Performance Analysis and Enhancements for IEEE 802.11e Wireless Networks

Qiang Ni, National University of Ireland

Abstract

The IEEE 802.11 WLAN legacy standard cannot provide QoS support for multimedia applications. Thus, considerable research efforts have been carried out to enhance QoS support for 802.11. Among them, 802.11e is the upcoming QoS-enhanced standard proposed by the IEEE working group. This article describes in detail the new QoS features of 802.11e based on the latest version of the standard draft. We investigate the performance of 802.11e through computer simulations. Using simple examples, we show the effectiveness of adaptive schemes under the 802.11e framework.

In next-generation wireless networks, it is likely that the IEEE 802.11 wireless LAN (WLAN) [1] will play an important role and affect the style of people's daily life. The 802.11 technology provides cheap and flexible wireless access capability. It is very easy to deploy an 802.11 WLAN in campuses, airports, stock markets, offices, hospitals, and other places. Meanwhile, multimedia applications are increasing tremendously. People want voice, audio, and broadband video services through WLAN connections. Unlike traditional best effort data applications, multimedia applications require quality of service (QoS) support such as guaranteed bandwidth and bounded delay/jitter. Providing such QoS support in 802.11 is challenging since the original standard does not take QoS into account: both the medium access control (MAC) layer and the physical (PHY) layer of 802.11 are designed for best effort data transmissions. Considering that the MAC layer is the key element that provides the QoS performance, the 802.11 standard group is working on a MAC amendment, 802.11e [2]. The main new features of 802.11e are examined in this article. We show through simulations the advantages of the 802.11e framework and the weakness of the default 802.11e configuration. We then describe how the flexibility provided by the upcoming 802.11e standard can be exploited to further enhance the performance over the basic parameter sets.

The rest of this article is organized as follows. We provide a brief overview of the 802.11 standard. We illustrate the lack of QoS support at the existing 802.11 MAC layer. The new features of 802.11e are described. We evaluate the performance of 802.11e. Some adaptive schemes to enhance the performance of 802.11e are discussed. We then conclude the article.

A Review of the IEEE 802.11 Standard

Two kinds of basic network configuration modes are provided in the 802.11 legacy standard: an *infrastructure mode*, where transmissions of all stations (STAs) have to go through a central access point (AP) device, and an *ad hoc mode*, where any STA can talk to another without an AP. The 802.11 standard defines the specifications of both PHY and MAC layers to construct a WLAN using either configuration mode.

The 802.11 PHY Layer

In 1997 the IEEE released the first version of the 802.11 WLAN standard with three kinds of PHY layer options: an infrared (IR) baseband PHY, a frequency hopping spread spectrum (FHSS) radio, and a direct sequence spread spectrum (DSSS) radio. All three options only support 1–2 Mb/s data rates. In 1999 two higher-rate PHY extensions were released by the IEEE:

- 802.11b, based on the DSSS technology, with data rates up to 11 Mb/s in the 2.4 GHz band
- 802.11a, based on orthogonal frequency-division multiplexing (OFDM) technology, with data rates up to 54 Mb/s in the 5 GHz band

In 2003 another version of the standard, 802.11g, that extends the 802.11b PHY layer to support data rates of up to 54 Mb/s in the 2.4 GHz band was finalized. Most recently, a new task group was created to work on the next-generation WLAN standard, 802.11n, which tries to support a maximum throughput of at least 100 Mb/s measured at the interface between the MAC and higher layers. Multiple-input multiple-output (MIMO) antenna technology and adaptive channel coding are likely choices for the main PHY technologies.

The 802.11 MAC Layer

The 802.11 MAC layer aims to provide access control functions to the wireless medium such as access coordination, addressing, frame check sequence generation, and security. There are several ongoing activities to extend the MAC layer protocols, including 802.11e, to enhance QoS performance; 802.11f, proposing an inter-AP protocol to allow STAs to roam between multivendor APs; and 802.11i, focusing on enhanced security and authentication mechanisms.

Two medium access coordination functions are defined in the original 802.11 MAC: a mandatory distributed coordination function (DCF) and an optional point coordination function (PCF).

0890-8044/05/$20.00 © 2005 IEEE

STAs	Voice (ms)	Video (ms)	Background (ms)
4	0.20	0.20	0.23
6	0.47	0.48	0.47
8	0.77	0.78	0.77
10	3.77	3.71	3.77
12	179.75	179.75	178.89
14	296.47	298.17	296.47
16	373.66	371.29	373.66
18	419.44	419.56	419.87

■ Table 1. *Mean delays of multimedia traffic vs. number of STAs.*

DCF: The basic DCF uses a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism to regulate access to the shared wireless medium. Before initiating a transmission, each STA is required to sense the medium and perform a binary exponential backoff. If the medium has been sensed idle for a time interval called DCF interframe space (DIFS), the STA enters a backoff procedure. A slotted *backoff time* is generated randomly from a contention window (CW) size: $\mathit{backoff\ time} = \mathit{rand}[0;\mathrm{CW}] \cdot \mathit{slot\ time}$. At the first transmission attempt, CW is set equal to a minimum value, CW_{min}. It is doubled after each unsuccessful transmission until reaching a maximum value, CW_{max}. It is reset to CW_{min} after successful transmission. The backoff time is decremented by each slot when the medium is sensed idle for that slot. It is frozen if the medium becomes busy, and resumes after the medium has been sensed idle again for another period of DIFS. Only when the backoff time reaches zero is the STA authorized to access the medium. If two or more STAs finish their backoff procedures and detect the medium as idle at the same time, they may transmit frames simultaneously; thus, a collision may occur. A positive acknowledgment (ACK) is used to notify the sender that the frame has been successfully received. The time duration between a data frame and its ACK is the shortest interframe space (SIFS). If an ACK is not received within a time period of $ACK_{Timeout}$, the sender assumes that there is a collision and schedules a retransmission by entering the backoff process again until the maximum retransmission limit is reached.

