

Bandwidth-Guaranteed Fair Scheduling with Effective Excess Bandwidth Allocation for Wireless Networks

Yaxin Cao, Ka-Cheong Leung, *Member, IEEE*, and Victor O. K. Li, *Fellow, IEEE*

Abstract—Traffic scheduling is key to the provision of quality of service (QoS) differentiation and guarantees in wireless networks. Unlike its wireline counterpart, wireless communications pose special channel-specific problems such as time-varying link capacities and location-dependent errors. These problems make designing efficient and effective traffic scheduling algorithms for wireless networks very challenging. Although many wireless packet scheduling algorithms have been proposed in recent years, issues such as how to improve bandwidth efficiency and maintain goodput fairness with various link qualities for power-constrained mobile hosts remain unresolved. In this paper, we devise a simple wireless packet scheduling algorithm called bandwidth-guaranteed fair scheduling with effective excess bandwidth allocation (BGFS-EBA), which addresses these issues. Our studies reveal that BGFS-EBA effectively distributes excess bandwidth, strikes a balance between effort-fair and outcome-fair, and provides a delay bound for error-free flows and transmission effort guarantees for error-prone flows.

Index Terms—Fairness, mobile communications, performance guarantees, quality of service (QoS), traffic scheduling, wireless networks.

I. INTRODUCTION

AS a vital component of quality of service (QoS) support, traffic scheduling algorithms are very important for both wireline and wireless packet-switched data networks. However, designing a good wireless packet scheduling algorithm is quite challenging, because wireless communications pose special problems that do not exist in wireline networks. Packet transmission in wireline networks enjoy very low error rate. However, wireless channels are more error-prone and suffer from interference, fading, and shadowing. As a result, the quality of a wireless communication link varies from time to time. Besides time-dependent failures, wireless link capacity is location-dependent. A base station can typically communicate with multiple mobile hosts simultaneously. At a certain time, different users may experience different link qualities due to different locations and moving speeds. Existing wireline packet scheduling algorithms cannot be applied directly in wireless networks, because they do not take such link variations into consideration.

Manuscript received August 18, 2005; revised February 12, 2007; accepted April 3, 2008. The associate editor coordinating the review of this paper and approving it for publication was Q. Zhang.

Y. Cao is with Broadcom Corporation, 100 Matawan Road, Suite 210, Matawan, NJ 07747-3913, USA (e-mail: yaxinc@broadcom.com).

K.-C. Leung and V. O. K. Li are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, China (e-mail: kcleung@ieee.org, vli@eee.hku.hk).

Digital Object Identifier 10.1109/TWC.2008.05650.

Good packet scheduling algorithms in wireless networks should opportunistically seek to exploit channel conditions to achieve better network performance [12]. For example, swapping service opportunities among traffic flows may be used to improve channel efficiency [4]. When a flow is scheduled to transmit but its link quality is relatively bad, it may give up its current service opportunity to another flow with a better link. To ensure fairness, the flow which gives up its service opportunity should be compensated for this temporary loss of service when its link recovers. However, determining how to compensate for the flow is not an easy task. Since bandwidth wastage due to packet losses, errors, and coding overheads are unavoidable in wireless links, the bandwidth or transmission capacity, known as effort, allocated to a flow may not equal to the actual goodput¹ achieved. A wireless packet scheduling algorithm must strike a balance between improving bandwidth efficiency and maintaining fairness, in terms of the goodput attained by users with different link qualities.

In wireless networks, mobile hosts are generally power-constrained. A good packet scheduling algorithm should be designed such that the number of scheduling related control messages exchanged with the mobile hosts is minimized. For example, a packet scheduling algorithm that requires the arrival time of every uplink packet to compute the transmission schedule is probably not a good choice, since the mobile hosts would consume a great deal of power for transmitting the packet arrival information to the base station. In addition, a packet scheduling algorithm should be of low complexity, thus ensuring a short execution time to accommodate the scheduling of real-time multimedia traffic with stringent timing requirements.

A. Our Contributions

The focus of this work, first described in [5], is to devise a simple, low-complexity opportunistic wireless packet scheduling algorithm, known as bandwidth-guaranteed fair scheduling with effective excess bandwidth allocation (BGFS-EBA). The proposed algorithm aims to account for all the above issues and is capable of providing QoS guarantees. We consider a time-slotted system in which the communication channel is shared at different time slots among the users. BGFS-EBA assigns each packet with a deadline and normally schedules

¹The goodput of a flow is defined as the data transmission rate realized for that flow.

a packet with the smallest deadline for transmission. BGFS-EBA permits a flow leading its target goodput to give up its current service opportunity to a lagging flow so as to strike a balance between effort-fair and outcome-fair. A scheduler is said to be effort-fair if it allocates the transmission capacity to each flow in proportion to its assigned weight. When a scheduler distributes the channel bandwidth such that the achieved goodput of each flow is in proportion to its assigned weight, the scheduler is said to be outcome-fair.

Flows with better links, which are able to meet target goodputs, are thus guaranteed to be served no more than their target rates if there exists a lagging flow. In the long run, flows with poor links, which are not able to meet their target rates, are guaranteed the minimum transmission bandwidths based on their preset goodput thresholds. Excess bandwidth is therefore allocated to compensate flows with inferior links. Our studies, based on a two-state probabilistic link model, reveal that, when compared with other existing wireless traffic scheduling algorithms such as channel-condition independent packet fair queueing (CIF-Q) [17], BGFS-EBA is a simple algorithm that can effectively distribute the excess bandwidth, strike a balance between effort-fair and outcome-fair, and provide a delay bound for error-free flows and transmission effort guarantees for error-prone flows. An error-free flow and an error-prone flow are flows in which packets are sent on an error-free link and an error-prone link, respectively.

B. Organization of the Paper

The remainder of this paper is organized as follows. Section II compares and discusses some existing wireless packet scheduling algorithms. The strengths and weaknesses of these algorithms are highlighted. Section III gives our system model to conceptualize our wireless scheduling problem. It also states the assumptions needed to simplify the subsequent discussion. Section IV presents our proposed wireless packet scheduling algorithm, BGFS-EBA, that can effectively distribute the excess bandwidth among traffic flows so as to strike a balance between effort-fair and outcome-fair. Section V examines the simulation results which compare the performance of CIF-Q and BGFS-EBA. Section VI concludes and discusses some extensions to our work.

II. RELATED WORK

One of the first scheduling algorithms that address the location-dependent problem and bursty errors in wireless networks is channel state dependent packet scheduling (CS-DPS) [3]. CS-DPS introduces the idea of deferring packet transmission when the channel for a flow is experiencing link errors and swaps the transmission opportunity with some other flow sent on a better link.

