Per-Session Weighted Fair Scheduling for Real Time Multimedia in Multi-Rate Wireless Local Area Networks

by

YASER POURMOHAMMADI FALLAH

B.Sc., Sharif University of Technology, 1998
M.A.Sc., University of British Columbia, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

March 2007

Abstract

Supporting real time multimedia applications is of utmost importance for future wireless data networks. In particular, it is crucial to support such applications in widely deployed and fast growing wireless local area networks (WLAN) that are based on IEEE 802.11 standard. However, achieving this goal requires features and mechanisms that have not been offered by the original IEEE 802.11 standard. To address this issue, several Quality of Service (QoS) enabling features have been added to the Medium Access Control (MAC) layer in the new IEEE 802.11e standard. Nevertheless, the new standard does not mandate a specific QoS solution, and intentionally leaves such task to developers and equipment vendors. Devising mechanisms that can efficiently provide the required QoS in WLANs has proved to be a challenging task. This is mainly due to the fact that WLANs such as 802.11e are based on the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) access method, which is inherently a distributed mechanism with random uplink or downlink access. Moreover, the physical layer of a WLAN allows each station to use a different transmission rate. The transmission rate could also dynamically change from one packet to the next, for the same station. The general problem of packet loss in wireless networks is also present in WLANs.

The existing solutions and the prioritized contention based mechanism provided by the standard are found to be inadequate for providing the required services, especially in heavily loaded networks. Considering the issues mentioned above, we present a solution in this thesis that employs the *controlled access* mechanisms of the 802.11e standard to provide per-session guaranteed QoS to multimedia sessions. We introduce a framework that centralizes the task of scheduling uplink and downlink flows in the access point through the concept of *virtual packets*. We propose a new queuing structure that works with a fair generalized processor sharing based scheduler, integrated with a traffic shaper, for scheduling controlled (polling) and contention access durations. To address the issues of physical channel impairment and variable rate operation of a WLAN, we extend our scheduling framework to provide throughput or temporal fairness in multirate WLANs. Through analysis and experiments, we demonstrate that our solution provides guaranteed fair access for multimedia sessions over WLANs.

Table of Contents

List of Tables

List of Figures

Dedication

To my wife, Mahsa, and my parents, Ziba and Ali

Co-Authorship Statement

Chapter 2 of this dissertation has been taken from a paper co-authored by Yaser Pourmohammadi Fallah (the author of this thesis), Samer El-Housseini and Hussein Alnuweiri. As a research group, we have been working on modeling IEEE 802.11 and 802.11e WLANs, and this was a common area where we shared our knowledge and work. The research and model analysis have been done in long group sessions on a round table. The greater part of the paper was written by Yaser who contributed to identification and design of the research work presented in the chapter, and developed the mathematical model presented therein; while Samer implemented the mathematical model in C, generated the simulation data and plots, and finished writing the last part of the paper. Prof. Alnuweiri has supervised the research work and contributed in directing and refining the work, as well as revising the manuscript.

Chapters 3 and 4 of this dissertation are co-authored by the author of this thesis, Yaser Pourmohammadi Fallah, and his PhD research supervisor, Prof. Hussein Alnuweiri. Yaser, the author of this thesis, is the primary author of these chapters and has contributed to identification and design of the research work presented in the chapters, as well as performing the presented research, data analysis and manuscript preparation. Prof. Alnuweiri has supervised the research work and contributed in directing and refining the work, as well as revising the manuscripts.

Chapter 1. Introduction and Background

1.1 Introduction and Motivation

Research towards new technologies is driven by the desire to improve the quality of life. Advancements in telecommunication technology in the past few decades certainly account for some of the most important instances of such improvements. The latest research efforts in this field have been focused on enabling true multimedia communication services, anywhere, anytime. To achieve this goal, many new communication technologies have been developed to provide higher communication speeds and better services to users. One of these technologies that provides broadband wireless access is the IEEE 802.11 standard [1] for wireless local area networks (WLAN).

Due to their simplicity and ease of use, WLANs are being deployed at a rapid pace around the world, bringing high speed access to many users. As a result, a diverse range of applications, including real-time multimedia applications, are expected to be used over WLANs. Examples of such applications are voice and video telephony and conferencing, video surveillance, streaming High Definition TV (HDTV) in a home environment or possibly in neighborhoods, as well as real-time online gaming and tele-robotics. All these applications have stringent quality of service (QoS) requirements such as low delay and jitter and guaranteed bitrate services.

Multimedia applications are also the driving force behind many new developments in other field of communications. For example, new video streaming applications are gradually appearing on cellular networks, forcing service providers to move to newer technologies and solutions. One solution for providing better broadband access to cellular mobile users is internetworking between high speed WLANs and cellular networks. The idea is to use the existing high speed WLAN access, wherever service is available, and switch some of the traffic from cellular networks to the local WLANs, while maintaining the main management and signaling functions within the core of the cellular network. Such mechanisms have received considerable attention from the standard bodies; an example of the specifications of such internetworking scenarios is found in [3]. Given the above scenario, it is imperative for WLANs to be able to provide the same QoS that cellular networks can, at a higher cost, offer to applications such as voice or video telephony.

An example of typical QoS requirements or performance targets in 3rd Generation cellular networks is shown in Table 1-1. Note that this table presents the end-to-end requirements; if a wireless network is only one part of the traffic path, stricter QoS requirements are applied. Therefore, in case of WLANs, we expect similar or more stringent QoS requirements than those described in Table 1-1. In fact, with more bandwidth availability in WLANs, high bitrate applications such as high quality video (e.g., with rates in the range of 1-24 Mbps) are also considered; whereas for 3G networks a rate of 384Kbps was considered to be the limit. This increased throughput requirements makes the already stringent end-to-end performance targets even stricter.

Table 1-1 3GPP End-user Performance Expectations - Conversational / Real-time Services (Modified from source: 3GPP TS 22.105 V8.0.0)

Medium	Application	Degree of symmetry	Data rate	Key performance parameters and target values	
				End-to-end One-way Delay	Information loss
Audio	Conversational voice	Two-way	4-25 kb/s	<150 msec preferred	< 3% FER
Video	Videophone	Two-way	32-384 kb/s	< 150 msec preferred Lip-synch : < 100 msec	< 1% FER
Data	Telemetry - two-way control	Two-way	<28.8 kb/s	< 250 msec	Zero
Data	realtime games	Two-way	< 60 kb/s	< 75 msec preferred	< 3% FER preferred, < 5% FER limit
Data	Telnet	Two-way (asymmetric)	< 1 KB	< 250 msec	Zero

The original 802.11 standard has not been designed for real-time services and multimedia applications; as a result, it fails to efficiently provide the required services for real time applications. To address this issue, a new standard, IEEE 802.11e, has been approved for enhancing the medium access control (MAC) layer of the original standard with Quality of Service (QoS) features. These features are presented through a Hybrid Coordination Function (HCF) and support two new access mechanisms of Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA). Such capabilities provide enough features for developers to devise QoS solutions that make real time multimedia applications in WLANs feasible.

The QoS features of the HCCA can be used to provide per-session guaranteed services to real time applications. However the standard does not mandate any scheduling solution to use these features and leaves it to developers to devise such a scheme. Devising such a mechanism is a complicated task due to the fact that WLANs are by nature distributed multiple access networks, and the physical layer in wireless networks is more error prone and may operate at different speeds at different times (also known as *multirate* operation). Addressing these

issues, we propose a scheduling solution in this thesis that provides per-session guaranteed services for multimedia applications in WLANs. Our solution, which is called Controlled Access Phase Scheduling (CAPS), addresses the variable rate operation of wireless networks and is applicable to any multiple access network with features similar to the 802.11e standard.

The proposed design in this dissertation is evaluated through mathematical analysis as well as simulation experiments. Since EDCA is the first choice of most vendors for QoS provisioning in WLANs, we present an extensive comparison between the performances of our method with that of EDCA. We also present a new analysis of the EDCA throughput under saturation condition in order to provide more insight into the operation of EDCA. This analysis complements and corrects the existing analyses of EDCA. We show through our experiments and analyses that although EDCA may provide adequate services in lightly loaded networks, it fails to provide the required QoS when the network is heavily loaded. Our method, on the other hand, significantly boosts the capacity and the provided QoS.

This thesis provides a detailed description of our design. Some prominent contributions of this thesis are described in the next section. In this chapter we also present an overview of the IEEE 802.11e technology and its specific features that concern QoS provisioning for multimedia applications. We review some related works that address specific aspects of QoS provisioning in wireless networks and identify the unresolved issues that the current solutions do not address. A guide to the organization of this thesis is presented in the last section of this chapter.

1.2 Thesis Contributions

This thesis proposes a QoS scheduling framework for 802.11e WLANs. The proposed method provides per-session fair services for multimedia sessions that make reservations with the access point, while sharing the remaining capacity of the WLAN using EDCA access method. Our solution takes into account the fact that WLAN stations may use different transmission rates in the physical layer, and proposes measures to maintain the fairness of the scheduling mechanism. The main contributions of this thesis, made through the course of developing the above solution, are the following:

1. Introducing a unified scheduling framework that centralizes the task of uplink/downlink scheduling in the access point of a WLAN.

2. Developing a new queuing/scheduling model that uses traffic shaping and fair scheduling in the above unified scheduling framework, and achieves efficient scheduling of HCCA and EDCA based access. This model provides guaranteed access services for HCCA flows, while sharing the remaining capacity in a contention based manner using EDCA.

3. Presenting an analysis of temporal and throughput fair scheduling based on the concept of Generalized Processor Sharing (GPS).

4. Extending the CAPS solution to maintain throughput or temporal fairness in multirate environments, and deriving the delay bounds provided by this algorithm under dynamic and static multirate scenarios.

5. Introducing a generalized saturation throughput analysis of EDCA that allows for an arbitrary number of priority levels to be considered, extending and correcting the existing methods.

6. Implementing the 802.11e MAC and the proposed scheduling framework in OPNET simulation environment.

1.3 Background: IEEE 802.11e Wireless Local Area Networks

In 1997 the IEEE adopted the 802.11 wireless local area network (WLAN) standard [1]. This standard defines the media access control (MAC) and physical (PHY) layers for a local area network (LAN) with wireless connectivity. It addresses local area networking where the connected devices communicate over the air with other devices that are within close proximity to each other. The standard identifies several main components for a WLAN architecture. These are Station (STA), Access Point (AP), Basic Service Set (BSS), Independent Basic Service Set (IBSS), Distribution System (DS), and Extended Service Set (ESS). BSS is a set of stations controlled by a single coordination function, i.e. in a single MAC domain. An IBSS is an Ad Hoc wireless network in which stations communicate in a peer to peer fashion. A DS is a system used to interconnect a set of BSSs and integrated local area networks (LANs) to create an extended service set (ESS). These concepts are illustrated in Figure 1-1.

WLANs are usually deployed in an infrastructure mode, in which an access point controls the network and provides a gateway to the outside world. The other mode of operation for WLANs is the Ad Hoc mode in which the network is managed in a distributed manner. Figure 1-1 depicts the typical topology of the network in each mode. A great deal of research has been

dedicated to Ad Hoc networks, while research on infrastructure based WLANs have been less extensive. In this thesis we focus on the infrastructure mode, which is preferred by service providers and is the widely used mode; however, some of the work of this thesis, especially the analysis in Chapter 2, is applicable to Ad Hoc networks as well.

The original standard has been amended with various enhancements; namely 802.11b (DSSS up to 11Mbps), 802.11a (OFDM, up to 54 Mbps), and 802.11g (backward compatible with 802.11b, with speed of up to 54 Mbps) standards were approved. These standards are only different in their physical layer and data rates; the MAC layer operation for all of them is the same, except for some PHY related parameters. The MAC layer has recently been amended with features to support QoS and increase efficiency as well. These features are approved under the IEEE 802.11e standard [2]. Further enhancements to the PHY and MAC layers are being considered in the upcoming 802.11n standard.

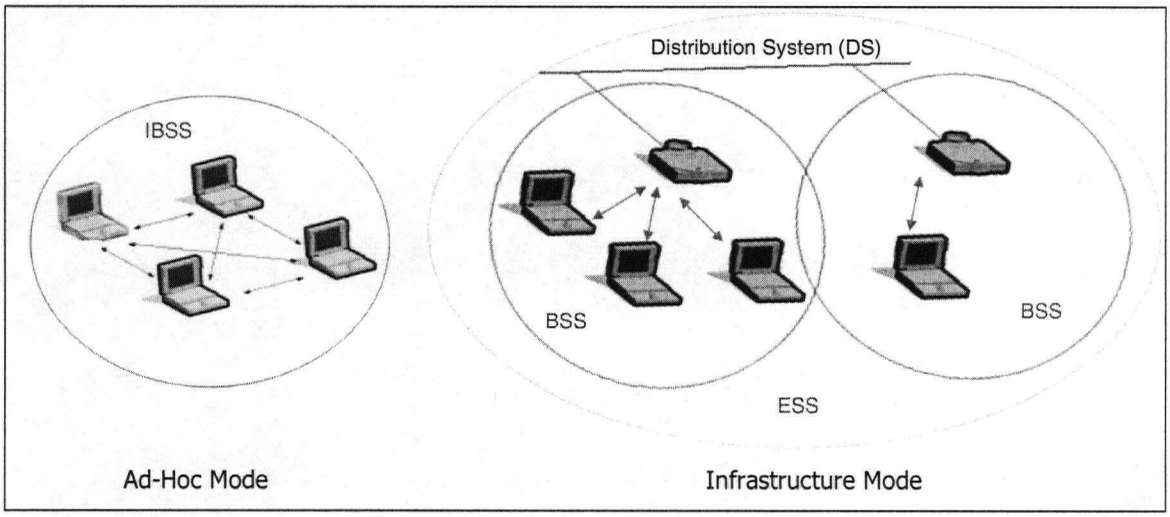

Figure 1-1 Different WLAN topologies: Ad Hoc vs. Infrastructure

In addition to 802.11, other Wireless LAN standards have been developed. For example HIPERLAN (and HIPERLAN2) standard has been developed by European

Telecommunications Standards Institute (ETSI) [4][5]. Although, the HIPERLAN standard seems to be more capable than 802.11 in terms of Quality of Service provisioning, it has not gained the widespread acceptance that 802.11 has. Therefore, we do not focus our research on this standard.

In this section we present an overview of the MAC layer functionality of the original 802.11 standard. We then review the enhancements that were introduced in the new 802.11e standard. In particular, we highlight the controlled access features that are used by our proposed QoS solution.

1.3.1 IEEE 802.11 MAC Layer

The basis for 802.11 MAC is a CSMA/CA mechanism (Carrier Sense Multiple Access with Collision Avoidance). Carrier sensing is done through physical sensing of the radio frequency (RF) carrier as well as a virtual carrier sensing in the MAC itself. Virtual carrier sensing is done through maintaining a Network Allocation Vector (NAV) signal that determines for how long the channel remains busy. NAV is set in a "duration" field of the MAC messages. Collision avoidance in 802.11 MAC is performed buy a mechanism called Distributed Coordination Function (DCF).

DCF is a protocol that describes how stations can access the wireless medium in a distributed manner. DCF uses inter-frame space (IFS) time intervals to coordinate channel access in the contention period (CP). Each frame type is allowed to access the network if it finds the medium idle for longer than a predetermined IFS time. The IFS time is different for each type of frame, Data frames have to use DIFS time, while ACK (Acknowledgment), CTS (Clear To Send), and Poll messages use SIFS time which is shorter than DIFS. This gives

ACK, Poll, or CTS messages higher medium access priority than the data frames. Stations that find the medium busy perform a random back-off and repeat the deference and channel sensing procedure again. The details of DCF can be found in [1]. Figure 1-2 shows the timing relationship in DCF.

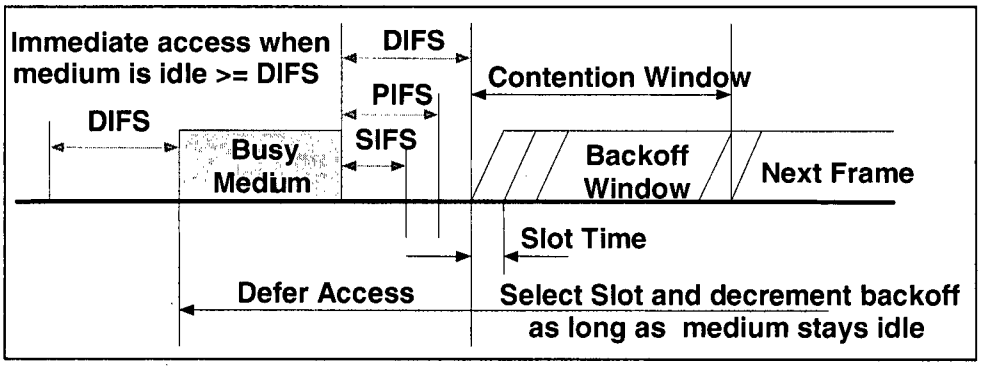

Figure 1-2 IFS Timing Relationships in DCF

The random backoff is performed by selecting a random number from a contention window of (0,CW). CW is the contention window size and is initially set to aCWmin. When a station transmits a data frame or RTS but does not receive an expected Ack or CTS, it assumes that a collision happened and doubles its contention window size, up to aCWmax. This increase continues for a few times, limited by the retry_limit parameter. The packet is dropped if this limit is reached. The backoff counter is reduced in each idle time slot observed by the station. If a transmission occurs before the counter reaches zero, the counter freezes for the duration of the busy period and resumes its count down with the next observed idle slot (after DIFS). This procedure is depicted in Figure 1-3. When the backoff counter reaches zero the station immediately transmits the frame.

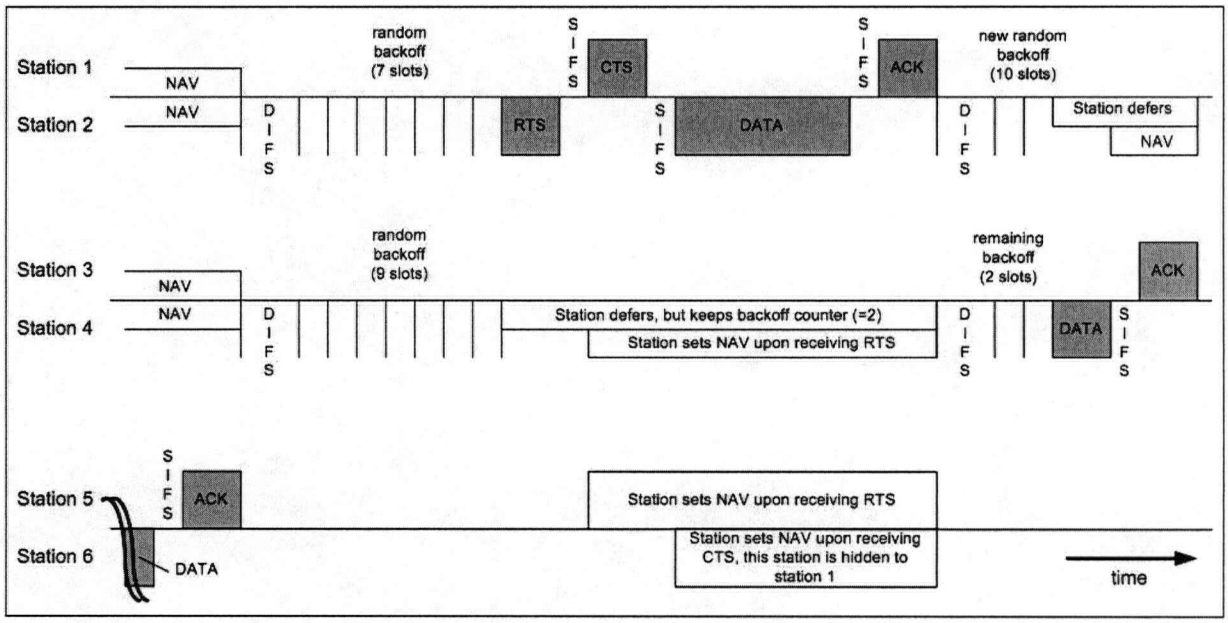

Figure 1-3 DCF Operation.

While DCF is a mandatory function in MAC, an optional mechanism, called Point Coordination Function (PCF), has also been defined in the standard. PCF resides in the access point and provides a contention free access method by polling individual stations whenever it wants to send to or receive data from them. Contention free periods, controlled by PCF, are repeated periodically (Figure 1-4). When the WLAN is controlled by an access point (AP), a beacon is also sent on a periodic basis. If PCF is supported, this beacon becomes the signal that indicates the start and end of a contention free period (CFP). The beacon sets the NAV parameter in all stations to the length of the CF period. To give the access point the priority in sending a beacon, the DCF specifies a PIFS (PCF Inter Frame Space) duration which is shorter than DIFS, but longer than SIFS. Using PIFS the access point can interrupt the normal contention, which uses the longer DIFS waiting time, and send a beacon immediately following any data exchange cycle. Beacon cannot interrupts Ack or CTS packets since these packets use the shorter SIFS.

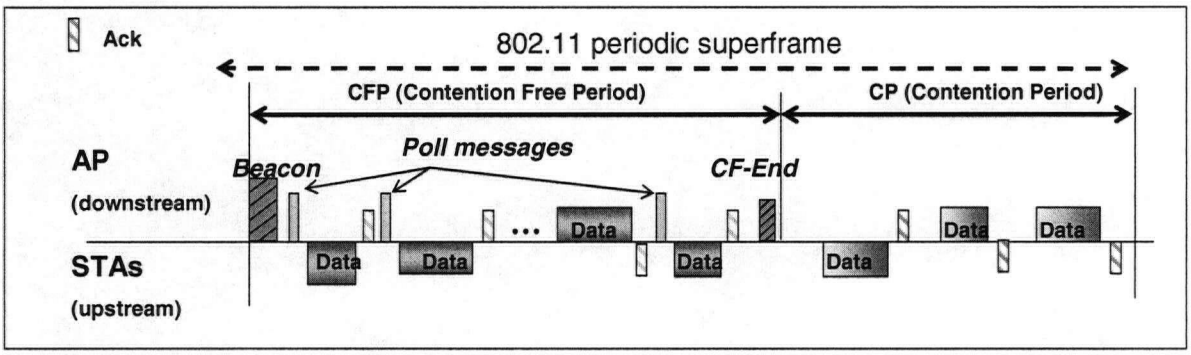

Figure 1-4 Periodic Contention and Contention Free Periods, PCF Operation.

1.3.2 IEEE 802.11e MAC Layer

The MAC in the original 802.11 standard (and a, b, and g variations), does not consider QoS and does not provide necessary mechanisms for properly supporting multimedia applications. Mechanisms for prioritizing between different classes of traffic and reserving guaranteed resources for flows were not present in the original 802.11. For this reason IEEE approved an amendment to the MAC layer specification, under the 802.11e standard, that provides necessary mechanisms for proper QoS support in the MAC layer [2].

The new standard introduces some concepts such as transmission opportunity (TXOP) and the capability to poll a station even during the contention period. There are also new frame formats with QoS information fields that help in enabling QoS in the MAC layer. The basic building block of MAC in 802.11e is again DCF, with some modifications. Accessing the medium is now controlled by a Hybrid Coordination Function (HCF) that works on top of DCF. Channel access in HCF is done using two protocols called EDCA (Enhanced Distributed Channel Access), and HCCA (HCF Controlled Channel Access). EDCA provides prioritized aggregate QoS through defining four traffic categories and using different contention parameters for each category. HCCA provides per session QoS through polling mechanisms and user defined scheduling and queuing schemes.

EDCA is in fact an enhanced version of DCF that allows different classes of traffic to use different contention and timing parameters, providing them with probabilistic differentiated services. The 802.11e standard defines 8 different traffic priorities in 4 access categories for EDCA. The DCF is enhanced by introducing an access category specific IFS waiting time, known as Arbitration IFS or AIFS, that replaces DIFS. Each access category or priority level *i* can use a different AIFS[i]. Control frames still use the same SIFS and PIFS waiting times that are shorter than AIFS. Figure 1-5 shows the timing relationship in 802.11e DCF and EDCA. The back-off windows are also different for different traffic priorities, so that each class *i* uses its specific aCWmin[i], and aCWmax[i]. Access categories with shorter AIFS and smaller contention window limits will have higher probabilities of transmission and will receive higher throughput and priority over other categories. This prioritization enables a relative QoS in 802.11e MAC.

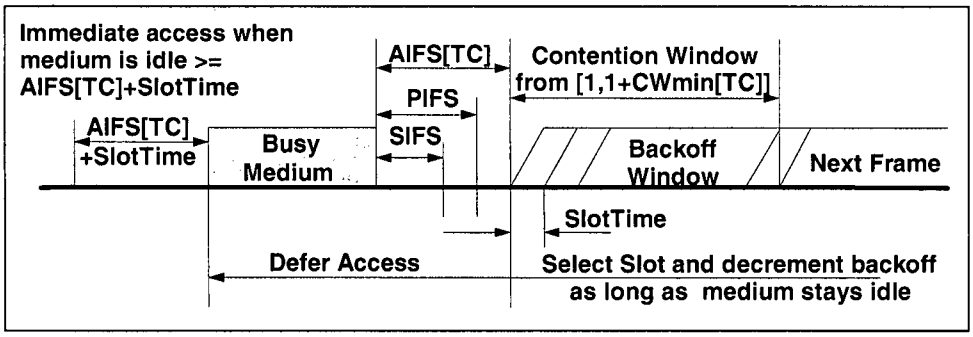

Figure 1-5 IFS relationships in 802.11e EDCA (DCF)

One of the new efficiency mechanisms that have been introduced by 802.11e is the possibility of transmitting multiple frames in one burst of frame exchanges, called transmission opportunity or TXOP (Figure 1-6). TXOP specifies the duration of time in which a station can hold the channel uninterrupted. Frame exchange in a TXOP is done with SIFS spacing which prevents interruption by other stations. A TXOP can be obtained by either winning contention

or through a CAP generation by the access point. Other than allowing for multiple frame exchange, which reduces collision and increases throughput, a TXOP can also be used to limit the problem of unfair access by slow stations. This problem occurs in a multirate WLAN in which a station with low transmission rate occupies the channel for a much longer time than that of stations with higher rates. Using TXOP, the slow stations can be forced to use fragmentation and limit their use of the channel. We have conducted a study of the TXOP feature and its effect on the efficiency of a WLAN. This study can be found in [31].

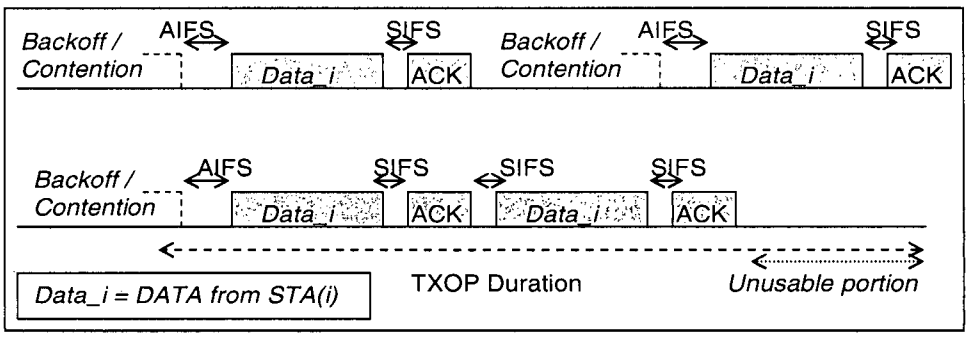

Figure 1-6 TXOP operation in 802.11e MAC

One of the most important features of the 802.11e is the capability of an AP to start a contention free duration, known as a Controlled Access Phase (CAP), even during contention periods. The AP is allowed to initiate a CAP after a PIFS following any frame exchange cycle in contention period. The original CFP (of the 802.11 standard) is in fact a CAP in the new standard (Figure 1-7). This feature allows for a more controlled and efficient polling mechanism in which periodic durations of contention and contention free periods are replaced with a contention period with occasional interruptions by controlled access phases. A CAP can be generated for either uplink or downlink directions. An uplink CAP is generated by sending a poll to a station and assigning a TXOP to it, a downlink CAP is generated by sending data frames to a station. Figure 1-8 elaborates on how CAPs are generated.

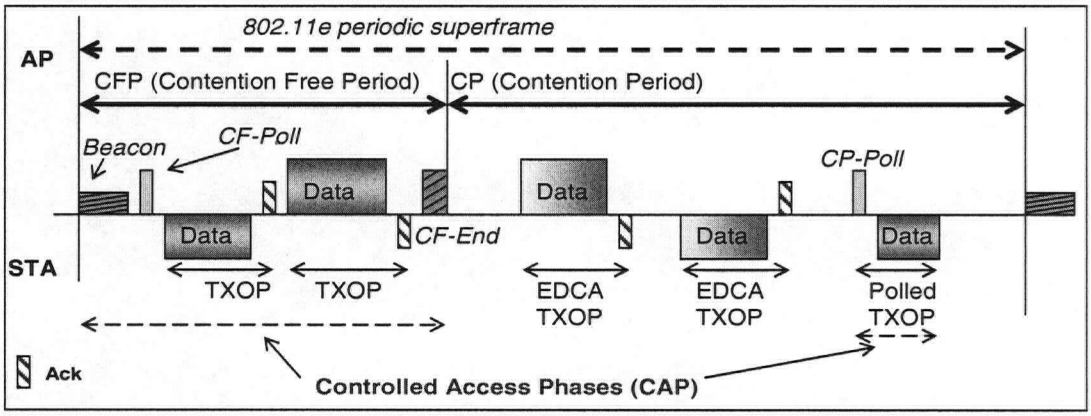

Figure 1-7 HCCA operation with periodic CFP

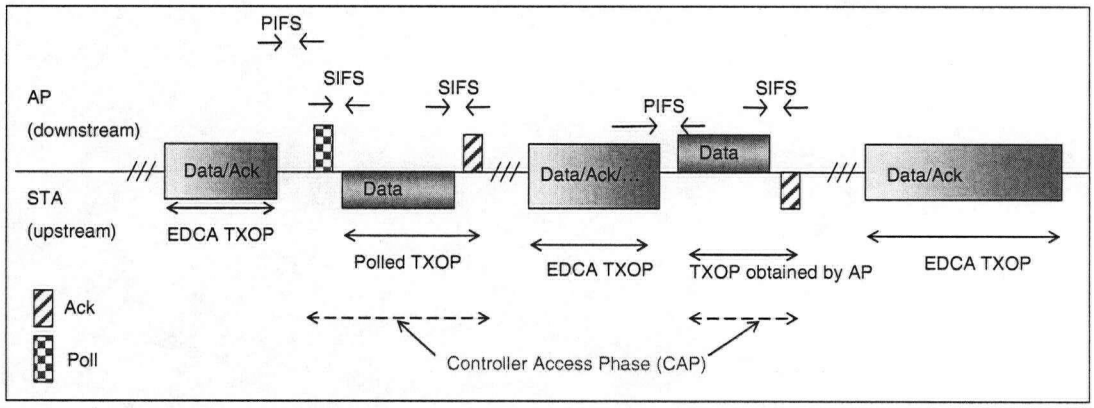

Figure 1-8 HCCA operation without CFP

The decision of when to generate a CAP lies with the scheduler that is implemented in the access point. The 802.11e standard does not mandate any scheduling and admission control mechanisms and leaves it to developers to devise their own solutions. However, a simple scheduler is presented in the standard as a reference (informative, not mandatory). This scheduler does not provide fair or guaranteed services, and is very inefficient for variable bitrate traffic [15][16]. This fact emphasizes the need for devising scheduling algorithms that can provide the required QoS for Multimedia applications, an issue that this thesis addresses.

