Institute of Computer Science and Information Engineering Nation Taiwan University

Thesis of Master Degree

Quality of Service Control for IEEE 802.11 Wireless LANs

Adivors： Dr. Ching-Chi Hsu

Dr. Chiou-Shann Fuh

Student: Shin-Neng Hsu

2003 July 20

Abstract

The **WLAN**s international standard of **IEEE 802.11** has defined two mechanisms for implementing the Media Access Control (**MAC**); one is distributed coordination function (**DCF**) and anther is point coordination function (**PCF**). The DCF mechanism employs the Carrier Sense Multiple Access with Collision Avoidance (**CSMA/CA**) strategy to provide data transmission. In general, the majority of frames transmitted via CSMA/CA strategy would not cause collision; but the probability of collision occurrence would get greater as the network traffic becomes larger. In that case, the network throughput isn't only lower but also the latency of transmitting data successfully would be more unpredictable. Thus DCF mechanism is only appropriate for non-real time data service. In PCF mechanism, the latency of transmitting data service is predictable since in PCF mechanism, there is a pointer coordinator (PC), which shall be implemented at the AP, to decide the order and timing of data transmissions of STAs via polling method. However, it results in that the latency of data transmission is unable to be guaranteed since PC doesn't apply any limitations to the number of STAs which want to join the polling list of PC. Thus PCF mechanism is unable to provide real-time transmission for real-time data. According to these reasons, we improve the PCF mechanism to provide the exact timing and period of data transmission and bandwidth guarantees upon different transmitted data, such as audio and video data. Thus this proposed mechanism is appropriate for real-time multimedia transmission. We nominate this mechanism as Quality of Service –PCF, abbreviated as **Q-PCF**, since it provides Quality of Service.

We design a simulation program to see if Q-PCF provides **QoS** and has the following functions. First, the different priorities for variety data services that include constant bit rate (**CBR**) and variable bit rate (**VBR**) data services are supported. The data with higher priority are guaranteed to be transmitted before lower priority data. Second, the STA attempting to acquire the QoS service should be served in a bounded time regardless of the loading of network. Third, the data are guaranteed to be transmitted within finite time regardless of network loading. Last, the bandwidth of some STA is guaranteed regardless of network loading. With providing these functions, Q-PCF mechanism can be called as QoS data service supporting protocol.

Although Q-PCF mechanism provides QoS data service, it is limited by too many hypotheses and this results in incompatibility with the IEEE 802.11 standard. For instance, all STAs in Q-PCF mechanism must disable power-saving mechanism, which is a critical issue for portable computers and equipments, to arrive to real-time data transmission. Another hypothesis is that the two adjacent STAs attempting to transmit data must recognize each other and their radio waves range must cover the other one in order to increase the network throughput and this hypothesis violates the IEEE 802.11 standard. Hence we give a further discussion and provide some amendment to become more compatible with IEEE 802.11. We nominate this revised mechanism as Enhanced Q–PCF, abbreviated as **EQ-PCF**.

EQ-PCF mechanism provides more compatibility with IEEE 802.11 and increases the network throughput besides that it is more appropriate for the real-time multimedia transmission.

keywords：IEEE 802.11、WLAN、MAC、DCF、PCF、CSMA/CA、Polling、Multimedia 、QoS、CBR、VBR

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

With the flourish development of wireless local-area network and the advancement of broadband technologies providing higher data rates, the research of wireless LAN transporting real-time multimedia such as audio and video data is a hot topic today. However, the mechanisms adhere to IEEE 802.11 which is the most popular wireless LAN standard is not enough to transport real-time multimedia immediately. The reason is not the lack of bandwidth but is the lack of assurance of the timing and bandwidth of transportation. Q-PCF [6] is presented to solve the lack of assurance and to provide different bandwidth upon different priorities. And EQ-PCF enhances the compatibilities with IEEE 802.11 standard.

1.1. Motivation

The DCF mechanism in IEEE 802.11 employs the fair competition to access the media for transporting the data completely, so its design doesn't consider the priority [4] and transportation of real-time data. Furthermore, it would decrease the throughput if the STAs participating in competition are too many. The PCF mechanism adopts the polling method that seems to solve the priority problem. Since IEEE 802.11 doesn't define what kind of STAs to join polling list of PC, the above priority becomes meaningless. Since PC doesn't limit the number of STAs which join in polling list, it results in that PC has not enough time to poll every STAs presented in polling list. Hence some STAs are unable to obtain the right to transmit or receive data within bounded time. The STA following to PCF mechanism is unable to require PC assurance of time and bandwidth according as STA's requirements. Besides, STA only transports a MAC Data Service Unit (MDSU) which size is limited and this bandwidth seems unable to transport real-time multimedia. The PCF mechanism defined by IEEE 802.11 is unable to transport real-time data in conclusion. We design a modified mechanism of PCF to increase network throughput and to guarantee the time and bandwidth of transportation. We expect that this mechanism is fit the requirement of real-time multimedia transportation and is compatible with IEEE 802.11 standard.

1.2. Objective

There are three objectives in this research.

Objective 1: supporting to transport real-time multimedia via wireless LAN

We add the Quality of Service to wireless LAN under the situation that doesn't increase network loading. There are four issues to support Quality of Service. (1) It must support different priorities of data transportations. (2) The period of time within that a STA attempts to join in polling list is limited. (3) The time point of data transportation is guaranteed for STAs in the polling list when they attempt to transport data. (4) The bandwidth of data transportation is also guaranteed besides the time point of it. If the above four issues are supported, wireless LAN shall have the ability of transporting real-time multimedia.

Objective 2: increasing the network throughput of wireless LAN

Upon IEEE 802.11 standard, the STAs employ access point (AP) to pass the data transportation between each other in the infrastructure mode and this would waste some bandwidth of wireless network. The bottleneck of this problem is the power-saving mechanism defined by IEEE 802.11. Since the STA_1 doesn't detect whether the other STA_2 that the STA_1 wants to communicate with is in power-saving mode, the communication would fail if the other one is in power-saving mode. If we can overcome the problem of power-saving, the data transportation between two STAs directly is practicable and hence the distribution of bandwidth is more efficient and flexible.

Objective 3: emphasizing the compatibility with IEEE 802.11

Every new mechanism shall increase its implementation and practicability if it is compatible with the current and existent mechanism. Hence this thesis would not only improve wireless LAN to provide real-time multimedia transportation and to increase the throughput of network, but also provide a modified mechanism to be compatible with IEEE 802.11. The Q-PCF mechanism would arrive to the first objective and EQ-PCF mechanism would arrive to the second and third objectives.

CHAPTER 2

Background and Related Works

The noticeable difference between wireless LAN and wired LAN is the media of transmission. The media of transmission evolves from physical wires used by wired LAN such as Ethernet to light such as ultra-red or radio wave used by wireless LAN such as IEEE 802.11. Hence the building of network is more flexible and the network has mobility since the STA may be portable. Those are also the reasons of flourish development of wireless LAN. IEEE 802.11 standard is one of the wireless LAN standards to define the protocols to apply the characterization of wireless LAN and it covers two major parts, namely physical layer (PHY) and media access layer (MAC). Briefly, PHY is responsible to transmit and receive frames via radio wave media. In transmitting part, PHY receives frames from MAC layer, processes these frames with modulation and encoding and transmits them out finally. In receiving part, PHY receives frames from radio wave media, processes these frames with demodulation and decoding and transmits them to MAC layer finally. The MAC layer is responsible to determine how and when to access the media and ensure that the data transportation is successful and correct.

In order to provide optimum wireless LAN service, the committee of IEEE 802.11 still improves the definition of PHY and MAC established at June 1996 until now and this tendency was shown at appendix A. The committee defines two more methods of PHY, namely 802.11a and 802.11b. Thus the data rate of 802.11b is improved from 1~2 Mbps which are the data rate of original 802.11 to 11 Mbps and the data rate of 802.11a is improved to 54 Mbps. The committee of IEEE 802.11 also reforms many issues relative to MAC layer except improvement of data rate. However, the data rate is not only determined by PHY but also determined by MAC. Furthermore, the MAC is a critical cause to improve data rate of wireless LAN. For example, PHY is analogous to a racing car with excellent performance and MAC is analogous to a professional racer. To win the championship they mush work in coordination very well to produce the most performance of the whole system. Hence this thesis starts at the protocol of MAC layer for improving the data rate of wireless LAN and goes deep into it.

2.1. WLANs

The environment of IEEE 802.11 is the single channel transportation. That means that only one transaction of some STA is permitted at any time and any STA attempting to communicate with another STA must take or seize the right and control to use the channel. The MAC layer is defined two communication architecture, namely Ad-Hoc mode and Infrastructure mode, as shown in Figure 2.1

and these two modes employ the same access mechanism such as DCF and PCF. Ad-Hoc mode provides STAs point-to-point communication. The receiving STA receives the data transmitted from transmitting STA directly and this transaction is not permitted to use any intermediary STA to pass the data. The wireless Ad-Hoc mode would have more flexibility, but less extensibility. Thus the wireless LAN using Ad-Hoc mode is appropriate to temporary occasions such as temporary conferences. The infrastructure mode of wireless LAN is a distribution networking system and an access point (AP) placed in infrastructure area is responsible to control which STA has the right to access media for transporting data. The set of an AP and the STAs which are controlled by that AP is called as basic service set (BSS) and several independent BSSs are integrated as extended service set (ESS) to connect to the whole distribution networking system. Thus the STAs placed at different BSS would transport data via the connectors or intermediaries (such as APs) of distribution networking system. Furthermore, the whole wireless LAN shall connect to wired-LAN via the Portal connector to set up the whole network. Since the popular protocol of wired-LAN is Ethernet, the AP substitutes for the Portal. This thesis would discuss infrastructure mode of wireless LAN and goes deep into it.

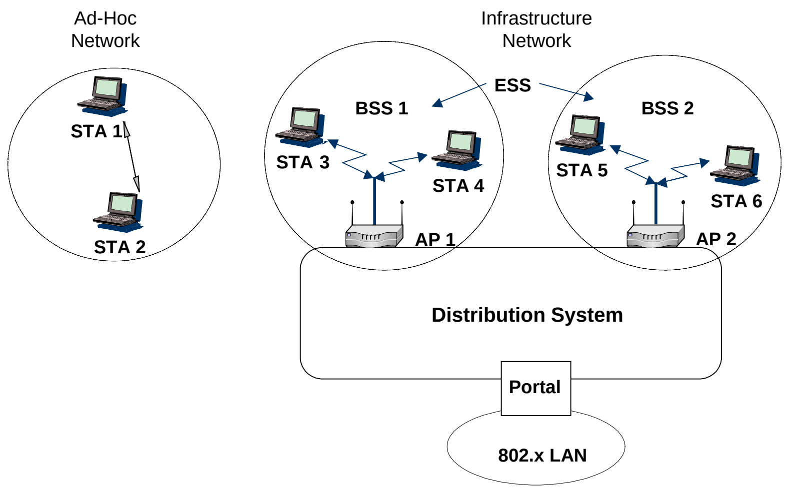


Figure 2.1 Complete IEEE 802.11 architecture

2.2. Distributed Coordination Function (DCF)

The MAC layer of IEEE 802.11 in the infrastructure mode provides two mechanisms to access the media, namely DCF and PCF. The DCF mechanism employs fair contention method to access the media to transport data and the PCF mechanism employs contention-free method, that is exactly polling method, to access the media. Thus PCF is unlike DCF that the collision of frame is occurred and noticeably, is only used in the infrastructure mode. Figure 2.2 illustrates the PCF is build on the top of the DCF mechanism.

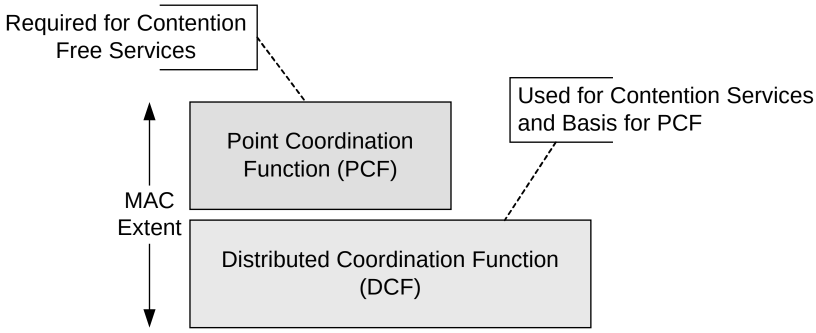


Figure 2.2 MAC architecture

Hence the DCF and PCF mechanisms are coexistent and one of them are used at different time. The period of time when DCF is working is called as contention period (CP) and the period of time when PCF is working is called as contention free period (CFP). Figure 2.3 illustrates that the set of these above periods is integrated as a super-frame and two periods alternate between each other in rotation.

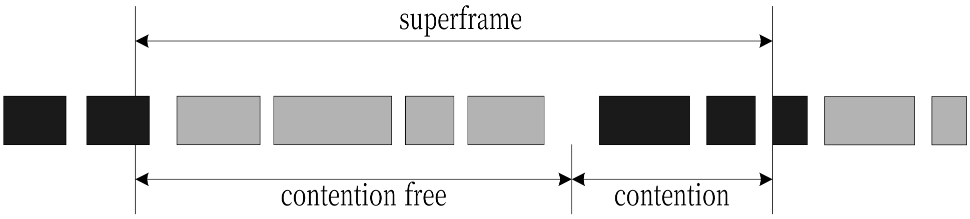


Figure 2.3 Super-frame

2.2.1. Carrier-sense mechanism

The DCF mechanism employs carrier sense multiple access with collision avoidance (CSMA/CA) to detect whether the channel media is idle of busy. The channel media is defined as idle if the strength of signal detected by STA is lower than some threshold. STA attempting to transporting data is permitted to transmit frame if the channel media is idle. On the other hand, the channel media is defined as busy if the strength of signal detected by STA is higher than some defined threshold. If the channel media is busy, STA attempting to access the channel media is required to defer the timing of frame transmission until the channel media is idle.

2.2.2. Interframe space (IFS)

IEEE 802.11 defines different waiting times upon different kind of frames and STA is allowed to transmit its frame until the corresponding waiting time is expired. IEEE 802.11 provides these different waiting times to different priorities level and the following description would show them in order, as shown in Figure 2.4. The first priority of waiting times is short IFS (SIFS) and used at frame transmission that is required to transmit acknowledgement immediately such as ACK. The second priority of waiting times is PCF inter-frame space (PIFS) and is responsible to wait the time before transmitting frame in contention free period (CFP). The third priority of waiting times is DCF inter-frame space (DIFS) and is responsible to wait the time before transmitting frame in contention period (CP). The forth priority of waiting times is extended IFS (EIFS) and is the waiting time to transmit resent frame. Thus the order of priorities from high to low is SIFS, PIFS, DIFS, and EIFS. The frames with higher priority of waiting time would be easier to be transmitted than them with lower priority of waiting time since the waiting time with higher priority is shorter than the waiting times with lower priorities. Hence this mechanism allows that the frame with higher priority of waiting time takes faster transmitting service and avoids frames with different priorities of waiting time from collision. However, the probability of collision occurred by frames with the same priority of waiting time is exist. The collision is occurred when STAs attempting the transmit frame with the same priority of waiting time detect that channel media is idle simultaneously after waiting for the same inter-frame space. In order to solve the above problem, IEEE 802.11 defines a random period of time to wait after inter-frame space (IFS) is expired and before transmitting frame is permitted in the DCF mechanism and this extra waiting time is called as backoff time. The probability of collision occurred by the frame with the same priority shall be decreased since backoff time generated by random method may be different.

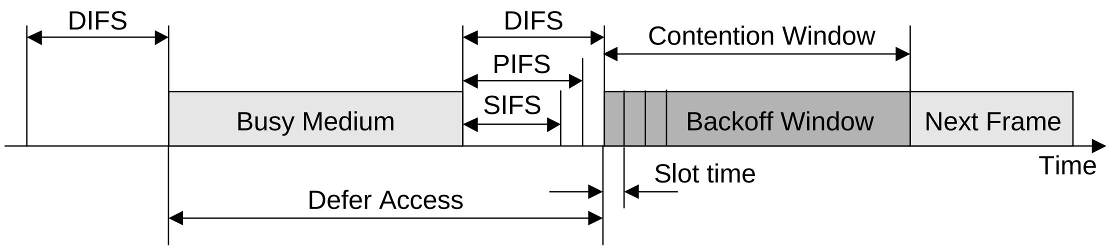


Figure 2.4 Basic access method

2.2.3. Random backoff time

In DCF mechanism, AP or STA must wait for a DIFS time before transmitting frame even though it detects that the channel media is idle. After waiting a DIFS time, AP or STA is permitted to transmit the frame if the channel media is idle. AP or STA must wait a DIFS again if it detects the channel media is busy after a DIFS time is expired or if it detects another AP or STA transmits its frame before a DIFS time is expired. AP or STA would enter the contention window (CW) if there is no another STA that is transmitting it frame before waiting time is expired. AP or STA would generate random backoff time during the contention window, as shown Figure 2.5 and it would transmit the frame after the backoff time is expired. The backoff time would be decreased progressively during the contention window (CW). The CW shall be closed and the backoff time would be saved to be decreased gradually during next CW if any AP or STA transmits its frame to channel media during this CW.

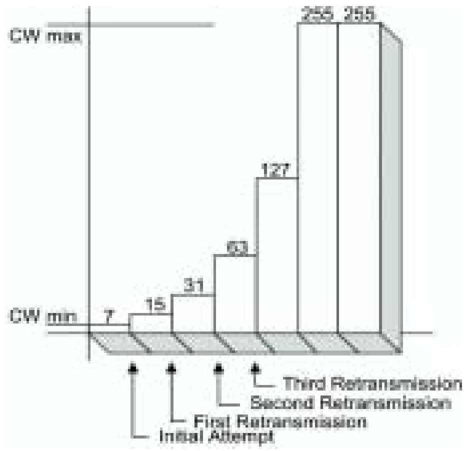


Figure 2.5 An example of exponential increase of CW

2.2.4. DCF access procedure

Besides CSMA/CA method IEEE 802.11 also employs RTS/CTS method to reduce the probability of collision in the DCF mechanism. The RTS/CTS method requests that transmitting STA or AP shall send a control frame, namely request to send (RTS), before transmitting its real frame and receiving STA or AP shall send immediately another control frame, namely clear to send (CTS), after receiving the RTS frame. If transmitter receivers CTS frame successfully, it means that the collision of RTS doesn't occur and then transmitting AP or STA would transmit the real frame. Even though the collision of RTS which consists of 20 bytes or the collision of CTS which consists of 14 bytes are occurred, the cost of retransmitting RTS or CTS would be lower than transmitting real frame.

The duration value carried by RTS frame provides other STAs or APs to estimate the time from transmitting RTS frame to the time when transmitter receives ACK frame of the first one of real data frames, as shown in Figure 2.6. Every STA or AP which receives the duration value carried by RTS must reset its net allocation value (NAV) to this value. Similarly, the duration value carried by CTS frame provides the time from transmitting CTS frame to the time when transmitter receives ACK frame of the first one of real data frames. Every STA or AP which NAV value is non-zero is not permitted to transmit its frame until its NAV value is decreased to zero value. Hence the probability of collision between transmitter and other STAs or APs shall be decreased since STAs or APs which NAV value is non-zero are blocked from transmitting any frame.

