Department of Electrical and Computer Systems Engineering

Technical Report MECSE-5-2004

A Survey of IEEE 802.11 MAC Mechanisms for Quality of Service (QoS) in Wireless Local Area Networks (WLANs)

D. Pham, A. Sekercioglu and G. Egan

MONASH UNIVERSITY

A Survey of IEEE 802.11 MAC Mechanisms for Quality of Service (QoS) in Wireless Local Area Networks (WLANs)

Minh-Duc Pham Y. Ahmet Şekercioğlu Gregory K. Egan

March 29, 2004

Contents

List of Figures

List of Tables

Abstract

Wireless local area network (WLAN) is becoming the edge network of choice in today's network infrastructure. The IEEE 802.11, which involves the Medium Access Control (MAC) and physical (PHY) layers, is so far the most widely used WLAN standard. However, it does not support Quality of Service (QoS) requirements of an increasing number of multimedia services being used on the networks. Therefore, a number of techniques for QoS support at MAC layer have been proposed. This article reviews some of the presented methods and explains the principles, the advantages and drawbacks of each method. From this survey, it seems that selecting the proper QoS mechanism and the best set of MAC parameters still remains a topic requiring more research.

Keywords

IEEE 802.11, IEEE 802.11e, MAC (Medium Access Control), QoS (Quality of Service), WLAN (Wireless Local Area Network), DCF (Distributed Coordination Function), PCF (Point Coordination Functions), EDCF (Enhanced Distributed Coordination Function), HCF (Hybrid Coordination Function).

List of Abbreviations

AC	Access Category
ACK	Acknowledgment
ADB	Age Dependent Backoff
AEDCF	Adaptive Enhanced DCF
AI	Association Information
AID	Association ID
AIFS	Arbitration Inter-Frame Space
AIFSD	AIFS Duration
AP	Access Point
AR	Association Request
ARME	Assured Rate MAC Extension
BC	Backoff Counter
BCUT	Backoff Counter Update Time
BEB	Binary Exponential Backoff
BI	Backoff Interval
BSS	Basic Service Set
CA	Collision Avoidance
CAP	Controlled Access Period
CF	Contention Free
CFACK	Contention Free Acknowledgment
CFB	Contention Free Burst
CFP	Contention Free Period
CFPRI	Contention Free Period Repetition Interval
CoS	Class of Service
CP	Contention Period

CSMA	Carrier Sense Multiple Access
CTS	Clear To Send
CW	Contention Window
CW_{min}	Contention Window Minimum
CW_{max}	Contention Window Maximum
DBRP	Distributed Bandwidth Reservation Protocol
DC	Deficit Counter
DCF	Distributed Coordination Function
DCF/SC	DCF/Shortened Contention Window
DDRR	Distributed Deficit Round Robin
DFS	Distributed Fair Scheduling
DIFS	DCF Inter-Frame Space
DLC	Data Link Control
DRR	Deficit Round Robin
DS	Distribution System
DU	Data Unit
DWFQ	Distributed Weighted Fair Queuing
EDCF	Enhanced DCF
EIED	Exponential Increase Exponential Decrease
FIFO	First In First Out
FTP	File Transfer Protocol
HAD	Hybrid Activity Detection
HCF	Hybrid Coordination Function
HIPERLAN	High Performance LAN
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFS	Inter-Frame Space
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
JW	Jamming Window
LAN	Local Area Network
LT	Life Time
MAC	Medium Access Control
M-DCF	Modified DCF
MF	Multiplicator Factor
M-PCF	Modified PCF
MPDU	MAC Protocol Data Unit
MPEG2	Motion Picture Experts Group 2
MPX	Motion Picture Extreme Compressed Movies
MS	Mobile Station
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
NRTDU	Non-Real-Time Data Unit
NTS	Nothing To Send
OFDM	Orthogonal Frequency Division Multiplexing
PC	Point Coordinator
PCF	Point Coordination Function
PF	Persistent/Persistence/Priority Factor
PHY	Physical layer
PI	Polling Information
PIFS	PCF Inter-Frame Space
P-MAC	Priority-based MAC
PVT	Packet Virtual Time

QoS	Quality of Service
QSTA	QoS Station
RCG	Reservation Cycle Generator
RF	Reservation Frame
RR	Round Robin
RTDU	Real-Time Data Unit
RTS	Request To Send
SAD	Statistical Activity Detection
SCFQ	Self-Clocked Fair Queuing
SD	Slow Decrease
SID	Sequence ID
S-PCF	Slotted PCF
SS	Slave Scheduler
STA	Station
SVT	Station Virtual Time
TBTT	Target Beacon Transmission Time
TC	Traffic Category
TCMA	Tiered Contention Multiple Access
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TOS	Type of Service
TXOP	Transmission Opportunity
UAT	Urgency Arbitration Time
UDP	User Datagram Protocol
UNII	Unlicensed National Information Infrastructure
UP	User Priority
VDCF	Virtual DCF
WLAN	Wireless Local Area Network

1 Introduction

Over the past few years, many multimedia applications such as voice telephony, streaming audio and video on demand have become increasingly popular under the Internet Protocol (IP) networks. However, IP was not designed to support multimedia services with stringent requirements on minimum data rate, delay and jitter. The desire to use these multimedia applications over IP networks has led to the need for enhancing the existing networks with end-to-end QoS support. QoS is the ability to offer some persistent data transmission over the network with different treatment for different traffic classes [36]. The Internet Engineering Task Force (IETF) is currently working on service differentiation at the IP layer to support various traffic classes. However, for optimal result, there is a need for QoS support from lower layers [52], especially Data Link Control (DLC) layer. The IEEE 802.11 WLAN standard, which covers the MAC sublayer of the data link layer and the physical layer, is gaining growing popularity, acceptance and is being deployed everywhere, such as hot spots in coffee shops, hotels and airports. However, the 802.11 standard does not currently provide QoS support for multimedia applications. As a result, numerous proposals for QoS support at the MAC layer have been proposed by an active research community. In this article, we present a survey of research efforts focusing on IEEE 802.11 MAC QoS mechanisms for WLANs. The article is organized as follows: The legacy IEEE 802.11 MAC is described in Section 2, while its 802.11e enhancements are presented in Section 3. Section 4 reviews the distributed mechanisms for QoS support based on Distributed Coordination Function (DCF). The centralized mechanisms for QoS support based on Point Coordination Function (PCF) are evaluated in Section 5. The article concludes with a summary and conclusion in Section 6.

2 Legacy IEEE 802.11 MAC

An IEEE 802.11 WLAN can operate in two modes: ad-hoc or infrastructure. In ad-hoc mode, mobile stations (MSs) can directly communicate with each other. In the infrastructure mode, an access point (AP) allows the MSs to communicate with each other, and also connects them to a distribution system (DS). In IEEE parlance, the cluster of communicating MSs (and the AP) is called Basic Service Set (BSS). In the infrastructure mode APs, through the DS, link the BSSs together. The traditional IEEE 802.11 MAC [1] contains algorithms to arbitrate media access (called coordination functions), which control the operation of MSs within a BSS to decide when a MS is allowed to send or receive frames via the wireless medium. Two coordination functions are defined, which include the compulsory DCF based on Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) and the optional PCF based on polling. The DCF may be used in both ad-hoc or infrastructure networks, while the PCF can only be used in the infrastructure networks. The DCF and PCF modes of operation are multiplexed in a superframe, which is repeated over time. In a superframe, a Contention Free Period (CFP), in which PCF is used for medium access, is followed by a Contention Period (CP), in which DCF is used. A diagram of two 802.11 superframes is shown in Figure 1. All the parameters such as Short Inter-Frame Space (SIFS), PCF Inter-Frame Space (PIFS), DCF Inter-Frame Space (DIFS), SlotTime, Contention Window Minimum (CW_{min}), Contention Window Maximum (CW_{max}) depend on the selection of the underlaying physical layer. The IEEE 802.11 standard consists of a family of standards. The original 802.11 standard [1] provides data rates up to 2 Mb/s at 2.4 GHz industrial, scientific, and medical (ISM) band. Its enhanced version IEEE 802.11b [3] achieves the data rates up to 11 Mb/s in the ISM band. Another version, IEEE 802.11a [2] extends the data rates up to 54 Mb/s, using orthogonal frequency division multiplexing (OFDM) at 5 GHz unlicensed national information infrastructure (UNII) band. However, regardless of the selected physical layer, following equalities are always valid: $PIFS = SIFS + SlotTime$ and $DIFS = PIFS + SlotTime$. The operation of DCF and PCF are described in details in Section 2.1 and Section 2.2, respectively.

