Cross Layer Design for the IEEE 802.11 WLANs: Joint Rate Control and Packet Scheduling

Qiuyan Xia, *Student Member, IEEE,* Xing Jin, *Student Member, IEEE,* and Mounir Hamdi, *Senior Member, IEEE*

***Abstract*— With the rapid development in wireless communication technologies, the IEEE 802.11 WLANs are experiencing a huge popularity and widespread deployment. Designed with traditional layered architecture, current WLANs adopt functional layer partitioning and aim at optimization at individual layers. However, in a highly dynamic and media sharing wireless environment, the capacity enhancements at individual physical layers may not necessarily benefit, and sometimes even degrade the system performance with multiple users. It has been shown that in a multiuser setting, one can increase the throughput substantially if partial knowledge of the channels at the receiver sides is known. The challenge is to make good matching of the instantaneous channel conditions of multiple users with the bandwidth and time allocation to each user. In this paper, we address the issue of cross layer design in the proposed "Weighted Fair Scheduling based on Adaptive Rate Control" (WFS-ARC) framework, where the PHY layer knowledge is shared with the MAC and LLC layer in order to provide efficient resource allocation. We evaluate the WFS-ARC approach in ns-2 and the simulation results demonstrate that our design can significantly improve the system throughput.**

***Index Terms*— Cross layer design, IEEE 802.11, multiuser diversity, rate control, weighted fair scheduling, WLAN.**

I. Introduction

In recent years, the IEEE 802.11 standard [1] has emerged as a prevailing standard in the market for WLANs (Wireless Local Area Networks), and it will play a major role in the next-generation of wireless communication networks. The current PHY (Physical) layer extensions provide multiple data rates by using different modulation and coding schemes. For example, the 802.11b [2] PHY layer provides 4 different data rates of 1, 2, 5.5 and 11 Mbps, and the 802.11a [3] standard defines 8 different data rates ranging from 6 Mbps up to 54 Mbps. Typically, higher data rates require higher SNR (Signal to Noise Ratio) to maintain a certain BER (Bit Error Rate); on the other hand, lower data rates can ensure a small BER, but the achieved throughput is also small.

In wireless systems, the propagation environments vary over time and space due to such factors as signal attenuation and fading, motion of objects, interference and so on, causing variations in the received SNR [4]. As a result, there is no single modulation and coding scheme (which decides the data rate) that can be optimal under all scenarios. However, the standard does not specify when and how to switch among different rates. As the multi-rate enhancements are within the PHY layer protocols implemented in the hardware, MAC (Medium Access Control) layer mechanisms are required to exploit this capability. A rate control algorithm is a major solution to handle this problem, which tunes the data rate to the user's current channel conditions. However, the transmitter only has limited, indirect CSIR (Channel State Information at the Receiver) feedback such as ACKs (Acknowledgements) and retry counts under current 802.11 implementations. Therefore, the rate control algorithm is required to be adaptive while simple enough with little overhead.

While the rate control scheme at the MAC layer aims at adjusting the data rate following the changes of the channel conditions to maximize the throughput of a single user, the overall system performance may not necessarily be optimized. Typically, in a multiuser setting, different users undergo different channel gains, which results in the phenomenon of "multiuser diversity". However, in most infrastructure WLANs, the multiuser diversity is rarely exploited and sometimes even degrades the system performance. For example, FIFO queueing at the Access Point (AP) may give rise to the HOL (Head-of-Line) Blocking problem, as the HOL packet transmitted through a low quality radio channel consumes much airtime and prevents other transmissions occurring over good channels. This leads to low channel utilization and "performance anomaly", i.e., when some communication peers use a lower data rate than the others, the performance of all users is considerably degraded [5]. Taking account of these factors, a transmission scheduler at the LLC (Logical Link Control) layer is required to exploit the multiuser diversity, by opportunistically selecting a feasible user with a good channel. At the same time, it should balance the system throughput and fairness requirements among "good" and "bad" users.

Applying the above ideas to our proposed cross layer design, the WFS-ARC approach dynamically adjusts the data rate parameters at the MAC layer, based on the available CSIR provided by the PHY layer. On top of the MAC layer, a LLC layer scheduler is implemented to opportunistically schedule the packet transmission to the most promising user and satisfy the fairness constraints. We remark that while exploring a much richer interaction between parameters across layers, we should not destroy the necessary independence and integrity of individual layers, which may lead to various undesirable consequences.

The rest of the paper is organized as follows. We briefly

Manuscript received February 24, 2006; revised November 8, 2006; accepted May 31, 2006. The associate editor coordinating the review of this paper and approving it for publication was M. Guizani. This work was supported by a grant from the Research Grants Council (RGC) under contract HKUST6260/04E.

The authors are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China (e-mail: {xiaqy, csvenus, hamdi}@cse.ust.hk).

Digital Object Identifier 10.1109/TWC.2007.06019.

introduce the IEEE 802.11 standard in Section II-A. Related work is described in Section II-B. In Section III, we first give an overview of our cross layer design framework and explain how it functions. Based on that, we present our adaptive rate control algorithm and the optimization model for weighted fairness scheduling in detail. Section IV evaluates the simulation performance of our approach. Finally, this paper concludes with Section V.

II. Background and Related Work

A. The IEEE 802.11 WLAN

The IEEE 802.11 [1] standard specifies the MAC and PHY layers for a WLAN system. There are currently three PHY layer extensions: 802.11b/a/g [2] [3] [6], all providing multiple data rates. The PHY layers perform the tasks of carrier sensing, transmitting, and receiving 802.11 frames [7]. The common MAC layer defines rules for orderly access to the shared medium in support of the LLC layer. Two medium access mechanisms are defined in 802.11: the Distributed Coordination Function (DCF) is a mandatory, contention-based protocol; the other one, the Point Coordination Function (PCF), is a priority-based, contention-free protocol. Throughout this paper, we present our cross layer design based on 802.11a WLAN working under the DCF. Our approach can easily be extended to WLANs with other PHY layers such as 802.11g.