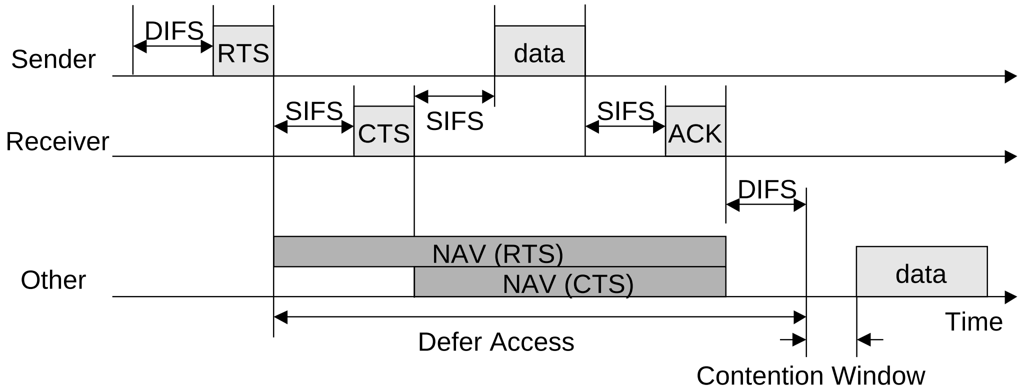


Figure 2.6 RTS/CTS/data/ACK and NAV setting

2.3. Point Coordination Function (PCF)

IEEE 802.11 provides PCF mechanism which employs contention free method to access the channel media to transmit time-bounded frame besides DCF mechanism which employs contention method to transmit frames. After a STA attempting to transport real-time data associates with AP, PC, which shall be implemented in AP, assigns the unique number, namely association ID (AID), to STA and records this AID to polling list of PC. Then PC would poll the STA in the polling list in order according to the order of polling list to ask STA if it need to transmit any data frame and thus STA in the PCF mechanism is only permitted to transmit its frame on condition that PC polls it.

2.3.1. CFP structure and timing

The PCF mechanism doesn’t need CSMA/CA, RTS/CTS, and NAV methods to reduce the probability of collision because the collision problem would not occur within polling method. The STA is permitted to transmit a frame within a MSDU size on condition that PC informs STA that it has the right to do it after PC registers STA to the polling list. If STA wants to join in the polling list of PC, it shall send association or re-association frame to AP within the DCF mechanism, as shown in Figure 2.7.

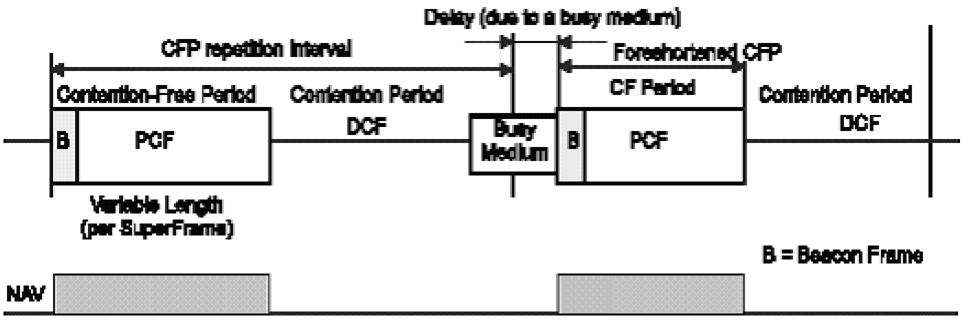


Figure 2.7 CFP/CP alternation

The STA shall get the cycle time of contention free period from CFPMaxDuration carried with Beacon broadcasted by AP periodically. If the contention free period (CFP) is longer than beacon interval, AP would send beacon frame during CFP and STAs recognize the CFPMaxDuration value carried with Beacon as remnant time of this contention free period.

Figure 2.8 illustrates that a super-frame is equal to two DTIM intervals. It means that one contention free period shall be started every two DTIM intervals and one DTIM interval is equal to three beacon intervals. Namely, the period of a super-frame is equal to six beacon intervals. AP would transmit six beacon frames and the contention free period is about equal to 2.5 beacon intervals.

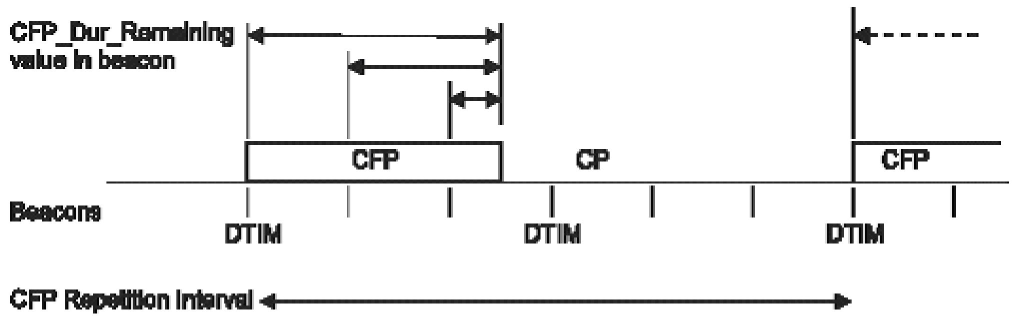


Figure 2.8 Beacons and CFPs

AP shall defer the predetermined beacon transmission time since some frame transactions are in progress while the target beacon transmission time (TBTT) arrived. It results in that the contention free period is forced to reduce and the lost time is equal to deferred time that beacon is deferred to broadcast and CFP would be closed at CFPMaxDuration timing and the deferred time shall be deducted from CFPDurRemaining value to ensure that CFP would be end at CFPMaxDuration exactly. The maximum time of deferred time is RTS frame + CTS frame + MSDU + ACK frame, as shown in Figure 2.9. If the STAs in the polling list are not overload, the contention free period may be end before CFPMaxDuration is coming and PC has the right to decide which STA has the right to transmit its more frame.

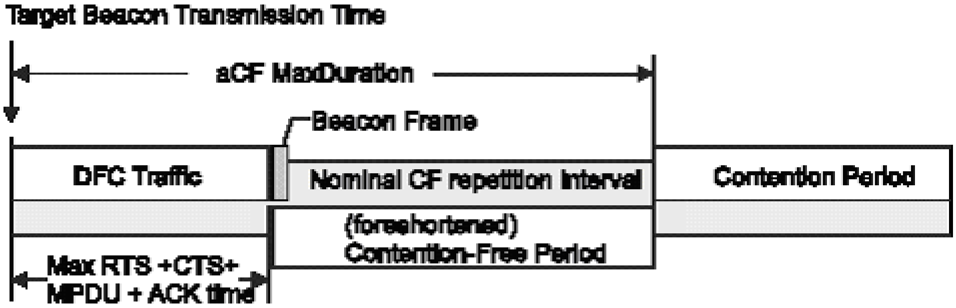


Figure 2.9 Example of delay beacon and foreshortened CFP

2.3.2. PCF access procedure

The PC transmits CF-Poll frame to inform STA joined in polling list about STA has the right to access channel media for transmitting one data frame. If polled STA hasn't any data to transmit, it replies AP with Null frame contained in ACK frame or if it is attempt to transmit data, it replies AP with data frame contained in ACK frame. If PC wants to transmit some data frames to polling STA, it transmits a unit of data frame contained in CF-Poll frame. After polled STA receives the CF-Poll frame, it transmits an ACK frame with a unit of data frame if it wants to transmit data frame or without data frame if it hasn't any data frame to transmit. AP would send ACK + CF-Poll frame if AP receives data frame from STA and wants to poll the next STA in the polling list. Furthermore, AP would transmit ACK + CF-Poll + Data frame if AP wants to transmit data frame to the next STA in the polling list. Figure 2.10 illustrates the typical example of frame transmission during the contention free period. The first order of frame transmission during the CFP is transmitting from PC to STA and second is transmitting back from STA to PC. Moreover, these orders are repeated until the end of CFP.

According to above description about PCF access procedure, it seems a very efficient mechanism to transmit real-time data and however, it derives a serious problem. If AP receives a data frame from STA during the CFP and the target STA of this data frame isn't in the polling list of this PC, it would occur that AP is unable to deliver this data frame during the CFP. In this case, AP shall save this data frame to its buffer and transmit it to target STA during contention period (CP) with DCF mechanism. Thus the real-time data would be saved and deferred at the process of the transportation and these break the rules of requirements of real-time data transportation.

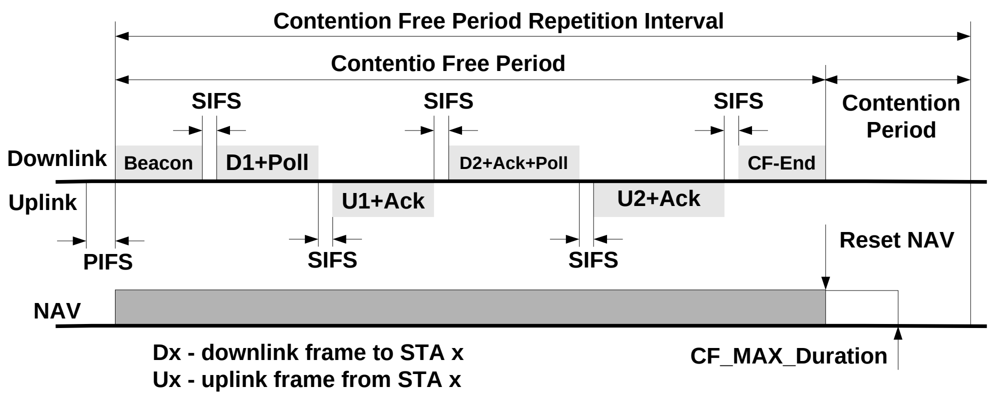


Figure 2.10 Example of PCF frame transfer

2.4. Related Works

O. Sharon and E. Altman [17] provided STRP mechanism to improve the performance of PCF mechanism. The PC divides the STAs in the polling list of PC to two kinds, namely active ring in which STA is permitted to transmit data as polling procedure and idle ring in which STA is permitted to transmit jamming noise as polling procedure to inform PC to change its type to active ring. PC polls two STAs, that one is in active ring and the other is in idle ring, at the same time during polling procedure. The polled STA in the idle ring transmits jamming noise during polling procedure and PC shall change the type of this STA from idle ring to active ring as receiving jamming noise signal. The most advantage of this mechanism is shortening the time to join to active ring and the biggest perplexity of this mechanism is near-far problem. SuperPoll [10] was provided to save bandwidth of wireless LAN. PC shall poll all STAs in the polling list once time per super-frame so a super-poll frame would consist of all members of the polling list. Every STA is permitted to transmit its data frame at its dedicated period and all other STAs covering the next STA are prohibited from transmitting its data frame even though the STA hasn't any data frame to transmit. Thus the waste of time and bandwidth is still existent.

According to the above rules of QoS, the system shall support priority of STAs. D.-J. Deng and R.-S. Chang [8] provided the architecture about priority DCF and the major method is shortening the backoff time of higher priority STA. Since the overlap of CW the STA of lower priority may join in polling list earlier that the STA of higher priority as the contention of two STAs with higher priority occurs. S.-T. Sheu and T.-F. Sheu [19] provided DBASE mechanism to solve above problem that the STA with lower priority may join in polling list earlier than STA with higher priority. The major method of DBASE mechanism is that contention occurs at the slots between PIFS and DIFS. Hence DBASE supports that real-time contention window is fixed and the maximum number is three and DIFS = SIFS + 5 * SlotTime. The STAs attempting to join in polling list may produce more contention since the few of slots. And the STAs under DCF mechanism would waste these slots time as real-time traffic is almost few.

In order to support the requirements of QoS, system must let real-time STA join in polling list as fast as possible. The method of collision resolution is better than the method of collision avoidance since the method collision resolution would be adjustable according to responses of STAs. J.-P. Sheu, C.-H. Liu, S.-L. Wu, and Y.-C. Tseng [18] modified initialization mechanism [15] and employed the method of copper coin throwing to solve collision problem. However, the process of copper coin throwing is quiet uncertain so their collision resolution process may be not stop forever.

CHAPTER 3

The Q-PCF Protocol

3.1. CFP structure and timing

Q-CFP mechanism divided CFP into three parts, namely prioritization period, collision resolution period, and polling period. The first two of three parts are integrated as registration period. During prioritization period, PC would execute a series of handshake to ensure that the STA of higher priority must join in polling list earlier than the STA of lower priority. During collision resolution period, PC would execute a series of handshake to ensure that the STA attempting to join in polling list can join within bounded time. During polling period, PC would transmit M-POLL frame to let STAs in the polling list transmit their data frame in order. At the end of polling period, PC would transmit CF-End frame to close contention free period and reset NAV value. In order to accord with IEEE 802.11, the shortest of CP time is assumed as the time that transmission of largest MPDU and a ACK frame. It's a worth notice that the busy of DCF would cause the deferment of CFP time, as shown in Figure 3.1.

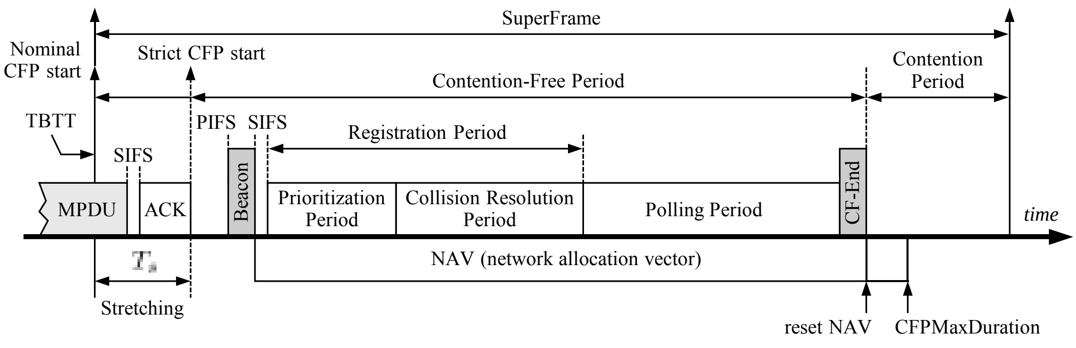


Figure 3.1 Superframe struct

Since the duration of CFPMaxDuration is limited, the bandwidth guarantees of STAs in the polling list may be interfered if the duration of registration period is boundless. Thus the run-time admission control is issued and designed to support PC with the ability to decide when to terminate the registration process.

3.2. Frame formats

Figure 13.2 depicts the MAC frame format of Q-PCF.

PE(Priority Enquiry)

(a) PE frame

Octets:	2	2	6	2	4
PE frame	Frame Control & Priority	RAB	BSSID	Δ_g	FCS

RE(Registration Enquiry)

(b) RE frame

Octets:	2	2	6	2	3	4
RE frame	Frame Control & Priority	RAB	BSSID	Δ_g	AddressPattern	FCS

PR(Priority Response)、RR(Registration Response)

(c) PR/RR frame

Octets:	2	2	6	2 D	2 G	4
PR/RR frame	Frame Control	AID	BSSID	Demanded Bandwidth	Guaranteed Bandwidth	FCS

M-POLL

(d) M-POLL frame

Octets:	2	2	6	2	4 × RecordCount		4
					Poll Record (4 octets)		
M-POLL frame	Frame Control	Duration	BSSID	Record Count	AID (2)	TXOP (2)	FCS

Data

(e) data frame (used in CFP)

Octets:	2	2 D	6	6	6	2	0～2312	4
data frame (used in CFP)	Frame Control	Demanded Bandwidth	Add. 1	Add. 2	Add. 3	Seq. Ctrl	Payload	FCS

To DS	From DS	Add. 1	Add. 2	Add. 3	Usage
0	1	DA	BSSID	SA	AP-to-STA traffic in a BSS
1	0	BSSID	SA	DA	STA-to-AP traffic in a BSS
0	0	DA	SA	BSSID	STA-to-STA traffic in a BSS

Figure 3.2 Frame format

3.3. Prioritization procedure

The priority levels are divided into H+1 levels from 0 to H. If some STA attempts to transmit some real-time data to somewhere and the priority level is bigger than zero, then it has an opportunity to join in the polling list. Figure 3.3 describes an example of how the Q-PCF supports priority level. First, PC transmits priority enquiry (PE) to inquire whether the STAs with H-th priority level want to join in the priority list and then transmits another priority enquiry (PE) to inquire whether the STAs with (H-1)-th priority level want to join in the priority list if the last PE is not replied and so on. If just only one STA with H-th priority level attempting to transmit real-time data responds to PE with priority response (PR) frame, the STA is permitted to join in polling list immediately and PC shall inquire the STAs with (H-1)-priority level. If there are at least two STAs responding to PE at the same time, PC would transmit registration enquiry (RE) frame to declare the starting of collision resolution period. During collision resolution period, PC shall execute a series of handshakes to find the STAs that attempt to transmit real-time data and produce the collision as responding to PE.

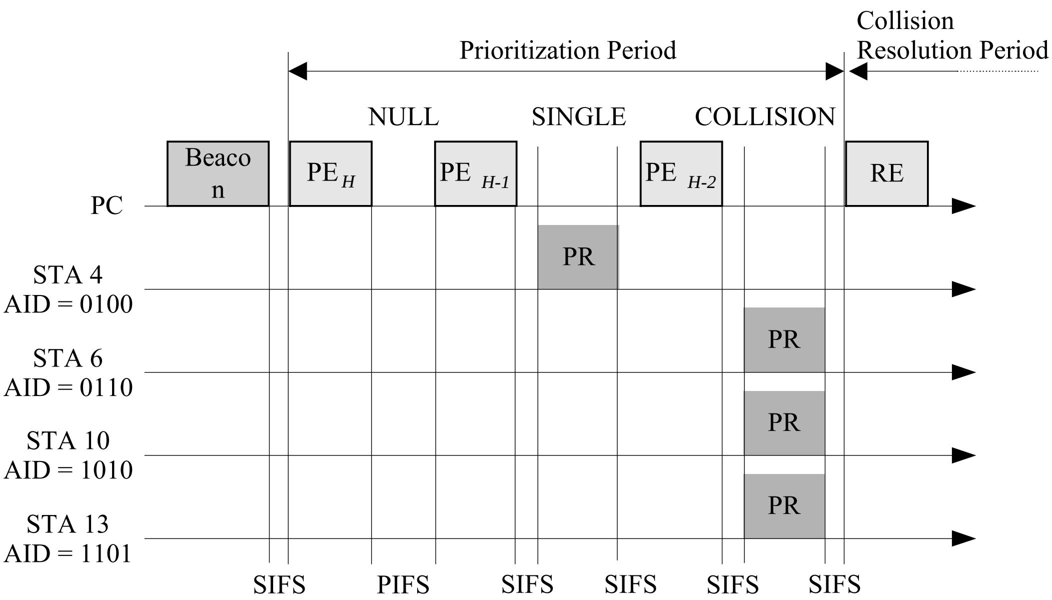


Figure 3.3 Priority period

3.4. Collision resolution procedure

This thesis adopts depth-first-search (FDS) traversal dimension tree splitting to solve collision resolution. The major theorem is that PC employs FDS method to find the STAs that produce the collision as responding to PE. Figure 3.4 gives an example that STA_6, STA_{10} and STA_{13} attempt to join in polling list and PC shall employ FDS method to find them. PC shall execute a series of handshakes to find STA_6, STA_{10} and STA_{13}. At the first step, PC inquires which STA has the H-th priority level (H is equal to H-2 in this example) and its address-pattern is {***0} and it attempting to join in polling list. If there is only one STA answering it, this STA is joined into polling list directly and then PC changes address-pattern to {***1} to inquire other STAs. If the collision is occurred, the PC shall reduce the range of address-pattern and shall change it to {**00}. Hence the STAs that result in collision can be found and joined into the polling list.

The major difference between Q-PCF and IEEE DCF is the strategy of solving collision. DCF mechanism employs collision avoidance method [2, 5, 11, 12, 15] to solve collision problem and it means that DCF mechanism attempts to use time delay to solve collision problem. Q-PCF mechanism employs collision resolution strategy to solve collision problem and Q-PCF also employs dynamic adjustment method. The worst cast of Q-PCF is occurred when the greater part of STAs with the same priority level wants to join in the polling list. However, the probability of above case is very low and the STAs that have joined in the polling list would not participate in contention process.

