Greediness Control Algorithm for Multimedia Streaming in Wireless Local Area Networks

Edward Casey
B.Eng. Hon

Thesis presented for the degree of
Masters in Engineering
to the
School of Electronic and Engineering
Faculty of Engineering Computing
Dublin City University
Ireland

Research Supervisor
Dr. Gabriel-Miro Muntean

July 2007

Declaration

I hereby certify that this material, which I now submit for assesment on the programme of study leading o the award of Masters of Engineering is entirely my own work and has not been taken from the work of others save to the extent such work has been cited and acknowledged within the text of my work.

Signed:

\-------------------------------

Edward Casey (50618217)

Date:

\-------------------------------

Acknowledgements

The support of the Irish Research Council for Science, Engineering and Technology is greatly appreciated.

I would like to thank my research supervisor, Dr. Gabriel-Miro Muntean for his guidance, advice and support over the past few years. I would also like to thank other staff in the Performance Engineering Lab. I am very grateful to the other members, past and present, of the Performance Engineering Lab.

I would like to thank my parents, Marian and Gerry, for their support and encouragement throughout my extended college career. Without them, I would not have made it to where I am today.

Finally I would like to thank Jan. Her advice, encouragement and honesty helped me enormously through my research. She was my guiding light at the end of the tunnel.

Abstract

This work investigates the interaction between the application and transport layers while streaming multimedia in a residential Wireless Local Area Network (WLAN). Inconsistencies have been identified that can have a severe impact on the Quality of Experience (QoE) experienced by end users. This problem arises as a result of the streaming processes reliance on rate adaptation engines based on congestion avoidance mechanisms, that try to obtain as much bandwidth as possible from the limited network resources. These upper transport layer mechanisms have no knowledge of the media which they are carrying and as a result treat all traffic equally. This lack of knowledge of the media carried and the characteristics of the target devices results in fair bandwidth distribution at the transport layer but creates unfairness at the application layer. This unfairness mostly affects user perceived quality when streaming high quality multimedia. Essentially, bandwidth that is distributed fairly between competing video streams at the transport layer results in unfair application layer video quality distribution. Therefore, there is a need to allow application layer streaming solutions, tune the aggressiveness of transport layer congestion control mechanisms, in order to create application layer QoE fairness between competing media streams, by taking their device characteristics into account.

This thesis proposes the Greediness Control Algorithm (GCA), an upper transport layer mechanism that eliminates quality inconsistencies caused by rate / congestion control mechanisms while streaming multimedia in wireless networks. GCA extends an existing solution (i.e. TCP Friendly Rate Control (TFRC)) by introducing two parameters that allow the streaming application to tune the aggressiveness of the rate estimation and as a result, introduce fair distribution of quality at the application layer. The thesis shows that this rate adaptation technique, combined with a scalable video format allows increased overall system QoE. Extensive simulation analysis demonstrate

that this form of rate adaptation increases the overall user QoE achieved via a number of devices operating within the same home WLAN.

Contents

1 Introduction **1**

1.1 Home Multimedia Streaming 1
1.2 Problem Statement 4
1.3 Contribution of this Work 6
1.4 Publications Arising from this Work 7
1.5 Thesis Structure 7

2 Wireless Multimedia Streaming **9**

2.1 Wireless Multimedia Streaming Process 9
2.2 Multimedia Encoding Standards 12
2.2.1 MPEG-1 12
2.2.2 MPEG-2 13
2.2.3 MPEG-4 14
2.3 Network and Transport Layer Protocols 15
2.3.1 Internet Protocol (IP) 15
2.3.2 User Datagram Protocol (UDP) 17
2.3.3 Transport Control Protocol (TCP) 19
2.3.4 Datagram Congestion Control Protocol (DCCP) 21
2.3.5 Real-time Transport Protocol (RTP) 24
2.4 IEEE 802.11 Wireless Standards 28
2.4.1 Introduction 28
2.4.2 Physical Layer (PHY) 29
2.4.3 Media Access Control (MAC) Sublayer 30
2.4.4 Institute of Electrical and Electronics Engineers (IEEE) 802.11e: MAC Enhancements for Quality of Service (QoS) 36
2.4.5 Other IEEE 802.11 Admentments 40

3 Literature Review 42
3.1 Multimedia Characteristics 42
3.1.1 Throughput 43
3.1.2 Loss 43
3.1.3 Delay / Jitter 44
3.2 End-to-End Centric Approach 45
3.2.1 Multimedia over TCP and UDP 45
3.2.2 TCP-Friendly Rate Control Protocols 46
3.2.3 Summary 51
3.3 Network Centric Approach to QoS Provisioning 52
3.3.1 Network Service Models 52
3.3.2 Wireless Scheduling 53
3.3.3 Admission Control 55
3.3.4 Summary 57
3.4 Video Quality Performance Metrics 58
3.4.1 Objective Video Quality Assessment 58
3.4.2 Subjective Video Quality Assessment 62
3.4.3 Summary 63

4 Greediness Control Algorithm 65
4.1 Introduction 65
4.2 Architecture Overview 66
4.3 GCA Sender 67
4.3.1 High Level State Operation 68
4.3.2 Low Level Operation 69
4.3.3 Rate Estimation Equation 71
4.4 GCA Receiver 73
4.4.1 High Level Operation 73
4.4.2 Low Level Operation 75
4.4.3 Loss Event Rate Calculation 77
4.5 Summary 80

5 Analysis and Testing 81
5.1 Analysis 81
5.2 Simulation Setup 82
5.2.1 Simulation Environment: Network Simulator (NS) . . . 84
5.2.2 Simulation Topology and Settings 86
5.3 Simulation Analysis 89

5.3.1 Non-Adaptive Streaming 89
5.3.2 TFRC Based Streaming Solution 97
5.3.3 GCA Streaming Solution 108
5.3.4 Summary 116

6 Conclusion and Future Work 119
6.1 Conclusion 119
6.2 Contributions 121
6.3 Future Work 121

List of Figures

1.1 Typical residential home network of the future 2
1.2 Fair bandwidth distribution at the transport layer does not equal, equal video quality distribution at the application layer 5

2.1 Wireless multimedia streaming architecture 10
2.2 Wireless multimedia streaming protocol layering 11
2.3 Wireless multimedia streaming protocol encapsulation 12
2.4 Open System Interconnect (OSI) and TCP / IP conceptual layered models 15
2.5 Internet Protocol version 4 (IPv4) packet format 17
2.6 UDP packet format 18
2.7 TCP packet format 19
2.8 DCCPs high level connection initiation, data transfer and termination phases 23
2.9 Characteristics throughput for CCID 2 and CCID 3 24
2.10 RTP Data Packet Format 25
2.11 Scope of the IEEE 802.11 standard 28
2.12 IEEE 802.11 modes of operation 29
2.13 IEEE 802.11 timing structure 32
2.14 IEEE 802.11 hidden and exposed node problems 34
2.15 RTS / CTS mechanism 34
2.16 IEEE 802.11b multirate Physical (PHY) 35
2.17 High level EDCA structure 38
2.18 IEEE 802.11e prioritization mechanism 39
2.19 TXOP 40

3.1 Illustration of End to End approach 45

3.2 Comparison of the instantaneous sending rate of TCP and Video Transport Protocol (VTP) 51
3.3 Illustration of network centric approach 52
3.4 Stimulus presentation timing in Absolute Category Rating (ACR) 63
3.5 Stimulus presentation timing in Degraded Category Rating (DCR) 63

4.1 Where GCA fits into the TCP/IP reference model 67
4.2 GCAs Architecture 67
4.3 GCAs sender state machine 69
4.4 GCAs RTT calculation 70
4.5 Receive Feedback Procedure 72
4.6 GCAs receiver state machine 74
4.7 GCAs receiver process when data is received 76
4.8 GCAs receiver send feedback 77
4.9 Loss intervals 78
4.10 Average loss interval weights calculation 79
4.11 Loss intervals weights 79

5.1 Analysis of GCA throughput estimation equation with various α and β parameters 83
5.2 NS Architecture 85
5.3 Wireless streaming topology 87
5.4 Disney's Pixar Ratatouille trailer used for simulation 88
5.5 Non-adaptive streaming to High Definition Television (HDTV) 1080 and File Transfer Protocol (FTP) background traffic . . . 91
5.6 Non-adaptive streaming analysis with HDTV 1080 and HDTV 720 media devices together with an FTP background traffic source 94
5.7 Non-adaptive streaming analysis with HDTV 1080, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source 96
5.8 Non-adaptive streaming analysis with two HDTV 1080s, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source 98
5.9 Throughput analysis 100

5.10 TFRC streaming analysis for a HDTV 1080 and HDTV 720 media device together with an FTP background traffic source 103
5.11 TFRC streaming analysis for a HDTV 1080, HDTV 720 and HDTV 480 media device together with an FTP background traffic source 105
5.12 TFRC streaming analysis for a HDTV 1080, HDTV 720, HDTV 480 and HDTV 1080 media device together with an FTP background traffic source 107
5.13 Illustration of GCA simulation consisting of a HDTV 1080 and FTP background traffic source 110
5.14 Illustration of GCA simulation consisting of a HDTV 1080, HDTV 720 and FTP background traffic source 113
5.15 Illustration of GCA simulation consisting of a HDTV 1080, HDTV 720, HDTV 480 and FTP background traffic source . . 115
5.16 Illustration of GCA simulation consisting of a HDTV 1080, HDTV 720, HDTV 480, a second HDTV 1080 and an FTP background traffic source 117

List of Tables

1.1 Device characteristic video requirements 6

2.1 Summary comparison of IEEE 802.11 PHY characteristics . . 30
2.2 Default EDCA parameter set. aCWmax and aCWmin values are specified by the PHY parameters 38

3.1 International Telecommunication Union (ITU) 5-point quality scale 62
3.2 ITU 5-point impairment scale 63

4.1 GCA Sender initialisation parameters 70
4.2 GCA receiver initialisation parameters 75

5.1 IEEE 802.11g WLAN parameters 86
5.2 Video stream parameters 88
5.3 Simulation device parameters 88
5.4 Non-adaptive streaming analysis summary for HDTV 1080 together with FTP background traffic source 92
5.5 Non-adaptive streaming analysis summary with HDTV 1080 and HDTV 720 media devices together with an FTP background traffic source 93
5.6 Non-adaptive streaming analysis summary for HDTV 1080, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source 95
5.7 Non-adaptive streaming analysis summary for two HDTV 1080s, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source 97
5.8 TFRC streaming analysis summary for a HDTV 1080 media device together with an FTP background traffic source 101

5.9 TFRC streaming analysis summary for HDTV 1080 and HDTV 720 media device together with an FTP background traffic source 102
5.10 TFRC streaming analysis summary for HDTV 1080, HDTV 720 and HDTV 480 media device together with an FTP background traffic source 104
5.11 TFRC streaming analysis summary for HDTV 1080, HDTV 720, HDTV 480 and HDTV 1080 media device together with an FTP background traffic source 106
5.12 GCAs α and β parameters applied multimedia enabled devices for the duration of this simulation 108
5.13 GCA streaming analysis summary for a HDTV 1080 media device together with an FTP background traffic source 111
5.14 GCA streaming analysis summary for a HDTV 1080 media device together with an FTP background traffic source 112
5.15 GCA streaming analysis summary for a HDTV 1080, HDTV 720 and HDTV 480 media device together with an FTP background traffic source 114
5.16 GCA streaming analysis summary for two HDTV 1080's, a HDTV 720 and a HDTV 480 media device together with an FTP background traffic source 116
5.17 Summary of key results for Non-Adaptive, TFRC and GCA streaming solutions 118

Chapter 1

Introduction

This thesis presents the findings of work performed in the area of quality-oriented adaptive wireless multimedia streaming. More specifically, it is focused on multimedia streaming in a residential environment. It identifies an anomaly that arises in the multimedia streaming process where by, bandwidth that is distributed fairly between competing video streams at the transport layer results in unfair application layer video quality distribution. A new transport layer rate adaptation scheme called the Greediness Control Algorithm is proposed to correct this anomaly.

1.1 Home Multimedia Streaming

Historically networks were built to provide a certain type of service. Telephone lines only carried voice communication and television was only distributed via cable, satellite, or terrestrial systems. Each service had its own network. This historic separation between services and networks began to breakdown with the evolution of the Internet. Dial-up Internet access was now being carried by telephone lines. These Internet access technologies later matured and developed into Digital Subscriber Lines (DSLs) which was also carried by telephone lines.

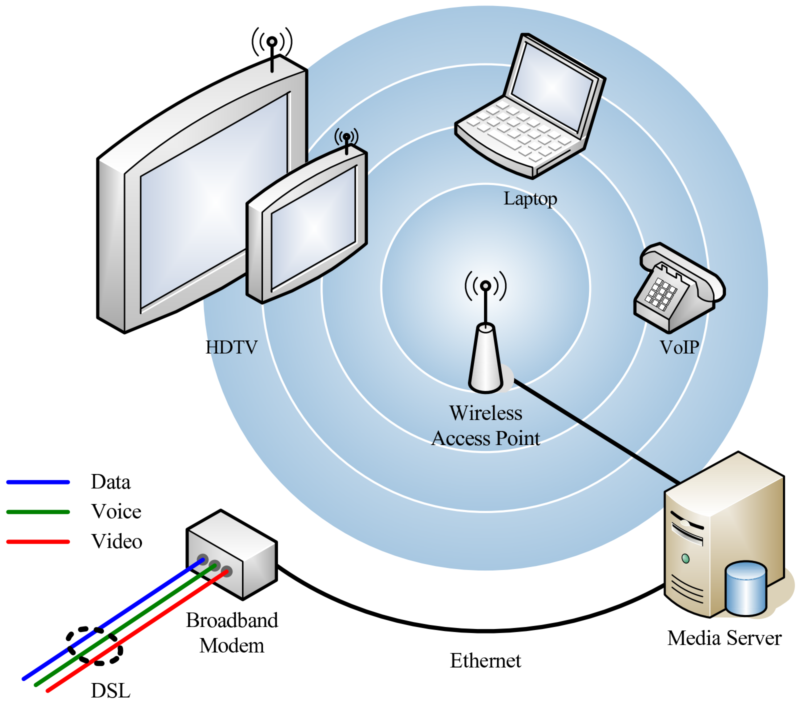


Figure 1.1: Typical residential home network of the future

In the future the role of these traditional network technologies will diminish and possibly become obsolete in the drive towards Internet Protocol (IP) convergence. This will lead to the creation on an ubiquitous network environment that would allow user to access the same content and services from a single or multiple service provider. This move toward an ubiquitous network environment has already had a huge impact on the telecommunications industry with the introduction of Voice over Internet Protocol (VoIP) services such as Skype[1] and Blueface[2]. This leaves only traditional television broadcast mediums as the final networks to converge with the Internet revolution. It is envisaged that data, voice and eventually television services will converge and be offered over a single IP-based broadband network. Homes

[1]http://www.skype.com
[2]http://www.blueface.ie

will receive all their service from their Internet Service Provider (ISP) [1].

The evolution of the Internet as a service oriented platform, coincides with evolution of multimedia devices. Today's multimedia devices are becoming increasingly sophisticated. Devices such as the iPod, XBOX 360 and Apple TV are revolutionising the way multimedia is experienced. Many devices now have huge Hard Disk Drives (HDDs) and most importantly wireless network connectivity. Imagine having the ability to watch the TV you want when you want. These technologies provide the building block for such scenarios. The a typical residential use case for this evolution is illustrated in Figure 1.1. Converged services are delivered to the home via a broadband Internet connection. These services are managed and distributed using a home media server. It is envisaged that these servers will intelligently learn your viewing tastes and download relevant content accordingly. Download media will then be streamed to the various multimedia enabled devices via the Wireless Local Area Network (WLAN). New architectures have been proposed in [2] for the delivery of this multimedia content for this very scenario.

Users will have a ability to subscribe (possibly using Really Simple Syndication (RSS)[3]) and download media to an in home media server using their broadband Internet connection. From here users can request and stream their downloaded content via their in-home wireless network to their High Definition Television (HDTV), Standard Definition Television (SDTV), laptop or mobile phone.

Precedents have already been set for these types of unscheduled broadcasts. TiVo[4] allows users to record TV and watch it when and where they want. Apples iTunes Store[5] offers users the ability to download an watch TV shows. The British Broadcasting Corporation (BBC)[6] has begun a project that will enable its viewers to access its archives through the Internet. Ap-

[3]http://www.rssboard.org/
[4]http://www.tivo.com/
[5]http://www.apple.com/itunes/
[6]http://www.bbc.co.uk

ple has also recently released Apple TV[7] allowing users to download movies via iTunes and stream them over an Institute of Electrical and Electronics Engineers (IEEE) 802.11n WLAN for play-out on a HDTV. Microsoft has also introduce products such as the Windows Home Server[8], Zune[9] and XBox 360[10], that can wirelessly stream multimedia. However these products and services are in their infancy and much more needs to be done before they reach mass market.

1.2 Problem Statement

This work concentrates on the delivery of content from the home media server to the various multimedia devices around the home. It does not consider the delivery of content from a service provider to the home media server.

Consider a typical residential IEEE 802.11g WLAN with a number of devices attached. Access to the wireless network is shared equally among these devices, resulting in them competing for and receiving a fair share of the available bandwidth. In general the streaming solutions will optimise video at the application layer to suit the characteristics of the device to which the media is being streamed. This discovery and optimisation mechanism is beyond the scope of this work. Once this coarse grained adaptation has taken place, fine grained adaptation is performed. This adaptation uses transport layer rate control feedback to further adjust the stream to suit the available network conditions. However these transport layer rate control mechanisms are based on transport layer congestion avoidance mechanisms that try to obtain as much bandwidth as possible while sharing bandwidth equally between competing streams. This results in greedy devices unfairly consuming excessive bandwidth that they do not necessarily require. By

[7] http://www.apple.com/appletv/
[8] http://www.microsoft.com/windows/products/winfamily/windowshomeserver/
[9] http://www.zune.net/
[10] http://www.xbox.com/

assuming that all devices have equal bandwidth requirements, there is an inefficient and unfair distribution of available bandwidth.

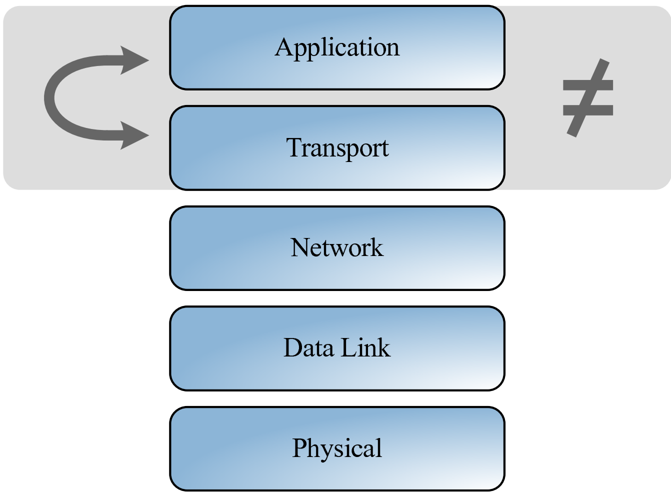


Figure 1.2: Fair bandwidth distribution at the transport layer does not equal, equal video quality distribution at the application layer

For example, consider the situation where three clients with various device characteristics, such as a 32” HDTV, 20” HDTV and 12” SDTV. Each device requests a unique H.264 video stream (see Table 1.1) from the media server to be streamed via the WLAN. If conventional transport layer rate control schemes were deployed in this scenario it would result in all clients receiving an equal share of available bandwidth. Assuming there is only *18 Mbps* of available bandwidth this may result in all devices receiving *6 Mbps* each. Based on their characteristics requirements this could result in the 32” HDTV receiving *66 %* of its required throughput, 20” HDTV receiving *100 %* of its required throughput and the 12” HDTV receiving *200 %* of its required throughput. Although this allocation of bandwidth might appear fair from a transport layer perspective, from the application layer’s Quality of Experience (QoE) point of view, this allocation is grossly unfair.

	Client 1	Client 2	Client 3
Device type	32' HDTV	20' HDTV	12' SDTV
Format	H.264	H.264	H.264
Resolution (pixels)	1920x1080	1280x720	640x480
Average Bit Rate (Mbps)	9	6	3

Table 1.1: Device characteristic video requirements

This problem stems from the fact that these rate control techniques do not consider the requirements of the media they are carrying or the device to which the media is being streamed. A fairer solution for this scenario would be for each of the clients to share the burden of the congested network equally. To overcome the applications greedy behaviour it is necessary to tune the parameters of the rate control algorithms to take into account the actual requirements of the device to which the media is being streamed. This can be achieved by introducing parameters that allow the control of the greediness of the rate control algorithm in order to achieve equal user satisfaction and increase overall QoE.

1.3 Contribution of this Work

This thesis proposes the Greediness Control Algorithm (GCA), an upper transport layer rate control mechanism designed to correct the fairness / quality inconsistency between the application and transport layer while streaming multimedia in a home environment. This is achieved by introducing α and β parameters that allow the aggressiveness of the rate control mechanism to be tuned to suit the characteristics of the multimedia device to which video is being streamed. This enables fair distribution of video quality at the application layer resulting in increased levels of user QoE.

1.4 Publications Arising from this Work

E. Casey, G.-M. Muntean, "A Priority-Based Adaptive Scheme for Wireless Multimedia Delivery", IEEE International Conference on Multimedia and Expo, Toronto, Canada, July 2006.

E. Casey, G.-M. Muntean, "A Mechanism for Greediness Management when Streaming Multimedia to Portable Devices", IEEE International Conference on Portable Information Devices, Orlando, USA, March 2007.

E. Casey, G.-M. Muntean, "Solution for Application and Transport Layer Inconsistency during Adaptive Multimedia Streaming", IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Orlando, USA, March 2007.

E. Casey, G.-M. Muntean, "TCP Compatible Greediness Control for Wireless Multimedia Streaming", IEEE 65th Vehicular Technology Conference, Dublin, Ireland, April 2007.

1.5 Thesis Structure

The thesis is structured as follows. Chapter 2 presents technologies related streaming multimedia in a wireless environment. This includes an overview of the multimedia streaming process followed by a detailed examination of the protocols involved in this process. It examines the application, transport, data link and physical layers of the streaming process. This chapter will provide background knowledge of the streaming process which is required for the thesis.

Chapter 3 provides a thorough examination of the literature pertinent to the proposed solution. This chapter presents and discusses works under three categories; end-centric approaches to provision of Quality of Service (QoS), network centric approaches to provision of QoS for streaming media and Transport Control Protocol (TCP) friendliness / compatibility issues.

Chapter 4 introduces and describes the proposed GCA for solving the problem outlined above in Section 1.2.

Chapter 5 assesses the performance of the proposed solution in various contexts and is compared with a number of variants of the standardised scheme.

The thesis concludes in Chapter 6 with a summary and discussion of results. Some ideas for future work are also outlined.

Chapter 2

Wireless Multimedia Streaming

This chapter presents various background material pertinent for streaming multimedia content in a residential WLAN. The chapter is divided into four main sections. Section 2.1 presents a brief introduction to the typical streaming scenario, outlining the various components and their interactions. Section 2.2 presents details of standards used for encoding multimedia content, while section 2.3 discuses the various transport and network layer protocols used for carrying this content. Finally, section 2.4 presents an overview of the major components of the IEEE 802.11 standard for WLAN.

2.1 Wireless Multimedia Streaming Process

Streaming multimedia in a wireless environment involves key elements that look after various stages of the streaming process. The major architectural elements of this process are illustrated in Figure 2.1. This solution consists of a server connected to a client over an IP-based network. In this case, IEEE 802.11 is used for the transport of multimedia between the server and the client. The server is connected to a wireless client via an IEEE 802.11 Access Point (AP) connected to the server using an Ethernet connection. These links form the Physical layer of the connection illustrated in Figure 2.2.

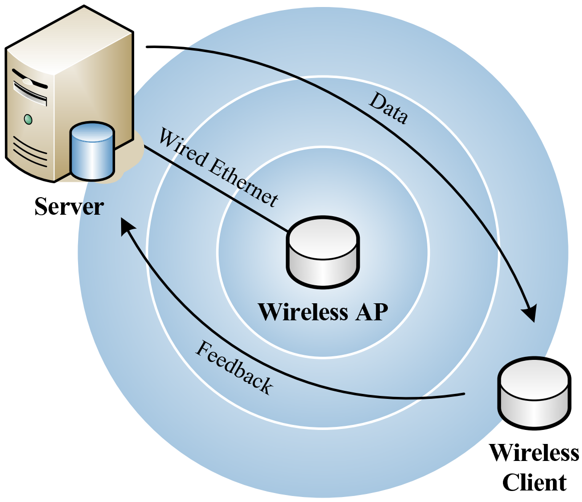


Figure 2.1: Wireless multimedia streaming architecture

The streaming process begins when a user switches on their multimedia enabled device. The device will automatically connect to the in-home WLAN. Once connected the devices will proceed to discover and negotiate a connection with the in-home media server using Digital Living Network Alliance (DLNA)[1] protocols. During this negotiation the device will inform the server its media requirements, such as screen size, resolution and network capabilities. The client will also download the Electronic Program Guide (EPG) from the server. The user now selects what program they want to view from the EPG. This interaction will be performed using an out of band protocol such as Hyper Text Terminal Protocol (HTTP) [3]. The server will then proceed to steam the requested multimedia content to the client via the WLAN.

The server contains a large repository of content downloaded from an Internet Content Distribution Network (CDN). The content is encoded in a scalable format to allow it to be easily temporally and / or spatially adapted to suit the characteristics of the client devices. When the server receives a

[1]http://www.dlna.org/

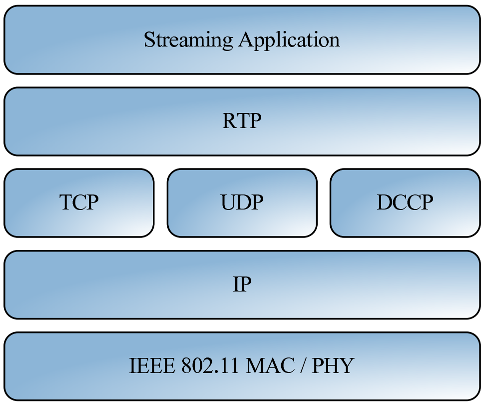


Figure 2.2: Wireless multimedia streaming protocol layering

request from a client for a particular media stream, the server will adapt and stream the requested media at desired bit rate and resolution negotiated by the client. This adaptation takes place at the application layer of the server. The adapted media is then passed to the Real-time Transport Protocols (RTPs) [4] protocol in the upper transport layer. RTP is responsible for framing, payload identification, sequencing and timing services. Once RTP services are applied the media is passed to the lower transports layer. This layer has three protocol options, User Datagram Protocols (UDPs) [5], TCPs [6] and Datagram Congestion Control Protocols (DCCPs) [7]. Each of these protocols have their own unique characteristics which are used for different streaming scenarios. In general they provide multiplexing, checksum and payload length services upper layers. The frame is now encapsulated in an IP packet. IP provides routing, addressing and fragmentation services on the network. The packet is now framed, scheduled and transmitted over the IEEE 802.11 WLAN, where is received by the wireless client device. The device will now reconstruct the data, decode it and play it to the user. Any errors that occur due to lost packets will be detected and the device's decoder will attempt to conceal these errors. The client will also transmit delivery related statistics in the form of feedback to the server which will use these statistics to adjust the sending rate of the media to suit the available network conditions.

