Wireless Pers Commun (2009) 51:499–516
DOI 10.1007/s11277-009-9757-5

A Cross-layer Dual Queue Approach for Improving TCP Fairness in Infrastructure WLANs

Juho Ha · Eun-Chan Park · Kyung-Joon Park · Chong-Ho Choi

Published online: 26 June 2009

Abstract Fairness is one of the most important performance measures in IEEE 802.11 Wireless Local Area Networks (WLANs), where channel is accessed through competition. In this paper, we focus on the fairness problem between TCP uplink and downlink flows in infrastructure WLANs from the cross-layer perspective. First, we show that there exists a notable discrepancy between throughput of uplink flow and that of downlink flow, and discuss its root cause from the standpoint of different responses to TCP data packet drop and TCP ACK packet drop at the access point (AP) buffer. In order to mitigate this unfairness, we propose a dual queue scheme, which works in a cross-layer manner. It employs two separate queues at the AP, one for the data packets of downlink TCP flows and another for the ACK packets of uplink TCP flows, and selects these queues with appropriate probabilities so that TCP per-flow fairness is improved. Moreover, we analyze the behavior of the dual queue scheme and derive throughputs of uplink and downlink flows. Based on this analysis, we obtain the optimal queue selection probabilities for fairness. Extensive simulation results confirm that the proposed scheme is effective and useful in resolving the TCP unfairness problem without deteriorating overall utilization.

J. Ha
LG Electronics, Seoul, Korea
e-mail: hazo77@naver.com

E.-C. Park
Department of Communication and Information, Dongguk University, Seoul, Korea
e-mail: ecpark@dongguk.edu

K.-J. Park (✉)
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA
e-mail: kjp@uiuc.edu

C.-H. Choi
Department of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea
e-mail: chchoi@csl.snu.ac.kr

Keywords Wireless LAN · Fairness · TCP congestion control · Cross-layer design

1 Introduction

Nowadays, Wireless Local Area Networks (WLANs) based on the IEEE 801.11 standard [1] are widely deployed. As the number of WLAN users has grown rapidly, service fairness among users has become an important issue. The IEEE 802.11 standard defines two different medium access control (MAC) mechanisms: the distributed coordination function (DCF) and the point coordination function (PCF). Currently, most of the WLAN devices employ only DCF due to its simplicity and efficiency. DCF provides fair channel access opportunities among wireless stations. The most popular network configuration of WLANs is the infrastructure mode, in which all the wireless stations communicate through an access point (AP). WLANs in the infrastructure mode can provide network access at public areas, such as campus, cafes and hotels.

In this paper, we focus on the unfairness problem among uplink TCP flows and downlink TCP flows in the infrastructure mode WLANs. As most Internet services use the TCP protocol, we will restrict our attention to TCP flows. While DCF allows all the competing stations to have equal opportunity for medium access, it does not guarantee fair service provision among sending and receiving stations when it is applied to the infrastructure mode. In the infrastructure mode, an individual sending station competes for channel access with an AP that is in charge of service for all the receiving stations. This behavior at the MAC layer leads to uplink/downlink asymmetry.

The authors of [2–5] proposed MAC protocols, which leads to fairness among uplink and downlink flows. In [2], the contention windows of wireless stations are dynamically adjusted. By increasing the contention windows for the wireless stations, the AP has more chances to access the wireless channel, and so the fairness among uplink and downlink flows improves. Similarly, the algorithm in [3] differentiates channel access delay between wireless stations and AP using different interframe space, which gives the AP higher priority in accessing the channel. In [4], each station defers channel access based on the next packet information that is collected at AP. The scheme proposed in [5] modifies the channel access mechanism with the mean of backoff distribution, in order to provide fairness among uplink and downlink flows. All these schemes are based on the MAC layer scheduling and give the AP more chances to access the channel than the wireless stations. Since these schemes do not consider the interaction between the MAC and TCP protocols, it is difficult to predict the performance when they are applied to TCP flows.

The issue of TCP fairness in WLANs has been recently addressed in [6–10]. In [6], the authors showed that the unfairness between uplink and downlink flows exacerbates in the case of TCP flows, and that both MAC-induced unfairness and TCP-induced unfairness lead to unfair service. They revealed that, in addition to the TCP-induced unfairness, the MAC-induced unfairness can be avoided if packet loss due to buffer overflow in the AP does not occur. Pilosof et al. [7] showed that the queue size of AP plays an important role in the unfairness problem. They presented a simple solution where the AP manipulates the TCP advertised window field of ACK packets. However this scheme is very complex to implement as the AP needs to manipulate TCP headers of all packets. Also, it can not be applicable to the case where IP payload is encrypted. To enhance fairness, the MAC QoS parameters were differentiated between TCP data packets and TCP ACK packets under IEEE 802.11e framework [8]. In this scheme, the TCP ACK packets are transmitted with minimal queueing, while the downlink TCP data packets are transmitted by transmission opportunity (TXOP) bursting. It can be deployed only in the 802.11e networks, and requires modifying the MAC protocol of wireless stations. In [9], per-flow queueing was deployed to alleviate the TCP unfairness. It performs well, but it is very complex to manage per-flow queues for all the uplink and downlink TCP flows, degrading scalability. Also, Kim [10] proposed a scheme which improves per-station fairness of TCP flows based on the channel access time of each wireless station. It works well but requires some computational overhead in AP.

We propose a dual queue scheme that can solve this unfairness problem induced by both of TCP and MAC layers. The key idea of this scheme is two-fold:

(i) to employ two separate interface queues in the AP, one for the data packets of downlink TCP flows and another for the ACK packets of uplink TCP flows,
(ii) to serve these queues with different probabilities to assure fairness between uplink TCP flows and downlink TCP flows.

In order to obtain the optimal queue selection probabilities for fairness, we analyze the behavior of the dual queue scheme. Through simulation, we observe the phenomenon that the congestion windows of downlink TCP flows are synchronized. Based on this observation, we derive throughputs of uplink and downlink TCP flows. Moreover, we analyze the effect of round trip time (RTT) on throughput, and show that the RTT difference does not significantly deteriorate the fairness when the proposed scheme is implemented in the AP. The proposed scheme is simple, easy to deploy and only requires minor changes in the AP queueing mechanism. We implemented the dual queue scheme in ns-2 [11] simulator and performed simulations to evaluate its performance. The simulation results confirm that, it improves fairness for uplink and downlink TCP flows significantly under various conditions. Also, the simulation results validate the analysis for the optimal queue selection probabilities and show that the overall utilization does not deteriorate in assuring fairness. A preliminary version of this study was presented in [12].