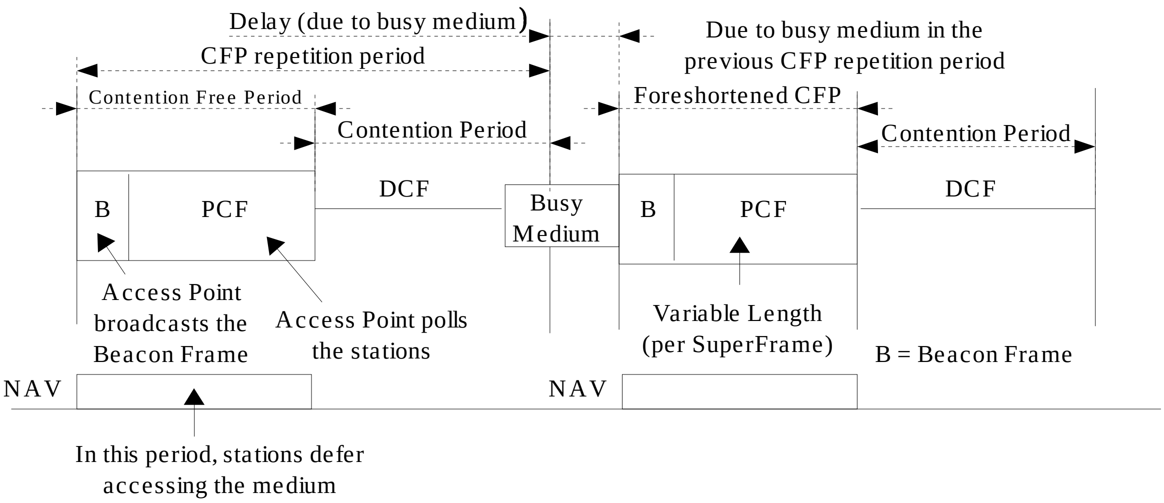

Figure 1: Two 802.11 superframes

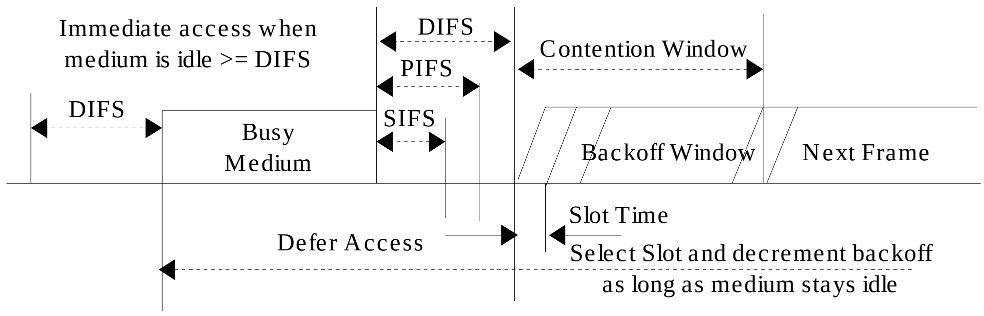

Figure 2: 802.11 DCF access method

2.1 Distributed Coordination Function

The access method of DCF is shown in Figure 2. Each MS has a First In First Out (FIFO) transmission queue. A station can sense whether the medium is busy (a station is transmitting) or idle (there is no transmission). When a frame (or in ISO terms, a MAC Service Data Unit (MSDU), which is the unit of data entering the MAC layer from higher layer) reaches the front of a station's transmission queue, the station checks the state of the medium. If the medium is busy, the station waits until the medium is idle. After that, it waits for a duration of DIFS. If the medium is still idle during the DIFS interval, the station initiates the backoff procedure. This is the Collision Avoidance (CA) mechanism to reduce the probability of frame collisions if two or more stations perceive the medium idle at the same time. Backoff Counter (BC) is chosen as a random integer in the uniformly distributed interval [0, CW], where CW is the current contention window of this station. The initial value of CW is CW_{min}. The Backoff Interval (BI) is the backoff time of the station, which is equal the BC multiplied by the SlotTime. If the medium is idle for SlotTime, the BC is decremented by 1. When BC is 0, the station transmits the frame. If during the backoff procedure, the medium becomes busy (another station finishes its backoff procedure and transmits a frame), the BC is suspended. This station will have to wait for the medium to become idle again, wait for extra DIFS duration before continuing the backoff procedure with the suspended BC value. If the transmitted frame is received successfully, the receiving station waits for a duration of SIFS and sends back an Acknowledgment (ACK) frame. The sending station, upon receiving this ACK frame, resets its CW to CW_{min}, defers for DIFS and initiates a post backoff procedure, even if there is no frame waiting in the transmission queue. The post backoff procedure guarantees that there is at least one BI between transmission of two successive MSDUs. If the sending station does not receive the ACK frame after some time (either due to two or more stations finish backoff process at the same time and the transmitted frames collide, the transmitted frame is lost or the ACK frame is lost), it assumes that the frame transmission is unsuccessful. It increases the CW to the new value of $2 \times (CW + 1) - 1$, with upper limit of CW_{max}, waits for the medium to become idle again, waits for DIFS and performs backoff process with this new value of CW. In case when an MSDU reaches the front of a station's transmission queue and the station is performing DIFS deferral or post backoff process, the frame is transmitted when the backoff procedure completes successfully. If the MSDU reaches the front of the queue and the backoff procedure has finished and the medium has been idle for at least DIFS, the frame is transmitted instantly. The MSDU can be of any size up to 2304 bytes. A large MSDU can be fragmented into smaller frames. To solve the hidden station problem, an optional Request To Send/Clear To Send (RTS/CTS) mechanism is introduced. Instead of transmitting a DATA frame after gaining access to the medium, a station sends a short RTS frame. The receiving station waits for SIFS and replies with a short CTS frame. Upon receiving the CTS frame, the sending station waits for SIFS and transmits the DATA frame. The RTS and CTS frames stores information about transmission duration of the DATA frame and are received by adjacent stations, which could be hidden from sending and/or receiving stations. These adjacent stations set their Network Allocation Vector

(NAV) and restrain from transmitting for this duration, thus avoid collision.

2.2 Point Coordination Function

The PCF is only applicable for infrastructure WLAN, which consists of an AP and a number of stations associated with it. The operation of stations is controlled by a station called Point Coordinator (PC). At the start of a superframe, the PC, which is usually located at the AP, generates a beacon frame periodically, whether PCF is active or not. Other stations know the time when the next beacon frame will arrive, this time is designated as Target Beacon Transmission Time (TBTT). The beacon frame transports the management information to other stations to synchronize local timers in stations and provide protocol related parameters. Stations can use the information in the beacon frame to associate with the PC during CP to have their transmissions scheduled in the next CFP. After the beacon frame, the PC waits for SIFS and polls the stations registered in its polling list by sending a CF-Poll frame to a station. If the PC has data or acknowledgment destined for this station, it piggybacks the CF-Poll frame on the DATA or CF-Ack frames, respectively. When a station receives the CF-Poll frame, it waits for SIFS and replies with a DATA, CF-Ack, DATA + CF-Ack or NULL frame (the NULL frame is transmitted when there is no data to send and no pending acknowledgment). If the polled station does not reply after PIFS, the PC polls the next station. This is repeated until the CFP expires. The PC sends a CF-End frame to indicate the end of the CFP. A typical 802.11 superframe is shown in Figure 3.

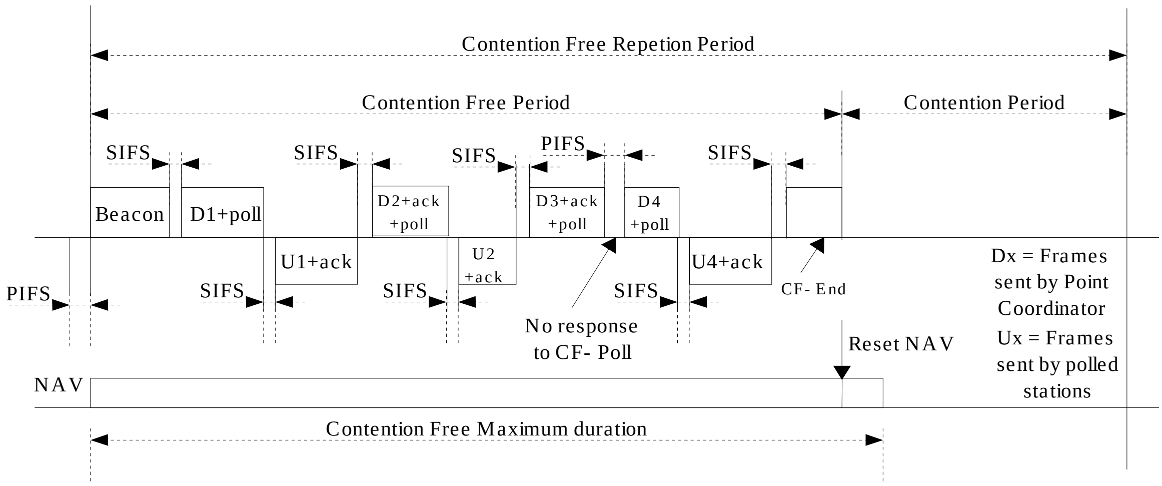

Figure 3: A typical 802.11 superframe

3 IEEE 802.11e MAC Enhancements

To support QoS, the IEEE 802.11 Task Group E propose enhancements to the above standard in the IEEE 802.11e draft [4]. Within 802.11e, the Enhanced Distributed Coordination Function (EDCF) is an enhancement of the DCF in the legacy 802.11 while a new medium access mechanism called Hybrid Coordination Function (HCF) is introduced. The EDCF is part of the HCF. The HCF combines aspects of both DCF and PCF with the multiplex of CFP and CP in a 802.11e superframe, which is repeated over time. EDCF is used only in CP, while HCF may be used in both CP and CFP. The EDCF and HCF are described in Section 3.1 and Section 3.2, respectively.

3.1 Enhanced Distributed Coordination Function

Support for QoS is provided by the introduction of Traffic Categories (TCs). Each station has up to four Access Categories (ACs) to support up to eight User Priorities (UPs). One or more UPs are assigned to one AC. The mapping of UPs to ACs is shown in Table 1.

Priority	Access Category	Designation
1	0	Best-effort
2	0	Best-effort
0	0	Best-effort
3	1	Video probe
4	2	Video
5	2	Video
6	3	Voice
7	3	Voice

Table 1: Mapping from Priority to Access Category in EDCF

In Table 1, the priorities are sorted in ascending order with the exception of relative priority 0 being placed between priorities 2 and 3. This is originated from IEEE 802.11d bridge specification [5]. Each AC is a variant of DCF with its own parameters AIFSD[AC], CW[AC], $CW_{min}[AC]$ and $CW_{max}[AC]$ instead of DIFS, CW, CW_{min} and CW_{max}. The AIFSD[AC], which is usually referred to as Arbitration Inter-Frame Space (AIFS), is calculated as $AIFSD[AC] = SIFS + AIFS[AC] \times SlotTime$, where AIFS[AC] is an integer greater than zero. Furthermore, the BC is chosen from [1, 1+CW[AC]] rather than [0, CW] as in the DCF. An AC with higher priority is assigned smaller CW_{min} and CW_{max} values and/or shorter AIFS. The IEEE 802.11e EDCF access method is shown in Figure 4.

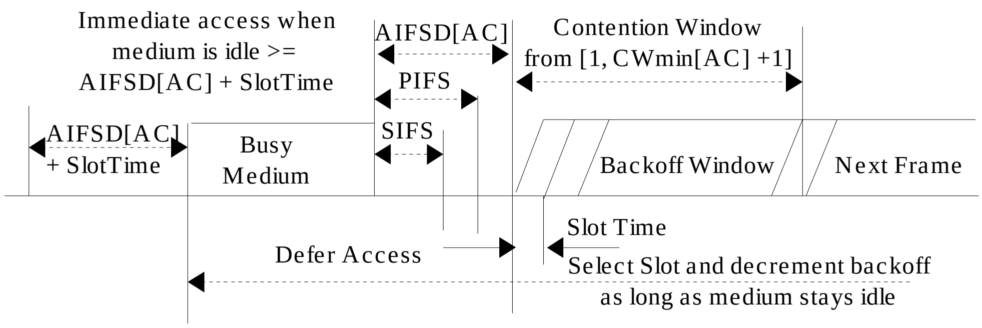

Figure 4: 802.11e EDCF access method

Each AC within a station behaves as a virtual station and contends for Transmission Opportunities (TXOPs), a time interval when it can initiate transmissions, using its own parameters and its own BC. If the BCs of two or more ACs within a station reach zero at the same time, the frame from the AC with highest priority receives the TXOP. ACs with lower priority behave as if there were an external collision in the medium and perform backoff process with increased CW values. The four ACs of a 802.11e station and one priority of a legacy 802.11 station and is shown in Figure 5. The AP determines and broadcasts the parameters for each AC plus the limit of a TXOP interval for each AC TXOPLimit[AC] in beacon frames periodically. During a TXOP, multiple MAC Protocol Data Units (MPDUs) from the same AC in a station can be transmitted with an SIFS time interval between an ACK and the next frame transmission. This multiple transmission of MPDUs is called Contention Free Burst (CFB).

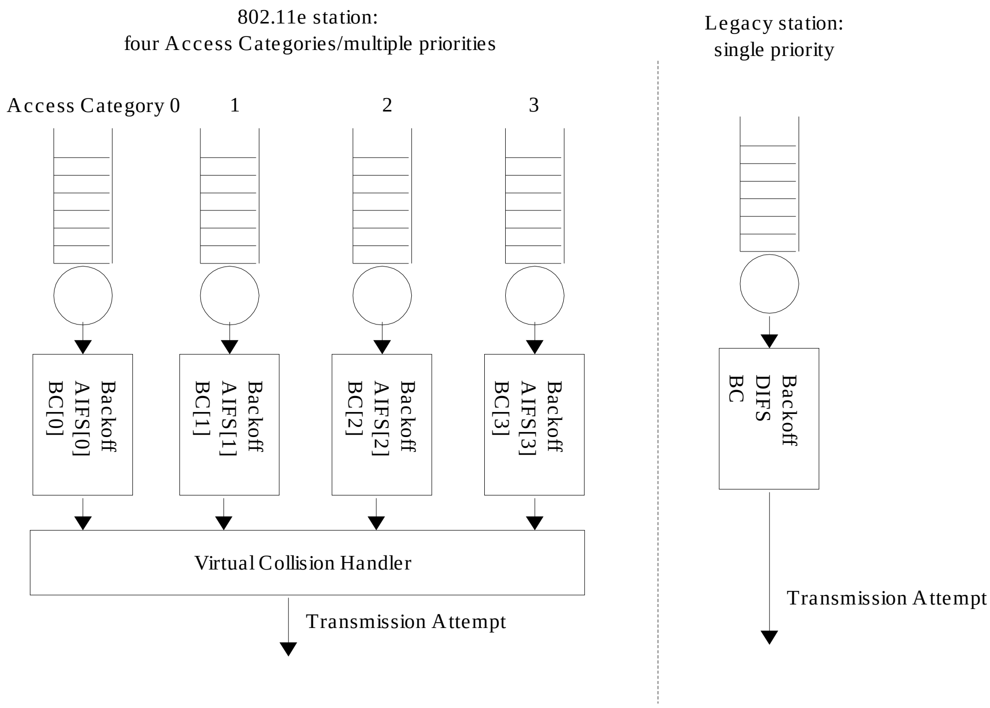

Figure 5: 802.11e EDCF queues vs 802.11 DCF queue per station

3.2 Hybrid Coordination Function

The HC, which is usually located at the AP, may assign TXOPs to itself or other stations after the medium is idle for PIFS without any backoff procedure, therefore it has higher priority than ACs since ACs, using (E)DCF cannot contend for medium access at least after DIFS idle duration. The HC traffic delivery and TXOP assignment may be scheduled during both CP and CFP. The beacon frame is broadcast by the HC at the beginning of each superframe. During CFP, only HC can allow access to the medium by giving TXOPs to stations using QoS CF-Poll frames, which specify the starting time and maximum duration of each TXOP. The CFP ends if the duration reaches the time specified in the beacon frame or the HC sends a CF-End frame. During CP, each TXOP begins when a station gains access to the medium using EDCF or when this station receives a QoS CF-Poll frame from the HC. A Controlled Access Period (CAP) is composed of several intervals within one CP when short bursts of frames are transmitted using polling. A typical 802.11e superframe is shown in Figure 6.