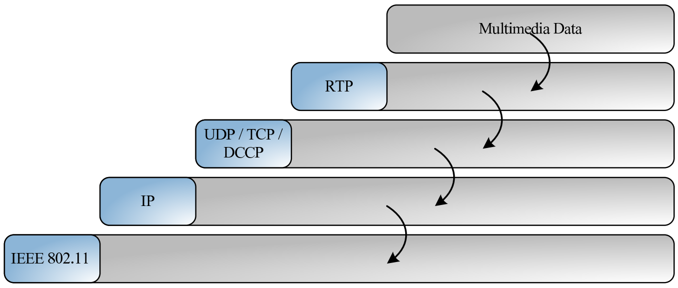


Figure 2.3: Wireless multimedia streaming protocol encapsulation

2.2 Multimedia Encoding Standards

2.2.1 MPEG-1

MPEG-1 [8] was the first standard developed by Motion Pictures Expert Group (MPEG) a working group of International Standards Organisation (ISO) / International Electrotechnical Commission (IEC). It defines the coding of multimedia content at bitrates of around 1.5 Mbps with resolutions of 320 x 240 pixels. This was motivated by the prospect that it would become possible to store video on a compact disc at a quality comparable to VHS.

MPEG-1 was published in five parts, Systems, Video, Audio, Conformance Testing and Software Simulation. Part 1 (MPEG-1 Systems) defines the syntax for combining multiple elementary audio and video streams into a single stream containing sequence and timing information, suitable for storage or transmission. Part 2 (MPEG-1 Video) defines a number of lossy and lossless compression techniques for reducing temporal and spatial redundancy in video sequences. This is achieved by first decomposing image into the three component RGB space and then converting this into the YUV. The YUV space is now divided into macroblocks and a Discrete Cosine Transform (DCT) transformation is applied to each block and is then Quantized. A zig-zag pattern in conjunction with run length encoding is now used to increase compression. Motion compensation and motion estimation is then used to identify casual temporal redundancies between pictures. Part 3 (MPEG-1 Audio) defines filters and sub-sampling mechanisms that exploit redundancies in the psychoacoustic model in order to achieve compression. Part 4 defines conformance testing, which specifies the methodology for verifying claims of conformance to the standard by manufacturers of equipment and producers of bitstreams. Part 5 proposes a full C-language implementation of the MPEG-1 standard (encoder and decoder).

2.2.2 MPEG-2

The MPEG-2 [9] standard was jointly developed by both the ISO/IEC and International Telecommunication Union (ITU). It was published in four parts. Part 1 (MPEG-2 System) specifies the system coding layer of the MPEG-2. It defines the multiplexing structure of elementary streams, that have a common time base. It is useful as a representation mechanism for audio and video data synchronization of elementary streams. It is designed for use in relatively error free environments. Part 2 (MPEG-2 Video) specifies the coded representation of video data and the decoding precess required to reconstruct pictures. It operates in a similar manner to MPEG-1 Video. However unlike MPEG-1, MPEG-2 targets very high bit rates of around 6 Mbps. It also introduces flexibility through the use of profiles and levels. Part 3 (MPEG-2 Audio) specifies the coded representation of audio data. It introduces multi-channel audio extensions. Part 4 specifies conformance testing mechanisms.

2.2.3 MPEG-4

MPEG-4 [10] is another ISO/IEC standard developed by the MPEG. It was originally intended as a standard for compressing audio and video at very low bit rates. However, the specifications for content-based compression opened many other possibilities for object manipulation, interactivity, rights management, inclusion of other types of media, so the final standard evolved in a framework for interactive multimedia content manipulation and management. It has been developed as an open standard to encourage interoperability and widespread use. As a result MPEG-4 has enjoyed wide acceptance in the research and commercial community due to its high bitrate scalability and compression efficiency. MPEG-4 is the successor to MPEG-1 [8] and MPEG-2 [9].

Like MPEG-1 and MPEG-2, the MPEG-4 standard has many parts. In total there are 23 parts to the MPEG-4 standard, of which the main ones are listed as follows. Parts 1 - 5 have similar purpose to their MPEG-2 counterparts. Part 1 (MPEG-4 Systems) describes synchronisation and multiplexing of video and audio streams. Part 2 (MPEG-4 Visual) defines the compression codec for visual data. Part 3 (MPEG-4 Audio) specifies compression codecs for perceptual coding of audio signals. Part 4 describes procedures for conformance testing. Part 5 provides reference software. Part 8 specifies procedures for transport of MPEG-4 data on IP networks. Part 10 (MPEG-4 Advanced Video Coding (AVC)) defines encoding techniques for video signals which is technically identical to the ITU-T H.264 standard. While parts 12, 14 and 15 define file formats for storing MPEG-4 content.

2.3 Network and Transport Layer Protocols

2.3.1 Internet Protocol (IP)

The vast majority of network enabled multimedia application make use of an IP-based network layer. All other layers are variable depending on the applications requirements or the physical medium to which nodes on which the application is running are connected to. As a result IP is a major component in the multimedia streaming process, providing essential services to higher layers in the TCP/IP (see Figure 2.4) conceptual model.

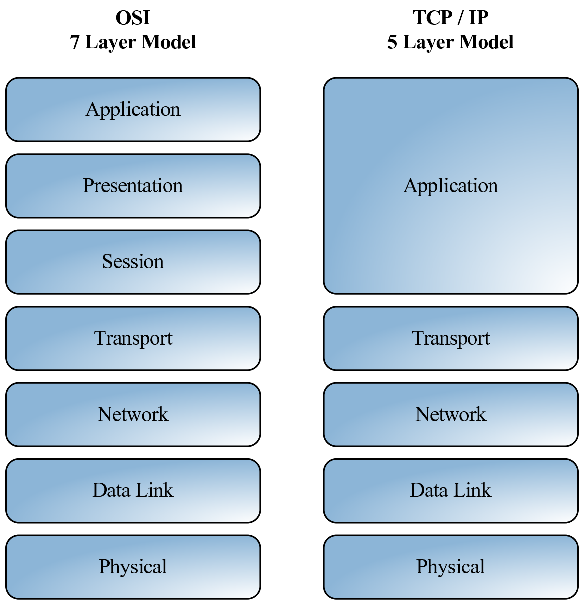


Figure 2.4: OSI and TCP / IP conceptual layered models

The services provided by IP can be seen as somewhat analogous to the postal service. In the traditional postal service, letters (data) is placed in envelopes (packets) which are marked with a destination addressed (IP address) and placed in a postbox (buffer) at any point in the postal system (network). Post boxes deliver messages to sorting centres (routers) that deliver letters to their required destination. Users of this network are willing to accept delays and loss.

IP is a network layer protocol. Like the postal network, an IP-based network is a connectionless network, in that IP enabled node does not know the actual route to the destination before a packet is transmitted. Like to postal network, IP provides a number of essential services: addressing, routing and fragmentation, to enabled applications utilise physical layer protocols such as Ethernet or IEEE 802.11 WLANs. It does not provide any re transmission, multiplexing or reordering services of packet, rather it relies on high layers such as UDP, TCP or DCCP which will be discussed in later Sections. Most importantly, IP is best effort protocol that is unaware of the content it carries or the route taken to deliver this content.

There are currently two version of IP, Internet Protocol version 4 (IPv4) [11] and Internet Protocol version 6 (IPv6) [12]. It is envisaged that IPv6 will eventually completely take over from IPv4. The primary difference between the two protocols is the larger address space provided by IPv6. IPv4 can support $2^{32} = 4,294,967,296$ addresses while IPv6 supports $2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$ addresses. IP addressing plays a fundamental role in the operation of any IP enabled network. It essentially provides the ability for networked nodes (hosts and router's) to uniquely identify each other. It also assists the routing protocols forward packets to their destination. Routing is the process of forwarding packets between nodes based on decision algorithms in order to route packets to their final destination. Routing resolves around a loose hierarchical structure, with a nodes IP address representing its Point of Attachment (POA) to the network. When a router receives a packet, it examines its destination IP address. This address is then compared with entries in the router routing table to determine on which port to forward this packet. The router will forward the packet to the next hop or the final destination. The routing tables used for forwarding decisions are compiled using routing protocols such as Open Shortest Path First (OSPF) [13] or Routing Information Protocol (RIP) [14].

The IPv4 packet, shown in Figure 2.5 consists of a number of fields that assist the operation of the protocol. The version field identifies the version of IP used, while the header length field gives the length of the header in terms of 32 bit words. Next, the type of service field was originally intended to specify the how an IP datagram would be handled as it traversed the network. It is now used for DiffServ [15] and Explicit Congestion Notification (ECN) [16]. The total length field defines the entire length of the datagram (header and payload). The identification field is primarily used for uniquely identifying fragments of an original IP datagram. The flags and fragment offset fields are is also used for fragmentation.The time to live field helps prevent datagrams from persisting in the network for too long. The protocol field defines the protocol used in the payload of the datagram. Header Checksum field is used for error-checking of the header. Source address and destination contain the 32 bit IP addresses of the source and destination of the datagram. Finally a rarely used options field ends the header.

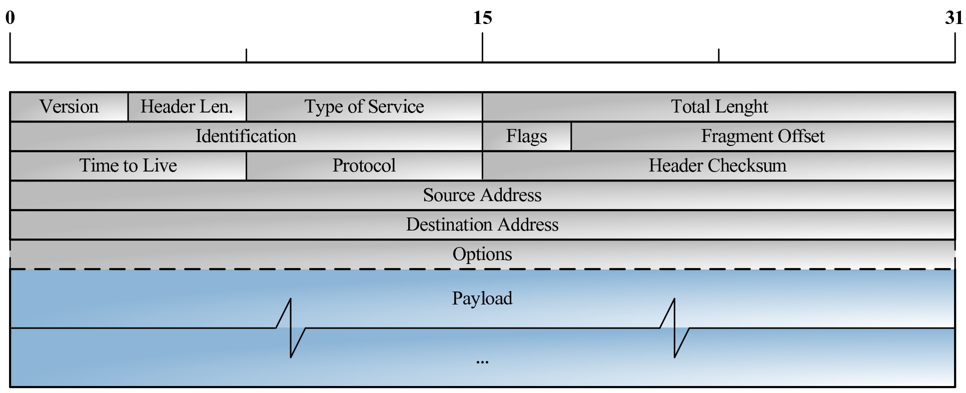


Figure 2.5: IPv4 packet format

2.3.2 User Datagram Protocol (UDP)

User Datagram Protocol (UDP) [5] is a connectionless transport protocol. It provides the basic functionality required for applications to send encapsulated

IP datagrams without having to establish a connection.

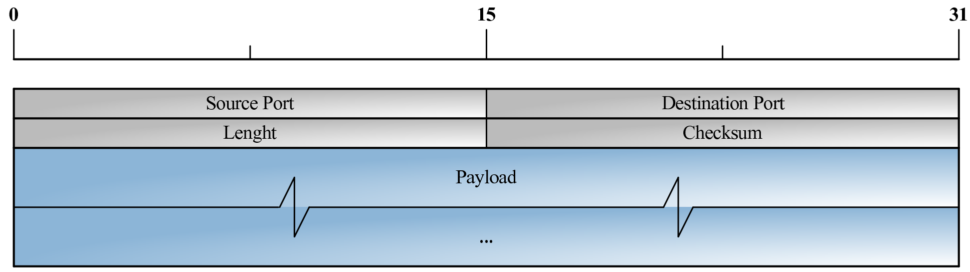


Figure 2.6: UDP packet format

A UDP datagram (see Figure 2.6) consists of a 8 byte header followed by a payload. The header consists of 4 x 2-byte fields: source port, destination port, length and checksum. The source and destination ports provide required information to allow transport layer daemon processes to route packets to their correct destination application. This multiplexing / demultiplexing feature is the main benefit UDP has over raw IP datagrams. The 16-bit length field specifies the length of the datagram in bytes of the entire datagram (header and data). The field size sets a theoretical limit of 65,527 bytes for the data carried by a single UDP datagram. Finally, a 16-bit checksum field is used for error-checking of the header and data.

UDP does not provides any reliability or congestion control features. As a result applications using the protocol must generally be willing to accept or deal with loss, duplication or out-of-order delivery and rely on network-based mechanisms to minimise potential of congestion collapse. The majority of applications using UDP often do not require reliability mechanisms and may even be hindered by them. Applications requiring high degrees of reliability should use a protocol such as TCP. These characteristics make UDP well suited for real-time multimedia streaming applications.

2.3.3 Transport Control Protocol (TCP)

Transport Control Protocol (TCP) [6] [17] is a reliable, connection-oriented, congestion controlled byte stream service.

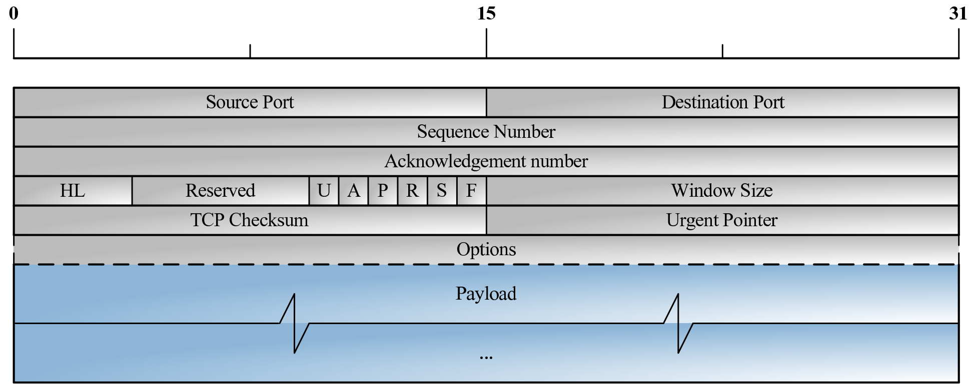


Figure 2.7: TCP packet format

A TCP packet consists of a 20 byte header followed by a payload as illustrated in Figure 2.7. The header includes a number of fields that enable the provision of TCPs key services. In the same way as UDP, TCP uses 16 bit source and destination port number fields for multiplexing data to various sending and receiving processes. The 32 bit sequence number field identifies the byte in the stream that the first byte of data in the segment represents. This field enables the reordering of out-of-order packets. The 32 bit acknowledgement field contains the sequence number of the next data segment the receiver expects to receive. this allows the sender to identify packets that have not been received yet. These two fields are essential for providing a reliable delivery service. 4 bit data offset / header length field specifies the length of the header. This is followed by a 6 bit field reserved for future use. Next, there are 6 flag bits. URG (U) is used to determine if the value in the urgent pointer field is valid. If set, the urgent pointer contains a sequence number offset, which corresponds to a TCP segment that contains urgent data and it should be expedited to its destination. ACK indicates if the acknowledgement number field is significant. It is used to by the receiver to inform the server that the packets it received are in order and intact. PSH is used to minimize the amount of buffering used before passing the data in this packet to the receiving process. The RST flag used to reset the connection, while the SYN and FIN flags are used for establishing and closing the TCP connection. The 16 bit window size field specifies the number of bytes the each end of the connection is willing to accept beginning with the one specified by the acknowledgement number. This field will enables connection flow control. Finally a checksum field covers the header and payload of the TCP segment.

Flow control is achieved by TCP using the window size field. This field identifies the number of bytes, starting with the byte acknowledged, that the receiver is willing to accept. If a receiver is busy or does not want to receive more data from the sender, this value can be set to 0. In addition to the flow control based on the window size TCP uses other complementary congestion control mechanisms such as Slow Start and Additive Increase, Multiplicative Decrease (AIMD). The slow start mechanism employed by TCP means that TCP data tries to avoid congestion by starting the transmission at a low rate and increasing the rate gradually to an acceptable level. AIMD means that the rate of transmitted data is increased slowly while the network appears capable of sustaining the current rate (i.e. no packet loss occurs), but as soon as the this rate appears excessive due to identification of lost packets the sender will dramatically reduce the data rate.

TCP is used for a number of best effort applications such as HTTP for web browsing and File Transfer Protocol (FTP). These applications are not time critical but require guarantees that the integrity of received data is maintained. For this reason is not the preferred choice for streaming media. Streaming media requires video delivered in a timely manner, maintain stable throughput while tolerating some loss. However, some research [18] has proposed TCP as the better mechanism for streaming media.

2.3.4 Datagram Congestion Control Protocol (DCCP)

Historically the majority of Internet traffic used TCP for both is reliability and congestion control features, while UDP was used for short request response transfers that wanted to avoid these features. UDP applications tended not to implement their own congestion control mechanisms. However, since UDP traffic volume was small in comparison to the congestion controlled TCP flows, the lack of this mechanism did not lead to network collapse.

As mentioned above, recent years have seen significant growth in streaming application that utilise the characteristic features inherent in the UDP protocol. These applications share a preference for timeliness over reliable delivery that make UDP ideal protocol choice. However the growth of this long-lived non-congestive controlled traffic poses a real threat to network stability. In most cases streaming applications employ their own congestion control mechanism. However experience has shown that congestion control is difficult to get right and many application writers would like to avoid reinventing the wheel. As a result the Internet Engineering Task Force (IETF) have proposed and standardised Datagram Congestion Control Protocol (DCCP) [7].

DCCP is a connection oriented transport layer protocol, providing congestion controlled, unreliable delivery mechanism for unicast flows. It is most beneficial for to streaming application that are willing to sacrifice in order reliable delivery for lower delay. It combines the benefits of congestion control offered by TCP with those offered by a UDP like connection less protocol. More specifically DCCP provides the following features:

- Unreliable flows of datagrams
- Reliable connection setup and teardown
- Reliable negotiation of options, including negotiation of a suitable congestion control mechanism

- Acknowledgement mechanisms
- Modular Congestion Control Mechanisms: TCP-like Congestion Control [19] and TCP Friendly Rate Control (TFRC) [20]. DCCP is easily extendible to further forms of unicast congestion control.

Connection Dynamics

DCCPs high level connection dynamics are similar those employed by TCP. Connections progress through three distinct phases: initiation, transfer and termination (see Figure 2.8). Although DCCP employs an Acknowledgment (ACK) framework, the information carried by these ACK packets is used for determining congestion control information. Unlike TCP, it is not used for reliable delivery. Applications wishing to employ a full or partially reliable delivery must do so at the application layer. DCCP can be formulated as shown in Equation 2.1 and Equation 2.2.

$$DCCP = TCP - BytestreamSemantics - reliability \tag{2.1}$$

$$DCCP = UDP + CongestionControl + Handshakes + Acknowledgements \tag{2.2}$$

Congestion Control Mechanisms

The major advantage associated with the use of DCCP for streaming applications is that it employs modularised congestion control framework. This gives developers the choice congestion control mechanisms or the option to implement their own. The mechanisms are identified by single byte Congestion Control IDs (CCIDs). The end-points negotiate their CCIDs during connection initiation. Currently CCIDs 2 and 3 are defined and 1, 2 and 4 - 255 are reserved.

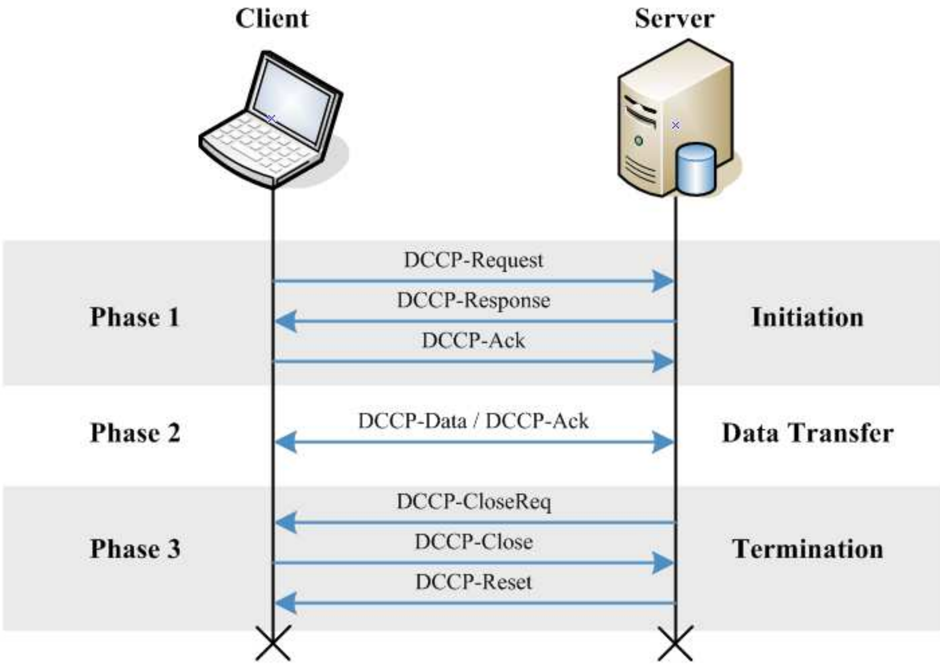


Figure 2.8: DCCPs high level connection initiation, data transfer and termination phases

CCID 2 provides TCP-like Congestion Control [19]. This mechanism is designed to emulate the behaviour of TCP congestion control mechanism. It is a window based mechanism that echos the operation of its TCP counterpart. Essentially a sender maintains a congestion window and send packets until window is full. Receiver acknowledges packets using a Selective Acknowledgement (SACK) based scheme. Dropped or ECN marked packets indicate congestion and case the congestion window to be halved. The characteristic throughput response of this CCID is illustrated in Figure 2.9.

CCID 3 provides TFRC [20]. TFRC is an equation based congestion control mechanism that provides a smoother response to congestion than CCID 2. It is designed to compete fairly with TCP over the long term. This can lead to throughput inaccuracies in the short term. TFRCs characteristic throughput response is illustrated in Figure 2.9.

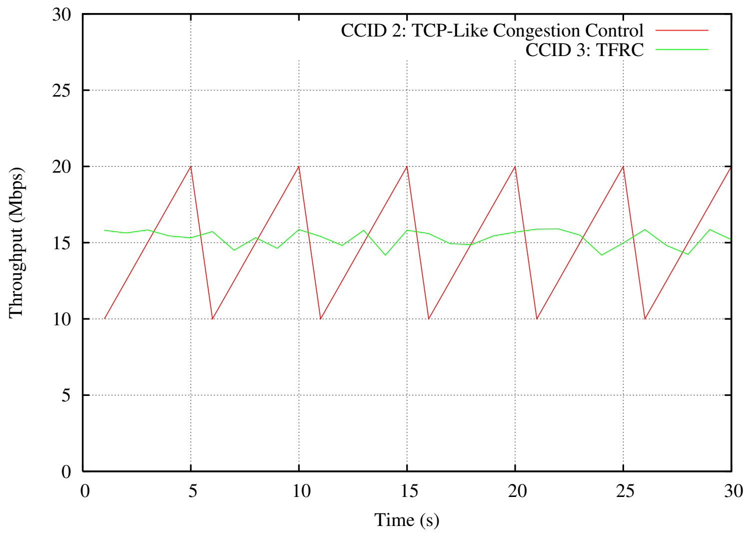


Figure 2.9: Characteristics throughput for CCID 2 and CCID 3

2.3.5 Real-time Transport Protocol (RTP)

Overview

Real-time Transport Protocol (RTP) [4] is an upper transport layer / lower application layer protocol which provides services for end-to-end delivery of data with time sensitive characteristics. The services offered by RTPinclude media framing, payload type identification, sequence numbering, time stamping and delivery monitoring. RTP is typically run on top of an existing transport layer protocol such as UDP or DCCP to make use of their multiplexing and checksum services. It is important to note that RTP does not provide mechanisms to ensure timely delivery, guarantee delivery, prevent out-of-order delivery or provide QoS guarantees. Rather, it relies on over and underlying protocols to utilize the services it offers in order to provide some of these requirements. It should also be noted RTP is designed for to be integrated into the application processing rather than be implemented as a separate layer.

The RTP specification actually defines two separate protocols. The first one is the RTP, defines the transport and delivery mechanics for carrying data with real-time properties. The second one is called the RTP Control Protocol (RTCP), an out-of-band signaling mechanism for monitoring quality of service and convey information about the participants in an on-going session.

RTP Data Transfer Protocol

The RTP portion of the specification defines the packet structure required for the transport of time sensitive data. The basic structure (see Figure 2.10) of a RTP data packet consists of a header followed by a payload. Note that the header does not contain a payload length field, checksum or port numbers. It relies on the underlying transport protocol to provide this functionality. As mentioned above RTP is generally carried by UDP and DCCP, which provides the length and checksum information as well as the multiplexing needs.

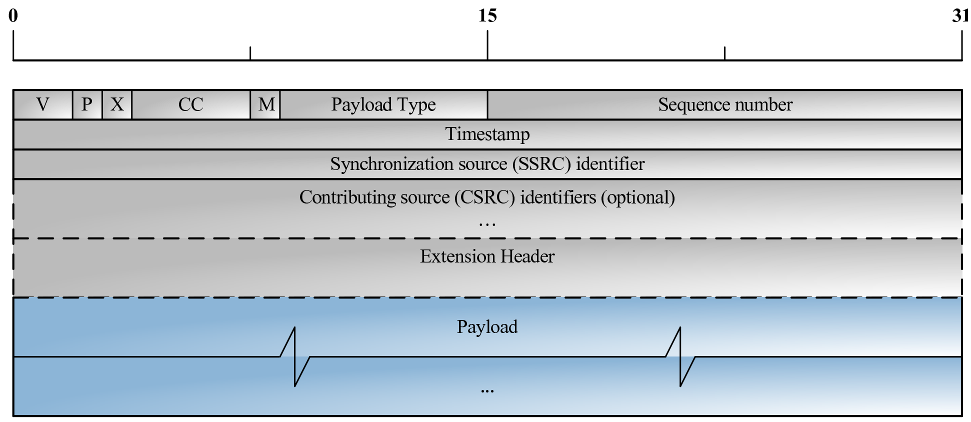


Figure 2.10: RTP Data Packet Format

The first two bits of the packet header identify the version of RTPused. The version defined by the specification in [4] is two. The next bit (P) indicates whether padding is used. Padding referees to the number of additional padding octets at the end which are not part of the payload. Padding is generally required by some encryption algorithms that require fixed block sizes. The extension bit (X) specifies if the header contains an extension header. This bit indicates that the fixed header is followed by exactly one extension header. Next the 4 bit CSRC (CC) count specifies how many contributing sources are specified in the RTP header. The one bit marker (M) is profile specific. It is intended to allow significant events to be marked in the packet stream (i.e. frame boundaries). Next, 7 bits are used to describe the payload carried. They define the format of the data carried.

Sequence number (16 bits) increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The 32 bit timestamp field specifies the sampling instant of the first octet in the RTP data packet. The clock frequency is dependent on the format of data carried in the payload. In the case of audio, the timestamp is normally incremented by the number of samples in the packets and not the amount of time that has passed since the last packet was transmitted. This allows the receiver to determine the exact play out time of the media carried. For video, a single frame may need to be transmitted using multiple packets. In this case each packet will contain incremental sequence number but the timestamp field will be the same in all packets.

The 32 bit Synchronization Source (SSRC) field uniquely identifies the sender of the RTP packets. This allows applications that support multiple sessions to determine which data is associated with which stream. Finally, a 32 bit contributing source (CSRC) identifies the number of contributing sources for the payload contained in the packet. The number of identifiers is given by the CC field. CSRCs are used by mixers, using SSRCs of the data which is contained within the payload of the packets.

RTP Control Protocol (RTCP)

The RTP protocol is complemented by an out-of-band control protocol RTCP. RTCP packets are periodically transmitted by every participant to every other participant in an RTP session. RTCP provides the following functionality:

- Their primary function is to provide feedback on the quality of the data distribution. This information allows applications to implement flow and congestion control functionality. It can also be used to control adaptive encoding schemes. The feedback can also be used to diagnose faults in the distribution chain.

- RTCP carries a persistent transport-level identifier for an RTP source called the canonical name or CNAME. This identifier allows RTP sessions to group certain stream together (i.e. groups audio and video together for synchronisation purposes). This information is not provided by RTP itself.

- RTCP packets must be rate controlled to prevent scalability issues. For this reason each participant independently observes the number of participants in a session by listening to other participants RTCP packets and adjusts the rate at which theses packets are sent.

- The final (optional) function provides a mechanism for the distribution of minimal session control information about a participant.

There are five types of RTCP packets that supply the above functionality. Sender Report (SR) are used for the transmission and reception statistics of participants that are active senders. Receiver Reports (RR) are sent by participants that are not active senders for conveying reception statistics. Source Descriptions (SDES) contain information which describes the participant while Application (APP) packets contain application specific data.

Finally, a BYE packet is used to indicate that a participant is about to leave a session.

