Mobile Netw Appl
DOI 10.1007/s11036-008-0111-z

Analytical Study of TCP Performance over IEEE 802.11e WLANs

Jeonggyun Yu · Sunghyun Choi · Daji Qiao

Abstract IEEE 802.11 Wireless LAN (WLAN) has become a prevailing solution for broadband wireless Internet access while the Transport Control Protocol (TCP) is the dominant transport-layer protocol in the Internet. Therefore, it is critical to have a good understanding of the TCP dynamics over WLANs. In this paper, we conduct rigorous and comprehensive modeling and analysis of the TCP performance over the emerging 802.11e WLANs, or more specifically, the 802.11e Enhanced Distributed Channel Access (EDCA) WLANs. We investigate the effects of minimum contention window sizes and transmission opportunity (TXOP) limits (of both the AP and stations) on the aggregate TCP throughput via analytical and simulation studies. We show that the best aggregate TCP throughput performance can be achieved via AP's contention-free access for downlink packet transmissions and the TXOP mechanism. We also study the effects of some simplifying assumptions used in our analytical model, and simulation results show that our model is reasonably accurate, particularly, when the wireline delay is small and/or the packet loss rate is low.

J. Yu · S. Choi (✉)
School of Electrical Engineering and INMC,
Seoul National University, Seoul, 151-744, Korea
e-mail: schoi@snu.ac.kr

J. Yu
e-mail: jgyu@mwnl.snu.ac.kr

D. Qiao
Department of Electrical and Computer Engineering,
Iowa State University, Ames, IA 50011, USA
e-mail: daji@iastate.edu

Keywords TCP · IEEE 802.11e WLAN · EDCA · TXOP · PIFS access

1 Introduction

In recent years, IEEE 802.11 WLAN has become the dominant technology for indoor broadband wireless networking, where the mandatory protocol of the 802.11 MAC is the Distributed Coordination Function (DCF) [8]. On the other hand, more than 90% of the Internet traffic is carried by the Transport Control Protocol (TCP) [10]. Consequently, most of the data traffic over the 802.11 WLAN is TCP traffic. Therefore, it is critical to have a good understanding of the TCP dynamics over WLANs.

It has been shown in [6, 17] that, with multiple stations carrying *long-lived* TCP flows in an 802.11 WLAN, the number of stations that are actively contending remains very small. Hence, the aggregate TCP throughput is almost independent of the total number of stations in the network. This interesting phenomenon is due to (i) the closed-loop nature of TCP flow control and (ii) the bottleneck downlink (i.e., AP-to-station) transmissions. More specifically, since a long-lived upload TCP flow usually operates in the congestion avoidance phase, the source station can only transmit a TCP Data after it receives a TCP Ack from the AP; meanwhile, each client station of a download TCP flow only replies with a TCP Ack upon reception of a TCP Data from the AP. On the other hand, the 802.11 DCF guarantees long-term equal channel access probabilities among contending stations, regardless whether it is an AP or a station. As a result, most of the outstanding TCP packets (in the form of either

TCP Ack for upload or TCP Data for download) are accumulated at the AP, and hence most of the stations remain *inactive* waiting for their turns to transmit.

The above observations do not always hold in the emerging IEEE 802.11e WLANs, because the 802.11e MAC supports service differentiation. With the 802.11e EDCA [9], the AP can perform service differentiation between itself and stations as well as among stations with different classes of traffic, thanks to its capability of setting flexible channel access parameters and transmission opportunity (TXOP) durations. For example, if the AP uses a smaller minimum contention window size or a longer TXOP duration than stations, the aforementioned TCP packet accumulation at the AP may be alleviated.

There have been efforts to understand the TCP dynamics over legacy 802.11 WLANs [3, 4, 6, 11, 16]. However, none of them analyzed the TCP dynamics over 802.11e WLANs. In [6], Choi et al. analyzed the number of active stations using a Markov chain model, but simply computed the state transition probabilities based on empirical results. In [11], Kherani et al. conducted a theoretical analysis on the TCP performance over multi-hop 802.11 WLANs; in contrast, our study focuses on infrastructure-based WLANs. Burmeister et al. analyzed the TCP over multi-rate WLANs [4]. However, they approximated the AP behavior as an M/G/1/B queuing system, and did not study the effects of contention-based MAC on TCP dynamics in detail. Bruno et al. addressed this problem using a p-persistent DCF model [3] under a critical assumption that no more than one TCP packet is enqueued in the buffer of an active station. This implies that a station becomes inactive after transmitting its packet successfully while each successful transmission from the AP always activates a different station, which is not true in practice. In a real network, the number of outstanding packets in a long-lived TCP flow is usually larger than one. Moreover, each successful transmission from the AP may not always activate a different station, because the AP could transmit a packet to an already active station. Yu et al. analyzed the TCP dynamics over legacy 802.11 DCF-based WLANs using Markov chain model [16], which is the basis of the approach employed in this paper.

In this paper, we conduct rigorous and comprehensive modeling and analysis of the TCP performance over EDCA-based IEEE 802.11e WLANs. Our analytical model is based on the p-persistent model developed by Cali et al. [5]. We study the effects of the minimum contention window sizes (CWmin), arbitration interframe space (AIFS) sizes, and the TXOP durations of both the AP and stations on the aggregate TCP throughput, and verify them using simulation results. We show that (i) the aggregate TCP throughput can be enhanced via a proper combination of CWmin values for the AP and stations, and (ii) the best aggregate TCP throughput performance can be achieved via AP's contention-free access for downlink packet transmissions and the TXOP mechanism.

The rest of the paper is organized as follows. Section 2 briefly introduces the IEEE 802.11e EDCA, the TXOP mechanism, the p-persistent CSMA/CA model, and the dynamics of long-lived TCP flows. Section 3 describes our analytical model for TCP over the 802.11e EDCA. Section 4 extends the analysis in Section 3 to consider the TXOP mechanism. AP's contention-free access for downlink TCP packet transmissions is presented in Section 5. In Section 6, our analytical models are evaluated via simulation, and the effects of AP/station access parameters and TXOP durations on the aggregate TCP throughput are studied. We also discuss briefly the effects of various assumptions on the accuracy of our models. We conclude the paper in Section 7.

2 Preliminaries

2.1 IEEE 802.11e EDCA and TXOP

The 802.11 DCF [8] does not support service differentiation. Basically, the DCF provides long-term fair channel access to contending stations with equal opportunities. In contrast, the 802.11e EDCA [9] provides four access categories (ACs) and the channel access function of each AC is an enhanced variant of the DCF. Specifically, each EDCA channel access function uses AIFS[AC], CWmin[AC], and CWmax[AC] instead of DIFS, CWmin, and CWmax as in the DCF. AIFS[AC] is determined by

$$\text{AIFS[AC]} = \text{SIFS} + \text{AIFSN[AC]} \cdot \sigma, \tag{1}$$

where AIFSN[AC] is an integer greater than 1 for stations and an integer greater than 0 for the AP, and σ is a backoff slot time. The backoff counter is selected randomly from [0, CW[AC]]. Basically, if an EDCA channel access function uses smaller AIFSN[AC] and CWmin[AC], it contends for the channel more aggressively. Hence, it may use more bandwidth than other ACs which use larger AIFSN[AC] and CWmin[AC].

In an 802.11e EDCA WLAN, after a station transmits a packet successfully, it resets its CW value to CWmin, and performs AIFS deference and random backoff. This is often referred to as *post backoff* since this backoff is done after, not before, a transmission.

Figure 1 Illustration of transmission opportunity (TXOP) operation

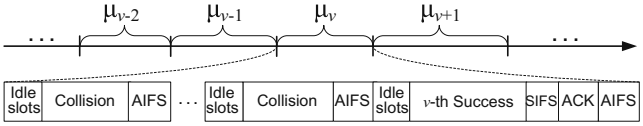


Figure 2 Illustration of virtual slots (μ) in the p-persistent model

Post backoff ensures that there is at least one backoff interval between two consecutive frame transmissions. On the other hand, when there is no on-going backoff and when the channel has been idle for longer than AIFS time, a frame can be transmitted immediately without additional backoff. This is referred to as *immediate access*.

TXOP (transmission opportunity) is a new concept introduced in the 802.11e. A TXOP is a time interval during which a particular station is allowed to transmit. As shown in Fig. 1, a station may transmit multiple packets (separated by SIFS) during a TXOP, which enhances the communication efficiency by reducing unnecessary backoffs [13]. A TXOP can be obtained by a successful EDCA contention. The maximum duration of a TXOP (called TXOPLimit) can be differently configured for different ACs. With a larger TXOPLimit, an AC can use more bandwidth than the others.

The values of AIFSN[AC], CWmin[AC], CWmax [AC], and TXOPLimit[AC], which are referred to as the EDCA parameter set, are determined and advertised to stations by the AP via Beacons and Probe Response frames. Stations should use the announced values for their channel access and transmissions while the AP can use the different values from those announced to stations. In this paper, we study how the aggregate TCP throughput performance may be enhanced by using different EDCA parameter sets for the AP and stations. Service differentiation among different traffic priorities (i.e., ACs) is not the focus of this paper. Accordingly, only a single AC, i.e., best effort AC, is considered.

2.2 p-persistent CSMA/CA model

Cali et al. [5] studied the DCF behavior with a p-persistent model. Instead of using the binary exponential backoff, the p-persistent model determines the backoff interval by sampling from a geometric distribution with parameter p. Thanks to the memoryless property of the geometric distribution, it is more tractable to analyze the p-persistent model. Based on the geometric backoff assumption, the processes that define the occupancy behavior of the channel (i.e., idle slots, collisions, and successful transmissions) are regenerative, where the regenerative points correspond to completions of successful transmissions. The time interval between two consecutive successful transmissions is defined as a virtual slot. As shown in Fig. 2, the v-th virtual slot, denoted by μ_v, consists of the v-th successful transmission, as well as the preceding idle periods and collision periods. An idle period is a time interval during which all the backlogged stations are performing backoff without transmission attempts. A collision period is the time interval during which two or more stations transmit simultaneously and collide with each other.

