Implementation and Evaluation of IEEE 802.11e Wireless LAN in GloMoSim

Jahanzeb Farooq, Bilal Rauf

February 2006
Master Thesis, 2x20 Credits
Supervisor: Thomas Nilsson
Examiner: Per Lindström

Department of Computing Science
Umeå University
Sweden

Umeå University

Abstract

IEEE 802.11 Wireless Local Area Network (WLAN) is one of the most widely deployed wireless network technologies in the world today. The success of IEEE 802.11 mainly owes to its cost effectiveness, easy deployment and high data transmission rates.

However, 802.11 MAC (Medium Access Control) algorithm is unable to support modern multimedia applications which require certain level of quality of service (QoS) guarantees in terms of consistent, in time and reliable data transfer. This lack of QoS support is a big hurdle in the evolution of multimedia applications over IEEE 802.11 networks.

Therefore, enabling QoS support in the IEEE 802.11 networks has been the focus of research activities for some time. The IEEE 802.11 Working Group is currently working on an enhanced version of the 802.11, known as 802.11e, to provide support for QoS. The access mechanism of 802.11e, referred to as Enhanced Distributed Channel Access (EDCA), assigns different types of data traffic with different priorities based on the QoS requirements of the traffic, and for each priority, uses a different set of medium access parameters to introduce QoS support.

This thesis presents the implementation of IEEE 802.11e EDCA simulation model in GloMoSim (Global Mobile Information System Simulator), which is a popular simulation software for wireless networks. Next it presents the performance evaluation of IEEE 802.11e EDCA with the help of a wide range of simulation scenarios in GloMoSim. The evaluation also includes the study of the role of different parameters IEEE 802.11e EDCA utilizes to realize service differentiation as well as the performance comparison of 802.11 and 802.11e.

The evaluation results depict that IEEE 802.11e introduces an effective and optimal mechanism to provide QoS support based on the requirements of traffic and thereby significantly improves the overall network performance.

Acknowledgments

First of all, we would like to express our heartiest gratitude to our supervisor Thomas Nilsson, for his all time guidance, support, and valuable suggestions. He also taught us the course *Mobile and Wireless Networks* and provided us with very comprehensive understanding of wireless communications, networks and protocols. He impressed us both as a teacher and researcher, and inspired us to carry research on quality of service in wireless networks. We have learnt a lot from him, from course lectures and from our numerous discussions.

We are very grateful to the Director of Studies, Per Lindström, for his cooperation and for managing all the necessary things.

Finally but most importantly, we would like to thank our parents for their support, patience, blessings and understanding while we were writing this thesis.

Jahanzeb Farooq, Bilal Rauf
Umeå, January 2006

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Specification

1.1.1 Background

The IEEE 802.11 Wireless Local Area Network (WLAN) [1] is one of the most widely deployed wireless network technologies in the world today. With the enhanced versions 802.11b and 802.11a, it supports data transmission rates of up to 11 and 54 Mbps, respectively.

The basic MAC (Medium Access Control) mechanism of 802.11 is called Distributed Coordination Function (DCF). DCF is based on distributed channel access and employs CSMA (Carrier Sense Multiple Access) protocol for the medium access. IEEE 802.11 also defines an optional access mechanism, called Point Coordination Function (PCF), based on centrally controlled access. Most of the 802.11 installations today use DCF, whereas the PCF is hardly implemented mainly due to its complex design and inefficient access mechanism.

Although IEEE 802.11 has become more and more popular due to its low cost and easy deployment, it does not provide quality of service (QoS) support. QoS refers to the ability of network to provide some consistent services for data transmission, and measured in terms of qualitative characteristics, such as throughput, delay, jitter and packet loss, which describes quality of data traffic over a network. Basically all types of data traffic are treated equally in both DCF and PCF, regardless of the QoS requirements of the traffic, which vary from application to application. Specifically, multimedia applications such as audio/video streaming, teleconferencing, Internet telephony and interactive games require certain level of QoS guarantees. Lost packets or delays can seriously destroy the performance of these applications. Some kind of service differentiation must be employed to let higher priority multimedia traffic get better served. This inability of 802.11 MAC mechanism in providing QoS support is a big hurdle in the adaptation of modern multimedia applications in 802.11 networks.

Thus, a lot of research works have been carried out to enhance the QoS support in IEEE 802.11 networks. IEEE 802.11 Working Group is currently focusing on an enhanced version of IEEE 802.11, known as 802.11e [2], in order to support Quality of Service. IEEE 802.11e is in its standardization process and the final draft has been released. IEEE 802.11e defines two new mechanisms for distributed and centrally controlled access, which are basically the improved versions of DCF and PCF in the original standard. The distributed access mechanism of 802.11e is called EDCA (Enhanced

Distributed Channel Access), and supports Quality of Service by introducing service differentiation. Different types of traffic are assigned with different priorities based on their QoS requirements, and service differentiation is introduced by using a different set of medium access parameters for each priority.

1.1.2 Task

The task is divided into two parts. The first part is to implement the IEEE 802.11e EDCA in GloMoSim (Global Mobile Information System Simulator) [3], which is a simulation software for mobile and wireless networks, developed by Parallel Computing Laboratory at UCLA. GloMoSim is built using a layered approach similar to the OSI seven layers network architecture, with standard APIs between the layers. A wide range of protocols are supported at each layer. IEEE 802.11 DCF is currently one of the several protocols available at the MAC layer.

The second part of the task is to evaluate the performance of IEEE 802.11e EDCA in comparison to IEEE 802.11 DCF using the implemented 802.11 EDCA. More specifically, the goal is to study the effectiveness of EDCA's service differentiation mechanism in providing QoS support to different types of data traffic and to evaluate its performance with the help of a wide range of traffic scenarios and performance metrics.

1.2 Outline

The report is organized as follows: Chapter 2 presents an overview of IEEE 802.11 and its medium access mechanism. Chapter 3 discusses the importance of quality of service (QoS) and the limitations of IEEE 802.11 in providing QoS support. Chapter 4 provides with an introduction of IEEE 802.11e and its QoS enhanced medium access mechanism, EDCA. Chapter 5 presents an overview of GloMoSim and its architecture. Chapter 6 provides with the details of the design and implementation of IEEE 802.11e EDCA in GloMoSim. Finally, Chapter 7 presents evaluation of IEEE 802.11e QoS scheme in comparison to IEEE 802.11 with the help of a wide range of simulation scenarios.

Chapter 2

Introduction to IEEE 802.11

2.1 Introduction

In 1997, IEEE (Institute of Electrical and Electronics Engineers) released the 802.11 Wireless Local Area Network (WLAN) standard [1]. As the name suggests, it belongs to the group of popular IEEE 802.x standards, e.g., IEEE 802.3 Ethernet [4] and IEEE 802.5 Token Ring [5]. IEEE 802.11 defines Media Access Control (MAC) and Physical (PHY) layers specifications for wireless LANs. Three different Physical layer specifications were defined, namely, Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS) and Infrared (IR), with the maximum data transmission rate of up to 2 Mbps. The DSSS and FHSS Physical layers operated in the license free 2.4 GHz ISM (Industrial, Scientific and Medical) band.

With the passage of time, while the original MAC remained intact, the technology continued evolving with the new Physical layer specifications.

In 1999, IEEE introduced two enhanced Physical layer specifications 802.11b [6] and 802.11a [7] with data transmission rates of up to 11 and 54 Mbps, respectively. 802.11b is also based on DSSS and operates in the 2.4 GHz band, and, 802.11a is based on OFDM (Orthogonal Frequency Division Multiplexing) and operates in the 5 GHz band. In 2003, IEEE released 802.11g [8] that extended 802.11b Physical layer to support data transmission rates of up to 54 Mbps in the 2.4 GHz band.

IEEE 802.11 gained immense popularity due to its cost effectiveness and easy deployment. Today IEEE 802.11 hotspots[1] are available at offices, campuses, airports, hotels, public transport stations and residential places, making it one of the most widely deployed wireless network technologies in the world.

IEEE 802.11 defines two different architectures, BSS (Basic Service Set) and IBSS (Independent Basic Service Set). In a Basic Service Set, number of wireless stations, called STAs[2], are associated to an AP (Access Point). All communications take place through the AP. In an Independent Basic Service Set, STAs can communicate directly to each other, providing that they are within each other's transmission range. This form of architecture is facilitated to form a wireless ad-hoc network in absence of any network infrastructure.

Several BSS can be connected together via a Distribution System (DS) to form an extended network, called Extended Service Set (ESS). Figure 2.1 illustrates the archi-

---

[1] The area where IEEE 802.11 access is available is referred to as *hotspot*.

[2] For simplicity, we prefer to use the word *station* instead of *STA* throughout the report.

tecture of IEEE 802.11 BSS.

IEEE 802.11 MAC defines two different access mechanisms, the mandatory Distributed Coordination Function (DCF) which provides distributed channel access based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), and the optional Point Coordination Function (PCF) which provides centrally controlled channel access through polling.

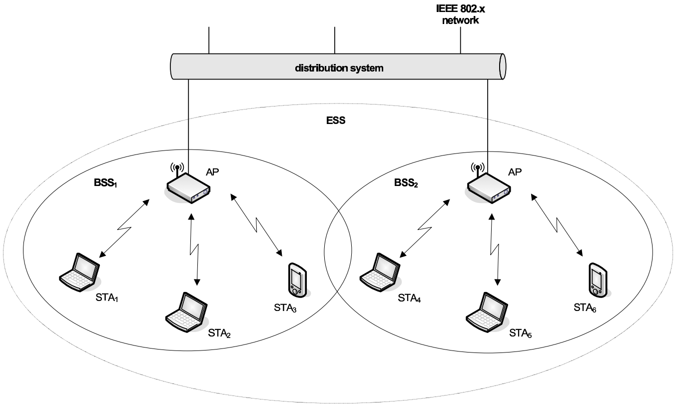


Figure 2.1: Architecture of IEEE 802.11 network.

In DCF, all stations contend for the access to the medium, in distributive manner, based on the CSMA/CA protocol. For this reason the access mechanisms is also referred to as contention-based channel access. In PCF, a Point Coordinator (PC), which is most often collocated in AP, controls the medium access based on the polling scheme, such that the PC polls individual stations to grant access to the medium based on their requirements. As in PCF stations do not contend for the medium and instead the medium access is controlled centrally, the access mechanism is sometimes referred to as contention-free channel access.

Only DCF is explained in the next section as it is the basis for the Enhanced Distributed Channel Access (EDCA) introduced in IEEE 802.11e, which we focus in this work.

2.2 DCF (Distributed Coordination Function)

DCF is the basic access mechanism of 802.11 and is based on Carrier Sense Multiple Access (CSMA). CSMA works as listen-before-talk, i.e., before transmitting a frame, the station senses the medium (carrier sensing). If the medium is found idle at least for DIFS (DCF Inter-Frame Space) time period, the station starts transmission, and other stations wait until medium becomes idle again at least for DIFS time period.

As the destination station successfully receives a frame, it acknowledges by sending back an ACK frame after SIFS (Short Inter-Frame Space) time period. Figure 2.2 illustrates the mechanism.

SIFS is the shortest of the three Inter-Frame Spaces (IFS) defined in IEEE 802.11 to control the access to the medium. IFS relationships can be seen in Figure 2.3 (other parts of figure are explained in next section). Subsequent frame transmissions are separated by these inter-frame spaces depending on the priority of the frame exchange sequence,

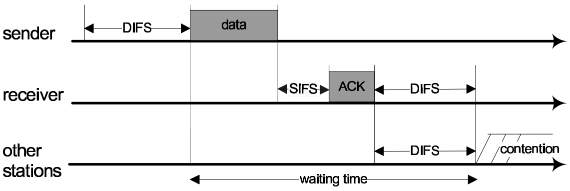


Figure 2.2: DCF basic access mechanism.

i.e., higher the priority of the frame exchange sequence, shorter is the inter-frame space used between the frames. The SIFS between the data and ACK frames, as seen in Figure 2.2, therefore prevents other stations to transmit at the same time the receiver transmits the ACK frame and thereby resulting in transmission failure, because other stations have to wait for the DIFS time prior to start transmission which is longer than SIFS. Thus, in this way, a station transmitting the ACK frame is given priority over the stations trying to transmit data frames.

The second shortest inter-frame space, PIFS (PCF Inter-Frame Space), is used by AP in the PCF, the optional access mechanism of IEEE 802.11, in which PC/AP centrally controls the access to the medium by polling individual stations. In PCF, PC/AP is given priority over ordinary stations such that it has to wait PIFS instead of longer DIFS prior to transmitting a frame.

The values of inter-frame spaces are dependent on the underlying Physical layer (PHY) and are defined in relation to a slot time. Slot time is derived from propagation delay, transmitter delay, and other PHY dependent parameters [9]. PIFS consists of a SIFS plus one slot time and DIFS consists of a SIFS plus two slot times.

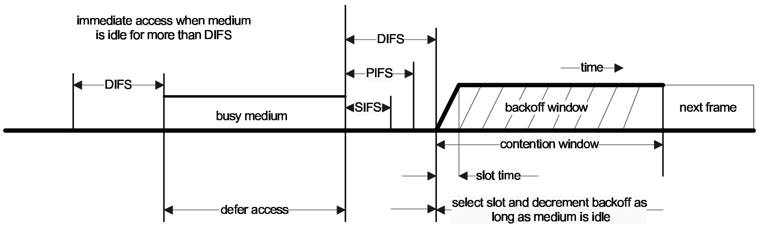


Figure 2.3: Medium access and IFS relationships.

Two types of carrier sensing are used to determine whether the medium is idle or busy. With Physical Carrier Sensing, the wireless channel is sensed itself at the Physical layer. On the other hand, Virtual Carrier Sensing is used at the MAC layer, such that as a station receives a frame that is not directed to it, it examines the *duration* field in the frame header, which specifies the time required to transmit the frame and to receive the ACK frame in response, and then defers the access to the medium for that particular period of time. The process is described in more detail in Section 2.2.3.

2.2.1 Collision Avoidance and Backoff Procedure

The above scenario may lead to collisions if two or more stations sense the medium idle and try to transmit at the same time. In order to avoid such collisions, station has to wait an additional time period prior to transmitting if the medium is sensed busy in the DIFS period, or, if the medium was busy just before the station started waiting the DIFS period. In these situations, the station defers access until the medium becomes idle, and chooses a random backoff value, which specifies the time period, measured in time slots, the station has to wait in addition to the DIFS after the medium becomes idle. This additional random delay in form of backoff helps to avoid collisions, otherwise all stations would try to transmit as soon as medium becomes idle for the DIFS period. This mechanism is called Collision Avoidance (CA), and thus the whole access mechanism is referred to as CSMA/CA.

The reason of using Carrier Sensing with Collision Avoidance instead of Collision Detection (CSMA/CD) used in wired networks, e.g., IEEE 802.3 Ethernet [4], is lack of collision detection capabilities in wireless networks. In wired networks, transceiver has the ability of receiving and transmitting simultaneously and therefore is able to detect collisions, but in wireless networks, the stations very often do not have the ability of simultaneous operation. Even if the station has the ability of receiving while transmitting, the fundamental characteristics of wireless communication do not allow it to detect other signals. Compared to the wired communication where the signal strength in the wire/cable does not drop below an acceptable level[3] and thus makes it possible for the sender to detect the colliding signal, in wireless communication it is not possible because in free space the strength of signal decreases proportionally to the square of distance to the sender. Moreover, various types of interferences and noises, and fading further attenuate the signal strength, thereby making it difficult for the sender to detect other signals in presence of its own signal, because the strength of its own signal is several magnitudes higher than the strength of the signal being detected [9, p. 70].

After choosing the backoff value, as the medium is sensed idle at least for DIFS time period, the station starts decrementing its backoff timer by one for each time slot. If the medium becomes busy during this backoff process, the station backoffs, i.e., it pauses its backoff timer. The backoff timer is then resumed as soon as the medium is sensed idle for the DIFS period again. The station is allowed to transmit as the backoff timer reaches zero.

While a new station has to choose a new backoff value, the station which attempted first continues to count down its paused backoff timer instead of choosing a new one. Thus, in this way a station that attempted first and thus waited longer is given advantage over a station that attempted after it, because it only has to wait for its remaining backoff time.

The random backoff value is uniformly chosen[4] from the interval [0, CW], called the Contention Window. At the first transmission attempt, CW is set to the minimum Contention Window size, CWmin. After each unsuccessful transmission, CW is doubled, actually increased exponentially, using the equation (2 x (CW + 1) - 1), until it reaches the maximum Contention Window size, CWmax. The values of CWmin and CWmax are dependent on the underlying Physical layer (PHY). For the most commonly used Physical layer, DSSS PHY, the values are 31 and 1023 respectively, in which case the Contention Window size increases in the form of 31, 63, 127, 255, 511, 1023, as shown in Figure 2.4. For this reason, the mechanism is also referred to as exponential backoff.

An unsuccessful transmission (in other words, a collision) is determined if the sender station does not receive ACK frame within a specified ACK timeout period. After the

---

[3] The strength of the signal in wire is directly proportional to the length of the wire. Thus if the length of wire stays within standardized limit, more or less same signal strength can be assumed all over the wire [9, p. 70].

[4] A better choice would be to use the word *draw* or *get* instead of *choose* because the backoff value is not explicitly chosen according to desire and instead is drawn randomly.

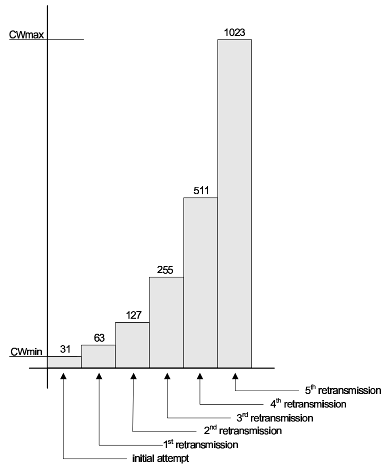


Figure 2.4: Exponential increase of Contention Window.

ACK timeout period, the station assumes that a collision has occurred and enters into the backoff period again after waiting for medium to be idle for DIFS, such that the new backoff value is chosen from the doubled CW. With the doubled CW size, the probability of a bigger random backoff value is higher, which reduces the probability of the stations colliding again.

DCF specifies a retransmit limit (also referred to as retry limit), i.e., the number of times a frame can be retransmitted. If an unsuccessful transmission is determined after reaching the retransmit limit, the frame is dropped.

The CW size is reset to CWmin after each successful transmission. The backoff mechanism is also used after a successful transmission before sending the next frame, i.e., if the sender station has another frame to send just after receiving ACK frame for the previous frame, it waits the medium to be idle for the DIFS time and chooses a new backoff value. This is referred to as *post backoff*, as it is done after the transmission not before. The post backoff ensures that there is at least one backoff interval between two consecutive transmissions. It allows other stations to decrement their backoff timers and thereby to get access to the medium.

The backoff procedure shows that the scenario described earlier, i.e., station transmitting immediately without waiting the backoff time, is rare, and only happens in the situations when, the time the frame arrives at MAC layer, the last post backoff has already been finished, i.e., the transmission queue is empty, and the medium has been idle longer than the DIFS time period. Only then the frame can be transmitted immediately after the DIFS time period. The frames immediately following this frame have to be transmitted after backoff, until the transmission queue becomes empty again.

The Collision Avoidance mechanism does not totally eliminate the risk of collisions. Collisions may still occur if the backoff timers for two or more stations reach zero at the same time, or if two or more stations accidentally get the same backoff values.

The latter also indicates that the probability of collision is inversely proportional to the size of Contention Window, i.e., smaller the Contention Window size, greater the rate of collisions and vice versa. Although a bigger Contention Window size reduces probability of collisions, it results in higher delays and inefficient bandwidth utilization.

An example of DCF operation follows next for further explanation of the backoff procedure and collision. It also highlights the important role of DIFS and CWmin and

CWmax parameters[5] in the access mechanism process.

2.2.2 Example of DCF Operation

Figure 2.5 illustrates an example scenario to explain the operation of DCF with backoff procedure. Here three stations are contending for the medium. The time Station 3 gets a frame to send, as the medium is idle and there is no ongoing backoff process, the station starts its transmission immediately after sensing the medium idle for the DIFS time period. The transmission is indicated by *busy* in the figure, and includes the complete frame exchange sequence of data + SIFS + ACK, as seen in Figure 2.2. Stations 1 and 2 arrive meanwhile and try to sense the medium idle for DIFS time period, but after sensing it busy, they defer the access.

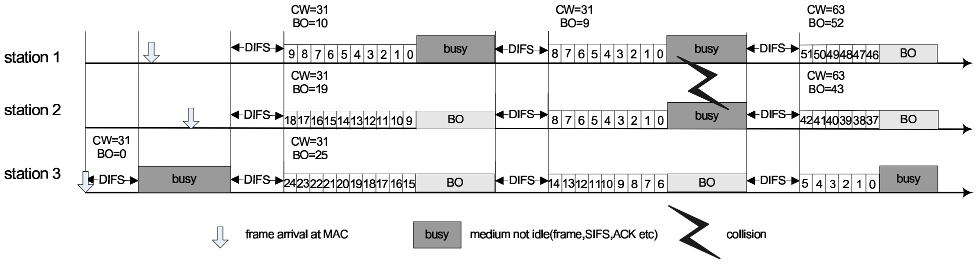


Figure 2.5: DCF access mechanism with backoff procedure.

After the transmission finishes and the medium becomes idle, all three stations wait it idle for the DIFS time period and then choose the random backoff values. Here it is assumed that the backoff values chosen by stations 1, 2 and 3 are 10, 19 and 25, respectively. The backoff performed by the Station 3 immediately after the transmission of first frame is referred to as post backoff, as described earlier. All three stations then invoke the backoff procedure by decrementing their backoff timers with each idle time slot. The station with the smallest backoff value, Station 1, is able to count its timer down to zero first, and thus wins the access to the medium. As Station 1 starts transmission, stations 2 and 3 pause their backoff timers, indicated by BO in the figure. After the medium becomes idle again, all three stations wait for the DIFS and start backoff procedure again. While Station 1 chooses a new backoff value of 10, stations 2 and 3 resume their paused backoff timers. As the remaining backoff time of Station 2 is also 10, both Station 1 and Station 2 start transmitting at the same time after counting their timers down to zero. This leads to a collision. As stations 1 and 2 start transmission, Station 3 senses the medium busy and pauses its backoff timer. Stations 1 and 2, being unaware of the collision (because of the reason described in Section 2.2.1), wait for the ACK frames from the receiver stations. As no ACK is received within the ACK timeout period, both stations assume that a collision has occurred, and thus, after waiting for the DIFS time period, double their Contention Windows and choose new backoff values. As the remaining backoff time of Station 3 is smaller compared to the new backoff values of stations 1 and 2, Station 3 is able to transmit first after waiting for the backoff time, and stations 1 and 2 pause their backoff timers.

---

[5] As these parameters are used to contend for the medium, they are sometimes referred to as *contention parameters.*

2.2.3 RTS/CTS Mechanism

An additional mechanism, RTS/CTS, is defined to solve the hidden terminal problem found in wireless networks that use CSMA. With RTS/CTS, the sender and receiver perform a handshake mechanism by exchanging RTS (Request To Send) and CTS (Clear To Send) control frames. The procedure is shown in Figure 2.6. After waiting the DIFS time, prior to transmit the data frame, the sender sends a RTS frame to the receiver, and the receiver responds with a CTS frame after waiting a SIFS time. The CTS frame indicates that the handshake is successful and ensures that the medium has been reserved for the particular sender and receiver for the transmission.

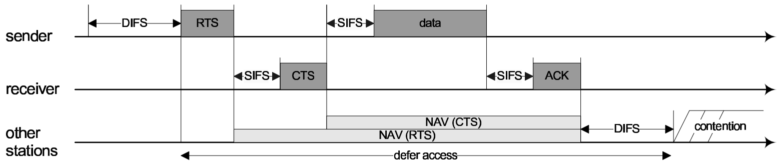


FIGURE 2.6: FRAME EXCHANGE SEQUENCE WITH RTS/CTS MECHANISM.

RTS/CTS uses Virtual Carrier Sensing, such that the RTS and CTS frames include the duration of the complete frame exchange sequence, inclusive of SIFS and ACK. All stations within the receiving range around the sender and receiver set their NAV (Network Allocation Vector) after receiving the RTS and CTS frames, and thus are informed that they have to wait until the current transmission finishes. Here an important thing to note is that the sets of stations receiving RTS and CTS frames can be different. NAV is a timer and is decremented in the similar way the backoff timer is decremented. The station is allowed to transmit after its NAV reaches to zero.

Collisions can only occur at the beginning when the RTS frame is transmitted, as two or more stations may start transmitting, either RTS or data frames, at the same time. Such collisions are determined if the sender does not receive the CTS frame within a specified CTS timeout period. In that case, the sender transmits the RTS frame again. As the size of the RTS frame is significantly smaller compared to that of data frame, the RTS/CTS mechanism provides a mean of fast recovery from collisions, because the sender becomes aware of failure and may retransmit more quickly compared to the case when long data frame is transmitted and failure is determined after ACK timeout.

As seen in Figure 2.6, here again the SIFS intervals between RTS, CTS and data frames prevent other stations to transmit and thereby interrupting the transmission.

RTS/CTS mechanism shall be used for large data frames. Using it for small data frames may result in significant overhead causing inefficient capacity utilization and higher delays.

2.3 Summary

IEEE released the 802.11 Wireless Local Area Network (WLAN) standard in 1997. While the original standard supported maximum data transmission rate of 2 Mbps, in 1999, its enhanced versions 802.11b and 802.11a increased data rates to 11 and 54 Mbps, respectively. IEEE 802.11 MAC defines two different access mechanisms. The mandatory Distributed Coordination Function (DCF) and an optional Point Coordination Function (PCF). Most of the 802.11 installations today deploy DCF, and PCF is hardly implemented due to its complex design.

DCF is based on Carrier Sense Multiple Access (CSMA). Before transmitting a frame, the station senses the medium, and if the medium is found idle at least for DIFS (DCF Inter-Frame Space) time period, the station starts transmission. Otherwise, if the medium is found busy during the DIFS period, the station defers access and chooses a random backoff value that specifies the additional time it has to wait after the medium becomes idle again. As the medium becomes idle for the DIFS time period again, the station starts decrementing its backoff time. If medium becomes busy during this backoff process, the station pauses the backoff timer, and resumes it as the medium becomes idle for the DIFS period again. The station is allowed to transmit as the backoff timer reaches to zero. The additional random backoff time helps avoiding collision, which is defined as the situation when two or more stations transmit at the same time and result in unsuccessful transmissions. As the destination station receives the frame, it acknowledges by sending back an ACK frame after SIFS (Short Inter-Frame Space) time period. SIFS, like DIFS, is one of the three inter-frame spaces (IFS) defined to control the medium access.

The random backoff value is uniformly chosen in the range (0, CW), where CW is called the Contention Window. CW is initialized to the minimum size CWmin and doubled after each unsuccessful transmission, until it reaches the maximum Contention Window size, CWmax. CW is reset to CWmin after every successful transmission.

An additional mechanism RTS/CTS is defined that allows the sender and receiver to handshake by exchanging RTS (Request To Send) and CTS (Clear To Send) frames prior to transmitting the data frame. In this way, the medium is reserved such that all stations which received RTS or CTS frames defer the access until the transmission is finished.

Chapter 3

Quality of Service and Limitations of IEEE 802.11

3.1 Introduction

The term Quality of Service (QoS) refers to set of qualitative and quantitative characteristics, such as throughput, packet loss, delay, jitter and bandwidth utilization, which describes the quality of data traffic over a network.

QoS can be seen from the network or application point of view. The application has certain QoS requirements, and the network capable of fulfilling these requirements is said to support QoS.

QoS requirements vary from application to application and can be classified in three dimensions: bandwidth, delay, and data loss [10].

Bandwidth: Bandwidth is one of the most important parameters and refers to the amount of data that can be delivered during a given period of time. Greater the bandwidth an application receives, larger the amount of data it can transfer and vice versa. Several other terms used interchangeably for bandwidth are data rate, transmission rate, bit rate, and capacity. For an application which requires to transfer data on a constant rate, any decrease in bandwidth results in undesired delays and data loss. Such an application is referred to as bandwidth-sensitive application. Multimedia applications such as streaming media, Internet telephony i.e., VoIP (Voice over IP), and videoconferencing, typically fall in this category. These applications require constant bandwidth and may seriously suffer because of variations in bandwidth [10]. Due to bandwidth-sensitive characteristic of these applications, they are sometimes referred to as inelastic applications [11]. Elastic applications, on the other hand, do not put tight constraints on bandwidth, i.e., they can tolerate variations in bandwidth. Elastic applications are generally data oriented. Some examples are email, file sharing, web, and instant messaging applications. Bandwidth is often measured with respect to throughput, which is the data transfer rate, measured as the number of bits transmitted per second.

Delay: Interactive real-time applications such as Internet telephony i.e., VoIP (Voice over IP), videoconferencing, VR (Virtual Reality) environments, and multiplayer network games are very sensitive to delays and enforce tight constraint on in time data transfer [10]. High delays severely destroy the performance of these applications.

An important measurement is the end-to-end delay, which is the total delay from the time the application at the sender side generates a packet to the time the application at the receiver side receives it. End-to-end delay includes all types of delays that may occur during the whole transmission, particularly transmission delay, propagation delay and any queueing delays [10, p. 38-39]. The *transmission delay* is the time required to transmit data into the wireless channel, and directly depends on the available bandwidth. For example, a packet with 1024 bytes of data takes approximately 8 ms to be transmitted with the available bandwidth of 2 Mbps. *Propagation delay* depends on the velocity of propagation of the signal across the transmission media [12], which, in the case of free space is equal to the speed of light, i.e., 3 x 10^8 m/s. *Queueing delay* refers to the time a packet has to wait in the queue before it is transmitted. The length of the queueing delay of a packet depends on the number of earlier arriving packets already waiting in the queue [10].

End-to-end delay, constraints are tightly applied in voice applications, and packet with delays above a few tens of milliseconds are considered useless. An example is Internet telephony, where end-to-end delays of more than 150 milliseconds are often unacceptable [10].

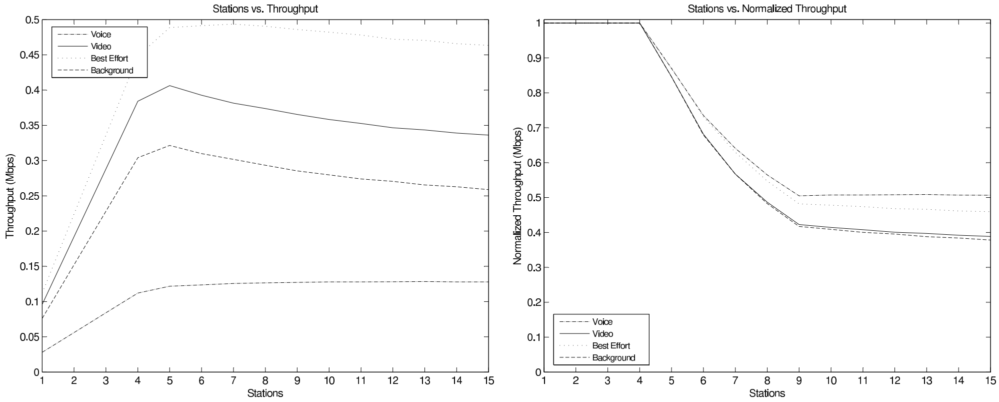
Another form of delay, often referred to as jitter, is defined as the variation in delays, and of high importance particularly in constant bit rate multimedia applications. Since the decoder application at the receiver application must decode the received data according to the bit rate it was encoded with at the sender application, high variations in delay (jitter) results in problems in decoding the data. Most of the real-time multimedia applications are very sensitive to jitter and buffering techniques are applied to reduce its effects, such that packets are first buffered in a buffer/queue at the receiver side, and then the decoding process is started. The number of packets buffered contained in the buffer before output starts is determined by the maximum expected jitter and the actual bit rate of data [12].

Data Loss: Most of the multimedia applications are classified as bandwidth and delay-sensitive but are generally loss-tolerant, i.e., they require strict bandwidth and delay guarantees but can tolerate certain amount of data/packet losses. Data/packet loss in these applications results in slightly reduced quality of output (e.g., jerks in video or voice output), but can be neglected to some extent. The effects of such losses on the application quality, and the amount of tolerable losses strongly depends on the application and the coding technology used [10]. On the other hand, data oriented applications such as email, file transfer, and web documents are generally loss-sensitive. They can tolerate occasional delays and low bandwidth but require reliable data transfer.