1.4 Previous Work

Providing QoS in wireless access networks has been the focus of many research works in recent years. As a result there are numerous solutions, proposed for different types of wireless networks. Our focus in this thesis is on the per-session QoS provisioning mechanisms, in particular fair scheduling algorithms. We also review some QoS solutions that are based on providing prioritized and aggregate services. In this section we first review these general solutions for wireless networks and then examine the works dedicated to 802.11 WLANs.

Devising fair scheduling algorithms for providing guaranteed access while maintaining service fairness in wired networks has been the center of attention in many recent research works (e.g. [18][19][20][30]). Most of the notable proposed algorithms, such as Weighted Fair Queuing (WFQ), are packet based approximations of the ideal Generalized Processor Sharing (GPS) scheduler [18]. GPS is the fluid model scheduler that serves each queue a fair (weighted) amount of data in any duration of time. These solutions were mainly developed for wired networks in which packet loss or variation in server speed were not of concern. As a result their application in wireless networks required modifications to most of them. These modifications were needed to maintain the fairness of the algorithms when service disruption happened for one session, usually in the form of packet loss or broken link. In general, wireless networks are more error prone that wired networks and the probable packet losses are not negligible. In some wireless networks such as 802.11 WLANs, the speed of transmission may change dynamically and may be different for each station, even for each packet. Therefore special measures are needed to maintain the fairness of the scheduling algorithm.

To address the issue of packet loss, several algorithms have been developed that try to achieve packet based approximations of GPS in wireless networks. A survey of such methods is found in [3][7]. Some notable algorithms from this class of schedulers are Wireless Fair Service (WFS) [8], Idealized Wireless Fair Queuing (IWFQ) and its variation Wireless Packet Service (WPS)[9], and CIF-Q (Channel-Condition Independent Fair Queuing) [10]. These algorithms rely on a lead/lag model in which each session keeps a variable indicating whether the service it received is leading or lagging that of an equivalent error free system. When a session experiences a bad channel, its share of service time is lent to other sessions, and it will be labeled as lagging. Later, when the channel condition improves, the lagging session can retrieve the service time borrowed by the leading sessions and compensate its lag. Depending on the pace of compensation for the lagging sessions, and the criteria for selecting the leading session from which service is taken back, different fairness and service guarantee properties are resulted.

IWFQ and WPS present coarse short-term fairness and throughput bounds. CIF-Q and WFS achieve short-term and long-term fairness, short-term and long-term throughput bounds, and tight delay bounds for channel access. Although these algorithms may provide the necessary solution for wireless environments such as cellular networks, they lack the features that make them practical in a WLAN. In fact, these algorithms are designed for a single direction scheduling (essentially on the downlink from the access point) and are based on the assumption of a single fixed rate server. These assumptions are not applicable to a CSMA/CA network such as IEEE 802.11. A WLAN based on 802.11 shares the medium at all times between uplink and downlink flows and is inherently a distributed environment, it also allows different operational transmission rates for each station. This means that these existing

algorithms that were mainly for cellular networks are not directly usable in an 802.11e (or 802.11) network. The issue of multirate operation has been considered in other notable algorithms, such as the works presented in [24][25][26]; However, these solutions lack the same features that are necessary for CSMA/CA networks. Yuan in [24] proposes a temporal fair algorithm, which is a variation of WFS, but the algorithm is suitable only for single direction scheduling in a TDD or FDD system and does not consider the shared medium nature of WLANs. Tinnirello in [26] studies service time fairness but does not present any scheduling scheme.

Considering that the above algorithms are not directly applicable to 802.11 WLANs, we examine another class of QoS solutions specially designed for 802.11 networks. These solutions can be categorized into two groups of priority services and guaranteed access services. Priority services are provided through contention access mode of the MAC layer, while guaranteed access services use the polling mechanism of the MAC.

Algorithms presented in [13] and [14] provide prioritized differentiated services to aggregated flows. These solutions are mainly based on the contention access mechanisms and provide QoS in a probabilistic manner to traffic aggregates. Ni in [13] present a survey of several QoS methods based on contention access. The work in [14] presents a distributed deficit round robin algorithm that tries to enforce fairness through adjusting 802.11 contention mode parameters. The QoS provided by this method is probabilistic and applicable to aggregate of flows, therefore no guarantees can be made and unfairness of the algorithm is not bounded.

Contrary to contention mode solutions, research on providing per-session guarantees in WLANs, especially using the new *controlled access* features of the 802.11e standard, has been very limited. The 802.11e standard itself proposes a simple algorithm (referred to as TGe in this dissertation), which does not provide fair services and is only effective for strict constant bit rate (CBR) traffic. The methods in [15] and [16] improve the proposed simple TGe scheduler, but are still not considered fair with guaranteed service provisioning.

The method in [15] extends the original algorithm by adjusting the transmission duration based on the collected queue size information from the stations, and an estimation of its future queue size. Although this method is more efficient than the TGe algorithm, it is based on an estimation of the queue sizes and is only fair in the long run. Also, [16] proposes another extension to TGe by addressing the issue of inefficiency for variable bit rate (VBR) traffic; however, this method is inherently not fair and similar to [15] uses transmission opportunity assignments instead of packet scheduling, hence it is susceptible to long delays due to simultaneous bursty transmission that may occur on all flows. Flow isolation is also very poor with either algorithm presented in [15] and [16] due to the fact that they use admission control on average rate basis while service assignment is burst size dependent. Physical layer impairments such as packet loss are also not addressed efficiently by these algorithms.

Our proposed solution, in this dissertation, uses the controlled access features of the 802.11e standard, a multirate fair GPS based scheduling algorithm, and a framework for combining uplink and downlink scheduling and achieves per session QoS in a WLAN. Where required, we revisit the solutions reviewed in this section and compare them with our proposed solution.

1.5 Thesis Scope and Organization

This dissertation focuses on providing QoS to multimedia applications in WLANs. This goal is achieved through scheduling solutions that provide fair guaranteed services for multimedia sessions in a WLAN. We target demanding applications such as voice and video telephony that have stringent delay and bitrate guarantee requirements. Such strict QoS requirements cannot be provided by the EDCA mechanism, especially in heavily loaded networks.

Since EDCA is the most readily available QoS solution mechanism that comes as part of the standard, most vendors will use it as the first step in migration from 802.11 based WLANs to the new 802.11e based QoS enabled networks. However, EDCA is based on probabilistic QoS provisioning and the level of service it provides, especially in heavily loaded networks, is not adequate for demanding application such as voice and video telephony. The delay and delay jitter measurements from experiments with EDCA based networks confirm this fact. Also, we observe, through analysis and simulation experiments, that the performance of EDCA is very sensitive to its parameters; and even with optimized parameters, its contention based nature will result in degraded service. We provide a throughput analysis of the EDCA mechanism in saturation mode in Chapter 2 and examine its performance, through simulation experiment in Chapter 3 and Chapter 4. We have also compared the performance of EDCA with that of an adaptive EDCA algorithm as well as a periodic CFP based scheduling in [23].

Having shown the shortcomings of EDCA, we develop a scheduling framework in Chapter 3 that provides a complete QoS solution for fair, guaranteed service provisioning in 802.11e based WLANs. This solution is applicable to any multiple access network similar to 802.11e WLAN. It can also use any admission control mechanism, such as those described in the

standard [2], its extensions [16] [17], or any generic wireless admission control mechanism. Admission control is outside the scope of this dissertation and we focus on the scheduling solution and bandwidth assignment algorithms for achieving real-time multimedia communications. As it was discussed in section 1.4, WLANs have certain features that prevent us from using the available QoS mechanisms for wired networks or cellular wireless networks. In a WLAN environment, deriving QoS mechanisms that achieve fairness and provide delay and loss guarantees, is much more challenging for the following reasons:

- The medium access is done in a distributed manner among stations and any scheduling technique has to co-ordinate stations instead of simply serving local queues.
- The conventional single server queuing model is not applicable if a mixture of prioritized and reservation based traffic are to be served in a distributed WLAN.
- The medium data rate might change during the lifetime of a session; contrary to the fixed server rate of the conventional schedulers. Packet loss and retransmission also affect the fairness of the algorithm.

Given the above facts we propose a scheduling framework in Chapter 3 that centralizes the scheduling task in the access point (addressing the first issue). We achieve this goal by introducing the concept of virtual packets, which are representations of stations actual packets in the access point, and combining the scheduling task for virtual and actual packets. Then, we introduce a new queuing model integrated with a generalized processor sharing based scheduler as well as a traffic shaping scheme to achieve fair guaranteed services using the centralized scheduler (addressing the second issue). This solution provides us with the desired QoS performance for a fixed rate WLAN.

We address the issue of multirate operation of a QoS enabled WLAN in Chapter 4, where we extend the scheduling algorithm to provide either throughput or temporal fairness. In this chapter, we develop the hypothesis of GPS based temporal and throughput fair scheduling, and analyze the complexity issues that may arise. We then describe our recommended CAPS solution with a modified SFQ at its core, and present an extensive delay analysis of this model, which is followed by its performance evaluation.

We conclude this thesis in Chapter 5, where we present a summary of the main contributions of this thesis to the field of wireless multimedia networking. Potential research subjects that can immediately follow this research are also presented in this chapter.

1.6 References

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.

[2] IEEE Standard 802.11e/ Amendment 8, "Medium Access Control (MAC) Quality of Service (QoS) Enhancements," July 2005.

[3] 3GPP TS 23.234: "3GPP Systems to Wireless Local Area Network (WLAN) Interworking; System Description (Release 6)," TS 23.234. v.1.10.0, May 2003.

[4] Broadband Radio Access Networks (BRAN); HIPERLAN Type 2 Functional Specification; Data Link Control (DLC) Layer; Part l: Basic Data Transport Function, ETSI Report TR 101 761-1, Ver. 1.1.1, April 2000.

[5] Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part2: Radio Link Control (RLC) Sublayer, ETSI Report TR 101 761-2, Ver. 1.1.1, April 2000.

[6] H. Fattah and C. Leung, "An Overview Of Scheduling Algorithms In Wireless Multimedia Networks", Wireless Communications, IEEE, vol. 9, Issue 5, pp. 76– 83, Oct 2002.

[7] T. Nandagopal , S. Lu and A. V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms", Wireless Networks, vol. 8 , Issue 2/3, pp. 231 - 247, March-May 2002.

[8] S. Lu, T. Nandagopal and V. Bharghavan, "Fair scheduling in wireless packet networks", ACM MOBICOM'98, October 1997.

[9] S. Lu, V. Bharghavan and R. Srikant, "Fair scheduling in wireless packet networks", IEEE/ACM Trans. on Networking, vol. 7 , Issue 4, pp. 473 - 489, August 1999.

[10] T.S. Ng, I. Stoica and H. Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors", Proc. of IEEE INFOCOM '98, vol. 3, pp. 1103–1111, March 1998.

[11] A. Kamerman and L. Montean, "WaveLAN-11: A High-Performance Wireless LAN far the Unlicensed Band", Bell Labs Technical Journal, pp. 118-133, 1997.

[12] J. Rosenberg, et. al., "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[13] Q. Ni, L. Romdhani, and T. Turletti. "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN", Wiley Journal of Wireless Communication and Mobile Computing (JWCMC), John Wiley and Sons Ltd., vol. 4, Issue 5, pp. 547-566, 2004.

[14] W. Pattara-Atikom, P. Krishnamurthy, S. Banerjee," Distributed mechanisms for quality of service in wireless LANs", Wireless Communications, IEEE, vol. 10 , Issue: 3 , pp. 26 – 34, June 2003.

[15] P. Ansel, Q. Ni, and T. Turletti, "An efficient scheduling scheme for IEEE 802.11e," WiOpt'04: Modeling and Optimization in Mobile, AdHoc and Wireless Networks, 2004.

[16] A. Grilo, M. Macedo and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks", IEEE Wireless Communications, vol. 10, Issue 3, pp. 36-43, June 2003.

[17] A. Banchs, M. Radimirsch, X. Perez, " Assured and expedited forwarding extensions for IEEE 802.11 wireless LAN" Quality of Service, 2002. Tenth IEEE Int. Workshop on, pp. 237 – 246, May 2002.

[18] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," IEEE/ACM Trans. on Networking, vol. 1, no. 3, pp. 344-357, June 93.

[19] J.C.R. Bennett, H. Zhang, "WF^2Q: Worst-Case Fair Weighted Fair Queueing", INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE, vol. 1, pp. 120–128, March 1996.

[20] P. Goyal, H.M. Vin, H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm For Integrated Services Packet Switching Networks", Networking, IEEE/ACM Transactions on, vol. 5, Issue 5, pp. 690-704, Oct. 1997.

[21] Y. Pourmohammadi Fallah, H. Alnuweiri "Hybrid Polling and Contention Access Scheduling in IEEE 802.11e WLANs", accepted for publication in the Journal of Parallel and Distributed Computing, Elsevier, 2006.

[22] Y. Pourmohammadi Fallah, H. Alnuweiri "A controlled-access scheduling mechanism for QoS provisioning in IEEE 802.11e wireless LANs" ACM international workshop on Quality of service & security in wireless and mobile networks Q2SWinet '05, pp. 122 – 129, 2005.

[23] Y. Pourmohammadi Fallah, H. Alnuweiri, "Enhanced Controlled-Access and Contention-Based Algorithms for IEEE 802.11e Wireless LAN", Wireless Networks,

Communications and Mobile Computing, 2005 International Conference on vol. 1, pp 409 – 414, June 2005.

[24] Y. Yuan, D. Gu, W. Arbaugh, J. Zhang, " Achieving Fair Scheduling Over Error-Prone Channels in Multirate WLANs", Wireless Networks, Communications and Mobile Computing, 2005 International Conference on, vol. 1, pp. 698–703, June 2005.

[25] Z. Jiang, N.K. Shankaranarayana , "Channel Quality Dependent Scheduling for Flexible Wireless Resource Management", Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE, vol. 6, pp. 3633-3638 , Nov. 2001.

[26] I. Tinnirello, S. Choi, "Temporal fairness provisioning in multi-rate contention-based 802.11e WLANs", World of Wireless Mobile and Multimedia Networks, Sixth IEEE International Symposium on a, pp. 220 – 230, June 2005.

[27] Y. Pourmohammadi Fallah , H. Alnuweiri, "Performance Analysis of Controlled Access Phase Scheduling Scheme for Per-Session QoS Provisioning in IEEE 802.11e Wireless LANs", Proc. of 2006 IEEE Wireless Communications and Networking Conf., vol. 3,pp. 1414-1420 , April 2006.

[28] J.Y. Le Boudec, "Application of Network Calculus To Guaranteed Service Networks", IEEE Trans on Information theory, vol. 44, Issue 3, May 1998.

[29] R. Agrawal, R. L. Cruz, C. Okino and R. Rajan, "Performance Bounds for Flow Control Protocols", IEEE Trans. on Networking, vol. 7, No3, pp. 310-323, June 99.

[30] H. Alnuweiri, H. Tayyar, "Analysis of virtual-time complexity in weighted fair queuing.", Computer Communications (2005), vol. 28, Issue 7, pp. 802–810, 2005.

[31] Y. Pourmohammadi Fallah, H. Alnuweiri, "Modeling and Performance Evaluation of Frame Bursting in Wireless LANs", Proc. of 2006 Int. Wireless Communications and Mobile Computing Conf., pp 869-874, July 2006.

Chapter 2. Generalized Saturation Throughput Analysis of EDCA

2.1 Introduction

The new IEEE 802.11e standard introduces enhancements to the Medium Access Control (MAC) layer of the original 802.11 standard for Wireless Local Area Networks (WLANs) [1]. While the original 802.11 standard supports only best effort services, the new standard offers priority-based access through several mechanisms. The new services are provided through a new access scheme called Enhanced Distributed Channel Access (EDCA). EDCA is built on top of the original distributed coordination function (DCF) of the 802.11 standard, and introduces class-specific contention window sizes and initial deferment (or Inter Frame Spacing, IFS) times. Probabilistic priorities are possible using these prioritized access parameters.

There have been several efforts to analyze the behaviour of the EDCA mechanism under *saturation* conditions, a model that assumes all stations queues remain continuously backlogged. In this regards, we found models developed by Robinson [3] and Xiao [4] noteworthy. However, the model developed in [4] only considers the contention window size for prioritizing frames, ignoring the important effect of different IFS times for different traffic classes (using Arbitration IFS, or AIFS timers). The model developed in [3] considers both

---

A version of this chapter has been submitted for publication. Yaser Pourmohammadi Fallah, Samer El-Housseini, Hussein Alnuweiri, A Generalized Saturation Throughput Analysis for IEEE 802.11e Contention-Based MAC.

contention-window and AIFS parameters, but assumes a post collision behaviour that does not comply with the 802.11 or 802.11e standards. When a collision occurs in an 802.11 WLAN, the stations involved in the collision wait for an Ack Timeout period after the end of collision before they can realize that a collision has occurred. Then, they will resume normal contention. Stations not involved in collision only wait until the end of the collided transmission and then resume normal operation. The model in [3] assumes that non-colliding stations will wait for an EIFS period before resuming contention.

Apart from the post collision behavior assumption, the study published in [3] presents a correct way of modeling the behavior of contending stations with two different priorities, or Access Categories. In this chapter, we present a correction of the model in [3] and generalize it to an arbitrary number of priorities instead of only two. Our model provides a complete solution for modeling 802.11e EDCA operating in saturation mode.

2.2 Model Development

EDCA is a contention-based access method whereby stations that need to access the medium after a busy period must wait for a random duration of time before they are allowed to start transmitting on the channel. This random waiting time is different for each priority and consists of two components: AIFS which is a short time-interval, and a back-off time randomly chosen from a class (access category) specific contention window (CW) size. IEEE 802.11 defines four different length AIFS values for each priority class. Similarly, the CW sizes are specified by priority-specific parameters *a*CWmin and *a*CWmax. Each queue must first sense an idle medium (channel) for an AIFS plus a random-backoff period before transmitting a frame. The backoff duration is chosen randomly between 1 and CW slots, and stored in a *backoff counter*. CW is initially set to *a*CWmin then doubled after each collision

until it reaches *a*CWmax. A successful transmission resets CW back to *a*CWmin.

Given that each queue in EDCA contends for the channel independently, we can model the backoff process of each queue separately, and apply the DCF model presented in [2] to each queue. An approximation in this modeling, also used by Robinson in [3], is to ignore the effect of internal contention resolution that happens between the queues of one station. In EDCA if two queues of the same station attempt to transmit at the same time, the queue with higher priority wins and the other queue repeats its backoff process without incrementing its retry counter. We verified the accuracy of the model in [2] and use it in the first stage of our analysis.

To model a single queue, we begin by mathematically defining the backoff process behavior for stations operating in a saturation mode. In this case, all transmissions consist of two stages: waiting for the AIFS expiry, and waiting for the backoff-counter to reach zero. To model the backoff process we first model the process for one queue with given EDCA parameters. This can be done following the same queuing strategy described in [2] and [3], whereby the backoff value was modeled as a two-dimensional Markov Process. One dimension of this chain describes the backoff counter value, while the other tracks the retransmissions stage, in effect modeling the exponential backoff procedure (refer to [2] for a detailed examination of this model). For such a model to be correct there are two fundamental assumptions. First, in each idle slot, each queue may attempt to transmit with an independent and constant priority τ. Second, in each transmission attempt and regardless of the number of past collisions, a packet transmission may result in a collision with an independent and constant probability p. However, since in EDCA each slot after a busy period is available to only a subset of access categories due to AIFS-differentiation, the number of contending stations may be different in

each slot. For this reason, an average conditional collision probability $\overline{p}$ is used instead of p, and the same analysis as in [2] is applicable. From this model the following equation is derived that describes τ_j as a function of $\overline{p}_j$ for *each* priority level j (refer to *Appendix A* for more details):

$$\tau_j = \frac{2}{1 + W_j + \overline{p}_j W_j \sum_{i=0}^{m_j - 1} (2\overline{p}_j)^i} \tag{2.1}$$

In (2.1) W_j is aCWmin for class j and m_j is the maximum number of exponential backoff stages (aCWmax = $2^{m_j} W_j$). To solve for all τ_j and $\overline{p}_j$, we need another set of equations describing $\overline{p}_j$ in terms τ_j for each priority level. For this purpose we need to find the probability of collision for each class of traffic in each slot after a busy period and then find the average collision probability for each class ($\overline{p}_j$). The next section describes how $\overline{p}_j$ is derived.

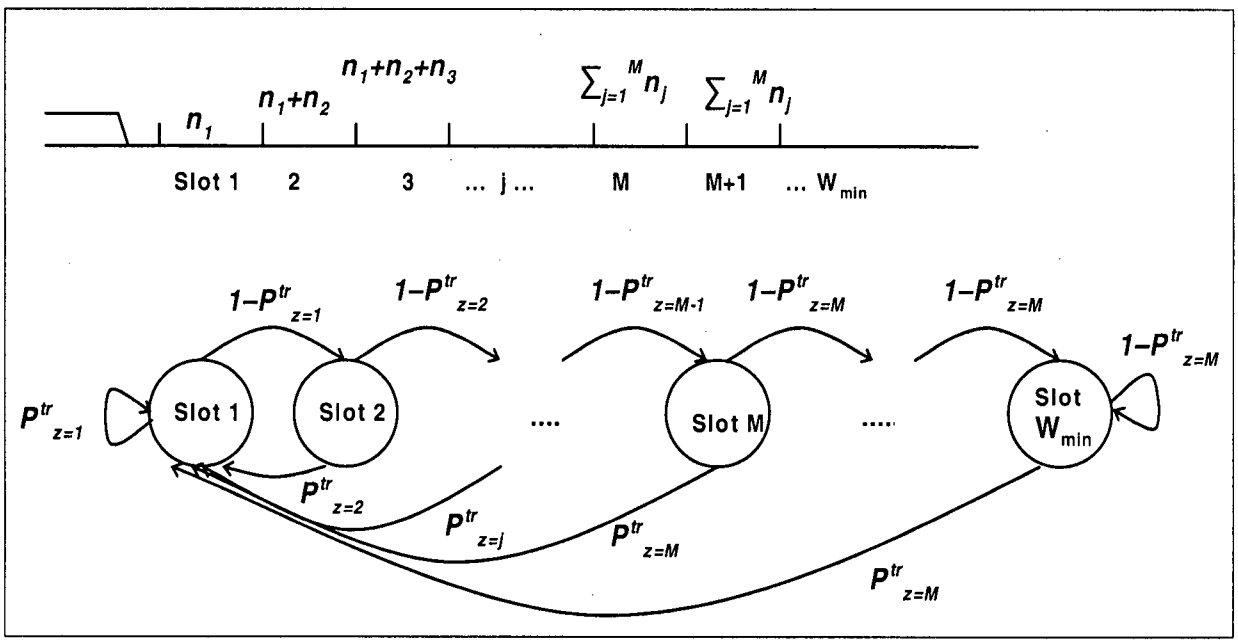

Figure 2-1 Modeling slot-occupancy for different contention zones

2.3 Slot Occupancy

Recall that each priority level has its own AIFS and Contention Window. To examine the contention behaviour after busy periods, we assume that there are K possible priority levels with each slot containing n_j (j=1, 2, ..., K) contending stations (or queues). Each priority level has an AIFS value equal to j slots, meaning that priority level j will start contention in the jth idle slot. For example if only the three priorities classes 1, 4,and 7 are present, then slots 1 to 3 will have n_1 contending stations, while slots 4 to 6 will have n_1+ n_4 stations, slot 7 onwards will see n_1+n_4+n_7 contending stations.

We model the contention slots after a busy period by a separate Markov Chain in order to derive the slot occupancy and probabilities of collision in each slot. Our model, depicted in Figure 2-1, extends the model presented in [3]. In Figure 2-1 W_{min} is the minimum of all the aCWmax values.

If n_i stations from priority i are present, then the equations that govern this model are:

$$P^{tr}_{z=1} = 1-(1-\tau_1)^{n_1}, \quad P^{tr}_{z=2} = 1-(1-\tau_1)^{n_1}(1-\tau_2)^{n_2}$$

$$P^{tr}_{Z=z} = 1-\begin{cases} \prod_{i=1}^{z}(1-\tau_i)^{n_i}, & 1 \le z \le M \\ \prod_{i=1}^{M}(1-\tau_i)^{n_i}, & z > M \end{cases} \tag{2.2}$$

Knowing $P^{tr}_{Z=z}$, the probability of at least one transmission in zone z, and assuming that system is in equilibrium with stationary distribution b_i (i :$1..W_{min}$) we have:

$$b_i = (1-P^{tr}_{z=i-1})\cdot b_{i-1} \qquad (i : 2 .. W_{min} -1)$$

$$b_{W\min}\cdot P^{tr}_{z=M} = b_{W\min-1}\cdot(1-P^{tr}_{z=M})$$

$$b_1(1-P^{tr}_{z=1}) = \sum_{i=2}^{W\min} b_i \cdot P^{tr}_{z=i}$$

$$\sum_{i=1}^{W\min} b_i = 1 \tag{2.3}$$

Using an approximation similar to [3], we assume W_{min} is the smallest CW_{max} in the system. Then b_1 becomes:

$$b_1 = 1/(\ 1 + \sum_{i=2}^{W_{\min}-1} \prod_{j=1}^{i-1} (1 - P_{Z=j}^{tr}) + \frac{1}{P_{Z=M}^{tr}} \prod_{j=1}^{W_{\min}-1} (1 - P_{Z=j}^{tr})\ ) \tag{2.4}$$

Using (2.4) and (2.3), all the stationary probabilities are computed based on the values of τ_j (where j is the traffic class). Next, we find the average conditional collision probability $\overline{P}_j$ for each priority j (defined as the probability that a transmitted packet of class j encounters collision). To do so we need to find the probability that a given slot z experiences a collision for a class-j station. Denoting this probability by Pc_z^j we can derive it as follows knowing the number of eligible stations for each slot, and the probability of each station transmitting in that slot (τ_j):

$$Pc_{z=1}^{j=1} = 1 - (1-\tau_1)^{n_1 - 1} \quad , Pc_{z=1}^{j=2} = 0 \quad , Pc_{z=2}^{j=1} = 1 - (1-\tau_1)^{n_1-1} \cdot (1-\tau_2)^{n_2}$$

$$Pc_{z=2}^{j=2} = 1 - (1-\tau_1)^{n_1} \cdot (1-\tau_2)^{n_2 - 1}$$

$$Pc_z^j = 1 - (1-\tau_j)^{n_j - 1} \cdot \prod_{k=1, k \neq j}^{z} (1-\tau_k)^{n_k} \qquad for \quad j \leq z$$

$$Pc_z^j = 0 \qquad\qquad for \quad j > z \tag{2.5}$$

From (2.5), we find the average conditional probability of collision for class-j stations by summing the slot specific probabilities weighted according to slot i occupancy b_i:

$$\overline{P}_j = \sum_{i=j}^{W\min} b_i \cdot Pc_{z=i}^j \tag{2.6}$$

Given that $\overline{P}_j$ is a function of τ_j for several j's, (2.6) and (2.1) form a system of nonlinear equations, amenable to solution by numerical methods. Using such methods we find the

average collision and transmission probabilities for each priority class ($\overline{P}_j, \tau_j$). Consequently we can derive the achievable throughput for each class.

2.4 Throughput Analysis

To find the throughput of the system we introduce the *transmission-event period* (T_p), defined as the expected time between such events as successful transmissions, collisions, and idle slots. Since the probabilities of success and collision in each slot after a busy period depend on the slot occupancy, we calculate the expected transmission event period by considering each slot and weighting its corresponding expected duration according to the slot occupancy probability b_i as follows:

$$T_p = \sum_{i=1}^{W\min} b_i \cdot ((1 - P_{z=i}^{tr}) \cdot \delta + \overline{P}_{z=i}^{S} \cdot T^s + \overline{P}_{z=i}^{Col} \cdot T^{col}) \tag{2.7}$$

Observe that T^S in (2.7) is the average duration of one successful transmission, including data and control frames such as Ack and the IFS deferment times [2]; this may also include a bursty transmission of multiple frames that is limited by a "transmission opportunity" (TXOP) duration and can be different for each priority level [6]. T^{col} is the average time spent in each collision (i.e., average frame length plus IFS deferments), and δ is the length of an idle slot which is the defined slot time in 802.11. Also $\overline{P}_{z=i}^{S}$ and $\overline{P}_{z=i}^{Col}$ denote the average probabilities of a given slot i seeing a successful transmission or a collision, respectively. The probability of an idle slot occurring in zone i is simply given by $(1 - P_{z=i}^{tr})$. To calculate $\overline{P}_{z=i}^{S}$, we first find $P_{j,z=i}^{S}$, the probability of successful transmission by a single station of class j in slot i. $P_{j,z=i}^{S}$ is in fact the probability that a station of class j transmits (τ_j), and no other station transmits ($\frac{1 - P_{z=i}^{tr}}{1 - \tau_j}$); thus we have:

$$P^{S}_{j,z=i} = \begin{cases} \tau_j \cdot \dfrac{1-P^{tr}_{z=i}}{1-\tau_j} & , \quad j \le i \\ 0 & , \text{ otherwise} \end{cases} \tag{2.8}$$

Therefore, the probability that the outcome of slot i is a successful transmission by any class is found as (given that the success probabilities are mutually exclusive between any two stations):

$$\overline{P}^{S}_{z=i} = \sum_{j=1}^{i} (n_j \cdot P^{S}_{j,z=i}) \tag{2.9}$$

and the probability of slot i seeing a collision is:

$$\overline{P}^{col}_{z=i} = P^{tr}_{z=i} - \overline{P}^{S}_{z=i} \tag{2.10}$$

As the final step in computing the WLAN throughput, we derive the expected payload size (EP_j) for each class j which is successfully served during a transmission period:

$$EP_j = \cdot \sum_{i=1}^{W_{\min}} b_i \cdot n_j P^{S}_{j,z=i} \cdot P_L \qquad (P_L \text{ refers to the average payload size in a frame}) \tag{2.11}$$

Dividing (2.11) by (2.7) we can derive the throughput for each class j:

$$Throughput_j = \frac{EP_j}{T_p} \tag{2.12}$$

2.5 Simulation Evaluation

To evaluate our analytical model we have solved the model equations using numerical methods and compared the results with simulation results generated from an accurate EDCA simulator (developed at UBC). The comparison depicted in Figure 2-2 shows that the mathematical model results and the simulation results are very close and almost identical. In Figure 2-2, we plotted the collision probability p and the throughput versus the number of stations for four priorities A to D (A being the highest priority). The four priorities correspond

to the four access categories (AC[3] to AC[0]) currently defined for IEEE 802.11e. As shown in Figure 2-2, both AIFS and CW play a significant role in providing throughput service differentiation. One notable fact is that the difference in collision probability is much more in the case of AIFS differentiation.