Statistically, standard DCF/EDCA operations and their p-persistent models are equivalent when the stations are always active (i.e., when they always have a frame to transmit). However, when a station carries long-lived TCP flows, it may not always have a frame to transmit; for example, during the congestion avoidance phase, such station is allowed to transmit an additional TCP Data only when it receives a TCP Ack for one of its outstanding TCP Data packets. Therefore, when analyzing the TCP dynamics over WLANs, there are slight differences between standard DCF/EDCA operations and their corresponding p-persistent models. Nevertheless, our analysis is based on p-persistent models for simplicity, and the accuracy of the analysis will be evaluated in Section 6.

2.3 Dynamics of long-lived TCP flows

Though there are different versions of TCP protocols, a TCP flow always operates in either slow-start or congestion avoidance phase regardless of the TCP version. The slow-start phase occurs during startup as well as when a timeout occurs at the sender due to packet loss. In this paper, we consider long-lived TCP flows so that we may ignore the effects of slow start since its fraction in the flow lifetime is very small.

In an early version of TCP, i.e., *TCP Tahoe*, the sender resets its congestion window to one Maximum Segment Size (MSS) and enters the slow start phase after each packet loss. This deficiency was remedied in later TCP versions such as *TCP NewReno*, which has been implemented in Microsoft Windows XP. The key difference between TCP NewReno and TCP Tahoe

is the addition of *Fast Retransmit* and *Fast Recovery* mechanisms [12]. Fast Retransmit prevents the timeout by retransmitting a packet when three duplicate ACKs are received at the sender. Fast Recovery cancels the slow start phase by setting the congestion window to approximately half of its current value and keeping the connection in the congestion avoidance phase. Although bursts of packet losses may still cause timeouts at the sender and then force slow start to be invoked, TCP NewReno recovers from slow start much faster than earlier versions of TCP [7].

Because the current 802.11 WLAN has a smaller bandwidth than the wireline network, WLAN is very likely to be the bottleneck in the round trip path of TCP traffic. Therefore, most of the outstanding TCP packets stay in the WLAN for most of their life time. Moreover, in the absence of frequent or bursty packet losses in the wireline network, the TCP send window may grow up to the maximum window size allowed by the receiver, i.e., the maximum TCP receive window size.

3 Analytical model for TCP dynamics over the 802.11e EDCA without TXOP

In this section, we analyze the average number of active stations and the aggregate TCP throughput in an 802.11e EDCA WLAN when the TXOP mechanism is not used. Then, we extend the analysis to take into consideration the TXOP mechanism in Section 4.

3.1 Network model and assumptions

We consider an infrastructure-based 802.11e WLAN with a single AP and N_{STA} stations; each station carries long-lived TCP flows. Stations either upload or download data from remote FTP servers. We conduct our analysis with the following assumptions for simplicity and the effects of some of the assumptions will be studied in Section 6.

A1. The queue sizes of the AP and stations are large enough to avoid buffer overflow.
A2. The successful transmission of each TCP Data packet is notified by an immediate TCP Ack instead of a delayed TCP Ack.
A3. As mentioned in Section 2.3, long-lived TCP flows usually operate in the congestion avoidance phase during most of their lifetime, as long as packet loss does not occur in bursts frequently. In our model, we assume ideal channel conditions (i.e., no hidden terminals, no channel error, and no capture effect). We will study the effects of packet loss on the aggregate TCP throughput in Section 6.5.
A4. In practice, the TCP receive window size is a small number, typically smaller than the TCP congestion window size during the congestion avoidance phase. We also know that the number of outstanding packets is determined by the minimum of the TCP congestion window size and the TCP receive window size. Based on this observation and assumption A3, we assume that the number of outstanding packets (including TCP Data and TCP Ack) for a TCP flow is equal to the maximum TCP receive window size (in packets).[1]
A5. There is no TCP Ack processing delay between a TCP Data reception and the corresponding TCP Ack transmission. We will study the effect of TCP Ack processing delay on the aggregate TCP throughput in Section 6.4.

3.2 State space

Table 1 lists the notations and symbols used in our analysis. We use a discrete-time Markov chain to model the distribution of the numbers of stations carrying different numbers of TCP packets in their queues at the end of the v-th virtual slot in the p-persistent CSMA/CA model.

$\vec{N}(v) = (N^0(v), N^1(v), \cdots, N^W(v))$ is the state vector, where $N^i(v)$ represents the number of stations, each of which carries i TCP packets in its queue at the end of the v-th virtual slot. We refer to such stations as *class-i* stations. Therefore, $N^0(v)$ is the number of stations without TCP packets in their queues. In other words, $N^0(v)$ is the number of inactive stations. W is the maximum TCP receive window size, or equivalently, the maximum number of outstanding TCP packets under assumption A4. For a class-i station, i packets reside in its queue while the remaining $(W - i)$ outstanding packets reside in the AP queue.

Our Markov chain is shown in Fig. 3 with the index of each state shown as the number above it. At *State k*, the following equation always holds:

$$N_{\text{STA}} = \sum_{i=0}^{W} N_k^i, \tag{2}$$

[1] The TCP window sizes are actually maintained in bytes. In our model, for simplicity, we assume that the TCP Data packets have a fixed size and are transmitted at a fixed rate. Hence, the TCP window sizes can be measured in packets. Similar analysis could be done for variable packet sizes and multi-rate WLANs, which is omitted due to space limitation.

Table 1 List of notations and symbols

Term	Definition
W	Maximum TCP receive window size (in packets)
M	Total number of states
N_{STA}	Total number of TCP stations in the WLAN (excluding the AP)
$\overrightarrow{N}_k$	*State k* vector
N_k^i	*i*-th element of the *State k* vector
η_k	Number of active TCP stations in *State k*
σ	A backoff slot time
$P_{m,k}$	State transition probability from *State m* to *State k*
Q_k^{AP}	Length of the AP queue in *State k*
$P_{\text{STA},k}^{\text{succ}}$	Probability that a non-AP station succeeds in a transmission attempt in *State k*
$P_{\text{AP},k}^{\text{succ}}$	Probability that the AP succeeds in a transmission attempt in *State k*
$P_{\text{AP},k}^{i}$	Probability that the AP succeeds in transmitting a packet to a class-*i* station in *State k*.
$P_{\text{STA},k}^{i}$	Probability that a class-*i* station succeeds in transmitting a packet in *State k*

where N_k^i is the i-th element of the *State k* vector, or equivalently, the number of class-i stations at *State k*. Hence, N_k^0 is the number of inactive stations at *State k*. Consequently, the length of the AP queue (Q_k^{AP}) and the number of active TCP stations (η_k) at *State k* can be calculated by

$$\begin{cases} Q_k^{\text{AP}} = \sum_{j=0}^{W} N_k^j \cdot (W - j), \\ \eta_k = N_{\text{STA}} - N_k^0. \end{cases} \tag{3}$$

Let M be the total number of states. The steady-state probability of the Markov chain is

$$\pi_k = \lim_{v \to \infty} \text{P}\left\{ \overrightarrow{N}(v) = \overrightarrow{N}_k \right\}, \text{ for each } k \in [0, M-1]. \tag{4}$$

In this Markov chain, the state transition occurs only at the end of each virtual slot, which ends with a successful transmission. A transition from a right state to a left state in Fig. 3 is caused by a station's successful transmission, and a transition from a left state to a right state is caused by a successful transmission by the AP. For example, at *State 2*, the number of active stations (including the AP) is three. Each of the two non-AP stations has exactly one TCP packet in its queue. If one of them ends the current virtual slot with a successful transmission, the transition from *State 2* to *State 1* occurs. On the other hand, if the AP ends the current virtual slot with its successful transmission, there are two possible cases. The first case is that the AP transmits a TCP packet to one of the $(N_{\text{STA}} - 2)$ inactive stations. In this case, the number of active stations becomes 2 + 1 = 3, hence the transition from *State 2* to *State 4* occurs. The second case is that the AP transmits a TCP packet to one of the two active stations, each of which already has one TCP packet in its queue. As a result, the number of active stations remains the same, while the station that just received the TCP packet from the AP is allowed to add one more packet to its queue. This means that the transition from *State 2* to *State 5* has occurred.

3.3 State transition probabilities

We now derive the state transition probabilities. As discussed above, during each transition, only two elements of the state vector change their values and each of them changes by one. This is because state transitions are triggered by a single successful transmission from either the AP or a station. Therefore, the transition probability from *State k* to *State m* is given by

$$P_{k,m} = \begin{cases} P_{\text{AP},k}^{i}, & \text{if } \exists \text{ a unique } i \in [0, W-1] \text{ such that} \\ & N_m^i = N_k^i - 1,\ N_m^{i+1} = N_k^{i+1} + 1, \\ P_{\text{STA},k}^{i}, & \text{if } \exists \text{ a unique } i \in [0, W-1] \text{ such that} \\ & N_m^i = N_k^i + 1,\ N_m^{i+1} = N_k^{i+1} - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

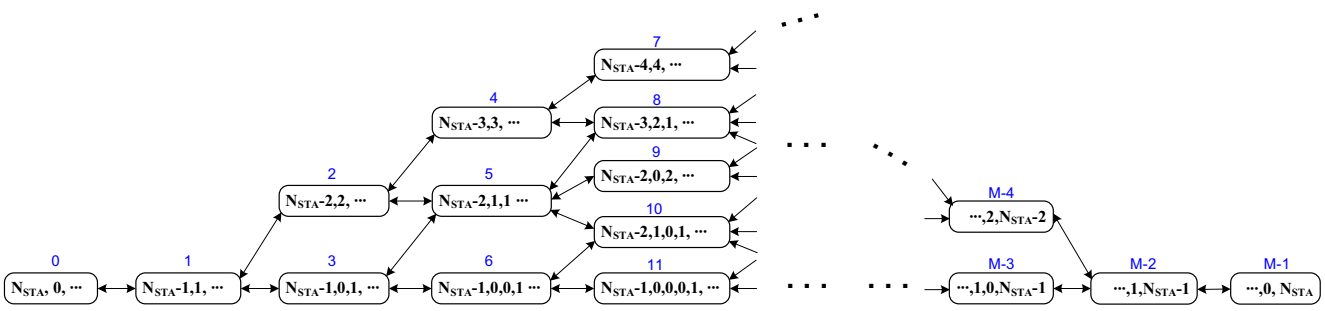


Figure 3 Our markov chain model to study TCP dynamics in WLANs

The first equation in Eq. 5 accounts for the fact that the AP ends the current virtual slot with its successful transmission. $P^i_{\mathrm{AP},k}$ is the probability that the AP successfully transmits its TCP packet to one of the class-i stations. The second equation accounts for the fact that a station ends the current virtual slot with its successful transmission. $P^i_{\mathrm{STA},k}$ is the probability that a class-$(i+1)$ station succeeds in transmitting its TCP packet.