2.4 IEEE 802.11 Wireless Standards

2.4.1 Introduction

The IEEE 802.11 [21] is a member of the IEEE 802 family, which is a series of specifications for Local Area Network (LAN) technologies. The IEEE 802 specification is focused on the lowest two layers of the OSI conceptual model [22]. All 802 networks have Media Access Control (MAC) and Physical (PHY) Layer components. The MAC layer defines the mechanisms that manage and control the access to the medium and the PHY controls the actual transmission and reception of data on the medium. The IEEE 802.11 specification defines these MAC and PHY components. The original IEEE 802.11 specification defined a MAC sublayer and two physical layer components. Later revisions and additions to the standard introduced new PHY components that specified higher data rates and MAC components which introduced QoS support.

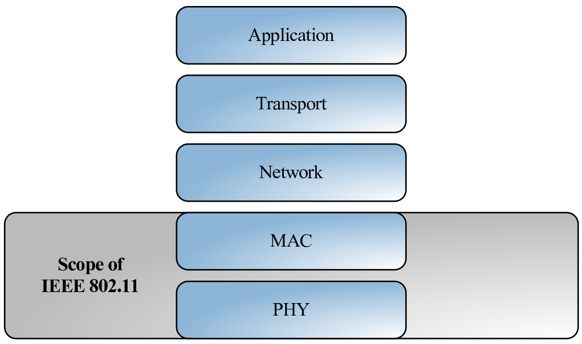
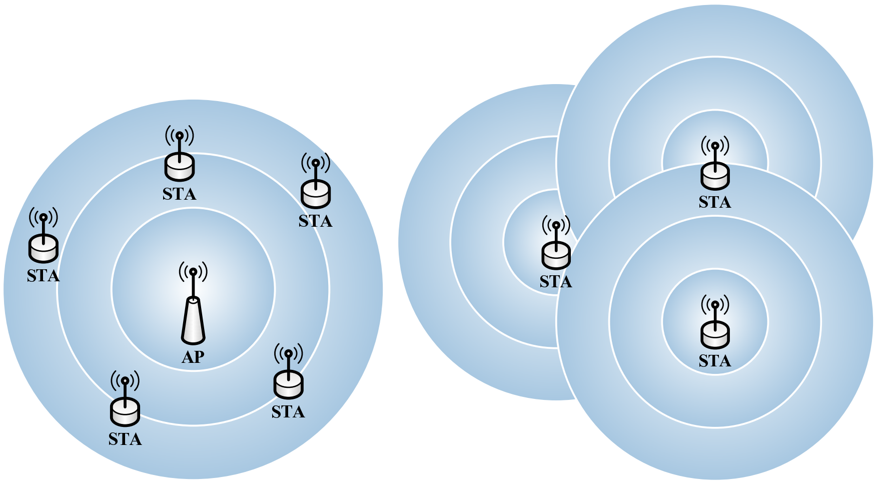


Figure 2.11: Scope of the IEEE 802.11 standard

An IEEE 802.11 network consists of three major physical components; Station (STA), AP and the wireless medium. The basic building block of a wireless network is the Basic Service Set (BSS) which is a group of STAs that communicate within a Basic Service Area (BSA). STAs within a BSA can communicate with other members of their BSS. A BSS can operate in either Ad-Hoc or Infrastructure mode as shown in Figure 2.4.1. Infrastructure BSSs are WLANs that include an AP. An AP handles all communication between STAs within a BSA. Ad-hoc mode is where a group of STAs within a BSA communicate directly with one another without the involvement of an AP. Ad-hoc networks are generally referred to as Independent Basic Service Set (IBSS).



(a) Infrastructure Mode (b) Ad-Hoc Mode

Figure 2.12: IEEE 802.11 modes of operation

2.4.2 Physical Layer (PHY)

Overview

The IEEE 802.11 PHY defines the modulation and transmission characteristics of a WLAN. A number of different PHY layers exist in the 802.11

standard each supporting the same MAC layer. For example, IEEE 802.11e can be used in conjunction with the IEEE 802.11a, IEEE 802.11b or IEEE 802.11g PHY. To achieve this degree of modularization the PHY is divided into two sub layers: the Physical Layer Convergence Procedure (PLCP) and Physical Medium Dependent (PMD). The PLCP is the interface between the MAC and the radio transmission. The PMD is responsible for transmitting any bits it receives from the PLCP into the air using the antenna. The physical layer also incorporates a Clear Channel Assessment (CCA) function to indicate to the MAC when a signal is detected. An overview of the various physical layer characteristics is outlined in Table 2.1.

	Legacy	802.11a	802.11b	802.11g	802.11n
Frequency Band (GHz)	2.4	5.0	2.4	2.4	2.4 / 5.0
Channel Width	22	18	22	22	
Indepentent Channels	18	3 / 4	3 / 4		
Indoor Range (m)	15	25	35	35	75
Outdoor Range (m)	75	100	125	125	150
Modulation	DSSS	OFDM	DSSS	OFDM / DSSS	OFDM
Max Data Rate (Mbps)	2	54	11	54	248
Typical Throughput (Mbps)	0.75	28.0	7	27.0	74.0

Table 2.1: Summary comparison of IEEE 802.11 PHY characteristics

2.4.3 Media Access Control (MAC) Sublayer

The IEEE 802.11 legacy MAC [21] specifies two coordination functions, which determine when a station operating within a BSS is permitted to transmit and receive frames from the wireless medium. These functions are necessary as only a single station can transmit on the medium at any given time. The mandatory Distributed Coordinator Function (DCF) is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and the optional Point Coordinator Function (PCF) is based on a pooling mechanism. The DCF enables distributed contention based access, while PCF provides contention free access to the wireless medium. Originally, it was hoped that the PCF would provide support for the QoS needs of real-time applications. However, due to its inherent complexity and incomplete standardization it has not reached mass market penetration. Most of todays IEEE 802.11 devices operate in the DCF mode only.

Point Coordinator Function (PCF)

PCF is optional MAC access mechanism in the IEEE 802.11 standard. It was invisiged that it would provide support for time-bound services by allowing STAs to have priority access ot the wireless medium. This access is coordinated using a Point Coordinator (PC) which usually resides in the AP in infrastructure mode.

When a BSS is using PCF the medium is divided into repeating Contention Free Period (CFP) and Contention Period (CP) timing intervals called superframes. Superframes begin with a beacon. PCF is used for accessing the channel during the CFPs, while DCF is used during the CPs. Beacons are management frames that allow STAs maintain synchronisation with the AP. During the CFP the PC/AP will poll STAs for pending frames and deliver any pending downstream frames. The PC will contiunue pooling other STAs until the CFP ends, at which point a *CFP-End* control frame is transmited by the PC to signal the end of the CFP.

Distributed Coordinator Function (DCF)

The DCF uses CSMA/CA to regulate access to the shared wireless medium. It is designed to reduce the probability of collisions using a combination of physical and virtual channel sensing. When a STA wants to transmit, it senses the medium to determine whether or not it is busy. If the medium has been sensed idle for a time interval called DCF interframe space Distributed Inter-Frame Space (DIFS), it proceeds to transmit the frame immediately.

However, if the medium is sensed busy, the station must defer its transmission attempt until the medium becomes idle again. Once the the medium becomes the STA must wait a DIFS and then enter into the backoff procedure delay. A backoff delay is calculated as a function of the Contention Window (CW) using Equation 2.3. For the first transmission attempt the CW is set to the minimum value CW_{min}. It is doubled for every unsuccessful transmission attempt up to a maximum value CW_{max}. If a successful transmission is achieved the CW is reset to the CW_{min} value. During this backoff procedure the backoff timer is decremented for each time slot that the medium remains idle. Should the medium become busy during this period the timer is paused. It is resumed once the medium is sensed idle for a duration of DIFS. The STA is permitted to transmit once the backoff timer reaches zero. A positive acknowledgment frame (ACK) is used to inform the sender that the frame has been successfully received. A receiver returns an ACK frame after a Short Inter-Frame Space (SIFS). If a sender does not receive an ACK within $ACK_{timeout}$, it assumes the packet has been lost due to collision or erroneous frame and reschedules the transmission by running the backoff procedure again. The above timing sequence is illustrated in Figure 2.13. The values of the MAC parameters used above are dependent on the underlying PHY.

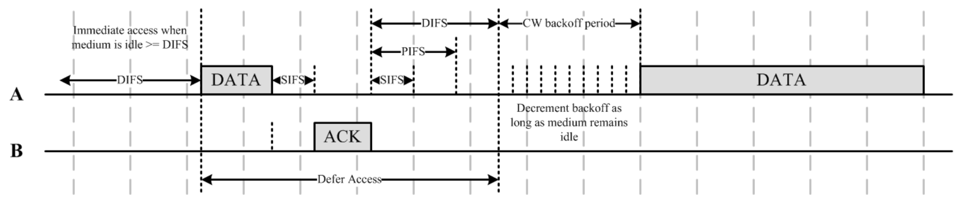


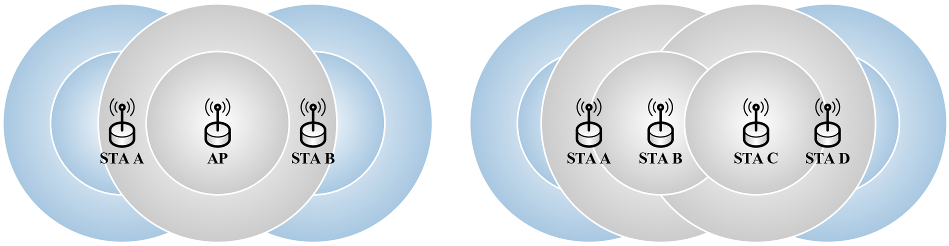
Figure 2.13: IEEE 802.11 timing structure

$$backoffDelay = random[0, CW] * slotTime \tag{2.3}$$

Request to Send (RTS) / Clear to Send (CTS)

The IEEE 802.11 standard employs a number of mechanisms to reduce the impact of collisions and errors experienced bay STAs on a WLAN. RTS / CTS is one such mechanism used to reduce frame collisions introduced by hidden and exposed node problems. The hidden node problem arises when two STAs, that are both within range of an AP but not each other, attempt to communicate with the AP that is within range of both. For example, consider the WLAN topology illustrated in Figure 2.14(a). STA A initiates a communciation with the AP using the CSMA/CA protocol. During STA A's transmission, STA B sucessfully initiates the CSMA/CA procedure and attempts to transmit its packet. Since STA B cannot hear STA A, both STAs transmit their packets at the same time causing a collision at the AP. These STA are know as hidden nodes. The exposed node problem occurs when a node is prevented from sending packets to other nodes because of a transmission from a neighbouring node. For example, consider the WLAN topology shown in Figure 2.14(b). In this example STAs B and C want to communicate with STAs A and D respectively. CSMA/CA will prevent this transmission from occurring even though STA A can receive STA B's tranmission without interference from the STA C's transmission because it is out of range of STA C.

The RTS / CTS handshke procedure is used in conjunction with CSMA/CA to overcome these inefficiencies. A STA wishing to transmit a frame on the WLAN first performs the usual CSMA/CA procedure followed by an RTS / CTS handshake (see Figure 2.15). This handshake involves the transmitting STA sending a RTS broadcast to all nodes within its carrier sense range. This causes all nodes that received the RTS broadcast to not contend for the medium for the duration time specified by the Network Allocation Vector (NAV) field in the RTS frame. Only the intended receiver of the data frame will respond to the RTS with a CTS, which is also received by all STAs within its range who also not contend for the medium for the duration time



(a) Hidden node (b) Exposed node

Figure 2.14: IEEE 802.11 hidden and exposed node problems

specified by the NAV field in the CTS frame. The transmitting STA can now proceed with the transmission of the data frame. Although this handsake reduces the number of collisions it also increases the overhead required to transmit a packet. As a result RTS / CTS implementations often use a frame size threshold under which no handshake is used.

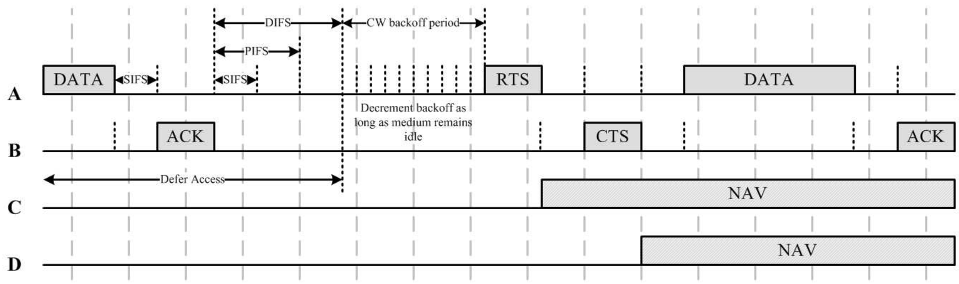


Figure 2.15: RTS / CTS mechanism

Multirate Support

IEEE 802.11a/b/g amendments enable support for enables support for multirate MAC. This provides nodes with the ability to dynamically adjust their PHY data rate in order to improve performance. Performance degration generally arises due to increased error rates due to poor signal quality caused by noise or interference, resulting in dropped frames. Changes in the levels of noise and interference generally occur due to a STA moving away from and AP or an object moving into the path of a STAs signal. This problem is generally compounded by the fact an increase in symbol rate leads to a increases the probability of an incorrect detection.

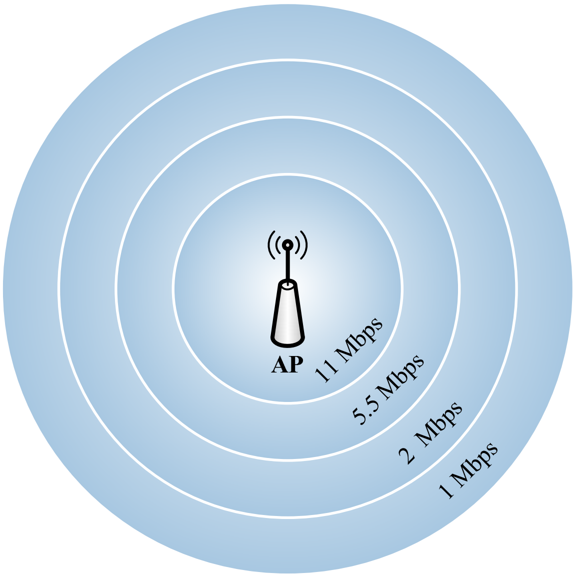


Figure 2.16: IEEE 802.11b multirate PHY

The IEEE 802.11 standard addresses this issue by offering multiple PHY modulation schemes. For example in IEEE 802.11b amendment there are four PHY modulation schemes providing data rates of 1,2, 5.5 and 11 Mbps. These data rates can be visualised as transmission zones radiating from a IEEE 802,11 enabled nodes. Although IEEE 802.11 allows nodes to change their PHY data rate, it does not actually specify the mechanism for doing so dynamically. The implementation of such a mechanism is left to the equipment manufacturers. One such scheme for adjusting this rate is Auto Rate Fallback (ARF). ARF uses relies on Adaptive Repeat Request (ARQ), mechanism employed by nodes to achieve reliable data transmission using acknowledgments and timeouts, to determine when to reduce the data. For example. as a STA moves away from an AP it begins to experience increased bit error rates. This will cause the ARQ mechanism to attempt to retransmit errored frames. If the ARF mechanism detects a number of consecutive ARQ retransmission attempts it will reduce the data rate to oedeer the increase the probability of a successful transmission attempt.

Although the provision of the multirate MAC was designed to increase performance by reducing the number of dropped frames, it actually reduces overall system performance. This is because the IEEE 802.11 standard does not consider the fact that transmission at 1 Mbps takes 11 times longer than an equal packet size transmission at 11 Mbps! The standard still guarantees all STAs the same long-term medium access probability. A comprehensive analysis of this anomoly can be found in [23].

2.4.4 IEEE 802.11e: MAC Enhancements for QoS

The original IEEE 802.11 standard was developed primarily for best effort data services. However, recent years have seen a dramatic increase in the amount of real-time traffic (i.e. streaming media, network games, VoIP, etc.) carried on these networks. This type of traffic imposes strict network related performance requirements in order to provide a certain level of QoS to end users. As a result, the IEEE proposed and ratified the 802.11e [24] supplement to the IEEE 802.11 standard that allows service differentiation of various traffic flows within a WLAN. Service differentiation is introduced by extending the standard 802.11 CSMA/CA contention mechanism to allow adjustment of MAC parameters that were previously fixed.

IEEE 802.11e specifies a new MAC Layer function called the HHybrid Coordination Function (HCF). The HCF provides both contention based and pooling-based channel access using Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA) respectively. APs and STAs that implement the QoS facilities are called QoS - Enhanced Access Point (QAP) and QoS - Enhanced Station (QSTA) respectively. In addition to these new coordination functions, the HCF also introduces the concept of Transmission Opportunity (TxOP), which refers to a time duration during which a QSTA is allowed to transmit a burst of data frames.

IEEE 802.11e also specifics other optional mechanisms. Block Acknowledgments (BAs) can be used to reduce overhead associated with the transmission of multiple frames within a single TxOP. NoAck allows QSTAs to specify whether a frame is to be acknowledged or not. This avoids retransmission of highly time-critical data. While, Direct Link Setup (DLS) allows direct QSTA-to-QSTA frame transfer within a QoS - Enhanced Basic Service Set (QBSS) where previously frames had to be transmitted via the AP. Again this mechanism is designed to reduce overhead and increase efficiency of QSTAs.

Enhanced Distributed Channel Access (EDCA)

Enhanced Distributed Channel Access (EDCA) is designed to provide prioritized QoS by enhancing the contention based DCF mechanism outlined above. This prioritization is achieved by associating a priority level with every packet entering the IEEE 802.11e MAC. These user level priorities are known as Traffic Categories (TC). EDCA also introduces four First-in, First-out (FIFO) queues at the MAC layer called Access Category (AC). Packets arriving at the MAC layer are filtered into their corresponding ACs (see Figure 2.17) in accordance with the IEEE 802.1D bridging protocol.

Each AC behaves as a single DCF contending entity with its own contention parameters (see Table 2.2), which are announced periodically by the QAP. Each AC is tuned to cater for a specific type of traffic; Background (BG), Best Effort (BE), Video (VI) and Voice (VO). Basically, an AC uses AIFS[AC], CWmin[AC] and CWmax[AC] instead of the DCF parameters DIFS, CWmin, and CWmax for the contention process to transmit a frame. These parameters are chosen to allow higher priority traffic gain access to the medium quicker than lower priority traffic. The smaller the values of

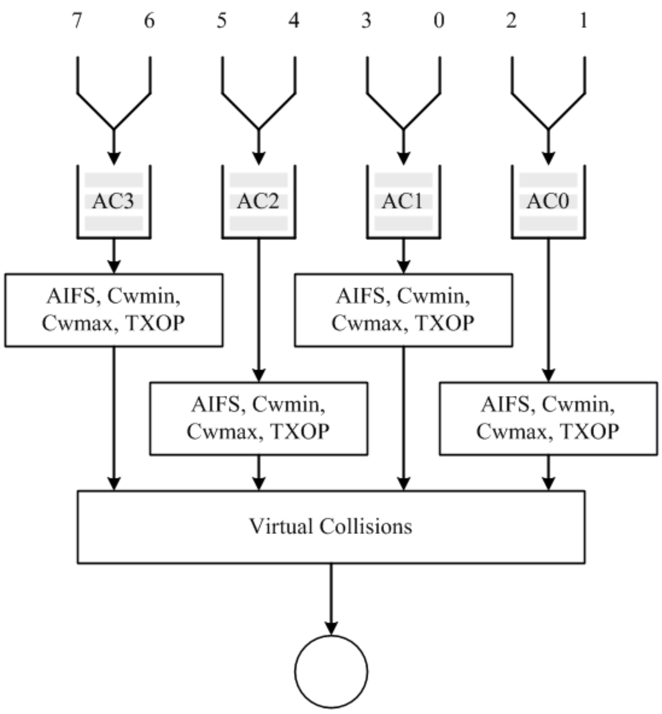


Figure 2.17: High level EDCA structure

CWmin[AC], CWmax[AC], and AIFS[AC], the shorter the channel access delays, and consequently the higher capacity share for a given traffic condition.

AC	Acronym	CWmin	CWmax	AIFSN
0	BG	aCWmin	aCWmax	7
1	BE	aCWmin	aCWmax	3
2	VI	(aCWmin + 1)/2-1 1	aCWmin	2
3	VO	(aCWmin + 1)/4-1	(aCWmin + 1)/2-1	2

Table 2.2: Default EDCA parameter set. aCWmax and aCWmin values are specified by the PHY parameters

The two key parameters that control how and when the various ACs gain access to the medium, are the CW and Arbitrary Inter-frame Spacing (AIFS). The AIFS determines the amount of time a AC should wait after a transmission has ended before attempting to transmit or backoff. The CW controls the length of the backoff delay that is introduced for each AC so as to avoid collisions. AIFS[AC] is calculated using Equation 2.4, where Arbitrary Inter-frame Spacing Number (AIFSN) is part of the EDCA parameter set for a given AC while the CW backoff delay for a given AC is calculated using 2.5. The CW range increases exponentially after each failed transmission attempt and reset after each successful transmission. The sutucture of these timing parameters is illustrated in Figure 2.18.

$$AIFS[AC] = (slotTime * AIFSN[AC]) + sifs \tag{2.4}$$

$$CW[AC] = random[1, CWmin + 1] * slotTime \tag{2.5}$$

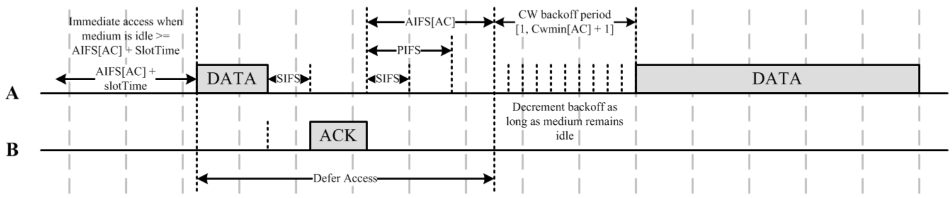


Figure 2.18: IEEE 802.11e prioritization mechanism

IEEE 802.11e EDCA also defines TxOP as the interval of time when a particular QSTA has the right to initiate transmissions. The TxOP interval for each AC is also announced by the QAP. During an EDCA TxOP, a QSTA is allowed to transmit multiple MAC Payload Data Units (MPDUs) from the same AC with a SIFS time gap between an ACK and the subsequent frame transmission. Figure 2.19 illustrates this mechanism. TxOP increase system throughput without degrading other system performance measures as long as as TxOP limit is not abused.

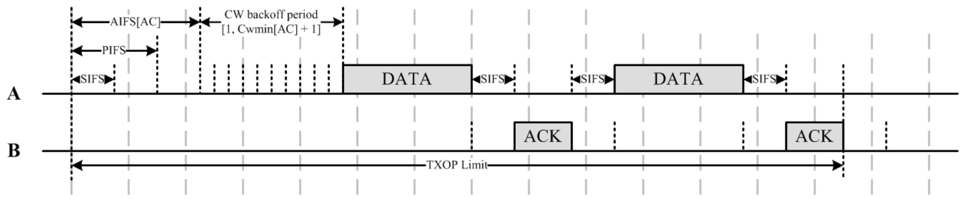


Figure 2.19: TXOP

HCF Controlled Channel Access (HCCA)

IEEE 802.11e HCCA operates in a similar manner to IEEE 802.11's PCF. However, with HCCA there is no division between CFPs and CPs. When in HCCA mode the medium is controlled by a QoS aware Hybrid Coordinator (HC). This functionality is the responsibility of the QAP when in infrastructure mode, and a QSTA can be nominated as the as the HC in ad-hoc mode. The HC has higher priority access to the medium than other stations in the QBSS which allows it to initiate a Controlled Access Phases (CAPs). A CAP is a timing period where a HC can initiate a downlink frame transfer with a QSTA or poll a QSTA for pending frames. Transfer of data or control frames is initiated after a Priority Inter-frame Spacing (PIFS) which allows the HC gain the priority access to the medium. For more information of the operation of the HCCA mechanism see [25].

2.4.5 Other IEEE 802.11 Admentments

Apart from the amendments discussed above, the IEEE 802.11 standard consists of a number of amendments and porposed admendments. Each of these admendments brings either imporvemnets or new features to the specification. The most significant ammendment is, IEEE 802.11n, which is currently in the draft stages of ratification. This amendment introduces support for higher data rates of at least 100 Mbps data throughput. Unlike previous amendments to the IEEE 802.11 standard, IEEE 802.11n aims to achieve this goal by using both physical and MAC layer enhancements. Several new

MAC features have been proposed to improve throughput efficiency. A detailed discussion of these improvements can be found in [26].

Other proposed and ratified ammendments include 802.11i for enhanced security, 802.11k for radio resource measurement enhancements and 802.11s for mesh networking.

Chapter 3

Literature Review

Wireless multimedia streaming has been an extremely active area of research over the past number years. Different research proposals aim at improving quality of perception, reducing network load, increasing utilisation etc. This research has proposed solutions at all layers of the OSI conceptual model. In general, this research has taken two approaches to QoS provisioning for streaming services. These approaches can be categorised as end-to-end or network centric. This chapter investigates the approaches taken by both of these categories. Section 3.1 presents the various network layer QoS metrics and what affect they have on quality of application layer multimedia. Section 3.2 outlines the various end-to-end based approaches. Section 3.3 discusses the proposals in the network-centric area. Finally, Section 3.4 discusses a number of objective and subjective methods for evaluating video quality.

3.1 Multimedia Characteristics

The effect of various network layer characteristics have on application level performance is a key part in the design of a suitable congestion control algorithm for multimedia application. Congestion control aims at controlling the network traffic so as to avoid a collapse due to congestion. These solutions try to optimise throughput, delay and jitter for multimedia traffic. These metrics can be categorised as Quality of Service (QoS). Quality of Experience (QoE) is another term that has recently been adopted by the ITU that represents the overall result of the QoS. It measures the acceptability of a service provided by an application from the point of view of the end user. For multimedia applications this often takes for form objective and subjective testing of content. QoE can be viewed as an extension of QoS providing a higher layer of abstraction that is closer to the user.

3.1.1 Throughput

Throughout is the metric that measures the amount of raw data that is transfered between two nodes on a network. Throughput is an important factor in providing certain levels of QoE for multimedia application. It can be generally assumed that the higher the bandwidht and consequently the throuput achieved by a multimedia application the better the QoE experienced by the end user. Multimedia applications have varying throughput requirements depending on the type of multimedia content they are carrying. Multimedia traffic requires certain bandwidth guarantees to be met in order to maintain acceptable levels of QoE. However, networks do not have any default mechanism to reserve bandwidth to meet such a requirement. Multimedia traffic is also susceptible to large throughput fluctuations, which can also impact QoE by causing delays etc.

3.1.2 Loss

Loss in IP-based networks can be broadly categorized as either congestive or transmission losses. Congestive loss occurs due when the combined data rate exceeds the available capacity on a given link. This causes the buffers of routers servicing that link to overflow resulting in dropped packets. Transmission losses occur due interference on the physical medium. In wired networks, congestive loss dominates, whereas in wireless networks transmission is more significant.

Loss is a serious issue for multimedia transmissions in WLANs. Lost packets can have potentially disastrous effects on QoE. Although video applications will work with loss, user QoE will be affected. To avoid this, the packet loss ratio must be maintained below a certain threshold to achieve acceptable QoE. However, loss can be counteracted with various error control techniques.

3.1.3 Delay / Jitter

There are a number of sources of delay in IP-based networks: serialization, queueing and propagation. Serialization delay occurs at the data link layer where frames are broken down into byte sequences which are then transmitted over the physical medium. Queueing delay arises due to the statistical multiplexing employed by nodes taking advantage off the bursty nature of most networked applications. Packets arriving simultaneously at a router destined for a common outbound link will experience transient congestion resulting in a delay while packets are multiplexed into the outbound queue. While propagation delay refers to the latency of the signal traversing the physical medium. Jitter is another type of delay experienced in IP-based networks. This refer to the variation in delay experienced by consecutive packets.

