國立交通大學

網路工程研究所

碩 士 論 文

一個無線區域網路上支援即時影音傳輸的適應性輪詢機制

An Adaptive Polling Scheme Supporting Real-Time Audio/Video Transmission in Wireless LANs

研 究 生：葉函儒

指導教授：陳耀宗 教授

中 華 民 國 九 十 六 年 七 月

一個無線區域網路上支援即時影音傳輸的適應性輪詢機制

An Adaptive Polling Scheme Supporting Real-Time Audio/Video Transmission in Wireless LANs

研 究 生：葉函儒　　　　Student：Han-Ru Yeh

指導教授：陳耀宗　　　　Advisor：Yaw-Chung Chen

國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

一個無線區域網路上支援即時影音傳輸的適應性輪詢機制

研究生：葉函儒　　　　　　　　　　　　　　　　指導教授：陳耀宗

國立交通大學網路工程所

摘要

IEEE 802.11e 標準主要是為了提供無線區域網路上的服務品質(QoS)。然而它卻依然無法做到即時媒體傳輸的品質保證，同時也會產生不少頻寬的浪費。本篇論文提出一種新的輪詢機制以達到更好的服務品質，像是改進時間顫動(jitter)、減少存取延遲(access delay)，以及提高頻寬的使用率。我們提出一個適應性的輪詢機制，此機制基於不同的工作站需要不同的服務區間，計算出每個工作站更合適的輪詢時間，以避免被輪詢到的工作站沒有訊框可以傳送，而造成浪費。不同的影音編碼方式，也會在這個新的輪詢機制考量範圍內。此外，為了偵測出更合適的輪詢時間以減少存取延遲，我們會在短暫的時間內，使用較短的服務區間來輪詢。同時我們會動態地增減語音傳輸資料流的服務區間，來適應此語音類型資料流的特徵。我們利用 NS-2 進行模擬實驗，數據結果顯示，我們提出的方法比起慣用的輪詢機制，可以減少更多存取延遲，也能降低時間顫動的標準差，同時提高整個無線頻寬的使用率。

An Adaptive Polling Scheme Supporting Real-Time Audio/Video Transmission in Wireless LANs

Student: Han-Ru Yeh　　　　Advisor: Yaw-Chung Chen

Institute of Network Engineering
National Chiao Tung University

Abstract

The main goal of IEEE 802.11e standard is to support Quality of Service (QoS) in wireless local area networks (WLANs). However, it makes a lot of unnecessary overheads and is still unable to guarantee QoS for real-time audio/video services. The motivation of this thesis is to provide better QoS, such as reducing jitter and access delay, and to improve the channel utilization. We propose an adaptive polling scheme which minimizes QoS-Null replies by calculating much appropriate polling time for all QoS traffic streams, based on much accurate service intervals for different QoS traffic streams. Also different codecs are taken into consideration in our proposed scheme. Besides, in order to reduce access delay and find a much precise polling time, we reduce polling interval to poll some traffic streams. Further, we dynamically increase or decrease service interval for voice transmission based on its talk-spurt and silence alternation characteristic. The simulation results through NS-2 show that our proposed scheme can reduce access delay and standard deviation of jitter as well as enhance channel throughput comparing with the commonly used round-robin polling scheme.

Keywords: QoS, jitter, access delay, polling scheme

Acknowledgement

First of all, I would like to express my sincerity to the Prof. Yaw-Chung Chen who gave me a lot of precious suggestions, supported me with his experience, professional knowledge and valuable time. He guided me through the difficulties to a proper goal and inspired me so much to complete this thesis. I would also appreciate my family for their encouragement and trust all the time. Besides, I need to thank my laboratory mates Chun-Yung Lin, Cheng-Yuan Ho, Zhen-Hua Shi, and my roommate Cheng-Han Jiang. They gave me many useful comments. Finally, I would like to thank all members in my laboratory for the wonderful time.

Table of Contents

Table of Figures

Table of Tables

Table of Equations

Chapter 1 Introduction

Deployment of wireless networks has grown rapidly in the last decade. It is convenient to connect mobile devices to the Internet without wired line in the public areas, such as libraries, train stations, hotel lobbies, cafeterias, airports, and campus. Nowadays, most of mobile devices such as notebooks, tablet PCs, and PDAs are equipped with one or more wireless communication interfaces. So far, IEEE 802.11 is the most mature and popular protocol amongst other wireless protocols such as Bluetooth, WiMAX, and UWB.

The goal of IEEE 802.11 working group established in 1990 is to develop a wireless local area network (WLAN) standard. In 1997, the group approved IEEE 802.11 as the first WLAN standard with data rates of 1 and 2 Mbps only, through three physical medium, infrared (IR), frequency hopping spread spectrum radio (FHSS) and direct sequence spread spectrum radio (DSSS). The IEEE 802.11 MAC sub-layer defines two medium access coordination functions, the base Distributed Coordination Function (DCF) and the optional Point Coordination Function (PCF). Two year later, in 1999, the working group approved two extended WLAN protocol standards. One is 802.11a, it is based on orthogonal frequency division multiplexing (OFDM), operates on U-NII band (Unlicensed National Information Infrastructure) in 5.4GHz, and with a maximum data rate of 54 Mbps. The other is 802.11b, it is based on DSSS, operates in 2.4GHz ISM band, with a maximum data rate of 11 Mbps. The family of IEEE 802.11 is shown in Table 1.1.

Table 1.1 The IEEE 802.11 family.

Standards	**Description**
IEEE 802.11a	Standard for 54 Mbit/sec at 5GHz
IEEE 802.11b	Standard to support 11 Mbit/sec at 2.4GHz
IEEE 802.11c	Bridge operation procedures
IEEE 802.11d	International (country-to-country) roaming extensions
IEEE 802.11e	Standard to support QoS
IEEE 802.11f	Inter-Access Point Protocol
IEEE 802.11g	Standard for 54 Mbit/sec at 2.4 GHz
IEEE 802.11h	Spectrum Management for European compatibility
IEEE 802.11i	Enhancement for security
IEEE 802.11j	Extensions only for Japan
IEEE 802.11k	Standard for radio resource management
IEEE 802.11m	Initiative to perform editorial maintenance, corrections, improvements, clarifications, and interpretations for the IEEE 802.11 family specifications.
IEEE 802.11n	Standard for high throughput improvements with MIMO (multiple input, multiple output antennas)
IEEE 802.11p	Standard for WAVE (Wireless Access for the Vehicular Environment)
IEEE 802.11r	Standard for fast roaming to specify fast BSS ("Basic Service Set") transitions
IEEE 802.11s	Enhancement for mesh networking
IEEE 802.11t	Wireless Performance Prediction (WPP) – test methods and metrics recommendation
IEEE 802.11u	Enhancement for interworking with external non-802 networks
IEEE 802.11v	Wireless network management

Recently, voice over Internet Protocol (VoIP) became one of the most popular applications. Some products of VoIP, such as Skype, are able to support good quality of voice communication over wired networks. VoIP market grows up quickly due to its low cost and easiness to construct IP network than traditional telecommunication networks. Everyone can use its computer to make cheaper or even free voice call instead of using the expensive cell phone services. According to statistics, it says that the number of residential VoIP users will rise from three million at 2005 to 27 million by the end of 2009 [13].

Therefore, more and more people try to use handheld devices to make VoIP calls through wireless LANs. For this reason, the quality of service (QoS) in WLAN becomes increasingly important. However, voice communication over WLAN features many challenges, such as low bandwidth, large interference, long latency, high loss rates, and jitter. The distributed coordination function (DCF) and the point coordination function (PCF) are unable to guarantee QoS effectively [3], this is because DCF cannot provide QoS trivially, PCF is not efficient for only one frame sent at each polling, point coordinator does not know the QoS requirement of traffic and does not guarantee the delay and jitter bound, so it is harder to provide QoS for real-time audio/video transmission in wireless networks than in wired networks.

IEEE 802.11e standard [1] defined at IEEE 802.11 Task Group E is expected to solve the QoS problem of latency sensitive applications over wireless local area networks. 802.11e standard proposes some MAC mechanisms to support time-sensitive applications. Hybrid Coordination Function (HCF) includes two methods, one is Enhanced Distributed Channel Access (EDCA) that combines DCF, and the other is HCF-Controlled Channel Access (HCCA) that is similar to PCF but with enhancement. Direct Link Protocol (DLP) is the option that allows stations to exchange packets directly with other stations, bypassing the AP. Block Acknowledgement, an optional function in implementation, improves channel efficiency by aggregating several acks into one frame.

However, IEEE 802.11e is still unable to support satisfactory QoS for time sensitive applications. EDCA provides QoS based on probability, not determinism. This means, in worst case, the quality of delay time or jitter bound is not desirable. Besides, HCCA polling scheme is not specified in the standard, and the study in [2] shows the HCCA polling overhead problem and its negative effect on the QoS of real-time applications in WLAN. Consequently, a good polling scheme can improve the channel performance and thus QoS for time sensitive applications. Nowadays, the most popular polling scheduler is the round robin (RR) polling scheduler [5] because it is simple in implementation. Many other polling schemes such as [4], [6], [7], [8] have been proposed.

In this study, we focus on HCCA polling scheme mainly. We amend a new time-based polling scheduler called Adaptive Time-Stamp Polling (ATSP) scheme. ATSP scheme shows that it can improve total channel utilization and decrease delay jitter variation. In our scheme, Hybrid Coordination (HC) operates at the access point (AP), receives traffic specification (TSPEC) from stations which require polling to send out QoS frames, and records the service start time and maximum service interval corresponding to each station. Then, to avoid polling all stations in turn within one contention-free period (CFP), and using the same service interval for all stations to be polled with different duration of interval, we start to poll a station at its start time of registered service, and the interval for polling this station is same as the interval registered in its TSPEC. By this way, we can reduce the number of polling responded with QoS-Null frame because we don't poll a station with excessive frequency than its frame sampling frequency, and do not poll a station before it starts communication.

In addition, we use a simple mechanism to detect silence mode of a VoIP conversation. Then, we reduced the frequency of polling a silence station in order to decrease unnecessary waste of time. When the station comes back to talk-spurt mode, we revert to the original frequency to poll.

The rest of this thesis is organized as follows. Chapter 2 introduces the background of 802.11, 802.11e, voice/video transmission characteristic, and a brief survey of current polling mechanisms. In Chapter 3, we discuss the proposed polling mechanism in detail. In Chapter 4, simulation and numerical results are demonstrated. Finally, the conclusion and future works are presented in Chapter 5.

Chapter 2 Background and Related Work

In this chapter, we will review IEEE 802.11 MAC, including DCF and PCF. Next, we review IEEE 802.11e, including how EDCA and HCCA work. Then we will give a brief of the advantages and disadvantages of WLAN MAC mechanisms. Furthermore, we will go through the attributes of multimedia transmissions and how their codec work. Finally, we will list some related works that have been proposed.

2.1 Review of IEEE 802.11 MAC

Access to the wireless medium is controlled by coordination functions. 802.11 defines two different basic exchange procedures. One is the distributed coordination function (DCF) which is used in contention-based services; the other is the point coordination function (PCF), if contention-free service is required. Contention-free services are provided only in infrastructure networks. The coordination functions are described as follows.

2.1.1 Distributed Coordination Function

DCF is a fundamental function in IEEE 802.11 protocol. It is the basis of the standard Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access mechanism with binary slotted exponential backoff. Like Ethernet, when a station needs to send packets, it first checks to see whether the channel is clear before transmitting. If the channel has been idle for longer than DCF Interframe Space (DIFS), the station will try to transmit packets. If the channel is busy, the station should wait for the channel to become idle for DIFS and prepare for the exponential backoff procedure in order to avoid collisions. 802.11 refers to the wait as access deferral (see Fig. 2.1).

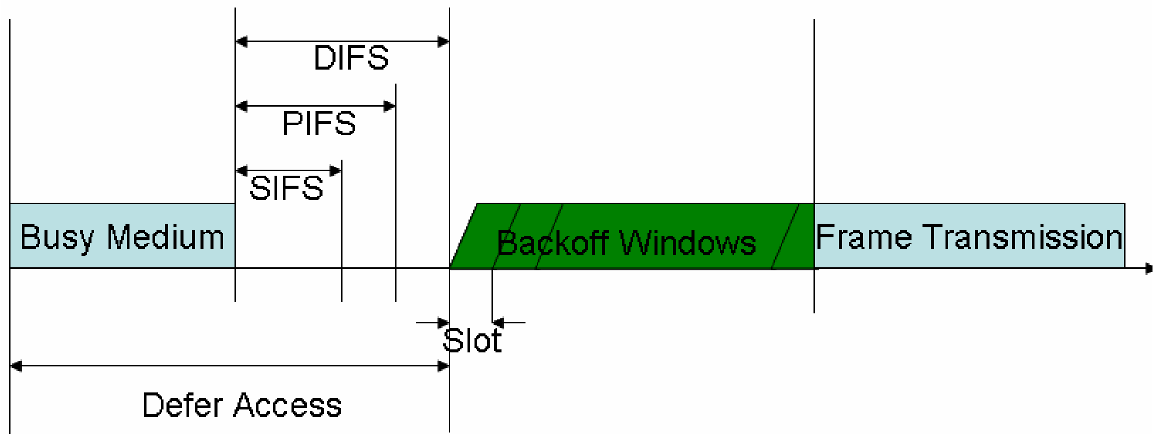

Figure 2.1 DCF transmission scheme.

The backoff window is divided into slots. Slot length is medium-dependent, generally, higher speed physical layers use shorter slot times. During the backoff procedure, station picks a random number of slots from contention window size. In other words, after a packet transmission was completed and DIFS has elapsed, several stations attempt to transmit again, the station that picks the lowest random number of slots wins.

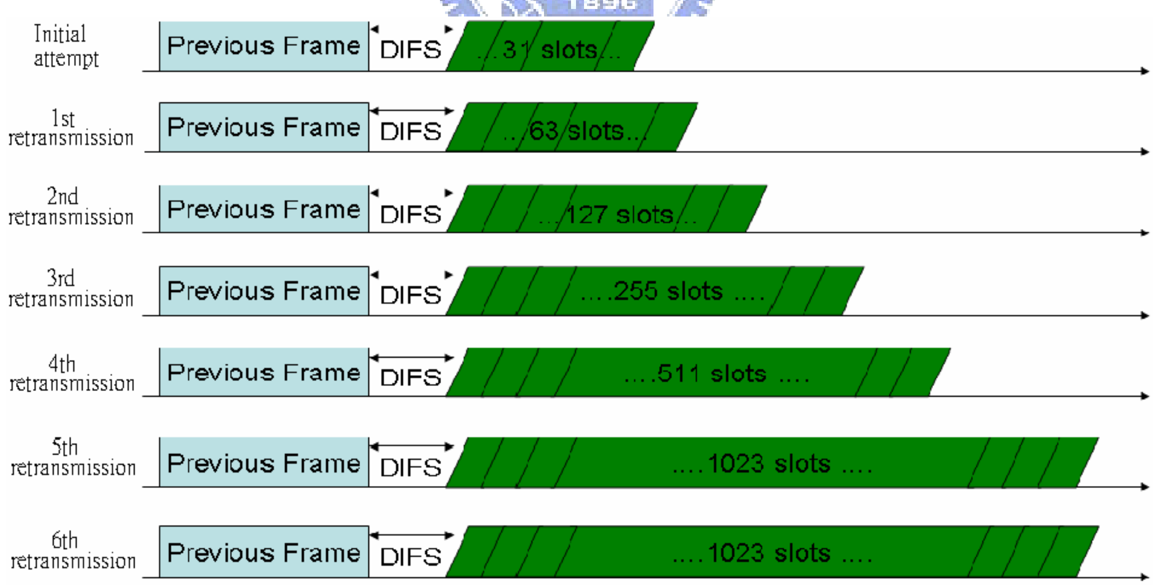

Figure 2.2 DSSS contention window size.