We assume that the AP and a station access the medium with attempt probabilities τ_{AP} and τ_{STA}, respectively, whose values will be derived in Section 3.6. Then, the probability $P^i_{\mathrm{AP},k}$ is given by

$$P^i_{\mathrm{AP},k} = P^{\mathrm{succ}}_{\mathrm{AP},k} \cdot P^{\mathrm{AP}(i)}_k, \tag{6}$$

where

$$\begin{aligned} P^{\mathrm{succ}}_{\mathrm{AP},k} &= P\left(\text{AP's success} \,\middle|\, \text{a tx success at } \textit{State } k\right) \\ &= \begin{cases} \frac{\tau_{\mathrm{AP}}(1-\tau_{\mathrm{STA}})^{\eta_k}}{\tau_{\mathrm{AP}}(1-\tau_{\mathrm{STA}})^{\eta_k}+\eta_k\tau_{\mathrm{STA}}(1-\tau_{\mathrm{STA}})^{\eta_k-1}(1-\tau_{\mathrm{AP}})}, \\ \qquad\qquad N^W_k < N_{\mathrm{STA}}, \\ 0, \qquad\qquad N^W_k = N_{\mathrm{STA}}, \end{cases} \end{aligned} \tag{7}$$

and

$$\begin{aligned} P^{\mathrm{AP}(i)}_k &= P\left(\text{AP xmit to a class-}i\text{ STA} \,\middle|\, \text{AP's success}\right) \\ &= \begin{cases} \frac{N^i_k(W-i)}{Q^{\mathrm{AP}}_k}, & Q^{\mathrm{AP}}_k > 0, \\ 0, & Q^{\mathrm{AP}}_k = 0. \end{cases} \end{aligned} \tag{8}$$

Similarly, the probability $P^i_{\mathrm{STA},k}$ is given by

$$P^i_{\mathrm{STA},k} = P^{\mathrm{succ}}_{\mathrm{STA},k} P^{\mathrm{STA}(i+1)}_k, \tag{9}$$

where

$$\begin{aligned} P^{\mathrm{succ}}_{\mathrm{STA},k} &= P\left(\text{STA's success} \,\middle|\, \text{a tx success at } \textit{State } k\right) \\ &= 1 - P^{\mathrm{succ}}_{\mathrm{AP},k}, \end{aligned}$$

and

$$\begin{aligned} P^{\mathrm{STA}(i+1)}_k &= P\left(\text{it is a class-}(i+1)\text{ STA} \,\middle|\, \text{STA's success}\right) \\ &= \begin{cases} \frac{N^{i+1}_k}{N_{\mathrm{STA}}-N^0_k}, & N^0_k < N_{\mathrm{STA}}, \\ 0, & N^0_k = N_{\mathrm{STA}}. \end{cases} \end{aligned} \tag{10}$$

3.4 Average number of active stations

Since the length of the virtual slot (i.e., the sojourn time) in each state varies, the average number of active stations (excluding the AP) is

$$E[N_{\mathrm{active}}] = \sum_{k=0}^{M-1} \eta_k p_k, \tag{11}$$

where

$$p_k = \frac{\pi_k \mu_k}{\sum_{j=0}^{M-1} \pi_j \mu_j}, \tag{12}$$

where μ_k is the sojourn time at *State k*. In the case of download TCP flows, μ_k is given by

$$\begin{aligned} \mu_k = {} & E\left[N^{\mathrm{col}}_{\mathrm{STA},k}\right] T^{\mathrm{ack}}_{\mathrm{col}} + E\left[N^{\mathrm{col}}_{\mathrm{AP},k}\right] T^{\mathrm{data}}_{\mathrm{col}} \\ & + \left(E\left[N^{\mathrm{col}}_k\right] + 1\right)\left(\mathrm{AIFS} + E\left[T^{\mathrm{idle}}_k\right]\right) \\ & + E\left[T^{\mathrm{data}}_k\right] + \mathrm{SIFS} + T_{\mathrm{ACK}}. \end{aligned} \tag{13}$$

Here, $T^{\mathrm{ack}}_{\mathrm{col}}$ and $E\left[N^{\mathrm{col}}_{\mathrm{STA},k}\right]$ are, respectively, the collision time and the average number of collisions due to contention among uplink TCP Acks. $T^{\mathrm{data}}_{\mathrm{col}}$ and $E\left[N^{\mathrm{col}}_{\mathrm{AP},k}\right]$ are, respectively, the collision time and the average number of collisions due to contention between a downlink TCP Data and uplink TCP Acks. Moreover, $E\left[T^{\mathrm{data}}_k\right]$ and T_{ACK} are the average TCP packet transmission time and the MAC-layer ACK frame transmission time, respectively. $E\left[N^{\mathrm{col}}_k\right]$ and $E\left[T^{\mathrm{idle}}_k\right]$ are the average number of collisions and the average idle time preceding a collision or a successful transmission, respectively. On the other hand, in the case of upload TCP, μ_k can be calculated by replacing $T^{\mathrm{ack}}_{\mathrm{col}}$ with $T^{\mathrm{data}}_{\mathrm{col}}$ in Eq. 13, because at least one of the colliding packets is a TCP Data, and hence the collision time is always $T^{\mathrm{data}}_{\mathrm{col}}$.

In the following, we derive individual elements used in Eq. 13. $E\left[N^{\mathrm{col}}_k\right]$, $E\left[N^{\mathrm{col}}_{\mathrm{AP},k}\right]$ and $E\left[N^{\mathrm{col}}_{\mathrm{STA},k}\right]$ can be expressed by

$$\begin{cases} E\left[N^{\mathrm{col}}_k\right] &= \sum_{i=0}^{\infty} i \cdot P\left(N^{\mathrm{col}}_k = i\right), \\ E\left[N^{\mathrm{col}}_{\mathrm{AP},k}\right] &= \sum_{i=0}^{\infty} j \cdot P\left(N^{\mathrm{col}}_{\mathrm{AP},k} = j\right), \\ E\left[N^{\mathrm{col}}_{\mathrm{STA},k}\right] &= E\left[N^{\mathrm{col}}_k\right] - E\left[N^{\mathrm{col}}_{\mathrm{AP},k}\right], \end{cases} \tag{14}$$

where $P\left(N^{\mathrm{col}}_k = i\right)$ and $P\left(N^{\mathrm{col}}_{\mathrm{AP},k} = j\right)$ are the probability that the total number of collisions is i and the probability that the number of collisions due to the AP's transmissions is j, respectively, which are given by

$$\begin{cases} P\left(N^{\mathrm{col}}_k = i\right) &= \left(P^{\mathrm{col}}_k\right)^i P^{\mathrm{succ}}_k, \\ P\left(N^{\mathrm{col}}_{\mathrm{AP},k} = j\right) &= \left(P^{\mathrm{col}}_{\mathrm{AP},k}\right)^j P^{\mathrm{succ}}_k. \end{cases} \tag{15}$$

P^{succ}_k is the conditional successful transmission probability at *State k* given that at least one station (including the AP) transmits, which can be calculated by

$$\begin{aligned} P^{\mathrm{succ}}_k &= P\left(N^{\mathrm{tx}}_k = 1 \,\middle|\, N^{\mathrm{tx}}_k \geqslant 1\right) \\ &= \begin{cases} \frac{\tau_{\mathrm{AP}}(1-\tau_{\mathrm{STA}})^{\eta_k}+\eta_k\tau_{\mathrm{STA}}(1-\tau_{\mathrm{STA}})^{\eta_k-1}(1-\tau_{\mathrm{AP}})}{1-(1-\tau_{\mathrm{AP}})(1-\tau_{\mathrm{STA}})^{\eta_k}}, \\ \qquad\qquad N^W_k < N_{\mathrm{STA}}, \\ \frac{\eta_k \cdot \tau_{\mathrm{STA}}(1-\tau_{\mathrm{STA}})^{\eta_k-1}}{1-(1-\tau_{\mathrm{STA}})^{\eta_k}}, \qquad N^W_k = N_{\mathrm{STA}}. \end{cases} \end{aligned} \tag{16}$$

Here, N_k^{tx} denotes the number of transmitting stations at *State k*. P_k^{col} is the conditional collision probability given that at least one station transmits, which is $P_k^{\text{col}} = 1 - P_k^{\text{succ}}$. Moreover, $P_{\text{AP},k}^{\text{col}}$ is the conditional probability that AP's transmission fails given that at least one station (including the AP) transmits, which is

$$P_{\text{AP},k}^{\text{col}} = \begin{cases} \frac{\tau_{\text{AP}}(1-(1-\tau_{\text{STA}})^{\eta_k})}{1-(1-\tau_{\text{AP}})(1-\tau_{\text{STA}})^{\eta_k}}, & N_k^W < N_{\text{STA}}, \\ 0, & N_k^W = N_{\text{STA}}. \end{cases} \tag{17}$$

The average idle time preceding a collision or a successful transmission in *State k* can be calculated as

$$\begin{aligned} E\left[T_k^{\text{idle}}\right] &= \sigma \sum_{i=0}^{\infty} iP\left(N_k^{\text{tx}} > 0\right)\left(P\left(N_k^{\text{tx}} = 0\right)\right)^i \\ &= \sigma \frac{(1-\tau_{\text{AP}})(1-\tau_{\text{STA}})^{\eta_k}}{1-(1-\tau_{\text{AP}})(1-\tau_{\text{STA}})^{\eta_k}}. \end{aligned} \tag{18}$$

In the case of download TCP, the average TCP packet transmission time can be expressed as

$$E\left[T_k^{\text{data}}\right] = T_{\text{TCP DATA}} P_{\text{AP},k}^{\text{succ}} + T_{\text{TCP ACK}} P_{\text{STA},k}^{\text{succ}}, \tag{19}$$

where $T_{\text{TCP DATA}}$ and $T_{\text{TCP ACK}}$ are TCP Data and TCP Ack transmission durations, respectively. In the case of upload TCP, the positions of $P_{\text{AP},k}^{\text{succ}}$ and $P_{\text{STA},k}^{\text{succ}}$ are exchanged in Eq. 19.