Non real-time delivery of multimedia content is not subject to the same strict delay constraints as real-time delivery of multimedia applications (i.e. video conferencing). The interactive nature of real-time applications requires bounded end-to-end delay. That is, every video packet must arrive at the destination in time to be decoded and displayed before the event horizon is reached. Non real-time multimedia applications can implement large buffers to negate the effect of this network delay. Bounded delay is only an issue during the startup phase of the multimedia stream. Large delays during this period can create lengthy channel hopping delays which cause annoyance to the user. Jitter is a problem for real-time multimedia applications as it causes loss and affects QoE but is less of a problem for non-real-time multimedia application where large receiver buffers can be implemented.

3.2 End-to-End Centric Approach

The end-to-end approach to multimedia streaming is based on the seminal work proposed in [27] on the end-to-end design principle which proposes that the network should be dumb and with all intelligence at the end points. It stipulates that the network should only be capable of providing the minimal service set required for transporting a packet from source to destination. All intelligence should be placed on the end points. An illustration of the endd-to-end approach is shown in Figure 3.1.

Figure 3.1: Illustration of End to End approach

3.2.1 Multimedia over TCP and UDP

Currently TCP and UDP are the predominant protocols for transporting data over IP based networks. They both follow an end-to-end approach to service provisioning. As a result they are a logical choice for streaming media. However, neither TCP nor UDP, are adequate for video applications. UDPs service model does not provide enough support, while TCPs provides too much. Unlike conventional applications, streaming media requires continuous bandwidth availability and limits end-to-end delay and jitter levels.

TCP is generally unsuitable for multimedia streaming mainly due to its fluctuating throughput. It was primarily designed for providing end-to-end reliability and fast congestion avoidance. This end-to-end reliability is achieved by acknowledging received packets and retransmitting unacknowledged (lost) packets. These reliability features also make it unacceptable for streaming media as they can cause unacceptable pauses in playback while a streaming application waiting for lost packets to be retransmitted. However, these problems could be overcome using can be counteracted by using large receiver side buffering [28]. Although this buffering smoothes the video playback, it creates unacceptable startup delays. Also, the majority of the wireless devices where multimedia streaming solutions are deployed, are small and mobile devices, where resources are limited making large buffering impractical. It should also be noted that the congestion control mechanism that is responsible for these dramatic fluctuations is also responsible for providing scalability and preventing congestion collapse [18] which made TCP such an overwhelming success.

Although UDP is not the ideal solution (in terms of reliability) for streaming media, it provides an adequate base on which to build extra functionality that makes it media friendly (for more detail see Section 2.3.2). These solutions employ partial or no reliability mechanisms eliminating the unacceptable end-to-end delay caused by retransmission. They also implement rate control mechanisms that provide smoother throughput variations in the short term then TCPs AIMD sawtooth while maintaining TCP friendliness.

3.2.2 TCP-Friendly Rate Control Protocols

Rate Adaptation Protocol (RAP) [28] is an end-to-end rate-based congestion control mechanism. It was one of the earlier attempts at rate-based congestion control. RAP is a sender based rate control mechanism. The sender transmits data packets to the receiver, who acknowledges each received data packet with a feedback packet. The sender uses this feedback mechanism to detect loss and estimate Round Trip Time (RTT). RAP considers losses to be congestion signals, and uses timeouts, and gaps in the sequence space to detect loss. The sender then adjusts the sending rate using an AIMD algorithm based on the receivers feedback. The sending rate is changed no more than once per RTT, otherwise the algorithm may become unresponsive.

Research has shown that the RAP algorithm does not compete fairly with TCP in many cases [29]. This issue can be rectified with the introduction of Random Early Drop (RED) [30] queueing routers in the core network. However, this adds to the cost and complexity of the implementation.

Streaming Media Congestion Control (SMCC) [31] protocol is another end-to-end-centric approach to rate / congestion control. SMCC is a receiver driven protocol that estimates the bottleneck bandwidth share of a connection using algorithms similar to those introduced in TCP Westwood [32]. Unlike RAP, SMCC does not send acknowledgements to the sender for each received packet. Instead, it sends Negative Acknowledgement (NACK) to the sender when the receiver identifies a lost packet. These NACKs are used to inform the sender that a loss has occurred, requesting a retransmission depending on whether the packet can be delivered before the event horizon has been reached. NACKs also carries the receivers current Bandwidth Share Estimate (BSE) allowing the sender to adjust its sending rate accordingly. The sender mimics TCPs congestion avoidance phase by increasing its sending rate by one packet per RTT until a NACK message is received, upon which the sending rate is set to the BSE. After this readjustment in the sending rate, the server resumes a linear sending rate increase of one packet per RTT. SMCC behaves well in random loss environments, since the bandwidth estimate is robust to sporadic packet loss.

TCP Friendly Rate Control (TFRC) [33] is an approach that trades off the benifits between UDP and TCP like approaches. TFRC is an equation-based congestion control algorithm explicitly designed for best-effort unicast multimedia traffic. It is designed to be reasonably fair when sharing bandwidth with TCP flows. TFRC also has a much lower throughput variation over time compared with TCP making it an ideal for streaming media. TFRC is not a full transport layer protocol, rather a congestion control mechanism that can be used by an existing transport protocol, such as UDP. It determines the sending rate (see Equation 3.1) as a function of the RTT, loss event rate (p) and packet size (s). These parameters are calculated on the receiver side of the connection where they are periodically sent back in the form of feedback to the sender. This dependency on the receiver makes TFRC a receiver driven protocol.

$$X = \frac{s}{RTT\sqrt{\frac{2bp}{3}} + 12 \times RTT \times p\sqrt{\frac{3bp}{8}}\,(1 + 32p^2)} \tag{3.1}$$

The key assumption behind TFRC is that any lost packet is caused by network congestion. However, this assumption does not hold true for wireless networks. Packet loss is caused by both congestion and the physical channel. In wired networks the vast majority of losses can be attributed to congestion due to the reliability of the wired medium. However, in wireless networks the losses due to the physical medium dominate. TFRC and other congestion control mechanisms that were primarily designed for wired networks have no way of distinguishing between congestion and propagation losses. This inability to distinguish between the two types of losses can impact severely on the performance of congestion control mechanisms, and TFRC in .

TFRC Wireless [34] was proposed to improve the efficiency of TFRC in the presence of wireless errors. As outlined above TFRC rate equation estimates available bandwidth based on packets size, loss event rate and RTT. However, the loss event rate calculation includes both congestion-based and random losses due to the wireless medium, as TFRC employs no mechanism for distinguishing between them. The lack of a loss discrimination mechanism causes TFRC avoid congestion incorrectly in the presence of wireless errors. TFRC Wireless employs an Loss Discrimination Algorithm (LDA) that allows it to recognise random wireless losses that are not caused by congestion and discount them in the calculation of the loss event rate. It detects these losses by measuring changes in RTT. Relatively higher RTTs measurements are an indication of congestion. As a result there is a higher probability of packets lost during this period being caused by congestion and not the wireless medium.

Another approach for achieving more efficient rate control while streaming media over wireless links is using a *TFRC-aware Snoop* module, similar to the mechanism proposed in [35]. This module sits on the AP between the LAN and WLAN. It performs local retransmissions when it detects TFRC packets that have been corrupted due to wireless channel errors. By effectively hiding wireless channel errors from the end-hosts, the TFRC based streaming application does not unnecessarily have to decrease its sending rate in these situations. The main advantage of this approach is its simplicity, and robustness to unpredictable wireless channel conditions. However it requires significant modifications to the network infrastructure as it require extra functionality to be implement on each AP. Explicit Loss Notification (ELN) and ECN [36] can also be used for detecting errors while streaming over wireless channels. However this solution also suffers the same disadvantages as a Snoop based approach.

TCP Friendly Rate Control with Compensation (TFRCC) proposed in [37] aims to provide better support for QoS requirements of multimedia applications without violating network fairness constraints. TFRCC is built upon TFRC, in that TFRCC calculates the TCP-friendly sending rate using the same mechanism outlined by TFRC. However, if the calculated sending rate is found to violate the QoS constraints imposed by the media application, TFRCC will temporarily adjust the sending rate to support the urgent QoS requirements of the application. This action will result in short-term

TCP-unfriendlieness characteristics. To correct this deviation from the TCP-friendly value, a rate compensation algorithm is proposed to maintain good long term TCP friendliness.

Although this proposal enables better support for the QoS constraints of the application it only considers the QoS constraints on a per stream basis. It still maintains long term TCP-friendliness which has the biggest effect on the long term end user perceived quality.

Video Transport Protocol (VTP) [38] is a rate control mechanism specifically designed for real-time streaming in wireless networks. It employs two techniques: Achieved Rate (AR) - measures the data rate which has successfully been received (throughput) and Loss Discrimination Algorithm (LDA) - distinguishes between congestion and wireless losses. VTP rate control is based on the analysis of TCP instantaneous sending rate. Similar to TCP, VTP linearly probes the available bandwidth until congestion is detected. However, unlike TCP, VTP does not perform multiplicative decrease. Instead, it reduces the sending rate to the AR. In this way VTP avoids the drastic rate reductions in sending rate which impact severely on video quality. VTP maintains the same average throughput as a similarly configured TCP stream without its characteristic fluctuations. VTP achieves this by reducing its sending rate by a smaller amount while maintaining this reduced rate for a longer period of time,. This is illustrated in Figure 3.2 where VTP ($A1$) achieves the same long term average throughput as TCP ($A2$) in the situation wher both schemes are trying to avoid congestion.

MULTFRC proposed in [39], is another mechanism that could potentially be used for streaming media over wireless network. It is based on work originally carried to investigate use of multiple concurrent TCP connections for streaming media [40]. These mechanisms open multiple connections in order to acquire more bandwidth from the transmission resource. More connections results in more competition with other flows. Since fairness between TCP-friendly applications is based on their individual transport layer con-

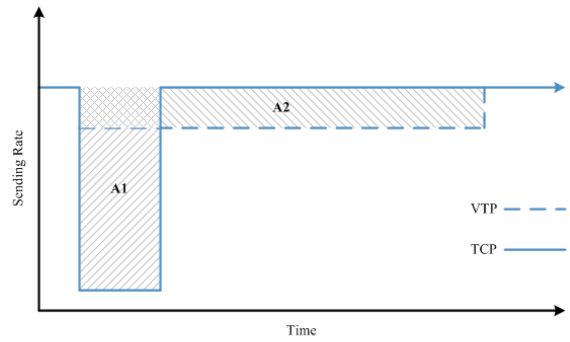


Figure 3.2: Comparison of the instantaneous sending rate of TCP and VTP

nections rather then their combined view from the application layer, using more connections than another application can result in individual applications acquiring higher throughput. It requires no modification to the existing network infrastructure. The drawback to this approach is that their is no limit to the number of connections that could be opened. This approach also requires a more complex scheduling algorithm to ensure the timely delivery of relevant data chunks. It would also require some sort of discovery mechanism and a utility function to discover devices and map device characteristics into relevant number of streams.

3.2.3 Summary

This section has presented the end-to-end approaches to providing a certain level of QoS for multimedia applications in both wired and wireless networks. These proposed solutions have a fundamental shortcoming: they focus on optimising the consumption of network resources and omit the perceived quality of the media streams. None of the above mechanism take account of the media applications requirements. Therefore it is desirable to devise a congestion control mechanism that is both TCP and media friendly.

3.3 Network Centric Approach to QoS Provisioning

The network centric approach to QoS provisioning builds intelligence into the network as opposed to the end-to-end approach which states that all intelligence should reside on the end points. An illustration of this approach is shown in Figure 3.3

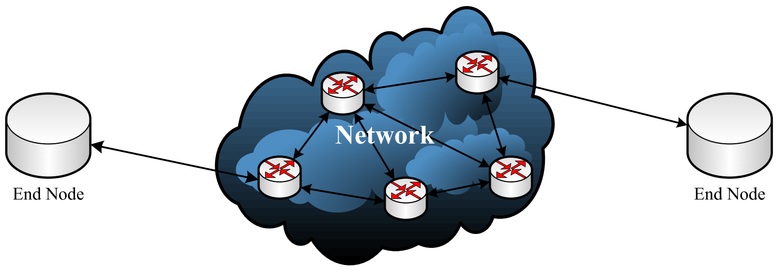


Figure 3.3: Illustration of network centric approach

3.3.1 Network Service Models

The *Integrated Services (IntServ)* [41] model was one of the first major architectures proposed by the IETF that specified the elements required to achieve certain QoS guarantees over interconnected heterogeneous networks. QoS was provided on a per flow basis using a signalling protocol such a Resource Reservation Protocol (RSVP) [42]. This approach adopted by this system was based on the principle where router's must reserve resources in order to provide the required QoS for certain traffic flows. Each router is required to state information for each flow in the network. As a result each router in the network required major modification in order to support the service.

The IETF later proposed the *Differentiated Services (DiffServ)* [15] model with the goal of overcoming the complexity and scalability issues inherent in the IntServ model. DiffServ provides a simple scalable mechanism for managing, classifying and providing course grained QoS guarantees to flows in an IP-based network. DiffServ is offered by a set of router's forming an administrative domain. The domain administrator defines a set of service classes which correspond to certain forwarding rules [43]. It uses the *Type of Service* field in the IP packet header to mark specific packets for preferential treatment. Using the same analogy of the postal outlined in Section 2.3.1 for describing the operation of IP, DiffServs class based approach to packet delivery is similar to letters receiving express, overnight or two-day delivery. Although the choice of service classes is left up to each operator, the IETF has defined *expedited forwarding* [44] and *assured forwarding* [45] to provide service compatibility for packets forward between different administrative domains.

3.3.2 Wireless Scheduling

Scheduling transmission of packets in wireless networks is one of the key mechanisms for providing a higher level of QoS. Extensive research has focused on scheduling mechanisms in wired networks that share bandwidth fairly between clients. Most of these wired mechanisms, *Weighted Fair Queueing (WFQ), Start-Time Fair Queueing (STFQ)* and *Earliest-Due-Date First (EDD)* are not well suited to the WLAN because they do not consider the characteristics of the wireless channel.

In wireless networks the original scheduling mechanisms employed by the IEEE 802.11 MAC provided fair scheduling for best effort traffic. However, it made no provision for multimedia content. More recently, priority based scheduling was introduced by the IEEE 802.11e via Enhanced Distributed Coordinator Function (EDCF). This enabled certain QoS guarantees for multimedia applications. Scheduling in wireless networks is particularly challenging due to the limited bandwidth, time-varying and location-dependent signal quality. The problem is further complicated by the limited capacity and distributed nature of WLANs. An overview of scheduling mechanisms for multimedia transmission is presented in [46].

Distributed Weighted Fair Queue (DWFQ) [47] adjusts the contention window size based on the difference between the actual and expected throughput. The bandwidth received by the flow is proportional to the queues weight. DWFQ uses the CW mechanism of the IEEE 802.11 MAC DCF to create this proportional bandwidth distribution. This is because the bandwidth received by a flow depends on its CW. The smaller the CW, the higher the achieved throughput. DWFQ enabled STAs to compete with each other with different CW. The authors propose two different algorithms using this strategy. In the first, if actual throughput is greater than the expected throughput, the CW will be decreased in order to increase the flow priority and vice-versa. In the second, the ratio of the estimated bandwidth requirement to the weight of the flow is calculated. The result is then compared with that of other flows and the CW is adjusted accordingly.

Persistent Factor DCF [48] does not use the binary exponential backoff technique used in the IEEE 802.11 standard. Instead a STA wishing to transmit a frame determines whether to attempt the transmission following an idle time of DIFS by the probability P. Each traffic class is assigned a persistent factor P. Higher priority classes are assigned a smaller value of P, while lower priority classes are assigned a larger value for P. In the backoff stage, a uniformly distributed random number r is assigned to every time slot. Each flow starts transmission only if the $r > P$ in the current time slot. The backoff interval is a geometrically distributed random variable with parameter P.

Vaidya et al. [49] propose an wireless scheduling technique called *Distributed Fair Scheduling (DFS)* based on the wired based Self-Clocked Fair Queueing (SCFQ) [50]. DFS introduce both prioritisation and fairness to the scheduling mechanism. DFS enabled STA performs a back-off for each packet it wants to transmit. This back-off interval is calculated as a function of packet size and weight of the STA. This weighing introduces prioritisation, as STAs with low weights will generate longer backoff intervals than those with high weights, thus getting lower priority. DFS achieves fairness by considering the packet size in the calculation of the back-off interval. This allows flows with smaller packets to be sent more often.

[51] proposed *Content Aware Adaptive Retry* scheduling mechanism to improve video transmission over WLAN. The proposed algorithm adapts the ARQ limit dynamically based on the type of content being carried. The mechanism adds functionality to the IEEE 802.11 MAC DCF that enables it to dynamically determines whether to send or discard a packet based on its re-transmission deadline which is assigned to each packet according to its temporal relationship and error propagation characteristics with respect to other video packets within the same Group of Pictures (GOP). It essentially tries really hard to re-transmit I-frames, moderately hard to re-transmit P-frames and not very hard to re-transmit B-frames in a GOP. Adapting the number of retries can reduce the impact of random backoff deference and co-channel interference that can cause late packets.

3.3.3 Admission Control

Admission control is a mechanism used in networks to manage QoS level. Admission control is trivial in networks where the transmission resource has a physical limit on the number of users it can support, such as a Time Division Multiple Access (TDMA) network where users are assigned a dedicated channel. However, this is not the case in networks where there is no physical limit on the number of users that can access the shared resource. A network employing an admission control algorithm will determine whether or not to admit a device wishing to join that network with its requested QoS requirement without violating the QoS requirements of existing users. This decision is made based on the requirements of the device being admitted, existing devices requirements and the network's current available resources. An admission control algorithm could be formulated as an optimisation problem. For example an admission control algorithm might try to maximise signal quality, revenue and transmission rate or minimise call dropping probability, delay and jitter, or maintain fair resource sharing.

Numerous admission control algorithms have been proposed for WLANs. A comprehensive survey of admission control schemes can be found in [52]. *Distributed Admission Control (DAC)* [53] was developed by the IEEE 802.11e Task Group to protect active quality of service streams such as voice and video streams. DAC uses beacons transmitted by the AP to announce the current transmission budget for each AC. This budget indicates the available transmission time per AC in the next beacon period in addition to what is being utilised. Each STA also calculates an internal transmission limit per AC for each beacon, based on the transmission count during the previous beacon period and the transmission budget announced from the AP. When the transmission budget for an AC is depleted, a new flow will not be able to obtain any transmission time, and existing flows will not be able to increase their transmission time either.

Virtual MAC and Virtual Source proposed in [54] are designed to allow a STA to passively observe the radio channel. This enables the STA to evaluate the channel, and estimate the achievable service qualities without actually loading the channel. The STA can then determine whether a new flow can be admitted. This mechanism operates in parallel with the real MAC. It handles virtual packets in same way the real MAC handles real packet. However, the virtual MAC does not actually transmit the packet on the physical medium. Instead it estimates the probability of collision if the virtual packet were to be actually sent. If a collision is detected, the Virtual MAC enters a backoff procedure. A Virtual Source algorithm is used in conjunction with the virtual MAC to estimate delay. The virtual source mimics a real application by generating virtual packets like a real application. These virtual packets then enter a virtual MAC where they are processed as outlined above. The main advantage of these virtual algorithms is that they do not use any bandwidth. However they do require significantly more processing power. Also, the main criteria for the admission decision are based on delay and collision estimations. They provide no estimation of achievable throughput.

[55] proposes a admission control scheme to control the packet dropping rate for video (MPEG4 and H.263) conference services over the uplink wireless systems. When a new call arrives, the required bandwidth of all existing users and the new call at the highest quality is determined. The new call is admitted only if the total required bandwidth is less than the total available bandwidth. If this test fails the algorithm lowers the quality of the new call (thus reducing its required bit rate) and again tries to gain access. If the required bandwidth is still higher than the available bandwidth the algorithm attempts to reduce the quality of existing calls in a sequential manner, in order to try to accommodate the new call. If all calls are reduced to their lowest quality level and the required bandwidth is still greater then the available bandwidth, then the new call is blocked. The algorithm also continiously checks the loss rate of existing connections to ensure an adequate service level is maintained. Should the loss rate exceed a threshold the algorithm will instruct clients to reduce quality in order to free up resources.

3.3.4 Summary

This section presented various mechanism for providing certain level of QoS guarantees using the network infrastructure. The majority of these mechanism contradict the end-to-end principle. However, some of these mechanisms, such as admission control, are a necessity for achieving high levels of QoE.

3.4 Video Quality Performance Metrics

Video quality performance metrics techniques quantify the quality of video sequences. These metrics can be categorised as either objective or subjective. Since 1997 the Video Quality Experts group (VQEG) has assessed various objective techniques for video quality assessment proposed by various research groups. The eventual goal of this analysis was to propose a quality metric for standardisation by the ITU. The performance of the proposed models was found to be statistically equivalent [56]. However it was also found that models were also statistically equivalent to that of Peak Signal to Noise Ratio (PSNR) (used reference objective model for these tests). As a result the VQEG did not propose one or more models for inclusion in ITU Recommendations on objective picture quality measurement.

The ITU has recently adopted the term QoE to represent both objective and subjective assessment of video images []. QoE describes users subjective perception of a system, application, event, or service relative to expectations. QoE is often used interchangeably with QoS, but they are two very different terms as they asses quality at different layers of the OSI stack. QoS focuses on transport layer performance metrics, while QoE focuses on application layer performance metrics. QoS parameters are objective, and although sometimes are difficult to measure, are quantifiable. QoE metrics are mostly subjective (although objective techniques exist, they try to mimic subjective assessment techniques), and often difficult to measure. QoE depends on many factors that are often outside the service providers control, i.e. end user context. The rest of this section will discuss the various assessment techniques for QoE.

3.4.1 Objective Video Quality Assessment

Objective metrics determine video quality without the need for human analysis. They compare the difference between the erroneous video and the original video using mathematical metrics based on formulae derived from psycho-visual experiments or model metrics based on the human visual system. Many objective techniques exist, the most common of which is PSNR. These metrics can be classified based on the approach they take in determining the quality of the video being processed (detailed of this categorisation can be found int [57]).

Peak Signal to Noise Ratio (PSNR)

PSNR is a mathematical metric that measures the maximum possible power of a signal with the power of the noise signal. For video sequences, PSNR is computed using on the luminance component in the YUV colour space. Each video frame contains $i \times jpixels$ each representing an $8 - bit(0 - 255)$ monochrome colour. PSNR is usually expressed in dB. Typical PSNR values vary in the range $20dB - 40dB$. PSNR is defined in Equation 3.3 using Mean Square Error (MSE) defined in 3.2.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \| \| I\left(i,j\right) - K\left(i,j\right) \| \|^2 \tag{3.2}$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{3.3}$$

PSNR can be used to determine the difference between two images. However, it can not tell how that difference will impact on human perception. For example, a small difference that reoccurs over the entire frame will produce the same MSE as a large difference that occurs in one particular area. However, one of these differences will be more noticeable than the other. This is because PSNR is derived directly from its engineering counterpart, it does not consider the characteristics that influence the human visual system [58].

Motion Picture Quality Metric (MPQM)

MPQM [58] is designed to mimic a basic vision model. It incorporates to key human perception phenomenon. The first accounts for the fact that a signal is detected by the eye only if its contrast is greater that some threshold. While the second phenomenon is related to the human vision response to the combination of several signals. It does this by decomposing the reference and the erroneous sequences into multiple perceptual channels. A channel-based distortion measure is then computed, accounting for contrast sensitivity and masking. Finally, the data is pooled over all the channels to compute the quality rating which is then scaled from 1 to 5 (from bad to excellent). MPQM does not take into consideration the chrominance, which led in the creation of Color MPQM.

Structural Similarity Index (SSIM)

SSIM [59] is another approach for video quality assessment. This metric measures structural distortion instead of error to evaluate video quality. This is based on the idea that the Human Visual System (HVS) is tuned for extracting structural information from the viewing field. Thus, a measurement on structural distortion should give a better correlation to the subjective impression. SSIM is a full reference metric that uses Equation 3.4 to determine video quality. In this equation $x, y, \sigma_x, \sigma_y, \sigma_{xy}$ are the estimates of the mean of x, mean of y, the variance of x, the variance of y and the covariance of x and y. C_1 and C_2 are constants. The result of this equation is between $-1.0 - +1.0$, where $+1.0$ represents no distortion.

$$SSIM = \frac{(2xy + C_1)\,(2\sigma_{xy} + C_2)}{(x^2 + y^2 + C_1)\left(\sigma_x^2 + \sigma_y^2 + C^2\right)} \tag{3.4}$$

Digital Video Quality (DVQ)

DVQ [60] is another full reference metric that incorporates many aspects of the HVS. Simplicity is one of the main goals of DVQ, since it would ideally like to used for real-time computation. DVQ comprises of several processing stages. The metrics algorithm first process the test reference and test clips by performing various sampling, cropping, and colour transformations that serve to restrict processing to a region of interest and to express the sequences in a perceptual colour space. Local contrast is then obtained using a blocking and DCT where local contrast is the ratio of DCT amplitude to DC amplitude for the corresponding block. Next, temporal filtering is used to determine the temporal contrast sensitivity function. The result of this filtering is then converted to just-noticeable differences by dividing each DCT coefficient by its respective visual threshold. This implements the spatial part of the contrast sensitivity function. The two sequences are now subtracted to produce a difference which is then subjected to a contrast masking operation. Finally the masked differences may be pooled in various ways to illustrate the perceptual error over various dimensions and the pooled error may be converted to visual quality.

Video Quality Metric (VQM)

VQM [61] [62] is a full reference video quality metric that uses feature extraction and analysis to calculate perceived video quality. VQM measures the perceptual effects of video impairment including, blurring, jerky motion, noise, block distortion and colour distortion. VQM has high correlation with subjective video quality assessment. It has a similar implementation to DVQ with difference in the conversion of local contrast to just noticeable differences and weighted pooling of mean and maximum distortion.

3.4.2 Subjective Video Quality Assessment

Subjective testing has existed for many years. It involves the evaluation of video quality by allowing a sample of participants to rate a particular video sequence. The main testing methodologies are presented in [63] and [64]. Subjective testing is generally regarded as the best method of evaluation video quality, provided a large enough group of test subjects are used to evaluate a particular sample. Some of the main testing methodologies are presented next.

Absolute Category Rating (ACR)

ACR is a category judgement method where the test sequences are presented one at a time and are rated independently on a category scale. Subjects are asked to rate the quality of the video using Table 3.1. The time pattern for the stimulus presentation and voting phase is can be illustrated in Figure 3.4, where A_e, B_e and C_e sequences A, B and C under test.

Grading Value	Estimated Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 3.1: ITU 5-point quality scale

Degraded Category Rating (DCR)

DCR is another subjective method for testing video quality. In DCR test sequences are presented in pairs. The first stimulus presented in each pair is always the source reference X_r without any impairments, which is followed by the same sequence with impairments X_e caused by the test conditions.

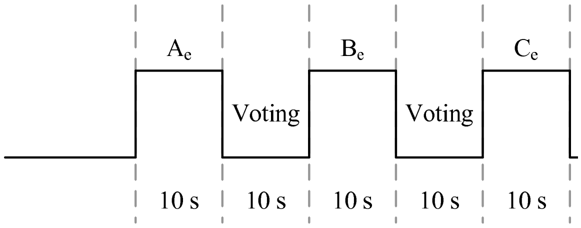


Figure 3.4: Stimulus presentation timing in ACR

The time pattern for the stimulus is illustrated in Figure 3.5. Voting time for each pair should be $< 10s$. Subjects are required to rate the impairment of the second stimulus in relation to the reference using Table 3.4.2.