Contention window sizes are always a power of 2 minus 1 (e.g., 31, 63, 127, 255). The size of the contention window depends on the physical layer. For example, the DS physical layer limits the contention window to 1023 transmission slots, and after each unsuccessful transmission, the contention window moves to the next greatest power of two (see Fig. 2.2) until it reaches the limit.

Furthermore, with hidden nodes problem, RTS/CTS exchange procedure is an optional scheme. The sender transmits a request-to-send (RTS) packet before data frame transmission, and the receiver answers a clear-to-send (CTS) packet. RTS and CTS packets include information called network allocation vectors (NAV), station will set NAV to the time that it expects to use the medium. Then, other stations count down from NAV to 0. When the NAV reaches 0, it means that the channel is idle (see Fig. 2.3). Since this scheme may generate huge overhead when data packets are small, RTS/CTS is introduced for only those packets whose sizes larger than a certain threshold.

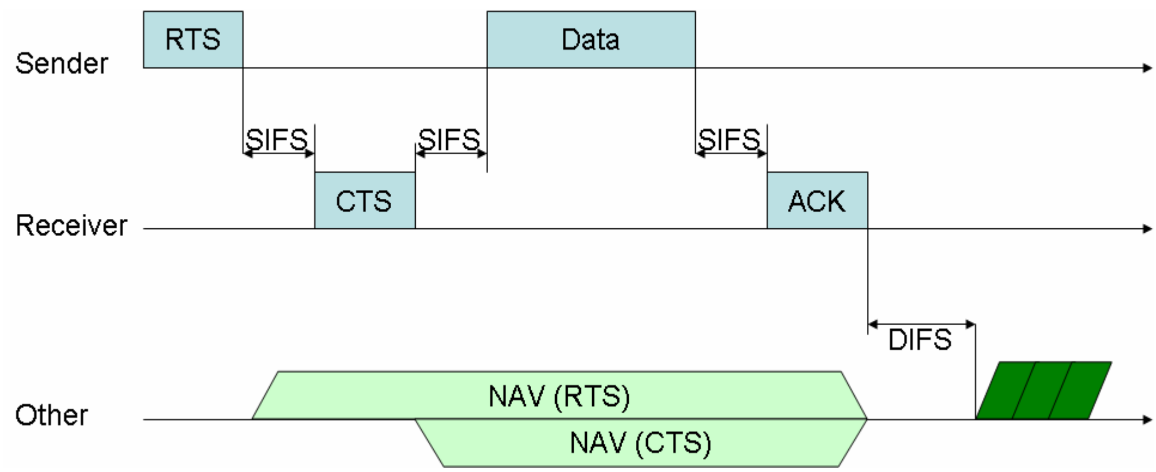

Figure 2.3 DCF timing chart

2.1.2 Point Coordination Function

The PCF is an optional access mechanism of the 802.11 specification, it is a centralized scheme which uses the AP as a point coordinator (PC). When the PCF is enabled, time on the channel is divided into the contention-free period (CFP) and the contention period (CP). The periods of contention-free service and contention-based service repeat at regular time intervals, which are called the contention-free repetition interval. In order to prevent interference, all transmissions in contention-free period are separated by the short interframe space (SIFS) and the PCF interframe space (PIFS). Since SIFS and PIFS are shorter than DIFS, DCF-based stations can not gain access to the medium using the DCF.

When a station is polled, it starts to send a data frame after SIFS, according to the specification, each poll is a license to send only one data frame. Multiple frames can be transmitted only if the access point sends multiple poll requests. To make sure that the point coordinator owns control of the channel, it may poll the next station on its polling list if there is no response after an elapsed PCF interframe space (see Fig. 2.4).

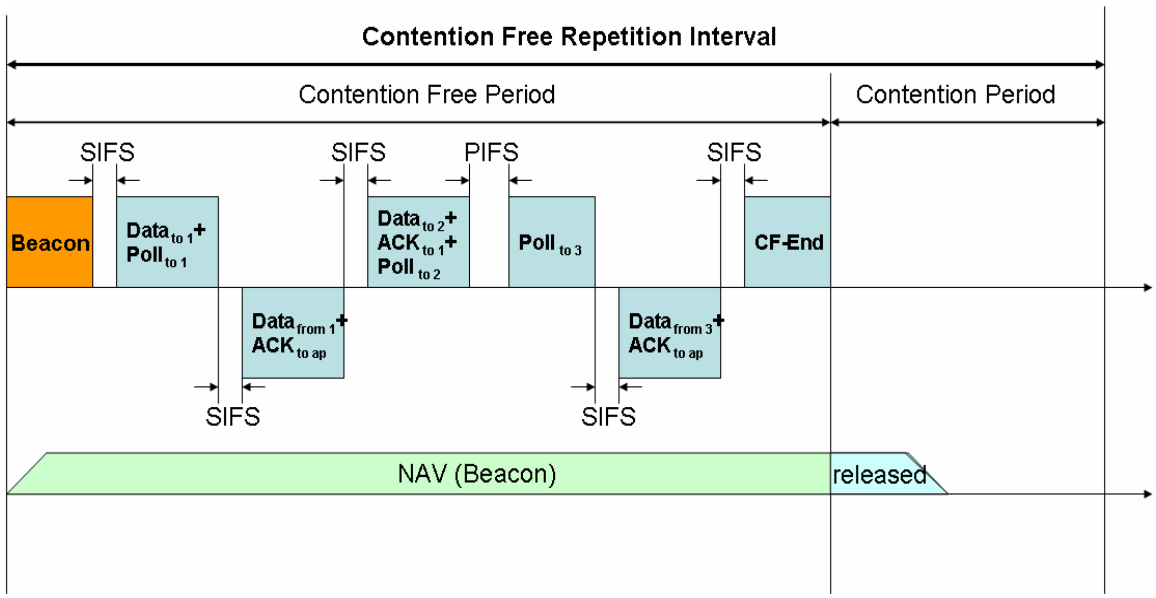

Figure 2.4 PCF timing chart

To improve efficiency, ACK, polling, and data transfer in the contention-free period may be combined, for example, Data+CF-Poll, Data+CF-ACK, Data+CF-Ack+CF-Poll, etc. And, the contention period must be large enough for the transmission of at least one maximum-size frame plus an ACK. When the contention-free period begins, the access point transmits a beacon frame. The beacon will announce the maximum duration of the contention-free period. All stations receiving the beacon would set the NAV to the maximum duration to avoid DCF-based access to the wireless.

If the existing frame exchange will occupy the contention-free period, it is allowed to complete the transmission. Then, the AP transmits a beacon frame to announce the start of the contention-free period. The contention-free period is shortened by the amount of the delay. The point coordinator may also terminate the contention-free period before its maximum duration by broadcasting a CF-End frame.

2.2 Introduction to IEEE 802.11e

The IEEE 802.11e defines the medium access control (MAC) procedures to support applications with quality of service (QoS) requirements, including the transport of voice, audio, and video over IEEE 802.11 wireless LANs (WLANs). The architecture of the MAC sublayer, including the distributed coordination function (DCF), the point coordination function (PCF), and the hybrid coordination function (HCF), can be described as shown in Figure 2.5 as providing the PCF and HCF through the services of the DCF.

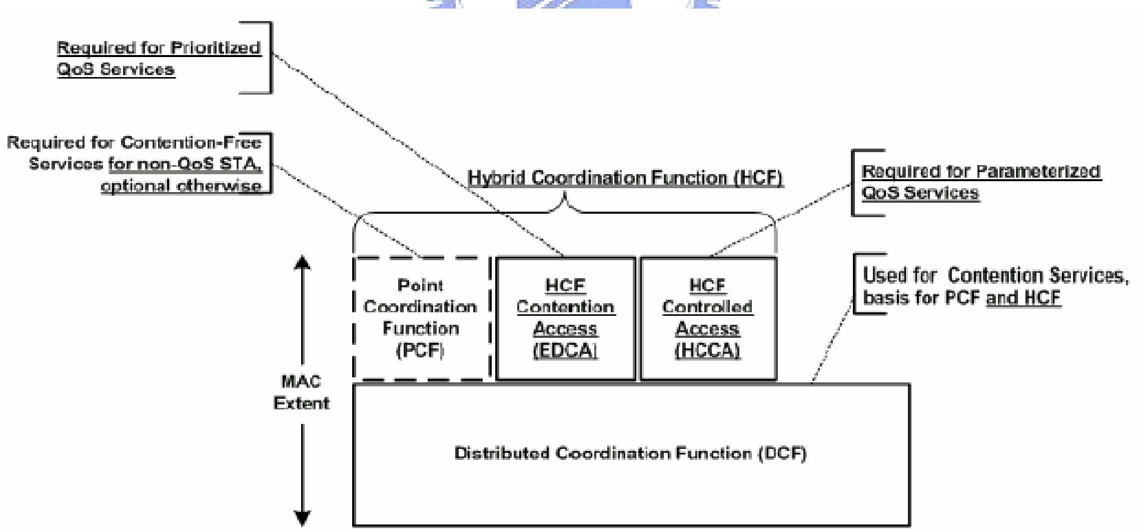

Figure 2.5 MAC architecture

The HCF uses both a contention-based channel access method, called the enhanced distributed channel access (EDCA) mechanism for contention-based transmission and a controlled channel access, called the HCF controlled channel access (HCCA) mechanism for contention-free transmission.

Stations that implement the 802.11e facility may obtain transmission opportunity (TXOP), which is an interval of time that station has the right to initiate frame exchange sequences on the wireless channel. A TXOP is defined by a starting time and a maximum duration, and the station that has obtained TXOP by successfully contending for the channel or assigned by the hybrid coordination (HC) can transfer one or more frames on the channel until the duration is over.

Each QoS data frame in 802.11e consists of a QoS control information, Figure 2.6 shows the 802.11e MAC frame format. The QoS Control field identifies that the traffic category (TC) or traffic stream (TS) which the frame belongs to and other QoS information, for example, the information about the corresponding MSDU, limit of TXOP, TXOP duration requested, ACK policy, etc.

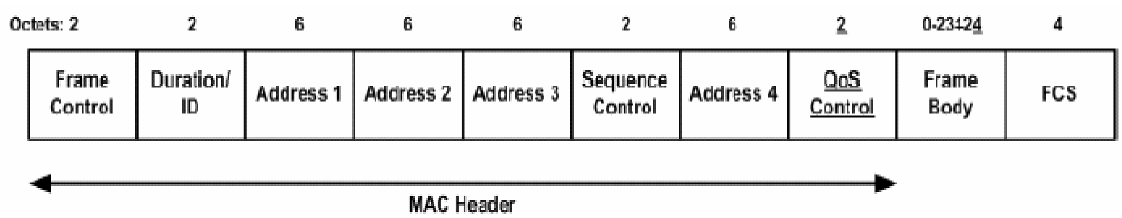

Figure 2.6 MAC frame format

In the following list, we will describe the major coordination function of 802.11e MAC architecture, the Enhanced Distributed Channel Access (EDCA) and the HCF-Controlled Channel Access (HCCA), and how they work to implement QoS requirement in detail.

2.2.1 Enhanced Distributed Channel Access (EDCA)

The EDCA mechanism defines four access categories (ACs) that provide support for the delivery of traffic with user priorities (UPs) at QoS stations (QSTAs). Voice, video, best-effort and background traffics are mapped into these four ACs. The ACs are used to contend for the channel in order to transmit data frames with certain priorities and are derived from UPs as shown in Table 2.1. Unlike DCF mechanism whose contention is between stations, in EDCA mechanism, each AC behaves as a single DCF contending entity and each AC has a corresponding parameter set. In fact, data frames are classified into four different FIFO queues in accordance with ACs (see Figure 2.7).

Table 2.1 UP-to-AC mapping.

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
Highest	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

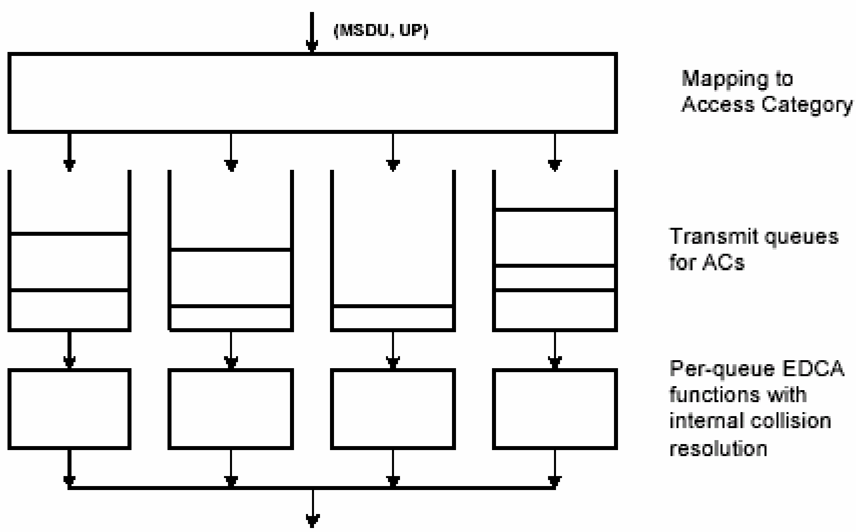

Fig. 2.7 Reference implementation model

For each AC, EDCA assigns different arbitration interframe space (AIFS), minimum contention window size (CWmin), maximum contention windows size (CWmax) and TXOP duration. Those parameter sets are announced by the AP periodically via beacon frames. It assigns smaller CWmin, CWmax and AIFS to ACs with higher priorities in order to bias the success probability that the QoS data frames can be sent with smaller delay. EDCA uses a new type of interframe space called AIFS, it is an time interval with arbitrary length and determined by the following equation, AIFS[AC] = AIFSN[AC] * aSlotTime + aSIFSTime. The parameter set is shown in Table 2.2.

In 802.11e standard, the value of AIFSN[AC] of QSTAs must be greater than or equal to 2, and the value of AIFSN[AC] of QoS access points (QAPs) must be greater than or equal to 1. Generally, AIFSN of AC_VI and AC_VO (see Table 2.2) for QAPs are set to 1, others for QAPs are same as that in Table 2.2, and DIFS = 2*aSlotTime + aSIFSTime. It means that AIFS of AC_VI and AC_VO is equal to DIFS for QSTAs, and AIFS of those is less than DIFS for QAPs. For this reason, the QAPs can transfer video and voice data frames with highest probability and the QSTAs transfer video and voice with second highest probability.

802.11e defines a TXOP limit as the interval of time during which a station has the right to initiate transmissions. The TXOP limit duration is introduced by the QAP as information element in the EDCA parameter set of beacon frames and probe-response frames. During a TXOP, stations can transmit one and more data frames from the same AC with a SIFS time between an ACK and the following data frame. A TXOP limit value of 0 means that a single MSDU only, in addition to a possible RTS/CTS frames exchange or CTS frame to itself, can be transmitted during each TXOP.

Figure 2.8 illustrates these EDCA parameters and the access procedure. Before each AC tries to send frames onto wireless channel, it needs to wait for an idle interval time of AIFS[AC], and a random backoff time from its corresponding CW. The purpose of different contention parameters for different queues of ACs is to give high-priority traffic more chances to gain the right to use the channel.

The end of backoff times of different ACs in one station may be the same sometimes, this situation will cause internal collisions and reduce the efficiency of service. In order to avoid these internal collisions, EDCA establishes a virtual layer inside MAC layer (see Fig 2.7) to allow the higher priority AC to send frames earlier. Therefore, the EDCA can support prioritized QoS for time-sensitive audio/video applications.

Table 2.2 Default EDCA parameter set element parameter values.