Idealized wireless fair-queueing algorithm (IWFQ) [15], channel-condition independent packet fair queueing (CIF-Q) [17], and wireless fair service algorithm (WFS) [16] have all been developed from wireline fair queueing. They all use an error-free fair queueing reference system and try to approximate the real service to the ideal error-free system. Based on the definitions in [17], a flow is said to be *leading*, *lagging*, or *satisfied* at any time instant if it receives more, less,

or the same amount of service as it would have received in the corresponding error-free system. In IWFQ, arriving packets are tagged with timestamps in virtual time just as in weighted fair queueing (WFQ) [18]. When it is ready to transmit a packet, the scheduler picks the one with the smallest finish time and the link on which the picked packet is sent is good. When the link of a flow recovers from link errors, the flow may have some packets with the set of smallest virtual tags. As a result, IWFQ is deficient in terms of short-term fairness as the scheduler may only serve this lagging flow exclusively for an extended period.

CIF-Q and WFS are designed to address the problem on short-term fairness. CIF-Q improves the scheduling fairness by associating compensation rate and penalty rate with a flow's allocated service rate and guaranteeing any flow sent on an error-free link with the minimal service rate. All the qualities are normalized by the service rate, giving fairer service allocation. WFS allows graceful degradation for leading flows and distributes the compensation bandwidth among lagging flows according to their lags. WFS also tries to decouple the rate and delay requirements. Instead of having only one weight associated with each flow in fair queueing, a flow is assigned with two weights, namely, rate weight and delay weight. Packets belonging to the same flow are drained into the scheduler according to the rate weight, but served according to the delay weight. WFS merely compensates a backlogged flow which has been denied service during its scheduled time slot if there is a packet belonging to another flow transmitted during this time slot.

Similar to IWFQ, every packet of each flow is timestamped based on the system virtual time to provide effort fairness for CIF-Q and WFS. Packets belonging to those flows which are feasible for transmission according to the compensation mechanism as described above are served in the increasing order of their timestamps. To ensure consistency, CIF-Q and WFS require that each mobile host with an uplink flow monitors the service progress of all other flows or the scheduler needs to broadcast the system virtual time periodically to the mobile hosts with uplink flows. The timestamping schemes used in CIF-Q and WFS are therefore impractical for uplink transmission. Furthermore, CIF-Q and WFS may fail to provide outcome fairness to traffic flows as the excess bandwidth is not distributed fairly based on the amount of goodputs achieved by lagging flows.

The underlying link models employed by all of the above algorithms are quite unrealistic. These algorithms assume that a wireless link is either in the *good* state or the *bad* state and link errors are bursty in nature. When a link is in the *good* state, a packet transmission is assumed to be always error-free. However, a transmission is assumed to always fail if a packet is sent on a link in the *bad* state. Thus, these algorithms would be ineffective in handling packet transmission when the link quality of a flow is generally bad.

All aforementioned wireless packet scheduling algorithms can provide some sort of effort fairness, but none of them can guarantee outcome fairness. Channel state independent wireless fair queueing (CS-WFQ) [11] and Effort-limited fair scheduling (ELFS) [7] attempt to achieve outcome fairness by increasing the fair queueing weights of the flows with larger

error rates. However, they do not consider swapping service opportunities for improving bandwidth efficiency.

Channel adaptive fair queueing (CAFQ) [21] attempts to provide outcome fairness by defining a new notion of fairness, known as channel-adaptive fairness (CAF), to account for the effect of throughput variation due to different states of a link. A parameter Δ is used to keep track of the difference of the service received by a real system and an idealized one. If a flow in the real system receives more service, it is a leading flow; otherwise, it is a non-leading flow. A lagging session can receive additional service through a virtual compensation session to help it catch up. However, CAFQ may fail to provide long-term outcome fairness since Δ is always incremented or decremented with the size of a packet, while the amount of data transmitted successfully varies and depends on the state of a link.

A forward (downlink) scheduling scheme for the code division multiple access high data rate (CDMA/HDR) system, known as the HDR scheduling algorithm [10], aims to maintain proportional fairness among traffic flows. This means that, when another scheduling algorithm is applied, an increase in the throughput of a certain user by $x\%$ over what the user can receive under the proposed algorithm causes a reduction of more than $x\%$ in the total throughput for all other users in the system. Best link lowest throughput first (BLOT) scheduling [9] extended the HDR scheduling algorithm so that a minimum service rate is guaranteed to each of the active flows concurrently maintained at a given user.

The problems of utility-based bandwidth allocation have been studied in [1], [13], [14], [20]. In [13], a set of stationary stochastic processes, which are functions of the channel conditions, are used to model the utilities or performance values achieved by the users in a time-slotted system. An opportunistic scheduling policy is to pick a user such that its selection function, which depends on the performance value achieved by that user, is the largest among all other users in the system. The selection function is constructed such that the expected fraction of time slots assigned to each user matches the pre-determined allocation value. The scheduling policy is optimal because it maximizes the sum of the expected utilities of all users in the system. Nevertheless, the proposed scheme assumes that all users are always backlogged in the system.

Wireless credit-based fair queueing (WCFQ) [14] generalizes the credit-based fair queueing used in wireline networks so as to provide throughput fairness among traffic flows. The current channel condition is taken into consideration as a parameter of the cost function, which is then used as part of the packet selection criteria. It has been shown [14] that WCFQ provides short-term and long-term statistical fairness guarantees for a continuous channel model. WCFQ does not assume that all flows are always backlogged in the system, but it does not provide any minimum bandwidth guarantees to flows nor compensation mechanisms for achieving goodput fairness.

A reward-based multiuser multirate time-varying channel scheduling problem has been addressed in [20]. The communication channel has a finite set of channel states such that each user can send at a certain user-specific rate when the channel is in a certain state. Each user is associated with a reward function in the current achieved throughput.

An optimization problem is formulated such that a chosen scheduling policy maximizes the total reward for all users in the system. The steepest descent or gradient method is applied to pick a user at any time slot for data transmission for solving the optimization problem. The work has been extended in [1] so that the minimum and maximum rate constraints for all users are incorporated. To realize the rate constraints in the long run, each user is associated with a token counter so that the token arrival rate depends on the sign of the token counter. Similar to [20], the gradient method is employed to choose a user for transmission at a time slot, where the selection function of each user is also a function of the token counter of that user.

Nevertheless, except for [14], all other mentioned utility-based bandwidth allocation schemes assume that all users are always backlogged in the system. The proposed scheduling policies may not be able to optimize the system performance for any flows with general traffic arrival patterns. In addition, all the proposed utility-based bandwidth allocation schemes do not have any compensation mechanisms for achieving goodput fairness. That is, the algorithms assume that, at any time slot, the user throughput and the data transmission rate are always the same. They offer no mechanisms to compensate any loss of goodput of a flow when a transmitted packet is corrupted or lost in the time slot.

The problem of scheduling over non-stationary multirate wireless channels has been investigated in [2]. A set of tracking algorithms are used to track the performance of a schedule maintained by an adversary in a measurement window. The tracking algorithm serves a user in a given time slot such that the corresponding counter value, based on the user and channel state, is the largest one. The counter is then decremented by one as the user is served at that slot. At the end of a measurement window, all counters are incremented with values determined from the solution of a linear programming problem for an optimal schedule at that measurement window. However, there is no provision of the minimum bandwidth guarantee nor any compensation mechanism offered by the proposed framework.