This paper is organized as follows. In Sect. 2, we present how and why the TCP unfairness problem occurs. The dual queue scheme is proposed and analyzed in Sect. 3. Its performance is evaluated through simulation in Sect. 4. In Sect. 5, we present our conclusion.

2 TCP Unfairness

In this section, we present simulation results of ns-2 to illustrate the unfairness problem. Figure 1 shows the network configuration, where uplink and downlink TCP flows are running through an AP. There are m downlink TCP flows and n uplink TCP flows. For the simplicity of analysis, we assume that each wireless station has only one flow. Here, m stations receive data traffic from the AP while n stations send data to the AP. We refer to these stations as `DN_STA`s and `UP_STA`s, respectively.[1] As shown in Fig. 1, all the flows share a link between the AP and the router in wired network. The capacity of the wired links is 100 Mbps and the propagation time is 5 ms for each link. The parameters for the wireless channel are compatible with the IEEE 802.11b MAC/PHY standard. The data rate of WLAN is 2 Mbps. Since the bandwidth of the wired link is much higher than that of the wireless link, the wireless link becomes a bottleneck link. The version of TCP used in the simulations is TCP/NewReno. The size of a TCP data packet is set to 1,500 bytes and the delayed ACK of TCP is not used. The advertised TCP window size of each flow is set to 42 packets, i.e., 63 KB, which is close to the value used in the real implementation.

[1] For the coherence of notation between a station and a flow, `DN_STA`/`UP_STA` indicate stations having a downlink/uplink flow, respectively.

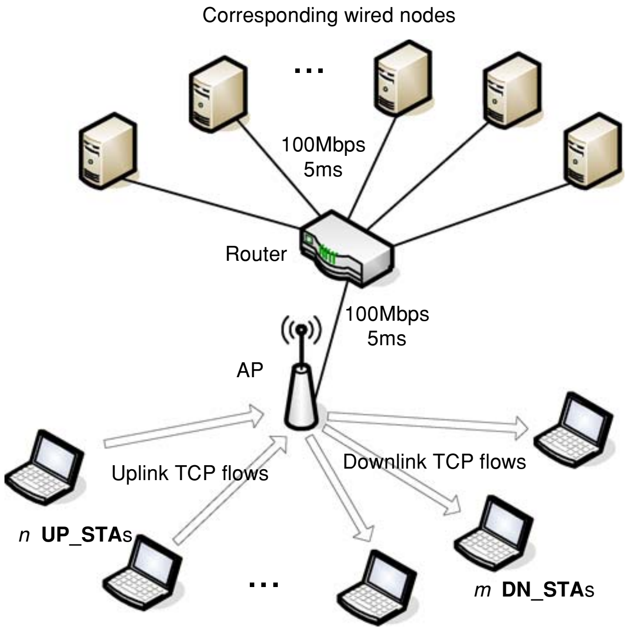
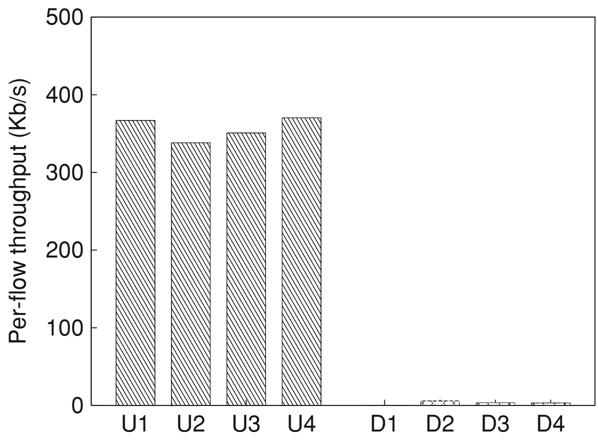


Fig. 1 Network topology

Fig. 2 Throughputs of TCP downlink and uplink flows with single queue; $m = 4$ and $n = 4$



First, we considered a particular case of $m = n = 4$, which means that the packets generated by four uplink TCP flows and four downlink TCP flows go through the AP. The queue size of the AP is set to 100 packets. Figure 2 shows the average throughputs of each flow. The flow indices U1, U2, U3 and U4 refer to the uplink TCP flows while the flow indices D1, D2, D3 and D4 refer to the downlink TCP flows. Through the simulation, we can observe the severe unfairness problem among uplink and downlink TCP flows. The average throughput of the downlink TCP flows is just 3 Kb/s while that of the uplink TCP flows is 357 Kb/s. The downlink TCP flows are almost starved by the uplink TCP flows. For various numbers of uplink and downlink TCP flows, the bias toward uplink TCP flows were always observed.

This phenomenon is mainly caused by TCP-induced asymmetry and MAC-induced asymmetry [6]. TCP-induced unfairness results from the asymmetric behavior of TCP flows responding to packet loss. Note that a TCP connection is bidirectional, i.e., a sender transmits a data packet to a receiver and the receiver sends the corresponding acknowledgement (ACK) packet to the sender. When the uplink and downlink TCP flows share a wireless channel, two different kinds of packets are buffered in the interface queue of the AP: data packets to DN_STAs and ACK packets to UP_STAs. In general, the capacity of a wireless link is smaller than that of a wired link. Moreover, in the infrastructure mode, AP always participates in the communication and so the wireless link easily becomes a bottleneck link and packets are occasionally dropped due to buffer overflow.

The downlink TCP flows and the uplink TCP flows react to this packet loss in different ways. When a data packet for DN_STA is lost, the receiver (DN_STA) transmits duplicate ACKs or a time-out occurs at the sender (wired node). Then, the sender reduces its congestion window size, decreasing the throughput of the downlink TCP flow. However, a loss of an ACK packet for UP_STA usually does not affect its throughput much due to the cumulative ACK mechanism of TCP. Even though an ACK packet is lost due to buffer overflow, the loss does not necessarily invoke the TCP congestion control mechanism of the uplink TCP flow. The cumulative ACK mechanism lets UP_STA tolerate a loss of ACK packet as long as the next ACK packet with a higher sequence number is delivered timely to UP_STA. Then, the sender (UP_STA) does not reduce its congestion window. Due to this asymmetric behavior in response to packet losses in the AP queue, the throughput of uplink TCP flow becomes higher than that of downlink TCP flow.