The DCF implementation is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. The basic access mode works as follows: before a station starts a frame transmission, it checks the medium status by carrier sensing. If the medium is idle, the transmission may proceed; if the medium is sensed busy, the station defers its transmission until the medium is determined to be idle for the DIFS interval and a random backoff procedure is invoked. While the medium stays idle, the backoff timer is decreased by one for each slot time, and the frame is transmitted when the timer reaches zero; otherwise, the backoff procedure is suspended and resumes after the channel has been idle for the DIFS. For each successful reception of a frame, the receiver immediately acknowledges the frame reception by sending an ACK frame. The ACK frame is transmitted after the SIFS interval, which is shorter than the DIFS. If the ACK is correctly received, the station defers for the DIFS and another random backoff procedure before transmitting the next frame. If the ACK frame is not received correctly within an "ACK timeout" period after the data transmission, the transmitter waits for the EIFS interval and a random backoff before the frame is retransmitted. The DCF also defines an optional mechanism to avoid the "hidden node" problem, which requires the transmitter and receiver to exchange short RTS and CTS control frames before the actual data transmission. A station has to insert proper duration information in the headers of RTS/CTS and Data frames, which indicates the amount of time the wireless medium is to be reserved for the next frame. Then every station overhearing this information can update its Network Allocation Vector (NAV) that represents the amount of time it has to defer. We assume basic access mode (the default one in practice) in this paper.

The 802.11 standard supports following topologies: Independent BSS (IBSS) and infrastructure BSS. An IBSS (or Ad Hoc network) is a stand alone BSS without backbone infrastructure and occupies a smaller area. Stations in an IBSS communicate directly with each other and thus must be within direct communication range. On the other hand, infrastructure networks typically use APs for all communications between them, including communication within stations in the same BSS. Each station must associate with an AP to obtain network services. In this paper, we focus on the infrastructure BSS with an AP, which is the most common deployment extending the Internet with a wireless last hop.

B. Related Work

Several rate control and packet scheduling algorithms have been proposed in the literature. However, most of them discuss these two problems separately, each focusing on the optimization of an individual layer, and do not take account of cross layer interactions.

1) Rate Control Algorithms: In the 802.11 standard, there is no specification for rate adaptation; in addition, no signaling mechanism is available for the receiver to notify the transmitter about the channel conditions. It is intentionally left open to vendors to implement it as they like. The Auto Rate Fallback (ARF) algorithm [8] in Lucent's WaveLAN-II devices, is one of the few used in commercial WLAN products and also available to the public. This scheme works as follows: if 2 consecutive ACKs are not received correctly, the sender drops the transmission rate for the second retry of the current packet as well as the subsequent transmissions to the next lower data rate, and starts a timer; if 10 consecutive ACKs are received successfully, the transmission rate is raised to the next higher data rate and the timer is cancelled; once the timer expires, the transmission rate is raised as before, but in the case where an ACK is not received for the very next packet (commonly referred to as a probing packet), the rate is reduced again and the timer is restarted. Obviously, this scheme is easy to deploy with existing 802.11 devices since it requires no changes to the standard. One disadvantage is that it cannot react quickly to fast channel variations. On the other hand, it may overreact when the channel quality is stable over a long period. Since it periodically tries to send data at the next higher rate in an attempt to gauge the channel condition, there may be repeated packet losses, which result in consistent performance degradation.

Different from the ARF algorithm, which is solely based on gathering ACK statistics, some rate adaptation algorithms assume certain communication between the transmitter and the receiver regarding the link condition. In the Receiver Based Auto Rate (RBAR) protocol [9], the receiver estimates the channel conditions using a sample of instantaneously received signal strength at the end of RTS reception, and then feeds back the selected data rate to the transmitter in CTS. Similar ideas can be found in [10], which further extends RBAR to opportunistically send multiple back-to-back data packets whenever the channel quality is good. However, these schemes are not practically implemented sine standard modification is inevitable.

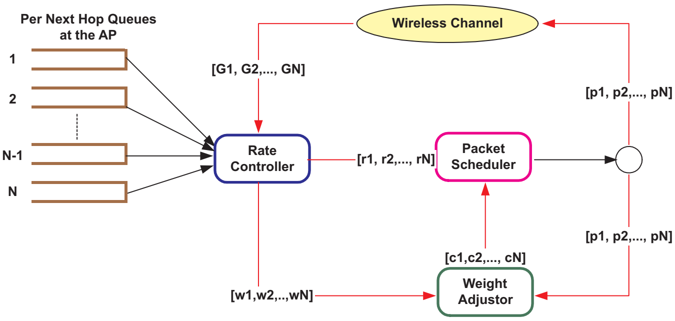


Fig. 1. WFS-ARC Framework: Rate Controller at the MAC layer adjusts PHY layer data rates. At the LLC layer, Packet Scheduler schedules packet transmissions to maximize the weighted goodput, and Weight Adjustor adjusts the control vector to satisfy the fairness constraints.

In the above schemes, only time diversity of single link is considered. By tuning data rate to time-varying link conditions, these methods mitigate channel variations rather than utilize them. The multiuser diversity is completely ignored.

2) Scheduling Algorithm: In an infrastructure WLAN, usually an AP maintains a single FIFO queue for all traffic flows. Since the HOL packet may be either transmitted at a low rate, or retransmitted many times before it succeeds or is finally dropped, it prevents other packets in the queue from being transmitted, leading to very low channel utilization. Note that the instantaneous channel gains from the AP to different clients are independent; however, due to the HOL blocking in the FIFO queue, we cannot make use of variations in channel quality.

The Channel State Dependent Packet Scheduling (CSDPS) scheme proposed in [11] addresses this problem. The basic idea is that, when a wireless channel is observed at the "bad" state, it switches to transmit on other "good" links, so that the HOL effect is removed. However, this method has some limitations. First, it uses a link status monitor to continuously track the channel quality. Such a mechanism is not available in current 802.11 WLANs due to implementation costliness. Second, the binary channel model assumed is too simple to account for the realistic time-varying wireless environment.

There are other scheduling schemes proposed in the literature. However, some are designed specifically for HDR or CDMA systems [12]; others do not take advantage of a rate adaptive MAC [13]. Again, few explicit models or solutions are proposed for jointly using the information from different layers, i.e., from a cross layer design perspective for efficient wireless resource allocation. Note that in infrastructure WLANs (e.g., a hot spot deploying an 802.11 cell and providing Internet access for multiple clients), the overall system performance (e.g., the aggregate throughput) is more desirable than the single user performance. On the other hand, as described in [14], there also exists a fundamental tradeoff between throughput and fairness. Therefore, the fairness constraints should be satisfied while trying to improve the system performance. We describe our cross layer design, i.e., Weighted Fair Scheduling based on Adaptive Rate Control (WFS-ARC) framework, in detail in the next section.