To deal with hidden terminal problems in which some STAs cannot hear each other and may transmit at the same time to a common receiver, an optional four-way handshake scheme known as request/clear to send (RTS/CTS) can be associated with the basic DCF when data frame size exceeds a value called the *RTS_threshold*. Before any data transmission, a sender can set a duration field in the MAC header of a data frame, or in RTS and CTS control frames. Other STAs hearing an RTS, a CTS, or a data frame can update their local timers, called network allocation vectors (NAVs), to this duration and will not start transmissions before their NAV timers reach zero. A collision may occur only on an RTS frame, and can be detected by the lack of a CTS reply. In this way, system performance can be significantly improved when there are hidden terminals. However, when the data frame size is small, the overheads of RTS/CTS are considerable, and it is recommended that this option be disabled.

PCF: PCF was introduced to support multimedia transmissions and can only be used if a WLAN operates in an infrastructure mode. It is an optional MAC function because the hardware implementation of PCF was thought to be too complicated at the time the standard was finalized. PCF is a polling-based contention-free access scheme, which uses an AP as a point coordinator. When a WLAN system is set up with PCF enabled, the channel access time is divided into periodic intervals called *beacon intervals*. A beacon interval is composed of a contention-free period (CFP) and a contention period (CP). During a CFP, the AP maintains a list of registered STAs and polls them according to the list. Only after a STA is polled can it start transmission. The size of each data frame is bounded by the maximum MAC frame size (2304 bytes). If the PHY data rate of every STA is fixed, the maximum CFP duration for all STAs, *CFP_max_duration*, can then be decided by the AP. However, the link adaptation (multirate support) capability makes the transmission time a frame variable, and may induce large delays and jitters that reduce the QoS performance of PCF. The time used by an AP to generate beacon frames is called the target beacon transmission time (TBTT). The next TBTT is announced by the AP within the current beacon frame. To give PCF higher access priority than DCF, the AP waits for a shorter interval than DIFS, called the PCF interframe space (PIFS), before starting PCF. But PCF is not allowed to interrupt any ongoing frame transmissions in DCF.

QoS Limitations of the 802.11 MAC

Providing QoS for upper layer applications is one of the most challenging functions a wireless MAC layer should support. In particular, wireless links have specific characteristics such as high loss rates, bursts of frame loss, high latency, and jitter. There are several ways to characterize QoS (e.g., parameterized or prioritized [2]). Parameterized QoS is a strict QoS requirement that is expressed in terms of quantitative values, such as data rate, delay bound, and jitter bound. Prioritized QoS is expressed in terms of relative delivery priority, without strict and quantitative service support. This section presents the QoS limitations of the basic 802.11 MAC functions.

QoS Limitations of DCF

In DCF, only best effort service is provided. Time-bounded multimedia applications (e.g., voice over IP, videoconferencing) require certain bandwidth, delay, and jitter guarantees. The point is that with DCF, all the STAs compete for the channel with the same priority. There is no differentiation mechanism to provide better service for real-time multimedia traffic than for data applications. To illustrate the problem, we perform the following simulation using the network simulator ns-2. A variable number of STAs are located within an area where there are no hidden terminals. Thus, the RTS/CTS option is disabled. They operate in ad hoc mode. The PHY layer data rate of each STA is set to 36 Mb/s. Each STA transmits three types of traffic flows (voice, video, and background data flows) using UDP as a transport layer protocol. The MAC layer queue size is set to 50. The voice flow is chosen as a 64 kb/s pulse code modulated (PCM) stream. The transmission rate of a video flow is 640 kb/s with a packet size of 1280 bytes. The transmission rate of background traffic is 1024 kb/s. We vary the load rate from 9.6 to 90 percent by increasing the number of STAs from 2 to 18. Table 1 shows the average delays of the three types of traffic vs. the number of STAs. The queuing delays are included. The mean delays of voice, video, and background flows are lower than 4 ms when the channel load is less than 70 percent (i.e., the number of STAs is below 10). However, when the number of STAs exceeds 10, the mean delays for the three types of flows increase up to about 420 ms and are almost the same for different types of flows. This experiment demonstrates that there is no service differentiation between the different types of flows, which causes a QoS problem for multimedia applications when traffic load is high.

