Scheduling Delay-Sensitive and Best-Effort Traffics in Wireless Networks

Yaxin Cao, Victor O. K. Li, and Zhigang Cao

Abstract—**In this paper we propose a novel wireless scheduling algorithm for delay-sensitive (DS) and best-effort (BE) traffics. Unlike the majority of the previous wireless scheduling, where the wireless links are modeled as having only two states, our algorithm is applicable to links with multiple states. For DS flows, the algorithm is capable of providing statistical *delay violation bounds*. Such bounds are derived, analytically, using the idea of the *statistical service envelope*. For BE flows, we propose a new notion of fairness, called long-term link-quality-weighted outcome-fair, which we believe is more suited to wireless networks than pure outcome-fair or effort-fair. The algorithm achieves a balance between bandwidth efficiency requirement and fairness requirement, and guarantees minimal goodput levels for BE flows.**

I. Introduction

Being one of the most important components of QoS provisioning and resource sharing, packet scheduling will play a key role in the future wireless data networks. Designing an efficient and effective packet scheduling algorithm is not a trivial task, because wireless communications pose special problems that do not exist in wired networks. The most difficult problems include limited bandwidth, high error rate, and transmission link variability. The quality of a wireless link is time-dependent as well as location-dependent.

A number of wireless packet scheduling algorithms have been reported in the recent years. [3] surveyed many representative algorithms and discussed their pros and cons. Most of the work models a wireless link with two states, in error or error-free. To reduce bandwidth wastage, the scheduler defers transmissions of flows whose links are in the error state and compensates those flows when their links recover. However, in reality the capacity of a wireless link does not jump between zero capacity and full capacity. To model the links more accurately, more states should be used. In particular, when adaptive modulation/coding is used, a flow clearly has different effective throughput levels for different modulation constellations or coding rates. Since bandwidth wastage due to losses, errors, and coding overheads is unavoidable in wireless links, the bandwidth allocated to a flow may not be equal to the actual goodput achieved. The above algorithms have not addressed this issue.

[1] and [2] point out the tradeoff between effort-fair and outcome-fair. Effort-fair means each flow should receive transmission bandwidth/effort in proportion to its assigned rate. Outcome-fair means the achieved goodput of each flow should be in proportion to its assigned rate. The algorithms in [1] and [2] try to achieve outcome-fair by increasing the fair queueing weights of the flows with larger error rates. To avoid overly reducing the overall bandwidth efficiency, an upper bound for the weight is set. However, the algorithms have some fundamental shortcomings. First, they do not exploit the benefit of improving bandwidth efficiency by swapping service opportunities among flows. Second, increasing the weights for flows with high error-rate links directly reduces the service effort other flows receive irrespective of their rate guarantees or link status. Similarly, [4] also associates the weight assignment of classical fair queueing algorithm to the link efficiency. Specifically, each flow's weight is determined by the link's efficiency raised by an exponent. By choosing different values of the exponent, the algorithm can adjust the resource allocation among flows taking into account the flows' efficiencies. However, when a link's status changes rapidly, the schemes adapting to link variations by adjusting weights are not effective.

Besides the aforementioned problems, none of the existing scheduling algorithms addresses the issue of providing delay guarantees for flows with error-prone links. Nevertheless, for delay-sensitive (DS) applications delay guarantees are essential. For best-effort (BE) applications, the major concerns of the scheduling algorithm should be bandwidth efficiency and fairness. Since links may be different not only in instantaneous qualities but also in average qualities, to achieve perfect fairness and to maximize bandwidth efficiency are conflicting objectives in most cases. When links are fast-changing and possess multiple states, maintaining short-term fairness becomes impractical and unnecessary. In addition, when links differ in average quality, how to guarantee bandwidth for flows with inferior link qualities and balance between efficiency and fairness objectives have not been well studied.