III. SYSTEM MODEL

N traffic flows, namely, Flow 1, Flow 2, ..., Flow N , share the wireless channel. Without loss of generality, we assume that the number of uplink and downlink flows are the same, and the uplink and downlink flows are scheduled independently. For each traffic flow, packets are expected to be reliably delivered in sequence. Thus, a transmitted packet remains at the head of the transmission queue or line (HOL) until it is successfully delivered over the link. We assume instant feedback on whether a transmission is successful.

In this work, we consider centralized scheduling of the uplink and downlink of a wireless communication channel shared by a set of traffic flows in time division multiple access (TDMA) manner. A central scheduler is installed at a base station. Time is divided into slots of equal duration. At the beginning of a time slot, the scheduler at the base station decides which flow is served during that time slot. For the uplink, the mobile host of the selected flow will transmit a

packet of that flow to the base station in that time slot. For the downlink, the base station sends a packet to the mobile host of the selected flow in that time slot. We assume that the uplink and downlink transmissions are carried out in two non-overlapping frequency bands so that they do not interfere with each other. For example, time division multiple access / frequency division multiple access (TDMA/FDMA) systems and time-slotted code division multiple access (CDMA) systems satisfy the requirements mentioned previously.

Denote by R the maximum data rate (in bytes per second) of the wireless communication channel. The value of R depends on the design of the wireless system using the channel, including, but not limited to, modulation technique and inter-symbol interference. Indeed, the data rate of the communication channel at any time also relies on the measured channel condition or state at that time. There are two types of communication links in the system, error-free and error-prone. An error-free link is just like a wireline link where the packet error probability is considered to be negligible. An error-prone wireless link is modelled by a two-state Markov model. The scheduler merely differentiates the quality of an error-prone link between two states, *good* or *bad*. The duration of each state of an error-prone link is exponentially distributed. The state of an error-prone link may be incorrectly estimated with a certain probability and is independent of each other. The average durations of the *good* and *bad* states of an error-prone link can differ from those of another link. Indeed, the channel model can be generalized to have a finite number of states. We outline how our proposed scheduling algorithm can be extended to better capture the adaptive behaviour of a multirate time-varying channel in Section IV-C.

A data packet can be sent in a time slot. A regular packet contains L bytes of data and does not contain redundant data for forward error correction (FEC). There is a much higher success probability to transmit a regular packet in the *good* state than in the *bad* state. When the scheduler decides to transmit a packet on a bad link and the HOL packet of the selected flow is a regular packet, the packet can be split into m low-rate packets. Each low-rate packet contains $\frac{L}{m}$ bytes of data drawn from a regular packet and an FEC code for error detection and correction. By doing so, the packet error probability can be maintained at an acceptable or desired level. Generally speaking, it is sufficient to take m to be two or three as inferred from our simulation study in Section V.

The scope of our work is to develop a simple, low-complexity opportunistic wireless packet scheduling algorithm, which aims to satisfy the following set of objectives:

- 1) Provide long-term fairness and goodput guarantees for flows with error-free links or occasional link errors.
- 2) Achieve high wireless channel utilization.
- 3) Minimize packet loss.
- 4) Provide delay bound for flows with error-free links or sporadic link errors.
- 5) Achieve low power consumption in mobile hosts.
- 6) Achieve medium algorithm complexity.

IV. OUR PROPOSED ALGORITHM: BGFS-EBA

In this section, we describe our proposed wireless packet scheduling algorithm, known as bandwidth-guaranteed fair

scheduling with effective excess bandwidth allocation (BGFS-EBA). The main philosophy of the proposed scheme is to guarantee a leading flow to be served no more than its target goodput if there exists a lagging flow, which is ensured to be allocated with the minimum transmission bandwidth based on its preset goodput threshold. This in turns strikes a balance between effort-fair and outcome-fair. BGFS-EBA can be employed for scheduling uplink and downlink packet transmissions, though uplink and downlink transmission activities are handled independently.

A. Algorithm Description

Each admitted Flow i has a target rate r_i , which is the desired average goodput. For a stable and realizable system, the sum of all target rates should not exceed the maximum data rate of the wireless communication channel R . The scheduler keeps track of the achieved goodput of each Flow i by maintaining g_i , the amount of data for that flow transmitted successfully. In an error-free system, the scheduler behaves as a latency-rate server [19]. The average service rate offered by the scheduler to a busy or backlogged period, over every interval starting at time θ_i from the beginning of the backlogged period, is at least equal to its reserved rate r_i . It has been shown [19] that the packet delay is bounded as long as a flow's service received in any backlogged period can be guaranteed such that:

$$W_i(t_1, t_2) \geq r_i \cdot (t_2 - t_1 - \theta_i) \quad (1)$$

where $W_i(t_1, t_2)$ is the effective service received by Flow i during $(t_1, t_2]$, r_i is the allocated rate of Flow i , t_1 is the start time of a backlogged period for Flow i , and θ_i is a non-negative constant, called *latency*, for a burst-constrained traffic flow with the average rate less than or equal to r_i . The latency at the scheduler represents the worst-case delay experienced by any packet of the traffic flow.

Instead of approximating the generalized processor sharing (GPS) service curve [18] as in channel-condition independent packet fair queueing (CIF-Q) [17] and wireless fair service algorithm (WFS) [16], we try to achieve the goodput target for each flow in any backlogged period as shown in (1) with $g_i(t_1, t_2]$ replacing $W_i(t_1, t_2]$, for all $i = 1, 2, \dots, N$. $g_i(t_1, t_2]$ denotes the amount of data of Flow i transmitted successfully during $(t_1, t_2]$. Similar to [17], Flow i is considered as *leading*, *lagging*, or *satisfied* at any time instant if it receives more, less, or the same goodput during the current backlog period as its target share r_i , for every $i = 1, 2, \dots, N$. Although we use the same terms as in CIF-Q, the reference system we are using is different from CIF-Q. CIF-Q compares the service received to the service a flow would receive in an error-free start-time fair queueing (SFQ) system, which is dependent on the traffic load. In our work, Flow i can be classified as *leading*, *lagging*, or *satisfied* by comparing $g_i(\tau, \tau + t]$ with the load-independent function $r_i t$, for all $i = 1, 2, \dots, N$.