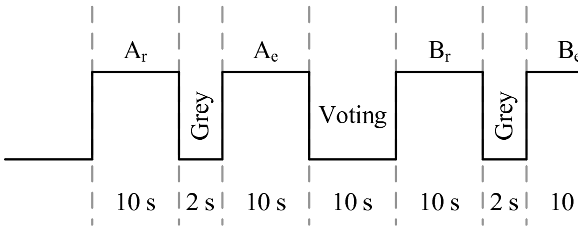


Figure 3.5: Stimulus presentation timing in DCR

Grading Value	Estimated Impairment
5	Imperceptible
4	Perceptible, but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

Table 3.2: ITU 5-point impairment scale

3.4.3 Summary

This section outlined various techniques for evaluating the quality of video using objective and subjective assessment. It is widely accepted that subjective assessment is a better method for determining, not just the quality of video, but also the QoE experienced by the user. Objective assessment is only capable of assessing a video in terms of the reference video and not the environment in which it is played. However, in most cases it is adequate to use objective assessment and most objective assessment techniques have good correlation with the subjective results.

Chapter 4

Greediness Control Algorithm

This chapter describes in detail the proposed Greediness Control Algorithm (GCA) for the problem where by bandwidth that is distributed fairly between competing video streams at the transport layer results in unfair application layer video quality distribution. The chapter begins with an introduction and overview of the architecture of GCA. This is followed by detailed discussion about the operation of the sender and receiver which includes details of the formulae used for estimating the available throughput.

4.1 Introduction

GCA is an equation based congestion control mechanism for streaming multimedia content in wireless networks. It enhances the IETF standardised TFRC congestion control mechanism, providing TCP compatible fairness. TCP compatibility implies that the protocol reacts to congestion in the same manner as TCP but does not necessarily compete fairly with other TCP flows. This allows GCA to be tailored specifically to the characteristics of the multimedia stream being carried. GCA inherits TFRC's smooth response to congestion and low throughput variation, making it suitable for multimedia streaming applications, while including new mechanisms that enhance the operation and efficiency of streaming applications in wireless networks. These mechanisms allow prioritization of multimedia flows and add robustness in the presence of random wireless loss. GCA depends on underlying protocols to provide the transport, real-time, multiplexing, length or checksum services. As a result GCA must be considered as a congestion control mechanism that could be used in a transport protocol such as DCCP, or in an application requiring end to end congestion control. GCA requires an application level entity to discover the characteristics of the multimedia enabled devices in order to adapt their video streams correctly.

4.2 Architecture Overview

GCAs end-to-end approach to rate / congestion control is accomplished using a client-server architecture. GCA sits between the application and transport layers (see Figure 4.1) in the simplified TCP/IP reference model where it has the ability to interpret the network conditions and control the video coding rate. It employs an equation based mechanism that estimates available bandwidth in a TCP compatible manner. It estimates the sending rate using a modified and simplified version of the TCP Reno throughput equation. the sending rate is calculated as a function of Round Trip Time (RTT), Loss Event Rate (p), packet size (s) and two novel greediness control parameters, α and β.

The high level view of the system is illustrated in Figure 4.2. Responsibility for various tasks is shared between the client and the server. The server is responsible for calculating and sending packets at a rate that suits current network conditions, while the client receives and calculates various parameters that assist the server's rate calculation. In this way the client aides the server by measuring the loss event rate of the received data stream and sends these measurements to the server using a feedback mechanism. The sender uses this feedback mechanism to measure the RTT using a combination of

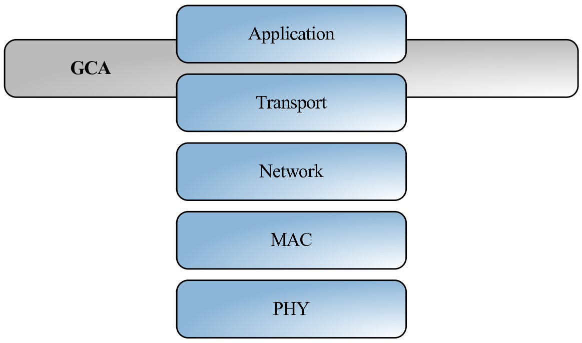


Figure 4.1: Where GCA fits into the TCP/IP reference model

the echoed timestamp (time last data packet was sent), delayed time (time delay between reception of last data packet and transmission of feedback) and reception time of feedback. Using the feedback information, loss event rate and RTT, the sender is able to estimate the available bandwidth using GCAs newly proposed throughput equation. The sender can now adjust its transmit rate to match the estimated rate.

Detailed descriptions of the operations of the sender and receiver mechanisms are presented next.

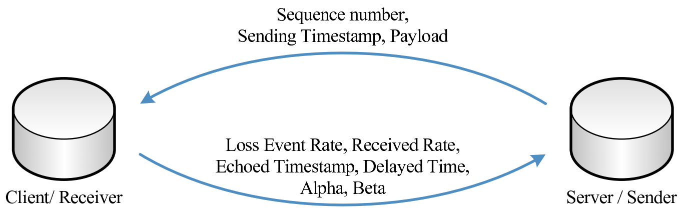


Figure 4.2: GCAs Architecture

4.3 GCA Sender

The sender's primary goal is to transmit data packets to the receiver in a TCP compatible manner. It does so by interpreting feedback packets that contain various receiver side parameters, such as loss event rate, echoed timestamps for RTT calculation and received rate. This information aides the sender in determining the available network bandwidth and thus an acceptable sending rate. The sender also sends inline downstream feedback to the receiver informing the receiver of RTT estimates that it has calculated.

4.3.1 High Level State Operation

The sender side of the GCA solution can be divided into three states: *slow start*, *congestion control* and *no feedback* as illustrated in Figure 4.3. Various events trigger changes of state based on the current state. The state machine is initialised with the transmission of the first data packet. The initialisation parameters are outlined in Table 4.1

After initialisation the sender enters the *slow start* state. In this state the sender doubles the sending rate up to the limit of double the received rate X_{recv} every RTT. This process continues until the state is exited when either a loss or no-feedback event has been encountered. In the *congestion control* state the sender maintains the sending rate in accordance with the result X of the throughput estimation Equation 4.7 and has an upper limit of the received rate estimated by the receiver. This state is triggered by the occurrence of a loss event. The *no-feedback* state is triggered by the expiration of the no-feedback timer. In this state the sending rate is halved and the no-feedback timer is reset.

As mentioned above changes of state are triggered by a number of events. These events, *loss*, *no-loss*, *no-feedback* and *session ended*, are caused by certain conditions being met. A *loss event* occurs when the sender receives a feedback packet reporting losses, while a *no-loss* event occurs when a feedback packet is received reporting zero losses. A *no-feedback* event occurs when the no-feedback timer expires as a result of no feedback being received at the sender for a certain period of time. A *session ended* event occurs when the session is terminated between the sender and receiver.

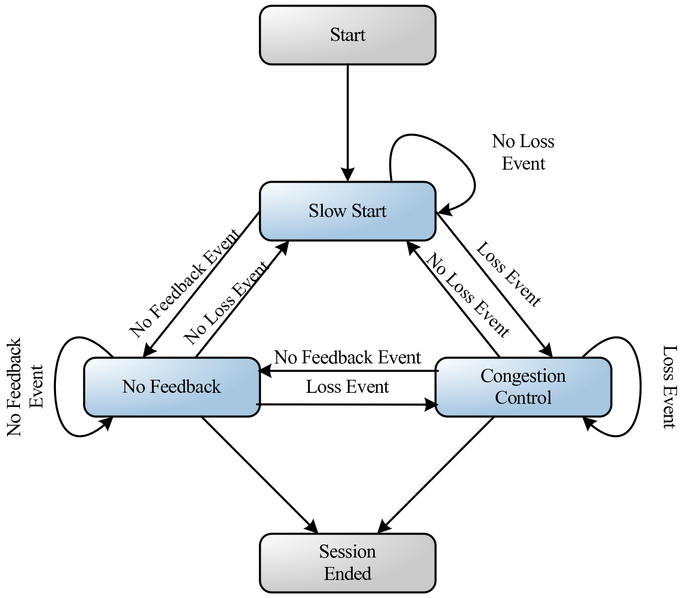


Figure 4.3: GCAs sender state machine

4.3.2 Low Level Operation

The sender is initialised with the values outlined in Table 4.1 and begins transmission of data packets at the rate of 1 packet per second. At this point the sender is in the slow start state. After a period of time the receiver will respond to the sender with a feedback packet. Feedback is transmitted at least once per RTT. Once this initial feedback packet is received the sender will perform the secondary initialisation. This secondary process initialises previously undefined RTT and RTO values using a combination of timestamp and sequence numbers contained within the data and feedback packets (see Figure 4.4). The sender can now estimate the R_{sample} using the *rxTime*, the time the feedback packet was received by the sender, *txTime*, transmission time of last received data packet at the receiver, *delayTime*, time at receiver between reception of last packet a transmission of feedback packets Equation 4.1. Subsequent RTT estimates are smoothed using an exponentially weighted moving average as shown in Equation 4.2, where $q = 0.9$. The weights determine the responsiveness of the transmission rate to changes in RTT.

Parameter	Value
X	1 pps
No Feedback Timer	2 s
RTT	undefined
RTO	undefined
Time Last Doubled	-1

Table 4.1: GCA Sender initialisation parameters

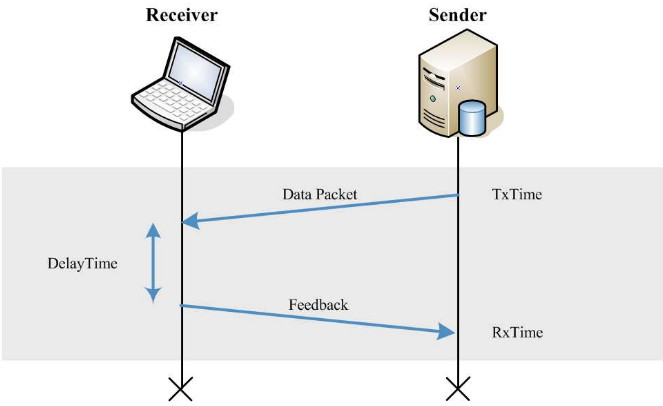


Figure 4.4: GCAs RTT calculation

$$RTT = (rxTime - txTime) - delayTime \quad (4.1)$$

$$RTT = (q * RTT_{prev}) + (1 - q) * ((rxTime - txTime) - delayTime) \quad (4.2)$$

$$RTO = 4 * RTT \quad (4.3)$$

Once the sender has determined the new value for the RTT it can easily calculate the RTO using Equation 4.3. Next the sending rate can be updated. The sender first gets the current loss event rate estimate p from the feedback packet. If $p == 0$ then the sender is in slow start state and should increase sending rate once per RTT in accordance with Equation 4.4. If feedback indicates that $p > 0$ then the Equation 4.5 should be used where X_{calc} is the rate estimation equation outlined in Section 4.3.3. The sender can now schedule data transmissions at the appropriate rate.

$$X = \max\left(\min\left(2 * X, 2 * X_{recv}\right), \frac{2}{t_{mbi}}\right) \tag{4.4}$$

$$X = \max\left(\min\left(2 * X, 2 * X_{recv}\right), \frac{s}{R}\right) \tag{4.5}$$

As outlined in the description of the high level state machine, the sender has a no-feedback timer. This timer keeps track of feedback connectivity from the receiver. It insures that feedback is received at regular intervals in order to maintain stable sending rate. The sender resets this timer each time the sending rate is updated using Equation 4.6. If the no-feedback timer expires, (i.e. no feedback has been received from the receiver) the sender immediately halves the sending rate.

$$t_{NoFeedback} = \max\left(4 * RTT, 2 * \frac{s}{X}\right) seconds \tag{4.6}$$

4.3.3 Rate Estimation Equation

As mentioned above, the GCA mechanism uses a novel rate estimation formula presented in Equation 4.7. It is based on a modified version of the TCP Reno throughput equation (which is designed to compete fairly with TCP) with modifications to allow to operate in a TCP compatible manner. This compatibility allows the aggressiveness of the formula to be tuned in order

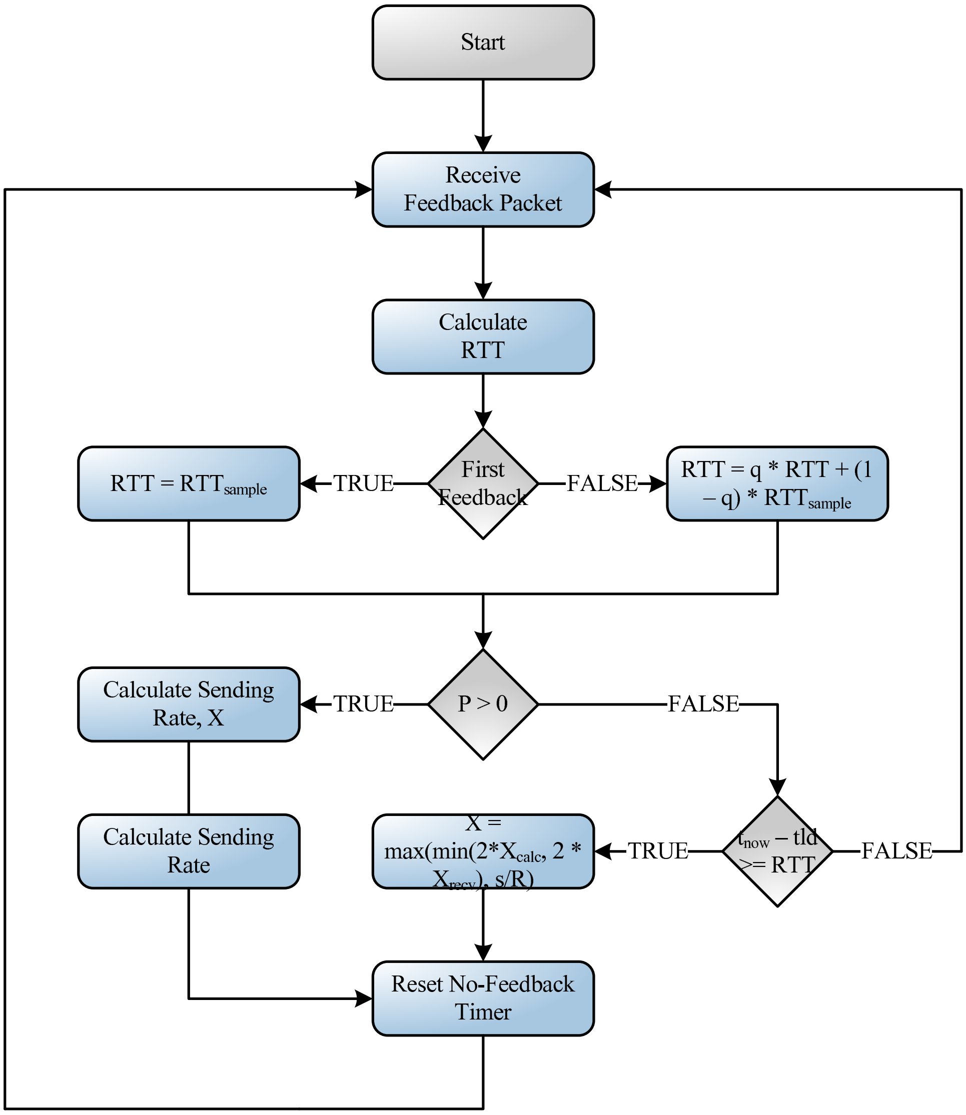


Figure 4.5: Receive Feedback Procedure

to obtain the required goal of application level fairness. The sending rate in Equation 4.7 is determined as a function of Round Trip Time (RTT), loss event rate (p) and packet size (s). α and β ($\delta = 1/\beta$) are specially proposed parameters that tune the aggressiveness of the rate estimation. The aggressiveness parameters are derived from the stochastic TCP model presented in

[65] and the methodology used in [66].

$$X = \frac{s}{RTT(\sqrt{\frac{2p(\delta-1)}{\alpha(\delta+1)}} + 12 \times p\sqrt{\frac{p(\delta-1)(\delta+1)}{2\alpha\delta^2}}(1 + 32p^2))} \tag{4.7}$$

Using this equation and by varying α and β, it is possible to configure GCA flows so that they are either more or less aggressive, thus adapting the transport layer rate estimation to suit the adapted application layer multimedia process.

4.4 GCA Receiver

GCAs receiver is responsible for processing received data packets, calculating loss event rate and providing feedback to the GCA sender. Feedback reports contain loss event calculation and timestamp information that allow the sender to estimate RTT. Regular feedback is generated and sent periodically, at least one feedback report per RTT unless sending rate is less than one packet per RTT. Emergency feedback is also sent immediately (without waiting for the next schedule feedback interval) if a loss event is detected.

4.4.1 High Level Operation

Like the sender, the GCA receiver can be in one of three states: *listen*, *calculate* and *send feedback* as illustrated in Figure 4.6. The *calculate* and *send feedback* states are transient. Various events trigger changes of state based on the current state. The state machine is initialised with the following parameters: loss event rate, feedback timer interval and receive rate.

The receiver is initialised with the reception of the first data packet from the sender. After initialisation the receiver enters the listen state. In this state the receiver will loop listening and receiving data packets from the sender. This state is exited when the logic detects a loss event or a scheduled

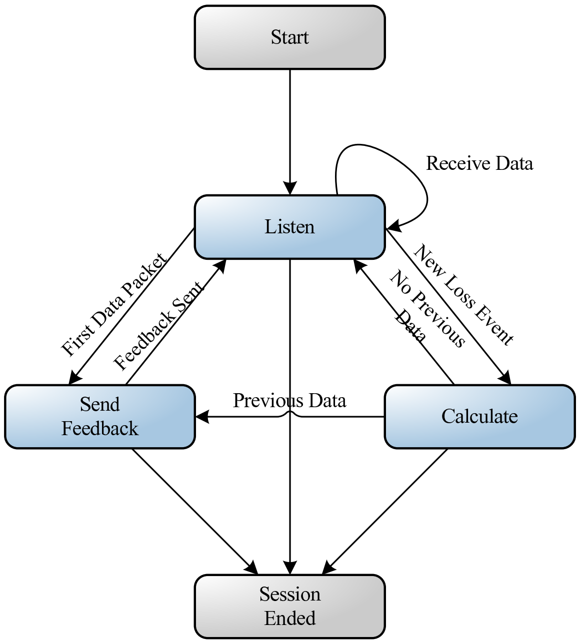


Figure 4.6: GCAs receiver state machine

feedback event. A state transition also takes place when a first data packet is received. This event requires the receiver to send a feedback packet in order to initialise the sender. The *calculate state* computes the received rate and resets the feedback timer. The *send feedback* state is also transient. A receiver will only be in this state while feedback is being transmitted. Once feedback has been sent receiver will return to the *listen* state.

Again changes in state are triggered by events. These events: *receive data*, *receive first data*, *previous data*, *no previous data*, *new loss event* and *feedback timer expired*, are fired when certain conditions are met. Receive data event occurs when data is received, and causes the listen state to perform some calculations. Using the information from these calculations the listen state either triggers a *receive first data event* or a *new loss event*. This will cause the sender to transition to either the *send feedback* or *calculate* states. A transition to the *calculate* state can also be trigger by the expiration of the *feedback timer*. Once in the *calculate state* more computations are performed. These calculations result in the triggering of either a *no previous data* event, which will return the receiver to the *listen* state, or the *previous data* event, causing the a transition to the *send feedback* state.

4.4.2 Low Level Operation

The process carried out by the receiver when data is received is illustrated in Figure 4.7. This process is initialised with the parameters listed in Table 4.2 when the first data packet is received.

Parameter	Value
Loss Event Rate (p)	0
Feedback Timer	RTT
Received Rate (X_recv)	0

Table 4.2: GCA receiver initialisation parameters

When a packet is received, payload is extracted and the header information is processed and added to the packet history. The information stored in this history included sequence number, transmission timestamp and reception timestamp. If this data packet was the first packet received, then the initialisation parameters should be set, feedback packet should be sent immediately to the sender and the feedback timer should be reset. Otherwise the the new loss event rate should be calculated. If this new loss event rate p is greater than the previous loss event rate p_{prev}, then the feedback timer should expire causing a feedback packet to sent. In other words, inform the sender immediately of change in loss event rate. Otherwise repeat the whole procedure again!

After initialisation, the receiver should send at least one feedback packet per RTT to the sender. This feedback is scheduled using a feedback timer. The timer continuously times-out and resets based on the senders current

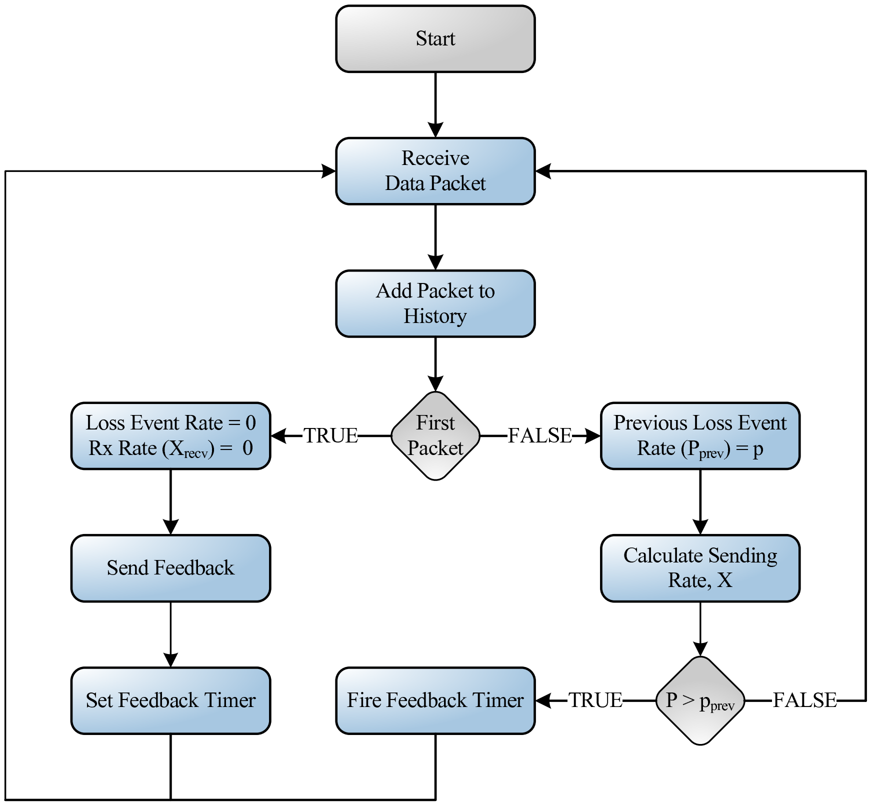


Figure 4.7: GCAs receiver process when data is received

RTT estimate. As outlined above the timer also times-out at unscheduled times when the loss event rate increases. The operation of the feedback timer is illustrated in Figure 4.8. When the feedback timer expires it checks to see whether data has been received since the last expiration. If no data has been received the timer is reset and the node continues to listen for data. If data was received the node proceeds to calculate the rate at which data has been received (X_{recv}) and the loss event rate (p). The exact details of how these parameters are calculated are outlined in Section 4.4.3. A feedback packet is now compiled and sent to the sender for processing. The receiver now sets the feedback timer to expire after the current RTT estimate. The receiver now returns to its listening state waiting for the arrival of the next data packet.

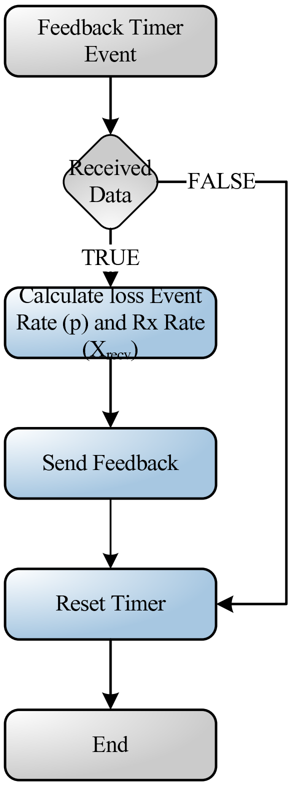


Figure 4.8: GCAs receiver send feedback

4.4.3 Loss Event Rate Calculation

As outlined above the receiver in the GCA session is responsible for calculation of the loss event rate. In general, the loss event rate is a measurement of the rate at which lost or marked (by routers using the ECN protocol as an early warning for potential congestion and thus loss) packets occur based on the sequence numbers of packets arriving at the receiver.

The GCA receiver implements the algorithm outlined in Figure 4.8 for the reception of each data packet. This procedure involves the analysis and maintenance of a data structure call the *loss history* that keeps track of which packets have arrived and which packets have not. More specifically this structure maintains timestamp and sequence number information of received packets. Using the sequence numbers of the packets stored in this structure, the receiver is able to determine whether or not a previous packet was lost or received out of sequence.

A loss is defined by the arrival of at least three packets with a higher sequence number than the lost packet. Loss intervals are used for determining the loss event rate. A loss interval, is defined as the number of packet received between successive loss events. These intervals are illustrated in Figure 4.9. In order to improve the robustness of the mechanism, a loss event is defined as the loss of one or more packets during the same RTT interval. This prevents the mechanism from reacting too aggressively to consecutive lost packets that are part of the same congestion event. TCP takes a similar approach to consecutive losses.

In order to derive the results outlined above the receiver must map the packet loss history into a loss event record. This mapping is accomplished by analysis the comparing the sequence numbers and received timestamps of packets in the Loss History. This loss event record determines the boundaries between successive loss events.

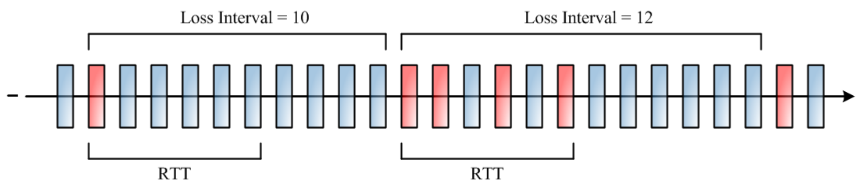


Figure 4.9: Loss intervals

Using this loss event record the receiver can now determine the loss intervals and thus the average loss interval. As outlined above the loss interval is the number of packets received between two consecutive loss events. This is essentially a count of the number of packets between a lost packet that begins a loss event and the next lost packet that is received at least an RTT

```
FOR i = 0 TO n − 1
IF i < n/2
w_i = 1
ELSE
w_i = 1 − (i − (n/2 − 1))/(n/2 + 1)
ENDIF
```

Figure 4.10: Average loss interval weights calculation

from the start lost packet. The weighted average of the last n loss intervals (typically $n = 8$) is now calculated using the algorithm outlined in Figure 4.10. The weights used in calculating this average are particularly important as they specify the degree of importance assigned to a given loss interval. The newer the loss interval the more weight it carries in determining the loss event rate as illustrated in Figure 4.11.

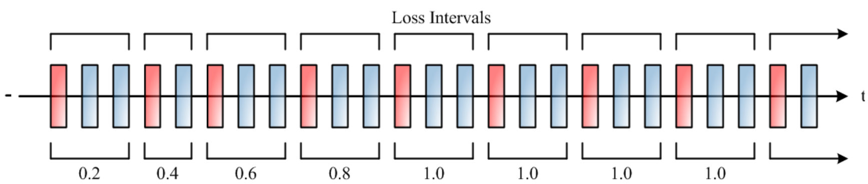


Figure 4.11: Loss intervals weights

The loss event rate p can now be calculated using the Equation 4.8, where I_n is the loss interval and W_i is the weight.

$$p = \frac{1}{\max\left(\frac{\sum_{i=0}^{n-1} I_i W_i}{\sum_{i=0}^{n-1} W_i}, \frac{\sum_{i=1}^{n} I_i W_{i-1}}{\sum_{i=1}^{n} W_{i-1}}\right)} \tag{4.8}$$

4.5 Summary

This chapter has given a comprehensive description of the operation of the various components of the proposed GCA mechanism. It has outlined the architecture and discussed where the GCA protocol fits into the streaming process. Detailed information about the behaviour of the sender and receiver were also presented.