AC	CWmin	CWmax	AIFSN	TXOP limit		
				For PHYs defined in Clause 15 and Clause 18	For PHYs defined in Clause 17 and Clause 19	Other PHYs
AC_BK	aCWmin	aCWmax	7	0	0	0
AC_BE	aCWmin	aCWmax	3	0	0	0
AC_VI	(aCWmin+1)/2 – 1	aCWmin	2	6.016 ms	3.008 ms	0
AC_VO	(aCWmin+1)/4 – 1	(aCWmin+1)/2 – 1	2	3.264 ms	1.504 ms	0

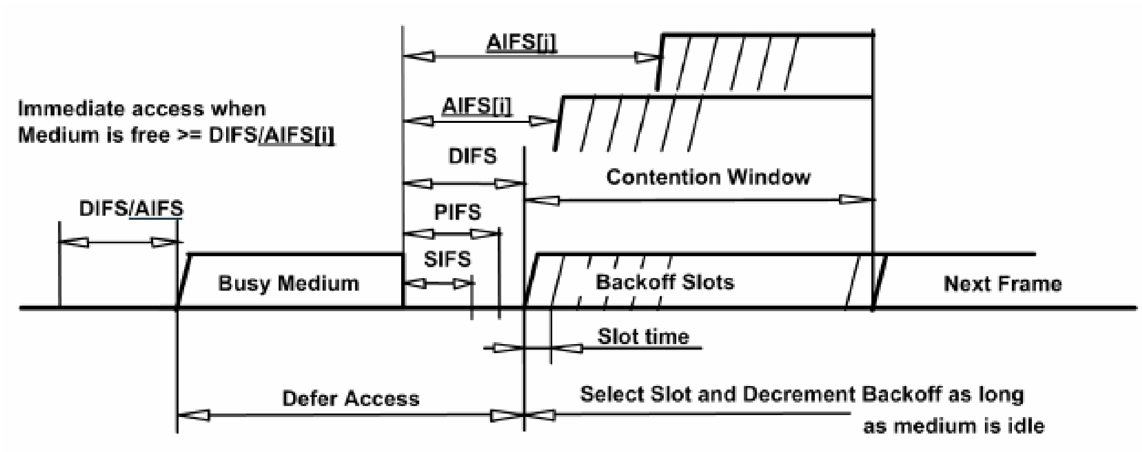

Fig. 2.8 IFS relationship and related terms

2.2.2 HCF Controlled Channel Access (HCCA)

The HCCA mechanism manages access to the channel, using an AP as hybrid coordinator (HC) that has higher access priority than stations. An AP can poll stations during contention free period (CFP) and contention period (CP). But in traditional PCF, polling is only allowed during CFP. This makes HCCA more flexible. During a CP, an AP is able to obtain the control of the channel for a certain time period, called controlled access phase (CAP), in order to poll multiple stations after sensing the channel to be idle for a PCF interframe space (PIFS). The AP has higher priority to start CAP and polling during CP because PIFS is shorter than DIFS and any AIFSs.

In HCCA, each QSTA which wants to be polled for QoS support must register a traffic stream (TS) to the HC by delivering the QoS parameter values in a particular traffic specification (TSPEC). The TSPEC includes information about the TS, such as mean data rate, delay bound, and maximum service interval (see Fig. 2.9). Then, QAP needs to provide services that conform to the demands registered in the TSPEC under controlled operating condition if this QAP admitted it and established the TS.

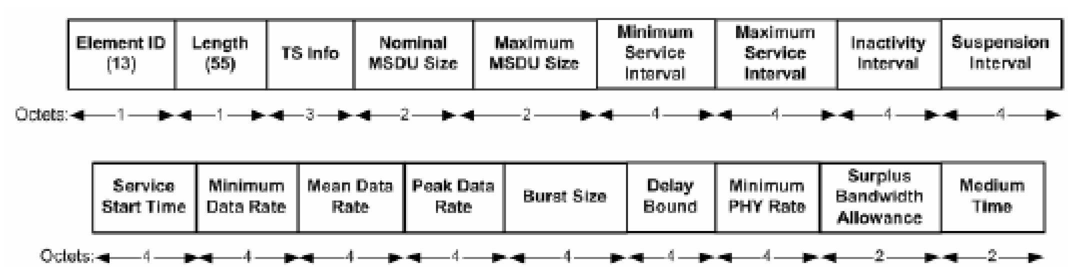

Fig. 2.9 TSPEC element format

After receiving a QoS request, the HC should determine the length of service interval (SI), and stations will be polled sequentially during each SI. In general, HC would calculate the smallest one among all maximum service intervals for those accepted TSPECs. Let the chosen one be SI_{min}. Then, HC chooses a value smaller than SI_{min} and this value should be a sub-multiple of the beacon interval. This value is the final SI for all QSTAs and admitted traffic streams, in this way, the beacon interval would be cut into several service intervals. Figure 2.10 shows the relationship of the CAP, CP, CFP and SI.

After HC calculated the SI for HCCA mechanism, it will also calculate the TXOP for those admitted TSs. First, HC calculates the number of frames that will arrive at the mean data rate during an SI by $N_i = \left\lceil \frac{SI * \rho_i}{L_i} \right\rceil$, where N_i is the number of frames sent from the stream i during an SI, ρi is the mean data rate for stream i and L_i is the nominal MSDU size of stream i. Then HC calculates the TXOP duration by $\mathrm{TXOP_i} = \max\left(\frac{N_i * L_i}{R_i} + O, \frac{M}{R_i} + O\right)$, where $\mathrm{TXOP_i}$ is the polled TXOP duration of stream i, R_i is the physical transmission rate, M is the maximum MSDU size and O refers to the transmission overheads.

By using those calculated TXOPs, we can make a simple admission control algorithm. The algorithm is to make sure that the sum of total TXOPs duration must be less than a CAP upper bound. When a new TS wants to be added into the HCCA polling list, it must meet the following equation,

$$\frac{TXOP_{new} + \sum_{i=1}^{n} TXOP_i}{SI} \leq \frac{CAP\ Limit}{Beacon\ Interval}, \quad \ldots\ldots(2.1)$$

where $\mathrm{TXOP_{new}}$ is the calculated TXOP duration of the new additional TS, $\sum_{i=1}^{n} TXOP_i$ is the sum of total TXOP duration of those current admitted TSs and CAP Limit means the maximum duration bound of HCCA. The algorithm ensures that the new traffic stream don't occupy all remaining time.

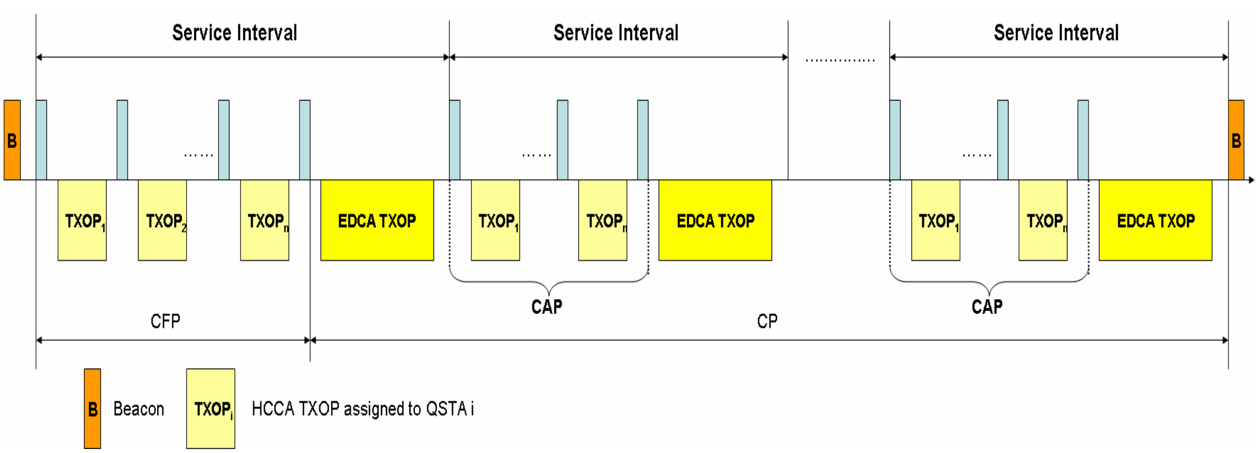

Fig. 2.10 CAP/CFP/CP/SI periods

However, this algorithm for calculating TXOP assumes all traffic streams are in constant bit rate. In practice, lots of real-time applications, such as VoIP or video conference, are using variable bit rate. So it may cause the queue overflow and packet drop that decreases the level of

QoS.

2.3 Summary of WLAN MAC Mechanisms

Two WLAN MAC mechanisms defined in IEEE 802.11 and 802.11e had been discussed in the previous chapter. One is contention-based, such as DCF and EDCA, and the other is contention-free-based (poll-based), such as PCF and HCCA. These two types of MAC mechanism have their own advantages and disadvantages, and we address them in this section.

The contention-based mechanism has several advantages. DCF and EDCA have a lower complexity of implementation than the contention-free-based mechanism which includes PCF and HCCA. Furthermore, contention-based mechanism is adaptive to dynamic change of traffic load and network condition. However, when more and more QSTAs enter the same channel, the waiting time of QoS data transmission may become longer and longer. This type of mechanism has hidden node problems that may cause extra collisions. The main drawback is that contention-based mechanism cannot guarantee QoS because it uses random backoff for QoS provisioning. It means probability takes the place of guarantee. Further, the backoff time is another overhead.

There are several advantages in contention-free-based mechanisms. The first advantage is the higher dependability on supporting QoS because it uses contention free transmission and centralized control mechanism on AP. It also can eliminate the hidden node problems. The channel utilization might be better than contention-based in the same situation because it reduces backoff time and collision overhead. Nevertheless, the complexity of implementing contention-free-based mechanisms is much higher than contention-based mechanisms since AP needs to schedule the sequence of QSTAs and might need to calculate transmission time. Furthermore, the excessive polling overhead is a critical problem if AP polls a station which has no QoS data frame in queue.

2.4 Attributes of Real-Time Multimedia Transmissions

We will introduce attributes of voice and video traffics briefly in this section, these include the packet duration, required bandwidth, the characteristics of those traffic streams. It shows that there are many voice and video requirements for the channel capacity over the wireless LANs.

First we discuss voice traffic, referring specifically to VoIP. Most applications of VoIP generate variable bit rate (VBR) traffic, this means that they generate packets only when the users of those applications are talking. Less applications of VoIP generate packets at constant bit rate (CBR), it means that they generate packets continuously with constant interval, 10ms, 20ms, 30ms, etc. Every VoIP codec might have different packet intervals and different transmission frequencies in Table 2.3 no matter it is CBR or VBR.

Table 2.3 VoIP codec standards

Codec	Bit Rate (kbps)	Payload (bytes)	Packet per Second (pps)	Packet Duration (ms)
ITU-T G.711	64	80	100	10
		160	50	20
		240	33.3	30
ITU-T G.723.1	6.4	24	33.3	30
		30	25.96	38.5
	5.3	20	33.3	30
		30	22.2	45
ITU-T G.726	32	120	33.3	30
ITU-T G.728	16	60	33.3	30
ITU-T G.729A	8	20	50	20
		30	33.3	30
GSM 06.10	13	33	20	50

It will cause extra polling overhead when the AP chooses the smallest one among all maximum service intervals as the final service interval for the AP to poll QSTAs (see Chapter 2.2.2). For example, there are two VoIP applications with different service intervals 20ms and 50ms, AP will poll both stations every 20ms. Hence, for the station with service interval 50ms, it should return QoS data frames 20 $(1000/50=20)$ times after AP polls it within 1000ms, but actually it will be polled 50 times. This causes 30 useless pollings which waste time. In 802.11b wireless LAN, we assume that the physical overhead is 192μs, SIFS is 10μs and CF-Poll frame transmission is at the basic rate (2 Mbit/sec) regardless of the data rate, and QoS-Null frame is transmitted at the maximum physical data rate (11 Mbit/sec), then we ignore the packet propagation time because this time is only few microseconds. So the time wasted would be $30 \times \{(192 + 36 \times 8 \div 2) + 10 + (192 + 36 \times 8 \div 11)\} \div 1000 \approx 17$ ms in every 1000ms.

According to Table 2.3, we also can see that VoIP requires relatively low bandwidth, the VoIP packet transmission frequency is very high and the payload size of voice traffic packets is smaller than other type of packets. Furthermore, it usually causes greater transmission overhead in VoIP than in general data transmission. For example, in G.711 over 802.11b wireless LANs, it needs 192μs for physical overhead duration, 52bytes for UDP, IP and MAC headers cost 38μs, and average backoff time is $CW_{\min} \div 2 \times Slot\ Time = 32 \div 2 \times 20 = 320\mu s$ whereas only 116μs is required to transfer a pure VoIP payload of 160bytes. In contention-free period based on polling mechanism, it still costs 336μs to transmit a CF-Poll frame instead of backoff time. Only about $116/(192 + 38 + 320 + 116) \times 100\% \cong 17.42\%$ of transmission time is used to transfer VoIP data even we use low rate VoIP codec at 802.11b wireless LANs.

The main attribute of voice transmission application is the alternation of talk-spurt and silence, that means users would stop talking and just listen to others sometimes, then they restart to talk later. This alternation is independent of the codec and application. If the voice application is VBR, during talk-spurt period the voice source works as isochronous source with stable interval time which is defined by the voice codec, and the voice source does not generate any voice frames during silence period. This attribution of voice transmission makes the original round robin (RR) polling schedule not efficient at all. This is because that the AP will not stop to poll VoIP stations which are in silence period. The AP continues to poll those VoIP stations even they respond no frame every time. It causes waste of bandwidth and increases unnecessary delay to other traffic streams.

Similarly, in video codec, there are several specifications, such as popular MEPG-1/2/4/7 standards defined by ISO and promising H.263 and H.264 standards defined by ITU-T. Video codecs have different sampling rates and intervals. For this reason, there might be lots of video transmissions with different service intervals in the same channel. Same as voice codec, it will still cause extra polling overhead when an AP chooses the smallest one amoung all maximum service intervals as the final service interval to poll QSTAs. Nevertheless, the payload size of video packet is usually up to 1000Bytes and most video streams do not have talk-spurt and silence alternation. So they cause less problem than voice streams.

2.5 Related Works

In our survey, most of those enhanced EDCA schemes tend to adjust the contention parameters of EDCA, for example, CWmin, TXOP duration, AIFS length. Some schemes try to change CWmin and AIFS length for the design of each AC such as that in [21], and some change those parameters dynamically according to the network condition such as traffic load [22]. Other adaptive schemes are focusing on adjusting the process of random backoff in contention depending on the network utilization or the priority of each AC such as that in [20]. In conclusion, the main improvement scheme of EDCA is to find better parameters for CWmin, AIFS length and etc, then let the frames of higher priority of AC be sent sooner and sooner.

For those enhanced polling schemes, there are fewer researches about HCCA than EDCA. Those adaptive schemes for HCCA usually try to find a mechanism to solve polling those stations without data frame ready in order to reduce the polling overhead. In [6], HC increases the polling interval of those stations that do not respond data frame to decrease the total polling overhead. In [9], AP will stop polling those stations which are considered silence stations, and increase the priorities of those stations during contention periods, and start to poll when they transmit QoS frames again. In [10], the VBR traffics and CBR traffics are transmitted in contention periods and contention-free periods respectively, this means AP only polls stations with CBR traffic. Another improvement schemes of HCCA focus on adjusting the TXOP duration to make polling process fairly. In [11], the determination of TXOP duration depends on TSPEC and current traffic status.

Admission control is important for AP to support QoS. Nowadays, there are two kinds of suggested admission control methods. One is that the AP measures the network condition continuously. To accept or deny a QoS request is based on the current throughput, average delay or other estimations. The other method is that the AP constructs a certain performance metric to forecast the status of the network and decide whether the QoS request is accepted or not. However, since there are a lot of interference factors which are difficult to be handled, the research of admission control in WLAN is still an active topic of study.

Chapter 3 Proposed Polling Scheme

In the previous chapter, the traditional round-robin polling scheme is not efficient for VBR traffic streams due to talk-spurt and silence alternation, and it causes large polling overhead because it uses the same service interval for all QoS traffic streams with requested different service intervals. Whether the AP polls the stations during silence periods or polls those stations with shorter service intervals than its actual frame intervals, both events will result in large polling overhead.

Unnecessary overheads will decrease the channel utilization, raise the packet access delay, and influence the deviation of jitter. If we want to provide higher quality of service for voice and video streams, we need a more efficient polling scheme that can avoid polling those stations which might not respond any data.

