described for VoIP; that is, the MOS evaluation and the desired eQoS curve. Due to the variable nature of the audio stream, the MOS evaluation has a high variance and is affected by the client software and hardware. For example, there are software algorithms designed to correct errors in the media stream induced by packet loss [187]. Hardware and the content of an audio stream can also give rise to intrinsic quality variations.

It is important to note that, like QoE, eQoS is an estimate of the perceived QoS and it is used for decision making at the node. Therefore the settings for α and β need to reflect the desired perceived QoS characteristics.

For an audio stream that is a mix of music, voice and sound, the QoS rapidly shows a loss in quality when only a few packets are lost. Therefore the adjacency contribution plays a more important role than that of loss. The equations defined for VoIP remain valid:

$$\beta > \alpha \tag{4.18}$$

and

$$\alpha + \beta = 1 \tag{4.19}$$

By definition, equation 4.10 restricts eQoS to values between 0 and 1.

The parameters α and β were set to 0.10 and 0.90 respectively for the simulations given in Chapter 6. These were chosen based on an eQoS evaluation of audio streaming and the previous discussion on the choice of α and β for VoIP. As detailed in section 3.1.1, PEAQ calculates the loss in quality perceived by the end user. The perception of quality loss for a high fidelity audio stream occurs when just a few packets are lost, therefore the values used for α and β will differ from those used for VoIP. Figure 4.6 shows the QoE curve when the number of dropped packets per second grows. Dropped packets are replaced by the last packet received, making it easy for the PEAQ algorithm to detect audio losses. QoE decreases rapidly when a few packets are dropped from the stream. Using the the PQevalAudio software[1] [188], the ODG [93] score varies between 0 and -4. This differs from MOS where a scale of values between 1 and 5 is used. In the case of VoIP, the MOS does not decline as rapidly, because VoIP packets contain less information: 8KHz sampling for VoIP results in 50 packet transmissions per second while audio streaming has a sampling rate of 44KHz resulting in 43 packet transmissions per second.

4.2.3 eQoS for Video Streaming

In the previous section the focus was on audio services; attention now shifts to the calculation of eQoS for video streaming services.

Video streaming exhibits a number of features that differ significantly from those of VoIP and audio streaming services. Video services consist of a variable flow of information in terms of both packets and frame sizes. Therefore the eQoS design needs to be extended to allow for this variability in the information flow.

Video streaming, like audio streaming, can be provided as either a live or a deferred stream. The deferred stream is partially stored at the destination and the transmission and play back times are not related.

[1] `http://www-mmsp.ece.mcgill.ca/Documents/Software/`

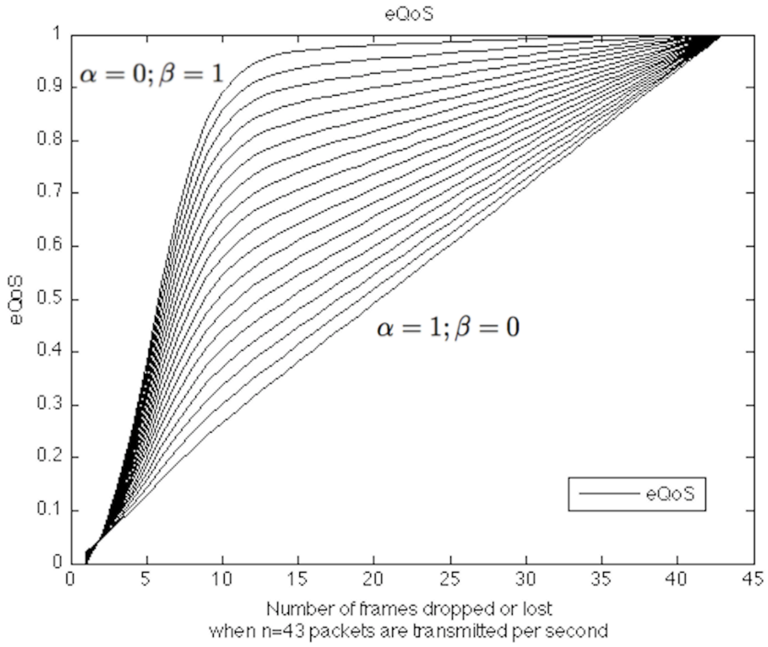

Fig. 4.5: eQoS curves for audio streaming with varying values of α and β

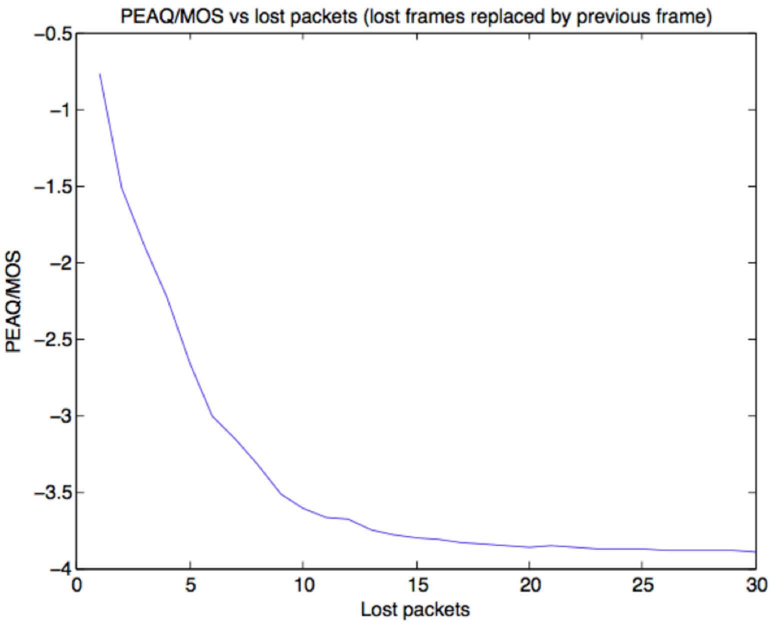

Fig. 4.6: MOS decreases as audio streaming packets are lost.

In the following calculations of eQoS the MPEG-4 Advanced Video Coding (AVC) [66] standard is used for video and audio streaming. However, the formulas obtained and conclusions reached are applicable for other video streaming standards.

The application of eQoS to a video streaming service does not immediately follow from that for VoIP and audio streaming. Some additional considerations and substantial changes to the eQoS formula are required.

Video streaming is considered as a VBR service using the MPEG-4 part 10, known as H.264 or MPEG-4 Advanced Video Coding (AVC) [66] standard. Packets are of variable size and frames are of different sizes and types. Figure 2.5 shows the relationship between I and P frames in terms of size. Other standards can provide a video streaming service with different characteristics, for example using a Constant Bit Rate (CBR) stream or fixed packet sizes.

In the example below basic profile H.264 [66] video with 30 frames per second is used. It uses only I and P frames, with a ratio of approximately one I frame to every 23 P frames. The ratio is calculated by the encoding software. This shows the applicability of the eQoS estimation method to irregular and non-symmetric data structures.

When B frames are included in the H.264 profile, the GoP(N, M) [67] configuration suggests that the number of B frames between two I frames is usually larger than the number of P frames. The loss of P and B frames have a similar consequence from the point of view of the user's perception of the service. Thus they can be considered together using a single adjacency contribution.

A long time interval is needed to evaluate the perceived quality of service for such a video service using a traditional algorithm that evaluates human perception. A method for determining eQoS based on mathematical deduction is the most appropriate one for obtaining an almost instantaneous estimate of the perceived QoS.

As before, eQoS is calculated by considering the contribution of lost packets and their adjacency. Unlike audio streaming, the loss contribution cannot be calculated based on the number of packets lost. Video streaming packets are of variable size and do not contain the same amount of information. This contrasts with VoIP and audio streaming where the amount of information in each packet is fixed. For video streaming the byte is the most appropriate unit of information to use. The loss contribution for a video stream is:

$$F(B_D, B_d) = \frac{B_D + B_d}{B_T} \tag{4.20}$$

Where:

B_D is the number of bytes lost due to packet drops,

B_d is the number of bytes lost due to delay and

B_T is the total number of bytes transmitted in the sampling interval.

The reduction in quality also depends on the number of packets lost per second and from the adjacency of these losses. It is also important to note that I frames contain more important information than P frames. I frames are, in general, much longer than P frames and so typically need more bytes than a P frame; however, they are less frequent than P frames.

Considering the loss contribution in bytes, a large amount of lost bytes can mean that a large amount of information is lost; this may correspond to the loss of an I frame or of a number of P frames or both. For this reason instead of calculating two loss contributions, one for I frames and one for P frames, the loss contribution is calculated in bytes. Another solution that gives more weight to the loss of I frames relative to P fames would be to use packet lengths.

It is necessary to add together two contributions to the adjacency component of eQoS for a video stream: one for the adjacency between packet loss events for an I frame and another, similar, contribution for adjacent loss events for a P frame. $G_I(P_{ID}, P_{Id})$ is the contribution for the loss of adjacent I frame packets. $G_P(P_{PD}, P_{Pd})$ is the contribution for the loss of adjacent P frame packets.

The eQoS formula for a video streaming service includes both these contributions:

$$eQoS = \alpha F(B_D, B_d) + \beta G_I(P_{ID}, P_{Id}) + \gamma G_P(P_{PD}, P_{Pd}) \tag{4.21}$$

where α, β and γ are the weights associated respectively with $F(B_D, B_d)$, $G_I(P_{ID}, P_{Id})$ and $G_P(P_{PD}, P_{Pd})$. These are considered in more detail in the following subsection.

$G_I(P_{ID}, P_{Id})$ and $G_P(P_{PD}, P_{Pd})$ are adjacency contributions for I frame and P frame loss. Both contributions are calculated using equation 4.6. If B frames are used then the

$G_P(P_{PD}, P_{Pd})$ contribution is replaced by a $G_{P,B}(P_{PBD}, P_{PBd})$ contribution. Comparing equation 4.21 with equation 4.1 as previously used for VoIP and audio streaming, it can be seen that there are two changes. The first change is the use of an adjacency contribution for each type of frame used; here this translates into a separate contribution for I frames and P frames. The second change is the use of bytes instead of packets as the unit of information in the calculation of the loss contribution.

The adjacency contribution for I frame packets is shown in figure 4.7. The curve that fits this discrete probability distribution is sigmoidal. In this simple example there are 30 frames per second; of these 1.3 are I frames and 28.7 are P frames. Each I fame is comprised of 12 packets of maximum size 1024 bytes; therefore I frames are transmitted with an average rate of 15 packets per second.

Figure 4.8 shows the sigmoidal curve that fits the discrete probability distribution for the P frames. In each second the number of P frames transmitted is approximately 29, with an average size of 2 packets per frame. This results in the transmission of 58 packets per second with a maximum packet size of 1024 bytes.

4.2.4 Setting the parameters α, β and γ for Video Streaming

To fully define eQoS for a video stream the three parameters α, β and γ need to be set. As for audio streaming there are two key aspects to the evaluation of perceived quality. First of all it is necessary to consider the video content, i.e. the details in the frames, the differences between frames, changes of scene, colours, etc. The second aspect is the physical characteristics of the video, i.e. resolution, frames per second, packet sizes etc.

The three parameters α, β and γ are set by drawing on the MOS evaluation and the desired eQoS curve. Perceived quality is also affected by the client software and hardware. Corrections by software algorithms or hardware can hide or amplify variations in quality. Both estimation methods are used together for video streaming.

α, β and γ have a larger range of values than for VoIP and audio streaming. They have to

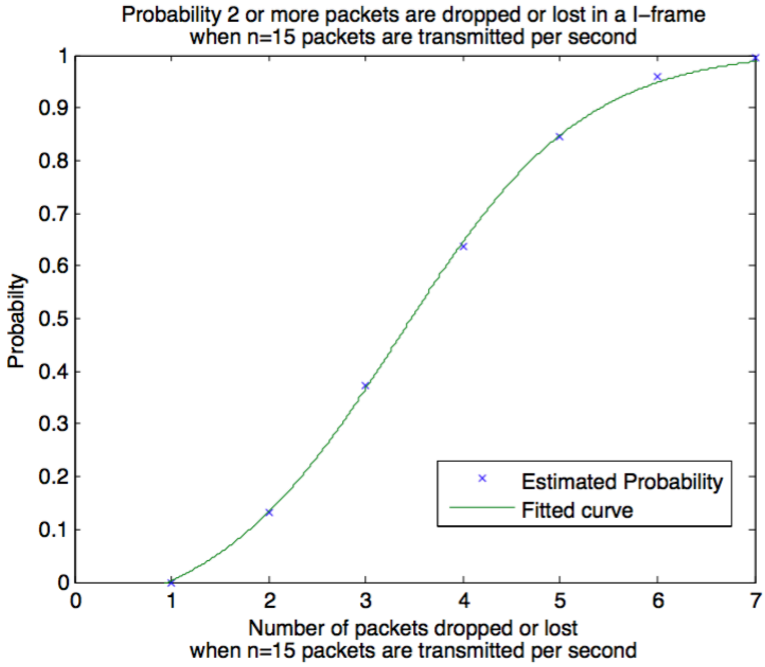

Fig. 4.7: Probability two or more packets are dropped in sequence from I frames as the number of packets dropped or lost increases.

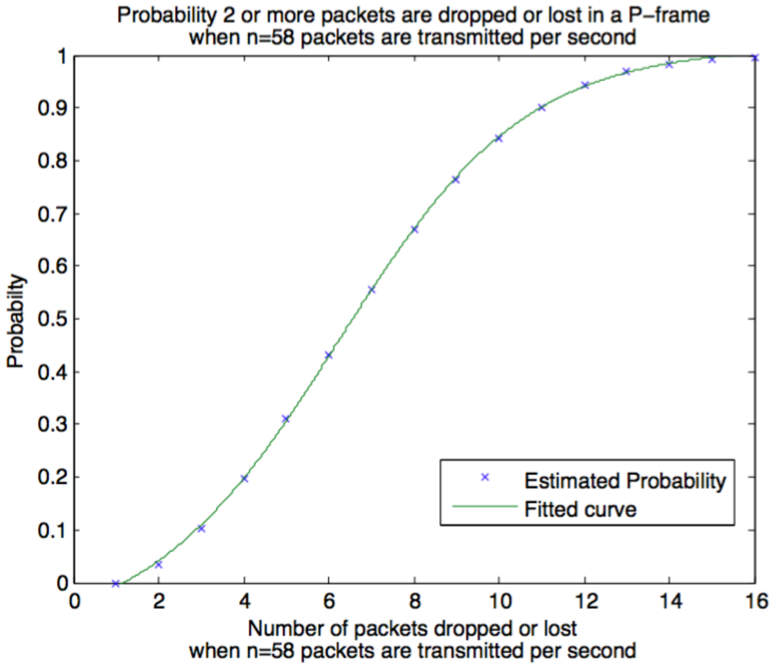

Fig. 4.8: Probability two or more packets are dropped in sequence from P frames as the number of packets dropped or lost increases.

satisfy the following two equations. The first equation limits eQoS to a range between 0 and 1,

$$\alpha + \beta + \gamma = 1; \tag{4.22}$$

while the second equation,

$$\beta + \gamma > \alpha, \tag{4.23}$$

gives more weight to the adjacency contribution for I and P frames than to the loss contribution. The values used in Chapter 6 are $\alpha = 0.4$, $\beta = 0.3$, $\gamma = 0.3$. As for VoIP, these three parameters are somewhat arbitrary in that they are chosen because they fit the desired and achievable QoE for the control video stream. This is a short video[2] with resolution 480×320. It is encoded using a basic H.264 encoding that uses only I and P frames. The video snippet included quick shot changes, many moving objects and a significant difference in contrast between objects. This increases the quantity of information to be conveyed per packet and per frame. The three parameters were chosen to balance the $F(B_D, B_d)$, $G_I(P_{ID}, P_{Id})$ and $G_P(P_{PD}, P_{Pd})$ contributions with the desired quality and they capture the independent probabilities of dropping events for $G_I(P_{ID}, P_{Id})$ and $G_P(P_{PD}, P_{Pd})$.

4.3 Considerations for the Practical Application of eQoS

eQoS needs some adjustments in order to be implemented in an AP. eQoS has to sample the number of dropped packets per second due to both queue overflow and queueing delays that exceed a maximum permitted value. The number of dropped packets per second can be counted using an array or a list for each flow which contains the time when the packet is dropped. This data structure has to be checked periodically to count the packets dropped per second and to remove packets that fall outside the one second window.

Mathematically the problem can be solved by implementing an EWMA to calculate the average number of packets dropped in the last second. Such an EWMA was used in the active queue management scheme RED to calculate the average queue length [9]. In this case the

[2] `https://www.youtube.com/watch?v=hbqMuvnx5MU`

EWMA is defined as:

$$avg_{(P_D+P_d)t} = (1 - w_{(P_D+P_d)}) \times avg_{(P_D+P_d)t-1} + w_{(P_D+P_d)} \times P_{D,d} \tag{4.24}$$

where:

$avg_{(P_D+P_d)t}$ is the normalised packet dropped average in a second at the instant t,

$avg_{(P_D+P_d)t-1}$ is the normalised packet dropped average in a second at the instant $t-1$,

$w_{(P_D+P_d)}$ is the weight associated with each dropping event and

$P_{D,d}$ is a 0-1 variable to indicate if the current packet is dropped or transmitted.

The weight, $w_{(P_D+P_d)}$, can assume values between 0 and 1. The dropping event $P_{D,d}$ is 0 if the packet is not dropped or 1 if the packet is dropped. Therefore, $avg_{(P_D+P_d)t}$ is normalised and assumes values between 0 and 1.

To calculate the number of packets dropped in a second, $avg_{(P_D+P_d)t}$ has to be multiplied by the total number of packets transmitted by the service during the sampling time interval. So the number of packets dropped in the sampling time interval is:

$$N_t \times avg_{(P_D+P_d)t}, \tag{4.25}$$

where N_t is the total number of packets transmitted in the sampling time interval.

The weight $w_{(P_D+P_d)}$ is a key parameter in equation 4.24. As discussed in [9], the EWMA is a low-pass filter and $w_{(P_D+P_d)}$ is the time constant for this filter. It serves two purposes in the average calculation in equation 4.24. It can determine the precision of the average calculation and it captures the reaction time needed for the prediction. Upper and lower bounds on $w_{(P_D+P_d)}$ can be obtained experimentally using the methodology given in [9].

An example is now used to illustrate this concept: if 10 packets are dropped per second from a sequence of 43 packets transmitted in an audio stream then figure 4.9 shows the average number of lost packets calculated using the EWMA given above when the lower limit set for $w_{(P_D+P_d)}$ is 0.01. Figure 4.10 shows the EWMA when the upper limit set for $w_{(P_D+P_d)}$ is 0.05. By the definition of EWMA [9], the standard deviation grows as $w_{(P_D+P_d)}$ grows. For a high value of $w_{(P_D+P_d)}$ the transient between the beginning of the stream and the calculation of 10 dropped packets per second is very small.

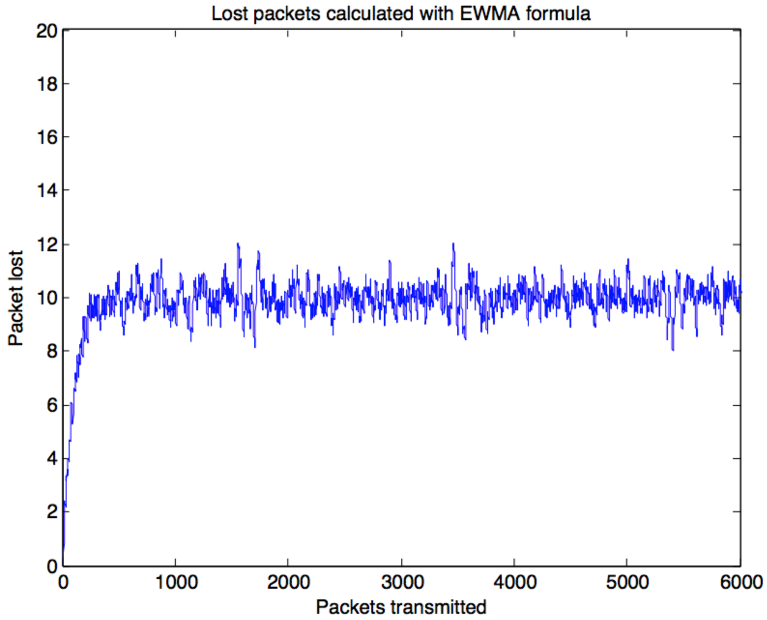

Fig. 4.9: Average number of lost packets for the example when $w_{(P_D+P_d)}$ = 0.01

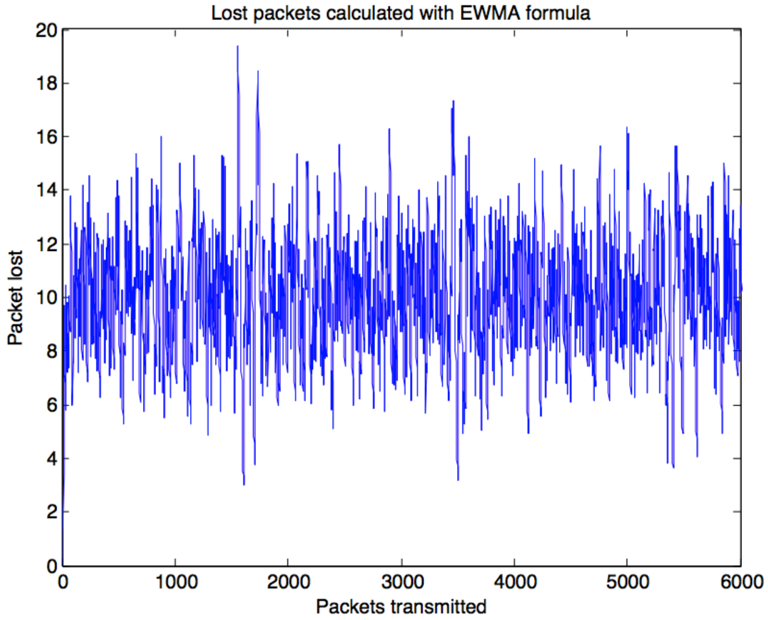

Fig. 4.10: Average number of lost packets for the example when $w_{(P_D+P_d)}$ = 0.05

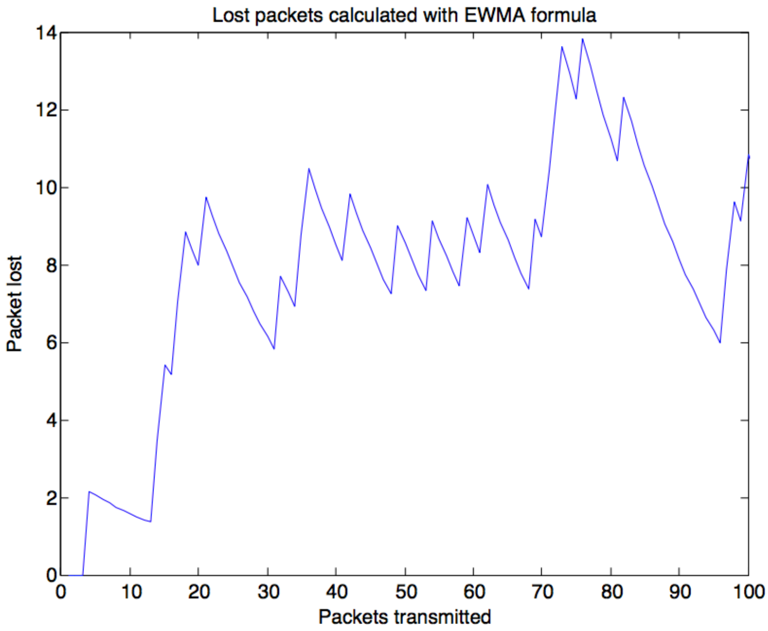

Fig. 4.11: Average number of lost packets for the example when $w_{(P_D+P_d)}$ = 0.05 for the first 100 packets transmitted.

Figure 4.11 shows the first 100 packets transmitted by the system shown in figure 4.10. The average value stabilises at approximately 10 dropped packets per second after less than half second. In this case the estimate provided by the system responds rapidly, but it should be noted that the standard deviation is higher than that obtained when lower values of $w_{(P_D+P_d)}$ are used.

High values of $w_{(P_D+P_d)}$ are preferable, in this case 0.05, because the purpose of eQoS is to estimate the value of QoE and use it to adjust the node properties so as to avoid future packet loss. A small transient is extremely helpful as it means the average number of dropped packets can be found quickly. It is possible to use other, more precise values for $w_{(P_D+P_d)}$. The value of 0.05 is sufficiently precise for use with the media stream discussed above as it is an acceptable compromise when oscillations in the EWMA are between 2 and 4 packets. Different values can be used depending on the specific application scenario used for eQoS.

Another important feature of the EWMA formula is that the number of packets dropped per second can be considered as continuous, rather than discrete. The EWMA equation can then be used to consider a small variation in the number of packets lost over time [121]:

$$avg((t+1)\delta) = (1 - w_{(P_D+P_d)}) \times avg(t\delta) + w_{(P_D+P_d)} \times P_{(t)\delta} \tag{4.26}$$

In the limit this yields the differential equation [121]:

$$\frac{d(\overline{avg})}{dt} = \frac{log_e(1 - w_{(P_D+P_d)})}{\delta}\overline{avg}(t) - \frac{log_e(1 - w_{(P_D+P_d)})}{\delta}\overline{P}(t) \tag{4.27}$$

This differential equation can then be used as a input in a control theoretic representation of the system [121].

4.4 Summary

In this chapter a new expected Quality of Service metric, eQoS, has been introduced. eQoS is defined as a perceived QoS metric and it captures the assessed QoS, i.e. the QoE. The goal of eQoS is to provide a better, near instantaneous estimate of the QoE at the AP in a infrastructure wireless network.

eQoS is designed to be used for real time services, it is calculated by counting the number of packets dropped or delayed beyond an acceptable limit in a one second time window. This chapter has shown how eQoS can be calculated for the three main real time services: VoIP, audio and video streaming. These time critical services are the most popular ones used on mobile devices in current and future wireless networks.

VoIP produces CBR traffic and the eQoS metric for it is a linear combination of two components: the first is the number of packets lost or delayed beyond the maximum limit allowed and the second is associated with the probability of dropping adjacent packets. The two components are appropriately weighted and graphically they are, respectively, a line and a sigmoidal curve.

Audio streaming does not, in general, produce CBR traffic. In this case the encoding algorithm chosen for the application of eQoS involved variable packet sizes, with a constant number of packet transmissions per second. Audio streaming produces a constant flow of information.

Hence eQoS for audio streaming can to be measured in the same way as, and with a similar formula to, VoIP.

Video Streaming uses VBR traffic; the encoding algorithm chosen for the calculation of the eQoS used variable packet sizes and a variable number of packet transmissions per second. Video streaming packets are not produced at regular intervals and their information content varies. The eQoS formula for video streaming uses a statistical average and is composed of three weighted factors: the loss contribution measured in bytes and the adjacency contribution for each type of frame as measured in packets.

The eQoS metric established in this chapter will form a key element of a novel Quality Queue Management, QQM, framework. The following chapter will introduce the components of the QQM algorithm and set out how they can operate together to manage traffic passing through a wireless access point.

Chapter 5

Quality Queue Management

One of the key components for managing traffic in wireless network controllers are the queues used to buffer data within the Access Point (AP). Traffic management algorithms can be implemented on these AP queues. If they are well designed then these clearly improve performance and provide Quality of Service (QoS) guarantees for real time services.

Queues can be managed using Active Queue Management (AQM) algorithms and some of these algorithms have been implemented on the network equipment most commonly used for routing in wired networks today. AQM schemes were originally designed to avoid and mitigate congestion on wired networks and they were not intended to provide quality assurance in a wireless environment. Such congestion avoidance mechanisms act to limit or prevent queue overflow. Typical parameters used in AQM schemes are the queue length, arrival rate and output speed. The schemes drop packets regardless of the impact this might have on the Quality of Experience (QoE) where the loss of a few packets can make an appreciable difference to the QoE provided.

Next generation wireless networks aim to achieve significantly higher throughput then existing systems and will make extensive use of multi queue systems; thus the wireless queue management algorithms of the future will carry out many functions beyond that of simple congestion avoidance. These new algorithms will address not only congestion avoidance but also

QoS assurance.

This chapter introduces a new mathematical model to describe packet transmission in a future wireless network where the Enhanced Distributed Channel Access (EDCA) method is employed. The mathematical model [15], derived using combinatorics, forms the basis for the design of a new class of traffic management algorithms called Quality Queue Management (QQM) schemes [15].

QQM schemes measure eQoS and implement congestion avoidance mechanisms whilst simultaneously managing contention windows (CW) and queueing priorities. QQM operates on a per flow basis across all services and traffic types. It is designed to reduce delay and discard traffic that is of poor quality.

Others have used combinatorial approaches for the modeling of the Distributed Coordination Function (DCF) in wireless sensor networks [130] [131] [189]. These methods were limited in scope, but indicated that a similar mathematical approach might provide insightful results for EDCA.

The chapter is organised as follows: firstly a novel combinatorial model that can be used to estimate the probability that a successful transmission occurs, the probability that a collision occurs and the probability that the channel is idle is presented. The QQM algorithm is then introduced along with its components: the active eQoS measurement system, the contention window controller and the queue management system. The systems are subsequently evaluated in Chapter 6.

5.1 Theoretical Analysis of a Multi Queue System

In this section the multi queue system detailed in chapter 2 is analysed. The analysis focuses on a theoretical description of packet behaviour in a future wireless network. Before presenting this analysis a brief review of Enhanced Distributed Channel Access (EDCA) is provided.

The formulation of a theoretical model necessitates the making of some underlying assumptions; the first of these relates to the wireless environment. It is assumed that the network uses a 160MHz channel with a Single Input and Single Output (SISO) configuration and a theoretical maximum throughput of 780Mbps on the channel, i.e. it is assumed that the VHT IEEE802.11ac [6] standard is used. All nodes in the wireless environment are assumed to use the IEEE802.11ac standard and packet headers contain all the information necessary for backward compatibility. Some SISO characteristics, such as Block Acknowledgment (BA) to improve the quality of the signal, Binary Convolutional Coding and Space-Time Block Coding (STBC) are omitted as they do not affect MAC layer behaviour. Packet transmission times for the wireless network are estimated using the relevant equation from the IEEE standard [6]. This will be discussed in more detail in Chapter 6. This first set of assumptions improves the design and optimisation of the mathematical model, and hence QQM, for use in a future wireless environment.

The second set of underlying assumptions relates to the channel access method used. It is assumed that the EDCA procedure implemented in IEEE802.11ae is used. This extension gives high priority to particular traffic and message types. It is assumed that Contention Free Bursting (CFB), also known as TXOP, is used. This was introduced along with EDCA in IEEE802.11e and was discussed in section 2.2. This second assumption makes sure that the mathematical model and QQM are suited for use on more modern mobile devices that implement TXOP.

The third assumption relates to the sequential transmission of packets in the wireless network. In the analysis it is supposed that the CSMA/CA feature is not used in the network. As mentioned previously, this feature negatively impacts on throughput and this is undesirable for high speed protocols such as IEEE802.11ac. At first glance this may seem to be contrary to the objectives of the QQM algorithm; that is, to optimise the throughput and efficiency of future wireless networks. The high speed of future wireless networks makes CSMA/CA an obstacle for enhancing IEEE802.11ac performance as CSMA/CA control packets are sent at a very low speed for backward compatibility. In the IEEE802.11ac protocol the MSDU dwarfs the amount of effective data contained in the packet. Collisions can be avoided using RTS and CTS packets but only at the expense of a significant reduction in throughput. The probability that a collision occurs can be reduced by employing techniques to conserve the channel bandwidth through avoidance of the transmission of control packets. This will be explored in section 5.5.1.2. Taken

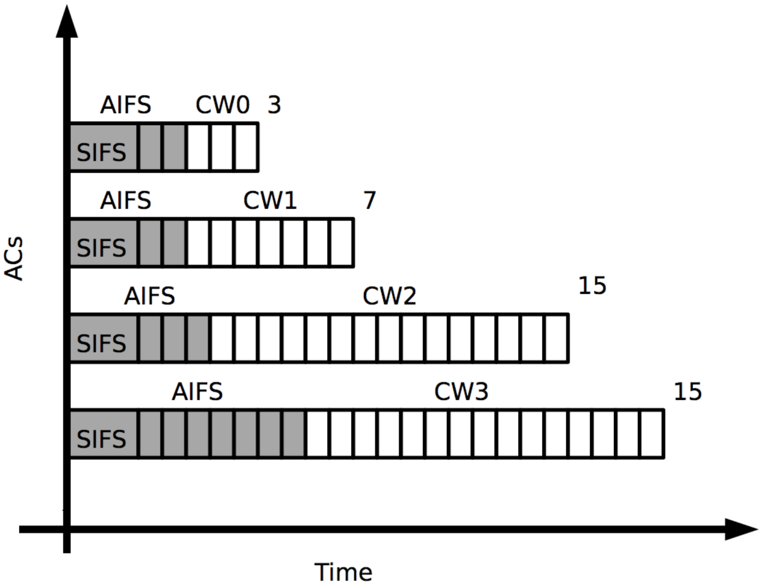

Fig. 5.1: EDCA: AIFS and Backoff times with CW_{min}

together, both QQM and the mathematical model aim to reduce the number of collisions and make CSMA/CA redundant. This third assumption focuses on eliminating the need for CSMA/CA through the deployment of QQM.

As set out in table 2.1, each AC has a backoff time of between 0 and CW_{min} time slots with an associated $AIFS[AC]$. For example, for AC_0 the backoff time is between 0 and three timeslots and $AIFS[0]$ is a SIFS time plus two time slots. Figure 5.1 summarizes graphically the relationship between each $AIFS$ and the CW_{min} time slots for each AC given in table 2.1.

For each new transmission attempt, each backoff time slot has an equal probability of being randomly chosen, therefore this probability follows a discrete uniform distribution across the interval $[0, CW]$. In the subsequent calculations the number of time slots in a contention window are considered rather than the contention window size. The probability of picking backoff time slot i follows a uniform distribution on the interval $[1, CW + 1]$ [190]:

$$\text{unif}\{1, CW + 1\}. \tag{5.1}$$

The key probabilities of interest in this thesis are the probability that a successful transmis-

Access Category	Prob. of TX	Prob. of collision	Prob. channel idle
AC_0	$\mathbb{P}_t(AC_0)_x$	$\mathbb{P}_{c_x}$	$\mathbb{P}_{e_x}$
AC_1	$\mathbb{P}_t(AC_1)_x$	$\mathbb{P}_{c_x}$	$\mathbb{P}_{e_x}$
AC_2	$\mathbb{P}_t(AC_2)_x$	$\mathbb{P}_{c_x}$	$\mathbb{P}_{e_x}$
AC_3	$\mathbb{P}_t(AC_3)_x$	$\mathbb{P}_{c_x}$	$\mathbb{P}_{e_x}$

Table 5.1: Notation for key probabilities associated with events that occur in time slot x

sion occurs, the probability that a collision occurs and the probability that the channel is idle. The probability a transmission occurs is denoted $\mathbb{P}_t$, the probability a collision occurs is $\mathbb{P}_c$ and the probability there are no packets queued for transmission is $\mathbb{P}_e$. The corresponding notation used for each access category is given in table 5.1. In the following subsections the focus shifts to determining these probabilities.

5.1.1 Determining the Probability a Packet is Successfully Transmitted

In this section we use combinatorics to determine the probability a packet is transmitted by a given AC [15] . The first challenge is to obtain a mathematical description of how each AC obtains access to the channel. The number of access categories with a packet ready for transmission is N_{AC}. This number will vary over time. As the focus of this section is on calculating the probability an AC gains access to the channel, a discussion on how N_{AC} can be estimated is deferred to section 5.4 below.

The probability of picking a time slot between $[1, CW_j + 1]$ is:

$$\mathbb{P}(AC_j) = \frac{1}{CW_j}, \tag{5.2}$$

where:

j is the index for each AC, for the system considered $j = 0, 1, 2, 3$ and

CW_j is the minimum time slot CW for each AC_j.

For example, if there are four ACs with a CW of eight backoff time slots, the probability of picking slot 0 is $\frac{1}{8}$. Using a similar methodology to [130], the probability that each of the other ACs chooses a slot other than slot 0 is:

$$1 - \mathbb{P}(AC_j) = \frac{CW_j - 1}{CW_j}. \tag{5.3}$$

In this example, this is $\frac{7}{8}$. Hence the probability that the first AC picks time slot 0 and the other three ACs pick time slots other than slot 0 will be the product of $\frac{1}{8}$ for the first AC and $(\frac{7}{8})^3$ for the other three. Since any one of the four ACs could have chosen slot 0, there are a total of four ways in which only one AC can pick slot 0 and so the result has to be multiplied for 4. Therefore, the probability that one AC picks slot 0 and the other three ACs do not will be $4 \times \left(\frac{1}{8} \times \left(\frac{7}{8}\right)^3\right)$. As will be shown below, this probability can be found more directly through the use of permutations and combinations.

Returning to the previous example of four ACs each with a CW of eight backoff time slots, the probability that an AC gains access to the channel is given by the ratio between the number of backoff time scenarios where only one AC is ready to transmit and the total number of backoff time scenarios possible. This can be found using combinatorics as follows: The number of possible ways in which three of the ACs are not ready to transmit is 7^3. This is the number of permutations with repetition, $P'_{n,r} = n^r$, of $r = 3$ numbers chosen from $n = 7$ numbers. The number of scenarios where one AC is ready to transmit and the other three are not is given by 4×7^3. The total number of possible backoff time scenarios possible is the permutation with repetition that gives the number of ways 8 objects can be taken 4 at a time i.e. 8^4. Therefore, the probability that one AC picks slot 0 and the other three ACs do not will be $\frac{4 \times 7^3}{8^4}$. This agrees with the previous calculation of this probability.

In order to describe the ECDA system it is necessary to generalise these calculations. Let each distinct access category in the system be indexed by j. The number of slots in the contention window associated with each AC_j is given by CW_j. Let the number of AC_js that are seeking to access the channel at a given instant in time be N_{AC_j}. Assuming that x is the number of ACs that are ready to transmit and that y is the index of the time slot to be used for the transmission then the total number of ways $(N_{AC_j} - x)$ ACs are not ready to transmit is given by the following permutation with repetition [130]:

$$P'_{(CW_j-y),(N_{AC_j}-x)} = (CW_j - y)^{N_{AC_j}-x}.$$

The following notational simplification is used to assist the reader in the subsequent discussion:

$$P'_{(CW_j-y),(N_{AC_j}-x)} = P'_{N_{AC_j}(x,y)}.$$

This gives the total number of ways $(N_{AC_j} - x)$ ACs are ready to transmit in slot y or later:

$$P'_{N_{AC_j}(x,y)} = (CW_j - y)^{N_{AC_j}-x}. \tag{5.4}$$

The repetitions included in $P'_{N_{AC_j}(x,y)}$ represent future collisions. Repetitions are selections of the same slot, as shown in [130], and in this scenario the backoff times will expire at the same time and a collision will occur.

The probability that an AC_j transmits in time slot i is:

$$\mathbb{P}_t(AC_j)_i = N_{AC_j} \times (\mathbb{P}_{AC_j})^{N_{AC_j}} \times P'_{N_{AC_j}(1,i+1)}. \tag{5.5}$$

This can be used to verify the calculations given previously for a system consisting of four ACs, each with a CW of eight backoff time slots. For this example, equation 5.5 gives the probability that an AC gains access to the channel at time slot 0 as:

$$\mathbb{P}_t(AC_j)_0 = N_{AC_j} \times (\mathbb{P}_{AC_j})^{N_{AC_j}} \times P'_{N_{AC_j}(1,0+1)} = 4 \times \left(\frac{1}{8}\right)^4 \times 7^3 = 4 \times \frac{1}{8} \times \left(\frac{7}{8}\right)^3.$$

The focus of this dissertation is on an infrastructure wireless network that implements EDCA. In this case there will be four different types of AC that compete for access to the channel. From figure 5.1, it can be inferred that at time slot 0 only two types of ACs are able to transmit: AC_0 and AC_1. Using equations 5.4 and 5.5 it is possible to calculate the probability that one AC_0 is transmitting in slot 0 as [130]:

$$\mathbb{P}_t(AC_0)_0 = N_{AC_0}(\mathbb{P}_{AC_0})^{N_{AC_0}} P'_{N_{AC_0}(1,1)} \times (\mathbb{P}_{AC_1})^{N_{AC_1}} P'_{N_{AC_1}(0,1)}. \tag{5.6}$$

Equation 5.6 captures the probability that slot 0 is chosen for a single AC_0 and that the remaining AC_0s and all AC_1s have chosen a time slot other than slot 0. Using a similar argument it is possible to calculate the probability that one AC_1 is transmitting in time slot 0 as:

$$\mathbb{P}_t(AC_1)_0 = (\mathbb{P}_{AC_0})^{N_{AC_0}} P'_{N_{AC_0}(0,1)} \times N_{AC_1}(\mathbb{P}_{AC_1})^{N_{AC_1}} P'_{N_{AC_1}(1,1)}. \tag{5.7}$$

No packets are transmitted when slot 0 is empty for all AC_0s and all AC_1s . The probability that slot 0 is empty, $\mathbb{P}_{e_0}$, is the product of the probability that slot 0 is not randomly chosen by any of the AC_0s and the probability that slot 0 is not randomly chosen by any of the AC_1s:

$$\mathbb{P}_{e_0} = \frac{P'_{N_{AC_0}(0,1)}}{P'_{N_{AC_0}(0,0)}} \times \frac{P'_{N_{AC_1}(0,1)}}{P'_{N_{AC_1}(0,0)}}. \tag{5.8}$$

Each term in the product is the ratio of the number of ways that any slot other than slot 0 can be chosen to the total number of possible slot choices when there are no restrictions on the slot that can be chosen.

5.1.2 Determining the Probability a Collision Occurs

The other important events in the network that need to be accounted for are collisions [15]. Collisions occur when the backoff periods for at least two ACs expire at the same time. For the purposes of this work it is assumed that collisions occur between a maximum of two stations at the same time [130]. This is a reasonable assumption as the number of ACs competing to access the channel depends on the number of ACs that are ready to transmit data and it is likely that only a few ACs are in saturation at the same time, i.e. it is likely that only a few ACs have packets waiting to be transmitted.

The probability a collision occurs, $\mathbb{P}_c$, at a given slot time is the sum of the collision probabilities between two AC_0s, two AC_1s or between one AC_0 and one AC_1. The probability of a collision in time slot 0, $\mathbb{P}_{c_0}$ is:

$$\mathbb{P}_{c_0} = \mathbb{P}_c(AC_0)_0 + \mathbb{P}_c(AC_1)_0 + \mathbb{P}_c(AC_0, AC_1)_0. \tag{5.9}$$

Using a similar methodology to [130], the probability two AC_0s give rise to a collision, $\mathbb{P}_c(AC_0)_0$, is given by:

$$\mathbb{P}_c(AC_0)_0 = \frac{\begin{pmatrix} N_{AC_0} \\ 2 \end{pmatrix} [(CW_0 - 1)^{(N_{AC_0}-2)}(CW_1 - 1)^{(N_{AC_1})}]}{(CW_0)^{N_{AC_0}}(CW_1)^{N_{AC_1}}}. \tag{5.10}$$

Equation 5.10 is obtained by calculating the number of possible combinations that arise when two N_{AC0} have chosen time slot 0 and all the N_{AC1}s have chosen a time slot other then 0. This is divided by all of the slot time choices between N_{AC_0} and N_{AC_1}. Similarly the probability two AC_1s give rise to a collision, $\mathbb{P}_c(AC_1)_0$, is:

$$\mathbb{P}_c(AC1)_0 = \frac{\begin{pmatrix} N_{AC_1} \\ 2 \end{pmatrix} [(CW_0 - 1)^{(N_{AC_0})}(CW_1 - 1)^{(N_{AC_1}-2)}]}{(CW_0)^{N_{AC_0}}(CW_1)^{N_{AC_1}}}. \tag{5.11}$$

The final probability to be calculated is that of a collision due to the time slot choices of one AC_0 and one AC_1. The probability that both an AC_0 and an AC_1 have chosen time slot 0 is:

$$\mathbb{P}_c(AC_0, AC_1)_0 = (N_{AC_0} \times N_{AC_1}) \times \frac{[(CW_0 - 1)^{(N_{AC_0}-1)}(CW_1 - 1)^{(N_{AC_1}-1)}]}{(CW_0)^{N_{AC_0}}(CW_1)^{N_{AC_1}}}.$$

The calculation of $\mathbb{P}_c$ is more complicated when more than two types of AC are involved. The two possible outcomes are that a collision occurs or that a collision does not occur. These two probabilities must sum to 1 because one or other of these possible outcomes must happen. In order to find the probability a collision occurs we must subtract the probability a collision does not occur from one. If a collision does not occur then either a successful packet transmission occurs or there are no packets queued for transmission. There are two possible ways that a successful transmission might occur: either a packet is successfully transmitted by AC_0 or a packet is successfully transmitted by AC_1. So the probability a collision occurs in time slot 0 is:

$$\mathbb{P}_{c_0} = (1 - ((\mathbb{P}_t(AC_0)_0 + \mathbb{P}_t(AC_1)_0) + \mathbb{P}_{e_0})) = 1 - \mathbb{P}_t(AC_0)_0 - \mathbb{P}_t(AC_1)_0 - \mathbb{P}_{e_0}. \tag{5.12}$$

This argument can be extended to find the probability a collision occurs when more than two ACs are competing for access in a given time slot.