Each traffic flow has its own queue. When a packet arrives, it is simply placed at the end of the queue for the flow the packet belongs to. No timestamping is performed for such a packet arrival. To decide which packet to transmit next, the

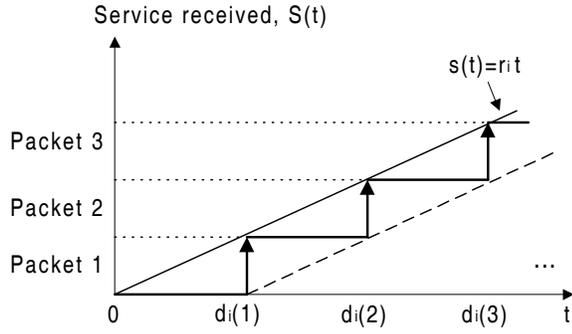


Fig. 1. Deadline calculation of a virtual flow.

scheduling process is performed in two phases. In the first phase, the scheduling decision is made on a fully-loaded error-free reference system. Apart from the real data flows in the network, the scheduler maintains a dummy flow which does not actually have packets to send but is only used to fill up the bandwidth. Suppose there are N real data flows, each of which has a target rate r_i , and the maximum data rate of the wireless communication channel is R . Denote by A the sum of the target rates for all served flows. The rate of the dummy flow will then be $R - A = R - \sum_{i=1}^N r_i$. However, the offered data rate of the wireless channel can be significantly smaller than R . For a stable system, the value of α is chosen such that the average offered data rate of the wireless channel is greater than αR , where $0 < \alpha < 1$. In our work, we expect that the proposed scheduler works with some admission control mechanisms so that a new flow is admitted for service only when the sum of all target rates for all flows, including the requested target rate for the new flow, is no more than αR . Alternatively, the target rates of all served flows are scaled so that $\sum_{i=1}^N r_i \leq \alpha R$.

The flows in the reference system are called *virtual flows*. In the reference system, all the virtual flows, including the dummy flow (marked as Flow $N + 1$), are assumed to be continuously backlogged. The imaginary packets of the virtual flows, which have the same fixed size as the real ones, are called *virtual packets*. The virtual packets are assigned with deadlines such that if all the virtual packets of Flow i are served before their deadlines, the service received by Flow i is no less than $r_i \cdot d_i(p)$ at every deadline of Packet p for Flow i , $d_i(p)$. The deadline calculation is further illustrated in Fig. 1. The arrows in the figure represent the deadlines. The service curve $s(t) = r_i t$ represents the service received by Flow i for a time period t if all virtual packets depart at their deadlines. The deadline of a virtual packet is thus the latest time that the virtual packet should depart for the flow to catch up with the service curve. The scheduler always selects a virtual packet with the smallest deadline.

In the second phase, the scheduler then decides whether a real packet of the traffic flow, of which the virtual packet is chosen in the first phase, should be transmitted. Since the virtual flows are always backlogged and the packet size is fixed, the deadline of Packet $p+1$ for Flow i can be derived from the deadline of Packet p for Flow i as $d_i(p+1) = d_i(p) + \frac{L}{r_i}$, where L and r_i are the packet size and the target rate for Flow i , respectively. Therefore, the scheduler only needs to

maintain one deadline for each Flow i , d_i , by which the head of the line (HOL) virtual packet should be served. After the HOL packet of a virtual flow is scheduled, no matter which real flow's packet receives the actual service in the second phase, the deadline of the virtual flow is updated as $d_i = d_i + \frac{L}{r_i}$. Since we have $\sum_{i=1}^{N+1} r_i = R$ (including the dummy flow), using such a deadline assignment and scheduling policy, it can be easily shown [6] that it is a schedulable system. That is, all virtual packets can meet their deadlines.

To determine how much Flow i is leading and lagging its target rate, the scheduler keeps track of a parameter G_i , called *normalized goodput gap*, which is defined as:

$$G_i(t) = \frac{g_i(t) - r_i t}{r_i t} \quad (2)$$

where $g_i(t)$ is the amount of data of Flow i transmitted successfully up to time t within its current backlog period. Note that $G_i(t)$ is normalized by the target goodput r_i and it is a fraction which represents how much a flow is leading and lagging compared to its target goodput. g_i , G_i , and t are reset to zero at the beginning of a backlog period for each flow.

The goodput threshold of Flow i , T_i , is set in such a way that $-1 \leq T_i \leq 0$. A flow with a bad link may still transmit a real packet if its normalized goodput gap falls below its goodput threshold or it cannot find any other flow with a good link. As discussed in Section IV-B, setting such a threshold guarantees a certain minimum transmission bandwidth for a flow. Whenever a flow substantially lags behind its target goodput, it would transmit real packet(s) even in the *bad* state. In this way, a flow with a link experiencing a long duration of the *bad* state would not be totally deprived of transmission. The smaller the goodput threshold, the smaller the guaranteed transmission bandwidth would be. Although this may lower overall bandwidth efficiency in terms of the total goodput, it is necessary for the scheduler to have the ability to strike a balance between bandwidth efficiency and bandwidth guarantees regardless of the states of any links.

Fig. 2 shows the complete scheduling operation for BGFS-EBA. The first three blocks are the operations performed in the first phase as described above. When a virtual flow is selected and its real flow is backlogged, a real packet of the flow will normally be scheduled for transmission so long as the link of the flow is in the *good* state. Otherwise, it will give up the current service opportunity to some other flow with a link in the *good* state. Nevertheless, there are some exceptional cases to be considered as follows.

When it is the turn for Flow i , which is currently backlogged, to transmit according to the reference system and its link is in the *good* state, it will give up its service opportunity to some other flow if all of the following conditions are satisfied:

- 1) Flow i is leading, i.e. $G_i > 0$.
- 2) $g_i + L \geq r_i \cdot (d_i - \tau_i)$, where d_i is the current deadline (after an update in the first phase) of a virtual packet for Flow i , and τ_i is the start time of the current backlog period for Flow i .
- 3) There exists at least one backlogged flow with a negative normalized goodput gap and the link of the flow is

the packet size, and the flow target rate. Thus, it does not need to timestamp any real packet nor access any control information of some other flows served in the system. This makes the computation much simpler and efficient. Besides, the scheduler may also need to verify whether any of the conditions are fulfilled for the exceptional cases described above. BGFS-EBA needs $O(N)$ counters to store its working variables. At each time a packet is scheduled, only a few sets of counters are needed to be updated and a small subset of conditions are generally required to be checked. Furthermore, each mobile host with an uplink flow is required to send a control message to the base station periodically indicating that the beginning or the end of a backlog period of each flow, independent of the number of packets sent. This leads to a very small volume of control message exchange among the mobile hosts and the base station. Thus, we consider that the computational, storage, and communication overheads are minimal.

B. Analytical Studies

1) *Minimum Bandwidth and Goodput Guarantees for Error-prone Flows:* Flow i is guaranteed a service rate r_i in the reference system. Whenever the goodput rate of the corresponding real flow falls below $r_i \cdot (T_i + 1)$, the real flow will send at a rate no less than r_i such that $r_i \geq r_i \cdot (T_i + 1)$, where the goodput threshold of Flow i , T_i , falls between -1 and 0. Flow i may send real packets at a rate lower than r_i only when its goodput is larger than $r_i \cdot (T_i + 1)$. In order to achieve such a goodput, at least the same amount of transmission bandwidth is allocated to the flow. Hence, for a continuously backlogged Flow i , its allocated transmission bandwidth is at least $r_i \cdot (T_i + 1)$ over a sufficiently long time. Consequently, if the error probability of a low-rate packet for Flow i , with a coding rate² of ψ , in the *bad* state is e_i , Flow i will be guaranteed an approximate long-term goodput of $r_i \psi \cdot (T_i + 1) \cdot (1 - e_i)$, even if the link of the flow is always in the *bad* state.