In this chapter, we propose an adaptive time-stamp poll scheming based on HCCA to provide better QoS in wireless LAN environment, and its associated auxiliary mechanisms that detect the silence periods and reduce the access delay. This scheme is able to reduce the number of unnecessary polling but not degrade the requested QoS, and the scheme operates on the top of MAC layer in the AP to manage the polling schedule and mitigate the access delay and jitter of those QoS traffic streams.

First, we will introduce our proposed scheme and explain how it works in detail. Then we discuss all kinds of situation that may happen in our proposed scheme and show how we handle them. We also rehearse this scheme by presenting some examples to compare it with round-robin polling scheme.

3.1 Main Architecture

The key idea of our proposed scheme is that HC calculates more suitable time to poll those traffic streams. The HC does not poll all traffic streams in the polling list during one CAP because each traffic stream has its own service start time, and requires a different service interval. So we propose an adaptive time-stamp poll scheduling which tries to calculate fairer and more efficient polling time for each traffic stream. After the HC calculates the suitable polling time, it polls those traffic streams exactly at their own calculated polling time.

However, only using this schedule is not sufficient for QoS of real-time applications. Therefore, we use a shorter interval to poll some traffic streams at the beginning of their transmission during a very small period, called the short interval polling function, in order to reduce the access delay. We also endeavor to solve the talk-spurt and silence problem of voice traffic streams such as VoIP. When a traffic stream has not generated data frames for a while, we use a simple count of receiving QoS-Null frames to determine whether it enters into silence mode or not.

Figure 3.1 shows an AP decision flow. When the current time equals or exceeds the calculated polling time, the HC sends a CF-Poll frame for this traffic stream. After it responds, the HC calculates the next polling time by adding different intervals depending on its current situation as shown in Figure 3.1.

Our proposed polling scheme does not modify the 802.11e standard; we do not need to add any new frame type. It enhances the original HCCA protocol with better poll scheduling policy. In the following, we will focus on those mechanisms mentioned previously in detail. We present adaptive time-stamp poll scheduling in Section 3.2 and show its advantages. In Section 3.3, we introduce the short interval polling function. And we show how we solve the talk-spurt and silence alternation problem in Section 3.4.

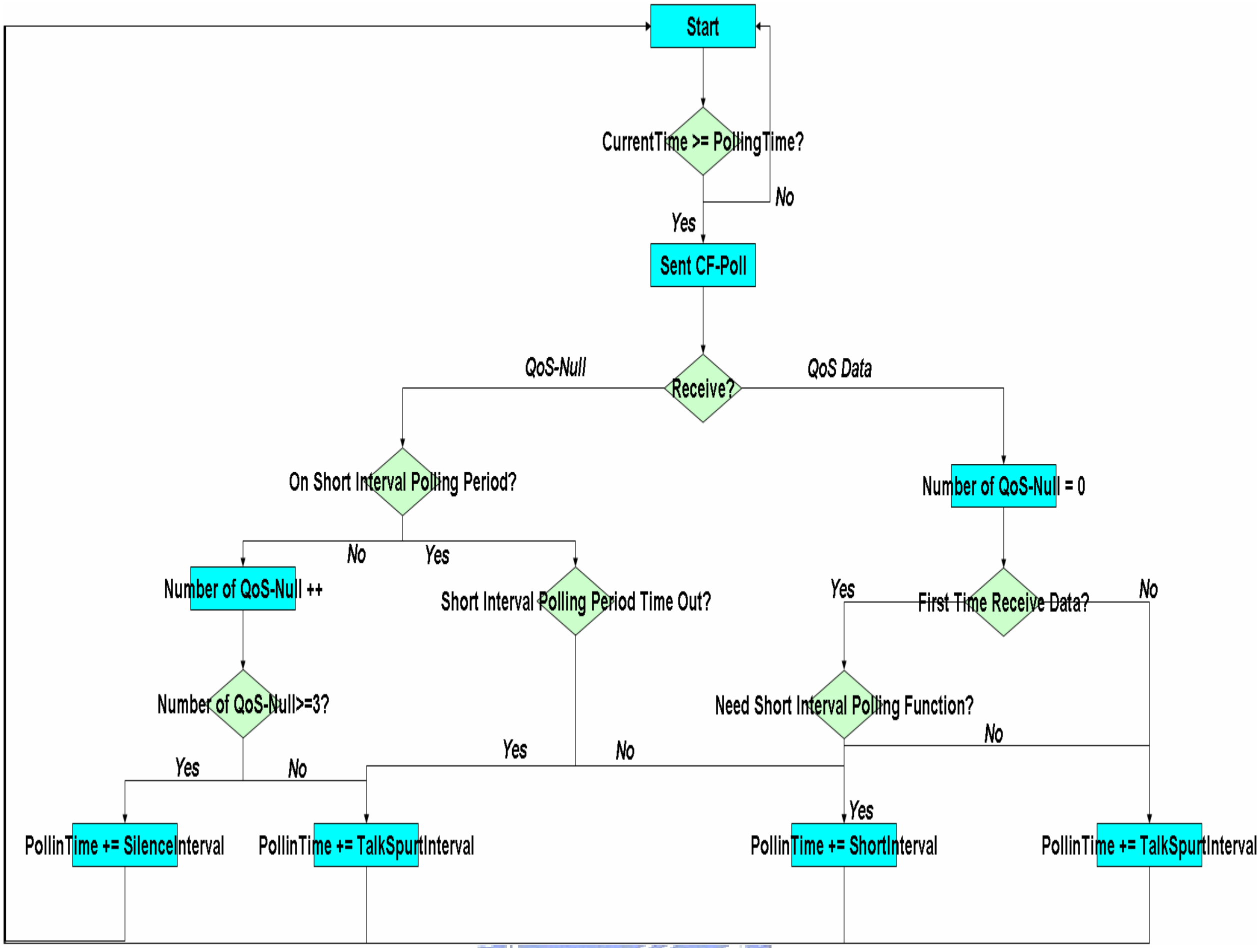

Fig. 3.1 AP decision flow chart

3.2 Poll Scheduling Mechanism

3.2.1 Time-Stamp Poll Scheduling

We propose an adaptive time-stamp poll scheduling method, the method starts to poll a new QoS station at its service start time which is registered in the accepted TSPEC (see Fig. 2.9). The service start time of a TSPEC means the time this QoS traffic stream needs the HC to poll it for transferring QoS data frame. Sometimes the original HCCA or PCF methods will start to poll the stations before their service start time because it tends to poll all accepted QoS traffic streams in a single contention-free period (CAP). It usually wastes time because these streams may not be ready to send frames. In order to avoid such time consumption, when a new traffic stream's QoS request is accepted, the HC starts to poll those stations exactly at its service start time.

When a QoS traffic stream has been polled, the HC calculates the next time to poll this stream by adding the maximum service interval in the TSPEC of this stream. Unlike original HCCA method, it uses adaptive polling intervals to each traffic stream instead of the unitary service interval. We do not use very small intervals to poll stations, instead we use maximum service intervals of them, so as to avoid unnecessary polling. Our basic polling time calculation formula is as follows:

$$1^{st}\, PollingTime_i = ServiceStartTime_i, \quad \ldots\ldots \quad (3.1)$$

$$2^{nd}\, PollingTime_i = 1^{st}\, PollingTime_i + MaximumServiceInterval_i, \ldots\ldots \quad (3.2)$$

$$n+1^{th}\, PollingTime_i = n^{th}\, PollingTime_i + MaximumServiceInterval_i, \quad \ldots\ldots \quad (3.3)$$

where $PollingTime_i$ is the predetermined time in which the traffic stream i will be polled, and $MaximumServiceInterval_i$ is the maximum service interval specified in the TSPEC of the traffic stream i.

In general, every HC has a clock to provide current time. When one of the polling time of a traffic stream is encountered, the HC tries to poll this traffic stream to give the temporal ownership of the channel to this stream for transmission. However, in some special case that we would discuss later, the HC can not poll these stations at their expected polling time exactly but delay for a while. In such situation, the HC should endeavor to poll this postponed traffic stream as soon as possible because it has passed the exact polling time.

The method of polling stations and exchanging frames during TXOP duration in our adaptive time-stamp scheme is same as the one in the original 802.11e HCCA scheme. When the HC detects that the present time equals to one of expected polling time of those traffic streams, the HC first checks to see whether the channel is clear before transmitting. If the channel is busy, the HC should wait for the channel to become idle for PCF Interframe Space (PIFS). If the channel has been idle for longer than PIFS, the HC sends CF-Poll for this assigned traffic stream to provide ownership and calculated TXOP duration. Both the HC and the station which had been polled would exchange frames with combining ACK, data or polling types at interval of SIFS.

If there is no data frame to transmit by this traffic stream before the end of TXOP duration, the HC sends the CF-End frame to all stations and comes back to the contention modes. And if there is one or more expected polling time occur among the current TXOP duration, the HC do not halt the current TXOP. The HC would poll these traffic streams just following the termination of the present TXOP duration by transferring Data+CF-ACK-CF-Poll frames or CF-Ack+CF-Poll frames like the method of HCCA.

In traditional round-robin poll scheduling of HCCA mechanism, it chooses the smallest service interval among all maximum service intervals of those accepted traffic streams as the interval in which the AP polls all stations as described in Section 2.2.2. However, it will cause extra polling when it uses a shorter service interval than that the station really needs.

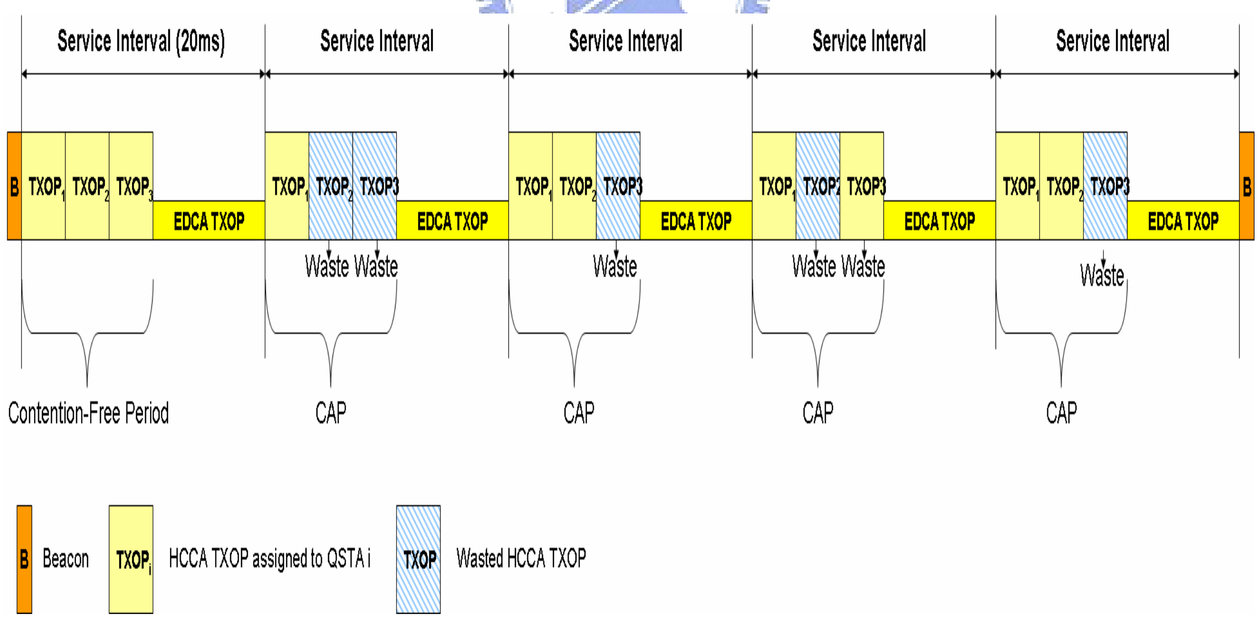

Fig. 3.2 Illustration of traditional RR polling scheme

For example, there are three stations which need to be polled for QoS transmission in HCCA method, station 1 requests polling with a maximum interval of 20ms that equals to the duration it generates the video or voice frames. Similarly, station 2 requests polling with a maximum interval of 30ms, station 3 requests polling with a maximum interval of 50ms and the beacon interval is 100ms. The hybrid coordinator (HC) will choose 20ms as the final service interval to start contention-free periods or controlled access phases (CAPs). During contention-free periods, all QoS-requested stations to be polled by the HC will be polled certainly in proper order. Figure 3.2 is an illustration for the previous example. After the beacon, it will start the contention-free period to poll three stations in turn, then after a service interval of 20ms, it starts the CAP to poll those stations one by one again, and it repeats every 20ms. This means that the HC polls all three stations every 20ms regardless of the true intervals in which those stations generate their video or voice frames actually. Therefore, station 2 generates QoS data frame in 30ms but is polled every 20ms and station 3 generates frames every 50ms; both of them will no doubt respond the QoS-Null frames after certain times of polling. It causes extra polling overheads and decreases the channel utilization.

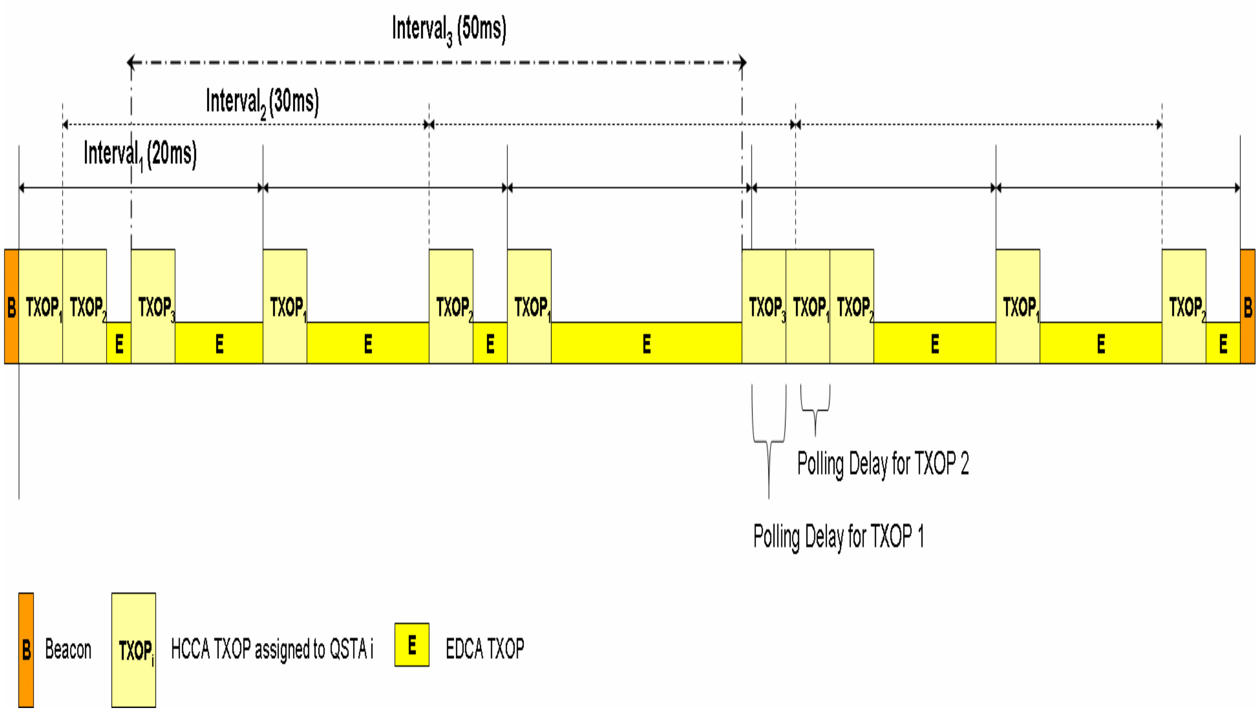

Fig. 3.3 Illustration of adaptive time-stamp polling scheme

However, the previous problem will be overcome in our proposed adaptive time-stamp scheme. For the same example, station 1 is polled at a maximum interval of 20ms, station 2 is polled at a maximum interval of 30ms, and station 3 is polled at a maximum interval of 50ms. They start to be polled at their own service start time with different service intervals. Figure 3.3 is an illustration for the same example using our proposed scheme. It looks like no contention-free periods, but the HC polls all stations during CAPs in contention periods. Each CAP period contains one or more polled TXOPs in accordance with the actual needs. In this illustration, there are some polled TXOP delays for station 1 and 2, because the polled TXOP duration of station 3 extends to the polling time of station 1, and in turn the TXOP of station 1 extends to the polling time of station 2. As Figure 3.2 and Figure 3.3 demonstrated, our proposed polling method can avoid unnecessary polling and improve the channel utilization.