3.5 Aggregate TCP throughput

In the *p*-persistent CSMA/CA model [5], the aggregate TCP throughput (in bits/second) is given by

$$\rho = \frac{\text{average length of a TCP Data (bits)}}{\text{average length of a virtual slot (seconds)}}. \tag{20}$$

In our analysis, with the assumption that there exist a fixed number of active stations in a given state, the download TCP throughput at *State k* can be derived as

$$\rho_k = \frac{L_{\text{TCP DATA}} P_{\text{AP},k}^{\text{succ}}}{\mu_k}, \tag{21}$$

where $L_{\text{TCP DATA}}$ is the TCP Data size and μ_k is the sojourn time at *State k*. Finally, the average aggregate TCP throughput can be calculated by

$$\rho_{\text{TCP}} = \sum_{i=0}^{M-1} \rho_i p_i. \tag{22}$$

For the upload TCP throughput, $P_{\text{AP},k}^{\text{succ}}$ is replaced by $P_{\text{STA},k}^{\text{succ}}$ in Eq. 21. For the mixed case of upload and download, we can calculate the TCP throughput by modifying Eqs. 13 and 21, and details are omitted due to space limitation.

3.6 Estimation of τ from CWmin at each state

In order to obtain the medium access probabilities τ_{AP} and τ_{STA}, we extend the Average Contention Window Estimation algorithm in [5] by considering different CWmin sizes for the AP and stations. We assume that $\tau = 1/(\overline{B}+1)$ where $\overline{B}$ is the average number of backoff slots, and is equal to $(\overline{\text{CW}}+1)/2$ [5]. Because the average contention window size $\overline{\text{CW}}$ is determined by CWmin and the number of active stations in a given state, we can estimate τ from CWmin in each state using an iterative algorithm. The algorithm used in this work is based on and improves upon the one we proposed in [16] by considering different CWmin sizes for the AP and stations.

4 Analytical model for TCP dynamics over the 802.11e EDCA with TXOP

In this section, we extend the analysis in Section 3 to consider the TXOP mechanism. For simplicity, we represent TXOPLimit as the maximum number of data packets which the AP or a station can transmit without additional backoff when it grabs the channel. Let ξ_{AP} and ξ_{STA} denote TXOPLimits (in packets) of the AP and a station, respectively.

The number of active stations and the aggregate TCP throughput depend on the distribution of TCP packets in the AP queue. That is, when the packets directed to different stations are uniformly distributed in the AP queue, a TXOP burst from the AP may activate many inactive stations. On the other hand, when the packets directed to the same station are grouped in the AP queue, a TXOP burst from the AP may activate less number of inactive stations. In our analysis, we assume that the packets in the AP queue are uniformly distributed. This is reasonable since all the TCP sessions start from the slow start phase and each station randomly accesses the channel.

4.1 State transition probabilities

With the TXOP mechanism, though the state space of the Markov chain remains the same, the state transition patterns are different and the transition probabilities need to be re-derived. The probability that the transition from *State k* to *State m* occurs can be derived as follows.

4.1.1 Transitions caused by TXOP of the AP ($k < m$)

With TXOP, the AP may transmit multiple TCP packets after a contention success. Hence, multi-hop state transitions may occur in the Markov chain. Recall that only single-hop transitions between linked states are possible when TXOP is not used. Moreover, with TXOP, the state transition probability from *State k* to *State m* is equal to the sum of the probabilities of all possible paths (in the increasing order of the state index along each path) from *State k* to *State m* in Fig. 3. The possible paths are determined by the order of packet transmissions in a TXOP. For example, there are two possible paths from *State 1* to *State 5* in Fig. 3, which are *1→2→5* and *1→3→5*. The probability that the state transition occurs following a particular path from *State k* to *State m* is the product of the transition probabilities between each pair of neighboring states along the path. Let *State i* and *State j* ($i < j$) be a pair of neighboring states along a path from *State k* to *State m*. Then from Eqs. 5 and 6, the transition probability from *State i* to *State j* given that the AP succeeds in accessing the channel at *State i* is

$$P'_{i,j} = \begin{cases} \frac{P_{i,j}}{P^{\text{succ}}_{\text{AP},i}}, & i < j, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Using Eq. 23, we construct an $M \times M$ matrix $\mathbb{A}$, where the element corresponding to row i and column j is $P'_{i,j}$. The diagonal and below-diagonal elements of $\mathbb{A}$ are zeros according to Eq. 23. Each above-diagonal element of $\mathbb{A}$ represents the transition probability from the corresponding row state to column state, assuming that the AP succeeds in accessing the channel at the row state; therefore, if the row and column states are not neighboring states in Fig. 3, the corresponding element of $\mathbb{A}$ is zero. Accordingly, if $Q^{\text{AP}}_m - Q^{\text{AP}}_k = n$, i.e., if the state transition from *State k* to *State m* is caused by n TCP packet transmissions by the AP, the probability that the transition from *State k* to *State m* occurs given that the AP wins a TXOP at *State k* is the element of the matrix $\mathbb{A}^n$ that corresponds to row k and column m. More generally, when the AP succeeds in obtaining a TXOP at *State k*, the probability matrix that represents all the transition probabilities from *State k* to other states can be expressed as

$$\mathbb{X}_k = \begin{cases} \mathbb{A}^{\xi_{\text{AP}}}, & Q^{\text{AP}}_k \geqslant \xi_{\text{AP}}, \\ \mathbb{A}^{Q^{\text{AP}}_k}, & 0 < Q^{\text{AP}}_k < \xi_{\text{AP}}, \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

where ξ_{AP} is the TXOPLimit (in packets) for the AP. Finally, we can derive the transition probabilities from *State k* to *State m* ($k < m$) as

$$P_{k,m} = P^{\text{succ}}_{\text{AP},k} X_k(k,m), \tag{25}$$

where $X_k(k,m)$ is the element of the matrix $\mathbb{X}_k$ corresponding to row k and column m, and is the conditional transition probability from *State k* to *State m* given that the AP succeeds in obtaining a TXOP at *State k*.

4.1.2 Transitions caused by TXOP of a station ($k > m$)

In the case of state transition caused by a station's TXOP, there exists only one possible path from *State k* to *State m* ($k > m$) because a station only transmits its packets to the AP. This is different from the multiple possible paths in the case of state transition caused by the AP's TXOP.

If there exists an $i \in [1, W]$ such that $N^i_m = N^i_k - 1$, the transition probability from *State k* to *State m*, assuming that a station succeeds in getting a TXOP at *State k*, can be expressed as

$$P''_{k,m} = \begin{cases} \frac{N^i_k}{N_{\text{STA}} - N^0_k}, & i > \xi_{\text{STA}}, \\ & N^{i-\xi_{\text{STA}}}_m = N^{i-\xi_{\text{STA}}}_k + 1, \\ & N^l_m = N^l_k,\ l \neq i,\ l \neq i - \xi_{\text{STA}}, \\ \frac{N^i_k}{N_{\text{STA}} - N^0_k}, & i \leq \xi_{\text{STA}}, \\ & N^0_m = N^0_k + 1, \\ & N^l_m = N^l_k,\ l \neq i,\ l \neq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{26}$$

where $l \in [0, W]$. Otherwise, $P''_{k,m}$ is always zero. Finally, the transition probability from *State k* to *State m* ($k > m$) is given by

$$P_{k,m} = P^{\text{succ}}_{\text{STA},k} P''_{k,m}. \tag{27}$$

4.2 Average number of active stations and aggregate TCP throughput

The sojourn time at *State k* in the case of download TCP flows is given by

$$\begin{aligned} \mu_k = {} & E\left[N^{\text{col}}_{\text{STA},k}\right] T^{\text{ack}}_{\text{col}} + E\left[N^{\text{col}}_{\text{AP},k}\right] T^{\text{data}}_{\text{col}} \\ & + \left(E\left[N^{\text{col}}_k\right] + 1\right) \left(\text{AIFS} + E\left[T^{\text{idle}}_k\right]\right) \\ & + E\left[T^{\text{TXOP}}_k\right], \end{aligned} \tag{28}$$

where $E\left[N^{\text{col}}_k\right]$, $E\left[N^{\text{col}}_{\text{AP},k}\right]$, $E\left[N^{\text{col}}_{\text{STA},k}\right]$, and $E\left[T^{\text{idle}}_k\right]$ can be calculated using Eqs. 14 and 18 because

using TXOP does not affect the channel contention. $E\left[T_k^{\text{TXOP}}\right]$ is the average TXOP duration used in *State k*:

$$E\left[T_k^{\text{TXOP}}\right] = E\left[T_{\text{AP},k}^{\text{TXOP}}\right] P_{\text{AP},k}^{\text{succ}} + E\left[T_{\text{STA},k}^{\text{TXOP}}\right] P_{\text{STA},k}^{\text{succ}}, \tag{29}$$

where $E\left[T_{\text{AP},k}^{\text{TXOP}}\right]$ and $E\left[T_{\text{STA},k}^{\text{TXOP}}\right]$ are the average TXOP durations which the AP and one of the active stations use for multiple TCP packet transmissions at *State k*, respectively, and are given by

$$E\left[T_{\text{AP},k}^{\text{TXOP}}\right] = \begin{cases} \min\left(Q_k^{\text{AP}},\ \xi_{\text{AP}}\right) \\ \cdot\left(T_{\text{TCP DATA}} + 2\text{SIFS} + T_{\text{ACK}}\right) \\ -\text{SIFS}, & Q_k^{\text{AP}} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{30}$$

$$E\left[T_{\text{STA},k}^{\text{TXOP}}\right] = \begin{cases} \left(\sum_{i=1}^{W} P_k^{\text{STA}(i)} \min\left(i,\ \xi_{\text{STA}}\right)\right) \\ \cdot\left(T_{\text{TCP ACK}} + 2\text{SIFS} + T_{\text{ACK}}\right) \\ -\text{SIFS}, & \eta_k > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{31}$$

where $P_k^{\text{STA}(i)}$ is the conditional probability that a class-i station succeeds in accessing the channel given that a station wins the channel access at *State k* as in Eq. 10. On the other hand, in the case of upload TCP, μ_k can be calculated by replacing $T_{\text{col}}^{\text{ack}}$ with $T_{\text{col}}^{\text{data}}$ in Eq. 28, $T_{\text{TCP DATA}}$ with $T_{\text{TCP ACK}}$ in Eq. 30, and $T_{\text{TCP ACK}}$ with $T_{\text{TCP DATA}}$ in Eq. (31).

Using Eqs. 11, 25, 27, and 28, we can calculate the average number of the active stations (excluding the AP). Moreover, the download TCP throughput at *State k* can be derived as

$$\rho_k = \frac{\min(Q_k^{\text{AP}}, \xi_{\text{AP}})\, L_{\text{TCP DATA}}\, P_{\text{AP},k}^{\text{succ}}}{\mu_k}. \tag{32}$$

In the case of upload TCP, the TCP throughput at *State k* is given by

$$\rho_k = \frac{\left(\sum_{i=1}^{W} P_k^{\text{STA}(i)} \min\left(i,\ \xi_{\text{STA}}\right)\right) L_{\text{TCP DATA}}\, P_{\text{STA},k}^{\text{succ}}}{\mu_k}. \tag{33}$$

Finally, the average aggregate TCP throughput can be calculated using Eq. 22.

5 AP PIFS access

5.1 AP PIFS access without TXOP

As described in Section 2.1, AIFSN[AC] is an integer greater than 1 for stations and an integer greater than 0 for the AP. Moreover, the values of CWmin[AC] and CWmax[AC] may be set to zero. Therefore, for the AP, we can use the smallest access parameter values of AIFSN[AC] = 1, CWmin[AC] = 0, and CWmax [AC] = 0 for downlink TCP packet transmissions. This allows the AP to transmit the pending TCP packets after a PIFS idle time without backoff. This scheme is referred to as *PIFS Access* in the rest of this paper. In [15], we have shown that, PIFS Access by the AP can improve the VoWLAN (Voice over WLAN) performance significantly.

When the AP does PIFS Access for TCP packet transmissions, all the outstanding TCP packets reside in the queues of stations. As a result, the number of active stations is the same as the number of stations in the WLAN, and hence the behavior of stations is identical to that of saturated UDP stations. The AP sends a TCP Data (or TCP Ack) at PIFS time after it receives a TCP Ack (or TCP Data) from one of the stations. That is, in Fig. 3, the state transitions only occur between *State (M-1)* and *State (M-2)*. When there are only download TCP flows, the aggregate TCP throughput with AP PIFS Access can be calculated by

$$\rho_{\text{PIFS}} = \frac{L_{\text{TCP DATA}}}{\mu_{M-1} + \text{PIFS} + T_{\text{TCP DATA}} + \text{SIFS} + T_{\text{ACK}}}, \tag{34}$$

where μ_{M-1} is the sojourn time at *State (M-1)* in Fig. 3. In the case of upload TCP, $T_{\text{TCP DATA}}$ in the denominator of Eq. 34 is replaced by $T_{\text{TCP ACK}}$.

5.2 AP PIFS access with TXOP

Employment of the TXOP mechanism does not affect the contention behaviors of the AP and stations. That is, the AP is assigned TXOP(s) to transmit the pending TCP packets after a PIFS idle time, and all the outstanding TCP packets reside in the queues of stations. Recall that ξ_{AP} and ξ_{STA} are the TXOPLimits (in packets) of the AP and stations, respectively. Therefore, when there are only download TCP flows, we can calculate the aggregate TCP throughput with AP PIFS Access and TXOP by revising Eq. 34 as follows:

$$\rho_{\text{PIFS}} = \frac{\xi_{\text{STA}}\, L_{\text{TCP DATA}}}{\mu_{M-1} + T_{\text{PIFS}}}, \tag{35}$$

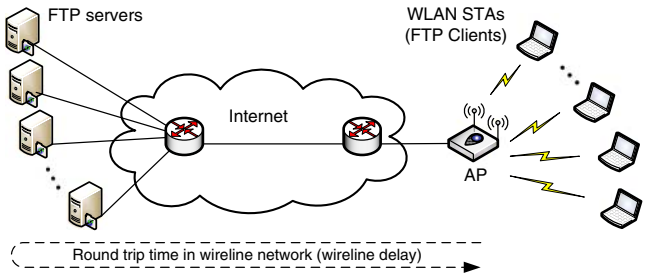


Figure 4 The simulated network topology

where T_{PIFS} is the time used to transmit TCP Data packets by the AP after receiving TCP Ack packets form a station, which is given by

$$T_{\text{PIFS}} = \text{PIFS}\left\lceil \frac{\xi_{\text{STA}}}{\xi_{\text{AP}}} \right\rceil - \text{SIFS}\left\lceil \frac{\xi_{\text{STA}}}{\xi_{\text{AP}}} \right\rceil + \xi_{\text{STA}}\left(T_{\text{TCP DATA}} + 2\text{SIFS} + T_{\text{ACK}}\right). \tag{36}$$

6 Simulation studies

In this section, we validate our model via ns-2 simulation [1]. We consider an IEEE 802.11b WLAN with a single AP and seven stations that either download or upload large data files to remote FTP servers. Figure 4 shows the network topology. As shown in the figure, the Round Trip Time (RTT) of TCP packets in the wireline network is referred to as the *wireline delay*. TCP NewReno is used for our simulation. Unless stated otherwise, we assume that the maximum TCP receive window size is $W = 4$ packets.[2] Parameters used to produce analytical and simulation results are summarized in Table 2. The considered values for CWmin are $\{2^n - 1 \mid 2 \leqslant n \leqslant 8\}$.

We first study how the TCP throughput performance is affected by the CWmin and TXOPLimit values, assuming no TCP Ack processing delay,[3] no packet loss, and no wireline delay. Then, we show the effects of these parameters on the aggregate TCP throughput in Sections 6.4 and 6.5.

[2]Actually, typical TCP configurations of commercial operating systems use a larger receive window size, e.g., the 12-packet (or 17520-byte) window used in MS Windows XP. We basically consider the 4-packet receive window size in order to reduce the calculation overhead; similar trends can be observed for other receive window sizes.

[3]In this paper, TCP Ack processing delay is defined as time duration between a TCP Data reception and the corresponding TCP Ack arrival at the MAC Service Access Point (SAP).

Table 2 Parameters used to produce analysis and simulation results

Parameters	Values
SlotTime	20 μs
SIFS, PIFS, AIFS	10 μs, 30 μs, 50 μs
CWmax	1023
TXOPLimit	Variable
Data, ACK Rates	11 Mbps, 2 Mbps
PHY Overhead	192 μs
ACK length	14 bytes
MAC Overhead	30 bytes
TCP data frame	Data (1460) + TCP/IP headers (40) + SNAP header (8)* + MAC/PHY overheads = 1538 bytes
TCP Ack frame	TCP/IP headers (40) + SNAP header (8) + MAC/PHY overheads = 78 bytes

*When an IP datagram is transferred over 802.11 WLANs, it is usually encapsulated in an IEEE 802.2 Sub-Network Access Protocol (SNAP) packet.

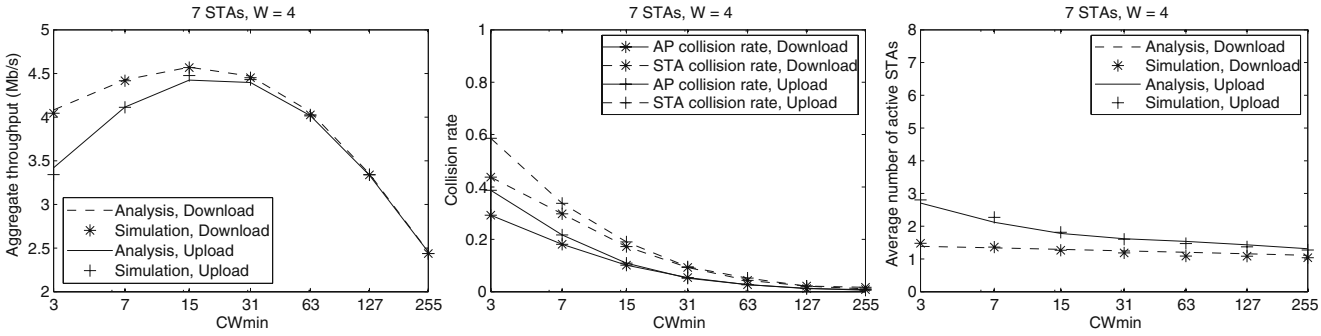
6.1 Effects of CWmin sizes

In this section, we study the effects of CWmin sizes of the AP and stations when the TXOP mechanism is not used.