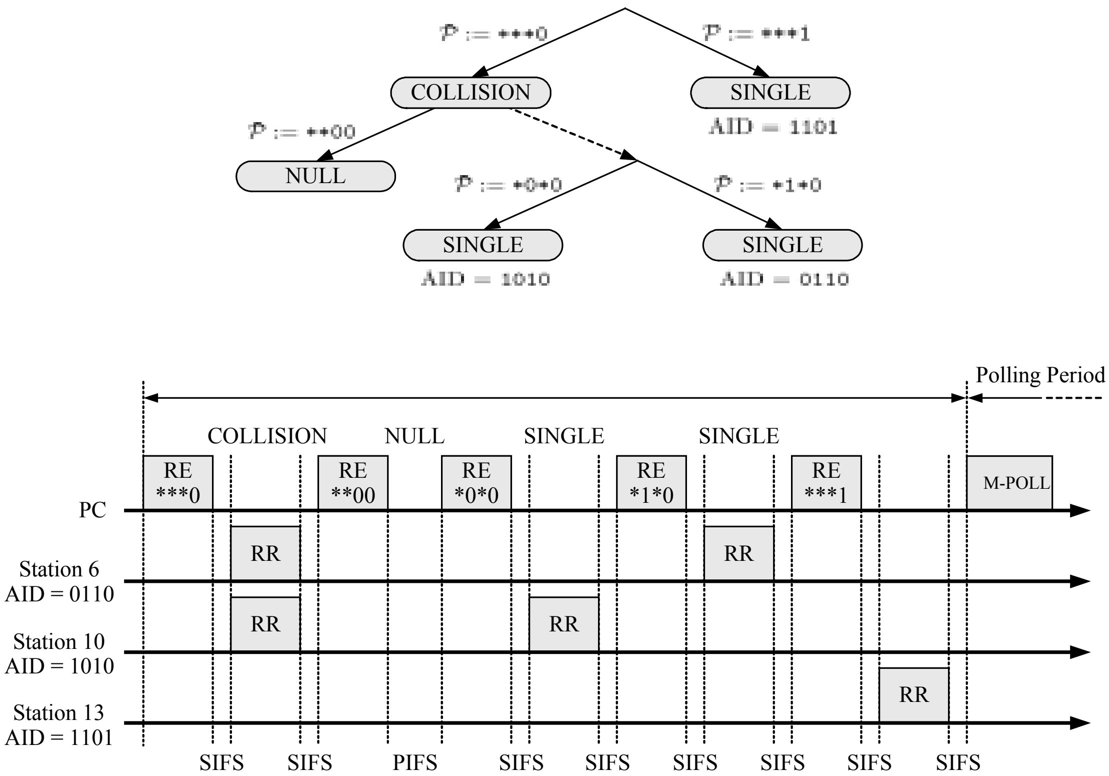


Figure 3.4 Collision resolution procedure

3.5. Polling procedure

After the end of collision resolution, the PC transmits the M-POLL frame and enters polling period. The M-POLL frame declares the order of STAs to be polled and transmission opportunity (TXOP) [9, 14] of these STAs. TXOP is defined as the longest time that polled STA can occupy. All polled STAs must listen to the channel carefully during this polling procedure. The polled STA shall transmit its data frame in order after the last STA finishes its transmission. Since the wireless LAN is easy to be interfered, the PC would detect that some polled STA doesn't transmit its data frame. It may be caused by STA failure and frame loss. In this case PC shall retransmit a new M-POLL frame during

PIFS period. This new M-POLL frame contains the remaining STAs in the polling list that doesn't get the right to access channel media. Thus PC still has the right of media control since Q-PCF permits PC to intervene in the polling procedure. Figure 3.5 illustrates the whole operation procedure during polling period. (a) After transmitting the M-POLL frame, STA_4, STA $_6$, STA_{10} and STA_{13} transmit their data frame in order. (b) After PIFS period, PC takes the right of media channel and retransmits a new M-POLL frame since STA_{10} doesn't transmit any frame.

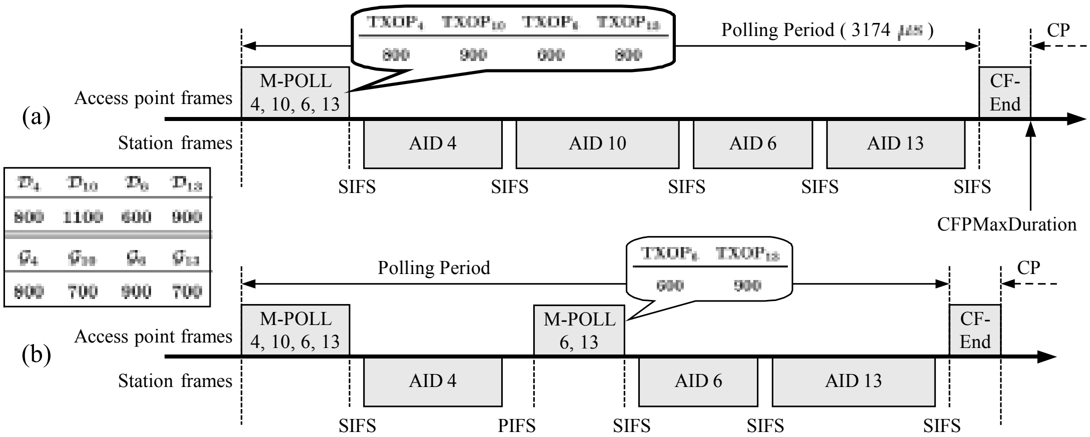


Figure 3.5 Polling period

3.6. Bandwidth allocation procedure

Q-PCF is designed to support real-time multimedia transportation. There are two characteristics of real-time multimedia transportation. First at all, data transportation is time bounded. If STA is unable to transmit real-time data within the bounded time, these data frames would be discarded. Second, the real-time data frames are great quantity and continuity of transportation is required. The real-time multimedia data are usually divided into two kinds, namely CBR and VBR. The bandwidth (demanded bandwidth) of CBR STA is fixed during every CFP. The demanded bandwidth (abbreviated as D) of VBR STA is dynamic. In order to get the guarantee of demanded bandwidth from PC, STA must inform PC about bandwidth guarantee (abbreviated as G) during registration period. Obviously, if D > R, the bandwidth of STA is not satisfied. Thus VBR STA should evaluate what G value is appropriate. G value is bigger and the number of STAs in the polling list is less. Symbol ε [16] is assumed the probability that the bandwidth of VBR STA is unable to be satisfied during a CFP so $Pr[D>G] < \varepsilon$. VBR STA would evaluate which ε is acceptable and produce G value according to defined ε. Since the required bandwidth of CBR STA is the same, the equation $G(CBR) = D^*(1-\varepsilon)$ is provided. Current papers are unable to guarantee the bandwidth of particular STA and it is easy to acquire this guarantee since Q-PCF employs the declaration of (D, G) value.

The following description will show how to provide (D, G) value. Figure 3.6 shows the process of providing (D, G) value. First, at the starting of registration, STA shall provide (D, G) value to PC.

Second, STA shall provide the next D value used during the next CFP as polled every time. When STA wants to be removed from polling list, it should set the field of more data to 0 during transmission of data frame. PC would remove the STA from polling list after receiving the data frame with 0 value of more data.

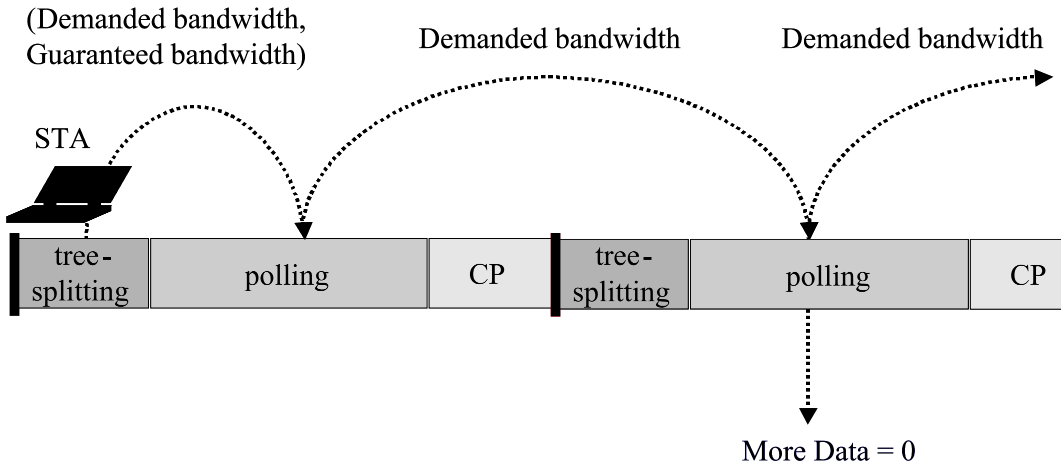


Figure 3.6 Demand Bandwidth

After STA declares its (D, G) value, PC shall calculate how many bandwidth (TXOP) is appropriate to distribute to every admitted STA. The strategy of Q-PCF is that remaining bandwidth is divided into STAs in the polling list under proportion of requirement after all bandwidth guarantees are satisfied. Assume that there are m STAs in the polling list (noted as (1, 2 … m)) and (D_i, G_i) is noted as (D, G) value of STA_i. PC shall calculate the following Y value before entering polling period.

$$Y = CFPMaxDuration - (Ts + PIFS + T_{beacon} + SIFS + T_{reg} + T_{M\text{-}POLL} + SIFS + T_{CF\text{-}End})$$

Then PC calculates how much time (RSB) is reminded during polling period to distribute to extra-requiring STA.

$$RSB = Y - \sum_{i=1}^{m} (\min\{D_i, G_i\} + SIFS)$$

The formula of calculating TXOP is following where RSB is known.

$$TXOP_i = \begin{cases} D_i & \ldots\ldots\ldots\ldots\ldots D_i < G_i \\ \min\left\{ D_i, G_i + \left\lfloor RSB \times \dfrac{D_i - G_i}{\sum_{D_i > G_i} D_i - G} \right\rfloor \right\} & \ldots\ldots\ldots\ldots\ldots D_i > G_i \end{cases}$$

For example, as shown in Figure 3.7, the STA_6 and STA_{10} have extra-requirement of bandwidth. Both of G_6 and G_{10} are 10 units and D6 and D_{10} are 14 and 16 units respectively. Thus the RSB value is equal to 5 and 5 is distributed to STA_6 and STA_{10} under the proportion 4:6. Therefore, $TXOP_6$ and $TXOP_{10}$ are 12 and 13 units respectively.

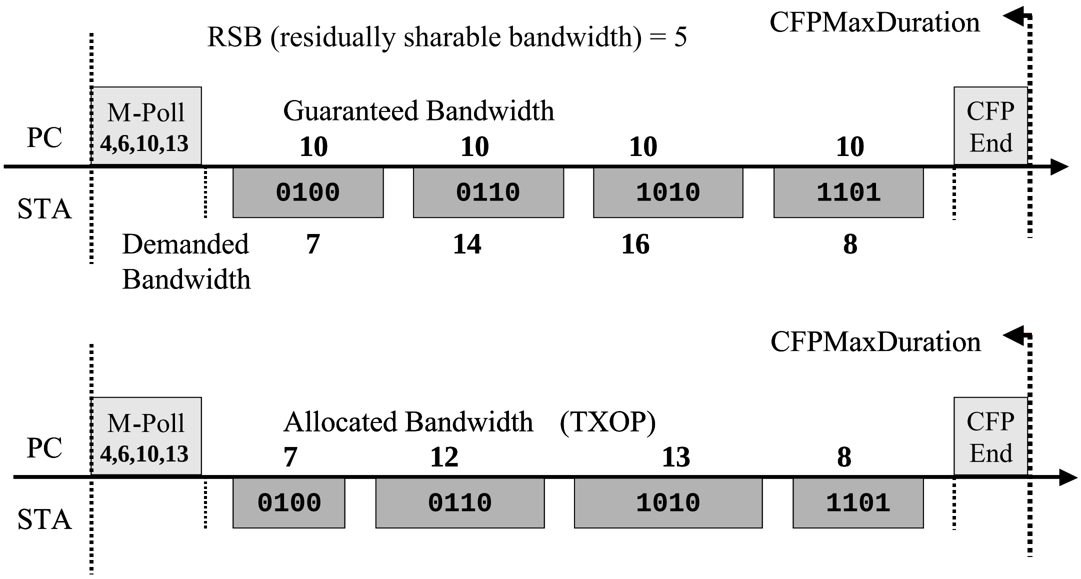


Figure 3.7 TXOP bandwidth allocation

3.7. Run-Time admission control

The based required bandwidth of STA_i is the minimum value of D_i and G_i (noted as $\min\{D_i, G_i\}$). In order to assure the based required bandwidth of admitted STA, the registration process cannot be executed too long and occupies the predefined time of polling period. On another side, PC shall handle the whole of registration procedure and PC cannot decrease the based requirement of bandwidth of every admitted STA since new STA is joined into the polling list. And the typically admission control [5, 7, 12] is not suitable here. The left part of Figure 3.8 shows the typical method of admission control and the right one is Q-PCF method. On typical method, STA transmits reservation request signal to PC for requirement of bandwidth. PC would reject the request if PC detects that the remaining time of CFPMaxDuration is not enough. This rejection procedure results in the failure of frame exchange and it would waste the rare bandwidth. On the Q-PCF method in the right side of following figure, frame exchanges are always valid so it doesn't waste the bandwidth.

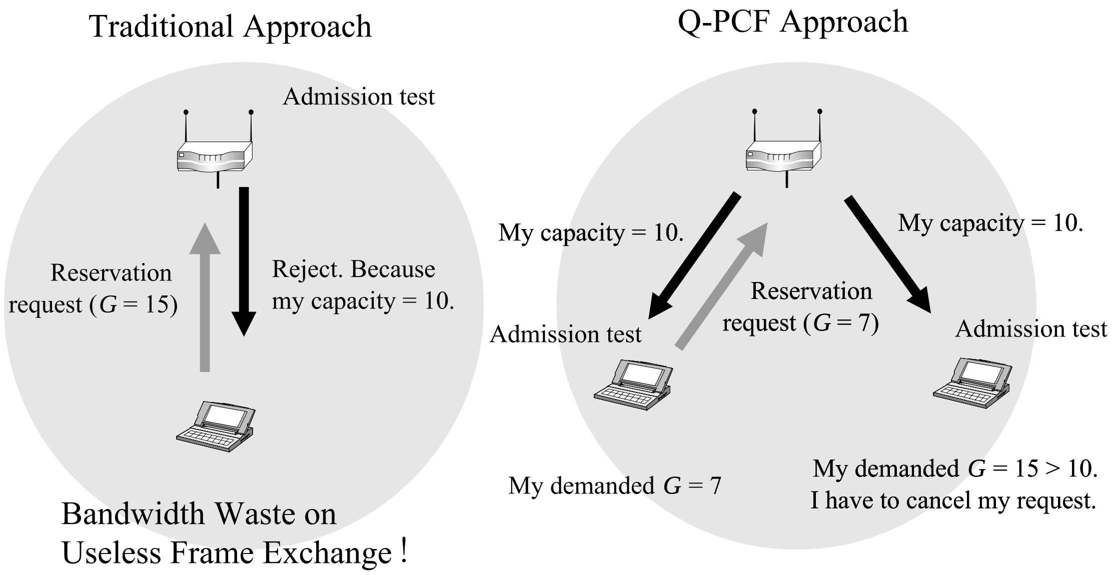


Figure 3.8 Poll method

The following will describe the advanced method of Q-PCF. PC would broadcast the remaining available bandwidth (RAB) to every STA and STA attempting to require bandwidth must compare the remaining bandwidth with the required bandwidth to determine whether it transmits PR or RR frame. If the requirement bandwidth is bigger than RAB, STA must not transmit PE or RE frame. The following will describe the operation of admission control used by Q-PCF. PC would transmit PE or RE frame with RAB value, and every STA attempting to join into polling list must execute admission test. If min{Gi, Di} < RAB, STAi can response to the PE or RE with PR or RR respectively. Otherwise, STAi must wait for the next CFP to transmit its require. PC would execute the next PE/PR or RE/RR handshaks according to the responses of STAs.

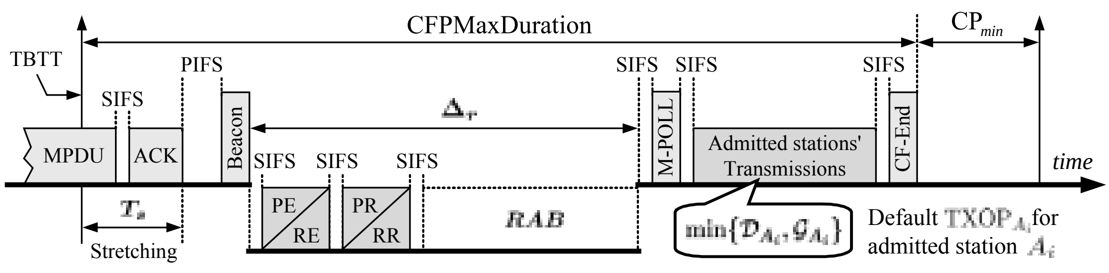


Figure 3.9 Run-Time admission control

The detailed algorithm refers to the Figure 3.10. It is worth to notice that PC must calculate two auxiliary variables, namely Δ_r and Δ_g, before transmitting the beacon.

$$\Delta_r = \text{CFPMaxDuration - }\left[\mathrm{T_S} + O_{CFP} + \sum_{i \in L} (SIFS + \min\{G_i, D_i\}) \right]$$

$$\Delta_g = \text{CFPMaxDuration} - \left[\hat{T}_s + O_{CFP} + \sum_{i \in L} (SIFS + G_i) \right]$$

where $O_{CFP} = \text{PIFS} + T_{beacon} + T_{M\text{-}POLL} + T_{CF\text{-}End} + 2*\text{SIFS}$

After broadcasting the beacon, the PC computes Δ_g, Δ_r, and the variable
$RAB := \Delta_r - (\delta_1 + \delta_2 + 3 \times \text{SIFS})$;
/* The variable RAB denotes the remaining available bandwidth
if the PC proceeds to the next PE/PR or RE/RR handshake. */
while ($\Delta_g > 0$ **and** $RAB > 0$ **and** (registration process is not finished)) {
The PC sends the PE/RE frame and announces (Δ_g, RAB);
/* On receiving the PE/RE frame, each active real-time station, say A_k,
takes the following admission test. */
if ($\mathcal{G}_{A_k} \le \Delta_g$ **and** $\min\{\mathcal{G}_{A_k}, \mathcal{D}_{A_k}\} \le RAB$)
Station A_k replies the PR/RR frame and declares ($\mathcal{D}_{A_k}, \mathcal{G}_{A_k}$);
$status$:= **receive**(PR or RR);
/* The PC updates the channel state variable $status$ according to
received PR/RR frames. */
switch ($status$) {
case SINGLE:
The PC places the real-time station A_k on the polling list;
$\Delta_g := \Delta_g - (\text{SIFS} + \mathcal{G}_{A_k} + \frac{4\times8}{\text{CDR}})$;
$\Delta_r := \Delta_r - (\delta_1 + \delta_2 + \min\{\mathcal{G}_{A_k}, \mathcal{D}_{A_k}\} + 3 \times \text{SIFS} + \frac{4\times8}{\text{CDR}})$; **break**;
/* Note that the length of the M-POLL frame will increase by 4 bytes
(32 bits) if a new real-time station is admitted. */
case NULL:
$\Delta_r := \Delta_r - (\delta_1 + \text{SlotTime})$; **break**; // PIFS = SIFS + SlotTime.
case COLLISION:
$\Delta_r := \Delta_r - (\delta_1 + \delta_2 + 2 \times \text{SIFS})$; **break**;
}
$RAB := \Delta_r - (\delta_1 + \delta_2 + 3 \times \text{SIFS})$;
}

Figure 3.10 Admission control algorithm

CHAPTER 4

Simulation Models

The Q-PCF and EQ-PCF simulator is coded in C language. The simulator would simulate the operations of MAC exactly and evaluate the performance of Q-PCF or EQ-PCF with different parameters further. The method of simulator is Event-Driven Scheme and the great parts of simulating parameters are extracted from Direct Sequence Spread Spectrum (DSSS) [13] of IEEE 802.11. The simulating time is $1.8*10^8$ (us). The simulating time is also noted 7200 super-frames with the unit of super-frame since the time of a super-frame is 25 ms. The number of STAs is 256 included STAs transmitting general data, CBR data and VBR data.

In order to simplify the complex environment of wireless LAN, this thesis makes three hypotheses in this simulation. (H1) Every STA must determine which priority level should be taken to transmit data during initial period and is prohibited from changing the priority level. The priority levels are divided into three kinds, namely general data (non-real-time data), CBR and VBR. (H2) All STAs have already associated with an AP before the simulator starts to be executed. (H3) The time period that the signal is transmitted in the air and the quality of signal upon the distance from AP to STA are ignored.