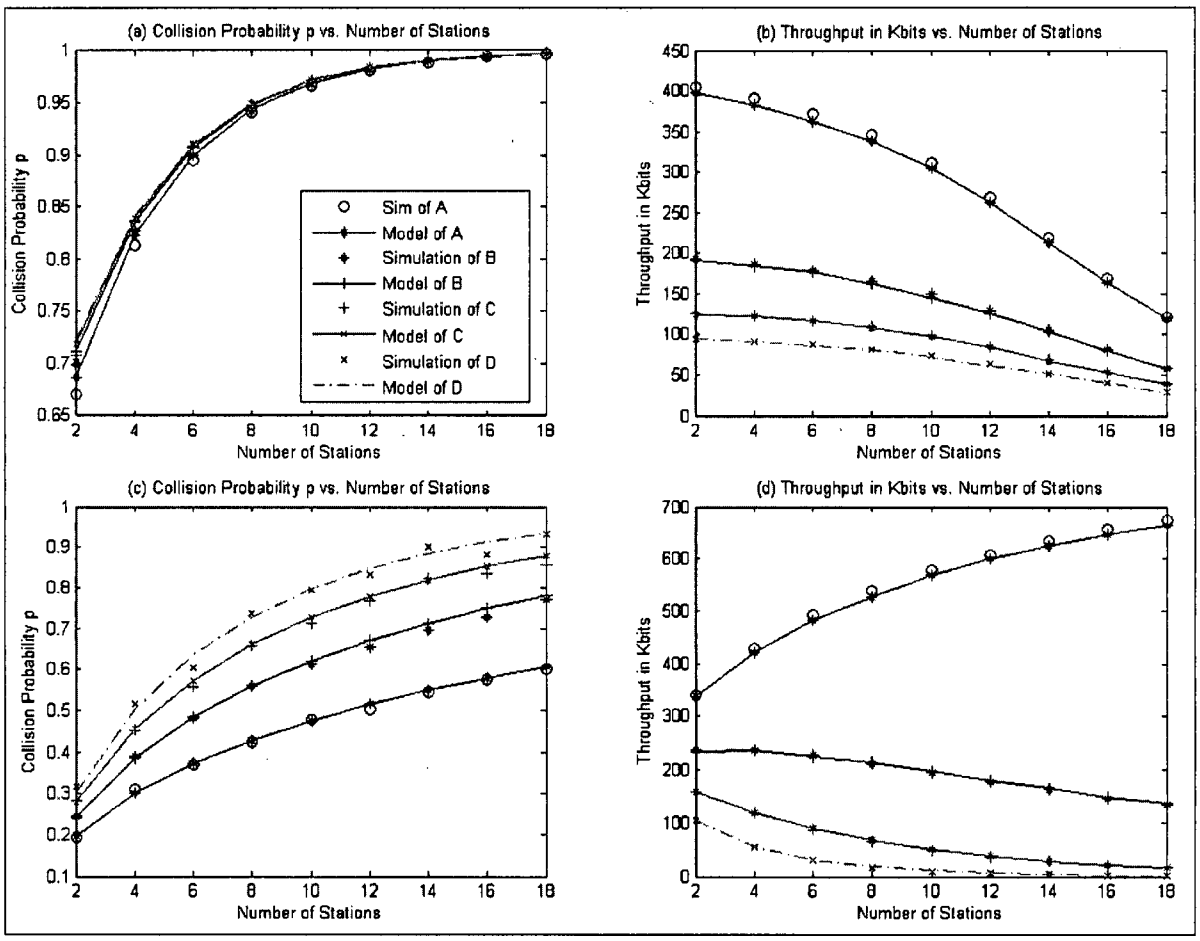

Figure 2-2: Collision Probability & Saturation Throughput vs. Num. of Stations (per class). Top plots- AIFS is 1 for all classes, CWmin, CWmax are { A(8,16) , B(16,32) , C(24,48) ,D(32,64)} Bottom plots - CWmin=32 & CWmax=64 for all classes, AIFS is 0, 1, 2, and 3 for classes A to D.

We also note that the throughput of each class, and the system, degrades considerably when the number of stations increases. We see in later chapters of this thesis that this performance degradation limits the ability of EDCA to provide services to multimedia applications in a WLAN. In particular, without changing the default EDCA parameters (which is allowed by the

standard) the system performance degrades to a level that makes the system unusable, especially for high priority traffic. Our model, presented in this chapter, can be used as an analytical tool for devising adaptive algorithms that adjust EDCA parameters in order to achieve better performance in an 802.11e WLAN. The presented model complements previous works on analyzing saturation operation of a WLAN.

2.6 References

[1] IEEE 802.11e Standard – Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, July 2005.

[2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE J. on Selected Areas in Communications, vol. 18, no. 3, March 2000.

[3] J. Robinson, and T. Randhawa, "Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function", IEEE J. on Selected Areas in Communications, vol. 22, Issue 5, pp. 917 - 928, June 2004.

[4] Y. Xiao, "Performance Analysis of IEEE 802.11e EDCF under Saturation Condition", IEEE International Conf. on Communications, vol. 1, pp. 170–174, June 2004.

[5] J. Hui, M. Devetsikiotis; "A Unified Model for the Performance Analysis of IEEE 802.11e EDCA", IEEE Transactions on Communications, vol. 53, Issue 9, pp. 1498 – 1510, Sept. 2005.

[6] Y. Pourmohammadi Fallah, H. Alnuweiri, "Modeling and Performance Evaluation of Frame Bursting in IEEE 802.11 WLANs", Proc. of 2006 Int. Wireless Communications and Mobile Computing Conf., pp. 869-874, July 2006.

[7] J.W. Tantra, C. H. Foh, A. B. Mnaouer, "Throughput and Delay Analysis of the IEEE 802.11e EDCA Saturation", 2005 IEEE International Conference on Communications, vol. 5, pp. 3450 – 3454, 16-20 May 2005.

Chapter 3. Per Session QoS Provisioning in WLANs

3.1 Introduction

Supporting real-time multimedia applications such as voice-over-IP, video telephony and TV over Wireless Local Area Networks (WLAN) requires realizing guaranteed services that are not currently provided by existing WLAN technologies such as IEEE 802.11[1]. To address this issue, the IEEE has approved a new standard, IEEE 802.11e [2], to enhance the original MAC layer of the 802.11 standard with features that facilitate guaranteed and differentiated service provisioning. However, the standard only specifies the features required for the new service provisioning and leaves the design of specific scheduling disciplines that utilize these features to the developers and equipment vendors. The solution proposed in this chapter fills this gap by showing how to utilize the available features to provide guaranteed services for real-time multimedia applications. We target the infrastructure mode of operation in which a central *access point* (AP) controls the network. Most commercial and residential WLANs use this mode.

The need for providing Quality of Service (QoS) for real-time applications in wireless networks has been driving research activities and standardization efforts for some time. In

---

A version of this chapter has been accepted for publication. Yaser Pourmohammadi Fallah, Hussein Alnuweiri, Hybrid Polling and Contention Access Scheduling in IEEE 802.11e WLANs, Journal of Parallel and Distributed Computing, Elsevier, Vol. 67, Issue 2, pp. 242-256, February 2006.

particular, there have been considerable efforts in devising fair scheduling algorithms for wireless environments [3]. These efforts were mostly concentrated on scheduling in cellular networks or generic wireless environments. For example, some notable algorithms such as WFS (Wireless Fair Service) [4], IWFQ (Idealized Wireless Fair Queuing) and its variation Wireless Packet Service (WPS)[5], and CIF-Q (Channel-Condition Independent Fair Queuing) [6] address the scheduling issue in a general wireless network. Other algorithms such as the work presented in [8] address the scheduling issue in an access network in which broadcast and peer-to-peer communications are combined.

IWFQ and WPS present coarse short-term fairness and throughput bounds. CIF-Q and WFS achieve short-term and long-term fairness, short-term and long-term throughput bounds, and tight delay bounds for channel access. However, these algorithms are designed for a single direction scheduling (essentially on the downlink from the access point) and are based on the assumption of a single fixed rate server. These assumptions are not applicable to a CSMA/CA network such as IEEE 802.11. A WLAN based on 802.11 shares the medium at all times between uplink and downlink flows and is inherently a distributed environment, it also allows different operational transmission rates for each station. This means that these existing algorithms that were mainly for cellular networks are not directly usable in an 802.11e (or 802.11) network. The multi-rate operation is considered in other notable algorithms, such as AWFS [9][10]; but these algorithms also lack the same features that are necessary for a distributed CSMA/CA environment and do not consider the shared medium nature of WLANs.

There also exists another set of QoS solutions specially designed for 802.11 networks. Some of these algorithms, such as the ones proposed in [11],[15], and [16] provide prioritized differentiated services to aggregated flows. These solutions are mainly based on the contention

access mechanisms and provide QoS in a probabilistic manner to traffic aggregates. In fact, research on providing per-session guarantees in WLANs, especially using the new *controlled access* features of the 802.11e standard, has been very limited.

The 802.11e standard itself proposes a simple algorithm (referred to as TGe in this thesis), which does not provide fair services and is only effective for strict constant bit rate (CBR) traffic. The methods in [13] and [14] improve the proposed simple TGe scheduler, but are still not considered fair with guaranteed service provisioning. The method in [13] extends the original algorithm by adjusting the transmission duration based on the collected queue size information from the stations, and an estimation of its future queue size. Although this method is more efficient than the TGe algorithm, it is based on an estimation of the queue sizes and is only fair in the long run. Also, [14] proposes another extension to TGe by addressing the issue of inefficiency for variable bit rate (VBR) traffic; however, this method is inherently not fair and similar to [13] uses transmission opportunity assignments instead of packet scheduling, hence it is susceptible to long delays due to simultaneous bursty transmission that may occur on all flows. Flow isolation is also very poor with either algorithm presented in [13] and [14] due to the fact that admission control is done on average rate basis while service assignment is burst size dependent. Physical layer impairments such as packet loss are also not addressed efficiently by these algorithms.

Our solution focuses on using controlled access mechanisms to provide per session fair quality of service for real-time applications. We present a framework that allows for efficient scheduling of controlled and contention access periods while maintaining service guarantee and short term fairness through employing Generalized Processor Sharing (GPS) based scheduling. We demonstrate that it is possible to provide guaranteed per-session QoS without

need to depart from the IEEE 802.11e standard specifications as is the case with most other solutions. We identified three characteristics of an 802.11e WLAN that need to be taken into account when devising such a QoS solution:

- First, the solution must provide a way of efficiently sharing the medium between uplink and downlink flows; meaning that the solution should provide a unified scheduling scheme for the combined traffic flows from both directions.
- Second, the 802.11e describes access to the medium in a prioritized contention based scheme that is intermittently interrupted by contention free periods. The scheduler must efficiently distribute contention free and contention periods with flexibility of adjusting the duration of each access type on demand.
- Third, the scheduler must achieve proportional (weighted) fairness among sessions and be able to handle the effects of wireless channel variation.

We have developed a scheduling solution that addresses all these issues. To the best of our knowledge this is the first design that addresses all the above issues in a single framework for IEEE 802.11e networks.

In this chapter, we first provide a short description of the 802.11e standard, highlighting its controlled access mechanism. We then present a new access scheduling framework designed for the 802.11e MAC, and capable of providing per-session QoS guarantees for such applications as interactive voice and video over WLAN. Essentially, the proposed solution provides guaranteed services to flows that make reservation with the WLAN Access Point (AP) by means of the available MAC signalling methods, while at the same time, allowing the normal contention based access to take place using the remaining capacity of the channel. This approach is different from the existing polling mechanisms in which long alternating

contention free and contention periods are generated (e.g., [19]), resulting in uncontrolled delay bounds and inefficient operation. Our design approach is called Controlled Access Phase Scheduling (CAPS). The CAPS algorithm is based on a number of novel concepts such as Virtual Packet generation and combined scheduling of uplink and downlink flows [17], as well as using the well established Generalized Processor Sharing (GPS) based scheduling discipline in a new unified queuing framework for both contention and controlled access mechanisms.

3.1.1 IEEE 802.11e MAC Specifications

The IEEE 802.11e standard introduces new features that enhance the MAC layer of the original 802.11 standard in order to provide QoS to real-time multimedia applications [2]. The offered QoS can be categorized into two classes of prioritized contention access and guaranteed contention free access. Both schemes are built on top of an enhanced version of the Distributed Coordination Function (DCF) which is the main function of the 802.11 MAC. In general, access to the medium is done in a prioritized contention manner during each Contention Period (CP). The original MAC allowed the AP to initiate Contention Free Periods (CFP) on a periodic basis. The 802.11e MAC redefines CFP as a Controlled Access Phase (CAP) and allows initiating mini CFPs or CAPs arbitrarily even during the contention period.

The basis for the 802.11 MAC is a CSMA/CA mechanism (Carrier Sense Multiple Access with Collision Avoidance). This mechanism is essentially a contention access method that uses a binary backoff procedure for collision resolution and inter-frame space (IFS) time intervals for prioritizing access to the medium. The rules describing the timing relations in the MAC are described by DCF. Stations that have frames to send are only allowed to transmit if they find the channel idle for a frame-specific IFS duration (Figure 3-1). For data frames in contention mode, this waiting time is extended by a random backoff interval as well. If priorities are

specified, as in 802.11e, the contention window from which the random backoff number is selected, and the IFS waiting times may be different for each priority level.

The IFS gap for data and RTS frames is AIFS (Arbitration IFS), while beacons and initial CAP messages (poll or data) use a shorter gap time, PIFS, that gives them a higher priority in accessing the channel. Acknowledgements (Ack), packet fragments, responses to polls and CTS messages use a SIFS gap which is the shortest IFS, giving them the highest access priority. SIFS is only used when contention has already been won, or during a contention free period; therefore, it provides an uninterrupted control of the channel for as long as frames are sent with SIFS gaps. Poll and data frames that are sent using PIFS (to start a CAP or CFP) are also able to grab the channel unchallenged if they follow a completed frame exchange sequence; this is because after a frame exchange cycle finishes, all stations have to use AIFS plus backoff interval before they can access the channel while AP can send after PIFS, in effect giving it absolute priority over others. However, if the medium was free for a long time after a busy period, the PIFS waiting for AP and the AIFS plus backoff for stations might coincide, resulting in collision, or a data frame might grab the channel sooner. In any case, the AP can recover quickly by grabbing the channel after PIFS waiting following the busy or collision situation. This is because it does not have to do a backoff before starting a CAP or CFP and only needs to wait a PIFS, thus having guaranteed contention free access [2].

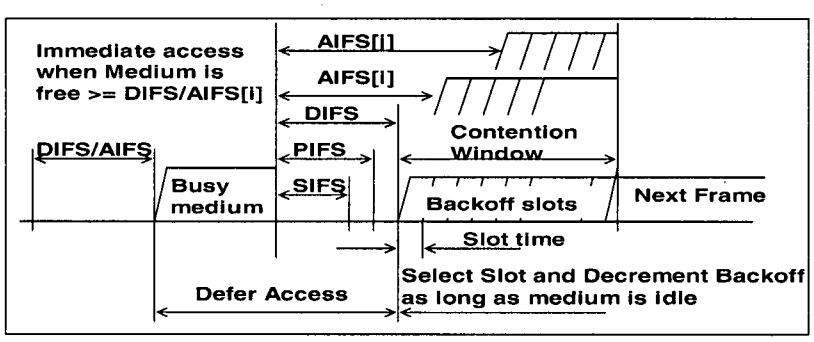

Figure 3-1 Some IFS relationships in DCF and EDCA

The 802.11e standard also introduces an important new concept: Transmission Opportunity (TXOP). A transmission opportunity specifies the duration of time in which a station can hold the medium uninterrupted and perform multiple frame exchange sequences consequently with SIFS spacing. A station can obtain a TXOP either through contention or be granted a TXOP by the AP. After completion of each frame exchange cycle during a TXOP, if enough time is left in the station's TXOP, it can retain control of the medium and commence a new frame exchange cycle after a SIFS period, otherwise it does not continue transmission using SIFS and enters the normal contention mode using AIFS deferred access and normal backoff.

MAC layer rules for controlling and coordinating access to the wireless medium in the 802.11e standard are specified under the Hybrid Coordination Function (HCF) protocol. HCF offers two access mechanisms; EDCA (Enhanced Distributed Channel Access) which is an enhanced version of DCF and is used for contention based access, and HCCA (HCF Controlled Channel Access) that replaces the Point Coordination Function (PCF) of the 802.11 standard and specifies the polling or controlled access schemes. The 802.11e standard defines 8 different traffic priorities in 4 access categories and also enables the use of *traffic stream* IDs (TSIDs), which allow per flow resource reservation.

Under EDCA access mechanism, depending on the type of a frame (Data or Control) and its priority, different AIFS values are used (Arbitration IFS or AIFS in Figure 3-1). The backoff windows are also different for each priority. Shorter AIFS times and smaller contention windows give higher access priority. This prioritization enables a relative and per-class (or aggregate) QoS in the MAC. The 802.11e standard allows for dynamically adjusting most EDCA parameters, facilitating performance enhancement using adaptive algorithms.

Under HCCA, access to the medium is controlled by the Access Point. HCCA is an enhanced version of the Point Coordination Function (PCF) of the original standard that controls the CFPs. The most important enhancement provided by HCCA is the new concept of Controlled Access Phase or CAP. A CAP is a usually short contention free period that is initiated during a contention period (Figure 3-2). An access point can start a CAP by sending a poll or data frame while it finds the medium idle for PIFS. Since PIFS is shorter than AIFS (used by EDCA), the AP is able to interrupt the contention operation and generate a CAP at almost any moment (with at most one packet length delay). A CFP (as described in 802.11) is also considered a CAP (Figure 3-2). However, with capability to generate CAPs at any time, there is no need for periodic CFPs. The CAP generation capability is the main feature that we use for providing per-flow QoS. The 802.11e standard does not specify the scheduling discipline that determines when CAPs are generated and leaves it to system developers to devise such a scheme.

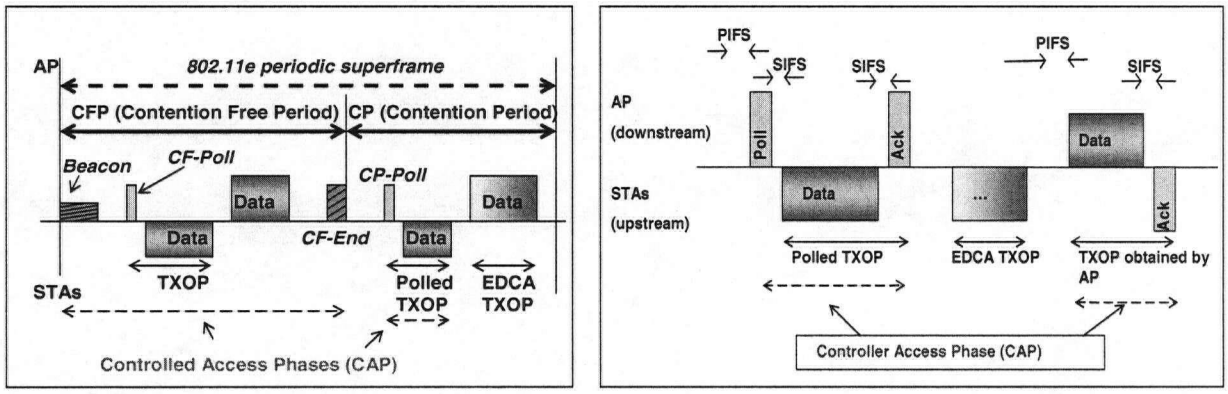

Figure 3-2 802.11e HCCA: CAP generation with CFP (left), without CFP (right)

The guaranteed access with bounded delay gives the AP the power to start a contention free access at any time with at most one packet length delay. This feature can be used to provide services for real-time applications that cannot tolerate unbounded delay or high jitter. At the start of a CAP the access point can send either a data frame (downlink CAP) or a poll message

(uplink CAP) after sensing the channel idle for PIFS. A CAP may include more than one consecutive frame exchange sequences that are limited by a station or flow specific TXOP.

When data frames are sent downlink, the AP decides for how long it will send frames to a particular destination; for uplink data frames, a station is only allowed to send frames for the duration of the TXOP granted by the AP. If this duration is short, the station must fragment its frames and only send the part that fits in the granted TXOP. If TXOP is set to zero the station is only allowed to send one frame (size limited by other MAC regulations).

The 802.11e standard draft provides flow IDs (Traffic Stream ID) in frame formats to enable per-flow QoS handling. It also specifies that it is the responsibility of stations to setup traffic streams (flows) and request resource reservation. This is done through sending an ADDTS request to the AP and asking for a traffic stream to be setup with specific traffic specifications. The information carried in the ADDTS request is used by the admission control and scheduling functions of the AP. The ADDTS response by AP completes the traffic stream setup procedure. The standard draft specifies the format in which the traffic stream specifications are described. In fact, we found this description to be very thorough. In particular fields such as service interval and start time are very useful in setting up scheduled access and poll messages.

3.2 CAPS: Controlled Access Phase Scheduling

Given the characteristics of an 802.11e WLAN, we present a unified QoS framework that addresses prominent aspects of a WLAN environment. Our scheduling framework has the following features: 1) Use of virtual packets to combine the task of scheduling uplink and downlink flows of a naturally distributed CSMA/CA environment into a central scheduler that resides in an AP; 2) Application of a GPS-based algorithm and an integrated traffic shaper in a

unified HCCA and EDCA queuing framework to provide guaranteed fair channel access to HCCA flows, and sharing the remaining capacity using EDCA (as illustrated in Figure 3-2). The following subsections describe the prominent features of our design, which is depicted in Figure 3-3, in more detail.

3.2.1 Centralizing the Scheduling Task: Combined Downlink/Uplink Scheduling

One important feature of CAPS is its ability to centralize the scheduling task in the inherently distributed WLAN environment. In an 802.11 WLAN, the medium is shared between downstream and upstream traffic at all times. Thus, any scheduling discipline must handle packet transmissions from individual stations to the AP (i.e. upstream), and from AP to the stations (i.e. downstream). Downstream packets are available in the AP buffers and can be directly scheduled, while upstream packets reside in the stations generating these packets and cannot be scheduled directly. However, the AP can use upstream traffic specifications, available through signalling or feedback, and schedule poll messages that allow for upstream packet transmission.

The key to realizing the above scheduling concept, is to represent packets from remote stations (i.e. the upstream packets) by "*virtual packets*" in the AP, then use a single unified scheduler to schedule virtual packets along with real packets (downstream packets). When scheduling virtual packets, the AP issues polling in the appropriate sequence to generate transmission opportunities for upstream packets. We call this mechanism *hybrid scheduling* because it combines upstream and downstream scheduling in one discipline. The performance of the scheduler will of course depend on the specific discipline used. In fact, the framework can use any conventional single server scheduler with some modifications. We propose to use

GPS based fair algorithms such as Start-time Fair Queuing (SFQ) [22], Weighted Fair Queuing (WFQ) [18], or Worst case Fair Weighted Fair Queuing (WF^2Q), [21]. For brevity, we name these CAPS options as CAPS-SFQ, CAPS-WFQ and CAPS-WF^2Q. Using a GPS based algorithm ensures fairness and bounded delay (thus controlled jitter) and increases the capacity of the system for supporting multimedia sessions. As will be shown later, we will modify these algorithms to suit them for the proposed framework. We will analyze these algorithms performance, and identify the best choice in different situations.

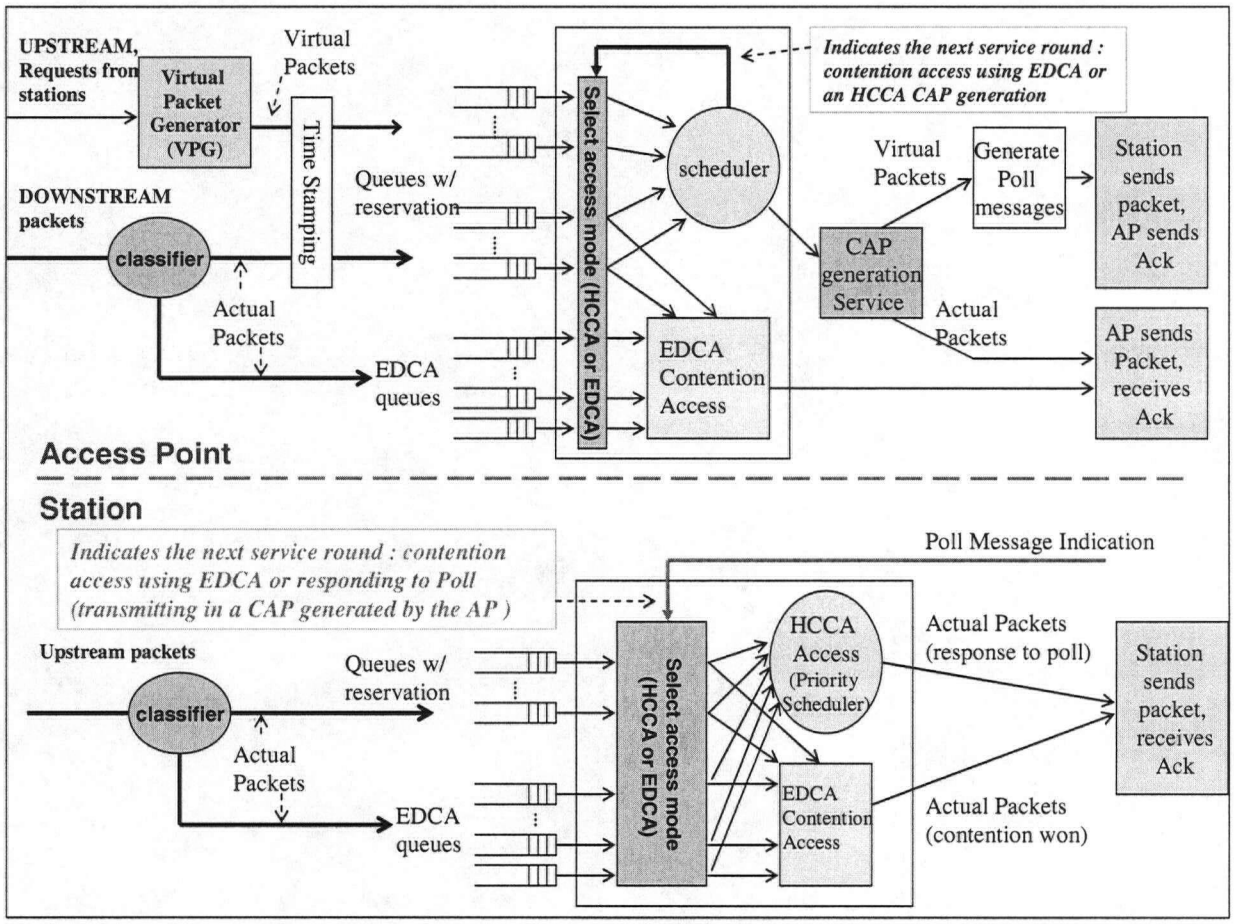

Figure 3-3 Architecture and queuing model of CAPS Access Point and Station

The task of generating virtual packets is performed by a module called Virtual Packet Generator (VPG), as depicted in Figure 3-3. VPG uses control plane requests (explicit through

ADDTS message or implicit through interpreting SIP, [20], calls in higher layers), or traffic pattern estimation to determine the patterns of virtual packets (or flows) that must be generated. For example, for a voice call, a periodic flow of packets similar to the real traffic is generated by the VPG. The generated virtual packets are classified along with actual downstream packets and are queued and scheduled for service based on the algorithm described in the next section.

Packets that are served by the scheduler are treated differently based on whether they are actual or virtual packets. Actual packets are directly transmitted in a downstream CAP, but for virtual packets an upstream CAP is generated by sending a poll message and assigning the appropriate TXOP to the station whose virtual packet is being served.

3.2.2 Scheduling, and Traffic Shaping

Using the hybrid scheduling model enabled by virtual packets, we can use a central queuing and scheduling model in the AP, as depicted in Figure 3-3. The integrated scheduler/shaper module combines EDCA and HCCA operation to achieve both fairness and service guarantee. In all stations (including the AP), the queuing model comprises all queues for flows with reservation (HCCA queues) plus the 4 (or 8) basic EDCA queues for each prioritized access category.

After each transmission or channel busy period, the scheduler examines the queues with reservation (virtual and actual flow queues) and determines whether a queue must be served. In this step only queues whose traffic is conformant to the declared traffic shape are examined. If a queue is found eligible for HCCA service and is selected by the scheduler, it is given controlled access through a CAP generation. But if no queue is found, the scheduler selects the

contention access mode and allows all actual packet queues in the system, including those with non conforming traffic, to contend for accessing the channel using EDCA rules.

When contention is allowed, all queues in the stations will contend for accessing the channel (including the HCCA queues). But in the AP we only allow EDCA queues plus the actual packet HCCA queues to contend; Virtual flows are excluded from contention because their corresponding actual flows in the stations are already involved in contention. The EDCA contention parameters used by contending HCCA queues are chosen locally based on the information collected during session setup.

The operation of CAPS can be divided into three tasks. The first task is admission control and generating virtual packets according to the declared session information. The second task includes time-stamping, pre-shaping and queuing the arriving packets. The third and main task is selecting the packet to be served and controlling the switching between HCCA and EDCA.

Task 1: Generating Virtual Packets & Admission Control

This task processes requests from stations to set up flows for sessions. Admission control rules are applied to determine whether a session can be admitted by the AP. Since admission control is outside the scope of this thesis we do not discuss it here. In fact, any admission control mechanism that works with fair scheduling algorithms can be used. For an admitted uplink session, this process generates virtual packets using the available information. If service interval S_i and average packet size P_i are specified, virtual flows of size P_i bits are generated every S_i seconds. If S_i is not declared, we can use the declared average rate r_i, and generate virtual packets of size P_i every (r_i/P_i) seconds. Note that this process provides bandwidth guarantees to flows specified by their average rate requirements. To provide delay guarantees

in the system, the maximum burst (b_i) size of each flow i must be supplied to the traffic shaper. Limiting the burst size is an essential requirement for providing delay guarantees in any GPS-based schedulers such as weighted-fair queuing and its variants.

One way of increasing the system capacity is to allow bursty transmission through TXOPs and reduce the overhead incurred by poll messages. This is achieved by CAPS by simply using larger virtual packets with proportionally longer service intervals (to keep the average rate constant).

For Applications such as Voice-over-IP where periods of silence and activity exist, a consistent stream of polls to silent stations will be wasteful. To address this issue the VPG must stop sending polls after detecting an empty queue (through the queue size field of the received poll response being set to zero or the more_data bit turned off). The VPG will resume generating VPs as soon as it receives a new frame for the session that arrives through EDCA. If EDCA may cause unacceptable delay the VPG can send polls at a lower rate to inquire about the activity of the voice source.

Task 2: Queuing Packets

Packets that are received by the CAPS scheduler are classified into three groups 1) *virtual packets* for uplink flows with reservations; 2) real packets belonging to downlink flows with reservations; 3) packets with no flow-association and no reservation. The first two types are called HCCA packets in this thesis and are assigned to HCCA queues. For scheduling purposes the length attribute of these packets must be adjusted to account for the different overheads incurred by each type. Virtual packets require an extra poll message at the

beginning of a CAP, so the transmission period for such packets must be increased accordingly.

When a packet without reservation is received, its access category field is examined and the packet is stored in a corresponding EDCA queue. For the HCCA packets, the Traffic Stream ID of the (virtual or real) packet is used to determine its corresponding session queue. Before queuing, the conformance of the arriving HCCA packet to its flow's declared traffic pattern is checked and the packet is properly tagged with an eligibility time indicating when the packet is eligible for HCCA service (section 3.3 elaborates on this issue more). The packets are then time-stamped with start or finish tags according to the algorithm used in the inner scheduler (e.g. SFQ, WFQ or WF^2Q). The packet start and finish times for these inner schedulers (SFQ, WFQ, and WF^2Q) are given by:

$$S_i^k = \max(F_i^{k-1}, V(t)) \tag{3.1}$$

$$F_i^k = \frac{L_i^k}{r_i} + S_i^k \tag{3.2}$$

Where S_i^k and F_i^k are the start and finish timestamps for the kth packet from the ith flow, L_i^k is the adjusted packet length, r_i is the rate assigned to the flow, and $V(t)$ is the *virtual time* function. The virtual time is calculated differently for each inner scheduler. For WFQ and WF^2Q, V(t) represents the progress time of a GPS scheduler that is fed with the packets from these queues and is calculated as :

$$V(t_{j-1}+T) = V(t_{j-1}) + \frac{T}{\sum_{i \in B_j} (r_i/C)} \quad , \quad T \le t_j - t_{j-1}, j = 2,3,... \tag{3.3}$$

where C is the server rate, T is the time between two subsequent events j and j-1 (i.e.. packet arrival or departure) in the GPS system and B_j is the set of backlogged sessions (queues) between these events. For SFQ the virtual time is described in a much simpler way as the start tag of the packet in service at time t. At the end of a busy period $v(t)$ is set to zero (or the last packet's finish time).