5.1.3 Extending the Model to AC_2 and AC_3

Figure 5.1 shows three ACs competing for access to the channel in time slot 1. In particular, it shows the arbitration inter-frame spacing, $AIFS$, for each of the access categories. For AC_0 and AC_1, $AIFS[AC_0]$ and $AIFS[AC_1]$ are both one SIFS and two time slots long. AC_0 and AC_1 have the highest priority and so the $AIFS$ for AC_2 and AC_3 will be greater. For AC_2, $AIFS[AC_2]$ is one SIFS and three time slots long. This means that for AC_2 time slot 0, the first slot it can use for transmission, corresponds to time slot 1 for AC_0 and AC_1. For AC_3 the arbitration inter-frame spacing, $AIFS[AC_3]$, is longer than for all the other ACs as it has the lowest priority. Its $AIFS$ is one SIFS and 7 time slots long. So time slot 0 for AC_3 corresponds to time slot 5 for AC_0 and AC_1 and time slot 4 from AC_2. $AIFS[AC]$ values are summarized in table 2.1 .

The probability an AC_2 transmits in a given time slot depends on the status of the AC_0s and AC_1s. If time slot 0 is not chosen by any AC_0s or AC_1s, or if they do not have any packets to transmit, then an AC_2 has the opportunity to transmit data. The probability an AC_2 transmits data in time slot 1, $\mathbb{P}_t(AC_2)_1$, is the product of three probabilities. The first two of these are the probabilities all AC_0s and AC_1s competing for the channel have chosen a time slot other than slot 1. The third is the probability that only one of the AC_2s has randomly chosen time slot 1. This gives:

$$\mathbb{P}_t(AC_2)_1 = \frac{P'_{N_{AC_0}(0,2)}}{P'_{N_{AC_0}(0,0)}} \times \frac{P'_{N_{AC_1}(0,2)}}{P'_{N_{AC_1}(0,0)}} \times N_{AC_2}(\mathbb{P}_{AC_2})^{N_{AC_2}} P'_{N_{AC_2}(0,1)}.$$

Though $CW_{AC1_{min}}$ overlaps with CW_{AC_3} in practice it is unlikely that $\mathbb{P}_t(AC_3)_5$ contributes significantly when AC_1 is competing to access the channel, see section 5.1.4. Thus the contribution of all AC_1s will be neglected in the calculation of $\mathbb{P}_t(AC_3)$, $\mathbb{P}_{c_5}$ and $\mathbb{P}_{e_5}$ below.

Assuming that a transmission from AC_3 occurs when the AC_0s and AC_1s are not competing to access the channel. The probability that AC_3 transmits a packet is:

$$\mathbb{P}_t(AC_3)_5 = \frac{P'_{N_{AC_2}(0,5)}}{P'_{N_{AC_2}(0,0)}} \times N_{AC_3}(\mathbb{P}_{AC_3})^{N_{AC_3}} P'_{N_{AC_3}(0,1)}. \tag{5.13}$$

$\mathbb{P}_{e_1}$ is the probability time slots 0 and 1 have not been randomly chosen by any of the AC_0s and AC_1s, and that time slot 0 is not randomly chosen by any of the AC_2s. It is given by:

$$\mathbb{P}_{e_1} = \frac{P'_{N_{AC_0}(0,2)}}{P'_{N_{AC_0}(0,0)}} \times \frac{P'_{N_{AC_1}(0,2)}}{P'_{N_{AC_1}(0,0)}} \times \frac{P'_{N_{AC_2}(0,1)}}{P'_{N_{AC_2}(0,0)}}. \tag{5.14}$$

Neglecting the contribution of the AC_1s, $\mathbb{P}_{e_5}$ is the probability that time slots 0 to 4 are not randomly chosen by the AC_2s, and that time slot 0 is not randomly chosen by any of the AC_3s. This gives:

$$\mathbb{P}_{e_5} = \frac{P'_{N_{AC_2}(0,5)}}{P'_{N_{AC_2}(0,0)}} \times \frac{P'_{N_{AC_3}(0,1)}}{P'_{N_{AC_3}(0,0)}}. \tag{5.15}$$

The procedure to estimate the CW used in this thesis, assumes that the probability two sequential time slots are empty is low and so it is neglected in these calculations i.e. It is assumed that the number of empty time slots does not exceed one. $\mathbb{P}_e$ should be interpreted as the probability there is an interval of length e time slots where no transmission occurs.

The probability of a collision in time slot 1 is:

$$\mathbb{P}_{c_1} = 1 - \mathbb{P}_t(AC_0)_1 - \mathbb{P}_t(AC_1)_1 - \mathbb{P}_t(AC_2)_0 - \mathbb{P}_{e_1}. \tag{5.16}$$

In time slot 5 the probability of a collision is:

$$\mathbb{P}_{c_5} = 1 - \mathbb{P}_t(AC_2)_4 - \mathbb{P}_t(AC_3)_0 - \mathbb{P}_{e_5}. \tag{5.17}$$

Using the methodology outlined above it is now possible to estimate the probability that a successful transmission occurs, the probability that a collision occurs and the probability that the channel is idle. Before looking at how these can be used in practice, some consideration must be given to both the contention window sizes and the queues associated with AC_0 and AC_1.

5.1.4 Contention Window Sizes and Queueing in AC_0 and AC_1

The first important consideration that merits further discussion relates to how the CW size is determined. The CW value is chosen randomly in the interval $[0, CW_{min}]$.

The time slot is chosen randomly and is decremented when the channel is idle and remains unchanged when the channel is busy. Therefore, the distribution of CW values is skewed towards time slot 0 over a range of values between 0 and CW_{min}.

A suitable methodology for instantaneous sampling of CW values at all the stations competing for access the channel is not straightforward and is beyond the scope of this work. On the other hand, it is possible to estimate CW sizes by considering each station separately. This can be estimated using either an empirical or a theoretical approach.

One empirical method that might be suited for CW estimation is the German Tank Problem [132]. This was discussed in section 3.2.1.1. With just a few samples this method can be used to estimate the average CW size to an acceptable level of precision. An alternate, theoretical approach would be to use the statistical expectation associated with a discrete uniform distribution. If time slots are randomly distributed in the range between 0 and $\mathbb{E}[CW_i]$, then an integer upper bound for $\mathbb{E}[CW_i]$ is [134]:

$$\lceil \mathbb{E}[CW_i] \rceil = \sum_{n=0}^{CW_{min}} \mathbb{P}_n \times n, \tag{5.18}$$

where $\mathbb{P}_n$ is the probability of picking a time slot between 1 and $(CW_{min} + 1)$, i.e. $\frac{1}{CW_{min}+1}$

Both these methods are dynamic: using the first method the CW size needs to be estimated periodically, while using the second method the CW size needs to be estimated every time the upper CW limit is exceeded. Despite their differences, both methods provide acceptable results for use in the theoretical calculations discussed above.

The results from subsection 5.1.2 can be used to determine the impact collisions have on the QoE and on the CW size used by the QQM algorithm. They cannot be used to determine the precise value of CW; for example, when it is doubled because a collision has occurred. For practical applications, and for the evaluation of the mathematical model presented above, the CW_{min} value is estimated by using the value for an individual mobile station.

A second important consideration relates to the AC_0 and AC_1 queues. Q_0 is the queue associated with AC_0 and Q_1 is the queue associated with AC_1. Both Q_0 and Q_1 are functions of the characteristics of AC_0 and AC_1 respectively.

Future wireless networks will have a very high throughput particularly in comparison to the packet transmission frequency for VoIP traffic. The exact instant a VoIP conversation starts is unpredictable. Protocol G.729 [57] transmits and receives at a frequency of 50 packets per second, $f_{VoIP} = 50$Hz, i.e. a packet is transmitted or received every $20ms$. The IEEE802.11ac protocol can be considered as a spatial stream with a 160MHz channel. In this case a packet is transmitted about every $0.265ms$, not including collisions and empty slots due to backoff. At a frequency of 3.8KHz and assuming a worst case scenario where CSMA/CA control packets are used, this corresponds to more than three thousand packets per second. Even if a very large number of VoIP conversations are present in the network, the frequency at which they access the channel is low when compared to the transmission frequency used for the IEEE802.11ac packets. Since AC_0 has highest priority, Q_0 is likely to contain at most one or two packets.

VoIP calls may be synchronised by application software or software running on the node itself or by network quality management software; therefore, VoIP packets from wired nodes may arrive at the AP at the same point in time. This problem is evident in class based networks [37] where the TXOP feature is used.

AC_1 has a lower priority than AC_0, but it is still subject to similar effects and considerations. An audio or video stream, encoded using MPEG4 [66] [67] as described in section 2.4.4, transmits one I frame per second and 29 P frames per second. I frames are, on average, composed of 12 packets with a size of 1024 bytes each. An average of 2 packets is needed for each P frame. The frequency of the video transmission, f_{Video}, is 30 frames per second but the number of packets per frame is variable. AC_1's TXOP feature is 2.008ms long and is sufficient for the transmission of 45 streaming packets.

From this discussion it can be concluded that AC_0 and AC_1 only periodically occupy the channel for packet transmissions. The frequency with which they transmit packets is related to the total number of VoIP and streaming packets passing through the AP. AC_2 manages traffic that has a low frequency of demand for access to channel. This traffic is similar to that of AC_1 and the considerations for AC_0 and AC_1 remain valid for AC_2. AC_3 will either make use of any remaining transmission time to send its data or else will transmit until it exceeds the upper limit imposed on its throughput.

In the next section a practical application of the theoretical model is proposed.

5.2 An Application of the Theoretical Model

This section describes an application of the mathematical model to estimate the average time needed to transmit a packet over a future wireless network. This application is also useful for estimating the throughput on a future wireless network [15].

The Bianchi model [123] can be used to estimate the average time needed to transmit a packet in a DCF environment. It can also be used to estimate the throughput. In this section this model is extended for use in future wireless networks, where the EDCA method is used to access the channel.

In subsection 5.1.4 it was noted that AC_0s have the smallest CW size and so have the best opportunity for access to the channel in order to transmit all their packets. AC_0s are in competition with AC_1s for access to the channel; however, AC_1 has a larger CW size than AC_0. Both

AC_0s and AC_1s transmit all their queued packets if the total volume of traffic to be transmitted does not exceed the available throughput. The number of AC_0s and AC_1s competing for access to the channel affects the associated transmission, collision and idle channel probabilities.

The average time required to transmit a packet is equal to the time need to transmit the packet itself plus the time lost due to collisions, when the channel is idle and due to backoff [123]. Both the collision time and the backoff time are estimated using the probabilities found in subsection 5.1.2. The average packet transmission time for AC_0 is [123]:

$$TxTime_{AC_0} = Tx_{AC_0} + Tc \times \mathbb{P}_{c_0} + Te \times \mathbb{P}_{e_0} \tag{5.19}$$

where:

Tx_{AC_0}: is the time to transmit CSMA/CA packets, a VoIP packet and ACK packet on IEEE802.11ac,

Tc: is the time lost for a packet transmission, or RTS if CSMA/CA is used, and the AIFS time when a collision occurs,

Te: is idle time, it can be considered as a single empty $9\mu s$ time slot.

Similarly, the average packet transmission time for AC_1 is:

$$TxTime_{AC_1} = Tx_{AC_1} + Tc \times \mathbb{P}_{c_0} + Te \times \mathbb{P}_{e_0}. \tag{5.20}$$

Both AC_0s and AC_1s manage UDP traffic with different packet sizes. When G.729 encoding is used AC_0s have a fixed packet size of 20 bytes and a fixed transmission rate. AC_1s have a variable packet size, but this can be approximated by an average value.

Packet transmission times for the other two Access Categories AC_2 and AC_3 are as follows:

$$TxTime_{AC_2} = Tx_{AC_2} + Tc \times \mathbb{P}_{c_1} + Te \times \mathbb{P}_{e_1}, \tag{5.21}$$

$$TxTime_{AC_3} = Tx_{AC_3} + Tc \times \mathbb{P}_{c_5} + Te \times \mathbb{P}_{e_5}. \tag{5.22}$$

Tx_{AC_2} and Tx_{AC_3} depend on the Transport Layer protocol used; for example, TCP and its ACK or UDP.

The throughput calculation depends on the network traffic configuration. Considering that AC_0 and AC_1 have the highest probability of access to the channel, $\mathbb{P}_{e_1}$ and $\mathbb{P}_{c_1}$ are calculated when both Q_0s and Q_1s are empty, that is when AC_0 and AC_1 have transmitted all their queued packets. AC_2 has an higher probability of access to the channel than AC_3, thus the probabilities $\mathbb{P}_{e_5}$ and $\mathbb{P}_{c_5}$ are calculated when Q_2 is empty or when the protocols used for AC_2 have reached their upper throughput limits.

In each one second interval the time spent transmitting AC_0 packets at the access point is:

$$Time_{AC_0} = TxTime_{AC_0} \times f_{VoIP} \times (N_{AC_0} - 1) + TXOPTime_{AC_0} \times f_{VoIP}, \tag{5.23}$$

where

f_{VoIP} is the VoIP frequency of access to the channel, for G.729 this is 50Hz i.e. 50 packets per second, and

$TXOPTime_{AC_0}$ is the time spent for the transmission of $(N_{AC_0} - 1)$ VoIP packets and, if necessary, the associated CSMA/CA control packets.

The audio is transmitted by AC_0 with a frequency f_{Audio}. For example, AAC encoding needs to transmit 47 packets per second, i.e. at a frequency of 47 Hz. f_{Audio} is approximated by f_{VoIP} and audio packets are considered to be transmitted with VoIP packets in $TXOPTime_{AC_0}$ when the audio is being streamed from the wired to the wireless network.

As packets may arrive at the AP at the same time; channel access using the TXOP feature has the same $\mathbb{P}_e$ and $\mathbb{P}_c$ as for a single packet transmission. In order to take account of this $TXOPTime_{AC_0}$ must be multiplied by $(N_{AC_0} - 1)$.

Future wireless networks have enough throughput available to transmit a large number of video streams. The number of streams that can be handled depends only on the bandwidth, in bits per second, that each stream requires. The time needed to transmit a video stream through AC_1 is:

$$Time_{AC_1} = TXOPTimeI_{AC_1} \times f_I \times N_{AC_1} + TXOPTimeP_{AC_1} \times f_P \times N_{AC_1}, \tag{5.24}$$

where $TXOPTimeI_{AC_1}$ and $TXOPTimeP_{AC_1}$ are the TXOP times needed to transmit I frames and P fames respectively. For example, an I frame may, on average, consist of 12 packets with 1024 bytes per packet and a P frame may have, on average, 2 packets with 1024 bytes per packet. Taking all frames together, a single video stream has a typical channel access frequency of 30Hz. For I frames alone the channel access frequency, f_I, is typically of the order of 1Hz, while for P frames the channel access frequency, f_P, is of the order of 29 Hz.

As mentioned above, when the Q_0s and Q_1s are empty then the AC_0s and AC_1s have no packets to transmit and the channel is shared between AC_2 and AC_3.

In each one second interval the respective times spent transmitting AC_2 and AC_3 TCP packets per flow at the Access Point (AP) are:

$$Time_{AC_2} = N_{f_{AC_2}} \times TxTime_{AC_2} \times CWavg_{AC_2} \times N_{RTT}, \tag{5.25}$$

$$Time_{AC_3} = N_{f_{AC_3}} \times TxTime_{AC_3} \times CWavg_{AC_3} \times N_{RTT}. \tag{5.26}$$

where $N_{f_{AC_2}}$ and $N_{f_{AC_3}}$ are, respectively, the number of flows in AC_2 and AC_3. $TxTime_{AC}$ includes the time to transmit the TCP data packet and the time it takes for the TCP acknowledgement packet to travel in the opposite direction. N_{RTT} is the average number of round trip times per second. $CWavg_{AC}$ is the average congestion window size for a given AC. The product of $CWavg_{AC}$ and N_{RTT} may be viewed as way of averaging across the TCP flows. If UDP traffic is present then it will have to be included in the $Time_{AC}$ calculation in a similar way to the traffic managed by AC_0. The product of $CWavg_{AC}$ and N_{RTT} acts to smooth bursty traffic and take account of intervals when other ACs are sending packets. An example of a source of AC_2 traffic is a Telnet connection.

The TCP protocol transmission speed is limited by the congestion window [49] and queue size. The expected TCP throughput is:

$$\lceil \mathbb{E}[Thr_{tcp}] \rceil = \frac{CongestionWindow_{Max}}{RTT} \times N_{TCP}, \tag{5.27}$$

where:

RTT: is the Round Trip Time,

$CongestionWindow_{Max}$ is the maximum contention window size for the TCP flow,

N_{tcp} is the number of TCP flows.

The total throughput is estimated over a long period of time; for example, over a one second interval. One possible application of this model is in the optimisation of throughput as it can be used to control the CW [114] [129] based on the observed traffic mix.

5.3 Model Limitations

This section summarises a few limitations of the model presented above.

By sampling just a few variables at the AP, it is possible to predict the volume of traffic generated by each of the four ACs. However, it is impossible to predict which station is transmitting and when. This is because the model is based on combinatorics and some of the variables used in the formulas are estimated using average values. For this reason it is not possible to use the model to provide specific detail on the traffic at each mobile station in the network.

The model estimates the probability that a collision occurs, but it does not provide an estimate of the number of packets involved in a collision. It is not possible to infer from the model if the packets involved in a collision are from VoIP, audio or video streams, best effort or background traffic. Finally, it is not possible to infer the number of times the same packet is involved in collisions.

5.4 Estimating the Number of ACs Simultaneously Contending for Channel Access

The transmission, collision and idle channel probabilities calculated using the theoretical model given above will be used in section 5.5 to design a system to manage the contention window and optimise transmissions in a wireless network. Before the complete Quality Queue Management algorithm is described it is necessary to find a method to use to determine the number of ACs simultaneously contending for channel access [15].

A wireless network with N VoIP calls in progress has, in effect, $N+1$ classes in competition for access to the channel because the AP must be included in the calculations. In this dissertation it is assumed that the AP accesses the channel with the same frequency as a mobile station. However, the frequency at which an AP seeks to access the channel may well differ from that of the mobile stations because the AP manages all traffic being transmitted to the mobile stations.

It is necessary to determine the relationship between the frequency of channel access requests for a service and the number of mobile stations simultaneously accessing the channel. In the following discussion this is explored for the three traffic types considered in this dissertation.

Protocol G.729, described in section 2.4.2, is used for VoIP traffic. It needs to transmit 50 packets per second, corresponding to a frequency of 50 channel access requests per second, i.e. one access attempt every $20ms$. Without considering TXOP, the worst case transmission time for a single packet, including the Layer 2 ACK control packet, on future wireless networks is about $150\mu s$. If a packet is transmitted every $20ms$, then, theoretically, a maximum of 120 packets can be transmitted in $20ms$.

If it is assumed that every mobile station with a phone call in progress needs to transmit a packet every $20\mu s$, then there are $N + 1$ ACs seeking access to the channel in each of the 120 $slots$ that make up a $20ms$ interval. The problem is to estimate the likelihood that more than one AC accesses the channel in the same $150\mu s$ slot as this will give rise to a collision.

Statistically, the number of ACs competing for the channel have the same likelihood of picking each individual time slot for transmission. It is necessary to find how many possible ways there are for at least two of the $N + 1$ mobile nodes to pick the same slot amongst the 120 $slots$ on offer. Using the same methodology as in section 5.1.1 this can be found using permutations with repetition. The required probability will be given by the ratio of the number of permutations with at least one repetition, i.e. the number of ways that at least one collision can occur, to the number all the possible permutations:

$$\mathbb{P}_{(rep)} = \frac{P'_{AC0_{(slots,N+1)}} - P_{AC0_{(slots,N+1)}}}{P'_{AC0_{(slots,N+1)}}} \tag{5.28}$$

Where $\mathbb{P}_{(rep)}$ is the probability that multiple ACs choose the same time slot. As in section 5.1.1, P and P' are the permutations without repetitions and with repetitions respectively. Here, $n = slots$ and $k = N + 1$, so that:

$$P_{AC0_{(slots,N+1)}} = \frac{(slots)!}{((slots) - (N + 1))!} \tag{5.29}$$

and

$$P'_{AC0_{(slots,N+1)}} = slots^{(N+1)} \tag{5.30}$$

The probability that a repetition occurs i.e. that two ACs compete for access to the channel is given by:

$$\mathbb{P}_{(rep=2)} = \frac{slots \times \begin{pmatrix} N+1 \\ 2 \end{pmatrix} \times P_{AC_{0(slots-1,N-1)}}}{P'_{AC_{0(slots,N+1)}}} \tag{5.31}$$

Figure 5.2 shows the variations in $\mathbb{P}_{(rep=2)}$ as the number of phone calls in the system grows. In the graph on the left there are 120 $slots$ in each $20ms$ interval. In the graph on the right a worst case scenario of 60 $slots$ in each $20ms$ interval is shown. From the graphs it can be seen that until there are more than ten calls in progress, the probability only two AC_0s compete for access to the channel at the same time is over 90%; therefore it is reasonable to consider that only two AC_0s are ready to transmit at any given time provided the number of phone calls flowing in the wireless network does not exceed ten.

Audio streams must be considered next. These are managed by AC_0 and are similar to unidirectional phone calls and have, approximately, the same channel access frequency as VoIP calls.

Finally, video streams must be considered. In this case the traffic rate is variable and there are 30 frames transmitted per second, i.e. one frame is transmitted every $33ms$. In the worst case, when many flows are being streamed from the same node, the transmission time for each frame containing video can be assumed to be twice the TXOP time. Each $slot$ is assumed to be $1.5ms$ long, therefore in every $33ms$ interval there are 22 $slots$ are available for transmission, or in the worst case only 11 $slots$ are available.

Figure 5.3 shows the experiments carried out to explore the probability that a maximum of two AC_1 video streams access the channel at the same time in the same slot. From this it can be inferred that for less than six flows, there is a 90% chance that only two of them are competing for access to the channel.

The discussion and experiments above confirm that the number of AC_0s and AC_1s competing for access to the channel at the same time does not exceed two if less than ten phone calls and audio streams or less than six video streams are present in a wireless network.

Based on the above discussion, the probability that one AC_0 is transmitting in slot 0 is now

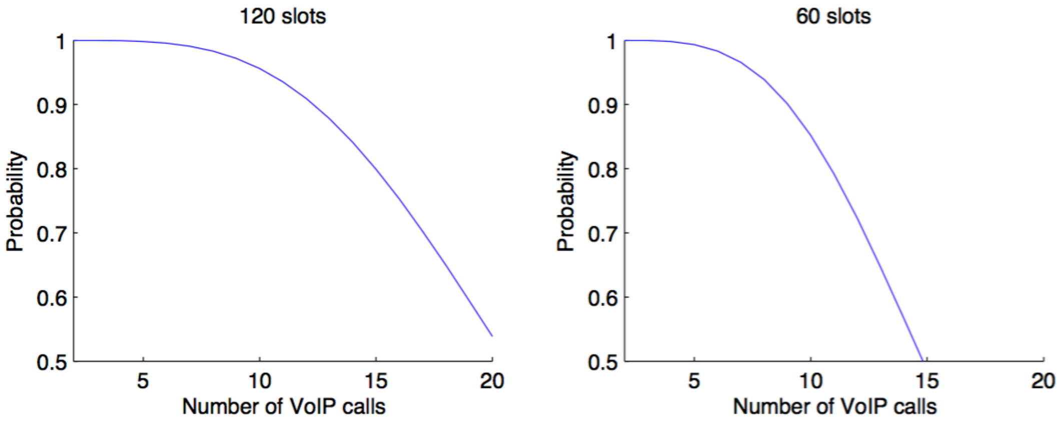

Fig. 5.2: Probability a maximum of two AC_0s access the channel at the same time in the same slot for VoIP calls.

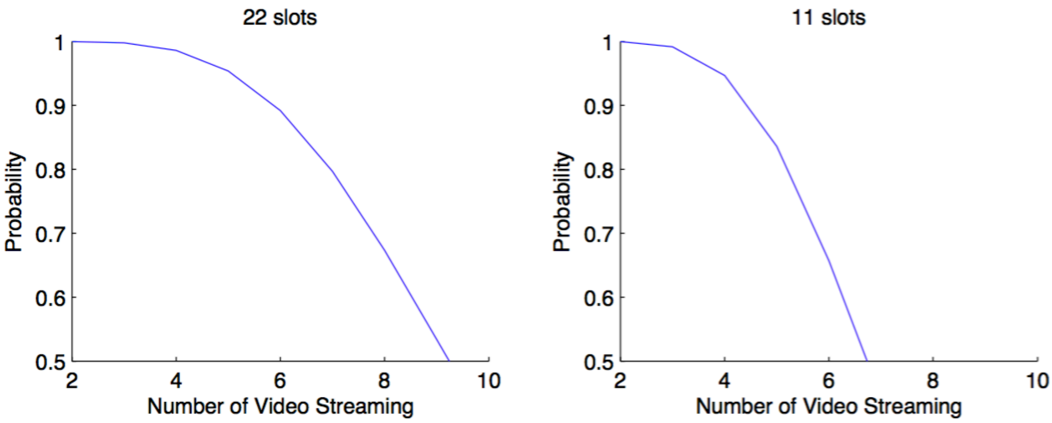

Fig. 5.3: Probability a maximum of two AC_1s access the channel at the same time in the same slot for video streams.

explored in more detail. N_{AC_0} and N_{AC_1} are fixed and equation 5.6 is plotted in figure 5.4. This shows how the probability an AC_0 chooses a backoff time of 0 varies with the CW size for both AC_0 and AC_1. In other words, it shows how the probability that one AC_0 transmits immediately after the $SIFS[0]$ is distributed.

In figure 5.4 the x axis represents the AC_0 contention window size, the y axis represents the AC_1 contention window size and the z axis represents the probability AC_0 transmits in the time slot 0 as found using equation 5.6. The resulting surface shows the probablity AC_0 transmits achieves a maximum value when CW_{AC_0} is small and CW_{AC_1} is large. It also shows that the probability AC_0 transmits achieves a minimum when CW_{AC_1} is small and CW_{AC_0} is large. When both CW_{AC_0} and CW_{AC_1} are small the probability AC_0 transmits does not achieve a minimum value, this is due to the increased probability of collisions.

Figure 5.5 shows the distribution of the probability that one AC_1 transmits in backoff timeslot 0 as calculated using equation 5.7. The graph is a mirror image of that in figure 5.4. Like figure 5.4 the x axis is the AC_0 contention window size, the y axis is the AC_1 contention window size and the z axis is the probability AC_1 transmits in time slot 0. The resulting surface shows that the probability AC_1 transmits achieves a maximum value when CW_{AC_1} is small and CW_{AC_0} is large. It can also be seen that the probability AC_1 transmits achieves its minimum

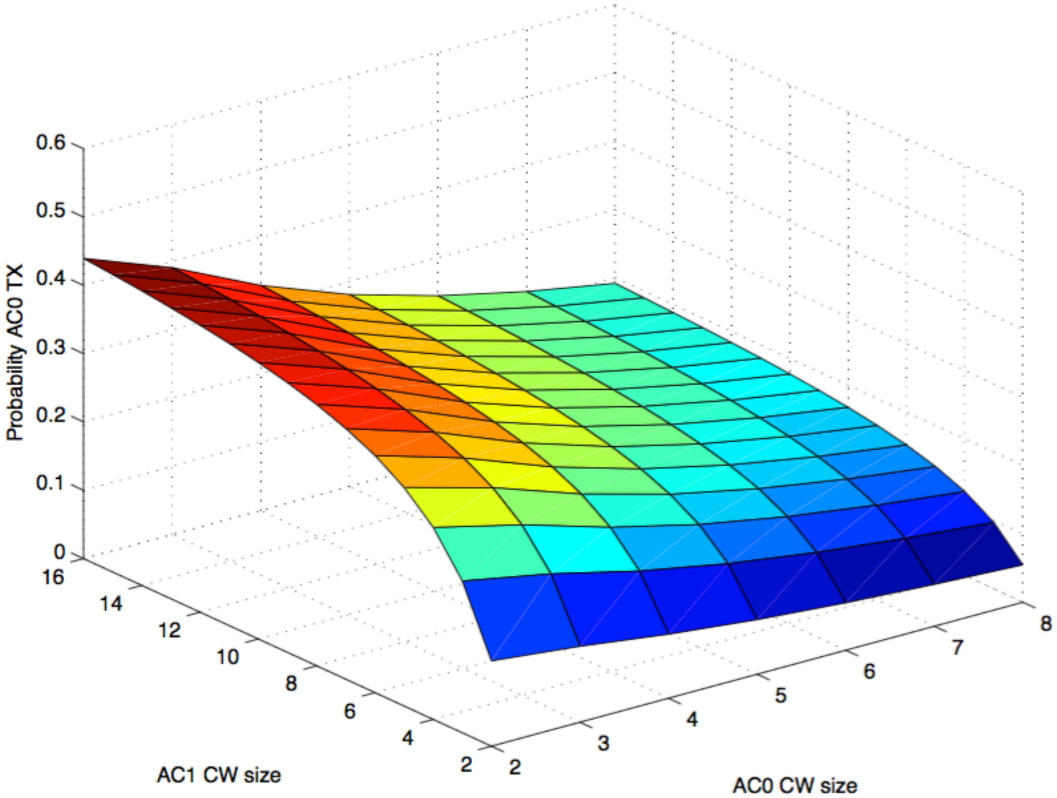

Fig. 5.4: Probability that one AC_0 is transmitting at slot 0 for a range of CW sizes for both AC_0 and AC_1

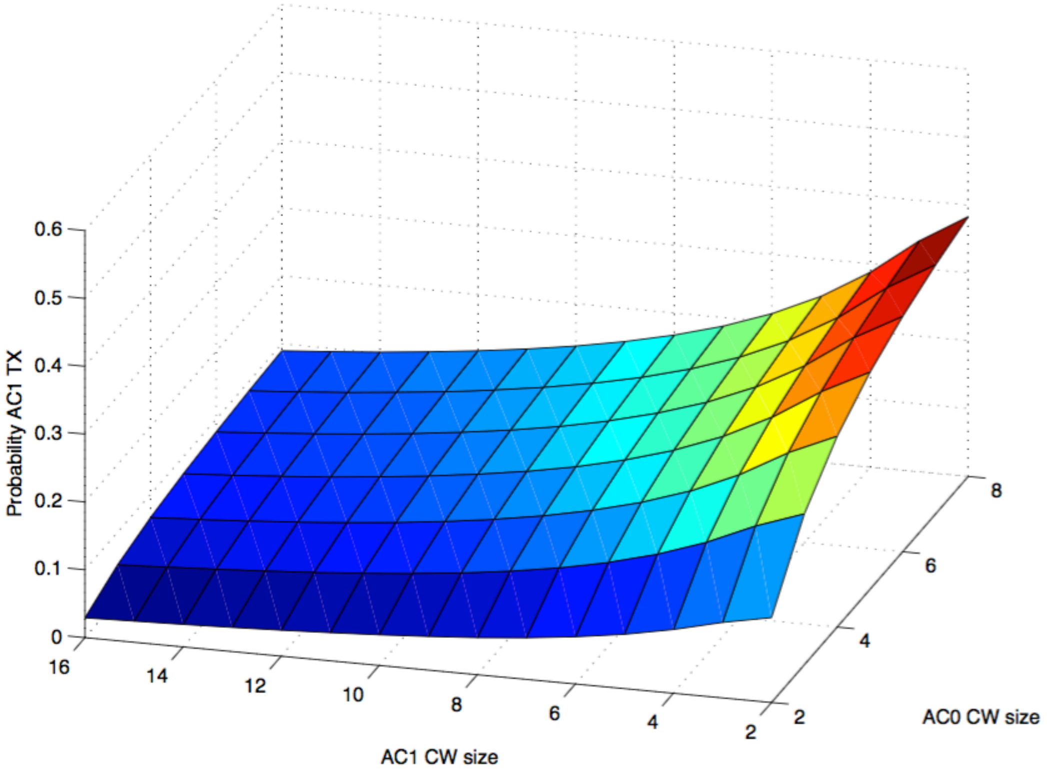

Fig. 5.5: Probability one AC_1 is transmitting at slot 0 for a range of CW sizes for both AC_0 and AC_1

value when CW_{AC_0} is small and CW_{AC_1} is large. When both CW_{AC_0} and CW_{AC_1} are small the probability AC_1 transmits is not a minimum due to the large probability of collisions.

The probability the channel is idle at the backoff timeslot 0, P_{e_0}, is shown in figure 5.6. It is calculated in MatLab [182] using equation 5.8. The x axis is the AC_0 contention window size, the y axis is the AC_1 contention window size and the z axis is P_{e_0}. The minimum P_e value is achieved when CW_0 and CW_1 are small; while P_e achieves a maximum value when CW_0 and CW_1 are large.

Figure 5.7 shows the probability a collision occurs in backoff timeslot 0, P_{c_0}. It is calculated in Matlab [182] using equation 5.9 . The x axis is the AC_0 contention window size, the y axis is the AC_1 contention window size and the z axis is P_{c_0}. P_{c_0} achieves a minimum value when

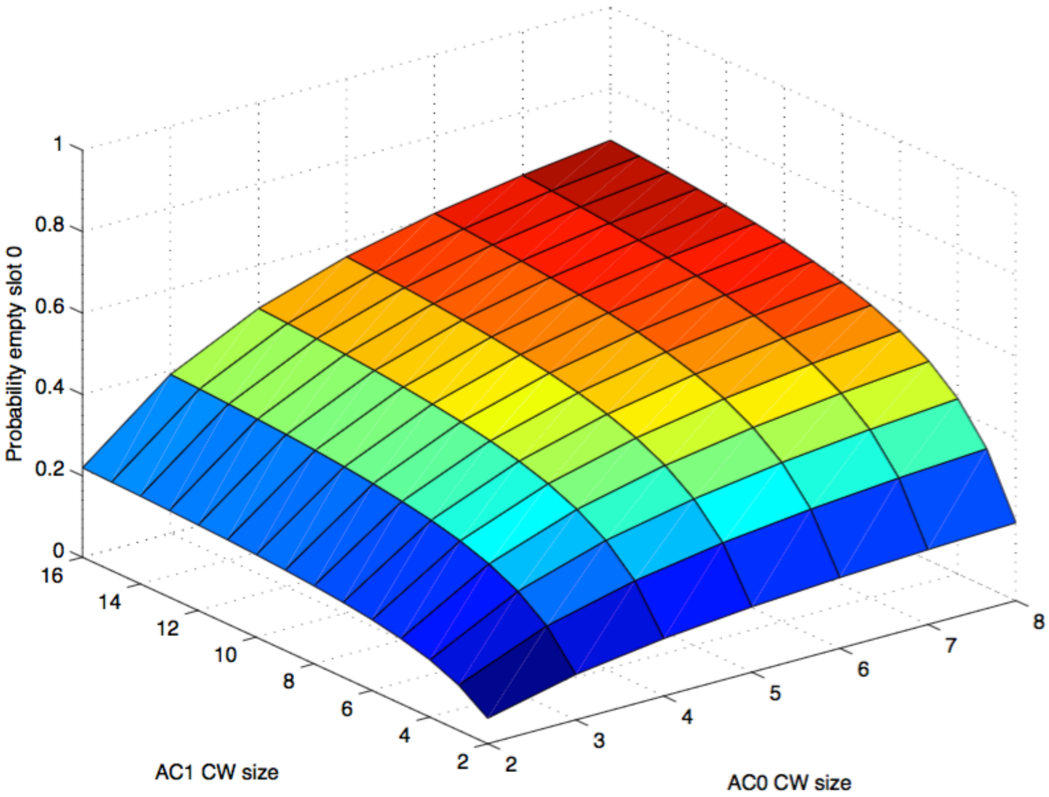

Fig. 5.6: Probability no transmissions occur at slot 0 for a range of CW sizes for both AC_0 and AC_1

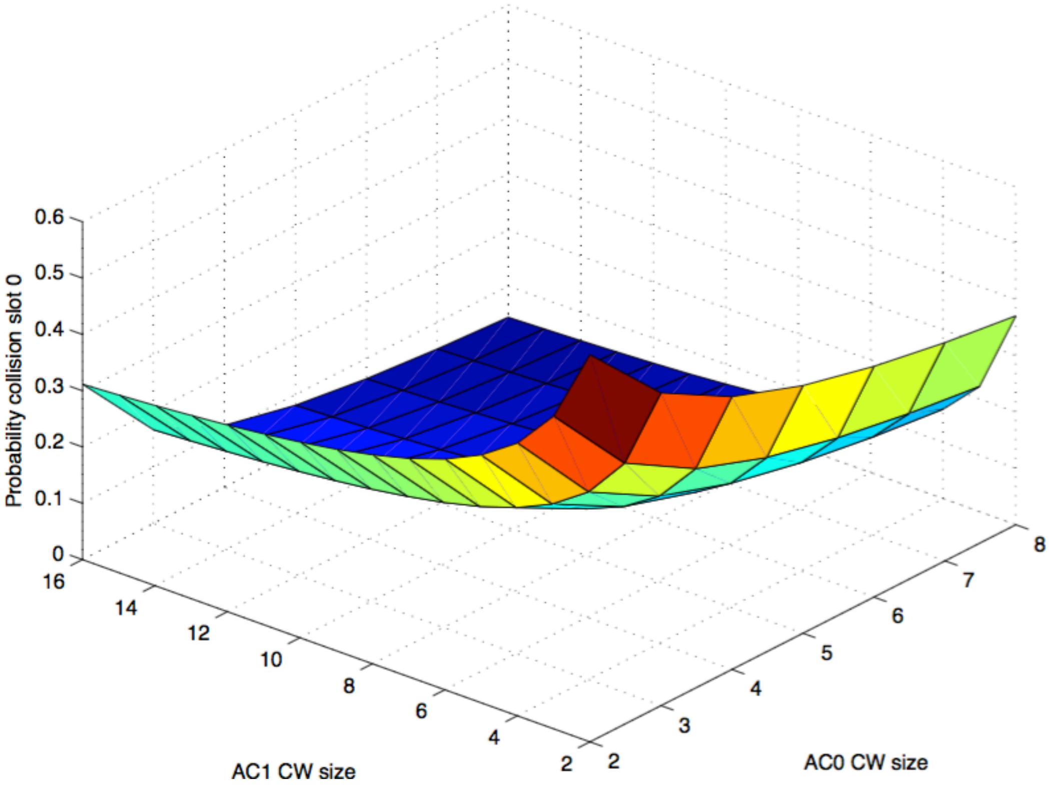

Fig. 5.7: Probability a collision occurs at slot 0 for a range of CW sizes for both AC_0 and AC_1

CW_0 and CW_1 are large and reaches a maximum when CW_0 and CW_1 are small.

As the three probabilities considered are mutually exclusive, the sum of the three surfaces in figures 5.4, 5.5 and 5.6 is a surface with a constant vertical axis probability value of 1.

Three observations can be made about the model. First, time is considered to be discrete therefore it is possible to use combinatorics to obtain probabilistic estimates about the behaviour of the wireless network.

The second is that the calculations are built on the hypothesis than only two AC_0s and two AC_1s are ready to transmit a packet at the same time and they both have the same CW_{min} size. In reality the two CW_0s or CW_1s competing to access the channel are not necessarily the same size as there is no communication between the nodes to set the same CW_{min} on both.

The final observation is that the discussion of the model focuses on $timeslot$ 0, and the goal of the algorithm is to transmit in time slot 0 with the predefined priority. As described in section 2.2, the CWs act differently; they do not start from the $AIFS[i]$s each time they stop the backoff time for a packet transmission.

The goal of the CW management part of QQM is to optimise the CW size based on the measured eQoS.

A collision event occupies the wireless channel for a time Tc [123]. Tc can be approximated as the $AIFS$ for AC_0 plus the time needed for an AC_0 packet transmission:

$$Tc = AIFS + Tx_{AC_0}. \tag{5.32}$$

The expected transmission time is, therefore:

$$Tc_0 = \mathbb{P}c_0 \times Tc \tag{5.33}$$

where $\mathbb{P}c_0$ is the probability that a collision event occurs.

The channel idle time, Te [123], has been estimated to be one $TimeSlot$. Te_0 can be approximated as:

$$Te_0 = \mathbb{P}e_0 \times Te \tag{5.34}$$

As discussed above, the probability the channel is idle grows with the size of CW_0 and CW_1; while the probability of a collision event decreases as CW_0 and CW_1 grow.

Figure 5.8 shows the intersection of $\mathbb{P}c_0$ and $\mathbb{P}e_0$ surfaces.

Using a similar methodology to [167], table 5.2 shows the CW_0 and CW_1 sizes needed to achieve the five different qualitative QoE levels for AC_0 and AC_1. These levels vary on a qualitative scale from Excellent quality to Bad quality. The value N/A is used to indicate that the service is not provided by the node. Table 5.2 reports the CWs used to randomly estimate the backoff time. This differs from the three dimensional surface shown in figure 5.8 as the surface shows the expected value for the CWs determined using equation 5.18.

The table is obtained from the optimal theoretical sizes for CW_0 and CW_1, as shown by the intersecting surfaces in figure 5.8 and from figures 5.4, 5.5, 5.6 and 5.7. Based on the measured eQoS it can be used to simultaneously optimise P_{e_0}, P_{c_0} and the probability to transmit a packet depending on the quality of the service provided. The sizes of CW_0 and CW_1 are set to the

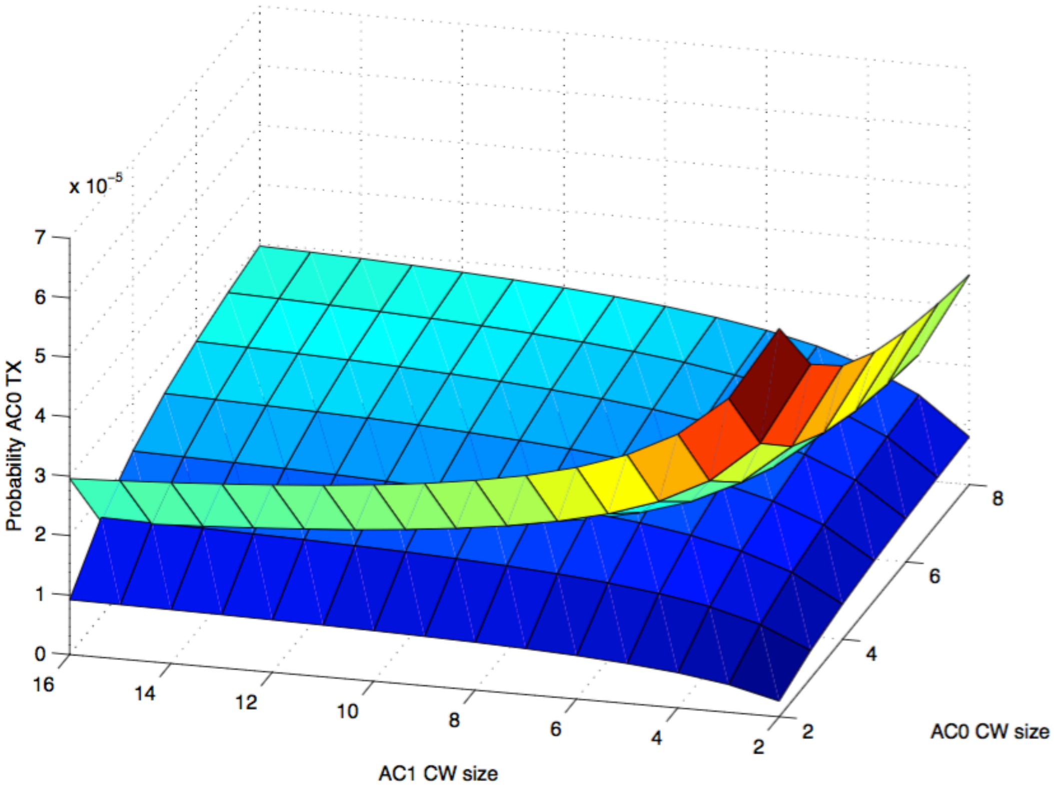

Fig. 5.8: Intersection of surfaces representing $\mathbb{P}c_0$ and $\mathbb{P}e_0$

.

CW_1 CW_0	Excell.	Good	Fair	Poor	Bad	N/A
Excellent	16 8	12 8	8 8	8 8	8 8	8 4
Good	16 6	12 6	8 6	8 6	8 6	8 4
Fair	16 4	12 4	8 4	8 4	8 4	8 4
Poor	16 4	12 4	8 4	8 4	8 4	8 4
Bad	16 4	12 4	8 4	8 4	8 4	8 4
N/A	8 4	8 4	8 4	8 4	8 4	8 4

Table 5.2: Video streaming details

maximum when the qualitative score is evaluated as Excellent, to the minimum size when the qualitative score is Fair, Poor or Bad and to the median value when the score is Good.

This configuration has two major advantages. First of all, when the quality is Excellent the larger CW value reduces the collision probability and optimises the throughput as a packet is always ready to be transmitted across the wireless network. Secondly, the use of a low CW value when the quality is Fair, Poor or Bad increases the probability of transmission in time slot 0 and avoids transmission delays. If packets are not transmitted then the flow quality deteriorates rapidly.

The differences in the CW_0 and CW_1 sizes reflects the different priorities assigned by the EDCA method. When a collision occurs the CWs are doubled. The optimal CW_0 and CW_1 values summarised in table 5.2 are used to design a fuzzy CW controller within the QQM algorithm. This will be discussed further in section 5.5.1.2.

5.5 The Quality Queue Management Algorithm

The Quality Queue Management (QQM) algorithm is detailed in this section [15]. It manages traffic at the Access Point (AP) to provides services over the network with the best possible Quality of Experience (QoE). It is designed to operate on future wireless networks and to manage the AC's parameters, and their interactions, according to the measured eQoS.