4.1. Traffic Models

Traffic Models are divided into three traffic models, namely Data (general data, non-real-time data), CBR (constant bit rate data), and VBR (variable bit rate data) upon the priority level of transportation. Every traffic model has its exclusive characteristics since its applications are different from others. Then the coding is also different and this sector would describe these traffic models respectively in the following statements.

4.1.1. Data traffic

The priority level of data traffic model is the lowest one of three and it employs the DCF mechanism to transport non-real-time data. To implement data traffic model, the following hypotheses are also given. (1) all STAs employing data traffic model must use RTS/CTS mechanism to confirm whether the data transmission is allowed before transmitting the real data frames. (2) Every STA is permitted to transmit data frames with fixed size of 2312 bytes per transportation. (3) The space time between the end of transmission and the first of transmission of next STA is determined via Poisson distribution. Table 4.1 illustrates the system parameters used in the data traffic model.

Table 4.1 System parameters used in the simulation

Parameter	Value	Unit
Channel bit rate	11	Mbps
Superframe length	25	ms
SIFS	10	us
PIFS	30	us
DIFS	50	us
SlotTime RTS frame length	20 20	us bytes
CTS frame length	14	bytes
ACK frame length	14	bytes
(CWmin, CWmax)	(31;1023)	slots
Reassociation Request frame length	38	bytes
Reassociation Response frame length	34	bytes
Beacon frame length	57	bytes
PE frame length	18	bytes
PR frame length	16	bytes
RE frame length	20	bytes
RR frame length	19	bytes
M-POLL frame length	16 + 4 * polling list size	bytes
CF-End frame length	20	bytes

4.1.2. CBR traffic

The CBR traffic model has the highest priority level and it employs the PCF mechanism to transmit real-time data such as voice data. To implement CBR traffic model, two hypotheses are given in this simulation. (1) The required data rate of CBR STAs is fixed. (2) The statuses of STAs in the CBR traffic model are divided into tow status, namely talkspurt and silent status and the front one occupies $1*10^6$ us (that is 40 frame); the rear one occupies $1.35*10^6$ us (that is 54 frame). The STA in the talkspurt status means that this STA is transmitting real-time data now via CBR traffic model and the STA in the silent status means that this STA is idle. Table 4.2 shows the traffic parameters for the CBR model.

Table 4.2 Traffic parameter values for the CBR models

CBR Traffic Parameter	Value	Unit
Conversation length	$1.8*10^8$	us
Principle talkspurt	$1.0*10^6$	us
Principle silent gap	$1.35*10^6$	us
Data bit rate (CBR)	64	Kbps
Maximum voice frame tolerable delay	25	ms

4.1.3. VBR traffic

The priority level of VBR traffic model is located between that of data traffic model and that of CBR traffic model and the VBR traffic model employs PCF mechanism to transmit real-time data with lower priority such as video data. To implement VBR traffic model, two hypotheses are also given. (1) The data rate of VBR STA is produced from truncated exponential distribution so the data rate of VBR is floating and the range of data rate is located from 120 Kbps to 420 Kbps. (2) According to the characteristics of multimedia transportation, the holding value [1][19] is produced from exponential distribution method and the STA would produce data bit rate in addition after the end of holding value. Table 4.3 shows the traffic parameters for VBR model.

Table 4.3 Traffic parameter values for the VBR models

VBR Traffic Parameter	Value	Unit
Peak bit rate	420	Kbps
Minimum bit rate	120	Kbps
Mean bit rate	240	Kbps
Mean state holding time	160	ms
Mean video call length	$1.8*10^8$	us
Maximum video frame tolerable delay	50	ms

4.2. Event-Driven Scheme

In general, the new theorem is unable to be tested or implemented in the real environment immediately. The performance and testing results of new theorem or mechanism may be acquired via the exclusive simulator if the performance and results of proof is necessary. The results produced by the simulator are meaningless if the variations between simulator and the real environment are too huge. The concept of Event-Driven scheme is to shorten the variations between simulator and the real environment and to represent the action of real environment. Since the environment of networking is variable, the loading of simulator to process all messages would become very heavy if simulator wants to consider all changes from networking. This research provides a programming mechanism to solve above problem. This mechanism adopts event-driven architecture to coding the simulator and this architecture is called as Event-driven scheme. This simulator simulate the events of wireless LAN MAC protocol faster than the real ones since the reductions of time to transmitting in the air and time of authentication and association and so on.

In order to understand the based operations of Event-driven scheme, first at all, the data structures used in Event-driven scheme must be realized. As shown in the following codes, this is a dynamic data structure that is linked with link list. All data used in Event-driven scheme would be saved in this dynamic data structure. Since using the dynamic memory allocation, the scheme doesn't need to occupy large memory size.

```
struct event_list_type
{
        char state;        // block
        char event;        // process
        int clock;         // the time when the event occurs
        short int nodeid;   // association ID
        struct event_list_type *next;    // point to the next data structure
};typedef struct event_list_type e_type;
e_type *first_d;
```

The following statement would describe the relation between block (state) and process (event). The block is a based functional unit and a block is implemented by at least one process. The following example would explain the relation between block and process concretely. The whole transmission of data is called as SEND block and this block has two incidents to be taken care. The first incident is processing the data transmission and the second one is processing the response from recipient(s) after transmitting the data. Thus the processing of these two incidents are called as process(SEND_DATA, RECV_ACK) and that is the event in the simulator. What time the event occurs depends on the relations among block, event, clock and nodeid.

During the initial period, the simulator program creates a time variable, namely gclock, and reset it to 0. Besides, every STA attempting to transmit or receive frames would insert an event to the dynamic link list and STA issuing the event should describe what block the event belongs to, what AID(coded as nodeid variable) it is, and when(coded as clock variable) the event should be executed. Every inserting event must sort the whole link list according to the clock variable to find its allocation of the list and then insert itself to list, as shown in Figure 4.1. After all STAs issues their events to link list, the program starts to extract the first event from the dynamic link list and this event should contain what block should be entered and what process in the block should be executed. The program assigns the value in the clock variable to the gclock variable after extracting the event and then executes the process(es) in the block designated by the event. The simulator produces another event after executing this event. And then the program extracts the second event from link list to process it. Thus every STA would produce another event and insert it to link list after current event of STA is processed. The simulator would not stop executing until the value in the gclock variable is equal to the predefined end time.

Additionally, it is an important concept that the control right of program is held by STA or AP when program is inserting the event to link list and the control right of program is held by main program when program is extracting the event to link list for processing it.

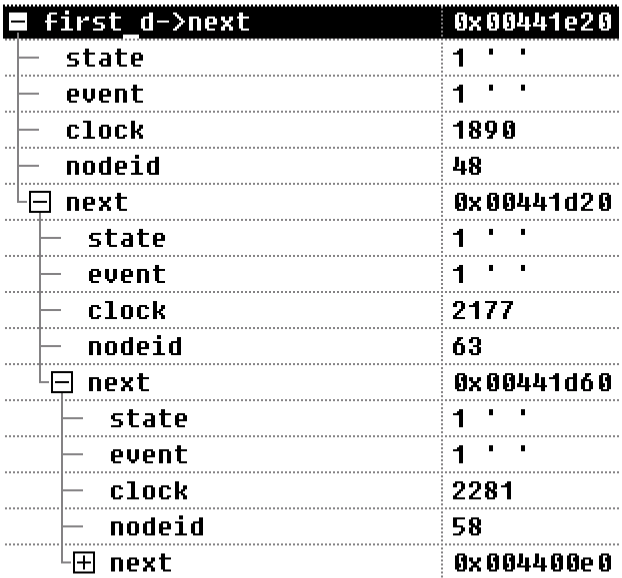


Figure 4.1 List of struct

4.2.1. Data struct

The DCF, Q-PCF and PCF of every STA have their own major data structure to record their status and these data structures are closely linked with the whole simulation. The following statements would describe these data structures.

DCF:

As shown in the following codes, this data structure records the status of STA under DCF mechanism.

```
typedef struct
{
int nav;              // recode the NAV value
        short int retry;         // count the retry times
short int partner;       // designate the recipient
}node_type;
```

Q-PCF:

As shown in the following codes, this data structure records the status of STA under Q-PCF mechanism.

```
struct node_state{
    short int   IDbit[LOGMAX];        // binary AID value
    short int   active;               // whether does the STA belong to CBR or VBR
    short int   pri;                  // value = 2, STA is CBR; value = 1, STA is VBR
    int         db;                   // bandwidth demand
    short int   enter;                // whether has the STA joined into polling list
    short int   more_data;            // the time point when CBR STA wants to join into polling list
                                      // or withdraw from polling list
    short int   more_data_count;      // the time counter when CBR STA joins into polling
                                      // list or withdraw from polling list
    int         loss;                 // the number of frames isn't transmitted
    short int   hold;                 // holding time
    int         TXOP;                 // TXOP value
    int         db2;                  // bandwidth demand + original buffer value
    double      regist_time;          // the time that STA spent for joining into polling list
    int         B[2];                 // VBR STA puts the data which doesn't want to send
                                      // out into buffer
};
```

PCF:

As shown in the following codes, this data structure records the status of STA under PCF mechanism.

```
struct node_state{
    short int   IDbit[LOGMAX];     // binary AID
    short int   active;            // whether the STA belongs to CBA or VBA STA
    short int   pri;               // the value = 2, STA is CBR; the value =1, STA is VBR
    int         db;                // bandwidth demand
    short int   enter;             // whether the STA has joined into polling list
    short int   REASSOC_IN;        // the time point when STA join into polling list
    short int   REASSOC_OUT;       // the time point when STA withdraw from polling list
    short int   succ_send;         // record whether STA transmits Data successfully
    short int   unpoll_count;      // record that STA isn't polled this time
    int         buffer;            // data saved in the buffer when STA is not polled
    short int   more_data;         // value = 0, STA wants to join in polling list;
                                   //value = 1, STA wants to withdraw from polling list
    short int   more_data_count;   // the counter to determine that STA joins in or withdraw from polling list
    int         loss;              // the number of frames isn't transmitted
    short int   hold;              // holding time
    short int   generate;          // value = 0, Re-association is successful
    int         db2;               // bandwidth demand + original buffer value
    double      regist_time;       // the time that STA spent for joining into polling list
};
```

4.2.2. State machines for DCF MAC Blocks

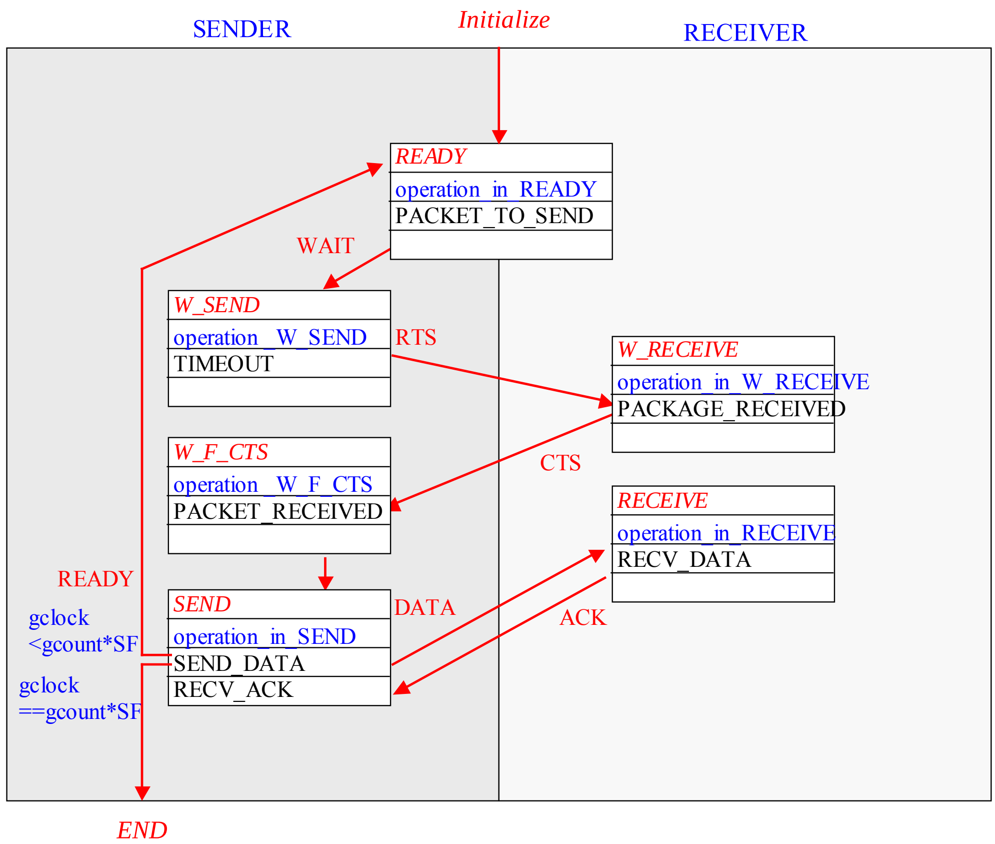


Figure 4.2 State machines for MAC Blocks

Initialize :

The following codes describe the total number of STAs is set as MAX_NODE and the time to issue the event is produced by Poisson distribution method.

```
for (i=0;i<MAX_NODE;i++){
        t=(int) poisson(rate);
        event_insert_d(i,PACKET_TO_SEND,t,READY);
}
```

READY STATE :

PACKET_TO_SEND EVENT :

The major function of this process is to calculate the backoff time for transmitter to insert RTS event. As shown in the following codes, the program picks a partner (recipient) via random method and then calculates the backoff time that is equal to current time plus a period produced by retry times.

```
node_d[id].partner=choose_a_partner_d(id);

event_insert_d(id,TIMEOUT,gclock_d+DIFS+backoff(node_d[id].retry),W_SEND);
```

W_SEND STATE :

TIMEOUT EVENT :

The major function of this process is to let transmitter send RTS frame. As shown in the following codes, the program compares NAV value with the current time. If the NAV value is not smaller than the current time, the NAV value is set and the RTS frame is prohibited from sending. And then the program would calculate the time to send RTS frame again and the time is equal to current time plus NAV value and backoff time. The backoff time is produced by retry parameter.

```
if (node_d[id].nav >= gclock_d )
        event_insert_d(id,TIMEOUT,node_d[id].nav+DIFS+backoff(node_d[id].retry),W_SEND);
```

If the NAV value is smaller than the current time, transmitter is permitted to send RTS frame. However, after receiving CTS frame it is realized whether the collision of RTS frame occurs. Figure 4.3 shows the collision process when STA_1 and STA_2 send their RTS frame simultaneously. STA_1 and STA_2 send their RTS frame at time point (1) so the collision would occurs from time point (1) to time point (2). After sending out RTS frame, STA_1 and STA_2 would wait for response of RTS frame respectively, that is CTS frame and they don't detect the collision. Since the collision occurred, every STA except STA_1 and STA_2 would keep silent and doesn't send any CTS frame. Thus neither STA_1 nor STA_2 would receive any CTS frame at the expectative time point (3). If neither STA_1 nor STA_2 receive any CTS frame at the time point (4), they assume the collision of RTS frame occurred and they must re-insert the RTS event to link list by re-calculating the backoff time. The retry counters of both STAs should increase 1. After time point (2), the channel media should be free and STA_3 attempting to transmit RTS frame would get the right to access this channel media and transmit RTS frame at time point (5).

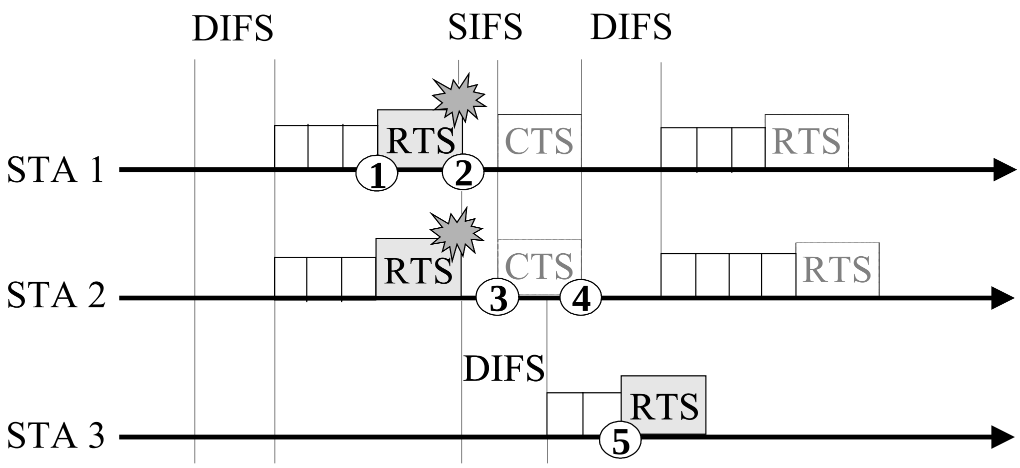


Figure 4.3 Forecast of Collision

This simulator doesn't simulate the real collision situation in the air so it takes a method to arrive the collision result. The program would check the clock variable member of all data structure in the dynamic link list to detect whether the collision occurs, as shown in the following codes.

```
while ((first_d->next != NULL)&&(first_d->next->clock == gclock_d)&&(first_d->next->event == TIMEOUT))
```

Besides the above two situations, the following codes describe the third situation that the collision doesn't occur. In that situation, the program set the NAV value and insert a CTS event to link list.

```
for(j=0;j<MAX_NODE;j++)
        node_d[id].nav=gclock_d+RTS_time+SIFS+CTS_time+SIFS+PACKET_time+SIFS+ACK_time;

event_insert_d(j,PACKET_RECEIVED,gclock_d+RTS_time,W_RECEIVE);
```

W_RECEIVE STATE :

PACKET_RECEIVED EVENT :

The major function of this process is to let recipient send CTS frame after receiving the RTS frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, a SIFS time and the time needed to transmit a CTS frame.

```
event_insert_d(psrc,PACKET_RECEIVED,gclock_d+SIFS+CTS_time,W_F_CTS);
```

W_F_CTS STATE :

PACKET_RECEIVED EVENT :

The major function of this process is to let transmitter prepare its data frame after receiving the CTS frame, as shown in the following codes. In the respect of time calculation, none of time variables needs to be calculated.

```
event_insert_d(psrc,SEND_DATA,gclock_d,SEND);
```

SEND STATE :

SEND_DATA EVENT :

The major function of this process is to let transmitter send its data frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, a SIFS and the time needed to transmit a data frame.

```
event_insert_d(pdst,RECV_DATA,gclock_d+SIFS+PACKET_time,RECEIVE);
```

RECV_ACK EVENT :

The major function of this process is to let transmitter receive ACK frame from recipient after transmitting the data frame, as shown in the following codes. After executing this process, it means that the data transportation is successful so the program would count the number of data frames (succ_AP) that are transmitted successfully. The meaning of successful transportation is that the data frame is transmitted to target recipient successfully regardless of the delivery of AP. This counter recording the successful transportation under DCF mechanism is only counted in the AP (the AID number of AP is 0) here. The program would issue another data transportation event of STA via Poisson distribution or one of AP directly without using Poisson distribution. In the respect of time calculation, the clock variable of the data transportation event is equal to addition of the current time, the time produced by Poisson distribution and the backoff time. The NAV value is set to current time and then all STAs can start to join in the contention.

```
if (id==0){
  t=0;
  succ_AP++;
}else{
  t=(int) poisson(rate);
}
event_insert_d(id,PACKET_TO_SEND,gclock_d+t,READY);

for(j=0;j<MAX_NODE;j++)
      node_d[j].nav=gclock_d;
```

RECEIVE STATE :

RECV_DATA EVENT :

The major function of this process is to let recipient send ACK frame after receiving data frame from transmitter, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, a SIFS and the time needed to transmit an ACK frame.

```
event_insert_d(psrc,RECV_ACK,gclock_d+SIFS+ACK_time,SEND);
```

End :

As shown in the following codes, the simulator of DCF mechanism would be closed and calculate the throughput of the whole network when the gclock variable is bigger than the predefined time (SIMU_TIME).

```
if (gclock_d>SIMU_TIME_d){
        Throughput_AP =succ_AP*(PACKET_time-(PHYheader+MACheader));
}
```