4 Distributed Approaches Based on DCF

In the distributed approach, each station in a WLAN determines to access the medium without control of a particular station. Service differentiation between traffic classes is based on differentiation of the time the traffic has to wait before transmission. Two main parameters that decide the waiting time of traffic are Inter-Frame Space (IFS) and CW. The distributed approach may be divided further into priority-based approach and fair scheduling approach. The priority-based approach and fair scheduling approach are described in Section 4.1 and Section 4.2, respectively.

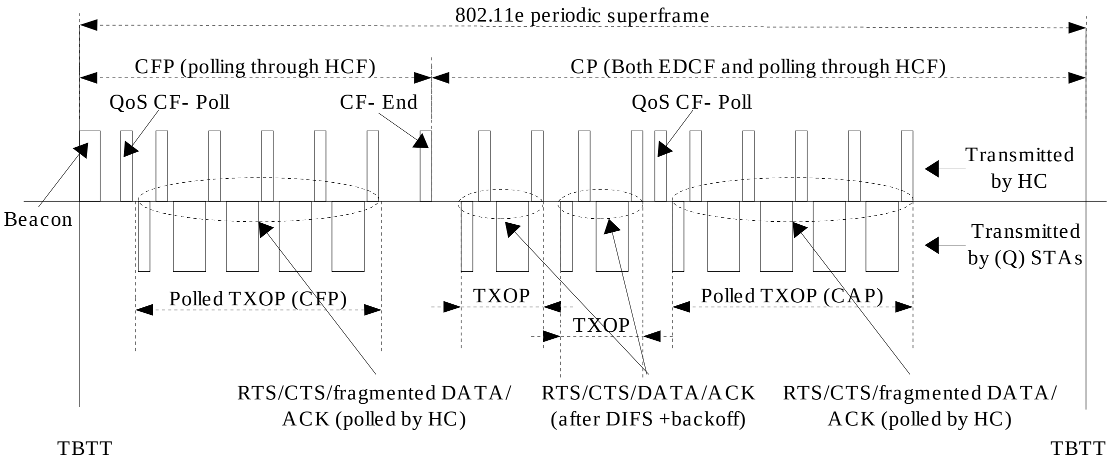

Figure 6: A typical 802.11e superframe

4.1 Approaches based on priority

In the priority-based approach, the priority is mapped into medium access. Traffic with higher priority is assigned smaller values for IFS and/or CW (leading to smaller BI value), therefore low-priority has to wait longer than high-priority before transmission.

4.1.1 Approaches based on IFS

Aad *et al.* [7] recommend a scheme in which higher priority $j + 1$ and lower priority j have IFS values of $DIFS_{j+1}$ and $DIFS_j$, respectively, such that $DIFS_{j+1} < DIFS_j$. The maximum random range RR_{j+1} of priority $j + 1$ is defined as the maximum BI of that priority. If the strict condition $RR_{j+1} < DIFS_j - DIFS_{j+1}$ is satisfied, then all packets of priority $j + 1$ have been transmitted before any packet of priority j is transmitted. In less stringent condition, $RR_{j+1} > DIFS_j - DIFS_{j+1}$, a packet which could not access the medium the first time may have its priority decreased in the subsequent attempts. Simulations were carried out and the results show that the method does not change the system efficiency, with data rate sums remain the same. The method works well for both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) flows with more significant effect on UDP flows than on TCP flows. It also works in noisy environment and keeps the same stability of the system.

Benveniste [15] suggests a technique to differentiate services based on Urgency Arbitration Time (UAT), which is the time a station has to wait before a transmission attempt following a period when the medium is busy. AIFS and Backoff Counter Update Time (BCUT) are generalization of DIFS and SlotTime, respectively. Traffic with higher priority is assigned shorter AIFS and/or shorter BCUT values. The highest priority has $AIFS(high\_prio) = PIFS$ and a minimum backoff time of 1 in order to prevent conflict with medium access by centralized protocol PCF. Simulation of two traffic classes with $AIFS(high\_prio) = PIFS$, $AIFS(low\_prio) = DIFS$, $CW(high\_prio) = [1, 32]$ and $CW(low\_prio) = [0, 31]$ was conducted. The outcomes reveal that the delay and jitter of high-priority traffic are decreased and under moderate load condition, the performance of low-priority traffic is also improved as compared with legacy DCF.

Deng *et al.* [19] propose a method to support two priorities. High and low priorities have to wait for the medium to be idle for PIFS and DIFS, respectively, before initiating backoff procedure. Simulation of video, voice and data traffic with priorities of 3, 2 and 0, and traffic ratios of 1 : 1 : 2, respectively, was performed. The results demonstrate that the approach, when combined with separation of random backoff time, can be used to support video, voice and data traffic in heavy load condition (say 90%). In heavy load condition, the highest priority video traffic uses most of the bandwidth (55%) and lower priorities use the remaining bandwidth, while in light load condition, the lower priority traffic has the required bandwidth. It is also illustrated that video and voice traffic have lower access delay and lower packet loss probability than in DCF, but data traffic has higher access delay and higher packet loss probability than in DCF.

4.1.2 Approaches based on CW

Separation of CW

Aad *et al.* [8] introduce a differentiation mechanism based on CW_{min} separation, in which higher priority traffic has lower CW_{min} value. Simulations of a WLAN consisting of an AP and three stations with CW_{min} values of 31, 35, 50 and 65, respectively, were conducted with both TCP and UDP flows. The results reveal that for the same set of CW_{min} values, the differentiation effect is more significant on UDP flows than on TCP flows. The per-flow differentiation is introduced, in which the AP sends back ACK packets with priorities proportional to priorities of the destinations. In other words, the AP waits for a period of time which is proportional to the delay from a destination before transmitting an ACK packet to that destination.

Barry *et al.* [12, 49] recommend using different values of CW_{min} and CW_{max} for different priorities, in which higher priority has lower CW_{min} and CW_{max} values than those of lower priority. Simulations of high-priority traffic with CW_{min} between [8,32] and $CW_{max} = 64$, and low-priority traffic with CW_{min} between [32,128] and $CW_{max} = 1024$ were performed. The outcomes show that the high-priority and low-priority traffic undergo different delay.

Deng *et al.* [19] propose a scheme based on separation of CW. Originally, the random BI is uniformly distributed between $[0, 2^{2+i} - 1]$, in which i is the number of times the station attempted transmission of the packet. In Deng's scheme, the high and low priorities have random BI values uniformly distributed in intervals $[0, 2^{2+i}/2 - 1]$ and $[2^{2+i}/2, 2^{2+i} - 1]$, respectively. This approach can be combined with the approach based on IFS mentioned in the last paragraph of Section 4.1.1. The simulation results reveal some improvement in delay and jitter for high priorities (voice and video).

Xiaohui *et al.* [53] present the Modified DCF (M-DCF) scheme, which uses different values of CW_{min} and CW_{max} for service differentiation. Simulations of ad-hoc WLAN with 10 data stations and between 10 and 35 voice stations were performed. Voice service had $CW_{min} = 7$ and $CW_{max} = 127$, while data service had $CW_{min} = 15$ and $CW_{max} = 255$. The outcomes illustrate that M-DCF decreases the total packet dropping probability and the dropping probability of voice packets as well as reduces the contention delay of both voice and data packets as compared with DCF.

Different Priority/Persistent/Persistence Factor (PF)

Aad *et al.* [6, 7] propose a method based on the backoff increase function. In the original DCF, the CW is multiplied by a Priority Factor (PF) of 2 after each collision. In Aad's method, a higher priority traffic has a lower PF P_j. Simulations of three priorities with PF values of 2, 6 and 8 were conducted. The results demonstrate that this method works well with UDP flows, but does not work well with TCP flows or in noisy environment. The efficiency is not lost but the stability of the system is decreased.

Benveniste [15] recommends a technique based on Persistent Factor (PF). After each collision, the CW is multiplied by a PF. Higher priority traffic has lower value for PF. For time sensitive applications and with capability for congestion estimation, PF value should be less than 1. Otherwise, a value between 1 and 2 should be chosen. Simulation of traffic with two priorities and $AIFS(high\_prio) = PIFS$, $AIFS(low\_prio) = DIFS$, $PF(high\_prio) = 0.5$ and $PF(low\_prio) = 2$ was carried out. The outcomes show that the high-priority traffic performance is improved without any significant effect on low-priority traffic. Furthermore, the delay and jitter is less than 10 ms, which could not be obtained with differentiation based on IFS alone.

Song *et al.* [46] introduce a scheme called Exponential Increase Exponential Decrease (EIED), in which after a collision or a successful transmission, the CW is multiplied or divided by factor

r_I and r_D, respectively. Simulations of 5, 10, 40 and 60 stations with packet arrival rate between 10 to 160 packets/s and packet length of 1024 bytes were performed using the Binary Exponential Backoff (BEB) and EIED with $[r_I, r_D]$ = [2, $2^{1/8}$], [2, $2^{1/4}$], [$2\sqrt{2}$, $2\sqrt{2}$], [2, 2]. The results reveal that the delay of EIED is smaller than that of BEB for all packet arrival rates and all number of stations. The throughput of EIED is the same as throughput of BEB when the packet arrival rate is small (less than 80 packets/s) and higher than throughput of BEB when the packet arrival rate is high.

4.1.3 Approaches based on combination of IFS and CW

Wong *et al.* [52] present a scheme named Age Dependent Backoff (ADB), which is an improvement of EDCF. For each TC, after each collision, the CW[TC] is multiplied by a Persistence Factor (PF) given by the formula

$$PF[TC] = \frac{-2}{LT[TC]} \times Age + 2$$

$LT[TC]$ and Age are the packet lifetime (LT) and age of the packet in transmission queue, respectively. Packets with $Age > LT$ are discarded since they would be obsolete by the time they get to the recipient. $PF[TC]$ is between 1 and 2 in the first half of packet lifetime and is between 0 and 1 in the second half. $CW[TC]$ is always less than $CW_{max}[TC]$ but could be less than $CW_{min}[TC]$ to differentiate between a retransmitted packet and a newly arrived one. Simulations of an ad-hoc network with 10 voice stations, 4 video stations and a number of File Transfer Protocol (FTP) stations (between 5 and 25), in which half of the FTP stations using DCF and the other half using EDCF, were conducted. LTs of voice and video packets were set to 25 ms and 75 ms, respectively. DCF had $DIFS = 50\mu s$, $CW_{min} = 31$, $CW_{max} = 1023$. For voice service, $AIFS[1](Voice) = 50\mu s$, $CW_{min}[1] = 7$, $CW_{max}[1] = 31$. The video service had $AIFS[2](Video) = 70\mu s$, $CW_{min}[2] = 15$, $CW_{max}[2] = 63$, while data service had $AIFS[3](Data) = 110\mu s$, $CW_{min}[3] = 15$, $CW_{max}[3] = 255$. The results show that ADB improves the packet delay, jitter and drop rate of both voice and video traffic as compared with EDCF with fixed PF values of 2 and 1.5. At the same time, the best-effort FTP traffic does not suffer from starvation.