QQM brings together, and manages, the information inferred by the mathematical model, the eQoS metric and a queue management algorithm. QQM not only manages the traffic crossing the AP but it can also give feedback to the applications to reduce the traffic generated by some services. Feedback can be provide in various forms; for example, it can be the quality estimated at the AP by QQM or an estimate of the traffic as determined from the queue lengths. Feedback can also be provided as a packet generated by the AP and forwarded to the application server with an appropriate protocol.

The QQM algorithm incorporates three key features. The first feature is quality assurance. This is achieved by implementing eQoS flow preservation; flows are dropped if the minimum eQoS cannot be guaranteed. The minimum eQoS is a QoE value; by definition once the eQoS drops below this level the end-user makes the decision to drop the service. The second feature is the management of the transmission, collision and idle channel probabilities through modulation of the contention window (CW). The third feature of QQM is that it seeks to manage the queue length and avoid congestion.

The novelty of QQM is that these three features are combined in a single system. The interaction and information exchange between the features contributes to provision of the service with the best quality possible over the network. The implementation of these features is now considered in detail.

The first feature of QQM is eQoS flow preservation. This is achieved through continuous checking of the eQoS for each flow and comparison of the values obtained with historical eQoS data. Flows for which the eQoS falls below an acceptable threshold are dropped. This feature guarantees that only flows that achieve a minimum and acceptable eQoS are transmitted.

Packets are lost in the wireless network when one of three types of dropping event occurs. The first such dropping event is a collision. In this case a packet retransmission is carried out by the QQM algorithm and packets may be dropped if the delay exceeds the maximum acceptable delay for the service. Quality thresholds and levels have been inferred from literature [45]. Major manufacturers of network equipment specify suitable ranges or bounds for the physical network parameters where the quality of service provided is not affected provided the parameters remain within these bounds. These may be specified using critical threshold values; for example, as a threshold beyond which the quality of the service is affected or as a threshold beyond which the service is not provided. The second type of dropping event occurs when the queue length exceeds the congestion threshold or if the queue is such that the delay will exceed the maximum delay allowed for the service. The third type of dropping event occurs due to eQoS quality preservation. This occurs when a flow does not achieve the minimum eQoS required to provide a satisfactory service to the end user and so all of the flow's packets are dropped.

The second feature of QQM exploits the theoretical model presented in section 5.1. The CW sizes at the access point are managed by a control system that uses information on the network status to adjust priorities. It manages the average packet transmission time using an optimisation aimed at providing services with the highest quality. This optimisation is a function of the backoff time and the probability of a collision occurring. This optimisation also takes account of the dynamic variability in the TXOP time size.

The third feature of QQM is a queue management algorithm. The packet dropping decision process differs from that of traditional AQM algorithms: it uses eQoS to guarantee, in so far as possible, the provision of services that meet at least the minimum quality standards.

A key feature of QQM is the application of fuzzy logic and a fuzzy control system to manage traffic on future wireless networks. These networks divide traffic into separate access categories and can be viewed in the frequency domain as a succession of channel access requests from the mobile stations and the AP. Traffic management can be viewed as an optimisation in the frequency domain acting on both the queue and the contention window. Traffic management limits the input traffic speed and optimises the output traffic.

In the next subsection these features of the QQM algorithm will be described and detailed for each AC. The interactions between the features will also be discussed.

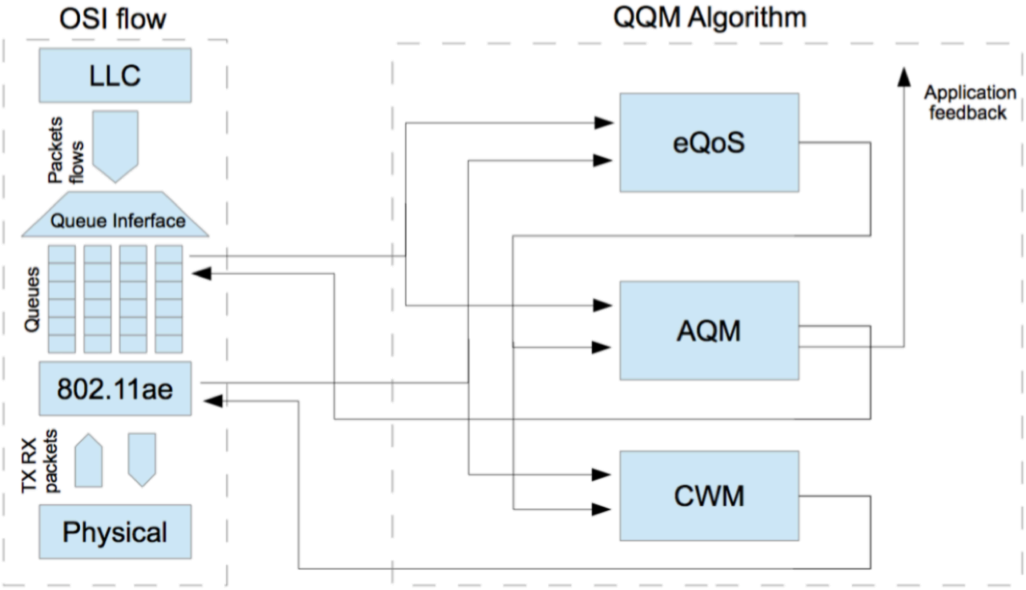

Fig. 5.9: QQM algorithm flow chart.

5.5.1 Algorithm Operation

Figure 5.9 shows a flow chart that captures the key operational elements of QQM. The algorithm is composed of three fuzzy controller blocks and it aims to manage the queues and the contention windows under the control of the eQoS block. These three blocks measure and control the traffic flows to guarantee an acceptable level of QoE for real time services, and a good level of QoS for all other services.

The eQoS block estimates the quality of service per flow. It is also an active block which interacts with the traffic through the AQM and contention window manager blocks, dropping packets that do not conform to the minimum quality level established for provision of the service.

The AQM block is at the heart of the QQM system. It provides queue management and queue length control. The AQM block can also provide application layer feedback to the traffic sources.

The contention window management block is based on the model described above and performs priority management of the contention windows. Its main goal is to optimise priorities across all ACs in order to reduce queue lengths.

All three QQM blocks are described in greater detail in the next three subsections.

5.5.1.1 The Active eQoS Measurement System

The eQoS block is designed to perform two important operations: the first one is to estimate the eQoS and the second is to drop packets when they do not conform to the desired eQoS. The block is a controller which provides services with the desired QoE by dropping non conformant traffic. Decisions on which packets to drop are made using a set of simple rules.

Figure 5.10 show the complete eQoS measurement and controller block diagram. The controller has four inputs. The eQoS error and the sign of its derivative are inputs for a closed loop fuzzy controller as described in section 3.4. The delay and the jitter are used to determine if an arriving packet is to be accepted or dropped. As a consequence their evaluation must be carried out at the packet level in order to calculate the eQoS. The controller uses these four inputs to determine the output decision.

The first operation performed by the controller is to measure the eQoS per flow using the EWMA method described in section 4.3. The measured eQoS has to be compared with the desired eQoS value in order to determine if the service is being provided with at least the desired QoE. The difference between the desired eQoS and the measured eQoS gives the eQoS error.

The range of possible eQoS values is rescaled to an integer value between 1 and 5 to simplify the calculation and allow for an immediate interpretation of the eQoS data. Five qualitative levels were chosen as five levels are used for MOS. As previously noted, there is not an exact correspondence between eQoS and MOS as the former is a nearly instantaneous measure while the latter is an averaged, long term value. When no packets are dropped the eQoS is excellent, but the qualitative score may not be 5.

Figure 5.11 show the fuzzification of the eQoS error input in the controller. The fuzzy values at the top of the graph are the five qualitative levels. The conversion between eQoS and the five qualitative level values, and vice versa, is performed using the fuzzification and defuzzification processes inside the controller. The eQoS error is 0 when the estimated eQoS is equal to the

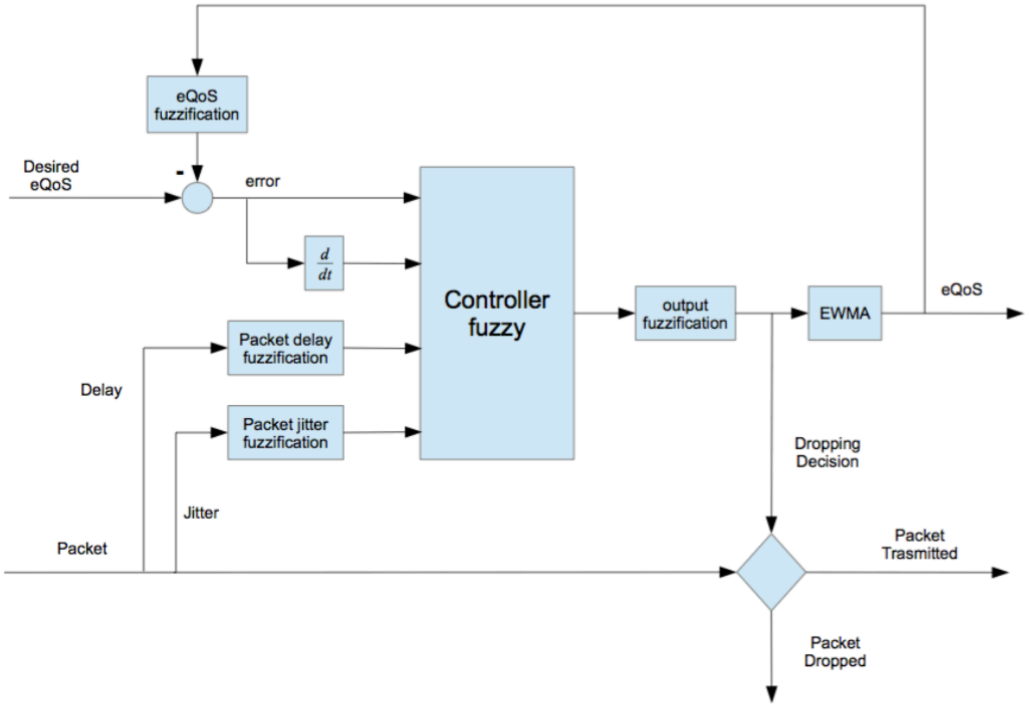

Fig. 5.10: eQoS Fuzzy Controller.

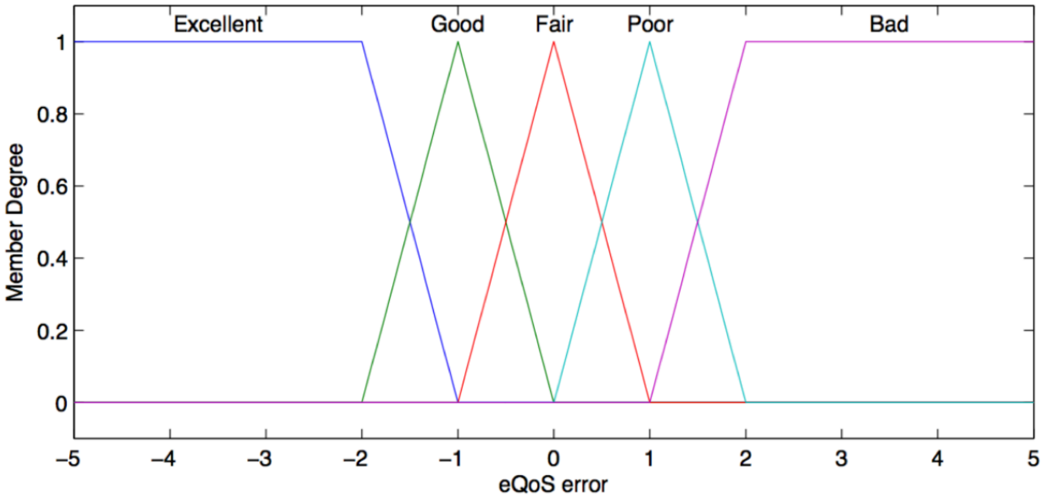

Fig. 5.11: eQoS Error Input

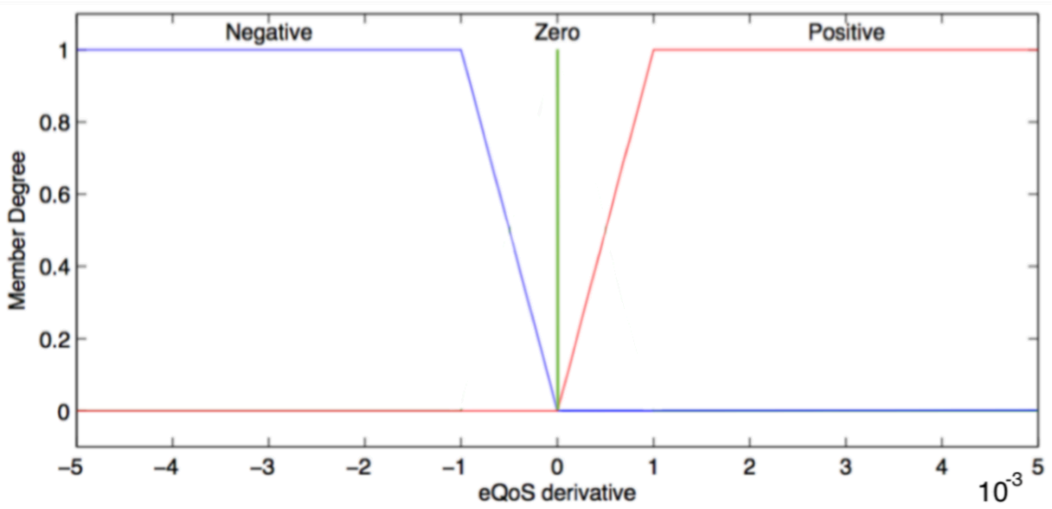

Fig. 5.12: eQoS Derivative Sign Input

desired eQoS, it is negative when the estimated eQoS is greater than the desired eQoS and positive when the estimated value is lower than the estimated eQoS.

The second controller input is the sign of the eQoS derivative. It calculates the variation between the eQoS actually measured and the previous eQoS value; it is the tangent to the curve of values assumed by the eQoS based on the previous two estimates. The derivative sign input indicates if the eQoS is instantaneously increasing, when it assumes positive values, decreasing, when it assumes negative values or stable, when the value is close to 0. Figure 5.12 shows the fuzzy set for the eQoS derivative.

Two other inputs need to be evaluated before a packet transmission can take place. A packet transmission is only permitted if the packet delay does not exceed a set limiting threshold. This is because packets that have accumulated a long delay are already considered as lost by the destination and so their transmission is a waste of wireless throughput. Packets with a jitter in excess of a declared threshold, typically 30ms as inferred from [45], are also considered to be unacceptable and should not be transmitted.

Like QoE, the delay and jitter limits depend on the application scenario under consideration. As discussed in section 2.4, the parameters used in this thesis are established values from the

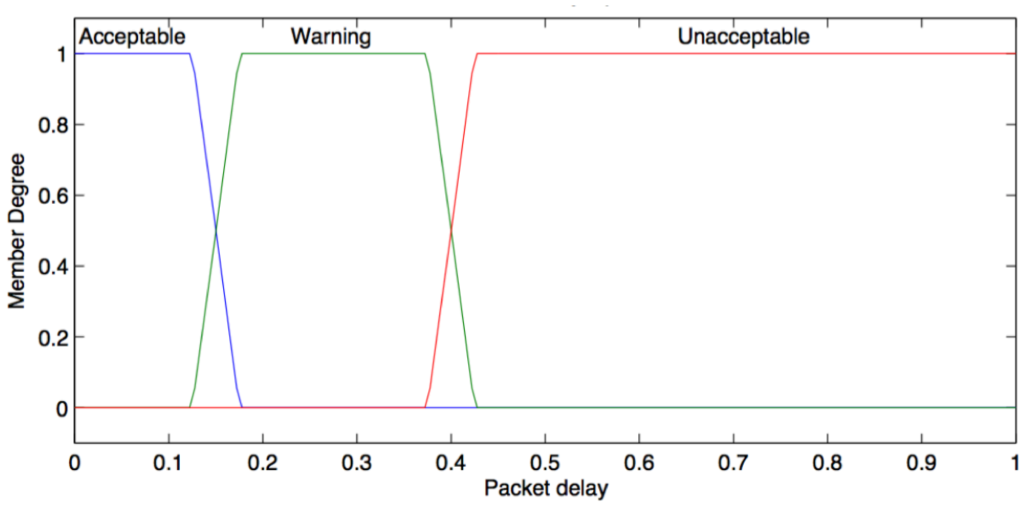

Fig. 5.13: Packet Delay Input

literature. In this case, the limits and threshold values used to guarantee good quality follow the VoIP guidelines in [45]. The fuzzy sets for delay and jitter are shown in figures 5.13 and 5.14 respectively.

The controller has two possible outputs shown in figure 5.15. If the output value is between 0 and 1 then no action has taken place, as indicated by NA in figure 5.15. While if the output is between 2 and 3 then the packet is dropped, as indicated by DP in figure 5.15. The output fuzzification rules are internally designed to rescale the output in two discrete values: 0 for NA and 1 for DP.

Fuzzy reasoning rules are inferred from the simple logic applied to the wireless system and are expressed by the following linguistic rules:

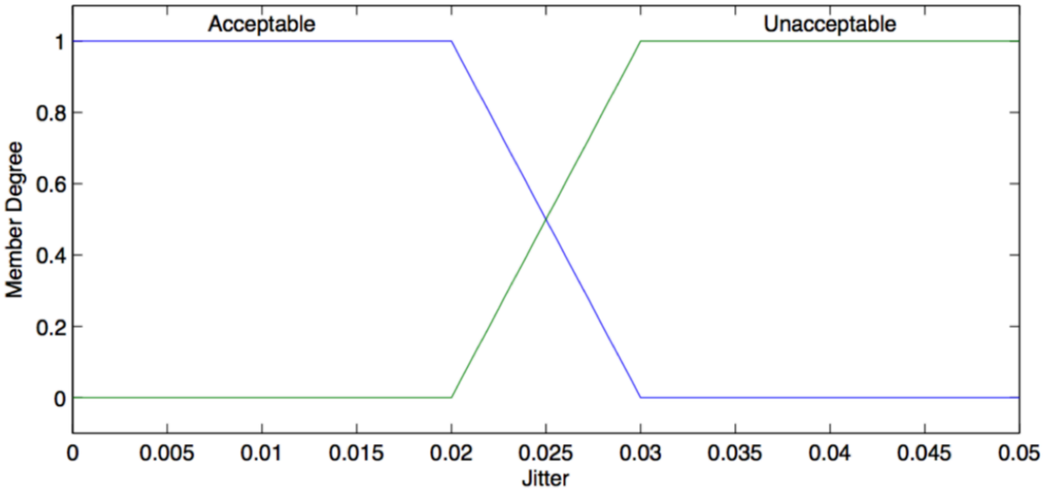

Fig. 5.14: Jitter Input

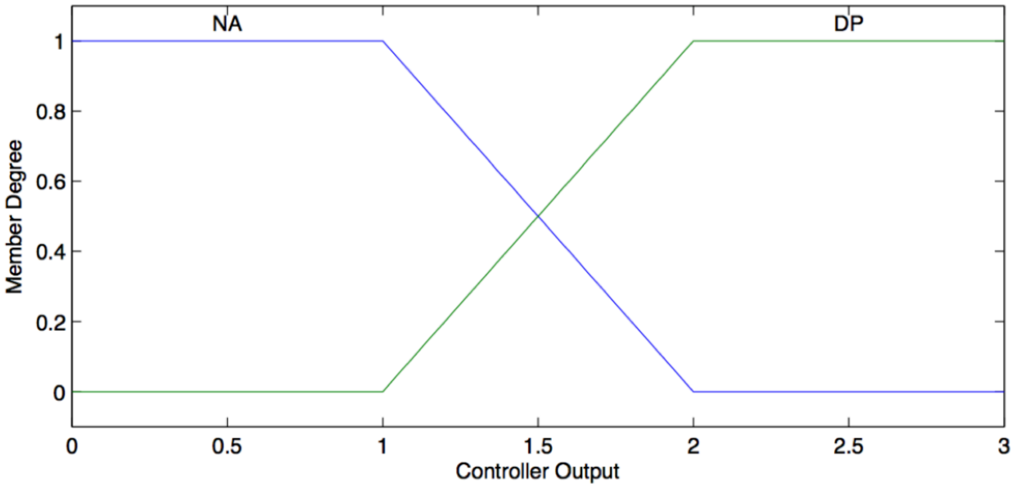

Fig. 5.15: eQoS Fuzzy Controller Output

```
if ((Packet delay < Unacceptable) || (Jitter < Unacceptable)) {
      if (eQoS >= Fair) {
            Transmit the packet
      } else {
            if (derivative eQoS >= Zero) {
                  Transmit the packet
            } else {
                  Drop the packet
            }
      }
} else {
      Drop the packet
}
```

The rules concerning delay and jitter form the first decision layer before the application of the other rules. The logical structure is, in fact, a controller with a controller. The defuzzification follows the maximum criterion [164]:

$$output^{crisp} \in \{argsup\{\mu_i(x)\}\} \tag{5.35}$$

where $\mu_i(x)$ is the member function of the crisp input i.

The eQoS calculation is performed in the EWMA block with the EWMA formula given in section 4.3. This makes use of the stored eQoS estimate from the previous cycle.

The controller is shown as a single box in figure 5.10; in reality the controller algorithm is split into a number of parts that operate at different points in the wireless network system. Packets enter the queue and are then passed to a transmission buffer before being transmitted.

The first point where the algorithm acts is at the queue input. Here the controller checks if the packet delay exceeds the maximum limit and that the queue is not full. The full queue check is not one of the controller rules but it is a system event that has to be considered. Jitter and delay checks are then performed. Finally, the eQoS measurement and its derivative are determined.

In the transmission buffer at the output of the queue the packet's delay and jitter are checked once more. This check is repeated for every packet retransmission attempt. eQoS considers a packet to have been successfully transmitted when its acknowledgement has been received. The eQoS is determined after these checks and the value stored in an array for use in subsequent calculations. The measured eQoS is used by the other parts of the QQM algorithm and this is described below.

5.5.1.2 The Contention Window Controller

This section details the CW size management system used to optimise transmissions by the wireless stations. Its design logic is inferred from the descriptive model of EDCA given in section 5.1.

The first step in the design of the Contention Window Management controller (CWMC) is to estimate the number of ACs contending for the channel at the same time. As described above, AC_0 and AC_1 access the channel with a frequency that depends on the number of flows in the class and the throughput needed to provide the given services. AC_2 and AC_3 do not access the channel with a fixed, regular frequency as these access classes often handle bursty traffic. Only CBR flows can access the channel with a fixed frequency.

The model estimates the transmission, idle channel and collision probabilities for the four ACs. The optimal CW size for each AC is estimated using the eQoS measured within the system. In addition, variations in the eQoS are used to manage the CW. As the eQoS decreases, the CW size is increased. This has the effect of increasing the transmission probability and reducing the collision probability. While there are other controllers that act in a similar manner on the CW, for example [167], these do not include any quality measure related to the end user experience.

The hypothesis formulated in section 5.4 was that mobile nodes are used for a single service such as a phone call, audio streaming or audio and video streaming. By contrast the AP manages multiple services and flows that originate in the backhaul.

In a wireless infrastructure network the mobile nodes and the AP are aware of the nature of this traffic from the backhaul network. The CW size is estimated for each flow based on the

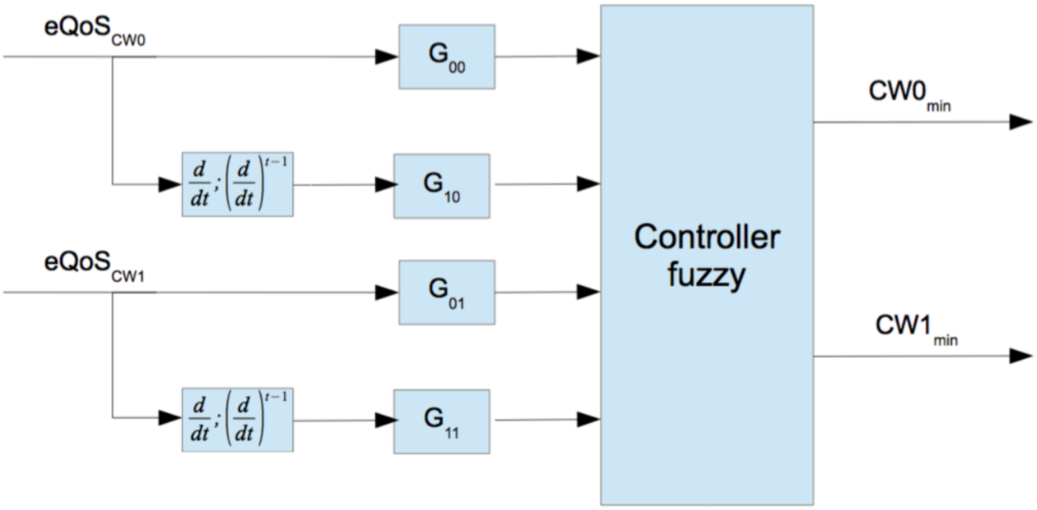

Fig. 5.16: Contention Window fuzzy controller.

eQoS. The algorithm is structured to independently manage the CW size on each mobile node and at the AP, therefore each node must adapt its CW to the traffic in its queue.

The inference rules for the contention window controller are given by the following expression:

```
if AC0 eQoS is X and AC1 eQoS is Y than CW0 is Z and CW1 is W;
```

The CW controller block diagram is shown in figure 5.16. It is a fuzzy controller applied at each node. It has two inputs for CW_0 and CW_1, and separate outputs for CW_0 and CW_1. It is specifically designed for the real time traffic in AC_0, and AC_1. The contention window controller does not manage AC_2 and AC_3 because delay is not a priority for best effort and background traffic. The AP manages more than one flow for each AC. The eQoS of the packet in the transmission buffer is used to estimate the CW, because this flow is transmitted together with the other flows during a TXOP time.

The inputs used for the CW_0 and CW_1 flows are the error, i.e. the difference between the measured and the desired values, and the sign of the derivative. These are similar to the error

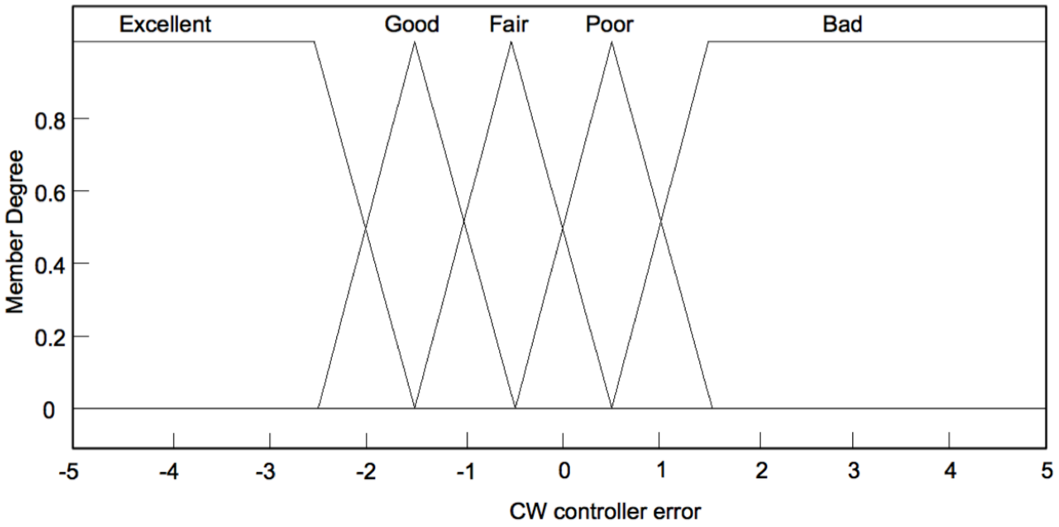

Fig. 5.17: Controller input error for CW0 and CW1

and derivative inputs for the active eQoS measurement system. They are shown in figures 5.17 and 5.18.

The lower input quality limit is fixed at the Fair level where the eQoS error is set to 0; this corresponds to a MOS value of 3, i.e. an eQoS value of approximately 0.25. Each CW controller also uses the second derivative of the eQoS error as an input; the first input is the actual derivative over time, and the second is the comparison of this derivative with that obtained for the previous instant in time.

These are used to determine if two consecutive eQoS derivative sign values are negative. In other words, it checks to see if two packets are dropped in sequence. The second input can be positive or negative.

No scaling is required for the eQoS error and so the gains for each kind of flow, G_{00} and G_{01} or G_{11} and G_{10}, are set equal to 1. By contrast, the eQoS error is decreased by a half of one unit to take account of the fact that the MOS is never 5.

The outputs are shown in figure 5.19. The output is obtained using the maximum criterion for defuzzification [164], as used in the eQoS measurement system. The values for CW max,

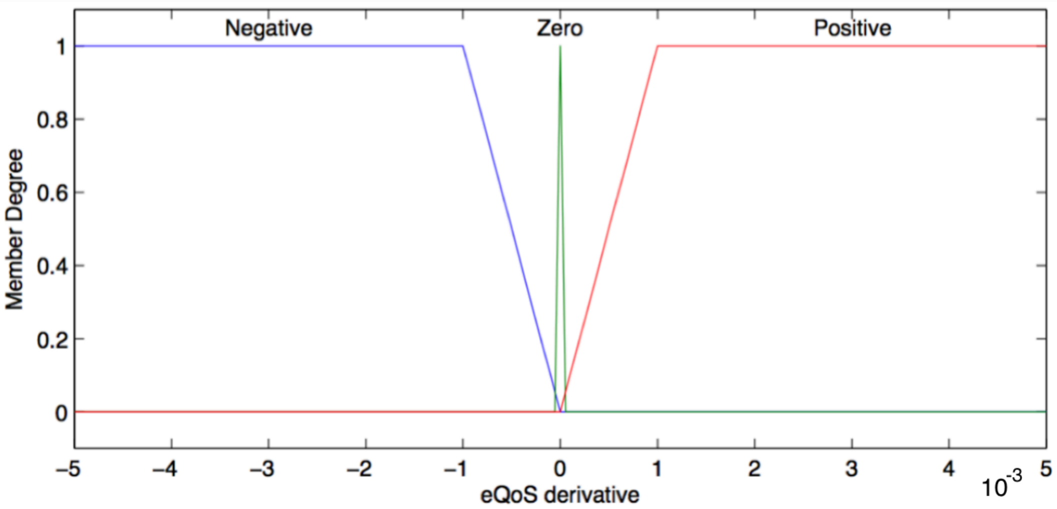

Fig. 5.18: Controller input derivative sign for CW0 and CW1

CW mid and CW min are, respectively, the maximum, the median and the minimum values of CW_0 and CW_1 based on the input error.

5.5.1.3 The Queue Management System

The queue management system is the third component of QQM. The specific task of this system is not only to avoid congestion by keeping the queue size within an acceptable bound, but also to maintain an acceptable QoE level for real time services and to optimise packet transmissions for all other types of traffic. Packet optimisation is performed indirectly through the packet dropping process and related to changes in eQoS through the optimisation of the contention window controller.

The EDCA method implemented maintains four queues. The traffic managed by each queue can have a constant or variable bit rate; it can be bursty, unresponsive or flow-controlled [10]. The Queue Management subsystem algorithm must be sufficiently versatile to be able to deal with all these traffic types.

Figure 5.20 shows the Queue Management system flowchart [191]. Packets arriving at the node are directed to one of the system's queues depending on their traffic type. The packets

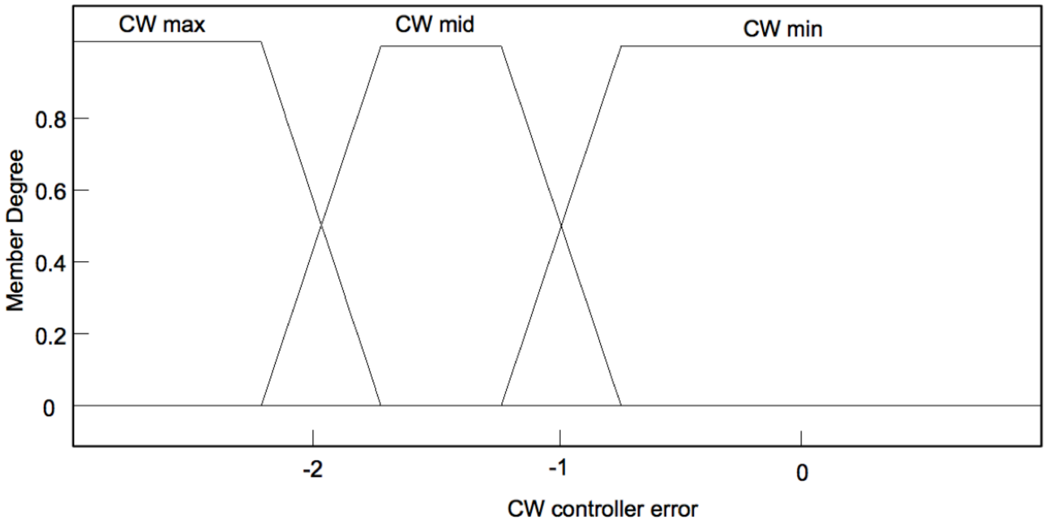

Fig. 5.19: Controller output for CW0 and CW1

are served by the node according to the priority assigned to each queue. As per[1] [192], packet delay has to be contained within acceptable limits. Delay checks are already performed by the active eQoS measurement system. Unlike [192], delay checks are not only performed at the queue output, they are also carried out before the packet is transmitted. This is particularly important for low priority ACs as the packets may have to wait several backoff times before being transmitted.

Delay control is applied in a different way to [192]. In addition to the checks performed by the active eQoS measurement system, the queue management system checks the delay between each packet transmission and averages it using an Exponentially Weighted Moving Average (EWMA) [9]. The average time needed to transmit a packet multiplied by the number of packets in the queue gives the theoretical total delay due to time spent in the queue. This gives a queue threshold for determining if packets are acceptable. This threshold is variable and represents the limit for a virtual queue [136]. The length of the virtual queue described in [136] is related to the channel capacity. In this case the queue length relates to the time needed to transmit each packet.

[1] `http://www.bufferload.net`

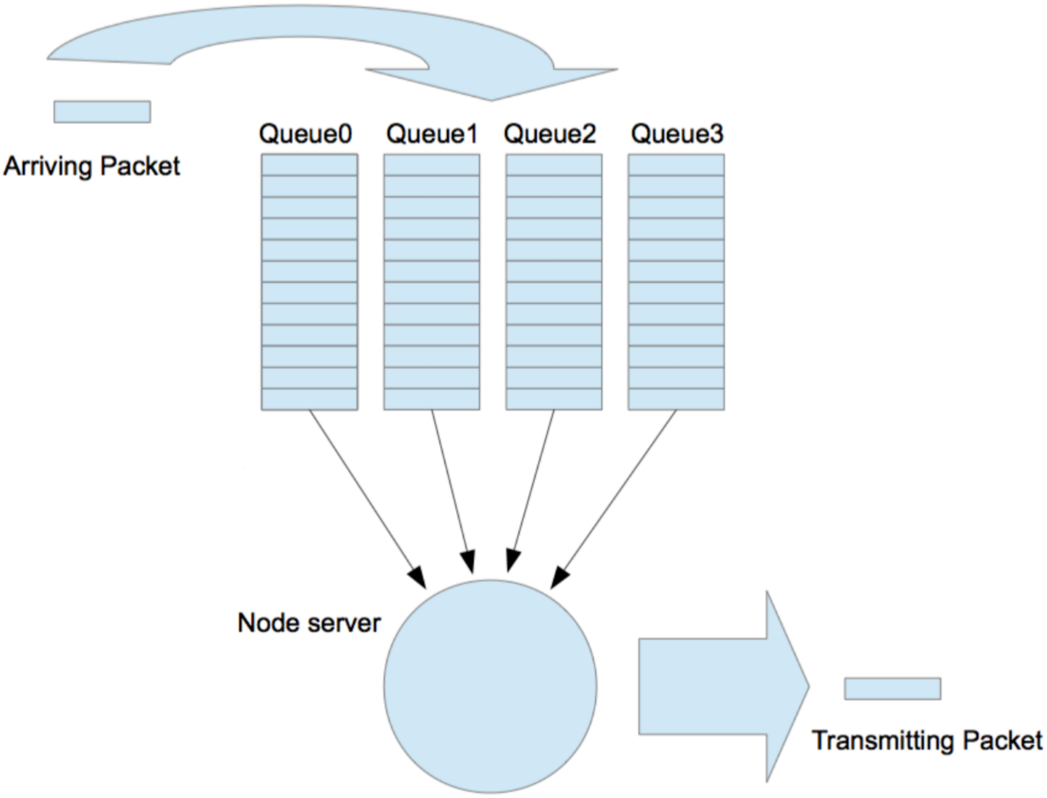

Fig. 5.20: The Queue Management System Flowchart

This process is slightly more complex than [192] but it has many advantages when the focus is on the delivery of a quality service for flows where the number of packets per second varies or for packets that have already accumulated variable delay within the wired network. The delay is used to estimate the maximum number of packets that can be managed by the queue without exceeding the maximum acceptable delay for provision of a good quality service. Packets that exceed the queue limit can be immediately dropped or observed for a certain amount of time before a decision is made as to whether to drop the flow or not. The decision will depend on the kind of service being provided and the AC.

The calculation of the average delay accumulated by the packets arriving at the queue is performed using EWMA [9]:

$$avg_{delay} = (1 - w_{delay}) \times avg_{delay_{t-1}} + w_{delay} \times Delay \tag{5.36}$$

where:

avg_{delay} is average of packet delay,
$avg_{delay_{t-1}}$ is the average of packet delay at time $t-1$,
w_{delay} is the weight and
$Delay$ is the delay of the arriving packet.

EWMA provides the average delay, it is updated every time a packet is moved from the head of the queue to the transmission buffer and it is not reset. Peaks and rapid variations are smoothed by the weights used in the EWMA formula. The average delay is immediately used to update the queue length limit.

Equation 5.36 is used for real time services on AC_0 and AC_1. AC_2 and AC_3 manage TCP and UDP traffic, so their flows may be unresponsive [10]. This allows for longer delays without any impact on the service. Nevertheless, delay and jitter remain valid measures by which to gauge the level of the network congestion.

The application of a queue limit based on the delay in a future wireless environment necessitates a modification of the features of EDCA. The main impact on the structure of the mathematical formulas is due to TXOP, where blocks of packets are transmitted at the same time. The TXOP is long enough to transmit a large number of packets, and it is frequently fully used each time the AC accesses the channel. Thus it is acceptable to hypothesise that each time either AC_0 or AC_1 accesses the channel a packet from each VoIP call and each audio frame are sent and a frame from each video stream is transmitted.

The calculation of the transmission delay accumulated at the transmission buffer is performed using EWMA [9]:

$$avg_{tx} = (1 - w_{tx}) \times avg_{tx_{t-1}} + w_{tx} \times TxTime \tag{5.37}$$

where:

avg_{tx} is average of packet transmission time,

$avg_{tx_{t-1}}$ is the average of packet transmission time at time $t - 1$,

w_{tx} is the weight and

$TxTime$ is the Transmission time of the packet.

The AP is the only node which manages packets in both directions. The mobile nodes manage only a single real time flow at a time for the service provided or, occasionally, multiple background traffic flows. Translating all these considerations for the AP and the mobile nodes into a mathematical formula, the queue limit $Q_{t_{max}}$ can then be calculated by dividing the average maximum delay allowed for a packet by the average time needed to transmit each packet. Mathematically, $Q_{t_{max}}$ is expressed as:

$$Q_{t_{max}} = \frac{P_{delay_{max}} - avg_{delay}}{avg_{tx}} \tag{5.38}$$

where:

$P_{delay_{max}}$ is the maximum delay allowed for a packet,

avg_{delay} is the delay accumulated by the packet arriving at the queue, and

avg_{tx} is the average transmission time.

The value of $P_{delay_{max}}$ is defined per service and $Q_{t_{max}}$ lies between minimum and maximum threshold values. In the simulations presented in Chapter 6 80$\%$ of $Q_{t_{max}}$ is used as the virtual queue threshold. In practice, $Q_{t_{max}}$ is very large if packets are transmitted regularly by the high priority ACs; in this case the network is not congested and can be considered to be working well. By contrast, a reduction in $Q_{t_{max}}$ indicates that the network is experiencing congestion as the average packet delay is increasing. In this case, the queue buffer needs to be extremely large as the delay at each queue is controlled by the queue management system.

The next three subsections describe how the queue management system is applied for each of the four access class.

5.5.1.4 AC_0 Queue Management System

Each mobile station involved in the provisioning of a VoIP service has only a small number of packets in its AC_0 queue, Q_0. The AP is the point where the wired and wireless network meet. It manages all the input and output traffic to and from the wireless network. The AP has in its Q_0 all the VoIP packets that arrive for transmission on the wireless network. Due to the synchronisation problem discussed in section 5.5.1.2 one packet from each flow is transmitted every TXOP. The AC_0 at the AP makes full use of the TXOP feature offered by the ECDA method.

VoIP produces CBR traffic and as there are only a few VoIP packets in each mobile node, the queue size can be easily kept under control by applying equation 5.38. Packets that exceed the dynamic threshold at a mobile node provide an indication of network congestion.

Equation 5.38 has a more significant application in queue size control and for quality management at the AP. The packet group transmission feature of TXOP does not require any modifications to the formula. The delay threshold, calculated using equation 5.38, includes an average over time of the packets transmitted at different TXOP times in the EWMA averaging formula.

Routing control and message packets are managed by AC_0 and they cross Q_0 together with the VoIP packets. They are not considered in the mathematical formulation presented because their influence on the system is negligible. The active eQoS measurement system functionalities require Q_0 to include features to enable it to drop VoIP flows with unacceptable QoE. At the queue level this means that it is possible to drop all the packets from an unacceptable flow or from the newest flow in the queue in order to increase throughput and provide other flows in the queue with an acceptable QoE.

5.5.1.5 AC_1 Queue Management system

. Video streaming is managed by AC_1. This is VBR, critical real time traffic. The number of flows and the throughput have to be carefully considered in order to manage the queue and guarantee acceptable QoE levels.

In this dissertation the streaming traffic in AC_1 includes services such as video calls, where there is bidirectional traffic between the AP and the mobile stations. The AP is then the node which manages the majority of the streaming traffic that is crossing the network. Therefore at the access point Q_1 will easily become full when there are only a few high throughput streams in the network.

Q_1 needs to have a threshold value that varies depending on packet delays. To avoid queue congestion, it should also be constrained by a maximum threshold value. At the AP frames arrive at the same time due to the same synchronisation problem found for traffic in AC_0. As a consequence the queue length exhibits large oscillations and these can be observed in the results provided in Chapter 6.

The queue management system drops all packets that exceed the maximum threshold value. This is done progressively and in proportion to the throughput for each flow. This balances the eQoS between the flows. Flows with a high throughput have more packets dropped than flows with low throughput, but in a proportion that means they experience the same eQoS. Equation 5.38 defines a dynamic threshold that varies with time. It is assumed that each AC_1's TXOP is sufficient long to transmit a frame from each flow.

When the queue continuously exceeds the upper threshold and the eQoS of each flow is close to the accepted limit, the flow with the worst eQoS according to the provider's quality specifications will be dropped. If required by the eQoS status of the flows, the dropping operation may then be repeated. Instead of dropping a flow it can be provided with feedback for the underlying application. This feedback requests a reduction in the throughput of the stream. The flow can then be readmitted to the system. New streaming flows are not admitted if the queue size is close to the threshold.

It is possible to further optimise the flow dropping process but the operations required are very expensive in terms of software complexity and beyond the scope of this work.

5.5.1.6 AC_2 and AC_3 Queue Management systems

The services crossing AC_2 must be provided with the best effort possible after the real time services in AC_0 and AC_1 have been provisioned for. In this dissertation the services in AC_2 are considered to be delivered using TCP and UDP traffic. The first TCP traffic type is short lived flows. These are used for bidirectional communications and are typical of connections like telnet or database requests. The second TCP traffic type is regular TCP traffic; for example, this may represent a request for a large data or file transfer at the best effort level.

The packets managed by AC_3 and crossing Q_3 are background traffic. This is composed of a mix of TCP and UDP flows as used, for example, for P2P file transfer, FTP file transfer etc. Unresponsive flows [10] crossing Q_3 cannot be controlled and it is difficult to maintain an acceptable queue length if the volume of input traffic to the queue exceeds the volume of output traffic.

The QQM system does not operate over AC_2 and AC_3. These two access categories require instant QoE estimation methods that differ from that provided by eQoS. This is because of the nature of the services AC_2 and AC_3 provide. This is beyond the scope of this dissertation; however, some key features emerge from the theory discussed in this chapter and these are considered below.

Equation 5.38 can be used for TCP and UDP traffic as it represents traffic that has exceeded a dynamic queue threshold or that has been delayed in the network. A distributed dropping policy can be applied to the flows to guarantee fairness. Dropping events regulate TCP flow throughput: when a TCP packet is dropped the source halves the congestion window, reducing the throughput after a RTT delay. Fairness between the flows is based on the same principles proposed in the CHOKe (SAC) [142] and TFRED [143] AQM algorithms as it depends on delays in the queue and between packet transmissions.

It is also possible to increase throughput by dropping TCP acknowledgement packets and banning flows that have dropped an excessive number of packets in a given interval of time. TCP acknowledgement packets can be dropped without affecting the TCP traffic when the Cumulative ACK [171] feature is implemented in the TCP protocol: the last TCP acknowledgement packet received by the source triggers an automatically assumption that all the previous TCP acknowledgement packets were correctly received.