Task 3: Scheduling and Traffic Shaping

With packets queued in either HCCA or EDCA queues, the main task of CAPS is to determine which mode of operation should be used and which queue must be served at each service time. A service time occurs after a transmission is completed and the AP senses that medium has been idle for one PIFS duration. At this time the algorithm described in Figure 3-4 indicates whether a CAP for a virtual or actual packet must be generated, or control should be given to EDCA.

The algorithm requires maintaining a queue budget parameter g_i for uplink traffic control. The queue budget parameter keeps track of the lost service time and the available TXOP time for a specific virtual flow at any given service time. Initially, g_i is set to zero; it increases with each transmitted poll, and decreases with each response received. The scheduling algorithm is explained in a two-step pseudo code format depicted in Figure 3-4.

The algorithm assumes that generated virtual flows are conformant to the reservations made during session setup, but actual downlink or uplink flows may not conform to their previously declared pattern. Therefore, traffic shaping and control is performed differently for actual and virtual flows. For uplink flows we only have an estimate of the flow pattern through virtual flow specifications and must wait for the actual packets to arrive before we can apply traffic

shaping. This is achieved through compensation as explained later. For actual downlink flows, we can apply the shaping measures directly to the flows through an eligibility flag that is explained in the next section. The scheduler only serves virtual flows with packets and actual flows with eligible HoL (Head-of-Line) packets. When no such packets are found, control is given to EDCA. Therefore the decision for switching to EDCA is made indirectly through traffic shaping and virtual packet generation processes.

```
Step1: /* Select the queue to serve: */
  {

 /* Find queue i with smallest HoL time stamp, from
the set of all virtual flow queues plus all downlink
HCCA queues with eligible HoL packets.*/
  A1:
    i = find_queue_to_serve()
    /* budget update for Virtual Flows*/
    if ( i Virtual Packet queue)
        g_i = min { b_i , g_i + vp_size }
        goto Step2;
    else if ( i downlink HCCA queue)
        goto Step2; /* actual downlink packet
                       to be served*/
    goto Step2; /* no packet to be served */
} /*end of Step 1*/
```

```
Step2:     /* Determine and apply EDCA or HCCA
operation*/
{
  If (no queue selected in Step1) /* yield to EDCA*/
    exit; /*exit the algorithm till next service round */
  else /*initiate a CAP, HCCA operation*/
  {  If (i: Virtual Packet queue)
        send a poll to queue i's destination;
     else if (i: actual packet queue)
        send the packet in a CAP;
  }
  WAIT for response or timeout;
  If (    data of size L received in response
          to poll from queue i   )
     g_i = g_i - L
  else (timeout or failure)
     do not update g_i;
}
WAIT until next service round; goto Step1;
```

Figure 3-4 Pseudo code for the scheduling task

3.2.3 Implementing the Traffic Shaper

The integrated traffic shaper in the system is needed for downlink actual packets. Since virtual packets are already conformant to a predefined shape (enforced by the VPG), we only need to use the shaper to ensure that actual downlink flows do not exceed their promised HCCA service. This way we make sure that CAPS only assigns the promised service times to HCCA and switches to EDCA for using the remaining capacity. If shapers were not used, mal-behaving downlink flows could take up all the channel capacity and starve the EDCA traffic.

To enforce the shaping decisions on downlink HCCA flows, we add a new time stamp called eligibility_time to each queued packet. Eligibility time is derived based on a token bucket shaper with envelope ($r_i t+b_i$). Upon arrival, each packet is tagged with the time when it becomes eligible (compared to system time). The inner scheduler only looks at HoL packets whose eligibility time is past the system time. However, for EDCA all HoL packets can contend.

There are two options for implementing the shaper for CAPS-WFQ and CAPS-WF^{2}Q. We can either implement the shaper in a separate queue, thus the packets in HCCA queues will all be eligible for scheduling. However when EDCA is active, the shaping queues are also used for contention if their corresponding HCCA queues are empty. The other option is to implement the shaper in the same queue, but use the eligibility_time tag to identify HoL packets eligible for HCCA scheduling. As expected, EDCA contention is applicable to all HoL downlink packets in this case. This method requires that we delay passing the packet arrival event to the GPS emulator for ineligible packets until they reach their eligibility time. Time stamping packets only happens after a packet becomes eligible too. For virtual time calculation the GPS emulator only uses the packet arrival event as an external trigger.

For CAPS-SFQ the shaping can be done in a much simpler way because virtual time is calculated using SFQ events. The scheduling tasks, including the time stamping and update of the virtual time, only apply to packets with eligibility time reached. Thus in each service round the scheduler only acts on HoL packets that are eligible. If no such packet is found the scheduler yields to EDCA, and takes over after EDCA operation completes (or PIFS passes). SFQ is in general much easier to implement than WFQ and WF^{2}Q; the fact that the shaping for

CAPS-SFQ is also very simple increases the advantage of CAPS-SFQ over other CAPS options.

3.2.4 Lost Service Compensation for Uplink Flows

Traffic shaping for uplink flows is mainly done through generating conformant virtual flows. However, in some cases the length of an uplink packet, sent in response to a poll, may be smaller than that of the virtual packet that generated the poll. In this case the budget g_i does not go to zero after receiving the poll response and increases (up to the burst size) by the unused amount of budget. The positive and increased budget for virtual flows is an indication of lost service for uplink flows. This lost service can be compensated in two ways: 1) "Immediate Compensation" in which the entire budget is assigned in one polled-TXOP when the next virtual packet for this queue is served, 2) "Deferred Compensation" for which the TXOP is always assigned based on the length of the virtual packet currently in service and any excess budget is used to generate additional virtual packets for the same virtual flow. Compensation occurs for the flow when these packets are later served.

With immediate compensation a small virtual packet may result in a large TXOP being assigned to the station to compensate for the lost service. We call this case Long Response (or LR). The LR case may result in a large (but still bounded) difference between CAPS operation and the ideal GPS for a short period of time. We analyze this situation later in this chapter.

With deferred compensation, since the TXOP assigned to a station as a result of serving a virtual packet is not derived from the budget parameter but from the virtual packet size, we ensure that the long response case does not happen and the subsequent service disturbance is avoided for other flows, as a result the service guarantees for other flows are still valid.

For deferred compensation, a virtual flow that has a positive g_i can exchange the accumulated budget with additional virtual packets that are then stored in its queue and will get service at the guaranteed rate. The compensation virtual packet is generated when an indication of non-zero queue size is received either through HCCA or EDCA packets from the station. Deferred Compensation is, in effect, similar to retransmitting a virtual packet (poll message) and re-assigning the TXOP until it is properly responded to. This mechanism isolates the compensation for a specific flow from the rest of the flows and enhances service guarantees. It, however, introduces implementation overhead. Therefore, we only use this option when we do not have a good estimation of uplink flows and the bounds on service discrepancy become unacceptably large. The analysis in section 3.3 helps us to make a choice more appropriately.

Note that the budget grows if there is not enough data in the station queue, meaning that at the end of the response TXOP the station queue is empty, so the extra budget should not be re-assigned through generating a virtual packet immediately, and the scheduler must wait until it receives a message from the station with non-zero queue size report. It then creates a virtual packet with the same length (up to the available budget) and stores it at the end of the queue.

3.2.5 Adapting to Wireless Channel

Physical channel impairments in a WLAN result in packet loss and consequently retransmission of packets by the MAC layer. If the quality is consistently low, the operational transmission rate for a station may be reduced as well. Channel impairment issues can be dealt with in many ways. One method is to use a lead/lag model as described in earlier works on single direction schedulers such as those described in [3], [4] or [6]. These models rely on

detecting channel quality beforehand and lending one stations transmission time to another to avoid transmitting in a bad channel. A lead/lag counter is maintained and the stations that are leading in their service will gradually give back service to the lagging stations. Such methods are not usually applicable if good channel estimations are not available. They also cannot be applied when uplink flows are concerned, since AP may not know of stations' conditions. As a result, we rely on the retransmission feature of the MAC and adapt a simpler model of readjustment of scheduling task in order to maintain fairness. If channel monitoring is efficiently possible in WLANs, the lead/lag method can also be used.

To deal with packet loss, the MAC layer can retransmit a packet a few times until it arrives at the receiver or is dropped after n attempts (n must be small enough to avoid causing excessive delay for the entire session). If retransmission happens during a CAP it may disturb the fairness of the scheduler since a station may take longer than expected to transmit the packet. To counter this problem we have several options: the first option is to avoid immediate retransmission and wait until the next service round for this queue. This is automatically achieved for virtual packets by the deferred compensation method discussed above. For downlink packets the HoL packet's time stamps are recalculated as if it was a new packet. This method prevents problems in this flow from disturbing other flows and ensures that service guarantees are still valid. Also, a good side effect is that immediate retransmission on the bad channel is avoided and situation may improve till the next service round.

Since the retransmitted packet will remain eligible for HCCA service, the retransmissions are indeed done at the expense of EDCA traffic, or in other words using the spare capacity of the channel. It is the responsibility of the admission control mechanism to reserve a portion of the channel capacity for dealing with packet retransmission.

Another option to maintain fairness in presence of retransmission is to move the packet that incurred problem to a special queue set up for retransmission (or to an EDCA queue) with separate reservations. This method is similar to Server Based Fair Algorithm (SBFA) described in [23]. This, in effect, isolates the effect of packet loss and retransmission from all other queues, and from the next packets in the same queue.

The re-adjustment of packet time stamps, as described above, must be reflected in virtual time calculation of the inner scheduler as well. Implementing this policy for CAPS-SFQ is very simple as its virtual time is calculated using real events from the scheduler; however for CAPS-WFQ and CAPS-WF^{2}Q, applying the length adjustments to virtual time, though feasible, is computationally expensive because virtual time is calculated from simulating a GPS server.

3.3 Performance Guarantee Analysis

Since CAPS is based on GPS and uses fair queuing algorithms, we expect it to be able to guarantee channel resources for each session. We examine this fact by proving that the difference between CAPS and ideal unidirectional GPS is bounded under different conditions and using different inner schedulers. To examine this point, we analyze the algorithm under worst case scenarios where the order of served packets in CAPS is different from the ideal order of its unidirectional inner scheduler, hence from GPS.

CAPS deviates from the ideal order of a unidirectional inner scheduler in two cases: when immediate compensation is used and the response to a poll message is longer than the corresponding virtual packet (i.e. Long Repose, LR, case), and when a short response is sent in response to a longer virtual packet (i.e. Short Response, SR, case) in both immediate and

deferred compensation modes. If each generated virtual flow exactly matches its corresponding uplink flow (poll response), CAPS behavior is equal to its inner scheduler. In this case all the performance bounds of the inner scheduler are applicable to CAPS as well. But in the case of LR and SR, the order of packets in CAPS and its inner scheduler may be different; as a result new performance bounds may be found for CAPS. In this section we first analyze the LR case for both immediate and deferred compensation options. We show that deferred compensation is indeed the preferred choice when strict performance guarantees are needed. Consequently we analyze the SR case under deferred compensation for several inner scheduler options.

3.3.1 Long Response Case: Immediate and Deferred Compensation

Using immediate compensation, a virtual flow queue may gather a large budget if its virtual packets are responded with short or no packets (null packets) for a long time. Since in immediate compensation the entire budget is assigned in one TXOP in each poll, the actual uplink frames corresponding to virtual frames may be of the maximum allowed size and larger in size than the corresponding virtual frames (LR case); this results in an order of service in CAPS that is different from its ideal inner scheduler and GPS. For such a case, the difference in order and service progress is bounded as will be shown; however, this bound may become very large for the LR case. The SR case is also applicable in the immediate compensation mode.

To examine the deviation of CAPS from GPS due to the LR case, we restrict the analysis of the immediate compensation mode to a situation where virtual packets are chosen smaller than the corresponding uplink packets (LR still happens due to null or lost packets). This way the

SR case does not happen and we can derive the deviation bounds due to LR for the immediate compensation method. Finding this bound is enough for the sake of demonstrating the inefficiency of the immediate compensation method. We argue that since this bound could be very large and increases when the SR case is considered, the deferred compensation method should be used instead of the immediate compensation. To establish this argument we consider the CAPS-WFQ algorithm in this section; a similar analysis is applicable to CAPS-SFQ and CAPS- WF^2Q, and resulting bound are very similar.

The increase of the deviation bound, when the SR case is also considered for immediate compensation, is limited by the contribution of actual packets that are scheduled ahead of a VP larger than its corresponding uplink packet (SR case). The reason is that the LR case only happens for the virtual packet queues. In this case, only the actual packet queues can introduce the SR situation and add to the LR bound that is independently found from virtual packet traffic. Since the SR case results in bounded deviation, as is shown in the next section, the bounds for the immediate compensation increases with at most the bound found in the next section. This fact further confirms the argument that the deferred compensation method is preferred over the immediate compensation method.

To find the difference in the service order of CAPS and GPS, consider a packet from queue k that is scheduled after a number of virtual packets j. If the virtual flows utilize their entire assigned TXOP in response to the short virtual packets, we may face a situation in which many frames from uplink flows j may be served ahead of frame k in CAPS, while in GPS k would finish service before all these packets. For example, for CAPS-WFQ, this order of scheduling is created if several virtual frames from flows j have smaller finish times than frame k. The finish times are calculated using the virtual packet lengths.

With immediate compensation, responses to virtual packets may be as long as the burst size (enforced by the budget parameter), and given that the scheduler works with virtual packet lengths, we may have more virtual packets scheduled before k and after the bursty response. There is also one other situation that can add to the difference between CAPS-WFQ and GPS. This situation is the same scenario mentioned in [18] that describes the inherent difference of WFQ and GPS; one example of this case is when a frame *m* arrives in an empty system and starts service under WFQ, but a short time later a frame *k* arrives (in another queue) and its calculated finish time is less than that of *m*. Since *m* has already started the service, *k* must wait until the end of service for *m*. Note that this situation may only happen for maximum one packet *m*. Given the described situation we can combine this case with the case where several small virtual packets from queues *j* are scheduled ahead of *k* and after *m*, but their actual frames are served after *k* in GPS. Considering the above worst case scenario the following theorem can be proved:

***Theorem 1*:** if t_i, and u_i denote the finish time for frame *i* in CAPS-WFQ and GPS respectively, the following inequality holds for frame *k* (as described above), if immediate compensation is used and long response case is incurred:

$$t_k - u_k \leq \frac{\sum_{j \in V}(b_j)}{R} - \frac{\sum_{j \in V}(V_j^{\min})}{R} + \frac{L_{\max}}{R} \tag{3.4}$$

L_{max} is the maximum packet length in the system, R is the channel rate, V is the set of all virtual flow queues, and V_j^{min} is the minimum virtual packet length of flow *j*. We also assume that the maximum response size to any virtual packet from flows *i* is bounded by the burst size b_i (according to immediate compensation rules).

Proof: denoting the amount of traffic served from queue i as S_i, and from all queues as S, we know that using immediate compensation the maximum amount of traffic that is served in CAPS-WFQ from each virtual queue j between the end of frame m (t_m) and beginning of frame k (i.e., $t_k - L_k/R$) includes: 1) a burst size response to a virtual packet (b_j) 2) sum of normal size responses (W_j) to the rest of virtual packets scheduled before k but after the virtual packet that resulted in the bursty response. Thus we have:

$$S_j(t_1) - S_j(t_k - L_k/R) \leq W_j + b_j \tag{3.5}$$

Also, denoting as X the sum of all traffic from downlink packets served in CAPS-WFQ between m and k (thus served between t_m and $t_k - L_k/R$), and taking a sum over all virtual flows traffic we have the following (V is the set of all VP queues, except queue k):

$$S(t_k - L_k/R) - S(t_m) \leq \sum_{j \in V} (W_i + b_j) + X \tag{3.6}$$

And since all packets are served at rate R, we have:

$$t_k - L_k/R - t_m \leq \frac{X + \sum_{j \in V} (W_i + b_j)}{R} \tag{3.7}$$

For the flows j virtual packets to have been scheduled before k by CAPS-WFQ, they must have all arrived and departed before u_k in the simulated GPS system, this includes virtual packets that resulted in a bursty response as well as those that incurred normal size responses (W_j). To conceive the worst case we can assume the smallest sizes , i.e., V_i^{min}, for the virtual packets that resulted in bursty response. Therefore, knowing that frame m arrives (at $t_m - L_m/R$) before other packets that contribute to the sum of traffic in (3.6) and knowing that k finishes after all these packets in GPS we have:

$$u_k \geq \frac{\sum_{j \in V}(V_j^{\min} + W_j) + X + L_k}{R} + (t_m - \frac{L_m}{R}) \tag{3.8}$$

Assuming maximum possible length for m, the theorem follows from deducting (3.8) from (3.7). Note that in (3.4) all the values for packet lengths and burst sizes already include the adjustments for MAC operation overhead (i.e. polling and acknowledgement). **Q.E.D.**

Expression (3.4) shows that the difference between service time in CAPS-WFQ and GPS is bounded (similar expressions can easily be found for SFQ and WF^2Q). However, it can become large if the burst sizes of virtual flows are large. We can also show that the backlog of each session under CAPS is more than GPS by a bounded amount. And since GPS is an ideal system, we will have bounded backlog for any session with reservation in CAPS. Since backlog is also the difference between arrival and service curves, it is enough for our purpose to show that the difference between served traffic in CAPS and GPS is bounded.

***Theorem 2*:** For any given time τ the difference between the amount of served traffic in CAPS-WFQ with immediate compensation (denoted $\hat{S}_j(0,\tau)$) and GPS (denoted $S_j(0,\tau)$) is bounded by the following:

$$S_j(0,\tau) - \hat{S}_j(0,\tau) \leq \sum_{j \in V}(b_j) - \sum_{j \in V}(V_j^{\min}) + L_{\max} \tag{3.9}$$

Proof: Let's assume that a packet of size L that finishes service at time τ in GPS, completes service at $t+L/R$ in CAPS. Since Packets are served in the same order in both systems (assuming all flows are conformant) we have:

$$S_j(0,\tau) = \hat{S}_j(0, t+L/R) \tag{3.10}$$

If for simplicity we rewrite *Theorem 1* as t_i-$u_i \leq A$, we will have: $(t+L/R) - A \leq \tau$. Also, from (3.10) we have:

$$S_j(0,t+L/R-A) \leq S_j(0,\tau) = \hat{S}_j(0,\ t+L/R) = \hat{S}_j(0,\ t)+L \tag{3.11}$$

Since we know that the slope of S_j is at most R:

$$S_j(0,t+L/R-A) \leq S_j(0,t)+L-A.R\ ; \tag{3.12}$$

deducting (3.11) from (3.12), we have $S_j(0,t)-\hat{S}_j(0,\ t) \leq A.R$, and the theorem follows. **Q.E.D.**

With *Theorems 1* and *2* we show that CAPS with immediate compensation performs different from GPS by a bounded amount, thus confirming that it can indeed provide fair and guaranteed services, as is possible with GPS and WFQ. However, as is seen from (3.4) and (3.9), in certain situations we may encounter a large deviation from GPS operation. This is the case when the difference of the sum of allowed burst sizes and minimum VP sizes amounts to a large value. In this case the algorithm provides fairness in a longer term than is possible by its inner scheduler. For such situations, we must use deferred compensation which has higher implementation complexity but eliminates the LR case altogether and results in shorter term fairness.

With deferred compensation the length of the frame that is sent in response to a poll is always equal or less than that of the virtual packet that generated the poll. This means that the LR case is in fact eliminated. In certain cases where we have virtual packet sizes that match the corresponding uplink packet size, we can show that deferred compensation can provide delay bounds equal to that of an ideal unidirectional scheduler, even if some packets are not present and polls are not responded to. For example, for CAPS-WFQ, the worst case situation

that has been described earlier reduces to the case where only one long packet may be served ahead of its order in WFQ if it starts service before other smaller packets arrive. This situation which is in fact similar to the worst case in WFQ system results in the following bounds:

$$t_k - u_k \leq \frac{L_{\max}}{R} \tag{3.13}$$

The above expression directly follows from the proof from *Theorem 1* with the knowledge that the service orders in CAPS strictly follows the finish time calculations based on GPS (except for the explained worst case packet m). Consequently we can revisit theorem 2 and derive the following inequality for the deferred compensation case under the same conditions:

$$S_j(0,\tau) - \hat{S}_j(0,\tau) \leq L_{max} \tag{3.14}$$

Expression (3.14) is simply proved by following the proof for (3.9) and replacing A with L_{max}.

Deferred compensation eliminates the LR case and can considerably improve the bounds on backlog and delay in worst case situations. Therefore we argue that the implementation overhead of deferred compensation is acceptable. Given this argument we assume that deferred compensation is used with CAPS and continue the analysis of CAPS in SR case under this assumption.

3.3.2 Performance Analysis of CAPS with Deferred Compensation

In this section we examine the deviation of CAPS operation from its inner scheduler in the SR case and derive the performance bounds for three inner schedulers, WFQ, WF^2Q, and SFQ. As mentioned before, the LR case is eliminated when deferred compensation is used. It is also important to note that the algorithm corrects the projected start and finish times of the next packets in the queues after detecting a SR case, preventing the propagation of the SR case.

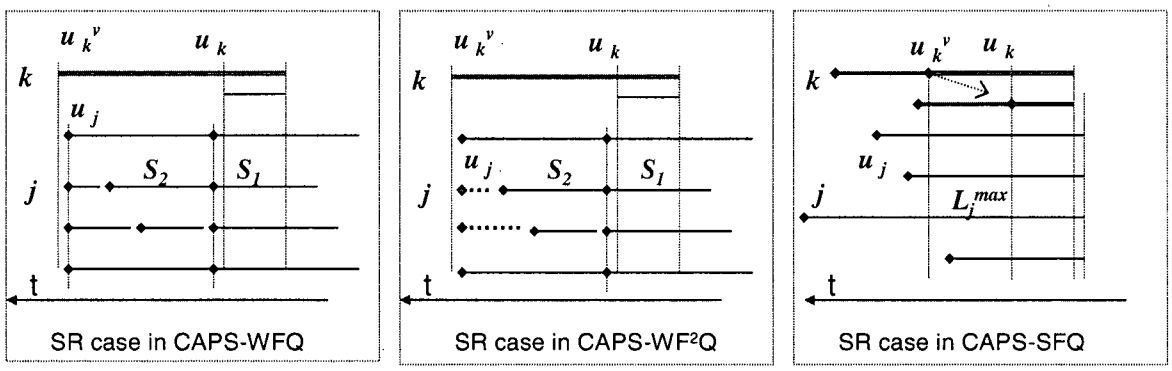

Figure 3-5 SR case for WFQ, WF²Q and SFQ

SR case for CAPS-WFQ

A worst case scenario for CAPS-WFQ with short response situation happens when a long virtual packet with finish time u_k^v is scheduled for a short uplink packet with ideal finish time of u_k under GPS. This means that all the packets j with GPS finish times $u_k<u_j<u_k^v$ were supposed to finish service after k in GPS, but under CAPS-WFQ they are sent ahead of k since $u_j < u_k^v$. With these assumptions we can now prove the following theorem:

***Theorem 3*:** if t_k, and u_k denote the finish time for frame k in CAPS-WFQ and GPS respectively, the following inequality holds for frame k, if deferred compensation is used:

$$t_k - u_k \le \frac{L_{\max}}{R} + \frac{\sum_{j\in Q, j\neq k} r_j}{R}\left(\frac{L_k^v - L_k}{r_k}\right) \quad Q\text{: the set of all queues} \tag{3.15}$$

Proof: To picture the worst case situation, we consider a busy period in which all queues in the system are backlogged for the duration of time in which CAPS service order is different from GPS finish times. Also consider packet i to have been the last packet served before k with $u_i<u_k$ (thus served in correct WFQ order). The maximum difference happens when we consider that for the duration between u_k and u_k^v all other queues always have packets for transmission. To find the service deviation we divide the packets, which are sent ahead of k in CAPS but finish after k in GPS, into two sets of W1 and W2. W1 includes packets j that start service in

GPS after i and before k and finish after u_k with a maximum GPS finish time of $u_j^{\max}$. W2 includes packets that start and finish between u_k and u_k^v.

Without loss of generality, we can divide packets in W2 into smaller packets and create a set W2′ such that its packet have a GPS finish time of $u_j^{\max}$. Denoting the amount of traffic in W2 as W_2, the packets in W2′ contain αW_2 bits, where $\alpha<1$. If we consider a hypothetical WFQ system including all the packets in W1 and W2′, as well as a hypothetical packet k' with $u_k^{'} = u_j^{\max} + \varepsilon > u_j^{\max}$, the behavior of this system is equivalent to CAPS until the last packet in W1 and W2′. Knowing that packet k' is served last in the hypothetical system, we can use the property of WFQ and write:

$$t_k^{'} = t_i + \frac{W_1 + aW_2 + L_k^{'}}{R} \le u_j^{\max} + \varepsilon + \frac{L_{\max}}{R} = u_k^{'} + \frac{L_{\max}}{R} \tag{3.16}$$

On the other hand, we can calculate the GPS finish time of k' from: $u_k^{'} = u_k + (\beta W_1 + aW_2 + L_k^{'} - L_k)/R$, where $\beta < 1$ and $\beta.W_1$ is the portion of W1 traffic (W_1) that is served by GPS after u_k. Now, considering that packet k finishes after all W1 and W2 packets in CAPS, we have $t_k = t_i + \frac{W_1 + W_2 + L_k}{R}$, which can be combined with the above equations to write:

$$t_k - u_k \le (\beta W_1 + W_2)/R + l_{\max}/R. \tag{3.17}$$

From inequality (3.17) we see that only the part of W1 and W2 traffic that is served in GPS after u_k is affecting the deviation of CAPS from GPS. Therefore, as an alternative, we can divide the packets in set W1 to smaller packets and create a set S1 with packets j that have $u_k < u_j < u_k + \varepsilon$, and a set S2 including packets that start and finish between u_k and u_k^v. Figure 3-5 shows these sets. Notice that (3.17) indicates that only packets in S2 contribute to the

deviation of CAPS from GPS. The size of S2 traffic (i.e. S_2) is in fact equal to $\beta W_1 + W_2$ in (3.17) and is found as:

$$S_2 \leq \sum_{j \in Q, j \neq k} \{ r_j \cdot (\frac{L_k^v - L_k}{r_k}) \} \tag{3.18}$$

Combining (3.18) and (3.17), and knowing $S_2 = \beta W_1 + W$, we obtain (3.15) and prove the theorem. As an alternative, one can also find (3.15) by considering a hypothetical set S1″ that includes S1 and a packet k'' with $u_{k''} = u_k + \varepsilon$. Since the set S1 includes all packets j that have $u_k < u_j < u_k + \varepsilon$, serving packets in S1″ in WFQ order, is equivalent to CAPS-WFQ serving S1 and packet k. Therefore, we can use the property of WFQ, in terms of its difference with GPS, and write the following for packet k'' from the set S1″:

$$t_{k''} - u_{k''} \leq \frac{L_{\max}}{R} \quad \Rightarrow \quad t_{k''} - u_k \leq \frac{L_{\max}}{R} + \varepsilon \tag{3.19}$$

Knowing that packet i was served before packets in set S1 and packet k'', and given that packet k'' is of length L_k, we have $t_{k''} = t_i + (S_1 + L_k)/R$. Combining with (3.19) we rewrite this equation as:

$$t_i - u_k \leq \frac{L_{\max} - S_1 - L_k}{R} + \varepsilon \tag{3.20}$$

Since packet k in CAPS-WFQ is served after all packets in S1 and S2 and packet i, and the service rate is R, using (3.18) we find the finish time of k as:

$$t_k = t_i + \frac{S_1 + S_2 + L_k}{R} \leq t_i + \frac{S_1 + L_k}{R} + \frac{\sum_{j \in Q, j \neq k} \{ r_j \cdot (\frac{L_k^v - L_k}{r_k}) \}}{R} \tag{3.21}$$

Deducting (3.21) from (3.20) and choosing ε close to zero we obtain (3.15). **Q.E.D.**

The above proof shows that we only need to find the amount of traffic in S2 to calculate the deviation of CAPS from GPS. This fact is later used in this section when we consider the WF^2Q algorithm. From expression (3.15), we see that although the deviation between GPS and CAPS-WFQ is bounded, it may become large if r_k is small or the difference of virtual and actual packet sizes is large. If more precise knowledge of the uplink flows is available and used in virtual packet generation, this bound will become smaller.

SR case for CAPS-WF^2Q

WF^2Q uses the same finish times as in WFQ; however, when scheduling packets according to their finish time, it only considers those packets that have already started service in the corresponding GPS at the scheduling moment. This mechanism positively affects the service difference bounds for CAPS in some situations.

The worst case scenario for CAPS-WF^2Q is more or less the same as in CAPS-WFQ, except for the packets that are served during (u_k, u_k^v) in GPS. These packets, although have $u_j < u_k^v$, may or may not have started service at the moment when the virtual packet k is eligible for service (Figure 3-5). If these packets have started service under GPS, the service difference bound for CAPS-WF^2Q is exactly like CAPS-WFQ, described in (3.15). Lower bounds are possible if we have further knowledge of the packet sizes for queues j.

Similar to the proof in theorem 3, we consider a set S1 whose packets j have finish times: $u_k < u_j < u_k + \varepsilon$. Also, consider a packet i as the last packet that was served in correct WF^2Q order. After this packet is served, at t_i, CAPS can be either ahead of GPS for this packet ($t_i < u_i$), or be lagging behind at most L_{max}/R according to [21], i.e., $u_i < t_i < u_i + L_{max}/R$. When CAPS is lagging at t_i, the GPS start time of the other packets are going to be less than the current CAPS

time of t_i, thus the packets are eligible. To conceive the worst case, we want to have more packets from queues j eligible for service at time u_j, thus we assume that CAPS was lagging at t_i and continues to be behind GPS when the last packet j is served from set S1 (CAPS finish time of the last packet of set S1 is $t_j > u_j$).