3.2.2 Polling Policy and Priority

We address the scheduling policy and priority of polled TXOP in this section. As discussed in the previous chapter, the HC may not be able to poll certain station exactly at its intended polling time because the channel is not idle at that time. In this case, the HC should reschedule the polling of this traffic stream immediately. Here we will present all possibilities in detail.

To start, no matter what was the reason of delay, the HC still needs to wait for one PIFS time to see whether the channel is clear before transmitting, then the HC could send a CF-Poll frame to poll a traffic stream. In our adaptive scheme, the contention function is same as 802.11e EDCA standard which was introduced in Section 2.2.1. It means, the best-effort data transmission could be sent just one frame at every successful contention, and both the voice and video data frames exchange between the AP and the QSTAs uses SIFS as intervals during EDCA TXOP durations. Similarly, the data exchange during a polled TXOP duration also uses SIFS as intervals.

Consequently, the HC must not hold any current TXOP to poll any deferred traffic stream. If being delayed by voice or video frames, the deferred polling must wait until the total TXOP duration finishes because SIFS is smaller than PIFS. If it is delayed by best-effort frames, the deferred polling just waits one data frame, one SIFS and one relative ACK because PIFS is smaller than DIFS. The reason is not only the interval length but also that EDCA TXOP is as important as polled TXOP, and best-effort data transfer is less urgent.

Suppose that there is a lot of polled TXOP durations which had been delayed, how could we reschedule multiple deferred TXOP durations. The method we proposed in our adaptive scheme is to poll those traffic streams sequentially in accordance with the order of the intended polling time. With this method, it can avoid excessive and aggregate delay in some particular traffic stream. In the worst case, if all expected TXOP durations have been delayed during the same time, the HC needs rescheduling all TXOP duration sequentially. The outcome will be the same as the original round-robin polling scheme and it will not make other unnecessary delay.

3.2.3 Jitter Deviation Reduction

For voice and video traffic streams, the jitter deviation is very important for the QoS applications. If the deviation of jitter is large, it means the frame inter-arrival time of this traffic stream varies seriously and this event will make users uncomfortable because the voice or video traffic streams might intermit randomly. So we need to minimize the jitter deviation to supply better QoS.

Our adaptive time-stamp polling scheme can reduce the deviation of jitter easily. In general, the packet generating duration of the traffic stream will be the same as the maximum service interval registered in the TSPEC. For example, the traffic with the codec G.726 (see Table 2.3) will generate packets every 30ms and the maximum service interval in registered TSPEC is 30ms. The HC will poll this traffic stream every 30ms. While the polling interval of this traffic stream is the same as the interval with which the frame of this stream is generated, the jitter deviation will be very small and stable.

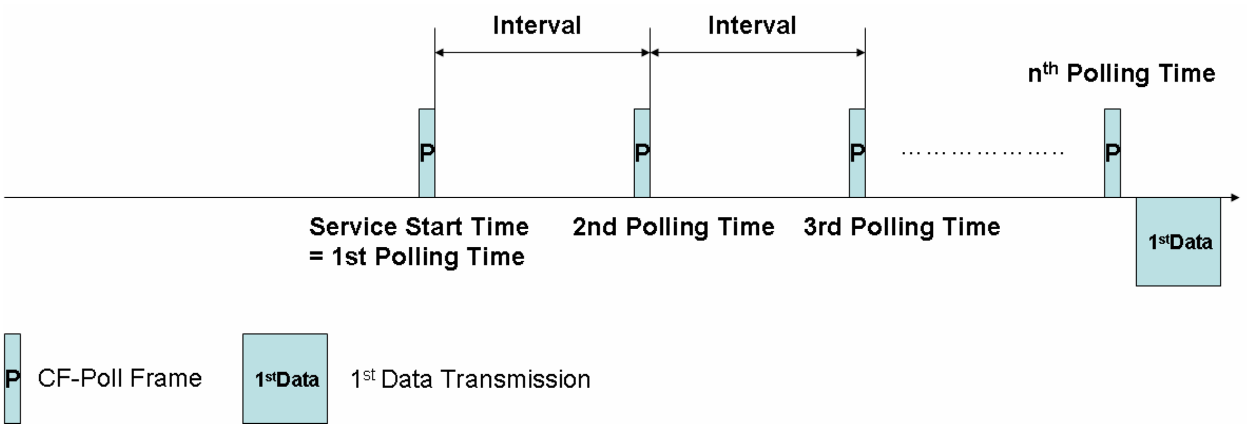

Fig. 3.4 Illustration of n^{th} polling with responding data

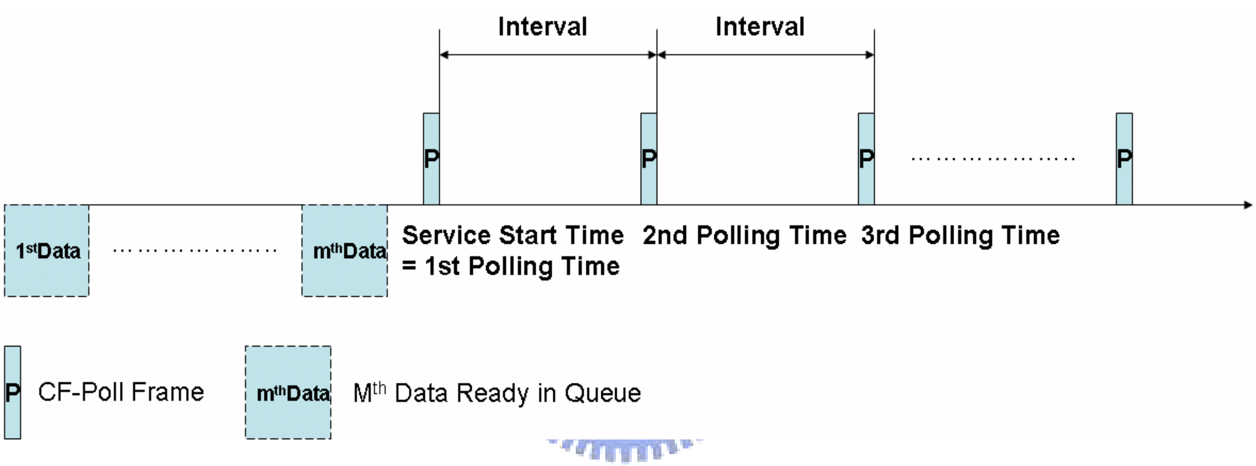

Fig. 3.5 Illustration of polling with responding m^{th} data

We use mathematical equations to express this jitter deviation reduction in our proposed scheme. Since we can not confirm whether the service start time equals to the first packet generating time, so we assume that the HC got response with data frames at first time in n^{th} polling, and the latest one of those transferred frame is the m^{th} frame generated (see Fig 3.4 and 3.5). Then we consider the jitter barely between the QSTAs and the AP. The equations about transmission time and access delay are as follows:

$$TransmissionTime = AccessDelay + PropagationDelay, \quad \ldots\ldots(3.4)$$

where propagation delay means the time of frame propagation in channel and is the same for the frames in the same traffic stream. The access delay is the time that the frame in queue waits to be transmitted.

$$1^{st}\ AccessDelay = n^{th}\ PollingTime + Overhead_1 - m^{th}\ FrameArrivingTime, \quad (3.5)$$

$$2^{nd}\ AccessDelay = n{+}1^{th}\ PollingTime + Overhead_2 - m{+}1^{th}\ FrameArrivingTime, \quad (3.6)$$

$$1{+}i^{th}\ AccessDelay = n{+}i^{th}\ PollingTime + Overhead_{n+i} - m{+}i^{th}\ FrameArrivingTime, \quad (3.7)$$

where polling time is calculated by Equation 3.1, 3.2 and 3.3. Frame arriving time means the time at which the frame was ready and arrived at queue, and overhead is the sum of the delay caused by other frames or TXOP duration, CF-Poll transmission time, inter-frame space before data frame transmission and other delays.

The jitter means the difference of the transmission times of the adjective frames, and the equations about jitter in our proposed scheme are as follows:

$$\begin{aligned}
1^{st}\ Jitter &= 2^{nd}\ TransmissionTime - 1^{st}\ TransmissionTime \\
&= (\ 2^{nd}\ AccessDelay + PropagationDelay\ ) \\
&\quad - (\ 1^{st}\ AccessDelay + PropagationDelay\ ) \\
&= 2^{nd}\ AccessDelay - 1^{st}\ AccessDelay \\
&= (\ n{+}1^{th}\ PollingTime + Overhead_2 - m{+}1^{th}\ FrameArrivingTime\ ) \\
&\quad - (\ n^{th}\ PollingTime + Overhead_1 - m^{th}\ FrameArrivingTime\ ) \\
&= (\ n{+}1^{th}\ PollingTime - n^{th}\ PollingTime\ ) + (\ Overhead_2 - Overhead_1\ ) \\
&\quad - (\ m{+}1^{th}\ FrameArrivingTime - m^{th}\ FrameArrivingTime\ ) \\
&= PollingInterval + (\ Overhead_2 - Overhead_1\ ) - FrameInterval \\
&= Overhead_2 - Overhead_1\ , \quad (3.8) \\
n^{th}\ Jitter &= Overhead_{n+1} - Overhead_n\ , \quad (3.9)
\end{aligned}$$

where the polling interval is the same length as the frame arriving interval in our proposed scheme for identical traffic stream, this is the reason why we can cancel polling interval and frame interval out at Equation 3.8.

From Equation 3.9, we know the general jitter comes from the overhead difference of adjacent frames. The adjacent overhead difference comes mainly from the delays caused by other frames or TXOP duration. However, the length of this delay is no more than the sum of single EDCA TXOP duration and all polled TXOP duration minus one polled TXOP duration, but the probability of this case is very little. This is because the polling start time and the service intervals of different traffic streams are varied.

For this reason, the overheads are usually little, also the differences of the adjacent overheads are very little. Therefore, we proved that the jitter deviation in our adaptive time-stamp polling scheme will be smaller than that in original RR polling scheme based on mathematical analysis as discussed in this section.

3.3 Access Delay Reduction

We can know the access delay of the preceding frame is very close to the access delay of the following one in the same traffic stream if we subtract the access delays (see Equation 3.7) of those adjective frames such as that in the following equation:

$$i+1^{th}\ AccessDelay - i^{th}\ AccessDelay = Overhead_{n+i+1} - Overhead_{n+i}, \quad \ldots\ldots\ldots\ldots\ldots\ldots \ (3.10)$$

we can easily understand that the difference of those adjective access delays will be close to zero, this means that access delays will be vary stable because most of the overheads are very small. However, if there was one QoS traffic stream that transmits first frame with much higher access delay, the access delay of its succeeding frames may be high accordingly.

3.3.1 Relationship of Frame Arriving Time and Access Delay

Assuming the access delay of the frame is the waiting time for transmission in the queue, and the frame arriving time is its arrival time at queue and is ready to be sent. So the average access delay would be decided by the difference of the frame arriving time and the relative polling time.

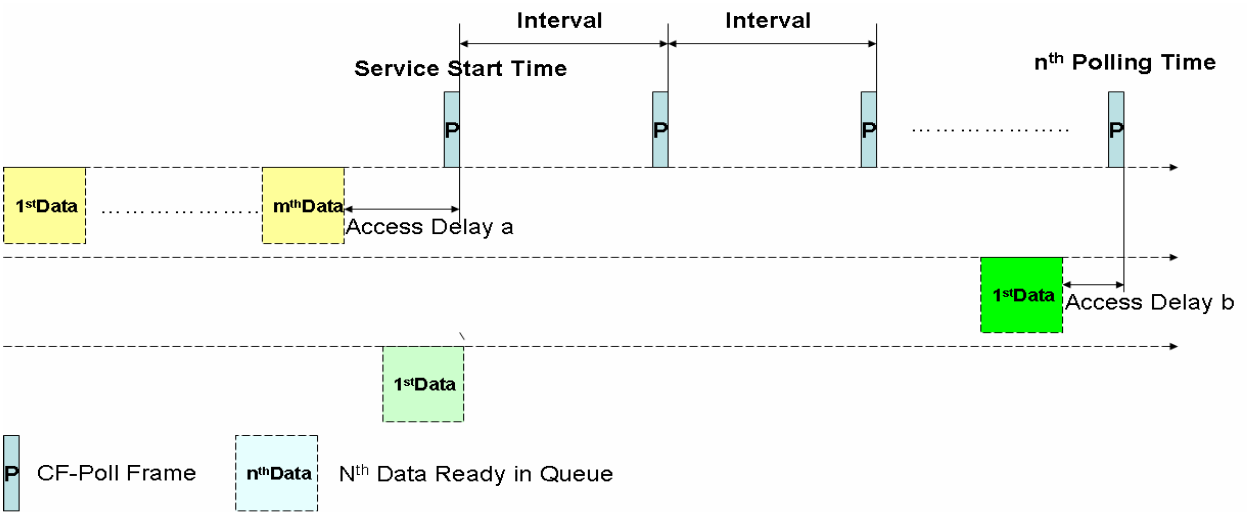

Fig. 3.6 Relationship of frame arriving time and access delay

As Figure 3.6 shows, there are three dotted lines, each represents one situations. If the QoS traffic stream starts before the service start time, the average access delay will be close to the access delay *a*. If it starts after the service start time, the average access delay will be close to the access delay *b*. Since the frame arriving time is not under control by the HC, the average access delay may be up to its polling interval. For example, there is a traffic stream with registered polling interval of 100ms, it may have an average access delay up to 100ms. This issue will be enlarged when the traffic streams have large polling intervals.

3.3.2 Short Interval Polling Function

In order to reduce the average access delay of each traffic stream, we propose a short interval polling function, which is able to reduce the access delay down to less than an artificial number. If we choose 10ms as the short interval for this function, the access delay won't be more than 10ms.

The function is that the HC polls the traffic stream with its short interval after it polled this traffic stream, and was replied QoS data frames at the first time. For example, as Figure 3.7 shows, the original polling interval is much longer than the defined short interval, and the HC received the first QoS data frame at n^{th} polling. Then the HC starts the short interval polling function to poll this traffic stream at short interval. So the new access delay will be less than the short interval and may be much less than the old access delay.

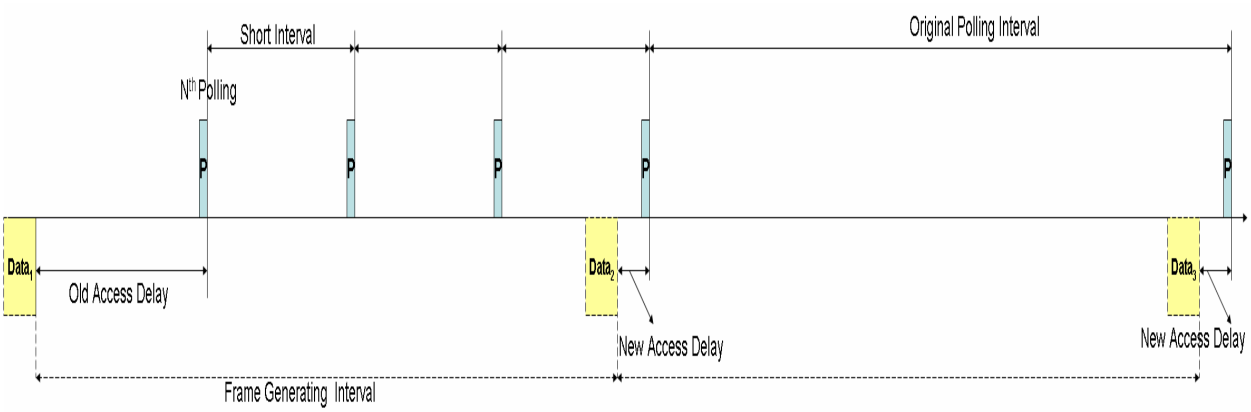

Fig. 3.7 Illustration of short interval polling function

We do not worry about traffic streams in silence mode at the beginning that may interfere this short interval polling function because the HC executes this function only after it received QoS data frames at the first time. It could reduce the possibility that the HC was replied QoS-Null frames for a long time in short interval polling function. However, if there really are such cases, we propose that each execution period of this function must be no more than the length of the original polling interval in order to avoid the wasting. When execution period exceeds the original polling interval, the HC will do poll using original polling interval.