Romdhani *et al.* [40] introduce a method called Adaptive EDCF (AEDCF), in which CW is dynamically calculated after each successful transmission or collision according to the network conditions. After each successful transmission of a packet of class i, the CW is updated as

$$CW_{new}[i] = max(CW_{min}[i], CW_{old}[i] \times MF[i])$$

where higher priority has lower multiplicator factor $MF[i]$. If $MF[i]$ is fixed at 0.5, then the scheme is called Slow Decrease (SD) scheme. Each traffic class has a different multiplicator factor (MF), which should not exceed 0.8 as suggested by the authors.

$$MF[i] = min((1 + (i \times 2)) \times f^j_{avg}, 0.8)$$

The average collision rate at step j is

$$f^j_{avg} = (1 - \alpha) \times f^j_{curr} + \alpha \times f^{j-1}_{avg}$$

where α is the weight or smoothing factor. The average collision rate is calculated every T_{update} expressed in SlotTime. The estimated current collision rate is

$$f^j_{curr} = \frac{E(collision_j[p])}{E(data\_sent_j[p])}$$

where $E(collision_j[p])$ and $E(data\_sent_j[p])$ are the number of collision and the number of packets sent at station p in the period j, respectively. After each collision, CW of class i is updated as

$$CW_{new}[i] = min(CW_{max}[i], CW_{old}[i] \times PF[i])$$

where higher priority traffic has lower persistent factor $PF[i]$. Simulations of an ad-hoc WLAN with the number of stations between 2 and 44 (corresponding to load rate from 6.7% to 149%) were carried out. Each station had three service classes, with high, medium and low classes having PF values of 2, 4 and 5, respectively. $T_{update} = 5000 \times SlotTime$ and $\alpha = 0.8$. High, medium and low classes had AIFS values of $34\mu s$, $43\mu s$ and $52\mu s$, respectively. $[CW_{min}, CW_{max}]$ of high, medium and low classes were [5,200], [15,500] and [31,1023], respectively. The outcomes demonstrate that AEDCF has lower mean access delay than EDCF and SD schemes, and its delay is less than 10 ms. AEDCF and SD has higher goodput than EDCF. Furthermore, AEDCF has higher gain on goodput than SD, especially under high load condition. The medium utilization decreases when traffic load increases, but AEDCF has higher utilization than SD, which has higher utilization than EDCF. Of all the three schemes, AEDCF has lowest collision rate while EDCF has highest collision rate.

Kim *et al.* [29, 30, 31] propose a technique called DCF with Shorten Contention Window (DCF/SC), which provides support for three service classes, premium (low latency and jitter), assured (guaranteed bandwidth) and best-effort (no guarantee). The super period, which originally consists of PCF and DCF, is divided into two periods. In the first period, only premium service is allowed to access the medium with DCF/SC after sensing the medium idle for PIFS. In the second period, premium and assured services access the medium with DCF/SC and best-effort service access medium with DCF after medium idle time of DIFS. DCF/SC has shorter CW than original IEEE 802.11 DCF. Simulations of infrastructure WLAN with premium service such as voice, Motion Picture Extreme Compressed Movies (MPX) and Motion Picture Experts Group 2 (MPEG2), assured service (assured bulk data) and best-effort service (bulk data) were conducted. DCF/SC had CW values from [11, 17, 23, 29, 35, 41] while DCF had CW values from [16, 32, 64, 128, 256, 512, 1024]. The results show that the method increases utilization (defined as the ratio of total successful transmission time of pure data (excluding backoff time, preambles and header) to the total simulation time) and increases stream throughput (defined as the ratio of the number of successful packets to the generated packets) as compared with the DCF. It also decreases the latency and jitter of the real-time services.

Banchs *et al.* [11] suggest a scheme for differentiation of real-time and best-effort services. Real-time stations waits for medium to be idle for PIFS and generates elimination burst EB 1, whose duration is multiple of SlotTime. The length of EB 1 has the following distribution

$$P_{E1}(n) = \begin{cases} p_{E1}^{n-1} \times (1 - p_{E1}), 1 \leq n < m_{E1} \\ p_{E1}^{m_{E1}-1}, n = m_{E1} \end{cases}$$

where n is the length of EB 1 in SlotTime, p_{E1} is a probability between 0 and 1 and m_{E1} is the maximum length of EB 1 in SlotTime. If the station senses the medium busy after transmission of EB 1 or EB 2, it defers access to the medium until the medium is idle for PIFS. The station with the longest EB 1 or EB 2 gains access to the medium. If two stations have the same EB 1 and EB 2, collision occurs and would be detected. Data stations wait for the medium to be idle for DIFS and use backoff algorithm. Each station has a share value corresponding to QoS level of this station. The basic service has share value of 1 and higher service has share value greater than 1. For all stations, the ratio $w_i = \frac{r_i}{s_i}$ should be equal, where r_i and s_i are throughput and the share assigned to station i, respectively. r_i is updated after a packet transmission as follows

$$r_i^{new} = (1 - e^{-\Delta t_i/k}) \times \frac{l_i}{\Delta t_i} + e^{-\Delta t_i/k} \times r_i^{old}$$

where l_i and Δt_i are the length and inter-arrival time of the transmitted packet, and k is a constant. Each station calculates its own w_i and includes this in the transmitted packet's header. A station observes a packet and notices the value of w_i in this packet header. If the station's w_i is greater than this packet's w_i, the station increases its CW by a small amount, and vice versa. The algorithm for calculation of a station CW is

if $(w_{own} > w_{rcv})$ **then**

CW ← $(1 + \Delta_1) \times$ CW
else if (*queue_empty*) **then**
CW ← $(1 + \Delta_1) \times$ CW
else
CW ← $(1 - \Delta_1) \times$ CW
end if

where $CW_{Min802.11} \leq CW \leq CW_{Max802.11}$ and w_{own} is calculated by the station, w_{rcv} is the value of w_i in the header of observed packet, Δ_1 is calculated as

$$\Delta_1 = k \left| \frac{w_{own} - w_{rcv}}{w_{own} + w_{rcv}} \right|$$

where k is another constant, which is different from k in the formula for r_i^{new}. It is claimed that the residual collision rate (two or more stations collide after transmission of EB 1 and EB 2) is very small and only depends on p_{E1}, m_{E1} and m_{E2}, but not the number of contending stations.

Sheu *et al.* [41, 42] present a scheme called Distributed Bandwidth Reservation Protocol (DBRP) to support both real-time and non real-time traffic. Data stations use DCF: they wait for the medium to be idle for DIFS and start backoff algorithm with CW_{min} and CW_{max} values of 31 and 1023, respectively. Voice stations stay in one of three states: initial state where each station starts with, reservation state where voice stations contend to reserve access to the medium and transmission state where voice stations transmit periodically without contention. A voice station has an Sequence ID (SID) for access sequence and an Active Counter (AC) for the number of active voice stations. A voice station that wants to access the medium at time t senses the medium for the reservation frame (RF) in period $(t, t + D_{max}]$, where D_{max} is the maximum acceptable delay of voice packets. The RF stores the number of active voice stations (AN). If there is no RF frame, and if the medium is idle in the period $(t + D_{max}, t + D_{max} + PIFS]$, the voice station performs backoff algorithm with backoff time uniformly distributed between 0 and $3 \times SlotTime$. When the backoff time goes to zero, the station enters *Send_RTS procedure* and transmits RTS frame to destination. If there is no collision, the station would receive CTS frame, enters *Transmission State* and becomes the Reservation Cycle Generator (RCG), which is the first voice station in WLAN and has to generate RF frame periodically. It sets its SID and Active Counter to 1 and transmits RF frame and its voice packet. If there is RTS collision, the colliding voice station would use p-persistent scheme for retransmission of RTS, with probability p_p in the next time slot. If there is RF frame in period $(t, t + D_{max}]$, the voice stations enters *RF_received procedure*, sets its Active Counter as the value of AN in the RF frame (denoted as RF.AN). The voice station enters *Wait_to_content procedure* and waits for a duration of $WT = RF.AN \times T_{voice}$ before trying to access the medium. If the medium is idle for SlotTime during this period, a voice station has left and WT should be decreased by one T_{voice}. After WT, the voice station goes to backoff procedure to send a RTS frame. The condition $D_{max} \geq RP_{max} + T_{maxMPDU}$ is required, where RP_{max} is the sum of the maximum voice packets reservation period and the voice station contention period, $T_{maxMPDU}$ is the transmission time of a maximum MPDU. If there is no collision of RTS/CTS, the station increases its Active Counter by 1 and sets its SID to the content of Active Counter. The station sends the first packet and enters *Transmission State*. All other stations increase their Active Counter by 1 when they hear a CTS. If there is collision of RTS, the colliding stations use p-persistent scheme for contention resolution. After the last access that is over the boundary of RP_{max}, a voice station senses the medium idle for PIFS. If the voice station SID is 1, it is the RCG and sends the RF frame and the voice packet immediately. Other stations sense the medium, and if it is idle for SlotTime, the RCG has left and all stations decrease its SID and Active Counter by 1. The new RCG has to transmit the RF frame before its voice packet. When an RF frame is received, a voice station sets its *cd_timer* as $cd\_timer = (SID - 1) \times T_{voice}$. When the *cd_timer* reaches zero, the station transmits its packet.

Chen *et al.* [16] recommend a method to differentiate between real-time and non-real-time traffic based on IFS, CW_{min} and retransmission procedure of real-time packets. After the medium is idle for DIFS, stations with real-time packets for retransmission generate jamming noise. The station with longest jamming noise then transmits its packet. The length of jamming noise has a truncated geometric distribution. The probability of a jamming noise with length of f in unit of SlotTime is:

$$P_j(f) = \begin{cases} p_j^{f-1} \times (1 - p_j), 1 \leq f < JW \\ p_j^{JW-1}, f = JW \end{cases}$$

p_j is a probability value between 0 and 1, Jamming Window JW is the maximum jamming time defined as $JW = min(JW \mid p_j^{JW-1} \leq \frac{1}{N})$, where N is the number of stations that simultaneously generate jamming noise. Simulations of seventy non-real-time flows and a number of active real-time flows were performed using Chen's scheme, Tiered Contention Multiple Access (TCMA) [13, 14] and Virtual DCF (VDCF) [17, 18]. (CW_{min}, IFS) for real-time and non-real-time were (15, 40 μs) and (31, 50 μs), respectively. p_j was 0.35 and JW was 9. The results show that Chen's scheme has lower mean MAC delay, lower packet dropping rate, similar average jitter and higher channel utilization than TCMA. This scheme also has lower mean MAC delay, similar packet dropping rate, lower average jitter and lower channel utilization than VDCF.

Sobrinho *et al.* [43, 44, 45] propose a scheme called Blackburst. Define τ to be the maximum propagation delay between any two stations. There are three inter-frame spacings that are required for the access procedures of data and real-time stations, which must satisfy the conditions $t_{short} + 2 \times \tau < t_{med}$ and $t_{med} + 2 \times \tau < t_{long}$. A data station that wants to transmit data senses the medium for a period t_{long}. If the medium is idle, this station transmits data. Otherwise, it waits for the medium to be idle for t_{long} and enters the backoff procedure, with the BI $f_{data}(c)$ in unit of white slot t_{wslot} as

$$f_{data}(c) = rand[f_{data}(0) \times 2^c]$$

where $rand[a]$ returns a random number between 0 and $a - 1$, c is the number of collisions that the data packet has experienced and $f_{data}(0)$ is the initial CW. A real-time station that wants to transmit packets senses the medium for a period t_{med}. If the medium is idle, this station transmits a packet. Otherwise, it waits until the medium is idle for t_{med} again and enters the blackburst contention period. The station jams the medium with a black burst in unit of black slot t_{bslot}, whose duration is proportional to the time that this station has been waiting. After the transmission of the black burst, the station waits for t_{obs} to see if any station is transmitting a longer black burst. The winning station, which is the station that has been waiting the longest time, would get access to the medium and other stations would contend for the medium with blackburst in the next cycle. In every black burst contention period, there is only one unique winner. By ensuring that $2 \times \tau < t_{obs} < min(t_{bslot}, t_{med})$, a real-time station can always see if its blackburst is shorter than that of another and real-time stations do not attempt to access the medium during the observation interval. When a real-time station has access to the medium, it transmits a packet that must be at least t_{pkt} and schedules the next access instant to be t_{sch} in the future. The real-time stations have higher priority than the data stations, since no data station would perceive the medium idle for t_{long} until all real-time stations have access to the medium. The real-time stations get access to the medium in the round robin order and share the medium in a Time Division Multiple Access (TDMA) manner. After all the real-time stations have synchronized their transmissions, the black burst contention is only initiated again if some transmission of data stations disturbs the order.