2) *Delay Bounds for Bucket-Constrained Error-Free Flows:*

Theorem 1: Suppose Flow i is served with an error-free link at a target rate r_i over a wireless communication channel having the maximum data rate R . Its traffic is constrained by a token bucket with parameters (σ_i, r_i) , where σ_i is the bucket depth and r_i is the average token rate, which is equal to the target rate for Flow i . The packet delay for Flow i , D_i , can be bounded as:

$$D_i \leq \frac{\sigma_i}{r_i} + \frac{2L}{r_i} + \frac{L}{R} \quad (3)$$

where $i = 1, 2, \dots, N$.

Proof: An error-free wireless channel can always achieve the maximum data rate R . It is proved [19] that, when the received service of Flow i , $i = 1, 2, \dots, N$, can be guaranteed in any backlog period as shown in (1), the packet delay can be bounded as:

$$D_i \leq \frac{\sigma_i}{r_i} + \theta_i \quad (4)$$

²The coding rate of a packet is defined as the ratio of the amount of user data to the total packet size.

where $i = 1, 2, \dots, N$.

In BGFS-EBA, it can easily be shown that Flow i is guaranteed to be served as in (1) with $\theta_i = \frac{L}{r_i}$ in the reference system. This corresponds to the dotted line with a slope of r_i in Fig. 1. If Flow i is an error-free flow, after the first packet of a backlog period is transmitted, the subsequent packets are guaranteed to be sent at a rate of r_i . Therefore, the service received by Flow i is also lower-bounded as in (1). We then only need to determine the latency θ_i for real Flow i . This latency is in fact its latency in the reference system plus the worst-case delay between when the first real packet of the flow in a backlogged period arrives and when its corresponding virtual flow is scheduled again in the reference system. Indeed, it is not difficult to show that such worst-case delay is $\frac{L}{R} + \frac{L}{r_i}$. Hence, the latency for real Flow i , θ_i , is $\frac{L}{R} + \frac{2L}{r_i}$. The captioned delay bound as shown in (3) is thus established. ■

C. Discussion on Generalization to Multirate Time-Varying Channel

Our proposed scheduling algorithm, BGFS-EBA, can be easily extended to handle traffic flows transmitted over a multirate wireless communication channel. An outline of the generalized algorithm is as follows. Suppose there is a set of C different data rates (in bytes per second) supported by the multirate channel, namely, R_1, R_2, \dots, R_C , where $R = R_1 > R_2 > \dots > R_C > 0$. Such data rates can be realized, for example, by sending a fixed-size packet with different amount of redundant data for forward error correction (FEC). The selection of a particular data rate for transmission depends on the channel state for the communication between a mobile host and the base station. The better the channel condition, the higher the data rate the wireless channel can support. A packet from Flow i can be sent on a wireless channel at State k , i.e. $S(i) = k$, when the packet is transmitted at rate R_k bytes per second. The generalized algorithm is similar to the algorithm discussed in Section IV-A with $C = 2$, except for the mechanisms to handle leading flows and to select a flow to receive excess or compensated bandwidth.

A leading backlogged flow, Flow i , can give up its service opportunity to some other backlogged flow, Flow j if the first two conditions as specified in Section IV-A are satisfied. The third condition is now replaced such that the normalized goodput gap of Flow j falls below its goodput threshold, or the normalized goodput gap of Flow j is negative and the channel condition for Flow j is no worse than that for Flow i , i.e. $S(i) \geq S(j)$.

To select a flow for receiving the excess or compensated bandwidth, the scheduler searches for a backlogged flow in the following ordered sets (in decreasing priorities). When there are more than one flow in the same set, a flow with the better channel condition will be selected for service. If there are two or more flows having the same best channel state, a flow with the smallest normalized goodput gap is chosen. Ties are then broken arbitrarily:

- 1) Any flow, Flow j , with $S(j) \leq c$ and a negative normalized goodput gap, where $c \leq C$ is a system parameter for the generalized algorithm.

TABLE I
SIMULATION RESULTS FOR SCENARIO I.

	Flow 1		Flow 2		Efficiency
	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)	
CIF-Q ($\alpha = 0$)	193.3	89.9	726.0	639.8	73.0%
CIF-Q ($\alpha = 0.9$)	115.5	45.7	799.1	702.5	74.8%
BGFS-EBA ($T_{\bullet} = -0.3$)	590.8	278.4	409.2	362.8	64.1%
BGFS-EBA ($T_{\bullet} = -0.5$)	526.4	250.0	473.6	408.0	65.8%

- 2) Any flow, Flow j , with $S(j) > c$ and its normalized goodput gap below its goodput threshold.
- 3) Any backlogged flow.

The generalized algorithm degenerates to the algorithm discussed in Section IV-A when $c = 1 < C = 2$.

V. PERFORMANCE EVALUATION

In this section, we present our simulation results, compare our proposed algorithm, known as bandwidth-guaranteed fair scheduling with effective excess bandwidth allocation (BGFS-EBA), with channel-condition independent packet fair queuing (CIF-Q) [17], and briefly discuss the issue of goodput threshold of our proposed algorithm.

A. Simulation Setup

Our simulations have been carried out in Network Simulator Version 2 (ns-2) [8]. We implemented both CIF-Q and BGFS-EBA, and conducted some simulations to compare the performance of these wireless packet scheduling algorithms. To simulate the fact that each link state corresponds to a range of bit error rate (BER) or signal-to-noise ratio (SNR) instead of a single point, we let all the packet error probabilities vary over a range. We denote by $u(a, b)$ an uniform distribution between a and b .

The parameters used in the simulations are shown as follows. The total available channel bandwidth is 1 Mbps. The duration of each test equals the transmission time of two million packets. All flows require in-sequence, reliable data delivery service. The packet size is 200 bytes. The error probabilities of a regular packet, which does not contain redundant data for forward error correction (FEC), are uniformly distributed as $u(0, 0.2)$ and $u(0.8, 1)$ in the *good* state and the *bad* state, respectively. In BGFS-EBA, when needed, a regular packet is split into $m = 2$ low-rate packets. Each low-rate packet contains 100 bytes of data drawn from a regular packet and an FEC code for error detection and correction. The error probabilities of a low-rate packet are uniformly distributed as $u(0, 10^{-3})$ and $u(0, 0.1)$ in the *good* state and the *bad* state, respectively. Suppose a flow is chosen for transmission in a certain time slot. When the link is estimated to be in the *good* state, a regular packet of the flow is sent. Otherwise, the link is estimated to be in the *bad* state and hence a low-rate packet of the flow is transmitted. The duration of each link state is exponentially distributed. The link state will be incorrectly estimated with probability 0.1. When the goodput thresholds for all flows in a test are set to the same value, it is denoted as T_{\bullet} .