6.1.1 When the AP and stations use the same access parameters (including CWmin)

Figure 5 plots (a) the aggregate TCP throughput, (b) the collision rates experienced by the AP and a station, and (c) the average number of active stations (excluding the AP), for both download and upload cases. We observe that analytical and simulation results match very well for all simulated scenarios. In Fig. 5a, in the case of download, the aggregate TCP throughput of the 802.11 DCF (similar to EDCA with CWmin = 31) is about 4.46 Mbps while the maximum download throughput of about 4.56 Mbps is achieved when CWmin = 15. From this result, one may conclude that TCP over DCF is operating near the optimal throughput point. Although this is true when the AP and stations are required to use the same access parameters, we will show in the subsequent subsection that the aggregate TCP throughput can be enhanced further by using different access parameters for the AP and stations.

The collision rate of a station is higher than that of the AP when CWmin is small. This is due to the fact that an active station contends with at least one station, i.e., the AP, because the AP is always active in this scenario. Moreover, note that a station transmits long TCP Data and short TCP Ack for upload and download cases, respectively. Therefore, as CWmin decreases, the



(a) Aggregate throughput (b) Collision rate (c) Average number of active stations (excluding AP)

Figure 5 TCP performance when the AP and stations use the same access parameters (including CWmin)

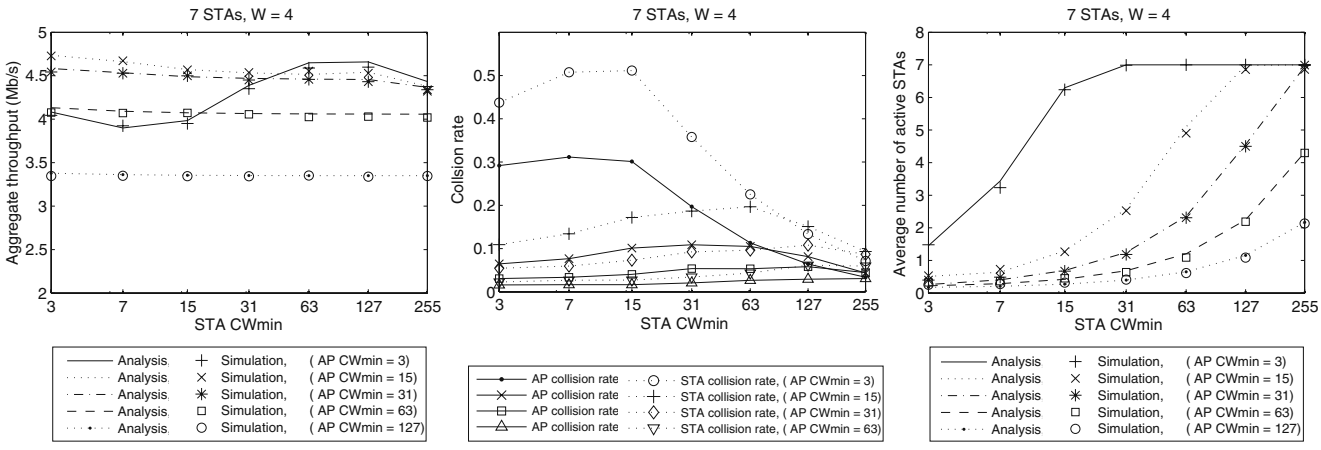
throughput in the upload case decreases more quickly than the download case because the wasted time due to collisions is more severe.

As shown in Fig. 5c, the average number of active stations remains small regardless of the CWmin size. This is due to (i) the closed-loop nature of TCP flow control and (ii) the bottleneck downlink (i.e., AP-to-station transmissions) in the infrastructure-based WLAN, as we have discussed in Section 1. Under such scenario, the AP is the bottleneck because it uses the same CWmin as stations but needs to serve multiple TCP flows.

6.1.2 When the AP and stations use different access parameters

We study the aggregate TCP throughput performance with various combinations of the CWmin sizes used by the AP and stations. Figure 6 plots (a) the aggregate TCP throughput, (b) the collision rates experienced by the AP and a station, and (c) the average number of active stations (excluding the AP), for the download case. Similar results have also been observed for the upload case, which are omitted due to space limitation.

We have two sets of observations. First, when $\text{CWmin}_{\text{AP}} \geqslant \text{CWmin}_{\text{STA}}$, the AP's downlink becomes a more severe bottleneck. The network is in a non-saturated state. As shown in Fig. 6c, the average number of active stations (excluding the AP) is less than or equal to one. As the gap between CWmin_{AP} and $\text{CWmin}_{\text{STA}}$ increases, the average number of active stations decreases. This is due to the fact that most stations spend more time idle, waiting for a TCP Data from the AP. On the other hand, it is interesting to see that the aggregate throughput does not increase monotonically as the gap decreases. For example, the combination



(a) Aggregate throughput (b) Collision rate (c) Average number of active stations (excluding AP)

Figure 6 Download TCP performance when the AP and stations use different CWmin

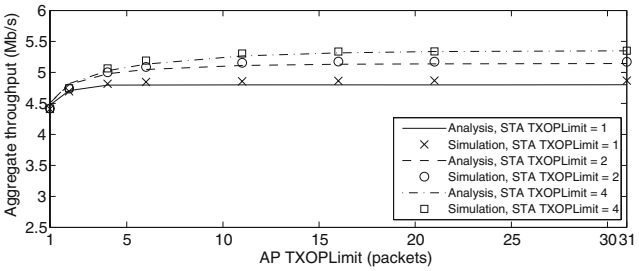
of $\text{CWmin}_{\text{AP}} = 31$ and $\text{CWmin}_{\text{STA}} = 7$ yields a higher throughput than the combination of $\text{CWmin}_{\text{AP}} = 3$ and $\text{CWmin}_{\text{STA}} = 3$. This is because, the collision rates (especially, the AP collision rate) increase as the gap decreases, which may offset the effects of smaller number of active stations under certain situations.

Secondly, when $\text{CWmin}_{\text{AP}} < \text{CWmin}_{\text{STA}}$, packet congestion at the AP bottleneck is alleviated. As the gap between CWmin_{AP} and $\text{CWmin}_{\text{STA}}$ increases, the average number of active stations increases, and is eventually saturated to 7 (i.e., all the stations are actively contending). On the other hand, as shown in Fig. 6b, the collision rates initially increase and then decrease eventually. The reason is as follows. For a given CWmin_{AP}, the number of active stations keeps increasing until $\text{CWmin}_{\text{STA}}$ reaches a certain threshold; when $\text{CWmin}_{\text{STA}}$ becomes larger than this threshold, the number of active stations is saturated to 7 but the idle backoff time continues to increase. For example, when CWmin_{AP} is 3, such threshold for $\text{CWmin}_{\text{STA}}$ is 31, which is shown in Fig 6c. In general, this threshold for $\text{CWmin}_{\text{STA}}$ varies with CWmin_{AP}, and a larger CWmin_{AP} corresponds to a larger threshold. Accordingly, we can clearly observe from the figure that, for small CWmin_{AP} values, the aggregate throughput first decreases (due to the increasing collision rate), then increases (due to the decreasing collision rate), and eventually decreases (due to the increasing idle backoff time). The reason that we do not observe similar behaviors in the figure for larger CWmin_{AP} values is because the corresponding thresholds for $\text{CWmin}_{\text{STA}}$ are very large, beyond the x-axis range of the figure.

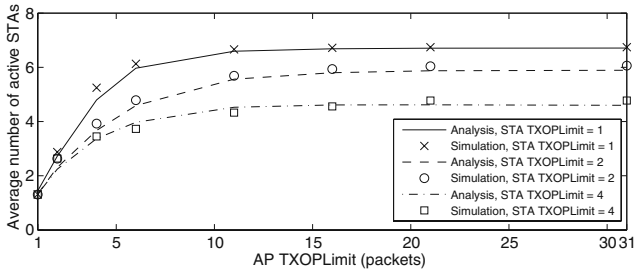
Comparing Fig. 6a with Fig. 5a, we can see that the maximum aggregate throughput in Fig. 6a is larger than that in Fig. 5a. It means that there exists at least one optimal combination of CWmin_{AP} and $\text{CWmin}_{\text{STA}}$ to maximize the aggregate throughput. This fact is not surprising since a similar fact has been well known for the saturation throughput maximization in 802.11 DCF WLANs. According to [2, 5], there exists an optimal CWmin value that maximizes the saturation throughput for a given number of stations. Unfortunately, it is very difficult, if not impossible, to derive a closed-form expression for the optimal CWmin combination in our study, because the CWmin values and the number of active stations are mutually dependent in the case of TCP.

6.2 Effects of TXOPLimit sizes

Figure 7 shows the aggregate download TCP throughput and the average number of active stations (excluding the AP) when the TXOP mechanism is used for both the AP and stations. CWmin is set to 31 for both the AP and stations. Note that setting the TXOPLimit to one packet is equivalent to not using the TXOP mechanism. As shown in the figure, when the TXOPLimits of both the AP and stations are large, the aggregate throughput can be improved further. This is because the TXOP mechanism allows multiple packet transmissions without additional contention overhead (i.e., backoff or wasted time due to collisions). When the AP TXOPLimit is small, different station TXOPLimit sizes do not make much difference in the aggregate throughput. This is because the AP is bottleneck, and hence the number of active stations is small as shown in Fig. 7b and active stations do not have many packets to transmit even though their TXOPLimits could be large. On the other hand, when the AP TXOPLimit is much larger than the station TXOPLimit, a larger station TXOPLimit yields a higher throughput, because most of the packets reside in the queues of stations and a larger station TXOPLimit allows more packets to be transmitted during a TXOP. Similar trends have also been observed with other CWmin values for the AP and stations, and simulation results are omitted due to space limitation.