4.2.3. State machines for Q-PCF MAC Blocks

The data structure of PCF and DCF are the same, as shown in the following codes. The simulator declares two structure variables for arriving independent operations of simulators.

```
struct event_list_type
{
        char        state;              // block
        char        event;              // process
        int         clock;              // the time when the event occurs
        short int   nodeid;             // Association ID
        struct event_list_type *next;
};typedef struct event_list_type e_type;

e_type *first_d;                        // for DCF
e_type *first;                          // for PCF
```

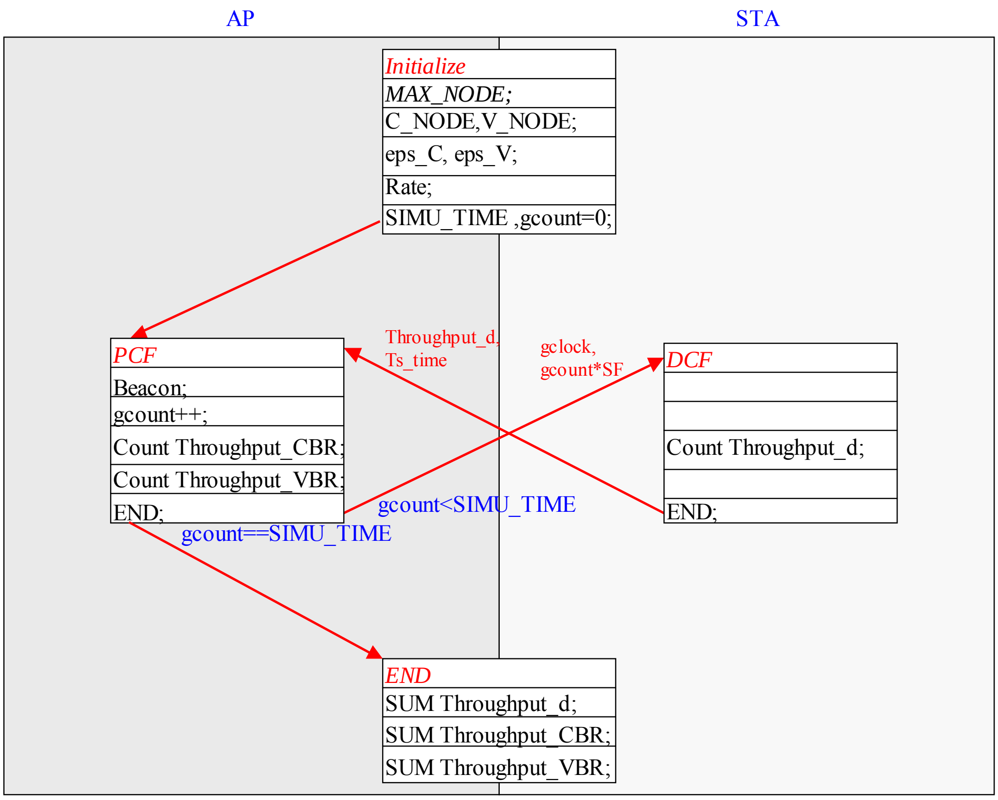


Figure 4.4 State machines for Q-PCF MAC Blocks

Q-PCF mechanism employs PCF mechanism to join STAs into polling list. Q-PCF blocks are divided into four blocks, namely Initialize, END, PCF and DCF blocks, where PCF and DCF blocks are executed in turn, as shown in the Figure 4.4. The simulator of Q-PCF mechanism would enter PCF block after every beacon and would change to DCF block when the end of CFPMaxDuration or when the data transportations are completed but the CFPMaxDuration is not end. There are two parameters to be delivered. One is gclock variable that is the current time of system. The other one is the end time of DCF (gcount*SF). If the delay of beacon occurs, the Delay time is a necessary parameter to be passed.

Initialize :

Determining the predefined value of MAX_NODE, C_NODE and V_NODE:

The simulator would ask user to enter the total number of STAs (MAX_NODE), the number of CBR STAs (C_NODE) and the number of VBR STAs (V_NODE). Then simulator determines which STA belongs to general data STA, CBR STA or VBR STA.

Determining the value of CBR and ε VBR:

The simulator would output the relation among ε_{CBR}, ε_{VBR} and Loss variable according to predefined ε_{CBR} and ε_{VBR}.

Rate :

The Rate is arrival rate that is the time produced by Poisson distribution.

SIMU_TIME :

The SIMU_TIME means the total time that simulator is executed such as 180 seconds for this research. The total frames that simulator is executed are 7200 frames since one frame is equal to 25 ms.

END :

As shown in the Figure 4.5, the simulator would calculate the results of DCF, CBR and VBR respectively.

```
CBR # in Polling List : 100
CBR # Register Time : 4673.640000

VBR # in Polling List : 5
VBR # Register Time : 93472.800000

DCF Succ=24964244, DCF Throughput=0.138671

CBR Succ=103435620, CBR Throughput=0.574563
CBR Loss=962640, CBR ErrorRatio=0.009221

VBR Succ=14917908, VBR Throughput=0.082866
VBR Loss=0,     VBR ErrorRatio=0.000000

ErrorRatio=0.008068
Throughput=0.796099
```

Figure 4.5 Result of simulation

State machines for Q-PCF MAC access point

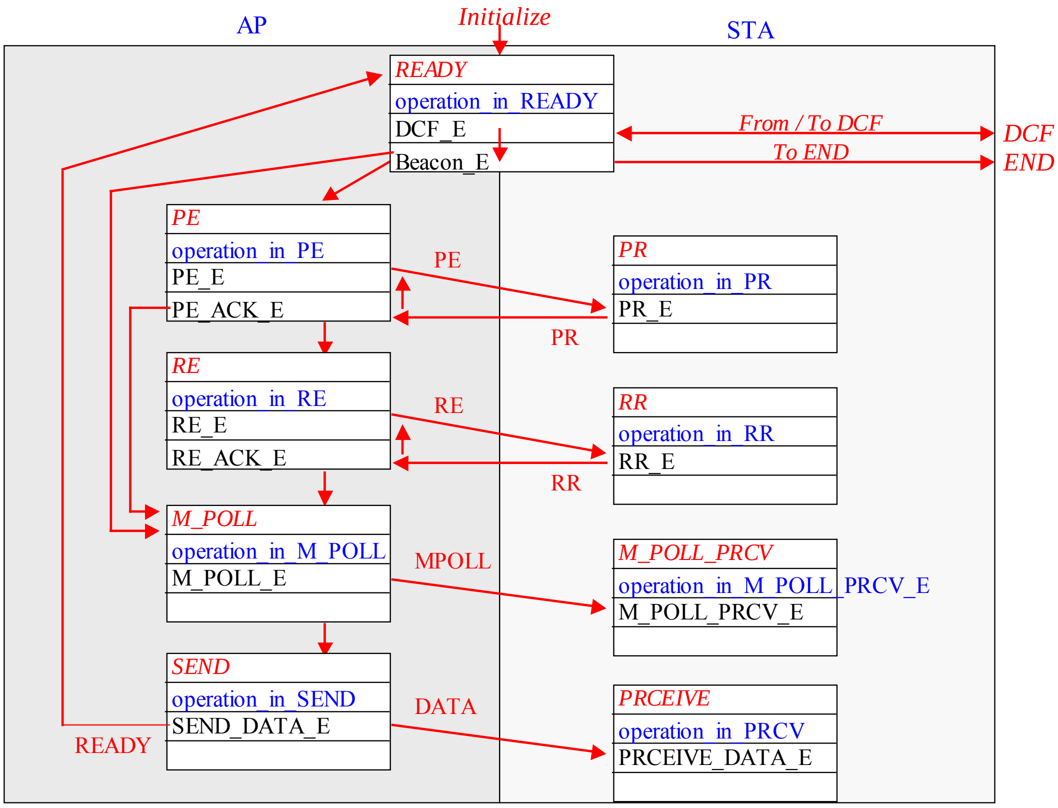


Figure 4.6 State machines for Q-PCF MAC access point

Figure 4.6 illustrates that unlike IEEE 802.11, the operations of joining into polling list under Q-PCF are changed from DCF to PCF mechanism.

READY STATE :

DCF_E EVENT :

The major function of this process is to let simulator program enter contention period (CP) as end of contention free period (CFP) and wait for the end of CP to enter CFP again, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, delay time caused by deferred beacon and a PIFS time.

As shown in the following codes, the program enters this process when the PCF mechanism is closed and the program would enter DCF mechanism directly.

```
Throughput_d=DCF(gclock,gcount*SF,&Ts_time);
event_insert(id,Beacon_E,gclock+Ts+PIFS,PRADY_S);
```

Beacon_E EVENT :

The major function of this process is to let PC transmit beacon, as shown in the following codes. After calculating the values of Δg (d_g), Δr (d_r) and RAB, the program compare Δg and RAB with zero. If one of Δg and RAB is not bigger than zero, it means that the bandwidth is fully occupied and the PC should transmit M-POLL frame. Otherwise the PC should transmit PE frame. In the respect of time calculation, the gclock variable is equal to addition of current time, the time needed to transmit beacon and a SIFS.

```
d_g = CFPmaxDuration-(RTS+CTS+maxMPDU+ACK+3*SIFS+OCFP+(3*k)+Gi_SUM);
d_r = CFPmaxDuration-(Ts+OCFP+(3*k)+Di_SUM);
RAB=d_r-(PE+PR+3*SIFS);
if ((d_g<=0)||(RAB<=0))
        event_insert(id,M_POLL_E,gclock+Beacon+SIFS,M_POLL_S);
else
        event_insert(id,PE_E,gclock+Beacon+SIFS,PE_S);
```

PE STATE :

PE_E EVENT :

The major function of this process is let PC transmit the frame to inquire STAs with specified priority level whether it wants to join in the polling list, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, the time needed to transmit a PE frame, and a SIFS.

```
event_insert(id,PR_E,gclock+PE+SIFS,PR_S);
```

PE_ACK EVENT :

As shown in the following codes, if one of Δg and RAB is not bigger than zero, it means that the bandwidth is fully occupied and the PC should transmit M-POLL frame. Otherwise, the PC enters the situation of transmitting PE frame or RE frame according to the six answers of PR frame.

```
if ((d_g<=0)||(RAB<=0)) {
        event_insert(id,M_POLL_E,gclock+M_POLL+SIFS,M_POLL_S);
        break;
}
if((pri==2)&&(collision==0)){
        pri--;event_insert(id,PE_E,gclock,PE_S);
}else if ((pri==2)&&(collision==1)){
        pri--;event_insert(id,PE_E,gclock,PE_S);
}else if ((pri==2)&&(collision==2)){
        event_insert(id,RE_E,gclock,RE_S);
}else if ((pri==1)&&(collision==0)){
        event_insert(id,M_POLL_E,gclock,M_POLL_S);
}else if ((pri==1)&&(collision==1)){
        event_insert(id,M_POLL_E,gclock,M_POLL_S);
}else if ((pri==1)&&(collision==2)){
        event_insert(id,RE_E,gclock,RE_S);
}
```

PR STATE :

PR_E EVENT :

The major function of this process is to let recipient transmit PR frame after receiving PE frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, a SIFS and the time needed to transmit a PR frame. The number of STAs transmitting PR frame would affect the RAB and Δr (d_r) values.

```
collision=2;
        event_insert(id,PE_ACK_E,gclock+PR+SIFS,PE_S);
        d_r = d_r-(PE+RE+2*SIFS);

collision=0;
        event_insert(id,PE_ACK_E,gclock+Slot_Time,PE_S);
        d_r = d_r-(PE+Slot_Time);

collision=1;
        event_insert(id,PE_ACK_E,gclock+PR+SIFS,PE_S);
        if (pri==2){
                d_g = d_g-(SIFS+CBRGB+3);
                d_r = d_r-(PE+PR+CBRGB+3*SIFS+3);
        }else{
                d_g = d_g-(SIFS+VBRGB+3);
                d_r = d_r- (PE+PR+Dmin(node[AID].db,VBRGB)+3*SIFS+3);
        }

RAB=d_r-(PE+PR+3*SIFS);
```

RE STATE :

RE_E EVENT :

The main function of this process is to let PC transmit frame to inquire which STA with specified priority level wants to join in polling list, as shown in the following list. In the respect of time calculation, the gclock variable is equal to addition of current time, the time needed to transmit a RE frame and a SIFS.

```
if ( stack_empty() !=1 )
        event_insert(id,RR_E,gclock+RE+SIFS,RR_S);
else
        event_insert(id,M_POLL_E,gclock,M_POLL_S);
```

RE_ACK EVENT :

As shown in the following codes, PC would judge whether d_g or RAB values is equal to or less than zero. If d_g or RAB values is equal to or less than zero, it means that the full bandwidth is occupied and PC should enter M-POLL period to transmit M-POLL frame. Otherwise, PC enters PE or RE period according to the six answers from response of PR frame.

```
if ((d_g<=0)||(RAB<=0))
        event_insert(id,M_POLL_E,gclock,M_POLL_S);
else
        event_insert(id,RE_E,gclock,RE_S);
```

RR STATE :

RR EVENT :

The major function of this process is to let recipient transmit PR frame after receiving RE frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, a SIFS and the time needed to transmit a PR frame. The number of STAs transmitting PR frame would affect the RAB and Δr (d_r) values.

```
collision=1;
        if (pri==2){
                d_g = d_g-(SIFS+CBRGB+3);
                d_r = d_r-(RE+RR+CBRGB+3*SIFS+3);
        }else{
                d_g = d_g-(SIFS+VBRGB+3);
                d_r = d_r-(RE+RR+Dmin(node[AID].db,VBRGB)+3*SIFS+3);
        }

collision=0
        d_r = d_r-(RE+Slot_Time);

collision=2;
        d_r = d_r-(RE+RR+2*SIFS);

RAB=d_r-(RE+RR+3*SIFS);

if (STATUS==0){
        event_insert(id,RE_ACK_E,gclock+RR+Slot_Time,RE_S);
else
        event_insert(id,RE_ACK_E,gclock+RR+SIFS,RE_S);
}
```

M-POLL STATE :

M-POLL EVENT :

The major function of this process is to let recipient transmit M-POLL frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition current time, the time needed to transmit original M-POLL frame, 3 us * the number of STAs attempting to be polled, and a SIFS.

```
event_insert(id,M_POLL_PRCV_E,gclock+M_POLL +( j *3)+SIFS,M_POLL_PRCV_S);
```

The following statement shows the calculation of TXOP value.

```
gamma=CFPmaxDuration - ((gclock - gzero) + (M_POLL + j *3) + (j * SIFS) + (SIFS + CF_End));
```

M-POLL_PRCV STATE :

M-POLL_PRCV EVENT :

The major function of this process is to let recipient receive the M-POLL frame from AP, as shown in the following codes. Since the recipient(s) receiving the M-POLL frame do(es) not need to response the M-POLL frame to AP, none of time value should be calculated.

```
event_insert(id,SEND_DATA_E,gclock,SEND_S);
```

SEND STATE :

SEND_DATA EVENT :

The major function is to let transmitter send its data frame, as shown in the following codes. In the respect of time calculation, the gclock value is equal to addition of current time, the time permitted to send data by PC (TXOP), and a SIFS.

```
event_insert(node[i].rpacket.dst,PRCV_DATA_E,gclock+node[ActiveSet[STA]].TXOP+SIFS,PRCEIVE_S);
```

The most important result in this process is to calculate Loss count. The define of Loss depends on CBR STA or VBR STA. The Loss of a CBR STA is confirmed when the data transportation is fail during **one** CFP. And the Loss of a VBR STA is confirmed when the data transportation is fail during **two** CFPs.

PRCEIVE STATE :

PRCEIVE _DATA EVENT :

The main function of this process is to let recipient deal with data frame sent from transmitter, as shown in the following codes. Since the recipient(s) receiving the data frame do(es) not need to response it to AP, none of time value should be calculated.

```
event_insert(psrc,SEND_DATA_E,gclock,SEND_S);
```

State machines for Q-PCF MAC STA

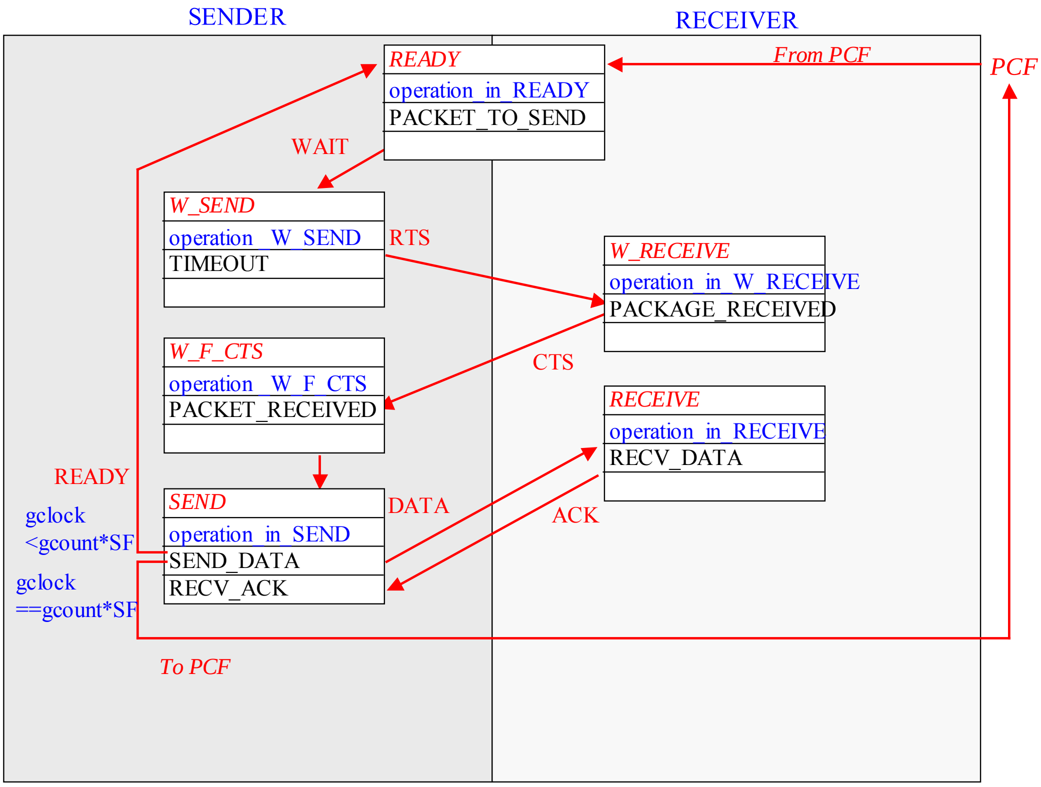


Figure 4.7 State machines for Q-PCF MAC STA

As shown in the Figure 4.7, the great parts of context are the same as above 5.2.1 sector for description of DCF mechanism. The most different part is that PC would ahead end PCF mechanism and change the right of control channel media to DCF mechanism, as shown in the following codes. AP should deliver tow parameters, namely current time (gclock variable) and expectative ending time (gcount*SF). And STA should return two values, namely the throughput of this time (Throughput_d) and the delay time of this time (&Ts_time).

```
Throughput_d=DCF(gclock,gcount*SF,&Ts_time);     // for AP
```

Before stating the DCF mechanism, the program would cancel all remaining events issued during CPF mechanism and reissue these events with new backoff time respectively, as shown it the following codes.

```
while ((first_d->next != NULL)&&(first_d->next->clock < gclock_p)){
        nid=first_d->next->nodeid;
        event_delete(first_d->next->nodeid,first_d->next->event);
        event_insert_d(nid,PACKET_TO_SEND,gclock_p,READY);
}
```

The transmitter is permitted to send RTS frame before the end of DCF mechanism so the program should know whether the time is over the end of contention period after finishing this data transportation.

```
if (gclock_d > SIMU_TIME_d)
          delay_time=gclock_d-SIMU_TIME_d;
```

The STA should return two values to AP, namely throughput and delay time. And the throughput value is equal to the number of successful data frames multiplied by the time needed to transmit a pure data frame excluded all PHY headers and MAC headers, as shown in the following codes.

```
Throughput=succ_d*(PACKET_time-(PHYheader+MACheader));
*Ts_time=delay_time;
```

4.2.4. State machines for 802.11 MAC Blocks

The architecture of IEEE 802.11 MAC is almost same as one of Q-PCF, as shown in the Figure 4.8. The most different part between IEEE 802.11 MAC and Q-PCF is that the STA followed IEEE 802.11 MAC attempting to join into the polling list employs DCF mechanism and does not employ PCF mechanism. Hence the first super-frame employs the DCF mechanism and the second one may employ PCF mechanism when at least one STA has joined into polling list. The DCF and PCF mechanism adhered to IEEE 802.11 MAC are executed in turn.