4.1.4 Discussion

Aad's scheme [7] can support a large number of priorities with the proper selection of $DIFS_j$ for each priority. Benveniste's approach [15] of using BCUT for differentiation is not possible because the higher priority could not use a shorter BCUT as the SlotTime used in the original DCF is the minimum possible. Deng's method [19] uses only two IFS values, which are PIFS and DIFS, therefore it only allows for differentiation of two priorities. In order to support more than two priorities, more IFS values must be used. However, except PIFS, all other values of IFS are at least DIFS and therefore there can be only one higher priority than the legacy DCF traffic if differentiation is merely based on IFS values. Furthermore, the total time a station has to wait before gaining access to the medium is the sum of IFS and the random backoff time. Therefore, even if high-priority traffic has lower IFS value, it can still have higher total wait time than low-priority traffic and loses the medium contention to low-priority traffic.

Although Barry's, Deng's and Xiaohui's methods [12, 19, 53] only provide service differentiation for two priorities, the same principle can be applied to support more priorities. Song's scheme [46] shows improvement in delay and throughput compared with DCF, however it does not introduce differentiation of services. Aad's methods [6, 7] based on IFS or backoff increase function does not work well for TCP flows, since the AP always uses its own priority to send back TCP-ACKs to different stations. For example, if two stations waits for t_1 and t_2 before transmitting a packet on average, then the data rate ratio is t_2/t_1. For TCP flow, the ACK produces additional delay t_0 and the data rate becomes $(t_2 + t_0)/(t_1 + t_0)$. If the AP is slow, t_0 is high and ratio $(t_2 + t_0)/(t_1 + t_0)$ is very much different from the desired ratio t_2/t_1 [8]. In using different CW_{min} and/or CW_{max} values for different priorities, low-priority traffic has to produce longer backoff time even if there is no high-priority traffic, leading to longer delay. Choosing a random BI in the CW range produces unpredictable variations in delay and throughput, which is undesirable for time-sensitive traffic. With regard to the PF, if PF value is greater than 1, after each collision, the CW is increased and the collided packet has to wait longer before being retransmitted, resulting in longer delay and higher jitter, which is not desirable for real-time packets. However, if PF value is less than 1, CW is reduced, and under the heavy load condition, congestion may increase leading to more collisions.

In Sheu's method [41, 42], $DIFS = PIFS + 4 \times SlotTime$, which is inconsistent with the current standard, in which $DIFS = PIFS + SlotTime$. Sobrinho's method [43] can provide guarantee on delay. However, it is only optimized for isochronous traffic and therefore can be a drawback for variable rate traffic [49]. Bandwidth is also wasted by sending bursts for each packet to access the medium [42]. The approach using combination of IFS and CW can support more priorities than each of the IFS or CW approach alone. However, the values of IFS, CW and BI are deterministic for each priority once chosen. This may lead to unfair medium access since high priorities tend to seize the medium and low priorities may suffer starvation. Romdhani's and Wong's methods [40, 52] calculate the CW dynamically based on network conditions, and therefore can adapt better to traffic load and can avoid starvation of low-priority traffic.

4.2 Approaches based on fair scheduling

Fair scheduling approach guarantees that traffic in each class has a fair chance for transmission. The approach aims to ensure that the bandwidth given to traffic flows is proportional to their weights.

4.2.1 Matching priority to IFS

Pattara-Aukom *et al.* [37] propose a method called Distributed Deficit Round Robin (DDRR) based on Deficit Round Robin (DRR) scheduling. Traffic is categorized into classes. Traffic class i has a *service quantum* Q bits every T_i seconds, such that Q/T_i is equal the desired throughput of class i. The Deficit Counter (DC) of traffic class i, at MS j at any given time is DC_i^j and is proportional to the bandwidth available to this traffic class at that time. DC_i^j is increased Q bits every T_i seconds and is decreased by the size of the frame whenever a frame is transmitted. At time t, IFS for traffic class i at MS j, $IFS_i^j(t)$ is calculated as

$$IFS_i^j(t) = DIFS - \alpha \times \frac{DC_i^j(t)}{Q} \times random(1.0, \beta)$$

$$DC_i^j(t) = DC_i^j(t') + \frac{Q}{T_i} \times (t - t')$$

$$DC_i^j(t) = DC_i^j(t) - Frame\_Size(t)$$

α is a scaling factor, $\beta > 1$ and $random(1.0, \beta)$ returns a random number uniformly distributed between 1 and β.

$$0 \leq DC_i^j \leq \frac{DIFS - PIFS}{\alpha} \times Q$$

If $DC_i^j < 0$, then traffic class i has to wait until DC_i^j is greater than 0 again before next transmission. The above equations leads to

$$PIFS \leq IFS_i^j \leq DIFS$$

From this last equation, DDRR is backward compatible with IEEE 802.11 When a MS senses the medium is idle, it waits for IFS_i before transmitting a traffic class i frame. If the medium is busy, the MS waits until it becomes idle again, waits for additional time IFS_i (as calculated at this time) and then transmits the frame. There is no backoff procedure in this scheme. The right to access the medium depends on DC_i^j but does not depend on the required throughput, resulting in fair share of the medium between different traffic classes.

4.2.2 Matching priority to CW

Banchs *et al.* [9] present an approach called Distributed Weighted Fair Queuing (DWFQ), in which for all stations, the ratio $L_i = \frac{r_i}{W_i}$ should be equal, where r_i and W_i are throughput and the weight assigned to station i, respectively. The assumption is all packets at a station are from one flow. IEEE 802.11 stations have default weight of 1, and other weight values must be greater than or equal to 1, which means better than or best-effort service. r_i is updated after a packet transmission as follows

$$r_i^{new} = (1 - e^{-t_i/K}) \times \frac{l_i}{t_i} + e^{-t_i/K} \times r_i^{old}$$

where l_i and t_i are the length and inter-arrival time of the transmitted packet, and K is a constant. Each station calculates its own L_i and includes this in the transmitted packet's header. A station observes a packet and notices the value of L_i in this packet header. If the station's L_i is greater than this packet's L_i, the station increases its CW by a small amount, and vice versa. The algorithm for calculation of a station CW is

if $(overload)$ **then**
  p $\leftarrow (1 + \Delta_2) \times$ p
else if $(L_{own} > L_{rcv})$ **then**
  $p \leftarrow (1 + \Delta_1) \times p$
else if $(queue\_empty)$ **then**
  p $\leftarrow (1 + \Delta_1) \times$ p
else
  p $\leftarrow (1 - \Delta_1) \times$ p
end if
$p = min(p, 1)$
$CW = p \times CW_{802.11}$

where Δ_2 is a constant, L_{own} is calculated by the station, L_{rcv} is the value of L_i in the header of observed packet, p is a scaling factor and Δ_1 is calculated as

$$\Delta_1 = k \left| \frac{L_{own} - L_{rcv}}{L_{own} + L_{rcv}} \right|$$

where k is another constant, which is different from k in the formula for r_i^{new}. The condition *overload* occurs when there are a large number of stations with high weights, resulting in stations choosing small values for CW and there are a large number of collisions. This condition could be tested like this

if $(av\_nr\_coll > c)$ **then**

$overload = true$
end if

where c is a constant that needs to be chosen properly. It is a trade-off between the efficiency of the medium and the closeness to the desired bandwidth allocation. If c is too small, flows with high weights may not reduce their CWs adequately resulting in insufficient bandwidth allocation. If c is too high, the number of collisions is high and the medium efficiency is decreased. The average number of collisions is updated after each successful transmission as follows

$$av\_nr\_coll = (1 - t) \times num\_coll + t \times av\_nr\_coll$$

where av_nr_coll in the right hand side of the equation is the last calculated value of average number of collisions and t is a small smoothing factor. For the case where a node i transmits n flows with weights $W_1, \ldots, W_n$, it uses the label

$$L_i = \frac{r_i}{\sum_{j=1}^{n} W_j}$$

where r_i i the total bandwidth of the node. Then this node uses weights $W_1, \ldots, W_n$ to choose the next packet in its flows for transmission.

Banchs *et al.* [10] suggest a scheme called Assured Rate MAC Extension (ARME), in which after each packet transmission, if the estimated sending rate of the station is higher (or lower) than the desired rate, the station CW is slightly increased (or decreased). The algorithm for CW calculation is as follows

if $(overload)$ **then**
  $p \leftarrow (1 + \Delta_4) \times p$
else if $(qlen = 0)$ **then**
  $p \leftarrow (1 + \Delta_1) \times p$
else if $(bsize < blim)$ **then**
  $p \leftarrow (1 + \Delta_2) \times p$
else
  $p \leftarrow (1 - \Delta_3) \times p$
end if
$p = min(p, 1)$
$CW = p \times CW_{802.11}$

where Δ_4 is a constant, $qlen$ is the queue length and CW is only decreased when this is greater than 0; $blen$ is the bucket length of the resource that the station has for transmission; $blim$ is the minimum length required for the bucket resource before a transmission occurs; Δ_1 is a constant and Δ_2 and Δ_3 are calculated as follows

$$\Delta_2 = \frac{blim - blen}{blim} \times \Delta_1$$

$$\Delta_3 = \frac{blen - blim}{bsize - blim} \times \Delta_1$$

where $bsize$ is the acceptable burstiness of the source, with the maximum burst length equal $bsize - blim$. The condition $overload$ occurs when there are a large number of stations with high weights, resulting in stations choosing small values for CW and there are a large number of collisions. This condition can be tested as mentioned in the previous paragraph.

Qiao *et al.* [39] recommend a priority-based MAC (P-MAC) protocol. There are n traffic classes with weights such that $0 < \phi_n < \ldots < \phi_2 < \phi_1 = 1$. Assume that each station only has one traffic flow. Let f_i be the set of stations that carry class traffic i. Assume that the traffic flows have the same MAC frame size. Then the weighted fairness is achieved if

$$\forall i, j \in \{1, \ldots, n\}, \forall s \in f_i, \forall s' \in f_j, \frac{SU_s}{\phi_i} = \frac{SU_{s'}}{\phi_j}$$

where SU_s is the probability that station s transmits a frame successfully. This is equivalent to

$$\forall u, v \in f_i, CW_u = CW_v$$

and

$$\forall j \in \{2, \ldots, n\}, CW_j = \frac{CW_1 - 1}{\phi_j} + 1$$

In P-MAC, the CW of each station is chosen to achieve the weighted fairness shown above and to reflect the number of stations in contention for medium access in order to maximize aggregate throughput. The number of traffic classes and their weights are assumed to be known in advance and CW_1 is set to a given initial value. Let *avg_idle* and *avg_wait_i* be the average number of consecutive idle slots and the average number of slots between two adjacent successful transmission of traffic class i, respectively. The stations sense the medium to determine at each slot whether the medium is idle or busy, if the busy period is due to collision or successful transmission of a packet of which traffic class. Let an idle-busy-cycle be the duration between the end of a busy period to the end of the next busy period. After each idle-busy-cycle, *avg_idle* is updated while *avg_wait_i* is only updated if a traffic class i packet is successfully transmitted. The calculation of these values uses the corresponding previous average value with a smoothing factor α. At the end of each observation window, the $|f_i|$ is updated and *avg_idle* and *avg_wait_i* are reset. The calculation of $|f_i|$ uses the previous value with smoothing factor β, where the instantaneous value is given by

$$|f_i| = \frac{(CW_i - 1) \times (avg\_idle + 1)}{2 \times avg\_idle \times (avg\_wait\_i + 1)}$$

The optimal value for CW_1 is calculated base on these values of $|f_i|$ and the optimal CW values for other traffic classes are updated according to the formula above.