QoS Limitations of PCF

While PCF was designed to support time-bounded multimedia applications, this mode has three major problems that lead to poor QoS performance [3–5]:

- PCF defines only a single-class round-robin scheduling algorithm, which cannot handle the various QoS requirements of different types of traffic.
- The AP schedules each beacon transmission at each TBTT, but the AP has to contend to access the medium. Depending on whether the wireless medium is idle or busy around the TBTT, the beacon frame may be delayed. With PCF, STAs are allowed to transmit even if the frame transmission cannot finish before the next TBTT. The duration of the beacon to be sent after the TBTT defers the transmission of data frames during the following CFP, which introduces delays to those data frames. In the worst case, the maximum delay of a beacon frame can be 4.9 ms in 802.11a, and the average delay of a beacon frame can reach 250 µs [4].
- It is difficult for PCF to control the transmission time of a polled STA. A polled STA is allowed to send a frame of any length between 0 and 2304 bytes, which may introduce variable transmission time. Furthermore, the PHY data rate of a polled STA can change according to varying channel conditions. Thus, the AP is not able to predict transmission time in a precise manner. This prevents the AP from providing guaranteed delay and jitter performance for other STAs present in the polling list during the rest of the CFP interval.

A common QoS problem for both DCF and PCF is that no admission control mechanism is specified in the 802.11 legacy MAC. When traffic load is very high, the performance of both functions can be degraded.

802.11e: Enhanced QoS Support for WLANS

The above mentioned QoS limitations of DCF and PCF have motivated numerous research efforts to enhance the performance of MAC (e.g., [2–10]). Based on different criteria, they can be classified into STA-based vs. queue-based, or DCF-based vs. PCF-based enhancements [5]. Among all the activities, 802.11e [2] is the most promising framework and is expected to become a new industrial standard soon; numerous manufacturers are committed to implementing 802.11e in their new WLAN devices. We describe the details of the 802.11e standard in this section.

In 802.11e a new MAC layer function called the hybrid coordination function (HCF) is proposed. HCF uses a contention-based channel access method, also called enhanced distributed channel access (EDCA), that operates concurrently with a polling-based HCF-controlled channel access (HCCA) method. The AP and those STAs that implement the QoS facilities are called QoS-enhanced AP (QAP) and QSTAs (QoS-enhanced STAs), respectively.

One main new feature of HCF is the concept of transmission opportunity (TXOP), which refers to a time duration during which a QSTA is allowed to transmit a burst of data frames. Thus, the problem of unpredictable transmission time of a polled STA in PCF (as mentioned earlier) can be solved. A TXOP is called an *EDCA-TXOP* when it is obtained by winning a successful EDCA contention, or an *HCCA-TXOP* when it is obtained by receiving a QoS poll frame from the QAP. In order to control the delay, the maximum value of a TXOP is bounded by a value called $TXOP_{Limit}$, which is determined by the QAP. A QSTA can transmit multiple frames within its TXOP allocation. This new feature also tends to provide time-based fairness between QSTAs, which can help remedy the performance anomaly of the legacy MAC (DCF/PCF) when different STAs operate at different data rates, and slow STAs may starve fast ones. The QAP allocates an *uplink* HCCA-TXOP to a QSTA by sending a QoS poll frame to it, while no specific control frame is required for a *downlink* HCCA-TXOP. In the rest of this section we describe first the two subfunctions, EDCA and HCCA, associated with the two different admission control algorithms. We then briefly summarize another two new features defined in 802.11e: BlockACK and direct link protocol (DLP). The latter two are not directly related to the QoS enhancements, but can improve the efficiency of the legacy MAC.

802.11e EDCA: The Contention-Based Part of HCF

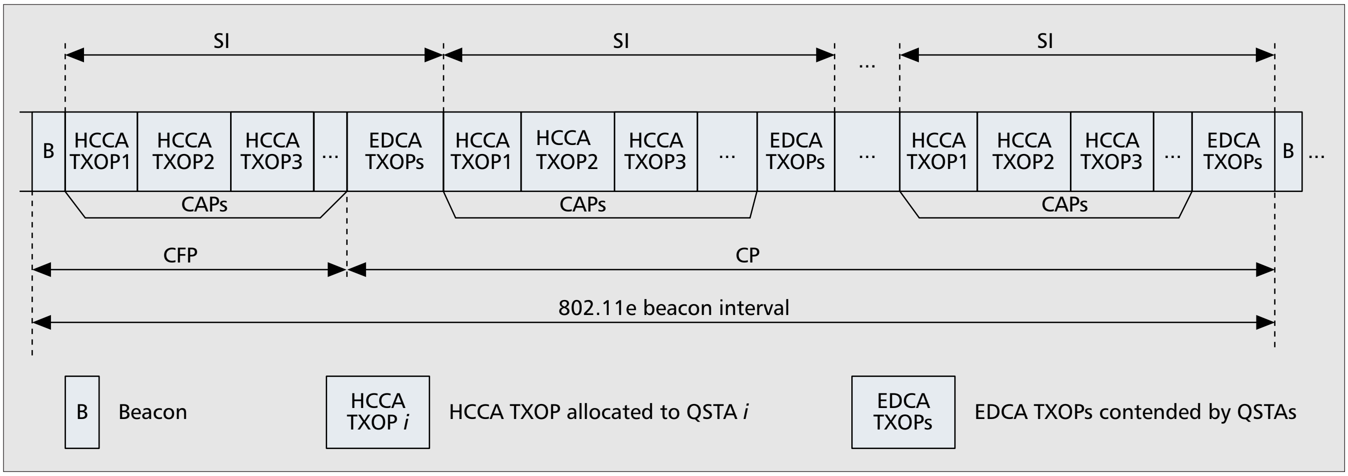
EDCA is designed to provide prioritized QoS by enhancing the contention-based DCF. Before entering the MAC layer, each data packet received from the higher layer is assigned a specific user priority value. How to tag a priority value for each packet is an implementation issue. At the MAC layer, EDCA introduces four different first-in first-out (FIFO) queues, called access categories (ACs). Each data packet from the higher layer along with a specific user priority value should be mapped into a corresponding AC according to a table [2]. Different kinds of applications (e.g., background traffic, best effort traffic, video traffic, and voice traffic) can be directed into different ACs. Each AC behaves as a single DCF contending entity with its own contention parameters ($CW_{min}[AC]$, $CW_{max}[AC]$, $AIFS[AC]$ and $TXOP_{Limit}[AC]$), which are announced by the QAP periodically in beacon frames. Basically, the smaller the values of $CW_{min}[AC]$, $CW_{max}[AC]$, and $AIFS[AC]$, the shorter the channel access delay for the corresponding AC and the higher the priority for access to the medium.