(a) Aggregate TCP throughput

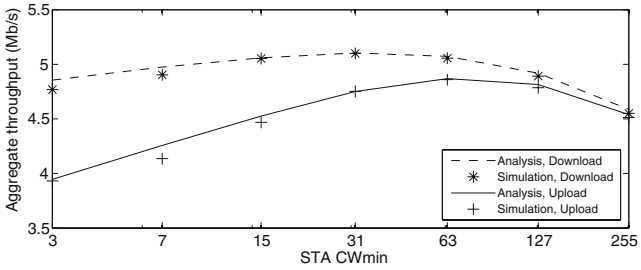


(b) Average number of active stations (excluding AP)

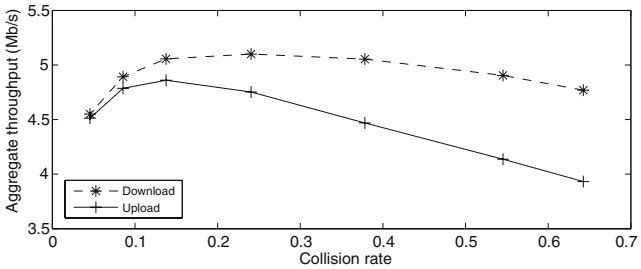
Figure 7 Download TCP performance when the TXOP mechanism is used. CWmin is set to 31 for both the AP and stations

6.3 Effects of AP PIFS access

Figure 8 plots the aggregate TCP throughput and the collision rate when the AP uses PIFS Access for its

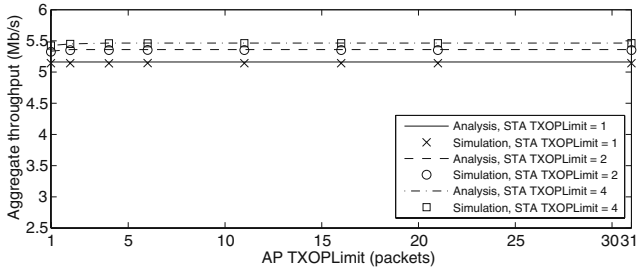


(a) Aggregate TCP throughput

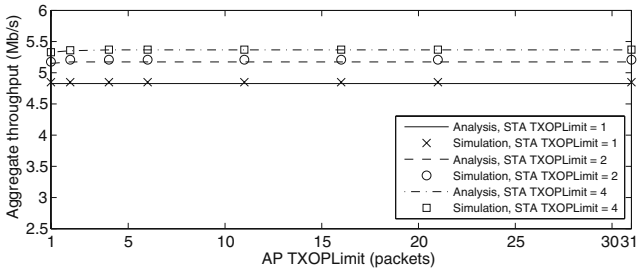


(b) Collision rate vs. aggregate throughput

Figure 8 TCP performance when the AP uses PIFS Access but the TXOP mechanism is not used



(a) Aggregate download TCP throughput



(b) Aggregate upload TCP throughput

Figure 9 TCP performance when the AP uses PIFS Access, stations use CWmin = 31, and the AP and stations use different TXOPLimits

downlink transmissions but the TXOP mechanism is not used. In the case of download, we observe that the maximum aggregate TCP throughput of PIFS Access is much higher than the previous two cases in Section 6.1. This is because, with PIFS Access, there is no collision between the AP and stations, and hence less time is wasted due to collisions. For example, when only download TCP flows are present in the network, the AP transmits long TCP Data to stations and stations respond by transmitting short TCP Ack back to the AP. In this situation, if with PIFS Access, there are only collisions among stations, and hence the wasted time is equal to the TCP Ack transmission time; on the other hand, if without PIFS Access, the AP's transmission may collide with stations' transmissions, the wasted time is then equal to the TCP Data transmission time, which is about 4 times the TCP Ack transmission time.

Note that, with PIFS Access, the AP's queue is always empty and all the stations always have packets to transmit. That is, PIFS Access makes the stations behave like saturated UDP stations. As a result, the difficult problem of maximizing the TCP throughput is simplified and becomes equivalent to the easier problem of maximizing the UDP saturation throughput. In [18], the authors showed that the maximum throughput of a saturated UDP network is independent of the number of stations and, with 1000-byte MAC frames, the maximum throughput is achieved when the collision rate experienced by a station is about 0.2. This conforms to our simulation results shown in Fig. 8b.

Figure 9 shows the aggregate TCP throughput of download case (Fig. 9a) and upload case (Fig. 9b) when the AP uses PIFS Access and the TXOP mechanism is used. CWmin is set to 31 for stations. Comparing with Fig. 8a (i.e., PIFS Access without TXOP), one can see that the throughput performance is much improved. This is due to the reduced contention among stations by the usage of TXOP. As shown in the figure, the aggregate throughput is not affected by the AP TXOPLimit size. Since AP PIFS Access makes most of the TCP packets reside in the queues of stations regardless of the AP TXOPLimit size, the aggregate throughput depends only on the transmissions of the stations. Therefore, a larger station TXOPLimit yields a higher throughput.

So far, we have evaluated the effects of CWmin values, TXOPLimit sizes, and AP PIFS Access with/out TXOP on the aggregate TCP throughput, but without considering TCP Ack processing delay, wireline delay, and packet loss. Next, we study the effects of these parameters via simulation.

6.4 Effects of TCP Ack processing delay

Figure 10 shows the effects of TCP Ack processing delay on the aggregate TCP throughput. We assume no

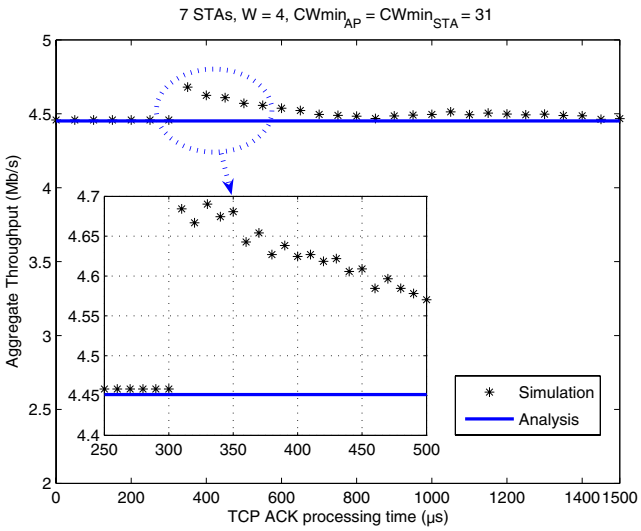


Figure 10 The effects of TCP Ack processing delay (with no wireline delay) when all stations download data from FTP servers

wireline delay in this scenario. In general, we observe that, with non-zero TCP Ack processing delay, the aggregate throughput deviates little from our analysis result (i.e., 4.46 Mbps) by at most 5%. This means that our analysis is reasonably accurate even with the simplifying assumption of no TCP Ack processing delay. As shown in the figure, the throughput increases when the TCP Ack processing delay is around 308 μs. This is due to the *immediate access* behavior that was discussed in Section 2.1. The reason can be explained as follows. Figure 11 shows the timing diagram of the AP and a station that receives a TCP Data from the AP. If the TCP Ack is generated within duration **A** in Fig. 11, the MAC starts backoff at the end of duration **A** because it senses the channel busy. On the other hand, if the TCP Ack is generated within duration **B**, the MAC might send the TCP Ack immediately without backoff. Note that the length of duration **A** is 308 μs = SIFS (10 μs) + T_{ACK} (248 μs) + AIFS (50 μs). In the latter case, in order to have immediate access, the station should satisfy the following two conditions when it receives TCP Data from the AP:

- It has already finished the post backoff;
- It is not contending for the channel to transmit a packet (i.e., TCP Ack).

Most stations satisfy the above conditions because they remain idle for most of the time while waiting for TCP Data from the AP due to AP bottleneck and TCP flow control.

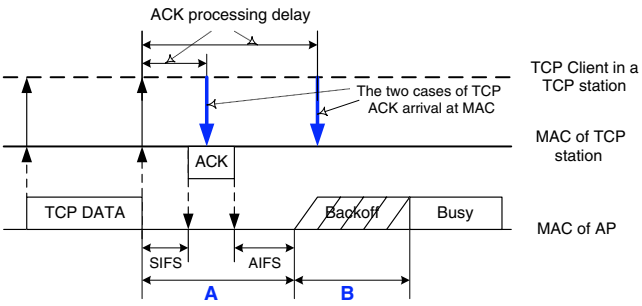
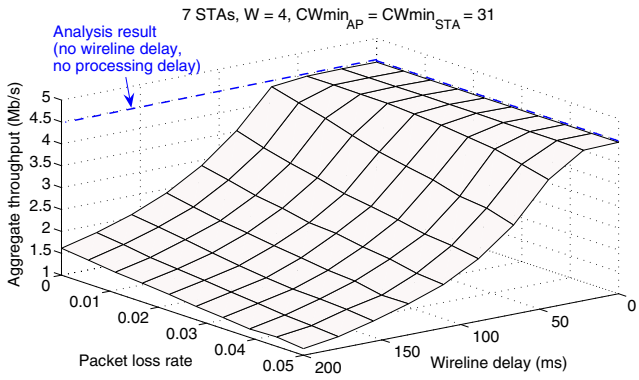
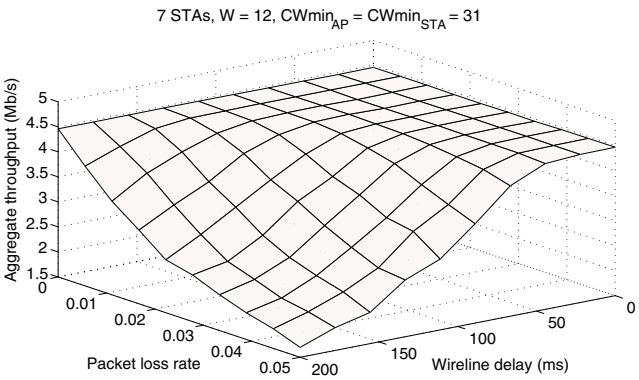


Figure 11 Illustration of the TCP Ack processing delay in a station when it downloads data from a server via the AP



(a) W = 4



(b) W = 12

Figure 12 The effects of wireline delay and packet loss on the aggregate TCP throughput when all stations download data from FTP servers. *W* is the maximum TCP receive window size (in packets)

6.5 Effects of wireline delay and packet loss

Figure 12 shows the effects of wireline delay (from 0 to 200 ms) and the end-to-end packet loss rate (from 0.1% to 5% according to [14]) on the aggregate TCP throughput. Results with the maximum TCP receive window sizes of $W = 4$ and $W = 12$ are shown in Figs. 12a and 12b, respectively. It is clear from the figure that our model produces accurate results when the wireline delay is small and the packet loss rate is low, i.e., when the WLAN is the bottleneck.