Wang *et al.* [21, 50] suggest a method in which each station considers all other stations as a whole entity and therefore only has the notation of itself and *the others*. Each station has a pre-defined fair share of the bandwidth that it should receive. When a station receives a packet, depending on the type of packet, it updates estimation of its share of bandwidth or the others' share of bandwidth accordingly. If a station receives a packet destined for it or for the others, it updates estimation of its own share of bandwidth or the others' share of bandwidth, respectively. If the optional RTS/CTS mechanism is used (this is obvious in reception of a RTS/CTS packet, however, in case a DATA or ACK packet is received, the corresponding DATA packet size would be greater than the RTS_THRESHOLD), the RTS/CTS packets are also included in the estimation of share of bandwidth. If a station receives a RTS, CTS or DATA packet destined for the others, it updates the transmission time of the corresponding DATA packet. When this station receives an ACK packet destined for the others, the latest transmission time value of the DATA packet destined to the others is used in calculation of the others' share of bandwidth. Define the estimated fairness index $FI_e = \frac{W_{ei}}{\Phi_i} / \frac{W_{eo}}{\Phi_e}$, where W_{ei} and W_{eo} are estimated throughput of station i and the others, respectively. Φ_i and Φ_o are the pre-defined fair share of station i and the others, respectively. If the station estimates that it has more than/enough/less than the required share of bandwidth (FI_e is greater than C/between $\frac{1}{C}$ and C/less than C, where C is a constant greater than but close to 1) it would double/keep/halve the current CW until the CW reaches the CW_{max} or CW_{min}.

4.2.3 Matching priority to BI

Vaidya *et al.* [48] introduce the Distributed Fair Scheduling (DFS) algorithm based on Self Clock Fair Queuing (SCFQ) and IEEE 802.11 DCF. The BI of a packet is chosen proportional to its finish tag and therefore the packet with smallest finish tag would be transmitted first. Assume each station only has one flow. Each packet has a start tag and a finish tag. When a station i hears or transmitted a packet at time t with finish tag Z, it sets its virtual clock to $maximum(v_i(t), Z)$, where v(t) is the virtual time at real time t. When packet P_i^k reaches the front of the queue at station i at real time f_i^k, it is marked with a start tag $S_i^k = v(f_i^k)$. Its finish tag is calculated as

$$F_i^k = S_i^k + Scaling\_Factor \times \frac{L_i^k}{\phi_i}$$

where $Scaling\_Factor$ is for suitable scale of virtual time, L_i^k is the length of packet and ϕ_i is the weight of station i. At time f_i^k, this packet is also assigned a backoff interval in unit of SlotTime

$$B_i = \lfloor F_i^k - v(f_i^k) \rfloor$$

where $\lfloor a \rfloor$ returns the maximum integer not greater than a. For collision reduction, B_i is randomized with a random variable ρ uniformly distributed in [0.9 1.1]. The final expression for B_i is

$$B_i = \lfloor \rho \times Scaling\_Factor \times \frac{L_i^k}{\phi_i} \rfloor$$

If there is a collision, the colliding stations choose a new backoff interval uniformly distributed between $[1, 2^{CollisionCounter-1} \times CollisionWindow]$, where $CollisionCounter$ stores the number of collisions that this packet has experienced and $CollisionWindow$ is a constant. If a station has multiple flows, when a packet reaches the front of a flow, its start tag and finish tag are marked. A station chooses the packet with the smallest finish tag among the packets at the front of its flows. The BI for this packet is then calculated as described above.

Dugar *et al.* [20] propose an approach combining the ideas from DFS and High Performance LAN/1 (HIPERLAN/1) to match packet length, priority and weight of a flow to which the packet belongs, to BI. Station i with weight ϕ_i chooses BI B_i for its k^{th} packet P_i^k with packet length L_i^k as follows

$$B_i = \lfloor \rho \times Scaling\_Factor \times \frac{L_i^k}{\phi_i} \rfloor$$

where ρ is a random variable uniformly distributed between [0.9, 1.1], $Scaling\_Factor$ is used for a suitable scale for backoff interval and $\lfloor a \rfloor$ returns the greatest integer not greater than a. The backoff interval is represented in a base-N format (for example, if backoff interval is 33 in decimal and base N is 2, then the base-2 representation of backoff interval is 100001). The station constructs a tuple $(p, c, n, d_{(n-1)}, \ldots, d_0)$ with the following elements. p is the priority of the station. If inter-round spacing irs is M slots, then M priority levels from 0 to M-1 could be supported. The traffic with lower priority level has higher priority. c is collision status of a station. Each station keeps a collision counter $ccntr_i$, which is incremented on each collision and is reset after each successful transmission. If $ccntr_i > 0$, then c is zero, otherwise c is 1. n is the number of digits in the base-N representation of the backoff interval. d_i is the i^{th} digit in the base-N representation of the backoff interval, with the 0^{th} digit is least significant. When a station senses the medium idle for inter-round spacing irs, it transmits a burst to start contention resolution procedure. The irs is chosen greater than the maximum idle time during the contention resolution, so that a station would not interrupt an ongoing contention resolution cycle. Each element in the tuple represents a phase in the contention resolution cycle. Only stations that survive the first i-1 phases go to the i^{th} phase. In the i^{th} phase, a station senses the medium for t_i slots, where t_i is the i^{th} digit from left in the tuple $(p, c, n, d_{(n-1)}, \ldots, d_0)$. If the station hears a transmission, it drops out of contention. After the last phase, there are potential winner(s) of the contention resolution cycle. When station i, the winner of the contention resolution cycle, gets permission to transmit, it piggybacks its backoff interval B_i and its priority level in the data packet. A station j with the same priority level, hearing this packet transmission, recalculates its backoff interval as $B_j = (B_j - B_i)$ if $B_j \geq B_i$. Station j then reconstructs the tuple from this new, smaller backoff interval. This procedure ensures that flows with the same priority level share the bandwidth fairly. If two or more stations with highest priority choose the same smallest backoff interval, then they are the winner of the contention resolution cycle and there is a collision. Each colliding station increments its collision counter by 1 and chooses a new backoff interval uniformly distributed between $[1, 2^{ccntr_i - 1} \times CollisionWindow]$ where $CollisionWindow$ is a constant. The station then constructs a tuple from this new backoff interval and waits for the medium to be idle for irs before contending with the new tuple.

Ogawa *et al.* [35] propose a scheme for control of Class of Service (CoS). When the k^{th} packet p_i^k reaches the front of the queue at station i, its start tag S_i^k, finish tag F_i^k and Packet Virtual Time (PVT) V_i^k are calculated as follows

$$\begin{cases} S_i^k = V_i \\ F_i^k = S_i^k + Scaling\_Factor \times L_i^k / w_i \\ V_i^k = V_i \end{cases}$$

where L_i^k is the packet size of packet p_i^k, w_i $(0 < w_i \leq 1)$ and V_i are weight and Station Virtual Time (SVT) of station i, respectively. The initial value of SVT and PVT are set to zero. When station i hears or transmits a packet with its finish tag z, its associated PVT and SVT are set as follows

$$V_i^k = \begin{cases} max(V_i^k, z), F_i^k > z \\ V_i^k, otherwise \end{cases}$$

$$V_i = max(V_i, z)$$

When a packet reaches the front of the queue, its BI in unit of SlotTime is calculated as

$$Backoff\ Time = \lfloor (F_i^k - V_i^k) \times Rand() \times 2^{2+j} \rfloor$$

where $\lfloor a \rfloor$ returns the greatest integer not greater than a, $Rand()$ returns a random number uniformly distributed between (0,1), j is the number of transmission attempts of the packet. The fair throughput is achieved with the scaling of $SlotTime$ according to weight. Define $Reference\_SlotTime$ as $SlotTime$ of weight 1, SlotTime of station i with weight w_i is $SlotTime = Reference\_SlotTime / w_i$.

4.2.4 Discussion

The principle of fair queuing is to regulate the time that traffic has to wait according to its priority, such that each traffic class has an equal opportunity for transmission and a bandwidth proportional to its priority. A fair queuing approach involves choosing a fair queuing mechanism, mapping of user requirements and using the parameters for service differentiation. The mechanism influences protocol complexity and computational cost [37].

Pattara-Atikom's method [37] is based on DRR, which has $O(1)$ complexity and provides simple mapping of QoS to IFS. There is no backoff procedure, therefore the throughput and delay variation are improved. The desired throughput and medium access right are mapped into separate parameters, service quantum rate and DC, respectively. This solves the fairness problem and prevents starvation of low-priority traffic. Banchs's approach [9] DWFQ offers a flow with average bandwidth according to its weight, however no guarantee can be granted for individual packets. Therefore this approach may show some short-term unfairness. It involves small changes in the calculation of CW and leads to legacy 802.11 stations behaving as best-effort stations. Banchs's method [10] ARME requires small modifications in the calculation of CW and legacy 802.11 stations receive best-effort service. It can provide absolute throughput guarantee in normal conditions. Qiao's P-MAC [39] does not consider traffic delay/jitter requirements, therefore it may not be fair in terms of delay/jitter. Wang's algorithm [50] can solve the fairness problem when packet lengths are variable and work with both basic and RTS/CTS access methods. Vaidya's mechanism [48] DFS is based on SCFQ, which is of complexity $O(log(n))$, where n is the number of flows [37]. DFS can provide relative throughput guarantees. Nevertheless, it does not consider delay in the differentiation analysis. Each station has to examine the finish tag of every packet and the header format of 802.11 must be changed to include the finish tag [10]. The scheme does not address the maximizing of the channel utilization [39].

5 Centralized Approaches Based on PCF

In centralized approach, stations obtain the right to access the medium from a certain station. This station coordinates the medium access by polling stations, the order in which stations are polled depends on the mechanism used.

5.1 Priority-based approaches

Ganz *et al.* [22] propose a robust SuperPoll method. In the SuperPoll protocols, a list of stations allowed to transmit in a certain period is announced at the start of that period. The PC determines the polling sequence in the current PCF period for all registered stations. It calculates $W_{PC} = (GroupSize) \times (T_{DATA} + T_{ACK})$, the maximum time that it would wait before ending CFP, which is at most the maximum CF duration, with T_{DATA} and T_{ACK} are the transmission time of data and ACK packets, respectively and $GroupSize$ is the number of stations in the initial SuperPoll. At the beginning of the current PCF period, the PC broadcasts a Beacon and an initial SuperPoll, which is a sequential polling list with IDs of the registered stations and sets C_{NTS} (the current number of Nothing To Send (NTS) packets sent) to 0. When the W_{PC} expires or the last station in the polling list finishes transmission, the PC broadcasts the CF-End to reset the NAV and end the CFP. All stations register with the PC during DCF period. Each station needs to listen to the PC and others to find out whether it is in the polling list, its order in the list, its predecessor (prior station in the polling list) and the maximum time it has to wait before transmission, which is $W_{max} = (order - 1) \times (T_{DATA} + T_{ACK}) + C_{NTS} \times (T_{DATA} + T_{ACK} + T_{NTS})$, where T_{NTS} is transmission time of NTS packet. If a station is not in the polling list, it does nothing. If it is the first station in the polling list (received from the initial SuperPoll or a data or NTS packet), it starts transmission. If it is not the first station, it calculates its W_{max} timer and when this timer expires, it sets C_{NTS} to 0 and starts transmission. If a station is in the polling list but has no data to transmit, it increments C_{NTS} by 1 and sends a NTS packet. A station transmitting (data or NTS packet) during the PCF period includes in its data or NTS packet a remaining SuperPoll, which has the initial polling list without all the stations prior to this current station in the polling order and the new C_{NTS} value. At the beginning of CFP, when a beacon is received, the station sets the NAV and stops its timer. At the end of CFP, when a CF-End is received, NAV is reset and timer is stopped.