On the other hand, the MAC-induced unfairness results from the contention-based channel access mechanism of 802.11 MAC. Since the AP competes with n UP_STAs, AP or each UP_STA occupies the channel with the probability of $1/(n+1)$. Moreover, AP sends data packets on behalf of m DN_STAs, each downlink flow has an opportunity to be served by the AP with $1/m(n+1)$, while each uplink flow has an opportunity to be served with $1/(n+1)$. The ratio of average throughput of uplink flows to that of downlink flows can be estimated as

$$\left(\frac{1}{n+1}\right) \Big/ \left(\frac{1}{n+1}\cdot\frac{1}{m}\right) = m.$$

In the case of TCP flows, this estimation does not agree with actual up/down throughput ratio due to the asymmetric behavior TCP congestion control.

3 Dual Queue Scheme for Uplink and Downlink TCP Flows

In this section, we propose a dual queue scheme to resolve the unfairness between uplink and downlink TCP flows. First, we describe its design rationale and observe synchronized packet dropping at the AP queue. Based on this observation, we derive throughputs of uplink and downlink TCP flows. Finally, we obtain an optimal value for the key design parameter, queue selection probability, to assure fairness.

3.1 Design Rationale

We propose a dual queue scheme to alleviate the asymmetric behavior between TCP uplink and downlink flows. There are two possible traffic in the AP, i.e., the downstream TCP data packets and ACK packets to acknowledge the upstream TCP traffic. Hence, we emply two

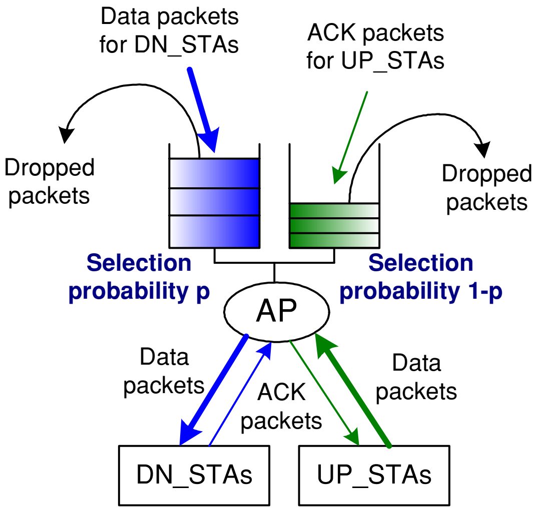


Fig. 3 Schematic diagram of an AP dual queue

queues at the AP: one for the downlink TCP data packets, and another for the ACK packets corresponding to the uplink TCP flows as illustrated in Fig. 3. We refer to these queues as the AP data queue and the AP ACK queue, respectively. With this scheme, data packet droppings of downlink TCP flows and ACK packet droppings of uplink TCP flows can be separated and independent. When the AP has a chance to send a packet, it selects one of the queues as follows. The AP data queue is selected with probability p, while the AP ACK queue is selected with probability $1 - p$. By properly differentiating these two traffic, the AP can control the share of upstream TCP traffic by properly dropping their ACK packets while that of downstream TCP by properly dropping their data packets. It will be further explained in detail in Sect. 3.4 how to set a proper value for p in order to achieve fairness among uplink and downlink TCP.

3.2 Preliminary Simulation

First, we perform a preliminary simulation to confirm the effectiveness of the proposed dual queue system, compared to the single queue system. The queue size of AP data queue and AP ACK queue was 50 packets for each, the selection probability p of AP data queue was 0.7[2], and other simulation environments were the same as those of Sect. 2. Figure 4 illustrates the throughput of each flow. By comparing Fig. 4 with Fig. 2, we can see that the unfairness was alleviated significantly in the dual queue system. The average throughputs of downlink and uplink TCP flows are 160 and 201 Kb/s, respectively. Moreover, the total throughput of eight flows with the dual queue scheme was 1,444 Kb/s while that with the single queue was 1,432 Kb/s, which indicates the proposed scheme improves fairness without decreasing the total throughput.

Figure 5 shows the changes in TCP congestion windows for the four downlink TCP flows. The congestion windows for the uplink TCP flows are not presented because it is seldom affected by a packet dropping at the AP ACK queue and its value is nearly equal to the advertised window size. We can see in Fig. 5 that the changes in the congestion windows for downlink TCP flows are mostly synchronized, implying that most of the packet dropping at

2 Later, we will discuss how to set the queue selection probability properly to improve fairness.

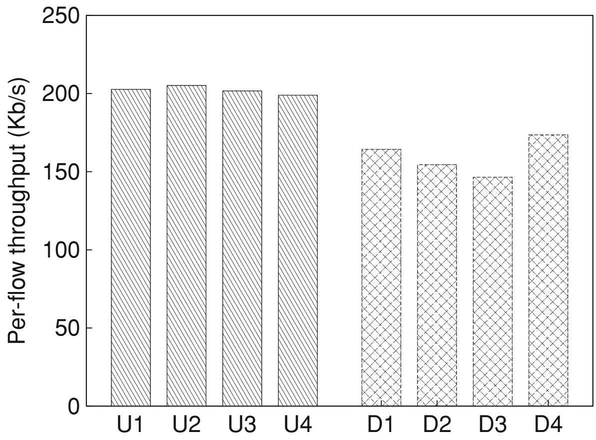


Fig. 4 Throughputs of TCP downlink and uplink flows with dual queue; $m = 4$ and $n = 4$

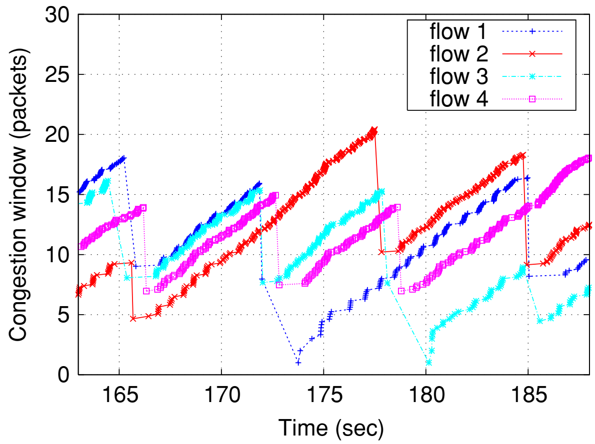


Fig. 5 Synchronization of congestion windows

the AP data queue for each flow occurs simultaneously. This is an important characteristic of TCP flows which share a wireless bottleneck link. There were exceptions where no packet of a certain flow was dropped while some of the newly arriving packets of the other flows were dropped at nearly the same time. For example, in Fig. 5, no dropping occurs for flow 2 at 172 s while other flows suffer packet loss. Also, we can observe that timeout occurred for flow 1 at 173 s and flow 3 at 180 s as shown in Fig. 5, and so the congestion window was reduced to one.