In EDCA a new type of IFS is introduced, the arbitrary IFS (AIFS), in place of DIFS in DCF. Each AIFS is an IFS interval with arbitrary length as follows:

$$AIFS[AC] = SIFS + AIFSN[AC] \cdot slot\ time,$$

where $AIFSN[AC]$ is called the arbitration IFS number. After sensing the medium idle for a time interval of $AIFS[AC]$, each AC calculates its own random backoff time ($CW_{min}[AC] \leq$ backoff time $\leq CW_{max}[AC]$). The purpose of using different contention parameters for different queues is to give a low-priority class a longer waiting time than a high-priority class, so the high-priority class is likely to access the medium earlier than the low-priority class. Note that the backoff times of different ACs in one QSTA are randomly generated and may reach zero simultaneously. This can cause an internal collision. In such a case, a virtual scheduler inside every QSTA allows only the highest-priority AC to transmit frames.

An EDCA contention-based admission control mechanism is suggested in 802.11e that needs the support of both QAP and QSTAs. With this function, QSTAs are required to maintain two local variables, *admitted_time* and *used_time*. They are initially set to zero at the time of association. Each AC in a QSTA transmits a QoS request to the QAP containing a traffic specification (TSPEC) of its application (e.g., mean/peak data rate, mean/maximum frame size). When the QAP receives the request, it decides whether to accept the



■ Figure 1. *An example of 802.11e beacon interval used in HCF scheduling algorithm.*

request. If it accepts the request, it calculates the amount of time per second for admitted traffic to access the medium, which is called *medium_time*. The algorithms used by the QAP to make an admission decision and calculate the medium_time are open. After that, the QAP sends back to the QSTA a response frame containing the derived medium_time. The admitted QSTA updates its local variable admitted_time to this medium_time. To control the total channel access time no longer than the admitted_time, the STA uses another local variable, used_time, to record how long the QSTA has accessed the medium. The used_time is updated after each transmission attempt, successful or not. If used_time is larger than admitted_time, the corresponding AC is not allowed to transmit any data frames until used_time is reset. If a QSTA needs more channel access time for an AC than the admitted_time, it has to send a new request to the QAP.

802.11e HCCA: The Contention-Free Part of HCF

In order to provide parameterized QoS support, HCCA has been proposed in 802.11e. HCCA solves the above three major problems of PCF as follows:

- Different traffic classes called traffic streams (TSs) are introduced in HCCA. Manufacturers can design multiclass scheduling algorithms to support different types of applications. In addition, scheduling algorithms are treated as implementation-dependent in HCCA, and can be enhanced by manufacturers without worrying about standard compliance problems.
- An 802.11e QSTA is not allowed to transmit a packet if the frame transmission cannot finish before the next beacon, which solves the beacon delay problem of PCF.
- A $TXOP_{Limit}$ is used to bound the transmission time of a polled QSTA.

Figure 1 shows an example of an 802.11e beacon interval. During a beacon interval, a QAP is allowed to start several contention-free bursts called controlled access periods (CAPs) using HCCA at any time after detecting a channel as being idle for a time interval of PIFS. Since PIFS is shorter than DIFS and AIFS, a QAP is given a greater opportunity to start HCCA than EDCA. HCCA is more flexible than PCF because the latter is only allowed in a CFP period, while a QAP can initiate HCCA whenever it wishes during the whole beacon interval. Even though PCF is allowed, the flexibility of HCCA makes PCF useless, and PCF is again defined as optional in 802.11e. To leave enough space for EDCA, the maximum duration of HCCA in a beacon interval is limited by a variable $T_{CAPLimit}$.