From theorem 3 we know that the traffic in S1 does not increase the deviation of CAPS and GPS. So we find the traffic served during (u_k, u_k^v). At the beginning of this period, when all the packets in S1 have been served, the CAPS service progress time (T_{CAPS}, defined as the time when CAPS finishes service for the last packet in S1) is assumed as lagging behind GPS finish time for these packets (i.e., $T_{GPS} = u_j$), meaning that GPS has progressed further at this point (due to earlier serving an L_{max} out of the GPS order for another queue) and at least one packet from each queue j is eligible. Notice that T_{CAPS} is t_i plus the time it takes to serve packets in set S1 ($T_{CAPS} = t_{i+} S_1/R$). Similar to the proof of theorem 3, assume a hypothetical set S1' that includes S1 and a packet k' with $u_k' = u_k + \varepsilon$. We know that packets in S1' being served in WF^{2}Q order, is equivalent to CAPS-WF^{2}Q serving S1 and packet k. Therefore, we can use the property of WF^{2}Q, in terms of its difference with GPS (refer to [21]), and write the following for packet k' from the set S1':

$$t_k' = t_i + \frac{S_1 + L_k}{R} = T_{CAPS} + \frac{L_k}{R} \le u_k' + \frac{L_{\max}}{R} \le u_j + \frac{L_{\max}}{R} + \varepsilon = T_{GPS} + \frac{L_{\max}}{R} + \varepsilon \qquad (3.22)$$

Inequality (3.22) means that the CAPS scheduler progress time is behind the GPS progress time, thus a set of packets from all queues may be served ahead of k, adding to the difference between CAPS and GPS. To worsen the situation we assume that queues j have (infinitesimally) small packets like a fluid system; these packets are served at rate $\sum_{j \in Q, j \neq k} r_j$ in GPS, and rate R in CAPS, thus CAPS advances faster and may reach and lead ahead of GPS

(finish time of a packet in CAPS be equal or less than its GPS finish time), making packets from queues j ineligible, and allowing packet k to be serviced. We also note that just before the moment when CAPS reaches GPS, one more packet can be served. We denote the length of this packet as L^l. Considering the amount of traffic that can be served in (T_{CAPS} -T_{GPS}) as S_s, we know that the maximum traffic that can be served between u_k and $u^v{}_k$ is:

$$S_m = \min\{S_s + L^l, (\sum_{j\in Q, j\neq k} r_j)\cdot(\frac{L_k^v - L_k}{r_k})\} \tag{3.23}$$

and S_s is found as follows using inequality (3.22) and choosing ε close to zero:

$$T_{CAPS} + \frac{S_s}{R} = T_{GPS} + \frac{S_s}{\sum_{j\in Q, j\neq k} r_j} \qquad \Rightarrow \qquad S_s \leq \frac{\sum_{j\in Q, j\neq k} r_j}{R - \sum_{j\in Q, j\neq k} r_j}(L_{\max} - L_k) \tag{3.24}$$

The maximum value of S_s is found when the equality holds in the above expression. The packet with length L^l must have a finish time less than $u^v{}_k$ to be counted in CAPS and GPS difference. Since this packet is sent at rate R, we can calculate its maximum allowable size as 'R multiplied by allowed time':

$$L^l = \max\{0\ ,\ R\cdot(u_k^v - u_k - \frac{S_s}{\sum_{j\in Q, j\neq k} r_j}) = R\cdot(\frac{L_k^v - L_k}{r_k} - \frac{S_s}{\sum_{j\in Q, j\neq k} r_j})\} \tag{3.25}$$

Note that $L^v{}_k$ may be larger than the L_{max} that is chosen from queues other than the one packet k belongs to. Having found S_s and L^l we can conclude that the maximum amount of traffic served ahead of k is given by $S_1 + S_w$, where S_w is found as:

$$S_w = \min\{S_s + L^l,\ \sum_{j\in Q, j\neq k} r_j\cdot(\frac{L_k^v - L_k}{r_k})\} \tag{3.26}$$

From (3.26) and (3.22) and knowing that $t_k - t_i = \frac{S_w + S_1}{R} + \frac{L_k}{R}$ we have indeed proved the following theorem:

***Theorem 4*:** if t_i, and u_i denote the finish time for frame i in CAPS-WF^{2}Q and GPS respectively, the following inequality holds for frame k if deferred compensation is used:

$$t_k - u_k \leq \frac{S_w}{R} + \frac{L_{\max}}{R} \tag{3.27}$$

The bound in (3.27) is indeed less than or equal to what is expressed in (3.15); thus, it may well depend on the flow parameters to determine whether using WF^{2}Q is helpful to reduce the effects of imprecise virtual packet generation. In the next subsection we study SFQ and find that it may in fact be a better choice for reducing the effect of virtual/actual flow mismatch.

SR case for CAPS-SFQ

For CAPS-SFQ with deferred compensation, the short response case can be contained and its effects eliminated. In fact, we prove that the following theorem holds for CAPS-SFQ.

***Theorem 5*:** if t_i, and u_i denote the finish time for frame i in CAPS-SFQ and GPS respectively, the following inequality holds for frame k if deferred compensation is used:

$$t_k - u_k \leq \frac{\sum_{j \in Q, j \neq k} L_j^{\max}}{R} + \frac{L_k}{R} - \frac{L_k}{r_k} \tag{3.28}$$

Proof: We first explain that we can eliminate the effect of SR case on CAPS-SFQ. To see this point, notice that in SFQ packets are time-stamped with start and finish times as follows for the i'th packet of flow k that arrives at time A_i^k:

$$S_i^k = \max\{V(A_i^k), F_{i-1}^k\}; \qquad F_i^k = S_i^k + L_i / r_k \tag{3.29}$$

where L_i denotes the packet size, and V(.) is the system virtual time, taken to be the start tag of the packet currently being served. As a modification to SFQ, in CAPS-SFQ, if a short response

occurs, the finish time of the current virtual packet should be adjusted to reflect the actual size of the uplink packet. This means that the next virtual packet backlogged in the queue will have the correct start time tag. Since in SFQ packets are served in order of start time, the SR case does not change the order of service in CAPS-SFQ from the inner ideal unidirectional SFQ scheduler. As a result, we know that the difference between CAPS-SFQ and GPS is the same as the difference between SFQ and GPS. Given that in GPS a packet i that arrives at HoL is served after L_i/r_k . And knowing that in worst case scenario for SFQ ([22]) such packet may be served after maximum packets from all other queues we can directly derive (3.28). **Q.E.D.**

From the bounds found in this section we see that the delay bound of WFQ and WF^2Q worsens in WLANs, compared to their bounds found for an ideal unidirectional scenario. This is not the case for SFQ. As it is shown in [22], the delay bound of ideal WFQ or WF^2Q is better than SFQ only for high bitrate flows, and in schedulers with a large number of session. For low bitrate flows, SFQ provides a much better delay bound. Given the bounds found above, this advantage of SFQ is strengthened with the increased deviation of WFQ or WF^2Q from GPS in non-ideal cases of a WLAN. These advantages, along with the ease of implementation and adoption of retransmission policies make CAPS-SFQ the best choice for our framework.

3.4 Performance Evaluation

To evaluate the service guarantee features of CAPS and measure its performance under different conditions, we conducted several experiments using an OPNET simulator that we developed for 802.11e WLANs. The applications that were selected for the experiments were the real-time applications that are most likely to use QoS services of an 802.11e WLAN, i.e.,

real-time voice and video conferencing. We assumed an 802.11b physical layer for our experiment. We compared the results from CAPS operation with those that are achieved by the EDCA mechanism. Some of the performance gains such as the total throughput increase of our algorithm will be similar to that of other HCCA algorithms such as the TGe scheduler and the works presented in [13] and [14]. However, the TGe scheduler has already been shown (in [13] and [14]) to be very inefficient for VBR traffic and is unlikely to be deployed in real products, thus we do not re-evaluate it here. Also, a direct comparison with HCCA algorithms presented in [13] and [14] would only be possible if implementation details of such algorithms and their associated admission control mechanisms were available. However, crucial implementation details are not presented in [13] and [14], making their implementation subject to reader's interpretation. These facts encouraged us to focus on EDCA which is the most likely contender with methods such as CAPS.

It is important to note that we considered absolute worst case scenarios in the previous section in order to prove the ability of CAPS to achieve fair and guaranteed services similar to GPS. In practice, these worst case scenarios do not happen very often, and we can achieve much better average delay and bitrate provision performance using CAPS. Through our experiments we demonstrate advantages of CAPS in providing guaranteed throughput, protection from background and from same class traffic, and increase in system capacity for multimedia applications.

The results in this section are valid for all three options of CAPS-WFQ, CAPS-SFQ and CAPS-WF^{2}Q, unless stated otherwise. In fact, we see that in most cases the worst case scenarios of the previous section do not occur easily, and the average or near worst case behavior of all three options of CAPS are very similar. To get more insight, we conducted an experiment in which the maximum delay for one 500Kbps video flow was measured as the

number of HCCA flows (each 500Kbps, with 2000B packets) increased. We simulated the SR case by generating 4000B virtual packets for video packets of average size 700B. The results, depicted in Figure 3-6, show that the near worst case behavior only happens when the load reaches the limits of available capacity; otherwise the behavior is very similar for CAPS-WFQ, CAPS-WF^{2}Q, and CAPS-SFQ.

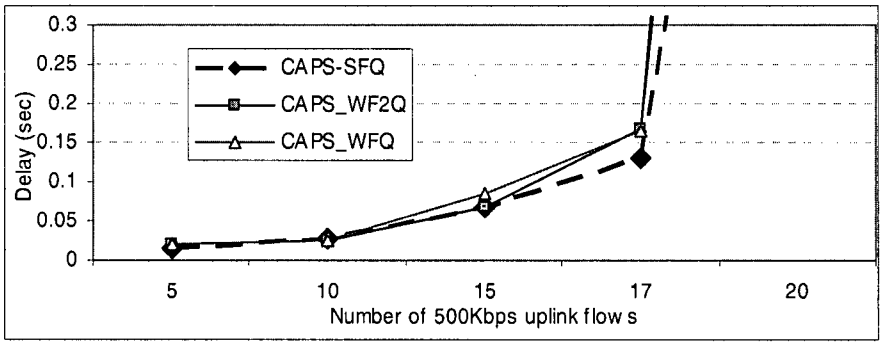

Figure 3-6 Max Delay observed for different inner schedulers

To examine and compare the delay performance of CAPS and EDCA we considered a home WLAN scenario in which an uplink video session co-existed with a heavy downlink traffic of 5Mbps. We also considered 2 (and 6) stations sending uplink background traffic of 200Kbps. The video was a CIF size H.264 foreman video with a bitrate of around 500Kbps. The cumulative distribution function of the measured delay for the video session is depicted in Figure 3-7. This figure shows that CAPS has a significantly better delay pattern than EDCA. For example if the deadline is set to 100msec, more than 10 to 20% of the packets in EDCA will miss their deadline, although the average delay of EDCA is far below this deadline. This experiment, which is based on real life scenarios, confirms that EDCA is not suitable for real time multimedia applications (for more details see Appendix B).

In our next experiment we demonstrate the ability of CAPS to guarantee a certain bitrate. For this purpose, we observe the achieved throughput of a CAPS flow with 100Kbps

reservation and other flows with no reservation and EDCA access. All the stations in this experiment are data sources with rate 200Kbps (200Bytes packets, with exponential inter-arrival, and highest EDCA priority). In different steps of the experiment we increased the number of stations to increase the load until the WLAN enters saturation. Figure 3-8 depicts the result. As is seen, at low loads all stations can get their 200Kbps traffic through. As the load increases, flows without CAPS service suffer from collision and problems of contention access, while the flow with reservation maintains at least its guaranteed rate under CAPS (i.e. 100Kbps in this example).

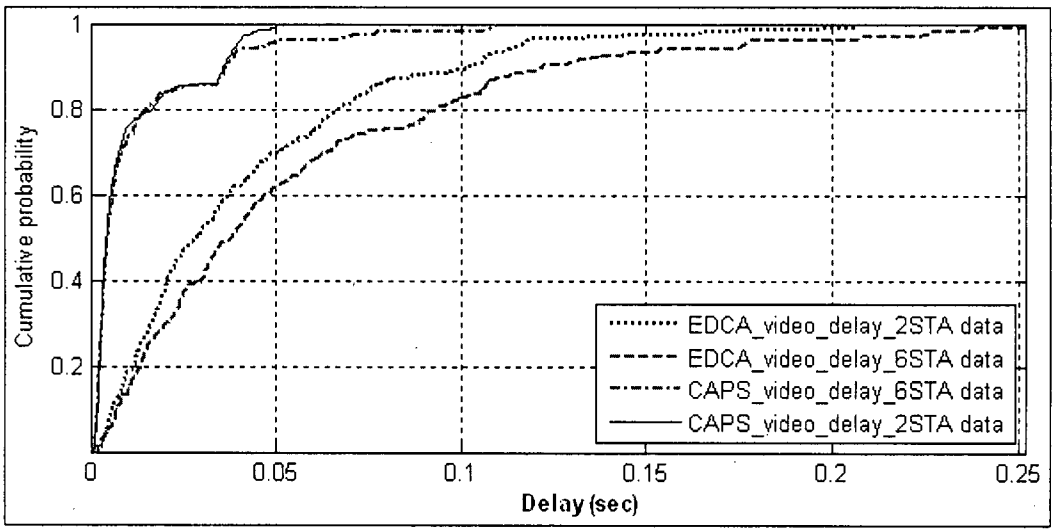

Figure 3-7 CDF of delay for an uplink Video flow (CAPS vs EDCA)

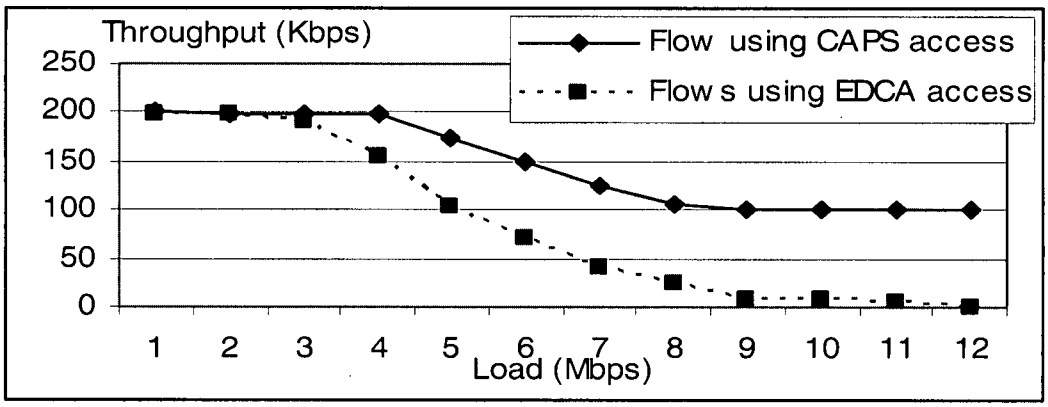

Figure 3-8 CAPS ability to guarantee bitrate (for all CAPS options)

To examine the ability of CAPS to increase WLAN capacity we evaluated the delay performance of a voice only WLAN and measured the number of G.711 voice flows that could be supported in an 11Mbps WLAN (e.g. in an 802.11b PHY). The voice flows were 64Kbps (80Kbps with RTP and IP overhead) with a rate of 50 packets per second. We increased the default minimum and maximum contention window sizes for EDCA voice access category to let it accommodate more stations. Without this increase EDCA would fail very quickly. We also allowed larger virtual packets, but with longer service intervals to allow for bursty operation (EDCA by default uses bursty operation for voice category). In this experiment no background or data traffic was present.

As is shown in Figure 3-9, when CAPS is used the average and maximum delay for voice sessions remains controlled for a higher number of voice sessions, demonstrating a substantial capacity boost despite the significant overhead of poll messages. For example, if the maximum specified delay for voice sessions is restricted to 100 ms within the WLAN, EDCA can admit no more than 20 flows while CAPS can serve more than 45 voice flows (CAPS-WFQ and CAPS-WF2Q performs identically, but slightly different from CAPS-SFQ).

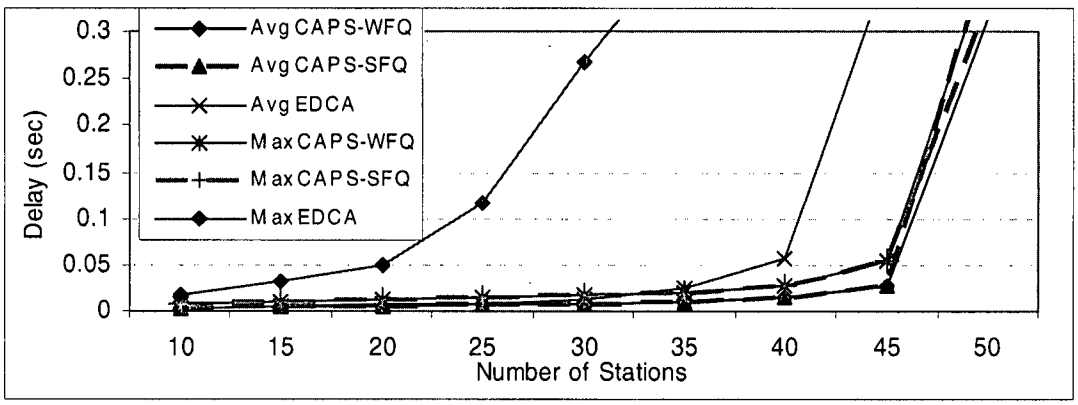

Figure 3-9 A Voice only WLAN, Maximum and Average delay for (CAPS vs. EDCA)

In another set of experiments we considered a 256Kbps H.264 Video traffic and observed the delay its packets incurred as we increased the background traffic of all classes (including

voice). Although the video was a variable bitrate media which caused SR case, we still achieved a very controlled delay performance using CAPS (all options) compared to EDCA. The results shown in Figure 3-10 confirm that the flow is protected from the background traffic.

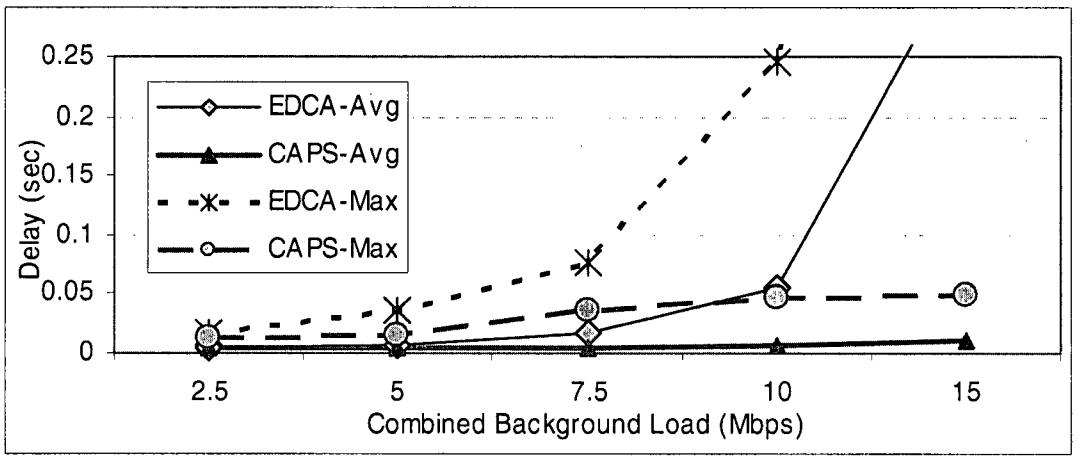

Figure 3-10 Delay of a single video session as background traffic increases

In the last set of experiments we examined the per-session services of CAPS versus the aggregate services of EDCA. We considered a WLAN in which the background data traffic was fixed but the number of video flows increased in the channel. We observed the delay incurred by a low bitrate video such as a 64Kbps cell-phone size video, while some higher bitrate video flows were being added to the WLAN. As shown in Figure 3-11, CAPS is able to protect the low bitrate stream against higher bit-rate flows from the same traffic class. Meanwhile, EDCA fails to isolate this flow and unfairly allows the higher bitrate flows to degrade the quality of lower bitrate flows. This deficiency in EDCA is because of its inherent aggregate service differentiation which fails to achieve flow isolation within the same traffic class.

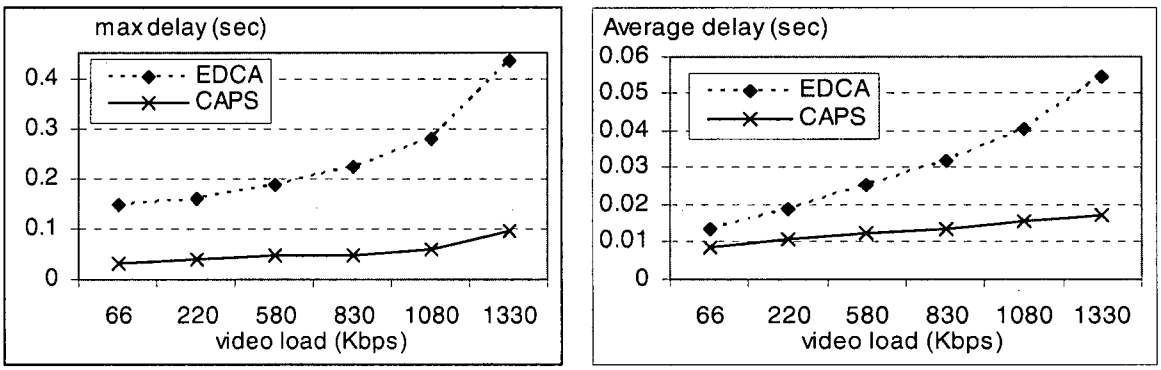

Figure 3-11 Protection from same class traffic: per flow vs. aggregate QoS

3.5 Concluding Remarks

Providing per-session QoS in WLANs requires special measures that are addressed by our proposed CAPS framework. The proposed design enables centralized scheduling of upstream and downstream flows in the access point. It also facilitates on demand use of controlled access phases under HCCA, while allowing EDCA operation for the remaining capacity. This feature allows very efficient service guarantee for time sensitive flows even under heavy traffic conditions. In particular, applications such as real-time voice and video over WLAN will greatly benefit from this design because of the inherent similarity of their operational environment to the cases that are targeted by this design.

The CAPS framework can be used in similar shared medium environments such as IEEE 802.16. Integrating the presented design with the power management features of 802.11e is also an open issue to be further studied. The flexibility provided by the combined uplink/downlink scheduling of CAPS can also be used to employ cross-layer optimizations in the MAC using information from application and physical layers.

3.6 References

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.

[2] IEEE Standard 802.11e/ Amendment 8, "Medium Access Control (MAC) Quality of Service (QoS) Enhancements", July 2005.

[3] T. Nandagopal , S. Lu and A. V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms", Wireless Networks, vol. 8 , Issue 2/3, pp. 231 - 247, March-May 2002.

[4] S. Lu, T. Nandagopal and V. Bharghavan, "Fair scheduling in wireless packet networks", ACM MOBICOM'98, October 1997.

[5] S. Lu, V. Bharghavan and R. Srikant, "Fair scheduling in wireless packet networks", IEEE/ACM Trans. on Networking, vol. 7 , Issue 4, pp. 473 - 489, August 1999.

[6] T.S. Ng, I. Stoica and H. Zhang, "Packet Fair Queueing Algorithms For Wireless Networks With Location-Dependent Errors", Proc. of IEEE INFOCOM '98, vol. 3, pp. 1103–1111, March 1998.

[7] A. Boukerche, and T. Dash "Performance Evaluation of a Generalized Hybrid TDMA/CDMA Protocol for Wireless Multimedia with QoS Adaptation", Computer Communication, vol. 28, pp. 1468-1480, 2005.

[8] A. Boukerche, T. Dash, and C. M Pinotti "Performance Analysis of a Novel Hybrid Push-Pull Algorithm with QoS Adaptations in Wireless Networks", Performance Evaluation, vol. 60, pp. 201-221, 2005.

[9] Y. Yuan, D. Gu, W. Arbaugh, J. Zhang, "Achieving Fair Scheduling Over Error-Prone Channels in Multirate WLANs" ,Wireless Networks, Communications and Mobile Computing, 2005 International Conference on, vol. 1, pp. 698–703, June 2005.

[10] Z. Jiang, N.K. Shankaranarayana , "Channel Quality Dependent Scheduling for Flexible Wireless Resource Management", Global Telecommunications Conference, 2001, GLOBECOM '01, IEEE , vol. 6, pp. 3633 – 3638, Nov. 2001.

[11] Q. Ni, L. Romdhani and T. Turletti ,"A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN", Wiley Journal of Wireless Communication and Mobile Computing (JWCMC), John Wiley and Sons Ltd., vol. 4, Issue 5, pp. 547-566, 2004.

[12] Q. Ni, "Performance Analysis and Enhancements for IEEE 802.11e Wireless Networks", IEEE Networks, vol. 19, Issue 4, pp. 21 - 27, July-Aug. 2005.

[13] P. Ansel, Q. Ni, and T. Turletti, "An efficient scheduling scheme for IEEE 802.11e," WiOpt'04: Modeling and Optimization in Mobile, AdHoc and Wireless Networks, 2004.

[14] A. Grilo, M. Macedo and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks", IEEE Wireless Communications, vol. 10, Issue 3, pp. 36-43, June 2003.

[15] R.S. Ranasinghe, L.L.H. Andrew, D.A. Hayes, D. Everitt, "Scheduling Disciplines for Multimedia WLANs: Embedded Round Robin and Wireless Dual Queue", Communications, 2001. ICC 2001. IEEE International Conf. on , vol. 4, pp. 1243 - 1248, 11-14 June 2001.

[16] W. Pattara-Atikom, P. Krishnamurthy, S. Banerjee, "Distributed Mechanisms For Quality Of Service In Wireless Lans", Wireless Communications, IEEE, vol. 10 , Issue: 3, pp:26 – 34, June 2003.

[17] Y. Pourmohammadi Fallah and H. Alnuweiri, "A Controlled-Access Scheduling Mechanism For QoS Provisioning In IEEE 802.11e Wireless LANs", ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks Q2SWinet '05, pp. 122 - 129 , 2005.

[18] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case", IEEE/ACM Trans. on Networking, vol.1, no. 3, pp. 344-357, June 93.

[19] X.J. Dong, M. Ergen, P. Varaiya, A. Puri, "Improving The Aggregate Throughput Of Access Points In IEEE 802.11 Wireless Lans", Local Computer Networks, Proceedings of 28th Annual IEEE International Conf. on, pp. 682 - 690, Oct. 2003.

[20] J. Rosenberg, et.al., "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[21] J.C.R. Bennett, H. Zhang, "WF^2Q: Worst-Case Fair Weighted Fair Queueing", INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies, Networking the Next Generation, Proceedings IEEE, vol. 1, pp. 120–128, March 1996.

[22] P. Goyal, H.M. Vin, H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm For Integrated Services Packet Switching Networks", Networking, IEEE/ACM Transactions on, vol. 5, Issue 5, pp. 690-704, Oct. 1997.

[23] P. Ramanathan and P. Agrawal, "Adapting Packet Fair Queueing Algorithms To Wireless Networks", ACM MOBICOM'98, October 1998.

Chapter 4. Fair Scheduling in Multi-Rate WLANs

4.1 Introduction

The widespread deployment of IEEE 802.11 Wireless Local Area Networks (WLAN) has made broadband wireless access a reality for many users [1]. As a result, a large variety of applications, such as real time multimedia applications, are being considered for these networks. Voice and Video telephony and real time TV are some examples. Supporting such applications requires guaranteed services that are not currently provided by existing 802.11 WLANs. The new IEEE 802.11e standard enhances the medium access controller (MAC) layer of the original 802.11 standard with features that facilitate guaranteed and priority service provisioning [2]. However, the new standard specifies only the features that are required for guaranteed service provisioning, and leaves the design of specific scheduling disciplines to developers.

Conventional scheduling techniques developed for wired networks or single direction wireless networks are not applicable to WLANs due to numerous reasons such as the fact that uplink and downlink traffic in WLANs share the same channel at all times and stations may operate at different rates at different times. Our proposed solution addresses these issues and efficiently utilizes the available features of the new standard to provide fair guaranteed

---

A version of this chapter has been submitted for publication. Yaser Pourmohammadi Fallah, Hussein Alnuweiri, Analysis of Temporal and Throughput Fair Scheduling in Multi-Rate IEEE 802.11e WLANs, submitted to Transaction on Mobile Computing, IEEE.

services for real-time applications. We target the infrastructure mode of operation in which a central access point (AP) controls the network. Most WLANs operate in this mode.

There have been considerable efforts in devising fair scheduling algorithms for wireless environments [3]. Some notable algorithms are Wireless Fair Service (WFS) [4], Idealized Wireless Fair Queuing (IWFQ), and Channel Independent Fair Queuing (CIF-Q) [6]. These algorithms are mainly designed for single direction scheduling, usually downlink from a base station, and are based on the assumption of a single fixed rate server; however, none of these assumptions are applicable to a multirate CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) network like IEEE 802.11. A WLAN based on 802.11 shares the medium at all times between uplink and downlink flows, and allows different operational transmission rates for each station. This means that these existing algorithms that were mainly for cellular networks are not directly usable in an 802.11e network and need to be modified.

There also exist another set of QoS solutions specially designed for 802.11 networks. These algorithms, such as the ones specified in [11],[15], and [16] provide prioritized differentiated services to aggregate flows. These solutions are mainly based on the contention access mechanisms and provide QoS in a probabilistic manner.

Other notable algorithms that consider the multirate operation, e.g., works presented in [22][23], also lack the same features that are necessary for CSMA/CA networks. Yuan in [22] proposes a temporal-fair algorithm, which is a variation of WFS, but the algorithm is suitable only for single direction scheduling in a TDD (Time Division Duplex) or FDD (Frequency Division Duplex) system and does not consider the shared medium nature of WLANs. Tinnirello in [23] studies service time fairness but does not present any scheduling scheme.

The works presented in [13][14] consider the specific controlled access mechanism of the 802.11e and propose extensions to the simple scheduler suggested in the standard, but the presented algorithms do not provide fine scale fairness and service guarantees in a multirate environment. In fact, very little work has been dedicated to providing solutions for per-session guarantees in multirate WLANs.

Our solution focuses on using controlled access mechanisms to provide per session QoS for real-time applications while maintaining fairness in short and long term. This is mainly achieved through using the Controlled Access Phase Scheduling (CAPS) mechanism that we first presented in [19] for a single rate WLAN. In this chapter, we extend CAPS to address the fairness issues in multirate and packet loss prone environment of a WLAN. Our extended solution addresses three main characteristics of an 802.11e WLAN that need to be taken into account when devising a QoS solution. First, the solution efficiently schedules both uplink and downlink flows to access the medium according to a Generalized Processor Sharing (GPS) based algorithm and without need for duplexing methods. Second, the scheduler efficiently distributes scheduled and contention based access periods with flexibility of changing the portion of each access type on demand. Third and most importantly the scheduler is able to provide throughput or service time (temporal) fairness and guarantee in a multirate environment. To the best of our knowledge this is the first design that addresses all these issues in a single framework for IEEE 802.11e networks.

In this chapter, we first briefly review multirate WLANs, in particular the 802.11e standard. Then, we present an overview of the CAPS mechanism that is the basis of the multirate solution presented in this chapter. We examine the effects of channel impairments on the modeling of a scheduler in WLANs and then extend the CAPS solution to address the issues of

multirate operation. We provide an analysis of different options of scheduling with CAPS, and present a performance evaluation of the final solution based on this study.

4.1.1 Scheduled Access in IEEE 802.11e MAC

The MAC layer of the 802.11 standard is based on a CSMA/CA mechanism. Collision avoidance in the MAC is achieved through Distributed Coordination Function (DCF) that specifies the timing rules of accessing the wireless medium. Stations running DCF have to wait a frame specific Inter Frame Space (IFS) time, before they can access the wireless medium. For data (and RTS) frames, each station has to wait an AIFS (Arbitration IFS) time plus a random backoff time, before it can access the medium. Acknowledgment (Ack) packets can be sent after a SIFS, which is shorter than AIFS. This ensures that an Ack messages can always be sent without interruption by other stations. The Access Point uses a PIFS (Point coordination function IFS), which is shorter than AIFS but longer than SIFS. This way, the access point can interrupt normal contention and take over the channel to create periods of contention free access called Controlled Access Phase (CAP). Since PIFS is longer than SIFS, the AP can only start a CAP after the current cycle of SIFS spaced frames completes.