5.6 Algorithm Limitations and Applicability

The QQM algorithm has been designed and developed for use in future wireless networks where the EDCA method is used. However, its range of application is not limited to these networks and it should be applicable on, and be beneficial to, the operation of a wider range of wireless networks.

The QQM algorithm is designed for practical application. The system's decision to drop a flow cannot be taken based on an instantaneous sampling of the flow's eQoS; rather a more longitudinal evaluation of the eQoS is needed. This evaluation is performed using three aspects of the calculated eQoS. The first is the eQoS level and if this is below the minimal acceptable value that has been fixed for that service. The second is to determine the length of time that the eQoS level remains below this threshold. The third factor is to establish the frequency with which the eQoS remains below the minimal acceptable. These three features allow the system to decide if, and when, a flow has to be excluded from the network. If a flow is excluded then its packets are immediately dropped, followed by the dropping of the service between the source and the destination. When a flow is dropped the system needs to wait for a few seconds to adjust to the new regime before it re-evaluates the quality parameters for each flow and any further decisions to drop a flow are reached.

QQM's design makes two assumptions about the nature of the wireless environment being used: that it uses multiple queues and that it transmits its traffic using a broadband channel. QQM assumes that the system is managing macro traffic types using multiple queues. If QQM is to be applied in a different wireless network then the Queue Management system will need to be adapted to suit the queueing regime and traffic characteristics of the new environment. QQM is designed to perform best in a broadband channel where throughput is high and traffic volumes are large. As a consequence features like CSMA/CA obstruct the proper functioning of the algorithm. This is because the CSMA/CA control packets reduce the throughput available for data packets. The QQM contention window controller performs best when EDCA is used without CSMA/CA.

The use of discrete system variables is a limitation of the QQM system. On one hand, these are advantageous when fuzzy logic and controllers are used to design the algorithm; on the other hand, the restriction to a discrete range of values reduces the algorithm's ability to act and react to traffic changes.

The algorithm combines multiple fuzzy logic controllers and their design will influence the complexity of the system. Each controller is implemented with one or more conditional loops. The active eQoS measurement system uses linguistic rules where the conditional loops do not need to be nested. As a consequence, the complexity of this controller is in the worst case $O(n)$ [193]. The contention window controller has a linguistic rule with a single conditional loop. In the worst case its complexity is $O(n)$. The final subsystem, to consider is the queue management system. It estimates the maximum number of packets with acceptable delay that the queue can contain, it uses the eQoS values to determine if a flow should be dropped. The first operation is a simple calculation and does not effect the complexity of the system. The checks and the decision making are performed in conjunction with the active eQoS measurement system. The operations consist of simple conditional loops and the complexity of this subsystem is, in the worst case, $O(n)$.

Even though the algorithm contains a large number of checks and conditional loops, it is still very simple and straightforward. The fuzzy controller and logic make the algorithm much simpler than those developed using traditional controllers. The number of operations needed per unit time are negligible for the hardware used in a simple wireless AP.

5.7 Summary

This chapter describes the Quality Queue Management (QQM) algorithm. QQM is composed of three subsystems. Each subsystem carries out measurement, adaptation and decision making. It is assumed that QQM is intended for use on future wireless networks where the ECDA method is used.

At the beginning of the chapter a novel mathematical model to capture the probability that a transmission occurs, a collision occurs or that the channel is idle was established.

The first QQM subsystem is the active eQoS measurement system. Its primary activity is to measure the eQoS of each flow. It interacts with the traffic, checking on packet delays and packet drops, in order to estimate the quality of the service being provided.

The second subsystem is the contention window controller, designed to use the measured eQoS in order to control the CW size and adjust priorities. It receives a measure of the eQoS for each flow and consequently acts to manage the CW size to adapt priorities. It aims to optimise the three key probabilities; that is, the probability that a transmission occurs, that a collision occurs or that the channel is idle.

The third subsystem is the quality queue management system. This is at the heart of the QQM algorithm and gives it its name. The queue management system performs decision making based on the traffic flows arriving at each queue. It uses the measured eQoS value to make decisions and, indirectly, alters the eQoS by dropping entire flows. Its decisions impact on the other two subsystems; that is, on the contention window controller and active eQoS measurement subsystem.

The chapter ends with some considerations regarding the limitations of the algorithm. These mainly arise from the theoretical aspects of the design and relate to the algorithm's applicability in a wireless network environment.

In the next, penultimate chapter the eQoS metric and the QQM algorithm are explored and evaluated by simulation.

Chapter 6

Algorithm Evaluation

In the previous chapter a novel queue management system, QQM, was introduced. It builds upon and exploits the features and functionality of the eQoS quality metric and the theoretical model of a multi queue system. QQM is more than just a simple queue management algorithm: it is an efficient mechanism for managing traffic in an AP and it is the key to quality management in future wireless networks.

In this chapter the efficiency of eQoS, the theoretical model and the QQM algorithm are explored via simulation. This evaluation of QQM not only validates the mathematical formulae derived and their underlying assumptions, but also serves as a demonstrator of the QQM algorithm and the significant role it could play in the implementation of future wireless networks.

The methods used to evaluate the efficiency of QQM concentrate on true-to-life, real time traffic scenarios in an infrastructure wireless network. Simulations where QQM implemented at the MAC layer are compared with simulations where the EDCA [35] method is implemented at the MAC layer. Simulation are perfomed using ns-2 [194] and, when necessary, using Matlab [182].

This chapter starts with a detailed description of the platform, traffic and quality metrics used in the simulations. It is then divided into three parts: the first of these validates the eQoS metric by comparing its performance with the traditional computational algorithm used to estimate the

MOS. The second part of the chapter validates the theoretical model by providing evidence to support the underlying hypotheses and the mathematical equations derived in Chapter 5. Finally the QQM subsystems and overall performance are evaluated using scenarios that have been carefully chosen to replicate the anticipated traffic volume and mix in future wireless networks.

6.1 Simulation Platform

This section details the software platform used to explore the efficiency of the QQM algorithm. The objective is to show how QQM can be used to enhance the quality of real time service provision in a network.

The methodology used to explore QQM's efficiency is focussed on an exploration of the settings required for the physical variables in a hypothetical everyday scenario. It is based on a realistic scenario, with a traffic mix that mirrors a real-life situation and existing QoS metrics. These will be described in more detail below. The impact that the correct setting of these physical variables has on meeting existing QoE guarantees and on the delivery of enhanced QoE are discussed.

The simulations were performed using the well-known network simulator ns-2 [194]. This is an open simulation platform written in C++ and OTcl, it has been developed through the collaborative effort of many individuals and it allows for simulation of the most popular wired and wireless networking protocols.

ns-2 is an old simulator but it is functional, efficient and widely used. While newer simulators exist, ns-2 can also be used to simulate new technologies and protocols. It is also easy to extend its functionality by developing and customising packages for it. In this thesis, Matlab [182] is used to analyse traffic data, carry out any mathematical calculations, perform the necessary statistical and fuzzy analysis, and to graph data.

Over the past fourteen years, ns-2 has evolved rapidly to enable researchers to simulate a wide variety of network scenarios. Its initial releases were designed for protocols with very low throughput, e.g. of around 1 Mbps, and for wireless sensor networks [5]. The simulation of very

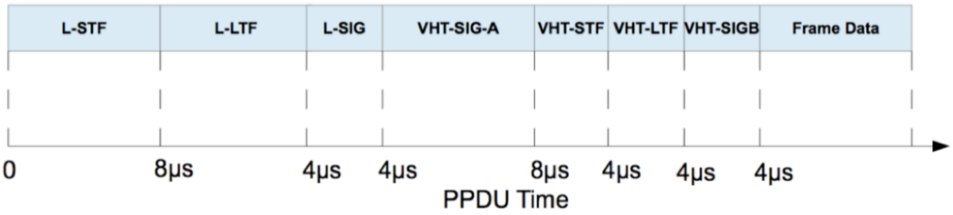

Fig. 6.1: IEEE802.11ac packet structure [6]

high throughput networks, such as those defined in the protocol IEEE802.11ac [6], pushes the ns-2 simulator to its limits but does not affect its simulation efficiency.

ns-2 has many existing modules for the simulation of IEEE802.11a/b, while some of its components have been developed to simulate other wireless networks, e.g. IEEE802.11e and IEEE802.16. For this thesis it was necessary to simulate the features of the protocol IEEE802.11ac [6]. To achieve this the existing code developed to simulate IEEE802.11e [36] was modified and extended.

The main code development work required for this thesis, consisted of the implementation of the IEEE802.11ac protocol within ns-2. It was a radical refinement and improvement of the existing IEEE802.11e codebase [36]. Four ACs are maintained with a parameterisation that is consistent with the IEEE802.11ac protocol. While it is possible to implement IEEE802.11ac if the node has only a single type of traffic to transmit, this is unlikely to reflect the usage pattern of future wireless networks.

One aspect of the implementation of IEEE802.11ac is in the modification of the header to reflect that of High Throughput (HT) and Very High Throughput (VHT), as used in IEEE802.11n and IEEE802.11ac [7]. In [36] the EDCA coordination function is implemented and reused to simulate the traffic AC. The IEEE802.11ac packet transmission speed is implemented using a time estimation method based on the standard [6].

The packet structure of a Very High Throughput PPDU is shown in figure 6.1. This structure is used to transmit both data and control packets, i.e. RTS, CTS ACK, packets. The number of VHT Long Training Fields (VHT-LTF) required in the header depends on the space-time stream as specified in the standard [6] . The transmission time, $TxTime$, estimated for a single frame in the IEEE802.11ac environment is obtained using the formula [6]:

$$\begin{aligned} TxTime \; &= T_{L-STF} + T_{L-LTF} + T_{L-SIG} + T_{VHT-SIG-A} + T_{VHT-STF} + \\ &\quad + N_{ES} \times T_{VHT-LTF} + T_{VHT-SIG-B} + T_{SYM} \times N_{SYM} \end{aligned} \tag{6.1}$$

where

T_{L-STF} is the Legacy short training sequence duration,

T_{L-LTF} is the Legacy long training sequence duration,

T_{L-SIG} is the Legacy SIGNAL field duration,

$T_{VHT-SIG-A}$ is the VHT SIGNAL A field duration,

$T_{VHT-STF}$ is the VHT short training field duration,

$T_{VHT-LTF}$ is the VHT LTF training field duration,

$T_{VHT-SIG-B}$ is the VHT SIGNAL B field duration,

N_{ES} is the number of BCC encoders,

T_{SYM} is the symbol transmission interval, i.e. $4\mu s$ and

N_{SYM} is the number of symbols.

N_{SYM} is given by:

$$N_{SYM} = m_{STBC} \times \left\lceil \frac{8 \times PktLen + 16 + 6}{m_{STBC} \times N_{DBPS}} \right\rceil \tag{6.2}$$

where

m_{STBC} is the Space Time Block code parameter,

$PktLen$ is the packet length and

N_{DBPS} is the number of data bits per symbol.

It should be noted that if STBC is not used then m_{STBC} is set to 1 [6]. The Guard Interval is set to $800ns$ and the Number of Binary Convolutional Codes, N_{ES}, is set to 2. All settings for the frame are extrapolated from the tables in [6] for a frame data rate of 780.0Mbps.

The main features of IEEE802.11ac [6] that are implemented in the simulations carried out are discussed below. Features of IEEE802.11ac [6] that are neglected are also detailed together with the rationale for their omission.

A 160MHz channel with a single spatial stream is implemented, i.e. MU-MIMO is not deployed. The code allows the user to set different channel sizes and alter the number of spatial streams used. The TXOP feature is implemented in this work.

The IEEE802.11ac CSMA/CA feature is not used for the following reasons. Firstly, it is expensive in terms of time. RTS and CTS control packets have to be transmitted with the same Layer 1 header used for data packets. Even if Layer 2 RTS and CTS frames are used they are very small in comparison to data frames, IEEE802.11ac throughput is so high that the transmission time for RTS and CTS control packets is comparable to that of data packets. Therefore RTS and CTS control packets can be considered as an inefficient use of bandwidth that gives rise to an unacceptable loss in channel throughput. Secondly, the QQM algorithm manages the contention window so as to reduce the number of collisions. If one considers the trade-off between the time lost when transmitting the RTS and CTS packets and the time lost when a collision occurs then it can be concluded that QQM makes CSMA/CA redundant.

Backward compatibility and a mix of various IEEE802.11 protocols on the wireless network are neglected in the code implementation, even if the Layer 1 packets are backward compatible. This is because the simulations aim to explore the improvements to be gained through deployment of the QQM algorithm and these will be significantly reduced when a mix of different protocols is used, mainly due to variations in the channel size. This is a reasonable assumption as it is likely that the IEEE802.11ac protocol will become dominant, fully replacing many of the protocols in use today. This is not an unreasonable assumption as this has been the case for its predecessors to date. Its advantages and the limited additional overhead cost make IEEE802.11ac the most likely candidate to replace current wireless network protocols.

The Frame Aggregation and Block Acknowledgement features were not implemented in the simulator. These can be used when there are a number of packets transmitted in sequence to the same destination. There are two important reasons for not implementing these two features. First of all, as assumed in section 5.5.1.2, the number of mobile stations that access the channel at the same time is small. This makes it improbable that more than one packet at a time is directed to the same destination for the number of traffic flows used in the simulations. Secondly,

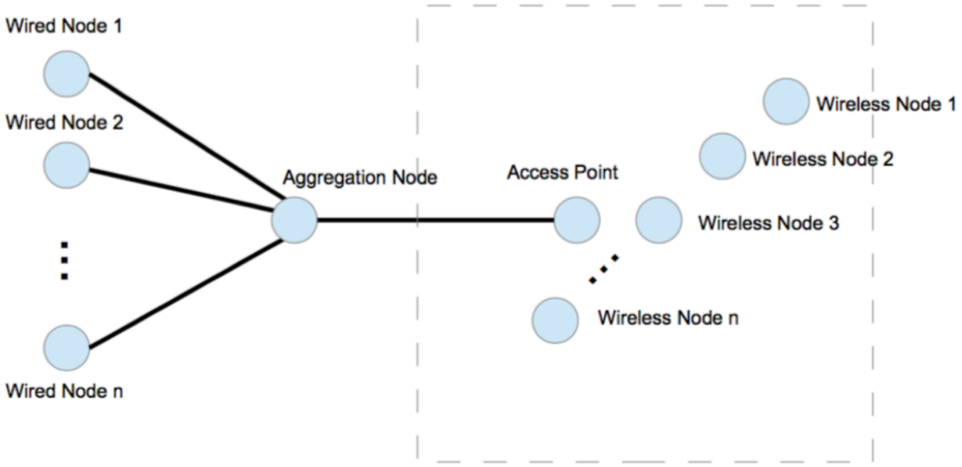

Fig. 6.2: Infrastructure wireless network used for the simulations.

these features provide an additional improvement to the network efficiency that goes beyond that offered by the QQM algorithm and, therefore, they are superfluous for the design, development and evaluation of QQM.

6.1.1 Simulation Environment

The simulation scenarios used reflect those of popular infrastructure wireless architectures, i.e. those typically used by Internet Service Providers.

The architecture used for the simulations is shown in figure 6.2. The number of source and destination nodes is set to a maximum of 50 and the number of ACs can vary for each simulation. Theoretically, as the model treats the AP like a node, up to 204 ACs can compete for access to the channel. Nodes in the network are not solely source or destination nodes: each node can be the source or destination for different kinds of traffic or a source and a destination at the same time. For example, nodes involved in a VoIP conversation are a source of traffic in one direction and a destination for traffic in the opposite direction.

The nodes are arranged along the diagonal of a 50 × 50 meter square. The AP is positioned close to the middle of the diagonal. The nodes are located at a distance of $\sqrt{2}$ meters from each other. The backbone and backhaul are replaced by a 100 Mbps link from the source to the aggregation router and by a 1 Gbps link from the aggregation router to the AP respectively.

The network delay is set to 10ms for each wired link between the sources and the aggregation router and from the aggregation router to the AP. The total RTT exceeds that of an ICMP request to a popular website, such as www.cisco.com. In the simulations it is assumed that a uniform number of packets cross the router in both directions and that the delay is set to the average link delay.

The radio propagation model used in the simulation is the free space model [195]. The propagation range is a circle. If a node is in the propagation range of another node then packet exchange between the nodes can take place. It is not unusual for packet transmissions to occur at mobile nodes that are located at different distances from the AP. This distance affects the receiving power. Packets are assumed to have been correctly received if their received power at the destination exceeds an established threshold. The transmission power of the station close to the AP is such that any transmitted packet will be correctly received, while the packets transmitted by more distant mobile stations are more likely to be lost. If a mobile station located far from the AP does not receive an acknowledgement from the AP then it assumes that the transmitted packet has been lost due to a collision. It then adjusts its CW and retransmits the packet at a later time.

Packet loss due to fading occurs at Layer 1 of the OSI model [5]; however, this loss also affects higher layers. This dissertation focuses on future wireless networks and fading is expected to have a marginal impact on such future technologies. Therefore fading is not considered in the simulations discussed below.

Limitations in the ns-2 design can also impact on the results obtained. As mentioned above, it is an older simulator and it was created at a time when network throughput was relatively low. It was optimised to simulate protocols that were in use or development at that time. High speed protocols, like IEEE801.11n, and more recently IEEE802.11ac, achieve far higher speeds than those anticipated by the initial designers of ns-2. For the architecture used in these simulations, detailed monitoring of the traffic showed that these potential limitations of ns-2 did not impact on the simulation results.

In the following subsection the traffic mix used for the simulatons is described in detail.

6.1.2 Traffic Mix

This subsection describes the different traffic types used to replicate real world traffic mixes in the simulations. The traffic mix was chosen to replicate real time traffic volumes that are consistent with everyday life settings where multimedia services are provided alongside other services. These other services are considered as background traffic.

The volume of traffic used for each experiment was chosen in order to balance three factors associated with the network. The first of these is the actual composition of the wireless network traffic [1] [2] [3] and, in particular, the use of a wireless network that is capable of carrying data from both a traditional wired network and a phone network. The second factor is the need to be able to push the network to the limit in order to explore the efficiency of the algorithm. The third consideration relates to the discussion on the contention window controller in section 5.5.1.2. This is needed to ensure the validity of the hypothesis relating to simultaneous access to the network.

The objective is to demonstrate by simulation that QQM operates satisfactorily in that is guarantees the highest level of QoE possible. This means there are some key features to the simulations and these are described below.

Firstly, the simulations must include traffic from every AC; however, it is not realistic to assume that network is in saturation at every instant in time i.e. that all nodes have at least one packet available to transmit at timeslot 0.

Secondly, the number of ACs accessing the channel must be comparable to a real-life scenario, with regular traffic volumes and a consistent number of flows. The simulations aim to show that QQM operates efficiently when dealing with a consistent volume of traffic. Logically, low traffic volumes are not as challenging for the QQM algorithm and are unlikely to demonstrate the added benefits of using QQM in a wireless network.

Thirdly, the traffic mix is not only determined based on experience, but also on observations of wireless networks. It is expected that future wireless networks will replace existing wireless networks and, to some extent, other networks such as wired access networks or wired and wireless phone networks. The traffic mix on such networks is likely to be a mix of traffic from a wide range of sources.

As previously discussed, VoIP traffic is transmitted as a CBR flow of UDP packets at a speed of 8 Kbps. In each second 50 packets are sent from the source to the destination and vice versa. This traffic is simulated in ns-2 by using CBR flows with the appropriate speed and packet sizes. The simulations use an extended and refined version of [1] [196] [2] and support codecs G.711 and G.729. The latter is used in the QQM evaluation simulations; in particular, in the evaluation of eQoS.

The number of VoIP calls in progress at the same same time will be determined by a number of hypotheses about the end users of the service. For example, a wireless network serving a business is used by employees during the course of their work. If a TDMA network is not available for regular phone calls, then it is reasonable to assume that a VoIP service over the wireless network will be used instead. In this case, the number of calls at the same time can be estimate using the standard Erlang [197] formulae.

The VoIP traffic model used in the simulations below is that of a constant presence of simultaneous VoIP calls on the network. The Erlang [197] calculation supposes 300 minutes of VoIP traffic per hour, generated by 5 VoIP calls of 5 minutes duration. Therefore, each AP manages at least 5 flows of VoIP traffic, and these are then increased up to a maximum of 10 simultaneous calls per Access Point.

The audio streaming traffic is represented in the simulations by UDP-like packets using Evalvid[3] [198] [199] [200] [97] [201] in ns-2. It was necessary to extend Evalvid to provide and monitor streaming services in the simulator.

Evalvid was adapted to the last release of ns-2 to include EDCA [36], VoIP [196] and

[1] http://www.tkn.tu-berlin.de/research/software_tools/

[2] https://www.tkn.tu-berlin.de/fileadmin/fg112/Hard_Software_Components/Software/qofis_v2.pdf

[3] http://www.tkn.tu-berlin.de/menue/research/evalvid/

IEEE802.11ac. A few modifications were made: information on the time when UDP packets were created was added to the header and TCP acknowledgements were moved from AC_0 to the same AC as the TCP data packets. The later is to prevent an increase in the priority of TCP acknowledgement packets and an associated increase in the queue length at AC_0. This means that these packets do not compromise the efficiency of the traffic prioritisation system or the delivery of VoIP and audio traffic.

Changes were also made to the software procedure used to create and rebuild streams. Audio was included in the streams and the Evalvid software procedures were modified to fill any gaps left by missing audio packets with the last one received. All the procedures to create and transmit a stream over the network were automatised using scripts. Finally, the software used to create and manage the streaming files was updated to the latest release.

The streamed audio traffic is assumed to be related to video transmission. The audio is configured as detailed in section 2.4.3 as CD quality; that is, 44KHz stereo sampling at 16 bits and encoded using AAC [62]. It is transmitted over the network using an average of 47 packets per second and the UDP protocol is used for both audio and video. Variable packet sizes do not give rise to any bias because the high speed of packet transmission on future wireless networks flattens any differences in packet transmission times. Audio traffic is managed by AC_0 together with the phone calls and in any associated calculations its transmission frequency is approximated at 50 packets per second i.e. to be the same as that used for phone calls.

The video stream used for the ns-2 simulations was the first 2 minutes and 20 seconds of a Star Wars movie [4]. This was chosen as it has a lot of rapid movements within each scene, uses a wide range of colours and contains contrasting scene elements. This means that it exhibits a great variability in frame sizes. Video streaming was simulated using UDP-like packets. It was implemented in ns-2 using an extended version of Evalvid [198] [199] [200]. Video was coded using MPEG-4 as described in section 2.4.4. The original video has a format of 1920 × 1080, i.e. a format ratio of 1.78, and required 69.8 MBytes of data to be streamed in 140 seconds. The stream chosen was sufficiently long to ensure it contains all possible frame types.

[4] `http://www.starwars.com/films/star-wars-episode-iv-a-new-hope`

Lower video resolutions will have a different standard ratio to that chosen above; however, the video quality evaluations discussed in this chapter do not focus on the quality of the frames, their compression or the video content. The loss of video streaming packets during transmission is not affected by the frame resolution ratio.

The streaming formats for video are summarised in table 6.1. The video streaming format 720×480, used for TV streaming in NTSC systems, is the best possible but is not recommended for use on future wireless networks, as the format requires a very large throughput and compromises the network's other real time services.

In the simulations presented in this thesis, a video resolution of 480×320 is used. This means that thirty frames are transmitted per second; with, on average, 13 packets for each I frame and 3 packets for each P frame. The packet sizes are set to a maximum of 1024 Bytes and the traffic is approximated by 100 video packets per second, of which 20% are I frames and 80% are P frames. Audio is simulated by the streaming of 47 frames per second.

As mentioned in the previous chapter, QQM was designed to manage real time traffic at the AP; therefore, AC_2 and AC_3 traffic is considered as background traffic. They are only of interest in terms of the volume of traffic produced; all other qualitative aspects of this traffic are neglected in the analysis. To simulate a large amount of AC_2 traffic, FTP flows with a packet size of 1024 bytes are created. These originate from sources distributed throughout the wired and wireless nodes. UDP flows in CBR configuration are also used to generate AC_2 traffic with a packet size of 256 bytes and at a rate of 50 Kb per second. The background AC_3 traffic is a mix of FTP and CBR flows. The FTP traffic is composed by TCP flows with a packet size of 1024 bytes and the CBR traffic is composed by UDP flows with a constant bit rate of 200 Kbits/s and a packet size of 256 bytes, i.e. at a rate of 49 packets per second in total.

The throughput per flow is sufficient to ensure the channel is always occupied and that there is always a packet waiting to be transmitted.

In ns-2 each flow has an unique identifier, $flowID$, in the packet header. The $flowID$ identifies the source, the destination and the traffic flow characteristics. All packets passing through the queue at Layer 2 are statically mapped at the node. It is possible to implement

Video Format	Stream size size (in MB)	I frame size (average in packets)	P frame size (average in packets)	Format ratio	Bitrate
720 × 480	22.20	19	4	1.5	976
480 × 320	17.36	13	3	1.5	662
320 × 240	13.66	8	2	1.3	457
160 × 120	8.49	4	2	1.3	227

Table 6.1: Video Stream Formats

dynamic packet queue switching by using deep packet inspection. For modern processors, this is not too costly in terms of the CPU time required.

In the next subsection the QoS metrics implemented within the simulator are presented.

6.1.3 QoS Metrics

The efficiency of QQM is assessed through simulation. Its performance is compared with that of networks that only use EDCA. The quality metrics used for this comparison are measures of perceived and intrinsic QoS. The previously defined perceived metric, eQoS, is used to estimate the assessed QoS i.e. the QoE. The intrinsic QoS is captured through physical parameters; improvements in these will indicate an increase in end-user satisfaction.

The most important intrinsic QoS metric used to evaluate the benefits of QQM are the number of dropped packets and the packet delay. Packets may be dropped due to network events like, for example, queue overflow, routing errors or an excessive number of transmission attempts. This also includes the new version of Controlling Queue Delay [192] where real time service packets are dropped because the delay induced by the queue latency and the serving time means that their overall delay would exceed the set threshold.

The packet delay and jitter are two intrinsic QoS metrics that are significant for QoE evaluation. The average delay is a key metric that can be used to estimate how QQM reduces the incidence of packet drops due to an inability to meet delay thresholds. In conjunction with packet delay, jitter is another mechanism for capturing changes in eQoS. Jitter is defined as the effective delay between the transmission of two sequential packets. The jitter metric is unsuitable for use with variable bit rate traffic. An upper limit on the transmission delay is sufficient to determine if packets can be transmitted without a loss of quality.

Other metrics, such as throughput, are of less interest when evaluating the systems under consideration in this thesis as they give indirect and imprecise information on the network's performance. For example, real time VoIP services send packets at regular intervals and their throughput is unchanged over a long period of time, i.e. over time periods of the order of 1 second. Throughput variations become evident when instantaneous throughput calculations are used. These capture the variation in packet delay times due to queueing delays at the node. However, metrics of this kind do not provide information on the aggregate benefit of QQM for the management of the entire wireless network.

6.2 Evaluation of the eQoS Metric

In this section the eQoS metric is applied to media streams [14]. The results are used to investigate a possible relationship between eQoS and a traditional computational algorithm used to estimate the MOS of a stream.

As explained in section 3.1.1, QoE, as evaluated using the MOS, and eQoS are different metrics; not only because QoE is subjective and eQoS is objective, but also because eQoS is evaluated almost instantaneously while QoE is determined over a much longer time frame. Therefore it is not possible to directly compare the two metrics. One way to over come this limitation is to evaluate both metrics for the same network stream under very specific conditions where a constant number of packets per second is lost and then compare an average of the eQoS values with those obtained using the MOS.

There are three reasons to conduct the evaluation in this way. The first reason is due to the different time frames needed for the calculation of eQoS and QoE. While instantaneous drops in quality are perceived by eQoS, they can have little or no impact on the traditional computational algorithm. This underlines a significant advantage of eQoS; namely its ability to capture instantaneous variations in quality.

Second, the use of network traffic focuses the eQoS and QoE evaluation on scenarios where some of the streamed data is lost. The causes of this loss of information; for example due to traffic variations or a increasing number of mobile stations, are not important for this evaluation. What is key is that some information, in the form of packets, is missing.

Third, the artificial loss of information is achieved by randomly dropping a few packets per second. This means that loss events are independent. As previously noted when loss events are caused by congestion it is more likely that packet drops occur in sequence; causing a rapid decrease in quality to the lowest MOS value.

The evaluation was performed using some streamed files. The files were altered in order to replicate the desired average packet loss. Each file was then streamed across the network and the MOS and eQoS calculated for the same stream in various scenarios. For each scenario a constant number of packets was dropped per second: in the first scenario one packet is dropped per second, in the next scenario two packets per second are dropped and so on. The number of loss events per second is increased until the encoded stream can no longer be rebuild by the receiver. The average eQoS and MOS values calculated for each scenario are compared. A statistical regression [202] analysis is used to show a possible relationship between the average eQoS and the MOS.

As concluded in the discussion about media sampling in sections 2.4.2, 2.4.3 and 2.4.4, the highest MOS score of 5 will never be measured by the computational algorithms even when no packets are lost. To simplify the calculations, the conversion of an eQoS value to a MOS score starts from the highest MOS score of 5 even though any measured value will be at most 4.5.

In the following three subsections this evaluation is performed for VoIP, audio and video streams.

6.2.1 Evaluation of eQoS for VoIP services

In this section a VoIP service is artificially corrupted to simulate packet loss and eQoS and PESQ [84] are calculated. The experimental platform uses a tailor-made system of scripts to simulate the loss of packets in the VoIP stream and to then evaluate the eQoS.

Each scenario consists of a stream, in this case a phone call, that has a constant number of packets dropped per second. In the first scenario one packet is dropped per second, and in the final scenario 16 packets are dropped per second. G.729 [57] encoding can tolerate the loss of up to 16 packets per second. If more than 16 packets are lost per second then it is not possible to decode the stream.

The software created works in conjunction with that described in [203] and [196] for simulating a VoIP service over ns-2 [194]. The ns-2 simulation environment consists of an AP and a mobile station, both are at fixed positions a few meters apart.

It was necessary to adapt the ns-2 software and procedures in order to use G.729 encoding. The simulation streams used for the VoIP evaluation are speech files that simulate a phone call. These streams are provided as examples by the ITU [84] [57]. Their content has the same characteristics as a phone call, for example, in terms of language and tone. The encoded file is processed by a script to simulate the loss of information described before. As the G.729 encoding process reduces the call quality through compression the initial MOS score is below the maximum value of 5.

The original transmission file, where no packets have been dropped, is compared with the file received where packets have been lost, using the PESQ [84] [203] algorithm. The results are plotted using Matlab [182]. PESQ is the algorithm used by the ITU-T to evaluate QoE for speech. eQoS is also calculated for each stream. The average value of eQoS for each stream is plotted using Matlab [182]. The results are shown in figure 6.3.

The relationship between the PESQ and average eQoS is demonstrated using a scatter plot and linear regression. Three curves are shown in figure 6.3: The central line is the regression line, while the two dashed curves either side of it are the associated 95% confidence interval. It can be seen that all points are inside the 95% confidence interval bounds. The statistics associated with the regression analysis are:

β_0=3.3929 β_1=-1.7149

R^2=0.9507 $adjR^2$=0.9474

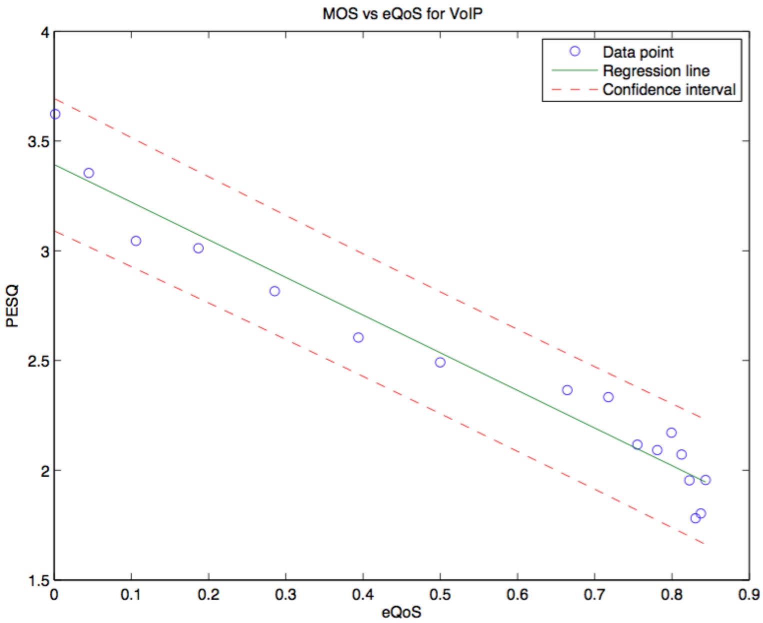

Fig. 6.3: PESQ MOS vs average eQoS for VoIP: Experiment 1

The high values of the coefficient of determination, R^2, and the adjusted R^2 value, suggest that the linear regression model is a good fit. The correlation coefficient is $r = -0.975$. This indicates a strong linear relationship between the measured PESQ values and the average of the calculated eQoS values. Just under 5% of variation in the relationship between PESQ and eQoS is not explained by the regression model. This may be due to unpredictable extrinsic streaming delays.

The experiment was repeated with another speech file [84]. The results are shown in figure 6.4. The regression analysis data for this second experiment are:

β_0=3.4666 β_1=-1.3615

R^2=0.8926 $adjR^2$=0.8849

In the second experiment the values of R^2 and $adjR^2$ are high and the coefficient of correlation is $r = -0.9448$. These all indicate a strong linear relationship between the two variables. All but one of the points lies inside the 95% confidence interval for the regression line.

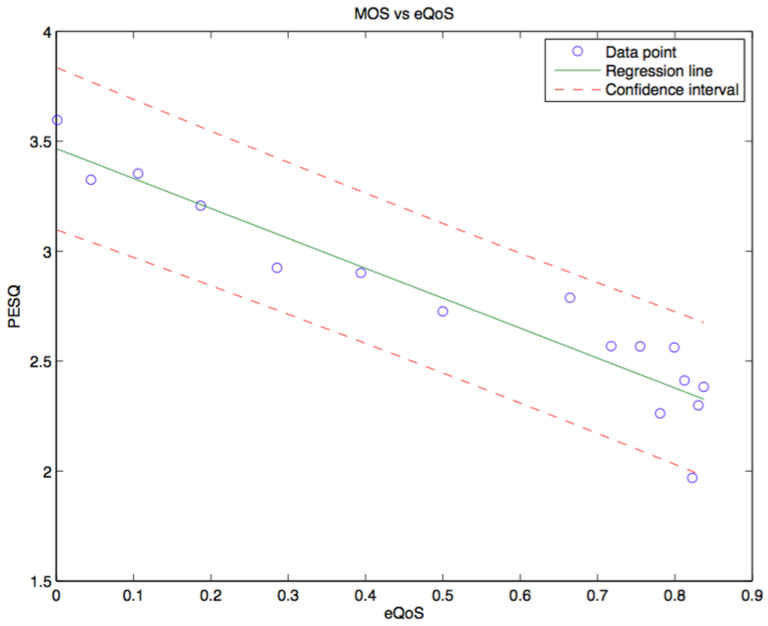

Fig. 6.4: PESQ MOS vs average eQoS for VoIP: Experiment 2

6.2.2 Evaluation of eQoS for Audio Services

In this section it will be shown that, when averaged, the eQoS metric for audio streaming has a direct linear relationship with the existing QoE metric PEAQ [93].

A tailor-made system was created to produce streams with known quality impairments and to then evaluate them using PEAQ [93] and the eQoS method. The system used to carry out the simulations was derived from Evalvid [198] [199] [200]. In particular, the procedures were modified and software packages were added to reproduce audio and video streams.

The audio stream is a stereo 16 bit, 44KHz audio file encoded using AAC. The encoded file was inserted into an MPEG-4 file as audio track together with associated video. It was than transmitted over a simulated wireless network. The evaluation was performed using the same methodology as for VoIP traffic.

A loss of quality was created by dropping a constant number of packets per second from the stream file at the receiver. The first stream had one dropped packet per second, and the experiment was then repeated, increasing the number of dropped packets per second each time up to a maximum of 30. The data from the impaired streams were then evaluated using a traditional QoE metric and eQoS. The results are reported in figure 6.5.

If an audio packet in a data stream is lost the system can attempt to recover it. For example, this could be done by a multimedia player used at the application level. Rebuilding processes, like interpolation or averaging between two received packets, are excluded from the calculations in this thesis. It is important to calculate the effect each dropped packet produces, even if it can eventually be rebuilt so that the streaming gap and associated loss in quality are hidden from the end user. A loss in quality may also be hidden by the hardware used; for example, by headphones for audio or by high definition screens in the case of video.

Since it is not possible to predetermine the application capabilities or the hardware equipment at the receiver, eQoS estimates the quality at the node. In the simulations lost packets are replaced by the last packet received as a simple, worst-case approximation of a data recovery mechanism and to make the results obtained more realistic.

As previously noted, comparisons between QoE and eQoS are only possible for the specific scenario used as they are very different metrics: eQoS is an objective instantaneous metric while QoE is a subjective metric. The evaluation shows a strong linear relationship between the average eQoS and the QoE, even in the case of streaming services.

Figure 6.5 shows a graph of PEAQ [204] [188] against average eQoS. The PEAQ software used for this experiment is PQevalAudio[5] [204] and it has two particular features. First, quality is measured using the ODG [93] [94]. Second, the audio files have a sampling rate of 48KHz. The regression line shows a strong linear relationship between the two metrics and the associated statistical parameters are:

β_0=-1.4435 β_1=-2.5783

R^2=0.9573 $adjR^2$=0.9558

The coefficient of correlation $r = -0.9784$, confirms a strong linear relationship between the two metrics. In the next section an example of eQoS calculations for video streaming is

[5] http://www-mmsp.ece.mcgill.ca/documents/downloads/PQevalAudio/

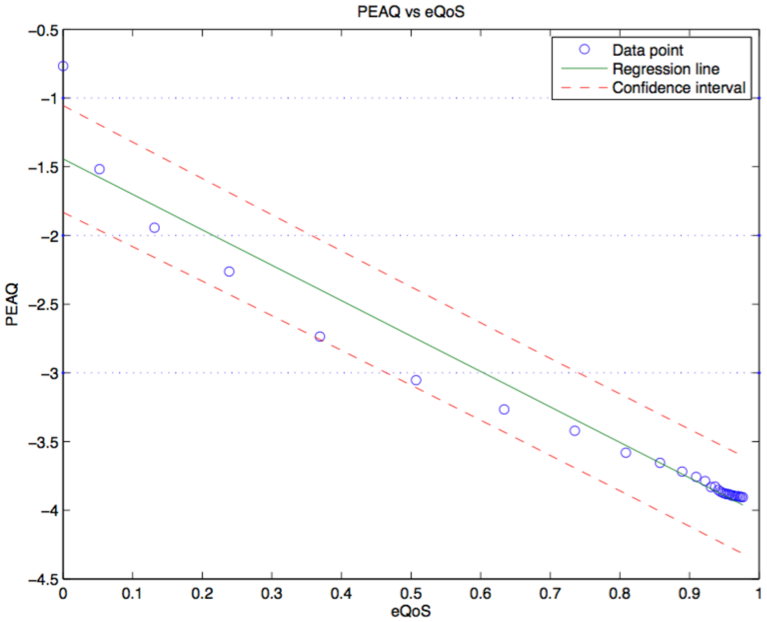

Fig. 6.5: Regression graph of PEAQ vs average eQoS for Audio Streaming

detailed.

6.2.3 Evaluation of eQoS for Video Services

In this section a possible relationship between the eQoS metric and PSNR/MOS is investigated. While PSNR is not the best estimator of QoE, it is widely used in the literature as a valid estimation metric [97]. PSNR values can be passed to the MOS metric [97] to give an estimate of QoE for a video stream. This follows the software procedures given in [200], [198] and [199].

The video stream is encoded using H.264. The encoded data was inserted into an MPEG-4 file and transmitted over a simulated wireless network to verify the correct encoding, MPEG-4 structure and packetisation. As in the previous subsection transmission was performed using ns-2 [194], version 2.35 with the addition of the Evalvid package for ns-2 [200].

The streamed file was 140 seconds long. The video size was 480 × 320 pixels at a rate of 30 frames per second. This is one of the most popular ways to provide streaming services on mobile devices and streaming websites.

Following the procedures for Evalvid [198] [201], at the beginning of the simulation the file is in a raw YUV format, where each frame is a complete picture. The file is then encoded using H.264. Transmission loss is introduced by dropping a constant number of packets per second in each simulation. Each dropped packet is replaced by zeros, and the stream is then decoded from H.264 to the raw YUV format.

PSNR is calculated by comparing the YUV files at the source with those at the destination [97]. The PSNR score is an average across the PSNR values calculated per frame. The PSNR value is then translated onto the MOS scale.

eQoS is calculated at the AP using equation 4.21 for video streaming. The values used for the parameters α, β and γ are: 0.4, 0.3 and 0.3 respectively. The weights given to the adjacency contribution for I and P frames are equal and are conformant with equations 4.22 and 4.23. α, β and γ were chosen by drawing the desired eQoS curve. The eQoS is set to measure a low MOS level when just a few packets are dropped as it is assumed that the end user perceives only a minimal loss in quality.

MOS results and averaged eQoS values are shown in figure 6.6. A linear regression relationship between PSNR/MOS and average eQoS is evident. The horizontal and vertical axes give the average eQoS and PSNR respectively. The horizontal dashed lines are the MOS levels, they assist in the visualisation of the relationship between the metrics. The linear regression analysis gives:

β_0=38.1162 β_1=-30.879

R^2=0.9772 $adjR^2$=0.9764

The coefficient of correlation is $r = -0.9886$. This shows a strong negative correlation between the two metrics.

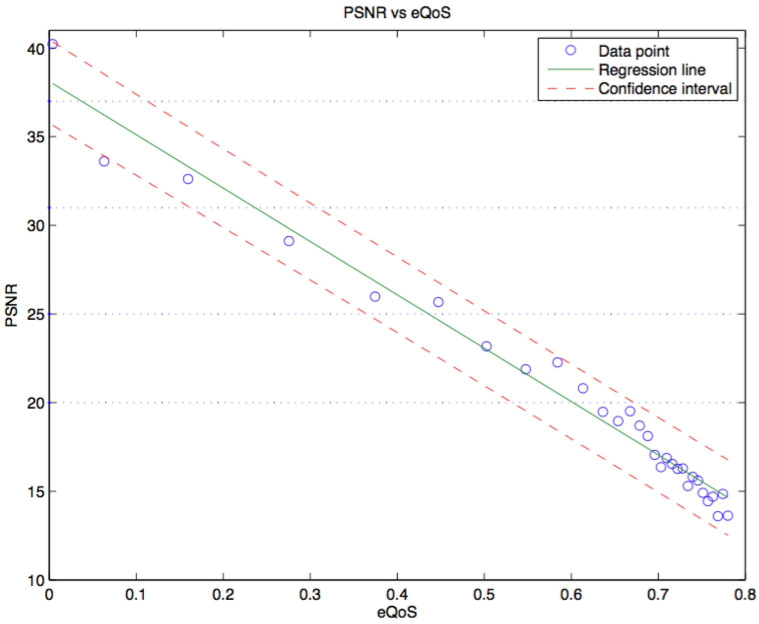

Fig. 6.6: Regression graph of PSNR vs average eQoS for Video Streaming

6.3 Packet Transmission Times & Throughput Evaluation for 802.11ac

This section shows a practical application of the theoretical model described in section 5.1 [15]. This model is used to estimate the probabilities that a packet is transmitted, that there is a collision or that the channel is found to be idle by an AC accessing it.

A practical application of the model is in the calculation of the average time needed to transmit a single packet in a future wireless network. This is done numerically by comparing the average number of packets transmitted in one second during a simulation with the expected time needed to transmit the same number of packets as inferred from the theoretical model. The time need to transmit a packet is inferred from equations 5.19, 5.20, 5.21 and 5.22.

The results of five experiments are presented in this section. In general, each experiment builds on the previous one by increasing the number of flows and mobile stations in the traffic mix. In each experiment the average number of packets transmitted in one second during the simulation is compared with the time needed to transmit the same number of packets using the theoretical model. Results are reported as the percentage difference between the estimate time obtained from the simulator and the value predicted by the model. These are used to show how accurate the theoretical model is at predicting the average transmission time for each packet.

The calculations for the theoretical model were carried out using Matlab [182], while the simulations were performed using ns-2 [194]. Matlab was used to draw the graphs. The number of packets transmitted per second and the time estimated by the theoretical model to transmit them are shown to be in close agreement. That means that the theoretical model can estimate the average time to transmit a packet in future wireless networks when the probability of events not captured by the model is low e.g. collisions between more than 2 ACs, channel interrupts etc.

The first experiment is a simulation of 5 VoIP calls together with 5 Video and Audio streaming flows. These flows are in the downlink direction from the wired network to the wireless nodes. The VoIP and audio traffic are managed by AC_0, while the video traffic is managed by AC_1. The best effort traffic on AC_2 is simulated by 10 TCP flows: 5 from wired nodes to wireless nodes, and, vice versa, 5 flows from wireless nodes to wired nodes. The background traffic on AC_3 is provided by a mix of 5 TCP flows and 5 UDP flows, i.e. CBR flows, from the wired to the wireless nodes.

The traffic is summarised in table 6.2. While this experiment is intended to model a real life situation, the traffic managed by AC_2 is greater than would be expected in order to saturate the network without a loss of any real time service packets.

Table 6.2 reports the CW sizes used in the calculations for the theoretical model and the maximum number of ACs competing for the channel.