As a reference design, a simple scheduling algorithm is suggested in 802.11e: Before any data transmission, a traffic stream (TS) is first established, and each QSTA is allowed to have no more than eight TSs with different priorities. Note that TSs (in HCCA) and ACs (in EDCA) can use different MAC queues. In order to initiate a TS connection, a QSTA sends a QoS request frame containing a traffic specification (TSPEC) to the QAP. A TSPEC describes the QoS requirements of a TS, such as mean/peak data rate, mean/maximum frame size, delay bound, and maximum required service interval (RSI). A maximum RSI refers to the maximum duration between the start of successive TXOPs that can be tolerated by a requesting application. Intuitively, there is a link between the maximum RSI and the delay bound for a given TS. Consequently, the 802.11e draft suggests that if both a maximum RSI and a delay bound are specified by a QSTA, the HCCA simple scheduler should only use the maximum RSI for calculating a TXOP schedule. On receiving all these QoS requests, the QAP scheduler first determines the selected service interval (SI), which should be the highest submultiple value of the beacon interval and also be no larger than all the maximum RSIs required by the different TSs from different QSTAs. Then an 802.11e beacon interval is divided into an integer number of SIs, and QSTAs are polled sequentially during each selected SI. In this way, all the admitted TSs should be polled once within the delay requirement of the most time-stringent TS. Lastly, the QAP scheduler computes the corresponding HCCA-TXOP values for different QSTAs by using their QoS requests in TSPECs ($TXOP_1$, $TXOP_2$, etc.), as shown in Fig. 1, and allocates them to those QSTAs.

Similar to the EDCA admission control, an HCCA admission control algorithm is also suggested in 802.11e. Using the TSPEC information, the QAP calculates a ratio of the transmission time reserved for HCCA of all existing K QSTAs over an SI:

$$\sum_{i=1}^{k} \frac{TXOP_i}{SI}.$$

In order to decide whether or not a request from a new traffic flow can be accepted in HCCA, the QAP scheduler only needs to check if the new request $TXOP_{K+1}$ plus all the current TXOP allocations are lower than or equal to the maximum fraction of time that can be used by HCCA:

$$\frac{TXOP_{K+1}}{SI} + \sum_{i=1}^{K} \frac{TXOP_i}{SI} \le \frac{T_{CAPLimit}}{T_{Beacon}}, \tag{1}$$

where $T_{CAPLimit}$ is the maximum duration bound of HCCA, and T_{Beacon} represents the length of a beacon interval.

	Voice	Video	Background (best effort)
Transport protocol	UDP	UDP	UDP
AC	VO	VI	BE
CW_{min}	3	7	15
CW_{max}	7	15	1023
AIFSN	2	2	3
Packet Size	160 bytes	1280 bytes	1500 bytes
Packet interval	20 ms	10 ms	12.5 ms
Sending rate	64 kb/s	1024 kb/s	960 kb/s

■ Table 2. *Simulation parameters for EDCA.*

BlockACK

Associated with TXOPs, the BlockACK mechanism is suggested to allow a group of data frames, back to back in a block, to be transmitted within $TXOP_{Limit}$. Each frame in a block is separated by a SIFS interval, which can reduce the overheads of the immediate ACK transmission. Followed by the data block, a single BlockACK request frame (Block- ACKReq) is transmitted by the sender, and a BlockACK frame from the destination acknowledges the bursts of transmitted data frames. The BlockACK frame contains a 32-byte ACK bitmap field that can indicate successful reception of the burst frames. Each bit in the bitmap acknowledges one data frame transmission. Before starting a block transmission, the sender should first win an EDCA-TXOP by using the EDCA contention mechanism, or gain an HCCA-TXOP during CAPs.

Direct Link Protocol (DLP)

In the legacy standard, if STAs are operated in an infrastructure mode, all communication between any two STAs must go through an AP. When communication is only between the pair, channel resources are wasted. In order to improve efficiency, a direct link between peer STAs is allowed in 802.11e, where STAs can communicate directly using both EDCA and HCCA. DLP allows the transmitter and receiver to exchange transmission rate sets and other information, including security elements, and then creates an active direct link between them. TXOPs associated with DLP are referred to as *directlink* TXOPs, and are also upper bounded by $TXOP_{Limit}$. When a direct link is active, the transmitter may use DLP probes to measure the quality of that link. Once a direct link is found to be inactive and no frames have been exchanged for the duration of *DLPIdleTimeout*, the two QSTAs revert to communicating via the QAP.

Performance Evaluation of 802.11e

Let us evaluate the performance of EDCA and HCCA. We use the ns-2 simulator and ignore random transmission errors, so a transmission failure occurs only when there is a collision. No admission control mechanisms are used.

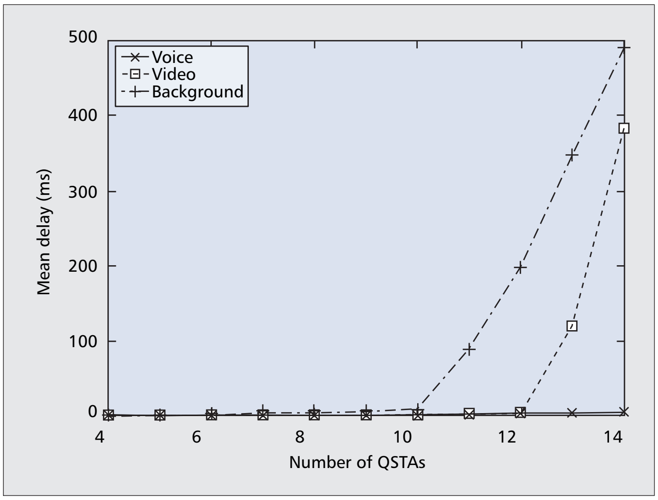
Simulation Analysis of EDCA

In this test each QSTA transmits three types of flows (voice, video, and background data) to the same destination. We choose 802.11a as the PHY layer, and the PHY data rate is set to 36 Mb/s. The simulation parameters are shown in Table 2. All the simulation results are averaged over five simulations, with random starting times for each flow. We vary the channel load from 22 to 80 percent by increasing the number of active QSTAs from 4 to 14.