Chapter 5

Analysis and Testing

This chapter investigates the equality inconsistency between transport and application layer outlined in Chapter 1 and demonstrates the results of the proposed solution, for enabling proportional distribution of bandwidth base on device requirements. The chapter is divided into three sections. Section 5.1 presents a analysis of the proposed solution in order to evaluate where possible performance gains can be made. Section 5.2 presents details of the simulation environment used for simulating the proposed solution, while section 5.3 presents simulation results that compare and contrast the benefits of the proposed solution against two other streaming solutions.

5.1 Analysis

In this section an analysis of the GCA throughput equation (Equation 4.7), presented in Section 4.3.3, is performed to determine how each of the parameters affect the overall throughput of the system. The results of this analysis are illustrated in Figures 5.1(a) through Figure 5.1(d). This analysis uses a constant packet size of 1,024 Bytes, RTT of 30 ms and loss event rate of 0.0001 unless otherwise stated. The simulated flow competes fairly with a similarly configured TCP flow when $\alpha = 1.0$ and $\beta = 0.5$. The throughput represented by these conditions in each plot is known as the reference point.

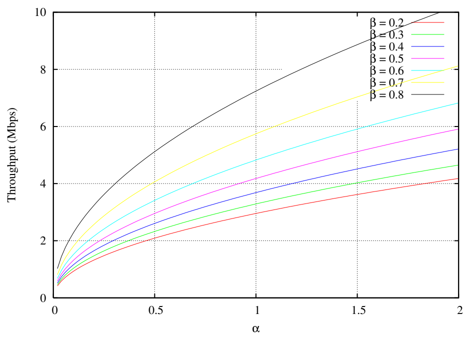
The first set of analysis results, illustrated in Figure 5.1(a), evaluates the effect varying α in the interval $0.0-2.0$ has on throughput for different values of β between $0.2-0.8$. The reference point of this plot returns a throughput of 4.1 Mbps. Analysis of the curve represented by $\beta = 0.5$, indicates that the throughput changes considerably over the range α. There is almost $6Mbps$ increase in throughput when α is varied between $0.0-2.0$. A similar variation is observed for other values of β. Further scrutiny of data concludes that $\sqrt{\alpha}$ difference in throughput is achieved for each instance of β.

Next, the effect of varying the β between $0.0 - 1.0$ for discrete values of α in the interval $0.25 - 1.75$ is evaluated. The result of this analysis is illustrated in Figure 5.1(b). First inspection of the results indicates that the throughput grows exponentially when β is varied between $0.0 - 1.0$. The reference point throughput only experiences a relatively small variation for β between $0.0 - 0.8$. Infinite throughput is experienced as β converges on 1.0. This makes $\beta > 0.8$ unsuitable for bit-rate tuning of multimedia streams due to its instability. The linearity between $0.0 - 0.8$ makes β suitable as a tuning parameter.

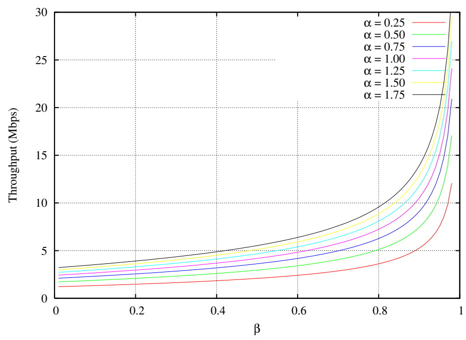
Figures 5.1(c) and 5.1(d) analyses the effect of varying loss event rate p have on certain values of α and β. As expected the higher the loss event the lower the throughput obtained. These graphs also illustrate GCAs ability to maintain proportional fairness between various flows for varying loss event rates. This is an essential characteristic for gaining the required compatibility to enable prioiritisation and thus fair video quality distribution at the application layer.

5.2 Simulation Setup

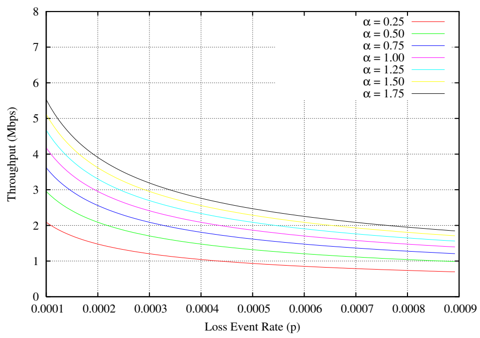
Simulation based analysis was chosen to highlight the benefits of the proposed GCA. The advantages of this approach are well known, as it provides a means



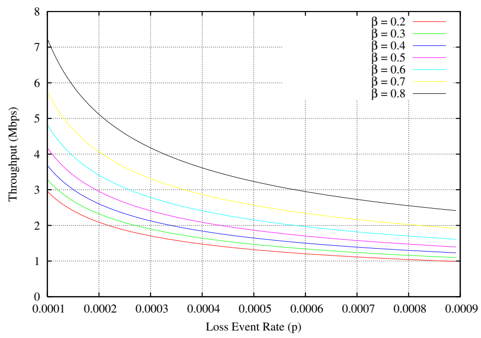
(a) Plot illustrates the effect varying the α parameter has on throughput for various values of β



(b) Plot illustrates the effect varying the β parameter has on throughput for various values of α



(c) Illustration of the effect on throughput of varying loss event rate between 0.0001 and 0.0009 for various values of α



(d) Illustration of the effect on throughput of varying loss event rate between 0.0001 and 0.0009 for various values of β

Figure 5.1: Analysis of GCA throughput estimation equation with various α and β parameters

of testing various scenarios in an efficient and cost-effective manner.

There are a number of network simulators in the market. Optimised Network Engineering Tools (OPNET)[1], Global Mobile Information Systems (GloMoSim)[2] and Network Simulator (NS)[3] are the most appropriate simulations environments for this work as all three simulators incorporate well-developed wireless models. However, they provide relatively little support for simulating video streaming. OPNET has an extensive feature set but also has steep learning curve associated with it. It is also a commercial simulator, and therefore has licences fees associated with it. Academic licenses are freely available on application for limited periods of time only. GloMoSim is the academic version of Qualnet[4]. It provides limited documentation of its libraries. NS is an open source simulator developed and contributed to by various members of the research community. As a result it has extensions for many different applications, protocols and traffic models. It also has been extensively tested. For these reasons NS was chosen as the simulation environment for this work.

Although simulators have many advantages associated with them, this comes at a price. Simulators have an inherent trade off between computational complexity and realism. As a result simulators will have decreased level of detail outside the area under analysis [67] [68]. However, it has been shown that by using increased levels of abstraction the validity of the simulations can be maintained [69].

5.2.1 Simulation Environment: Network Simulator (NS)

NS is an open source discrete event network simulator. It supports the simulation of a wide range of network protocols in both wired and wireless environments.

[1] http://www.opnet.com/
[2] http://pcl.cs.ucla.edu/projects/glomosim/
[3] http://www.isi.edu/nsnam/ns/
[4] http://www.qualnet.com/

NS is a variant of the Realistic and Large (REAL) network simulator that was developed in 1988. The first version of NS was released in 1995. Development was initially supported by Defence Advanced Research Projects Agency (DARPA) through the VINT project. A second version of NS was released in 1996, which became known as NS2. Currently NS development is supported through the DARPA by the Measurement and Analysis for Networks (SAMAN) project and through National Science Foundation (NSF) by the Collaborative Simulation for Education and Research (CONSER) project, which both work in collaboration with other researchers including ICSI Centre for Internet Research (ICIR). NS also includes substantial contributions from other researchers, such as wireless code from Carnegie Mellon University (CMU) Monarch project[5] and Sun Microsystems[6].

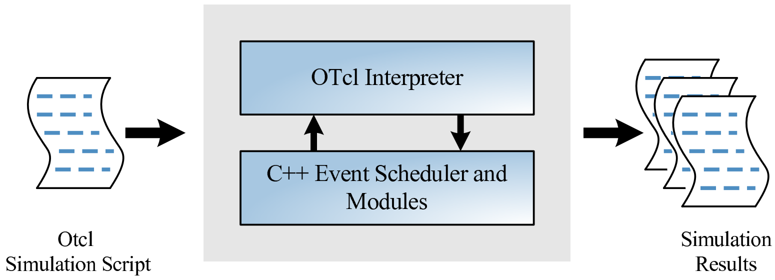


Figure 5.2: NS Architecture

The current version of NS, NS2 is an object oriented simulator written in C++ and Object-Oriented Tcl (OTcl). It is essentially an OTcl script interpreter that interfaces with a C++ discrete event network simulation. This is done to create balance between ease of use and efficiency. Efficiency is achieved by separating the control path from the data path implementations. The data path components, event scheduler and basic network component objects, are written and compiled using C++ to reduce event execution time. These objects are then linked to an OTcl interpreter through linkage

[5]http://monarch.cs.cmu.edu/
[6]http://www.sun.com/

that creates a matching OTcl object for each C++ object. This linkage also makes the control functions and the configurable variables specified by the C++ object act as member functions and member variables of the corresponding OTcl object. In this way the entire simulation environment can be easily controlled using user configurable OTcl scripts. An illustration of the architecture of NS is presented in Figure 5.2.

5.2.2 Simulation Topology and Settings

An infrastructure based WLAN topology was used for simulations as illustrated in Figure 5.3. This topology consisted of a *media server* connected to a AP via a high capacity wired link with negligible delay and sufficient bandwidth to carry all traffic without congestive loss.

The topology's WLAN was configured to simulate a IEEE 802.11g environment using the parameters outlined in Table 5.1. All nodes were positioned within carrier sense range to ensure that no hidden / exposed nodes existed. They were also positioned to ensure that all nodes were able to transmit at the highest data rate supported by IEEE 802.11g.

Parameter	Value
Data Rate	54 Mbps
Basic Rate	6 Mbps
CW_min	15
CW_max	1023
Slot Time	9 us
SIFS	16 us
CCA Time	4 us
Short Retry Limit	7
Long Retry Limit	4
RTS/CTS	Enabled

Table 5.1: IEEE 802.11g WLAN parameters

The topology consists of a varying number of multimedia enabled nodes.

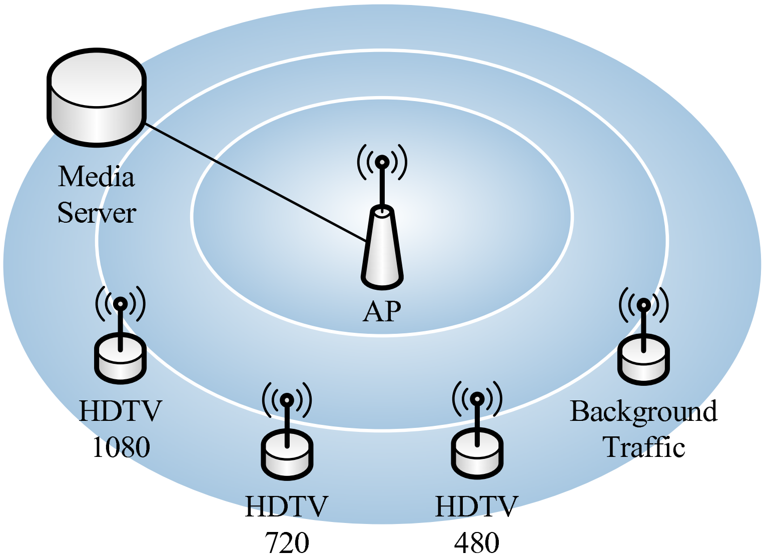


Figure 5.3: Wireless streaming topology

Each node had specific characteristics which required it to receive a certain type of video in order to achieve maximum QoE. There were three main types of devices supported by the simulation: HDTV 1080, HDTV 720 and HDTV 480. The characteristics of these devices are outlined in Table 5.3. These devices required video with characteristics outlined in Table 5.2 to achieve 100% level of user QoE relative to the devices properties.

Simulations evaluated the streaming of multimedia content specifically tailored to the characteristics of the destination device under a number of different scenarios. These scenarios were chosen to highlight the benefit achieved using the proposed GCA. Comparison were performed against non-adaptive [70] and TCP Friendly Rate Control (TFRC) [33] based streaming solutions.

Results were obtained using a combination of NS simulations and the Evalvid Video Framework [71]. This enabled traffic traces to be extracted from simulations to be applied to real video files. This process simulated the transmission of video through a WLAN environment. This methodology was applied to the Disney's Pixar Ratatouille trailer shown in Figure 5.4. The characteristics of this trailer are detailed in Table 5.2.

Figure 5.4: Disney's Pixar Ratatouille trailer used for simulation

	1080	720	480
Video Name		Ratatouille	
Run Time		2 min 29 s	
Encoding	H.264	H.264	H.264
Resolution (pixels)	1920x800	1280x532	848x352
Frame Rate (fps)	23.98	23.98	23.98
Average Bit Rate (kbps)	9,853.12	6,421.22	2,329.86

Table 5.2: Video stream parameters

	HDTV 1080	HDTV 720	HDTV 480
Screen Resolution (pixels)	1920x1080	1280x720	x480
Screen Size (inches)	32	24	16
Location	Living Room	Kitchen	Bedroom

Table 5.3: Simulation device parameters

5.3 Simulation Analysis

This section presents the simulation analysis for a streaming solutions employing: non-adaptive, TFRC and the proposed GCA rate control mechanisms. The results compare the three schemes in relation to transport layer throughput, loss, delay and jitter metrics and application layer PSNR video quality assessment metrics. The section is concluded with a discussion of the results.

5.3.1 Non-Adaptive Streaming

The first series of simulations evaluate the streaming multimedia content presented in Table 5.2 to the devices presented in Table 5.3 using a non-adaptive streaming solution. Non-adaptive streaming is the least complex of all streaming solutions. With non-adaptive streaming, multimedia data is streamed between two end-points that do not consider available network conditions. These solutions stream content at bit rates that might not be supported under current network conditions. They do not employ congestion control mechanisms to reduce loss, maintain network stability, enable scalability, while maximising throughput.

The non-adaptive streaming solutions considered for these simulations are assumed to employ device discovery mechanisms, such as DLNA[7], that can determine the media requirements of the devices to which the media is being streamed. This discovery mechanism allows the streaming solution stream appropriately configured media in terms of bit-rate to the end devices given their screen size in order to achieve maximum QoE under ideal network conditions. It is assumed that content is streamed at the bit rate at which it is encoded using a Constant Bit Rate (CBR) approach.

[7]http://www.dnla.org

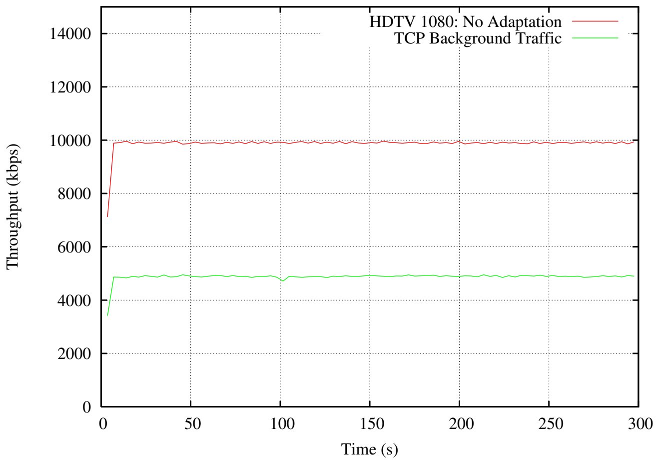
Simulation A

The first simulation evaluated the performance of the non-adaptive solution for a single HDTV 1080 device in a WLAN with a single background traffic source. The background source was configured to simulate the transfer of a large file using FTP over TCP. The simulation was configured to run for $300.0s$. The HDTV streaming flow and FTP background traffic source were added to the network at $t = 1.0s$ and $t = 2.0s$ respectively. The throughput and loss analysis of this simulation is illustrated in Figures 5.5 and detailed summary is presented in Table 5.4.

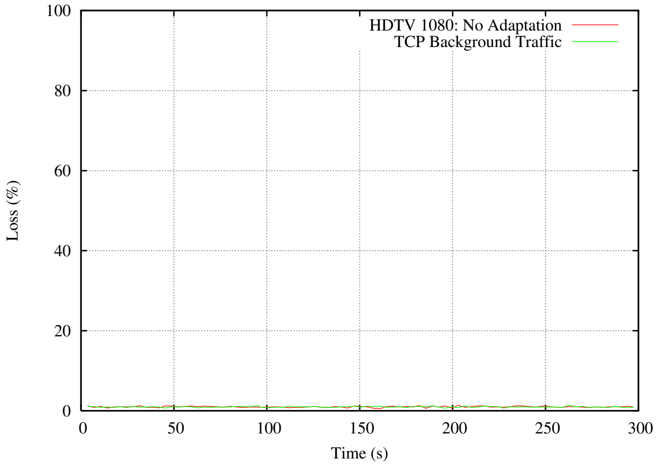
These results indicate very good performance of the non adaptive scheme. Although bandwidth is not shared equally between the applications each of them are experiencing very good application layer metrics. The HDTV 1080 device is achieving an average throughput of nearly *9.6 Mbps* with very small variance. This throughput experienced at the transport layer is almost equal to the throughput required by the application to achieve the desired 100% level of QoE at the application layer. This result is confirmed by application layer PSNR measurements of *88 dB*. The difference between achieved PSNR and expected ideal PSNR can be attributed to the slightly lower achieved throughput and losses. These losses remain at acceptable levels as reflected in the high average PSNR measurement. The FTP background traffic sources is achieving average throughput of nearly *4.9 Mbps* and experiencing very small losses of less than *1 %*. Its reduced throughput in comparison with the HDTV 1080 device, has little impact on the FTPs payload due to it best effort, non time critical nature. End-to-end delays and jitter for both applications remain within acceptable ranges for the duration of the simulation.

Simulation B

The second non-adaptive simulation evaluated the performance of three devices, a HDTV 1080, HDTV 720 and the FTP background traffic source. Theses devices joined the WLAN at $t = 1.0s, 2.0s, 3.0s$ respectively. The



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.5: Non-adaptive streaming to HDTV 1080 and FTP background traffic

	HDTV 1080	FTP
Throughput (kbps)	9,606.17	4893.64
Loss (%)	0.99	0.96
Delay (ms)	21.12	20.61
Jitter (ms)	0.8272	0.5027
PSNR (dB)	88.62	-

Table 5.4: Non-adaptive streaming analysis summary for HDTV 1080 together with FTP background traffic source

duration of the simulation was 300.0s. The throughput and loss analysis of this simulation is illustrated in Figure 5.6 and detailed summary is presented in Table 5.5.

The throughput analysis for this simulation shows both streaming devices, HDTV 1080 and HDTV 720, receiving adequate throughput to achieve high application layer QoE. This is confirmed by both application layer and transport layer measurements. The each receive *9.5 Mbps* and *6.1 Mbps* throughput respectively. This again translates into high application layer PSNR scores of *82.73 dB* and *80.98 dB* respectively. Loss for these devices also remains acceptably low, however it has increased slightly compared with the previous simulation. Although the application and transport layer results for both multimedia enabled devices are ideal for streaming and maximising overall QoE, this is at the experience of the background traffic source. The lack of congestion control implementation in the streaming application is beginning to result in starvation of the FTP host. Although this traffic is best effort, non time critical in nature, the network still needs to provide it with some QoS. Otherwise this will being to impact of the QoE of the user or system process transferring the possibly critical data over the network. This source has experienced an *83 %* drop in throughput compared with the previous simulation while the HDTV 1080 device has experience no degradation of throughput. Again loss, delay and jitter measurements remain within acceptable ranges for all devices.

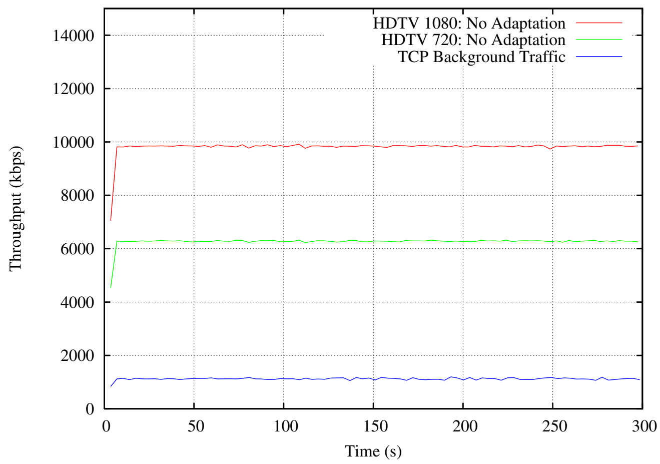
	HDTV 1080	HDTV 720	FTP
Throughput (kbps)	9,565.82	6,284.95	1,125.08
Loss (%)	1.61	1.68	7.13
Delay (ms)	20.87	21.29	21.75
Jitter (ms)	0.8323	1.30	0.78
PSNR (dB)	82.73	80.98	-

Table 5.5: Non-adaptive streaming analysis summary with HDTV 1080 and HDTV 720 media devices together with an FTP background traffic source

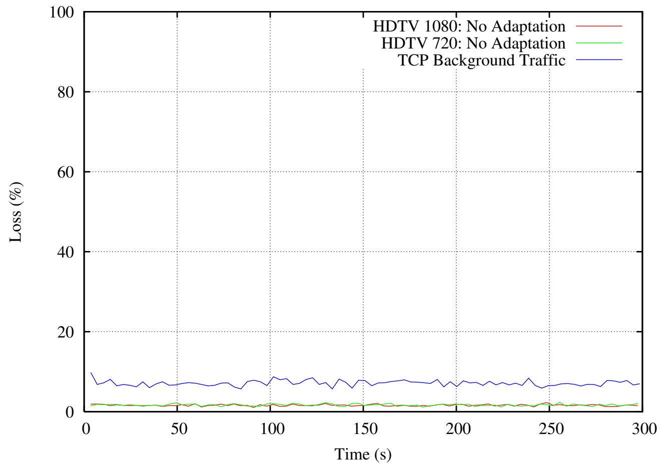
Simulation C

The third simulation added a fourth media device to the scenario. The simulation now consisted of a HDTV 1080, HDTV 720, HDTV 480 and the FTP background traffic source. AS in previous simulations these devices joined the WLAN at $t = 1.0s, 2.0s, 3.0s$ and $4.0s$ respectively. The duration of the simulation was $300.0s$. The throughput and loss analysis of this simulation is illustrated in Figures 5.7 and detailed summary is presented in Table 5.6.

The addition of the third media device (HDTV 480) has had very little impact on the throughput of the other media devices. However, greediness of the non-adaptive solution becomes apparent with the unacceptable starvation experienced by the FTP source. The HDTV 1080 device is still receiving *9.5 Mbps*. However the loss measurements have quadrupled resulting in much lower values of PSNR. Delay for this device still remains within an acceptable range. The HDTV 720 is receiving approximately *6 Mbps* which is *5 %* less throughput than required and experiences a loss rate of about *5 %*. A similar throughput reduction and increased loss is experienced by the HDTV 480 device. From an application layer perspective each of these devices are not achieving the same high PSNR results as previous simulations, due to the increased loss rate. However, the TCP based FTP source is experienc-



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.6: Non-adaptive streaming analysis with HDTV 1080 and HDTV 720 media devices together with an FTP background traffic source

	HDTV 1080	HDTV 720	HDTV 480	FTP
Throughput (kbps)	9,458.68	6,088.82	2,233.46	5.87
Loss (%)	4.47	4.75	5.09	43.55
Delay (ms)	22.36	23.89	24.10	22.21
Jitter (ms)	0.8573	1.3459	3.5120	0.4334
PSNR (dB)	67.34	62.46	57.84	-

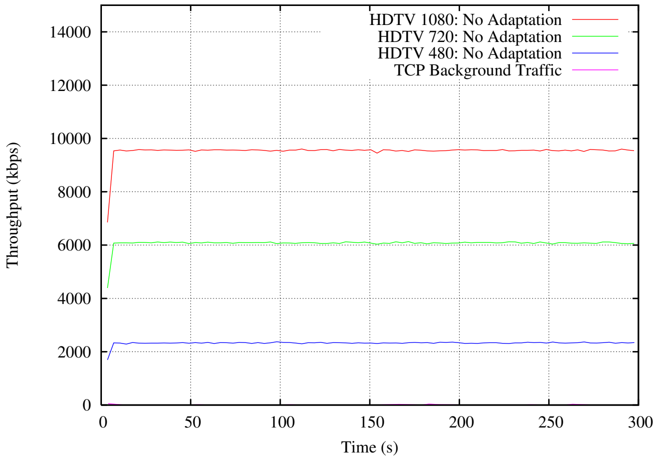
Table 5.6: Non-adaptive streaming analysis summary for HDTV 1080, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source

ing severe starvation which resulting in unacceptable QoE for the user or application using this transport mechanism.

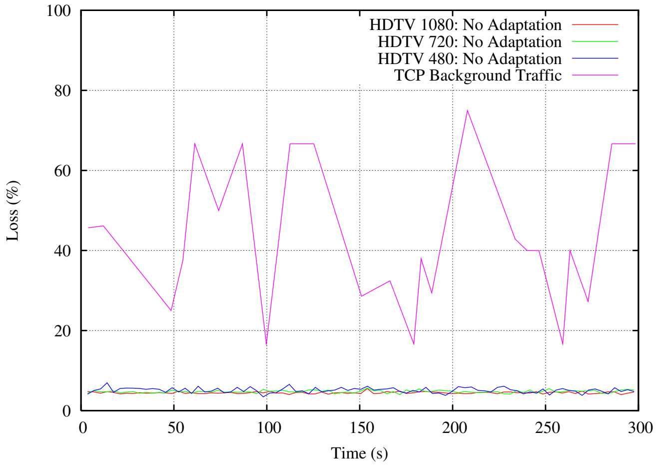
Simulation D

The simulation now included a HDTV 1080, HDTV 720, HDTV 480, another HDTV 1080 and the FTP background traffic source. As in previous simulations these devices joined the WLAN at $t = 1.0s, 2.0s, 3.0s, 4.0s$ and $4.0s$ respectively. The duration of the simulation was $300.0s$. The throughput and loss analysis of this simulation is illustrated in Figures 5.8 and detailed summary is presented in Table 5.7.

The final non-adaptive simulation again illustrates the problems of this technique for streaming media. As outlined in the previous paragraph this simulation involves five clients. Initial inspection of the graphs again shows the starvation experienced by the FTP, and increased loss rates for each of the media devices. The addition of the a second HDTV 1080 device has resulted in the required throughput exceeding the WLANs available throughput. The non-adaptive solutions are unaware of these network conditions and continue to transmit as though they have access to an unlimited bandwidth link. This results in unacceptably high loss rates for streaming media. The HDTV 1080 is still managing to achieve relatively high throughput and low loss of *8.8*



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.7: Non-adaptive streaming analysis with HDTV 1080, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source

	HDTV 1080	HDTV 720	HDTV 480	HDTV 1080	FTP
Throughput (kbps)	8,799.08	4,376.29	1,548.35	3,262.79	1.21
Loss (%)	12.05	31.53	36.98	67.32	32.74
Delay (ms)	23.15	24.56	30.23	27.43	20.45
Jitter (ms)	0.9313	1.8926	5.2928	2.5063	1.2105
PSNR (dB)	25.13	10.64	8.62	5.09	-

Table 5.7: Non-adaptive streaming analysis summary for two HDTV 1080s, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source

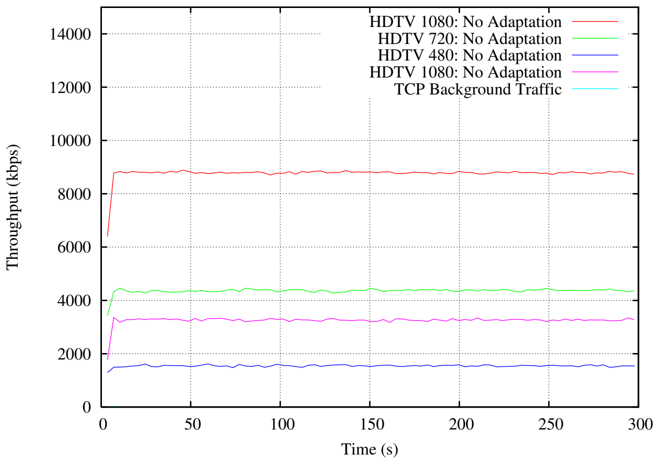
Mbps and *12 %* compared with other devices. The HDTV 720 and HDTV 480 are also experiencing *35 %* reduced throughput and *35 %* increase in loss. This reduction and increase in throughput and loss is reflection in decrease application layer PSNR measurements for these devices. However, the second HDTV 1080 device is only receiving a throughput of *3.3 Mbps* and is experiencing almost *70 %* loss. These low transport layer measurements result in unacceptable average application layer PSNR of *5.09 dB*.