State machines for 802.11 MAC access point

In order to simplify the complex, this simulator make a hypothesis that all CBR and VBR STAs must join into polling list and all recipients also must join into polling list.

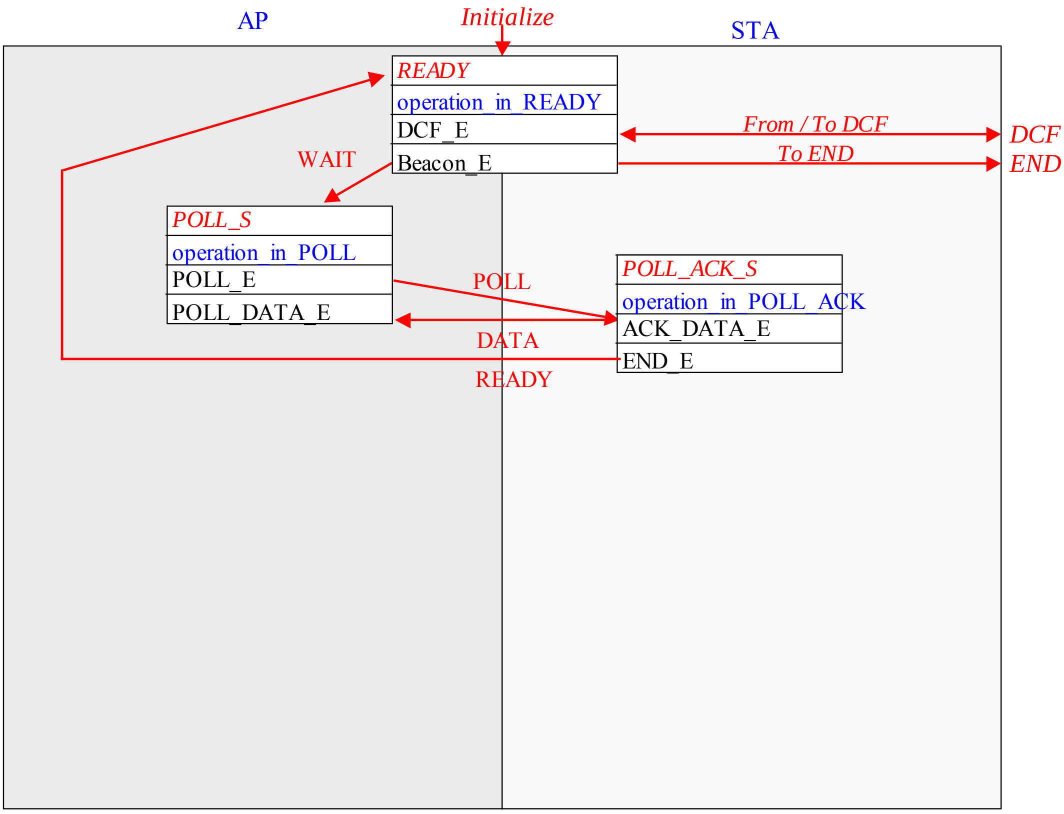


Figure 4.8 State machines for 802.11 MAC access point

READY STATE :

DCF_E EVENT :

The major function of this process is to delivery parameters between PCF and DCF mechanisms.

Beacon_E EVENT :

The major function of this process is to send a beacon frame.

POLL_S STATE :

POLL_E EVENT :

The major function of this process is to let PC send POLL frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, the time needed to transmit a POLL frame and a SIFS.

```
event_insert(psrc,ACK_DATA_E,gclock+POLL+SIFS,POLL_ACK_S);
```

POLL_DATA_E EVENT :

The major function of this process is to let PC send POLL+DATA+ACK frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, the time needed to send POLL+DATA+ACK frame and a SIFS.

```
event_insert(psrc,ACK_DATA_E,gclock+node[ActiveSet[STA]].db+SIFS,POLL_ACK_S);
```

POLL_ACK_S STATE :

POLL_ACK_E EVENT :

The major function of this process is to let recipient send DATA+ACK frame after receiving POLL frame or POLL+DATA frame, as shown in the following code. In the respect of time calculation, the gclock variable is equal to addition of current time, the time needed to send DATA+ACK frame and a SIFS.

```
event_insert(psrc,ACK_DATA_E,gclock+node[ActiveSet[STA]].db+SIFS,POLL_ACK_S);
```

END_E EVENT :

The major function of this process is to let PC send CF_END frame when the end of CFP is coming, as shown in the following code. In the respect of time calculation, the gclock variable is equal to addition current time, the time needed to send CF_END frame and a SIFS.

```
event_insert(pdst,DCF_E,gclock+CF_End,PRADY_S);
```

State machines for 802.11 MAC STA

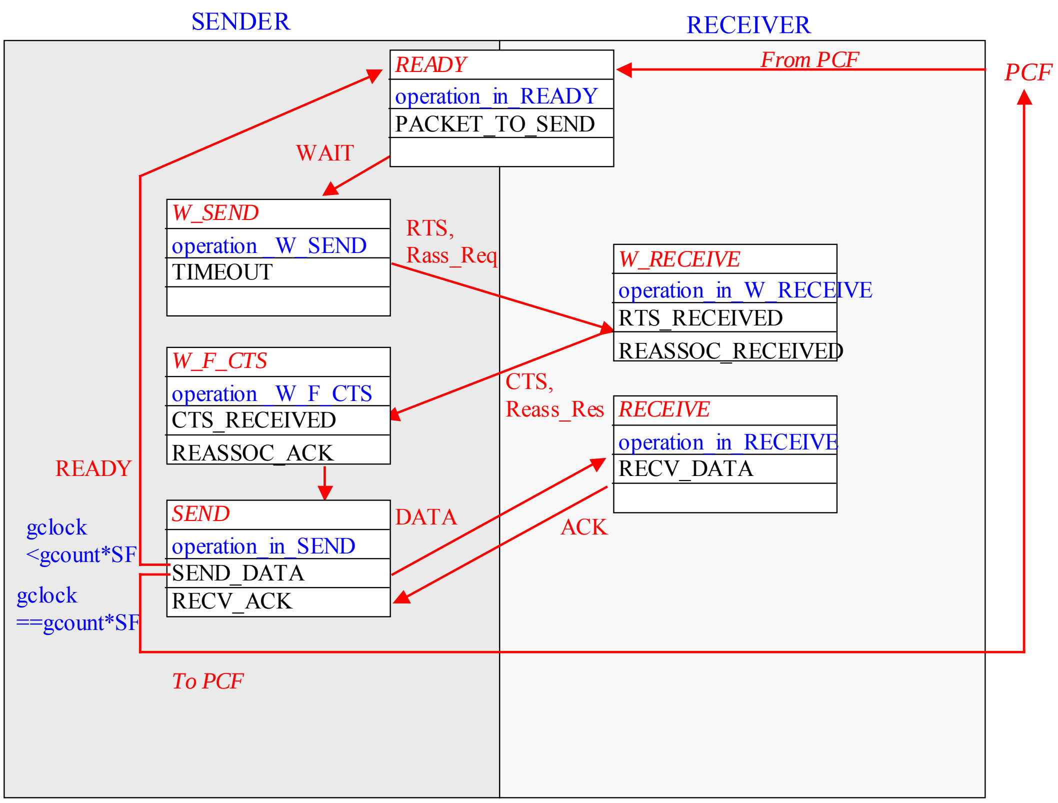


Figure 4.9 State machines for 802.11 MAC STA

As shown in the Figure 4.9, the great parts of this state machine are almost same as those of DCF parts of Q-PCF. The most different part is re-association request frame adding to RTS part and re-association response frame adding to CTS part.

W_SEND STATE :

TIMEOUT EVENT :

The context of this process is almost same as above description of RTS part in DCF mechanism of Q-PCF architecture. The most different part is that STA attempting to join in or withdraw from polling list would transmit a Re-association Request frame, as shown in the following codes. In the respect of time calculation, the gclock time is equal to current time plus the time needed to send a Re-association Request frame.

```
event_insert_d(id,REASSOC_RECEIVED,gclock_d+REASSOC_REQ_Time,W_RECEIVE);
```

W_RECEIVE STATE :

RTS_RECEIVED EVENT :

The major function of this process is to let recipient send CTS frame after receiving the RTS frame, as shown in the following codes. In the respect of time calculation, the gclock variable is equal to addition of current time, a SIFS and the time needed to send a CTS frame.

```
event_insert_d(psrc,CTS_RECEIVED,gclock_d+SIFS+CTS_time,W_F_CTS);
```

REASSOC_RECEIVED EVENT :

The major function of this process is to let recipient send Re-association Response frame after receiving the Re-association Request frame, as shown in the following codes. In the respect of time calculation, the system current time (gclock variable) is equal to addition of current time, a SIFS and the time needed to send a Re-association Response frame.

```
event_insert_d(psrc,REASSOC_ACK,gclock_d+SIFS+ACK_time,W_F_CTS);
```

W_F_CTS STATE :

CTS_RECEIVED EVENT :

The major function of this process is to let transmitter prepare data frame to send it, as shown in the following code. In the respect of time calculation, none of time variable needs to be calculated.

```
event_insert_d(id,SEND_DATA,gclock_d,SEND);
```

REASSOC_ACK EVENT :

The major function of this process is to let STA stop sending Re-association Request regardless of joining in or withdrawing from polling list after that operation is successful. In this process, the program would change the status of STA to polled status and join STA in polling list, as shown in the following codes.

```
if (node[id].REASSOC_IN==0){
        node[id].REASSOC_IN=1;
        node[id].enter=1;
}
if (node[id].REASSOC_OUT==0){
        node[id].REASSOC_OUT=1;
        node[id].enter=0;
}
```

CHAPTER 5

Simulation Results

It is necessary to give clear definitions of Throughput [3, 20] and Loss Rate first. Throughput means that the number of data sent to target recipient successfully regardless of delivery of AP is divided by the whole time of execution and its unit is bit per second (bps). The data contain three kinds of data with different priority levels, namely general data, CBR data and VBR data. Loss Rate means that the data should be sent successfully but they don't [12].

5.1. Throughput of DCF

Objective:

Testing the throughput in DCF mechanism with different number of STAs where AP is exist and is responsible to pass all data of STAs.

Condition:

DCF： $\lambda=10^2$ frame / sec (DCF)

STA：1,2,4,8,16,32,64,128

Result:

Figure 5.1 illustrates that as the number of STAs increases, less throughput can be acquired The reason of this situation is that all data from STAs must be delivered by AP and AP should join in contention procedure as it deliver these data.

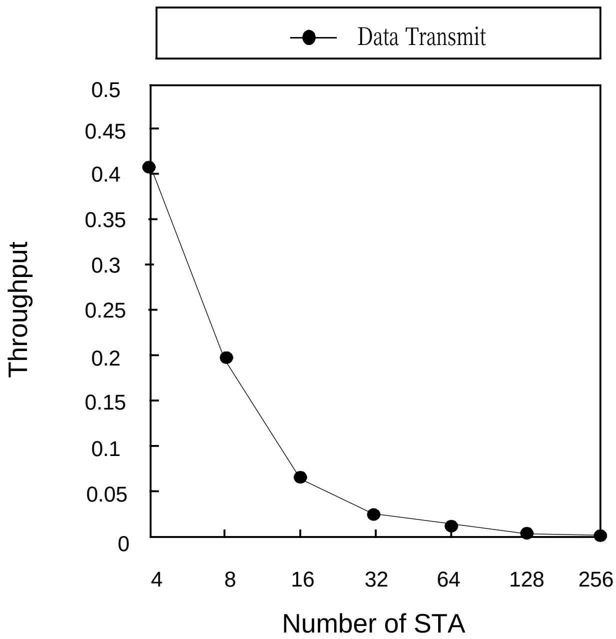


Figure 5.1 Throughput of DCF

5.2. Q-PCF

5.2.1. Ability of admission control

Objective:

(a) Calculating the number of CBR STAs and VBR STAs that join into polling list under Q-PCF mechanism and PCF mechanism respectively.

(b) Calculating the Loss Rate of CBR STAs and VBR STAs under Q-PCF and PCF mechanisms respectively.

Condition:

STA=256, ε_{CBR}=0, ε_{VBR}=0.5 , λ=0.1 frame / sec (DCF) and two kinds of parameters: (1) CBR=150,VBR=0 and (2) CBR=0,VBR=150

Result:

(a) The number of CBR STAs and VBR STAs that join in polling list under Q-PCF mechanism adheres to the theorem of Upper Bound. And that under PCF mechanism seems unlimited unless the number of STA in the DCF mechanism is too many.

(b) The Loss Rate of CBR STAs under Q-PCF mechanism is zero since ε_{CBR}=0 and the Loss Rate of VBR STAs under Q-PCF mechanism is also zero since ε_{VBR}=0.5[16]. However, under PCF mechanism the Loss Rates of CBR STAs and VBR STAs increase when the number of STAs attempting to join in polling list increases, as shown in the Figure 5.2.

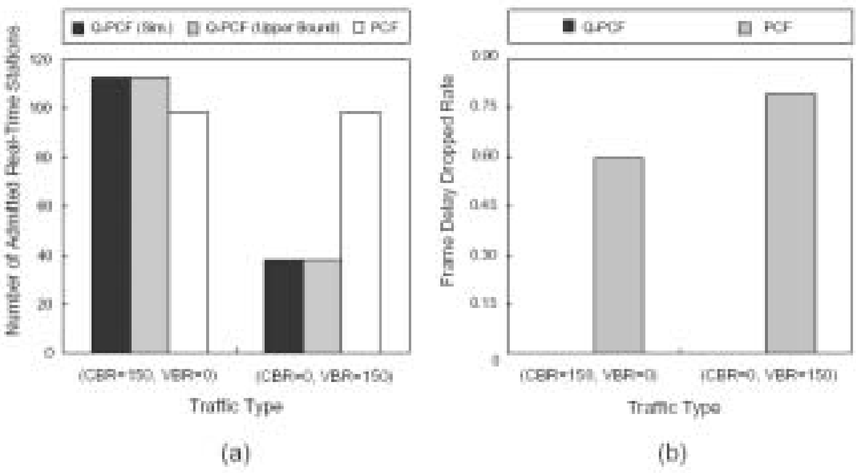


Figure 5.2 Throughput, Polling List and Loss for CBR and VBR

5.2.2. Effect of DCF

Objective:

(a) Calculating the number of CBR STAs and VBR STAs that join into polling list under **PCF** mechanism respectively when changing the time space of data transmission during DCF mechanism form long to short.

(b) Calculating the number of CBR STAs and VBR STAs that join into polling list under **Q-PCF** mechanism respectively when changing the time space of data transmission during DCF mechanism form long to short.

Condition:

STA=256, ε_{CBR}=0, ε_{VBR}=0.5, λ =1 or 10 frame / sec (DCF) and two kinds of parameters: (1) CBR=150, VBR=0 and (2) CBR=0, VBR=150

Result:

(a) The number of CBR STAs and VBR STAs joining in polling list under PCF mechanism is affected by the time space of data transmission under DCF mechanim.

(b) The number of CBR STAs and VBR STAs joining in polling list under Q-PCF mechanism is **not** affected by the time space of data transmission under DCF mechanism, as shown in the Figure 5.3.

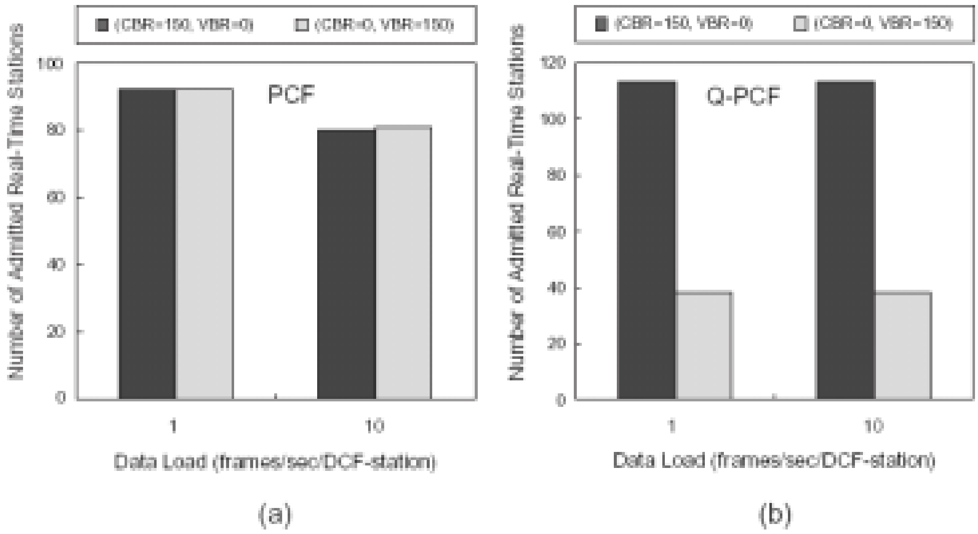


Figure 5.3 λ of DCF for PCF and QPCF

5.2.3. Effect of ε for CBR

Objective:

(a) Calculating the relation of Throughput and Loss Rate of CBR STAs under Q-PCF mechanism when changing ε value.

(b) Calculating the number of CBR STAs that join into polling list under Q-PCF mechanism when changing ε value.

Condition:

STA=256, CBR=150, VBR=0, λ=0.1 frame / sec (DCF)

ε_{CBR}=0, 0.02, 0.04 , 0.08 , 0.1 0.12

Result:

(a) The relation of ε value and Loss Rate is an direct ratio. If ε value is equal to 0.1, the Loss Rate is closed to 10%.

(b) The number of CBR STAs joining into polling list increases when the ε value increases. In fact, the Loss Rate increases too, as shown in the Figure 5.4.

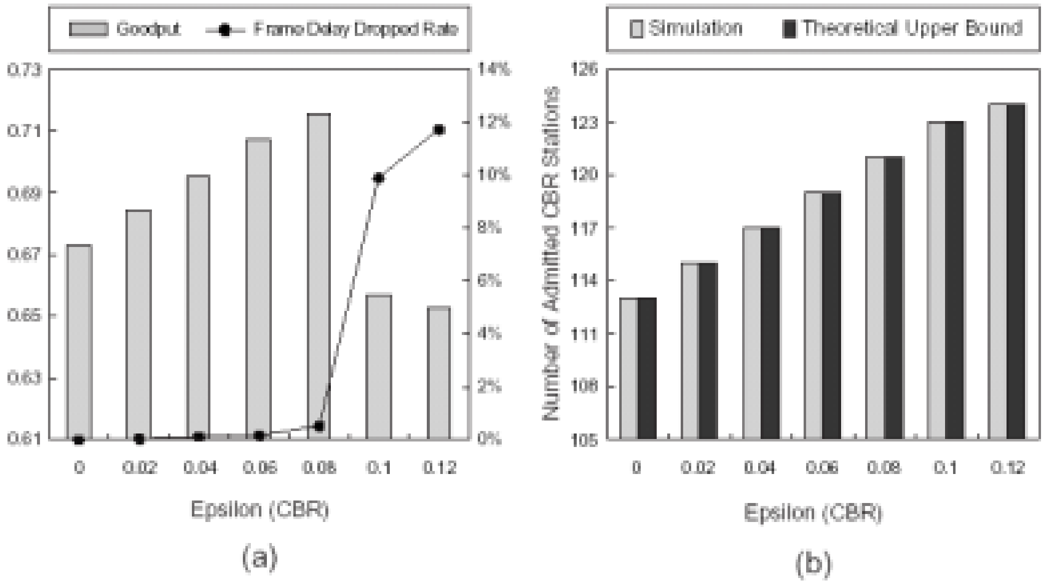


Figure 5.4 Throughput and Loss Rate for CBR

5.2.4. Effect of ε for VBR

Objective:

(a) Calculating the relation of Throughput and Loss Rate of VBR STAs under Q-PCF mechanism when changing ε value.