TABLE II
FLOW PARAMETERS FOR SCENARIO II.

	Target Rate (Kbps)	Average Link State Duration (s)	
		<i>good</i>	<i>bad</i>
Flow 1	200	0.03	0.07
Flow 2	100	0.03	0.07
Flow 3	200	0.09	0.01
Flow 4	200	0.09	0.01
Flow 5	100	0.09	0.01

B. Scenario I: Minimum Transmission Bandwidth and Goodput Guarantees

We start with a very simple scenario to demonstrate the idea of guaranteeing the minimum transmission bandwidth and goodput for flows scheduled by our proposed algorithm, BGFS-EBA. In this example, there are only two flows in the network. Each flow has a target rate of 500 Kbps. The average duration of the *good* and *bad* state of the link experienced by Flow 1 are 0.01 s and 0.09 s, respectively. The average duration of the *good* and *bad* state of the link experienced by Flow 2 are 0.09 s and 0.01 s, respectively. Both Flows 1 and 2 are greedy traffic sources, so that they always have packets for transmission.

The simulation results are shown in Table I. The parameter α , as defined in CIF-Q, is the minimum fraction of service retained by a leading session. Efficiency is defined as the total goodput divided by the total available bandwidth (or 1 Mbps). Both flows are associated with the same goodput threshold $T_1 = T_2 = T_{\bullet}$ when BGFS-EBA is employed.

As exhibited from the performance results for CIF-Q, there is no link-independent transmission bandwidth guarantee for a flow with a very poor link, because a flow always gives up its bandwidth (or effort) to others when its link is in the *bad* state. Note that for CIF-Q, Flow 1 receives very little bandwidth and goodput. Even when α is 0 (where the lagging Flow 1 has the most transmission opportunities when scheduled by CIF-Q), its effort rate is still less than 40% of its target rate. However, when flows are scheduled by BGFS-EBA, each flow is guaranteed a certain amount of bandwidth regardless of its link quality, thereby achieving effort fairness. Moreover, Flow 1 is given more transmission effort to compensate for its inferior link quality to help it achieve outcome fairness. Indeed, we observe that Flow 1 receives far more bandwidth and goodput when using BGFS-EBA than CIF-Q. Moreover, Flow 1 receives more bandwidth when the goodput threshold T_{\bullet} is -0.3 than when T_{\bullet} is -0.5, as a larger T_{\bullet} results in greater guaranteed minimum transmission bandwidth and goodput.

We notice that the total effort rate when employing CIF-Q is smaller than the total available bandwidth. The reason is that the CIF-Q scheduler will stay idle if it cannot find any

TABLE III
SIMULATION RESULTS OF CIF-Q FOR SCENARIO II.

	$\alpha = 0$		$\alpha = 0.3$		$\alpha = 0.7$	
	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)
Flow 1	249.9	156.4	160.8	111.0	151.3	105.9
Flow 2	124.9	71.7	124.8	80.0	124.8	80.4
Flow 3	249.9	225.2	284.0	251.9	290.3	255.9
Flow 4	249.9	225.2	282.3	250.3	284.3	251.8
Flow 5	125.0	115.6	147.5	130.3	148.8	130.4
Efficiency	79.4%		82.3%		82.4%	

TABLE IV
SIMULATION RESULTS OF BGFS-EBA FOR SCENARIO II.

	$T_{\bullet} = -0.2$		$T_{\bullet} = -0.3$		$T_{\bullet} = -0.5$	
	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)
Flow 1	279.8	200.1	278.3	199.5	277.9	199.3
Flow 2	149.8	101.8	145.5	101.0	143.9	100.2
Flow 3	227.8	201.5	230.1	204.1	231.0	205.3
Flow 4	227.5	201.4	230.0	204.0	231.0	205.2
Flow 5	115.0	100.7	116.1	102.3	116.5	103.2
Efficiency	80.6%		81.1%		81.3%	

TABLE V
SIMULATION RESULTS OF CIF-Q FOR SCENARIO III.

	$\alpha = 0$		$\alpha = 0.3$		$\alpha = 0.7$	
	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)
Flow 1	283.9	177.4	173.6	128.2	164.2	118.4
Flow 2	142.4	83.1	140.3	84.2	134.1	85.7
Flow 3	285.0	265.3	341.0	304.7	347.9	309.0
Flow 4	285.5	266.1	341.6	305.3	350.1	310.8
Efficiency	79.2%		82.2%		82.4%	

TABLE VI
SIMULATION RESULTS OF BGFS-EBA FOR SCENARIO III.

	$T_{\bullet} = -0.2$		$T_{\bullet} = -0.3$		$T_{\bullet} = -0.5$	
	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)	Effort (Kbps)	Goodput (Kbps)
Flow 1	276.4	200.0	276.1	199.6	276.1	199.6
Flow 2	159.1	116.6	159.1	116.7	159.1	116.7
Flow 3	282.4	249.0	282.6	249.4	282.6	249.4
Flow 4	282.1	249.1	282.2	249.3	282.2	249.3
Efficiency	81.5%		81.5%		81.5%	

flow with a link having the *good* state. Hence, a small amount of bandwidth is wasted while the channel is idle. In addition, the channel can achieve a higher efficiency for BGFS-EBA than for CIF-Q, since this is the tradeoff between effort-fair and outcome-fair by providing the minimum bandwidth guarantees. In fact, this is an extreme case, where the link quality for Flow 1 is pretty bad and the target rate for Flow 1 is quite high. Generally speaking, the difference in efficiency between these two algorithms would not be so large and this can be acceptable.

C. Scenario II: Outcome Fairness

For Scenario II, we consider a more complex scenario. There are a total of five greedy flows sharing the same wireless channel. The target rates and the average link state durations of each flow are summarized in Table II.

The simulation results are displayed in Tables III and IV. When CIF-Q is used for packet scheduling, the average goodputs of the flows with better links (i.e. Flow 3, 4, and 5) are at least 10% higher than their target rates while Flows 1 and 2 obtain average goodputs far below from their target

rates. As discussed in Section V-B, this is due to the way the excess bandwidth is distributed and the discrimination of CIF-Q against flows with worse average link qualities. Hence, the flows with good links are over-provisioned in transmission effort. When BGFS-EBA is used instead, the goodput of nearly every flow reaches its target rate. This is because the scheduler utilizes the excess bandwidth to compensate flows with inferior links while trying to make sure that such compensation does not jeopardize the service target rates of the flows with better links. Since the goodput thresholds are not set very aggressively, efficiency can still be maintained at a high level. Outcome fairness is therefore achieved for BGFS-EBA.

D. Scenario III: Distribution of Excess Bandwidth

For Scenario III, we eliminate Flow 5 from Scenario II to see how the channel bandwidth will be distributed when there is more than enough bandwidth for flows to reach their target rates. The parameter setting is the same as in Scenario II, except for the one stated above. The simulation results are exhibited in Tables V and VI.

TABLE VII
FLOW PARAMETERS FOR SCENARIO IV.