Summary

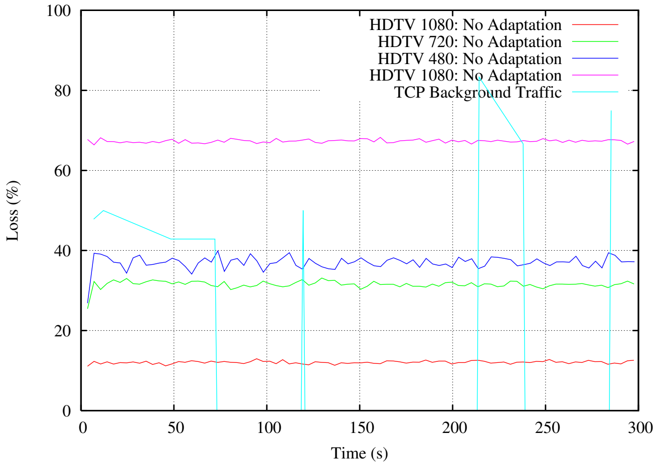
Although initial simulations provided promising results for this non-adaptive streaming solution, the increased number of devices in latter simulations led to a drastic decrease in overall QoE of media enabled and best effort applications. The high loss levels and starvation experienced by TCP enable streams is unacceptable. The initial simulations achieved the required goal of application layer fairness. However, without stability and scalability the non-adaptive streaming solutions is a non runner.

5.3.2 TFRC Based Streaming Solution

TFRC based streaming solution is now evaluated using the same simulation scenarios used for the non-adaptive streaming solution. This simulations



(a) Throughput Analysis



(b) Loss Analysis

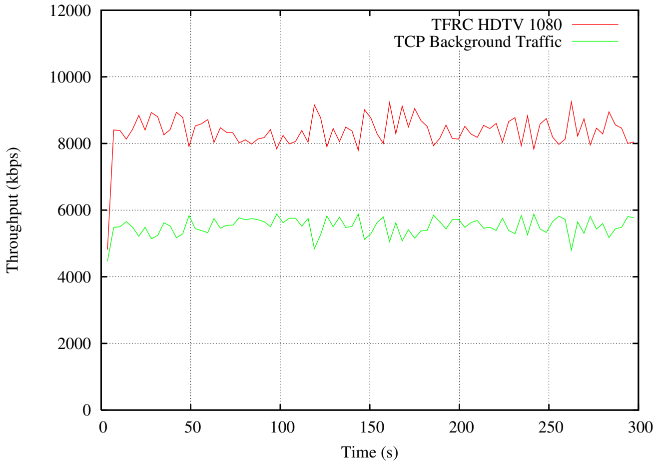
Figure 5.8: Non-adaptive streaming analysis with two HDTV 1080s, HDTV 720 and HDTV 480 media devices together with an FTP background traffic source

also used the media content outlined in Table 5.2 for playback on devices presented in Table 5.3. TFRC streaming is a more intelligent solution for delivery of multimedia content. It employs a feedback mechanism that allows the sender to estimate the available network conditions and adapt the transmission rate media accordingly. The mechanisms employed by TFRC avoid congestion collapse, maintain fairness between different flows, while maximising throughput and minimising loss.

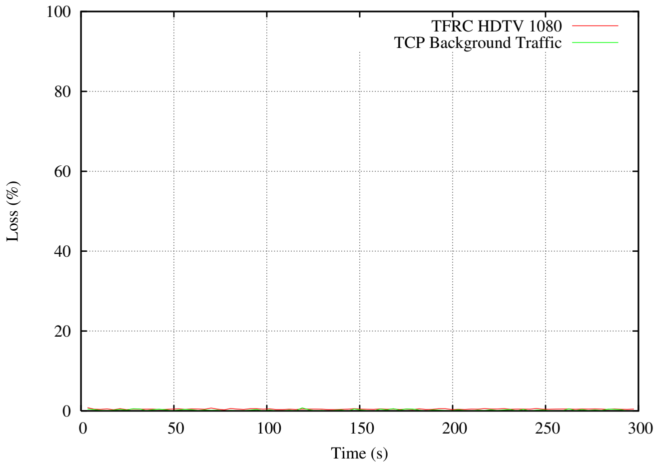
Simulation A

Like the non-adaptive, this simulation evaluated the performance of the TFRC streaming solution for a single HDTV 1080 device in a WLAN with a single TCP based traffic source. The background source was configured to simulate the transfer of a large file using FTP. The acHDTV and FTP traffic source were added to the network at $t = 1.0s$ and $t = 2.0s$ receptively. The throughput and loss analysis of this simulation is illustrated in Figures 5.9 and detailed summary is presented in Table 5.8.

Initial inspection of results for this scenarios indicate increased levels of competition between the TFRC streaming solution and the FTP traffic source. although there is still nearly *3 Mbps* difference in bandwidth between the media and data streams they are competing fairly for available resources. As a result of this competition the HDTV is receiving approximately *15 %* less bandwidth than required, while the FTP traffic source is receiving more than adequate throughput. Measured loss in both cases remains exceptionally low, due to the efficiency of the rate control mechanism at avoiding congestion. Delay for both applications also remains stable at *24 ms.* From an application layer perspective the HDTV 1080 PSNR measurements are only *58 dB*. This is considerably lower then *88.62 dB* what was achieved during the non-adaptive simulation.



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.9: Throughput analysis

	HDTV 1080	FTP
Throughput (kbps)	8,387.87	5,523.83
Loss (%)	0.45	0.22
Delay (ms)	24.93	25.27
Jitte (ms)r	0.3889	0.4297
PSNR (dB)	58.76	-

Table 5.8: TFRC streaming analysis summary for a HDTV 1080 media device together with an FTP background traffic source

Simulation B

An additional media enabled devices is now added to the simulation. This device has the same characteristics as the HDTV 720 outlined in Table 5.3. The HDTV 1080, HDTV 720 and FTP traffic source were added to the network at $t = 1.0s, 2.0s$ and $t = 3.0s$ receptively. The throughput and loss analysis of this simulation is illustrated in Figures 5.10 and detailed summary is presented in Table 5.9.

The most noticeable difference with this simulation is the increased throughptu fluctuations experienced by each of the clients in the network. Each of the media enabled client are competing fairly with one another for available bandwidth and a competing slightly more aggressively than the FTP application. First, consider the media enabled devices. The HDTV 1080 device is receiving *5.7 Mbps* of available bandwidth, which is *3.6 Mbps* or *36 %* less than required to achieve *100 %* user QoE. However, the HDTV 720 is receiving *5.4 Mbps* of available bandwidth or *12 %* less than it requires. This difference between achieved and required throughput leads to unequal distribution of quality at the application layer. This inequality becomes apparent on analysis of application layer PSNR measurements. The smaller HDTV 720 device is achieving a higher PSNR measurements of *56 dB* compared with the larger HDTV 1080 which is achieving just *46 dB*. This anomaly did not occur during the non-adaptive streaming due to the inherent greediness of non-adaptive streaming. Loss and delay measurements remain within acceptable ranges for the duration of the simulation for all devices and application. Starvation of the TCP based FTP application is reduced due to the fairness of the TFRC streaming solution. The inconsistency that occurred during this simulation is what the proposed GCA algorithm aims to solve.

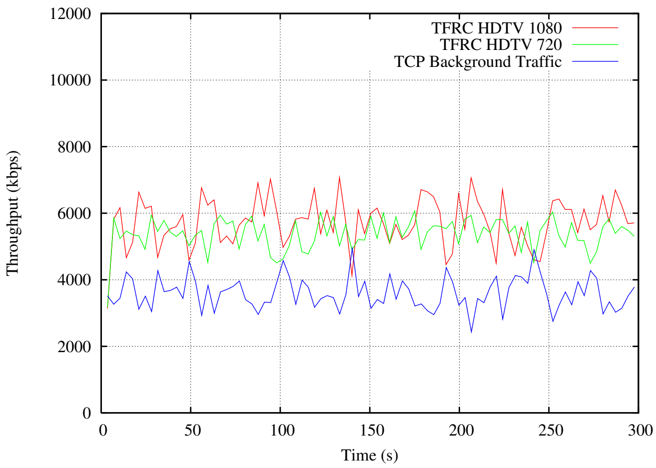
	HDTV 1080	HDTV 720	FTP
Throughput (kbps)	5,746.36	5,395.12	3,601.76
Loss (%)	0.99	1.02	1.24
Delay (ms)	22.13	22.29	23.03
Jitter (ms)	0.4817	0.5023	0.6218
PSNR (dB)	46.79	56.37	-

Table 5.9: TFRC streaming analysis summary for HDTV 1080 and HDTV 720 media device together with an FTP background traffic source

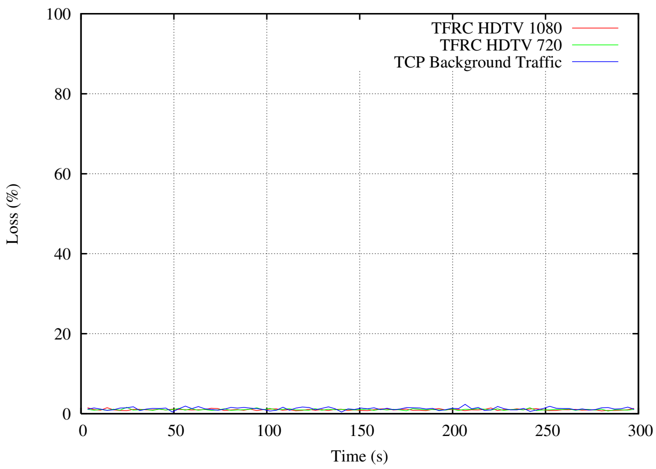
Simulation C

An addition media enabled devices is now added to the simulation. This device has the same characteristics as the HDTV 720 outlined in Table 5.3. The HDTV 1080, HDTV 720, HDTV 480 and FTP traffic source were added to the network at $t = 1.0s, 2.0s, 3.0s$ and $t = 4.0s$ receptively. The throughput and loss analysis of this simulation is illustrated in Figures 5.11 and detailed summary is presented in Table 5.10.

This simulation shows more evidence of the inconsistencies discussed in the previous simulation. Initial comparison of this simulation with the non-adaptive scenario show better bandwidth distribution better the various streams. The most noticeable difference is that the TCP based FTP application is not starved of bandwidth. The HDTV 480 receives its required bandwidth of $\approx 2.4Mbps$ of available bandwidth, resulting in high application layer PSNR measurements of *72 dB*. The HDTV 1080 and HDTV 720 nearly equal average throughput of *4.8 Mbps*. This is only 50% of the re-



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.10: TFRC streaming analysis for a HDTV 1080 and HDTV 720 media device together with an FTP background traffic source

quired throughput of the HDTV 1080 and 80% of what is required by the HDTV 720. This inconsistency is reflected in the application layer PSNR measurements of all three devices. The HDTV 480 is achieving better quality than the HDTV 720 which is achieving better quality than the HDTV 1080. All devices and applications experience very low loss rate of around 2.5% and also stable delay measurements.

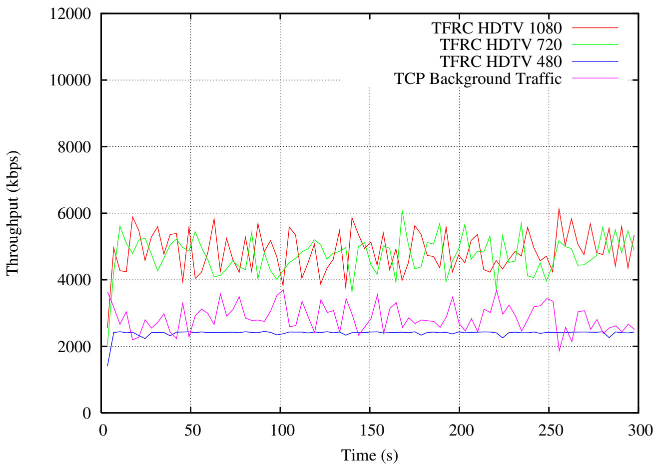
	HDTV 1080	HDTV 720	HDTV 480	FTP
Throughput (kbps)	4,863.13	4,749.06	2,405.37	2,867.88
Loss (%)	1.25	1.27	1.42	1.83
Delay (ms)	22.28	22.33	22.52	23.00
Jitter (ms)	0.5477	0.5555	0.7861	0.7102
PSNR (dB)	39.86	52.24	72.11	-

Table 5.10: TFRC streaming analysis summary for HDTV 1080, HDTV 720 and HDTV 480 media device together with an FTP background traffic source

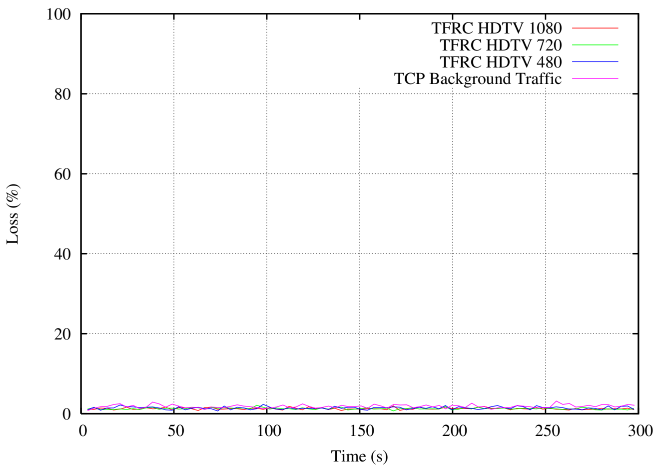
Simulation D

Finally, a second HDTV 1080 device is now added to the simulation. This device has the same characteristics as the HDTV 1080 outlined in Table 5.3. The two HDTV 1080s, HDTV 720 HDTV 480 and FTP traffic source were added to the network at $t = 1.0s, 2.0s, 3.0s, 4.0s$ and $t = 5.0s$ receptively. The throughput and loss analysis of this simulation is illustrated in Figures 5.12 and detailed summary is presented in Table 5.11.

The results of the final simulation show that the fair competition between the four media devices. The application layer inconsistencies observed in previous simulation are also present leading to unfair distribution of QoE. Closer examination of transport layer measurements show throughput of the two HDTV 1080s and the HDTV 720 converging at *3.5 Mbps*. This means that the two HDTV 1080s are only achieving *36 %* of their required throughput while the HDTV 720 is receiving *56 %* of required throughput to achieve



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.11: TFRC streaming analysis for a HDTV 1080, HDTV 720 and HDTV 480 media device together with an FTP background traffic source

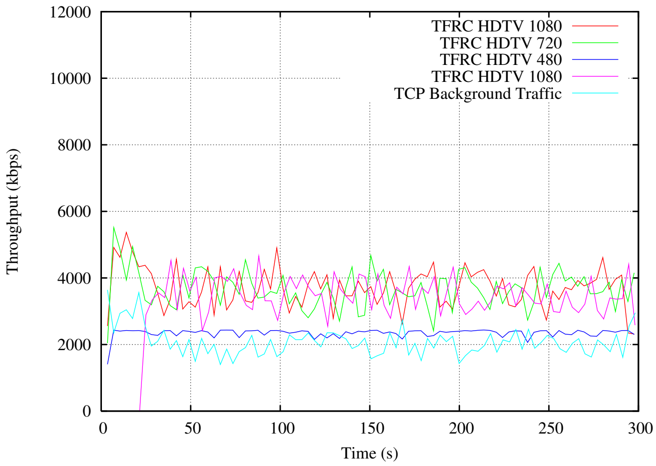
maximum QoE. However, the HDTV 480 is still obtaining approximately its required throughput of *2.4 Mbps* and achieving very high PSNR scores. Loss and delay for this simulation is with ideal range for streaming video. Also the FTP source is avoiding starvation due to the fair competition within the network.

	HDTV 1080	HDTV 720	HDTV 480	HDTV 1080	FTP
Throughput (kbps)	3,719.56	3,684.49	2,354.21	3,249.69	2,088.73
Loss (%)	1.72	1.69	1.74	3.04	2.88
Delay (ms)	22.87	22.89	22.99	22.98	23.35
Jitter (ms)	0.60	0.6058	0.7611	0.6194	0.7895
PSNR (dB)	29.46	39.76	74.46	30.46	-

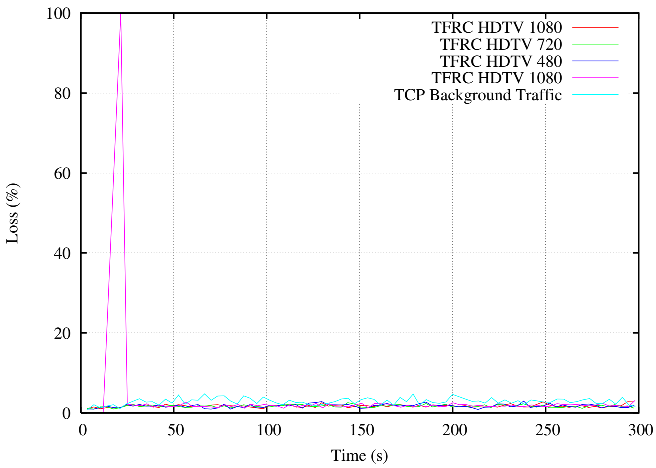
Table 5.11: TFRC streaming analysis summary for HDTV 1080, HDTV 720, HDTV 480 and HDTV 1080 media device together with an FTP background traffic source

Summary

The results of this set of TFRC simulations are conclusive. It clearly illustrates the application layer inconsistency created by TCP-friendly streaming solution. Results have illustrated that when using a multimedia enabled device with a TFRC based streaming solution, the best quality is achieved by the device with the lowest bandwidth requirement. From another perspective, the highest quality is achieved on the device with the lowest requirements! IsThis is contrary to what is expected by the user. It should also be noted that the some degree of TCP-friendliness is required to prevent starvation of TCP enabled application operating in parallel with streaming applications.



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.12: TFRC streaming analysis for a HDTV 1080, HDTV 720, HDTV 480 and HDTV 1080 media device together with an FTP background traffic source

5.3.3 GCA Streaming Solution

This set of simulations evaluates the performance of the proposed GCA based streaming solution using the same simulation scenarios considered for both the non-adaptive and TFRC based streaming solutions. Again, the media content outlined in Table 5.2 was used for streaming to the devices presented in Table 5.3. GCA is designed to increase the overall QoE in the network by adapting the rate control mechanism to suit the characteristic requirements of the multimedia content being carried. This is achieved by choosing suitable α and β parameters (see Table 5.12) that allow the rate control mechanism to dynamically adjust the streaming rate proportional to the contents requirements, thereby increasing overall QoE. The rate control mechanism employed by GCA avoid congestion collapse, maintain fairness at the application layer, while maximising throughput and minimising loss.

The GCA based streaming was compared using the same simulation scenarios used for both the non-adaptive and TFRC based streaming solutions. That is, four topology configurations of multimedia enabled devices request video streams that are adapted to suit their characteristic requirements and maximise QoE. Section 5.3.3 presents results for a single HDTV 1080 device and a FTP background traffic sourc, then the effect adding another HDTV 720 and HDTV 480 device to the above scenario. Finally, another HDTV 1080 and the results are analysed.

	HDTV 1080	HDTV 720	HDTV 480
α	4.0	3.0	1.0
β	0.6	0.5	0.5

Table 5.12: GCAs α and β parameters applied multimedia enabled devices for the duration of this simulation

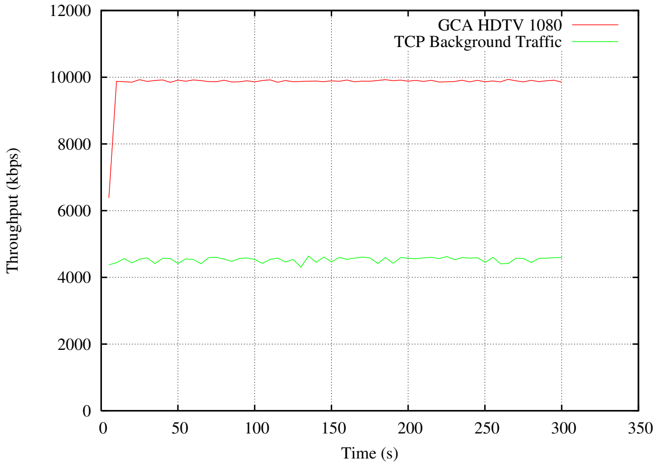
Simulation A

As mentioned previously this scenario simulates a residential network that contains a single HDTV 1080 device receiving content from a central media server. The simulation also includes a FTP background traffic source to simulate the transfer of a large file. These devices begin their respective tasks at $t = 1.0s$ and $t = 2.0s$ receptively. An analysis of the results is illustrated in Figures 5.13 and summarised in detail in Table 5.13. Initial inspection of the results indicate that the GCA enabled streaming solution is achieving similar performance to the non-adaptive simulation and substantially better performance then the TFRC simulation for this scenario.

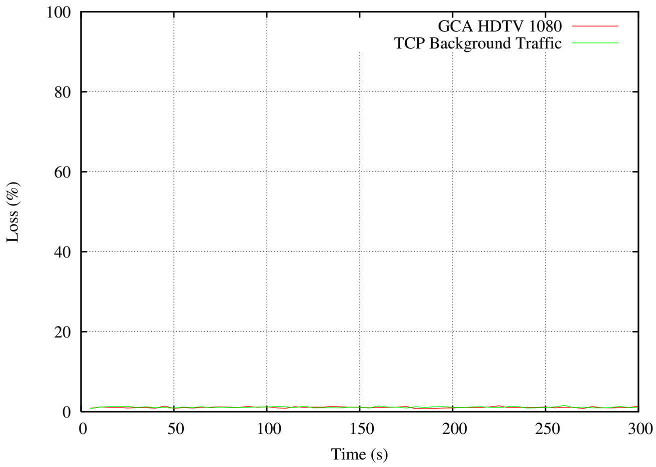
Detailed analysis and comparison of the supports this initial inspection. The α and β vales assigned to the HDTV have enabled it to compete more aggressively with the FTP file transfer. This aggressiveness has allowed it to achieved its required bandwidth to obtain very high QoE. The HDTV 1080 acquired *9.4 Mbps* of throughput while the FTP source acquires *4.5 Mbps* of throughput. When compared with the TFRC simulation, this represents an *18 %* increase in throughput for the HDTV 1080 and an *18 %* decrease in throughput for the FTP source. For the HDTV 1080 this translates into near perfect application level QoE as can be seen from the PSNR measurement of *70 dB*. This PSNR measurement also represents a increase when compared with the TFRC simulation. Although the FTP traffic source has experienced a decrease in throughput, this has very little if any impact on the best effort, non time critical traffic it carries. It should also be noted that other QoS metrics remain low and compare well with previous simulations.

Simulation B

This simulation consists of a HDTV 1080, HDTV 720 and a FTP background traffic source. The simulation begins a $t = 0.0s$, while these devices begin their respective tasks at $t = 1.0s$, $t = 2.0s$ and $t = 3.0s$ receptively. An analysis of the results is illustrated in Figures 5.14 and summarised in detail



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.13: Illustration of GCA simulation consisting of a HDTV 1080 and FTP background traffic source

	HDTV 1080	FTP
Throughput (kbps)	9,458.97	4,544.82
Loss (%)	1.06	1.11
Delay (ms)	20.90	21.07
Jitter (ms)r	0.3740	0.5377
PSNR (dB)	70.16	-

Table 5.13: GCA streaming analysis summary for a HDTV 1080 media device together with an FTP background traffic source

in Table 5.14.

Preliminary inspection of the results for this simulation also indicate performance improvements over both the non-adaptive and TFRC based simulations. It is also clear that good service differentiation is achieved between each of the clients. The aggressiveness assigned to the HDTV 1080 and HDTV 720 has resulted in the FTP source receiving a considerably smaller share of available bandwidth. Both the HDTV 1080 and the HDTV are receiving *8.8 Mbps* and *5.8 Mbps* respectively, which represents relatively fair sharing of available bandwidth based on their requirements. From an application layer perspective, both HDTVs are still achieving relatively high PSNR scores of *57 dB* and *55 dB* respectively.

These results again show a significant improvement over TFRC and slightly lower performance than non-adaptive simulations. The HDTV 1080 and HDTV 720 in the GCA based simulation are achieving approximately *55 %* and *10 %* increase in throughput when compared with the TFRC, while the FTP source is receiving 60% less throughput. GCAs ability to create application layer fairness is also apparent in this simulation. In the TFRC simulation both HDTVs were receiving approximately the same throughput, which translated into poor application layer quality fairness. In the GCA simulation both HDTVs are receiving throughput proportional to their requirements. This results in near equal quality distribution at the application layer.

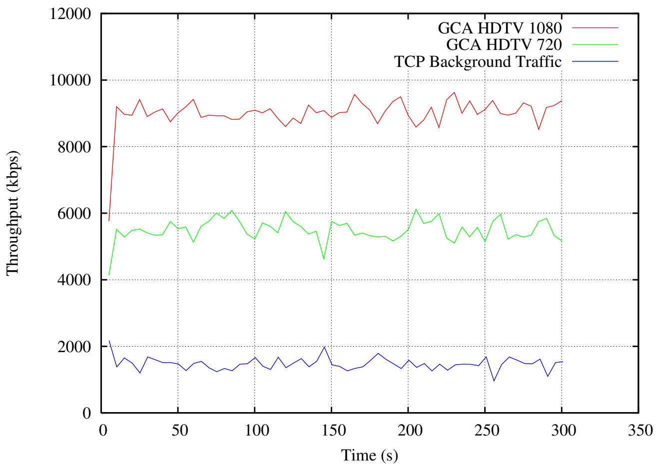
	HDTV 1080	HDTV 720	FTP
Throughput (kbps)	8,814.90	5,846.89	1,476.92
Loss (%)	2.22	2.41	5.75
Delay (ms)	22.15	22.16	22.38
Jitter (ms)r	0.2917	0.3605	0.7979
PSNR (dB)	57.38	55.31	

Table 5.14: GCA streaming analysis summary for a HDTV 1080 media device together with an FTP background traffic source

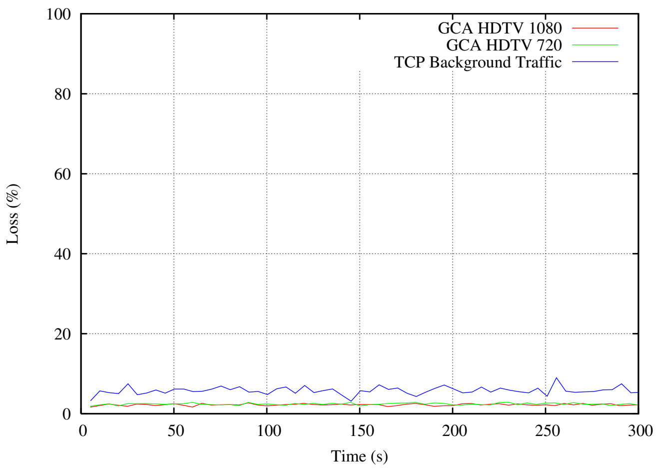
Simulation C

This simulation consists of a HDTV 1080, HDTV 720, HDTV 480 and a FTP background traffic source. The simulation begins a $t = 0.0s$, while these devices begin their respective tasks at $t = 1.0s$, $t = 2.0s$, $t = 3.0s$ and $t = 4.0s$ receptively. An analysis of the results is illustrated in Figures 5.15 and summarised in detail in Table 5.15.