(b) Calculating the number of VBR STAs that join into polling list under Q-PCF mechanism when changing ε value.

Condition:

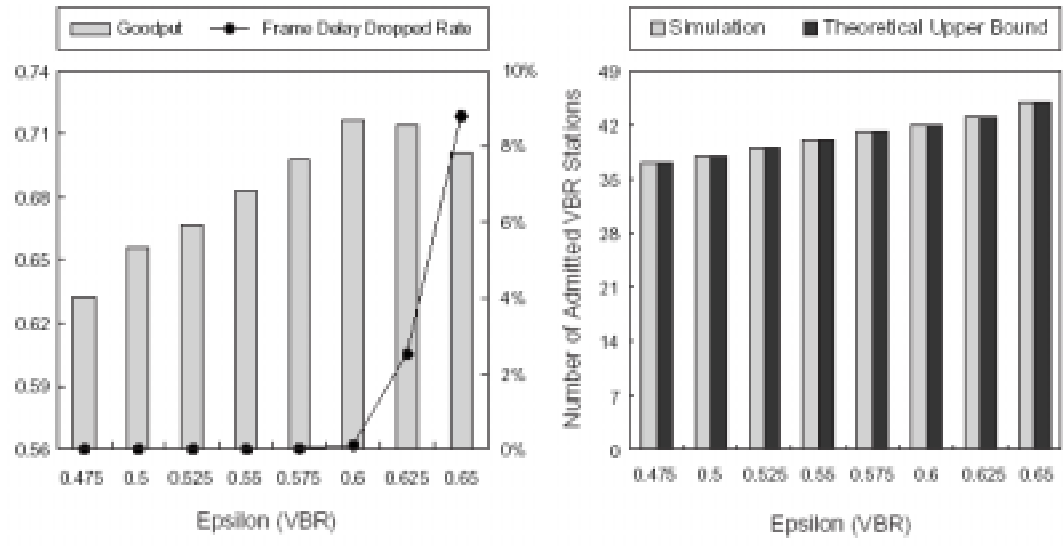
STA=256,CBR=0, VBR=150, λ =0.1 frame / sec (DCF)

ε_{VBR}=0.475, 05, 0.55 , 0.575 , 0.6 0.625 0.65

Result:

(a) The relation of ε value and Loss Rate is an direct ratio. If ε value is bigger than 0.5, the Loss Rate would occur.

(b) The number of CBR STAs joining into polling list increases when the ε value increases. In fact, the Loss Rate increases too, as shown in the Figure 5.5.



(a)

Figure 5.5 Throughput and Loss for VBR

5.2.5. Throughput and loss

Objective:

The objective is comparing the Throughput and Loss Rate under Q-PCF and PCF mechanisms respectively.

(a) Comparing the Throughput of Q-PCF and PCF mechanisms when the loading of network increases.

(b) Comparing the Loss Rate of Q-PCF and PCF mechanisms when the loading of network increases.

Condition:

STA=256, ε_{CBR}=0:02, ε_{VBR}=0.5, VBR = 50, CBR=100, λ=1 frame / sec (DCF)

Result:

(a) The throughput of Q-PCF mechanism is better than that of PCF.

(b) Even though ε=0.02, the Loss Rate of CBR STAs under Q-PCF mechanism is closed to zero since the CBR STAs have the chance to withdraw from polling list. The Loss Rate of VBR STAs under Q-PCF mechanism is closed to zero since ε_{VBR}=0.5. However, the Loss Rates of CBR and VBR STAs under PCF mechanism are very high, as shown in the Figure 5.6.

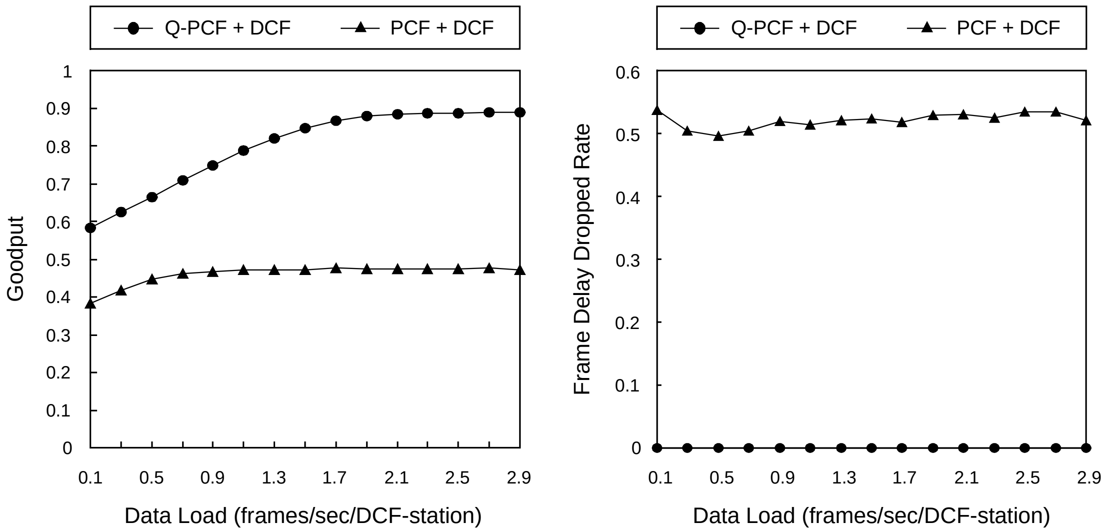


Figure 5.6 Performance Evaluation

CHAPTER 6

The EQ-PCF Protocol

Q-PCF is very efficient especially in control of bandwidth. However, Q-PCF mechanism assumes several hypotheses and these critical hypotheses seem conflicting with IEEE 802.11 standard that is the most popular standard of wireless LAN. Thus Q-PCF has the problem to be compatible with IEEE 802.11 standard and it is worth to discuss this problem. This chapter would start on the compatibility of Q-PCF with IEEE 802.11 and goes deep into it. Furthermore, this chapter would provide a modified Q-PCF mechanism to improve the compatibility of Q-PCF. Hence this improved mechanism is called as enhanced Q-PCF, abbreviated as EQ-PCF.

6.1. Registration Period

According to the theorem of Q-PCF, if the time of CFPMaxDuration is remaining, PC would execute the inquiry of priority after every beacon even though none of STAs wants to send CBR or VBR data. In this case, the overhead of bandwidth would increase and the situation becomes worst when there are a lot of priority levels, such as eight priority levels defined by IEEE 802.11e draft[16].

EQ-PCF mechanism divides the first part of three parts in the Q-PCF procedure, namely prioritization period, into two phases, namely join and prioritization phases, as shown in the Figure 6.1. The difference between join and prioritization phases is that AP would transmit join enquiry (JE) frame to inquire all STAs whether STA attempts to join in polling list. If there is some STA attempting to join in the polling list, STA should response to JE with join response (JR) frame regardless of its priority level. Figure 6.2 shows the frame format of join phase. PC would execute a series of handshakes with STA attempting to join in polling list during join phase and then enter to the prioritization period. The procedure of prioritization period is the same as Q-PCF. PC would execute a series of handshakes to ensure that the STAs with higher priority level would join in the polling list earlier than the STAs with lower priority level. The procedure of prioritization period in EQ-PCF is not necessary to be executed. PC shall ignore the procedures of prioritization period and collision resolution period if none of STA answers the inquiry that PC inquire STAs whether joins in polling list during join phase. After skip of procedures of prioritization and collision resolution periods, PC transmits M-POLL frame if there are data to be transmitted or end the CFP if none of data wants to be transmitted.

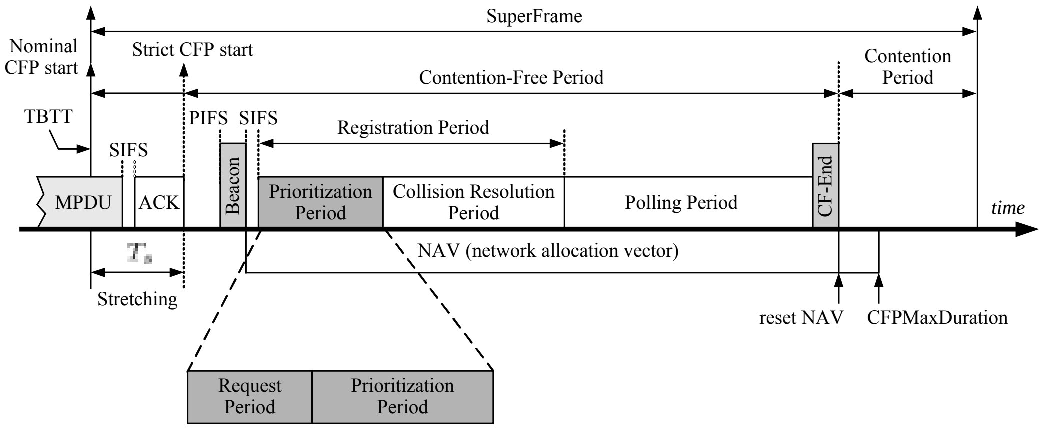


Figure 6.1 Superframe structure

The reason of adding inquiry frame during join phase is very simple. The probability that PC deals with the STA requirement of joining in polling list is lower than the probability that there is none of STA attempting to join in polling list. Furthermore, if the priority levels are increased more, the waste of bandwidth becomes worst. Hence the inquiry for every STAs during join phase would determine whether the procedure of prioritization and collision resolution period is executed.

(a)

Octets:	2	2	6	2	4
JE frame	Frame Control	RAB	BSSID	Δ_g	FCS

(b)

Octets:	2	2	6	2 $\mathcal{D}$	2 $\mathcal{G}$	4
JR frame	Frame Control	AID	BSSID	Demanded Bandwidth	Guaranteed Bandwidth	FCS

Figure 6.2 Frame format

6.2. Direct Transfer

Q-PCF mechanism makes a strange hypothesis that all STAs under infrastructure mode must be adjacent. The meaning of adjacency of STA_i and STA_j is that the range transmitting RF of STA_i and STA_j must cover each other. Thus STA_i and STA_j should transport data to each other directly without passing AP. However, according to IEEE 802.11 standard, all transportation between STAs must be delivered by AP under infrastructure mode since AP should have the highest priority to access the channel media.

It would save the bandwidth and increase the throughput that STAs can transport data to each other directly with point-to-point method. Hence, this thesis would improve this hypothesis and make it reasonable. There are two methods.

(1) Decreasing the transmitting range of RF of AP

According to IEEE 802.11 standard, all registered STAs must be covered under the range of AP and it is not necessary that the range of all STAs should cover each other. The range of all STAs under the range of AP would cover each other if the radius of range of AP becomes half and the range of STAs are the same and are double radius of AP, as shown in Figure 6.3. Thus the range of AP used by Q-PCF mechanism would become a quarter of normal range of AP used by PCF mechanism.

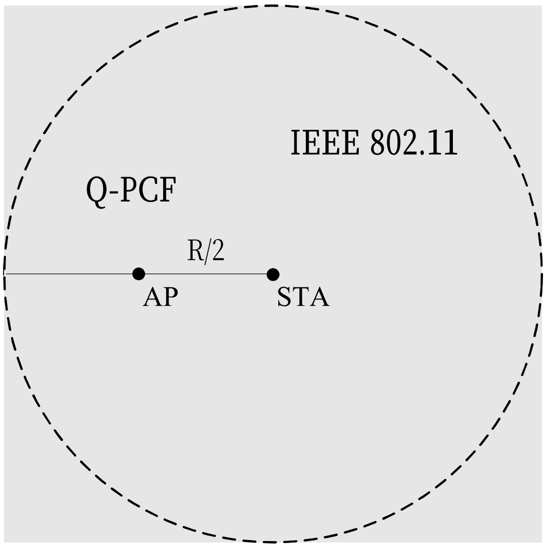


Figure 6.3 Transmission Range

(2) Calculating the probability of adjacency of STAs under the normal range of AP

This sector would calculate the probability of adjacency of STAs and input this value to simulator to build EQ-PCF architecture. The method of calculating probability employs the computer. First, defined radiuses of AP range and STA range is assumed and the radiuses of AP and STA ranges are the same. Then STA_1 and STA_2 are selected respectively with random method and STA_1 and STA_2 are covered under the range of AP. Finally, the distance between STA_1 and STA_2 are calculated and compared to determine whether the ranges of STA_1 and STA_2 cover STA_2 and STA_1 respectively, as shown in Figure 6.4. This thesis employs the computer to execute above algorithm with 100 million times and the results are the probability of adjacency of STAs under the normal range of AP is 58% and the probability of non-adjacency is 42%. That is only 58% of the point-to-point transportation between STAs under Q-PCF is permitted.

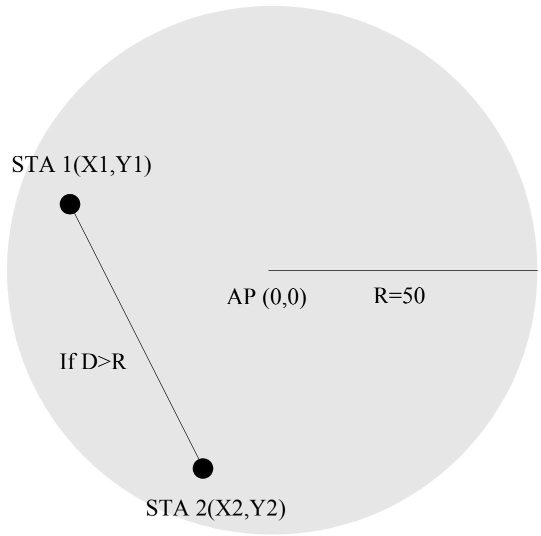


Figure 6.4 Distance measure

This one of objectives in this research is point-to-point transportation between STAs directly, as shown in Figure 6.5.

If the point-to-point transportation is prohibited, the throughput would increase to double times since the all transportation must be delivered by AP. The power-saving mechanism defined by IEEE 802.11 standard would disturb the point-to-point transportation since the transmitting STA may not know the power-saving mode of receiving STA when it wants to transmit data. Thus this problem will be discussed at the next sector.

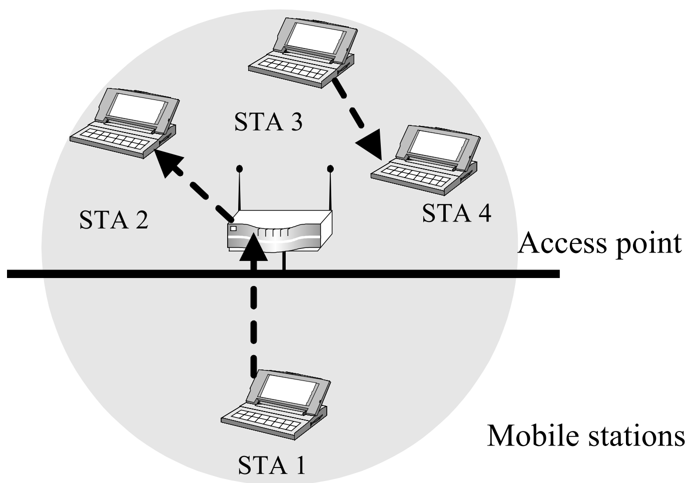


Figure 6.5 Data Transfer Path

6.3. Power-saving

For supporting point-to-point transportation under infrastructure mode, the Q-PCF mechanism assumes that all STAs do not support power-saving since the transmitting STA would not know the power-saving mode of receiving STA. However, for portable equipments and the standard of IEEE 802.11, power-saving mechanism is a very important mechanism.

According to IEEE 802.11 standard, AP shall transmit TIM contained in beacon to inform sleeping STAs about that AP wants to deliver the buffered data to STAs. After receiving TIM, STA transmits PS-Poll frame to inform AP about that STA is ready and awake to receive data. In this procedure, AP would not know the power-saving mode of receiving STAs.

It is acceptable that AP saves the data that should be transmitted to STAs under DCF mechanism since the transportation under DCF mechanism is not real-time data transportation. However, it is not acceptable under real-time transportation.

The six-byte destination address (DA) would be added to JR, PR and RR frames as shown in Figure 6.6.

(a)

	Octets: 2	2	6	6	2 D	2 G	4
JR frame	Frame Control	AID	BSSID	Add.	Demanded Bandwidth	Guaranteed Bandwidth	FCS

(b)

	Octets: 2	2	6	6	2 D	2 G	4
PR/RR frame	Frame Control	AID	BSSID	Add.	Demanded Bandwidth	Guaranteed Bandwidth	FCS

Figure 6.6 Frame format

AP needs a list to record the power-saving status of all STAs. If the destination STA is in sleeping mode, AP would reject STA that attempts to transmit data to destination STA to join into the polling list. If the destination STA is not in sleeping mode, AP should accept the request of STA to join in the polling list if the remaining of DCFMaxDuration is exist. Hence M-POLL frame as shown in Figure 6.7, would the destination address of STAs besides the order of polled STAs and the transmission time. The destination STAs in the M-POLL frame should not enter the sleeping mode.

(c)

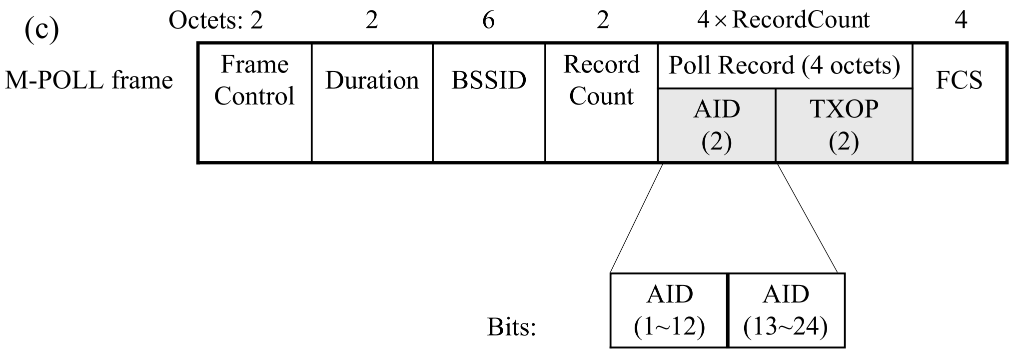


Figure 6.7 M-POLL Frame format

6.4. Necessary DCF

Since the procedure of joining into polling list is transformed from DCF mechanism to EQ-PCF mechanism, DCF mechanism is meaningless. However, the channel selection, authentication and association should be executed in DCF mechanism. There are two issues to discuss or notice if DCF mechanism wants to be ignored. First at all, without DCF mechanism the STAs shall join into polling list before they wants to transmit any frame that embraces registration and authentication frames. And this situation would cause the increase of loading and collision so it is more difficult for STAs to join into the polling list. Second, not all data frame from STAs are real-time multimedia and not all STAs need request guarantee of bandwidth. And this situation would waste the bandwidth of wireless LAN. Hence DCF mechanism is needful.

CHAPTER 7

Performance Evaluation of EQ-PCF

This chapter would show the result of EQ-PCF simulation and the state machine of simulation.

7.1. State machines for EQ-PCF MAC Blocks

Figure 7.1 illustrates the MAC blocks of EQ-PCF and the their data flows.