III. WFS-ARC Framework

A. Cross Layer Design

There has been increased interest in cross layer design in an effort to improve the performance of wireless networks [15]. This approach results from the fundamental difference between the wireless medium and the wired one. In wireless networks, where channel quality can vary dramatically in both time and frequency, knowledge of the channel state can be exploited to significantly improve performance [16]. For example, modern wireless interface cards usually employ rate adaptation methods (e.g., ARF) to adapt their data rates over time based on time-varying channel fading. Both high error rates or low data rates at the PHY layer affects the higher-layer performance. In a multiuser setting, channel quality varies independently across different users, which results in the phenomenon of "multiuser diversity". As the number of users in a system increases, the probability that one user has a very good channel also increases. Exploiting this diversity results in a total system throughput gain that increases with the number of users, since it provides a user with an opportunity to transmit data to one of its neighbors with good channel quality before those with bad channel quality. However, this must be balanced with higher-layer issues (e.g., fairness). All of the above coupling effects demonstrate the need to consider higher-layer QoS (Quality of Service) issues jointly with the PHY layer issues such as data rate. Note that cross layer design does not mean getting rid of protocol layers, or integrating all layers. However, since there is direct coupling between the PHY layer and upper layers, this inter-layer coupling can be exploited for further optimization, instead of a sub-optimal solution resulting from several local optimizations in each layer of the protocol stack. We must consider carefully which layers should respond to channel variations, and what layers should be jointly designed or optimized [15].

Our solution serves to provide some insights regarding the design of cross layer paradigms. We are trying to demonstrate that significant performance gains can be achieved by jointly considering several layer issues in an integrated framework. Our goal is to utilize the multi-rate PHY to produce a rate adaptive MAC, and exploit the multiuser diversity to produce a MAC assuming a multiuser PHY, so that the MAC can take advantage of both time and location dependent diversities to address the channel vulnerability and the HOL blocking, while considering fairness issues among multiple users.

We focus primarily on the downlink model, a situation where the arriving traffic is buffered at the AP until it is transmitted, and resources are allocated as a function of each flow's channel state. The AP acts as a centralized controller and makes all resource allocation decisions. It maintains a set of queues for each back logged flows at the LLC layer, as shown in Fig. 1. We assume saturated traffic here, so at any time, the queues are filled to a certain extent and packets arriving when queues are full are dropped. In our WFS-ARC scheme, LLC/MAC/PHY layers work cooperatively to optimize the wireless resource (airtime in this paper) allocation, while satisfying the fairness requirement. We note that the framework presented here is quite general and can accommodate a variety of physical layer models and wireless network topologies.

There are three functional blocks in the proposed cross layer framework (Fig. 1). The rate controller at the MAC layer directly resides above the PHY layer and adjusts the data rate parameter used by the PHY layer. We note that this is reasonable, since in modern network cards, the interface typically exports some tunable parameters, such as the minimum contention window size, retry limit, data rate and so on, to the MAC layer as controllable variables. Users can choose proper settings for these parameters to achieve the desired QoS requirements. Obviously, the data rate plays a critical part in throughput performance. The goal of the rate controller is to dynamically adjust the data rate according to the channel quality variations and therefore, improve the channel efficiency. It collects statistics of ACKs and retry counts, for different downlink nodes. Based on the collected information, the AP estimates the downlink channel quality and selects the data rate which is the "best" among all possible rates for each active client.

Suppose there are N users in total in the system. Each user is assigned a weight w_i. The weight vector $\vec{w}$ can be constant or dynamically updated. We define the Contention Period (CP), as the time duration lasting for a frame's transmission and reception, including the backoff, deferring, and ACK duration. The scheduling decision is made at the beginning of each CP. The rate controller collects the transmission statistics, such as ACKs and retry counts to measure the channel qualities. The output of the rate controller is a vector of data rates, $\vec{r}(k) = [r_1(k), r_2(k), \ldots, r_N(k)]$, for active nodes at the contention period k. Based on the rate vector, the expected goodput vector during k_{th} CP $\vec{G}(k) = [G_1(k), G_2(k), \ldots, G_N(k)]$ can be computed approximately using the formulas (1), (2) and (3):

$$G_i(k) = \frac{Data\,payload\,length \quad s_i(k)}{ETT_i(k)}, \tag{1}$$

$$\begin{aligned} ETT_i(k) = {} & Backoff + DIFS + PHY\,ov + MAC\,ov \\ & + MAC\,data\,duration + SIFS + ACK + \sigma, \end{aligned} \tag{2}$$

$$MAC\,data\,duration = \frac{MAC\,data\,payload\,length}{PHY\,data\,rate \quad r_i(k)}. \tag{3}$$

The goodput vector $\vec{G}(k)$, together with the control vector $\vec{c}(k) = [c_1(k), c_2(k), \ldots, c_N(k)]$ are feed to the scheduling block, which is positioned on top of the MAC layer. The scheduler selects the most promising user to transmit a packet to, based on the scheduling policy. The scheduling decision is given by the vector $\vec{p}(k) = [p_1(k), p_2(k), \ldots, p_N(k)]$. Define $p_i(k)$ $(1 \le i \le N)$ as follows (i.e., one by one scheduling):

$$p_i(k) = \begin{cases} 1 & user_i\,is\,scheduled\,in\,k_{th}\,CP, \\ 0 & otherwise. \end{cases} \tag{4}$$

Following the scheduling decision, a packet is transmitted to the wireless channel, and the transmission result is collected at the rate controller for rate adaptation in the next CP. The decision vector $\vec{p}(k)$ also serves as an input to the weight adjustor, which adapts the control vector $\vec{c}(k)$, so that the long term expected goodput of each user approaches the assigned weight. We discuss these three blocks in detail in the following sections.

B. Adaptive Rate Control Algorithm

In this section, we propose a simple but efficient ARF-like rate control algorithm implemented in the rate controller. The basic ideas included are: 1) Multi-rate retransmission; 2) AIMD (Additive Increase Multiplicative Decrease) adjustment of the success threshold ST.