In the 802.11e standard, the MAC layer rules for controlling and coordinating access to the wireless medium are specified under the Hybrid Coordination Function (HCF) protocol that works on top of DCF [2]. Using the services of DCF, HCF offers two access mechanisms: EDCA (Enhanced Distributed Channel Access), which is an enhanced version of the DCF of the original standard and is used for contention based access, and HCCA (HCF Controlled Channel Access) that specifies the polling or controlled access schemes. The 802.11e standard defines 8 different traffic priorities in 4 access categories and also enables the use of traffic flow IDs, which allow per flow resource reservation. Access to the medium is normally done

through EDCA; but the AP can interrupt the contention period (CP), at almost any time by waiting a PIFS time, and initiate a CAP to allow HCCA access (Figure 4-1). This feature allows scheduled HCCA access to the channel; however, the standard does not mandate any specific scheduling algorithm for HCCA, and devising the scheduler is left to developers.

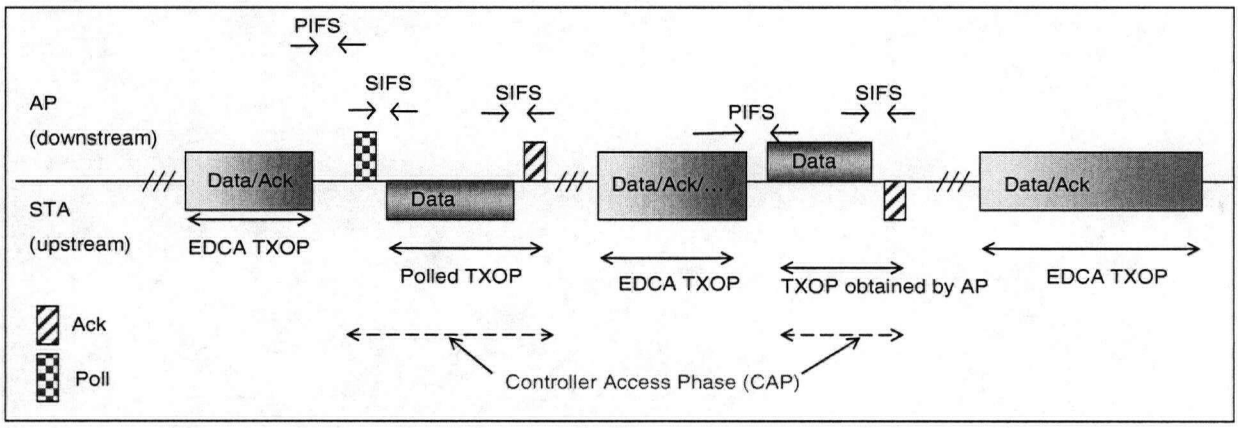

Figure 4-1 HCCA CAP generation

The 802.11e standard also introduces the concept of Transmission Opportunity (TXOP). TXOP specifies the duration of time in which a station can hold the medium uninterrupted and perform multiple frame exchange cycles consequently with SIFS spacing.

Under the EDCA access mechanism, different AIFS values are used for different classes of traffic. The backoff windows are also different for each priority. Shorter AIFS times and smaller contention windows give higher access priority. This prioritization enables a relative and per-class (or aggregate) QoS in the MAC. The 802.11e standard allows for dynamically adjusting most EDCA parameters, facilitating performance enhancement using adaptive algorithms [21].

4.1.2 MultiRate Operation of a WLAN

Wireless channels are significantly more error prone than wired channels. As a result, more error recovery and robustness methods are employed in wireless environments. One method of increasing the robustness of communications over a wireless channel is to reduce the number of bits that are modulated into one symbol (e.g., from QAM-64 to QAM-16). The other method is through increasing the FEC encoding ratio. These methods increase the probability of a frame being correctly received at the receiver, but consequently reduce the transmission rate. The 802.11 standard utilizes these methods and defines several modes of operation for the PHY that result in different transmission rates over the 20Mhz available wireless channel.

Through employing different modulation and FEC methods, a variety of transmission rates are available in the 802.11 WLANs. For example, the 802.11b standard allows four rates of 1, 2, 5.5 and 11Mbps, while 802.11a/g provides a set of eight rates (6, 9, 12, 18... 54 Mbps). The upcoming 802.11n standard has an even larger set of available rates, ranging from 6 to almost 300 Mbps (600 Mbps for 40 MHz channels). Any of these transmission rates can be applied to individual frames and stations. As a result, an 802.11 network includes stations that may use different transmission rates. Moreover, individual packets belonging to the same station may also be sent at different rates.

The decision to use a specific modulation and FEC method is made in the transmitter using the available channel condition information. Such information may be continuously collected through monitoring the SNR, interference level and other measurable parameters. It is therefore possible for a station to change the transmission rate dynamically, or just choose an appropriate rate at the time of associating with the access point. The information that is gathered by the transmitter is used to determine a PHY mode that results in the highest

transmission rate with an acceptable Bit Error Rate (BER) or Packet Error Rate (PER). Usually a BER value of 10^{-5} is considered acceptable, and if the packet length is known, a PER of 10% is deemed acceptable.

The mechanism that is used for selecting the appropriate transmission rate (PHY mode), from the set of available rates, is called "link adaptation". The standard does not mandate any specific link adaptation algorithm, and it is up to developers to devise such methods. Several algorithms have been designed for determining the process of selecting the best PHY mode. The work in [6] describes an Auto Rate Fall-back (ARF) protocol that uses a heuristic method in which the transmitter selects the next lower rate after two failed transmissions (ACKs not received for two consecutive transmissions), and raises the rate to the next higher rate after ten successful transmissions. The method also maintains a timer to reset the rate if no transmissions happen after a rate reduction. However, such methods may not be very efficient, since they are affected by MAC layer operation and respond very slowly to channel variations. A new mechanism described in [7] provides a more intelligent method that uses channel BER information and results in a more sustainable throughput.

Temporal Fairness vs. Throughput Fairness

The use of different transmission rates by each station or frame has significant implications on the fairness of the services given to stations in a WLAN. Since lower rates mean longer transmission times, it is clear that a station that is experiencing lower channel quality and is using lower transmission rate takes more time to send the same amount of data. As a result, it may starve the other stations of service. This situation may be deemed unfair, if we define fairness in terms of service time. This type of fairness is also called "temporal fairness" in which the goal is to provide the stations with the same amount of service time, regardless of

their transmission rate. In contrast to temporal fairness, we can also define fairness in terms of the throughput that is assigned to stations in a given duration of time (usually long-term); this is called "throughput fairness". In this case, fairness is achieved if all stations achieve the same throughput in a given duration of time. It must be noted that the issue of unfairness in WLANs is also linked to the packet loss problem, since multirate operation does not completely eliminate the packet loss probability in WLANs.

The original 802.11 MAC was designed to be throughput fair, resulting in temporal unfairness in multirate scenarios. However, the temporal unfairness that results from lower rate stations taking up long times to transmit may be unacceptable. To tackle this issue, the concept of transmission opportunity has been introduced in 802.11e to limit the maximum transmission time for each station or class of traffic. Using TXOP, it is possible to achieve temporal fairness in contention mode (EDCA) of the MAC layer. Throughput fairness is also possible in EDCA by removing TXOP limitations. For controlled or scheduled access using HCCA, the scheduling process can be designed to provide either temporal or throughput fairness based on the overall performance objective. We address this issue in the current chapter.

In the next section, we will briefly describe a framework for centralizing the scheduling task for a WLAN. This centralized design is extended in later sections to deal with service variations specific to multirate 802.11 wireless LANs.

4.2 Centralized Hybrid Scheduling in WLANs

Considering the distributed multiple access environment of the 802.11e WLAN, with possible packet loss and multirate operation, we propose a detailed QoS solution that addresses

the following three prominent aspects of scheduling in such environments: (1) Unified Uplink/Downlink scheduling, our QoS framework uses virtual packets and combines the task of scheduling uplink and downlink flows of a naturally distributed CSMA/CA environment into a central scheduler that resides in an AP; (2) Service guarantee and scheduling HCCA and EDCA access, our framework uses a GPS based algorithm for the centralized scheduler, as well as an integrated traffic shaper, to provide guaranteed fair channel access to HCCA flows with reservation, and share the remaining capacity using EDCA; (3) Multirate operation, we extend the single server/scheduler model, to maintain fairness and service guarantees under multi rate operation and channel impairments such as packet loss. Since our solution presented here is an extension of the CAPS framework[19], we will not repeat the details of CAPS and only provide an overview in the next subsections. A complete description of CAPS framework, initially designed for single rate WLANs, is found in chapter 3 or [19].

4.2.1 Combining Downlink and Uplink Scheduling

In a CSMA/CA WLAN, the medium is shared between downlink and uplink traffic at all times. Therefore, the scheduling discipline must consider both uplink and downlink traffic for scheduling at all times. Moreover, it must enforce the scheduling decision. Downlink packets are available in the AP buffers and can be directly scheduled, while uplink packets reside in the stations generating these packets, and cannot be scheduled directly. However, the AP can use uplink traffic specifications, available through signalling (e.g., MAC ADDTS messages) or feedback, and schedule poll messages that allow for uplink packet transmission.

The key to realizing the above scheduling concept is to represent packets from remote stations (i.e. the uplink packets) by "*virtual packets*" in the AP, then use a single unified scheduler to schedule virtual packets along with real packets (downlink packets). When

scheduling virtual packets, the AP issues polling in the appropriate sequence to generate transmission opportunities for uplink packets. This *hybrid scheduling* scheme combines uplink and downlink scheduling in one discipline, and allows the use of centralized single server schedulers. This design is depicted in Figure 4-2.

Virtual packets are generated using the specified patterns of uplink traffic flow (e.g., specified in the MAC layer ADDTS message). For example, for a voice call, a periodic flow of virtual packets similar to the real traffic is generated. The virtual packets are classified along with actual downlink packets, and are queued and scheduled for service by the inner scheduler. Packets that are served by the scheduler are treated differently based on whether they are actual or virtual packets. Actual packets are directly transmitted in a downlink CAP, but for virtual packets an uplink CAP is generated by sending a poll message and assigning the appropriate TXOP to the station whose virtual packet is being served.

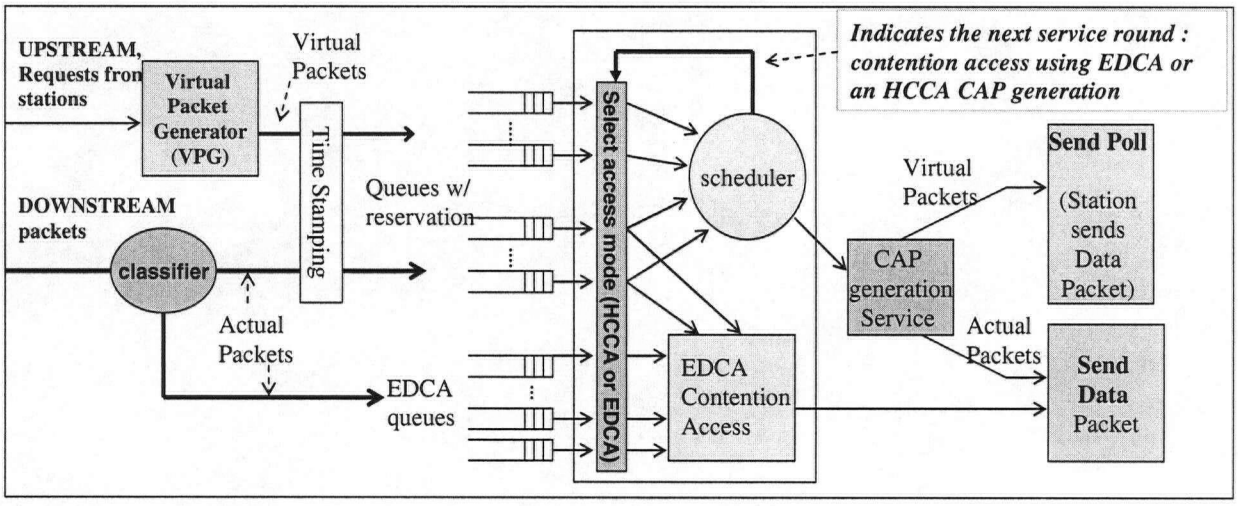

Figure 4-2 Architecture and queuing model of a CAPS Access Point

4.2.2 Scheduling and Switching between HCCA and EDCA

CAPS generates durations of HCCA access (CAP) to serve downlink and uplink flows that have made reservation with the access point, and leaves the remaining capacity to be used by

EDCA in a fair contention manner. To achieve this goal, traffic shaping on both virtual and actual downlink flows is needed. Since virtual packets are already generated according to an accepted traffic pattern, we only need to apply the traffic shaping policy to downlink HCCA flows. This is achieved by time stamping all the downlink HCCA packets with a service eligibility time. At each service round, the CAPS scheduler serves only downlink HCCA queues with eligible head of line (HoL) packet, plus any non-empty virtual flow queue (a service time occurs after a transmission is completed and the AP senses that medium has been idle for one PIFS duration). If a queue is not found for HCCA service, control is given to EDCA. To share the channel fairly in EDCA, we allow all downlink HCCA queues (including those with non-conforming traffic) plus the existing 4 (or 8) EDCA queues to contend for the channel. Virtual flow queues are not allowed to contend since their corresponding uplink flows in stations are already involved in contention.

When the inner scheduler determines that HCCA service must take place, it selects one of the eligible queues based on the scheduling discipline used. We use a GPS based scheduler in order to provide fairness and service guarantee. In chapter 3, we investigated the use of several GPS based fair algorithms such as Start-time Fair Queuing (SFQ, [18]), Weighted Fair Queuing (WFQ, [16]), or Worst case Fair Weighted Fair Queuing (WF^2Q, [17]). These algorithms require tagging each packet with start and finish times and then serving them in the order of either start time (SFQ) or finish time (WFQ, and WF^2Q). They also maintain a virtual time function that keeps track of the system progress in comparison to an ideal GPS. The packet start and finish times for these inner schedulers (SFQ, WFQ, and WF^2Q) are given by:

$$S_i^k = \max(F_i^{k-1}, V(t)) \tag{4.1}$$

$$F_i^k = \frac{L_i^k}{r_i} + S_i^k \tag{4.2}$$

Where S_i^k and F_i^k are the start and finish timestamps for the kth packet from the ith flow, L_i^k is the packet length, r_i is the rate assigned to the flow, and $V(t)$ is the virtual time function. The virtual time is calculated differently for each inner scheduler. For WFQ and WF^{2}Q, $V(t)$ represents the progress time of a GPS scheduler that is serving the same set of packets from these queues at rate C, and is calculated as:

$$V(t_{j-1} + T) = V(t_{j-1}) + \frac{T}{\sum_{i \in B_j} (r_i/C)} \quad , \quad T \le t_j - t_{j-1}, j = 2,3,... \tag{4.3}$$

where T is the time between two subsequent events j and j-1 (i.e.. packet arrival or departure) in the GPS system and B_j is the set of backlogged sessions (queues) between these events.

For SFQ, the virtual time is described in a much simpler way as the start tag of the packet in service at time t. At the end of a busy period, $v(t)$ is set to the maximum of finish tag of packets served in that busy period (or zero) [18]. The multirate behavior of a WLAN and the choice of temporal or throughput fairness affect how the corresponding GPS scheduler progresses; thus, the above time stamps and virtual time calculations may need to be modified. We will discuss the modifications later in this chapter.

In chapter 3 (also in [19]) we investigated the use of several GPS based fair algorithms such as SFQ, WFQ, and WF^{2}Q. We identified that if precise knowledge of uplink flows is not available, the deviation of CAPS from an ideal GPS is different from that of CAPS inner scheduler and the ideal GPS. The deviation bounds are found in [19] and chapter 3, and it is shown that if precise knowledge of uplink packet sizes is not available, the deviation of CAPS-WFQ and CAPS-WF^{2}Q from an ideal GPS may be much larger than the deviation of the inner schedulers from GPS. This weakness of CAPS-WFQ and CAPS-WF^{2}Q somewhat decreases

their suitability, in comparison to CAPS-SFQ, which is found to always behave identically to SFQ. In this chapter, we analyze the effects of multirate operation and packet loss on the performance of the inner scheduler of CAPS to identify the enhancements needed for providing either temporal or throughput fair services, and to identify the best inner scheduler choice.

4.2.3 Service Compensation for Uplink Flows

Traffic shaping for uplink flows is mainly done through generating conformant virtual flows. However, in some cases the length of an uplink packet, sent in response to a poll, may be smaller than that of the virtual packet that generated the poll. To keep track of this lost service for the station, and compensate it later, the algorithm maintains a queue budget parameter g_i for each uplink flow i. The budget is always limited to the maximum allowed burst size of each flow. It is incremented by the size of each transmitted poll, and decremented by the size of each response received. When the response packet is shorter than the virtual packet, the budget becomes positive.

The positive budget is an indication of lost service. To compensate this lost service we use a "Deferred Compensation" method in which the excess budget is used to generate additional virtual packets for the same virtual flow (for more details and also other compensation options refer to section 3.2.4 or [19]). These additional virtual packets are then stored in the corresponding queue, and will receive service at the flow's guaranteed rate. In effect, this is similar to attempting to retransmit a virtual packet (poll message). This mechanism isolates the compensation for a specific flow from the rest of the flows, and enhances service guarantees and fairness.

4.2.4 Adapting to Wireless Channel Quality

As it was described in previous sections, we can use the CAPS framework and centralize the scheduling task in the access point of a WLAN. In conventional schedulers, the server is considered to be a fixed rate server (e.g. 100Mbps Ethernet), while for wireless environments this assumption does not hold and many sources of impairments make the server behave differently from a fixed rate server. For example, physical channel impairment in a WLAN results in packet loss or reduction in transmission rate over the wireless medium. Collisions that are inherent in CSMA networks also introduce a random but bounded medium access delay to the MAC layer HCCA operation. These impairments in MAC and PHY operations mean that the transmission services provided by a WLAN are indeed variable, and the WLAN has to be modeled by a variable rate server with occasional interruptions in service. We examine the effect of each of these service disruptions on the modeling of WLAN as a server for the proposed fair scheduling framework.

Effect of collision on CAP generation delay

In HCCA mode, a CAP can only be initiated by an access point after sensing the medium idle for a PIFS time. If the medium is busy, the AP has to wait until the current TXOP finishes and then it can take over the channel. Since all non-AP stations have to wait longer than PIFS, there is no possibility of collision in this case. If the medium was not busy when an AP needs to generate a CAP, a collision may happen if another station's waiting time for EDCA access elapses at the same time, and it attempts to transmit over the channel. In this case the AP has to wait until the end of the collided frame, before it can take over the channel. Since a TXOP is usually longer than a frame length, the AP medium access delay is at most one TXOP length.

To handle the effect of HCCA access delay and maintain the fairness of the inner scheduler, certain steps must be taken. If schedulers such as WFQ and WF^2Q are used, it will be enough to freeze the calculation of virtual time for the duration of the access delay. For algorithms like SFQ, this delay has no effect on the fairness of the algorithm because the algorithm does not change its virtual time during the delayed HCCA access anyways. Assuming this adjustment, we do not need to consider this delay for these scheduling algorithms.

Effect of Packet Loss

The 802.11 standard allows retransmission in the MAC layer to increase the probability of correct reception for packets that are lost due to physical layer errors. One way to model the retransmission process in HCCA mode is to consider it as a rate reduction. For example, if a packet is retransmitted n times, we can assume that it has been served at an n times lower rate (counting MAC and PHY overhead as well). Using this model for retransmitted packets we can simplify the problem to the rate reduction (Multirate) case.

If dynamic and precise channel estimation is available, we can also utilize the lead/lag model used in [4][6]. These models rely on detecting channel quality beforehand and lending one station's transmission time to another; however, this situation in WLANs is dealt with using the multirate operation. Therefore we do not study these models in this chapter and focus on the abovementioned retransmission policies and multirate operation instead.

Effect of Multi-Rate Operation

An 802.11 WLAN that uses multirate operation to combat channel impairments can be simply modeled as a multirate server that serves packets at a given rate for the entire duration of the packet. The multirate operation can be either static or dynamic. In static multirate, each station selects a specific rate at association time and may change the rate occasionally, but not

very often. In dynamic multirate scenario, the station may use a different rate for each packet. In the next sections we study the effects of multirate operation on the fairness of different CAPS scheduling options.

4.3 Fairness and Multirate operation of WLANs

Multi-rate operation introduces complexities for fair scheduling algorithms that normally assume a fixed service rate for all queues. In particular, GPS based algorithms must ensure that they emulate a GPS server that is operating under the same variable rate conditions in order to correctly provide fair services. Although such a task may be computationally very expensive for some algorithms, it is in general feasible.

When the transmission rate for a specific station changes, the transmission durations for its flows become different from what the scheduler may have originally anticipated. Therefore, the scheduler may have to either cut the transmission time for this station, or borrow from the service time of other stations. In effect, the scheduler is faced with two different choices of either maintaining service time fairness amongst stations (temporal fairness) or continuing to provide the promised throughput to all stations (throughput fairness). Admission control or network management entities can decide on which approach to apply, based on application requirements and network capability. The general idea is to borrow from the excess capacity of the server (if there is any) and provide the guaranteed service to a station whose transmission rate has dropped. This can be seen as providing throughput fairness by borrowing from system's excess capacity. When the extra capacity is depleted, the policy of the scheduler should change to maintaining the current service level for other stations and not penalizing them for the rate reduction in one station; effectively providing temporal fairness.

Assuming that the decision of providing temporal or throughput fairness is made by the admission control and management entities we examine how temporal and throughput fairness can be achieved in CAPS, using different types of GPS based inner schedulers. As a first step we investigate the centralized scheduler behavior under WLAN multirate conditions and then extend the analysis to include CAPS specific performance.

4.3.1 Fair Generalized Processor Sharing with MultiRate Operation

The GPS algorithm is based on a fluid model server that provides fair (weighted) service to all flows, assuming a constant transmission rate [16]. When working in a multirate environment, we need to redefine GPS so that it remains either throughput fair or temporal fair. In a throughput fair GPS server, in each round of service all queues are served an amount of traffic proportional to their weight. This description can be applied to the following throughput fairness formulation for GPS in order to derive the amount of guaranteed throughput for each flow. Throughput fairness for a GPS server is defined as:

$$\frac{W_i(\tau,t)}{W_j(\tau,t)} \geq \frac{\varphi_i}{\varphi_j} \tag{4.4}$$

where $W_i(\tau,t)$ is the traffic (in bits) served for queue i during time period (τ,t), and φ_i is the queue i weight (without loss of generality we assume $\sum_{j\ all\ queues} \varphi_j = 1$ in the rest of this chapter).

Given the description of throughput fair GPS, we know that if the total amount of served traffic in (τ,t) is B bits, each queue i will receive a share of $\varphi_i \cdot B$. Transmitting this amount at a queue transmission rate C_i takes $\varphi_i B / C_i$ time, assuming constant rate for the queue in (τ,t). Hence, the total time to serve all queues is: $t - \tau = \sum_{j\ all\ queues} (\varphi_j B / C_j)$

Taking the sum over all queues j in (4.4) and noting that the sum of all W_j's is B, we derive the guaranteed rate of service g_i for queue i by dividing $W_i(\tau,t)$ by $(t-\tau)$ as follows:

$$g_i \geq \varphi_i \frac{B}{\sum_{i\ all\ queues} \varphi_j \frac{B}{C_j}} = \frac{\varphi_i}{\sum_{j\ all\ queues} \frac{\varphi_j}{C_j}} = \varphi_i.C^* \tag{4.5}$$

where we define $C^* = 1/(\sum_{j\ all\ queues} \frac{\varphi_j}{C_j})$ as the average server rate. The above expression shows that the guaranteed throughput for one queue is dependent on the transmission rate of other queues.

The other option of fairness, as described before, is temporal fairness. We can formulate the temporal fairness of a scheduler as follows:

$$\frac{T_i(\tau,t)}{T_j(\tau,t)} \geq \frac{\varphi_i}{\varphi_j} \tag{4.6}$$

in which $T_i(\tau,t)$ is the portion of time interval (τ,t) spent serving queue i. Thus a temporally fair GPS scheduler serves each queue, in each round of service, for duration of time proportional to the service weight of the queue. Following this definition and taking a sum over all flows j in (4.6) we know that in a period (τ,t), the scheduler spends $T_i(\tau,t) \geq \varphi_i \cdot (t-\tau)$ for queue i; thus serving $W_i(\tau,t) \geq \varphi_i(t-\tau).C_i$ bits for this queue. Dividing this amount by $(t-\tau)$, we have the following guaranteed throughput for each queue i served by a Temporal Fair GPS scheduler:

$$g_i \geq \varphi_i C_i \tag{4.7}$$

In contrast to (4.5), the above equation shows that for a temporal fair GPS, the guaranteed throughput is not dependent on other queues transmission rates, and is directly proportional to the queue's transmission rate. Next, we examine how packet based approximations of GPS such as WFQ or WF^2Q can emulate a throughput or temporal fair GPS; where appropriate, we will distinguish two types of Multirate operation, static and dynamic, in our analysis.

4.3.2 Throughput and Temporal Fair WFQ and WF^2Q

WFQ and WF^2Q are packet based approximations of GPS. We achieve temporal or throughput fairness for these algorithms under multirate operation if we modify them to approximate a throughput or temporal fair GPS. To do so, we consider a realization of these algorithms that use simulation of the corresponding GPS scheduler [16][17]. This method uses a virtual time that keeps track of the progress of a hypothetical GPS scheduler serving the same queues. Then it tags packets with their finish time from the simulated GPS system and serves them based on the increasing order of finish times (for WFQ). For WF^2Q only packets that have started service under GPS are considered for scheduling. Virtual time and packet finish and start times are described in equations (4.1) to (4.3) for a single rate scheduler. In this section we present throughput and temporal fair WFQ and WF^2Q and analyze their implementation complications. An analysis of the deviation of these algorithms from their corresponding GPS schedulers is given in Appendix C as a reference.

4.3.2.1 Throughput Fair WFQ and WF^2Q

A throughput fair WFQ should use the virtual time that is derived from a throughput fair GPS system, as described in the previous section. To find out how such virtual time can be calculated, we first present the following definition of virtual time: virtual time $v(t)$ represents the number of completed service rounds in a GPS system until the given time t. Service rounds

in a throughput fair system are service visits by the scheduler during which a fair proportion of traffic is served from each backlogged queue. For example, for each queue i this amount is $\varphi_i \sigma$ bits, where without loss of generality we set σ to 1 bit (and consider the number of rounds as a real number). A more rigorous definition of σ as an infinitesimal amount of data can be used, but is unnecessary for our discussion. Clearly, if the number of backlogged queues increases, then the time it takes to complete a round increases as well. In fact, for a constant rate server with rate C (bits per second), the number of rounds completed during an interval of length τ in which the set of backlogged sessions (B) remains unchanged is:

$$V(t+\tau) - V(t) = (\textit{ the number of completed rounds in } \tau) = \frac{\tau}{\sum_{i \in B} \frac{\varphi_i}{C}} = \frac{C \cdot \tau}{\sum_{i \in B} \varphi_i} \tag{4.8}$$

This expression is in fact what equation (4.3) describes. The corresponding start and finish times for the kth packet that arrives at time a_i^k are also given as:

$$S_i^k = \max(F_i^{k-1}, V(a_i^k)) \qquad \text{and} \qquad F_i^k = S_i^k + \frac{L_i^k}{\varphi_i} \tag{4.9}$$

in which $\frac{L_i^k}{\varphi_i}$ denotes the number of rounds of service needed to serve packet k. This is regardless of the transmission rate.

Static Multirate Case

To consider the effects of multirate operation, we first consider the *static* multirate case. Here, each queue i has a fixed but different transmission rate C_i. Recalling the definition of throughput fair GPS whereby φ_i bits are served in each round for queue i, and knowing that this amount takes φ_i / C_i seconds to serve, equation (4.8) becomes:

$$V(t+\tau) - V(t) = \frac{\tau}{\sum_{i \in B} \frac{\varphi_i}{C_i}} \qquad or \qquad \frac{dv}{dt} = \frac{1}{\sum_{i \in B} \frac{\varphi_i}{C_i}} \tag{4.10}$$

Note that $\frac{dv}{dt}$ denotes the slope of the virtual time function. Finish and Start time tags for the multirate case remain as in (4.9), since they are not directly dependent on the transmission rate.

Dynamic Multirate Case

In a WLAN environment, it is possible that in certain cases each individual packet k belonging to the same station (queue i) be sent at a different transmission rate C_i^k. Since the rate does not change during one packet transmission, we find that the time it takes to serve one flow i's share is φ_i / C_i^k. Taking a sum over all flows i belonging to the set of backlogged queues, equation (4.10) changes to:

$$V(t+\tau) - V(t) = \frac{\tau}{\sum_{i \in B} \frac{\varphi_i}{C_i^k}} \quad or \quad \frac{dv}{dt} = \frac{1}{\sum_{i \in B} \frac{\varphi_i}{C_i^k}} \tag{4.11}$$

where k^{th} packet of queue i is being served during $(t\ ,\ t+\tau)$. Although the above equation looks similar to (4.10), it can be considerably harder to compute. In (4.10), updating the slope of virtual time function can be done at start and end of each queue's busy periods; while in (4.11) this has to happen at each packet arrival and departure. Computing (4.11) becomes even much harder, if we consider the fact that in dynamic multirate cases, the rate at which a packet is sent in the future may not be known in advance.

Since GPS may run ahead of WFQ and WF^2Q by at most L_{max} bits for any given packet, where L_{max} is the maximum packet length in the system, it is possible that a packet assumes a

rate by GPS while it receives a different rate by the actual scheduler. This situation leads to incorrect calculation of time stamps (and potentially wrong scheduling order for at most one packet). In this case, all the virtual time and related tags must be recalculated in order to correctly schedule successor packets. This requires that we back track the GPS simulation until the arrival of the packet in question and recalculate all the time stamps for successor packets in the same queue, as well as the virtual time calculated at that time, and all the finish and start time tags dependent on this virtual time. Such re-adjustments require that we store the arrival time of all packets and re-simulate the behavior of the system from the arrival time of the concerned packet. This is potentially a very costly solution, especially for large systems. Nevertheless, such a task is possible, despite complexity.

4.3.2.2 Temporal Fair WFQ and WF^{2}Q

Similar to the throughput fair GPS scheduler, we can model the progress of a temporal fair GPS scheduler using a virtual time function. However, in this case we change the definition of a service round (virtual time represents the number of completed service rounds in a GPS system until the given time t) as follows: A round is a service visit by the scheduler in which the server spends a fair proportion of time serving each backlogged queue. For example, for each queue i this amount is $\varphi_i \delta$ seconds, where δ is a very small unit of time. Assuming a static or dynamic multirate case, the length of a service round is $\sum_{i \in B} \varphi_i \delta$, and the following expresses the virtual time progress during an interval of length τ in which the set of backlogged sessions (B) remains unchanged: (without loss of generality we can set δ to $1/C$ seconds, where C is the server nominal rate, and obtain an equation similar to (4.8)):

$$V(t+\tau) - V(t) = \frac{\tau}{\delta \sum_{i \in B} \varphi_i} = \frac{C \cdot \tau}{\sum_{i \in B} \varphi_i} \tag{4.12}$$

The above formulation for the virtual time is applicable to both dynamic and static multirate operations. To compute the corresponding start and finish times for the kth packet that arrives at time a_i^k, we must note that such packet requires L_i^k / C_i^k seconds of service in dynamic multirate case (L_i^k / C_i for static multirate). This time divided by the amount of service $\varphi_i \delta$ given to this queue in each round will give us the number of service rounds needed to serve this packet, thus:

$$S_i^k = \max(F_i^{k-1}, V(a_i^k)) \quad \text{and} \quad F_i^k = S_i^k + \frac{L_i^k}{\varphi_i \cdot \delta \cdot C_i^k} = S_i^k + \frac{L_i^k}{\varphi_i \cdot C_i^k / C} \tag{4.13}$$

Clearly, the finish time is directly dependent on the transmission rate of the packet. The above equations are valid for both static and dynamic multirate operations (replacing C_i^k with C_i for static case). With dynamic rate change, re-adjustment of time stamps is necessary (as in throughput fair case) to correct the situation in which the actual transmission rate is different from what was assumed at scheduling time.