In this paper we study the above issues and propose a novel scheduling algorithm that applies to wireless links with multiple states. For DS flows, the algorithm provides statistical delay violation bounds even when the links are not perfect. For BE flows, we propose the notion of *long-term link-quality-weighted outcome-fairness (LT-LQW outcome-fairness)* taking into account the heterogeneous link qualities. In addition, we show that by using a simple scheduling scheme we can guarantee certain effort and goodput levels for each BE flow, improve bandwidth efficiency and maintain LT-LQW outcome-fair.

II. System Model

We consider a typical cell-structured wireless network, where centralized scheduling is performed. Transmission bandwidth is shared by the users/flows in a TDM manner, where time is slotted and only one flow is served in each time slot. Each wireless link between a user and the base station is time-varying with multiple states determined by the link's physical capacity and the particular error-correction code and modulation level used. The parameters used to characterize a state of a link include *effective output* and *state probability*. The *effective output* is the number of useful information bits successfully delivered in the particular state in one slot. We assume that this parameter has taken into account the coding overhead, the modulation level, and the corresponding error rate associated with the link state. The *state probability* is the probability of being in the state in each slot. We focus on fast-fading scenarios where the link states in two different slots can be considered independent of each other. Without loss of

This research is supported in part by the NSFC/RGC Joint Research Scheme under Grant No. N_HKU709/00.

Y. Cao and V. Li are with the Dept. of Electrical and Electronic Engineering, the University of Hong Kong, Pokfulam, Hong Kong ({yxcao, vli}@eee.hku.hk).

Z. Cao is with the Dept. of Electronic Engineering, Tsinghua University, Beijing, China (czg-dee@tsinghua.edu.cn).

0-7803-7802-4/03/$17.00 © 2003 IEEE

generality, we assume that the marginal distribution of the link states is stationary and ergodic. Assuming no interference and each user's location and moving speed to be independent of each other, all the links are independent of each other and may have different link statistics.

There are two types of data flows, DS flows and BE flows. Each DS flow has two target parameters, *delay bound* and *delay violation probability bound*. The objective of scheduling a DS flow is to guarantee that the probability of the flow's traffic delayed by more than the *delay bound* is no larger than the *delay violation probability bound*. To provide any delay guarantee, the incoming traffic must be constrained in burstiness and rate. We consider token-bucket constrained traffic for delay-sensitive flows. For BE flows, the scheduler tries to achieve high bandwidth efficiency while maintaining LT-LQW outcome-fair among the flows. Since the upper layer packets are usually large and need to be segmented at the wireless link layer, from the link layer's view point, the incoming traffic is considered as bit-streams. Each flow has its own queue with infinite buffer.

III. Delay Violation Bound for a Single DS Flow

First we discuss how to provide statistical delay violation bounds to a single flow. As time is slotted, we model the system as a discrete-time system with fluid-modeled traffic arriving and departing the queue only at the boundaries of time slots. The model is validated with reasonable requirements that the bits arriving in one slot can not be served in the same slot, and the bits served in a slot is said to complete the service only at the end of the slot. In this section all the variables that represent time are in the unit of a time slot. For a link I with L states, denote the *effective output* and *state probability* of state l by u_{Il} and p_{Il}, respectively. Denote the effective service received by the corresponding flow I during an interval $[\mu, \mu + t]$ by $W_I(t)$, and denote the total incoming traffic during the same interval by $A_I(t)$. If $A_I(t)$ and $W_I(t)$ satisfy

$$A_I(t) \leq_{st} B_I(t) \quad \text{and} \quad W_I(t) \geq_{st} S_I(t) \tag{1}$$

where $B_I(t)$ and $S_I(t)$, called *statistical traffic envelope* and *statistical service envelope*, respectively, are non-negative non-decreasing random processes, and $\leq_{st}$ represents stochastic inequality, then given a d_0, the probability of a bit being delayed for more than d_0, $P\{D_I > d_0\}$ is bounded as follows

$$P\{D_I > d_0\} \leq P\{\max_{t \geq 0}\{B_I(t) - S_I(t + d_0)\} > 0\} \tag{2}$$

where D_I is the delay experienced by a bit. We utilize the above theorem, which is proven in [7], to find a solution for bounding the delay violation probability.