The multirate retransmission mechanism is used to react quickly to the short-term channel variations and reduce fluctuations in the long-term data rate. The idea is that, each next-hop FIFO $queue_i$ is associated with a long-term rate r_i. When the HOL packet of $queue_i$ is being transmitted, the rate controller fills in the retry series of the transmission descriptor for the packet with the current value of r_i and lower retry rates than r_i accordingly. If the first attempt fails, retries at the same or lower rates automatically take place without changing r_i. Therefore, small-scale variations can be quickly resolved to avoid premature long-term rate adjustments.

We adjust the long term data rate r_i based on a threshold scheme similar to ARF. It has been shown that a fixed value of ST is very sensitive to the changing speed of the link quality [9]. Under stable channel conditions, the rate adaptation algorithm can finally converge to the best rate; however, after that, it periodically attempts to transmit at a higher data rate, leading to periodical failures. Intuitively, it is better to increase the ST value so that we can minimize the undesired rate increments for slow fading channels; on the other hand, when the channel condition is fluctuating rapidly, it is critical to locate an optimal rate and stay there as long as possible.

In our algorithm, we propose a dynamic ST setting process, based on the transmission history. If the previous ST has been reached, which is followed by a rate increment, we conclude that the channel quality is improving. Since the data rate is already higher now, we increase the ST to stay in the current high rate as long as possible. If a transmission failure immediately follows the rate increment, we also increase the ST to discourage premature rate increments. For the failure threshold FT, a high value may degrade the system performance, since too many transmissions failed before the data rate is reduced. On the contrary, it is very likely that a single failure is due to collision, which can be handled by the multi-rate retransmission strategy, while bursty failures are due to stable deterioration in channel quality. Therefore, ARC requires that once 2 consecutive packets are retransmitted, the data rate is reduced as well as the ST, to encourage potential rate increments. Specifically, the ST value is adjusted in the interval of $[ST_{min}, ST_{max}]$, using a similar way as the one of TCP's AIMD congestion window adjustment. Algorithm 1 illustrates the ideas presented above. Note that the data rate in the algorithm refers to a PHY mode index. For 802.11a, the PHY modes are indexed from 0 (6 Mbps) to 7 (54 Mbps).

C. Optimization Model for Weighted Fair Scheduling

Our focus in this section is on systems where packets destined for each downlink user are queued at the AP and all

Algorithm 1 Adaptive Rate Control

```
ARC with Multi-rate Retransmission and AIMD ST Adjustment.
while receiving a packet P dequed from one of the FIFO queues do
  NextHop = daddr(p);
  P.dataRate = NextHop.dataRate;
  send(P);
  if recvACK() then
    if retry == 0 then
      Success++; Failure = 0; Recovery = 0;
      if Success ≥ ST then
        NextHop.dataRate++;
        Success = 0;
        ST+ = α; Recovery = 1;
      end if
    else if retry > 0 then
      Success = 0; Failure++;
      oldRate = NextHop.dataRate;
      if Failure ≥ FT || Recovery == 1 then
        NextHop.dataRate−−;
        Failure = 0;
      end if
      if Recovery == 1 then
        ST+ = α;
      else if oldRate < NextHop.dataRate then
        ST/ = β;
      end if
      Recovery = 0;
    end if
  end if
  if ACKtimeout() then
    P.dataRate = lookupMultirateRetry();
    if retry < retryLimit then
      retry++; retransmit(P);
    else
      Error++; Success = 0; Failure = 0;
      Recovery = 0; drop(P);
    end if
  end if
end while
```

flows have infinite backlogs of bits. The task of the scheduler at the AP is to schedule a packet transmission to one of the most profitable users, in the hope that such a transmission maximizes the system performance. One greedy scheduling policy is to always choose the best user (i.e., the user with the best channel quality). However, in the case where one user always having a good channel, other users suffer from starvation. Therefore, we need to consider a general tradeoff model which maximizes an utilization (objective) function subject to the assigned fairness constraints.

1) Packet Scheduler: Following the framework described in Section III, we present our packet scheduler formulation. Without loss of generality, the aggregate goodput (application layer throughput) is used as the system performance metric. Consider N users in total (not including the AP) in the system. Each is based on the same PHY layer, and therefore, has the same rate set $R = \{R_1, R_2, \dots R_M\}$. The objective of the scheduler is to choose one $user_i$ out of N at the beginning of the k_{th} CP, based on the output $\vec{r}(k) = [r_1(k), r_2(k), \dots, r_N(k)]$ of the rate controller, where $r_i(k) \in R$ is the current data rate of $user_i$. The scheduler output $\vec{p(k)}$ (defined in (4)) states that if $user_i$ is scheduled, $p_i(k) = 1$; otherwise, $p_i(k) = 0$. Since only one user is scheduled in one CP, it follows that $\sum_{i=1}^{N} p_i(k) = 1$. The expected goodput of the scheduled $user_i$ in the k_{th} CP is $g_i(k)$. Note that $g_i(k) = g(r_i(k), s_i(k), x_i(k))$, where $s_i(k)$, $x_i(k)$ are the packet length and channel condition respectively, of the $user_i$ during the k_{th} CP. However, as we do not have the exact knowledge of $x_i(k)$, we can only compute the goodput using (1) approximately. Following this, the goodput in the current CP can be represented as $G(k) = \sum_{i=1}^{N} p_i(k)g_i(k)$. In the long run, the system goodput is given by:

$$G = E[G(k)] = E[\sum_{i=1}^{N} p_i(k)g_i(k)].$$

Alternatively, let $G_i = E[p_i(k)g_i(k)]$ be the long term goodput of $user_i$. The system goodput can be rewritten as:

$$G = \sum_{i=1}^{N} E[p_i(k)g_i(k)] = \sum_{i=1}^{N} G_i.$$

The optimization model is to maximize the system goodput, given the assigned weight vector $\vec{w}$, i.e.,

$$\max \quad G = \sum_{i=1}^{N} G_i, \quad s.t. \tag{5}$$

$$\frac{G_i}{w_i} = \frac{G_j}{w_j}, \quad \forall 1 \le i, j \le N. \tag{6}$$

Using a similar technique as the one in [17], (5) and (6) can be transformed to the following equivalent problem:

$$\max \quad Z = \frac{G_i}{w_i}, \quad (1 \le i \le N). \tag{7}$$

Since all $(\frac{G_i}{w_i})'s$ are identical, once again, the objective function in (7) is equivalent to:

$$\max \quad Y = (\sum_{i=1}^{N} c_i w_i)Z, \quad where$$

$$c_i \ge 0, \quad Z = \frac{G_i}{w_i}, \quad (1 \le i \le N).$$

It follows that the above problem can be rewritten as:

$$\max \quad Y = \sum_{i=1}^{N} c_i G_i, \quad s.t. \tag{8}$$

$$\frac{G_i}{w_i} = \frac{G_j}{w_j}, \quad \forall 1 \le i, j \le N. \tag{9}$$

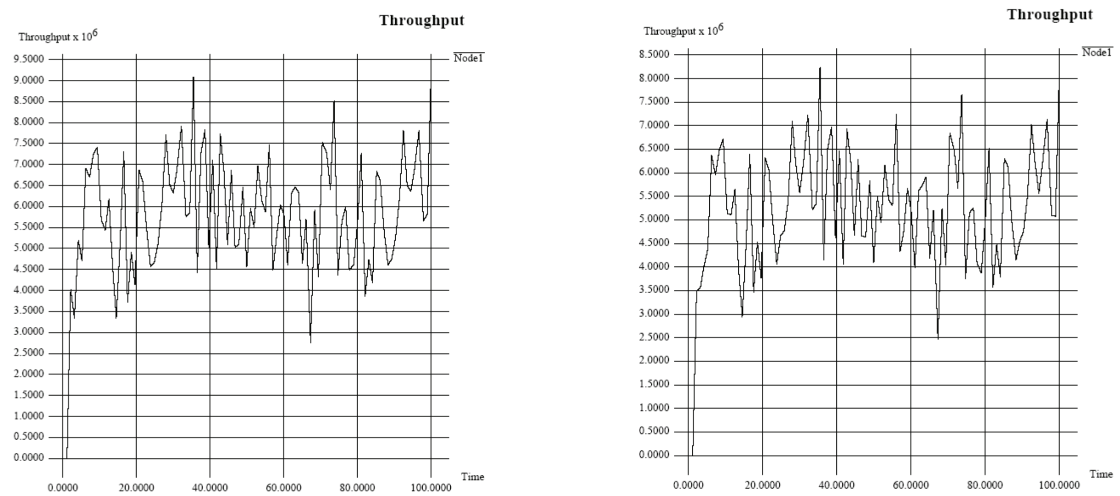
Therefore, the original problem defined in (5) and (6) is equivalently transformed to the optimization problem defined in (8) and (9), where $\vec{c}$ is a vector with non-negative values. Suppose we are able to choose an appropriate control vector $\vec{c}^*$, such that it can drive $(G_i)'s$ to satisfy (9), then we only need to consider the optimization model

$$\max \quad Y = \sum_{i=1}^{N} c_i^* G_i. \tag{10}$$

Based on the above analysis, define the scheduling policy at the LLC layer of the AP to be

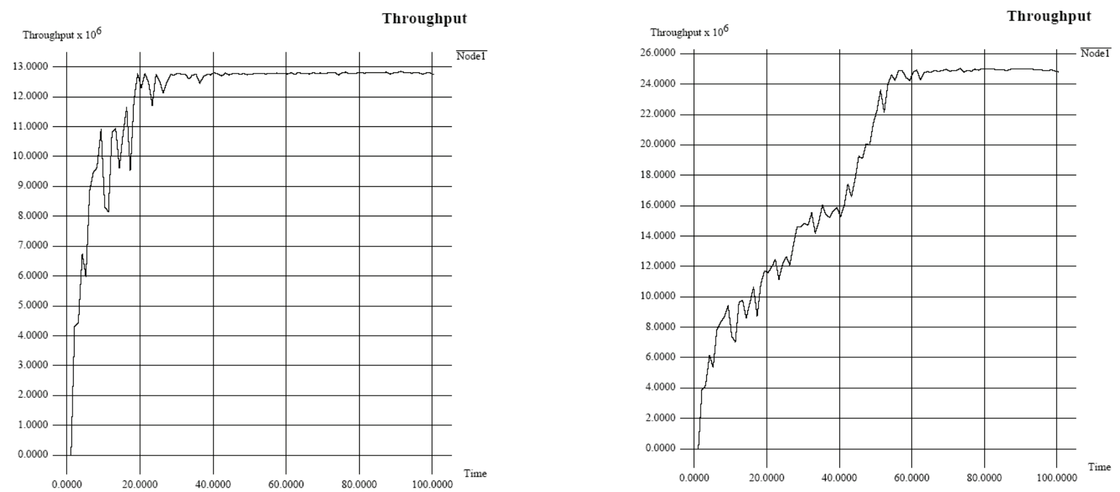
$$S^*(\vec{G}(k)) = \arg\max_i c_i^* G_i(k), \tag{11}$$

such that in each CP, the weighted goodput is maximized. Obviously, the policy $S^*(\vec{G}(k))$ yields a solution to the problem defined in (10).



(a) Goodput of 802.11a/18 Mbps (b) Goodput of 802.11a/ARC

Fig. 2. AP sends CBR traffic to the single static user far away from it. In Fig. 2(a), the user transmits at the optimal data rate 18 Mbps, assuming the channel quality is known a prior. In Fig. 2(b), the user transmits with ARC enabled. ARC can fast tune to the optimal rate 18 Mbps. No a prior knowledge of the channel quality is required.



(a) Goodput of 802.11a/18 Mbps (b) Goodput of 802.11a/ARC

Fig. 3. AP sends CBR traffic to the single user moving towards it. The channel quality is improving. In Fig. 3(a), the goodput gain is limited without a rate control scheme. In Fig. 3(b), ARC increases the data rate responsively until the maximum data rate is reached.

2) Weight Adjustor: Note that we introduce the control vector $\vec{c}^*$, which is dependent on the distribution of $\vec{G}$, to approach the fairness constraints. However, since the full knowledge of the channel conditions is not known apriori, a fixed $\vec{c}^*$ obtained at the very beginning and leading to the optimal value is not available. Rather, we define an online updating process to dynamically adjust the control vector $\vec{c}^*$, which is the task of the weight adjustor introduced here. In a special case where c_i^* is always 1 for all $i's$, the scheduling policy reduces to the greedy one mentioned previously, i.e., the one that always schedules the user with the largest goodput in the current CP computed by (1).