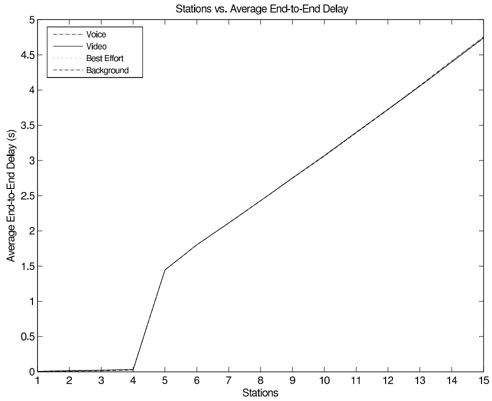
3.2 QoS Limitation of IEEE 802.11 DCF

IEEE 802.11 is based on best-effort service model, i.e., the protocol makes its best-effort to deliver frames from the sender to receiver as quickly as possible, but it does not make any guarantees about delay, bandwidth and packet loss. Basically it serves all types of applications in the same way, irrespective of what QoS requirements an application may have. There is no support for service differentiation. Bandwidth, delay and loss-sensitive applications are served in the same way, on the best-effort basis. As a result, all types of data traffic suffer from same amounts of delays, losses and variations in bandwidth as the network becomes congested.

Figure 3.1 shows the throughput and end-to-end delay results for four different types of traffic streams generated by a station. The four traffic types are modeled as video, voice, best effort and background, based on their characteristics, i.e., packet size, interval, and bit rate, as shown in Table 7.3. As seen in Figure 3.1a, regardless of the QoS requirements of different traffic streams, all four traffic streams are served equally and thus experience same amounts of degradations in bandwidth. Figure 3.1b presents the same throughput results as in Figure 3.1a but in normalized form, which is the ratio of the data actually transmitted to the total offered data. It is evident from the results that all four types of data traffic receive almost same ratios of the total bandwidth, and suffer from similar degradation as the network becomes saturated. Figure 3.1c shows the end-to-end delay results. As it is seen, each of the four streams faces same amount of delay regardless of the delay requirements of the traffic/application.



(a) Average throughput (absolute). (b) Average throughput (normalized).



(c) Average end-to-end delay.

Figure 3.1: Throughput and delay results for four different applications in 802.11.

The point is that all stations/traffic compete for the medium with the same priority. There is no differentiation mechanism to guarantee bandwidth, delay and jitter for high priority multimedia traffic. Some kind of service differentiation must be employed to let certain types of traffic get better served. This lack of guarantees is a big hurdle in providing QoS services for multimedia applications in 802.11 networks.

The Internet Engineering Task Force (IETF) has defined two different frameworks, Integrated Services (IntServ) [13] and Differentiated Services (DiffServ) [14], to support QoS for the traffic over Internet. Compared with wired networks, to provide QoS in wireless networks is more challenging since wireless networks have limited bandwidth, and suffer from higher bit error rates because of many factors such as propagation loss, noise, interference, multipath, shadowing, fading, weather and so on.

Realizing the importance of QoS, the modern packet-oriented telecommunication networks are also employing QoS support mechanisms. GPRS (General Packet Radio Service) enables users to specify a *QoS-profile*, which determines the priority of service the user is acquiring. Three priorities are defined, high, normal and low, to support QoS for different types of applications. Additionally, three classes are defined, know as *reliability class*, *delay class*, and *throughput class*, to let the users specify the packet loss, delay and throughput constraints [9]. Similarly, UMTS (Universal Mobile Telecommunication System)[15] also defines four different service classes based on the delay requirements of the traffic. The classes are named *conversational*, *streaming*, *interactive*, and *background*. Here the *conversational* class has the most strict delay requirements since it is intended for real time voice conversation applications. The *streaming* class is defined for audio/video streaming and have relatively lower delay constraints. The other two classes are defined for low priority data oriented traffic [16].

Thus, with the evolving need of QoS enabled networks, a large amount of research work has been carried to enhance the QoS capabilities of 802.11 since last few years. Some of the comprehensive studies on quality of service issues and enhancement in 802.11 networks are found in [17, 18, 19, 20, 21, 22]. A different approach to provide QoS support, discussed in [23], is based on utility functions which express user or application utility as a function of allocated resources, such as bandwidth, and the techniques to optimize these functions.

For last four, five year, parallel to the activities in the research community, the IEEE 802.11 Working Group has also been working on a new version of 802.11, called 802.11e [2], to introduce QoS support in 802.11 networks. 802.11e has been available in draft forms, and recently the final version of the draft, draft 13th, has been released. The next chapter provides a comprehensive description of 802.11e and its QoS enhanced MAC mechanism. Some of the good performance evaluation works based on different draft versions of the 802.11e standard we consulted are found in [24, 20, 25, 26, 27, 28, 29, 30, 31, 32, 19, 33, 34, 35, 36, 37, 38, 39].

3.3 Summary

The term Quality of Service (QoS) refers to set of characteristics, such as throughput, packet loss, and delay, which describes the quality of data traffic over a network. Applications have certain QoS requirements and the network capable of fulfilling these requirements is said to support QoS. QoS requirements vary from application to application and can be classified in three dimensions. Bandwidth: Greater the bandwidth an application receives, larger the amount of data it can transfer and vice versa. Bandwidth is often measured in terms of throughput, which is the data transfer rate measured as the number of bits transmitted per second. For applications which require data transfer with constant rate, any variations in bandwidth results in undesired delays and data loss. Such applications are referred to as bandwidth-sensitive applications. Delay: Interactive and real-time multimedia applications such as VoIP, videoconferencing and network games are very sensitive to delays. A common measurement, end-to-end delay, is the total delay from the time the sender application generates a packet to the time it is received by the receiver application. End-to-end delay constraints are tightly applied in voice applications and packet with delays above a few tens of milliseconds are considered useless. Data Loss: In contrast to multimedia applications which are bandwidth and delay sensitive but are generally loss-tolerant, the data oriented applications such as email, file transfer, and web documents are loss-sensitive. Data/packet losses in these applications result in partial or complete corruption of data.

802.11 MAC mechanism is unable to provide QoS support for different types of applications. Basically it serves all types of applications in the same way irrespective of what QoS requirements an application may have. There is no support for service differentiation. This proved to be a big hurdle for adaptation of modern multimedia applications which are very sensitive to variations in bandwidth and delay.

Realizing the importance of QoS, different QoS enhanced mechanisms have been introduced in wired networks (Internet) and in modern telecommunication networks such as GPRS and UMTS. Basically the data traffic is assigned with different priorities based on its QoS requirements and then served accordingly. Different priorities and classes have been introduced for bandwidth, delay and loss sensitive applications/traffic. IEEE 802.11 Working Group has also been working on a new version of 802.11, called 802.11e, to introduce QoS support in 802.11 networks and recently the final version of the draft standard has been released.

Chapter 4

Introduction to IEEE 802.11e

4.1 Introduction

IEEE is currently working on a new standard, called IEEE 802.11e [2], which is an enhanced version of the legacy[1] 802.11 MAC in order to support quality of service (QoS). IEEE 802.11e is in standardization process and the final draft has been released.

IEEE 802.11e supports quality of service by introducing priority mechanism. All types of data traffic are not treated equally as it is done in the original standard, instead, 802.11e supports service differentiation by assigning data traffic with different priorities based on their QoS requirements. Furthermore, four different Access Categories (AC) have been defined each for data traffic of a different priority. Access to the medium is then granted based on the priorities of data traffic, such that each frame with a particular priority is mapped to an Access Category, and service differentiation is realized by using a different set of contention parameters to contend for the medium, for each AC.

In IEEE 802.11e, the AP and STA that provides QoS services are referred to as QAP (QoS Access Point) and QSTA[2] (QoS Station) respectively, and the BSS they are operating in is called QBSS (QoS Basic Service Set).

IEEE 802.11e introduces a new coordination function, called Hybrid Coordination Function (HCF), to provide QoS support. Subsequent sections describe HCF together with the detailed description of its service differentiation mechanism.

4.2 HCF (Hybrid Coordination Function)

IEEE 802.11e defines a new coordination function called Hybrid Coordination Function (HCF). HCF is a centralized coordination function that combines the aspects of DCF and PCF with enhanced QoS mechanisms to provide service differentiation. HCF provides both distributed and centrally controlled channel access mechanisms similar to DCF and PCF in the original standard. The distributed, contention-based channel access mechanism of HCF is called Enhanced Distributed Channel Access (EDCA), and the centrally controlled, contention-free channel access mechanism is called HCF Controlled Channel Access (HCCA).

IEEE 802.11e introduces Transmission Opportunity (TXOP), defined as the time period during which a QSTA has the right to transmit. In other words, in 802.11e

---

[1] The word *legacy* is often used to refer to the original 802.11 standard.

[2] For simplicity, most of the times in this report we use the word *station* for both *STA* and *QSTA*.

when a station gets access to the medium, it is said to be granted the TXOP. TXOP is characterized by a starting time and a maximum duration, called TXOP Limit. As a QSTA gets the TXOP, it can then start transmitting frames such that the transmission duration does not exceed the TXOP limit. TXOP Limit is specified by the QAP.

The next section describes EDCA, the distributed access mechanism of HCF. The detailed functionality of the centrally controlled access mechanism HCCA is beyond the scope of this report as we focus on the EDCA.

4.3 EDCA (Enhanced Distributed Channel Access)

The EDCA provides differentiated, distributed access to the medium using different priorities for different types of data traffic. The detailed description of the components and operation of EDCA is presented next.

4.3.1 Access Categories (ACs)

EDCA defines four Access Categories (ACs) for different types of data traffic, and service differentiation is introduced such that for each AC, a different set of parameters is used to contend for the medium. These parameters are referred to as EDCA parameters and are described in the next subsection.

Frames from different types of data traffic are mapped into different ACs depending on the QoS requirements of the traffic/application the frames belong to. The four Access Categories are named AC_BK, AC_BE, AC_VI and AC_VO, for Background, Best Effort, Video and Voice data traffic, respectively, where AC_BK has the lowest and AC_VO has the highest priority.

Each frame from the higher layer arrives at the MAC layer along with a priority value. This priority value is referred to as User Priority (UP) and assigned according to the type of application/traffic the frame belongs to. There are eight different priority values ranging from 0 to 7.

Priority	**User Priority (UP)**	**Access Category (AC)**	**Designation**
Lowest	1	AC_BK	Background
.	2	AC_BK	Background
.	0	AC_BE	Best Effort
.	3	AC_BE	Best Effort
.	4	AC_VI	Video
.	5	AC_VI	Video
.	6	AC_VO	Voice
Highest	7	AC_VO	Voice

TABLE 4.1: USER PRIORITY (UP) TO ACCESS CATEGORY (AC) MAPPINGS.

How to assign such a priority to each frame is a higher layer implementation issue. Interestingly, the draft standard does not specify anything how such a priority is assigned at the higher layers. Generally, it can be assigned by application generating the traffic, or by the user using the application. The latter solution indicates that every application has to be updated in order to be compatible with 802.11e. Another possibility would be to adaptively assign the priority at the Application layer, based on the traffic characteristics e.g., data rate, packet[3] interval, packet size etc. How this priority will travel through different layers down to the MAC layer further indicates that modifications at the higher layers would be inevitable.

At the MAC layer, a frame with a particular UP is further mapped to an AC. ACs are derived from the UPs as illustrated in Table 4.1.

4.3.2 EDCAF (Enhanced Distributed Channel Access Function)

Every station maintains four transmit queues one for each AC, and four independent EDCAFs (Enhanced Distributed Channel Access Function), one for each queue, as illustrated in Figure 4.1. EDCAF is an enhanced version of DCF, and contends for the medium on the same principles of CSMA/CA and backoff, but based on the parameters specific to the AC it is contending for[4]. Next section discusses these parameters, referred to as EDCA parameters .

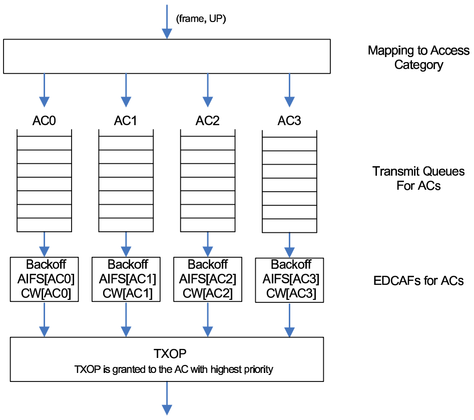


Figure 4.1: Four ACs, each with its own queue, AIFS, CW and backoff timer.

4.3.2.1 EDCA Parameters

An EDCAF contends for medium based on the following parameters associated to an AC:

- AIFS - The time period the medium is sensed idle before the transmission or backoff is started.
- CWmin, CWmax - Size of Contention Window used for backoff.
- TXOP Limit - The maximum duration of the transmission after the medium is acquired.

---

[3] A packet at the MAC layer is generally referred to as *frame*. Therefore sometimes the words *packet* and *frame* are used interchangeably.

[4] AC and EDCAF are sometimes referred interchangeably because of very close relationship between them.

The values of EDCA parameters[5] are different for different ACs. The higher priority ACs wait a small AIFS time period while the lower priority ACs have to wait a longer AIFS time before they can access the medium. The size of Contention Window varies such that the higher priority ACs choose backoff values from a smaller Contention Window compared to the lower priority ACs. TXOP Limit is also set in a way that the higher priority ACs get the access to the medium for longer durations. Basically, the higher the priority of an AC, the smaller the AIFS, CWmin and CWmax, and larger the TXOP Limit. As the values of EDCA parameters are AC specific, they are sometimes referred to as AIFS[AC], CWmin[AC], CWmax[AC] and TXOP Limit[AC].

Thus, basically the main difference between DCF and EDCAF is that EDCAF uses AC specific parameters AIFS[AC], CWmin[AC] and CWmax[AC] instead of using fixed values DIFS, CWmin, and CWmax.

EDCA parameters are periodically advertised by the QAP. QAP can adapt these parameters dynamically depending on the network conditions. The draft standard specifies default values of EDCAF parameters if not advertised by the QAP.

A brief description of each of the EDCA parameters and its role in providing service differentiation is presented in next section.

AC	**CWmin**	**CWmax**	**AIFSN**	**TXOP Limit**	
				FHSS	**DSSS**
AC_BK	CWmin	CWmax	7	0	0
AC_BE	CWmin	CWmax	3	0	0
AC_VI	(CWmin+1)/2-1	CWmin	2	6.016ms	3.008ms
AC_VO	(CWmin+1)/4-1	(CW+1)/2-1	2	3.264ms	1.504ms

TABLE 4.2: DEFAULT EDCA PARAMETER VALUES.

AIFS (Arbitration Inter-Frame Space) - The minimum time period for which the medium must be sensed idle before an EDCAF/station may start transmission or backoff is not the fixed value DIFS, as it is in DCF, but is a variable value, AIFS, that depends on the AC for which the EDCAF is contending for. AIFS is derived from the following equation:

$$\text{AIFS} = \text{AIFSN x aSlotTime} + \text{aSIFSTime},$$

where aSlotTime is the slot time, aSIFSTime is the SIFS time period and AIFSN (Arbitration Inter-Frame Space Number) is used to determine the length of the AIFS. AIFSN specifies the number of time slots in addition to the SIFS time period the AIFS consists of. Different AIFSN values are used for different ACs such that the high priority ACs use smaller values compared to the low priority ACs. The minimum possible value of AIFSN is 2. As a DIFS is equal to 2 x aSlotTime + aSIFSTime, it shows that the minimum length of AIFS is same as of DIFS. For QAP operating in HCCA, the minimum possible value of AIFSN is 1, which makes it equal to PIFS as PIFS is 1 x aSlotTime + aSIFSTime.

The default AIFSN values for all four ACs can be seen in Table 4.2. Figure 4.2 further explains how priority is given to different ACs based on the AIFS time periods.

---

[5] The terms *EDCA parameters*, *AC parameters*, *QoS parameters*, and *contention parameters* are used interchangeably to refer to these parameters throughout the report.

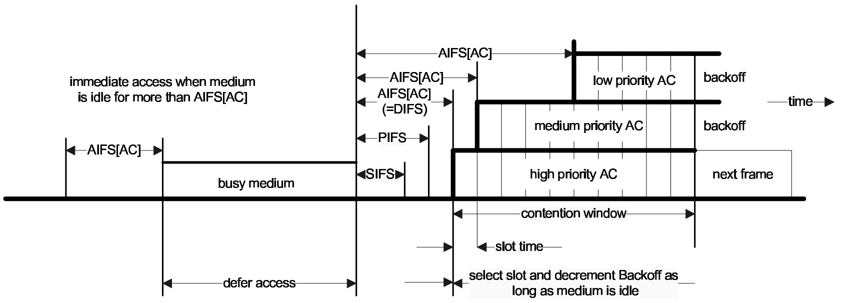


Figure 4.2: Prioritization based on AIFS.

The smaller AIFSN value for a higher priority AC explains that the corresponding EDCAF has to wait shorter time period before it can start transmission or counting down its backoff timer compared to the EDCAF for a low priority AC. In this way, the higher priority ACs are guaranteed greater share of the bandwidth. Moreover, smaller AIFS lengths ensure that the higher priority ACs will not suffer from long delays, which are vary critical for the delay-sensitive applications/traffics, as discussed in Section 3.1. The lower priority ACs may suffer from longer delays because of the larger AIFS durations they have to wait, but since these ACs are designed for delay-tolerant applications/traffics, certain amount of delays do not degrade their performance beyond an acceptable limit.

CWmin and CWmax - The minimum and maximum Contention Window size limits are not fixed as it is in DCF, but are variable depending on the AC. The higher priority ACs have smaller CWmin and CWmax values compared to lower priority ACs. The default values of CWmin and CWmax parameters for each of the four ACs are presented in Table 4.2. The Contention Window parameters specific for the physical layers are presented in Table 4.3.

	FHSS	**DSSS**
CWmin	15	31
CWmax	1023	1023

Table 4.3: Contention Window parameters for different Physical layers.

A smaller Contention Window for an AC will cause the corresponding EDCAF to choose smaller random backoff values, and thereby waiting shorter time period in addition to AIFS as the medium becomes idle. It gives such an AC priority over the AC with a larger Contention Window, which results in larger backoff values and thereby longer delays.

As seen in Table 4.2, for the commonly used Physical layer DSSS, the CWmin values for lower priority ACs, AC_BE and AC_BK, are same as it is for the legacy 802.11 DCF, but these values for higher priority ACs, AC_VO and AC_VI, are as small as one half or quarter of those of the lower priority ACs. This results in smaller backoff values for the high priority ACs and thereby shorter medium access delays. The negative aspect of small Contention Window sizes for higher priority ACs is, that they suffer from higher number of collisions. The reason is, as described in Section 2.2.1, that the probability of choosing the same backoff values or counting the backoff timers to zero at the same time increases with the decreasing size of Contention Windows.

CWmax values for high priority ACs are also set such that they are equal or less than the CWmin values for the lower priority ACs, i.e., Contention Windows are non-overlapping. This shows that after doubling the Contention Window size in case of an unsuccessful transmission, i.e., collision, its size still remains smaller than the CWmin size of lower priority ACs. Furthermore, it also indicates that while a low priority AC has to double its CW size after each unsuccessful transmission, until it reaches the CWmax, and with higher probability, has to choose a bigger backoff value for each retransmission, the Contention Window size of a high priority AC becomes constant after fewer retransmissions, allowing it to consistently choose smaller backoff values and thereby winning access to the medium. In this way, high priority ACs are given consistent and greater share of the bandwidth in the situations when the network has become congested. On the other hand, this may severely degrade the performance of the low priority ACs since they might not be able to decrement their backoff timers because of the smaller post backoff durations of the higher priority ACs. The situation is further explained with an example later in this section.

As it can be seen in Table 4.2, the default values for CWmin and CWmax for both AC_BE and AC_BK are same, but priority is given to AC_BE over AC_BK by assigning it a much smaller AIFSN, i.e., 3 compared to 7, indicating that AC_BK has to wait four additional slots prior to starting transmission or backoff procedure. It also shows that AC_BK suffers from much high delays compared to the other ACs.

TXOP (Transmission Opportunity) - As described above, TXOP is the time duration an EDCAF may transmit after winning access to the medium. TXOP is characterized by a maximum duration, called TXOP Limit. As an EDCAF gets the TXOP, it can then start transmitting frames such that the transmission duration does not exceed the TXOP Limit. The transmission duration covers the whole frame exchange sequence, including the intermediate SIFS periods and ACKs, and the RTS and CTS frames if RTS/CTS mechanism is used.

Table 4.2 shows the default TXOP limits for different ACs. A non zero value of TXOP Limit indicates that the EDCAF may transmit multiple frames in a TXOP, provided that the transmission duration does not exceed the TXOP Limit and the frames belong to the same AC. This is then referred to as Contention Free Bursting (CFB). The consecutive frame transmissions in a TXOP are then separated by SIFS time periods instead of AIFS plus the post backoff periods, as illustrated in Figure 4.3. It is important to note that the multiple frame transmission is granted to EDCAF (or AC) and not to the station, i.e., it is only allowed for the transmission of frames of the same AC as of the frame for which the TXOP was obtained.

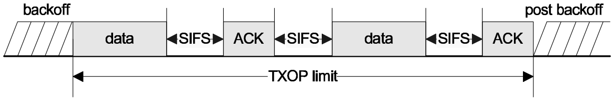


FIGURE 4.3: CONTENTION FREE BURSTING (CFB).

If RTS/CTS mechanism is used with CFB, then the RTS CTS frames handshake is done only once, before the first frame, instead of for every frame in the CFB.

The TXOP Limit of zero indicates that CFB is disabled, and thus only one frame, in addition to RTS/CTS if enabled, can be transmitted in a TXOP. In that case, if there is a risk that the transmission duration of the first frame may exceed the TXOP limit, then the frame should be fragmented.

As it can be seen in Table 4.2, the default values of TXOP limits for the low priority ACs, AC_BK and AC_BE, are zero, indicating that CFB is disabled for these ACs. For high priority ACs, CFB allows to seize the medium for certain amount of time periods, which results in significantly reduced delay. However, too large TXOP limits for high priority ACs may result in higher delays for the low priority ACs.

Thus, service differentiation is introduced through TXOP limits by allowing higher priority ACs to gain continuous access to the medium for longer time periods compared to the lower priority ACs.

In the case when CFB is enabled, the virtual carrier sensing is applied such that the *Duration* field in frame header is set to the remaining duration of the whole TXOP and thus all stations receiving the frame set their NAVs for the duration of whole TXOP instead of that of one frame, i.e., first frame in TXOP, plus the intermediate SIFS times and ACK.

4.3.3 Example of EDCA Operation

Besides the different AIFS, CWmin, CWmax and TXOP limit values for different ACs, the rest of medium access mechanism is same as in DCF, i.e., as the medium becomes idle at least for AIFS time period, the EDCAF chooses a random backoff value from its Contention Window and starts decreasing its backoff timer. The EDCAF can start transmission as its backoff timer reaches to zero.

Figure 4.4 shows an example of EDCA operation to further explain how individual EDCAFs in a station contend for the medium, assuming that all four EDCAFs have frames to transmit.

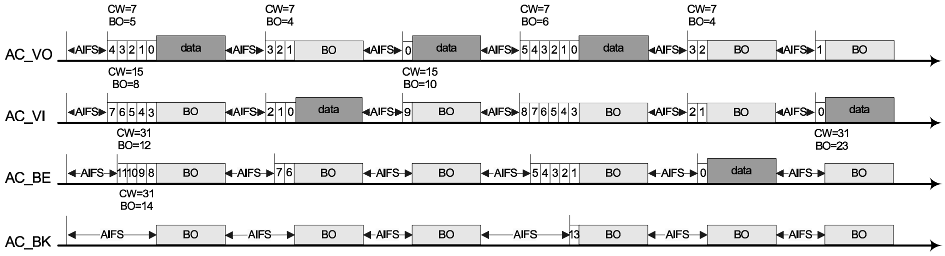


Figure 4.4: EDCA access mechanism.

Figure 4.4 shows all four EDCAFs inside a station contending for the medium. The current Contention Window size and backoff value for each of the AC/EDCAF is also shown. As the EDCAFs for higher priority ACs, AC_VO and AC_VI, have to wait smaller AIFS time periods, they starts counting their backoff timers down prior to EDCAFs for lower priority ACs, AC_BK and AC_BE. The figure assumes the default EDCA parameter values. As the default AIFSN values for AC_VO and AC_VI are same, i.e., 2, the corresponding EDCAFs start to count their backoff timers down at the same time. EDCAFs for AC_BK and AC_BE have to wait some additional slots because of their longer AIFS time periods. As a high priority AC has smaller minimum and maximum Contention Window limits compared to a lower priority AC, EDCAF for a high priority AC most of the time gets smaller backoff values and thus has to wait less time, as shown in the example.

As an EDCAF gets access to the medium, others pause their backoff timers, and continue to count down as soon as the medium becomes idle again for the AIFS time period. Thus, at a certain time, a lower priority EDCAF will have smaller backoff value because while the higher priority EDCAF has to choose a new backoff value for every next frame, the lower priority EDCAF continues to count down its paused backoff timer. This avoids the starvation of the lower priority ACs in a similar way it is avoided for different stations in the original standard, as described in Section 2.2.1.

The example clearly shows the role of EDCA parameters in achieving the service differentiation, i.e., a higher priority AC gets the larger share of the bandwidth by transmitting more frames compared to a lower priority AC. It also shows that the EDCAF for lowest priority AC, AC_BK, is starved. The reason is that since it has to sense the medium to be idle for a much longer AIFS time period, most of the time it is unable to decrement its backoff timer because another EDCAF starts transmitting and hence the medium becomes busy before its AIFS is finished. The other big reason is the smaller Contention Window sizes for high priority ACs, which with high likelihood results in shorter post backoff durations. As seen in Figure 4.4, the EDCAF for AC_BK is able to decrement its backoff timer either if there are no pending frames for high priority ACs, or, if the EDCAFs for AC_VO and AC_VI get post backoff values greater than 5 and the EDCAF for AC_BE gets post backoff value greater than 4.

***Internal Collisions* -** As shown in Figure 4.4, as the four EDCAFs at the AC transmit queues behave like virtual stations inside the real station such that each EDCAF contends for the medium independently of other EDCAFs, there exist two levels of contention, internal contention among different EDCAFs/ACs inside the same station, and external contention among different stations. This may result into a situation where more than one EDCAF in the same station count their backoff timers to zero and try to transmit at the same time. This leads to a situation referred to as *internal collision* or *virtual collision*. In such situation, the access to the medium is granted to the EDCAF for the highest priority AC among the colliding EDCAFs, and the lower priority colliding EDCAF doubles its Contention Window and backoffs, just as if an external collision occurred. Figure 4.5 shows an example of internal collision.

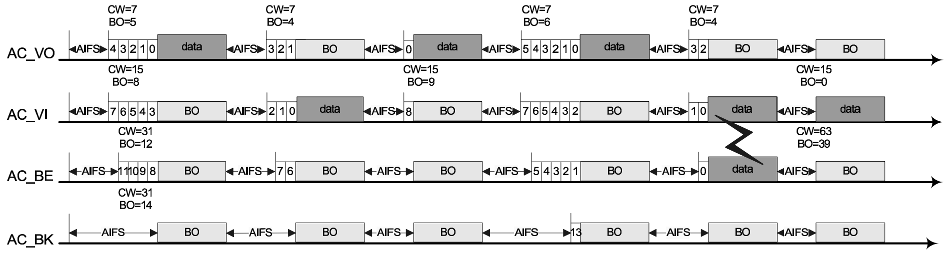


FIGURE 4.5: EDCA ACCESS MECHANISM AND INTERNAL COLLISION.

As described above, after the internal collision, out of the colliding EDCAFs the EDCAF for lower priority AC, AC_BE, doubles its Contention Window and chooses a new backoff value, and the EDCAF for higher priority AC, AC_VI, starts the transmission without any backoff. This explains that the traffic of higher priority AC does not suffer from additional delays after the occurrence of internal collisions, although it may starve the lower priority AC even more, i.e., it will take long time for EDCAF for AC_BE to count its new backoff of 39 down to zero. This condition may be far worse if EDCAF for AC_BK collides; it will hardly be able to decrement its backoff timer after higher priority EDCAFs will have transmitted dozens of frames.

An external collision occurs if backoff timers of the EDCAFs at two or more stations reach zero at the same time, or, if the EDCAFs at two or more stations accidentally get the same backoff values and win access to the medium. Similar to the original standard, after the external collision the colliding EDCAFs increase (double) their Contention Windows and choose new backoff values, and the rest of the EDCAFs retain their paused backoff timers. Figure 4.6 shows an example of external collision where EDCAFs for AC_VO and AC_VI in two different stations count their backoff timers down to zero and try to transmit at the same time. After determining the collision, both colliding EDCAFs double their Contention Windows and start decrementing the newly chosen backoff values while other EDCAFs in both stations continue to decrement their paused backoff timers.

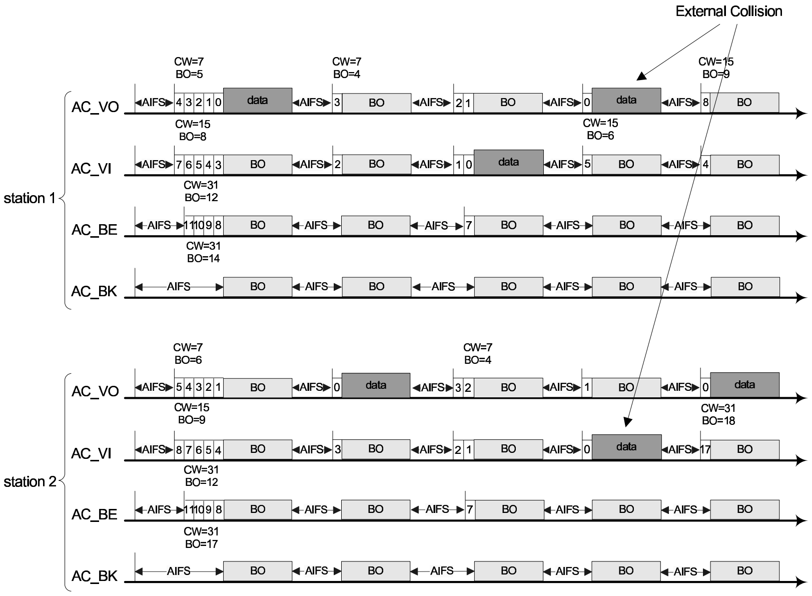


Figure 4.6: EDCA access mechanism and external collision.

4.4 Architecture and Important Frame Formats

Together with HCF and its two access mechanisms EDCA and HCCA, IEEE 802.11e also includes the two coordination functions from the original 802.11, DCF and PCF, in order to provide backward compatibility. Figure 4.7 illustrates the architecture of 802.11e MAC.

For backward compatibility, a QSTA can also operate in a non-QoS BSS (nQBSS) by associating itself to a non-QoS AP (nQAP), in case a QAP is not available. On the other hand, a non-QoS STA (nQSTA) may also associate with a QAP in a QBSS, such that it operates just like an ordinary STA in 802.11 and the transmissions from QAP to nQSTA do not use frame formats specific for QoS services.

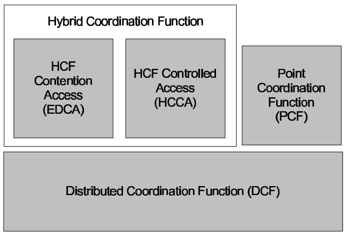


FIGURE 4.7: IEEE 802.11E MAC ARCHITECTURE.

The centrally controlled, contention-free channel access mechanism of HCF, i.e., HCCA, uses a centralized coordinator called HC (Hybrid Controller), which is collocated in QAP. HC operates concurrently with the EDCA just like in the legacy 802.11, i.e., a Contention Free Period (CFP) is followed by a Contention Period (CP), such that the EDCA operates in CP while HC operates both in CP and CFP. This is in contrast with legacy 802.11 where PC can only operate in CFP. It indicates that HC is capable of polling QSTAs both in CP and CFP, and explains why it is referred to as Hybrid Controller.

The one bit *QoS subfield* in *Frame Control* field of MAC header indicates whether the station is acting as a QSTA or an nQSTA. The field is set to 1 if the station is QSTAs, and 0 otherwise. Figure 4.8 shows the QoS subfield in the MAC header.

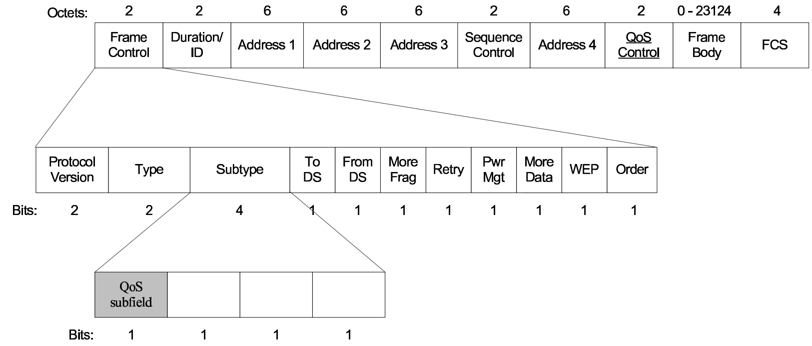


FIGURE 4.8: MAC DATA FRAME HEADER AND QOS SUBFIELD.