The ns-2 simulation results and the average values are shown in figure 6.7. The x-axis shows the simulation time, while the y- axis shows the packets per second exchanged between the AP and the wireless nodes. The blue lines are the simulation results and the red dashed lines are the averages.

The lines marked with AC_VO represent the traffic managed by AC_0. The lines representing AC_0 traffic overlap because VoIP and audio traffic is transmitted using a constant number

AC	CW size	Flows from AP to MS	Flows from MS to AP	Maximum number of competing MS
AC_0 / AC_VO	4	5 phone + 5 audio	5 phone	2
AC_1 / AC_VI	8	5 video		1
AC_2 / AC_BE	16	3 TCP	2 TCP	2
AC_3 / AC_BG	16	3 TCP + 2 UDP (CBR)		2

Table 6.2: Summary of CW Sizes and Traffic Flows for Experiment 1

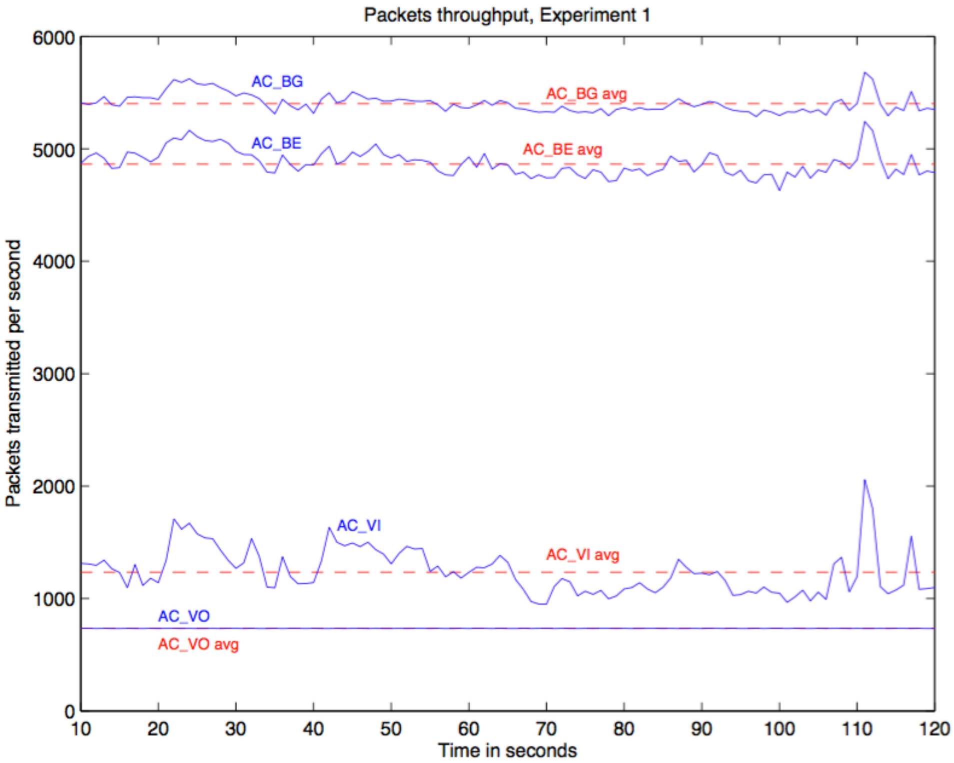

Fig. 6.7: Simulation results and traffic averages for 5 VoIP, 5 Audio and 5 Video streams, 5 TCP flows and a mix of 3 TCP and 2 UDP flows

of packets per second. The lines marked AC_VI represent the sum of the traffic managed by AC_1 with the AC_VO traffic. All the video traffic flows start at the same instant in time. This represents a worst case scenario for video traffic because all the peaks in video transmission overlap and so these peaks are amplified.

The lines marked with AC_BE represent the sum of the traffic managed by AC_2 and the AC_0 and AC_1 traffic.

The lines marked with AC_BG represent the sum of the traffic managed by AC_3 and the AC_0, AC_1 and AC_2 traffic.

The calculations performed using the theoretical model are based on those given in section 5.2. All flows managed by AC_0 and AC_1 are transmitted first and they are competing to access the channel. The calculations for these two ACs include the TXOP feature. After this traffic has been transmitted, the traffic managed by AC_2 is transmitted. This is TCP/FTP traffic. The remaining time is then reserved for the transmission of the traffic managed by AC_3, when only AC_3 flows are competing to access the channel. In general, AC_2s and AC_3s transmit UDP/CBR traffic before any TCP/FTP packets are sent in their respective transmission time slots. In both cases the TCP/FTP traffic consists of data packets in one direction and acknowledgement packets in the opposite direction.

As discussed in section 5.2, the number of AC_2s and AC_3s accessing the channel at the same time has to be fixed. It should be noted that the number of AC_0s and AC_1s accessing the channel at the same time is also fixed, see section 5.4. AC_2 has three TCP/FTP flows from the wired to the wireless network and two flows in the opposite direction. As mentioned above for each TCP/FTP data packet received at the destination a TCP/FTP acknowledgement is sent back. Considering that AC_2 and AC_3 have a large CW_{min}, the timeslot to access the channel for AC_2 and AC_3 can be approximated as $\lceil \mathbb{E}[\frac{CW_{min}}{2}] \rceil$ using equation 5.18 . Therefore following the theoretical model four timeslots will be added to the transmission time of AC_2 and AC_3 packets and the competition to access the channel will be between only AC_2s or only AC_3s.

It is assumed that the RTT for TCP/FTP flows exceeds the typical delay accumulated on the wired network; that is, it exceeds 40 milliseconds [47]. This means there are less than 25

RTTs per second. In each RTT a flow can transmit a maximum of 32 packets if the congestion window is large [47]. In the case of Experiment 1, less than 4000 packets are transmitted by the AC_2s. This includes both data and acknowledgement packets. In theory, one of the 5 AC_2s in the simulation can transmit 400 packets. This corresponds to an average congestion window size of 16 packets. Based on the discussion in section 5.4 it can be assumed that, in theory, each AC_2 is accessing the channel less than 16 times every 40 milliseconds. About 13 packets can be transmitted every 2.5 milliseconds; this corresponds to a slot as described in section 5.4.

A large number of flows are from the wired to the wireless network; so the AP has always a packet ready to transmit. This means that a maximum of 2 AC_2s are competing to access the channel at any instant in time. AC_3 traffic sources all originate on the wired network and converge at the AP; therefore, the only traffic flowing from the wireless to the wired network is TCP/FTP acknowledgement packets. From the discussion in section 5.4 it can be assumed that a maximum of 2 AC_3s are competing to access the channel at the same time.

In the first experiment an average of about 5400 packets are transmitted each second. This is shown in figure 6.7. The theoretical model described in section 5.1 estimates that it will take 1.001684 seconds to transmit this number of packets.

This means that the average transmission time for each packet estimated by the theoretical model differs from the average time observed in the simulation by 0.1684$\%$. The difference is 0.001684 seconds and this corresponds to the time needed to transmit 9 packets of the total.

In Experiment 2 the number of flows managed by AC_2 and AC_3 is increased. Table 6.3 summarises the details of the flows used in this simulation.

In the second experiment, it can be observed that an average of about 5410 packets are transmitted per second. This is shown in figure 6.8. The theoretical model estimates that 1.00464 seconds are needed to transmit this number of packets. The average transmission time estimated for each packet by the theoretical model differs from the average time observed in the simulation by 0.464$\%$. The difference of -0.00464 seconds corresponds to the theoretical time needed to transmit 25.1 packets of the total.

In Experiment 3 the flows managed by AC_2 are changed, while the flows used for AC_3

AC	CW size	Flows from AP to MS	Flows from MS to AP	Maximum number of competing MS
AC_0 / AC_VO	4	5 phone + 5 audio	5 phone	2
AC_1 / AC_VI	8	5 video		1
AC_2 / AC_BE	16	5 TCP	5 TCP	4
AC_3 /AC_BG	16	5 TCP + 5 UDP (CBR)		2

Table 6.3: Summary of CW Sizes and Traffic Flows for Experiment 2

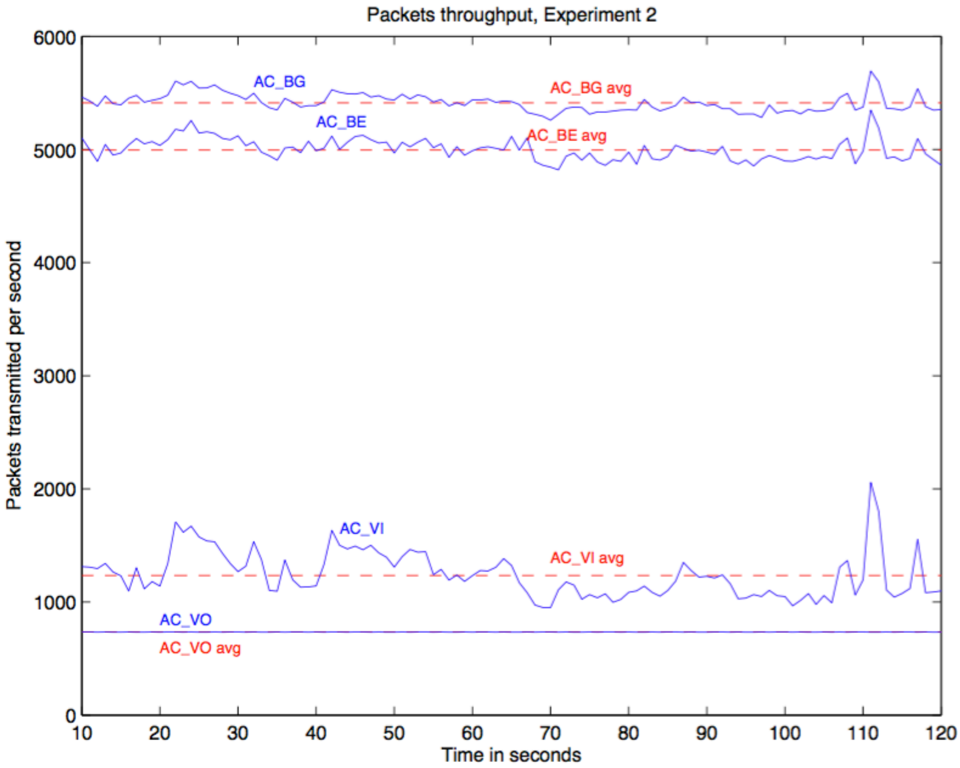

Fig. 6.8: Simulation results and traffic averages for 10 VoIP, 5 Audio and 5 Video streams, 10 TCP flows and a mix of 5 TCP and 5 UDP flows

AC	CW size	Flows from AP to MS	Flows from MS to AP	Maximum number of competing MS
AC_0 / AC_VO	4	5 phone + 5 audio	5 phone	2
AC_1 / AC_VI	8	5 video		1
AC_2 / AC_BE	16	3 TCP (cw=2) + 2 UDP	3 TCP (cw=2)	3
AC_3 / AC_BG	16	3 TCP + 2 UDP (CBR)		2

Table 6.4: Summary of CW Sizes and Traffic Flows for Experiment 3

remain the same as those used in Experiments 1. Table 6.4 summarises the details of the flows used in this simulation.

The AC_2 traffic is now bidirectional TCP/FTP traffic with a congestion window size of two packets and CBR traffic with a throughput of 50 Kbits per second and packet size of 256 bytes. These settings were chosen to match those of real world best effort traffic; for example, telnet and p2p traffic.

The results obtained by simulation and using the theoretical model are shown in figure 6.9. The simulation shows that an average of about 4822 packets are transmitted each second. This means that the average time needed to transmit a packet is greater than in the previous experiments. There are two reasons for this. Firstly, once all the possible packets managed by AC_0, AC_1 and AC_2 are transmitted there is still sufficient bandwidth available to allow the AC_3s to transmit a large number of packets. Secondly, the number of packets transmitted in the wireless network is smaller than Experiments 1 and 2 because AC2 flows have a low CBR rate and small TCP congestion window.

The number of AC_2s and AC_3s accessing the channel at the same time can be set to two. The timeslot to access the channel for AC_3 can be approximated as $\lceil \mathbb{E}[\frac{CW_{min}}{2}] \rceil + 1$ to compensate the low CBR rate and the small tcp congestion window of AC_2. The average transmission time estimated for each packet by the theoretical model differs from the average time calculated using the simulator by 0.296%, that is 0.00296 for a total time of 1.00296 or 14.3 packets.

The traffic settings for Experiment 4 are summarised in table 6.5. This experiment is similar

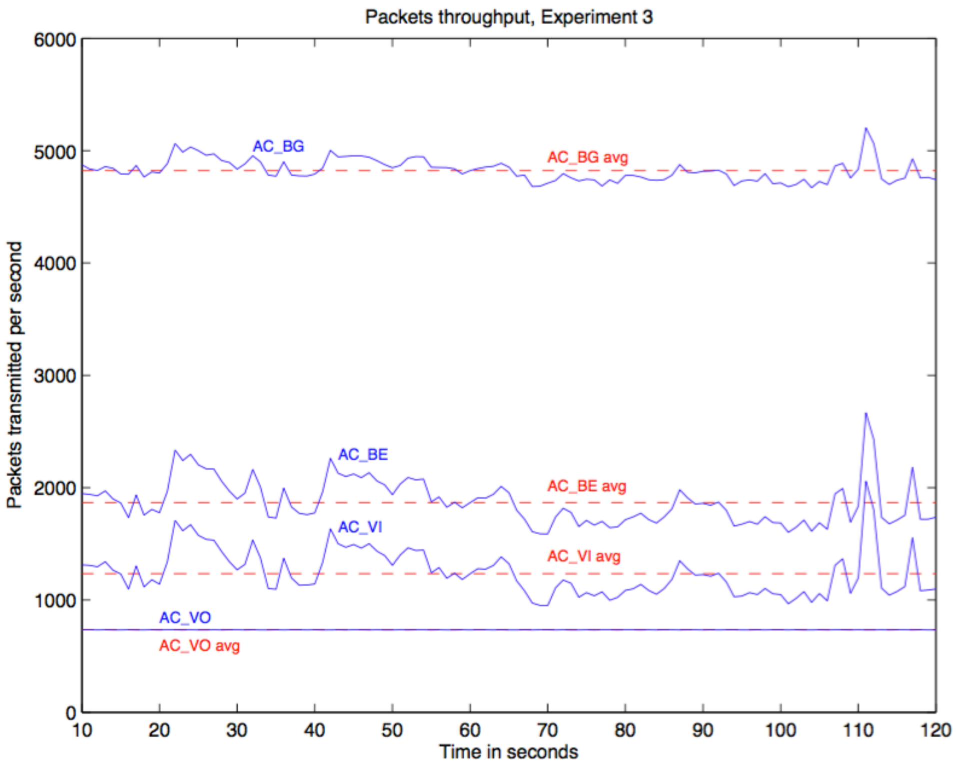

Fig. 6.9: Simulation results and traffic averages for 10 VoIP, 5 Audio and 5 Video streams, 6 TCP flows, 2 UDP flows and a mix of 5 TCP and UDP flows

AC	CW size	Flows from AP to MS	Flows from MS to AP	Maximum number of competing MS
AC_0 / AC_VO	4	5 phone + 5 audio	5 phone	2
AC_1 / AC_VI	8	5 video		1
AC_2 / AC_BE	16	5 TCP (cw=2) + 5 UDP	5 TCP (cw=2)	4
AC_3 / AC_BG	16	5 TCP + 5 UDP (CBR)		3

Table 6.5: Summary of CW Sizes and Traffic Flows for Experiment 4

to Experiment 3; however, the number of flows managed by AC_2 and AC_3 is increased. The number of AC_2s and AC_3s accessing the channel at the same time is still set to a maximum of two in the theoretical model.

The simulator shows that an average of about 4890 packets are transmitted each second. This is shown in figure 6.10. The wireless network is still not in saturation, even though the number of flows has been increased. The average transmission time estimated for each packet by the theoretical model differs from the average time calculated using the simulator by 0.06%, that is 0.0006 for a total time of 1.0006 or 3 packets. . This indicates that, in this case, the estimates obtained using the theoretical model are in line with those observed in the simulator.

In the final experiment in this section the traffic managed by each of the ACs is increased. Table 6.6 shows the number of flows used for each AC. Based on the discussion in section 5.4, the number of flows competing to access the channel is now set to three for AC_0 and two AC_1. The number of AC_2 flows competing to access the channel is increased to a maximum of four. The increased traffic for AC_0 and AC_1 means that there are longer gaps between transmissions for the AC_2 traffic. The number of AC_3 flows competing to access the channel is increased to a maximum of eight.

The results obtained from the simulator and the theoretical model are shown in figure 6.11. The simulator transmits an average of about 5362 packets each second. Due to the large volume of traffic being transmitted the system is in saturation.

The average transmission time estimated for each packet by the theoretical model differs

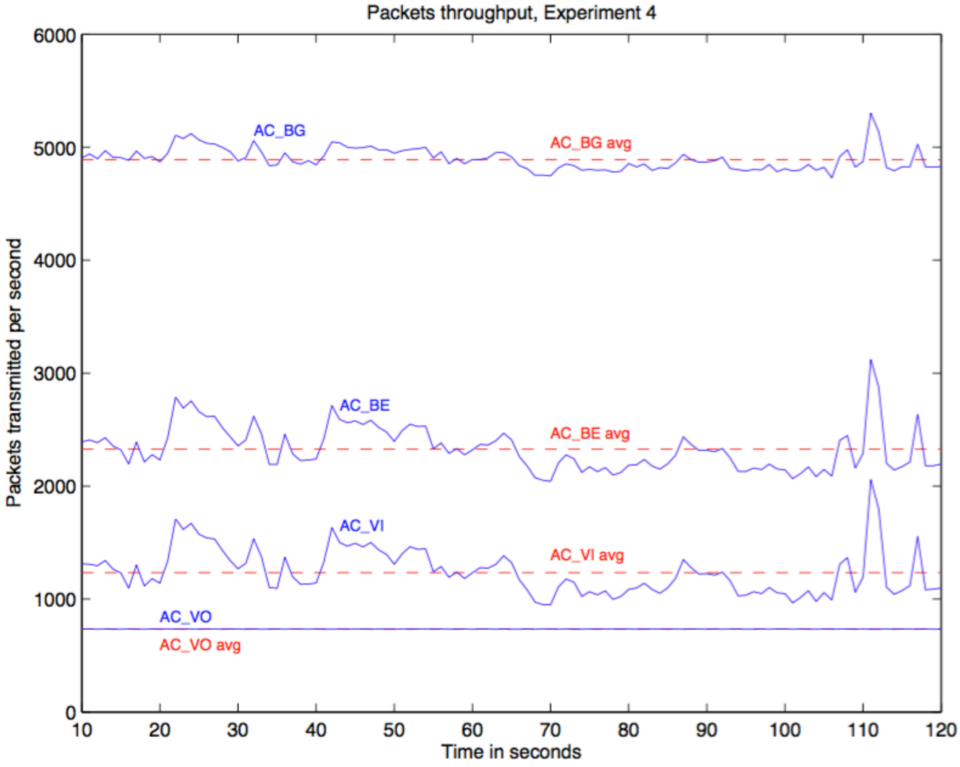

Fig. 6.10: Simulation results and traffic averages for 10 VoIP, 5 Audio and 5 Video streams, 10 TCP flows, 5 UDP flows and a mix of 5 TCP and 5 UDP flows

AC	CW size	Flows from AP to MS	Flows from MS to AP	Maximum number of competing MS
AC_0 / AC_VO	4	10 phone + 5 audio	10 phone + 5 audio	3
AC_1 / AC_VI	8	5 video	5 video	2
AC_2 / AC_BE	16	5 TCP (cw=2) + 5 UDP	5 TCP (cw=2)	4
AC_3 / AC_BG	16	10 TCP + 5 UDP (CBR)	5 TCP	8

Table 6.6: Summary of CW Sizes and Traffic Flows for Experiment 5

from the average time calculated using the simulator 0.703%, that is 0.00703, for a total time of 1.00703 or 37.7 packets.

In this section a practical application of the theoretical model presented in section 5.1 has been explored. In the following section attention shifts to QQM and, in particular, to an evaluation of the subsystems that form a key part of QQM.

6.4 Performance Evaulation of the QQM Subsystems

In this section each of the QQM subsystems is evaluated by simulation [15]. QQM's efficiency is demonstrated through simulations that compare the performance of future wireless networks where EDCA is implemented with that of the same networks where QQM is implemented.

Six distinct configurations are used. The first scenario is labelled configuration 0 and the last scenario is configuration 5. Configuration 0 includes: 5 VoIP calls managed by the AC_0s; 10 audio and video unidirectional flows from the wired to the wireless network managed by the AC_0s and AC_1s respectively; 5 audio and video bidirectional flows managed by the AC_0s and AC_1s respectively; 5 TCP bidirectional flows with a congestion window size of eight packets and a packet size 1024 bytes, managed by the AC_2s; five UDP/CBR unidirectional flows from the wired to the wireless network with a packet size of 256 bytes and a speed of 50 Kb managed by the AC_2s; 5 TCP unidirectional flows from the wired to the wireless network with a packet size of 1024 bytes managed by the AC_3s; 5 UCP CBR unidirectional flows from the wired to the

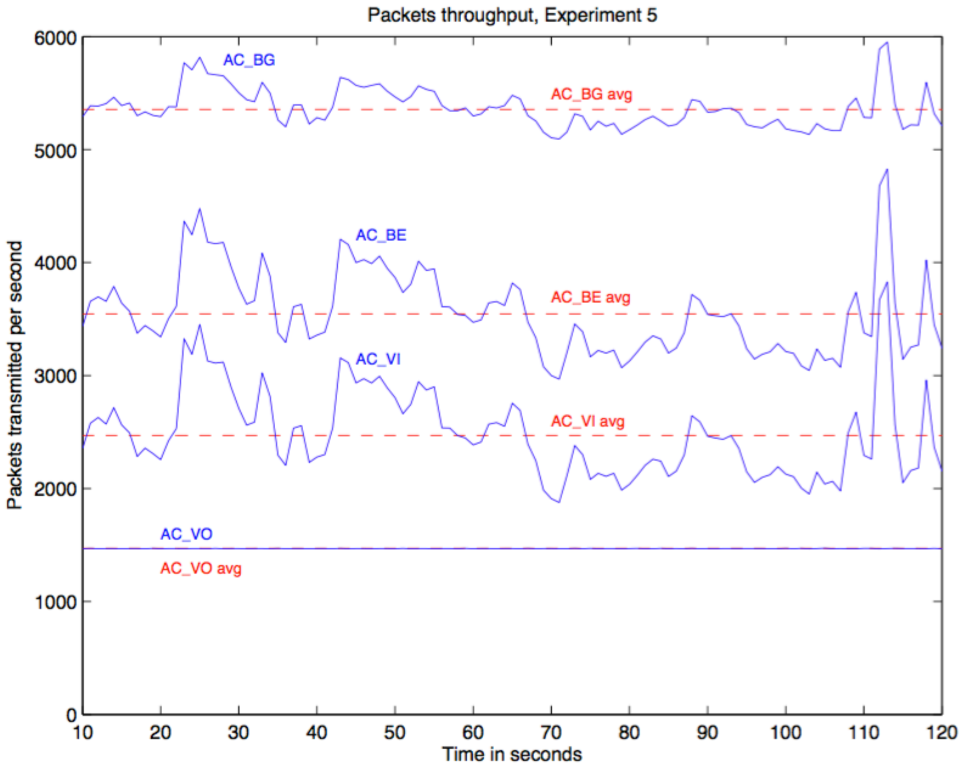

Fig. 6.11: Simulation results and traffic averages for 20 VoIP, 10 Audio and 10 Video streams, 10 TCP flows, 5 UDP flows and a mix of 15 TCP and 5 UDP flows

wireless network with a packet size of 256 bytes and a speed of 200 Kb managed by the AC_3s.

Each subsequent configuration adds one more VoIP call to those managed by the AC_0s and one additional audio and video bidirectional flow to those managed by the AC_0s and AC_1s respectively.

VoIP phone calls are replicated in the network by ten CBR flows from the wired nodes to the wireless mobile nodes and ten CBR flows from the mobile nodes to the wired nodes. Video traffic is simulated using an extended and refined version of Evalvid [198] [199]. The audio and video streams are each 140 seconds long.

Traffic volumes were chosen in order to explore the upper bound on the real time traffic that can be delivered by the network while, at the same time, guaranteeing the minimum amount of traffic needed to maintain the best effort and background traffic access categories.

In the first set of simulations the IEEE802.11ac wireless protocol is implemented. The eQoS controller and the EDCA [35] method are also used. For each of the six configurations the Access Point estimates the eQoS directly by measuring the number of packets dropped at the queue, the transmission delay for each packet and the jitter. The jitter metric used is the delay between two packet transmissions. Delay and jitter checks are performed at the access point's output queue and in the transmission buffer of each node.

The eQoS controller performs two operations: it estimates the eQoS and determines packet transmission delays and jitter. The delay and jitter are not used to actively drop the packets in the experiments performed with the original system, i.e. with IEEE802.11ac with EDCA only,. Each node in the wireless network stores the EWMA estimate of the number of packets dropped in a one second sliding window. This is done at the MAC layer and in the queues. When this variable is updated it triggers an immediate re-calculation of the eQoS and QoE.

There are three ways that packet drops can occur. Firstly, a packet is dropped if the queue buffer is full and there is no space to store it. Secondly, if a packet has exceeded its transmission or packet delay threshold then it is dropped at the queue output. Thirdly a packet is dropped if it has exceeded the number of acceptable retransmissions after a collision. The number of packets dropped is decreased when an acknowledgement packet is received as this confirms that a packet has been successfully transmitted.

Two new features were added to the ns-2 simulator. The first is the association of a priority with TCP acknowledgements. By default ns-2 associates a priority of zero with these, causing them to be managed by AC_0 instead of by the relevant AC_2s or AC_3s. The second is the addition of a timestamp to the UDP packet headers to facilitate delay calculations at the node.

In the next subsections the QQM algorithm and its components are tested using the scenarios described above.

6.4.1 The Contention Window Controller

The first experimental configuration uses the IEEE802.11ac [6] protocol with EDCA [35]. The system also measures the eQoS but the values obtained are not used in the active eQoS controller; therefore packets exceeding the maximum jitter and delay accepted to deliver the service with a high quality are not dropped. This means that the system will overestimate the QoE.

The contention window controller is implemented with the IEEE802.11ac protocol. Table 5.2 is used to set the initial contention windows for the AC_0s and AC_1s depending on the measured eQoS. When a collision occurs the contention window is doubled. The AC_2s and AC_3s are not considered by the contention window controller.

As mentioned in section 5.5.1.2 a large CW size is used when the quantitative score exceeds 4 and a small CW size is used when the score is below 3. This has two main advantages: first of all it reduces the number of collisions and secondly it gives to a clear definition of the quality of a flow. The smaller CW value means that the system tries to transmit each packet as quickly as it can. This will give rise to an increase in the quality. If traffic conditions mean that a large CW size is used then this will increase the packet transmission delay. This, in turn, increases the likelihood of packet drops and so the quality is reduced.

Figure 6.12 compares configurations 0 and 1 using IEEE802.11ac with the EDCA channel access method, shown in red, with the corresponding configuration where the contention window controller is deployed, shown in blue. The left hand figures show the number of packets involved in collisions per second. The right hand side figures show the packets received per second.

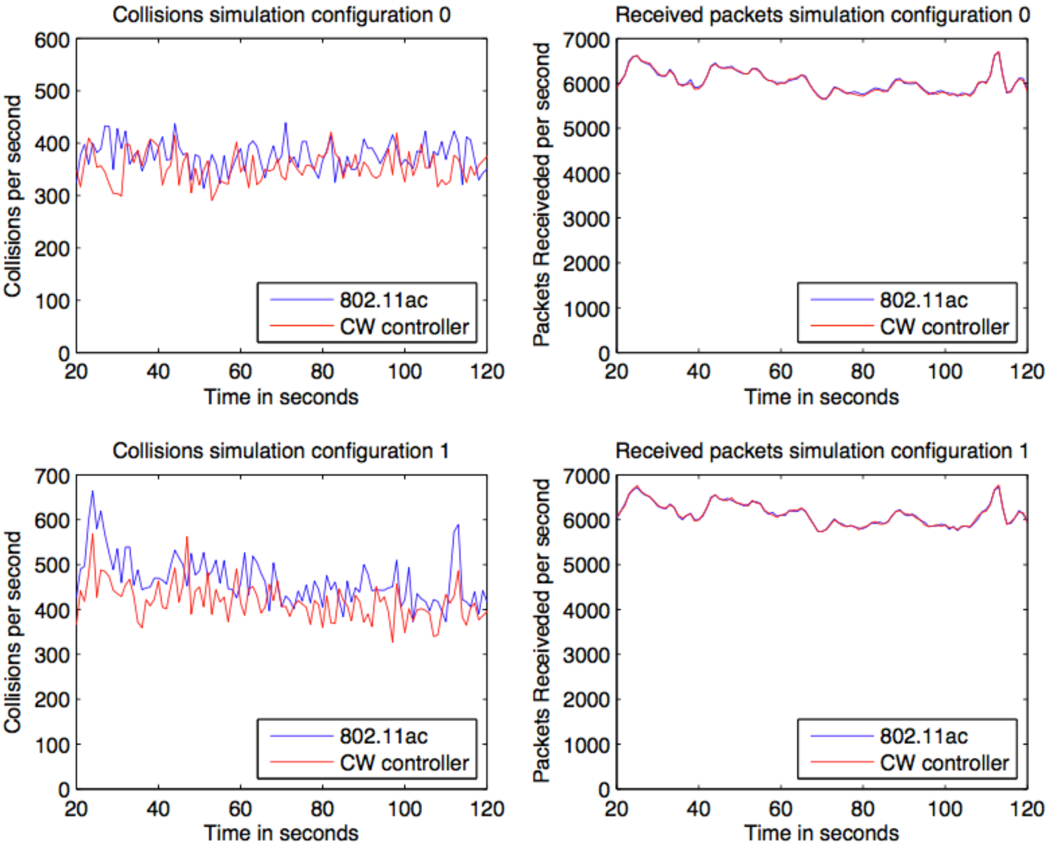

Fig. 6.12: Simulation results for configurations 0 and 1 comparing the number of collisions per second and the number of packets received per second for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of the contention window controller

The experiments are repeated for configurations 2 and 3 and the results shown in figure 6.13. The results for configurations 4 and 5 are shown in figure 6.14.

Figures 6.12, 6.13 and 6.14 show a reduction in the number of packets involved in collisions per second when the CW controller is used. By contrast, the number of packets transmitted per second is practically the same. This is due to the large volume of traffic on the network. The reduction in collisions becomes more evident as the number of VoIP, audio and video streams grows.

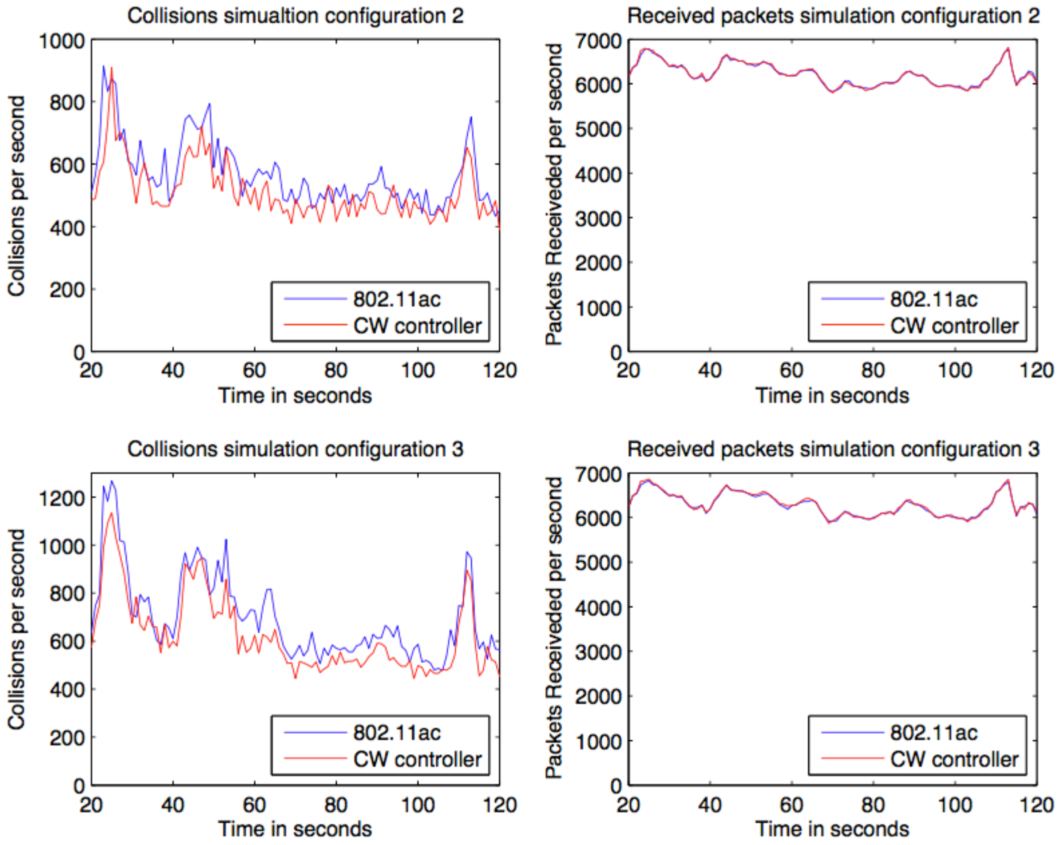

Fig. 6.13: Simulation results for configurations 2 and 3 comparing the number of collisions per second and the number of packets received per second for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of the contention window controller

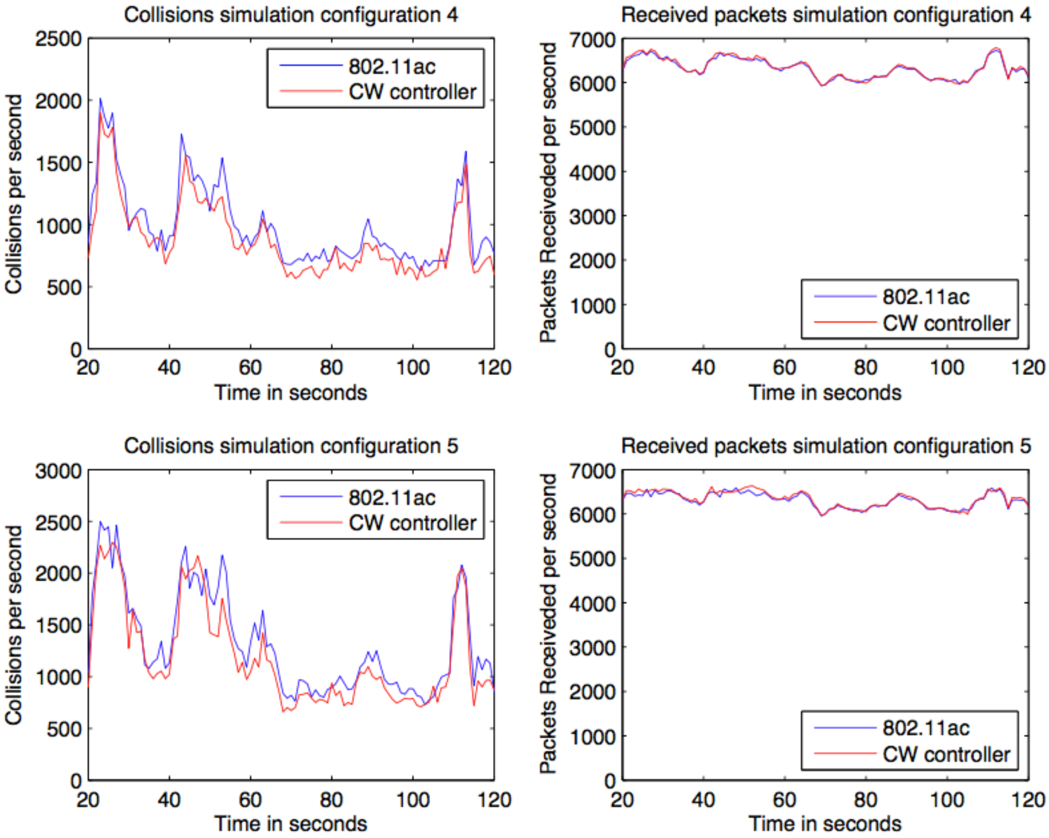

Fig. 6.14: Simulation results for configurations 4 and 5 comparing the number of collisions per second and the number of packets received per second for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of the contention window controller

6.4.2 The Active eQoS Controller

The active eQoS controller drops packets that exceed the maximum thresholds for delay and jitter. As described in section 5.5.1.1, packets from a real time stream that exceed the maximum delay and jitter threshold values are transmitted across the wireless network despite that fact that they are of no use to the destination node. Therefore, in an ideal world, these packets should be dropped before they are transmitted so that they do not occupy the available wireless channel with unnecessary traffic.

The active eQoS controller has a direct effect on the estimated eQoS values. If the active eQoS controller is not deployed then all enqueued packets are transmitted and the eQoS is not affected. When the active eQoS controller is implemented, packets that exceed the maximum thresholds for delay and jitter are dropped before transmission across the wireless network.

The packet delay and packet transmission delay calculations are performed at the transmission buffer, because packets that exceed these delays should be dropped before they are transmitted. The active eQoS controller uses the packet start time in the UDP header to estimate the delay. The jitter, or delay variation [45], is calculated at the transmission buffer by looking at the variation in delay between packets from the same flow. When an ACK is received or before a packet is dropped the transmission time is recorded so that it can be used in the calculation of the jitter associated with any subsequent packet from the same flow. Delay checks at the queue input are performed by the queue management controller.

The acceptable value for the jitter was chosen using [45]. The upper jitter limit for two video packets transmitted in sequence was set to be 0.040 seconds. For two audio and VoIP packets this is set to 0.025 seconds in order to conform to the video conferencing standard. The maximum delay is set to 0.2 seconds for all real time traffic packets.

In figures 6.15, 6.16 and 6.17 the performance of IEEE802.11ac with EDCA is compared with that of the same system where both the contention window and active eQoS controllers are included. As described in the previous subsection the results for IEEE802.11ac alone are shown in blue, while the results for the system that includes both the contention window and the active eQoS controllers are shown in red.

The number of packets that collide per second is lower when the active eQoS controller is used. The differences between this system and IEEE802.11ac alone is more noticeable then the case where just the contention window controller is deployed. This is because some packets are now dropped before transmission as they have exceeded the maximum limits for delay and jitter.

6.4.3 The Queue Management Controller

The queue management controller estimates the theoretical maximum queue size and automatically drops packets the exceed this threshold. As mentioned in section 5.5.1.3, the maximum queue size is variable and it is found using the average time needed to transmit a packet. The transmission time is checked at the queue output and when the MAC Layer acknowledgement is received by the transmitting station. The average delay accumulated by arriving packets is calculated at the queue input. An EWMA is used to smooth differences between the average delay and any delay peaks, the system then drops packets whose average delay exceeds 80% the theoretical maximum queue size.

Simulations have been performed using implementations of all three controllers: the contention window controller, the eQoS controller and the queue management controller.

Figures 6.18, 6.19, 6.20, 6.21, 6.22 and 6.23 show Queue 0 ad Queue 1 during the simulation of configurations 0, 1, 2, 3, 4 and 5 respectively.

In each figure the red line is the theoretical maximum queue size, the black line shows the fixed maximum queue length for the simulations, that is 250 packets. The blue line is the instantaneous queue length. In the simulations of configurations 0, 1, 2 and 3, the theoretical maximum queue length differs significantly from the real queue length and it is concluded that the queue management controller is not affecting the simulations. On the other hand as the traffic grows in configurations 4 and 5, the theoretical maximum queue length is less than the maximum queue length and the instantaneous queue length and so it is becomes necessary to drop some packets.