This simulation provides further evidence of how the proposed GCA streaming solution can provide proportional fairness based on device requirements while preventing starvation of TCP background traffic streams. The HDTV 1080, HDTV 720 and HDTV 480 are receiving *7.8 Mbps*, *5.5 Mbps* and *1.8 Mbps* respectively. This represents approximately *81 %*, *84 %* and *72 %* of their actual required throughput. This compares very well with the TFRC simulation in Section 5.3.2, where the HDTV 1080, HDTV 720 and HDTV 480 devices obtained *4.8 Mbps*, *4.7 Mbps* and *2.5 Mbps* respectively representing *49 %*, *72 %* and *96 %* of their required throughput. GCAs proportional distribution of resources based on the devices requirements translates into an even distribution of quality among each of the device (as can be see from the PSNR measurements). This leads an increase in the overall QoE for all users in the system. Although GCA is not achieving the same



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.14: Illustration of GCA simulation consisting of a HDTV 1080, HDTV 720 and FTP background traffic source

throughput for these devices compared with the non-adaptive solution in section 5.3.1 , it is achieving lower loss and is not starving the background traffic of network resources.

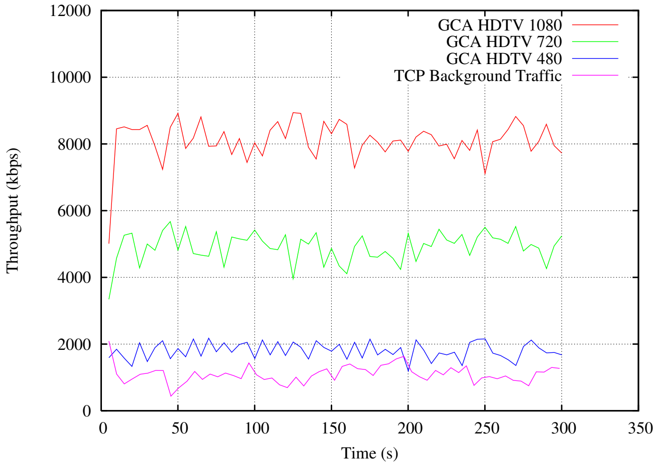
	HDTV 1080	HDTV 720	HDTV 480	FTP
Throughput (kbps)	7,817.70	5,451.47	1,832.89	1,096.28
Loss (%)	2.59	2.86	2.92	7.09
Delay (ms)	22.90	22.89	22.81	23.08
Jitter (ms)r	0.3356	0.3896	0.6379	23.07
PSNR (dB)	52.37	48.95	49.56	-

Table 5.15: GCA streaming analysis summary for a HDTV 1080, HDTV 720 and HDTV 480 media device together with an FTP background traffic source

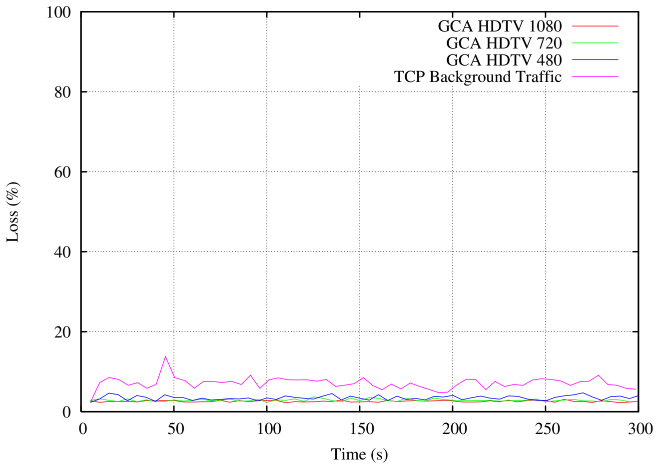
Simulation D

Like the non-adaptive and TFRC based streaming solutions the final GCA simulation consists of a HDTV 1080, HDTV 720, HDTV 480, a second HDTV 1080 and a FTP background traffic source. These devices begin their respective tasks at $t = 1.0s$, $t = 2.0s$, $t = 3.0s$, $t = 4.0s$ and $t = 5.0s$ receptively. An analysis of the results is illustrated in Figures 5.16 and summarised in detail in Table 5.16.

This simulation is designed to demonstrate the scalability of GCA. This is where GCA has the advantage over non-adaptive simulations which nearly matched GCA in performance for the previous scenarios. TFRC has the ability to scale but not the ability to provide application layer proportional fairness. The addition of the extra HDTV 1080 increase competition in the simulation. In the non-adaptive simulation in Section 5.3.1 the required bandwidth exceeds the available bandwidth resulting in lost packets. This is caused by the lack of congestion avoidance mechanism. GCA on the other hand is able to scale and avoid this congestion. In the GCA simulation, the



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.15: Illustration of GCA simulation consisting of a HDTV 1080, HDTV 720, HDTV 480 and FTP background traffic source

HDTV 1080, HDTV 720, HDTV 480 and second HDTV 1080 are receiving *5.5 Mbps*, *4.0 Mbps*, *0.9 Mbps* and *5.2 Mbps* respectively. This represents approximately *56 %*, *62 %*, *40 %* and *53 %* of their required throughput to achieve perfect quality. Although the throughput for each of these devices is substantially lower then the required throughput the PSNR shows that each device is still receiving adequate quality and near equal quality. More importantly the loss measurements remain low, in contrast to those obtained for the non-adaptive simulation.

	HDTV 1080	HDTV 720	HDTV 480	HDTV 1080	FTP
Throughput (kbps)	5,436.33	4,013.48	847.54	5,126.48	549.78
Loss (%)	4.03	4.53	5.74	4.14	11.08
Delay (ms)	23.53	23.59	23.63	23.70	21.29
Jitter (ms)	0.3761	0.4078	0.66	0.3832	0.71
PSNR (dB)	49.54	46.76	39.23	48.92	-

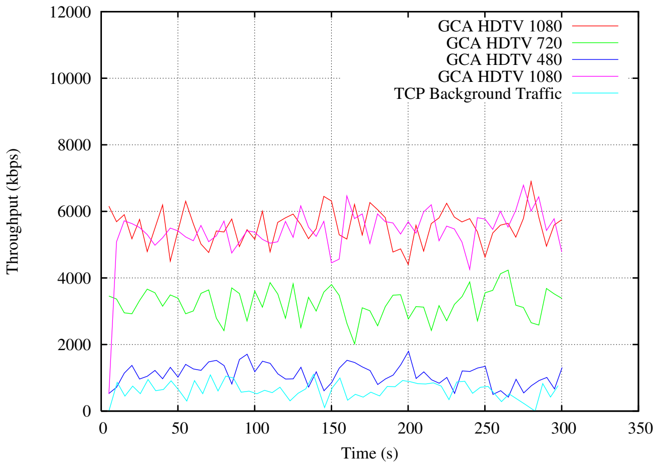
Table 5.16: GCA streaming analysis summary for two HDTV 1080's, a HDTV 720 and a HDTV 480 media device together with an FTP background traffic source

Summary

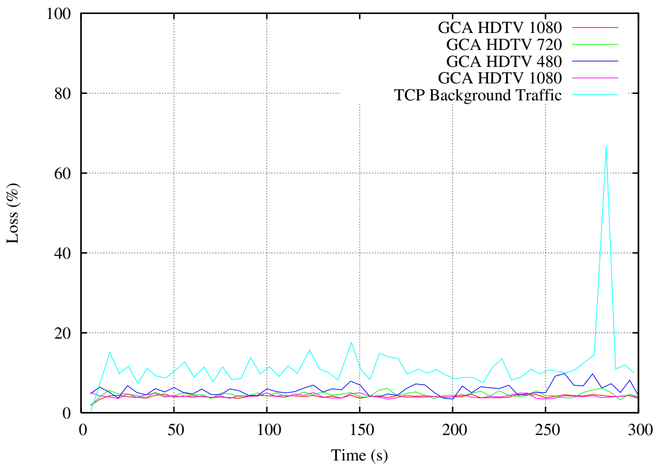
This section presented the results for the GCA based streaming simulations. The performance of the GCA streaming solution was compared with the non-adaptive and TFRC based streaming solutions and benefits were highlighted. GCA showed it had the ability to scale and maintain proportional fairness between media flows while preventing starvation of background traffic flows.

5.3.4 Summary

This section presented a conclusive simulation analysis of the GCA streaming solution. Non-adaptive and TFRC streaming solutions were used to highlight the benefits of the proposed solution. These benefits are summarised in



(a) Throughput Analysis



(b) Loss Analysis

Figure 5.16: Illustration of GCA simulation consisting of a HDTV 1080, HDTV 720, HDTV 480, a second HDTV 1080 and an FTP background traffic source

Table 5.17. GCA had similar performance to the non-adaptive solution for the multimedia devices for a number of the scenarios. However, GCA out performed it in terms of its ability to scale and prevent starvation of TCP background traffic sources. TFRC on the other hand had the ability to scale with GCA but was unable to provide the proportional fairness required to eliminate the inconsistency between in application layer quality fairness.

	Non-Adaptive	TFRC	GCA
Scalable	No	Yes	Yes
Low Loss	No	Yes	Yes
Proportional Fairness	Yes/No	No	Yes
No TCP Starvation	No	Yes	Yes

Table 5.17: Summary of key results for Non-Adaptive, TFRC and GCA streaming solutions

Chapter 6

Conclusion and Future Work

This chapter presents a summary of the work presented in this thesis. Possible future directions for this work are also presented.

6.1 Conclusion

In recent years there has been significant development in the area of multimedia streaming. This research has focused on various aspects related to streaming media to multimedia enabled devices. Numerous application layer and transport layer rate / congestion control schemes have been developed for multimedia streaming. These solutions are employed in applications that stream media on a per stream basis and do not consider their effect on other streams. This is not a problem if all streams are transmitted to devices that have equal characteristics, however this is rarely the case. In a residential environment the diversity of the characteristics of multimedia enabled devices leads to an inconsistency to occur between the application layer and transport layer mechanisms. Devices will not receive proportional share of available resources based on their requirements because the transport layer rate control mechanism is designed to distribute bandwidth evenly among streams. This thesis focused on this inconsistency in the multimedia streaming process. In particular this work was focused on proposing a mechanism to allow the transport layer rate / congestion control mechanism to distribute bandwidth among multimedia device based on their characteristic requirements.

After describing the motivation behind the work in this thesis, Chapter 2 presented background details relating to some of the protocols used in a wireless multimedia streaming process. Chapter 3 examined literature related to various aspects pertinent to the proposed solution. It examined various end-to-end and network centric approaches to streaming media. Each of the proposed solutions were described and their advantages and disadvantages were highlighted. It also presented an overview of video quality assessment techniques.

Chapter 4 presents details of the newly proposed Greediness Control Algorithm (GCA). It introduces the architecture and gives detailed information about the operation of the protocol. GCA is an equation based congestion control mechanism for streaming multimedia content in wireless networks. It enhances the IETF standardised TFRC congestion control mechanism, providing TCP compatible fairness. This allows GCA to be tailored specifically to the characteristics of the multimedia stream being carried.

This is followed by Chapter 5, which presents a detailed analysis and simulation of the proposed GCA. GCA is simulated using the NS2 simulator and compared with non-adaptive and TFRC based streaming solutions. The results of these simulations prove to be conclusive. GCA outperformed both non-adaptive and TFRC streaming scenarios. When compared with the non-adaptive solution, GCA had the ability to scale and prevent TCP starvation, while neither the non-adaptive or TFRC streaming solutions were able to provide the proportional fairness required to eliminate the inconsistency between in application layer quality fairness.

6.2 Contributions

The principal contribution of this thesis is the Greediness Control Algorithm (GCA). GCA is designed to solve an inconsistency, related to the fact that the application layer quality is not distributed proportionally between devices due to their varying requirements. The proposed GCA is an end-to-end solution for this problem. It can be deployed on any network and provide required level of service differentiation between streams without any modification to network infrastructure. The results of the tests performed show that the proposed GCA outperforms other non-adaptive and TFRC based streaming solutions. It enables the required proportional distribution bandwidth based on the devices requirements, thus increasing the QoE experienced by the user. It also outperformed it in terms of its ability to scale and prevent starvation of TCP background traffic sources. IEEE 802.11e does not have the granularity to fix this inconsistency because all video traffic would occupy the same AC. This video traffic would be separated from other types of traffic but not from other multimedia enabled devices that cause the problem. GCA increases the overall QoE of users in the network by removing this inconsistency.

6.3 Future Work

The work presented in this thesis focussed on the distribution of content within a residential environment between an in home media server and various multimedia enabled devices. It dealt specifically with the inconstancy in the distribution of quality between device with different characteristics. In order to solve this problem the streaming solution was required to employ a discovery mechanism. This mechanism discovered the characteristics of media enabled devices within the network and assigned them appropriate values of α and β based on their characteristics. At present this discovery is performed manually as network was simulated. However, in order for this solution to be deployed in a real environment this discovery mechanisms would have to incorporate some sort of utility function that automatically mapped device characteristics into specific values of α and β. This could possibly be built on-top of existing discovery mechanisms such as the DLNAs[1] Universal Plug and Play (UPnP)[2] varient or Apple's Bonjour[3].

As mentioned above this work focussed on the delivery between a LAN based media and media enabled devices. This work could be extended to investigate the how a Wide Area Network (WAN) based media server would influence results. In this situation the bottleneck would be the broadband connection.

Another interesting direction could be to investigate the content delivery mechanism between the WAN service provider and the LAN media server. This research could focus on the delivery of content using Peer 2 Peer (P2P) such as BitTorrent[4] rather than the traditional CDNs approach such as Akami[5]. This work could focus on tuning the performance of the P2P protocol specifically for multimedia content. At present this overlay network is tuned for raw data, and does not consider the characteristics of the multimedia content.

[1] http://www.dlna.org/
[2] http://www.upnp.org/
[3] http://developer.apple.com/networking/bonjour/
[4] http://www.bittorrent.org/protocol.html
[5] http://www.akamai.com/

Acronyms

AC Access Categories

AC Access Category

ACK Acknowledgment

ACR Absolute Category Rating

AIFS Arbitrary Inter-frame Spacing

AIFSN Arbitrary Inter-frame Spacing Number

AIMD Additive Increase, Multiplicative Decrease

AP Access Point

ARF Auto Rate Fallback

ARQ Automatic Repeat Request

ARQ Adaptive Repeat Request

BA Block Acknowledgment

BBC British Broadcasting Corporation

BE Best Effort

BG Background

BSA	Basic Service Area
BSS	Basic Service Set
BSS	Basic Service Set
CAP	Controlled Access Phase
CBR	Constant Bit Rate
CCA	Clear Channel Assessment
CCID	Congestion Control ID
CDN	Content Distribution Network
CFP	Contention Free Period
CMU	Carnegie Mellon University
CONSER	Collaborative Simulation for Education and Research
CP	Contention Period
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
CW	Contention Window
CW	Contention Window
DAC	Distributed Admission Control
DARPA	Defence Advanced Research Projects Agency
DCCP	Datagram Congestion Control Protocol
DCF	Distributed Coordinator Function

DCR	Degraded Category Rating
DCT	Discrete Cosine Transform
DFS	Distributed Fair Scheduling
DiffServ	Differentiated Services
DIFS	Distributed Inter-Frame Space
DLNA	Digital Living Network Alliance
DLS	Direct Link Setup
DSL	Digital Subscriber Line
DVQ	Digital Video Quality
DWFQ	Distributed Weighted Fair Queue
ECN	Explicit Congestion Notification
EDCA	Enhanced Distributed Channel Access
EDCF	Enhanced Distributed Coordinator Function
EDD	Earliest-Due-Date First
ELN	Explicit Loss Notification
EPG	Electronic Program Guide
FIFO	First-in, First-out
FTP	File Transfer Protocol
FTP	File Transfer Protocol
GCA	Greediness Control Algorithm

GloMoSim	Global Mobile Information Systems
GOP	Group of Pictures
HC	Hybrid Coordinator
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HDD	Hard Disk Drive
HDTV	High Definition Television
HTTP	Hyper Text Terminal Protocol
HVS	Human Visual System
IBSS	Independent Basic Service Set
ICIR	ICSI Centre for Internet Research
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Standards Organisation
ISP	Internet Service Provider

ITU	International Telecommunication Union
LAN	Local Area Network
LDA	Loss Discrimination Algorithm
MAC	Media Access Control
MPDU	MAC Payload Data Unit
MPEG	Motion Pictures Expert Group
MPQM	Motion Picture Quality Metric
MSE	Mean Square Error
NACK	Negative Acknowledgement
NAV	Network Allocation Vector
NS	Network Simulator
NSF	National Science Foundation
OPNET	Optimised Network Engineering Tools
OSI	Open System Interconnect
OSPF	Open Shortest Path First
OTcl	Object-Oriented Tcl
P2P	Peer 2 Peer
PC	Point Coordinator
PCF	Point Coordinator Function
PHY	Physical

PIFS	Priority Inter-frame Spacing
PLCP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
POA	Point of Attachment
PSNR	Peak Signal to Noise Ratio
PSNR	Peak Signal to Noise Ratio
QAP	QoS - Enhanced Access Point
QBSS	QoS - Enhanced Basic Service Set
QoE	Quality of Experience
QoS	Quality of Service
QSTA	QoS - Enhanced Station
RAP	Rate Adaptation Protocol
REAL	Realistic and Large
RED	Random Early Drop
RIP	Routing Information Protocol
RSS	Really Simple Syndication
RSVP	Resource Reservation Protocol
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTS	Request to Send

RTT	Round Trip Time
SACK	Selective Acknowledgement
SAMAN	Measurement and Analysis for Networks
SCFQ	Self-Clocked Fair Queueing
SDTV	Standard Definition Television
SIFS	Short Inter-Frame Space
SMCC	Streaming Media Congestion Control
SSIM	Structural Similarity Index
STA	Station
STFQ	Start-Time Fair Queueing
TC	Traffic Categories
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
TFRC	TCP Friendly Rate Control
TFRCC	TCP Friendly Rate Control with Compensation
TxOP	Transmission Opportunity
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
VI	Video
VO	Voice

VoIP	Voice over Internet Protocol
VQEG	Video Quality Experts group
VQM	Video Quality Metric
VTP	Video Transport Protocol
WAN	Wide Area Network
WFQ	Weighted Fair Queueing
WLAN	Wireless Local Area Network

Bibliography

[1] F. Thouin and M. Coates, “Video-on-Demand Netwroks: Design Approaches and Future Challenges,” in *IEEE Network*, March / April 2007.

[2] F. Callaly and P. Corcoran, “Architecture of a PVR Appliance with 'Long-Tail' Internet-TV Capabilities,” in *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, May 2006.

[3] R. Fielding, J. Gettys, H. Mogul, J. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, “Hypertext Transfer Protocol,” IETF, RFC 2817, June 1999.

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” IETF, RFC 3550, July 2003.

[5] J. Postel, “User Datagram Protocol,” IETF, RFC 768, August 1980.

[6] ——, “Transmission Control Protocol,” IETF, RFC 793, September 1981.

[7] E. Kohler, M. Handley, and S. Floyd, “Datagram Congestion Control Protocol (DCCP),” IETF, RFC 4340, March 2006.

[8] L. Chiariglione, “Short MPEG-1 description,” ISO/IEC, Tech. Rep. JTC1/SC29/WG11 N, June 1996.

[9] ——, “Short MPEG-2 description,” ISO/IEC, Tech. Rep. JTC1/SC29/WG11 N, October 2000.

[10] R.Koenen, “Overview of the MPEG-4 Standard,” ISO/IEC, Tech. Rep. JTC1/SC29/WG11 N4668, March 2002.

[11] “Internet Protocol (IP),” IETF, RFC 791, September 1981.

[12] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” IETF, RFC 2460, December 1998.

[13] J. Moy, “OSPF Version 2,” IETF, RFC 2328, April 1998.

[14] G. Malkin, “RIP version 2,” IETF, RFC 2453, November 1998.

[15] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services,” IETF, RFC 2475, December 1998.

[16] K. Ramakrishnan, S. Floyd, and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” IETF, RFC 3168, September 2001.

[17] ——, “The Addition of Explicit Congestion Notification (ECN) to IP,” IETF, Tech. Rep., 2001.

[18] C. Krasic, K. Li, and J. Walpole, “The Case for Streaming Multimedia with TCP,” in *8th International Workshop Interactive Distributed Multimedia Systems: ,IDMS 2001, , UK, September 4-7, 2001, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2158/2001. Lancaster, UK: Springer Berlin / Heidelberg, 2001, p. 213.

[19] S. Floyd and E. Kohler, “Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control,” IETF, RFC 4341, March 2006.

[20] S. Floyd, E. Kohler, and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)," IETF, RFC 4342, March 2006.

[21] IEEE, "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification," IEEE, Tech. Rep., 2000.

[22] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. 28, pp. 425 – 432, 1980.

[23] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *IEEE Conference on Computer Communications (INFOCOM)*, 2003.

[24] IEEE, "IEEE 802.11e: Medium Access Control (MAC) Quality of Service Enhancements," IEEE, Tech. Rep., 2005.

[25] S. Mangold, S. Choi, G. Hiertz, O. Klein, and B. Walke, "Analysis of IEEE 802.11e for QoS Support in Wireless LANs," *IEEE Wireless Communications*, vol. 10, pp. 40 – 50, 2003.

[26] Y. Xiao, "IEEE 802.11n: enhancements for higher throughput in wireless LANs," in *IEEE Wireless Communications*, vol. 12, no. 6, December 2005.

[27] J. Saltzer, D. Reed, and D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, pp. 277 – 288, 1984.

[28] R. Rejaie, M. Handley, and D. Estrin, "RAP: An End-to-End Rate-Based Congestion Control Mechanism for Realtime Streams in the Internet," in *IEEE Computer and Communications Society (INFOCOM)*, vol. 3, March 1999, pp. 1337–1345.

[29] F. Menta and G. Schembra, "Efficient design of RED routers for TCP/RAP fairness optimization," *IEEE International Conference on Communications*, vol. 4, pp. 2186 – 2190, 2002.

[30] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet," IETF, RFC 2309, April 1998.

[31] N. Aboobaker, D. Chanady, M. Gerla, and M. Sanadidi, "Streaming Media Congestion Control using Bandwidth Estimation," in *IEEE International Conference on Management of Multimedia and Mobile Networks and Services*, October 2002.

[32] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Mobile Computing and Networking*, 2001, pp. 287–297.

[33] M. Handley, S. Floyd, J. Padhye, and J. Midmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," IETF, RFC 3448, 2003.

[34] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703 – 717, 2003.

[35] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," in *IEEE / ACM Transactions on Networking*, vol. 5, no. 6, December 1997.

[36] S. Floyd, "Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field," IETF, Tech. Rep., 2006.

[37] P. Zhu, W. Zeng, and C. Li, “Joint Design of Source Rate Control and QoS-Aware Congestion Control for Video Streaming OVer the Internet,” in *IEEE Transaction on Multimedia*, 2007.

[38] G. Yang, M. Gerla, and M. Sanadidi, “Adaptive video streaming in the presense of wireless errors,” in *Management Multimedia Networks & Services (MMNS)*, San Diego, October 2004.

[39] M. Chen and A. Zakhor, “Multiple TFRC Connections Based Rate Control for Wireless Networks,” in *IEEE Transactions on Multimedia*, vol. 8, no. 5, October 2006, pp. 1045 – 1061.

[40] J. Crowcroft and P. Oechslin, “Differentiated end to end internet Services using weighted fair sharing,” in *ACM computer Communications Review*, vol. 28, no. 3, July 1998.

[41] R. Braden, D. Clark, and S. Shenker, “Integrated Services in the Internet Architecture: An Overview,” IETF, RFC 1633, June 1994.

[42] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RSVP),” IETF, RFC 2205, September 1997.

[43] A. Tanenbaum, *Computer Netwroks*, 4th ed. Pearson Education, 2003.

[44] B. Davie, A. Charny, J. Bennet, K. Benson, J. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, “An Expedited Forwarding PHB (Per-Hop Behavior),” IETF, RFC 3246, March 2002.

[45] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured Forwarding PHB (Per-Hop Behavior) Group,” IETF, RFC 2597, June 1999.

[46] H. Fattah and C. Leung, “An overview of scheduling algorithms in wireless multimedia networks,” in *IEEE Wireless Communications*, vol. 9, no. 5, October 2002, p. 76 83.

[47] A. Banchs and X. Perez, "Distributed Weighted Fair Queueing," in *International Communications Conference (ICC)*, vol. 5, April 2002, pp. 3121 – 3127.

[48] G. Ye and J. Hou, "An analytical model for service differentiation in ieee 802.11," in *IEEE International Conference on Communications (ICC)*, 2003.

[49] N. Vaidya, P. Bahl, and S. Gupa, "Distributed Fair Fcheduling in a Wireless LAN," in *Annual International Conference on Mobile Computing and Networking*, August 2000, pp. 167 – 178.

[50] S. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *IEEE Infocom*, 1994.

[51] M.-H. Lu, P. Steenkiste, and T. Chen, "Video Streaming Over 802.11 WLAN with content-Aware Adaptive Retry Limit," in *IEEE International Conference on Multimedia & Expo*, July 2005.

[52] M. Ahmed, "Call Admission Control in Wireless Networks: A comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 7, pp. 50 – 69, 2005.

[53] Y. Xiao and H. Li, "Evaluation of Distributed Admission Control for the IEEE 802.11e EDCA," in *IEEE Communication Magazine*, vol. 42, no. 9, 2004, p. S20 S24.

[54] M. Barry, A. Campbell, and A. Veres, "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks," in *IEEE INFOCOM*, vol. 1, 2001, p. 582 90.

[55] P. Koutsakis, M. Paterakis, and S. Psychis, "Call Admission Control and Traffic Policing Mechanisms for the Wireless Transmission of Layered Videoconference Traffic from MPEG-4 and H.263 Video Coders," in

IEEE international Symposium of Personal Indoor and Mobile Radio Communication, vol. 5, September 2002, p. 2155 2159.

[56] I. T. Union, "Objective perceptual assesment of video quality: Full reference televicion," International Telecommunications Union, Tech. Rep., 2004.

[57] D. Miras, "Network QoS needs of advanced Internet applications: A Survey," in *Internet2 QoS Working Group*, 2002.

[58] C. Lambrecht and O. Verscheure, "Perceptual Quality Measure using spatio-temporal model of the human visual system," in *SPIE*, 1996.

[59] Z. Wang and A. Bovik, "A Universal Video Quality Index," in *IEEE Signal Processing Letters*, 2002.

[60] A. Watson, J. Hu, and J. McGowan, "DVQ: A digital video quality metric based on human vision," in *Journal of Electronic Imaging*, 2001.

[61] ITU, "Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference," ITU, ITU-T Recommendation J.144, 2004.

[62] ——, "Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference," ITU, ITU-R Recommendation BT.1683, 2004.

[63] ——, "Subjective video quality assessment methods for multimedia applications," International Telecommunication Union, ITU-T Recommendation P.910, 1999.

[64] ——, "Methodology for the subjective assessment of the quality of television pictures," International Telecommunication Union, ITU-R Recommendation BT.500-11, 2002.

[65] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP reno performance: a simple model and its empirical validation,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 133 – 145, 2000.

[66] A. Kuzmanovic and E. Knightly, “A performance vs. trust perspective in the design of end-point congestion control protocols,” *International Conference on Network Protocols (ICNP)*, vol. 1, pp. 96 – 107, 2004.

[67] J. Heidemann, N. Bulusu, and J. Elson, “Effects of Detail inWireless Network Simulation,” in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2001.

[68] D. Kotz, C. Newport, R. Gray, J. liu, Y. Yuan, and C. Elliott, “Experimental evaluation of wireless simulation assumptions,” in *International Symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004.

[69] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, “Advances in network simulation,” in *IEEE Computer*, 2000.

[70] G. Muntean and N. Cranley, “Resource Efficient Quality-Oriented Wireless Broadcasting of Adaptive Multimedia Content,” *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 362–368, March 2007.

[71] J. Klaue, B. Rathke, and A. Wolisz, “EvalVid - A Framework for Video Transmission and Quality Evaluation,” in *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, September 2003, pp. 255–272.