At MAC layer, each frame is assigned a priority in the form of a traffic identifier (TID). *TID* field in the newly added *QoS Control* field in the MAC header contains this TID value. UP of the frame is then determined based on this TID value, such that when QAP receives a frame from a QSTA, it gets the UP value from *TID* field, ranging from 0 to 7. Figure 4.9 illustrates the *QoS Control* and *TID* fields in the MAC header.

The priority value in *TID* field is supported only if the station has its *QoS subfield* in the *Frame Control* field set to 1, i.e., the station is associated with a QAP and thus working as a QSTA. If no QAP is available and a QSTA is associated with an ordinary AP, i.e., nQAP, then the QSTA is functioning just like an ordinary STA, which is indicated by setting the *QoS subfield* to 0. In that case, the TID value is meaningless and

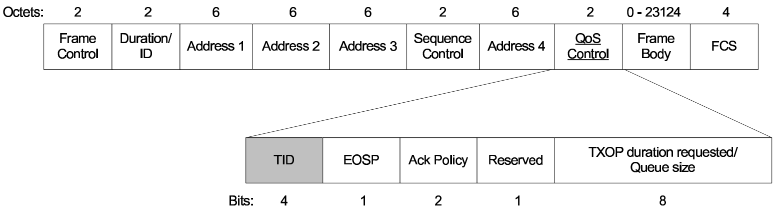


Figure 4.9: TID field in QoS Control field.

all frames from the station are treated as frames with priority of *Contention*, indicating that they shall be transmitted without any priority, as it is done in the DCF. Similarly, if an ordinary station, i.e., STA or nQSTA, is associated with a QAP, all frames from the station are treated as frames with priority of 0.

The *Queue size* field in *QoS Control* field of the frame header, as seen in Figure 4.9, specifies the total number of frames of the particular priority/TID the station have in its AC transmit queue, excluding the current frame.

TXOP are obtained both in EDCA and HCCA, such that the former is referred to as EDCA TXOP, and the later is referred to as HCCA TXOP or Polled TXOP. An EDCA TXOP is obtained as soon as a QSTA wins access to the medium while operating in EDCA. A HCCA TXOP is granted by the HC while operating in HCCA, such that the HC polls individual stations to grant HCCA TXOPs based on their requirements.

A QSTA can specify the intention to transmit multiple frames in a TXOP by setting the *Duration/ID* field in the frame header, such that it also includes the time required to transmit the additional frames. While operating in HCCA, a QSTA can request the TXOP of particular duration by setting the *TXOP duration requested* subfield of *QoS Control* field shown in Figure 4.9. The *TID* field in that case indicates the AC for which the TXOP is being requested. The HC/QAP may then assign a TXOP of the size requested or of a smaller size.

EDCA parameters are defined in *EDCA Parameter Set* element, and are periodically advertised by the QAP in selected frames (beacons). QAP can adapt these parameters dynamically, depending on the network condition. Figure 4.10 shows the structure of EDCA Parameter Set element.

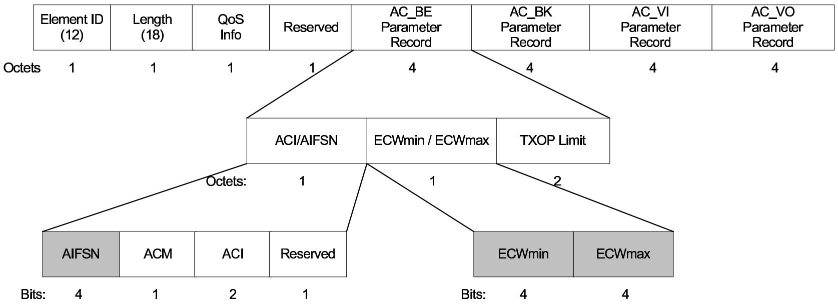


Figure 4.10: EDCA Parameter Set element.

The values of EDCA parameters are specified in the subfields *AIFSN*, *ECWmin*, *ECWmax*, and *TXOP Limit*, in the EDCA Parameter Set element. All QSTAs that receive the EDCA Parameter Set element from QAP update their EDCA parameter

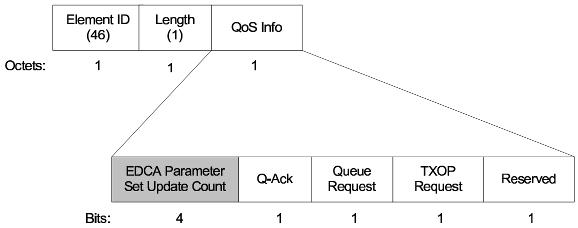


FIGURE 4.11: QOS CAPABILITY ELEMENT AND QOS INFO FIELD.

values and use new values to contend for the medium. The draft standard specifies the default values of EDCA parameters if not advertised by the QAP, as presented in Table 4.2.

Every time QAP updates the EDCA parameters, it increments the value of *EDCA Parameter Set Update Count* field in *QoS Info* field in the *QoS Capability element* sent in selected frames. The QSTAs use this information to confirm that they are using the latest set of EDCA parameters. The structure of QoS Capability element is illustrated in Figure 4.11.

4.5 Summary

IEEE 802.11e is an enhanced version of IEEE 802.11 WLAN standard to support quality of service (QoS). IEEE 802.11e provides support for QoS by introducing service differentiation. Four Access Categories (ACs) are defined to serve different types of data traffic. These Access Categories are named AC_BK, AC_BE, AC_VI and AC_VO, for Background, Best Effort, Video and Voice data traffic types, respectively. At the higher layers, each frame from a traffic stream is assigned a priority value, called User Priority (UP), ranging from 0 to 7. At the MAC layer this priority is mapped to one of the four Access Categories. For each Access Category, an enhanced version of DCF, called Enhanced Distributed Channel Access Function (EDCAF), contends for the medium based on a different set of contention parameters. These parameters are: (1) AIFS - the time period the medium is sensed idle before the transmission or backoff is started, (2) CWmin, CWmax - size of the Contention Window used for backoff, and, (3) TXOP Limit - the maximum duration of TXOP (Transmission Opportunity), which is the time period during which an EDCAF has the right to transmit after the medium is acquired. AIFS (Arbitration Inter-Frame Space) is derived from the equation AIFS = AIFSN x Slot Time + SIFS, where AIFSN (Arbitration Inter-Frame Space Number) specifies the number of time slots in addition to the SIFS time period an AIFS consists of. The values of EDCA parameters are AC specific and are set in a way that a higher priority AC waits a shorter AIFS time period before it can access the medium, chooses backoff values from a smaller Contention Window, and occupies the medium for a longer time period, compared to a lower priority AC. Basically, the higher the priority of an AC, the smaller the AIFS, CWmin and CWmax, and larger the TXOP Limit. These parameters are referred to as EDCA Parameters and are periodically advertised by the AP. The standard specifies the default values for EDCA Parameters if not advertised by the AP.

Besides the different sets of EDCA parameters for different ACs, the rest of medium access mechanism is same as in DCF, i.e., as the medium becomes idle for the AIFS time period, the EDCAF chooses a backoff value and starts decreasing its backoff timer.

Transmission is started as the backoff timer reaches to zero.

As the four EDCAFs inside the station contend for the medium independently of each other, it may result into a situation where more than one EDCAF in the same station count their backoff timers to zero and try to transmit at the same time. This is then referred to as *internal collision* and is resolved such that among the colliding EDCAFs, the EDCAF for higher priority Access Category transmits and the lower priority EDCAF backoffs, just as if an external collision occurred.

A new feature of 802.11e, called Contention Free Bursting (CFB), enables an EDCAF to transmit multiple frames once the medium/TXOP is acquired, without contending for the medium for every frame. The consecutive frame transmissions are separated by SIFS time periods. The transmission duration of CFB is bounded by TXOP Limit.

Chapter 5

Introduction to GloMoSim

5.1 Introduction

GloMoSim [3] stands for *Glo*bal *Mo*bile Information System *Sim*ulator and is a scalable simulation environment for mobile and wireless networks, developed at UCLA Parallel Computing Laboratory. GloMoSim is a discrete event simulator built using PARSEC [40], a C based environment designed for parallel simulations, also developed at UCLA Parallel Computing Laboratory. GloMoSim is built using a layered approach similar to the OSI seven layers network architecture, with standard APIs between layers. This makes it easy to implement and integrate new protocols and models at different layers. A wide range of models and protocols are supported at different layers as shown in Table 5.1. GloMoSim is a scalable simulator and has capabilities to simulate thousands of mobile nodes.

Layers	**Protocols**
Mobility	Random Waypoint, Random Drunken, Trace Based
Radio Propagation	Two-Ray and Free Space
Radio Model	Noise Accumulating
Packet Reception Models	SNR bounded, BER based with BPSK/QPSK modulation
Data Link (MAC)	CSMA, IEEE 802.11 and MACA
Network (Routing)	IP with AODV, Bellman-Ford, DSR, Fisheye, LAR scheme 1, ODMRP, WRP
Transport	TCP and UDP
Application	CBR, FTP, HTTP and Telnet

TABLE 5.1: PROTOCOLS AND MODELS SUPPORTED AT EACH LAYER.

5.2 Architecture

GloMoSim is designed using a layered approach with standard APIs between layers, as shown in Figure 5.1. The layered design benefits from the features of modular development, such that the layers as well as the protocols and models at different layers are treated as independent modules and can be modified or replaced without affecting other layers. The modular design allows people to develop and implement new protocols at different layers such that the design conforms with the standard API used between the layers. The *plug and play* feature enables people to work without the concern for the inner working of the simulator.

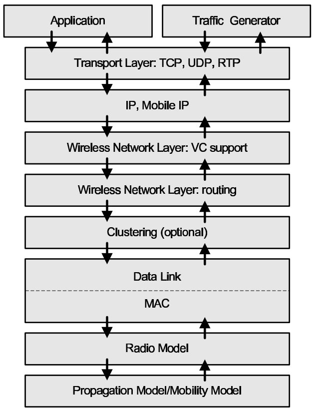


FIGURE 5.1: ARCHITECTURE OF GLOMOSIM.

GloMoSim is written in C and utilizes PARSEC (Parallel Simulation Environment for Complex Systems) [40], which is a simulation environment developed for parallel and sequential execution of discrete event simulations. By discrete event it means that the execution is mainly based on event handling, i.e., the execution consists of set of events and as an event occurs, the appropriate action is taken in its response. An event is defined as an incident which results in the change of state of the system. A certain event or combination of events may invoke other events and so on, and this is how the execution proceeds. Some examples are arrival of packet at a particular layer, expiry of a timer, etc. Events only occur at discrete units of time are not permitted to occur in between these units of time.

GloMoSim is freely available for educational purposes. However the free version only supports the sequential execution of simulations.

5.3 Summary

GloMoSim *Glo*bal *Mo*bile Information System *Sim*ulator is a scalable simulation environment for mobile and wireless networks, developed at UCLA Parallel Computing Laboratory. GloMoSim is a discrete event simulator written in PARSEC, a C based language. GloMoSim is built using a layered approach similar to the OSI seven layers network architecture. A wide range of models and protocols are supported at different layers. The layered design benefits from the features of modular development, and enables people to develop and implement new protocols at different layers. GloMoSim supports parallel and sequential execution of discrete event simulations. By discrete event it means that the execution is mainly based on event handling. The execution consists of set of events, where event is defined as an incident which results in the change of state of the system. As an event occurs, the appropriate action is taken in its response. A certain event or combination of events may invoke other events and so on, and this is how the execution proceeds. GloMoSim is freely available for educational purposes. However the free version only supports the sequential execution of simulations.

Chapter 6

Design and Implementation

The chapter presents an overview of the design and implementation of IEEE 802.11e EDCA in GloMoSim v2.03. GloMoSim currently supports IEEE 802.11 DCF together with many other MAC protocols. The task was to design and implement IEEE 802.11e EDCA on the basis of the available DCF design.

The following sections describe the design of important EDCA components, that includes priority, AC transmit queues, Access Category and EDCAF, together with the details of updates to the input and output of the system.

6.1 Inclusion of Priority

One of the first implementation issues was to decide how to assign priority to a frame (or application). As discussed in Section 4.3.1, the draft standard states that each frame from the higher layer arrives at the MAC layer along with a priority value (called User Priority), but it does not specify how such a priority is assigned and carried through the higher layers. However, assuming that the most appropriate way would be to have this priority assigned by the application itself, in our design the priority is specified along with the specification of the application in the input file, as described in the following section.

6.1.1 Defining Priority

The line defining a CBR application in the input file *app.conf* now includes the priority as the 8th parameter:

```
CBR <src><dest><items to send><item size><interval><start time><end time><priority>
```

The value of priority parameter ranges from 0 to 3, enabling total four priorities to be assigned. The draft standard defines 8 different priorities which then are mapped into 4 ACs (Access Categories) at the MAC layer (see Section 4.3.1), but we found it sufficient to use only 4 priorities and then map them directly to 4 ACs. Another difference is that in our design priority is used in opposite order, i.e., 0 is the highest and 3 is the lowest priority. The reason for using it in opposite order is described in subsequent sections.

Below is an example how a 802.11e scenario with CBR applications is defined in the *app.conf* file. The example shows two stations with multiple CBR applications generating traffics of different priorities.

```
CBR 1 0 100 512 0.05S 0S 0S 0
CBR 1 0 100 1460 0.9S 9S 0S 3
CBR 1 0 50 512 0.25S 20S 0S 1
CBR 1 0 50 1024 1.5S 0S 40S 2
CBR 2 0 20 512 0.07S 15S 0S 2
CBR 2 0 30 1460 0.1S 20S 0S 0
CBR 2 0 20 512 1.10S 0S 30S 1
CBR 2 0 30 512 0.25S 20S 0S 1
CBR 2 0 30 1024 0.2S 10S 0S 0
```

The design has been updated such that the value of priority parameter does not have any effect when a MAC protocol other than 802.11e is used.

As we only modified CBR to adapt for the priority, therefore only CBR application can be used with 802.11e.

6.1.2 Adaptation of Priority

Priority is carried from Application layer to Network layer in the same way it is done to carry priority of different applications and routing protocols in the existing design. Figure 6.1 shows the adaptation of priority at different layers.

At the Network layer, priority is included in the IP header in the field *ip_ tos* and sent to the MAC layer. At the MAC layer, this is next mapped into the corresponding AC. As described in Section 4.3.1, 802.11e defines 4 different ACs, namely, AC_BK, AC_BE, AC_VI and AC_VO, for Background, Best Effort, Video and Voice data traffic, respectively. Table 6.1 shows how priority is mapped to AC.

Priority	Access Category
0	AC_VO
1	AC_VI
2	AC_BE
3	AC_BK

TABLE 6.1: PRIORITY TO ACCESS CATEGORY MAPPING.

6.2 Implementation of AC Transmit Queues

Four AC transmit queues have been implemented at MAC layer. In the existing design of GloMoSim, Network layer also contains three queues for the IP protocol. These queues are used to give priority to certain applications on others, and to routing protocols on applications. One possible design was to use same queues as 802.11e AC transmit queues as well, but we found it better to implement 802.11e AC transmit queues at MAC layer to remain sincere to the standard.

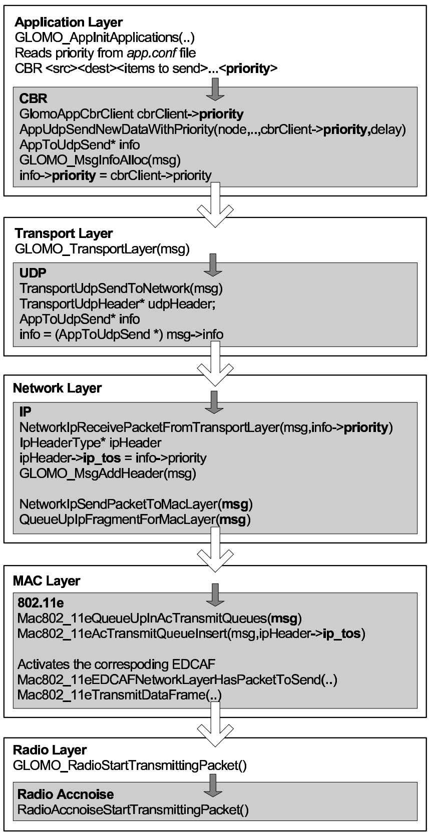


FIGURE 6.1: ADAPTATION OF PRIORITY.

6.2.1 IP Queues in the Existing Design

In the existing design, there are three queues for IP protocol at the Network layer. These queues, called IP Queues, are used to give priority to packets associated to different applications and routing protocols. The packets related to routing protocols are assigned the highest priority of 0 and are inserted in the first queue, packet from application CBR are assigned priority of 1 and are inserted in second queue, and packets from rest of the applications (FTP, Generic FTP, HTTP and Telnet) are assigned priority of 2 and are inserted in third queue. While forwarding a packet to MAC layer, packets are retrieved in the order of priority, i.e., packets from the first queue are retrieved first, then from the second queue and then from the third queue. As routing packets are always inserted in first queue, they are always sent first.

6.2.2 802.11e AC Transmit Queues

Four queues have been added at the MAC layer as the 802.11e AC Transmit Queues. When using 802.11e as the MAC protocol, the Network layer IP queues are bypassed and packets are directly inserted into AC transmit queues at MAC layer. Packets are then retrieved and dequeued from AC transmit queues throughout the simulation.

The CBR packets are inserted in these four queues such that the packets of highest priority 0 (AC_VO) are inserted in the first queue, packets of next highest priority 1 (AC_VI) are inserted in the second queue and so on.

As described in previous section, we used priorities in opposite order, i.e., 0 is the highest and 3 is the lowest priority. This is done to avoid modifications in the existing code, especially in the routing protocols files, because, as described above, the existing design uses priorities in this order, i.e., routing packets are assigned the highest priority of 0.

AC Transmit Queue	Source
1	CBR with priority 0 (AC_VO), Routing Protocols
2	CBR with priority 1 (AC_VI)
3	CBR with priority 2 (AC_BE), FTP, HTTP and Telnet
4	CBR with priority 3 (AC_BK)

TABLE 6.2: AC TRANSMIT QUEUES USED FOR DIFFERENT TYPES OF APPLICATIONS.

In our design, routing packets are also inserted into the first queue, i.e., the queue for packets with priority of AC_VO. This shows that routing protocols packets are transmitted with the priority of AC_VO. Another possible design was to use a separate queue for routing packets so that they could always be transmitted prior to CBR packets with priority of AC_VO, but it leads to the confusion that which EDCAF, in other words, what AC parameter values, should be used to transmit routing packets. But in principle as routing packets are always transmitted before data packets, we found it feasible to use same queue for both type of packets.

As we did not modify the rest of the applications (FTP, Generic FTP, HTTP and Telnet) to adapt for the 802.11e protocol, the packets of these applications are always inserted in the third of the four AC transmit queues, i.e., queue for AC_BE, because they are already assigned with priority of 2 in the existing design.

Note that if 802.11e is used as the MAC protocol but with an application other than CBR, then the fourth AC transmit queue, i.e., queue for AC_BK, will always remain empty. Moreover, if both 802.11e and CBR are used, and any of the other applications, i.e., FTP, Generic FTP, HTTP, Telnet, is also used, then third AC transmit queue, i.e., queue for AC_BE, will contain packets of both CBR and that application. Table 6.2 summarizes how packets associated to different applications are inserted into AC transmit queues when 802.11e is used as the MAC protocol.

6.3 Implementation of EDCA

Four Access Categories (ACs) have been implemented each with its own AC transmit queue and channel access function, EDCAF, as illustrated in Figure 6.2. Each of the EDCAFs implements an independent instance of CSMA/CA protocol with backoff. These EDCAFs at AC transmit queues behave like virtual stations inside the real station and contend for the medium independently of each other.

Following sections describe the key data structures used in the implementation of EDCA.

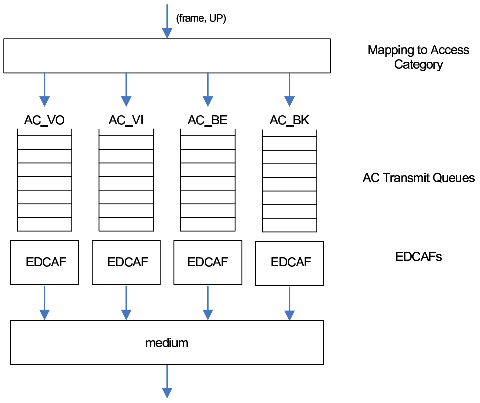
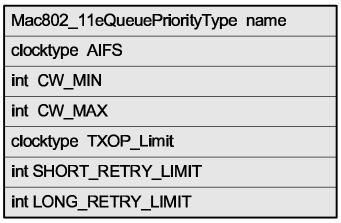


Figure 6.2: AC transmit queues and EDCAFs.

6.3.1 M802_11eAccessCategory

Access Category (AC) is defined by the data structure M802_11eAccessCategory, illustrated in Figure 6.3.



Mac802_11eQueuePriorityType name
clocktype AIFS
int CW_MIN
int CW_MAX
clocktype TXOP_Limit
int SHORT_RETRY_LIMIT
int LONG_RETRY_LIMIT

Figure 6.3: Data structure for Access Category.

As it is seen above, the data structure contains fields to keep the values of EDCA parameters, i.e., AIFS, CWmin, CWmax, and, TXOPLimit. Here the field *name* keeps the name of AC, i.e., the designations assigned by the draft standard to represent Access Categories, namely AC_BK, AC_BE, AC_VI, and, AC_VO.

6.3.2 M802_11eEDCAF

EDCAF is defined by the data structure M802_11eEDCAF. Important fields of M802_11eEDCAF are illustrated in Figure 6.4.

The data structure GlomoMac802_11e, that contains all necessary parameters and states related to the MAC 802.11e, now keeps four EDCAFs, namely EDCAF_BK, EDCAF_BE, EDCAF_VI, and EDCAF_VO, that are instances of data structure M802_11eEDCAF illustrated in Figure 6.4.

As it is seen from the design, each of the four EDCAFs keeps its own state, current size of Contention Window (CW), current backoff timer (BO), and, NAV, just like in DCF in the existing design of GloMoSim for the 802.11 MAC.

The EDCAF keeps the information of the AC it contends for in the instance of M802_11eAccessCategory (see Figure 6.3). The EDCA parameters, AIFS, CW_MIN, CW_MAX, and TXOP_Limit, are initialized to the values in the input file *config.in* if

M802_11eAccessCategory AC
int state
int prevState
clocktype CW
clocktype BO
clocktype lastBOTimeStamp
clocktype NAV
int SSRC
int SLRC
int timerSequenceNumber
long pktsToSend
long pktsSentUnicast
long retxDueToCts
long retxDueToAck
long pktsDropped
long pktsLost
clocktype macDelayTimeStamp
clocktype totalMacDelay
long totalBackoffsSet
long accumulatedBackoff
long accumulatedCW
long internalCollisions

FIGURE 6.4: DATA STRUCTURE FOR EDCAF.

specified by the user, or, to the default values, otherwise. The default EDCA parameter values defined by the draft standard are presented in Table 4.2.

The data structure also contains the information such as number of packets the EDCAF/AC has to send (i.e, number of packets in the AC transmit queue), number of packets sent, retransmissions due to ACK and CTS timeouts, number of packets dropped due to retry limit, number of packets lost due to queue overflow, and, number of times EDCAF collided with other EDCAFs (internal collisions). The information of total MAC delay, number of times EDCAF set its backoff, sum of all backoff values, and, sum of Contention Window sizes, is used to calculate average MAC delay, average backoff value and average Contention Window size, respectively.

6.3.3 GlomoMac802_11e

The data structure GlomoMac802_11e represents the MAC of the entire station. Since a station contains four EDCAFs one for each AC, as described in Section 4.3.2, the data structure GlomoMac802_11e keeps four EDCAFs, namely EDCAF_BK, EDCAF_BE, EDCAF_VI, and EDCAF_VO, that are instances of data structure M802_11eEDCAF, illustrated in Figure 6.4. Contrary to the design of DCF for 802.11, data structure GlomoMac802_11e does not keep MAC information related to the MAC functionality, since that is now kept by individual EDCAFs. Figure 6.5 shows important fields of GlomoMac802_11e.

As it is seen in Figures 6.4 and 6.5, and described above, information such as number of packets the EDCAF/AC has to send, numbers of packets sent, retransmissions due to CTS and ACK timeouts, number of packets dropped due to retry limit, and, number of packets lost due to queue overflow, is kept in M802_11eEDCAF instead of

int state
int prevState
BOOL IsInExtendedIfsMode
unsigned int timerSequenceNumber
int totalInternalCollisions
clocktype ctsOrAckTransmissionDuration
long pktsSentBroadcast
long pktsGotUnicast
long pktsGotBroadcast
int rtsPacketsIgnoredDueToBusyChannel
int rtsPacketsIgnoredDueToNAV
Mac802_11eAcTransmitQueueType* ACTransmitQueues
M802_11eEDCAF* EDCAF_BK
M802_11eEDCAF* EDCAF_BE
M802_11eEDCAF* EDCAF_VI
M802_11eEDCAF* EDCAF_VO
M802_11eTXOP TXOP
BOOL queuesAreEmpty

FIGURE 6.5: DATA STRUCTURE FOR 802.11E.

GlomoMac802_11e because it is more relevant to individual EDCAF compared to the overall station. On the other hand, information such as number of packets received (both unicast and broadcast), and, number of RTS packets ignored, is contained in GlomoMac802_11e instead of M802_11eEDCAF since this information is more relevant to entire station compared to individual EDCAF. The fields *totalInternalCollisions* and *queuesAreEmpty* keep the information of total number of internal collisions occurred between EDCAFs and whether the AC transmit queues are empty or not, respectively.

6.3.4 Working of EDCA

As a frame arrives at MAC layer, first its priority is mapped into AC, as described in previous sections. Next the frame is inserted into corresponding AC transmit queue.

Each of the four EDCAFs which receives frame in its AC transmit queue from the above layer starts contending for the medium independently of other EDCAFs, behaving like a virtual station inside the real station. This means in a station maximum four EDCAFs contend for the medium, depending if all of them have frames to send. As an EDCAF for higher priority AC has to wait a smaller duration of AIFS, it starts counting down its backoff timer prior to an EDCAF for lower priority AC. Moreover, as a higher priority AC has smaller minimum and maximum Contention Window limits compared to a lower priority AC, EDCAF for higher priority AC most of the time gets a smaller backoff value and thus has to wait less time. As an EDCAF is able to access the medium, other EDCAFs in the same station pause their backoff timers, and continue to count down once the medium becomes idle again for the AIFS time period. Thus, at a certain time, a lower priority EDCAF will have smaller backoff value because while the higher priority EDCAF has to choose a new backoff value for every next frame, the lower priority EDCAF continues to count down its paused backoff timer. The whole process is illustrates in Figure 4.4 in Section 4.3.3. The example assumes that all four EDCAFs have packets to send and the medium has been busy.

Just like in the original design, an EDCAF can transmit a frame immediately after

AIFS, without waiting for backoff time, in case the time the frame arrives at the MAC layer, the last post backoff has already been finished, i.e., the corresponding AC transmit queue is empty, and the medium has been idle longer than the AIFS time period. In case there are more than one EDCAF with pending frames, then the EDCAF at the highest priority AC queue transmits and others choose backoff values. On the other hand, if the medium is sensed busy, as indicated either by the virtual (NAV) or physical carrier sensing, then all EDCAF (EDCAFs with frames in their queues) choose backoff values after waiting the medium idle at least for the AIFS time period. The EDCAF which counts its timer down to zero first gets access to the medium and others pause their backoff timers.

As an EDCAF finishes transmission of a frame and has a pending frame in its AC queue, it performs the post backoff after waiting the medium to be idle at least for the AIFS time period. Other EDCAFs which paused their backoff timers (in case there were more than one EDCAF with frames to send) also continue to decrement their backoff timers after waiting for the AIFS time period. In case there is an EDCAF which did not contend for the medium the last time, i.e., its queue was empty at that time, but has pending frame now, then it also chooses backoff value after waiting for the AIFS time period.

Internal collisions may occur if the backoff timers for two or more EDCAFs reach zero at the same time, or when two or more EDCAFs accidentally get the same backoff values, in which case the EDCAF with higher priority AC gets the access to the medium and the lower priority EDCAF doubles its Contention Window and chooses a new backoff value, as shown in Figure 4.5.

As a station receives a frame that is not directed to it, while all four EDCAFs set their NAVs to the time required to transmit the frame and to receive the corresponding ACK, only the EDCAFs which have frames to send start decrementing their NAV timers. As the NAV for an EDCAF expires, it waits the medium to be idle for the AIFS time period, chooses a backoff value (if no backoff value was already chosen), and starts decrementing its backoff timer.

An unsuccessful transmission, i.e., an external collision, is determined if the sender EDCAFs at the colliding stations do not receive ACK frame within a specified ACK timeout period. In that case, in a colliding station, all EDCAFs with pending frames contend for the medium again, such that the EDCAF which faced unsuccessful transmission chooses a new backoff value from a doubled Contention Window, and together with other EDCAFs which paused their backoff timers, continues to count its backoff timer down after waiting the medium to be idle for the AIFS time period. Figure 4.6 in Section 4.3.3 explains the external collision and the subsequent actions.

6.3.4.1 States of M802_11eEDCAF

As it is seen from Figures 6.4 and 6.5, both data structures M802_11eEDCAF and GlomoMac802_11e maintain their own states.

Just like in the design for the original standard in GloMoSim, an EDCAF enters into several states like a real station. The only exceptions are the transmitting states M802_11E_X_ACK and M802_11E_X_CTS, and the waiting state M802_11E_S_WFDATA. An EDCAF does not enter into the transmitting states M802_11E_X_ACK and M802_11E_X_CTS because ACK and CTS frames are not transmitted by any particular EDCAF and instead by the overall station. As the transmission of ACK and CTS frames does not have anything to do with EDCA parameters, and as they are transmitted after SIFS (following the reception of data or RTS frame) which is of the same length for all ACs, it is not necessary that they are transmitted by any particular EDCAF.

Therefore after an ACK or CTS frame is transmitted, the whole station enters into the corresponding state instead of any particular EDCAF. The third state an EDCAF does not enter into is the waiting state WFDATA, which the station enters into after transmitting the CTS frame.

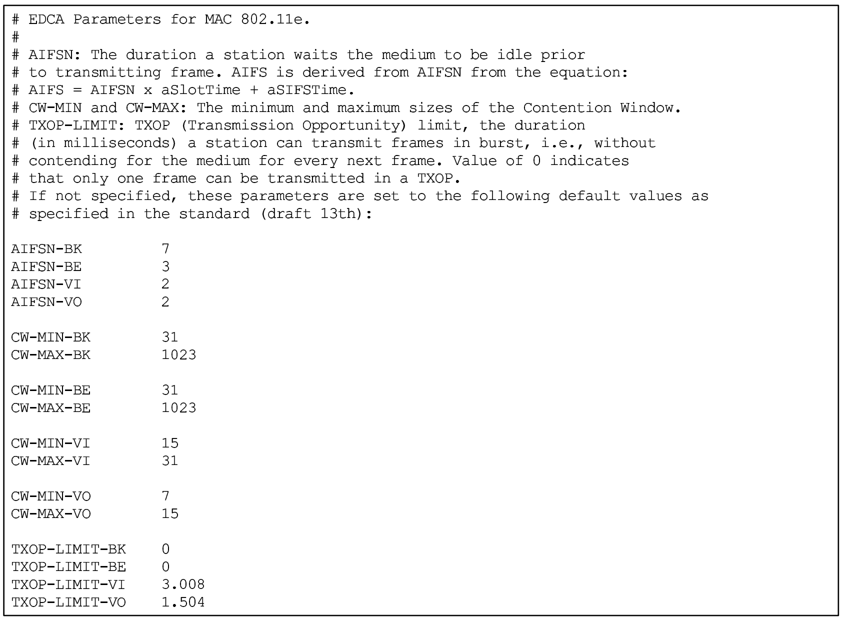
Another possible design was to include the priority/AC information in the header while transmitting a data or RTS frame, to know at the receiver side which EDCAF at the sender side sent it and thus which EDCAF at the receiver side should transmit the corresponding ACK or CTS frame, and wait for data frame in the latter case. The design was canceled because of the unnecessary computation.

6.4 Updates to the System Input

The system enables user to specify a range of parameter values in the input file *config.in*. It includes the EDCA parameter values as well as some other important parameters as described in the following subsections.