However, when the wireline delay gets larger, the WLAN is not the bottleneck anymore. That is, the aggregate TCP throughput is dominantly-determined by the wireline delay. We observe that the aggregate TCP throughput decreases as the wireline delay increases. Moreover, packet loss accelerates the throughput degradation. We observe that the higher the packet loss is, the more quickly the aggregate throughput decreases as the wireline delay increases. The reason is as follows. High packet loss rate causes TCP timeout and in turn reduces the number of outstanding TCP packets, because TCP send window shrinks to one MSS after timeout.

Figure 12b shows that a larger maximum TCP receive window results in a bigger applicable region (i.e., the flat region) for our model. This is because, with a larger maximum TCP receive window, more outstanding TCP packets are allowed in the network, and hence the WLAN may remain as the bottleneck even in the presence of larger wireline delay and/or higher packet loss rate.

7 Conclusion

In this paper, we conduct rigorous and comprehensive modeling and analysis of the TCP performance over EDCA-based IEEE 802.11e WLANs, and show that the aggregate TCP throughput can be enhanced via a proper selection of channel access parameters for the AP and stations. In particular, we show that the best aggregate throughput performance can be achieved via AP's contention-free access for downlink transmissions and the TXOP mechanism. Moreover, some of the assumptions used in our model are evaluated via simulation, and results show that our model is reasonably accurate, particularly, when the wireline delay is small and/or the packet loss rate is low. Future work includes optimization of the 802.11e performance when considering other types of traffic such as VoWLAN.

Acknowledgements This work was in part supported by the ITRC support program of MKE/IITA (IITA-2008-C1090-0801-0013 and IITA-2008-C1090-0803-0004).

References

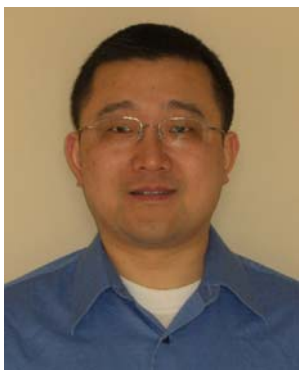
1. The Network Simulator — ns-2. http://www.isi.edu/nsnam/ns/
2. Bianchi G (2000) Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J Select Areas Commun 18(3):535–547
3. Bruno R, Conti M, Gregori E (2004) Analytical modeling of TCP clients in Wi-Fi hot spot networks. In: Proc. IFIP/TC6 networking. Athens, 9–14 May 2004
4. Burmeister C, Killat U (2006) TCP over rate-adaptive WLAN-An analytical model and its simulative verification. In: Proc. IEEE WoWMoM. Niagara-Falls, Buffalo, 26–29 June 2006
5. Cali F, Conti M, Gregori E (2000) Dynamic tuning of the IEEE 802.11 protocol. IEEE/ACM Trans Netw 8(6):785–799
6. Choi S, Park K, Kim C (2006) Performance impact of interlayer dependence in infrastructure WLANs. IEEE Trans Mobile Comput 5(7):829–845
7. Fall K, Floyd S (1996) Simulation-based comparisons of Tahoe, Reno, and SACK TCP. ACM SIGCOMM Comput Commun Rev (CCR) 26(3):5–21
8. IEEE std (1999) IEEE 802.11-1999, Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications
9. IEEE std (2005) IEEE 802.11e, Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: medium access control quality of service enhancements
10. Jiang H, Dovrolis C (2005) Why is the internet traffic bursty in short time scales? In: Proc. ACM SIGMETRICS. Banff, 6–10 June 2005
11. Kherani AA, Shorey R (2006) Modelling TCP performance in Multihop 802.11 networks with randomly varying channel. In: Proc. WILLOPAN. New Delhi, January 2006
12. Stevens WR TCP/IP illustrated, vol 1, the protocols, 2nd edn. Addison-Wesley Reading
13. Tinnirello I, Choi S (2005) Temporal fairness provisioning in multi-rate contention-based 802.11e WLANs. In: Proc. IEEE WoWMoM. Taormina, 13–16 June 2005
14. Yamasaki Y, Shimonishi H, Murase T (2005) Statistical estimation of TCP packet loss rate from sampled ACK packets. In: Proc. IEEE GLOBECOM. St. Louis, 28 November–2 December 2005
15. Yu J, Choi S (2006) Comparison of modified dual queue and EDCA for VoIP over IEEE 802.11 WLAN. European Trans Telecomms 17(3):371–382
16. Yu J, Choi S (2007) Modeling and analysis of TCP dynamics over IEEE 802.11 WLAN. In: Proc. IEEE/IFIP WONS. Obergurgl, 24–26 January 2007
17. Yu J, Choi S, Lee J (2004) Enhancement of VoIP over IEEE 802.11 WLAN via dual queue strategy. In: Proc. IEEE ICC. Paris, 20–24 June 2004
18. Zhai H, Chen X, Fang Y (2005) How well can the IEEE 802.11 wireless LAN support quality of service? IEEE Trans Wireless Commun 4(6):3084–3094

Jeonggyun Yu received his B.E. degree in School of Electronic Engineering from Korea University, Seoul, Korea in 2002. He is currently working toward his Ph.D. in the School of Electrical Engineering at Seoul National University (SNU), Seoul, Korea. His research interests include QoS support, algorithm development, performance evaluation for wireless networks, in particular, IEEE 802.11 wireless local-area networks (WLANs). He is a student member of IEEE.

Sunghyun Choi is currently an associate professor at the School of Electrical Engineering, Seoul National University (SNU), Seoul, Korea. Before joining SNU in September 2002, he was with Philips Research USA, Briarcliff Manor, New York, USA as a Senior Member Research Staff and a project leader for three years. He received his B.S. (summa cum laude) and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1992 and 1994, respectively, and received Ph.D. at the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor in September, 1999.

His current research interests are in the area of wireless/mobile networks with emphasis on wireless LAN/MAN/PAN, next-generation mobile networks, mesh networks, cognitive radios, resource management, data link layer protocols, and cross-layer approaches. He authored/coauthored over 120 technical papers and book chapters in the areas of wireless/mobile networks and communications. He has co-authored (with B. G. Lee) a book "Broadband Wireless Access and Local Networks: Mobile WiMAX and WiFi," Artech House, 2008. He holds 15 US patents, nine European patents, and seven Korea patents, and has tens of patents pending. He has served as a General Co-Chair of COMSWARE 2008, and a Technical Program Committee Co-Chair of ACM Multimedia 2007, IEEE WoWMoM 2007 and IEEE/Create-Net COMSWARE 2007. He was a Co-Chair of Cross-Layer Designs and Protocols Symposium in IWCMC 2006, 2007, and 2008, the workshop co-chair of WILLOPAN 2006, the General Chair of ACM WMASH 2005, and a Technical Program Co-Chair for ACM WMASH 2004. He has also served on program and organization committees of numerous leading wireless and networking conferences including IEEE INFOCOM, IEEE SECON, IEEE MASS, and IEEE WoWMoM. He is also serving on the editorial boards of IEEE Transactions on Mobile Computing, ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), and Journal of Communications and Networks (JCN). He is serving and has served as a guest editor for IEEE Journal on Selected Areas in Communications (JSAC), IEEE Wireless Communications, Pervasive and Mobile Computing (PMC), ACM Wireless Networks (WINET), Wireless Personal Communications (WPC), and Wireless Communications and Mobile Computing (WCMC). He gave a tutorial on IEEE 802.11 in ACM MobiCom 2004 and IEEE ICC 2005. Since year 2000, he has been a voting member of IEEE 802.11 WLAN Working Group.

He has received a number of awards including the Young Scientist Award (awarded by the President of Korea) in 2008; IEEK/IEEE Joint Award for Young IT Engineer of the Year 2007 in 2007; the Outstanding Research Award in 2008 and the Best Teaching Award in 2006 both from the College of Engineering, Seoul National University; the Best Paper Award from IEEE WoWMoM 2008; and Recognition of Service Award in 2005 and 2007 from ACM. Dr. Choi was a recipient of the Korea Foundation for Advanced Studies (KFAS) Scholarship and the Korean Government Overseas Scholarship during 1997–1999 and 1994–1997, respectively. He is a senior member of IEEE, and a member of ACM, KICS, IEEK, KIISE.



Daji Qiao is currently an assistant professor in the Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa. He received his Ph.D. degree in Electrical Engineering-Systems from The University of Michigan, Ann Arbor, Michigan, in February 2004. His current research interests include modeling, analysis and protocol/algorithm design for various types of wireless/mobile networks, including IEEE 802.11 Wireless LANs, mesh networks, and sensor networks. He is a member of IEEE and ACM.