The weight adjustor estimates the control vector using a standard stochastic approximation algorithm [12] [17]. Stochastic approximation was introduced by Robbins and Monro (1951) for finding a solution to the equation $f(x^*) = 0$ when f is observed with error

$$y^k = f(x^k) + \xi^k,$$

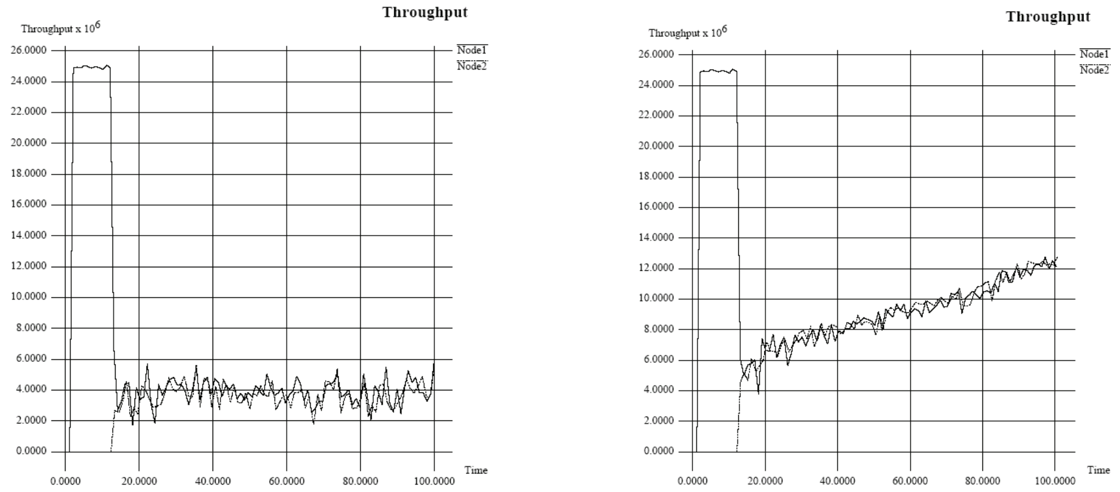
where $\{\xi^k\}$ is a sequence of random errors. In this paper we assume $\{\xi^k\}$ are independent with mean zero and bounded variances. The idea of the Robbins-Monro algorithm is to iterate the sequence

$$x^{k+1} = x^k - a^k y^k$$

until convergence. The convergence property of the Robbins-Monroe algorithm under proper conditions can be found in [18] and [19].

In our case, define

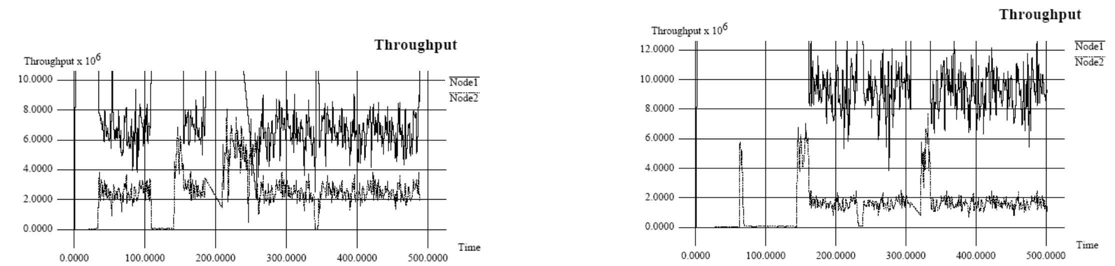
$$f_i(\vec{c}) = \frac{G_i}{\sum_{i=1}^N G_i} - \frac{w_i}{\sum_{i=1}^N w_i}, \quad i = 1, ..., N.$$



(a) Goodput of 802.11a/ARC: two static users

(b) Goodput of 802.11a/ARC: one static, one moving

Fig. 4. AP sends CBR traffic to two users. In Fig. 4(a), there are two static users, $user_1$ with a good channel and $user_2$ with a bad channel. However, both achieve approximately the same goodput performance, despite of different data rates. In Fig. 4(b), there is a static $user_1$ with a good channel, and a distant $user_2$ moving towards the AP. The goodput performance achieved by these two users, although increasing, is still approximately the same.



(a) Goodput of 802.11a/WFS-ARC: weight ratio (2.4:1)

(b) Goodput of 802.11a/WFS-ARC: weight ratio (5:1)

Fig. 5. AP sends CBR traffic to two static users. In Fig. 5(a), $user_1$ with a good channel is assigned a weight 2.4, and $user_2$ with a bad channel is assigned a weight 1. The goodput ratio of the two users approaches the weight ratio (2.4:1). The aggregate goodput is increased by 50% compared with Fig. 4(a). In Fig. 5(b), $user_1$ with a good channel is assigned a weight 5, and $user_2$ with a bad channel is assigned a weight 1. The goodput ratio of the two users approaches the weight ratio (5:1). The aggregate goodput gain is even more compared with Fig. 4(a) and Fig. 5(a).

We use the stochastic approximation to generate a sequence of iterations $\vec{c}^1$, $\vec{c}^2$, ..., that converges to $\vec{c}^*$. In each iteration, the scheduling policy is given by:

$$S^k(\vec{G}(k)) = \arg\max_i c_i^k G_i(k).$$

Hence, we need an estimation of y^k at $f(\vec{c}^k)$. Note that in each CP, we have a noise observation of $f_i(\vec{c}^k)$, i.e.,

$$y_i^k = p_i(k) - \frac{w_i}{\sum_{i=1}^N w_i}, \quad i = 1, ..., N,$$

where $p_i(k)$ is defined in (4). The observation error in this case is:

$$E[\xi_i^k] = E[p_i(k) - \frac{G_i}{\sum_{i=1}^N G_i}] = 0.$$

Therefore, we can use the stochastic approximation algorithm to adaptively find $\vec{c}^*$ by

$$c_i^{k+1} = c_i^k - a^k y_i^k,$$

where the step size a^k should be appropriately chosen to converge to zero, e.g., $a^k = 1/k$.