6.4.1 EDCA Parameter Values

The draft standard states that the AP is capable to adapt the values of EDCA parameters dynamically, according to the network conditions. In our design, therefore, the user is allowed to specify the desired EDCA parameter values in input file *config.in*. It allows the user to run simulations with different EDCA parameter values besides the default values specified by the standard, in order to get more realistic results. Figure 6.6 illustrates how EDCA parameter values are specified in the *config.in* file.



```
# EDCA Parameters for MAC 802.11e.
#
# AIFSN: The duration a station waits the medium to be idle prior
# to transmitting frame. AIFS is derived from AIFSN from the equation:
# AIFS = AIFSN x aSlotTime + aSIFSTime.
# CW-MIN and CW-MAX: The minimum and maximum sizes of the Contention Window.
# TXOP-LIMIT: TXOP (Transmission Opportunity) limit, the duration
# (in milliseconds) a station can transmit frames in burst, i.e., without
# contending for the medium for every next frame. Value of 0 indicates
# that only one frame can be transmitted in a TXOP.
# If not specified, these parameters are set to the following default values as
# specified in the standard (draft 13th):

AIFSN-BK        7
AIFSN-BE        3
AIFSN-VI        2
AIFSN-VO        2

CW-MIN-BK       31
CW-MAX-BK       1023

CW-MIN-BE       31
CW-MAX-BE       1023

CW-MIN-VI       15
CW-MAX-VI       31

CW-MIN-VO       7
CW-MAX-VO       15

TXOP-LIMIT-BK   0
TXOP-LIMIT-BE   0
TXOP-LIMIT-VI   3.008
TXOP-LIMIT-VO   1.504
```

FIGURE 6.6: SPECIFYING EDCA PARAMETERS IN CONFIG.IN FILE.

6.4.2 AC Transmit Queue Sizes

The draft standard does not specify anything about the AC transmit queue sizes for individual ACs. Anyhow, since some of the important simulation results are heavily influenced by the queue sizes, such as end-to-end delay and packet loss ratio, in our design the user is allowed to specify the desired queue sizes in the input file. Figure 6.7 illustrates how the queues sizes are specified in the *config.in* file.

```
# AC queue sizes for MAC 802.11e.
# The size is specified in terms of number of frames.
# The standard (draft 13th) does not specify any values for AC queue sizes.
# If not specified, the default value is 100 frames.

QUEUE-SIZE-BK 100
QUEUE-SIZE-BE 100
QUEUE-SIZE-VI 100
QUEUE-SIZE-VO 100
```

FIGURE 6.7: SPECIFYING AC TRANSMIT QUEUE SIZES IN CONFIG.IN FILE.

6.4.3 AC Retry Limits

Although the draft standard indicates that the station maintains retry counter for each AC, it does not specify any values for retry limits for individual ACs. The retry limit values used in the original standard are 7 and 4 for short and long retry limits, respectively. But for 802.11e, using the same values for all four ACs does not look accurate. Since the higher priority traffic, e.g., voice or video, is very sensitive to end-to-end delays, bigger retry limit values for higher priority ACs may have significant effect on their performance. It is due to the reason that bigger values of retry limits result into higher end-to-end delays (because of higher number of retransmissions). Therefore, the user is enabled to specify the desired retry limit values in the input file. Figure 6.8 shows how retry limit values are specified in the *config.in* file.

```
# AC short and long retry limits for MAC 802.11e.
# The standard (draft 13th) does not specify any values for AC short and long retry limits.
# If not specified, the default values are 7 and 4 for short and long retry limits,
respectively.

SHORT-RETRY-LIMIT-BK 7
LONG-RETRY-LIMIT-BK 4

SHORT-RETRY-LIMIT-BE 7
LONG-RETRY-LIMIT-BE 4

SHORT-RETRY-LIMIT-VI 7
LONG-RETRY-LIMIT-VI 4

SHORT-RETRY-LIMIT-VO 7
LONG-RETRY-LIMIT-VO 4
```

FIGURE 6.8: SPECIFYING AC RETRY LIMITS IN CONFIG.IN FILE.

6.5 Updates to the System Output

The statistics information related to the 802.11e MAC layer is printed in the output file *glomo.stat*, just like it is done for the other layers. A wide range of useful information is printed to the output file for each AC. It includes number of packets sent, number of retransmissions, number of packets drops and losses, average MAC delay, average Contention Window size, average backoff value, and number of internal collisions. Figure 6.9 illustrates a snapshot of *glomo.stat* file.

The Application layer has also been updated to print the important information for each AC. It includes the important information of end-to-end delay, number of packets received, and, throughput. A snapshot of the *glomo.stat* file is presented in Figure 6.10.

```
Node:      1, Layer:        802.11e, Total pkts from network: 8140
Node:      1, Layer:        802.11e, AC_BK pkts from network: 2035
Node:      1, Layer:        802.11e, AC_BK UCAST (non-frag) pkts sent to chanl: 1
Node:      1, Layer:        802.11e, AC_BK retx pkts due to CTS timeout: 0
Node:      1, Layer:        802.11e, AC_BK retx pkts due to ACK timeout: 2
Node:      1, Layer:        802.11e, AC_BK pkt drops due to retx limit: 0
Node:      1, Layer:        802.11e, AC_BK pkt losses due to queue overflow: 1934
Node:      1, Layer:        802.11e, AC_BK average MAC delay [s]: 73.411827826
Node:      1, Layer:        802.11e, AC_BK average CW size: 63
Node:      1, Layer:        802.11e, AC_BK average backoff value: 19
Node:      1, Layer:        802.11e, AC_BK internal collisions: 0
Node:      1, Layer:        802.11e, AC_BE pkts from network: 2035
Node:      1, Layer:        802.11e, AC_BE UCAST (non-frag) pkts sent to chanl: 126
Node:      1, Layer:        802.11e, AC_BE retx pkts due to CTS timeout: 0
Node:      1, Layer:        802.11e, AC_BE retx pkts due to ACK timeout: 76
Node:      1, Layer:        802.11e, AC_BE pkt drops due to retx limit: 0
Node:      1, Layer:        802.11e, AC_BE pkt losses due to queue overflow: 1809
Node:      1, Layer:        802.11e, AC_BE average MAC delay [s]: 1.583286044
Node:      1, Layer:        802.11e, AC_BE average CW size: 61
Node:      1, Layer:        802.11e, AC_BE average backoff value: 29
Node:      1, Layer:        802.11e, AC_BE internal collisions: 0
Node:      1, Layer:        802.11e, AC_VI pkts from network: 2035
Node:      1, Layer:        802.11e, AC_VI UCAST (non-frag) pkts sent to chanl: 1143
Node:      1, Layer:        802.11e, AC_VI retx pkts due to CTS timeout: 0
Node:      1, Layer:        802.11e, AC_VI retx pkts due to ACK timeout: 1227
Node:      1, Layer:        802.11e, AC_VI pkt drops due to retx limit: 4
Node:      1, Layer:        802.11e, AC_VI pkt losses due to queue overflow: 790
Node:      1, Layer:        802.11e, AC_VI average MAC delay [s]: 0.171813104
Node:      1, Layer:        802.11e, AC_VI average CW size: 23
Node:      1, Layer:        802.11e, AC_VI average backoff value: 11
Node:      1, Layer:        802.11e, AC_VI internal collisions: 25
Node:      1, Layer:        802.11e, AC_VO pkts from network: 2035
Node:      1, Layer:        802.11e, AC_VO UCAST (non-frag) pkts sent to chanl: 2026
Node:      1, Layer:        802.11e, AC_VO retx pkts due to CTS timeout: 0
Node:      1, Layer:        802.11e, AC_VO retx pkts due to ACK timeout: 2126
Node:      1, Layer:        802.11e, AC_VO pkt drops due to retx limit: 4
Node:      1, Layer:        802.11e, AC_VO pkt losses due to queue overflow: 0
Node:      1, Layer:        802.11e, AC_VO average MAC delay [s]: 0.079350391
Node:      1, Layer:        802.11e, AC_VO average CW size: 11
Node:      1, Layer:        802.11e, AC_VO average backoff value: 5
Node:      1, Layer:        802.11e, AC_VO internal collisions: 25
Node:      1, Layer:        802.11e, Total UCAST (non-frag) pkts sentto chanl: 3296
Node:      1, Layer:        802.11e, Total retx pkts due to CTS timeout: 0
Node:      1, Layer:        802.11e, Total retx pkts due to ACK timeout: 3431
Node:      1, Layer:        802.11e, Total pkt drops due to retx limit: 8
Node:      1, Layer:        802.11e, Total pkt losses due to queue overflow: 4533
Node:      1, Layer:        802.11e, Total internal collisions: 25
Node:      1, Layer:        802.11e, RTS Packets ignored due to Busy Channel: 0
Node:      1, Layer:        802.11e, RTS Packets ignored due to NAV: 0
```

Figure 6.9: MAC layer statistics in glomo.stat file.

```
Node:      0, Layer:    AppCbrServer, (0) Client address: 1
Node:      0, Layer:    AppCbrServer, (0) Priority 0 (AC_VO), First packet received at [s]: 0.082010109
Node:      0, Layer:    AppCbrServer, (0) Priority 0 (AC_VO), Last packet received at [s]: 199.962872150
Node:      0, Layer:    AppCbrServer, (0) Priority 0 (AC_VO), Average end-to-end delay [s]: 0.198946935
Node:      0, Layer:    AppCbrServer, (0) Priority 0 (AC_VO), Session status: Not closed
Node:      0, Layer:    AppCbrServer, (0) Priority 0 (AC_VO), Total number of bytes received: 2074624
Node:      0, Layer:    AppCbrServer, (0) Priority 0 (AC_VO), Total number of packets received: 2026
Node:      0, Layer:    AppCbrServer, (0) Priority 0 (AC_VO), Throughput (bits per second): 83019
Node:      0, Layer:    AppCbrServer, (0) Client address: 1
Node:      0, Layer:    AppCbrServer, (0) Priority 1 (AC_VI), First packet received at [s]: 0.121813048
Node:      0, Layer:    AppCbrServer, (0) Priority 1 (AC_VI), Last packet received at [s]: 199.934615286
Node:      0, Layer:    AppCbrServer, (0) Priority 1 (AC_VI), Average end-to-end delay [s]: 0.060719509
Node:      0, Layer:    AppCbrServer, (0) Priority 1 (AC_VI), Session status: Not closed
Node:      0, Layer:    AppCbrServer, (0) Priority 1 (AC_VI), Total number of bytes received: 2398780
Node:      0, Layer:    AppCbrServer, (0) Priority 1 (AC_VI), Total number of packets received: 1643
Node:      0, Layer:    AppCbrServer, (0) Priority 1 (AC_VI), Throughput (bits per second): 96009
Node:      0, Layer:    AppCbrClient, (3) Server address: 1
Node:      0, Layer:    AppCbrClient, (3) Priority 0 (AC_BK), First packet sent at [s]: 0.000000000
Node:      0, Layer:    AppCbrClient, (3) Priority 0 (AC_BK), Last packet sent at [s]: 199.977144107
Node:      0, Layer:    AppCbrClient, (3) Priority 0 (AC_BK), Session status: Not closed
Node:      0, Layer:    AppCbrClient, (3) Priority 0 (AC_BK), Total number of bytes sent: 700000
Node:      0, Layer:    AppCbrClient, (3) Priority 0 (AC_BK), Total number of packets sent: 8750
Node:      0, Layer:    AppCbrClient, (3) Priority 0 (AC_BK), Throughput (bits per second): 28000
Node:      0, Layer:    AppCbrClient, (2) Server address: 1
Node:      0, Layer:    AppCbrClient, (2) Priority 1 (AC_BE), First packet sent at [s]: 0.000000000
Node:      0, Layer:    AppCbrClient, (2) Priority 1 (AC_BE), Last packet sent at [s]: 199.898333881
Node:      0, Layer:    AppCbrClient, (2) Priority 1 (AC_BE), Session status: Not closed
Node:      0, Layer:    AppCbrClient, (2) Priority 1 (AC_BE), Total number of bytes sent: 2400240
Node:      0, Layer:    AppCbrClient, (2) Priority 1 (AC_BE), Total number of packets sent: 1644
Node:      0, Layer:    AppCbrClient, (2) Priority 1 (AC_BE), Throughput (bits per second): 96009
```

Figure 6.10: Application layer statistics in glomo.stat file.

The priority/AC has not been included in any of the frame headers, including the MAC header. Instead, the priority field has been added only in the data structures representing the CBR Server and CBR Client. At the Application layer, the CBR Client sets its priority to the value specified in the *app.conf* file, includes the priority information in the data structure CBRData and adds the CBRData in the packet before forwarding packet to the Transport layer. At the Application layer at the receiver side, the corresponding CBR Server retrieves the priority of the packet (or CBR Client) from the CBRData. We found these modifications enough to print the statistics in glomo.stat file, instead of including the priority in all headers.

6.6 Summary

GloMoSim currently supports IEEE 802.11 DCF together with many other MAC protocols. The task was to design and implement IEEE 802.11e EDCA on the basis of the available DCF design. One of the first implementation issues was to decide how to assign priority to a packet. The most appropriate method has been adapted such that the priority is assigned at the Application layer, which then travels down to the MAC layer through different layers. For it the syntax defining a CBR application in the input file *app.conf* has been updated to includes the priority. At the Network layer, priority is included in the IP header and sent to the MAC layer.

For the implementation of EDCA, four Access Categories (ACs) have been implemented each with its own AC transmit queue and EDCAF. Access Category is defined by the data structure M802_11eAccessCategory, which contains fields to keep the values of EDCA parameters, AIFS, CWmin, CWmax and TXOPLimit. EDCAF is defined by the data structure M802_11eEDCAF, which enables an EDCAF to keep its state, Contention Window, backoff timer and NAV, just like in DCF in the existing design of 802.11 in GloMoSim. The data structure `GlomoMac802_11e`, that represents the overall MAC layer of a station, keeps four instances of M802_11eEDCAF as the EDCAFs for the four ACs.

As a frame arrives at MAC layer, first its priority is mapped into AC and then it is inserted into corresponding AC transmit queue. As an EDCAF receives frame, it starts to contend for the medium independently of other EDCAFs. The EDCAF for a high priority AC has to wait less for the medium because of its smaller AIFS, CWmin and CWmax values. As an EDCAF is able to access the medium, other EDCAFs pause their backoffs timers, and resume as the medium becomes idle again for AIFS period.

In case of an internal collision, the EDCAF with the highest priority AC among the colliding EDCAFs gets access to the medium and others backoff. In case of an unsuccessful transmission (external collision), in the colliding station, all EDCAFs with pending frames contend for the medium again such that the colliding EDCAF chooses a new backoff value from a doubled Contention Window while others resume their paused backoff timers.

The draft standard states that the AP is capable of adapting the values of EDCA parameters dynamically, according to the network conditions. Therefore, in our design the user is allowed to specify the desired EDCA parameter values in input file *config.in*. In addition to EDCA parameters, the user is also enabled to specify transmit queue sizes and retry limits for all ACs. The output has been updated to print the information for 802.11e layer as well as the Application (CBR) layer in the output file *glomo.stat*. A wide range of statistics information is printed for all ACs.

Chapter 7

Evaluation

7.1 Introduction

This chapter presents the performance evaluation of IEEE 802.11e EDCA MAC mechanism using the 802.11e EDCA model we implemented in GloMoSim. The evaluation phase is divided into two parts. The first part evaluates the IEEE 802.11e EDCA QoS scheme with some simple scenarios in order to more clearly observe the role of individual EDCA parameters in providing service differentiation. It also serves as the verification of our IEEE 802.11e EDCA design. The second part focuses on the evaluation of EDCA by considering some more realistic simulation scenarios. Comparisons of 802.11e EDCA and 802.11 DCF are also presented to analyze their performance with respect to different real world traffic scenarios. For 802.11 DCF, the model already implemented in GloMoSim is used.

Performance Metrics - The performance metrics we used for verification and evaluation are as follow:

- Throughput: The amount of data transferred in a given amount of time, usually expressed in bits per second (bps). Throughput is directly proportional to the available bandwidth/capacity.
- End-to-end delay: The total delay from the time the application at the sender side generates a packet to the time the application at the receiver side receives it. As the term suggests, end-to-end delay includes all types of delays occur at different layers as well as those occur during the transmission, i.e., transmission delay, propagation delay, queueing delays etc. [10, p. 38-39].

  As discussed in Section 3.1, end-to-end delay is the most critical parameter for interactive real-time multimedia applications such as Internet telephony, videoconferencing, VR (Virtual Reality) environments, and multiplayer network games [10, p. 83]. These applications enforce tight constraints on in time data transfer and frames with delay of more than a few tens of milliseconds are often not acceptable.
- MAC delay: MAC delay is defined as the delay occurs from the time a packet arrives at the MAC layer and the station starts to contend for it, to the time it is successfully transmitted and the corresponding ACK frame is received in

response. Thus, in contrast to end-to-end delay which includes all types of delays, MAC delay only includes the delay specific to the MAC operation, i.e., accessing the medium and then successfully transmitting. Another reason of considering MAC delay in addition to end-to-end delay is that the latter is highly influenced by the delay a frame experiences at the queue, known as the queuing delay [10, p. 38], as discussed in Section 3.1, which makes it difficult to study the traffic performance particularly in highly loaded network situations.

- Collision Rate: A collision occurs if two or more stations try to transmit at the same time. It may happen if the backoff timers for two or more stations reach zero at the same time, or if two or more stations accidentally get the same backoff values. Collisions result in retransmissions, which in turns result in wastage of bandwidth.
- Internal Collision Rate: While external collisions occur among stations, internal collisions occur when two or more EDCAFs inside the same station try to transmit at the same time.

  As discussed in Section 2.2.1, the collision rate (both internal and external) is highly dependent on the Contention Window size, i.e., larger the Contention Window size, lower the number of collisions and vice versa. The reason is that the probability of choosing same backoff values, or of decrementing backoff timers to zero at the same time, increases with decreasing size of Contention Window.
- Packet Transmission Ratio: The ratio of packets successfully transmitted to the total number of packets arrived at the MAC layer to transmit. Packet transmission ratio is directly proportional to the throughput. Higher the received throughout, higher the number of packets transmitted and vice versa.
- Packet Loss: A packet is lost due to queue overflow (at the sender side), i.e., by the time the packet arrives at the MAC layer, the corresponding AC transmit queue is already full. The packet loss rate is directly proportional to the throughput/bandwidth. Higher the throughput, faster the packets are transmitted and thus lower the chances of packet losses.

  Packet loss is also highly dependent on two parameters, the size of the queue, and, the inter-packet arrival interval. Larger the size of the queue, the lower the number of packet losses and vice versa, and, smaller the packet arriving interval, the higher the number of packet losses and vice versa.
- Packet Drops: A packet is dropped after reaching the retransmit limit (or retry limit), as discussed in Section 2.2.1. Packet drop is directly proportional to collision rate. Higher the collision rate, higher the number of retransmissions and thus higher the packet drop rate.

Next section presents the simulation model used for the evaluation together with important Physical (PHY) and MAC layer parameters.

7.2 Simulation Model

Network Topology - The basic network topology consists of an AP and varying number of stations depending on the simulation scenario, as illustrated in Figure 7.1.

All communications take place between the stations and the AP, i.e., there is no direct communication between stations. All stations are stationary, and transmission powers are set such that all stations are within each other's transmission ranges.

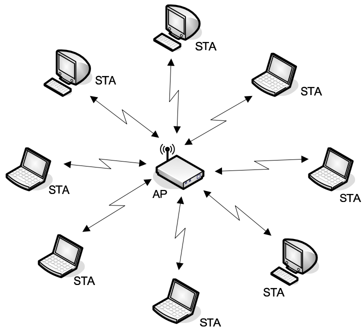


Figure 7.1: Basic network topology used for simulations.

EDCA Parameters - The default set of EDCA parameters is used as specified in the standard [2], and is summarized here in Table 7.1 for the DSSS PHY, for the comfort of the reader.

AC	CWmin	CWmax	AIFSN
AC_VO	7	15	2
AC_VI	15	31	2
AC_BE	31	1023	3
AC_BK	31	1023	7

Table 7.1: Default EDCA parameter values.

PHY Parameters - The legacy 802.11 PHY is used for simulations because currently it is the only available 802.11 PHY in GloMoSim. The legacy 802.11 PHY offers maximum data transmission rate of 2 Mbps. This proved to be a hurdle in simulating more realistic traffic scenarios since most of the 802.11 installations today are based on 802.11b and 802.11a PHYs with maximum data transmission rates of 11 and 54 Mbps respectively, and thus are able to support multimedia applications of much higher bit rates.

Important parameters for legacy 802.11 DSSS PHY are presented in Table 7.2.

Data Rate	2 Mbps
aSlotTime	20 μs
aSIFSTime	10 μs

Table 7.2: Important PHY parameters.

Furthermore, if not stated otherwise, the following assumptions are made about the simulations.

- The RTS/CTS mechanism is disabled.
- The placement of stations is randomly determined by the simulator.

- Static routing protocol is used by all stations.
- Frame fragmentation is disabled.
- The size of each AC transmit queue is 100 frames.
- The Two-Ray propagation path loss model is implemented.
- All traffic/application types are configured as CBR (Constant Bit Rate).
- UDP is implemented as the Transport layer protocol.
- The standard radio model with accumulated noise is applied.
- The CFB functionality is disabled, i.e., only one data frame is allowed to be transmitted after the medium is acquired.
- Every simulation is run for 200 seconds.

7.3 Evaluation Part 1

The section presents the performance evaluation of 802.11e QoS mechanism with the help of a set of simple scenarios. Evaluation with simple scenarios enables to more efficiently observe the effect of EDCA parameters in providing service differentiation. Moreover, it also serves as the verification of our EDCA design. Additionally, the role of individual EDCA parameters (AIFSN, CWmin and CWmax) in providing service differentiation is also studied.

Two scenarios are considered. In Scenario 1, one station is transmitting traffic streams corresponding to all four ACs to the AP, such that the load is increased from 0.1 to 4.0 Mbps by increasing the data rate of each traffic stream. Scenario 2 is similar to Scenario 1 except that the number of stations is increased to ten.

The two scenarios help to study the performance of traffic streams of all four ACs in both lightly and highly loaded network situations. Moreover, evaluation with one and ten stations allows to observe the performance of ACs with and without the presence of external contention, respectively.

Next two sections present the results and discussions for the two scenarios defined above. For each scenario, three additional cases are considered such that the EDCA parameters are *switched off* one by one in order to observe the influence of individual EDCA parameters on service differentiation. In the first case, the AIFSN for all four ACs is set to a fixed value. This eliminates the role of AIFS and thereby allows to study the role of CWmin and CWmax in realizing service differentiation. Similarly, in second case, same CWmin and CWmax values are used for all four ACs to study the service differentiation based on only AIFS. In the third case, service differentiation based on both AIFS and CWmin and CWmax is eliminated by using the same values of both parameters for all ACs.

The performance metrics considered are throughput, average end-to-end delay, MAC delay, packet transmission ratio, packet loss, and the internal collision rate. Fixed packet size of 1024 bytes is used for traffic of all four ACs in order to clearly observe the performance of each AC, specifically for throughput and delay results which are heavily influenced by the packet size.

7.3.1 Scenario 1: Results and Discussions

This section presents the results for the Scenario 1 described above. Figure 7.2 shows the results for the six performance metrics: throughput, end-to-end delay, MAC delay, transmission ratio, and internal collisions. Discussion on the results for each of the performance metrics follows next.

Throughput: Figure 7.2a presents the average throughput results for all four traffic streams, and clearly shows how differentiation is introduced through different ACs. While the throughput for low priority ACs starts to drop as the network becomes congested, the high priority ACs, AC_VI and AC_VO, keep receiving the constant shares of bandwidth. Throughput for the traffic streams of two lowest priority ACs, AC_BK and AC_BE, starts to drop as the total network load approaches to 1.6 and 1.9 Mbps, respectively. The early drop in throughput for the low priority ACs before network becomes saturated[1] is mainly due to the effect of starvation. As the AIFS time periods and Contention Window sizes for high priority AC are much smaller than those for lower priority ACs, most of the time the lower priority ACs are not able to decrement their backoff timers, as illustrated in Figure 4.4, and therefore are starved. The throughput of low priority ACs then keeps dropping gradually as the network starts to become saturated.

As there is only one station transmitting in this scenario, there is no external contention present, and thus the relative performances of ACs are only influenced by the internal contention among them, which makes it easier to understand the impact of EDCA parameters on the AC performances. External contention is taken into account in Scenario 2 in which total ten stations are transmitting, and thus the effectiveness of 802.11e QoS scheme will be more clearly observable.

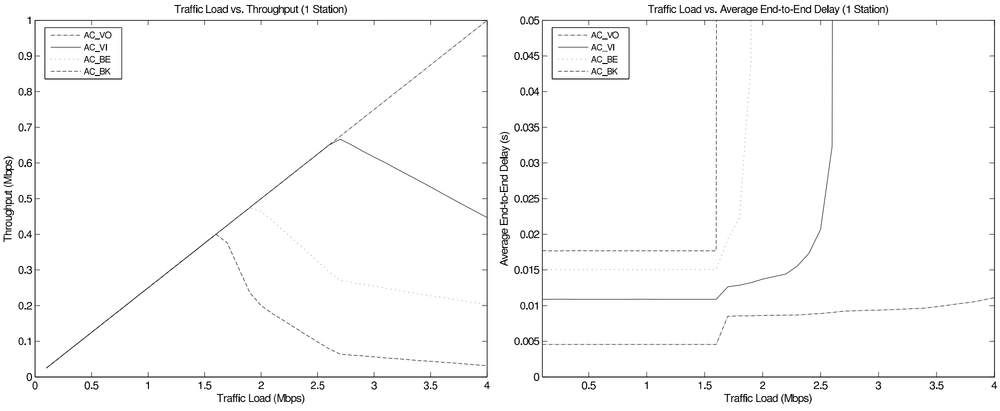
End-to-end delay: Figure 7.2b illustrates the average end-to-end delay faced by the traffic streams. From the figure, the effectiveness of service differentiation scheme is evident, seeing that while the delay for the low priority ACs, AC_BK and AC_BE, increases exponentially as the network starts to become congested, the delay for high priority ACs, specifically that of AC_VO, remains consistent.

The end-to-end delay is very critical for real-time and interactive multimedia applications such as teleconferencing and internet telephony, as discussed in Section 3.1 and in beginning of the chapter. In these applications the packets delivered with end-to-end delay of more than a few tens of milliseconds are often not acceptable. The performance of QoS scheme with respect to end-to-end delay is further analyzed in the next section where the simulation results for the more realistic scenario with ten stations are discussed.

MAC delay: Figure 7.2c shows the average MAC delay faced by the traffic streams. By comparing Figures 7.2b and 7.2c, we see that while both MAC and end-to-end delay figures are almost same initially, the effect of queueing delay becomes evident as the network becomes saturated, in that the end-to-end delay increases exponentially. It is because the frames start to face longer delays while residing in the AC transmit queues due to the fact that the high priority ACs get the maximum share of bandwidth, i.e., low priority ACs are starved, as discussed above.

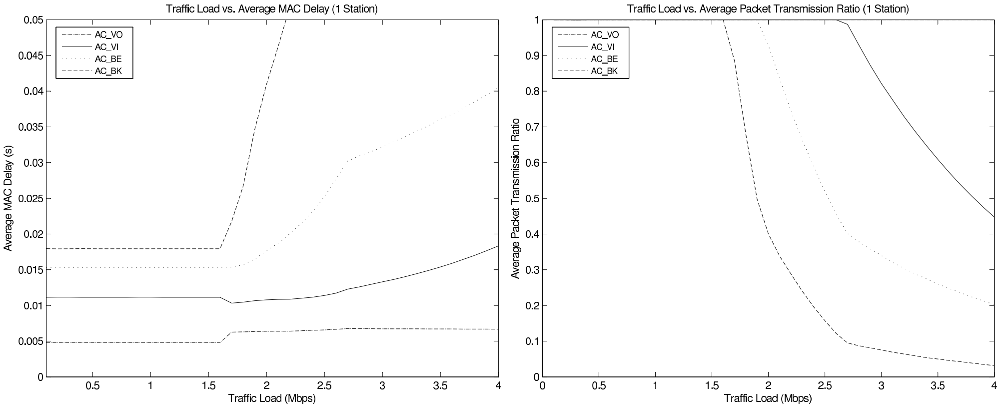
---

[1] The term *saturated* refers to the situation when the total data being transmitted is more than the maximum capacity/bandwidth of the network. Another term often used is *overloaded*.



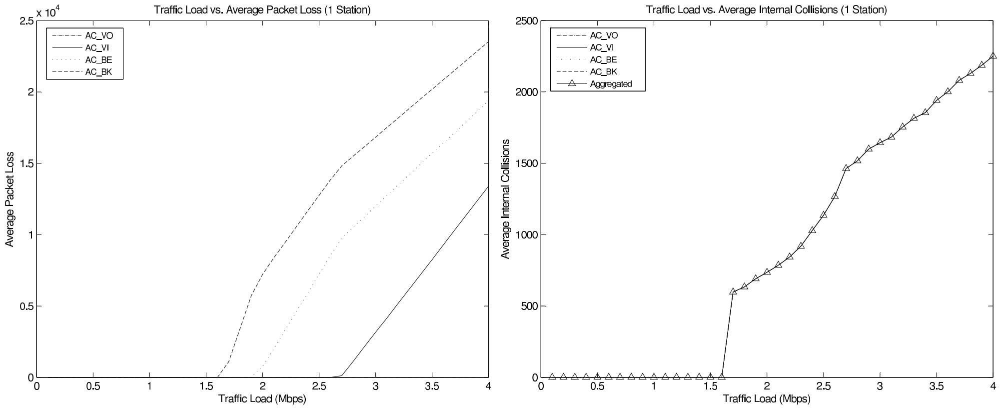
(a) Average throughput.

(b) Average end-to-end delay.



(c) Average MAC delay.

(d) Average packet transmission ratio.



(e) Average packet loss.

(f) Average internal collisions.

FIGURE 7.2: RESULTS FOR EVALUATION PART 1, SCENARIO 1.

Packet transmission ratio: Figure 7.2d shows the average packet transmission ratio for each of the ACs. The results for each AC traffic are in accordance with the results shown in Figure 7.2a, as the higher the bandwidth/throughput, the higher the packet transmission ratio and vice versa. As seen from the figure, the lower priority ACs, AC_BK and AC_BE, are only able to transmit below 30% and 10% percent of the total packets as the network becomes saturated.

Packet loss: Figure 7.2e shows the average packet loss for each of the ACs. As seen in the figure, there is no packet loss for the traffic stream of AC_VO, but other ACs suffer from high losses as the network approaches to its saturation point. This is in accordance with the results shown above. Comparing it to the results shown in Figure 7.2a, we see that higher the bandwidth share (throughput) for an AC, higher the number of packets transmitted for that AC, and thus lower the chances that the AC transmit queue becomes full and packets are lost. Comparing it to the results shown in Figure 7.2b, we see that both the average end-to-end delay and average packet loss for an AC starts to increase at the same point, indicating that as the queues start to fill toward their maximum sizes, the packets starts to face higher end-to-end delays and the number of lost packets starts to increase.

Internal collisions: Figure 7.2f illustrates the internal collisions faced by all four traffic streams, as well as the aggregated internal collisions. The drawback of smaller Contention Window sizes is very evident here in that the more than 90% of the total internal collisions occur between the higher priority ACs AC_VO and AC_VI, which have the minimum Contention Window sizes of 7 and 15 respectively, as presented in Table 7.1.

7.3.1.1 Service Differentiation based on CWmin and CWmax

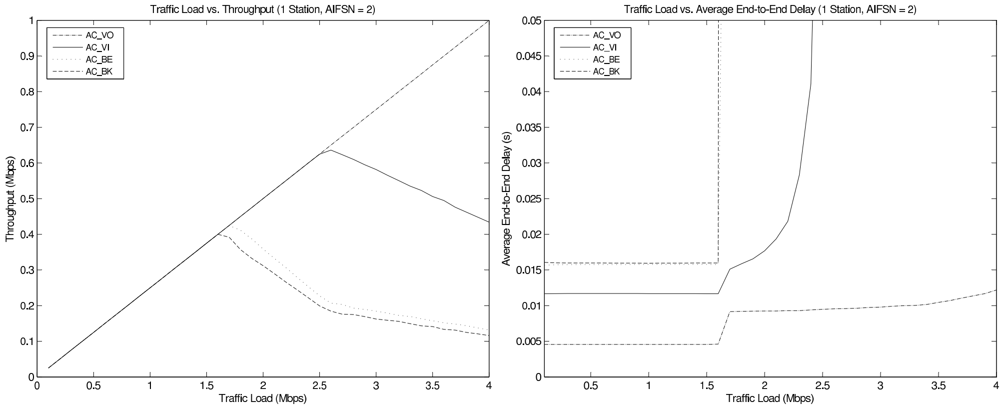
This section presents the results for the first of the three additional cases considered for the two scenarios. In the first case, same AIFSN value of 2 is used for all ACs. As described above, the aim is to observe the role of CWmin and CWmax in providing the service differentiation. Figure 7.3 presents the results for the same six performance metrics considered above for the original case.

Figure 7.3a shows the average throughput results for all four traffic streams. As it is seen, the CWmin and CWmax parameters are greatly effective in providing priorities to different ACs. The traffic streams for high priority ACs, AC_VO and AC_VI, receive larger share of bandwidth due to their smaller CWmin and CWmax values, 7 and 15, and, 15 and 31, respectively, as presented in Table 7.1. On the other hand, the throughput for the lower priority ACs, AC_BK and AC_BE, drops much earlier because of the larger CWmin and CWmax values they have.