	Target Rate (Kbps)	Traffic	Average Link State Duration (s)	
			<i>good</i>	<i>bad</i>
Flow 1	200	Exponential on/off	Error-free	
Flow 2	100	Poisson	Error-free	
Flow 3	200	Poisson	0.09	0.01
Flow 4	200	Greedy	0.09	0.01
Flow 5	200	Greedy	0.03	0.07
Flow 6	100	Greedy	0.09	0.01

TABLE VIII
PACKET DELAYS OF CIF-Q FOR SCENARIO IV.

	$\alpha = 0.1$		$\alpha = 0.3$	
	Maximum (s)	Average (s)	Maximum (s)	Average (s)
Flow 1	0.0095	0.0008	0.0096	0.00077
Flow 2	0.019	0.0024	0.0191	0.0026
Flow 3	0.276	0.057	0.277	0.0386

For CIF-Q, Flow 1 still misses its goodput target even when there is a sufficient amount of available bandwidth. At the same time, Flows 3 and 4 receive far more than their fair shares of bandwidth. For BGFS-EBA, after the scheduler manages to satisfy the goodput target of every flow, it does the favour for Flows 3 and 4 by allocating them the excess bandwidth. This matches the policy in BGFS-EBA that flows with positive lags can receive the excess bandwidth when there exists no lagging flow. Therefore, the target rate of every flow has been met, whereas the scheduler also tries to maintain high efficiency in bandwidth usage.

E. Scenario IV: Packet Delay

For this scenario, we demonstrate that the delay bounds for error-free flows can be guaranteed when BGFS-EBA is used. There are a total of six flows, where the channel is fully loaded as the sum of all the target rates is equal to the channel bandwidth.

Table VII shows the parameter setting of the traffic sources and the links for all flows. Flow 1 corresponds to an exponential on/off source, where the average “on” and “off” periods are 500 ms and 100 ms, respectively. When the source is “on”, it sends data at a rate of 400 Kbps. When the source is “off”, it sends no data. Flows 2 and 3 correspond to two Poisson sources, each of which has an average rate of 200 Kbps. The traffic bursts for Flows 1, 2, and 3 are token-bucket constrained in order to provide delay guarantees. The token bucket rate of each flow is the same as its target rate. The bucket depth is 2000 bytes, which is equal to the total size of ten data packets. For BGFS-EBA, if a regular packet is split into $m = 2$ low-rate packets, all its low-rate packets have to be received successfully or its transmission is considered as incomplete.

The performance of CIF-Q and BGFS-EBA in packet delay are shown in Tables VIII and IX. The analytical bounds are computed based on (3) derived in Section IV-B2. Based on the simulation results, we find that all packets belonging to error-free flows (i.e. Flows 1 and 2) are sent, through the BGFS-EBA scheduler, within the analytical bounds. Moreover, the maximum and average packet delays for error-free flows scheduled through either of these two algorithms are similar.

However, the maximum and average packet delays for the error-prone flow (i.e. Flow 3) are much smaller when BGFS-EBA is employed, since error-prone flows can send their packets more aggressively.

F. Discussion on Goodput Threshold

Although the goodput thresholds for all flows are the same when BGFS-EBA is employed in the simulation study, it is indeed not necessary for all these flows to have the same threshold value. In fact, to improve bandwidth efficiency, it would be advantageous to set a higher goodput threshold for a flow with a better link. Similarly, it is also beneficial to set a lower goodput threshold for a flow with a worse link. In general, the goodput thresholds for flows with poor links should not be set too aggressively.

For a high-priority flow, its goodput threshold can be set to zero or close to zero so that the flow will be unlikely to give up its service opportunity even when its link is in the *bad* state. For a flow without any bandwidth guarantee, it may be desirable to set the goodput threshold to be -1 so that the flow will always give up its service opportunity when its link is in the *bad* state.

Besides, the goodput threshold may also be changed dynamically according to the link quality and traffic load. In our future work, we shall study how to design a proper admission control scheme and set the goodput thresholds to maintain a certain efficiency target, and how to dynamically adjust these goodput thresholds.

VI. CONCLUSIONS

In this paper, we propose a low-complexity wireless packet scheduling algorithm called bandwidth-guaranteed fair scheduling with effective excess bandwidth allocation (BGFS-EBA). BGFS-EBA assigns each packet with a deadline and normally schedules a packet with the smallest deadline for transmission. To strike a balance between effort-fair and outcome-fair, BGFS-EBA permits a flow leading its target goodput to give up its current service opportunity to a lagging flow. Flows with better links, which are able to meet the target goodputs, are thus guaranteed to be served no more than their target rates if there exists a lagging flow. In the long run, flows with poor links, which are not able to meet their target rates, are guaranteed the minimum transmission bandwidths based on their preset goodput thresholds. Excess bandwidth is therefore allocated to compensate flows with bad links. We have developed a delay bound for error-free flows and transmission effort guarantees for error-prone flows.

TABLE IX
PACKET DELAYS OF BGFS-EBA FOR SCENARIO IV.

	$T_{\bullet} = -0.3$		$T_{\bullet} = -0.5$		Analytical Bound
	Maximum (s)	Average (s)	Maximum (s)	Average (s)	
Flow 1	0.0098	0.0009	0.0096	0.0007	0.0122
Flow 2	0.019	0.0023	0.018	0.0018	0.0242
Flow 3	0.104	0.014	0.11	0.015	N/A

Our studies, based on a two-state probabilistic link model, reveal that, when compared with other existing wireless traffic scheduling algorithms such as channel-condition independent packet fair queueing (CIF-Q) [17], BGFS-EBA is a simple algorithm that can effectively distribute the excess bandwidth, strike a balance between effort-fair and outcome-fair, and provide a delay bound for error-free flows and transmission effort guarantees for error-prone flows.

There are several possible extensions to our work, some of which are listed as follows:

- extend our proposed algorithm with a generalized channel model with a set of channel states, like the one described in [21], so as to better capture the channel adaptive behaviour with the state of a link;
- design a proper admission control scheme and combine it with the setting of goodput thresholds to maintain a certain efficiency target; and
- devise an adaptive algorithm to dynamically adjust the goodput thresholds.

ACKNOWLEDGEMENT

This research is supported in part by the Research Grants Council, Hong Kong Special Administrative Region, China (Project No. HKU 7148/06E). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which assisted us in improving the quality of this manuscript.