3.3 Derivation of Per-Flow TCP Throughput

In order to determine the selection probability p of AP data queue appropriately, we need to analyze TCP behavior with the dual queue scheme. Here, we derive throughput of TCP uplink and downlink flows. Let Q_D and Q_A denote the AP data queue size and the AP ACK queue size, respectively. Also, let w_r denote the receiver advertised window size and R denote the average number of packets that the AP transmits per unit time. Then, pR is the sending rate of the AP data queue while $(1-p)R$ is that of the AP ACK queue. Let $\tau_{d,i}$ and $\tau_{u,j}$ denote

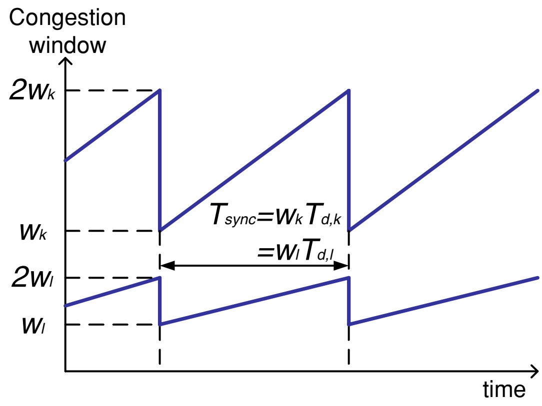


Fig. 6 Changes in congestion windows

the round trip time (RTT) without queueing delay at AP for downlink TCP flow i and that for uplink TCP flow j, respectively. We assume that the delayed ACK of TCP is not used.

First, we derive the throughput of downlink TCP flow. Since TCP data packet dropping occurs depending on the size of AP data queue, we investigate the behavior of downlink TCP flow for two different cases; *(i)* $Q_D < mw_r$ and *(ii)* $Q_D \geq mw_r$.

When $Q_D < mw_r$, data packets of m downlink TCP flows can fill up the AP data queue. In this case, the AP data queue will be a bottleneck for downlink TCP flows and packet dropping may occur at the queue. As shown in Fig. 5, most of the downlink TCP flows were synchronized and packets were dropped simultaneously. For simplicity of analysis, we ignore the exceptional cases in the synchronization of packet dropping and the timeout events. Figure 6 shows the changes in congestion windows for flow k and flow l. Since we ignore the TCP ACK timeout, each downlink TCP flow operates in the congestion avoidance phase in the steady state. Let w_i denote the TCP congestion window size of downlink TCP flow i after the packet dropping event. Then the TCP congestion window size of this flow grows to $2w_i$ before the next packet dropping. Since the average occupied AP data queue length is $\frac{3}{4}Q_D$ and the average sending rate of AP data queue is pR, the average queueing delay at the AP data queue $T_{d,Q}$ is

$$T_{d,Q} = \frac{3}{4}\frac{Q_D}{pR}.$$

Then the average RTT of downlink TCP flow i, $T_{d,i}$ is

$$T_{d,i} = \tau_{d,i} + T_{d,Q} = \tau_{d,i} + \frac{3}{4}\frac{Q_D}{pR}.$$

Since the congestion window increases by one for every RTT in the congestion avoidance phase, the time T_{sync}, taken between the events of synchronized packet droppings as depicted in Fig. 6, becomes

$$T_{\text{sync}} = w_i T_{d,i} = w_i(\tau_{d,i} + T_{d,Q}) \quad \text{for all } i,\ 1 \leq i \leq m. \tag{1}$$

Note that T_{sync} is equal for all flows due to the assumption of synchronization of packet dropping.

On the other hand, the size of AP data queue can be represented in terms of congestion window size. Since data packets are dropped when the AP data queue is full, Q_D becomes approximately

$$Q_D = \sum_{i=1}^{m} 2w_i, \tag{2}$$

under the assumption that most of the in-flight TCP packets are buffered at AP data queue. Finally, we can derive the throughput of downlink TCP flow i, $R_{d,i}$ as

$$R_{d,i} = \frac{\int_{w_i}^{2w_i} w\mathrm{d}w}{T_{\text{sync}}} = \frac{3}{2}\frac{{w_i}^2}{T_{\text{sync}}} = \frac{3}{2}\frac{T_{\text{sync}}}{(\tau_{d,i} + T_{d,Q})^2}. \tag{3}$$

Note that $R_{d,i}$ can be obtained by solving the simultaneous Eqs. 1 and 2 for $m + 1$ unknowns T_{sync} and w_i $(1 \le i \le m)$.

Next, we focus on the case of $Q_D \ge mw_r$, i.e., the downlink TCP data packets can not fill up the data queue and no dropping occurs at the data queue. In this case, the average queueing delay of packets at AP can be calculated by using M/M/1 queueing model [9] as

$$T_{d,Q} = \frac{1}{pR - \sum_{i=1}^{m} R_{d,i}}.$$

Since no dropping occurs at the AP data queue in this case, the TCP window size of downlink TCP flow increases up to w_r. Then the throughput of downlink TCP flow i can be expressed as

$$R_{d,i} = \frac{w_r}{T_{d,i}} = \frac{w_r}{\tau_{d,i} + T_{d,Q}} = \frac{w_r}{\tau_{d,i} + 1/(pR - \sum_{i=1}^{m} R_{d,i})}. \tag{4}$$

In (4), $R_{d,i}$ is not given in the closed form and it should be obtained by solving the simultaneous equations (4) for the unknowns $R_{d,i}$ $(1 \le i \le m)$.