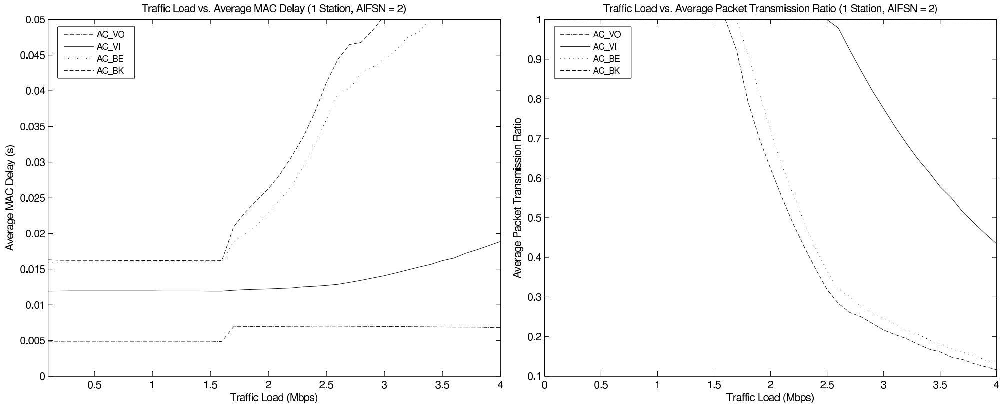
As it is seen, setting the AIFSN values to 2 for all ACs does not have much effect on the throughput performance of traffic streams for AC_VO and AC_VI, since the AIFSN value for both of the ACs is already 2, as shown in Table 7.1. Thus, by setting the value to 2 for all ACs, the aim is to eliminate the priority the high priority ACs receive in terms of AIFS.

Studying the results further, the relatively big margin between the throughput for AC_VI and those of AC_BK and AC_BE is thus due to the fact that the values of CWmin and CWmax for AC_BK and AC_BE are two times bigger than those for AC_VI.

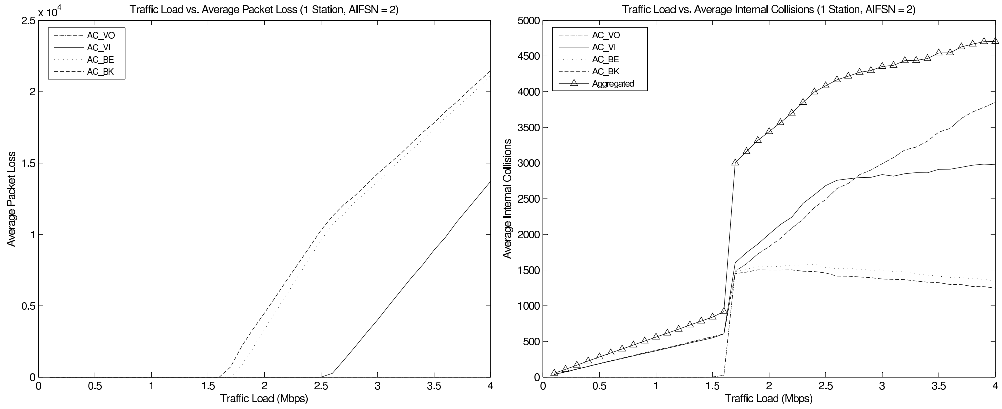
There is only a small difference between the throughput results for AC_BK and AC_BE, because both ACs have the same value of CWmin and CWmax, i.e., 31 and 1023, respectively, as shown in Table 7.1. One might wonder why AC_BE receives a little better throughput than AC_BK even though both have the same values of AIFSN,



(a) Average throughput. (b) Average end-to-end delay.



(c) Average MAC delay. (d) Average packet transmission ratio.



(e) Average packet loss. (f) Average internal collisions.

FIGURE 7.3: RESULTS FOR EVALUATION PART 1, SCENARIO 1, CWMIN/CWMAX BASED DIFFERENTIATION.

CWmin and CWmax. The reason is, as described in Section 4.3.2.1, that as two or more ACs collide internally, the higher priority AC is granted the access to medium and the lower priority AC backoffs. Thus, the difference in throughput results for AC_BK and AC_BE depicts the priority given to AC_BE in case of internal contention.

Comparing Figures 7.2a and 7.3a, it is observed that the throughput for AC_BK does not drop as drastically as it is in the original case. It is mainly due to the fact that the AIFSN value for AC_BK in the original case is much bigger, i.e., 7 compared to 2. The fact is more notable in the results presented in next sections for the Scenario 2.

Figures 7.3b, 7.3c, 7.3d and 7.3e show results for end-to-end delay, MAC delay, transmission ratio, and packet loss, respectively, and are in accordance to the results for throughput; the traffic streams for AC_VO and AC_VI are well served and achieve priority over traffic for lower priority ACs, and, the results for traffic streams for AC_BK and AC_BE are different with small margins due to their same CWmin and CWmax values.

Figure 7.3f shows the internal collisions as faced by all four traffic streams. As it is seen, AC_BK and AC_BE suffer from almost same number of collisions, which is in accordance to their same CWmin and CWmax values. The traffic for higher priority ACs, AC_VO and AC_VI, still suffers from higher collisions because of their small CWmin and CWmax values. An interesting observation is the influence of AIFSN values on the number of internal collisions. A comparison between Figures 7.2f and 7.3f shows that using same AIFSN values greatly increases the number of internal collisions, particularly for AC_BK and AC_BE, and the aggregated number of internal collisions are roughly doubled. Although the CWmin and CWmax values for AC_BK and AC_BE are same in both results and thus the probability of collisions should be same in terms of CWmin and CWmax, but using same AIFSN values in the second case drastically increased the number of internal collisions among AC_BK and AC_BE.

7.3.1.2 Service Differentiation based on AIFS

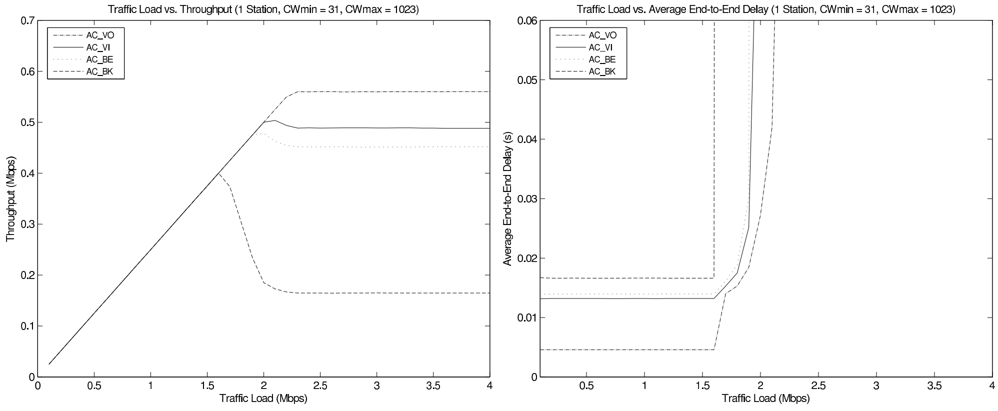
In order to study the effect of AIFS on service differentiation, this section presents the results for the same scenario as above, but such that this time the CWmin and CWmax values are set to fixed values 31 and 1023, respectively, for all four ACs. Figure 7.4 presents the results for the six performance metrics considered above.

Figure 7.4a illustrates the average throughput results for all four traffic streams. The stronger role of AIFSN in providing service differentiation is very evident from the results. From high to low priority AC, the relative differences in throughput results clearly reflect the AIFSN values of ACs, i.e., 2, 2, 3 and 7. The large difference between throughput of AC_BE and AC_BK compared to others shows the influence of using the AIFSN value as big as 7.

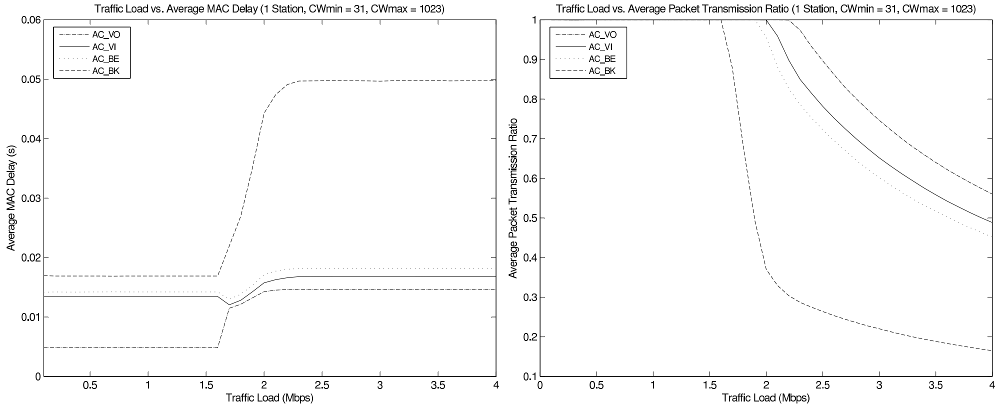
A comparison of Figure 7.4a with Figures 7.2a and 7.3a shows that the throughput for traffic stream of AC_BK does not degrade as much as for the other two cases. An interesting point here is that the values for AIFSN, CWmin and CWmax for AC_BK are same in the two cases shown in Figures 7.2a and 7.4a, but while its throughput continues to drop in the first case, it remained consistent in the second one. It shows that the low priority ACs are served better when only AIFS is used for service differentiation. But no conclusions can be drawn since, as specified earlier, these results are without the presence of any external contention, which may dramatically change the results.

From Figure 7.4f, the influence of using same and large CWmin and CWmax values on the number of internal collisions is evident seeing the dramatic drop in the internal collisions. Comparing it to Figures 7.2f and 7.3f further strengthens the conclusion.

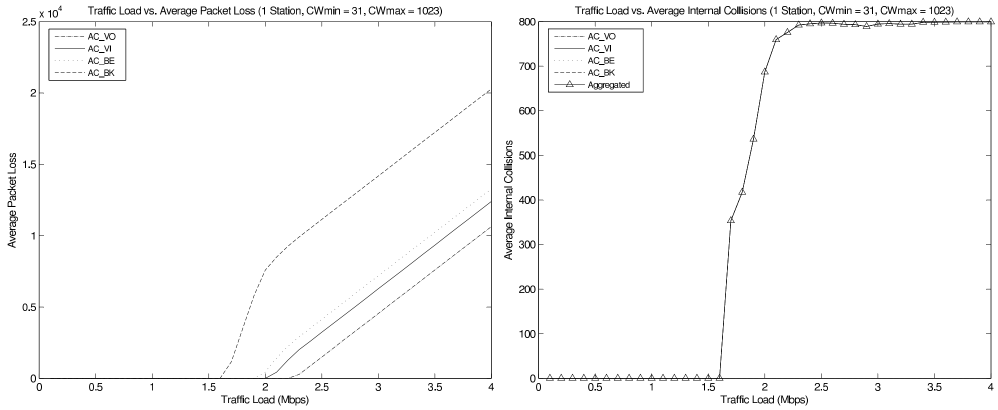
Comparing the role of CWmin and CWmax, and, AIFSN, in providing the service



(a) Average throughput. (b) Average end-to-end delay.



(c) Average MAC delay. (d) Average packet transmission ratio.



(e) Average packet loss. (f) Average internal collisions.

FIGURE 7.4: RESULTS FOR EVALUATION PART 1, SCENARIO 1, AIFS BASED DIFFERENTIATION.

differentiation, specifically in terms of throughput and internal collisions, shows that AIFS provides more efficient service differentiation compared to CWmin and CWmax.

Finally, in the last of the three additional cases, the effects of both EDCA parameters are switched off. Figure 7.5 presents the results.

As discussed before, the slightly better performance of a high priority AC compared to a low priority AC even when EDCA parameters for all four ACs have been set to same values is due to the fact that as two or more ACs contending for the medium internally collide, i.e., if the backoff timers for two or more ACs reach zero at the same time, then the highest priority AC among the colliding ACs is granted the access to the medium and others backoff. The procedure is explained with an example in Section 4.3.2.1.

Figure 7.5f again illustrates the strong influence of using the same AIFSN values on internal collisions, as discussed in Section 7.3.1.1. Comparing Figures 7.4f and 7.5f shows that using the same AIFSN value in addition to CWmin and CWmax for all ACs dramatically increased the internal collisions which were very low when only CWmin and CWmax values were kept same.

7.3.2 Scenario 2: Results and Discussions

While the first scenario serves mainly as the verification of service differentiation mechanism of IEEE 802.11e and our EDCA design as well, this section shows the results for a more realistic scenario in that the number of stations is now increased to ten. Two more performance metrics are taken into account, namely packet drops and external collisions. These metrics were meaningless for the first scenario since there was no external contention present and thus collisions and packet drops were not possible.

The same packet size of 1024 bytes is used for the traffic streams of all four ACs in order to closely observe the effectiveness of the EDCA parameters in providing service differentiation. More realistic traffic patterns with packet sizes and bit rates used by the real world applications are considered in the second part of the evaluation, presented in Section 7.4.

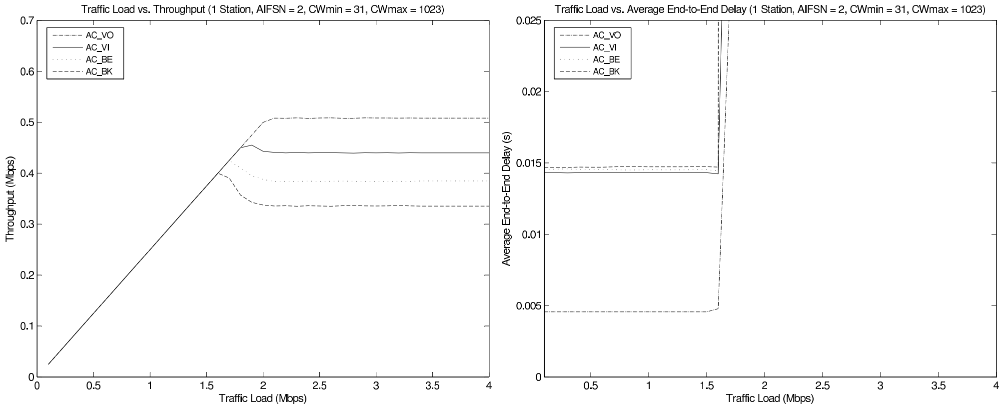
Figures 7.6 and 7.7 present the results for the eight performance metrics. Discussions on the results for each of the performance metrics follow next.

Throughput: Figure 7.6a presents the average throughput results for all four ACs, and, like Figure 7.2a, clearly shows the effectiveness of QoS scheme realized by introducing different ACs for data traffic of different priorities.

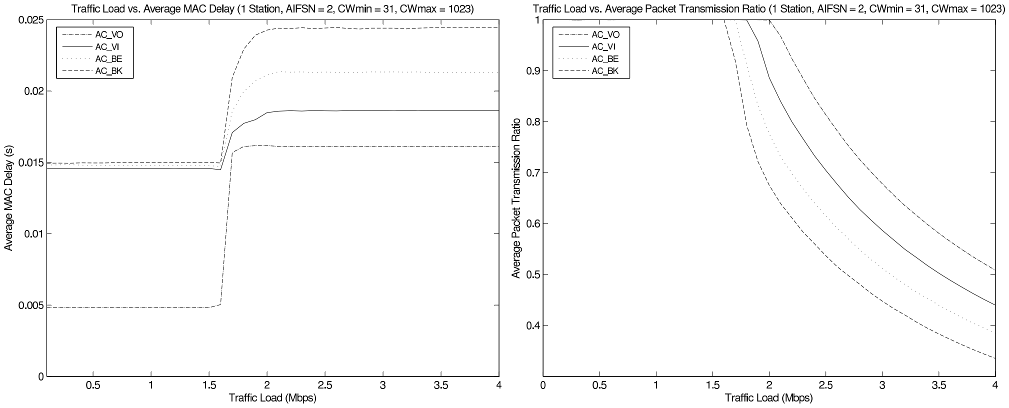
The throughput results for the traffic streams of high priority ACs, AC_VO and AC_VI, show that they receive constant share of bandwidth even in the saturation conditions. Throughput for AC_VI traffic stream drops when network is loaded with 2.4 Mbps of data, whereas, throughput for AC_VO traffic starts to decrease near the 3 Mbps point, although does not drop as quickly as for the AC_VI traffic.

The throughput results for high priority traffic show that the scheme is able to support multimedia applications, especially the bandwidth-sensitive inelastic applications, with a reasonable quality of service. As discussed in Section 3.1, bandwidth-sensitive, inelastic applications require constant share of bandwidth for successful operation.

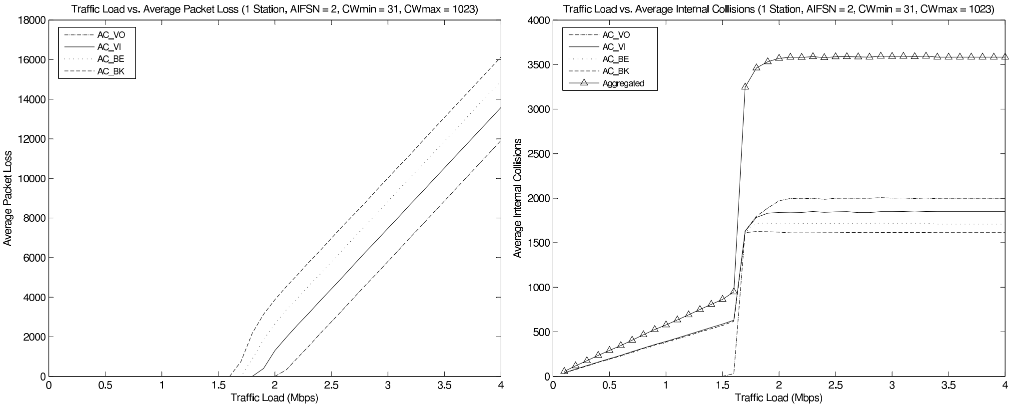
Observing the performance of traffic streams for low priority ACs, it is straightforward to conclude that they receive very small bandwidth as the network becomes saturated. The traffic stream for AC_BK becomes starved mainly due to the fact that it experiences both levels of contention, namely, internal and external. A comparison of Figures 7.2a and 7.6a further explains the influence of external collision on the starvation of traffic stream for AC_BK. But still the influence of internal contention on the starvation of AC_BK is greater compared to that of external contention, mainly because AC_BK always has to compete internally with three high priority ACs which



(a) Average throughput. (b) Average end-to-end delay.



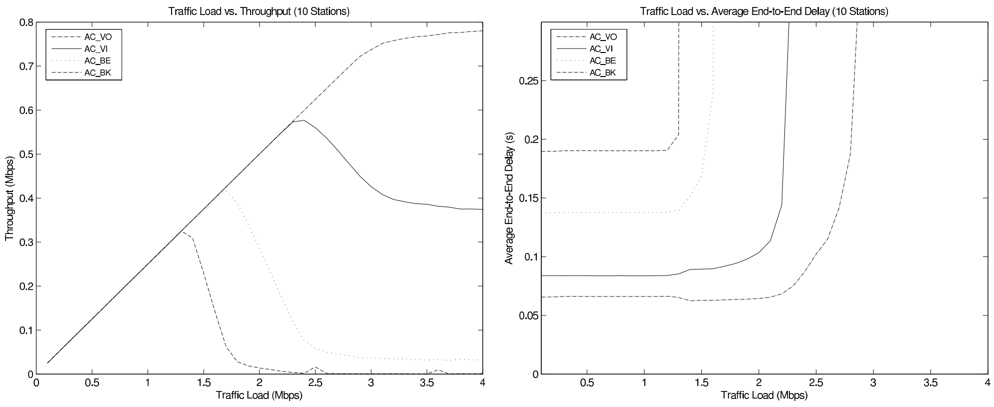
(c) Average MAC delay. (d) Average packet transmission ratio.



(e) Average packet loss. (f) Average internal collisions.

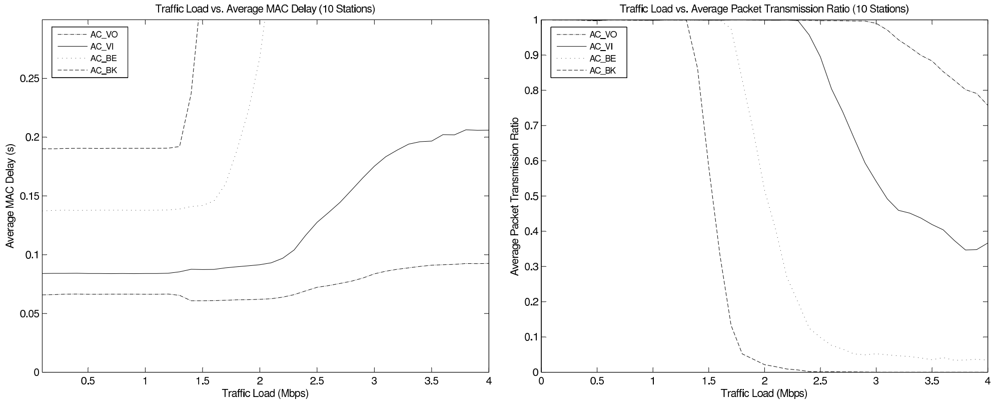
FIGURE 7.5: RESULTS FOR EVALUATION PART 1, SCENARIO 1, NO AIFS AND CWMIN/CWMAX DIFFERENTIATION.

act just like three contending stations. Any further introduction of contending stations thus makes the situation even worse and performance decreases drastically.



(a) Average throughput.

(b) Average end-to-end delay.



(c) Average MAC delay.

(d) Average packet transmission ratio.

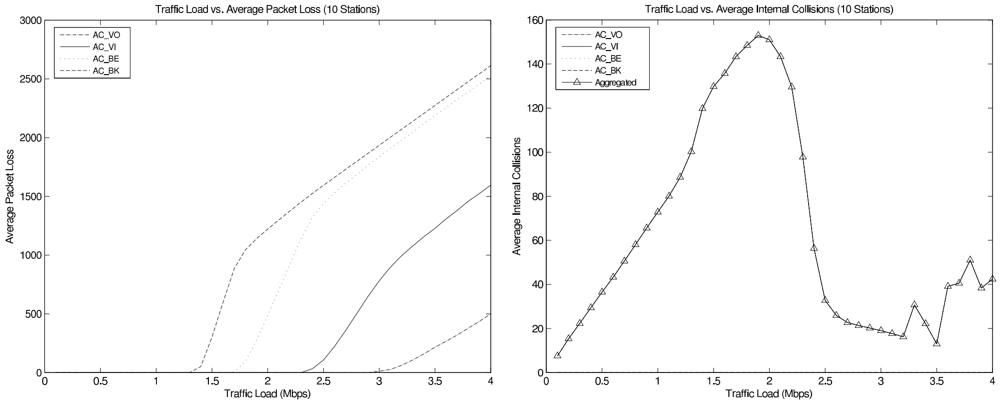
FIGURE 7.6: RESULTS FOR EVALUATION PART 1, SCENARIO 2.

The influence of external contention seen from Figures 7.2a and 7.6a is also due to the fact that the contending stations are transmitting the traffic of higher priority ACs. The starvation condition would be less worse if the contending stations transmit only traffic of low priority ACs.

The observations regarding starvation of low priority ACs are also presented in simulation studies in [20] and [18].

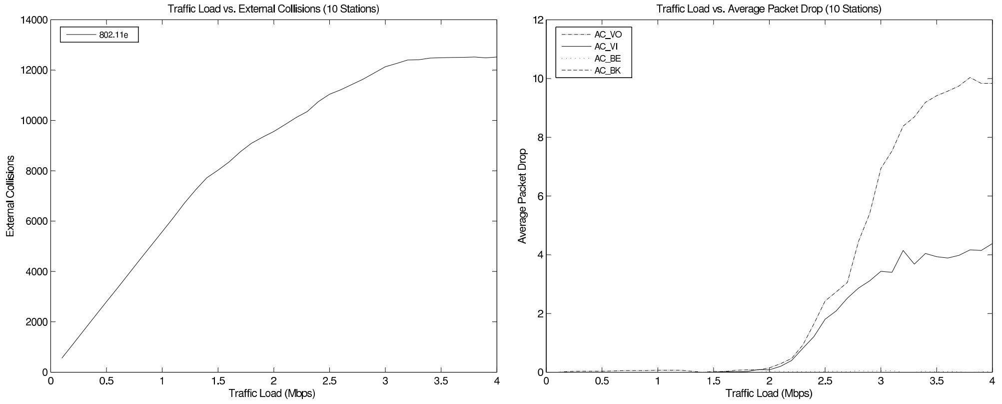
End-to-end delay: Figure 7.6b shows the average end-to-end delay faced by the four traffic streams. As discussed in Section 3.1 and in beginning of this chapter, end-to-end delay is the most critical parameter for interactive and real-time multimedia applications. These applications are highly sensitive to in time delivery of data and frames with delay of more than a few tens of milliseconds are often useless. Thus the performance of high priority ACs, AC_VO and AC_VI, which are mainly designed for multimedia applications, is of high importance.

As it is seen from Figure 7.6b so far, 802.11e successfully supports QoS for the traffic streams of AC_VO and AC_VI, in that they maintain the low delay figures up until the network becomes overloaded. The delay for the AC_VI is below 90 ms and that for AC_VO is below 70 ms which should be acceptable for most of the multimedia video



(a) Average packet loss.

(b) Average internal collisions.



(c) External collisions.

(d) Average packet drop.

FIGURE 7.7: RESULTS FOR EVALUATION PART 1, SCENARIO 2.

and voice applications.

A point to note here is that the actual delay for voice should be much less because for this scenario the packet size for AC_VO is 1024 bytes which is too big for most of the voice applications used today. Larger the size of packet, greater the delay and vice versa. Evaluation with some more realistic scenarios is presented in the second part in Section 7.4, in which traffic characteristics used by the real world applications are considered.

The relatively earlier increase in the delay for AC_VI compared to AC_VO is in accordance to the traffic types they are designed for, i.e., video streaming applications can tolerate some delays but voice applications apply more stringent constraints on delay. The delay for the low priority ACs, AC_BK and AC_BE, increases exponentially as the network starts to become congested. But as these ACs are mainly designed for the delay-tolerant applications, certain amount of delay does not degrade their performance beyond an acceptable limit. As discusses in Section 3.1, data oriented applications such as email, file transfer, web, and instant messaging are generally delay-tolerant but enforce high constraints on reliable data transfer, i.e., they cannot tolerate packet losses.

As discussed in Section 3.1, certain real-time multimedia applications are very sensitive to delays and packets with delays beyond an acceptable limit are useless at the receiver side, because the data is required to be decoded at the receiver side with the same rate it was encoded at the sender side. For these applications sometimes only studying the average end-to-end delay is not enough since the average might be rather low even if a large number of packets delivered with unacceptable delays. Thus, another interesting study is to investigate how the delays of packets are distributed, i.e., how many packets suffered from delays beyond the acceptable limit, as studied in [20].

Another performance metric, referred to as jitter, is defined as the variation in delays, and of high importance particularly in the multimedia applications that require data delivery with constant bit rates, as discussed in Section 3.1. An example is video and teleconferencing applications in which the received data must be decoded at the same bit rate it is encoded at the sender side. Any variations in the timing of receiving the data therefore results in problems in decoding. Most of the multimedia applications are very sensitive to jitter and in order to minimize its effects, buffering techniques are utilized such that certain number of packets are buffered at the receiver side before the decoding process stars [12, p. 46]. Generally, the larger the amount of jitter, the larger the buffers required to mitigate its effects, which results in additional delays before the output starts (e.g. video streaming). Investigation studies about performance of EDCA with respect to jitter can be found in [26] and [18].

MAC delay: Figure 7.6c shows the average MAC delay for each of the traffic streams. The improvement in delay performance of high priority ACs, AC_VI and AC_VO, over the lower priority ACs is clearly observable in that they maintain consistent delay figures in the highly loaded situations, while the low priority ACs suffer from large delays.

Packet transmission ratio: Figure 7.6d presents the average packet transmission ratio results for the four ACs. As seen, the traffic stream for AC_BK is completely starved the point the network is overloaded, such that the packet transmission ratio goes down to zero indicating that it is not able to transmit any packets at all. The ratio for AC_BE does not drop as low as that for AC_BK although it is also able to transmit only 5% of total packets after the network becomes saturated. The difference between the performance of the two ACs is mainly because of the large difference in their AIFSN values as seen in Table 7.1. AC_BK has to wait three additional slots before it can access the medium or start decrementing backoff timer.

Packet loss: Figure 7.7a shows the average packet loss results for the four traffic streams. The results are in accordance with the throughput results in Figure 7.6a; the higher the throughput a traffic stream receives, the lower the packet loss it suffers from. As described in beginning of the chapter, a packet is lost if by the time it arrives at MAC layer, the corresponding AC transmit queue is already full.

As discussed in Section 3.1, the data oriented applications are generally more sensitive to packet losses than the multimedia applications, and are relatively less sensitive to delays too. A possible strategy therefore would be to use larger queue sizes for the ACs designed for data applications, i.e., AC_BK and AC_BE, compared to the ACs for multimedia applications, i.e., AC_VO and AC_VI. It will improve the packet loss ratio for the data oriented applications. Although it will result in increased delays too, but certain amounts of delays are negligible for data applications such as email, file transfer, web, etc.

Internal collisions: Figure 7.7b shows the result for internal collisions faced by all four traffic streams, as well as the aggregated internal collisions. As it is seen, the internal collisions among the low priority ACs are as few as none, while those for higher priority ACs are very high, which shows the drawback of small Contention Window sizes as discussed in Section 7.3.1, and further in subsections 7.3.1.1 and 7.3.1.2.

Collisons: Figure 7.7c shows the total number of collisions (external) occurred among the stations. As it is observed, the number is very high, which is mainly due the fact that most of the collisions occurred while transmitting AC_VO or AC_VI packets, because of their small Contention Window sizes. Later in this section, the individual effect of AIFS and CWmin and CWmax on the collision rate is studied. A comparison study of collisions in 802.11 and 802.11e is also presented in the subsequent sections.

Packet drops: Figure 7.7d shows the average packet drops for all ACs. This performance metric was not considered for Scenario 1 because Scenario 1 does not consider external contention, i.e., there is only one station transmitting and therefore it does not have to contend with other stations. Revising from Section 2.2.1, a packet is dropped after the number of retransmissions reaches to the retry limit. The higher packet drops for traffic streams for AC_VO and AC_VI are due the fact that the collision rate (or rate of unsuccessful transmissions) for these ACs is much higher compared to the lower ACs, because of their small Contention Window sizes. It is in accordance to the number of collisions shown in 7.7c.

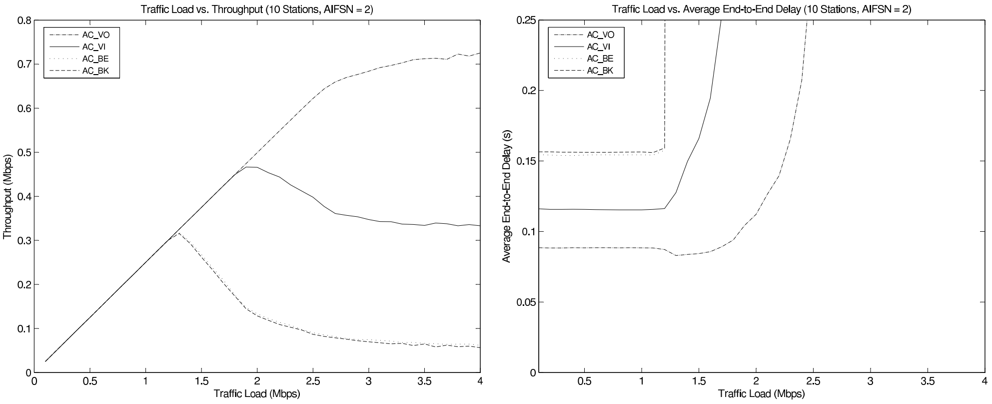
The higher number of packet drops for AC_VO and AC_VI does not question the effectiveness of the service differentiation scheme, since, as discussed in Section 3.1, the multimedia applications these ACs are design for, specifically the video streaming applications, are generally loss-tolerant, i.e., they require strict bandwidth and delay guarantees but can tolerate certain amounts of packet losses. Packet loss in these applications results in slightly reduced quality of output (e.g., jerks in video or voice output), but can be neglected to some extent. Additionally, as the data oriented applications are very sensitive to packet losses, the results for AC_BK and AC_BE show how effective the service differentiation mechanism is in providing QoS supports to different ACs according to the traffic/applications they are designed for.

The packet drop rate as well as the delay is dependent on the number of retransmissions. The higher the number of retransmissions, the larger the delay and the packet drop rate. Furthermore, the maximum number of retransmissions is bounded by the retry limits. Thus the retry limit values for individual ACs should be set such that the ACs for real-time multimedia applications have smaller values and the ACs for data oriented applications have larger values. Large values of retry limits result in higher delays but smaller number of packet drops, which is acceptable for the data oriented applications since the main concern is the reliable data transfer, and, small values of retry limits result in higher number of packet drops but lower delays, which is desirable for real-time multimedia applications.