Figure 2 shows that EDCA can provide the expected service differentiation between different types of traffic: the highest-priority voice flows have lower delays than the video and data flows. Even when the channel is highly loaded (e.g., the number of QSTAs is 14, 80 percent channel load), the mean delay of voice is still around 5 ms. However, the average delay of the video flows increases up to about 400 ms and the data flows have a mean delay of about 500 ms when the medium is highly loaded. The large delays for the video flows are due to the fact that the default *CW* sizes provided in 802.11e are too small for the network to accommodate such heavy traffic. This test demonstrates that while the service differentiation provided by the default EDCA parameter settings is effective, the performance of video and data flows can be degraded when the channel is heavily loaded. Similar observations are also obtained in [6] and an admission control algorithm is shown to be required to protect the existing traffic in EDCA. The above behavior of EDCA comes from the fact that the default values of CW_{min} and CW_{max} for video traffic are very small (7 and 15, respectively). The same backoff time for different video flows are likely to be generated, and thus causes considerable collisions between the video flows as well as between the video flows and the lower-priority data flows when there are a large number of QSTAs. On the other hand, the sending rate of voice traffic is very low (64 kb/s), which encounters few collisions between the voice flows. The service differentiation mechanism sustains 14 voice flows from the contentions with the lower-priority video and data flows. This confirms that the differentiation mechanisms can protect a higher-priority class from a lower-priority class, but cannot reduce the contentions between traffic flows within the same priority class. Fortunately, 802.11e EDCA offers great flexibility to a QAP to adjust backoff parameters for different ACs if the channel condition varies. Based on the 802.11e EDCA framework, we discuss later how adaptive tuning can be performed in EDCA.

Simulation Analysis of HCCA

In order to study the performance of the reference scheduling algorithm in 802.11e, we implemented it in ns-2. The simulation topology contains one fixed QAP and 18 QSTAs with one TS stream per QSTA. In the simulation, six QSTAs sent highest-priority voice streams (64 kb/s PCM coded), six QSTAs sent variable bit rate (VBR) video streams (200 kb/s on aver-



■ Figure 2. *Mean delays for different ACs under EDCA.*

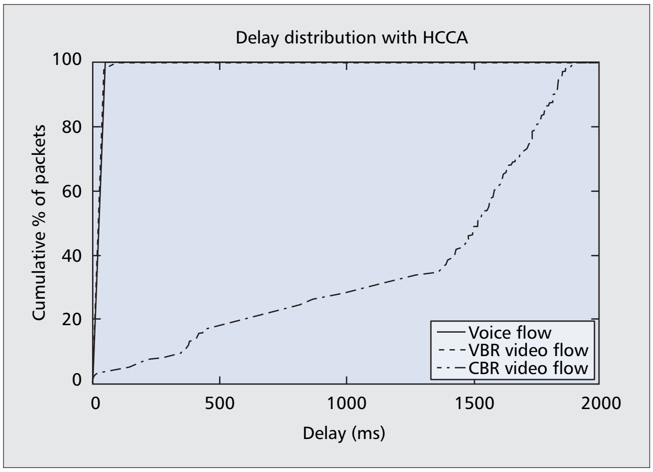


Figure 3. *Delay distribution of the reference scheduler for HCCA.*

age) with medium priority, and six QSTAs sent constant bit rate (CBR) MPEG4 video streams (3.2 Mb/s) with lowest priority to the QAP. In this implementation, neither a sender nor a receiver drops a packet actively if its delay is larger than the delay bound. A packet is dropped only if the MAC queue is full. The maximum queue size is 50. All streams use UDP as the transport layer protocol. The beacon interval is 500 ms. The RSI of voice traffic is 50 ms, and the RSIs of both CBR and VBR types of video traffic are 100 ms. According to the HCCA scheduling algorithm described earlier, the SI is chosen as 50 ms. Several VBR video flow traces have been obtained with the VIC videoconferencing tool using the H.261 codec and QCIF format for typical "head and shoulder" video sequences. The mean transmission rate and mean packet size of the VBR video flows are precalculated and specified in the TSPEC, and are 200 kb/s and 660 bytes, respectively.

Figure 3 plots the delay distribution for different flows with HCCA. We observe that HCCA delivers 99 percent of voice packets within a delay of 50 ms and 97 percent of CBR video packets within 50 ms, which is equal to the selected SI duration. It shows that for CBR traffic, HCCA can guarantee the delay requirement by using the reference scheduler.[1] On the other hand, the delays of VBR video flows are completely uncontrolled. In the worst case, the packet delays can be around 2000 ms. This is due to the fact that those packets have been queued for a duration of 40 times SI until other earlier arrived packets in the buffer are delivered given that the maximum queue size is set to 50. In this test, the video applications report their mean transmission rates to the QAP. During each SI, the simple HCCA scheduler allocates a fixed TXOP to each QSTA based on its mean rate requirements. Since the actual transmission rates of VBR flows are fluctuating and sometimes larger than the mean rates, the packets are queued and delays are increased. If the peak transmission rates of VBR video applications are reported to the QAP and used to calculate the TXOPs, the TXOPs will be large enough for delivering packets, and the delays of VBR applications can be controlled. However, a smaller number of VBR flows can be admitted than in the case where the mean rates are used. In such a case the channel will be underutilized, especially when the gaps between the peak and mean rates are considerable. Furthermore, to poll all the TSs with the same period (the selected SI) is not efficient since different types of traffic may generate bursts at different frequencies. Hence, adaptive scheduling algorithms that take into account fluctuations of traffic transmission rates are suggested [8, 9]. Next, we describe an adaptive scheduler, AHCCA, which shows how the performance of the 802.11e reference scheduler can easily be improved.