In a similar way, we can obtain $R_{u,i}$, the throughput of uplink TCP flow i. Since, for uplink TCP flows, the ACK packet dropping little affects the congestion window size due to the cumulative ACK mechanism, we assume that the congestion window size of uplink TCP flows are fixed at w_r. Then, $R_{u,i}$ is

$$R_{u,i} = \frac{w_r}{T_{u,i}} = \frac{w_r}{\tau_{u,i} + T_{u,Q}}, \tag{5}$$

where $T_{u,i}$ is the average RTT of uplink TCP flow i and $T_{u,Q}$ is the average queueing delay at the AP ACK queue.

If $Q_A < nw_r$, the AP ACK queue is always filled with ACKs. Therefore, every ACK packets should wait for $T_{u,Q} = Q_A/(1-p)R$, i.e., the time taken for Q_A packets to be transmitted. Then the throughput of uplink TCP flow i would be

$$R_{u,i} = \frac{w_r}{\tau_{u,i} + Q_A/(1-p)R}. \tag{6}$$

Otherwise if $Q_A \ge nw_r$, no packet dropping occurs at the AP ACK queue, and the average queueing delay at the AP corresponds M/M/1 queueing delay and is represented as

$$T_{u,Q} = \frac{1}{(1-p)R - \sum_{i=1}^{n} R_{u,i}}.$$

Hence the throughput of uplink TCP flow i is expressed as

$$R_{u,i} = \frac{w_r}{\tau_{u,i} + 1/((1-p)R - \sum_{i=1}^{n} R_{u,i})}. \tag{7}$$

3.4 Optimal Queue Selection Probability

In this subsection, we investigate the optimal value of queue selection probability based on the TCP throughput derived in the previous subsection. The TCP throughput represented in (3), (4), (6), and (7) can be calculated once R, $\tau_{d,i}$ and $\tau_{u,i}$ are given. However, it is difficult to measure them timely and accurately. In order to circumvent this difficulty, we assume that the queueing delay at AP is dominant in the RTT, i.e., $\tau_{d,i} \ll T_{d,Q}$ and $\tau_{u,i} \ll T_{u,Q}$.[3]

With the approximation that $\tau_{d,i} \approx 0$, (1) and (2) leads to

$$w_1 = \cdots = w_m = \frac{Q_D}{2m},$$

and we can derive the throughput of downlink TCP flow R_d from (3) in the case of $Q_D < mw_r$

$$R_d = R_{d,1} = \cdots = R_{d,m} = \frac{pR}{m}. \quad (8)$$

For the case where no buffer overflow occurs at the AP data queue, i.e., $Q_D \geq mw_r$, (4) becomes

$$R_d = R_{d,1} = \cdots = R_{d,m},$$
$$R_d = w_r(pR - mR_d),$$

by applying $\tau_{d,i} \approx 0$. It can be easily solved and consequently leads to

$$R_d = \frac{w_r pR}{1 + mw_r} \approx \frac{pR}{m}, \quad (9)$$

which is the same as (8) implying that R_d is independent of the size of AP data queue.

Similarly, the throughput of uplink TCP flow for the case of $Q_A < nw_r$ is derived from (6) with the approximation $\tau_{u,i} \approx 0$, i.e.,

$$R_u = R_{u,1} = \cdots = R_{u,n} = \frac{w_r}{Q_A}(1 - p)R. \quad (10)$$

For the other case of $Q_A \geq nw_r$, (7) becomes

$$R_u = R_{u,1} = \cdots = R_{u,n},$$
$$R_u = w_r((1 - p)R - nR_u),$$

by applying $\tau_{u,i} \approx 0$. Then, it leads to

$$R_u = \frac{w_r(1 - p)R}{1 + w_r n} \approx \frac{(1 - p)R}{n}. \quad (11)$$

Consequently, the up/down throughput ratio γ, becomes

$$\gamma = \frac{R_u}{R_d} = \begin{cases} \frac{m(1-p)}{p}\frac{w_r}{Q_A}, & \text{if } Q_A < nw_r \\ \frac{m(1-p)}{np}, & \text{if } Q_A \geq nw_r \end{cases} \quad (12)$$

from (8–11).

[3] In the next subsection, the assumptions of $\tau_{d,i} \ll T_{d,Q}$ and $\tau_{u,i} \ll T_{u,Q}$ will be validated, and the effects of $\tau_{d,i}$ and $\tau_{u,i}$ will be discussed.

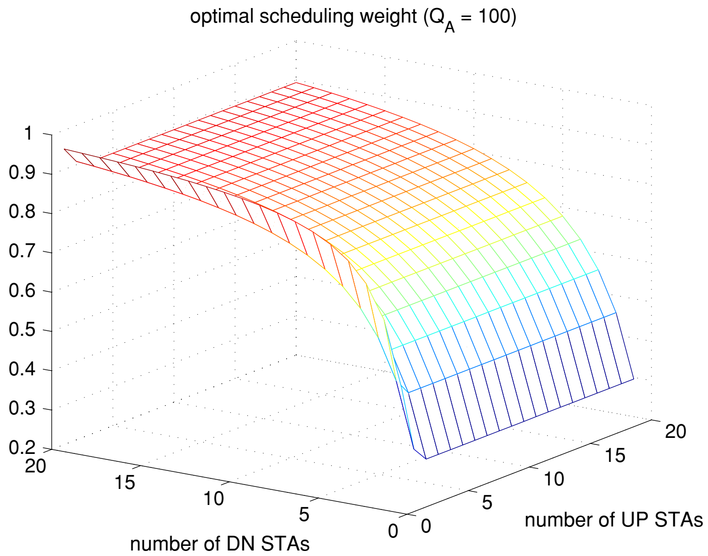


Fig. 7 Optimal value of p as a function of the number of DN STAs and UP STAs when $Q_A = 100$

In order to achieve the fairness among uplink and downlink TCP flows, the average throughput of uplink and downlink TCP flows should be same, i.e., $\gamma = 1$. From (12), we can derive the optimal value for p as

$$p = \begin{cases} \frac{mw_r}{mw_r + Q_A}, & \text{if } Q_A < nw_r \\ \frac{m}{m+n}, & \text{if } Q_A \geq nw_r \end{cases}. \tag{13}$$

An illustrative example of the optimal p is given in Fig. 7.