This short interval polling function should be an optional one and not all traffic streams need this function to reduce their own access delays. The HC does not execute this function on those traffic streams if their polling intervals are small enough. It is only executed on those streams whose original polling intervals are twice longer than the defined short interval. Nevertheless, to define the length of the short interval is difficult. The smaller short intervals cause larger polling overheads. In our opinion, 10ms or 20ms is appropriate for the short interval because usually the minimum of those maximum service intervals is only 20ms in most VoIP applications. This function wastes time and enlarge overheads, but the duration to execute it is short and usually each voice or video stream will last for more than tens of seconds. Since we are sure that the access delays would be reduced to a certain extent, this function will do more good than harm.

3.4 Talk-spurt and Silence Detection and Management

In order to perform higher channel utilization, we should consider the talk-spurt and silence alternation of voice traffic streams and try to poll those QSTAs during their talk-spurt periods. There are two problems regarding this issue in point coordination function, the first one is when should be the ends of these talk-spurt periods, and the other problem is when the silence traffic streams would come back to talk-spurt modes.

Generally, it is very difficult for the HC to know the exact starting time and ends of those talk-spurt periods. The HC may still poll a voice traffic stream even during its silence periods, this causes inefficiency. Polling those silence traffic streams wastes bandwidth and increases the access delays of those talk-spurt traffic streams, because it increases the polling overhead when the silence stations replied with a QoS-Null frame.

Consequently, we refer to [6] and [9], then we propose an improved management process to determine whether traffic stream is in talk-spurt mode or silence mode and handle those traffic streams in different modes with different ways. Our method judges the alternation based on the reply to CF-Poll frames. When the HC polled a traffic stream and be replied with QoS data frame, the HC understands that this traffic stream is on the talk-spurt period and the current polling interval should be the same as the interval of its packet arriving interval.

If a traffic stream replied CF-Poll frames with QoS-Null frames for a consecutive number of times, the HC considers this traffic stream as being in the silence mode. We set this specific number to three in the adaptive time-stamp polling scheme. The management process is shown as in Figure 3.8. If three consecutive QoS-Null replies are caused by the short interval polling scheme as we introduced in section 3.3, the HC will not consider this traffic stream as coming back into the silence state. The consecutive replies with QoS-Null frames are only for the normal polling, not for the short interval polling.

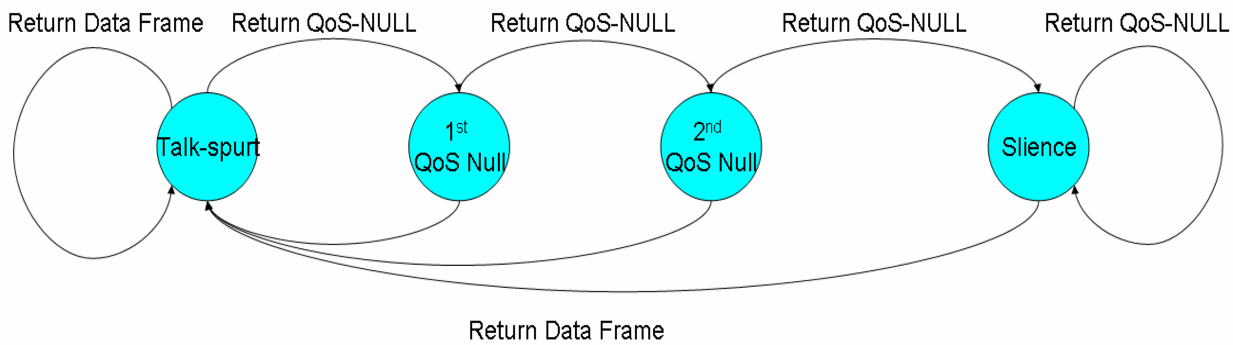

Fig. 3.8 Illustration of talk-spurt and silence management process

According to what we showed in Figure 3.8, the HC uses different polling intervals for the same traffic stream at different states. When those traffic streams are in talk-spurt, 1st QoS-Null or 2nd QoS-Null states, the HC polls them at the intervals which are the same as the intervals of the frame arriving intervals of those traffic streams. When those traffic streams are in silence states, the polling intervals will be increased acutely but no more than 300ms, this is because when delay time of these QoS traffic frames exceeds 300ms, the voice communication quality would be intolerable to users, and we will not take a risk to increase the delay of voice frames. If the traffic stream in silence state replies QoS data frames after the HC polled it, the HC calculates next polling time by Equation 3.3 with original polling interval.

Here we calculate the interval of those silence traffic streams by the following equation:

$$SilencePollingInterval = TalkSpurtPollingInterval \times \left\lfloor \frac{300}{TalkSpurtPollingInterval} \right\rfloor, \quad ...(3.11)$$

where talk-spurt polling interval means the original polling interval used in talk-spurt periods and is same as the maximum service interval registered in the TSPEC. When any QoS traffic stream replies QoS-Null frames for consecutive three times, the HC calculates its next polling time by Equation 3.3 with silence polling interval calculated using Equation 3.11.

With this silence polling interval, we can reduce the polling overhead by increasing the polling interval and make sure the delay of first packet generated by the QoS traffic source from silence state to talk-spurt state will be no longer than 300ms. However, a delay time of 300ms is too large, and after the HC decides that the traffic stream has entered silence state, it is possible that the QoS data frame arrives right away. So we propose an additional method to accommodate this circumstance.

The method is that the silence traffic stream can forward QoS frames through the EDCA function. The HC can easily know which traffic stream this QoS frame belongs to because all QoS frame have QoS control information in their MAC header (see Figure 2.6), which contains the TSID. When the HC discovers that the silence traffic stream is transmitting QoS data frames in a contention period, it will confirm that this traffic stream is in talk-spurt state now and poll this stream with original interval.

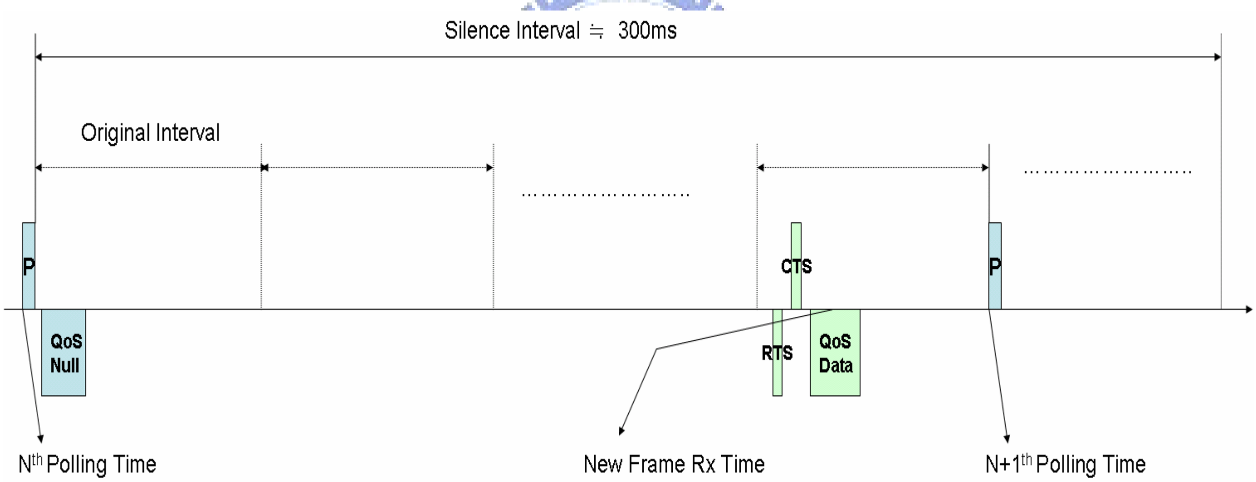

Fig. 3.9 Illustration of TS that comes back to talk-spurt state

In this case, the HC can not use previous equation to calculate the next polling time, this is because the last polling time might be too long ago so that the HC can not estimate correct polling time just by adding one original interval. Other reason is that to use the time of receiving QoS frame as new service start time will alter the original access delay. Hence, the correct next polling time should be calculated by a new equation as follows:

$$n+1^{th}\ PollingTime = TalkSpurtPollingInterval \times \left(\left\lfloor \frac{NewFrameRxTime - n^{th}\ PollingTime}{TalkSpurtPollingInterval} \right\rfloor + 1 \right),$$

……………………………………………………………………………………...(3.12)

where the new frame Rx time means the receiving time for the first QoS frame which was generated by the traffic source from silence state to talk-spurt state (see Fig 3.9).

3.5 Polling List Table

We need a new polling list format and management policy to support our adaptive time-stamp polling scheme. This polling list contains some information that we need, and we can set up this polling list by combining it with the TSPEC table or establishing it in the independent field. No matter what method is used, we will update the information in this polling list, therefore the polling list management will be more complicated than the management in the round-robin polling scheme.

3.5.1 Polling List Format

In order to implement our adaptive time-stamp polling scheme, we construct a new polling list. This polling list contains all information we need such as when and which station the HC should start to poll, and other variables. For this reason, we propose a new polling list format that contains what we need in our adaptive time-stamp polling scheme. Table 3.1 shows the proposed polling list format.

Table 3.1 Proposed polling list format.

TSID	MAC Address	Polling Time	Maximum Service Interval	Number of QoS-Null	Detection State
$TSID_1$	MAC $Address_1$	Polling $Time_1$	Maximum Service Interval $_1$	0	True
$TSID_2$	MAC $Address_2$	Polling $Time_2$	Maximum Service Interval $_2$	3	False
⋮	⋮	⋮	⋮	⋮	⋮
$TSID_n$	MAC $Address_n$	Polling $Time_n$	Maximum Service Interval $_n$	2	False

Each element in the proposed polling list encapsulates the following information:

- TSID: This is the traffic stream identifier to distinguish the traffic stream with a particular specification. The TSID is assigned in the layers above the MAC and the same TSID may be used for multiple traffic streams at different QSTAs.
- MAC Address: This is the MAC address of the QSTA that owns this traffic stream. This element can be convenient to find the corresponding TSPEC .
- Polling Time: This is the expected polling time that is calculated by the HC for the associated traffic stream. The calculation formula is shown in Equations 3.1, 3.2 and 3.3. When the current time exceeds this value, the HC should poll the corresponding QSTA for traffic stream.
- Maximum Service Interval: This is the maximum service interval which is registered in the corresponding TSPEC. This element is used by calculating next polling time at Equations 3.2 and 3.3 in section 3.1.1.
- Number of QoS-Null: This is the number of times that this traffic stream replies the CF-Poll frames with QoS-Null frames consecutively. It is used to determine whether this traffic stream is in talk-spurt or silence periods in the section 3.4.
- Detection State: This is used to decide whether currently the traffic stream needs the

short interval polling function in Section 3.2.2. If this value is true, the HC will execute this option function just after it received the QoS data frames from the corresponding traffic stream. If this value is false, the HC never execute the short interval polling function.

Beside the information in the polling list we proposed, we still need some information that can be recorded in other way. For example, we need to record the value of this short interval which we defined, but it is the same for all traffic streams and is inefficient if we remove it from the polling list. Also some information will vary with the different implementations. For example, the maximum number of times or the end time of executing the short interval polling functions.

3.5.2 Polling List Update

When we try to establish the polling list, we can set up it with an array and then look for the one that has the earliest polling time, or we can build it into a dynamic list which is arranged in order of time when those stations should be polled. No matter what method is used, the HC recorded those initial informations from the TSPEC in the polling list after it accepted this request. The value of Polling Time is the service start time. Then the HC looks for the traffic stream which has the earliest Polling Time and waits until the expected time is up, then polls this stream. If the Polling Time is earlier than the current time, the HC polls as soon as possible. After the HC polls some traffic stream, it must calculate the next polling time and updates this Polling Time in the polling list and repeat the process.

The default value of Detection State element for all traffic streams should be True except those streams whose values of Maximum Service Interval are less than the double of short interval. If the HC receives the QoS data frames from some traffic stream whose Detection State is True, the HC must execute the short interval polling function for this traffic stream. During the period of carrying out this function, the HC would change the Detection State value into False when it receives the following QoS data frames.

The default value of Number of QoS-Null element is zero, the HC increases this element by one every time the HC found this associate traffic stream replies QoS-Null frames after the polling, and the HC will reset this element to zero when it received the QoS data frames from the corresponding traffic stream. When this element is greater than or equals to three, the HC will consider that the traffic stream is in silence state and polls it at the silence polling interval. However, the HC does not change the Number of QoS-Null value during this short interval polling duration because it assumes that this element is implemented correctly.

3.6 TXOP Calculation

When the HC sends CF-Poll frames, it also grants TXOP duration for those traffic streams. Nevertheless, we need to modify the calculation method in the standard a little bit to calculate much suitable TXOPs because we use different polling intervals for different traffic streams in different situations.

To calculate the TXOP intervals for those traffic streams in our adaptive time-stamp polling scheme, the HC uses the following TSPEC parameters: mean data rate, nominal MSDU size, and maximum service interval. When this traffic stream is not in silence state (the value of Number of QoS-Null in the polling list is less than three), the TXOP calculation equation is determined as follows:

(1) Calculate the number of MSDUs that are expected to be sent with the mean data rate for the traffic stream i during the associated polling interval which is same as the maximum service interval:

$$N_i = \left\lceil \frac{MeanDataRate_i \times MaximumServiceInterval_i}{MSDUSize_i} \right\rceil, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.13)$$

where N_i is the expected number of MSDUs to be sent, and $MSDUSize_i$ means the nominal MSDU size for traffic stream i.

(2) Calculate the TXOP duration for the traffic stream i:

$$TXOP_i = \max\left\{\frac{N_i \times MSDUSize_i}{PhyTxRate_i} + O, \frac{MaxMSDUSize}{PhyTxRate_i} + O\right\}, \quad \text{......................... (3.14)}$$

where $PhyTxRate_i$ is the physical transmission rate for traffic stream i, MaxMSDUSize is the maximum allowable size of MSDU, i.e., 2304bytes, and O means the overhead for this transmission.

If the traffic stream is in silence state, we calculate the TXOP duration by Equation 3.14 but the number of MSDUs expected to be sent will be calculated using a different Equation as follows:

(3) Calculate the number of MSDUs during silence periods:

$$N_i = \left\lceil \frac{MeanDataRate_i \times SilencePollingInterval_i}{MSDUSize_i} \right\rceil, \quad \text{.............................. (3.15)}$$

where SilencePollingInterval is the interval which the HC uses to poll this silence traffic stream and is calculated based on Equation 3.11. The N_i is used in Equation 3.14 to calculate the suitable TXOP duration for this silence stream.

When this traffic stream is in the short interval polling duration, the TXOP calculation method is the same as the one which is used in the talk-spurt state. The actual number of MSDUs to be sent during the short interval polling periods would be the same as the number which is expected to be sent during the original polling interval (see Figure 3.7). This is because the packet generating duration or sampling frequency of real-time application may not be altered often.

3.7 Admission Control

We need a precise admission control mechanism in our proposed scheme. Here, we suggest an admission control algorithm which is similar to the algorithm suggested in 802.11e standard. We count total polling overhead, protocol overhead and data transmission time to make sure it won't exceed the available bandwidth. The admission control mechanism is determined as follows:

(1) Choose the maximum interval as the admission control duration among those currently used service intervals for the accepted traffic streams. For example, if some traffic streams are polled at silence interval, we will choose the maximum silence polling interval as the admission control duration.