REFERENCES

- [1] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *Proc. IEEE INFOCOM 2005*, vol. 4, pp. 2415-2424, Miami, FL, USA, Mar. 2005.
- [2] M. Andrews and L. Zhang, "Scheduling over nonstationary wireless channels with finite rate sets," *IEEE/ACM Trans. Networking*, vol. 14, no. 5, pp. 1067-1077, Oct. 2006.
- [3] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs," *Wireless Networks*, vol. 3, no. 1, pp. 91-102, Mar. 1997.
- [4] Y. Cao and V. O. K. Li, "Scheduling algorithms in broad-band wireless networks," in *Proc. IEEE*, vol. 89, no. 1, pp. 76-87, Jan. 2001.
- [5] Y. Cao and V. O. K. Li, "Wireless packet scheduling for two-state link models," in *Proc. IEEE GLOBECOM 2002*, vol. 1, pp. 819-823, Taipei, Taiwan, China, Nov. 2002.
- [6] R. L. Cruz, "Quality of service guarantees in virtual circuit switched networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1048-1056, Aug. 1995.
- [7] D. A. Eckhardt and P. Steenkiste, "An Internet-style approach to wireless link errors," *Wireless Commun. and Mobile Computing*, vol. 2, no. 1, pp. 21-35, Feb. 2002.
- [8] K. Fall and K. Varadhan, "The ns manual (formerly ns notes and documentation)," The VINT Project, Apr. 2008.
- [9] A. Gyasi-Agyei and S.-L. Kim, "Cross-layer multiservice opportunistic scheduling for wireless networks," *IEEE Commun. Mag.*, vol. 44, no. 6, pp. 50-57, June 2006.
- [10] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE VTC 2000-Spring*, vol. 3, pp. 1854-1858, Tokyo, Japan, May 2000.
- [11] P. Lin, B. Bensaou, Q. L. Ding, and K. C. Chua, "CS-WFQ: a wireless fair scheduling algorithm for error-prone wireless channels," in *Proc. IEEE ICCCN 2000*, pp. 276-281, Las Vegas, NV, USA, Oct. 2000.
- [12] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1452-1463, Aug. 2006.
- [13] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 2053-2064, Oct. 2001.
- [14] Y. Liu, S. Gruhl, and E. W. Knightly, "WCFQ: an opportunistic wireless scheduler with statistical fairness bounds," *IEEE Trans. Wireless Transmissions*, vol. 2, no. 5, pp. 1017-1028, Sept. 2003.
- [15] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 473-489, Aug. 1999.
- [16] S. Lu, T. Nandagopal, and V. Bharghavan, "Design and analysis of an algorithm for fair service in error-prone wireless channels," *Wireless Networks*, vol. 6, no. 4, pp. 323-343, July 2000.
- [17] T. S. E. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," in *Proc. IEEE INFOCOM '98*, vol. 3, pp. 1103-1111, San Francisco, CA, USA, Mar. 1998.
- [18] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 344-357, June 1993.
- [19] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 611-624, Oct. 1998.
- [20] V. Tsibonis and L. Georgiadis, "Optimal downlink scheduling policies for slotted wireless time-varying channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1808-1817, July 2005.
- [21] L. Wang, Y.-K. Kwok, W.-C. Lau, and V. K. N. Lau, "Efficient packet scheduling using channel adaptive fair queueing in distributed mobile computing systems," *Mobile Networks and Applications*, vol. 9, no. 4, pp. 297-309, 2004.



Yaxin Cao received the B.Eng. degree from Tsinghua University in 1997 and the Ph.D. degree from University of Southern California, Los Angeles, California, USA, in 2002, both in Electrical Engineering. From 2000 to 2002, he was Visiting Research Associate in the Department of Electrical and Electronic Engineering at the University of Hong Kong, Hong Kong, where he conducted research on wireless networks and systems. He joined Broadcom Corporation in Matawan, New Jersey, USA, in 2003 and is currently System Design Manager working on 2G/3G wireless baseband chip design and development.



Ka-Cheong Leung (S'95–M'01) received the B.Eng. degree in Computer Science from the Hong Kong University of Science and Technology, Hong Kong, in 1994, the M.Sc. degree in Electrical Engineering (Computer Networks) and the Ph.D. degree in Computer Engineering from the University of Southern California, Los Angeles, California, USA, in 1997 and 2000, respectively. He worked as Senior Research Engineer at Nokia Research Center, Nokia Inc., Irving, Texas, USA from 2001 to 2002. He was Assistant Professor at the Department of Computer

Science at Texas Tech University, Lubbock, Texas, USA, between 2002 and 2005. Since June 2005 he has been with the University of Hong Kong, Hong Kong, where he is Visiting Assistant Professor at the Department of Electrical and Electronic Engineering. His research interests include wireless packet scheduling, routing, congestion control, and quality of service guarantees in high-speed communication networks, content distribution, high-performance computing, and parallel applications.



Victor O. K. Li (S'80–M'81–SM'86–F'92) received SB, SM, EE, and ScD degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, in 1977, 1979, 1980, and 1981, respectively. He joined the University of Southern California (USC), Los Angeles, California, USA in February 1981, and became Professor of Electrical Engineering and Director of the USC Communication Sciences Institute. Since September 1997 he has been with the University of Hong Kong, Hong Kong,

where he is Chair Professor of Information Engineering at the Department of Electrical and Electronic Engineering and Associate Dean of Engineering. He also served as Managing Director of Versitech Ltd., the technology transfer and commercial arm of the University, and on various corporate boards. His research is in information technology, including all-optical networks, wireless networks, and Internet technologies and applications. He is a Co-Director of the Area of Excellence in Information Technology funded by the Hong Kong Government. Sought by government, industry, and academic organizations, he has lectured and consulted extensively around the world. Prof. Li chaired the Computer Communications Technical Committee of the IEEE Communications Society 1987–1989, and the Los Angeles Chapter of the IEEE Information Theory Group 1983–1985. He co-founded the International Conference on Computer Communications and Networks (IC3N), and chaired its Steering Committee 1992–1997. He also chaired various international workshops and conferences, including, most recently, IEEE INFOCOM 2004 and IEEE HPSR 2005. Prof. Li has served as an editor of *IEEE Network*, *IEEE JSAC Wireless Communications Series*, and *Telecommunication Systems*. He also guest edited special issues of *IEEE JSAC*, *Computer Networks and ISDN Systems*, and *KICS/IEEE Journal of Communications and Networks*. He is now serving as an editor of *ACM/Springer Wireless Networks*. Prof. Li has been appointed to the Hong Kong Information Infrastructure Advisory Committee by the Chief Executive of the Hong Kong Special Administrative Region (HKSAR). He is a part-time member of the Central Policy Unit of the Hong Kong Government. He also serves on the Innovation and Technology Fund (Electronics) Vetting Committee, the Small Entrepreneur Research Assistance Programme Committee, the Engineering Panel of the Research Grants Council, and the Task Force for the Hong Kong Academic and Research Network (HARNET) Development Fund of the University Grants Committee. He was a Distinguished Lecturer at the University of California at San Diego, at the National Science Council of Taiwan, and at the California Polytechnic Institute. Prof. Li has also delivered keynote speeches at many international conferences. He has received numerous awards, including, most recently, the Changjiang Chair Professorship at Tsinghua University from the Ministry of Education, China, UK Royal Academy of Engineering Senior Visiting Fellowship in Communications, the Outstanding Researcher Award of the University of Hong Kong, the Croucher Foundation Senior Research Fellowship, and the Order of the Bronze Bauhinia Star, Government of HKSAR, China.