It is important to note that p in (13) depends on the number of downlink and uplink flows and the size of ACK queue, not on the size of data queue. If the size of AP ACK queue is large enough to contain all the ACK packets for uplink flows, i.e., $Q_A \geq nw_r$, the data queue selection probability becomes in proportion to the number of downlink flows, while ACK queue selection probability becomes in proportion to the number of uplink flows accordingly. Otherwise if $Q_A < nw_r$, the number of uplink flows does not affect both data and ACK queue selection probabilities. Instead, the sizes of ACK queue and advertised congestion window are used in determining the queue selection probabilities.

In order to implement the dual queue with this value p, AP should be able to classify a packet as either a TCP data packet or a TCP ACK packet. For this purpose, AP may simply check the packet size because ACK packets are much smaller than data packets in size. To determine the optimal value of p, the AP should keep track the numbers of uplink and downlink TCP flows, which are difficult to estimate accurately. This difficulty can be avoided by serving the data queue and ACK queue in a round-robin manner, i.e., $p = 0.5$. As will be evaluated through simulation in the next section, the dual queue system with an appropriate value of p, even though not optimal, outperforms the single queue system significantly. Furthermore, the dual queue scheme can be extended to the case where both TCP and UDP flows coexist by employing another queue for UDP flows and serving these queues with modified queue selection probabilities. This extension is beyond the scope of this paper and is our future work.

3.5 Effect of Delay on Fairness

In this subsection, we analyze TCP throughput in the case where $\tau_{d,i}$ and $\tau_{u,i}$ are not negligible so that each flow has different throughput. Though the throughput of each flow cannot be solved easily, the throughput ratio between downlink TCP flows can be derived directly from (3). Since T_{sync} does not change for all downlink TCP flows even in this case, the throughput ratio becomes

$$\begin{aligned} R_{d,i} : R_{d,j} &= 1/(\tau_{d,i} + T_{d,Q})^2 : 1/(\tau_{d,j} + T_{d,Q})^2 \\ &= 1/T_{d,i}{}^2 : 1/T_{d,j}{}^2 \end{aligned}.$$

This implies that the throughput of downlink TCP flows are inversely proportional to the square of RTT, which is in contrast to the well-known fact that the throughput of TCP flow is inversely proportional to the RTT in wired networks [13]. But in this case, the flows share a same wireless bottleneck link and packet droppings are synchronized, so the inverse relationship does not hold. The throughput difference among downlink TCP flows with different RTT cannot be reduced in the proposed dual queue scheme since it does not adopt other active queue management mechanism. The difference is a natural characteristic of TCP flow. Even though there exists unfairness among downlink TCP flows, the influence of RTT is reduced by queueing delay which is the dominant component of RTT. We evaluated the range of throughput ratio in the case where data rate of WLAN was 2 Mbps, AP data queue size was 50 packets, p was 0.667 and TCP data packet size was 1500 bytes. When $\tau_{d,1}$ was 20 ms and $\tau_{d,2}$ varied from 1 to 500 ms, i.e., $\tau_{d,1}/\tau_{d,2} = 20 \sim 0.04$, the throughput ratio $R_{d,1}/R_{d,2}$ changed only from 0.9 to 3.8.

On the other hand, the throughput of uplink TCP flows are simply inversely proportional to the RTT from (5). In this case, the influence of RTT is much smaller than that of downlink TCP flows. The throughput ratio $R_{u,1}/R_{u,2}$ changed from 0.95 to 1.95, for the same condition of the downlink TCP flows as above, when $\tau_{u,1}$ was 20 ms and $\tau_{u,2}$ varied from 1 to 500 ms.

In the range of normal operation, the RTT is not as much influential on the throughput of TCP flows as the packet droppings.

4 Simulation Study

We performed extensive simulations and compared the performance of dual queue scheme with that of single queue scheme to verify the effectiveness of the proposed scheme and to validate the analysis. Simulation environment was the same as in Sect. 2.

4.1 Effect of Queue Selection Probability on Fairness

In the first simulation, we investigated the effect of p on fairness under the following traffic scenarios;

- CASE1: The number of downlink TCP flows varied from one to ten while that of uplink TCP flows was fixed at two, i.e., $m = 1 \sim 10$ and $n = 2$.
- CASE2: In opposite to the first case, the number of uplink TCP flows varied while that of downlink flows was fixed, i.e., $m = 2$, $n = 1 \sim 10$.
- CASE3: Both the numbers of downlink and uplink flows increase from one to ten, i.e., $m = n = 1 \sim 10$.

Simulation was performed on four schemes: the single queue and the dual queue with either the optimal probability p given in (13), or fixed probabilities $p = 0.5$ or $p = 0.7$. Figure 8

Fig. 8 Throughput ratio of single queue scheme and dual queue scheme for CASE1, $m = 1 \sim 10, n = 2$

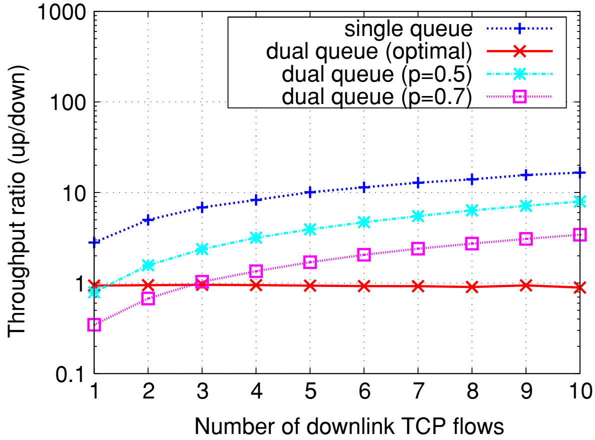
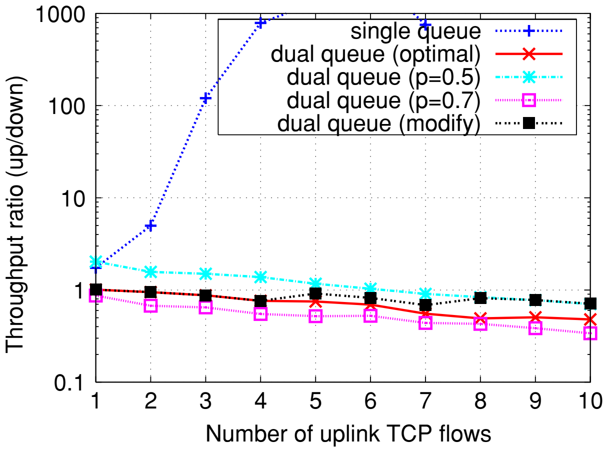