(2) Calculate maximum overhead and data transmission time of normal traffic stream during the admission control duration:

$$\mathrm{Time}_{normalTS} = \left\lceil \frac{AdmissionControlDuration}{TalkSpurtPollingInterval} \right\rceil \times (PollingOverhead + ProtocolOverhead + DataTxTime), \quad \ldots(3.16)$$

where talk-spurt polling interval means the original polling interval used in talk-spurt periods, polling overhead is one PIFS time and transmission time for one CF-Poll frame, protocol overhead is transmission time for PHY and MAC header of responding frame as well as other IFS time, and DataTxTime is the transmission time for QoS data payload.

(3) Calculate maximum overhead of one silence traffic stream during the admission control duration:

$$Time_{SilenceTS} = \left\lceil \frac{AdmissionControlDuration}{SilencePollingInterval} \right\rceil \times (PollingOverhead + ProtocolOverhead), \quad \ldots(3.17)$$

where silence polling interval is calculated by Equation 3.11. Here, there is no data transmission time because it is silence traffic stream which won't generate QoS data payload and the protocol overhead includes transmission time of QoS-Null frame.

(4) Calculate maximum overhead and data transmission time of traffic stream which executes short interval polling function now during the admission control duration:

$$Time_{sipfTS} = \left\lceil \frac{TalkSpurtInterval}{ShortInterval} \right\rceil \times (PollingOverhead + ProtocolOverhead) + DataTxTime$$

$$+ \left(\left\lceil \frac{AdmissionControlDuration}{TalkSpurtPollingInterval} \right\rceil - 1 \right) * (PollingOverhead + ProtocolOverhead +$$

$$DataTxTime). \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.18)$$

Here, the first part is in short interval polling function so that there is only one QoS data responding, and the second part is in talk-spurt period. This equation is based on the worst cast which has the largest number of polling.

By summation of total calculated time of all current traffic streams, we can simply obtain the channel utilization. We consider that new traffic stream is normal one so that the time for it is calculated by Equation 3.16. Therefore we accept this new traffic stream only when

$$\frac{\sum Time_{normalTS} + \sum Time_{silenceTS} + \sum Time_{sipfTS} + Time_{newTS}}{AdmissionControlDuration} \le \alpha, \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.19)$$

where Timenewts is the expect time for this new traffic stream which is calculated by Equation 3.16. α is set to 0.8, but it can be adjusted depending on the requirement. This algorithm won't fully occupy the channel with HCCA traffic streams, and best-effort traffic could be still transferred.

3.8 Summary

Here we will give a simple summary to walkthrough the total scheme. When the HC accepts a new QoS traffic stream, it records the TSPEC and defines polling time as the service start time. If the current time has already exceeded service start time, the HC polls this traffic stream right away.

Each time the HC sends CF-Poll frames to this stream, it should also calculate next polling time to update the polling list no matter what are responded to the HC. If this traffic stream responds with QoS data frames, the Number of QoS-Null value in polling list will be zero and the HC should decide whether this stream needs the short interval polling function. If this traffic stream responds with QoS-Null frames, the Number of QoS-Null element will be increased by one. When the Number of QoS-Null is more than three, the HC considers this traffic stream as in the silence state.

The HC calculates the polling time by increasing maximum polling interval of this traffic stream only when this stream is neither in the silence state nor during short interval polling duration. If this stream is in the silence state, the HC calculates the polling time by adding the silence polling interval to eliminate the unnecessary polling. If this stream is during short interval polling duration, the HC calculates the polling time by increasing the short interval until it receives the QoS data frames from this stream or the short interval polling period is time out.

Although the ways to implement proposed scheme depends on actual ideas of users, we still propose a simple algorithm as an example to show how we could complete this proposed scheme step by step. The following is the sample pseudo-code which we proposed to implement our adaptive time-stamp polling scheme:

```
Choose the minimum of all expected polling time as pollingTime

   when currentTime >= pollingTime
      poll this TS and wait for reply or timeout;
      if reply is QoS-Data
            NumofQoSNull is set to 0;
            if this is first time to receive QoS data
               and this TS needs the short interval polling function
                  pollingTime = pollngTime + shortInterval;
            else
                  pollingTime = pollngTime + maximumServiceInterval;
            return;
      else
            if this TS is during short interval polling period
                  pollingTime = pollingTime + shortInterval;
                  return;
            else
                  NumofQoSNull increases by one;
                  if NumofQoSNull >= 3
                        pollingTime = pollngTime + silencePollingInterval;
                  else
                        pollingTime = pollngTime + maximumServiceInterval;
                  return;
```

Fig. 3.10 Algorithm of proposed scheme.

Chapter 4   Simulation and Numerical Results

4.1 Simulation environment

We use NS-2 tool (version 2.29) [23] with ns-2 802.11 support [24]. The simulation environment consists of one QAP and a varying number of stations. The number of these QSTAs depends on the requirement of the simulation. All stations operate on IEEE 802.11a. The MAC and physical parameters are shown in Table 4.1.

Table 4.1   MAC and physical parameters

Data rate	54 Mbps
Basic rate	6 Mbps
PHY header	20 us
SIFS	16 us
PIFS	25 us
DIFS	34 us

We use three codec for VoIP, G.711 (160bytes payload, 20ms packet duration), G.723.1 (24bytes payload, 30ms packet duration) and GSM 06.10 (33bytes payload, 50ms packet duration), and three type of video transmission with different frequencies (10fps, 15fps and 30fps) but the same frame size (1000bytes payload). In this simulation, the VoIP transmission consists of alternating talk-spurts and silence intervals. According to [19], we set the talk-spurts period with mean length of 7.24 sec and silence period with mean length of 5.69 second. All video transmissions are constant bit rate (CBR). FTP transmission is used to transfer packet with 1000 bytes payload.

We assumed that there are no hidden stations, thus RTS/CTS feature is turned on, and we compared our scheme with the round robin scheduler. The short interval is defined as 10ms. Those stations start their voice or video transmissions randomly between 2 second and 3 second, and all stations start FTP at 2 second. The simulation ended at 18sec. We analyze the simulation results just between 3 second and 18 second because the simulation network and transmission will be more stable during this period. Simulation results emphysize the comparison of the jitter deviation, the access delay of CBR and VBR traffic streams, the average throughput of total channel between our adaptive time-stamp scheme and the original round-robin polling scheme.

4.2 Numerical Results Analysis

In this chapter, we will compare our adaptive time-stamp polling (ATSP) scheme with the original round-robin polling scheme. Then we will discuss those simulation results and explain that our proposed scheme is better than the original polling scheme.

Table 4.2 Simulation result and comparison

Adaptive Time-Stamp Polling Scheme									
	CBR			VBR			FTP		Total
CBR / VBR	Access Delay Ave/StDev (ms)	Jitter Deviation (ms)	Throughput (KB/sec)	Access Delay Ave/StDev (ms)	Jitter Deviation (ms)	Throughput (KB/sec)	Access Delay Ave/StDev (ms)	Throughput (KB/sec)	Throughput (KB/sec)
3/3	5.36/0.52	1.2870	53.34	5.30/0.34	0.8271	3.35	0.44/0.41	1115.82	1172.51
6/6	5.42/0.73	1.7399	106.38	5.34/0.33	0.8005	5.71	0.47/0.58	1020.83	1132.92
9/9	5.60/1.10	2.6873	159.61	5.38/0.46	1.1445	9.90	0.58/1.08	922.59	1092.10
12/12	5.75/1.11	2.6838	211.73	5.48/0.66	1.5643	15.68	0.73/1.39	805.01	1032.42
15/15	6.08/1.65	4.0470	264.04	5.63/0.94	2.2652	21.02	0.85/1.66	700.39	985.45
18/18	6.16/1.70	4.0900	317.20	5.74/1.02	2.0226	25.74	1.04/2.25	605.47	948.41

Round-Robin Polling Scheme									
	CBR			VBR			FTP		Total
CBR / VBR	Access Delay Ave/StDev (ms)	Jitter Deviation (ms)	Throughput (KB/sec)	Access Delay Ave/StDev (ms)	Jitter Deviation (ms)	Throughput (KB/sec)	Access Delay Ave/StDev (ms)	Throughput (KB/sec)	Throughput (KB/sec)
3/3	14.50/5.74	15.1721	53.22	13.17/4.94	13.2185	3.35	0.48/0.63	1028.58	1085.15
6/6	15.20/5.96	15.3292	105.90	14.78/6.30	17.7249	5.70	0.67/1.77	855.86	967.46
9/9	15.10/5.90	15.5656	158.81	14.41/5.91	15.9606	9.88	0.99/2.91	682.10	850.79
12/12	15.40/5.99	15.6554	210.82	15.79/5.97	16.2959	15.66	1.55/4.32	495.44	721.92
15/15	15.37/6.18	16.3778	262.82	14.87/6.19	15.8832	21.01	2.83/7.92	325.91	609.74
18/18	15.68/6.37	16.5981	315.86	15.32/6.28	16.1110	25.73	6.07/16.52	171.74	513.33

We show our total simulation result first in Table 4.2 in order to present the perspective of those two quite different schemes. The first column is the number of CBR and VBR traffic streams, 3/3 means there are three CBR and three VBR traffic streams and so on. Then we present the average access delay and its standard deviation, the jitter deviation and throughput of those CBR, VBR and FTP traffic. The last column is the throughout of the channel.

In the following sections, we discuss those simulation results of three aforementioned criteria separately, these include average throughputs, jitter deviations and access delays.

4.2.1 Throughput Enhancement

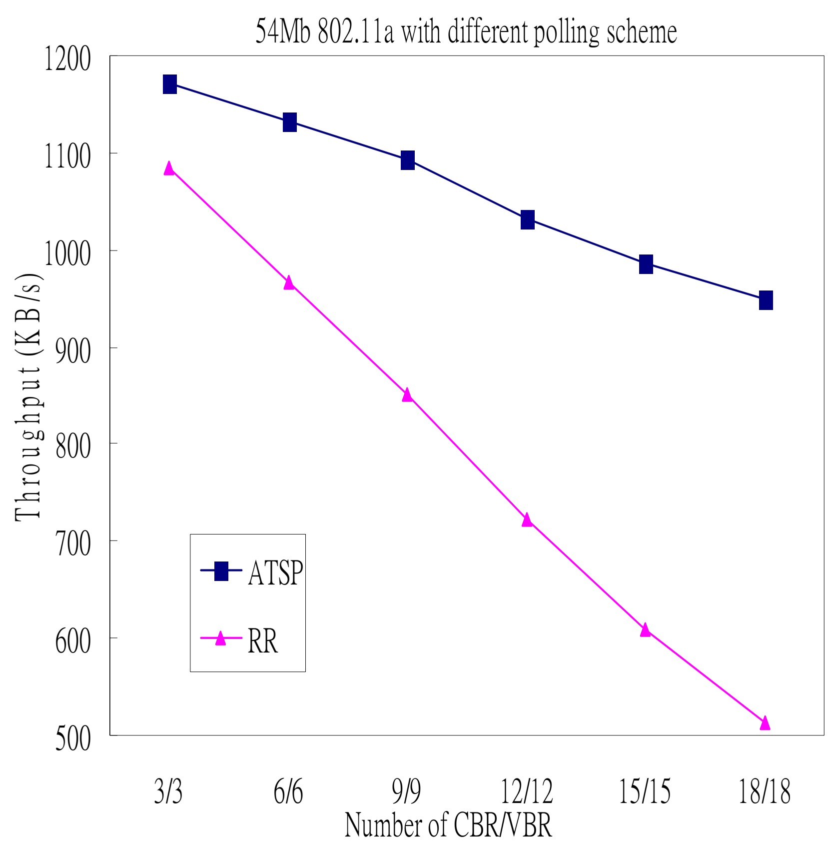

Fig. 4.1 Average throughput against QSTA number for ATSP and RR schemes

Figure 4.1 shows the average throughout between 3sec and 18sec in different network conditions. When more and more QoS traffic streams appears on the same channel, it will cost longer time to access this channel, it will also waste more time on polling overhead, and FTP traffic streams would have less time to contend for the channel. So the average throughput will decrease with the growth of the QoS traffic streams.

We can also notice a more gradual decrease in the average throughput for the adaptive time-stamp polling scheme comparing to round-robin scheme from Figure 4.1. RR scheme starts at 1085 .15 KB/sec and ATSP scheme starts at 1172.51 KB/sec when the numbers of CBR and VBR traffic streams are three. However, the gap between ATSP and RR schemes is getting larger when the number of QoS stations increases. The reason is that the throughputs of CBR and VBR traffic streams are almost the same for both schemes but the throughputs of FTP traffic streams are completely different when the network size is growing up (see Table 4.3). Therefore, ATSP scheme ends at 948.41 KB/sec and RR scheme ends at only 513.33 KB/sec when the numbers of CBR and VBR traffic streams are eighteen. From Figure 4.1 and Table 4.3, while both numbers of CBR and VBR traffic streams increase, ATSP scheme achieves a much better performance than RR scheme on the average throughput.

Table 4.3 Average throughput for different polling schemes

CBR/VBR	***Average Throughput (KB/sec)***					
	ASTP_FTP	*RR_FTP*	*ASTP_CBR*	*RR_CBR*	*ASTP_VBR*	*RR_VBR*
3/3	1115.82	1028.58	53.34	53.22	3.35	3.35
6/6	1020.83	855.86	106.38	105.90	5.71	5.70
9/9	922.59	682.10	159.61	158.81	9.90	9.88
12/12	805.01	495.44	211.73	210.82	15.68	15.66
15/15	700.39	325.91	264.04	262.82	21.02	21.01
18/18	605.47	171.74	317.20	315.86	25.74	25.73

4.2.2 Jitter Deviation Reduction

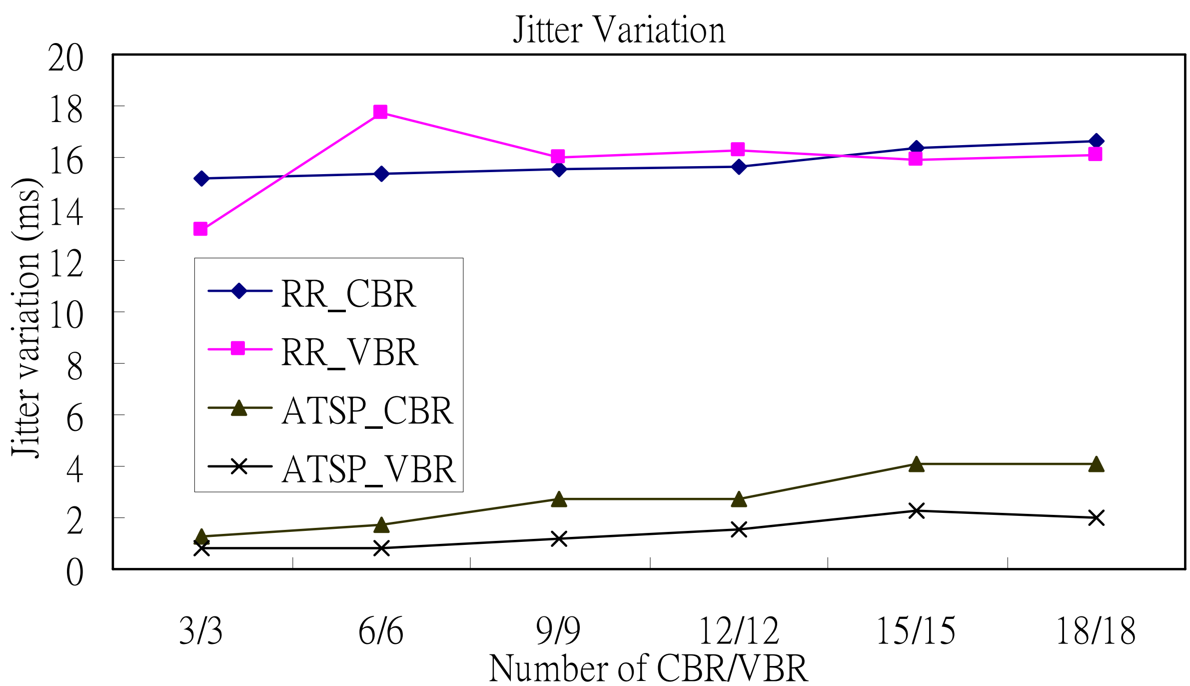

Fig. 4.2 Standard deviation of jitter against number of QSTA

From Figure 4.2, we see the standard deviation of jitter in different numbers of CBR and VBR traffic streams. The jitters are calculated by subtracting previous transmission time from current transmission time of the same QoS traffic stream between the AP and the stations. Generally, we can notice that the jitter standard deviations increase when there are more and more QoS traffic streams on this network just because heavy traffic load will deteriorate the stability of the network.