In practice, statistical traffic envelopes are difficult to enforce. In this paper we consider deterministic traffic envelopes, in particular, token bucket constrained traffic. For a flow I regulated by a token bucket with parameter set (α, ρ), where α is the bucket depth and ρ is the token filling rate, $A_I(t) \leq \alpha + \rho t$.

Suppose we allocate one slot to flow I every M slots. Now the problem we need to solve is: Given (α, ρ), d_0 and delay violation probability bound P_v, what is the minimal bandwidth , i.e. the largest M, which needs to be allocated to the flow in order to guarantee that $P\{D_I > d_0\} \leq P_v$?

Without loss of generality, suppose at time 0, flow I's queue is empty. Then for any $t > 0$, $A_I(t) \leq \alpha + \rho t$, and $W_I(t) = \sum_{i=1}^{\lfloor \frac{t}{M} \rfloor} U_I$ where U_i is the information bits transmitted by the flow in the ith transmission/service, and $\lfloor \phi \rfloor$ is the largest integer that is no larger than ϕ. Note in our model, U_i, $i = 1, 2, ...$ are i.i.d. Let $B_I(t) = \alpha + \rho t$, and $S_I(t) = \sum_{i=1}^{\lfloor \frac{t}{M} \rfloor} U_I$, then

$$P\{B_I(t) - S_I(t + d_0) > 0\} = P\{\alpha + \rho t - \sum_{i=1}^{\lfloor \frac{t+d_0}{M} \rfloor} U_i > 0\} \tag{3}$$

Let $\lfloor \frac{t+d_0}{M} \rfloor = n$ and $\frac{t+d_0}{M} - \lfloor \frac{t+d_0}{M} \rfloor = \delta$. Then

$$t = Mn + M\delta - d_0 \tag{4}$$

Following (3) and (4),

$$P\{B_I(t) - S_I(t + d_0) > 0\} \tag{5}$$

$$\leq P\{\alpha + \rho M n + \rho M \delta - \rho d_0 - \sum_{i=1}^{n} U_i > 0\} \tag{6}$$

$$\leq P\{\sum_{i=1}^{n} (\rho M - U_i) > \rho d_0 - \rho M - \alpha\} \tag{7}$$

The last step above follows because $0 \leq \delta < 1$. Since the average arrival rate of the information data is ρ, and the average departure rate is $\frac{E\{U_i\}}{M}$, to have a stable queue we should have

$$\rho M < E\{U_i\} \tag{8}$$

Let $V_i = \rho M - U_i$, and $\rho d_0 - \rho M - \alpha = a$. Then (7) becomes

$$P\{B_I(t) - S_I(t + d_0) > 0\} \leq P\{\sum_{i=1}^{n} V_i > a\} \tag{9}$$

where $E\{V_i\} < 0$. Since the flow's service period is M slots, for the scheduler to guarantee the delay violation bound, d_0 should be no less than M. Therefore, for any $t > 0$, there exists an integer $n \geq 1$ according to mapping (4). Following (9),

$$P\{\max_{t \geq 0}\{B_I(t) - S_I(t + d_0)\} > 0\} \leq P\{\max_{n \geq 1}\{\sum_{i=1}^{n} V_i\} > a\}$$

Now we apply a corollary of Wald's identity [8]. The corollary is as follows. Let X_i, $i = 1, 2, ...$ be i.i.d, and $E\{X_i\} < 0$. Let $\theta(r) = ln(E\{exp(rX_i)\})$ be the semi-invariant moment generating function of each X_i. Let $S_n = X_1 + ... + X_n$. If there exists an $r^* > 0$ such that $\theta(r^*) = 0$, then given $b \geq 0$,

$$P\{\max_{n \geq 1}\{S_n\} \geq b\} \leq exp(-r^* b) \tag{10}$$

The proof is given in [8].