It is important to note that some packets that are successfully transmitted by the system with

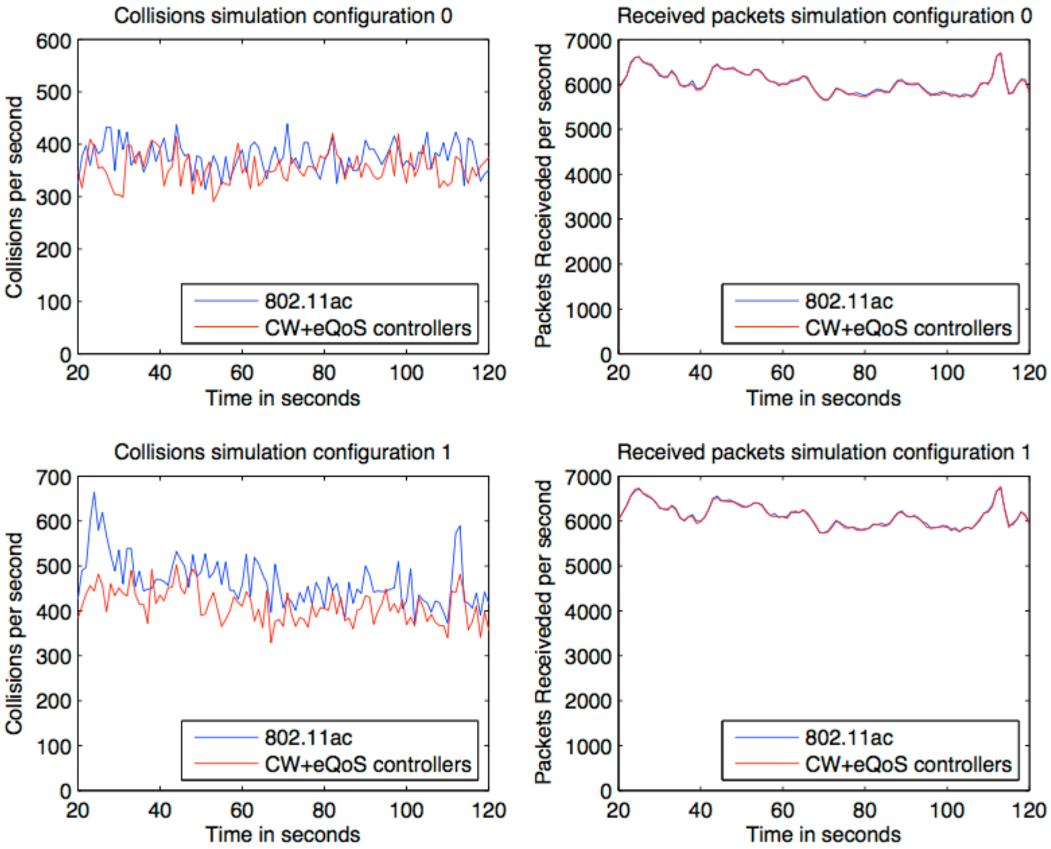

Fig. 6.15: Simulation results for configurations 0 and 1 comparing the number of collisions per second and the number of packets received per second for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of both the contention window and active eQoS controllers

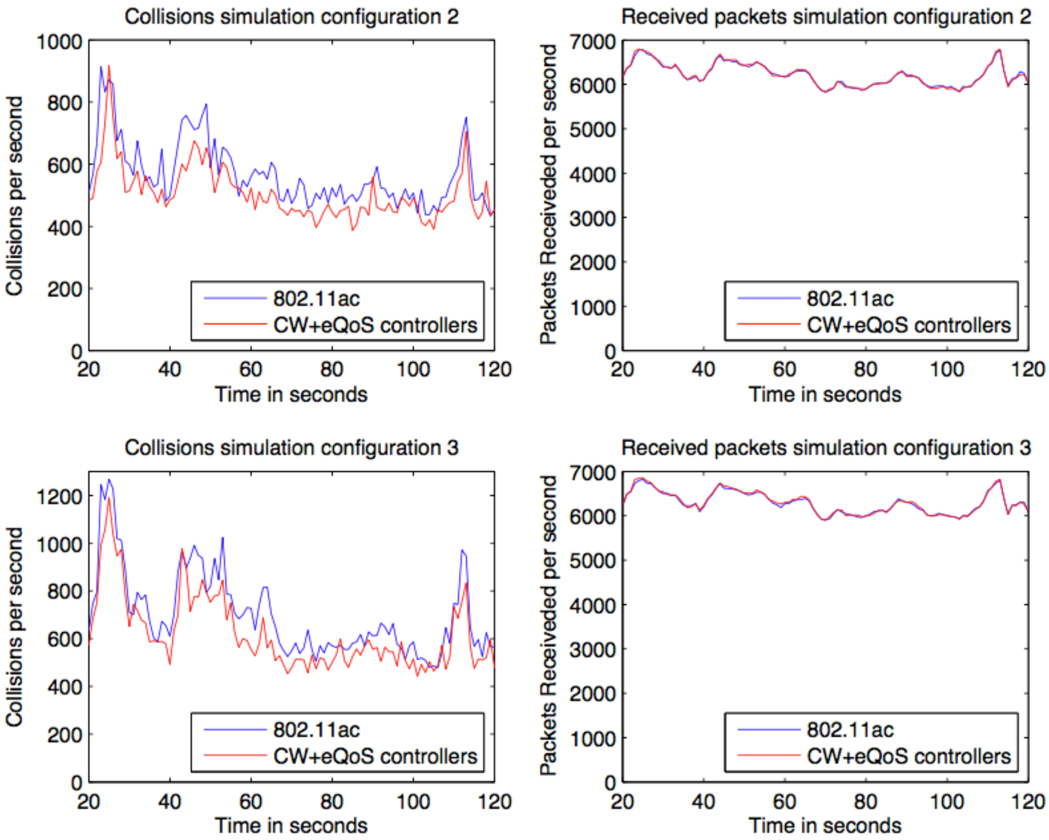

Fig. 6.16: Simulation results for configurations 2 and 3 comparing the number of collisions per second and the number of packets received per second for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of both the contention window and active eQoS controllers

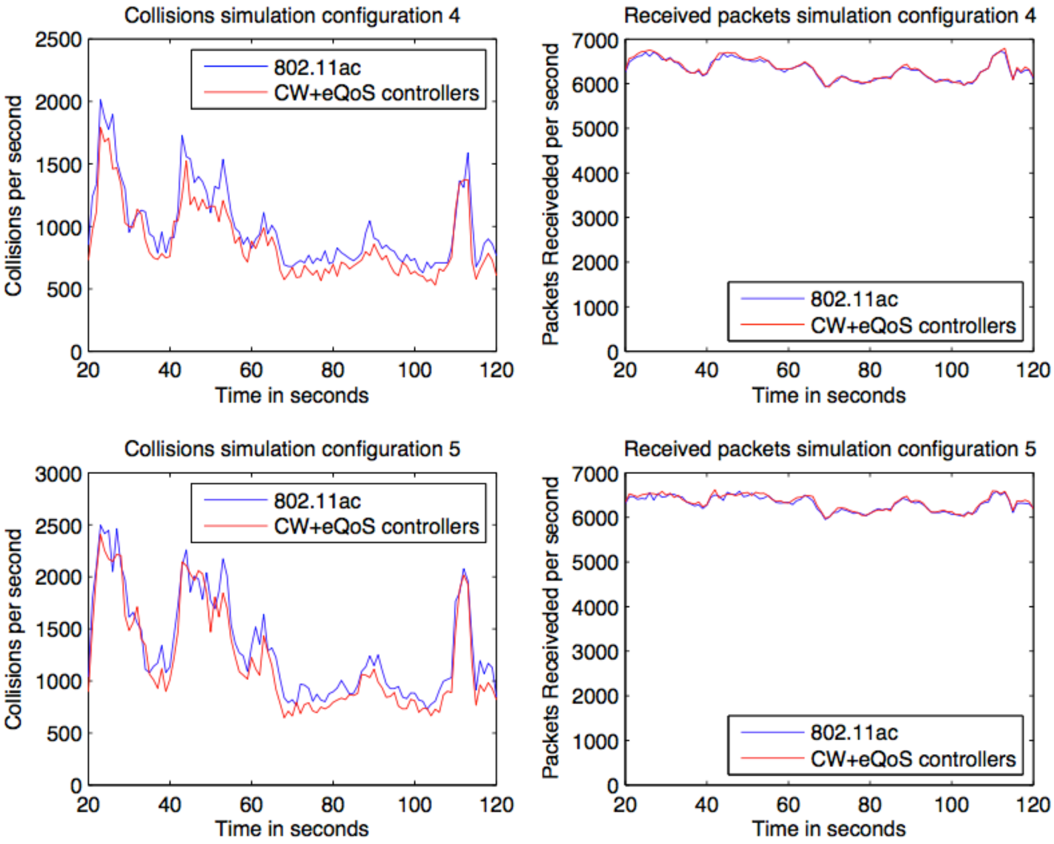

Fig. 6.17: Simulation results for configurations 4 and 5 comparing the number of collisions per second and the number of packets received per second for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of both the contention window and active eQoS controllers

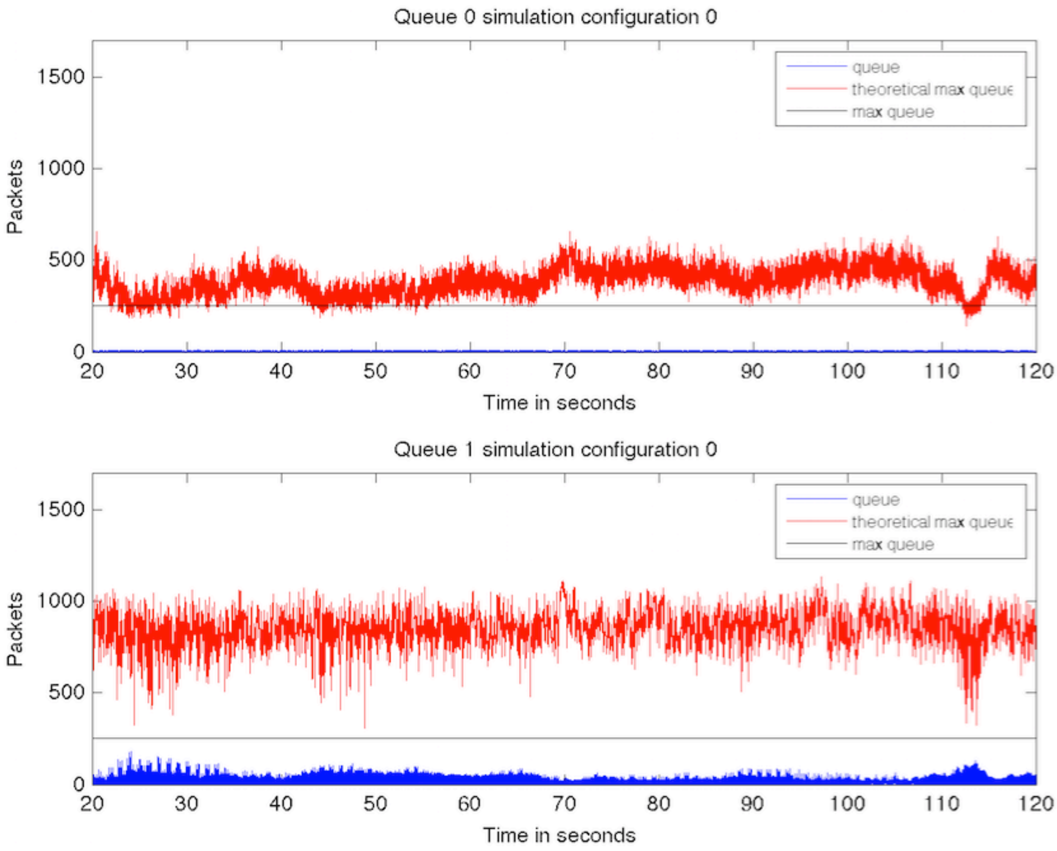

Fig. 6.18: Real and Virtual lengths of Queue 0 and Queue 1 for the Configuration 0 Simulation

IEEE802.11ac and EDCA alone are dropped when all three controllers are implemented. The eQoS measured in the system without the three controllers is not affected by the transmission of these packets as they do not enhance the QoE of the service being delivered.

6.4.4 The QQM Controller

In this subsection the three controllers evaluated above are brought together using the QQM algorithm. This algorithm manages the interactions between the three controllers. It is configured to work in the worst case, i.e. with minimal changes to the traffic, to show how QQM can have a significant impact on the provision of high quality services.

QQM is implemented at the queue inputs and, as described in section 5.5, it operates and

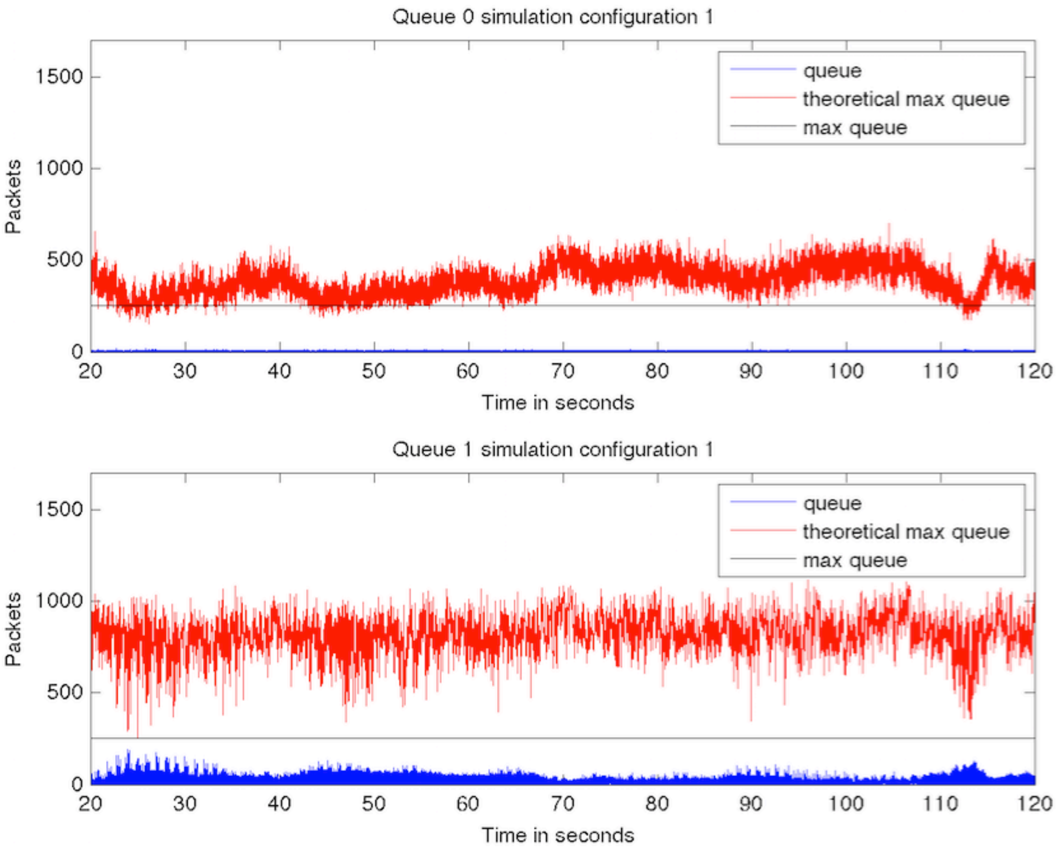

Fig. 6.19: Real and Virtual lengths of Queue 0 and Queue 1 for the Configuration 1 Simulation

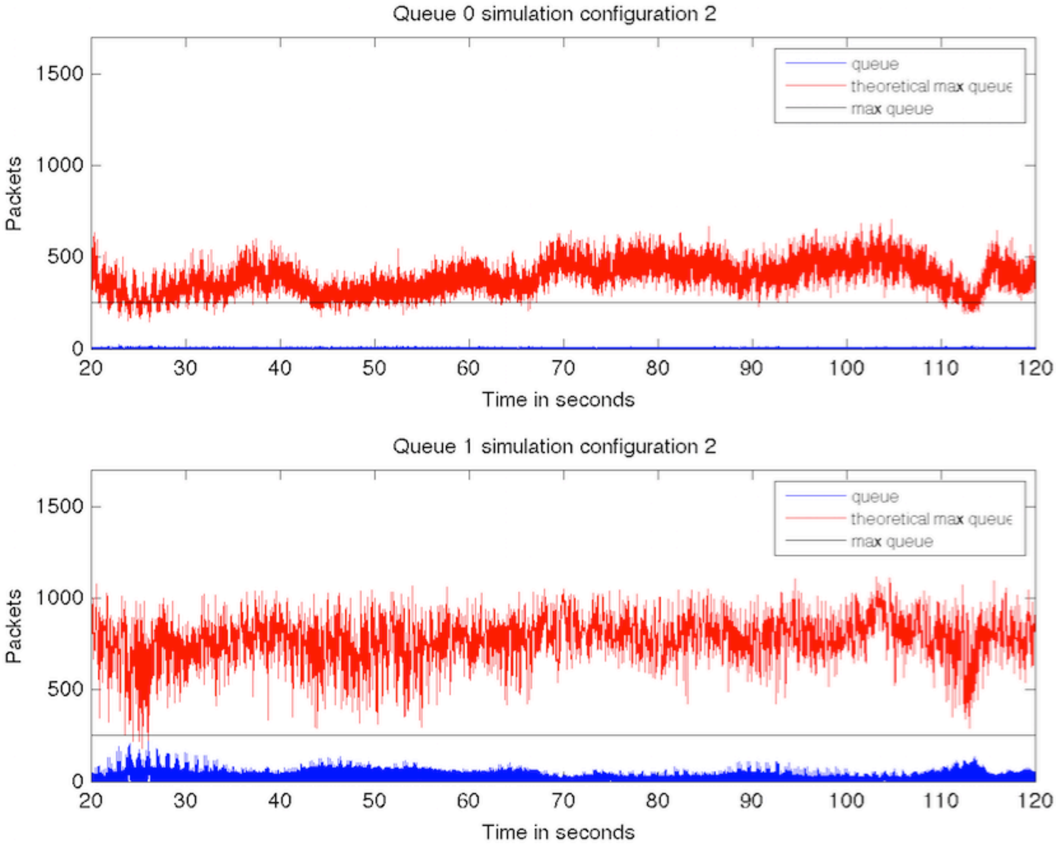

Fig. 6.20: Real and Virtual lengths of Queue 0 and Queue 1 for the Configuration 2 Simulation

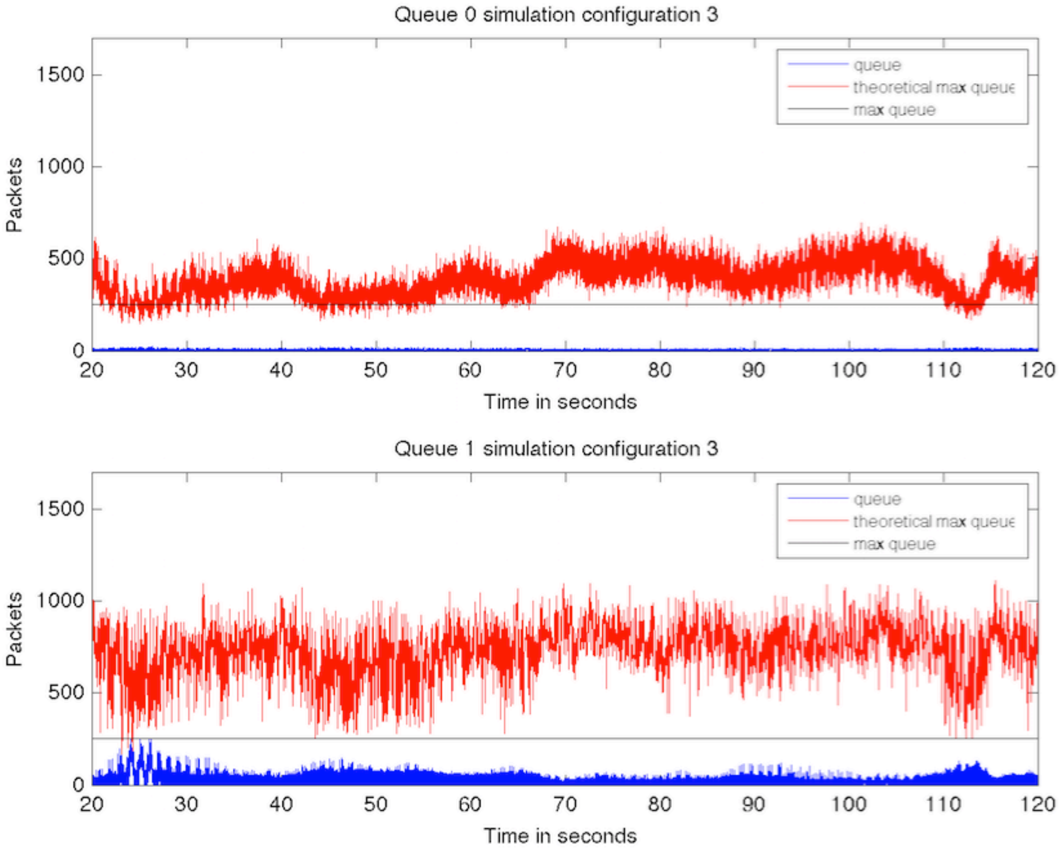

Fig. 6.21: Real and Virtual lengths of Queue 0 and Queue 1 for the Configuration 3 Simulation

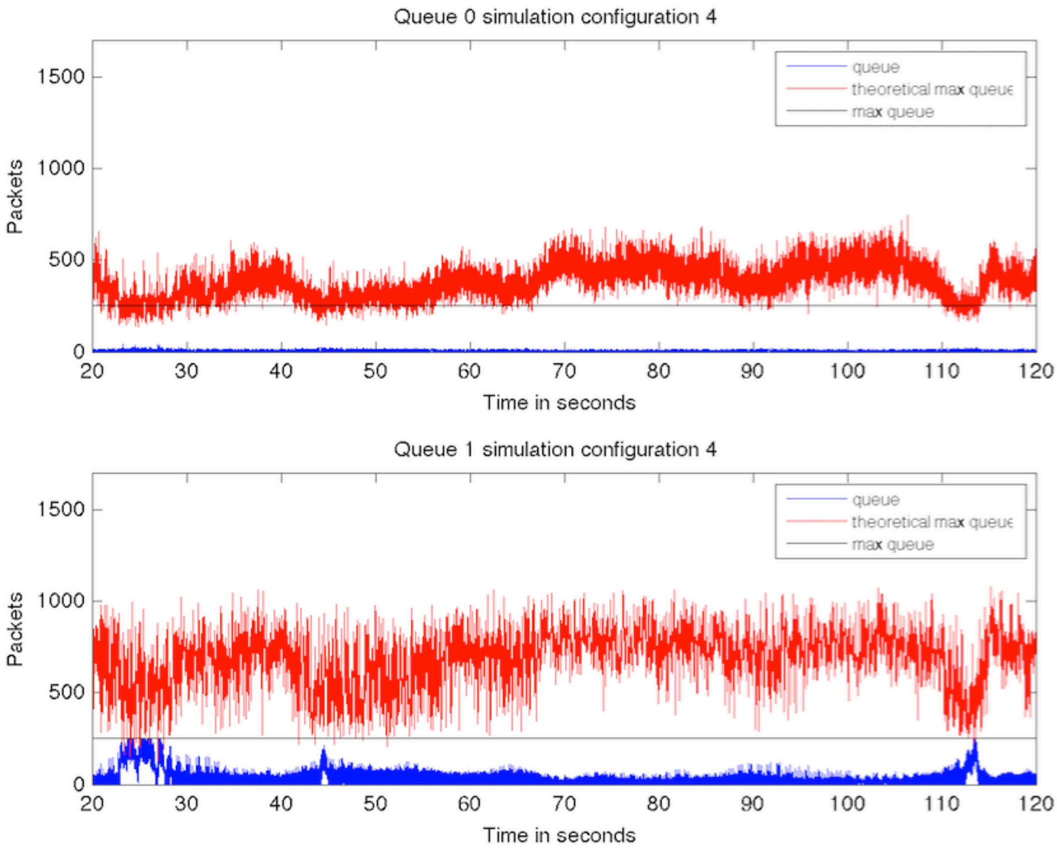

Fig. 6.22: Real and Virtual lengths of Queue 0 and Queue 1 for the Configuration 4 Simulation

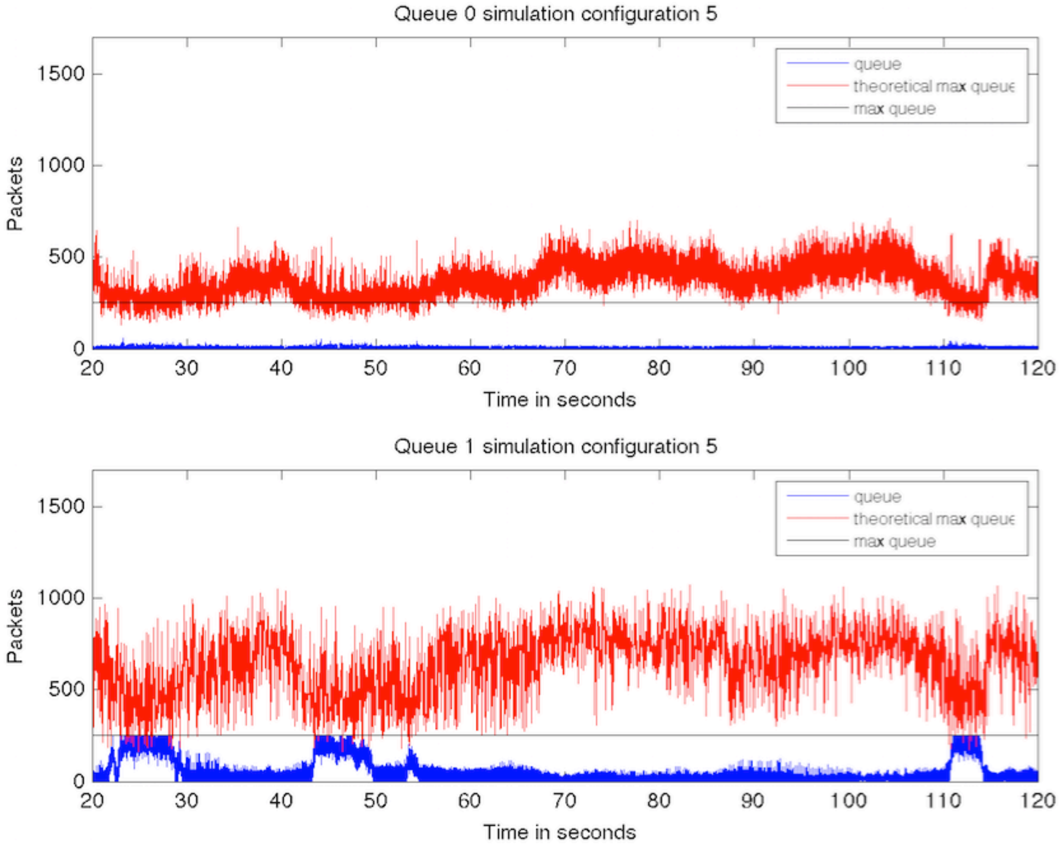

Fig. 6.23: Real and Virtual lengths of Queue 0 and Queue 1 for the Configuration 5 Simulation

interacts with the controllers through the eQoS metric. To assist the reader in their understanding of the QQM rules and of the graphs presented below, the eQoS estimates are translated onto a scale that is comparable with that used for the MOS score. This is indicated on the graphs as QoE.

For each packet that arrives at the node, QQM checks the QoE for the associated flow. If the QoE is less than the the acceptable threshold value of 3 and the previous two packets have been dropped, then the flow is marked. When the QoE for a marked flow stays below the threshold value of 3 for one second the flow is dropped. QQM then drops any other flows associated with the marked flow; for example, in the case of a phone call the flow in the opposite direction will also be dropped. This can be easily achieved using a control packet that is generated by the node that dropped the original flow. To limit the number of dropped flows, QQM drops only flows associated with the same service; e.g. VoIP, audio or video. A more aggressive choice would be to drop the entire service; for example, to drop an entire video conference if the audio in one direction is not working. These choices can be made by the TELCO depending on how they market their services.

When a flow is dropped the node does not drop any other flow for the next five seconds. This time interval is needed for the system to adapt to the loss of the dropped flow. The time interval settings adjust QQMs tolerance to both the magnitude and duration of quality loss events. For the evaluation presented below, the system is set to the worst case with long time intervals.

Simulations of configurations 0, 1, 2 and 3 were chosen to have a sufficient amount of traffic to enable the three controllers to work independently, but without sufficient loss to cause QQM to have an impact on the system. Configurations 4 and 5 have flows with a loss in quality that will be detected by QQM and cause it to take action .

In the configuration 4 simulation only a VoIP flow is dropped at the beginning of the simulation, therefore the subsequent analysis focuses on the configuration 5 simulation.

Table 6.7 summarises the VoIP flows in progress during the simulation of configuration 5. The table reports the name and type of service in operation, the source node, the destination node and the flowID, that is a unique number that identifies the flow in the packet header. In-

Service	Source	Destination	flowID
VoIP call 1	wired node 1	wireless node 1	31
VoIP call 1	wireless node 1	wired node 1	41
VoIP call 2	wired node 7	wireless node 7	37
VoIP call 2	wireless node 7	wired node 7	47
VoIP call 3	wired node 11	wireless node 11	311
VoIP call 3	wireless node 11	wired node 11	411
VoIP call 4	wired node 17	wireless node 17	317
VoIP call 4	wireless node 17	wired node 17	417
VoIP call 5	wired node 21	wireless node 21	321
VoIP call 5	wireless node 21	wired node 21	421
VoIP call 6	wired node 27	wireless node 27	327
VoIP call 6	wireless node 27	wired node 27	427
VoIP call 7	wired node 31	wireless node 31	331
VoIP call 7	wireless node 31	wired node 31	431
VoIP call 8	wired node 37	wireless node 37	337
VoIP call 8	wireless node 37	wired node 37	437
VoIP call 9	wired node 41	wireless node 41	341
VoIP call 9	wireless node 41	wired node 41	441
VoIP call 10	wired node 47	wireless node 47	347
VoIP call 10	wireless node 47	wired node 47	447

Table 6.7: Summary of VoIP Flows in Configuration 5

stantaneous variations in QoE for each VoIP flow are reported in figures 6.24, 6.25, 6.26, 6.27 and 6.28.

In figures 6.24, 6.25, 6.26, 6.27 and 6.28 configuration 5 is simulated for the original system, i.e. for IEEE802.11ac with EDCA only, and for the enhanced system where the QQM algorithm is deployed. Results for the former are on the right hand side of each figure; while those for the latter are on the left side.

It can be seen that, in general, the quality of flows from the wireless to the wired network are most affected by the introduction of QQM. Drops in QoE are smoothed and reduced in amplitude

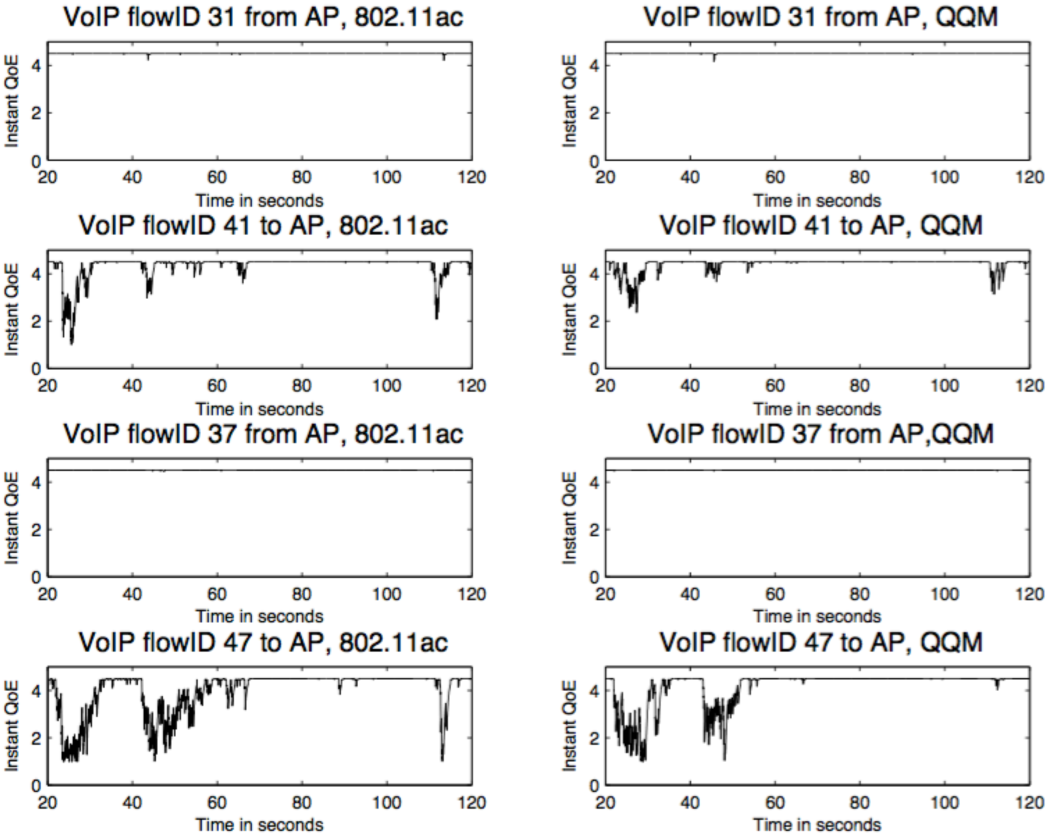

Fig. 6.24: Comparison of the QoE of VoIP Flows at Wireless Nodes 1 and 7 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

by the QQM algorithm through the actions and interactions of the three individual controllers and the dropping of a few flows from the system. After 113 seconds the flow with ID number 411 is dropped along with the related flow with ID 311 and VoIP call 3 is terminated.

A similar comparison between a simulation of IEEE802.11ac with EDCA and the same system with the use of QQM are repeated for video and audio flows. Table 6.8 summarises the details of the unidirectional video and audio flows used in the simulation of configuration 5.

Figures 6.29, 6.30, 6.31, 6.32 and 6.33 show the comparison between the QoE of the original system, i.e. for IEEE802.11ac with EDCA only, and the same system with the addition of QQM.

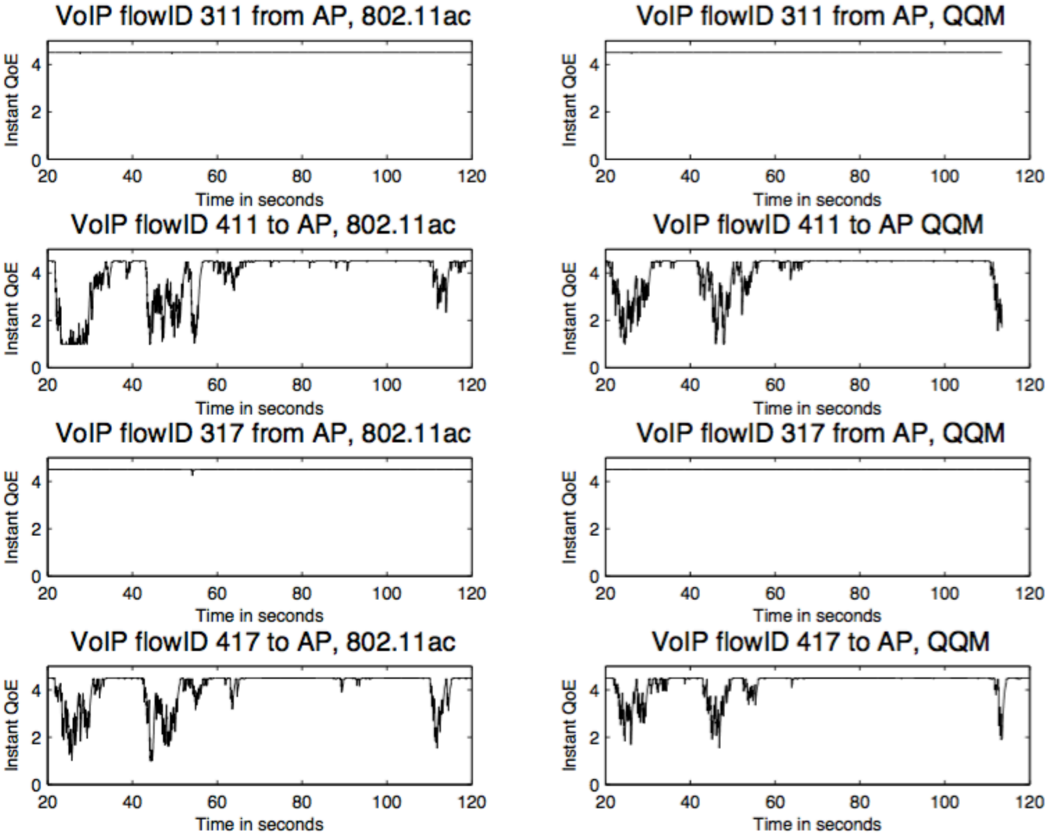

Fig. 6.25: Comparison of the QoE of VoIP Flows at Wireless Nodes 11 and 17 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

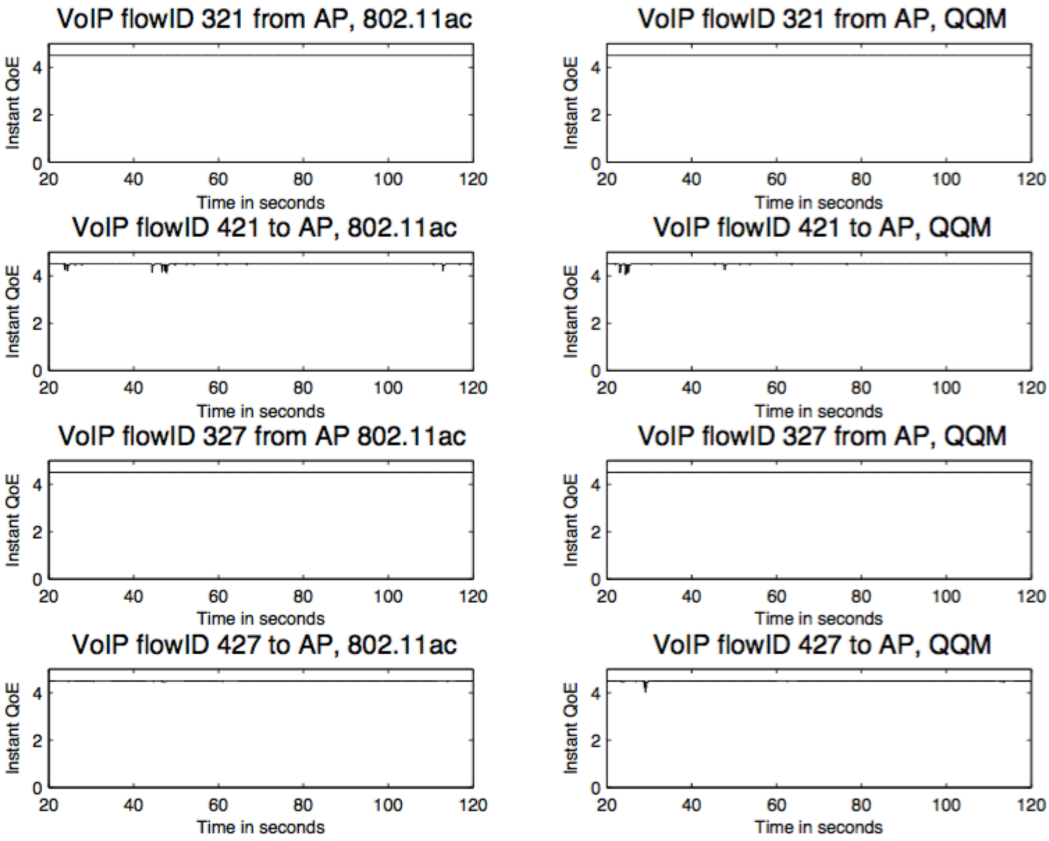

Fig. 6.26: Comparison of the QoE of VoIP Flows at Wireless Nodeses 21 and 27 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

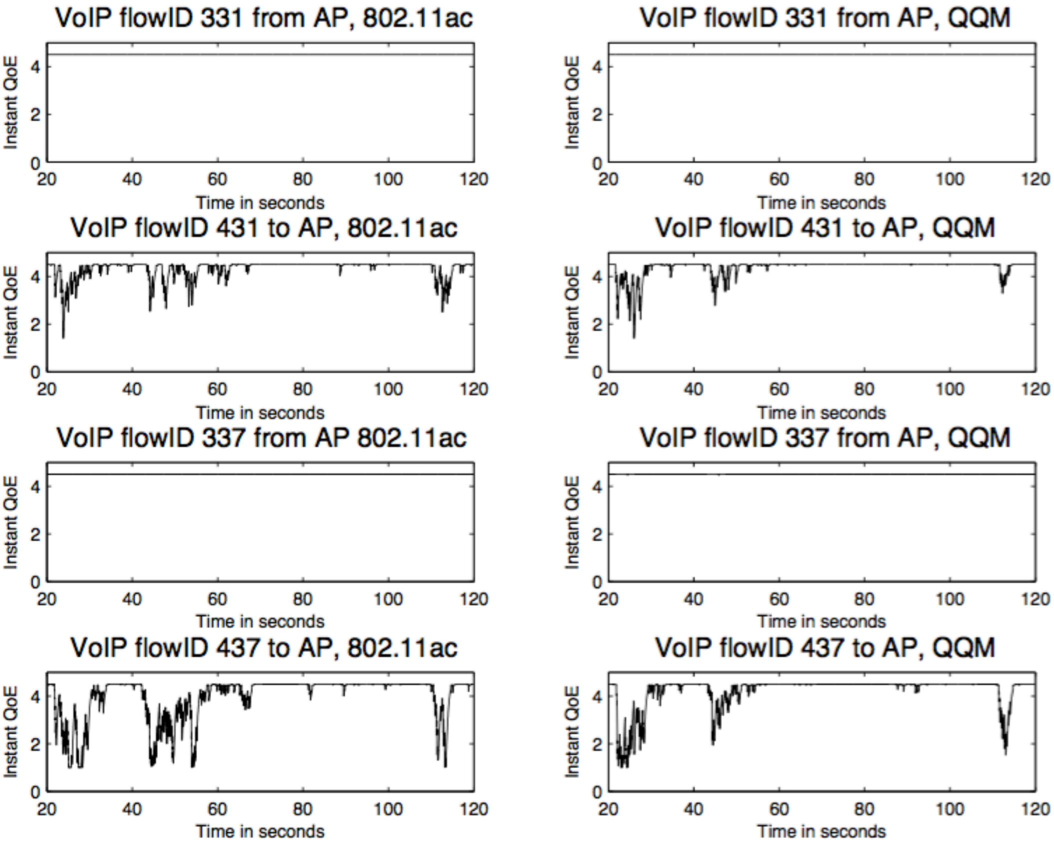

Fig. 6.27: Comparison of the QoE of VoIP Flows at Wireless Nodes 31 and 37 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

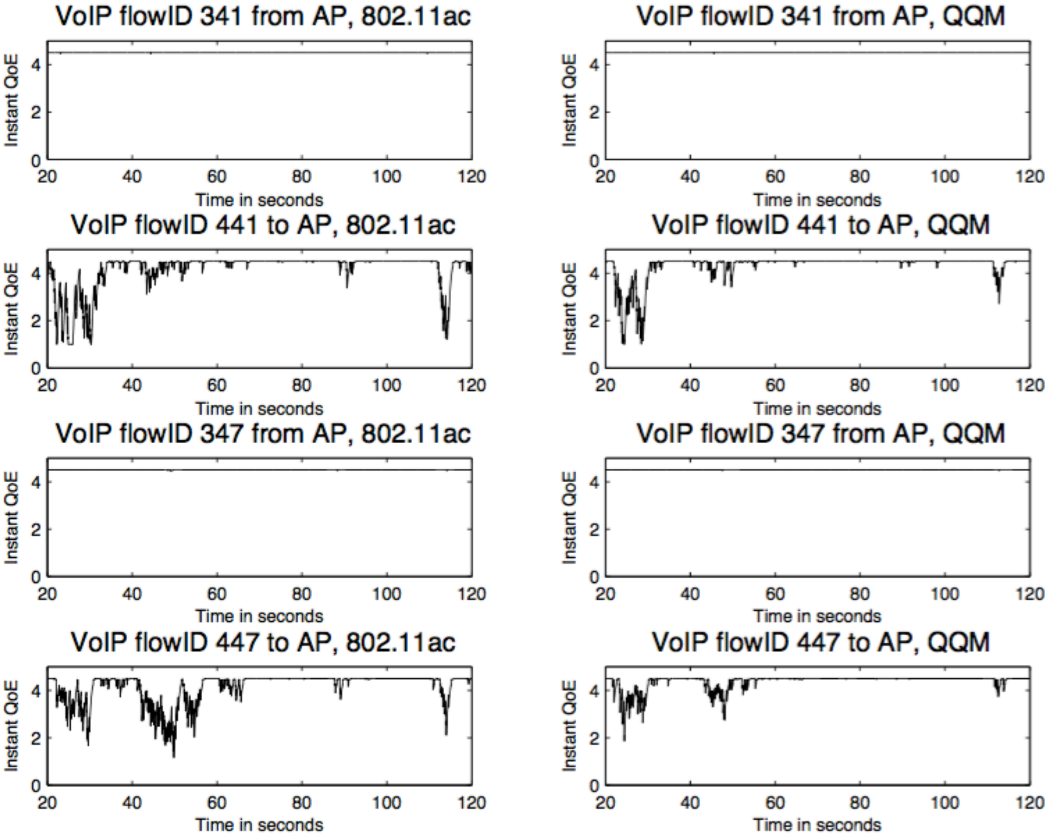

Fig. 6.28: Comparison of the QoE of VoIP Flows at Wireless Nodes 41 and 47 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

Service	Source	Destination	flowID
Audio 1	wired node 0	wireless node 0	20
Video 1	wired node 0	wireless node 0	10
Audio 2	wired node 8	wireless node 8	28
Video 2	wired node 8	wireless node 8	18
Audio 3	wired node 10	wireless node 10	210
Video 3	wired node 10	wireless node 10	110
Audio 4	wired node 18	wireless node 18	218
Video 4	wired node 18	wireless node 18	118
Audio 5	wired node 20	wireless node 20	220
Video 5	wired node 20	wireless node 20	120
Audio 6	wired node 28	wireless node 28	228
Video 6	wired node 28	wireless node 28	128
Audio 7	wired node 30	wireless node 30	230
Video 7	wired node 30	wireless node 30	130
Audio 8	wired node 38	wireless node 38	238
Video 8	wired node 38	wireless node 38	138
Audio 9	wired node 40	wireless node 40	240
Video 9	wired node 40	wireless node 40	140
Audio 10	wired node 48	wireless node 48	248
Video 10	wired node 48	wireless node 48	148

Table 6.8: Summary of VoIP flows in configuration 5

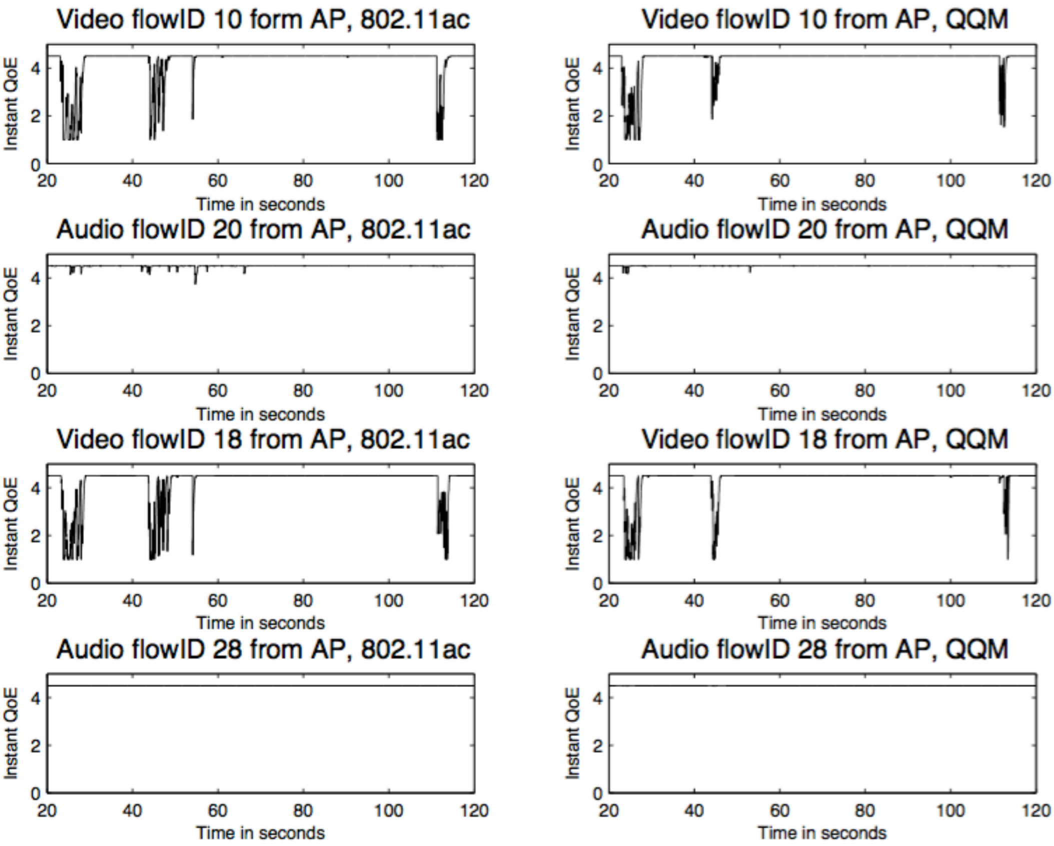

Fig. 6.29: Comparison of QoE of audio and video unidirectional flows at Wireless Nodes 0 and 8 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

In figure 6.31 the flow with ID 128 is dropped by the QQM algorithm after 26 seconds. Audio quality is not affected in the unidirectional flows from the wired to the wireless network.