Suzuki *et al.* [47] present two polling schemes for PCF. In priority scheme 1, the AP adds all multimedia stations to its polling list at the beginning of CFP and polls the stations in round robin manner. However, the stations use a static priority scheduling algorithm for transmission of MPDUs. Voice, video and data MPDUs have high, medium and low priorities, respectively. According to priority scheme 2, at the beginning of CFP, the AP adds all multimedia stations to its high-priority and low-priority polling lists. If data stations want to transmit data to AP using PCF, they are added to the low-priority polling list. The AP polls stations in the high-priority polling list in round robin. After all stations in this high-priority list finish transmission of voice and video MPDUs and are dropped from the high-priority list or the high-priority polling period expires, AP polls stations in low-priority list in round robin manner. When all stations are dropped from the low-priority list or the CFP expires, AP terminates the CFP.

Yeh *et al.* [54] introduce four polling schemes for PCF. In the Round Robin (RR) scheme, the PC polls the stations in turn, starting from the station with lowest address until the CFP duration reaches its maximum. The types of message from PC to stations include: CF-Poll, CF-Ack-Poll, CF-Data-Poll, and CF-Data-Ack-Poll. In the FIFO scheme, the PC transmits frames to stations according to the order of frames in its queue and piggybacks the polling frames to the corresponding stations. If the PC does not receive ACK from the polled station, it keeps polling until reaching the retry limit. If there are no frames in the PC queue, the FIFO scheme becomes RR scheme. When there is a new frame in the PC queue, the FIFO scheme is resumed. In the Priority scheme, the PC transmits traffic or poll frames to the stations from higher Type of Service (TOS) to lower TOS in turn. After all stations with TOS > 1 has been served, best-effort stations are polled in round robin order until frames of TOS > 1 enter the PC queue. The PC always piggybacks polling frame with data frame and piggybacks ACK frame if it needs to respond to a station transmitting a data frame just now. For the last scheme, which is called Priority Effort Limited Fair (Priority-ELF), the PC transmits traffic or poll frames to the stations from higher TOS to lower TOS in turn. However, the PC checks counters that record the number of frames transmitted from each station in a given period. The PC only polls the stations with counter values greater than 1, even if the PC has data for the station in its queue or the station has data to send.

5.2 TDMA-like approaches

Ni *et al.* [34] mentions a method, in which TDMA-like time slots are set up and allocated to stations for service differentiation. Each station only transmits in its time slots and there is little requirement from the AP to intervene with packet transmissions.

Wei *et al.* [51] recommend a method called Slotted PCF (S-PCF). During CP, the PC identifies the stations and their priority in a list. The PC polls stations for transmission requests and determines the transmission order based on traffic priority. After the CP period, the PC sends a beacon frame with transmission parameters to all stations, assigning slots to stations from higher priority to lower priority. The PC could terminate the CFP at any time with a Contention Free Acknowledgment (CFACK) frame. When a station receives the beacon frame, it sets its transmission parameters such as NAV. If a station receives a traffic frame, it should reply with a small frame. A station always waits until its reserved slot to transmit a frame. If there are many transmission requests during S-PCF period, lower priority stations wait for the next DCF or S-PCF period. A station sets new NAV after receiving a CFACK frame.

Qiang *et al.* [28, 38] introduce a method using distributed schedulers. At the AP, there is both the Master Scheduler and one Slave Scheduler (SS) while there is only one SS at each station. The Master Scheduler assigns time slots to SSs and each SS then assigns its allocated time slots among its connections. A Contention Free Period Repetition Interval (CFPRI) consists of a period for contention-free traffic and a period for contention-based traffic. A maximum bandwidth portion (say 25%) is assgined for non-real-time traffic. If the number of Non-Real-Time Data Units (NRTDUs) reported to the Master Scheduler is less than or equal to 0.25 $\times$ the total number of Data Units (DUs) that could be supported in a CFPRI, then the unused bandwidth by NRTDUs could be allocated for Real-Time Data Units (RTDUs) and their delay could be decreased. The NTRDUs have a large timeout value which is decremented with time at the SSs. When this value reaches the timeout value of RTDUs, the NRTDU would be considered as a RTDU and its delay requirement is reported to the Master Scheduler. If the number of NRTDUs is greater than 0.25 $\times$ the total number of DUs that could be supported in a CFPRI, bandwidth portion for NRTDUs would be limited to 25%. Some NTRDUs staying long enough in the the queue would get some bandwidth. If a station has more time slots than the RTDUs (occur when the sum of total RTDUs and total NRTDUs reported to Master Scheduler is less than the total number of DUs supported in a CFPRI), the unused slots could be used by the remaining stations or AP could end CFP and the unused bandwidth is used in CP. If there is no NRTDUs, the total duration of CFPRI minus time for Association Request (AR) frame (used by a new station to associate with AP) transmission is assigned for RTDUs.
At the beginning of CFPRI, the Master Scheduler at AP allocates the bandwidth for each SS according to the data unit priorities. The assigned priority value depends on the remaining timeout of DUs and channel error. When a station or the AP has data for transmission, the Association Information (AI) is piggybacked in uplink data from remote SS to the Master Scheduler the in AP. An SS schedules for transmission of NRTDUs in CP. For those NRTDUs whose timeout reaches the timeout value of RTDUs, SS sends this AI to the Master Scheduler. The bandwidth is shared fairly between different DU categories and stations. After the beacon frame, the AP broadcasts Polling Information (PI) once, which includes the order of stations and their assigned time slots for transmission. If a station is allocated time slots in this CFP, the ACK and AI are piggybacked in its uplink data. If a station does not have time slots allocated, mini slots for transmission of ACK and AI would be allocated before the end of CFP. The AP broadcasts one ACK with a bitmap for all successful transmission of uplink DUs in this CFP.

Zhao *et al.* [55] recommend an approach called Modified-PCF (M-PCF), in which stations access the medium in a hub-poll manner. During CP, when a station successfully gains access to the medium, the AP adds this station to its polling list, gives it a polling sequence number and broadcasts to other stations. In CFP, after the first station is polled, all stations in polling list are allowed automatic access to the medium in turn without polling. Furthermore, a station receiving a non-real-time frame replies with an ACK, while a station receiving a real-time frame does not.

5.3 Adaptive polling approach

Kim *et al.* [32] suggest a polling scheme for voice transmission. The Statistical Activity Detection (SAD) performs the following tasks in each phase of a voice activity. In phase 1, *detection period of talkspurt-to-silence state change*, the SAD could detect the station changing to *silence* state if the *more data* field in the MAC header is set to *false*. During phase 2, *silence period*, the SAD assumes there is no packet for transmission unless there is *silence-to-talkspurt state change*. Within phase 3, *detection period of silence-to-talkspurt state change*, after the arrival of a packet, the SAD assumes the voice activity is changed to *silence* state and must stay there for a period of time at least $T_{threshold}$ before changing back to *talkspurt* state. In phase 4, *talkspurt period*, the SAD assumes the station remains in *talkspurt* state as long as the *more data* field in the MAC header is set to *true*. There are four logical lists: pollable list, active node list, inactive node list and adaptive polling list. Each station is represented with an Association ID (AID). The pollable list has the AIDs of all the stations. The active and inactive lists keep the AIDs of stations currently in *talkspurt* and *silence* states, respectively. The adaptive polling list is used by the AP to poll stations during CFP. At the beginning, all stations in the pollable list are assumed to be in *talkspurt* state. If a station changes to *silence* state, its AID is removed from the active and adaptive lists and is moved to the inactive list. The SAD has a timer for each inactive station, and when this timer exceeds $T_{threshold}$, the station AID is moved from the inactive list to the adaptive list. If this station has a DATA frame to send, then its AID is added to the active list. The adaptive polling list is shorter than the exhaustive polling list in IEEE 802.11 round robin scheme.

5.4 Contention-based multipolling approach

Lo *et al.* [33] propose a contention-based Multipoll technique. This method reduces the number of polling frames that the PC sends, as compared with the SinglePoll method used in PCF or HCF, in which the PC sends one polling frame for each polled station in the polling list. Different backoff intervals are assigned to stations in the polling list and the stations perform backoff procedure after receiving the CP-Multipoll frame. The polling order is transformed into the contending order, which is the same as the order of the assigned backoff interval in ascending order. In the CP-Multipoll, stations perform restricted backoff procedure without waiting for medium to be idle for DIFS, therefore contention of these stations is protected from other stations performing backoff procedure in DCF mode. When a station receives a CP-Multipoll frame, this station checks whether it belongs to the polling list. If it is in the polling list, it sets its backoff time, maximum TXOP duration and determines the actual transmission duration. If the medium is idle, the backoff interval is decremented until this interval reaches zero, when RTS/CTS frames are exchanged and if there is no collision, DATA frame(s) are transmitted. If there is a collision in RTS/CTS frame transmission, after three unsuccessful retransmission attempts, the station aborts its transmission. If the medium is busy, the NAV is set after receiving a frame and the station defers medium access until the time specified in the NAV. For the PC, if the medium is busy, the PC performs the initial backoff after the medium is found idle for PIFS. The PC sets its internal backoff time and sends a CP-Multipoll frame. If it receives a RTS frame, it responds with a CTS frame and also records the successful stations. When the medium is idle, the backoff interval is decremented. When the backoff interval reaches zero, the PC sends a null CP-Multipoll frame to cancel all the pending polled stations and polls the failed stations individually. A station receiving the null CP-Multipoll abandons the unfinished actions in the previous CP-Multipoll.

5.5 Discussion

Ganz's scheme [22] works better than single poll scheme in noisy or hidden terminal conditions because stations have more opportunities to receive the poll. However, each station has to listen to the PC and other stations for the SuperPoll frame and the redundant polling frames use more medium space. In Suzuki methods [47], for a given voice and video MPDU dropping probability, priority schemes 1 and 2 can accommodate a larger number of multimedia stations. However, the non-priority scheme can offer higher data throughput. For scheme 2, the effect of high-priority polling period is a trade-off between throughput of multimedia and data stations. Yeh's priority and Priority-ELM schemes [54] can provide QoS and prevention against link errors. However, the priority scheme starves the best-effort traffic. The Priority-ELF scheme has higher total throughput when there are link errors and also prevents starvation of best-effort traffic better than the priority scheme. For balanced traffic model, the total throughput of round robin and DCF schemes are similar and much higher than that of the FIFO scheme. On the other hand, with unbalanced traffic model, the total throughput of FIFO scheme is much higher than DCF scheme, which is higher than round robin scheme. In Ni's and Wei's methods [34, 51], stations only transmit in their allocated slots and there is no need for polling frames, thus more bandwidth can be used for traffic transmission. Mathematical analysis shows that Wei's method performs better than DCF in terms of collision probability, bandwidth rate, average delay and data frame loss. Qiang's approach [38] can guarantee the delay and throughput of real-time services and performs better than DCF in terms of throughput and delay. Zhao's M-PCF [55] can support voice and data traffic better than PCF with higher voice throughput and lower packet dropping probability in both ideal channel or erroneous channel conditions. For Kim's adaptive polling [32], in heavy load conditions, SAD and Hybrid Activity Detection (HAD) have good performance in terms of goodput and delay because unnecessary polling attempts are avoided while *talkspurt* occurrence is detected. However, for video support, traffic characteristics of video need to be considered.