Adaptive Schemes for 802.11E

In this section we discuss how adaptive approaches can be helpful for improving the performance of both EDCA and HCCA in varying traffic conditions.

Adaptive Tuning Mechanism for EDCA

As shown earlier, the default CW parameter sets of the video class are too small when there are a large number of video flows. One can choose larger CW sizes as the default parameter sets. However, when traffic load is low, larger CW sizes induce larger channel access delays and thus reduce channel efficiency. Therefore, an adaptive mechanism for a QAP to adjust CW sizes according to the channel condition is attractive. In order to perform a perfect tuning mechanism that achieves a theoretical optimal performance, the QAP should have exact knowledge of the network contention level, which may be hard to attain in practice. As an alternative, QAP can tune the CW_{min} values of different classes using some simple measurements (e.g., estimating channel collision rates) while maintaining the relative service differentiation. We design a simple tuning mechanism called AEDCA based on the 802.11e EDCA framework [7]. In AEDCA, each QSTA monitors the interface, calculates the collision rates, and reports them to the QAP. In order to estimate the collision rates, each AC class in a QSTA records the number of unsuccessful transmissions (N_{coll}) and the total number of transmissions during that period (N_{trans}). Supposing that a channel is error-free, all failed transmissions without receiving ACK frames can be regarded as collisions. The collision rates can then be calculated: $p^i_{curr} = N_{coll}/N_{trans}$. We use the following smoothing function to obtain the average collision rates:

$$p^i_{avg} = (1-\alpha)\cdot p^i_{curr} + \alpha \cdot p^{i-1}_{avg}, \qquad (2)$$

where i refers to the ith measuring period, and α is a smoothing factor in the range of [0,1].

After receiving the average collision rate measures from different QSTAs, the QAP tunes the CW sizes for different classes [7]. Figure 4 shows the average delays of different ACs. A comparison between Figs. 4 and 2 demonstrates that

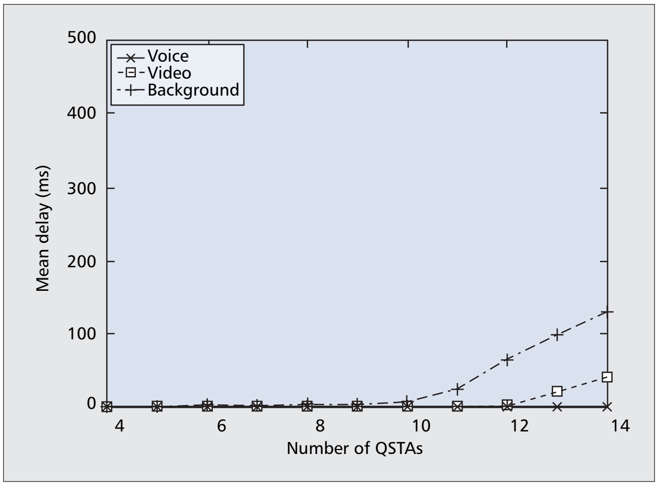
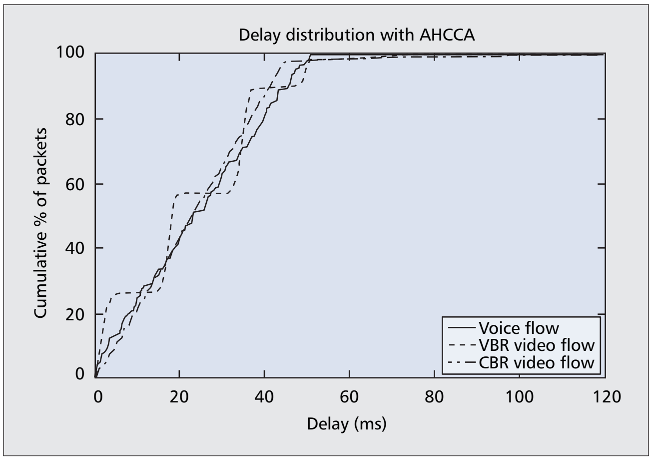


Figure 4. *Mean delays for different ACs under AEDCA.*

---

[1] *Note that voice traffic is modeled by an on/off VBR source. Its peak rate is used for scheduling, so it receives a large enough TXOP allocation for transmission.*



■ Figure 5. *Delay distribution for AHCCA.*

with a simple tuning mechanism, the delays of video and background flows are much lower than those provided by the default EDCA configurations when a channel is highly loaded.

An Adaptive Scheduling Algorithm for HCCA

Recently, several adaptive scheduling algorithms (e.g., [8, 9]) have been proposed for the HCCA framework. In [8] a delay-bound-based earliest due date (SETT-EDD) scheduling algorithm is proposed, which requires some additional information from applications (e.g., maximum burst size, peak data rate). The AHCCA scheduling algorithm [9] proposes to use the queue length information and its estimate for adaptation. Note that allowing a QAP to obtain previous queue length information from QSTAs is a new feature supported in 802.11e.