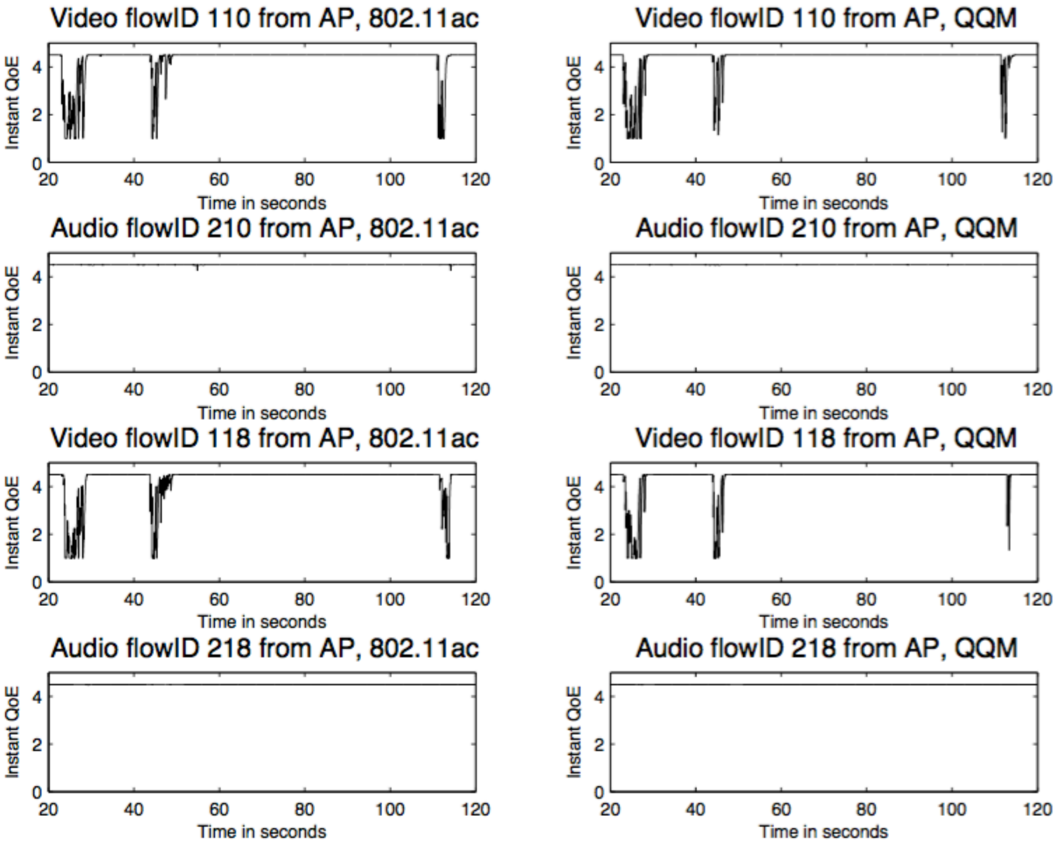

Fig. 6.30: Comparison of QoE of audio and video unidirectional flows at Wireless Nodes10 and 18 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

Tables 6.9 and 6.10 summarise the details of the bidirectional video and audio flows used in the simulation of configuration 5.

Figures 6.34, 6.35, 6.36, 6.37 and 6.38 show the comparison between the QoE of the original system, i.e. for IEEE802.11ac with EDCA only, and the same system with the addition of QQM for table 6.9. In this case no flows are dropped by the QQM algorithm.

Figures 6.39, 6.40, 6.41, 6.42 and 6.43 show the comparison between the QoE of the original system, i.e. for IEEE802.11ac with EDCA only, and the same system with the addition of QQM for table 6.10. In this case the flows with ID numbers 246 and 247 have been dropped by the

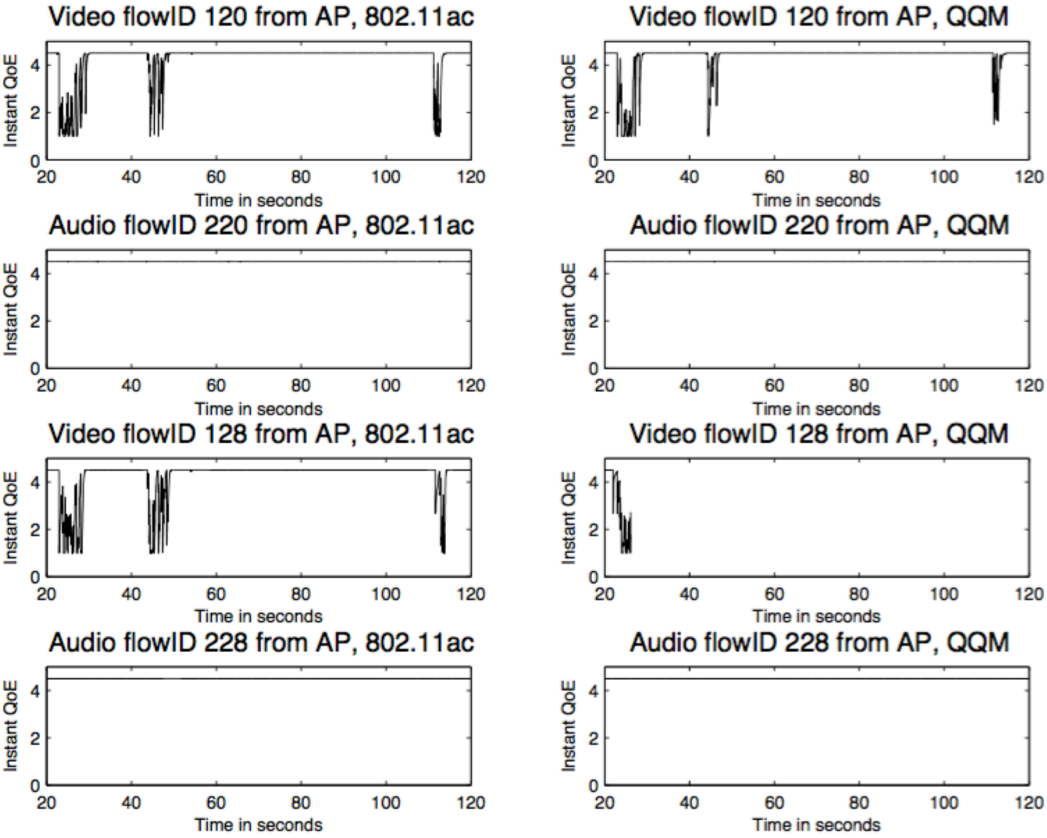

Fig. 6.31: Comparison of QoE of audio and video unidirectional flows at Wireless Nodes 20 and 28 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

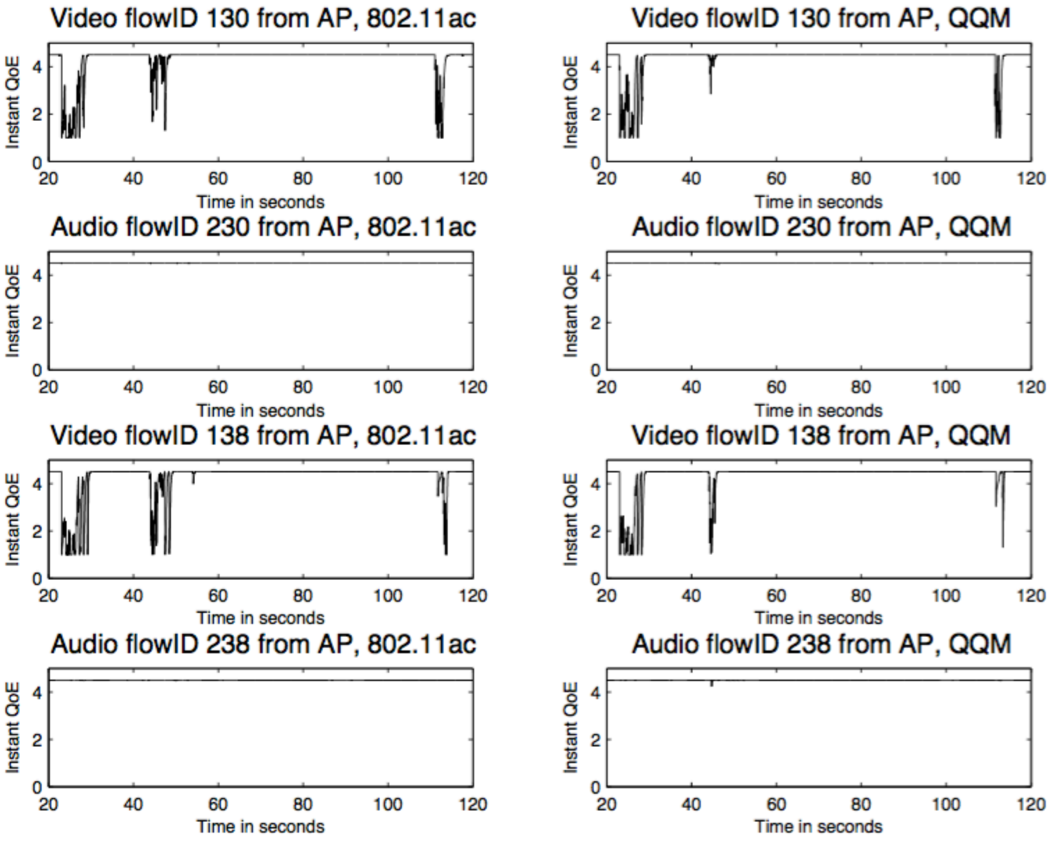

Fig. 6.32: Comparison of QoE of audio and video unidirectional flows at Wireless Nodes30 and 38 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

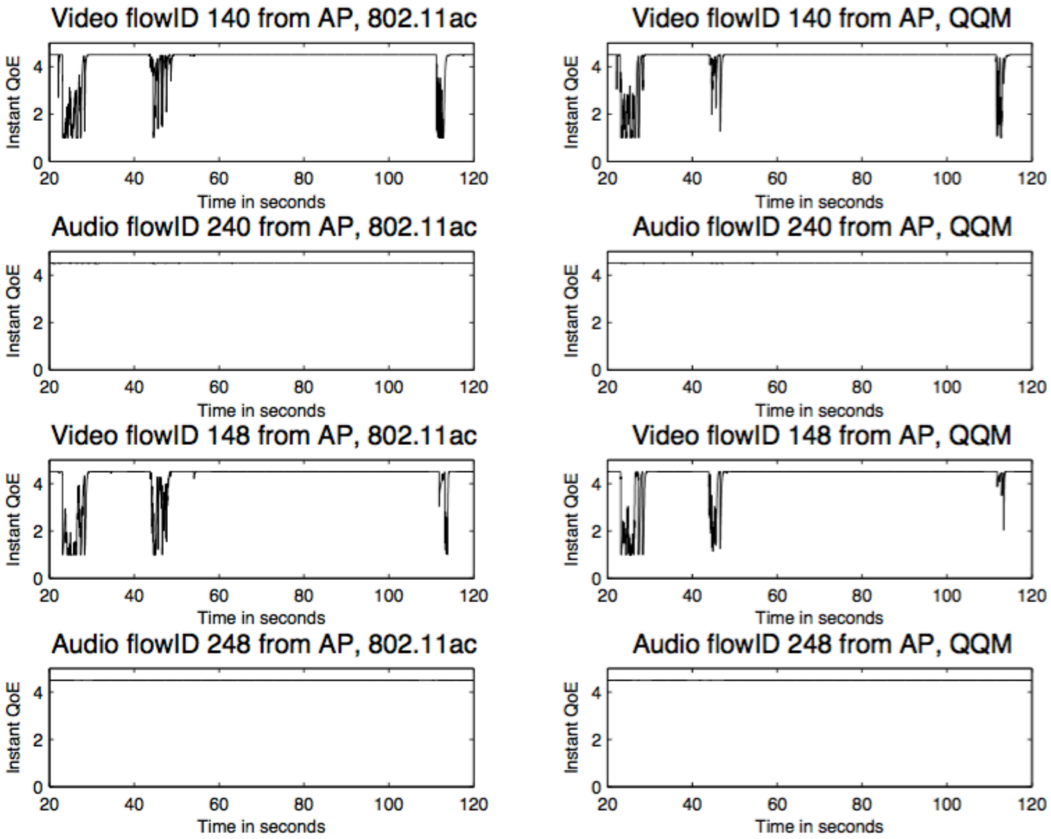

Fig. 6.33: Comparison of QoE of audio and video unidirectional flows at Wireless Nodes 40 and 48 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

Service	Source	Destination	flowID
Audio 11	wired node 6	wireless node 6	26
Video 11	wired node 6	wireless node 6	16
Audio 11	wireless node 6	wired node 6	27
Video 11	wireless node 6	wired node 6	17
Audio 12	wired node 9	wireless node 9	29
Video 12	wired node 9	wireless node 9	19
Audio 12	wireless node 9	wired node 9	211
Video 12	wireless node 9	wired node 9	111
Audio 13	wired node 16	wireless node 16	216
Video 13	wired node 16	wireless node 16	116
Audio 13	wireless node 16	wired node 16	217
Video 13	wireless node 16	wired node 16	117
Audio 14	wired node 19	wireless node 19	219
Video 14	wired node 19	wireless node 19	119
Audio 14	wireless node 19	wired node 19	221
Video 14	wireless node 19	wired node 19	121
Audio 15	wired node 26	wireless node 26	226
Video 15	wired node 26	wireless node 26	126
Audio 15	wireless node 26	wired node 26	227
Video 15	wireless node 26	wired node 26	127

Table 6.9: Summary of VoIP flows in configuration 5

Service	Source	Destination	flowID
Audio 16	wired node 29	wireless node 29	229
Video 16	wired node 29	wireless node 29	129
Audio 16	wireless node 29	wired node 29	231
Video 16	wireless node 29	wired node 29	131
Audio 17	wired node 36	wireless node 36	236
Video 17	wired node 36	wireless node 36	136
Audio 17	wireless node 36	wired node 36	237
Video 17	wireless node 36	wired node 36	137
Audio 18	wired node 39	wireless node 39	239
Video 18	wired node 39	wireless node 39	139
Audio 18	wireless node 39	wired node 39	241
Video 18	wireless node 39	wired node 39	141
Audio 19	wired node 46	wireless node 46	246
Video 19	wired node 46	wireless node 46	146
Audio 19	wireless node 46	wired node 46	247
Video 19	wireless node 46	wired node 46	147
Audio 20	wired node 49	wireless node 49	249
Video 20	wireld node 49	wireless node 49	149
Audio 20	wireless node 49	wired node 49	251
Video 20	wireless node 49	wired node 49	151

Table 6.10: Summary of VoIP flows in configuration 5

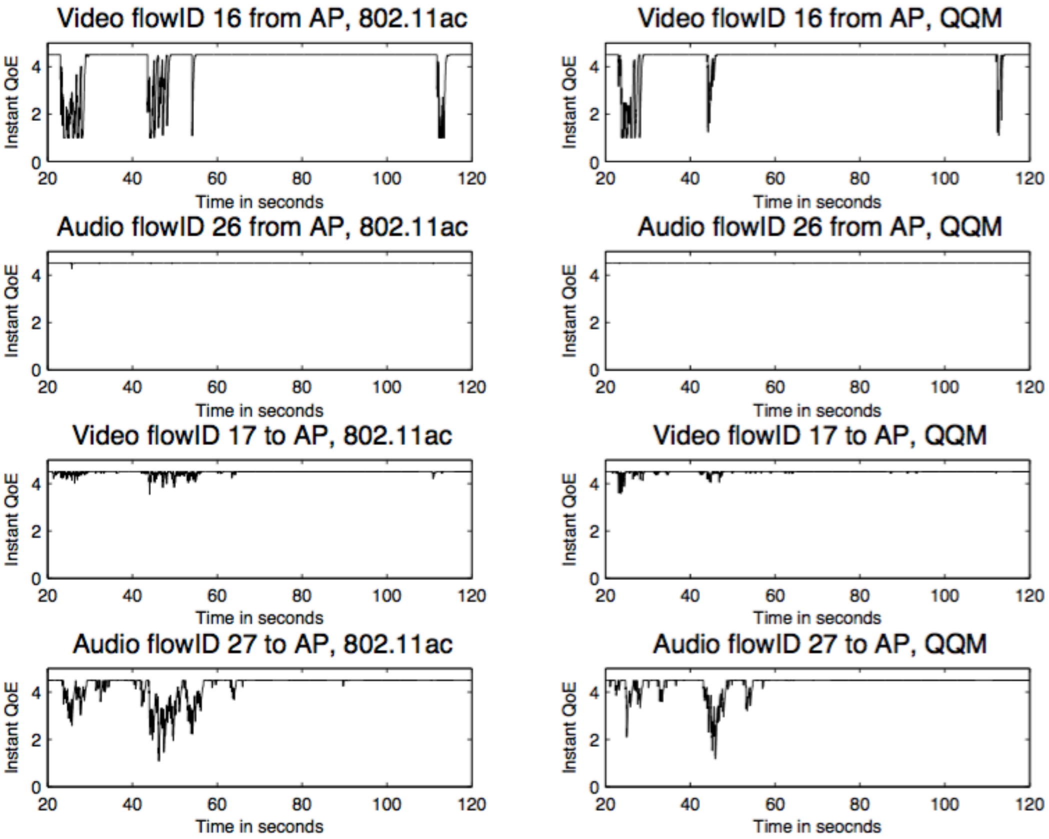

Fig. 6.34: QoE of audio and video bidirectional Flows at Wireless Node 6 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

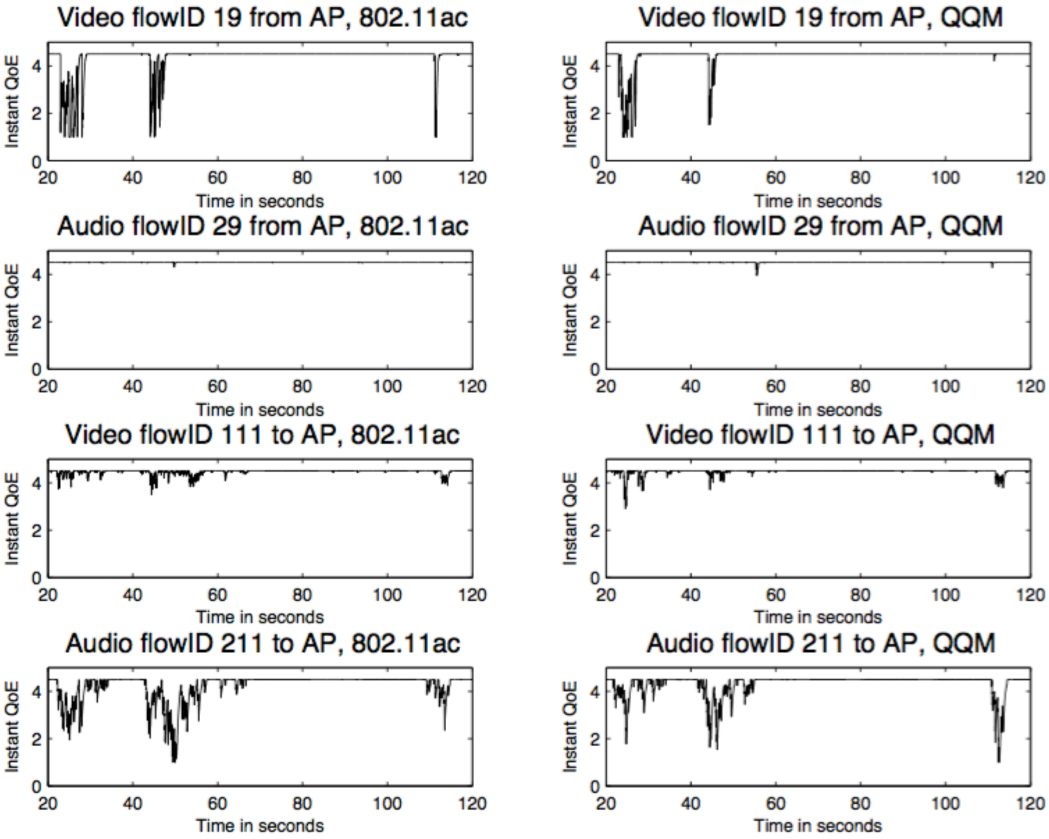

Fig. 6.35: QoE of audio and video bidirectional Flows at Wireless Node 9 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

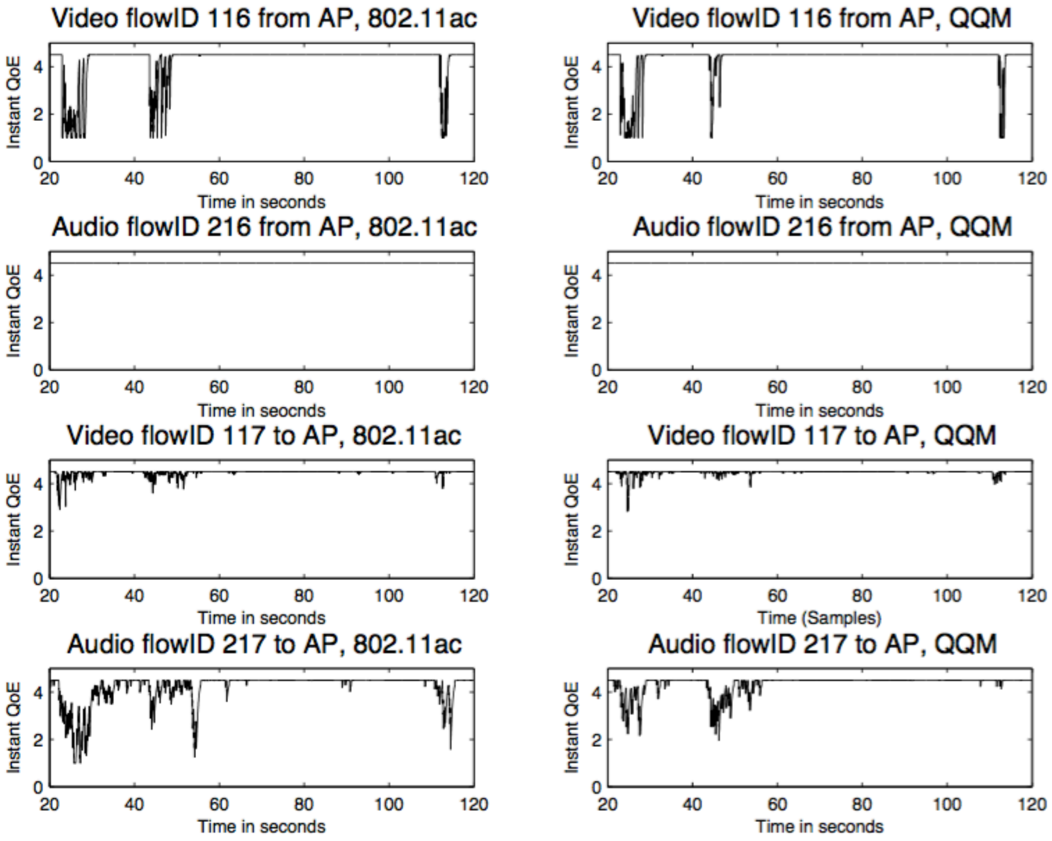

Fig. 6.36: QoE of audio and video bidirectional Flows at Wireless Node 16 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

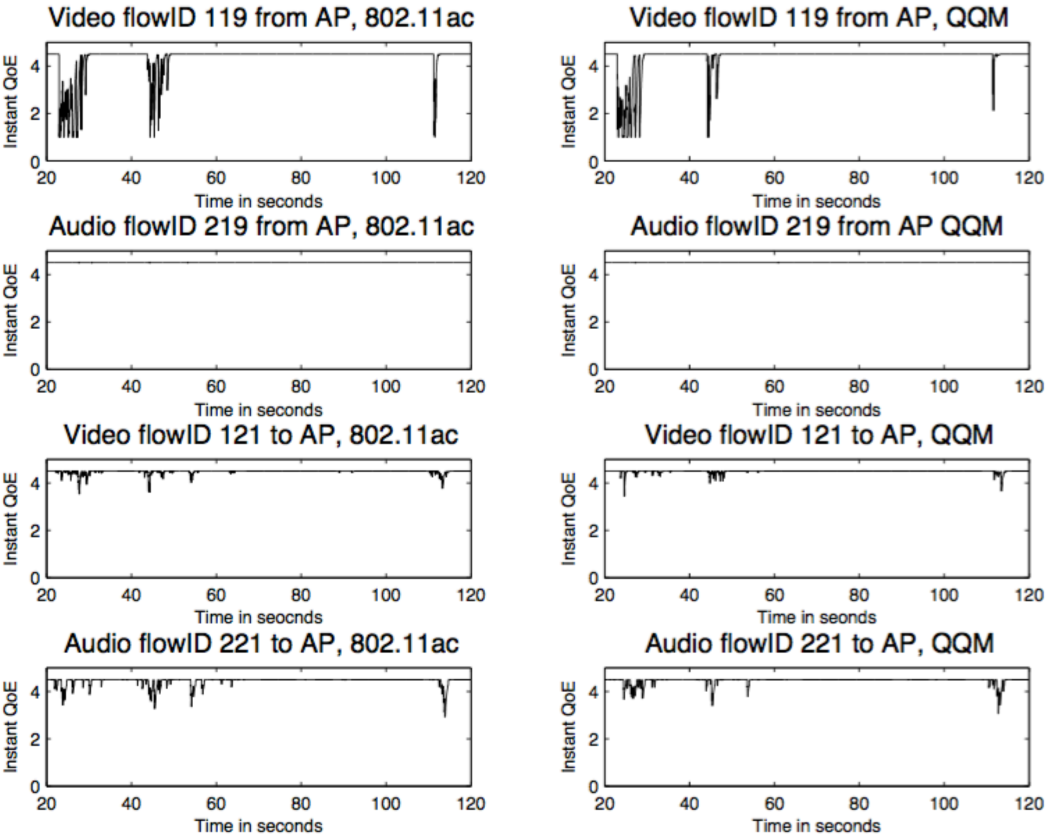

Fig. 6.37: QoE of audio and video bidirectional Flows at Wireless Node 19 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

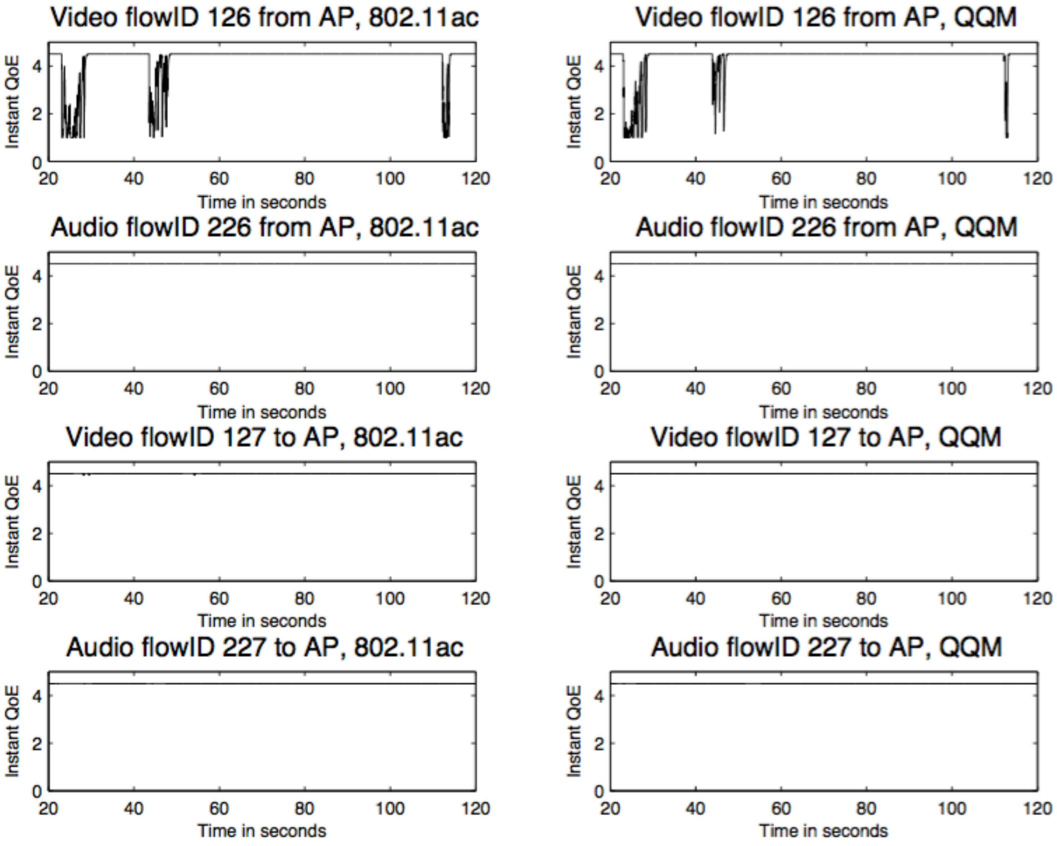

Fig. 6.38: QoE of audio and video bidirectional Flows at Wireless Node 26 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

QQM algorithm.

Two observations can be made about the VoIP, audio and video flows. Firstly, issues with quality are only visible for flows from the wireless to the wired network. The reason for this is that the mobile nodes access the channel with frequency 30Hz for AC_1 and 50 Hz for AC_0; whereas the AP accesses the channel much more frequently due to the large number of flows going from the wired to the wireless network. Secondly, as mentioned above, the active eQoS controller drops packets whose delay and jitter exceed the maximum thresholds, whereas IEEE802.11ac with EDCA alone would transmit these packets. Therefore the effective eQoS measured at the receiving node is lower than the eQoS measured at the AP, as shown by the QoE in the figures above.

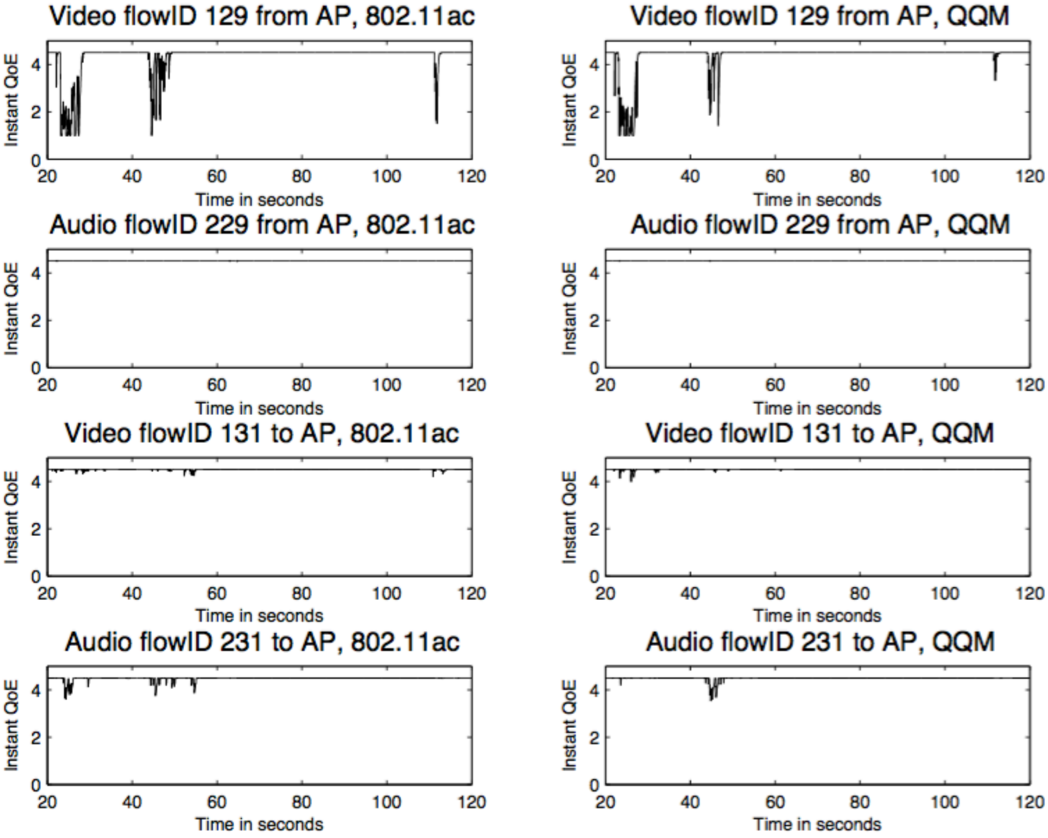

Fig. 6.39: QoE of audio and video bidirectional Flows at Wireless Node 29 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

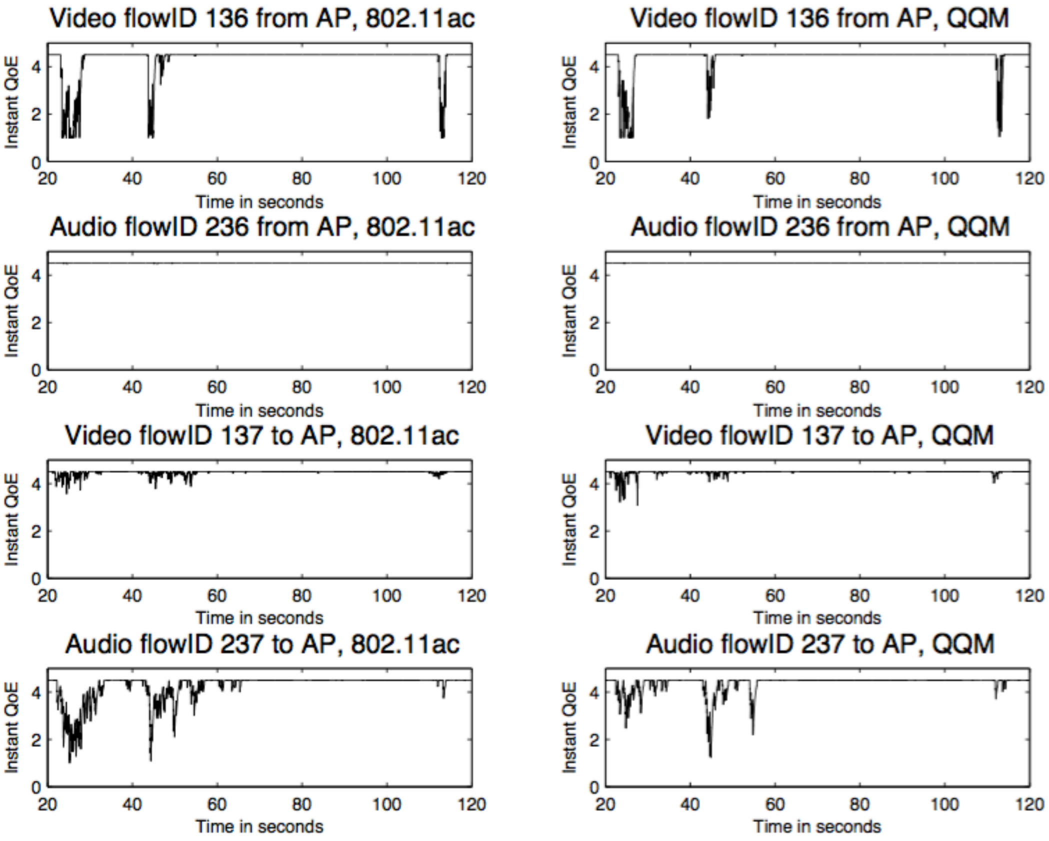

Fig. 6.40: QoE of audio and video bidirectional Flows at Wireless Node 36 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

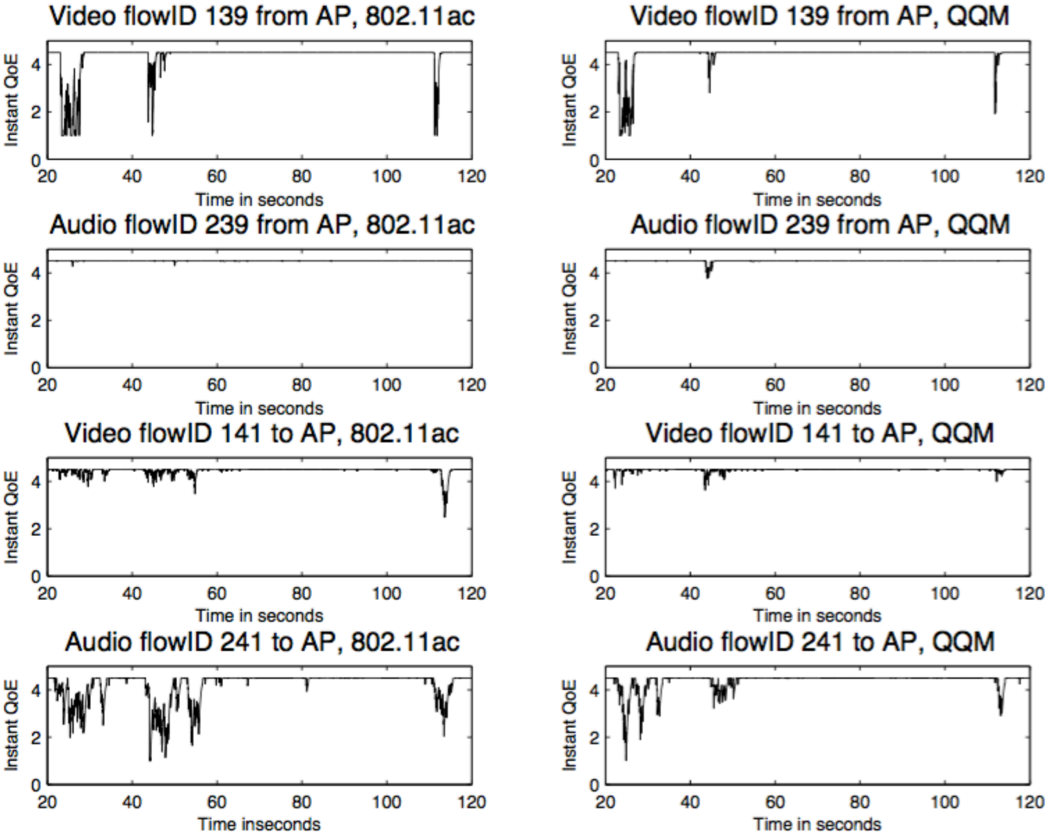

Fig. 6.41: QoE of audio and video bidirectional Flows at Wireless Node 39 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

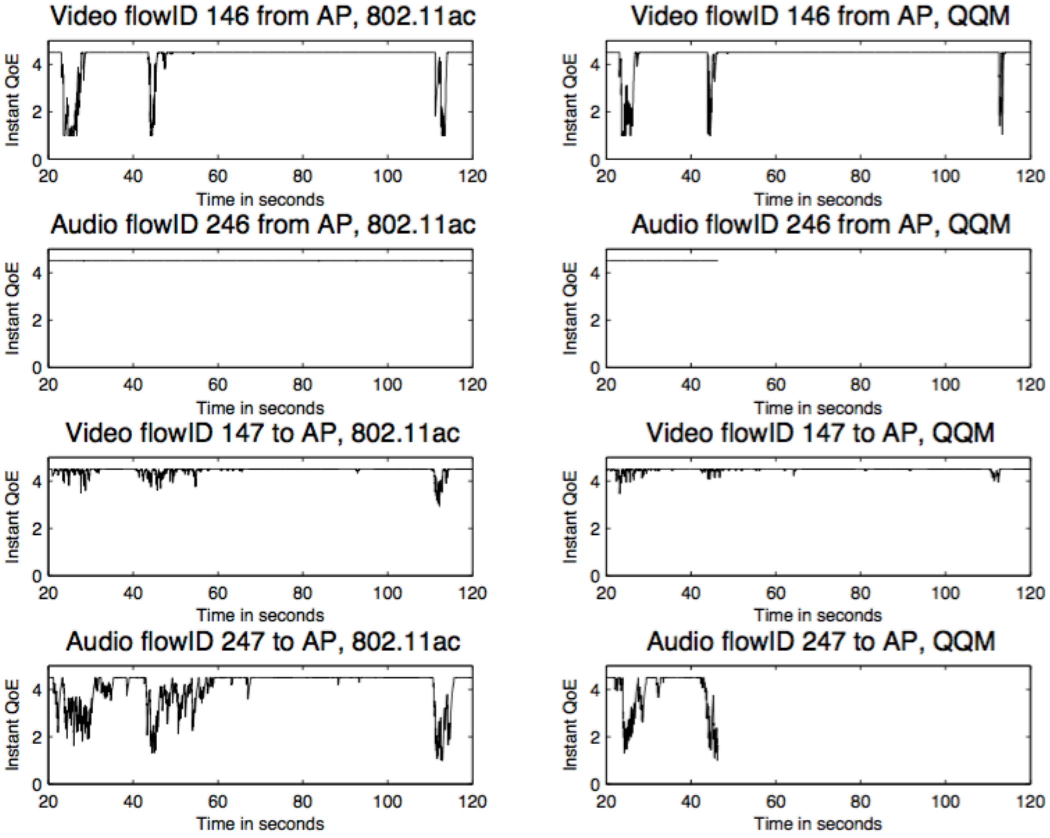

Fig. 6.42: QoE of audio and video bidirectional Flows at Wireless Node 46 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

6.5 Overhead and Complexity Analysis

One of the key theoretical design goals for the QQM algorithm was that it should have low overhead and complexity. Its complexity can be summarised as $O(n)$, as the algorithm does not include any loops or recursive functions. However, the algorithm does include some nested conditional expressions.

The active eQoS measurement system performs the eQoS calculation by counting the packets dropped at the queue and the packets that exceed the maximum delay or jitter thresholds.

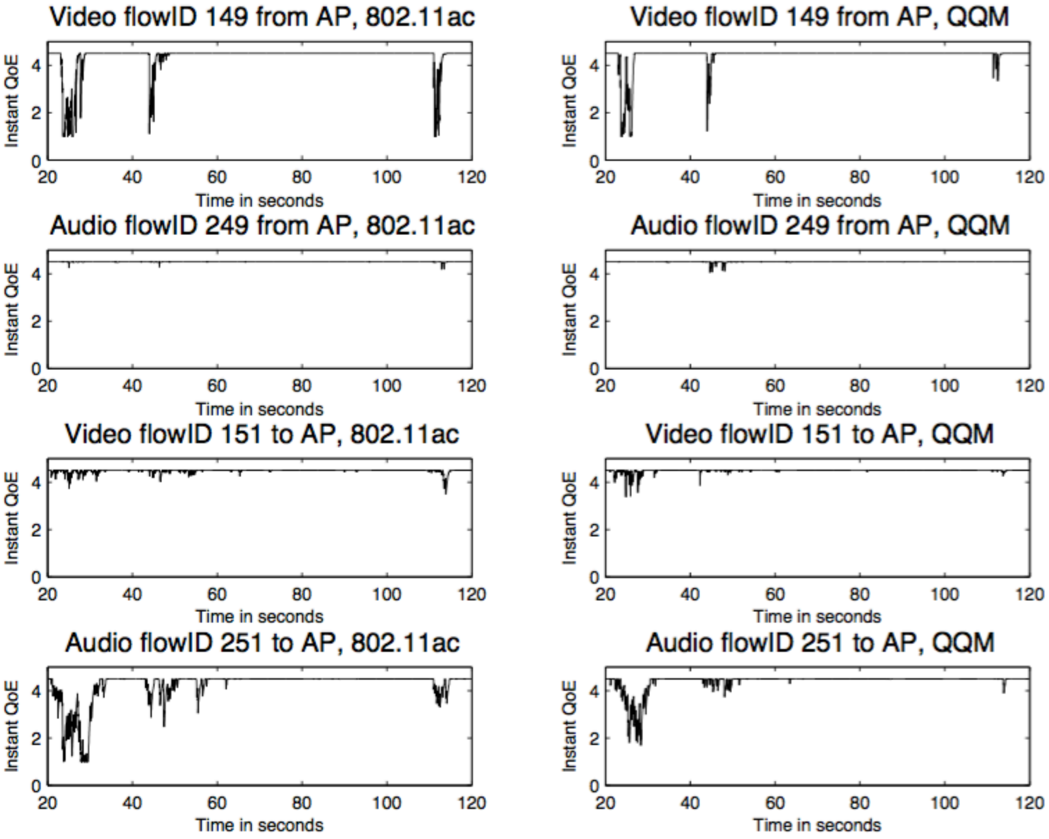

Fig. 6.43: QoE of audio and video bidirectional Flows at Wireless Node 49 for two systems: (i) IEEE802.11ac with EDCA and (ii) the same system with the addition of QQM for the Configuration 5 Simulation

The operations required to achieve this are comparisons and a few conditional expressions.

The contention window controller compares the eQoS value with the contention window size table. The operation performed by the contention window controller are, once again, just a few conditional expressions.

The queue management system samples the time to transmit a packet. The packet transmission time is then used to calculate the maximum threshold for the queue length and the average transmission time. These operations do not require any conditional expressions.

The resources required by the QQM algorithm are very limited. A few variables are needed to keep track of each flow and each queue in order to store information on lost packets, quality and sampling times. The CPU effort required is minimal compared with the capabilities of modern CPUs. The operations performed by the algorithm are mainly packet header analysis and variable comparisons, operations similar to a those carried out by most routing algorithms.

6.6 Summary

This chapter presented an evaluation of eQoS metric, theoretical model and the Quality Queue Management (QQM) algorithm and its subsystems through computer simulation. The range of scenarios explored clearly demonstrate the efficiency of the metric and the algorithms.

eQoS not only provides long term average estimates of the QoE that are shown to be consistent with existing full reference methods via simulation, it also provides an almost instantaneous estimate of the QoE.

In the second part of the chapter the mathematical model is evaluated. Simulations compare the average time needed to transmit packets from each AC, with the time estimated by the theoretical model via the sampling of packets in the queue. These demonstrated that the theoretical model can be used to predict traffic and to estimate the probabilities associated with transmission, collision and idle events on the wireless network.

The third part of the chapter evaluated the performance and efficiency of the QQM algorithm subsystems through computer simulations. Traffic simulations of the QQM controller with protocol IEEE802.11ac and EDCA were compared with those obtained from a system that also implemented protocol IEEE802.11ac and EDCA. The queue management controller was evaluated using similar scenarios. The active eQoS controller and the contention window controller were evaluated separately to highlight the benefits of both these algorithms. The simulations were performed using suitable estimates of real time flows and traffic.

The simulation results indicate the validity of the theoretical model derived in this thesis and provide compelling evidence of the effectiveness of QQM. The following chapter concludes this dissertation.

Chapter 7

Conclusion

This dissertation has analysed and solved the challenging problem of the provision of Quality of Experience (QoE) to real time services on future wireless networks. Future wireless network will soon replace existing wireless networks and this thesis investigated a mechanism for guaranteeing these new wireless networking technologies a level of QoE comparable to that achieved on existing cellular telephony networks.

Future wireless networks are innovative because they introduce VHT and a novel Layer 1 technical design, which makes it easy to apply EDCA efficiently. Moreover, they are inexpensive and it is anticipated that they will rapidly become dominant.