6 Summary and Conclusion

In this article, the operation of the legacy IEEE 802.11 MAC functions, including DCF and PCF, is described. The 802.11e enhancements to the original MAC, consisting of EDCF and HCF functions, is also introduced. Afterward, the QoS mechanisms for WLAN, based on the distributed and centralized approaches are summarized and evaluated. Generally, distributed methods are simpler for implementation and have smaller overhead as compared with centralized ones. Distributed protocols are also more flexible than centralized protocols when dealing with highly bursty traffic of real-time services. However, centralized mechanisms can guarantee bandwidth requirements, while distributed mechanisms cannot. Distributed approaches can be further divided into priority-based and fair scheduling. In priority-based approach, the differentiation based on IFS is simpler than mapping priority to CW or BI. Furthermore, it does not introduce throughput and delay fluctuation as does the approach based on CW or BI using a random BI. The priority-based mechanisms, however, do not give traffic equal opportunity for medium access and tend to starve lower priority traffic, hence the need for fair scheduling schemes.The fair scheduling methods usually employ additional fields in the MAC header and require stations listen to every frame transmitted and inspect these fields to find out the load condition, the throughput and delay experienced by each flow. They also require the assignment of the flow weights, the calculation of the fairness index resulting in some complexity. Of all the methods evaluated, most of them only consider throughput guarantee but not the delay/jitter requirements. These aspects of QoS are of increasing importance for video streaming and anticipated increases to interactive video applications. Although the proposed techniques can offer QoS differentiation, in order to guarantee QoS, admission control and resource allocation are also required. Moreover, in order to select the suitable QoS mechanism with the proper set of parameters, more research including theoretical studies and simulations is required.

References

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

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

[3] IEEE Std. 802.11b 1999. 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.

[4] IEEE 802.11e/D4.0. Draft Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), November 2002.

[5] IEEE Std. 802.1d 1998. Part 3: Media Access Control (MAC) Bridges, 1998.

[6] I. Aad and C. Castelluccia. Introducing Service Differentiation into IEEE 802.11. In *Fifth IEEE Symposium on Computers and Communications ISCC 2000*, pages 438–443, July 2000.

[7] I. Aad and C. Castelluccia. Differentiation Mechanisms for IEEE 802.11. In *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001* [23], pages 209–218.

[8] I. Aad and C. Castelluccia. Remarks on Per-flow Differentiation in IEEE 802.11. In *European Wireless EW 2002*, Florance, Italy, February 2002.

[9] A. Banchs and X. Perez. Distributed Weighted Fair Queuing in 802.11 Wireless LAN. In *IEEE International Conference on Communications ICC 2002* [24], pages 3121–3127.

[10] A. Banchs and X. Perez. Providing Throughput Guarantees in IEEE 802.11 Wireless LAN. In *IEEE Wireless Communications and Networking Conference WCNC 2002*, volume 1, pages 130–138. IEEE Communications Society, March 2002.

[11] A. Banchs, X. Perez, M. Radimirsch, and H. J. Stuttgen. Service Differentiation Extensions for Elastic and Real-time Traffic in 802.11 Wireless LAN. In *IEEE Workshop on High Performance Switching and Routing*, pages 245–249, May 2001.

[12] M. Barry, A. T. Campbell, and A. Veres. Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks. In *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001* [23], pages 582–590.

[13] M. Benveniste. Overview of Tier Contention Multiple Access (TCMA). IEEE Document, 802.11-01/019, January 2001.

[14] M. Benveniste. Proposed Normative Text for TCMA with Backoff Adaptation. IEEE Document, 802.11-01/135r1, March 2001.

[15] M. Benveniste. Tiered Contention Multiple Access (TCMA), a QoS-based Distributed MAC Protocol. In *The* 13^{th} *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 598–604, September 2002.

[16] Wen-Tsuen Chen, Bo-Bin Jian, and Shou-Chih Lo. An Adaptive Retransmission Scheme with QoS Support for the IEEE 802.11 MAC Enhancement. In *IEEE* 55^{th} *Vehicular Technology Conference VTC 2002* [26], pages 70–74.

[17] G. Chesson, A. Singla, W. Diepstraten, and H. Tenniusen et. al. VDCF State Machine Description. IEEE Document, 802.11-00/412r1, November 2000.

[18] G. Chesson, A. Singla, W. Diepstraten, and H. Tenniusen et. al. VDCF Proposed Draft Text. IEEE Document, 802.11-01/131, March 2001.

[19] J. Deng and R. S. Chang. A Priority Scheme for IEEE 802.11 DCF Access Method. *IEICE Transactions on Communications*, E82-B(1):96–102, January 1999.

[20] A. Dugar, N. Vaidya, and P. Bahl. Priority and Fair Scheduling in a Wireless LAN. In *IEEE Military Communications Conference MILCOM 2001, Communications for Network-Centric Operations: Creating the Information Force*, volume 2, pages 993–997, October 2001.

[21] Zuyuan Fang, Brahim Bensaou, and Yu Wang. Performance Evaluation of a Fair Backoff Algorithm for IEEE 802.11 DFWMAC. In 3^{rd} *ACM International Symposium on Mobile Adhoc Networking and Computing*, pages 48–57, Lausanne, Switzerland, July 2002.

[22] A. Ganz, A. Phonphoem, and Z. Ganz. Robust Superpoll with Chaining Protocol for IEEE 802.11 Wireless LANs in Support of Multimedia Applications. *Wireless Networks*, 7(1):65–73, 2001.

[23] IEEE Communications Society. *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001*, April 2001.

[24] IEEE Communications Society. *IEEE International Conference on Communications ICC 2002*, April 2002.

[25] IEEE Communications Society. *IEEE Wireless Communications and Networking Conference WCNC 2003*, March 2003.

[26] IEEE Vehicular Technology Society. *IEEE* 55^{th} *Vehicular Technology Conference VTC 2002*, May 2002.

[27] IEEE Vehicular Technology Society. *The 57th IEEE Semiannual Vehicular Technology Conference VTC 2003*, April 2003.

[28] L. Jacob, R. Radhakrishna Pillai, and B. Prabhakaran. MAC Protocol Enhancements and a Distributed Scheduler for QoS Guarantees over the IEEE 802.11 Wireless LANs. In *IEEE* 56^{th} *Vehicular Technology Conference VTC 2002*, volume 4, pages 2410–2413. IEEE Vehicular Technology Society, September 2002.

[29] Kanghee Kim, A. Ahmad, and Kiseon Kim. A Wireless Multimedia LAN Architecture Using DCF with Shortened Contention Window for QoS Provisioning. *IEEE Communications Letters*, 7(2):97–99, February 2003.

[30] Kanghee Kim, A. Ahmad, and Kiseon Kim. A Wireless Multimedia LAN Architecture Using DCF with Shortened Contention Window for QoS Provisioning. In *IEEE Wireless Communications and Networking Conference WCNC 2003* [25], pages 1308–1311.

[31] Kanghee Kim, Seokjoo Shin, and Kiseon Kim. A Novel MAC Scheme for Prioritized Services in IEEE 802.11a Wireless LAN. In *Joint* 4^{th} *IEEE International Conference on ATM (ICATM 2001) and High Speed Intelligent Internet Symposium*, pages 196–199, April 2001.

[32] Young-Jae Kim and Young-Joo Suh. Adaptive Polling MAC Schemes for IEEE 802.11 Wireless LANs. In *The 57th IEEE Semiannual Vehicular Technology Conference VTC 2003* [27], pages 2528–2532.

[33] Shou-Chih Lo, Guanling Lee, and Wen-Tsuen Chen. An Efficient Multipolling Mechanism for IEEE 802.11 Wireless LANs. *IEEE Transactions on Computers*, 52(6):764–778, June 2003.

[34] Q. Ni, L. Romdhani, T. Turletti, and I. Aad. QoS Issues and Enhancements for IEEE 802.11 Wireless LAN. Technical Report 4612, INRIA, November 2002.

[35] M. Ogawa, I. Shimojima, and T. Hattori. CoS Guarantee Control for Wireless LAN. In *IEEE 55th Vehicular Technology Conference VTC 2002* [26], pages 50–54.

[36] W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee. Distributed Mechanisms for Quality of Service in Wireless LANs. *IEEE Wireless Communications*, 10(3):26–34, June 2003.

[37] W. Pattara-Aukom, S. Banerjee, and P. Krishnamurthy. Starvation Prevention and Quality of Service in Wireless LANs. In *The 5th International Symposium on Wireless Personal Multimedia Communications*, volume 3, pages 1078–1082, October 2002.

[38] Qiu Qiang, L. Jacob, R. Radhakrishna Pillai, and B. Prabhakaran. MAC Protocol Enhancements for QoS Guarantee and Fairness over the IEEE 802.11 Wireless LANs. In *Eleventh International Conference on Computer Communications and Networks*, pages 628–633, October 2002.

[39] Daji Qiao and K. G. Shin. Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF. In *Tenth IEEE International Workshop on Quality of Service*, pages 227–236, May 2002.

[40] L. Romdhani, Qiang Ni, and T. Turletti. Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-hoc Networks. In *IEEE Wireless Communications and Networking Conference WCNC 2003* [25], pages 1373–1378.

[41] Shiann-Tsong Sheu and Tzu-Fang Sheu. DBASE: a Distributed Bandwidth Allocation/Sharing/Extension Protocol for Multimedia over IEEE 802.11 Ad hoc Wireless LAN. In *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001* [23], pages 1558–1567.

[42] Shiann-Tsong Sheu, Tzu-Fang Sheu, Chih-Chiang Wu, and Jiau-Yu Luo. Design and Implementation of a Reservation-Based MAC Protocol for Voice/Data over IEEE 802.11 Ad-hoc Wireless Networks. In *IEEE International Conference on Communications ICC 2001*, volume 6, pages 1935–1939. IEEE Communications Society, June 2001.

[43] J. Sobrinho and A. S. Krishnakumar. Real-time Traffic over the IEEE 802.11 Medium Access Control Layer. *Bell Labs Technical Journal*, 1(2):172–187, 1996.

[44] J. L. Sobrinho and A. S. Krishnakumar. Distributed Multiple Access Procedures to Provide Voice Communications over IEEE 802.11 Wireless Networks. In *Global Telecommunications Conference GLOBECOM '96*, volume 3, pages 1689–1694, November 1996.

[45] J. L. Sobrinho and A. S. Krishnakumar. Quality-of-Service in Ad hoc Carrier Sense Multiple Access Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1353–1368, August 1999.

[46] Nah-Oak Song, Byung-Jae Kwak, Jabin Song, and Leonard E. Miller. Enhancement of IEEE 802.11 Distributed Coordination Function with Exponential Increase Exponential Decrease Backoff Algorithm. In *The 57th IEEE Semiannual Vehicular Technology Conference VTC 2003* [27], pages 2775–2778.

[47] T. Suzuki and S. Tasaka. Performance Evaluation of Priority-Based Multimedia Transmission with the PCF in an IEEE 802.11 Standard Wireless LAN. In *The 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages G70–G77, September 2001.

[48] N. H. Vaidya, P. Bahl, and S. Gupta. Distributed Fair Scheduling in a Wireless LAN. In *The 6th Annual International Conference on Mobile computing and Networking*, pages 167–178, Boston, Massachusetts, United States, 2000.

[49] A. Veres, A. T. Campbell, M. Barry, and Li-Hsiang Sun. Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control. *IEEE Journal on Selected Areas in Communications*, 19(10):2081–2093, October 2001.

[50] Y. Wang and B. Bensaou. Achieving Fairness in IEEE 802.11 DFWMAC with Variable Packet Lengths. In *Global Telecommunications Conference GLOBECOM '01*, volume 6, pages 3588–3593, November 2001.

[51] A. Wei and S. Boumerdassi. A New PCF Scheme for Multimedia Traffic in IEEE 802.11 Wireless LAN. In *IASTED International Conference Computer Science and Technology CST 2003*, May 2003.

[52] G. W. Wong and R. W. Donaldson. Improving the QoS Performance of EDCF in IEEE 802.11e Wireless LANs. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing PACRIM 2003*, volume 1, pages 392–396, August 2003.

[53] Jin Xiaohui, Li Jiandong, and Guo Feng. M-DCF: a MAC Protocol Supporting QoS in Ad hoc Network. In *International Conference on Communication Technology WCC-ICCT 2000*, volume 2, pages 1718–1721, August 2000.

[54] Jing-Yuan Yeh and Chienhua Chen. Support of Multimedia Services with the IEEE 802-11 MAC Protocol. In *IEEE International Conference on Communications ICC 2002* [24], pages 600–604.

[55] Liqiang Zhao and Changxin Fan. M-PCF: Modified IEEE 802.11 PCF Protocol Implementing QoS. *Electronics Letters*, 38(24):1611–1613, November 2002.