Fig. 9 Throughput ratio of single queue scheme and dual queue scheme for CASE2, $m = 2, n = 1 \sim 10$



shows the average up/down throughput ratio, γ, in log scale. Each point was marked with the average value of ten simulations. For the single queue, γ increased as the number of downlink TCP flows increased and it exceeded ten when $m = 10$. For the dual queue with the optimal value of p, γ remained near 1 which implies excellent fairness among the uplink and downlink TCP flows, when p was calculated as (13). When it is difficult to estimate the number of uplink or downlink TCP flows, we can set the queue selection probability p to a fixed value. For example, we can simply set p to 0.5 for the round robin scheduling or 0.7 as a near optimal value. In Fig. 8, when p was fixed at either 0.5 or 0.7, the up/down throughput ratio increased as the number of downlink TCP flows increased, but it was much closer to 1 than that of the single queue case.

The outstanding performance of the dual queue scheme can be observed for the CASE2 ($m = 2$ and $n = 1 \sim 10$) and CASE3 ($m = n = 1 \sim 10$), and their throughput ratios are given in Figs. 9 and 10, respectively. The dual queue scheme showed much better fairness than the single queue scheme. In the case of single queue, γ increased exceeding 1,000 for some cases, while γ in the case of dual queue did not increase exceeding 4 for all the cases, regardless of the values of p. For certain cases, the dual queue with fixed p showed better performance than the dual queue with optimal p.

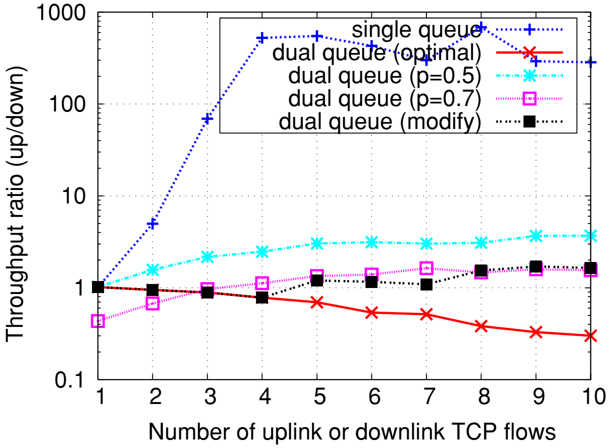


Fig. 10 Throughput ratio of single queue scheme and dual queue scheme for CASE3, $m = n = 1 \sim 10$

Table 1 Total throughput of single queue scheme and dual queue scheme for several values of uplink and downlink flows

(m, n)	Total throughput (Kb/s)		
	Single queue queue	Dual queue (optimal)	Dual queue (modify)
(2,2)	1360	1395	1395
(2,5)	1438	1467	1472
(2,10)	1480	1491	1502
(5,2)	1363	1371	1371
(5,5)	1458	1416	1446
(10,2)	1369	1357	1357
(10,10)	1481	1378	1471

In Figs. 9 and 10, we can see that γ with the optimal queue selection probability gets smaller as n increases. The reason for this phenomenon is that some of the assumptions in Sect. 3 failed to hold when n was very large. In this case, too many ACK packets belonging to an uplink TCP flow could have been dropped simultaneously. This in turn may have led to TCP timeout. Therefore, it can be said that the uplink TCP flows did not have the throughput as large as evaluated in Sect. 3, and p calculated as (13) was larger than needed. We made some modification on p to resolve this problem. We simply reduced the value of p by 10% when $n > 4$, and p by 20% when $n > 7$. Since the optimal modification rule was difficult to find analytically, these reduction factors were selected in a heuristic manner. The legend "dual queue (modify)" in Figs. 9 and 10 show the results when this modification was applied. The value of γ for the modified dual queue was closer to 1 than that of the original dual queue scheme.

As well as the throughput ratio, we computed the total throughput for several cases of (m, n), which is listed in Table 1. Table 1 confirms that the proposed scheme does not reduce the total throughput in assuring fairness.

Table 2 Effect of queue size and delay on fairness of single queue scheme and dual queue scheme

System parameter	Throughput ratio	
	Single queue	Dual queue
Queue size (packets)		
50	785	0.780
100	526	0.934
200	5.09	1.01
500	1.01	1.01
1000	1.01	1.01
Delay (ms)		
5	526	0.780
10	272	0.823
20	221	0.755
50	229	0.755
100	200	0.714

4.2 Effect of Queue Size and Delay

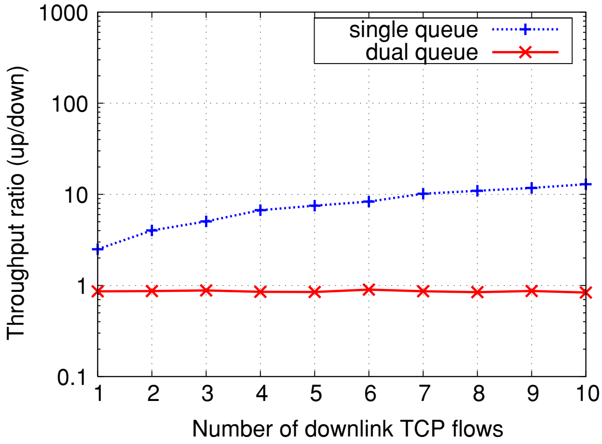
To see the impact of the queue size on the fairness, we performed simulations for various values of queue size. In this simulation, we set $m = n = 4$. In the case of the single queue scheme, we can see in Table 2 that the fairness improves by increasing the queue size. The reason is that the packet droppings of downlink TCP flows are reduced as the queue size increases. As long as the queue size is larger than 500 packets, the throughput ratio is almost equal to one. Since the number of the maximum in-flight packets generated by eight TCP flows is mw_r (= 336 packets in this case), the fairness may be achieved when the queue size of AP is greater than this value. But increasing the queue size induces a large queueing delay. In the dual queue scheme, the fairness is achieved even when the queue size is as small as 50. This result confirms that the dual queue scheme enables fair services without employing a large AP queue.