Future wireless networking technologies must be capable of providing popular real time services. The ability to provide real time services with the same high quality that TELCOs offer as a premium product will bring many advantages for the end user. Firstly, there will be a dramatic reduction in the cost of high quality services; secondly, there will be a huge increase in the throughput available for mobile devices.

Real time services have been shown to be very sensitive to packet loss and packet delay. The quality with which they are provided to the end user is measured by the QoE metric. This is an indicator of the customer's level of satisfaction with the service they are receiving. QoE is one of the most important quality and, indirectly, network performance metrics available to TELCOs. It is the end-user's subjective measure of the combined impact of the network's physical parameter settings on their experience.

This dissertation has determined a relationship between these physical parameters and QoE, through a novel quality measure called eQoS. Packet dropping events that occur at the node are mathematically related to the human perception of the impact of packet dropping events that occur in sequence. Therefore, eQoS is a perceived QoS measure that can be used to obtain an instantaneous estimate of the QoE.

As shown in this thesis, the access point of a wireless network is a critical node in the network for the maintenance of acceptable quality levels. On wired networks the throughput is high and the percentage of packets lost is negligible; by contrast, wireless networks struggle to provide good QoE to their users. Packets can easily accumulate delays at the AP or be dropped when congestion occurs. Therefore an appropriate traffic management algorithm at the AP, that can also be used in its associated mobile nodes, is strategically essential for future quality assurance purposes.

Traffic management algorithms are based on theoretical models which describe the network environment. This dissertation analysed the environment that future wireless networks are likely to inhabit. It proposed a novel theoretical model, based on probabilistic methods and combinatorics, to describe the EDCA method to access the channel applied in a Very High Throughput wireless environment. One of the key contributions of this work is the design of a traffic management algorithm that guarantees high levels of QoE. This algorithm brings together priority classes, quality metrics, traffic flow management and queue length control mechanisms.

Decision making within QQM is driven by the QoE metric. Its key design goals are the provision of high QoE to real time traffic and optimisation of packet transmission scheduling.

Through QQM this dissertation achieves its aim of ensuring services with an expected QoE and optimising throughput on future wireless networks.

7.1 Achievements and Contributions

The dissertation provides a comprehensive description of wireless networks and their traffic management, within the context of which the three research contributions were presented.

The first contribution is the novel eQoS metric. This fills an important research gap as it provides an almost instantaneous estimate of the QoE for real time services. eQoS has two important and evident advantages. Firstly, by providing an almost instantaneous estimate of the QoE it supersedes existing systems where there is a non-negligible delay in the calculation of the QoE estimate. Moreover, it can be successfully incorporated into a management system for the network; providing quality information that can be used directly by the node. eQoS is obtained through the use of combinatorics and is composed of two elements: the number of packets dropped in a one second window and the probability the packets are dropped in sequence. eQoS provides an almost instantaneous estimate of the QoE. It has been shown that, on average, it is comparable to one of the most popular automatic perceived quality metrics, MOS.

A new efficient and simple theoretical model to describe the channel access probabilities is the second contribution of this thesis. The theoretical model is designed to work on future wireless networks where the EDCA method is used to access the channel. It replaces other, more complex, models in the design of a controller for wireless networks. It fully captures the EDCA method and is suitable for use in a fuzzy controller. The model uses combinatorics to estimate the probability of collisions, the probability the channel is idle and the probability a collision occurs between packets.

QQM is the final contribution of this dissertation. It combines eQoS and the theoretical model to create a system to manage traffic on a future wireless network. The novelty of this traffic management method is in the combination of three subsystems that usually work in isolation. To be precise, it creates mutual collaboration between three systems: a queue management system, a contention window controller and an eQoS controller. All with the stated aim of improving the provision of real time services over the wireless network. QQM provides as many real time flows as possible with the best QoE possible in the network.

7.2 Applicability and Limitations

The three main contributions presented in this work are clearly defined and can be used to achieve near optimal results in some specific environments. All contributions can be brought together in a system designed to manage the traffic with the goal of providing real time services with the best possible QoE available.

The first consideration is that the system achieves its best performance when the available throughput is sufficiently large to allow for the exchange of a significant amount of real time traffic. Future wireless networks offer an ideal environment for the utilisation of QQM. Wireless protocols, like IEEE802.11ac, and wireless networks, like WiMax and 4G networks, have sufficient available throughput to make best use of QQM.

Each of the main QQM subsystems requires some particular environment characteristics to be correctly used. eQoS, for example, operates to its full potential when it is used as part of a traffic management algorithm, like QQM, which acts to instantaneously adjust the network parameter settings. To apply eQoS, the network characteristics of each real time flow have to be known in advance. eQoS does not depend on the number and structure of the queues in the system.

QQM can be used efficiently with multiple queue systems where the queue structure is well defined and the characteristics of each real time streaming flow are known; but it can also be used in systems where a single queue is present.

With just a few modifications, QQM can be applied in a wide variety of wireless environments. However, the design will need more significant changes for use in networks where the method of channel access differs significantly from EDCA. It can be adapted for use in WiMax networks with time sharing or in cellular telephony networks but the substantial modifications required to achieve this are beyond the scope of this work.

QQM works to maintain the QoS parameters within predetermined thresholds, but it is more efficient when it works together with higher level applications to smooth any fluctuations in the QoS parameters.

7.3 Future Work

QQM allows for the investigation of various future directions in QoE measurement and the use of traffic management to provide services with QoE guarantees.

QQM was designed to be independent of the wireless protocol, therefore the main ideas for future work are three: the application of QQM in a different scenario from WiFi to extend its functionalities, to expand eQoS functionalities in the network and, finally, to modify QQM to work on networks that uses lower throughput than VHT.

The first suggestion for future work is to extend the functionality of QQM to wireless protocols that do not use EDCA to access the channel. This includes the development of a QQM-like system for a cellular architecture. Such a system could look to guarantee high QoE when a mobile node moves between two access points, i.e. when frequent handover events occur.

Another interesting starting point is to expand QQM functionalities. For further exploration, it is interesting to investigate on the distribution of eQoS measurements over the network and at the application layer. It might be of benefit in the development of an application layer protocol to manage real time service flows and unresponsive flows.

The challenging proposal is the investigation of QQM applicability and performance when the throughput of the network is lower than VHT. QQM is designed to work with VHT networks, but in low throughput network, for example sensor networks [5], it is possible to deliver very limited real time services. When a very low throughput is available, like in underwater sensor networks [205], a few packets are transmitted per second. In this case no realtime services can be delivered, therefore QQM has to be redesigned to guarantee the quality of the delivered packets.

7.4 Final Remarks

Even though combinatorics, statistics and fuzzy logic are widely used in the design of QQM and eQoS, they are straightforward to implement and do not place an unacceptable computational burden on the system. The hardware and software resources required are those that are common to almost all APs and mobile devices.

eQoS and QQM were developed and implemented entirely at Layer 2, they do not affect the Layer 2 protocol or other layer protocols. As noted during the evaluation in chapter 6, they perform best when they are applied at all nodes in the wireless network.

List of Acronyms

3G	Third Generation	10
4G	Fourth Generation	2
AAC	Advanced Audio Coding	31
AC	Access Category	21
ACK	Acknowledgement	16
AIFS	Arbitration Inter-Frame Space	22
AIMD	Additive Increase Multiplicative Decrease	59
AP	Access Point	13
APSD	Automatic Power Save Delivery	23
AQM	Active Queue Management	3
AQoS	Assessed QoS	40
AVC	Advanced Video Coding	33
BA	Block Acknowledgment	25
BPSK	Binary Phase Shift Keying	17
BCC	Binary Convolutional Coding	24
BS	Base Station	11
BSS	Basic Service Set	14
CBR	Constant Bit Rate	93
CCK	Complementary Code Keying	17
CoS	Class of Service	21
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance	15

CTS Clear To Send . . . 16

CW Contention Windows . . . 51

DCF Distributed Control Function . . . 15

DIFS Distributed Inter Frame Spacing . . . 15

DS Distribution System . . . 14

DSL Digital Subscriber Line . . . 13

DSS Distribution System Services . . . 15

EDCA Enhanced Distributed Channel Access . . . 7

ECN Explicit Congestion Notification . . . 55

FR Full Reference . . . 42

eQoS expected Quality of Service . . . 6

ESS Extended Service Set . . . 15

EWMA Exponentially Weighted Moving Average . . . 71

FDD Frequency Division Duplex . . . 11

GCA General Contention window Adaptation . . . 52

HCF Hybrid Coordination Function . . . 21

HCCA HCF Controlled Channel Access . . . 21

HLS HTTP Live Streaming . . . 28

HSPA High Speed Packet Access . . . 11

HT High Throughput . . . 2

HTTP Hypertext Transfer Protocol . . . 28

IBSS Independent Basic Service Set . . . 14

IQoS Intrinsic QoS . . . 39

ISPs Internet Service Providers . . . 12

IP Internet Protocol . . . 2

MDI Media Delivery Index . . . 44

MIMO Multiple Input Multiple Output . . . 12

MOS Mean Opinion Score . . . 40

MPDU MAC Sublayer Protocol Data Unit 18

MPEG Moving Picture Experts Group 33

MPEG-DASH MPEG-Dynamic Adaptive Streaming over HTTP 28

MSDU MAC Service Data Unit 15

MU-MIMO Multi User - Multi Input Multi Output 24

NR No Reference 42

ODG Objective Difference Grade 44

OFDM Orthogonal Frequency-Division Multiplexing 12

PAN Personal Area Network 48

PCF Point Coordination Function 16

PCM Pulse Code Modulation 29

PEAQ Perceptual Evaluation of Audio Quality 43

PESQ Perceptual Evaluation of Speech Quality 42

PEVQ Perceptual Evaluation of Video Quality 44

PHB Per Hop Behavior 50

POLQA Perceptual Objective Listening Quality Assessment 43

PQoS Perceived QoS 39

PSK Phase-Shift Keying 17

PSNR Peak Signal-to-Noise Ratio 44

QAM Quadrature Amplitude Modulation 12

QoE Quality of Experience 3

QoS Quality of Service 2

QPSK Quadrature Phase Shift Keying 17

QQM Quality Queue Management 6

RFC Request for Comments 49

RR Reduced Reference 42

RTCP Real-time Transport Control Protocol 28

RTP Real-time Transport Protocol 28

RTS Request To Send ... 16

RTT Round Trip Time ... 27

SIFS Short Inter Frame Spacing ... 16

SISO Single Input and Single Output ... 25

SNR Signal-to-Noise Ratio ... 20

SOHO Small Office/Home Office ... 13

SS Station Services ... 15

SSIM Structural Similarity ... 44

STBC Space-Time Block Coding ... 24

TC Traffic Categories ... 21

TCP Transmission Control Protocol ... 4

TDD Time Division Duplex ... 12

TELCOs Telephone Companies ... 1

TXOP Transmission Opportunity ... 22

UDP User Datagram Protocol ... 4

UE User Equipment ... 11

UMTS Universal Mobile Telecommunications System ... 11

VBR Variable Bit Rate ... 35

VCEG Video Coding Experts Group ... 33

VHT Very High Throughput ... 2

VHT-LTF VHT Long Training Fields ... 157

VoIP Voice over IP ... 8

VQM Video Quality Metric ... 44

MPQM Moving Pictures Quality Metric ... 45

W-CDMA Wideband Code Division Multiple Access ... 11

WAN Wide Area Network ... 10

WiFi WIreless FIdelity ... 10

WiMax Worldwide Interoperability for Microwave Access ... 2

WLAN Wireless Local Area Network ... 10

Bibliography

[1] Index, Cisco Visual Networking, “Global mobile data traffic forecast update, 2013–2018,” *Cisco White Paper*, 2014.

[2] Index, Cisco Visual Networking, “The zettabyte era–trends and analysis,” *Cisco white paper*, 2013.

[3] Index, Cisco Visual Networking, “Forecast and methodology, 2013-2018,” 2014.

[4] Q. Li, G. Li, W. Lee, M. i. Lee, D. Mazzarese, B. Clerckx, and Z. Li, “MIMO techniques in WiMAX and LTE: a feature overview,” *IEEE Communications Magazine*, vol. 48, pp. 86–92, May 2010.

[5] G. Mullett, *Wireless telecommunications systems and networks*. Thomson Delmar Learning, 2006.

[6] “IEEE Standard for Information technology– Telecommunications and information exchange between systemsLocal and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz.,” *IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012)*, pp. 1–425, Dec 2013.

[7] “IEEE 802.11ac: Enhancements for very high throughput WLANs,” in *2011 IEEE*

22nd International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 849–853, Sept 2011.

[8] R. Stankiewicz, P. Cholda, and A. Jajszczyk, “QoX: What is it really?,” *IEEE Communications Magazine*, vol. 49, pp. 148–158, April 2011.

[9] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *Networking, IEEE/ACM Transactions on*, vol. 1, no. 4, pp. 397–413, 1993.

[10] C. V. Hollot, Y. Liu, V. Misra, and D. Towsley, “Unresponsive flows and AQM performance,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1, pp. 85–95 vol.1, March 2003.

[11] G. Pibiri, C. M. Goldrick, and M. Huggard, “Enhancing AQM performance on wireless networks,” in *Wireless Days (WD), 2012 IFIP*, pp. 1–3, Nov 2012.

[12] G. Pibiri, C. Mc Goldrick, and M. Huggard, “Using Active Queue Management to Enhance Performance in IEEE802.11,” in *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, PM2HW2N ’09, (New York, NY, USA), pp. 70–77, ACM, 2009.

[13] T. Hoßfeld, P. Tran-Gia, and M. Fiedler, “Quantification of Quality of Experience for Edge-Based Applications,” in *Managing Traffic Performance in Converged Networks: 20th International Teletraffic Congress, ITC20 2007, Ottawa, Canada, June 17-21, 2007. Proceedings* (L. Mason, T. Drwiega, and J. Yan, eds.), (Berlin, Heidelberg), pp. 361–373, Springer Berlin Heidelberg, 2007.

[14] G. Pibiri, C. Mc Goldrick, and M. Huggard, “Expected Quality of Service (eQoS) A network metric for capturing end-user experience,” in *Wireless Days (WD), 2012 IFIP*, pp. 1–6, IEEE, 2012.

[15] G. Pibiri, C. Mc Goldrick, and M. Huggard, “Ensuring quality services on wifi net-

works for offloaded cellular traffic," in *Wireless On-demand Network Systems and Services (WONS), 2017 13th Annual Conference on*, pp. 136–143, IEEE, 2017.

[16] "IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Prioritization of Management Frames," *IEEE Std 802.11ae-2012 (Amendment to IEEE Std 802.11-2012)*, pp. 1–52, April 2012.

[17] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: MAC Enhancements for Robust Audio Video Streaming," *IEEE Std 802.11aa-2012 (Amendment to IEEE Std 802.11-2012 as amended by IEEE Std 802.11ae-2012)*, pp. 1–162, May 2012.

[18] Cisco, "Understanding delay in packet voice networks. White Paper," *Cisco White Paper, Document ID:5125*, Sept. 2003.

[19] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, July 2003.

[20] R. Ratasuk, A. Prasad, Z. Li, A. Ghosh, and M. A. Uusitalo, "Recent advancements in M2M communications in 4G networks and evolution towards 5G," in *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, pp. 52–57, Feb 2015.

[21] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput,"

IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009), pp. 1–565, Oct 2009.

[22] Ravichandiran, C. and Raj, Dr P. and Vaidhyanathan, Dr, “A Perspective Analysis of the Fourth Generation (4G) Communication Standard and the Respective Technologies,” *IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), Vol.1, No. 1*, pp. 23–31, October 2011.

[23] “IEEE Standard for Air Interface for Broadband Wireless Access Systems,” *IEEE Std 802.16-2012 (Revision of IEEE Std 802.16-2009)*, pp. 1–2542, Aug 2012.

[24] “IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 3: Advanced Air Interface,” *IEEE Std 802.16m-2011(Amendment to IEEE Std 802.16-2009)*, pp. 1–1112, May 2011.

[25] E. Seidel, “Progress on "LTE Advanced"-the new 4G standard,” *White Paper, Nomor Research GmbH*, 2008.

[26] Cisco, “802.11n: The next generation of wireless performance.,” *Cisco White Paper*, 2009.

[27] J. Jun, P. Peddabachagari, and M. Sichitiu, “Theoretical maximum throughput of IEEE 802.11 and its applications,” in *Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on*, pp. 249–256, April 2003.

[28] “IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. 1–1076, June 2007.

[29] K. Medepalli, P. Gopalakrishnan, D. Famolari, and T. Kodama, “Voice capacity of IEEE 802.11b, 802.11a and 802.11g wireless LANs,” in *Global Telecommunications Conference, 2004. GLOBECOM ’04. IEEE*, vol. 3, pp. 1549–1553, Nov 2004.

[30] A. Kajackas and A. Vindašius, “Applying IEEE 802.11 e for Real-Time Services,” *Elektronika ir Elektrotechnika*, vol. 89, no. 1, pp. 73–78, 2015.

[31] M. van der Schaar, Y. Andreopoulos, and Z. Hu, “Optimized scalable video streaming over IEEE 802.11 a/e HCCA wireless networks under delay constraints,” *Mobile Computing, IEEE Transactions on*, vol. 5, no. 6, pp. 755–768, 2006.

[32] “IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band,” *IEEE Std 802.11a-1999*, pp. 1–102, Dec 1999.

[33] “IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band,” *IEEE Std 802.11b-1999*, pp. 1–96, Jan 2000.

[34] “IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*, pp. i–67, 2003.

[35] “IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Phys-

ical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements," *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, pp. 1–212, Nov 2005.

[36] S. Wiethölter and C. Hoene, "Design and verification of an IEEE 802.11e EDCF simulation model in ns-2.26," in *Technische Universität Berlin, Tech. Rep. TKN-03-019*, November 2003.

[37] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 365–386, Aug 1995.

[38] S. Singh and R. Tripathi, "Enhancement in QoS for Hybrid Networks Using IEEE 802.11 e HCCA with Extended AODV Routing Protocol," *International Journal of Communications, Network and System Sciences*, vol. 8, no. 06, p. 236, 2015.

[39] "IEEE Draft Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges (Revision of IEEE Std 802.1D -1998 Incorporating IEEE Std 802.1T -2001 IEEE Std 802.1W -2001) (Replaced by 802.1D-2004)," *IEEE Std P802.1D/D4*, 2003.

[40] X. Lei and S. H. Rhee, "Enhancement of the IEEE 802.11 Power Saving Mode by Prioritized Reservations," *Int. J. Distrib. Sen. Netw.*, vol. 2015, pp. 9:9–9:9, Jan. 2015.

[41] K. Kosek-Szott, M. Natkaniec, S. Szott, A. Krasilov, A. Lyakhov, A. Safonov, and I. Tinnirello, "What's new for QoS in IEEE 802.11?," *IEEE Network*, vol. 27, pp. 95–104, November 2013.

[42] A. de la Oliva, P. Serrano, P. Salvador, and A. Banchs, "Performance evaluation of the IEEE 802.11aa multicast mechanisms for video streaming," pp. 1–9, June 2013.

[43] C. Zhu, Y. Kim, O. Aboul-magd, and C. Ngo, "Multi-user support in next generation wireless LAN," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1120–1121, Jan 2011.

[44] "802.11ac: The Fifth Generation of Wi-Fi," *Technical White Paper, Cisco*, 2012.

[45] T. Szigeti and C. Hattingh, "Quality of service design overview," *Cisco, San Jose, CA, Dec*, 2004.

[46] C.-F. Wong, W.-L. Fung, C.-F. J. Tang, and S. H. G. Chan, "Tcp streaming for low-delay wireless video," in *Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE'05)*, pp. 6 pp.–41, Aug 2005.

[47] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validations," in *ACM SIGCOMM Computer Communication Review*, vol. 28, pp. 303–314, ACM, 1998.

[48] V. Misra, W. Gong, and D. Towsley, "Stochastic differential equation modeling and analysis of TCP-windowsize behavior," in *Proc. Performance'99*, Citeseer, 1999.

[49] K. Zhou, K. L. Yeung, and V. O. K. Li, "On bursty packet loss model for TCP performance analysis," in *HPSR. 2005 Workshop on High Performance Switching and Routing, 2005.*, pp. 292–296, May 2005.

[50] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications." RFC 3550 (Standard), July 2003. Updated by RFCs 5506, 5761, 6051, 6222.

[51] J. Lazzaro, "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport." RFC 4571 (Proposed Standard), July 2006.

[52] Network Working Group and others, "Rfc 2616: Hypertext transfer protocol–http/1.1," *R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee*, 1999.

[53] R. Pantos, "HTTP Live Streaming draft-pantos-http-live-streaming-13," *Published by the Internet Engineering Task Force (IETF)*, 2014.

[54] I. Sodagar, “White paper on MPEG-DASH standard: The standard for multimedia streaming over internet,” *ISO/IEC JTC1/SC29/WG11 W13533*, 2012.

[55] J. R. A. Bruce Carlson, Paul Crilly, *Communication Systems (4th Edition).* McGraw-Hill Science/Engineering/Math, June 25, 2001.

[56] “ITU-T recommendation G.711 : Pulse Code Modulation (PCM) of voice frequencies,” November 1988.

[57] “ITU-T recommendation G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP),” Jan 2007.

[58] Cisco, “Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation,” *Technical Report*, February 2006.

[59] “ITU-T recommendation G.114 : SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS,” May 2003.

[60] Y. Wang and M. Vilermo, “A compressed domain beat detector using MP3 audio bitstreams,” in *Proceedings of the ninth ACM international conference on Multimedia*, pp. 194–202, ACM, 2001.

[61] International Electrotechnical Commission and others, *ISO/IEC 11172-3: Information Technology: Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to about 1, 5 Mbit/s.* ISO/IEC, 1993.

[62] K. Brandenburg, “MP3 and AAC explained,” in *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*, Audio Engineering Society, 1999.

[63] C. C. Todd, G. A. Davidson, M. F. Davis, L. D. Fielder, B. D. Link, and S. Vernon, “AC-3: Flexible perceptual coding for audio transmission and storage,” in *Audio Engineering Society Convention 96*, Audio Engineering Society, 1994.

[64] A. Robert, O. Alvarez, and G. Doerr, “Adjusting bit-stream video watermarking systems to cope with HTTP adaptive streaming transmission,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7416–7419, May 2014.

[65] Rec, ITUT, “H. 262 | ISO/IEC 13818-2,” *Information technology—Generic coding of moving pictures and associated audio information—Video*, 2000.

[66] S. kak Kwon, A. Tamhankar, and K. Rao, “Overview of H.264/MPEG-4 part 10,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 186 – 216, 2006.

[67] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the Scalable H.264/MPEG4-AVC Extension,” in *2006 International Conference on Image Processing*, pp. 161–164, Oct 2006.

[68] M. Smart and B. Keepence, “Understanding mpeg-4 video,” *White paper IC-COD-REP012*, Jun 2008.

[69] S. Sen, J. K. Dey, J. F. Kurose, J. A. Stankovic, and D. Towsley, “Streaming cbr transmission of vbr stored video,” in *Voice, Video, and Data Communications*, pp. 26–36, International Society for Optics and Photonics, 1998.

[70] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, “REM: active queue management,” *IEEE Network*, vol. 15, pp. 48–53, May 2001.

[71] “ITU-T recommendation E.800: Definitions of terms related to quality of service,” Sept. 2008.

[72] “ITU-T recommendation E.802: Framework and methodologies for the determination and application of QoS parameters,” Feb. 2007.

[73] “ITU-T recommendation G.1000: Communications Quality of Service: A framework and definitions,” Nov. 2001.

[74] E. Crawley and R. Nair, “RFC2386-A framework for QoS-based routing in the internet,” *USA: Network Working Group*, vol. 5, 1998.

[75] E. Ibarrola, J. Xiao, F. Liberal, and A. Ferro, “Internet QoS regulation in future networks: a user-centric approach,” *IEEE Communications Magazine*, vol. 49, pp. 148–155, Oct 2011.

[76] “ITU-T recommendation P.10/G.100: Vocabulary for performance and quality of service,” July 2006.

[77] “ITU-T recommendation P.800: Methods for subjective determination of transmission quality,” August 1996.

[78] T. Hoßfeld, D. Hock, P. Tran-Gia, K. Tutschku, and M. Fiedler, “Testing the IQX hypothesis for exponential interdependency between QoS and QoE of voice codecs iLBC and G. 711,” in *Proc. of 18th ITC Specialist Seminar on Quality of Experience, Karlskrona, Sweden*, pp. 105–114, 2008.

[79] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service,” *IEEE Network*, vol. 24, pp. 36–41, March 2010.

[80] P. Reichl, S. Egger, R. Schatz, and A. D’Alconzo, “The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment,” in *Communications (ICC), 2010 IEEE International Conference on*, pp. 1–5, May 2010.

[81] R. Serral-Gracià, M. Yannuzzi, E. Marin-Tordera, X. Masip-Bruin, and S. Sánchez, “Quality of experience enforcement in wireless networks,” in *Wired/Wireless Internet Communications*, pp. 180–191, Springer, 2010.

[82] “A_PSQA: Efficient real-time video streaming QoE tool in a future media internet context,” in *2011 IEEE International Conference on Multimedia and Expo*, pp. 1–6, July 2011.

[83] P. Le Callet, C. Viard-Gaudin, S. Péchard, and E. Caillault, “No reference and reduced reference video quality metrics for end to end qos monitoring,” *IEICE transactions on communications*, vol. 89, no. 2, pp. 289–296, 2006.

[84] “ITU-T recommendation P.862 (02/01): Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs; P.862 Amedment 2 (11/05): Revised Annex A – Reference implementations and conformance testing for ITU-T Recs P.862, P.862.1 and P.862.2; P.862 Corrigendum 1 (10/07),” Oct. 2007.

[85] “PESQ: An Introduction,” *White Paper, PSytechnics Limited, Ipswich, United Kingdom*, September 2001.

[86] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs,” in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP ’01). 2001 IEEE International Conference on*, vol. 2, pp. 749–752 vol.2, 2001.

[87] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, “ViSQOL: The Virtual Speech Quality Objective Listener,” in *Acoustic Signal Enhancement; Proceedings of IWAENC 2012; International Workshop on*, pp. 1–4, Sept 2012.

[88] A. Hines and N. Harte, “Speech intelligibility prediction using a neurogram similarity index measure,” *Speech Communication*, vol. 54, no. 2, pp. 306–320, 2012.

[89] “ITU-T recommendation P.863: Perceptual objective listening quality assessment,” January 2011.

[90] J. G. Beerends, C. Schmidmer, J. Berger, M. Obermann, R. Ullmann, J. Pomy, and M. Keyhl, “Perceptual objective listening quality assessment (POLQA), the third generation ITU-T standard for end-to-end speech quality measurement part I - Temporal alignment,” *Journal of the Audio Engineering Society*, vol. 61, no. 6, pp. 366–384, 2013.

[91] "ITU-T recommendation P.563: Single-ended method for objective speech quality assessment in narrow-band telephony applications," May 2004.

[92] E. Biersack, C. Callegari, and M. Matijasevic, *Data traffic monitoring and analysis. From measurement, classification, and anomaly detection to quality of experience*, vol. 7754. Springer, 2013.

[93] "ITU-T recommendation BS.1387: Method for objective measurements of perceived audio quality," 11 2001.

[94] Assembly, TIR, "ITU-R BS. 1284-1: EN-General methods for the subjective assessment of sound quality," tech. rep., ITU, 2003.

[95] "ITU-T recommendation J.247: Objective perceptual multimedia video quality measurement in the presence of a full reference," 08 2008.

[96] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[97] J. Gross, J. Klaue, H. Karl, and A. Wolisz, "Cross-layer optimization of OFDM transmission systems for MPEG-4 video streaming," *Computer Communications*, vol. 27, no. 11, pp. 1044–1055, 2004.

[98] M. Vranješ, S. Rimac-Drlje, and D. Žagar, "Objective video quality metrics," in *49th International Symposium ELMAR-2007 focused on Mobile Multimedia*, 2007.

[99] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, April 2004.

[100] RFC4445, IETR, "A Proposed Media Delivery Index (MDI)," *Massachusetts: Network Working Group*, vol. 4, 2006.

[101] Y. Wang, "Survey of objective video quality measurements," *Worcester Polytechnic Institute, Department of Computer Science*, 2006.

[102] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, pp. 312–322, Sept 2004.

[103] C. J. Van den Branden Lambrecht and O. Verscheure, "Perceptual quality measure using a spatiotemporal model of the human visual system," in *Electronic Imaging: Science & Technology*, pp. 450–461, International Society for Optics and Photonics, 1996.

[104] "ITU-T recommendation G.1010: End-user multimedia QoS categories," Nov. 2001.

[105] "ITU-R recommendation BT.500-13: Methodology for the subjective assessment of the quality of television pictures," Jan. 2012.

[106] "ITU-T recommendation P.911: Subjective audiovisual quality assessment methods for multimedia applications," Dec. 1998.

[107] "ITU-T recommendation P.911 corrigendum 1: Subjective audiovisual quality assessment methods for multimedia applications," Sept. 1999.

[108] "ITU-T recommendation P.913: Methods for the subjective assessment of video quality, audio quality and audiovisual quality of internet video and distribution quality television in any environment," Jan. 2014.

[109] I. Ali, M. Fleury, S. Moiron, and M. Ghanbari, "Enhanced prioritization for video streaming over QoS-enabled wireless networks," in *Wireless Advanced (WiAd), 2011*, pp. 268–272, June 2011.

[110] Q. Zhang, W. Zhu, and Y. Zhang, "QoS-Adaptive Multimedia Streaming over 3G Wireless Channels," *MMSA2000 Nov*, pp. 9–10, 2000.

[111] P. Papadimitriou and V. Tsaoussidis, "QRP04-4: End-to-end Congestion Management

for Real-Time Streaming Video over the Internet," in *IEEE Globecom 2006*, pp. 1–5, Nov 2006.

[112] A. Nafaa, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Communications Magazine*, vol. 46, pp. 72–79, January 2008.

[113] H. Kim, S. Yun, I. Kang, and S. Bahk, "Resolving 802.11 performance anomalies through QoS differentiation," *IEEE Communications Letters*, vol. 9, pp. 655–657, July 2005.

[114] J. Naoum-Sawaya, B. Ghaddar, S. Khawam, H. Safa, H. Artail, and Z. Dawy, "Adaptive approach for QoS support in IEEE 802.11e wireless LAN," in *WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.*, vol. 2, pp. 167–173 Vol. 2, Aug 2005.

[115] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers." RFC 2474 (Proposed Standard), Dec. 1998. Updated by RFCs 3168, 3260.

[116] D. Grossman, "New Terminology and Clarifications for Diffserv." RFC 3260 (Informational), Apr. 2002.

[117] F. Baker, K. Chan, and A. Smith, "Management Information Base for the Differentiated Services Architecture." RFC 3289 (Proposed Standard), May 2002.

[118] J. Babiarz, K. Chan, and F. Baker, "Configuration Guidelines for DiffServ Service Classes." RFC 4594 (Informational), Aug. 2006. Updated by RFC 5865.

[119] K. Chan, R. Sahita, S. Hahn, and K. McCloghrie, "Differentiated Services Quality of Service Policy Information Base." RFC 3317 (Informational), Mar. 2003.

[120] F. Zheng and J. Nelson, "An H_∞ approach to congestion control design for AQM routers supporting TCP flows in wireless access networks," *Computer Networks*, vol. 51, no. 6, pp. 1684–1704, 2007.

[121] V. Misra, W. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, p. 160, 2000.

[122] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP." RFC 3168 (Proposed Standard), Sept. 2001. Updated by RFCs 4301, 6040.

[123] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535–547, March 2000.

[124] P. Rajmic, J. Hošek, M. Fusek, S. Andreev, and J. Stecík, "Simplified Probabilistic Modelling and Analysis of Enhanced Distributed Coordination Access in IEEE 802.11," *The Computer Journal*, p. bxu081, 2014.

[125] Y. Lin and V. W. S. Wong, "Saturation throughput of IEEE 802.11e EDCA based on mean value analysis," in *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, vol. 1, pp. 475–480, April 2006.

[126] I. Inan, F. Keceli, and E. Ayanoglu, "Performance Analysis of the IEEE 802.11e Enhanced Distributed Coordination Function Using Cycle Time Approach," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, pp. 2552–2557, Nov 2007.

[127] A. Banchs, P. Serrano, and L. Vollero, "Providing Service Guarantees in 802.11e EDCA WLANs with Legacy Stations," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 1057–1071, Aug 2010.

[128] J. W. Tantra, C. H. Foh, and A. B. Mnaouer, "Throughput and delay analysis of the IEEE 802.11e EDCA saturation," in *IEEE International Conference on Communications, 2005. ICC 2005. 2005*, vol. 5, pp. 3450–3454 Vol. 5, May 2005.

[129] Y. Yang, J. Wang, and R. Kravets, "Distributed optimal contention window control for

elastic traffic in wireless LANs," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1, pp. 35–46 vol. 1, March 2005.

[130] F. Bouabdallah and N. Bouabdallah, "The tradeoff between maximizing the sensor network lifetime and the fastest way to report reliably an event using reporting nodes' selection," *Computer Communications*, vol. 31, no. 9, pp. 1763–1776, 2008.

[131] S. Singh and C. S. Raghavendra, "PAMAS - power aware multi-access protocol with signalling for ad hoc networks," *SIGCOMM Comput. Commun. Rev.*, vol. 28, pp. 5–26, July 1998.

[132] R. Ruggles and H. Brodie, "An empirical approach to economic intelligence in World War II," *Journal of the American Statistical Association*, vol. 42, no. 237, pp. 72–91, 1947.

[133] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions*. John Wiley & Sons, 2011.

[134] A. D. Polyanin and A. V. Manzhirov, *Handbook of mathematics for engineers and scientists*. CRC Press, 2006.

[135] Y. Hadjadj Aoul, A. Mehaoua, and C. Skianis, "A fuzzy logic-based AQM for real-time traffic over internet," *Computer Networks*, vol. 51, no. 16, pp. 4617–4633, 2007.

[136] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 123–134, 2001.

[137] W. chang Feng, K. G. Shin, D. D. Kandlur, and D. Saha, "The BLUE active queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 513–528, Aug 2002.

[138] R. Mahajan, S. Floyd, and D. Wetherall, "Controlling high-bandwidth flows at the congested router," in *Network Protocols, 2001. Ninth International Conference on*, pp. 192–201, Nov 2001.

[139] S. Floyd, R. Gummadi, S. Shenker, *et al.*, “Adaptive RED: An algorithm for increasing the robustness of RED’s active queue management.” Technical report, ICSI, 518-522, 2001.

[140] C. Joo, S. Bahk, and S. S. Lumetta, “Hybrid active queue management,” in *Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, pp. 999–1004 vol.2, June 2003.

[141] R. Pan, B. Prabhakar, and K. Psounis, “CHOKe - a stateless active queue management scheme for approximating fair bandwidth allocation,” in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 942–951 vol.2, 2000.

[142] Y. Jiang, M. Hamdi, and J. Liu, “Self adjustable CHOKe: an active queue management algorithm for congestion control and fair bandwidth allocation,” in *Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, pp. 1018–1025 vol.2, June 2003.

[143] J. Huang, J. Wang, and W. Jia, “Downlink Temporal Fairness in 802.11 WLAN Adopting the Virtual Queue Management,” in *2007 IEEE Wireless Communications and Networking Conference*, pp. 3035–3040, March 2007.

[144] Smitha and A. L. N. Reddy, “LRU-RED: an active queue management scheme to contain high bandwidth flows at congested routers,” in *Global Telecommunications Conference, 2001. GLOBECOM ’01. IEEE*, vol. 4, pp. 2311–2315 vol.4, 2001.

[145] Cisco, IOS, “Quality of Service solutions configuration guide, Release 12.2.” QC-79, 2010.

[146] K. Wallace, *Cisco IP Telephony Flash Cards: Weighted Random Early Detection (WRED).* Cisco Press, 2004.

[147] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, “A control theoretic analysis of RED,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1510–1519 vol.3, 2001.

[148] N. S. Nise, *Control Systems Engineering*. 4th Edition, ed. John Wiley and Sons. Inc., 2004.

[149] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, “On designing improved controllers for AQM routers supporting TCP flows,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1726–1734 vol.3, 2001.

[150] N. Bigdeli and M. Haeri, “Predictive functional control for active queue management in congested TCP/IP networks,” *ISA transactions*, vol. 48, no. 1, pp. 107–121, 2009.

[151] F. Yanfei, R. Fengyuan, and L. Chuang, “Design of an active queue management algorithm based fuzzy logic decision,” in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, vol. 1, pp. 286–289 vol.1, April 2003.

[152] S. Floyd, “Recommendations on using the gentle variant of RED,” May 2000.

[153] C. Zhang, M. Khanna, and V. Tsaoussidis, “Experimental assessment of RED in wired/wireless networks,” *International Journal of Communication Systems*, vol. 17, no. 4, pp. 287–302, 2004.

[154] D. D. Clark and W. Fang, “Explicit allocation of best-effort packet delivery service,” *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, Aug 1998.

[155] W.-C. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, “Stochastic fair blue: a queue management algorithm for enforcing fairness,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1520–1529 vol.3, 2001.

[156] C. Long, B. Zhao, X. Guan, and J. Yang, "The YELLOW active queue management algorithm," *Computer Networks*, vol. 47, no. 4, pp. 525–550, 2005.

[157] B. Wydrowski and M. Zukerman, "GREEN: an active queue management algorithm for a self managed Internet," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 4, pp. 2368–2372 vol.4, 2002.

[158] J. Sun, G. Chen, K.-T. Ko, S. Chan, and M. Zukerman, "PD-controller: a new active queue management scheme," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 6, pp. 3103–3107 vol.6, Dec 2003.

[159] Z. Heying, L. Baohong, and D. Wenhua, "Design of a robust active queue management algorithm based on feedback compensation," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 277–285, ACM, 2003.

[160] Y. Qiao and H. Xiaojuan, "A new PID controller for AQM based on neural network," in *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, vol. 1, pp. 804–808, Oct 2010.

[161] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Computer Networks*, vol. 36, no. 2, pp. 203–235, 2001.

[162] Y. Bai and D. Wang, "Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications," in *Advanced Fuzzy Logic Technologies in Industrial Applications*, pp. 17–36, Springer, 2006.

[163] M. G. Simoes, "Introduction to fuzzy control," *Golden, Colorado, USA: Colorado School of Mines-Engineering Division*, 2010.

[164] K. M. Passino and S. Yurkovich, *Fuzzy control*, vol. 42. Citeseer, 1998.

[165] S. Leghmizi and S. Liu, "A survey of fuzzy control for stabilized platforms," *arXiv preprint arXiv:1109.0428*, 2011.

[166] M. I. H. Nour, J. Ooi, and K. Y. Chan, “Fuzzy logic control vs. conventional PID control of an inverted pendulum robot,” in *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*, pp. 209–214, Nov 2007.

[167] S. Vittorio, E. Toscano, and L. L. Bello, “CWFC: A contention window fuzzy controller for QoS support on IEEE 802.11e EDCA,” in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1193–1196, Sept 2008.

[168] S. Floyd, “Notes on the Holt-winters Procedure,” *Unpublished notes*, 1993.

[169] X. Chang, X. Lin, and J. K. Muppala, “A control-theoretic approach to improving fairness in DCF based WLANs,” in *2006 IEEE International Performance Computing and Communications Conference*, pp. 7 pp.–86, April 2006.

[170] X. Lin, X. Chang, and J. K. Muppala, “VQ-RED: An efficient virtual queue management approach to improve fairness in infrastructure WLAN,” in *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN’05)l*, pp. 7 pp.–638, Nov 2005.

[171] Q. Xia, X. Jin, and M. Hamdi, “Active Queue Management with Dual Virtual Proportional Integral Queues for TCP Uplink/Downlink Fairness in Infrastructure WLANs,” *IEEE Transactions on Wireless Communications*, vol. 7, pp. 2261–2271, June 2008.

[172] Y. Xue, H. V. Nguyen, and K. Nahrstedt, “CA-AQM: Channel-Aware Active Queue Management for Wireless Networks,” in *2007 IEEE International Conference on Communications*, pp. 4773–4778, June 2007.

[173] G. Min and X. Jin, “Performance Modelling of Random Early Detection Based Congestion Control for Multi-Class Self-Similar Network Traffic,” in *2008 IEEE International Conference on Communications*, pp. 5564–5568, May 2008.

[174] Y. Dong, D. Makrakis, and T. Sullivan, “Network congestion control in ad hoc IEEE 802.11 wireless LAN,” in *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, vol. 3, pp. 1667–1670 vol.3, May 2003.

[175] S. Yi, M. Kappes, S. Garg, X. Deng, G. Kesidis, and C. R. Das, “Proxy-RED: an AQM scheme for wireless local area networks,” in *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pp. 460–465, Oct 2004.

[176] B. Abbasov, “AHRED: A robust AQM algorithm for wireless ad hoc networks,” in *Application of Information and Communication Technologies, 2009. AICT 2009. International Conference on*, pp. 1–4, Oct 2009.

[177] R. Alasem and H. Abu-Mansour, “EF-AQM: Efficient and Fair Bandwidth Allocation AQM Scheme for Wireless Networks,” in *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*, pp. 169–172, July 2010.

[178] C. Demichelis and P. Chimento, “RFC 3393,” *IP packet delay variationmetric for IP performance metrics (IPPM)*, 2002.

[179] S. Jelassi and G. Rubino, “A perceptually sensitive Markovian model of packet loss processes during voip conversations,” in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 964–969, July 2013.

[180] E. Kreyszig, *Advanced engineering mathematics*. John Wiley & Sons, 2010.

[181] J. Park, R. VanZee, W. Lal, D. Welter, and J. Obeysekera, “Sigmoidal activation of proportional integral control applied to water management,” *Journal of water resources planning and management*, vol. 131, no. 4, pp. 292–298, 2005.

[182] MATLAB, *version 7.10.0.499 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[183] T. Williams, C. Kelley, C. Bersch, H.-B. Bröker, J. Campbell, R. Cunningham, D. Denholm, G. Elber, R. Fearick, C. Grammes, *et al.*, “gnuplot 5.0,” 2015.

[184] K. Yu, X. Xu, Q. Liang, Z. Hu, J. Yang, Y. Guo, and H. Zhang, “Model predictive control for connected hybrid electric vehicles,” *Mathematical Problems in Engineering*, vol. 2015, 2015.

[185] D. Auslaender, S. Auslaender, and M. Fussenegger, “Designer PH sensor as universal transgene controller,” Dec. 31 2015. US Patent 20,150,376,647.

[186] R. Rutledge, “Sigmoidal curve-fitting redefines quantitative real-time pcr with the prospective of developing automated high-throughput applications,” *Nucleic acids research*, vol. 32, no. 22, pp. e178–e178, 2004.

[187] F. Casu, J. Cabrera, F. Jaureguizar, and N. García, “A protection scheme for multimedia packet streams in bursty packet loss networks based on small block low-density parity-check codes,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 1–14, 2015.

[188] P. Kabal, “An examination and interpretation of itu-r bs. 1387: Perceptual evaluation of audio quality,” *TSP Lab Technical Report, Dept. Electrical & Computer Engineering, McGill University*, pp. 1–89, 2002.

[189] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance anomaly of 802.11b,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, pp. 836–843 vol.2, March 2003.

[190] P. Patras, A. Banchs, and P. Serrano, “A Control Theoretic Approach to Distributed Optimal Configuration of 802.11 WLANs,” *Mobile Networks and Applications*, vol. 14, no. 6, pp. 697–708, 2009.

[191] H. Perros, “Open queueing networks with blocking-a personal log,” *Performance Evaluation: Stories and Perspectives, Austrian Computer Society*, 2003.

[192] K. Nichols and V. Jacobson, “Controlling queue delay,” *Communications of the ACM*, vol. 55, no. 7, pp. 42–50, 2012.

[193] R. Sedgewick, *Algorithms in C.* Addison-Wesley, 1990.

[194] S. McCanne, S. Floyd, K. Fall, K. Varadhan, *et al.*, "Network simulator ns-2," 1997.

[195] H. T. Friis, "A Note on a Simple Transmission Formula," *Proceedings of the IRE*, vol. 34, pp. 254–256, May 1946.

[196] C. Hoene and S. Wientholter, "Simulating playout schedulers for voipsoftware manual," *TKN, Technical University of Berlin, Alemanha*, 2004.

[197] M. Schwartz, *Mobile wireless communications.* Cambridge University Press, 2005.

[198] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid-A framework for video transmission and quality evaluation," *Computer Performance Evaluation. Modelling Techniques and Tools*, pp. 255–272, 2003.

[199] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid-A Video Quality Evaluation Tool-set," *Telecommunication Networks*, 2011.

[200] C. Ke, C. Shieh, W. Hwang, and A. Ziviani, "An evaluation framework for more realistic simulations of mpeg video transmission," *Journal of information science and engineering*, vol. 24, no. 2, pp. 425–440, 2008.

[201] A. Lie and J. Klaue, "Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video," *Multimedia Systems*, vol. 14, no. 1, pp. 33–50, 2008.

[202] E. Mullins, *Statistics for the quality control chemistry laboratory*. Royal Society of Chemistry, 2003.

[203] G. Joubert and E. Barnard, "The effect of network degradation on speech recognition," in *Poster papers section, the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa, Langebaan, South Africa*, pp. 43–46, 2005.

[204] P. Kabal, "An examination and interpretation of ITU-R BS. 1387: Perceptual evaluation of audio quality," *McGill University*, 2002.

[205] I. F. Akyildiz, P. Wang, and S.-C. Lin, “Softwater: Software-defined networking for next-generation underwater communication systems,” *Ad Hoc Networks*, vol. 46, pp. 1–11, 2016.