Figure 4.3 shows the jitter of one CBR traffic stream in RR polling scheme. Figure 4.4 shows the jitter of the same traffic stream in our ATSP scheme. Both are in the networks with eighteen CBR and eighteen VBR traffic streams. We can notice that the variations of jitter in RR polling scheme are relatively large, and the maximum jitter is higher than 30ms (see Figure 4.3). Hence, the standard deviation of jitter for RR scheme will be large certainly. On the contrary, the line in the Figure 4.4 vibrates slightly, and the values are almost restricted within 5ms. This means the jitters of our ATSP scheme are almost negligible, and the simulation result and analysis by Equation 3.9 are matched neatly.

From Figure 4.2, 4.3 and 4.4, ATSP scheme shows a much better jitter reduction than RR scheme regardless of the network size and the numbers of CBR and VBR traffic streams.

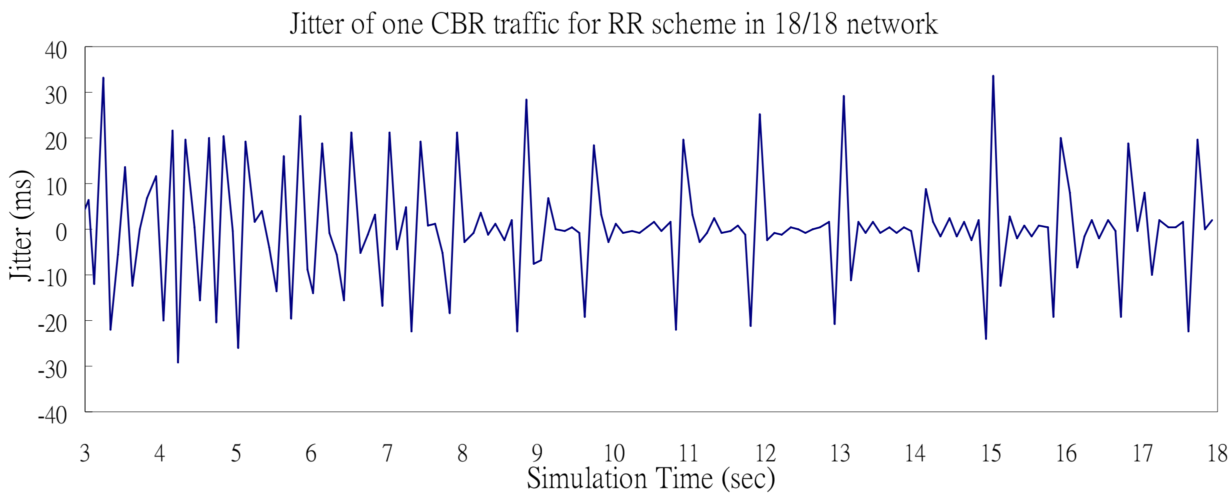

Fig. 4.3 Jitter between packets in RR scheme

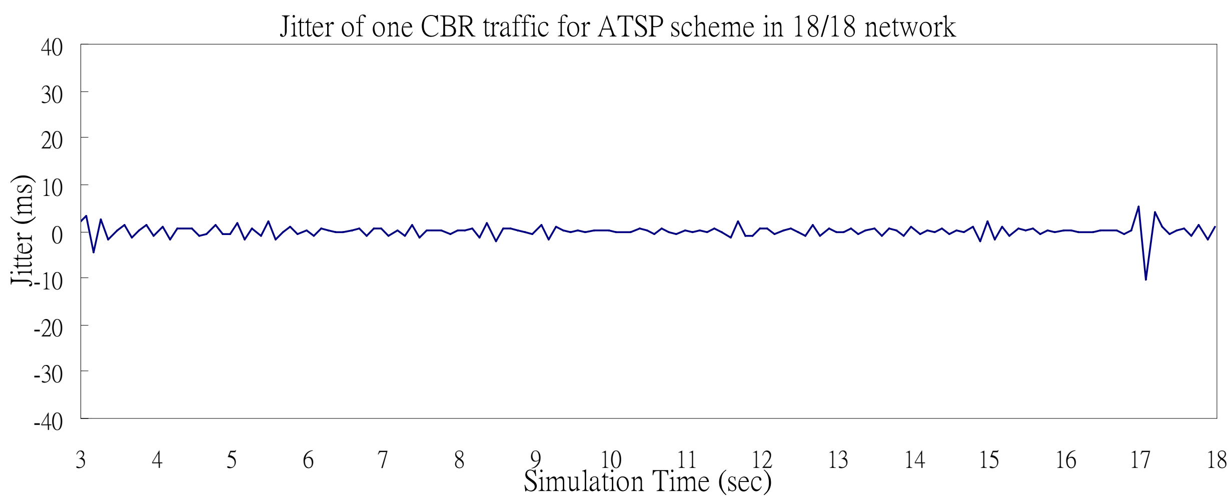

Fig. 4.4 Jitter between packets in ATSP scheme

4.2.3 Access Delay Improvement

Figure 4.5 shows the average access delay against different network sizes for different types of traffic stream and different polling schemes. This figure shows significant gaps between RR scheme and ATSP scheme regardless of the types of traffic streams. The higher delay in RR scheme is due to the polling overheads and inaccurate polling time. While ATSP scheme shows a lower delay comparing to the RR schemes, this is because we can decrease our access delay by reducing the polling overhead and using the short interval polling function. From Figure 4.5, ATSP scheme presents lower access delay than RR scheme regardless of the network size.

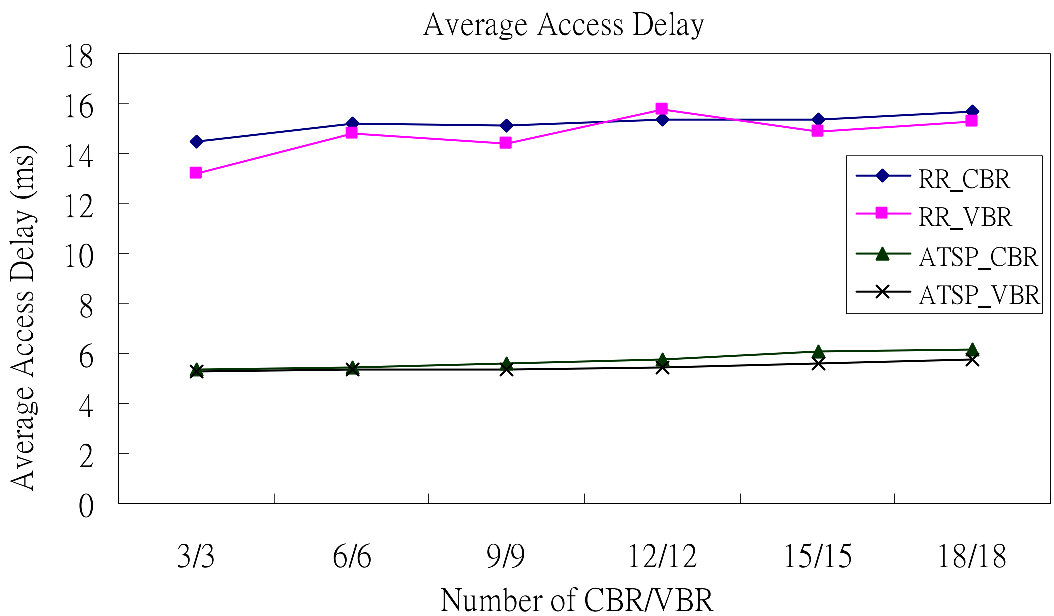

Fig. 4.5 Average access delay against number of QSTA

Chapter 5 Conclusion and Future Works

In order to reduce the polling overhead and provide higher QoS for real-time traffic streams, we proposed a new polling scheme, called Adaptive Time-Stamp Polling (ATSP) scheme. In our proposed scheme, it contains three enhancement parts; one is the adaptive time-stamp poll scheduling which is responsible for calculating the next polling time of each traffic stream. Basically, it calculates by adding the maximum service interval which is registered in the corresponding TSPEC to the current polling time, instead of adding the same service interval for all traffic streams. This method is more flexible and able to reduce the number of unnecessary polling and lessen the jitter deviation.

The second part is the short interval polling function which focuses on the access delay reduction by using the personalized short interval to poll some traffic stream between receiving first QoS data frame and second QoS data frame. During short interval polling periods, it calculates next polling time by just adding the short interval to detect a more accurate polling time. The last part is used to solve talk-spurt and silence alternation problem of voice traffic streams such as VoIP, it detects the silence traffic streams by counting the consecutive QoS-Null replies up to three, and considers them as in the talk-spurt state when receiving QoS data. When it considers that some traffic stream is in silence state, it polls this stream with a longer interval to reduce the overhead.

We use NS-2 tool (version 2.29) with ns-2 802.11 support which contains 802.11e module (both HCCA and EDCA) to simulate our adaptive time-stamp polling scheme and compare it with round-robin polling scheme. The results show that ATSP has significant improvement in terms of throughput, access delay and jitter deviation. The standard deviation of jitter for real-time traffic in ATSP scheme reduces by more than 60% comparing to RR scheme. The average delay in ATSP scheme also decreased more than 50% comparing to RR scheme and the standard deviation of access delay is very little, this means frames will unlikely experience high access delay with our ATSP scheme. The most important thing is that we would not sacrifice the best-effort transmission for real-time applications. Best-effort traffic would not suffer from starvation when there are more and more QoS traffic streams added to the channel, and the total utilization of channel will be also improved.

The complexity of our ATSP scheme is a big issue because the HC needs to calculate polling time of each traffic stream respectively and observe which traffic stream has been in the silence state or during the short interval polling period. The short interval polling period of the traffic stream is much shorter than the whole real-time transmission and we use simple method to detect silence streams, so the HC won't spend much time to observe those traffic streams. Truly complex part is to calculate polling time for each traffic stream. The more QoS traffic streams, the more the calculations. However, those calculations are almost simple additions, so we could conclude that ATSP scheme achieves a significant improvement.

In the future, we will improve our silence detection method to make those decisions more reliable, and find a more efficient mechanism about access delay reduction to minimize extra polling. Besides, the Direct Link Setup (DLS) and Block Acknowledge will be taken into account to support a more reliable and effective transmission for real-time applications on wireless LANs.

References

[1] IEEE Std 802.11e-2005.

[2] Ping Wavg, Hai Jiang, Weihua Zhuang, “IEEE 802.11E Enhancement for Voice Service”, IEEE Wireless Communications, February 2006.

[3] Q. Ni, “Performance Analysis and Enhancements for IEEE 802.11e Wireless Network”, IEEE Network, Vol. 19, pp.21-27, July-Aug. 2005.

[4] Ming-Chuan Hsu; Yaw-Chung Chen, “Enhanced PCF Protocols for Real-time Multimedia Services over 802.11 Wireless Networks”, Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on, pp.56 – 56, 04-07 July 2006.

[5] Minoru Matsumoto and Tadashi Itoh, “QoS-guarantee Method for Public Wireless LAN Access Environments”, 2005 International Conference on Wireless Networks, Communications and Mobile Computing.

[6] Jungbo Son, Hosuk Choi ,Sin-Chong Park, “An Effective Polling MAC Scheme for IEEE 802.11e”, international Symposium on Communications and Information Technologies 2004 (lSCIT 2004), Sapporo. Japan, October 26 - 29, 2004.

[7] Jun Zheng, Emma Regentova, “An Improved Polling Scheme for Voice Support in IEEE 802.11Wireless Network”, Proceedings of the International Conference on Information Technology (ITCC’05).

[8] Ray Y. W. Lam and Victor C. M. Leung, “Polling-Based Protocols for Packet Voice Transport Over IEEE 802.11 Wireless Local Area Networks”, IEEE Wireless Communications, February 2006.

[9] Xiyan Ma, Yanfeng Zhu and Zhisheng Niu, “Dynamic polling management for QoS differentiation in IEEE 802.11e wireless LANs”, 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile, vol.1, pp.152-156, 2004.

[10] Rongbo Zhu and Yuhang Yang, “Adaptive Scheduler to Improve QoS in IEEE 802.11e Wireless LANs”, First International Conference on Innovative Computing, Information and Control (ICICIC '06), vol. 1, pp.377-380, 2006.

[11] Ping-Chi Wang, Kuochen Wang, Lung-Sheng Lee, "A QoS scheme for digital home applications in IEEE 802.11e wireless LANs", IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 3, pp. 1845-1849, 2005.

[12] T. D. Lagkas and G. I. Papadimitriou, "Priority Oriented Adaptive Polling for wireless LANs", 11th IEEE Symposium on Computers and Communication (ISCC'06), 2006.

[13] Xian Ma, Cheng Du, Zhishing Niu, "Adaptive Polling List Arrangement Scheme for Voice Transmission with PCF in Wireless LANs", 10'th Asia-Pacific Conference on Communications and 5th International Symposiumon Multi-Directional Mobile Communication.

[14] Bandinelli. M, Chifi. F, Fantacci. R, Tarchi. D, Vannuccini. G, "A link adaptation strategy for QoS support in IEEE 802.11e-based WLANs", IEEE Wireless Communications and Networking Conference, Vol.1, pp. 120-125, 2005.

[15] Hyun-Jin Lee and Jae-Hyun Kim, "A optimal CF-poll piggyback scheme in IEEE 802.11e HCCA", The 8th International Conference Advanced Communication Technology, Vol. 3, pp.6 ,Feb. 2006.

[16] Il-Gu Lee, Jung-Bo Son, Sung-Rok Yoon and Sin-Chong Park, "Efficient block size based polling scheme for IEEE 802.11e wireless LANs", 2005 IEEE 61st Vehicular Technology Conference, vol. 5, pp. 2869-2873, 2005.

[17] Fang-Yie Leu, Ching-Chien Kuan, Dr-Jiunn Deng, "A QoS provision multipolling mechanism for IEEE 802.11e standard", International Conference on Wireless Networks, Communications and Mobile Computing, vol.1, pp. 392-397, 2550

[18] Paul T. Brady, "A model for generating ON-OFF speech patterns in two-way conversations", Bell System Technical Journal, Vol. 48, pp. 2445-2472, Sept 1969.

[19] Shuang Deng, "Traffic characteristics of packet voice", IEEE International Conference on Communications, Vol. 3, pp. 1369-1374, 1995.

[20] J.Naoum-Sawaya, B. Ghaddar, S. Khawam, H. Safa, H. Artail and Z. Dawy, "Adaptive Approach for QoS Support in IEEE 802.11e Wireless LAN", IEEE WMob'05, Vol. 2, pp/ 167-173, Aug. 2005.

[21] T. Raimondi and M. Davis, "Design Rules for a Class-based Differentiated Service QoS Scheme in IEEE 802.11e Wireless LANs", ACM MSWiM'04, Oct. 2004.

[22] L. Romdhani, Q. Ni and T. Turletti, "Adaptive EDCF: Enhanced Service Differentiation for

IEEE 802.11 Wireless Ad-Hoc Networks", IEEE WCNC, Vol. 2, pp. 1373-1378, Mar. 2003.

[23] http://www.isi.edu/nsnam/ns/

[24] http://yans.inria.fr/ns-2-80211/