Algorithm 2 shows the structure of our weighted fair scheduling scheme:

Algorithm 2 Weighted Fair Scheduling

Weighted Fair Scheduling: $\frac{G_i}{w_i} = \frac{G_j}{w_j}, \quad \forall 1 \le i, j \le N.$
while receiving a packet P from the upper layer **do**
    $Enque(P)$;
    if not blocked **then**
        return $P = Dequeue()$;
    end if
end while

void $Enque(P)$:
$qid = findEnqueID(P.daddr())$;
$flowQueue[qid].enque(P)$;

Packet* $Dequeue()$:
if has not dequed any packet **then**
    $init\_wi()$;
else
    $update\_wi()$ using stochastic approximation;
end if
for all $NextHop_i$ **do**
    compute $G_i = E[G_i(r_i, s_i)]$ by (1);
    $WeightedG[i] = c_i G_i$;
end for
$qid = selectBest(WeightedG[])$;
return $flowQueue[qid].deque()$;

TABLE I
MAC AND PHY PARAMETERS OF STANDARD IN SIMULATION

CW_{min}	15
CW_{max}	1023
SlotTime	9 μs
DIFSTime	34 μs
SIFSTime	16 μs
BasicRate	6 Mbps
DataRate	6, 9, 12, 18, 24, 36, 48, 54 Mbps
Preamble Duration	16 μs
PLCP Header	4 μs
ACK Payload	14 Bytes
MAC Header	28 Bytes
ARC ST_{min}	8
ARC ST_{max}	50
ARC FT	2
ARC Linear Increase Factor α	16
ARC Mutiplicative Decrease Factor β	2

IV. Performance Evaluation

In this section, we present the performance evaluation of the proposed WFS-ARC framework in ns-2 [20]. Table I summarizes some parameters used to characterize the MAC/PHY layers of the IEEE 802.11a. For more realistic consideration, we make necessary changes to the original ns-2 MAC/PHY layers and incorporate the Ricean Fading channel model introduced in [21]. In our simulation, CBR traffic is used to study the saturation behavior of both the WFS-ARC algorithm and the 802.11a with a single FIFO queue. We run each simulation for at least 100 seconds with data payload size of 1000 Bytes. Each reported result is averaged over at least 10 runs. The goodput performance is measured in different scenarios.

A. ARC Performance

In this section, we consider the single receiver case and evaluate the performance of the ARC algorithm. In the first case, suppose the only static user is far away from the AP. With a plain MAC without rate adaptation, the AP always sends packets at a specified data rate, which may either cause lots of transmission errors or a waste of bandwidth due to a too low data rate. In our scenario, the optimal rate is 18 Mbps (since no packets sent at 24 Mbps can be received and 12 Mbps leads to lower goodput), as shown in Fig. 2(a). With our ARC scheme, the AP can tune to the best data rate (Fig. 2(b)). As expected, the throughput is a little lower than the optimal one, due to the delay overhead of maintaining transmission statistics, and the occasionally wrong decisions of the data rates to use.

Now suppose the single user is moving towards the AP. Therefore, the channel condition is improving and the packet error rate at 18 Mbps is lower than that of the previous case. Without ARC, the maximum goodput is approximately 12.8 Mbps only (Fig. 3(a)). With ARC, the data rate can reach the maximum 54 Mbps and the ARC algorithm can quickly tune to the improving channel quality, as shown in Fig. 3(b).

B. WFS-ARC Performance

Here we study the performance of 802.11a only with ARC enabled and 802.11a with the whole WFS-ARC framework. We first give a simple two-user scenario, where $user_1$ is close to the AP and the other $user_2$ is far away from it. The AP starts the flow of $user_1$ first and later of $user_2$. With ARC, although the AP can always transmit to $user_1$ at the highest data rate 54 Mbps and can only send at 18 Mbps to $user_2$, the goodput of the two users are approximately the same (Fig. 4(a)). This can be explained by the fact that the two users have roughly the same opportunity to access the channel, but the slower user consumes more radio time and leads to very low channel utilization. This phenomenon is more evident in another two-user case, where $user_2$ is moving towards the AP. The goodput of the two users are still approximately the same throughout the time (Fig. 4(b)). However, both are increasing since the channel quality of $user_2$ is now improving, and the time fraction that it takes up is less compared to the previous case.

In Fig. 5(a), the weights for $user_1$ and $user_2$ in previous two-static-user case (Fig. 4(a)) are set to be 2.4 and 1 respectively, taking account of different channel qualities. We can see that our framework converges quickly to the assigned weights and the system goodput is improved by 50% while not sacrificing fairness too much. The improvement is even more if we set the weights to be 5 and 1 for $user_1$ and $user_2$ respectively (Fig. 5(b)). Both results reflect the well-known throughput and fairness tradeoff: the higher the system throughput is, the lower the fairness is.

Fig. 6 plots the aggregate goodput of both schemes as a function of the number of users, assuming saturated downlink CBR traffic. For 802.11/ARC, the system performance is greatly dependent on the behavior of the "worst user", which has the worst channel quality. For 802.11/WFS-ARC, with properly assigned weights (e.g., the weights either satisfying the QoS requirement, or dynamically updated according to the channel quality), significant system goodput gain can be achieved.

V. Conclusion

In this paper, we propose a cross layer design for the IEEE 802.11 WLANs, WFS-ARC, for joint rate control and packet scheduling, so that the LLC/MAC layers can exploit the multi-rate PHY layer capability and the multiuser diversity. The problem is modelled to maximize the system goodput with a rate adaptive MAC layer, while satisfying the assigned fairness constraints. We study the saturation behavior in different cases and the simulation results demonstrate that through well-designed cooperation of different layers, superior performance gain can be achieved. This scheme can be easily adopted by the sate-of-the-art IEEE 802.11 AP products, since it can be implemented in the device driver and no modification to the hardware or the standard is required. It should also be easy to extend this framework to Ad Hoc networks.

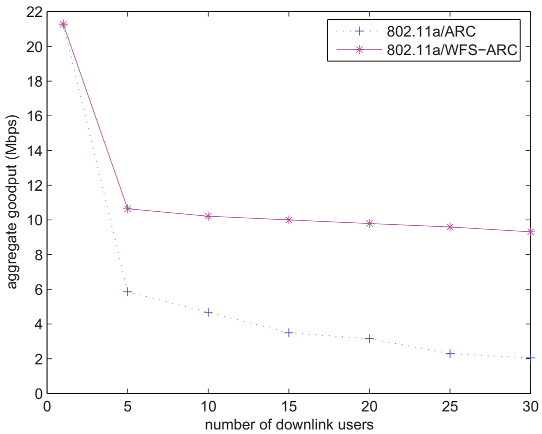


Fig. 6. Aggregate goodput: 802.11a/ARC vs. 802.11a/WFS-ARC as a function of the number of users. Significant system goodput gain can be achieved with the WFS-ARC framework.