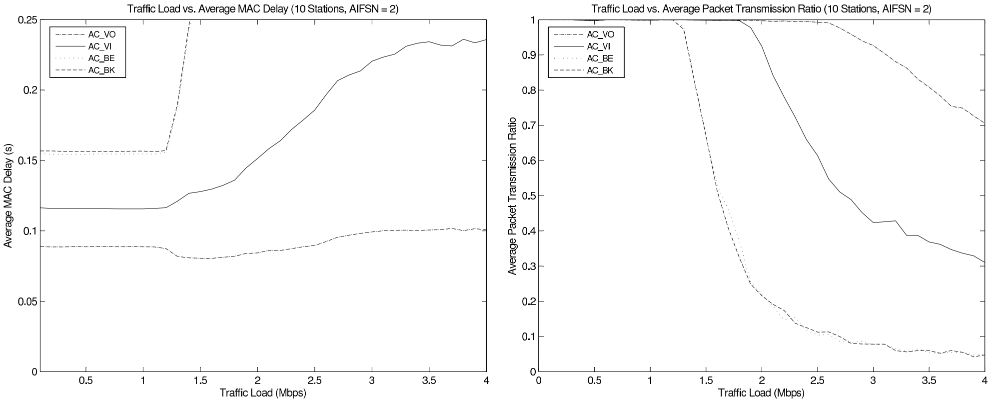
Interestingly, the standard does not specify any values for retry limits for ACs. It has been left up to the user to choose retry limit values for different ACs. In our implementation, the user is allowed to chose different retry limit values for different ACs, as seen in Section 6.4.3.

7.3.2.1 Service Differentiation based on CWmin and CWmax

This section presents the results for the same scenario as above, but with AIFSN values set to 2 for all ACs. The aim is to observe the impact of CWmin and CWmax parameters in realizing service differentiation for different ACs. Figures 7.8 and 7.9 show the results for the same eight performance metrics considered before.



(a) Average throughput. (b) Average end-to-end delay.

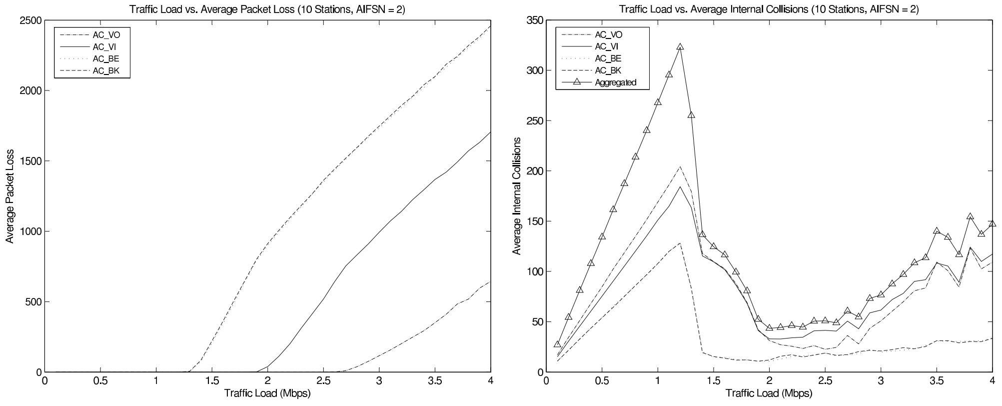


(c) Average MAC delay. (d) Average packet transmission ratio.

Figure 7.8: Results for Evaluation Part 1, Scenario 2, CWmin/CWmax based differentiation.

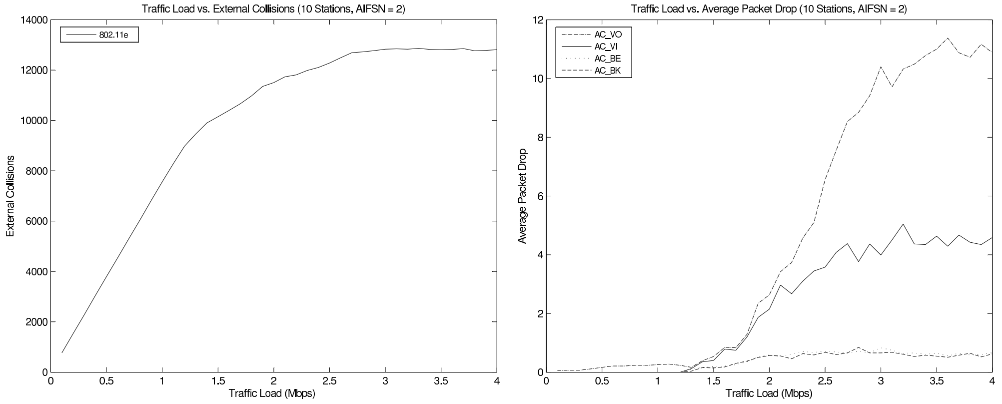
Figure 7.8a shows the average throughput results for all four traffic streams. As it is seen, the traffic streams for AC_BK and AC_BE receive equal amounts of bandwidth resources due to their same CWmin and CWmax values, i.e., 31 and 1023, respectively. The small difference is due to the fact, as discussed in Section 7.3.1.1, that a higher priority ACs is always given priority over a lower priority AC if the two accidently tries to access the medium at the same time. The relatively smaller difference between throughput results of AC_BK and AC_BE comparing to the results for CWmin and CWmax based differentiation in Scenario 1 in Figure 7.3a explains that the impact of internal contention reduces in comparison to the external contention with the increasing number of contending stations.

A significant observation is that the performance of AC_BK is improved signifi-



(a) Average packet loss.

(b) Average internal collisions.



(c) External collisions.

(d) Average packet drop.

FIGURE 7.9: RESULTS FOR EVALUATION PART 1, SCENARIO 2, CWMIN/CWMAX BASED DIFFERENTIATION.

cantly; it is not starved as terribly as in the earlier two cases. The improvement confirms that the primary reason of starvation of AC_BK is its large AIFSN value of 7. The heavy influence of AIFSN on the performance of AC is described in detail with the help of Figure 4.4 in Section 4.3.3.

From the results, it is concluded that the starvation of low priority ACs can be avoided by using the same AIFSN values for all ACs and only varying the CWmin and CWmax, a behavior also investigated in the simulation studies in [20] and [18].

An interesting study would be to investigate the performance of AC_BK with an AIFSN value smaller than 7 (but still greater than that of AC_BE). AIFSN of 5 could be a good choice.

Figures 7.8b and 7.8c present the results for average end-to-end delay and MAC delay, respectively, and show that while using the same AIFSN for all ACs decreases the delay for AC_BK, it results in negative effect on the delays of higher priority ACs. Specifically, the delays for AC_VO and AC_VI increased with approximately 20 ms and the delay for AC_BE increased with approximately 10 ms, even though the former two still have the AIFSN of 2 and the latter one now has a AIFSN one slot less than its original AIFSN of 3. The reason, basically, is that eliminating the AIFSN prioritization decreases the margin of priority a high priority AC has on a lower priority AC, and thus all ACs contend for the medium with lower differentiation among them.

Figure 7.8d shows the average packet transmission ratio results for all four ACs. The results are in accordance to the throughput results in Figure 7.8a, in that the traffic stream for AC_BK is not starved completely and is able to transmit 6-8% of packets in the saturated condition.

Figure 7.9b shows the internal collisions as faced by all four ACs. Comparing it to Figure 7.7b further strengthens the observation that using same AIFSN for all ACs greatly increases the number of internal collisions, particularly among AC_BK and AC_BE. Moreover, the drop in results after the load is reached to 1.2 Mbps also shows that the impact of internal contention decreases with the increase in external contention.

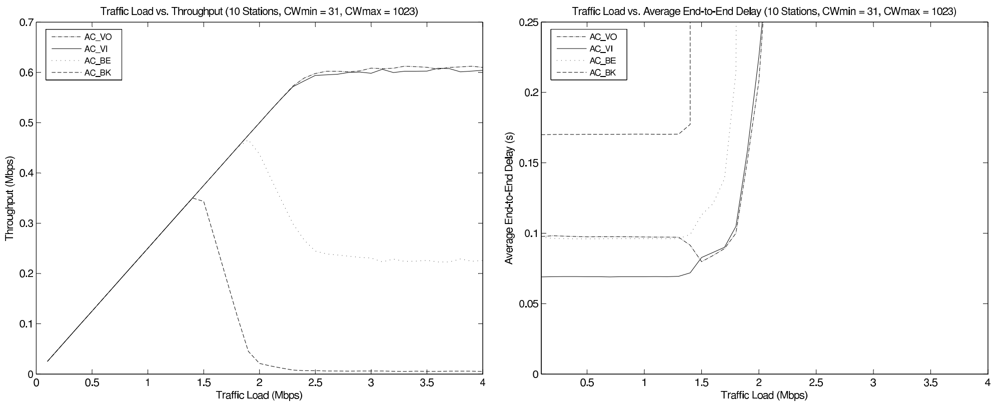
Figure 7.9c shows the total external collisions occurred among the stations. Comparing it to Figure 7.7c shows that using same AIFSN does not have much influence on total number of external collisions. The reason, as described earlier, is that the external collisions are mainly due to the high priority ACs, AC_VI and AC_VO, which has small CWmin and CWmax values, and since there is no change in the AIFSN, CWmin and CWmax values for these ACs in this case of the scenario, the number of external collisions does not decrease. The earlier increase in the external collisions compared to Figure 7.7c is due the fact that now lower priority ACs, AC_BK and AC_BE, are also contributing towards the external collisions since they are also receiving considerable share of bandwidth.

Figure 7.9d shows the average packet drop results for the four traffic streams. Traffic streams for AC_VO and AC_VI still face high packet drops similar to the results in Figure 7.7d. It is for the obvious reason that CWmin and CWmax values are same in both cases. The results differ in that this time traffic streams for AC_BK and AC_BE also face packet drops. This is in accordance to the throughput results in Figure 7.8a comparing to Figure 7.6a, which shows that this time AC_BK and AC_BE are receiving higher throughput too, concluding that with increase in throughput, more packets are sent and thus more collisions occur resulting in higher packet drops.

7.3.2.2 Service Differentiation based on AIFS

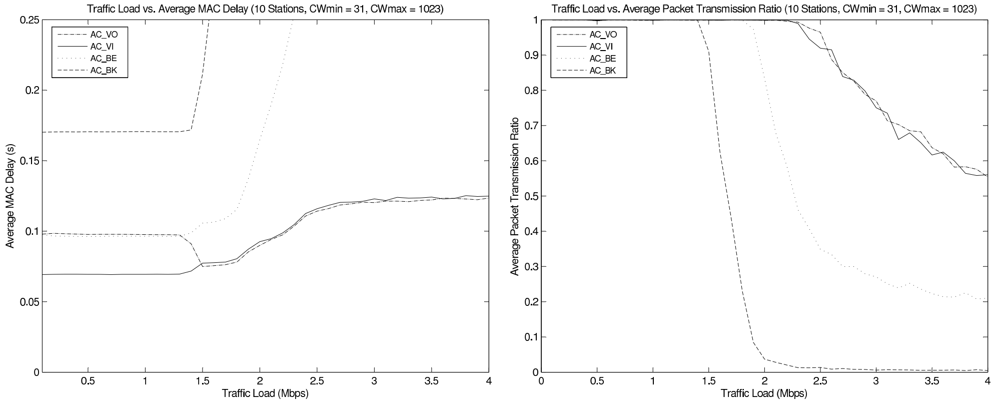
This section presents the results for the same scenario as above, but such that this time the CWmin and CWmax values are set to fixed values 31 and 1023, respectively. As before, the aim is to observe the role of AIFS in providing service differentiation. Figures 7.10 and 7.11 present the results for the same eight performance metrics we considered above for the original case.

Figure 7.10a shows the average throughput results for the four ACs. Comparing it to the results for CWmin and CWmax based differentiation in Figure 7.8a shows that AIFS provides superior and more reliable service differentiation. The reasons are: (1) The throughput received by an AC is directly in accordance to its AIFSN value. AC_VI and AC_VO receive almost equal share of bandwidth because of their same AIFSN values of 2, and AC_BE and AC_BK receive throughput according to their AIFS values of 3 and 7. (2) The large margin between throughput results of AC_VI and AC_BE shows that even AIFS difference of a single slot time is very effective in realizing service differentiation, a fact also studied in [25]. (3) AIFS provides consistent and more reliable differentiation independent of the different network conditions. It is in contrast to the CWmin and CWmax based service differentiation which becomes less effective with increase in network congestion. It is because as the traffic/stations increases, number



(a) Average throughput.

(b) Average end-to-end delay.



(c) Average MAC delay.

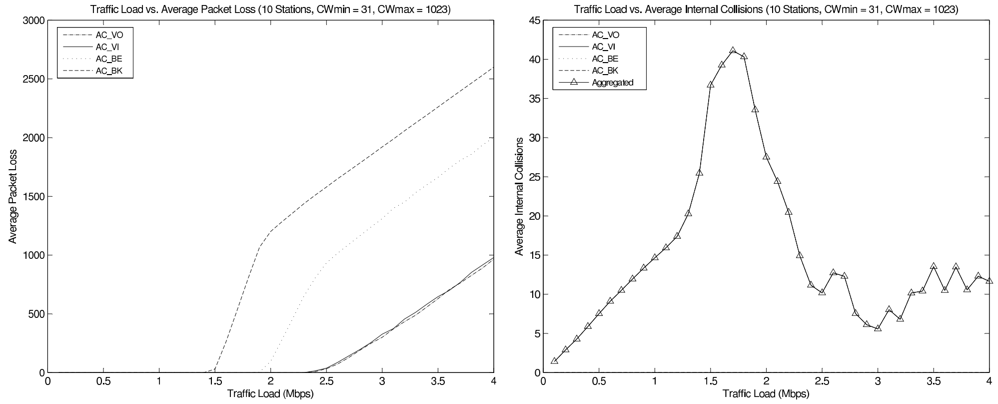
(d) Average packet transmission ratio.

Figure 7.10: Results for Evaluation Part 1, Scenario 2, AIFS based differentiation.

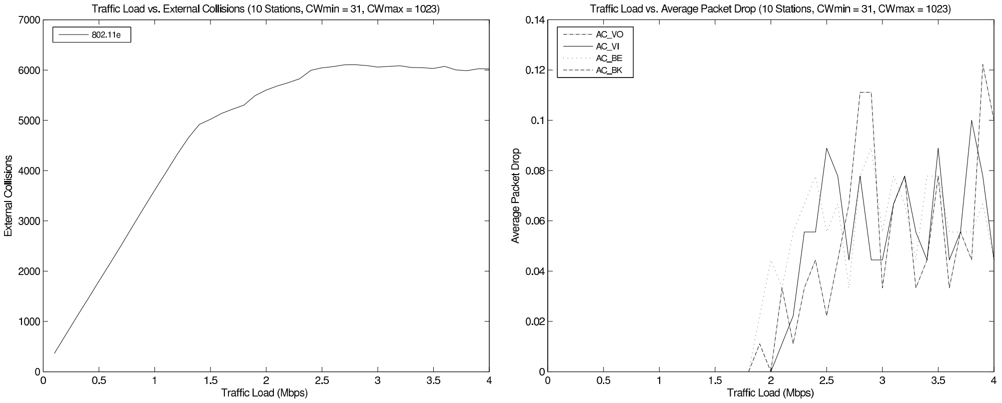
of collisions/retransmissions also increases because of the small CWmin and CWmax values, and, the average Contention Window size also increases resulting in reduced impact of CWmin on service differentiation.

An important fact is that under lightly loaded network conditions, the role of CWmax in introducing service differentiation is less significant compared to that of CWmin. It is because as there are less collisions/retransmissions, most of the times the Contention Window size does not approach to CWmax. The role of CWmax becomes more and more stronger with the increase in network congestion, since, as described in Section 2.2.1, Contention Window size is doubled after each unsuccessful transmission until it reaches to CWmax. Moreover, once set to CWmax, it stays there until the frame is successfully transmitted. It also indicates that in congested network situations, larger CWmax values for AC_VO and AC_VI could remarkably reduce the collision rate, which is the primary reason of lowered performance of CWmin and CWmax based differentiation.

Comparison between Figures 7.10a and 7.8a shows that the throughput results for AC_VO and AC_VI traffic in AIFS based service differentiation are slightly lower, and the traffic for AC_BK is almost completely starved. The reason of starvation, as discussed in Sections 7.3.1.1, 7.3.1.2 and 7.3.2, is that AC_BK has AIFSN value of 7



(a) Average packet loss. (b) Average internal collisions.



(c) External collisions. (d) Average packet drop.

Figure 7.11: Results for Evaluation Part 1, Scenario 2, AIFS based differentiation.

which is too big in comparison to the AIFSN values of other ACs. It shows that AIFS based service differentiation is highly sensitive to the AIFSN values and too big AIFSN values for lower priority ACs lead them to starvation. The results helps to conclude that in lightly loaded situations, CWmin and CWmax based differentiation serves both the high and low priority ACs slightly better in terms of throughput compared to the AIFS based differentiation.

A comparison of results for the normal scheme (when both AIFS and CWmin and CWmax are used for differentiation) and the AIFS based differentiation, presented in Figures 7.6a and 7.10a respectively, leads to the interesting observation that the latter significantly increased throughput for AC_BE, although it has same AIFSN, CWmin and CWmax values in both cases. The same thing was also observed from Figures 7.2a and 7.4a. The obvious reason is that the bandwidth is distributed among all four traffic streams, and thus the more one traffic stream receives the less the other traffic streams receive. In other words, AC_BE received higher throughput because high priority ACs, AC_VI and AC_VO, received lower throughput. It also shows that high priority ACs are provided differentiation on the expense of lowered performance of lower priority ACs.

Figure 7.11b shows the internal collisions as faced by all four ACs. Comparing it to Figure 7.9b strengthens the claim that using different values of AIFSN drastically reduces the number of internal collisions. The fact was also observed in Section 7.3.1 in Figures 7.3f and 7.4f.

Figure 7.11c shows the total external collisions occurred among the stations. Comparing it to Figures 7.7c and 7.9c shows that using same and larger CWmin and CWmax values for all ACs, as well as different AIFSN values, have big influence on external collisions, which have reduced to one half of those in earlier cases.

Together from the results of internal and external collisions so far, it is concluded that the larger the difference in AIFSN values, the lower the number of collisions, both internal and external, and vice versa. The effect is also studied in [19].

Figure 7.11d shows dramatic decrease in average packet drops compared to the results in Figure 7.9d; it decreased from average (approximately) 12 packets to as low as average 0.12 packets. The results are in accordance to the external collision results in Figure 7.11c, and show the huge influence of CWmin and CWmax values on the average number of packet drops.

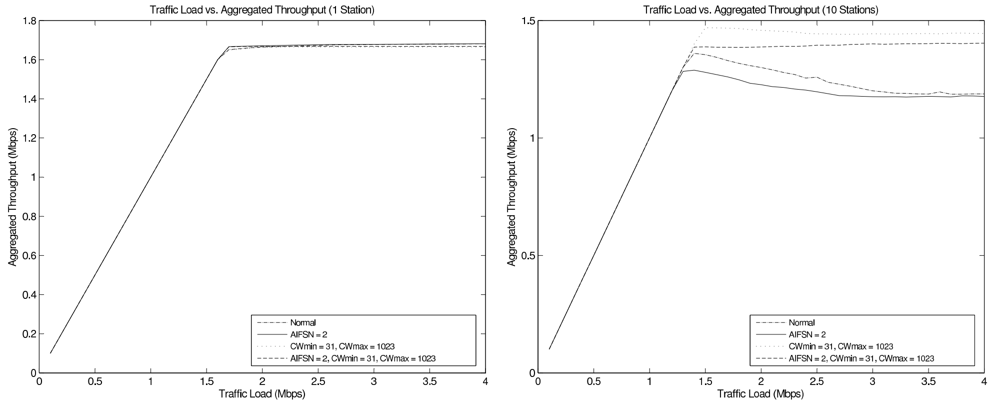
Finally, in the last of the three additional cases, both EDCA parameters are switched off. The results can be seen in Figures B.1 and B.2 presented in Appendix B.

7.3.3 Throughput Comparisons

After studying the effectiveness of service differentiation scheme of EDCA as well as the roles of individual EDCA parameters, in this section, an overall comparative study is presented.

7.3.3.1 EDCA Parameter Aggregated Throughput Comparison

In the first comparison study, aggregated AC throughput results are compared for the four cases studied in Scenarios 1 and 2 in Sections 7.3.1 and 7.3.2, respectively, i.e., the EDCA parameters are *switched off* one by one. For each of the four cases, the aggregated throughput is calculated by adding the throughput values corresponding to all four traffic streams. Figure 7.12 shows the comparison results.



(a) Aggregated throughput comparison with 1 station.

(b) Aggregated throughput comparison with 10 stations.

FIGURE 7.12: AGGREGATED THROUGHPUT COMPARISON OF EDCA PARAMETERS.

Results and Discussions: Figure 7.12a shows the results for the 1 station case. As it is seen, there is not much difference in the aggregated throughput in absence of external contention. Figure 7.12b presents the results for the case of 10 stations. It is evident from the results that the aggregated throughput for AIFS based service differentiation is higher compared to the normal case when both AIFS and CWmin and CWmax are used for this purpose. On the other hand, CWmin and CWmax based service differentiation receives the lowest aggregated throughput, which, as thoroughly discussed in Sections 7.3.1 and 7.3.2, is mainly because of the small CWmin and CWmax values for high priority ACs resulting in high collision rate. The observation is also presented in studies in [20] and [25]. Thus considering the results together with the results presented in Sections 7.3.1 and 7.3.2, and specifically in Section 7.3.2.2, it is further concluded that AIFS based differentiation enables greater aggregated throughput compared to CWmin and CWmax based service differentiation which suffers from reduced throughput because of the high collision rate.

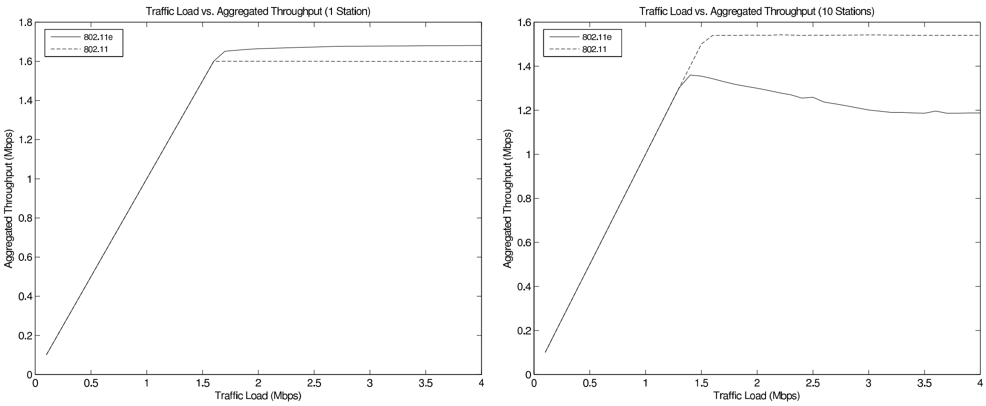
Another notable observation is that the normal scheme, i.e., using both AIFS and CWmin and CWmax for service differentiation, receives lower throughput compared to the AIFS based and CWmin and CWmax based differentiation schemes. It further concludes that service differentiation is provided on the expense of lower overall throughput.

Seeing the remarkably superior performance of AIFS in realizing service differentiation together with the fact that CWmin and CWmax based differentiation is highly suffered because of the high collision rate as the network becomes congested, one might wonder why not only AIFS is chosen for service differentiation by the IEEE Working Group. One reason could be that as more likely initially not all 802.11 networks could be upgraded to 802.11e at once, and instead 802.11e and 802.11 networks would coexist (802.11e provides backward compatibility, see Section 4.4), CWmin and CWmax based service differentiation would ensure 802.11e AC_VO and AC_VI traffic priority over 802.11 stations. It is because as the AIFS for AC_VO and AC_VI is equal to DIFS in 802.11, the service differentiation for these ACs would be achieved solely on the basis of their smaller CWmin and CWmax values (7 and 15, and, 15 and 31, respectively) compared to 31 and 1023 in 802.11.

7.3.3.2 802.11 and 802.11e Aggregated Throughput Comparison

In second comparison study, aggregated throughput results of 802.11 DCF and 802.11e EDCA are compared, for both cases of 1 and 10 stations. The traffic pattern is same as used for the two scenarios above, i.e., each station is transmitting traffic corresponding to all four traffic types, packet size for all traffic types is 1024 bytes, and, the total load is increased by increasing the load corresponding to each traffic stream. The aggregated throughput is the result of adding the throughput values for all four traffic streams. The comparison results are presented in Figure 7.13.

Results and Discussions: As seen in Figure 7.13a, compared to 802.11 DCF, 802.11e EDCA provides slightly better results in terms of throughput. As the results are for the single station case, no consideration is given to the external contention. Figure 7.12b shows the comparison results for 10 stations. It presents very important fact that the EDCA receives lower throughput in comparison to DCF, which helps to conclude that EDCA provides service differentiation on the expense of lower throughput, as discussed in the first comparison above, and also presented in [26]. Another important observation is that while DCF maintains consistent throughput after the network becomes saturated, the throughput for EDCA keeps dropping gradually with the increase in load. It is



(a) Aggregated throughput comparison with 1 station.

(b) Aggregated throughput comparison with 10 stations.

FIGURE 7.13: AGGREGATED THROUGHPUT COMPARISON OF 802.11 AND 802.11E.

mainly because of the small CWmin and CWmax sizes of high priority ACs; with the increase in traffic load, the collision/retransmission rate increases resulting in wastage of bandwidth resources.

It should be noted that the goal of EDCA is to provide service differentiation, in which it is very successful as it is evident from the results presented in Sections 7.3.1 and 7.3.2. Better performance in terms of bandwidth should not be expected since the basic access mechanism is still based on the contention-based distributed channel access just like the DCF. The fundamental difference is only that it supports priority based differentiation.

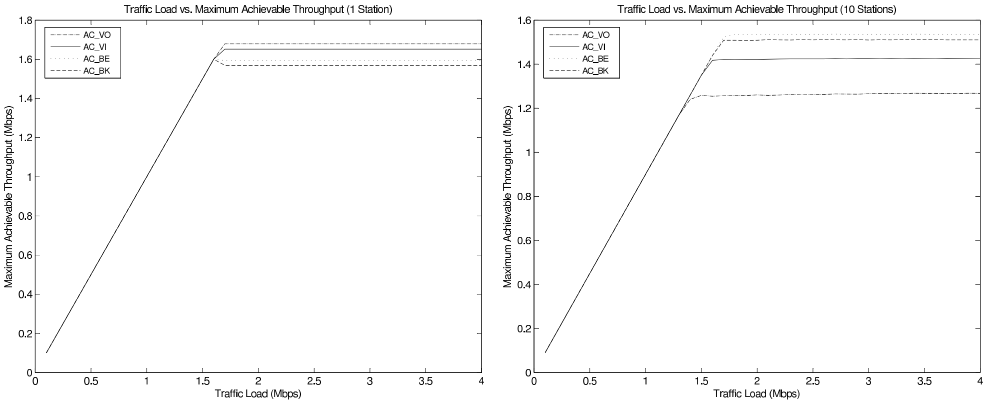
7.3.3.3 AC Maximum Achievable Throughput Comparison

In the third comparison study, the performances of ACs are compared with respect to their maximum achievable throughput results. The maximum achievable throughput for an AC is measured such that one traffic stream corresponding to only one AC is transmitted by a station.

The packet size of 1024 bytes is used for all ACs, and the total load is increased by increasing the load for the single traffic stream the station is transmitting. Figure 7.14 shows the comparison results.

Results and Discussions: Figure 7.14a shows the results for the 1 station case. As it is seen, higher the priority of AC, higher the maximum achievable throughput it receives, due to smaller EDCA parameter values it uses. Both types of contention, namely, internal and external, are non-existent in this scenario since only one station is transmitting traffic corresponding to only one AC. This also shows that CWmax value does not play any role in this scenario since the Contention Window sizes is never doubled. A similar study to measure maximum achievable throughput is also presented in [24].

The results in Figure 7.14b take into account the existence of external contention since now 10 stations are transmitting, although each station is transmitting traffic stream corresponding to only one AC. It gives us very important and interesting result that AC_VO receives the lowest maximum achievable throughput. It is due to the fact



(a) Maximum achievable throughput comparison with 1 station.

(b) Maximum achievable throughput comparison with 10 stations.

FIGURE 7.14: MAXIMUM ACHIEVABLE THROUGHPUT COMPARISON OF ACs.

that AC_VO has very small CWmin and CWmax values, 7 and 15, respectively, and thus suffers from high number of collisions. High collision rate results in wasted bandwidth resources and thus severally degrades the traffic performance. The maximum throughput AC_VO receives is only 1.2 Mbps compared to 1.7 Mbps in the case without external contention. From the figure, it is seen that the similar degradation in throughput is also faced by the traffic of AC_VI, also due to the small CWmin and CWmax values, although the results are significantly better due to the slightly larger values, 15 and 31, compared to 7 and 15 for AC_VO, a behavior also studied in [18].

The significantly reduced performance of these high priority ACs highlights their weak points. It shows that the performance of a high priority AC drops with the addition of every new station transmitting the traffic of high priority AC. A large number of stations with AC_VO or AC_VI traffic will severally destroy the traffic performance of every individual station.

This problem introduces the importance of an admission control mechanism, which enables the AP to allow or disallow a new station, or a new traffic stream within a station, to use a high priority AC, according to network conditions. The requirement of such an admission control mechanism is also expressed in performance studies in [27], [20], [35], [41] and [32].

A possible solution for the above problem would be to dynamically increase the CWmin and CWmax values with every new station or traffic stream being added to the network. The impact of such a solution on the performance of high priority traffic is presented in [18] and [32].

This also points towards the importance of optimizing the traffic performance by fine-tuning the values of EDCA parameters. The standard specifies that the AP can dynamically adapt the values for EDCA parameters according to the load on the network and advertise the updated values to all stations which then use new values to contend for the medium.

Another interesting observation is that in these types of conditions, AC_BE receives better throughput compared to all other ACs. Although the CWmin and CWmax values for both AC_BE and AC_BK are same, AC_BE is served better mainly because of its smaller AIFSN value of 3 compared to 7 for AC_BK.

7.4 Evaluation Part 2

In second part of the evaluation phase, some more realistic traffic scenarios are considered. In order to study the performance of EDCA in hotspots, in each of the scenarios, the number of stations is increased with the time such that every new station increases the total network load. Furthermore, instead of using same bit rates and packet sizes for all ACs, more realistic values are considered based on the real world applications a particular AC models. The traffic types and their characteristics used for evaluation part 2 are presented in Table 7.3. It should be noted that the traffic bit rates, specifically that of video, are set to very small values because of the limitation of 2 Mbps bandwidth as discussed in the beginning of the chapter. Most of the video applications today use higher bit rates. Applications with bit rates higher than 1 Mbps are common nowadays.

Total six different scenarios are studied for the second part. Next section presents the description along with results and discussions for each of the scenarios.

Traffic Type	**AC**	**Packet Size (bytes)**	**Interval (ms)**	**Rate (Kbps)**
Voice	AC_VO	80	22	28
Video	AC_VI	1460	122	96
Data (Best Effort)	AC_BE	1024	73.1	112
Data (Background)	AC_BK	1024	107.8	76

TABLE 7.3: TRAFFIC CHARACTERISTICS.

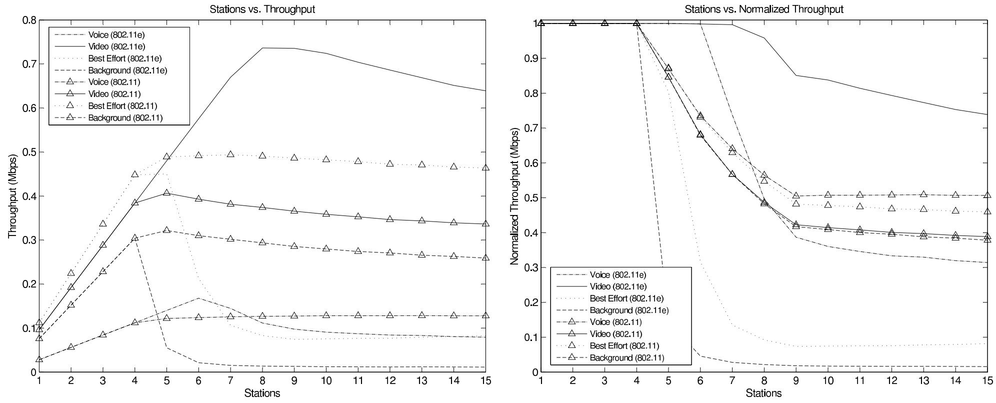
7.4.1 Scenario 1

Description: Number of stations increases from 1 to 15, with each station transmitting streams corresponding to all four traffic types, as summarized in Table 7.3. The goal is to investigate the performance of individual ACs, i.e., how well traffic corresponding to a particular AC is served, and, to compare the performance of 802.11e EDCA and 802.11 DCF as well. It is the same scenario as discussed to explain the QoS limitations of 802.11 in Section 3.2. Similar simulation studies have been carried in [24] and [28].