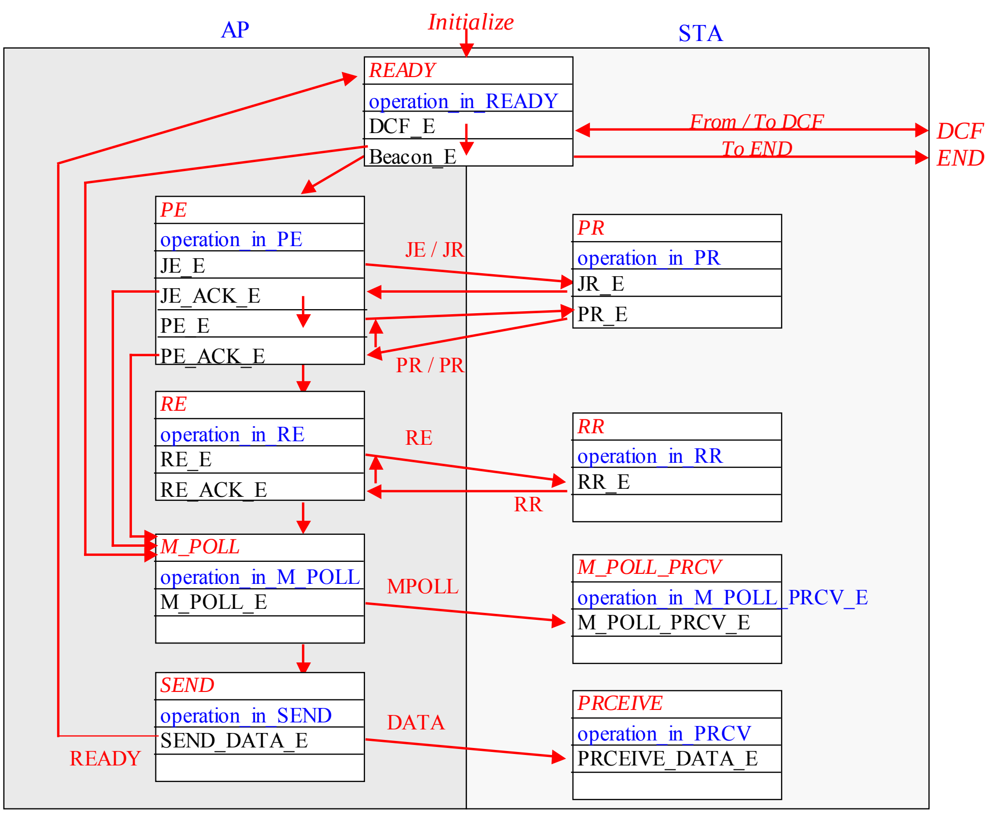


Figure 7.1 State machines for EQ-PCF MAC Blocks.

The major differences of MAC blocks between Q-PCF and EQ-PCF are the block of PE and RE and the others are the same as Q-PCF described above. The following statement would describe the blocks of PE and RE respectively.

PE STATE：

JE EVENT：

PC would inquire STA whether it wants to join into polling list in this block and as shown in the following program codes, the time to trigger the event is current time + **JE** frame transmission time + a SIFS time.

```
event_insert(id,JR_E,gclock+JE+SIFS,PR_S);
```

JE_ACK EVENT：

PC would receive the response from STA in this block and as shown in the following program code, the time to trigger this event depends on the situation of the response from STA.

```
if ((d_g<=0)||(RAB<=0))
        event_insert(id,M_POLL_E,gclock+M_POLL+SIFS,M_POLL_S);
if(collision==0){
        event_insert(id,M_POLL_E,gclock,M_POLL_S);
}else if (collision==1){
        event_insert(id,M_POLL_E,gclock,M_POLL_S);
}else if (collision==2){
        event_insert(id,PE_E,gclock,PE_S);
}
```

PR STATE：

JR EVENT：

STA would response PC whether it wants to join into polling list in this block and as shown in the following program codes, the time to trigger the event is current time + **JR** frame transmission time + a SIFS time.

```
collision=2;
        event_insert(id,JE_ACK_E,gclock+JR+SIFS,PE_S);
        d_r = d_r-(JE+JR+2*SIFS);

collision=0;
        event_insert(id,JE_ACK_E,gclock+Slot_Time,PE_S);
        d_r = d_r-(JE+Slot_Time);

collision=1;
        event_insert(id,JE_ACK_E,gclock+JR+SIFS,PE_S);
        if (pri==2){
                d_g = d_g-(SIFS+CBRGB+4);
                d_r = d_r-(JE+JR+CBRGB+3*SIFS+4);
        }else{
                d_g = d_g-(SIFS+VBRGB+4);
                d_r = d_r-(JE+JR+Dmin(node[AID].db,VBRGB)+3*SIFS+4);
        }
```

7.2. Simulation Results

This sector would show the simulation results from the aspects of register time and throughput in the network. The register time of EQ-PCF mechanism is much better than original PCF and a little worse than Q-PCF. And the worse register time of EQ-PCF is acceptable. The throughput of EQ-PCF is located between PCF and Q-PCF. Even though the performance of EQ-PCF is a little worse than Q-PCF, EQ-PCF is very compatible with IEEE 802.11 standard and Q-PCF is not.

7.2.1. Register time

Objective:

Calculating the average register time of PCF, Q-PCF and EQ-PCF respectively with the simulator.

Condition:

STA=256, ε_{CBR}=0, ε_{VBR}=0,CBR=100,VBR=0, λ=1 frame / sec (DCF)

Result:

The performances of Q-PCF and EQ-PCF mechanisms are almost the same when the members in the polling list are at full quantity or the remaining bandwidth is not available. The reason of this phenomenon is that PC is not going to poll any STA anymore since the remaining bandwidth of PC is not available. However, the PC is going to poll STAs even though none of

STAs wants to join into polling list when the remaining bandwidth of PC is available. In this situation, the PC would poll STAs upon the priority levels of STAs and the PC would poll n times when the system divides all STAs to n priority levels. In this situation, the performance of EQ-PCF is much better than that of Q-PCF mechanism. In this testing, the number of CBR STAs is given and the number of VBR STAs is zero. The PC would poll VBR STAs under Q-PCF mechanism and these operations would waste the bandwidth of network.

As shown in the Figure 7.2, the register time of EQ-PCF is 100 (us) longer than that of Q-PCF and this result is acceptable. This result would be different when the priority levels are become more than 3. And the register time of EQ-PCF is 8,500 (us) faster than that of PCF as the number of STAs is 128 and that of EQ-PCF is 12,200 (us) faster than that of PCF as the number of STAs becomes 256. According to the result, the performance of PCF is quite unideal especially when the number of STAs is more than 256.

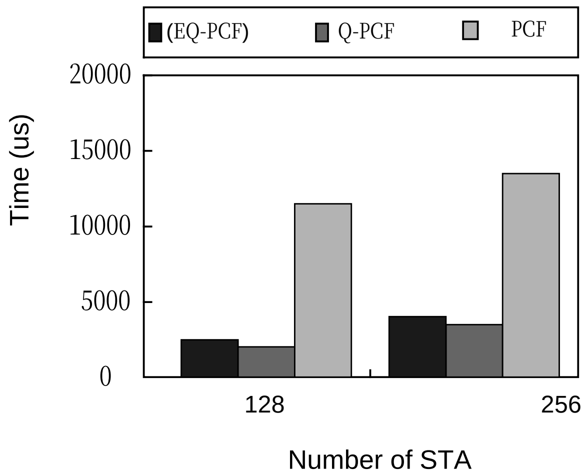


Figure 7.2 Register time

7.2.2. Throughput

Objective:

Calculating the average network throughput of PCF, Q-PCF and EQ-PCF respectively with the simulator.

Condition:

STA=256, ε_{CBR}=0:02, ε_{VBR}=0.5,CBR=100,VBR=30

Result:

Figure 7.3 shows that the throughput of Q-PCF doubles that of PCF. The major cause is that Q-PCF employs point-to-point transportation directly during CFP period in the infrastructure mode and PCF defined by IEEE 802.11 standard employs an intermediary, that is AP, to deliver the transaction to the target. The throughput of EQ-PCF is a little less than that of Q-PCF and is much better than that of PCF. This performance is acceptable and it is worth to be notice that EQ-PCF mechanism is fully compatible with PCF but Q-PCF mechanism is not.

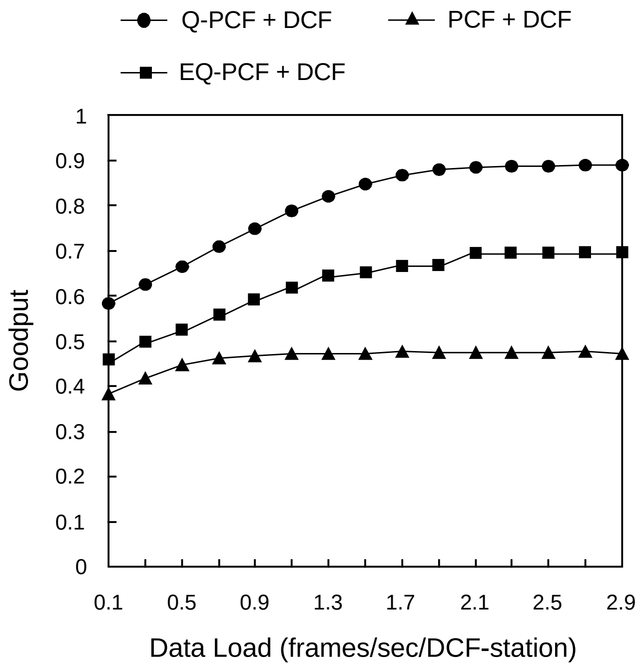


Figure 7.3 Performance Evaluation

CHAPTER 8

Conclusions and Future Works

Users are permitted to require PC to give the guarantees of transmission time and bandwidth according to the dynamic requirement of users on the method provided by this research. The result is it is practicable to transport real-time multimedia with wireless LAN. And this research also improve the throughput of wireless network since the design of point-to-point transportation between adjacent STAs. Besides, EQ-PCF mechanism improves the compatibility with IEEE 802.11 standard. Those are the best contributions to real-time multimedia transportation with wireless LAN in this thesis.

The result of this research has arrived at the above objectives and however, there are several issues to be discussed in the future. First at all, It may be considered to change STAs employing DCF mechanism to PCF mechanism if the loading of DCF is too heavy and not only STAs with real-time multimedia transportation have right to use PCF mechanism. STAs with non-real-time multimedia transportation also have the right to use PCF mechanism if the loading of DCF is too heavy so the probability of collision would be decreased. Second, the Q-PCF and EQ-PCF mechanisms permit STAs with higher priority level to join into polling list before those with lower priority level at the same time. However, the STAs with higher priority level are unable to join into the polling list if the whole CFP time is allocated by the STAs with lower priority level at the early period. It is a choice to force STAs with lower priority level to withdraw from polling list and to accept STAs with higher priority level to join into polling list. Third, even though the priority levels are applied to application programs in the computer, it is appropriate to apply the priority levels to human-life application such as the fee of networking coffee shop. Forth, the power-saving performance is unideal under the requirements of QoS. There is space to discuss or design a mechanism to take care both sides.

References

[1] I. F. Akyildiz, D. A. Levine, and I. Joe. A Slotted CDMA Protocol with BER Scheduling for Wireless Multimedia Networks. IEEE Transactions on Networking, Vol. 7, No. 2, pp. 146–158, April 1999.

[2] D. Bertsekas and R. Gallager. Data Networks, Second Edition, Prentice-Hall, 1992.

[3] Giuseppe Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, pp. 535–547, March 2000.

[4] Sunghyun Choi. Emerging IEEE 802.11e WLAN for Quality-of-Service (QoS) Provisioning. SK Telecom Telecommunications Review, Vol. 12, No. 6, pp. 894–906, December 2002.

[5] S. Choi and K. G. Shin. A Unified Wireless LAN Architecture for Real-Time and Non-Real-Time Communication Services. IEEE/ACM Transactions on Networking, Vol. 8, No. 1, pp. 44–59, February 2000.

[6] Zi-Tsan Chou, Ching-Chi Hsu, and Ferng-Ching Lin：A New Quality-of-Service Point Coordination Function for IEEE 802.11 Wireless Multimedia LANs

[7] C. Coutras, S. Gupta, and N. B. Shro®. Scheduling of Real-Time Trffic in IEEE 802.11 Wireless LANs. ACM/Baltzer Wireless Networks, Vol. 6, pp. 457–466, 2000.

[8] D.-J. Deng and R.-S. Chang. A Priority Scheme for IEEE 802.11 DCF Access Method. IEICE Transactions on Communications, pp. 96–102, January 1999.

[9] M. Fischer. QoS Baseline Proposal. IEEE Document, IEEE 802.11-00/360. [Online] http://grouper.ieee.org/groups/802/11/Documents/DT351-400.html

[10] A. Ganz, A. Phonphoem, and Z. Ganz. Robust SuperPoll with Chaining Protocol for IEEE 802.11 Wireless LANs in Support of Multimedia Applications. ACM/Kluwer Wireless Networks, Vol. 7, pp. 65–73, 2001.

[11] J. Y. Juang and B. W. Wah. Unified Window Protocols for Contention Resolution in Local Multiaccess Networks. IEEE INFOCOM, pp. 97–104, April 1984.

[12] S. Keshav. An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network. Addison-Wesley Publishing Company, 1997.

[13] LAN MAN Standards Committee of the IEEE Computer Society. IEEE Standard 802.11-1999 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE,

November 1999.

[14] LAN MAN Standards Committee of the IEEE Computer Society. IEEE 802.11e/D3.2, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS). IEEE, August 2002.

[15] K. Nakano and S. Olariu. Randomized Initialization Protocols for Ad Hoc Networks. IEEE Transactions on Parallel and Distributed Systems, Vol. 11, No. 7, pp. 749–759, July 2000.

[16] S. Ross. A First Course in Probability, Fifth Edition. Prentice-Hall, Inc., 1998.

[17] O. Sharon and E. Altman. An Efficient Polling MAC for Wireless LANs. IEEE/ACM Transactions on Networking, Vol. 9, No. 4, pp. 439–451, August 2001.

[18] J.-P. Sheu, C.-H. Liu, S.-L. Wu, and Y.-C. Tseng. A Priority MAC Protocol to Support Real-Time Multimedia Traffic in Ad Hoc Networks. ACM/Kluwer Wireless Networks, to appear, 2003.

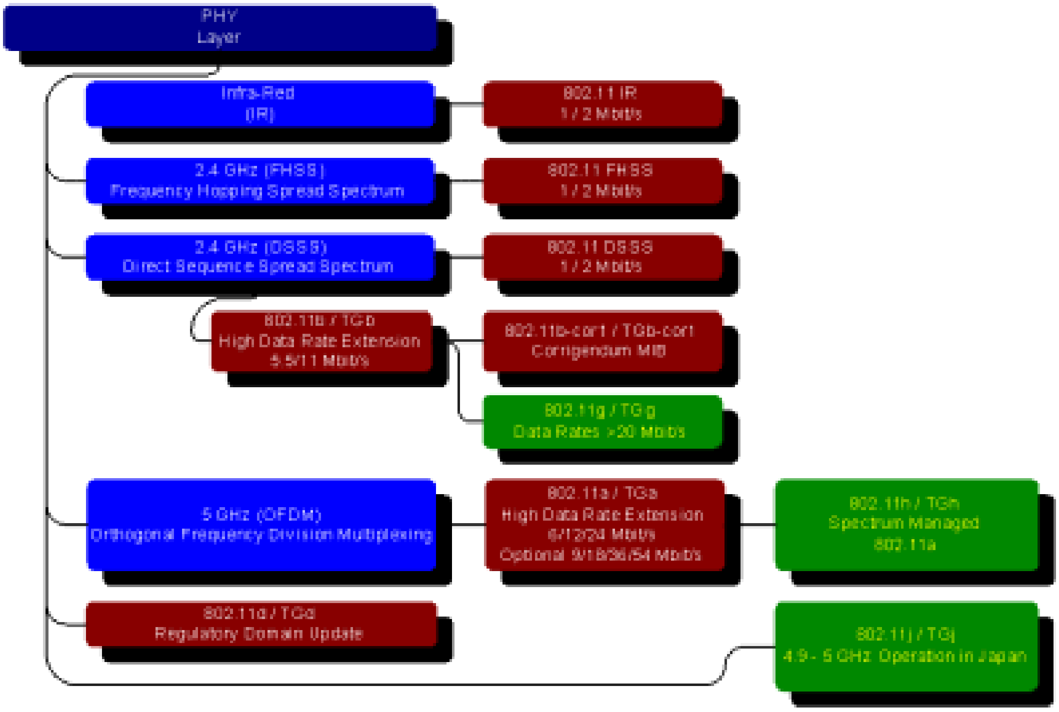
[19] S.-T. Sheu and T.-F. Sheu. A Bandwidth Allocation/Sharing/Extension Protocol for Multimedia Over IEEE 802.11 Ad Hoc Wireless LANs. IEEE Journal on Selected Areas in Communications, Vol. 19, No. 10, pp. 2065–2080, October 2001.

[20] Y. C. Tay and K. C. Chua. A Capacity Analysis for the IEEE 802.11 MAC Protocol. ACM/Kluwer Wireless Networks, Vol. 7, pp. 159–171, 2001.

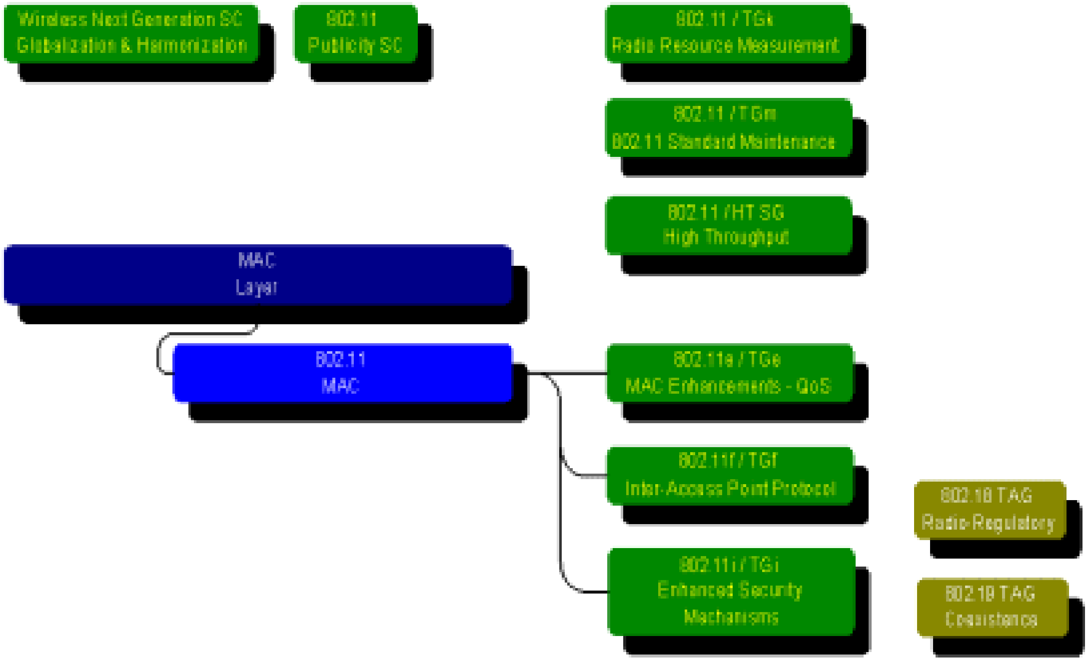
Appendix A

The following figures of IEEE 802.11 family and status of MAC and PHY are referred from http://www.ieee802.org/11/ in the internet.

802.11 Activities - PHY:



802.11 Activities - MAC & Others:



Status

802.11a：The scope of the project is to develop a PHY to operate in the newly allocated UNII band.

802.11b：Work has been completed and is now part of the Standard as an amendment - Published as IEEE Std. 802.11b-1999

802.11b-Cor1：Work has been completed and is now part of the Standard as an amendment - Published as IEEE Std. 802.11b-cor1 2001

802.11c：Work has been completed and is now part of the ISO/IEC 10038 (IEEE 802.1D) Standard

802.11d：Work has been completed and is now part of the Standard as an amendment - Published as IEEE Std. 802.11d 2001

802.11e：Ongoing - Note: the Security portion of the TGe PAR was moved to the TGi PAR as of May 2001. TGe has completed letter ballot 51 with a 83% approval rate and is now in comment resolution.

802.11f：Work has been completed and is now part of the Standard as a recommended practice.

802.11g：Work has been completed and is now part of the Standard as an amendment.

802.11h：TGh has completed the 2nd Sponsor Recirculation Ballot with a 98% approval rating and is now in the comment resolution phase.

802.11i：Ongoing - Note: the Security portion of the TGe PAR was moved to the TGi PAR as of May 2001. TGi has completed WG Recirculation Letter Ballot 57 with a 78% approval rating and is now in the comment resolution phase.

802.11j：Ongoing - Initial meeting January 2003 TGj has completed Letter Ballot 56 with an approval rating of 79% and is now in the comment resolution phase.

802.11k：Ongoing - Initial meeting January 2003 and is preparing its first draft.

802.11m：Initial meeting March 2003 (Subject to SEC approval of the Task Group PAR)