Since V_i, $i = 1, 2, ...$ are i.i.d. and $E\{V\} < 0$, and let

$$a = \rho d_0 - \rho M - \alpha \geq 0\ , \tag{11}$$

then following (2) and (10), we have

$$P\{D_I > d_0\} \leq P\{\max_{n \geq 1} \sum_{i=1}^{n} V_i > a\} \leq exp(-r^* a) \tag{12}$$

where r^* is the positive root of the equation $ln(E\{exp(rV)\}) = 0$. Therefore, given M and d_0, (12) is an upper bound (and a reasonable estimate) of the delay violation probability. In particular, for a link I with L states,

$$ln(E\{exp(rV)\}) = ln\{\sum_{l=1}^{L} p_{Il} \cdot exp[r(\rho M - u_{Il})]\} \tag{13}$$

where u_{Il} and p_{Il} are the *effective output* and *state probability* of state l, respectively.

From the above analyses we make the following conclusion. Given delay violation probability bound P_v, and all the other necessary parameters, if we find an $M \geq 1$ such that (8), (11) and the following conditions are all satisfied,

$$exp\{-r^*(\rho d_0 - \rho M - \alpha)\} \leq P_v \tag{14}$$

the delay violation probability bound can be guaranteed. Note r^* is the positive root of the semi-invariant moment generating function of each $V_i = \rho M - U_i$.

Note $r^* > 0$ and $P_v \leq 1$; therefore, condition (14) implicitly requires that $\rho d_0 - \rho M - \alpha \geq 0$. That is, condition (11) is incorporated in (14) already.

Theorem 1: Let a flow's traffic be token-bucket constrained with bucket parameters (α, ρ), and the number of information bits it can send in each service slot i be U_i, $i = 1, 2...$, which are i.i.d. Given a delay bound d_0 and a delay violation probability bound P_v, we can guarantee that $P\{D_I > d_0\} \leq P_v$ by guaranteeing that the flow receives at least one service slot in every M time slots, if both of the following conditions are satisfied.

$$\begin{cases} M < \frac{E\{U_i\}}{\rho} \\ exp\{-r^*(\rho d_0 - \rho M - \alpha)\} \leq P_v \end{cases} \tag{15}$$

where r^* is the positive root of $ln(E\{exp[r(\rho M - U_i)]\})$.

Following the above theorem, to guarantee a delay violation probability bound with the least required bandwidth, we just need to pick the largest M that satisfies the conditions in (15).

The next question is: when there are N_d flows, how does one guarantee a service effort period M_I for each flow I, $I = 1, 2, ..., N_d$? We will elaborate more on this when discussing how to schedule multiple DS and BE flows simultaneously.

IV. Issues of Scheduling BE Flows

For BE traffic, no strict guarantees are provided. Nevertheless, the network still needs to facilitate fair sharing of the resource and avoid large differences in different flows' goodputs.

In most of the existing wireless scheduling, since the links are modeled as either error-free or 100% in error, only transmission in error-free state is possible. Therefore, improving efficiency is not a concern in such a scenario. Maintaining both short-term and long-term fairness has been the main focus of the previous research. However, things are more complicated when links have multiple effective throughput levels. Furthermore, links may be different not only in instantaneous quality but also average quality. More service opportunities need to be assigned to flows with inferior links to ensure that they have similar goodput levels as the others. However, this will decrease bandwidth efficiency. On the other hand, if the scheduler always chooses to serve the flow with the best instantaneous quality, some flows may be starved.

In wireless networks, the same amount of transmission effort may result in different outcome, due to users' different link qualities. Consequently, as we have discussed, there are two types of fairness notions, effort-fair and outcome-fair. As for end users, the effective service received is directly related to the useful data (outcome) sent/received, not the bandwidth (effort). Therefore, guaranteeing only effort-fair without considering the actual outcome is not meaningful. On the other hand, guaranteeing only outcome-fair irrespective of the link quality differences may result in very low bandwidth efficiency. We believe that the fairness notion should be based on outcome and also related to link qualities. Since the exact amount of any short-term outcome depends on the fast-changing instantaneous link status, which is not within the control of the scheduler, it is very difficult, if not impossible, to maintain short-term fairness based on outcome. Guaranteeing short-term effort-fairness is achievable by scheduling flows just according to classical wireline scheduling policies that provide isolation mechanisms among flows. However, such isolation prevents flows from swapping transmission opportunities, which may cause low bandwidth efficiency. In all, maintaining short-term fairness for BE flows in wireless networks is impractical and unnecessary. Following the above reasoning, we propose a new fairness notion, called *long-term link-quality-weighted outcome-fairness*, which is defined as follows. Considering two continuously backlogged flows over a sufficiently long time interval T, if the average effective output levels of the two flows' links are $\bar{O}_1$ and $\bar{O}_2$, respectively; and the total effective output achieved, $W_1(T)$ and $W_2(T)$ satisfy