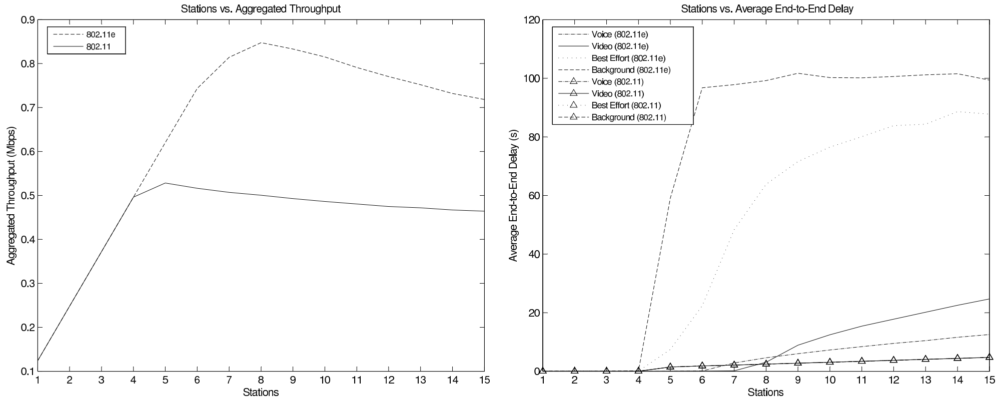
Just to revise, while in 802.11e the four traffic streams are served with four different ACs, there is no such mechanism in 802.11 and thus all four traffic streams are served with same priority.

Four performance metrics are studied, namely, throughput, aggregated throughput, end-to-end delay, and, number of collisions. The aggregated throughput is measured by summing up the throughput figures for all four traffic streams. Only external collisions are studied for the comparison of 802.11e and 802.11 because internal collisions do not exist in 802.11. Figure 7.15 shows the results.

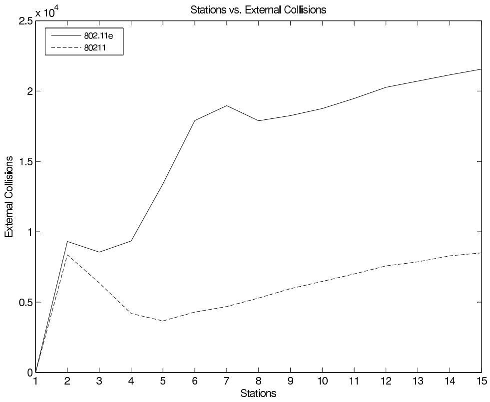
Results and Discussions: Figures 7.15a and 7.15b present the throughput results in absolute and normalized forms, respectively. Normalized throughput is the ratio of the data transmitted to the total data offered, and is presented to more closely observe the results and for the comparison with Figure 3.1b as well. From the results it is evident that while 802.11 DCF treats all types of traffic equally, 802.11e EDCA successfully introduces service differentiation through different ACs. As it is seen, in 802.11, the throughput for all four traffic streams starts to drop equally as the fourth and fifth stations are added to the network. This clearly shows the fundamental problem



(a) Average throughput (absolute). (b) Average throughput (normalized).



(c) Average aggregated throughput. (d) Average end-to-end delay.



(e) External Collisions.

FIGURE 7.15: RESULTS FOR EVALUATION PART 2, SCENARIO 1.

with 802.11, i.e., all types of traffic are treated equally, there is no means of prioritization/differentiation, as discussed in Section 3.2. On the other hand, 802.11e enables the high priority traffic (voice and video) to receive higher and constant throughput through its service differentiation mechanism. We observe that both voice and video traffic streams receive remarkably improved throughput as compared to 802.11. The throughput for voice starts to drop as the sixth station is added to the network, and the throughput for video starts to drop as the sixth station is added.

An interesting observation is that although 802.11e enables higher throughput for voice and video traffic, in comparison to 802.11, the throughput drops more rapidly as the number of stations increases. The reason is, as discussed before, that the corresponding ACs, AC_VO and AC_VI, have smaller CWmin and CWmax values, and therefore as the number of stations increases, the number of collisions also increases resulting in dramatic drop in performance. It is in accordance with the observation in Section 7.3.3.3 that there should be some mechanism of dynamically increasing the CWmin and CWmax values with increasing number of contending stations. Larger values of CWmin and CWmax improve the chances of successful transmissions, a fact also studied in [18] and [19].

The drastic drop in throughput for background (AC_BK) and best effort (AC_BE) traffic streams in 802.11e compared to 802.11 once again confirms that 802.11e provides better service to high priority traffic on the expense of worse performance of low priority traffic. It also revises the earlier observation that under high loads of high priority traffic, the low priority traffic is starved. The behavior is also studies in research work in [20] [26] and [18].

Figure 7.15c presents the aggregated throughput results and reconfirms that 802.11e provides remarkably improved throughput in comparison to 802.11.

Average end-to-end delay results are presented in Figure 7.15d, which show the effectiveness of service differentiation scheme of 802.11e, in that while in 802.11 all types of traffic suffer from same amount of delay regardless of their QoS requirements, in 802.11e, higher priority traffic suffers from significantly lower delays compared to the lower priority traffic. In 802.11, the delay for all traffic types starts to increase as soon as the fourth station is added to the network. Comparatively, in 802.11e the delay for traffic streams of video (AC_VI) and voice (AC_VO) starts to increase with the addition of sixth and seventh station.

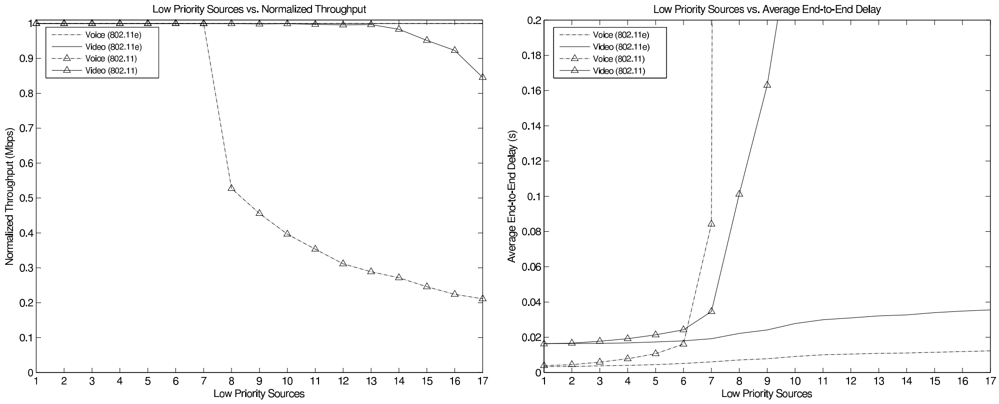
Finally, Figure 7.15e shows the comparison of number of collisions in 802.11 and 802.11e and concludes that 802.11e suffers from much high collision rates, mainly due to the small CWmin and CWmax values for AC_VI and AC_VO, as discussed throughout in Sections 7.3.1, 7.3.2 and 7.3.3.

7.4.2 Scenario 2

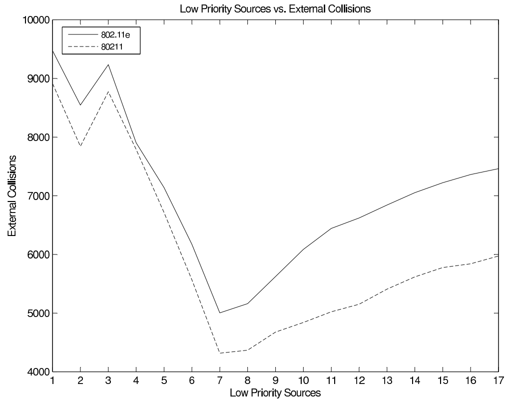
Description: Two stations are transmitting voice traffic and two other stations are transmitting video traffic. The total number of stations increases gradually from 4 to 21, such that each increasing station is transmitting two streams of low priority traffic background and best effort. Such a station is referred to as *low priority source* in the results. The goal is to study the performance of high priority traffic with respect to increasing number of low priority traffic. A performance study of a similar simulation scenario is presented in [24].

Three performance metrics considered are throughput, end-to-end delay and number of collisions. The results are presented in Figure 7.16.

Results and Discussions: Figure 7.16a presents the throughput results for 802.11 and 802.11e. Here again the normalized throughput is plotted instead of absolute throughput, in order to more effectively compare the results for traffic streams with different bit rates. Moreover, only the throughput results for video and voice traffic are shown



(a) Normalized throughput. (b) Average end-to-end delay.



(c) External collisions.

FIGURE 7.16: RESULTS FOR EVALUATION PART 2, SCENARIO 2.

since the aim is to study the behavior of high priority traffic under high load of low priority traffic. The aggregated throughput for a traffic type is shown by summing up the throughput of the two stations transmitting that particular traffic type. The effectiveness of 802.11e QoS scheme is clearly evident from the results. While in 802.11 the voice and video traffic throughput figures start to drop with the addition of 7th and 13th low priority source, in 802.11e both voice and video traffic maintain constant throughput even in the saturated conditions.

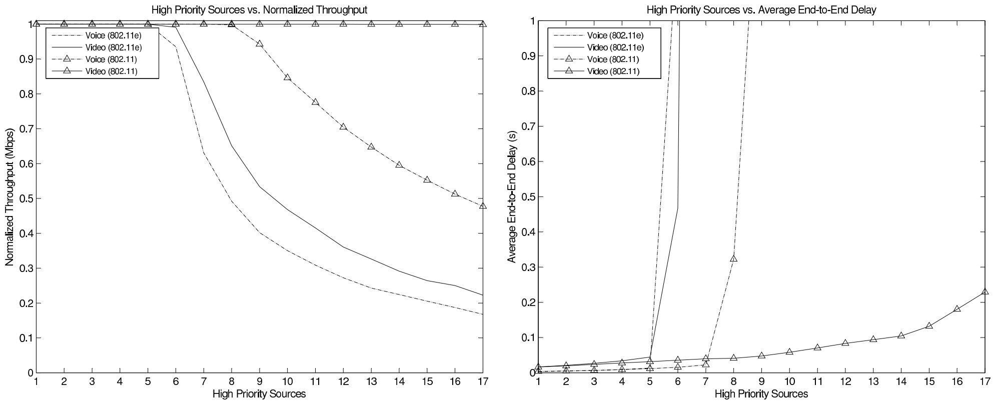
Figure 7.16b shows the results for end-to-end delay, and like many observations before, further confirms the capability of 802.11e in providing service differentiation in comparison to 802.11. Here we see that in 802.11e, the delays for video and voice traffic remain under acceptable limit and there is no significant increase with too many stations are added to the network. On the other hand, under same circumstances, the delay for voice and video traffic in 802.11 approaches to infinity.

Figure 7.16c presents the results for collisions, and as before, shows that 802.11e suffers from higher collisions. The relatively smaller difference in number of collisions for 802.11e compared to 802.11 is due to the fact that only 4 stations are transmitting with small CWmin and CWmax values, i.e., AC_VO and AC_VI traffic, compared to 17 stations transmitting AC_BK and AC_BE traffic with the CWmin and CWmax values same as in 802.11.

7.4.3 Scenario 3

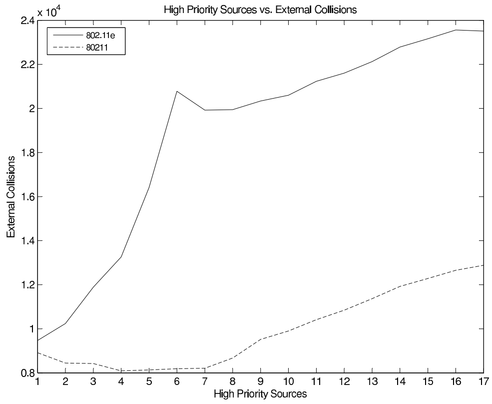
Description: Scenario 3 is similar to Scenario 2, except that this time each increasing station is also transmitting high priority traffic of voice and video type instead of low priority traffic. Such a station is referred to as *high priority source* in the results. The aim here is to analyze the performance of high priority traffic under increasing load of of high priority traffic. A similar simulation study is performed in [20].

The same three metrics throughput, end-to-end delay, and, number of collisions, are considered here. The results are shown in Figure 7.17.



(a) Normalized throughput.

(b) Average end-to-end delay.



(c) External collisions.

FIGURE 7.17: RESULTS FOR EVALUATION PART 2, SCENARIO 3.

Results and Discussions: Figures 7.17a and 7.17b present the results for throughput and end-to-end delay, and show that the performance is terribly reversed in this scenario. The performance of voice and video traffic in 802.11e is by big margin worse than that of 802.11. From Figure 7.17a, we see that the throughput for voice and video in 802.11e starts to drop much earlier compared to 802.11. Same results are seen for the end-to-end delays from Figure 7.17b. It again highlights the major drawback of the 802.11e QoS scheme recently studied in Sections 7.3.3.3 and 7.4.1, that too many stations transmitting high priority traffic severally degrade the overall performance.

Seeing the performance of high priority ACs in this scenario, it is interesting to conclude that IEEE Working Group expects high priority traffic (multimedia applications) users to be a relatively smaller ratio compared to the low priority traffic (data oriented applications) users in the network at a given time. This is in accordance to the recent studies that user generated traffic is generally more data oriented, as presented in [42].

Figure 7.17c presents the results for collisions and shows the huge difference in the number of collisions for 802.11e compared to 802.11. The behavior is not unexpected since for 802.11e all stations are transmitting traffic of AC_VO and AC_VI types with CWmin and CWmax values of 7 and 15, respectively, compared to CWmin and CWmax values of 31 and 1023 in 802.11.

7.4.4 Scenario 4

Description: Scenario 4 is similar to Scenario 2, except that this time the traffic types are reversed, i.e., two stations are transmitting background traffic and two other stations are transmitting best effort traffic, and the number of stations is increased gradually such that every new station is transmitting voice and video traffic. The goal is to study the performance of low priority traffic under the high load of high priority traffic. A similar study is presented in [20].

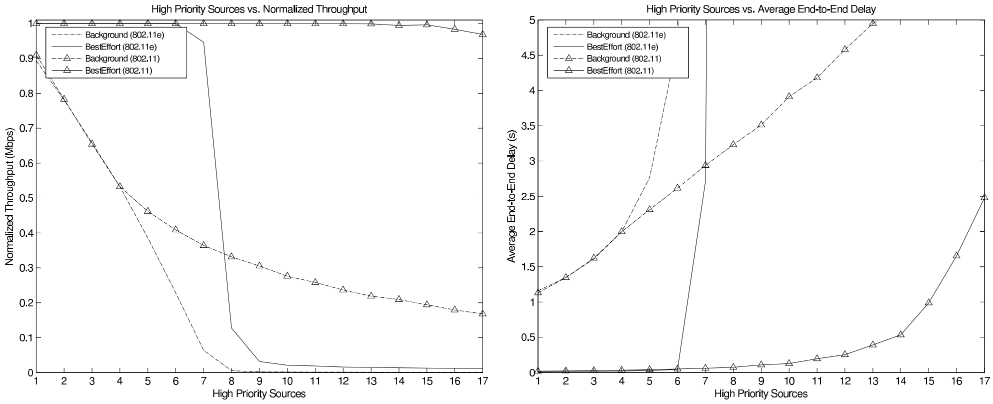
Figure 7.18 presents the results, for the same three metrics considered before.

Results and Discussions: Figure 7.18a presents the throughput results for 802.11 and 802.11e. As expected, in 802.11e the performance of background and best effort traffic is severally degraded in presence of high priority traffic. As it is seen, background (AC_BK) is completely starved as soon as 8th station is added to the network. Best effort (AC_BE) traffic suffers from similar degradation since it receives only 5% (approximately) share of the network bandwidth. We revisit the conclusion from Sections 7.3.2 and 7.4.1 that high load of high priority traffic starves the low priority traffic.

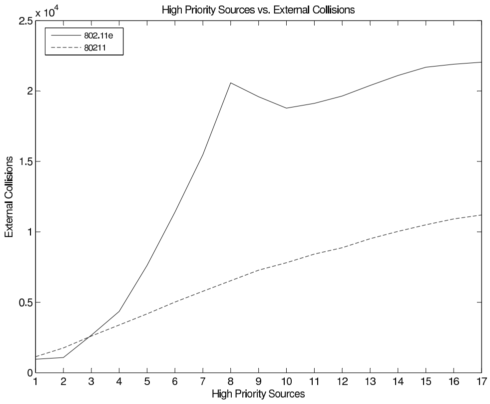
Observing the better results received by 802.11 in Figures 7.18 and 7.17, it is concluded that 802.11 provides better results when the traffic generated by the users belongs to same priority. 802.11e enhances the results only when the traffic becomes heterogenous in terms of priority. As we analyzed earlier, 802.11e performs significantly worse if the traffic is of the same, but of high priority. The reason, as discussed several times, is the smaller CWmin, CWmax sizes for high priority ACs, AC_VO and AC_VI. It is thus concluded that in 802.11e, at a given network condition if most of the traffic belongs to the same priority, no matter what the priority is, then the CWmin and CWmax values for the corresponding priority/AC should be switched to 31 and 1023, respectively. It requires continuous monitoring by the AP, which dynamically adapt the CWmin and CWmax values depending on the ratio of stations generating traffic of same priority.

7.4.5 Scenario 5

Description: The downlink/uplink fairness problem is well known for 802.11 networks. As the number of uplink transmissions (stations transmitting to AP) increases, the performance of downlink transmissions (AP transmitting to stations) is severally degraded. The obvious reason is that as the AP also contends for the medium with the same priority as of an ordinary station, it has to compete with large number of stations to win the access to the medium. Therefore, as the number of uplink transmissions increase,



(a) Normalized throughput. (b) Average end-to-end delay.



(c) External collisions.

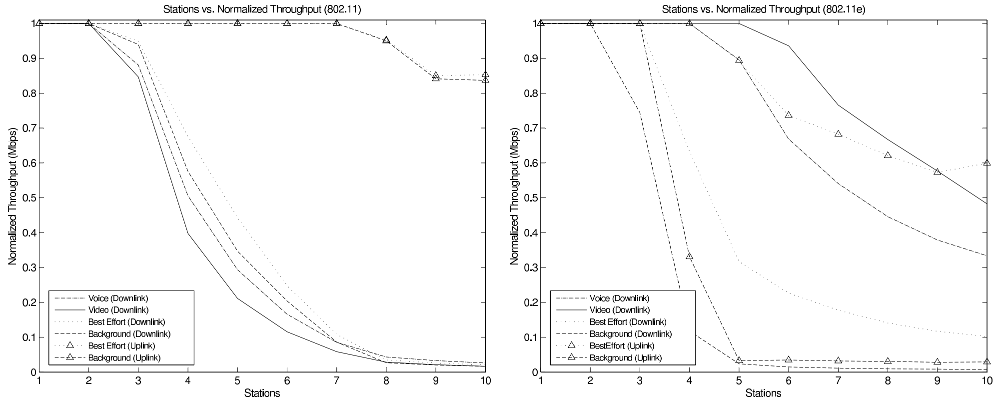
FIGURE 7.18: RESULTS FOR EVALUATION PART 2, SCENARIO 4.

the bandwidth share for AP drops rapidly. (In other words, just like in 802.11e, the increasing number of high priority traffic starves the lower priority traffic, the increasing number of uplink traffic starves the AP).

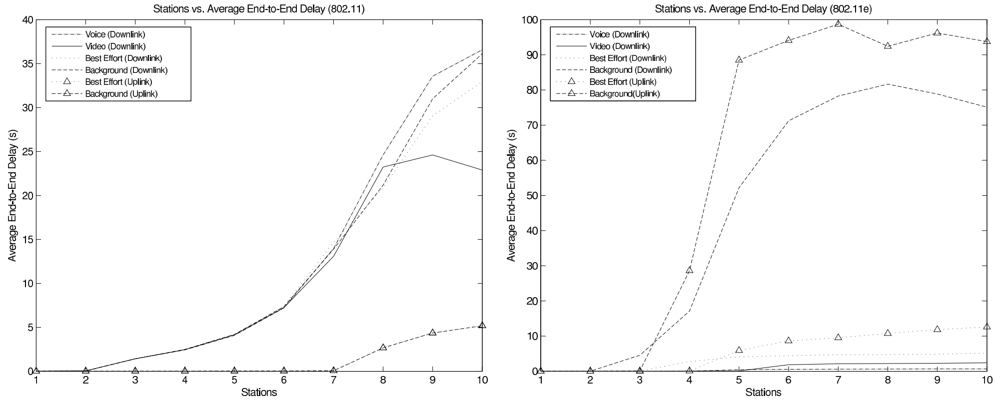
The issue is more crucial for realtime and interactive multimedia applications in which in time response from the server is of very high importance. For example, in voice applications (i.e., VoIP), any excessive delays from the server/AP side destroy the speech quality, or, may even result in termination of the communication.

With the help of this scenario, the aim is to study how well 802.11e is able to solve the downlink/uplink fairness problem, especially since the main purpose behind the introduction of 802.11e is to support QoS for realtime and interactive multimedia applications.

The scenario considers two different cases. In the first case, the stations are transmitting traffic of background and best effort types, and in the second case the stations are transmitting traffic of voice and video types. For both cases, the number of stations is increased from 1 to 10, and AP is transmitting traffic corresponding to all four types, background, best effort, video and voice, to all stations. The two cases are considered for both 802.11 and 802.11e for comparison reason. Similar simulation studies are presented in [29] and [28]. Figures 7.19 and 7.20 present the results.



(a) Normalized throughput (802.11). (b) Normalized throughput (802.11e).



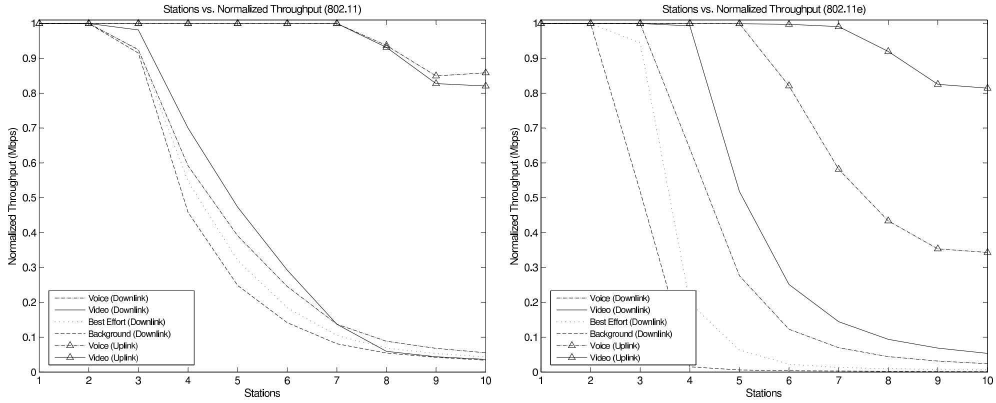
(c) Average end-to-end delay (802.11). (d) Average end-to-end delay (802.11e).

FIGURE 7.19: RESULTS FOR EVALUATION PART 2, SCENARIO 5, CASE 1

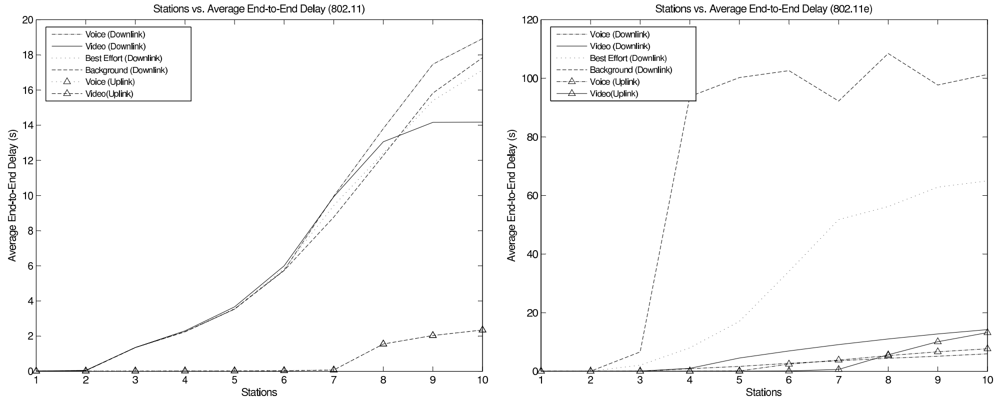
Results and Discussions: The results for the first case for 802.11e and 802.11 are presented in Figures 7.19a and 7.19b, respectively. From the results, the effectiveness of 802.11e QoS mechanism is apparent. While in 802.11, the throughput for the downlink transmissions for all four traffic types drops drastically, in 802.11e, all four downlink traffic are well served. Specifically, remarkable improvements are seen in the performance of voice and video traffic, which is the primary concern of this scenario. An interesting observation is that the uplink traffic, specifically, best effort, still receives good throughput.

Figures 7.20a and 7.20b present the results for the second case. We see that in the presence of high priority voice and video uplink traffic, 802.11e is still able to show significant improvements over 802.11. Although the uplink voice and video traffic is still served better than downlink voice and video traffic, but the downlink traffic does not suffer from as much drop in performance as it is in 802.11. It is seen that the performance of voice transmissions between AP and stations remains up to the acceptable limits until after the 5th and 6th stations are added.

From the above two results, it is concluded that 802.11e effectively solves the uplink/downlink fairness problem found in 802.11 and thereby efficiently enhances the capability of network of supporting voice communications.



(a) Normalized throughput (802.11). (b) Normalized throughput (802.11e).



(c) Average end-to-end delay (802.11). (d) Average end-to-end delay (802.11e).

FIGURE 7.20: RESULTS FOR EVALUATION PART 2, SCENARIO 5, CASE 2.

7.5 Summary

Performance evaluation of IEEE 802.11e EDCA MAC mechanism is presented using the 802.11e EDCA model implemented in GloMoSim. A wide range of performance metrics is used including throughput, end-to-end delay, MAC delay, collision rate, internal collision rate, packet transmission ratio, packet loss and packet drops. The default set of EDCA parameters is used as specified in the draft standard. The legacy 802.11 PHY with data transmission rate of 2 Mbps is used for simulations because currently it is the only available 802.11 PHY in GloMoSim.

7.5.1 Evaluation Part 1

The evaluation phase is divided into two parts. The first part evaluates 802.11e EDCA with a set of simple scenarios in order to more clearly observe the role of individual EDCA parameters (AIFSN, CWmin and CWmax) in realizing service differentiation. Two scenarios are considered with 1 and 10 stations transmitting traffic corresponding to all four ACs to the AP, such that the load is increased from 0.1 to 4.0 Mbps by increasing the data rate of each traffic stream. The two scenarios help to study the performance in both lightly and highly loaded network situations, and, with and without the presence of external contention.

Throughput results for the two scenarios clearly shows the how effectively service differentiation is introduced through different ACs. Higher the priority of an AC, higher the throughput it receives and vice versa. While the throughput for low priority ACs starts to drop as the network becomes congested, the high priority ACs, AC_VI and AC_VO, keep receiving consistent shares of bandwidth. Results for the Scenario 2, when the total number of stations is ten, help to conclude that in presence of higher external contention the low priority ACs receive very small share of bandwidth after the network becomes congested. The starvation of low priority ACs is not very unexpected since as they have larger AIFS, CWmin and CWmax values compared to the high priority ACs, they have to wait much longer before they can access the medium, and under heavily loaded situations most of the time they are not able to decrement their backoff timers. AC_BK is completely starved which is due to the fact that its AIFSN of 7 is too big in comparison to the AIFSN values of other ACs. Another factor is that it experiences both levels of contention, namely, internal and external. Internal contention has a stronger influence on the starvation because AC_BK always has to compete internally with three high priority ACs which act just like three contending stations. Packet transmission ratio results further highlight this fact showing that ratio for AC_BK goes down to as low as zero. AC_BE performs relatively well, mainly due to the fact that it has a much smaller AIFSN value of 3 in comparison to that of AC_BK.

Result for average end-to-end delay are also very promising, in that the delay faced by a particular AC is according to the requirements of traffic it is designed for. The high priority ACs, AC_VO and AC_VI, maintain low and consistent delay figures which is what desirable for the modern multimedia applications. The delay for the low priority ACs increases exponentially as network becomes congested, but as these ACs are mainly designed for the delay-tolerant applications, certain amount of delay does not degrade their performance beyond an acceptable limit.

Internal collisions results for the two scenarios show the drawback of smaller Contention Window sizes in that more than two third of the total collisions occur between the higher priority ACs, AC_VO and AC_VI, which have CWmin and CWmax values of 7 and 15, and 15 and 31, respectively. Results for collisions (external) in Scenario 2 further strengthens the fact that high priority ACs suffer from very high number of collisions, which severely degrades their performance.

Results show that AC_VO and AC_VI suffer from higher number of packet drops, which is a consequence of their high collision rate, i.e., higher the number of collisions, higher the number of retransmissions and thus higher the packet drop rate. Though high packet drop does not question the effectiveness of the service differentiation scheme since these ACs are designed for multimedia applications which are generally loss-tolerant.

It also points out that the retry limit values for ACs should be set such that the ACs for multimedia applications (AC_VO and AC_VI) have smaller values and the ACs for data oriented applications (AC_BK and AC_BE) have larger values, because large retry limits values result in lower number of packet drops which is desirable for the data oriented applications, and, small values result in lower delays which is desirable for real-time multimedia applications.

Next, for each of the two scenarios, three additional cases are considered such that the EDCA parameters are *switched off* one by one in order to observe the influence of individual EDCA parameters on service differentiation.

7.5.1.1 Service Differentiation based on CWmin and CWmax

In the first case, the AIFSN for all four ACs is set to 2. The results show how effectively service differentiation is introduced through only varying the CWmin and CWmax values for different ACs. Smaller the CWmin and CWmax values for an AC, higher the throughput it receives and vice versa. AC_VO receives and highest throughput because it has the smallest CWmin and CWmax values of 7 and 15, respectively. AC_VI receives the second highest share of bandwidth. And AC_BK and AC_BE receive equal share because they have the same CWmin and CWmax values of 31 and 1023, respectively.

From the results it is evident that throughput for AC_BK is improved significantly, mainly due to the fact that its AIFSN value is set to 2 instead of original value of 7. It confirms that the primary reason of starvation of AC_BK is its large AIFSN value.

Comparing the throughput results in the two scenarios with one and ten stations shows that the impact of internal contention reduces in comparison to the external contention with increasing number of contending stations, as in the first case the difference between throughput of AC_BK and AC_BE is much bigger compared to the second case when external contention is introduced.

The results again show that AC_VO and AC_VI suffer from higher collisions because of their small CWmin and CWmax values. It is the biggest weakness of CWmin and CWmax based differentiation. As the load increases, the number of collisions also increases resulting in lowered overall throughput. It indicates that only CWmin and CWmax parameters are not enough for service differentiation, since as smaller values of these parameters are necessary to introduce differentiation for the high priority ACs, there is a tradeoff between high throughput and service differentiation. Another significant observation is that using same AIFSN values greatly increases the number of internal collisions, particularly for AC_BK and AC_BE.

An important fact is that in lightly loaded network conditions, the role of CWmax in introducing service differentiation is less significant compared to that of CWmin, although it then becomes more and more stronger with the increase in network congestion. It also indicates that in congested network situations, larger CWmax values for high priority ACs could remarkably reduce the collision rate.

7.5.1.2 Service Differentiation based on AIFS

In the second additional case, the CWmin and CWmax values are set to fixed values 31 and 1023, respectively, to clearly observe the role of AIFS on service differentiation.

By comparing the results to the CWmin and CWmax based differentiation, it is evident that AIFS provides superior and more reliable service differentiation. The reasons are: (1) The throughput received by an AC is directly in accordance to its AIFSN value. From high to low priority AC, the relative differences in throughput figured clearly reflect the AIFSN values of ACs, i.e., 2, 2, 3 and 7. (2) The large margin between throughput results of AC_VI and AC_BE shows that even AIFS difference of a single slot time is very effective in realizing differentiation. (3) AIFS provides consistent and more reliable differentiation independent of the different network conditions. It is in contrast to the CWmin and CWmax based differentiation which becomes less effective with increase in network congestion. It is because as the congestion increases, number of collisions also increases because of the small CWmin and CWmax values, and, the average Contention Window size also increases resulting in reduced impact of CWmin on service differentiation.

The throughput results show that AC_BK is almost completely starved once again because of its large AIFSN value of 7. It confirms that AIFS based differentiation is highly sensitive to the AIFSN values and too big AIFSN values for lower priority ACs lead them to starvation.

A comparison with the results of the normal scheme (using both AIFS and CWmin and CWmax for differentiation) shows that the performance of high priority ACs is slightly lowered in AIFS based differentiation, due to the fact that high priority ACs now also have same CWmin and CWmax values as of lower priority ACs, i.e., the additional differentiation based on CWmin and CWmax does not exist. Another observation is that AC_BE now receives significantly better throughput, which is obvious since as the bandwidth is distributed among all four traffic streams, the more one gets the less the others get. It also shows that high priority ACs are provided differentiation on the expense of lowered performance of lower priority ACs.