4.3.3 Throughput and Temporal Fair SFQ in a MultiRate Scenario

The SFQ algorithm is inherently throughput fair. This fact was established in [22] by proving that the difference of weighted service given to any given pair of queues is bounded. This was shown by proving that the following bound exists:

$$\left| \frac{W_i(\tau,t)}{\varphi_i} - \frac{W_j(\tau,t)}{\varphi_j} \right| \le \frac{L_i^{\max}}{\varphi_i} + \frac{L_j^{\max}}{\varphi_j} \tag{4.14}$$

Since there is no assumption on the transmission rate of individual queue's or packet's transmission rates, SFQ maintains its throughput fairness in both static and dynamic multirate scenarios. With a simple modification to the SFQ implementation, the algorithm can work properly without needing the advanced knowledge of individual packet transmission rates. The reason is that packets are served in order of start time tags, and the time stamp of head of line packets can be adjusted after the current packet finishes service, which means its transmission rate is known. This modification had already been included in CAPS-SFQ in order to eliminate the effect of virtual and actual packet length mismatch [19].

To achieve temporal fairness using SFQ, we need to modify the algorithm so that its inherent throughput fairness translates to temporal fairness. This property is achieved by modifying the definition of finish time tag as follows:

$$F_i^k = S_i^k + \frac{L_i^k / C_i}{\varphi_i} = S_i^k + \frac{L_i^k}{\varphi_i \cdot C_i} \tag{4.15}$$

The start time remains unchanged at $S_i^k = \max(F_i^{k-1}, V(a_i^k))$, and virtual time selected as the start time of the packet in service. For brevity, we call this modified SFQ as **STFQ** (Start-time Temporal Fair Queuing). To prove that this modification yields a temporal fair service, we first need to define a temporal fairness measure similar to (4.14). We will use the following measure for two queues i and j that are backlogged over the interval (τ, t):

$$H_{tmf} = \left| \frac{T_i(\tau,t)}{\varphi_i} - \frac{T_j(\tau,t)}{\varphi_j} \right| \tag{4.16}$$

where $T_i(\tau, t)$ is the amount of service time provided to queue i. For the static multirate case we have the following lemma.

Lemma 1: A STFQ system with weights φ_i and transmission rate C_i for each queue has the following bound on temporal fairness measure:

$$H_{tmf} = \left| \frac{T_i(\tau,t)}{\varphi_i} - \frac{T_j(\tau,t)}{\varphi_j} \right| \leq \frac{L_i^{\max}}{C_i \varphi_i} + \frac{L_j^{\max}}{C_j \varphi_j} \tag{4.17}$$

Proof: In SFQ the weights are un-interpreted numbers and the algorithm is throughput fair with regards to these weights. For a SFQ system with weights $C_i \varphi_i$, we have (from (4.14)):

$$\left| \frac{W_i(\tau,t)}{C_i \varphi_i} - \frac{W_j(\tau,t)}{C_j \varphi_j} \right| \leq \frac{L_i^{\max}}{C_i \varphi_i} + \frac{L_j^{\max}}{C_j \varphi_j} \tag{4.18}$$

On the other hand, we see that using the rate $C_i \varphi_i$ in finish time equation (4.15) yields the required STFQ system. We also know that during an interval (τ, t) the amount of server time spent on each queue i is $T_i(\tau,t) = \frac{W_i(\tau,t)}{C_i}$. The Lemma follows from this equality. **Q.E.D.**

For the dynamic multirate case we redefine the finish time in (4.15) as follows:

$$F_i^k = S_i^k + \frac{L_i^k}{\varphi_i \cdot C_i^k} \tag{4.19}$$

where C_i^k is the transmission rate for the kth packet of queue i. We show next that this definition results in a temporal fair service as well.

Lemma 2: A STFQ system with weight φ_i and transmission rate C_i^k for each packet k of queue i has the following bound on temporal fairness measure:

$$H_{tmf} = \left| \frac{T_i(\tau,t)}{\varphi_i} - \frac{T_j(\tau,t)}{\varphi_j} \right| \leq \frac{L_i^{\max}}{C_i^{\min} \varphi_i} + \frac{L_j^{\max}}{C_j^{\min} \varphi_j} \tag{4.20}$$

Proof: Consider a SFQ scheduler with weights φ_i assigned to each queue i with packets of length ($L'^{k}_{i} = L^{k}_{i} / C^{k}_{i}$). This system is clearly equivalent to the dynamic multirate STFQ. Since SFQ is throughput fair the following inequality holds for this system:

$$\left| \frac{W'_i(\tau,t)}{\varphi_i} - \frac{W'_j(\tau,t)}{\varphi_j} \right| \leq \frac{L'^{\max}_i}{\varphi_i} + \frac{L'^{\max}_j}{\varphi_j} \tag{4.21}$$

where $W'_i(\tau,t)$ is the amount of traffic served for queue i in the SFQ server. If we assume that M packets have been served for queue i in interval (τ ,t), we have: $W'_i(\tau,t) = \sum_{n=1}^{M} L'^{n}_i = \sum_{n=1}^{M} (L^n_i / C^n_i)$; We also know that the amount of time spent serving the same M packets in STFQ, during the same interval (τ ,t) is $T_i(\tau,t) = \sum_{n=1}^{M} (L^n_i / C^n_i)$, thus $W'_i(\tau,t) = T_i(\tau,t)$ and we can rewrite (4.21) as : $\left| \frac{T_i(\tau,t)}{\varphi_i} - \frac{T_j(\tau,t)}{\varphi_j} \right| \leq \frac{L'^{\max}_i}{\varphi_i} + \frac{L'^{\max}_j}{\varphi_j}$.

Since $L'^{\max}_i = (L^{\max}_i / C^{\min}_i)$, where $C^{\min}_i$ is the minimum transmission rate for a packet in any interval (τ , t). The lemma follows by substituting $L'^{\max}_i$ in (4.21). **Q.E.D.**

4.4 Delay Bound Analysis of CAPS-SFQ

We have shown in chapter 3 (also in [19]) that in a singe rate WLAN, WFQ and WF^{2}Q do not have an advantage over other fair algorithms such as SFQ that use start time scheduling. In fact, we have shown that the advantage of WFQ and WF^{2}Q over SFQ, in terms of their lower deviation from GPS in some ideal situations, diminishes when we consider the short response case and WLAN operation. Given this fact, and considering the significant complexity of implementing time stamp adjustment for WFQ and WF^{2}Q algorithms in a dynamic multirate

environment, we select the modified SFQ as our choice for the inner scheduler of CAPS. Choosing SFQ has other advantages as well; for example, the delay bound provided by SFQ for a flow with low bitrate, can be much less than that provided by WFQ and WF^2Q [18].

Having chosen SFQ as the inner scheduler, we next derive the delay bound of CAPS-SFQ under different fairness assumptions and multirate conditions. It must be noted that the same adjustment of time stamps that corrects the time stamps in SFQ is also effective in eliminating the short response case described in Chapter 3. Therefore, the analysis in this section applies to both a standalone modified SFQ as well as CAPS-SFQ, which is the focus of this chapter.

4.4.1 Delay Bound Analysis of Throughput Fair SFQ in Static Multirate Case

The delay bound found for the SFQ algorithm in [18] assumes an average service rate C for all queues. This is clearly not applicable to the multirate environment we are considering. In this section we present a series of lemmas that will lead to finding the delay bound of the static multirate CAPS-SFQ, and proving that it is given by inequality (4.33) of theorem 1.

The work in [18] defines the scheduling delay of a packet as the time it takes the packet to depart the system after its Expected Arrival Time (*EAT*). EAT_i^k is defined as the time that we expect a packet k from queue i to start service (reach head of line in a corresponding GPS) if the queue has been continuously served at a reserved rate r_i. Thus, the scheduling delay of concern in this section excludes the queuing delay inside a queue, since that delay can be separately found. For brevity we call the scheduling delay bound as "delay" in this section. This definition leads to the following observation:

$$Departure\ of\ packet\ p_i^k = Dep_i^k \le EAT_i^k + Scheduling\ Delay$$

$$EAT_i^k = \max\{a_i^k, EAT_i^{k-1} + \frac{L_i^{k-1}}{r_i}\} \tag{4.22}$$

where a_i^k is the arrival time of packet k. If the queue remains backlogged after m^{th} arrival we also have:

$$EAT_i^k = EAT_i^m + \sum_{n=m}^{k-1} \frac{L_i^n}{r_i} \tag{4.23}$$

To calculate the delay, we must consider the packets whose service time does not finish before the expected arrival time of packet k of a queue f. To formulate this situation, assume (without loss of generality) that packet k arrives in queue f that has been backlogged with packets m to k-1 such that packet m arrived at an empty queue. With these assumptions and assuming that the weight φ_i of each flow i in SFQ is set to r_i, the start time tags are found as:

$$S_f^m = V(a_f^m) \quad , \quad S_f^k = S_f^m + \sum_{n=m}^{k-1} \frac{L_f^n}{r_f} \tag{4.24}$$

Let $V_1 = S_f^m$ and $V_2 = S_f^k$ be the system virtual times, corresponding to the start time tags of packets m and k. From (4.24) we derive $V_2 - V_1 = \sum_{n=m}^{k-1} \frac{L_f^n}{r_f}$; combining with (4.23) we have:

$$EAT_i^k = EAT_i^m + V_2 - V_1. \tag{4.25}$$

Figure 4-3 shows that the worst case departure time of packet k is after the service time of all the packets shown in the figure, plus the arrival time of packet m. We divide these packets into the following two sets of E1 and E2:

E1: $\{\text{all packets } n \mid V_1 \le S_i^n \le V_2 \quad and \quad F_i^n \le V_2\}$

E2: $\{\text{all packets } n \mid V_1 \le S_i^n \le V_2 \quad and \quad F_i^n > V_2\}$

Therefore, the departure time for packet k is $Dep_f^k \le EAT_f^m + T(E1) + T(E2) + (L_f^k / C_f^k)$, where C_f^k is the transmission rate for packet k of flow f., and $T(E)$ is the time it takes to serve packets of set E. If we deduct EAT_f^k from both sides of this expression, we will have the delay for packet k ($Delay = Dep_f^k - EAT_f^k$) as:

$$Delay \le EAT_f^m + T(E1) - EAT_f^k + T(E2) + \frac{L_f^k}{C_f^k} \tag{4.26}$$

If we show that $EAT_f^m + T(E1) - EAT_f^k \le 0$ we have proven that

$$Delay \le T(E2) + \frac{L_f^k}{C_f^k} \tag{4.27}$$

We can rewrite this condition, knowing that $EAT_f^k = EAT_f^m + V_2 - V_1$, as following:

$$T(E1) \le V_2 - V_1 \tag{4.28}$$

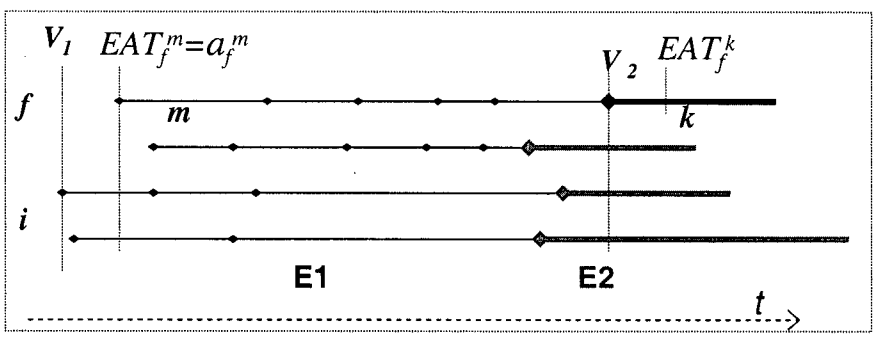

Figure 4-3 Delay Analysis of CAPS-SFQ

Note that up to this point we have not made any assumptions on the transmission rate of each packet, so condition (4.28) must be met for both dynamic and static multirate cases in order to get the delay bound in (4.27). To prove (4.28), we present the following lemmas under different multirate assumptions. As a first step we introduce the *stability condition* for a throughput fair SFQ.

Lemma 3: Given weights r_i and transmission rate C_i for individual queues in a throughput fair SFQ scheduler, the scheduler is stable and can provide the expected throughput of r_i, if the following condition is met for the flows:

$$\sum_{all\,flows\ i} \frac{r_i}{C_i} \leq 1 \tag{4.29}$$

Proof: A throughput fair system, with rates r_i assigned as weights for each queue, must provide $r_i(t-\tau)$ of service to each queue in an interval (τ, t). Knowing that each queue has a transmission rate C_i, we can calculate the time it takes to serve one queues traffic in (τ, t) as $r_i(t-\tau)/C_i$. Summing over all queues i, we know that the system is stable and able to provide the service, only if this sum of times is less than the original interval (τ, t) : $\sum_{all\,flows\ i} \frac{r_i(t-\tau)}{C_i} \leq (t-\tau)$. The lemma follows from this inequality. **Q.E.D.**

Lemma 4: In a Static Multirate SFQ, the time it takes to serve the traffic of set E1 is less than $V_2 - V_1$, for V_1 and V_2 as described in (4.24); i.e., we have $T(E1) \leq V_2 - V_1$.

Proof: Packets n (n from 1 to some x for each queue i) that belong to set E1 have the start and finish tags as follows:

$$S_i^1 \geq V_1, \quad F_i^x \leq V_2 \tag{4.30}$$

The maximum amount of traffic in E1 is observed when the queues are always backlogged, and we have $F_i^x = S_i^1 + \sum_{n=1}^{x} \frac{L_i^n}{r_i} \leq V_2$. Combining this equation with (4.30), we get the following:

$$\sum_{n=1}^{x} \frac{L_i^n}{r_i} \leq V_2 - V_1 \tag{4.31}$$

Dividing both sides of (4.31) by C_i and then taking a sum over all flows i we get:

$$\sum_{all\,flows\ i} \sum_{n=1}^{x} \frac{L_i^n}{C_i} \leq \sum_{all\,flows\ i} \frac{r_i}{C_i}(V_2 - V_1) \tag{4.32}$$

The left side of the above inequality is indeed $T(E1)$. Considering the result of Lemma 3 in (4.29) for the stability of a static multirate SFQ, we know that the right side of the above equation is always less than $(V_2\text{-}V_1)$, thus:

$$T(E1) = \sum_{all\,flows\ i} \sum_{n=1}^{x} \frac{L_i^n}{C_i} \leq \sum_{all\,flows\ i} \frac{r_i}{C_i}(V_2 - V_1) \leq (V_2 - V_1) \qquad \textbf{Q.E.D.}$$

Having proven the above lemmas, we now present the main result of this section for the static multirate case.

Theorem 1: The delay bound for any packet k belonging to queue f with weight (assigned rate) r_f and transmission rate C_f, in a stable throughput fair SFQ scheduler under static multirate assumptions, is as follows:

$$Delay_f^k \leq \sum_{j \in Q,\, j \neq k} \{\frac{L_j^{\max}}{C_j}\} + \frac{L_f^k}{C_f} \tag{4.33}$$

Proof: Following lemmas 3 and 4 we know that inequality (4.27) is valid. Also, since all packets in E2 contribute to the delay incurred by packet k after its EAT, we should consider the maximum length for these packets. Consequently, considering the transmission rate of each flow the theorem follows. **Q.E.D.**

The above theorem implies that the delay bound of one queue may in fact increase with the rate reduction in other queues. This fact is later examined in this chapter through experiments.

4.4.2 Delay Bound Analysis of Throughput Fair SFQ in Dynamic Multirate Case

To find the delay bound in dynamic multirate scenario we again need to prove that inequality (4.28) is valid, and packets belonging to set E1 do not contribute to the delay bound. To do so we first define the stability condition in the following lemma; we then prove, in lemma 6, that the condition in (4.28) is valid for this case.

Lemma5: Given weights r_i for each queue i and transmission rate C_i^k for each packet k of this queue in a throughput fair SFQ scheduler, the scheduler is stable and can provide the expected throughput of r_i for flow i, if the following condition is met in any interval (τ, t):

$$\sum_{all\,flows\ i} \sum_{n \in Bi} \frac{L_i^n}{C_i^n} \leq \frac{1}{r_i} \sum_{n \in Bi} L_i^n \tag{4.34}$$

Proof: Consider an interval $T=t-\tau$, for the server to be fair and able to guarantee the promised rate (be stable), it must be able to serve a set B_i of packets from queue i during T, such that the following inequality holds (the left side is the amount of received service):

$$\sum_{n \in Bi} L_i^n \geq r_i \cdot T \tag{4.35}$$

On the other hand we know that packets from set B_i take $\Delta T_i = \sum_{n \in Bi} \frac{L_i^n}{C_i^n}$ to transmit. Taking a sum over all queues i we must have $\sum_{all\ flows\ i} \Delta T_i = \sum_{all\ flows\ i} \sum_{n \in Bi} \frac{L_i^n}{C_i^n} \leq T$ in order to have a stable server. Combining this condition with (4.35) the proof completes. **Q.E.D.**

Lemma 6: In a Dynamic Multirate SFQ, the time it takes to serve the traffic of set E1 is less than $V_2 - V_1$ for V_1 and V_2, as described in (4.24); i.e., we have: $T(E1) \leq V_2 - V_1$.

Proof: Packets n (n from 1 to some x for each queue i) that belong to set E1 have the start and finish tags as follows:

$$S_i^1 \geq V_1, \quad F_i^x \leq V_2 \tag{4.36}$$

The maximum amount of traffic in E1 is observed when the queues are always backlogged and we have $F_i^x = S_i^1 + \sum_{n=1}^{x} \frac{L_i^n}{r_i} \leq V_2$. Combining with (4.36) we get the following:

$$\sum_{n=1}^{x} \frac{L_i^n}{r_i} \leq V_2 - V_1 \tag{4.37}$$

Considering an interval of length $T(E1)$, we know that packets served in this interval for all queues take:

$$T(E1) = \sum_{all\,flows\ i} \sum_{n=1}^{x} \frac{L_i^n}{C_i} \tag{4.38}$$

If the system is assumed stable, we know that inequality (4.34) from Lemma 5 is valid for the interval $T(E1)$; thus we have $T(E1) = \sum_{all\,flows\ i} \sum_{n=1}^{x} \frac{L_i^n}{C_i} \leq \sum_{n=1}^{x} \frac{L_i^n}{r_i}$. Knowing that for any queue in set E1 we have $\sum_{n=1}^{x} \frac{L_i^n}{r_i} \leq V_2 - V_1$ (according to (4.36)), the lemma is proved. **Q.E.D.**

Theorem 2: The delay bound for any packet k belonging to queue f with weight (assigned rate) r_f and transmission rate C_f^k in a stable throughput fair SFQ scheduler under dynamic multirate assumptions is as follows:

$$Delay_f^k \leq \sum_{j \in Q,\, j \neq k} \{\frac{L_j^{\max}}{C_j^{\min}}\} + \frac{L_f^k}{C_f^k} \tag{4.39}$$

where($L_j^{\max}$, $C_j^{\min}$) are a pair of length and transmission rates for queue j that have the maximum transmission duration.

Proof: Following lemmas 5 and 6 we know that inequality (4.27) is valid. Also, since all packets in E2 contribute to the delay incurred by packet k after its EAT, we should consider the maximum length for these packets. Consequently, considering the transmission rate of each flow the theorem follows. **Q.E.D.**

4.4.3 Delay Bound of Temporal-Fair SFQ in Static and Dynamic Multirate Cases

To find the scheduling delay bound for the temporal fair SFQ (STFQ), we can use the same concept of Expected Arrival Time as in the previous section. The scheduling delay bound is a measure of other flows' effect on the departure time of a given packet in a flow; whereas, EAT represents the time we expect the packet to reach the head of line in its own flow and start service. EAT calculation in (4.22) assumes that each queue is served by a fluid server at a rate which is in fact the weight assigned to the queue in throughput fair SFQ. For Temporal fair SFQ, the relation between the rate of the hypothetical fluid server and the weights assigned to each queue is not straight forward. The reason is that a temporal fair scheduler does not directly guarantee throughput; especially in dynamic multirate scenarios, the fixed weight for a queue of STFQ does not guarantee a fixed throughput for it. However, for the static multirate scenario, the fixed weight φ_i, and transmission rate C_i of a queue i do indeed guarantee a throughput of $r_i = C_i \cdot \varphi_i$ (if $\sum_{j\ all\ queues} \varphi_j = 1$).

We can model the fluid server in the temporal fairness case as a server with rate $r_i = C_i \cdot \varphi_i$ for the static multirate case, and a variable rate server with per packet rate of $r_i^k = C_i^k \cdot \varphi_i$ for the dynamic multirate scenario. Employing these rates in (4.22) gives us the expected arrival

time in the temporal fair system. Since these values for rates are indeed the same weights as described in (4.15) and (4.19), the temporal fair system can be viewed as a throughput fair system with the above rates as the weights. Consequently, the analysis of the previous section is still applicable, which suggests that the traffic in set E1 does not contribute to the scheduling delay bound. As a result, the following theorem holds.

Theorem 3: In a static (or dynamic) multirate scenario, an STFQ scheduler with transmission rate C_i (or C_i^k) for each queue i (and packet k), can guarantee the following scheduling delay bound for each queue f and packet k:

$$Delay_f^k \le \sum_{j \in Q, j \ne k} \{\frac{L_j^{\max}}{C_j}\} + \frac{L_f^k}{C_f} \qquad \text{for static multirate case} \qquad (4.40)$$

$$Delay_f^k \le \sum_{j \in Q, j \ne k} \{\frac{L_j^{\max}}{C_j^{\min}}\} + \frac{L_f^k}{C_f^k} \qquad \text{for dynamic multirate case} \qquad (4.41)$$

The stability conditions for the throughput fair SFQ (as described in Lemma 3) change to the following condition in temporal fair SFQ, which is already true: $\sum_{j \text{ all queues}} \varphi_j \le 1$.

It is imperative to note that although the scheduling delay bounds are the same for throughput and temporal fair SFQ, the guaranteed throughput and consequently the cumulative delay bounds are quite different. This is because the total delay (cumulative delay) a packet may incur is very much dependent on the throughput it receives from the scheduler and is equal to the sum of scheduling and queuing delays. For example, a packet k that arrived in a backlogged queue with n other packets has to wait at most $\sum_{n=1}^{k} (L_i^n / r_i^n)$ before it reaches the head of line, if a rate of r_i^n is guaranteed for the n'th packet of this queue. The study of

queuing delay is a well developed subject and is not repeated here. It is worth to note that to derive the bounds on the queuing delay we need to assume a traffic shape such as token bucket regulated traffic for the incoming flow and consider the throughput provided to the flow by the scheduler. Consequently, well-known mechanisms such as the one described in [25] will provide the queuing delay.

4.5 Performance Evaluation

To evaluate the performance of CAPS-SFQ in multirate environments, we conducted several simulation experiments using an OPNET based 802.11e simulator that we have developed at UBC. We observed different features of throughput and temporal fair CAPS-SFQ , and examined the difference between the two fairness modes. We assumed an 802.11b PHY in these experiments.

In our first experiment, we observed the throughput granted to two stations (STA1 and STA2) in a static multirate case by throughput-fair CAPS-SFQ and by EDCA. We considered the transmission rate for STA1 to be 2Mbps, while other stations including STA2 had 11Mbps transmission rate. The number of stations increased from 5 to 55; each station had a load of 200 Kbps (packet size uniformly distributed in [250,1750]). The virtual packets generated for STA1 and STA2 in CAPS-SFQ had a rate of 200Kbps with 1000B packets. The results, depicted in Figure 4-4, show that throughput-fair CAPS is able to guarantee the 200Kbps throughput for both STA1 and STA2, despite the lower transmission rate for STA1. This is not the case for EDCA that provides a lower throughput for STA1.

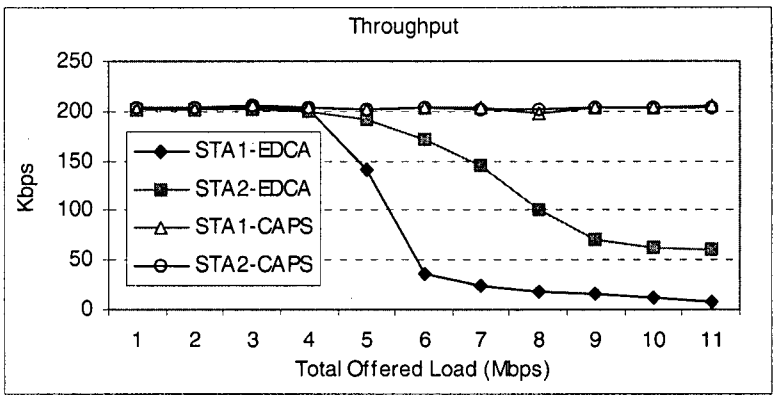

Figure 4-4 The ability of Throughput Fair CAPS-SFQ to guarantee bitrate

Providing throughput fairness to a low transmission rate station may degrade the level of service received by other stations. We observe this fact in another experiment in which we measured the throughput received by stations with transmission rate 2Mbps and 11Mbps. In this experiment, CAPS-SFQ was used and the two modes of throughput and temporal fairness were considered. We increased the number of 300Kbps stations in this experiment, to see how the scheduler divides the available capacity. As it is seen from Figure 4-5, when temporal fairness is used, the high bitrate stations do not see service degradation since they are isolated from the effect of service loss for other stations. However, when throughput fairness is considered, both types of stations suffer equally. In effect, service from stations with 11Mbps is taken and given to lower rate stations.

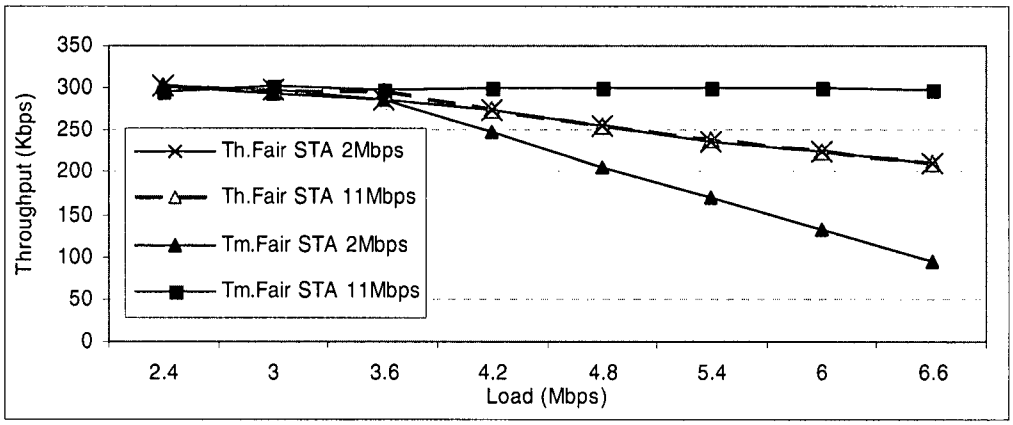

Figure 4-5 Throughput Fairness vs. Temporal Fairness

We repeated the above experiment with 12 stations (to keep the network stable) and obtained the distribution of the packet delays for each type of station. The delay measurements were done for both cases of temporal and throughput fair CAPS-SFQ and for EDCA. The results, depicted in Figure 4-6, show that high bitrate stations have the best delay performance under temporal fair CAPS-SFQ, while by compromising service for these stations we can achieve reasonable service levels for both types of stations using throughput fairness (under the conditions of the experiment). The general idea is to allow throughput fairness for as long as it is possible to meet the minimum service requirements of all stations. When the load increases beyond this point, temporal fairness must be used to guarantee that at least stations with higher transmission rate (good channel condition) receive the promised service.

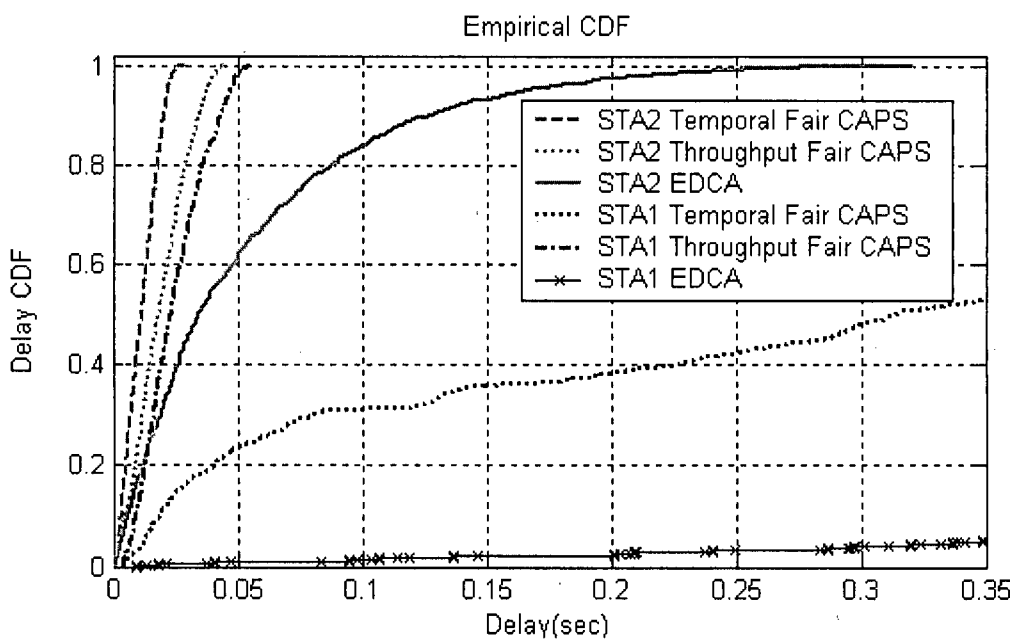

Figure 4-6 Delay distribution for CAPS (Temporal & Throughput Fair) and EDCA

4.6 Concluding Remarks

This chapter extends the solution proposed in chapter 3, and presents a design for providing per flow service guarantees in the MAC layer of a multirate WLAN. The solution introduced in this chapter employs the CAPS-enabled centralized scheduling of uplink and downlink flows in the access point, and uses a modified SFQ based algorithm along with traffic shaping to provide guaranteed services, while providing either throughput or temporal fairness. Application such as real-time Voice and Video over WLAN will greatly benefit from this design, because of the inherent similarity of their operational environment to the cases that are targeted by the proposed solution. Currently, we are examining the application of our design to other networks such as IEEE 802.16, and the integration of the presented algorithms with the power management features of the 802.11e standard.

4.7 References

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.

[2] IEEE 802.11e Standard – Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, July 2005.

[3] T. Nandagopal , S. Lu and V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms", Wireless Networks, vol. 8 , Issue 2/3, pp. 231 - 247, March-May 2002.

[4] S. Lu, T. Nandagopal and V. Bharghavan, "Fair scheduling in wireless packet networks", in: ACM MOBICOM'98 (October 1997).