In AHCCA, before each SI interval, the QAP scheduler estimates the varying queue length of each TS. Since the estimate may be incorrect, the QAP scheduler adjusts its estimate by minimizing prediction errors after obtaining the real queue length information from polling the QSTA. The AHCCA scheduler then compares the adjusted estimate with the *ideal queue length* before allocating a TXOP to a QSTA. Here, the ideal queue length refers to the queue size at the beginning of the next SI if the transmission rate is constant, as specified in the TSPEC request. Actually, the HCCA reference scheduling algorithm assumes that the queue length follows such an ideal rule, but this is not always true since traffic is VBR. Based on the difference between the estimated and ideal queue lengths, the AHCCA scheduler adjusts HCCA-TXOP allocations among different TSs with the constraint that the total TXOP allocations should be bounded by $T_{CAPLimit}$. It allocates more TXOPs to QSTAs that have larger queue sizes than the ideal queue length, or removes some time allocations from those QSTAs with smaller queue sizes than the ideal case. In this way, channel efficiency is improved. When it is time for the QAP to poll a QSTA, the QAP scheduler allocates the adjusted TXOP to the QSTA. Figure 5 shows that with the AHCCA enhancement, the maximum delays of all flows can be bounded by the selected SI (50 ms), as expected by the 802.11e reference scheduler.

Conclusion

In this article we have described the QoS limitations at the 802.11 MAC layer. We have also examined the upcoming QoS-enhanced standard, 802.11e. Our performance evaluation of 802.11e shows that:

- The contention-based EDCA mechanism can provide effective service differentiation between different types of traffic. However, the default CW values provided in the 802.11e draft are too small for a large number of users. Adaptation of backoff parameters can be helpful when the channel condition is varying.
- The polling-based HCCA mechanism is more flexible than the legacy PCF scheme. The simple scheduler suggested in the standard draft performs well when traffic is CBR-like. In the case of VBR traffic, adaptive scheduling algorithms can easily be implemented under the HCCA framework.

Admission control mechanisms are important for both EDCA and HCCA, and require further study. Accurate 802.11e analytical models under different channel conditions and real testbed experiments are needed to optimize the performance of 802.11e networks.

Acknowledgments

This work was supported in part by the French National Telecommunication Research Project (VTHD++) and Science Foundation Ireland WLAN Project (grant no. 03/IN3/I396). The author would like to thank his colleagues at INRIA, the Hamilton Institute, the guest editors of *IEEE Network*, and the anonymous reviewers for their valuable comments and help.

References

[1] IEEE 802.11 WG, "International Standard for Information Technology — Local and Metropolitan Area Networks-Specific Requirements — Part 11: Wireless LAN MAC and PHY specifications," 1999.

[2] IEEE 802.11 WG, "Draft Supplement to Standard for Telecommunications and Information Exchange between Systems-LAN/MAN Specific Requirements — Part 11: Wireless MAC and PHY Specifications: MAC Enhancements for QoS," IEEE 802.11e/draft 11.0, Oct. 2004.

[3] S. Choi *et al.*, "IEEE 802.11e Contention-based Channel Access (EDCF) Performance Evaluation," *Proc. IEEE ICC*, Anchorage, AK, May 2003.

[4] S. Mangold *et al.*, "IEEE 802.11e Wireless LAN for Quality of Service," *Proc. Euro. Wireless*, Florence, Italy, Feb. 2002.

[5] Q. Ni, L. Romdhani, and T. Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN," *Wiley J. Wireless and Mobile Comp.*, vol. 4, no. 5, Aug. 2004, pp. 547–66.

[6] Y. Xiao, and H. Li, "Evaluation of Distributed Admission Control for the IEEE 802.11e EDCA," *IEEE Commun. Mag.*, vol. 42, no. 9, Sept. 2004, pp. S20–S24.

[7] L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks." *Proc. IEEE WCNC*, New Orleans, LA, Mar. 2003.

[8] A. Grilo, M. Macedo, and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE802.11E Networks." *IEEE Wireless Commun.*, vol. 10, no.3, June 2003, pp. 36–43.

[9] P. Ansel, Q. Ni, and T. Turletti, "An Efficient Scheduling Scheme for IEEE 802.11e." *Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* Cambridge, U.K., Mar. 2004.

[10] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE JSAC*, vol. 18, no. 3, Mar. 2000, pp. 535–48.

Biography

QIANG NI [M] (Qiang.Ni@ieee.org) received B.Eng., M.S., and Ph.D. degrees from Huazhong University of Science and Technology (HUST), China, in 1993, 1996, and 1999, respectively. From 1999 to 2001 he was a post-doctoral fellow in the wireless and multimedia communication laboratory at HUST. He visited Microsoft Research Asia Laboratory in 2000. In 2001 he joined INRIA, France, where he was a research staff member in the Planete group. He is currently a senior researcher at Hamilton Institute, National University of Ireland Maynooth. Since 2002 he has been a voting member of the IEEE 802.11 wireless LAN standard working group. He has served as a TPC member for several communication conferences such as IEEE GLOBECOM '05, WirelessCom '05, WiOPT '05, and VTC '03. His research interests include wireless communication network protocol design and performance analysis.