Comparison of the results for AIFS based and CWmin and CWmax based differentiation leads to the interesting observation that in lightly loaded situations, the latter serves both the high and low priority ACs slightly better in terms of throughput.

Comparing to the results of the normal scheme and CWmin and CWmax based differentiation concludes that there is dramatic reduction in number of internal and external collisions by (1) using different values of AIFSN for different ACs, and (2) using large values of CWmin and CWmax.

7.5.1.3 Throughput Comparisons

After evaluating with the simple scenarios with one and ten stations, as well as the additional cases to analyze the individual impact of EDCA parameters on service differentiation, three overall comparative studies are presented.

In the first comparison study, the performances of different EDCA parameters based differentiation schemes studied in Scenarios 1 and 2 are compared in terms of their aggregated throughput, which is calculated by adding throughput figures for of all four ACs. The results show that AIFS based service differentiation receives the highest aggregated throughput, and, CWmin and CWmax based differentiation receives lowest aggregated throughput, which is mainly because of the high collision rate it suffers from.

Moreover, the lower throughput results for the normal scheme compared to the AIFS based and CWmin and CWmax based differentiation schemes conclude that differentiation is realized on the expense of lower overall throughput.

The remarkably superior performance of AIFS based differentiation raise the question why only AIFS is not chosen as the service differentiation parameter by the IEEE Working Group. One reason could be the issue of coexistence of 802.11 and 802.11e. As AIFS for high priority ACs in 802.11e is equal to DIFS in 802.11, the only possible way is to use smaller CWmin and CWmax values for these ACs to introduce differentiation.

In the second comparison study, the aggregated throughput results of 802.11 DCF and 802.11e EDCA are compared. For this, each station is transmitting traffic corresponding to all four traffic types, and aggregated throughput is calculated by summing up the throughput of all four traffic streams. The results show that the EDCA receives lower throughput in comparison to DCF, which further strengthens the fact that EDCA provides service differentiation on the expense of lower overall throughput. Next, while DCF maintains consistent throughput after the network becomes congested, the throughput for EDCA keeps dropping gradually with the increase in load. The primary reason is the small CWmin and CWmax values for high priority ACs in 802.11e resulting in high collision rate.

In the third comparison study, the performances of ACs are compared in terms of their maximum achievable throughput results. Maximum achievable throughput for an AC is calculated such that the station transmits only one traffic stream corresponding to that AC. It gives very important result that AC_VO receives the lowest maximum achievable throughput. The reason, as always, is that AC_VO suffers from high number of collisions because of very small CWmin and CWmax values. AC_VI also suffers from similar degradation, although relatively lower because of its slightly larger CWmin and CWmax values. It is concluded that the performance of a high priority AC drops with the addition of every new station transmitting the traffic of high priority AC.

The results highlights the importance of an admission control mechanism which enables the AP to allow or disallow a new station or traffic stream to use a high priority AC according to network conditions. A possible solution would be to dynamically increase the CWmin and CWmax values with every new station or traffic stream. This points towards the importance of optimization of AC performance by adjusting the EDCA parameters according to network conditions and available resources.

Another interesting observation is that in these types of conditions, AC_BE receives better throughput compared to all other ACs, which is mainly due to its small AIFSN value of 3 compared to 7 for AC_BK.

7.5.2 Evaluation Part 2

The second part of the evaluation considers some more realistic simulation scenarios, with traffic characteristics used by real world applications. Comparisons of 802.11e EDCA and 802.11 DCF are also presented.

7.5.2.1 Scenario 1

In Scenario 1, number of stations increases from 1 to 15, with each station transmitting traffic corresponding to all four traffic types. The comparison of 802.11e EDCA and 802.11 DCF shows that although 802.11e enables higher throughput for voice and video traffic, in comparison to 802.11, the throughput drops more rapidly as the number of stations increases. As before, it is due to the fact that AC_VO and AC_VI suffer from high collision rate which results in dramatic drop in performance. It strengthens the conclusion that there should be some mechanism of dynamically updating the CWmin and CWmax values with increasing number of stations. Larger values of CWmin and CWmax improve the chances of successful transmissions.

The drastic drop in throughput for background (AC_BK) and best effort (AC_BE) traffic streams in 802.11e compared to 802.11 once again confirms that 802.11e provides better service to high priority traffic on the expense of worse performance of low priority traffic. It also revises the earlier observation that under high loads of high priority traffic, the low priority traffic is starved.

7.5.2.2 Scenario 2

In Scenario 2, two stations are transmitting voice and two other stations are transmitting video traffic, and total stations increase from 4 to 21, such that each new station is transmitting two streams of low priority background and best effort traffic. The goal is to study the performance of high priority traffic in high load of low priority traffic.

Results confirms the capability of 802.11e in providing service differentiation, in that while in 802.11 the voice and video throughput start to drop as too many stations are added to the network, in 802.11e they maintain consistent throughput even in the saturated conditions. End-to-end delay for video and voice traffic also remain consistent and under acceptable limits, while it approaches to infinity in 802.11.

7.5.2.3 Scenario 3

Scenario 3 is similar to Scenario 2, except that this time each increasing station is also transmitting high priority traffic of voice and video type. The goal is to study the performance of high priority traffic in high load of high priority traffic. Throughput and end-to-end delay results for 802.11e show that the performance is terribly reversed, and is by big margin worse than that of 802.11. It again highlights the major drawback of the 802.11e that too many stations transmitting high priority traffic severally degrade the overall performance.

7.5.2.4 Scenario 4

Scenario 4 is similar to Scenario 2, but traffic types are reversed, i.e., two stations are transmitting background traffic and two other stations are transmitting best effort traffic, and the number of stations is increased gradually such that every new station is transmitting voice and video traffic. The goal is to study the performance of low priority traffic under the high load of high priority traffic.

As expected, throughput results show that for 802.11e the performance of background and best effort traffic is severally degraded in presence of high priority traffic. As it is seen, background (AC_BK) is completely starved as soon as network becomes congested, and best effort (AC_BE) traffic suffers from similar degradation and receives very little share of the bandwidth. The results confirm that high load of high priority traffic starves the low priority traffic.

7.5.2.5 Scenario 5

Scenario 5 considers the downlink/uplink fairness problem well known in 802.11 networks. The issue is more crucial for realtime and interactive multimedia applications (e.g., VoIP) in which in time response from the server/AP is of very high importance. The goal of scenario is to study how well 802.11e is able to solve the problem. The scenario considers two cases. In the first case, the stations are transmitting traffic of background and best effort types, and in the second case the stations are transmitting traffic of voice and video types. For both cases, the number of stations is increased from 1 to 10, and AP is transmitting traffic corresponding to all four traffic types.

From the results, the effectiveness of 802.11e QoS mechanism is apparent. While in 802.11, the throughput for the downlink transmissions for all four traffic types drops drastically, in 802.11e, all four downlink traffic streams are well served. Specifically, remarkable improvements are seen in the performance of voice and video traffic, which is the primary concern of this scenario. For the second case, the results show that in the presence of voice and video uplink traffic, 802.11e is still able to show significant improvements over 802.11. Although the uplink voice and video traffic is still served better than downlink voice and video traffic, the downlink traffic does not degrade as much as in 802.11. It is concluded that 802.11e effectively solves the uplink/downlink fairness problem found in 802.11 and thereby efficiently enhances the capability of network of supporting real time multimedia applications.

Chapter 8

Conclusions and Further Research

8.1 Introduction

This chapter presents the concluding remarks together with some of the potential future research areas. Section 8.2 presents the conclusions drawn from the thesis work and Section 8.3 discusses the future research areas and improvements.

8.2 Conclusions

A large amount of research work has been carried out to enhance the QoS support in IEEE 802.11 networks. Recently IEEE released the final draft of an upcoming standard, called 802.11e, which is an extension to IEEE 802.11 to provide QoS support. In our work, we implemented the basic access mechanism of IEEE 802.11e, called Enhanced Distributed Channel Access (EDCA) in GloMoSim v2.03, and evaluated its performance in introducing QoS support in 802.11 networks.

The first step was to study and understand the IEEE 802.11e standard available in the form of final draft. More specifically, complete and thorough understanding of IEEE 802.11e EDCA mechanism was essential in order to develop the simulation model for it. It required considerable effort to understand the architecture and working of EDCA, also due to the fact that the standard documentations are written in rather vague manner, with lack of technical details. Detailed and careful examination of standard draft helped in understanding how 802.11e EDCA works.

The second step was to learn GloMoSim. It included understanding its system architecture and the APIs operating at each layer, more specifically the MAC sub-layer where the new protocol was to be implemented. Detailed study of GloMoSim facilitated in developing the simulation model more efficiently. Well designed architecture of GloMoSim and the data structures it utilizes made the learning task simpler. However, the biggest drawback of using GloMoSim is the lack of enough supporting documentation. A reasonable amount of time, therefore, was spent on understanding GloMoSim thoroughly. Well written APIs between the layers as well as efficient modular design further favored the implementation task.

The primary feature of the EDCA is to assign packets with different priorities based on the QoS requirements of the traffic they belong to. Interestingly, the draft standard does not specify how such a priority is assigned at the higher layers and how it travels to the MAC layer. The programming of EDCA confirmed that more or less the whole network stack would required to be updated since modifications at each of the higher layers are inevitable to the pass the priority to the next layer. It also highlighted the potential drawback of adapting to 802.11e, since every application has to be updated in order to be compatible with 802.11e.

A comprehensive evaluation study based on a wide range of simulation scenarios has been presented. Performance metrics considered are throughput, end-to-end delay, MAC delay, collision rate, internal collisions rate, transmission ratio, packet loss and packet drop. From the results, it is concluded that IEEE 802.11e introduces a very effective service differentiation mechanism to provide QoS support. With the service differentiation mechanism, the high priority traffic receives larger and consistent share of bandwidth while delay remains under acceptable limits. In the first part of the evaluation, a set of simple simulation scenarios is considered in order to more clearly study the role of individual EDCA parameters in realizing service differentiation. The results show that both EDCA parameters, AIFS and CWmin and CWmax, are very effective in realizing service differentiation when used individually. In the second part, evaluation with some more realistic scenarios is presented considering traffic characteristics used by the real world applications.

Following is the list of key points extracted from the evaluation phase.

1. Under highly loaded network situations, high priority ACs, AC_VO and AC_VI, suffer from high number of collisions due to their small CWmin and CWmax values. High collision rate severally degrades the performance of these ACs, and proved to be the biggest weakness of service differentiation based on CWmin and CWmax.
2. Small values of CWmin and CWmax for high priority ACs result in high number of internal collisions too. Results show that more than two third of the total collisions occur between the higher priority ACs, AC_VO and AC_VI.
3. Retry limits for AC_VO and AC_VI should be set to smaller values since multimedia applications are generally delay-sensitive, and those for AC_BK and AC_BE should be set to larger values since data oriented applications are generally loss-sensitive.
4. In lightly loaded network conditions, the role of CWmax in introducing service differentiation is less significant compared to that of CWmin, although it then becomes more and more stronger with the increase in network congestion. It also indicates that in congested network situations, larger CWmax values for high priority ACs could remarkably reduce the collision rate.
5. High priority ACs, AC_VO and AC_VI, suffer from higher number of packet drops which is also a consequence of high collision rate these ACs suffer from.
6. High collision rate further indicates that only CWmin and CWmax parameters are not enough for reliable and robust service differentiation. It is due to the fact that while small values of these parameters are necessary to introduce differentiation for the high priority ACs, they also result in degradation of system throughput due to high collision rate. Thus there is a tradeoff between high throughput and service differentiation.
7. The results for CWmin and CWmax based differentiation show that using same AIFSN values for all ACs greatly increases the number of internal collisions. Basically, larger the difference in AIFSN values for different ACs, lower the internal collisions and vice versa.

8. Comparing the throughput results in scenarios with and without external contention shows that the impact of internal contention (in comparison to external contention) reduces with increasing number of contending stations.
9. The poor performance of low priority ACs (AC_BE and AC_BK) under high load of high priority ACs (AC_VO and AC_VI) shows that high load of high priority traffic starves the low priority traffic.
10. AC_BK is completely starved under high load of high priority traffic, which is due to the fact that it has too large AIFSN value of 7 in comparison to other ACs. Another factor is that it experiences both levels of contention, namely, internal and external contention.
11. In high load of high priority traffic, AC_BE receives better throughput compared to all other ACs mainly due to its larger CWmin and CWmax values compared to AC_VO and AC_VI, and, smaller AIFSN value compared to AC_BK.
12. Comparison of CWmin and CWmax based differentiation with AIFS based differentiation shows that the latter provides superior and more reliable service differentiation because:
13. (a) The performance of an AC is consistently controlled by its AIFSN value. For example, from high to low priority AC, the relative differences in throughput results clearly reflect the AIFSN values of ACs, i.e., 2, 2, 3 and 7,
14. (b) AIFS difference of even a single time slot is very effective in realizing differentiation, and,
15. (c) AIFS provides consistent and more reliable differentiation independent of the different network conditions. It is in contrast to the CWmin and CWmax based differentiation which becomes less effective with increase in network congestion. It is because as the congestion increases, number of collisions also increases due to the small CWmin and CWmax values, and thus the average Contention Window size also increases, resulting in reduced impact of CWmin on service differentiation.
16. AIFS based service differentiation is highly sensitive to the AIFSN values and too big AIFSN values for lower priority ACs lead them to starvation.
17. The above points indicate that the starvation of low priority ACs could be avoided by using same AIFS for all ACs and using only CWmin and CWmax as the differentiation parameters.
18. Differentiation based on only AIFS highlights the issue of coexistence of 802.11 and 802.11e since the AIFS for high priority ACs in 802.11e is equal to DIFS in 802.11. However, as it is obvious, CWmin and CWmax are therefore used as additional differentiation parameters, i.e., smaller CWmin and CWmax values are used to introduce differentiation for high priority ACs in 802.11e.
19. Comparison of aggregated throughput results shows that AIFS based service differentiation receives the highest aggregated throughput, while CWmin and CWmax based differentiation receives the lowest, mainly because of the high collision rate it suffers from.

20. Comparing the aggregated throughput results shows that the normal scheme, i.e., using both AIFS and CWmin and CWmax for differentiation, suffers from lower throughput, which further strengthens the conclusion that differentiation is realized on the expense of lower overall throughput.

21. Comparison of aggregated throughput of 802.11 DCF and 802.11e EDCA shows that EDCA receives lower throughput in comparison to DCF. It also strengthens the fact that EDCA provides service differentiation on the expense of lower overall throughput.

22. Comparison of aggregated throughput of 802.11 DCF and 802.11e EDCA also shows that while DCF maintains consistent throughput after the network becomes congested, the throughput for EDCA continues to drop gradually with the increase in load. The primary reason is the small CWmin and CWmax values for high priority ACs in 802.11e, resulting in high collision rate.

23. Comparison of maximum achievable throughput of the four ACs shows that AC_VO receives the lowest maximum achievable throughput. The reason, as before, is that AC_VO suffers from high number of collisions. AC_VI receives the second lowest throughput because of the same reason.

24. High collision rate for high priority ACs concludes that performance of a high priority AC drops with the addition of every new station transmitting the traffic of high priority AC. Additionally, too many stations transmitting high priority traffic severally degrade the overall performance of the network.

25. The above point highlights the importance of an admission control mechanism which enables the AP to allow or disallow a new station or traffic stream to use a high priority AC according to network conditions.

26. An effective solution would be to dynamically increase the CWmin and CWmax values with addition of every new station or traffic stream, since large values of CWmin and CWmax improve the chances of successful transmissions. Moreover, a particular set of CWmin and CWmax values is optimal only for a certain number of stations/traffic, after which it is required to be updated. This points towards the importance of optimization of AC performance by adjusting the EDCA parameters according to network conditions and available resources.

27. The drastic drop in throughput for background and best effort traffic streams in 802.11e compared to 802.11 shows that 802.11e provides service differentiation to high priority traffic on the expense of worse performance of low priority traffic.

28. 802.11e service differentiation mechanism effectively solves the downlink/uplink fairness problem well known in 802.11 networks. Downlink traffic of high priority ACs is very well served in the presence of uplink traffic of low priority ACs as well as high priority ACs.

8.3 Further Research

The following list specifies the areas of further research together with the required modifications to the GloMoSim.

1. *Evaluation of HCCA* - Performance evaluation of the centrally controlled channel access mechanism of 802.11e, called HCCA (HCF Controlled Channel Access), in comparison to PCF (Point Coordination Function) in legacy 802.11. It would first require implementation of PCF in GloMoSim and then upgrading it to the HCCA.

2. *Upgrading HTTP, FTP and TCP* - Currently only CBR and UDP have been upgraded for compatibility with IEEE 802.11e. Upgrading HTTP, FTP and TCP would enable simulating more realistic traffic scenarios, specifically voice and bursty web traffic.

3. *Evaluation with higher data rates* - Currently GloMoSim only supports the legacy 802.11 PHY, which is a big hurdle in experimenting more realistic high bit rate multimedia traffic scenarios because of the limited data rate of 2 Mbps it offers. Upgrading GloMoSim for 802.11b and 802.11a is desirable which will enable data transmission rates of up to 11 and 54 Mbps, respectively. It would require modification at the Channel and Radio layers of GloMoSim.

4. *Optimization of EDCA parameters* - The draft standard states that the AP is capable of dynamically adapting the EDCA parameters according to the network conditions. Currently the default EDCA parameter value are used in our implementation. Evaluating 802.11e with varying sets of EDCA parameter values, especially deviating from the default values, would help in studying the optimal performance of the network. As observed from the evaluation results, increasing number of high priority traffic sources severely degrade the overall performance of the network, and, larger AIFSN values for lower priority ACs result in their starvation. Adapting the CWmin, CWmax and AIFSN values according to network conditions will therefore have remarkable improvements on the network performance.

5. *Evaluation with Contention Free Bursting* - In 802.11e, once a station/EDCAF receives the TXOP (access to the medium), it is allowed to transmit multiple frames bounded by a time limit, called TXOP Limit. The process is referred to as CFB (Contention Free Bursting). Implementation of CFB will enable analyzing how well the high priority traffic is served as well as the affects on the performance of low priority traffic.

6. *Block Acknowledgement Mode* - 802.11e defines a block acknowledge mechanism which allows multiple data frames to be acknowledged with only one ACK frame, instead of a separate ACK for each data frame. The implementation and evaluation of Block Acknowledgement Mode will enable studying the improvements it offers in the bandwidth sharing.

7. *Evaluation with Admission Control Mechanism* - The increasing number of high priority traffic sources severely degrades the overall performance of high priority traffic and starves the low priority traffic as well. It is mainly due the fact that high priority ACs have small CWmin and CWmax values which result in high collisions. The importance of an admission control mechanism is therefore apparent, which enables AP to allow or disallow a new high priority traffic stream (or station), keeping in view the current network conditions. Implementation of such an admission control mechanism will allow studying the improvements it offers in the overall network performance.

8. *Coexistence of 802.11 and 802.11e* - It would be interesting to study the performance of 802.11e in presence of 802.11 stations knowing that the AIFSN for high priority ACs in 802.11e is equal to DIFS in 802.11.

References

[1] IEEE Std. 802.11, Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* 1997.

[2] IEEE 802.11e/D13.0, Draft Supplement to Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements.* January 2005.

[3] X.Zeng, R. Bagrodia, and M. Gerla. *GloMoSim: A Library for the Parallel Simulation of Large scale Wireless Networks*. In Proceedings of the 12th Workshop on Parallel and Distribution Simulation PADS, 1998.

[4] IEEE Std. 802.3, Part 3: *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*. 1985.

[5] IEEE Std. 802.5, *Token Ring Access Method and Physical Layer Specificationss*. 1985.

[6] IEEE Std. 802.11b, Supplement to Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band.* 1999.

[7] IEEE Std. 802.11a, Supplement to Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 5 GHz Band.* 1999.

[8] IEEE Std. 802.11g, Supplement to Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band.* 2003.

[9] J. Schiller. *Mobile Communications.* Addison-Wesley, second edition, 2003.

[10] J. F. Kurose and K. W. Ross. *Computer networking: a top-down approach featuring the Internet.* Addison-Wesley, third edition, 2003.

[11] W. Stallings. *High-Speed Networks and Internets: Performance and Quality of Service.* Prentice-Hall, second edition, 2002.

[12] F. Halsall. *Multimedia Communications, Applications, Networks, Protocols and Standards.* Addison-Wesley, 2001.

[13] R. Braden, D. Clark, and S. Shenker. *Integrated services in the Internet architecture: An overview.* RFC 1633, June 1994.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An architecture for differentiated service.* RFC 2475, December 1998.

[15] H. Holma and A. Toskala, editors. *WCDMA for UMTS.* John Wiley & Sons, second edition, 2002.

[16] T. Ojanperä and R. Prasad. *WCDMA: Towards IP Mobility and Mobile Internet.* Artech House Publishers, 2001.

[17] G. Bianchi. *Performance Analysis of the IEEE 802.11 Distributed Coordination Function*. In IEEE JSAC, volume 18, pages 535–547, March 2000.

[18] S. Kuppa and R. Prakash. *Service Differentiation Mechanisms for IEEE 802.11-based Wireless Networks*. In IEEE Wireless Communications and Networking Conference (WCNC), volume 2, pages 796–801, March 2004.

[19] L.Scalia and I.Tinnirello. *Differentiation Mechanisms for Heterogeneous Traffic Integration in IEEE 802.11 Networks*. In BroadWim04, San Jose', CA. US., October 2004.

[20] A. Lindgren, A. Almquist, and O. Schel. *Quality of Service Schemes for IEEE 802.11 Wireless LANs: An Evaluation*. Mobile Networks and Applications, 8:223–235, 2003.

[21] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran. *A Survey of Quality of Service in IEEE 802.11 Networks*. IEEE Wireless Communications Magazine, 11(4):6–14, August 2004.

[22] I. Aad and C. Castelluccia. *Differentiation Mechanisms for IEEE 802.11*. In IEEE INFOCOM, pages 209–218, April 2001.

[23] T. Nilsson. *Resource Allocation and Service Differentation in Wireless Local Area Networks.* Licentiate Thesis, Dept. of Computing Science, Umeå University, June 2005.

[24] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. *IEEE 802.11e Wireless LAN for Quality of Service*. In European Wireless '02,, Florence, Italy, February 2002.

[25] G. Bianchi, T. Vergata, I. Tinnirello, and L. Scalia. *Understanding 802.11e Contention-Based Prioritization Mechanisms and Their Coexistence with Legacy 802.11 Stations*. In IEEE Network, volume 19, July 2005.

[26] P. Garg, R. Doshi, R. Greene, M. baker, M. Malek, and X. Cheng. *Using IEEE 802.11e MAC for QoS over Wireless*. Computer Science Department, Stanford University.

[27] S. Choi, J. DelPrado, S. Shankar, and S. Mangold. *IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation*. In ICC 2003, Anchorage, AL, USA, May 2003.

[28] He and C. Shen. *Simulation Study of IEEE 802.11e EDCF*. In IEEE VTC, 2003.

[29] C. Casetti and C.-F. Chiasserini. *Improving Fairness and Throughput for Voice Traffic in 802.11e EDCA.* In IEEE PIRMC'04, Barcelona, September 2004.

[30] 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), 4:547–566, 2004.

[31] J. W. Tantra, C.H.Foh, and A.B.Mnaouer. *Throughput and Delay Analysis of the IEEE 802.11e EDCA Saturation*. In IEEE International Conference on Communications, ICC, volume 2, pages 219–223, 2005.

[32] Y. Xiao, H. Li, and S. Choi. *Protection and Guarantee for Voice and Video Traffic in IEEE 802.11e Wireless LANs*. In IEEE INFOCOM, pages 2153–2163, 2004.

[33] S. Kuppa, S. Gandham, and R. Prakash. *Undesirable Service Differentiation in Future WLANs*. In First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (short paper), pages 304–307, October 2004.

[34] Y. Xiao. *IEEE 802.11e: QoS Provisioning at the MAC Layer*. In IEEE Wireless Communications, volume 11, pages 72–79, 2004.

[35] J.-D. Kim and C.-K. Kim. *Performance Analysis and Evaluation of IEEE 802.11e EDCF*. Wireless Communications and Mobile Computing, 4:55–74, 2004.

[36] A. Banchs, A. Azcorra, C. Garcia, and R. Cuevas. *Applications and Challenges of the 802.11e EDCA Mechanism: An Experimental Study*. In IEEE Network, volume 19, July 2005.

[37] N. Ramos, D. Panigrahi, and S. Dey. *Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions*. In IEEE Network, volume 19, July 2005.

[38] E. P.E. and Østerbø O.N. *Analysis of QoS in WLAN*. In Telektronikk, volume 1, 2005.

[39] Y. Ge, J. C. Hou, and S. Choi. *An Analytic Study on Tuning Systems Parameters in IEEE 802.11e Enhanced Distributed Channel Access*. Accepted for publication in Computer Networks Journal, December 2005.

[40] R. Bagrodia, R. Meyer, M. Takai, Y. an Chen, X. Zeng, J. Martin, and H. Y. Song. *PARSEC: A Parallel Simulation Environment for Complex Systems.* IEEE Computer, 31(10):77–85, October 1998.

[41] D. Gao, J. Cai, and K. N. Ngan. *Admission Control in IEEE 802.11e Wireless LANs*. In IEEE Network, volume 19, July 2005.

[42] ATT Plan 2000 for expected traffic percentage distributions, Webpage - http://post.fts2k.gsa.gov/gsa docs/FCSWG 9311/fig4-20.gif.

Appendix A

List of Abbreviations and Acronyms

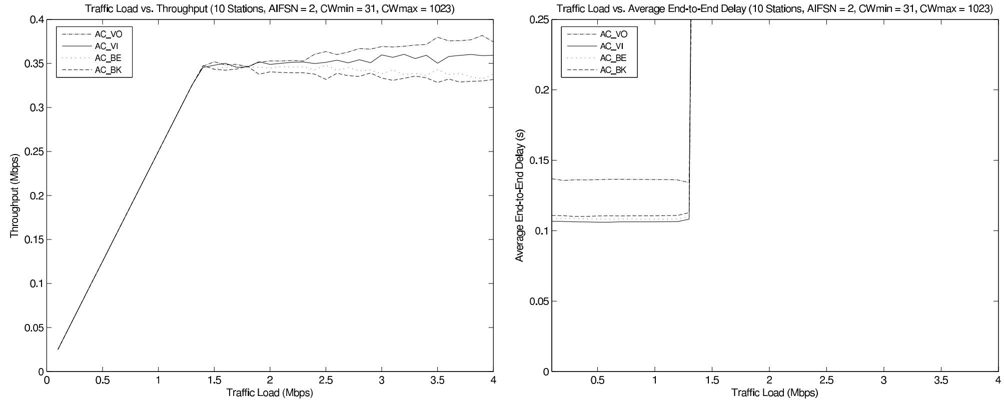
AC	Access Category
ACK	Acknowledgement
AIFS	Arbitration Inter-Frame Spacing
AP	Access Point
API	Application Programming Interface
CA	Collision Avoidance
CBR	Constant Bit Rate
CFB	Contention Free Bursting
CFP	Contention Free Period
CP	Contention Period
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
CW	Contention Window
CWmax	Contention Window Maximum
CWmin	Contention Window Minimum
DCF	Distributed Coordination Function
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
EDCA	Enhanced Distributed Channel Access
EDCAF	Enhanced Distributed Channel Access Function
ESS	Extended Service Set
FEC	Forward Error Correction
FHSS	Frequency Hopping Spread Spectrum
FTP	File Transfer Protocol
HC	Hybrid Coordinator
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HTTP	Hyper-Text Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IR	Infrared
ISM	Industrial, Scientific and Medical

MAC	Medium Access Control
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open Systems Interconnection
PARSEC	Parallel Simulation Environment for Complex Systems
PC	Point Coordinator
PCF	Point Coordination Function
PIFS	PCF Inter-Frame Spacing
QAP	Quality of Service Access Point
QSTA	Quality of Service Station
QBSS	Quality of Service Basic Service Set
QIBSS	Quality of Service Independent Basic Service Set
QoS	Quality of Service
RTS/CTS	Request To Send/Clear To Send
SIFS	Short Inter-Frame Spacing
TCP	Transmission Control Protocol
TID	Traffic Identifier
TXOP	Transmission Opportunity
UDP	User Datagram Protocol
UP	User Priority
VoIP	Voice over IP
VR	Virtual Reality
WLAN	Wireless Local Area Network

Appendix B

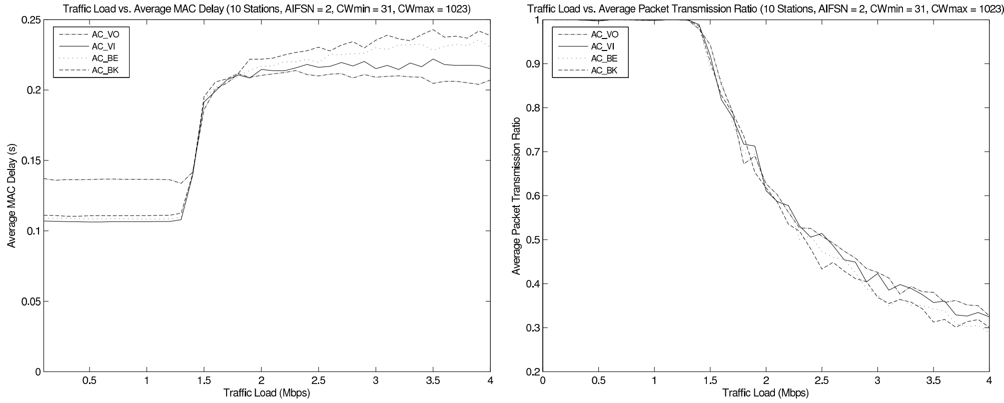
Additional Results

Results for the last of the three additional cases for Scenario 2 in evaluation part 1 (See Section 7.3.2) are presented in Figures B.1 and B.2.



(a) Average throughput.

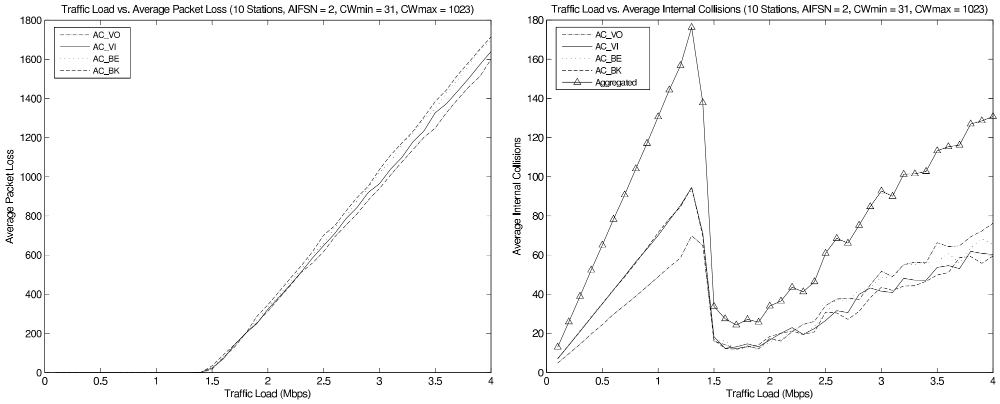
(b) Average end-to-end delay.



(c) Average MAC delay.

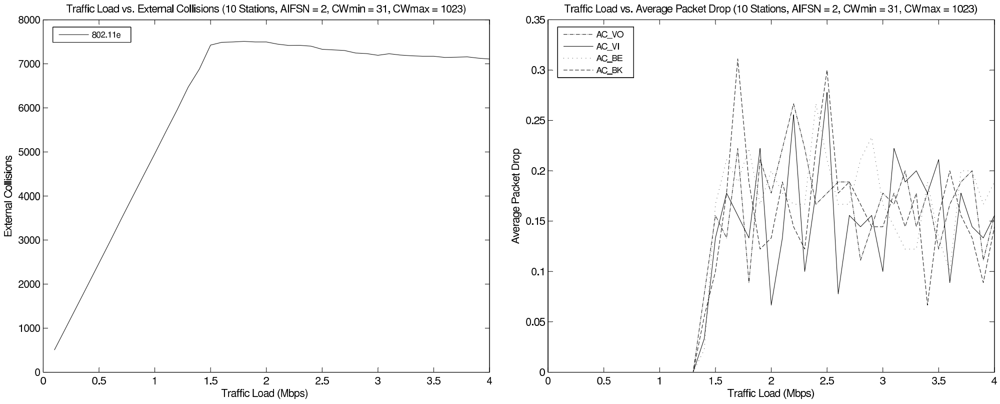
(d) Average packet transmission ratio.

Figure B.1: Results for Evaluation Part 1, Scenario 2, no AIFS and CWmin/CWmax differentiation.



(a) Average packet loss.

(b) Average internal collisions.



(c) External collisions.

(d) Average packet drop.

Figure B.2: Results for Evaluation Part 1, Scenario 2, no AIFS and CWmin/CWmax differentiation.