We repeated the simulation to study the influence of RTT on the fairness. In this simulation, we modified the propagation delay of two wired links, one for an uplink TCP flow and another for a downlink TCP flow, from 5 ms to $10 \sim 100$ ms. Each of the modified link connects the router and a corresponding wired node, as depicted in Fig. 1. Table 2 lists the average throughput ratio, which shows that the dual queue scheme outperforms the single queue scheme even when there exist significant RTT differences among flows and that the performance of the dual queue scheme is almost immune to the RTT differences.

4.3 Effect of Wireless Channel Error

Lastly, we considered the case where wireless channel error exists. Simulation was performed for two uplink TCP flows and various numbers of downlink TCP flows with bit error rate (BER) of 10^{-5}, which is considered to be the worst operating environment. If the BER is high, packets can be lost due to wireless channel error, as well as buffer overflow at the AP buffer. The TCP sender reacts to packet losses without differentiating their causes and the congestion window is unnecessarily reduced for the packet loss due to wireless channel error.

Fig. 11 Throughput ratio with wireless channel error, $m = 2$, $n = 1 \sim 10$



Thus, we need to evaluate the performance of dual queue scheme at the presence of high BER. From Fig. 11, we see that the throughput ratio of the dual queue scheme is almost equal to one, i.e., the dual queue scheme maintains fairness with little influence from the wireless channel error.

5 Conclusion

In this paper, we studied on the unfairness problem among uplink and downlink TCP flows. We found that the TCP-induced asymmetry as well as the MAC-induced asymmetry leads to significant unfairness on throughput of TCP flows. In order to solve this problem, we presented the dual queue scheme. The proposed dual queue scheme employs two separate queues, one for data packets of downlink TCP flow and another for ACK packets of uplink TCP flow, and serves these queues with proper probabilities. We analyzed the behavior of TCP flows on the dual queue and derived a simple formula for the ratio of average throughput of uplink flows to that of downlink flows by observing the synchronization phenomena of the TCP packet dropping. In order to assure fairness among uplink and downlink flows, we showed how to set the queue selection probabilities for the downlink data queue and the uplink ACK queue. The simulation results show that the dual queue scheme improves the TCP fairness significantly compared to that of the single queue scheme. Simulations on various numbers of flows, queue sizes, propagation delays, and channel error show that this scheme is feasible and robust. The advantages of the dual queue scheme are that it is very simple, effective, and requires only small changes in the AP interface queue and no changes in the MAC layer.

References

1. IEEE 802.11 WG (1999). Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification. *IEEE 802.11 Standard*.
2. Bottigliengo, M., Casetti, C., Chiasserini, C.-F., & Meo, M. (2003). Smart traffic scheduling in 802.11 WLANs with access point. In *Proceedings of IEEE VTC 2003 Fall*, October 2003.

3. Kim, S., Kim, B., & Fang, Y. (2005). Downlink and uplink resource allocation in IEEE 802.11 wireless LANs. *IEEE Transactions on Vehicular Technology, 54*(1), 320–327.
4. Ogawa, M., Kuno, Y., Sakata, T., & Kubota, S. (2005). MAC protocol for fair two-way communications in wireless LANs. In *Proceedings of ICPPW 2005*, June 2005.
5. Jeong, J., Choi, S., & Kim, C. (2005). Achieving weighted fairness between uplink and downlink in IEEE 802.11 DCF-based WLANs. In *Proceedings of IEEE QShine 2005*, August 2005.
6. Park, E.-C., Kim, D.-Y., & Choi, C.-H. (2006). Analysis of unfairness between TCP uplink and downlink flows in Wi-Fi hot spots. In *Proceedings of IEEE Globecom 2006*, November 2006.
7. Pilosof, S., Ramjee, R., Raz, D., Shavitt, Y., & Sinha, P. (2003). Understanding TCP fairness over wireless LAN. In *Proceedings of IEEE INFOCOM 2003*, April 2003.
8. Leith, D. J., Clifford, P., Malone, P., & Ng, A. (2005). TCP fairness in 802.11e WLANs. *IEEE Communications Letters, 9*(11), 964–966.
9. Wu, Y., Niu, Z., & Zheng, J. (2005). Study of the TCP upstream/downstream unfairness issue with per-flow queueing over infrastructure-mode WLANs. *Wirless Communications and Mobile Computing, 5*(4), 459–471.
10. Kim, D.-Y. (2006). Access time control for TCP fairness in WLANs. Ph. D. Dissertation, Seoul National University.
11. Network Simulator 2 (NS-2) http://www.isi.edu/nsnam/ns.
12. Ha, J., & Choi, C.-H. (2006). TCP fairness for uplink and downlink flows in WLANs. In *Proceedings of IEEE Globecom 2006*, November 2006.
13. Padhe, J., Firoiu, V., Towsley, D., & Kurose, J. (1998). Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM 98*.

Author Biographies

Juho Ha received his B.S. and M.S. degrees from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea in 2000 and 2007, respectively. He is currently a research engineer of LG Electronics, Seoul, Korea. He worked for Locus, Seoul, Korea as a research engineer in 2000–2001, and for C&S Technology, Seoul, Korea as a research engineer in 2001–2004. His research interests include personal area networks and mobile connectivity solutions.

Eun-Chan Park received the B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, in 1999, 2001, and 2006, respectively. He was a senior engineer in Samsung Electronics, Ltd. from 2006 to 2008. He is currently an assistant professor of the Department of Communication and Information, Dongguk University, Seoul, Korea. His research interests include performance analysis, resource allocation, quality of service, congestion control, and cross-layer optimization in wired and wireless networks.

Kyung-Joon Park received his B.S., M.S., and Ph.D. degrees all from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea in 1998, 2000, and 2005, respectively. He is currently a postdoctoral research associate in the Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA. He worked for Samsung Electronics, Suwon, Korea as a senior engineer in 2005–2006, and was a visiting graduate student, supported by the Brain Korea 21 Program, in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign in 2001–2002. His current research interests include characterization and design of medical-grade protocols for wireless healthcare systems, design and analysis of self-adjusting protocols for wireless environments, and distributed control of physical carrier sense in wireless networks.

Chong-Ho Choi received the B.S. degree from Seoul National University, Seoul, Korea, in 1970 and the M.S. and Ph.D. degrees from the University of Florida, Gainesville, in 1975 and 1978, respectively. He was a senior researcher at the Korea Institute of Technology from 1978 to 1980. He is currently a professor in the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea. His research interests include modeling, optimization, and control of communication networks.