[5] T.S. Ng, I. Stoica and H. Zhang, "Packet Fair Queueing Algorithms For Wireless Networks With Location-Dependent Errors", Proc. of IEEE INFOCOM '98, vol. 3, pp. 1103–1111, March 1998.

[6] A. Kamerman and L. Montean, "WaveLAN-II: A High-Performance Wireless LAN far the Unlicensed Band", Bell Labs Technical Journal, pp. 118-133 , 1997.

[7] S. A. Khan, "Adjacent Channel Interference in Overlapping Access Points of OFDM WLANs", M.A.Sc Thesis, University of British Columbia, February 2006.

[8] Y. Pourmohammadi-Fallah, A. Elfeitori, H. Alnuweiri, "A Unified Scheduling Approach for Guaranteed Services over IEEE 802.11e Wireless LANs", Broadband Networks, First Int. Conf. on, pp. 375 – 384, Oct 2004.

[9] J. Rosenberg, et.al., "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[10] Q. Ni, L. Romdhani, and T. Turletti. "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN". Wiley Journal of Wireless Communication and Mobile Computing (JWCMC), John Wiley and Sons Ltd., 2004; vol. 4, Issue 5, pp. 547-566, 2004.

[11] R.S. Ranasinghe, L.L.H. Andrew, D.A. Hayes, D. Everitt, "Scheduling disciplines for multimedia WLANs: embedded round robin and wireless dual queue", Communications, 2001. ICC 2001. IEEE Int. Conf. on , vol. 4 , pp:1243 – 1248, June 2001.

[12] W. Pattara-Atikom, P. Krishnamurthy, S. Banerjee, " Distributed mechanisms for quality of service in wireless LANs", Wireless Communications, IEEE, vol. 10 , Issue: 3 pp:26 – 34, June 2003.

[13] P. Ansel, Q. Ni, and T. Turletti, "An efficient scheduling scheme for IEEE 802.11e", WiOpt'04: Modeling and Optimization in Mobile, AdHoc and Wireless Networks, 2004.

[14] A. Grilo, M. Macedo, and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks", IEEE Wireless Communications, vol. 10, Issue 3, pp. 36-43, June 2003.

[15] A. Banchs, M. Radimirsch, X. Perez, " Assured and expedited forwarding extensions for IEEE 802.11 wireless LAN" Quality of Service, 2002. Tenth IEEE Int. Workshop on, pp. 237 – 246, March 2002.

[16] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," IEEE/ACM Trans. on Networking, vol.1, no. 3, pp. 344-357, June 93.

[17] J.C.R. Bennett, H. Zhang, "WF^2Q: Worst-Case Fair Weighted Fair Queueing", INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE, Vol. 1, pp. 120–128, March 1996.

[18] P. Goyal, H.M. Vin, H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm For Integrated Services Packet Switching Networks", Networking, IEEE/ACM Transactions on, vol. 5, Issue 5, pp. 690-704, Oct. 1997

[19] Y. Pourmohammadi Fallah and H. Alnuweiri, "Hybrid Polling and Contention Access Scheduling in IEEE 802.11e WLANs", Journal of Parallel and Distributed Computing, Elsevier, Vol. 67, Issue 2, pp. 242-256, February 2006.

[20] Y. Pourmohammadi Fallah, H. Alnuweiri "A controlled-access scheduling mechanism for QoS provisioning in IEEE 802.11e wireless LANs" ACM international workshop on Quality of service & security in wireless and mobile networks Q2SWinet '05, pp. 122 - 129 , 2005.

[21] Y. Pourmohammadi Fallah, H. Alnuweiri, "Enhanced Controlled-Access and Contention-Based Algorithms for IEEE 802.11e Wireless LAN", Wireless Networks, Communications and Mobile Computing, 2005 International Conference on, vol. 1, pp. 409 – 414, June 2005.

[22] Y. Yuan, D. Gu, W. Arbaugh, J. Zhang, " Achieving Fair Scheduling Over Error-Prone Channels in Multirate WLANs ",Wireless Networks, Communications and Mobile Computing, 2005 International Conference on, vol. 1, pp. 698 – 703, June 2005.

[23] I. Tinnirello, S. Choi, "Temporal fairness provisioning in multi-rate contention-based 802.11e WLANs", World of Wireless Mobile and Multimedia Networks, 2005. Sixth IEEE International Symposium on a, pp. 220 – 230, June 2005.

[24] Y. Pourmohammadi Fallah , H. Alnuweiri, "Performance Analysis of Controlled Access Phase Scheduling Scheme for Per-Session QoS Provisioning in IEEE 802.11e Wireless LANs", Proc. of 2006 Wireless Communications and Networking Conf., vol. 3,pp. 1414-1420 , April 2006.

[25] J.Y. Le Boudec, "Application of Network Calculus To Guaranteed Service Networks", IEEE Trans. on Information Theory, vol. 44, Issue 3, May 1998.

[26] R. Agrawal, R. L. Cruz, C. Okino and R. Rajan, "Performance Bounds for Flow Control Protocols", IEEE Trans. on Networking, vol. 7, no. 3, pp 310—323, June 99.

[27] H. Alnuweiri, H. Tayyar, "Analysis Of Virtual-Time Complexity In Weighted Fair Queuing", Computer Communications, vol. 28, Issue: 7, pp. 802 – 810, 2005.

Chapter 5. Summary and Conclusions

5.1 Summary and Concluding Remarks

This research was motivated by the need for providing QoS in wireless local area networks. The goal of this thesis is to devise a scheduling framework that provides per-session QoS to multimedia applications in a WLAN such as IEEE 802.11e. Multimedia applications are demanding applications with stringent QoS requirement. In particular, application such as voice and video telephony, require low delay and jitter services from a WLAN.

We studied several proposals that provide QoS solutions for wireless networks, and identified that these solutions lack the features required for QoS provisioning in multirate CSMA/CA based wireless networks. In particular we identified the following three features that are required from any QoS solution: 1) the ability to schedule both uplink and downlink directions at the same time; 2) efficiently dividing the use of wireless channel between contention and controlled access modes (EDCA and HCCA) while providing guaranteed services with low access delay to HCCA sessions 3) the ability to maintain throughput or temporal fairness in multirate environments. We developed a hybrid polling solution that possesses all these features and provides a flexible framework that allows for further extension.

To gain detailed knowledge of the operation of the 802.11e MAC, we implemented all the features of the standard in software (C language). We imported the implementation into the OPNET simulation tool and evaluated the features of the MAC layer under different scenarios.

We verified the results from our simulator with those of other published works and ensured the accuracy of the implementation. Through our study we identified that the existing throughput analyses of the EDCA under saturation mode were incomplete or flawed. We corrected the existing models and presented a generalized analysis of saturation throughput for EDCA in Chapter 2. Analytical models of EDCA under finite load conditions (non-saturated networks) are presented in other researchers' works.

Following our analysis and several simulation experiments we observed that the performance of EDCA mechanism is very sensitive to its parameters and network conditions. As a result, we considered HCCA based solutions and presented CAPS. CAPS, described in Chapter 3, is a hybrid scheduling mechanism that achieves the goal of per-session guaranteed service provisioning through the use of virtual packets, GPS based scheduling, and traffic shaping. CAPS utilizes GPS based scheduling at its core. Methods such as WFQ , WF^2Q or SFQ were examined for CAPS and it was shown that for situations where exact knowledge of uplink traffic is not available in the access point CAPS may behave differently from its core scheduler (except for CAPS-SFQ). Nevertheless, it was proved that CAPS is able to provide per session guaranteed services.

Extending the design in Chapter 3, we examined the effects of multirate operation of a WLAN on the fairness of our scheduling algorithm in Chapter 4. We presented a detailed analysis of throughput and temporal fairness and extended the definition of GPS scheduling to introduce temporal or throughput fair GPS. We then presented temporal and throughput fair version of the packet based approximations of GPS (WFQ, WF^2Q and SFQ).

Considering the issues of dynamic multirate operation in WLANs we demonstrated that although the implementation of throughput or temporal fair CAPS-WFQ and CAPS-WF^{2}Q in dynamic multirate WLANs is possible, it is computationally very costly. Knowing from Chapter 3 that CAPS-WFQ and CAPS-WF^{2}Q do not have an advantage over CAPS-SFQ in single rate WLANs, we recommend fair CAPS-SFQ as our QoS solution in multirate WLANs. We presented several simulation experiments in Chapter 4 to demonstrate that throughput and temporal fair CAPS-SFQ is in fact able to provide the expected services and outperforms other methods such as EDCA (which can be made throughput or temporal fair). The work presented in Chapter 3 and Chapter 4 provides a complete scheduling solution for per-session QoS provisioning in multirate 802.11e WLANs.

5.2 Summary of Contributions

The main contributions of this thesis can be summarized as follows:

- A unified scheduling framework has been introduced that uses the concept of virtual packets (representing uplink flows) and combines the scheduling of virtual packets with actual downlink packets to achieve the goal of centralizing the task of uplink/downlink scheduling in the access point.

- A new queuing/scheduling model has been developed that uses traffic shaping and GPS based scheduling in the above unified scheduling framework and achieves efficient scheduling of HCCA and EDCA based access. This model provides guaranteed access services for HCCA flows while sharing the remaining capacity in a contention based manner using EDCA.

- An analysis of temporal and throughput fairness based on the concept of GPS based scheduling has been presented. This analysis was used to devise and analyze temporal and throughput fair versions of WFQ, WF2Q and SFQ algorithms.

- An analysis of the performance of CAPS-SFQ in multirate environments has been presented, deriving the delay bounds provided by this algorithm under dynamic and static multirate scenarios.

- A generalized saturation throughput analysis of EDCA has been presented that allows for an arbitrary number of priority levels to be considered; extending and correcting the existing methods.

- A complete implementation of the 802.11e MAC and the proposed scheduling framework have been done and tested in OPNET simulation environment.

It is important to note that providing QoS for multimedia applications is crucial to most service providers as it enables efficient deployment of revenue producing applications such as voice and video telephony and TV over WLANs. Applications such as voice only WLANs or real time video surveillance are examples of applications that are currently used, but at very low efficiency. We have shown in Chapter 3 that using CAPS we can significantly boost the capacity of the network while still maintaining fairness and per-session service protection required by such applications.

Another possible application of CAPS is in WLANs that inter-network with Wireless Wide Area Networks such as 3rd generation (3G) or 4th generation (4G) cellular networks. It is expected that in future networks, 3G or 4G mobile users that enter the service area of a WLAN

switch to the WLAN and use its higher bitrate services. Such service transfer requires that the WLAN be able to provide the same level of QoS that is provided by 3G and 4G networks. With popularity of live multimedia applications in cellular networks, it is imperative that WLANs support such application too. CAPS design is perfectly suitable for such scenarios.

5.3 Future Research Directions

There are a number of related issues that can be investigated as an extension to the presented research. We present some of these subjects in this section.

In designing CAPS we assumed the availability of some controlled access mechanisms of the 802.11e WLANs; however, we can easily apply the design and idea of CAPS to any multiple access network that uses controlled access mechanisms such as polling. For example, wireless metropolitan area networks (WMAN) such as IEEE 802.16 can use the proposed solution with some modifications.

Power management is one of the important issues for low power WLAN devices. Given the ability of CAPS (and HCCA) to schedule access for individual stations in a deterministic fashion, one can extend these methods to take into account the power management issues as well. For example, a VoIP station that uses CAPS to send its uplink packets can reduce its ON durations if it receives periodic services from the access point, while using contention based methods requires more transmission attempts, thus higher power usage.

Admission Control is a topic that has received much attention in recent years. We do not consider any specific admission control mechanism in this dissertation. The standard does not mandate such methods either. An analysis of different admission control mechanisms and their

performance in multirate WLAN environments can complement the work presented in this thesis.

Mobility is an important aspect of most wireless networks. Mobility in WLANs is supported in the LLC and MAC layers. Studying the effects of mobility on the performance of polling based mechanisms such as CAPS is another issue that can be beneficial in devising a complete QoS solution for mobile WLANs or similar networks.

APPENDICES

Appendix A. Modeling 802.11 DCF

In this appendix, we present a summary of the model developed by Bianchi[1] to derive the equations governing the backoff process behavior of DCF or a single queue in EDCA. We verified this model to be an accurate representation of the station behavior in saturation mode. Using this model the probability τ of a stations attempting transmission in a slot time is found as a function of n (the network size), W (the minimum contention window size), and m, which describes the maximum backoff stage and expresses the maximum contention window size as $2^m \cdot W$. In each stage of retransmission the contention window size is $W_i = 2^i \cdot W$.

As previously explained, in saturation mode all stations always have frames available for transmission, which means that for each frame transmission the stations have to run the backoff procedure. Since the backoff value is derived randomly we can model it as a random variable $b(t)$. The value that is derived for the backoff counter at the beginning of each backoff process depends on the previous retransmission history of the system, thus $b(t)$ is non-Markovian. To handle this problem Bianchi uses another process $s(t)$ to model the contention window size and models the backoff procedure using a two dimensional process $[s(t), b(t)]$. The backoff state represented by this two dimensional stochastic process is Markovian and can be modeled as in Figure A-1 with the following state transition probabilities:

$$
\begin{aligned}
&P\{i,k|i,k+1\} = 1 && k \in (0, W_i - 2) \quad i \in (o,m) \\
&P\{0,k|i,0\} = (1-p)/W_0 && k \in (0, W_0 - 1) \quad i \in (o,m) \\
&P\{i,k|i-1,0\} = p/W_i && k \in (0, W_i - 1) \quad i \in (1,m) \\
&P\{m,k|m,0\} = p/W_m && k \in (0, W_m - 1)
\end{aligned} \tag{1}
$$

[1] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE Journal on Selected Areas in Communications, vol. 18, no. 3, March 2000.

One important assumption in this model is that the probability p that a transmission attempt in a slot results in collision, is constant in each slot and is independent of the retransmission history of the system. Solving the markov chain model, we find the following equation that expresses the probability of transmission:

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)} \tag{2}$$

In the case of DCF, using the fact that stations whose backoff counters reach zero will attempt a transmission, we can find the probability of collision as a function of τ:

$$p = 1-(1-\tau)^{n-1} \tag{3}$$

Note that (3) is not applicable to EDCA since probability of collision is dependent on the transmission probability of other classes as well. For DCF, one can solve (2) and (3) using numeric methods and find τ and p based on the known values of *CWmin* , *CWmax*, and *n*.

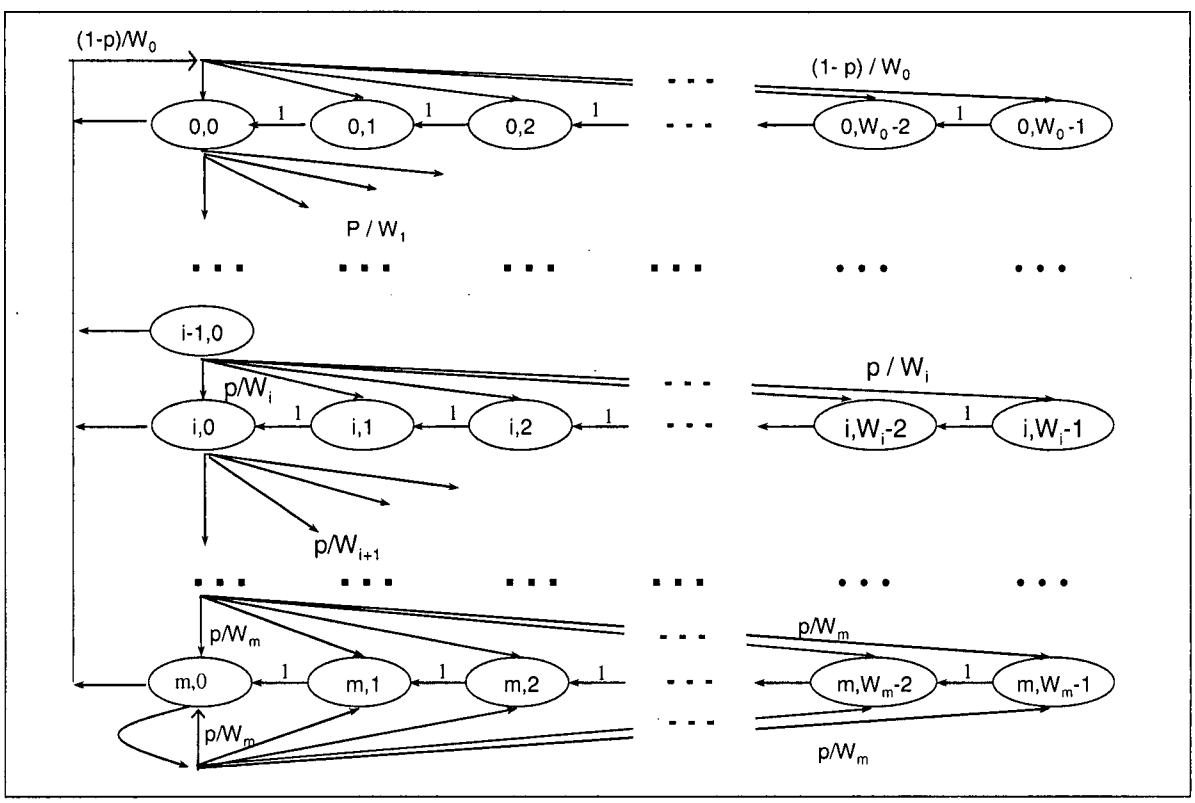

Figure A-1 Markov model for the backoff procedure

Appendix B. Performance of H.264 Video over 802.11e WLAN

The H.264 standard has recently emerged as the most prominent video compression technology. Considering this fact, most of our experiments with video files, reported in this thesis, use the traces from H.264 encoded video sequences. To further analyze the performance of H.264 video transmission over WLANs and examine the quality of the delivered video, we have devised a simulation framework that allows simulating real-time video communications scenarios using an offline simulator. This framework, depicted in Figure B.1, consists of a core network simulator (in this case, the 802.11e OPNET model) and several interfaces that apply the simulation result to a decodable H.264 file containing the video sequence.

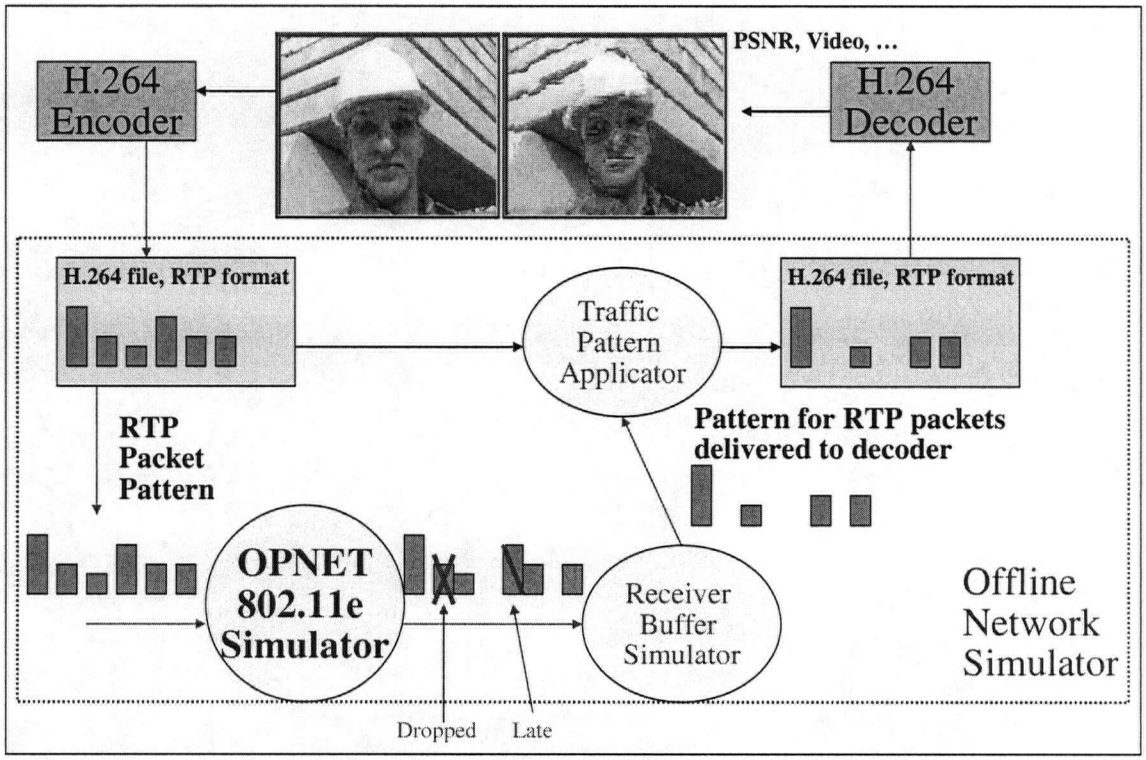

Figure B.1 Offline Video Communication Simulator

Using the above framework we have conducted several experiments to observe the quality of an H.264 video transmitted over an 802.11e WLAN. We considered the physical layer errors and the effect of MAC layer operation. The WLAN that was used for this experiment was comprised of one uplink video source (CIF size H.264 encoded foreman video with 500Kbps bitrate and 700 Byte slice sizes) and a number of stations generating background traffic (30 stations, 200Kbps bitrate, with 1000Byte packets with exponential inter-arrival). We considered two cases of very low PHY error (10^{-6}, denoted as no error in Figure B.2) and high PHY error of 10^{-5}. We also repeated the case with PHY errors for 6 background stations in order to verify that the difference between CAPS and EDCA exists even in lightly loaded networks. The cumulative distribution function of the measured delay in each case is depicted in Figure B.2.

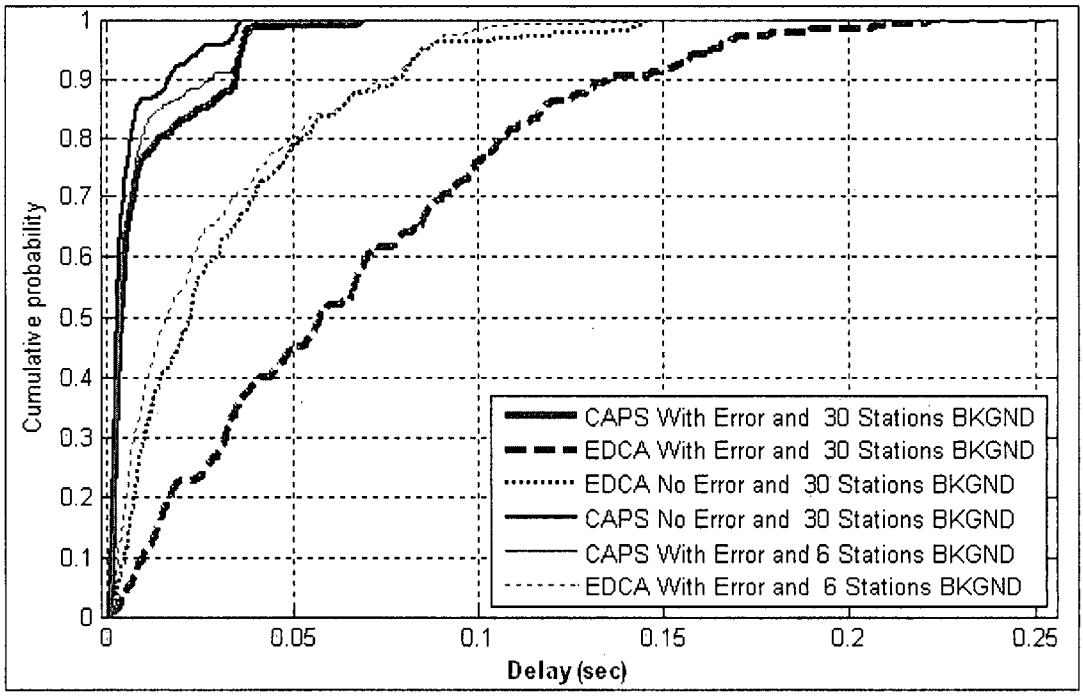

Figure B.2 CDF of delay in a WLAN with and without PHY errors

From Figure B.2 we see that introducing errors in the PHY layer has a significant effect on EDCA operation because it incurs retransmission, effectively increasing the load of the network and the probability of collision. The PHY error effects are very limited in CAPS. It is also seen that although the average delay for the EDCA case is acceptably low, the jitter levels are very high. This is seen from the CDF of the delay for EDCA case, where it is spread over a wide range.

Using the offline network simulator, depicted in Figure B.1, we applied the effects of PHY errors and MAC delay to the 500Kbps foreman video (from the experiment) and observed the output video quality. Some snapshots of the played back video are depicted in Figure B.3. We considered two different delay deadlines of 100msec and 250msec for the received packets, and the EDCA case with 30 stations. As we expected, CAPS performance was clearly superior to that of EDCA and the video quality is considerably better. The delay deadlines and the simulation scenario are shown on each snapshot window in Figure B.3.

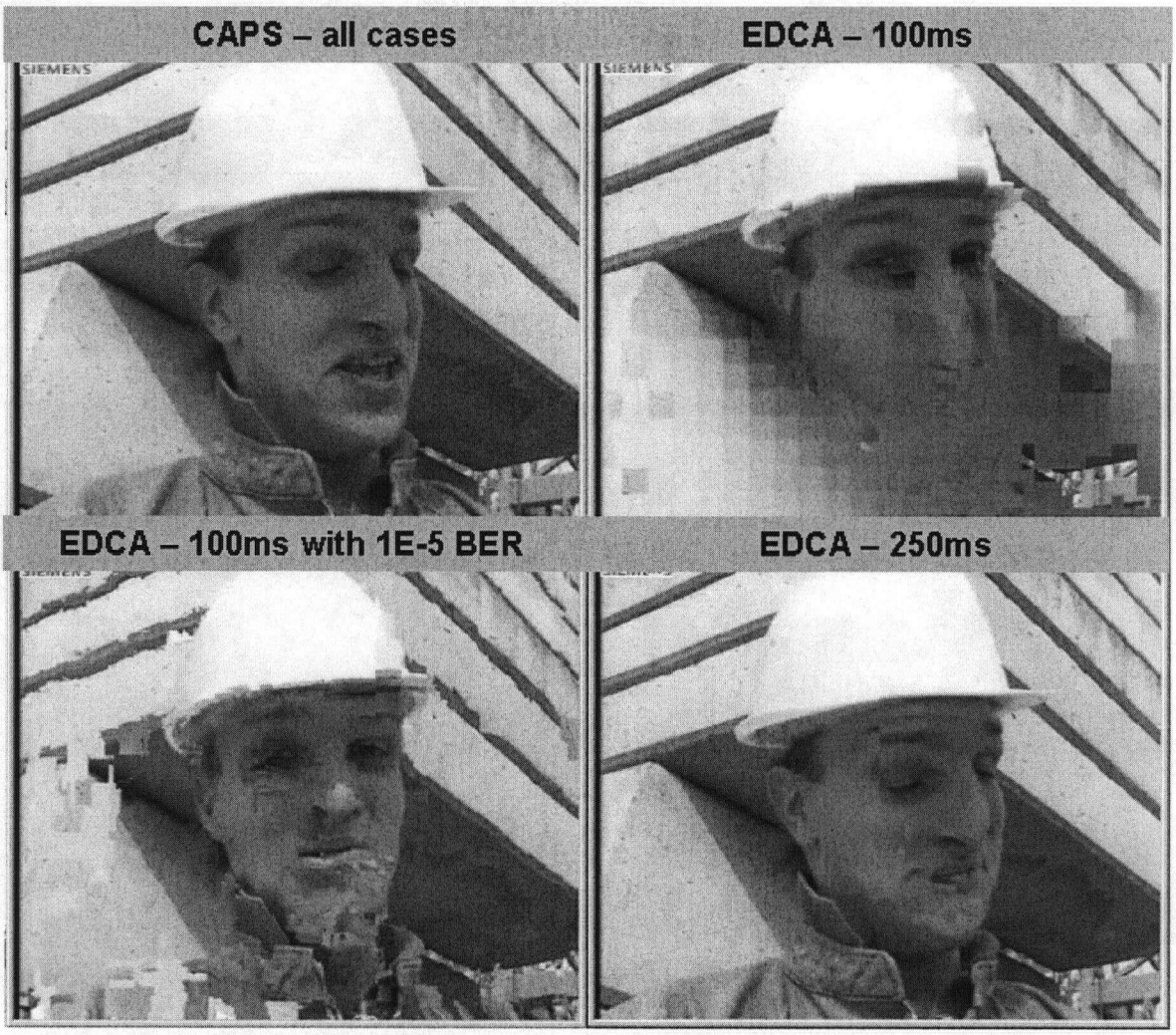

Figure B.3 Snapshots of foreman video, transmitted in four different WLAN scenarios

Appendix C. Analysis of Throughput and Temporal Fair WFQ and WF^{2}Q

Service discrepancy bounds for Throughput or Temporal WFQ in multirate environments are indeed very similar to the single rate case. The bound on the lag of WFQ compared to GPS, in serving any packet, is found by Parekh[1] to be $(L^{\max}/C)$. This bound is found considering a scenario in which a packet m (with length $L_m = L^{\max}$), that has a GPS finish time of $u_m \geq u_i$ { i:m+1 .. k}, is served ahead of packets m+1 to k in WFQ; thus packet k has a WFQ service completion time of $t_k \leq u_k + \frac{L_m}{C}$, behind its GPS finish time u_k. This is shown using the following facts:

$$u_k \geq \min\{a^i\} + \{time\ it\ takes\ to\ serve\ m+1\ to\ k)\ \ and \qquad \min\{a^i\} \geq t_m - \frac{L_m}{C}$$

$$u_k \geq \sum_{i=m+1}^{k} \frac{L_i}{C} + t_m - \frac{L_m}{C} = t_k - \frac{L_m}{C}$$

We can apply the same argument to the multirate case in which each packet is served at a different rate. The above expression changes in this case to:

$$u_k \geq \sum_{i=m+1}^{k} \frac{L_i}{C_i} + t_m - \frac{L_m}{C_m} = t_k - \frac{L_m}{C_m}$$

Note that the above inequality holds for the multirate case because the time that it takes to serve packets m+1 to k in either the throughput or temporal fair GPS and WFQ is $\sum_{i=m+1}^{k} \frac{L_i}{C_i}$,

[1] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case", IEEE/ACM Trans. on Networking, vol.1, no. 3, pp. 344-357, June 93.

and we have already defined k as the last packet served in WFQ order. Therefore, for the multirate scenario we have:

$$t_k \le u_k + \frac{L_m}{C_m} \qquad \text{where we find } m \text{ so that: } \frac{L_m}{C_m} \ge \max\{\frac{L_i}{C_i}\} \quad \text{for all packets } i.$$

The same service lag bound of WFQ is applicable to WF^2Q (proven in the article by Bennet[2]). The justification is that the bound in WF^2Q is found by modeling the scheduler as a Regulated or R-WFQ and the GPS server as a Regulated GPS or R-GPS. Since regulated schedulers are only used to delay the patterns of arrivals until the GPS service eligibility time, the order and times of departure remain unchanged[2]. Such an adjustment is still possible and correct under either temporal or throughput fair multirate cases.

The bounds given above are found based on the assumption that the uplink packet lengths are known. Also we assume that advanced knowledge of the transmission rate of each packet is available. For situations where these two assumptions do not hold, these bounds have to be re-evaluated. The reason is that in the above analysis we assumed that the order of service in the inner scheduler and CAPS are the same, which may not be true when dynamic multirate or short response cases are considered.

[2] J.C.R. Bennett, H. Zhang, "WF^2Q: Worst-Case Fair Weighted Fair Queueing", INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE, Vol. 1, pp. 120–128, March 1996.