References

[1] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std 802.11-1999, Aug. 1999.

[2] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE Std 802.11b-1999, Sep. 1999.

[3] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*, IEEE Std 802.11a-1999, Sep. 1999.

[4] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2001.

[5] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. IEEE INFOCOM*, Mar. 2003, vol. 2, pp. 836–843.

[6] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band*, IEEE Std 802.11g-2003, June 2003.

[7] J. Geier, *Wireless LANs*, 2nd ed. Indianapolis, IN: Sams Publishing, July 2001.

[8] A. van der Vegt, "Auto Rate Fallback Algorithm for the IEEE 802.11a Standard," Utrecht University, Tech. Rep., 2002.

[9] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. ACM MOBICOM*, July 2001, pp. 236–251.

[10] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "OAR: An opportunistic auto-rate media access protocol for ad hoc networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 39–53, Jan. 2005.

[11] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling," in *Proc. IEEE INFOCOM*, Mar. 1996, pp. 1133–1140.

[12] X. Liu, E. Chong, and N. Shroff, "Transmission scheduling for efficient wireless utilization," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 776–785.

[13] L. B. Jiang and S. C. Liew, "An adaptive round robin scheduler for head-of-line-blocking problem in wireless LANs," in *Proc. IEEE WCNC*, Mar. 2005, pp. 1219–1224.

[14] D. Pong and T. Moors, "Fairness and capacity trade-off in IEEE 802.11 WLANs," in *Proc. IEEE LCN*, Nov. 2004, pp. 310–317.

[15] "ICC panel on defining cross-layer design in wireless networking," May 2003 [Online]. Available: `http://www.eas.asu.edu/~junshan/ICC/JunshanICC.ppt`

[16] R. A. Berry and E. M. Yeh, "Cross-layer wireless resource allocation," *IEEE Trans. Signal Processing*, vol. 21, pp. 59–68, Sep. 2004.

[17] Y. Liu and E. Knightly, "Opportunistic fair scheduling over multiple wireless channels," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 1106–1115.

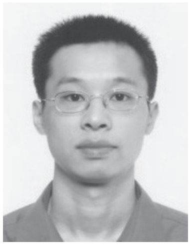
[18] H. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

[19] I. J. Wang, E. K. P. Chong, and S. R. Kulkarni, "Weighted averaging and stochastic approximation," *Math. Control Signals Syst.*, vol. 10, no. 1, pp. 41–60, Mar. 1997.

[20] NS2, *The network simulator2*, 2003 [Online]. Available: `http://www.isi.edu/nsnam/ns`

[21] Antenna & Radio Communications (ARC) Group, Carnegie Mellon University, *Additions to the NS network simulator to handle Ricean and Rayleigh fading*, 2000 [Online]. Available: `http://www.ece.cmu.edu/wireless(underDownloads)`

Qiuyan Xia (S'05) received her B.S. degree in Computer Science from the Nanjing University, Nanjing, China, in 2003. She is currently working towards the Ph.D degree in Computer Science and Engineering, at the Hong Kong University of Science and Technology, Hong Kong, China. Her research interests include cross layer design and optimization, adaptive algorithms, link adaptation, resource management for the IEEE 802.11 WLANs, and quality of service provision.



Xing Jin (S'04) received the B.Eng. degree in computer science and technology from Tsinghua University, China, in 2002. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. Mr. Jin was awarded the Microsoft Research Fellowship in 2005. He is a junior editor of the *Journal of Multimedia* since 2006. His research interests include overlay multicast with applications and QoS issues, Internet topology inference, end-to-end measurements and peer-to-peer streaming.

Mounir Hamdi (S'90-M'91-SM'06) received the B.S. degree in Electrical Engineering from the University of Louisiana in 1985, and the MS and the PhD degrees in Electrical Engineering from the University of Pittsburgh in 1987 and 1991, respectively. He has been a faculty member in the Department of Computer Science at the Hong Kong University of Science and Technology since 1991, where he is now Full Professor of Computer Science, Director of the Computer Engineering Program, and Director of the Master of Science in Information Technology. He is a member of the University Senate and University Council. From 1999 to 2000 he held visiting professor positions at Stanford University, USA, and the Swiss Federal Institute of Technology, Lausanne, Switzerland. His general area of research is in high-speed wired/wireless networking in which he has published more than 220 research publications, received numerous research grants, and graduated more 20 PhD/Master students. In addition, he has frequently consulted for companies in the USA, Europe and Asia on high-performance Internet routers and switches as well as high-speed wireless LANs.

Dr. Hamdi is/was on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS, *IEEE Communication Magazine*, *Computer Networks*, *Wireless Communications and Mobile Computing*, and *Parallel Computing*. He was a guest editor of *IEEE Communications Magazine*, guest editor-in-chief of two special issues of the IEEE JOURNAL ON SELECTED AREAS OF COMMUNICATIONS, and a guest editor of *Optical Networks Magazine*, and has chaired more than 7 international conferences and workshops including the IEEE International High Performance Switching and Routing Conference, the IEEE GLOBECOM/ICC Optical networking workshop, the IEEE ICC High-Speed Access Workshop, and the IEEE IPPS HiNets Workshop. He is/was the Chair of the IEEE Communications Society Technical Committee on Transmissions, Access and Optical Systems, and Vice-Chair of the Optical Networking Technical Committee, as well as member of the ComSoc technical activities council. He is/was on the technical program committees of more than 120 international conferences and workshops. He received the best paper award at the International Conference on Information and Networking in 1998 out of 152 papers.

In addition to his commitment to research and professional service, he is also a dedicated teacher. He received the best 10 lecturers award and the distinguished engineering teaching appreciation award from the Hong Kong University of Science and Technology, and various grants targeted towards the improvement of teaching methodologies, delivery and technology. He is a member of IEEE and ACM.