$$\left| \frac{W_1(T)}{w_1 \bar{O}_1} - \frac{W_2(T)}{w_2 \bar{O}_2} \right| < \epsilon \tag{16}$$

where ϵ is a small constant, and w_1 and w_2 are generic bandwidth weights, we say LT-LQW outcome-fair is achieved.

Denote the total available bandwidth by B_w, of which B_d is allocated to DS flows. The bandwidth available to BE flows is thus $B_b = B_w - B_d$. Each BE flow J is assigned a generic bandwidth weight w_J. We assume that the weights are determined by some call-level bandwidth allocation module that may have already taken into consideration each flow's traffic demand and importance. Denote by $\bar{O}_J$ the average effective output that can be achieved on a link J in one time slot, and s the slot size in bits. According to the call-level allocation with perfect isolation among flows, each flow J should receive $B_J = B_b \cdot w_J / (\sum_{I=1}^{N_b} w_I)$ of bandwidth, and, consequently, achieve effective goodput at

$$r_J = B_J \frac{\bar{O}_J}{s} \tag{17}$$

where N_b is the total number of BE flows. We call r_J, which represents estimated goodput level the network operator would like flow J to achieve without considering packet-level adaptation, the *goodput target* of flow J.

It is easy to see that (16) is equivalent to the following.

$$\left| \frac{W_1(T)}{r_1} - \frac{W_2(T)}{r_2} \right| < \epsilon \tag{18}$$

As flows with worse links have smaller goodput targets than those with the same bandwidth weights but better links, LT-LQW outcome-fair facilitates more efficient use of bandwidth than pure outcome-fair. Furthermore, this fairness notion focuses only on long-term performance. As no strict short-term fairness is required, it gives the packet scheduler more freedom in improving bandwidth efficiency by selectively scheduling transmissions on links with better instantaneous qualities.

If maximizing the bandwidth efficiency is the sole objective of packet scheduling, the scheduler just needs to always schedule the flow with the best instantaneous link quality in each time slot. However, as fairness must also be considered, the objectives of scheduling for BE flows should be multi-dimensional. First, the scheduling policy should provide certain minimal goodput level (its target goodput) for each flow. Second, the policy should try to maintain LT-LQW outcome-fair. Third, the policy should try to achieve high bandwidth efficiency while maintaining fairness.

```
Scheduling Procedure: (when $\mathcal{C} \neq \emptyset$)
1.  $id = \{i \mid \min \Delta_i : i \in \mathcal{A}\}$;
2.  $\Delta_{id} = \Delta_{id} + \frac{s}{B_{id}}$;
3.  if $(id \in \mathcal{D})$ then
4.    $sid = id$;
5.  else if $(id \in \mathcal{B})$ then
6.    if $(g_{id} < t)$ then
7.      $sid = id$;
8.    else
9.      $fid = \{i \mid \max O_i : g_i \leq g_{id}, O_i \geq O_{id}, i \in \mathcal{B}\}$;
10.     if ($fid$ exists) then
11.       $sid = fid$;
12.     else
13.       $sid = id$;
14.     end if
15.   end if
16. else    (// $id$ is not backlogged)
17.   $fid = \{i \mid \min g_i : i \in \mathcal{B}\}$;
18.   if ($fid$ exists) then
19.     $sid = fid$;
20.   else
21.     $sid = \{i \mid \max O_i : i \in \mathcal{D}\}$;
22.   end if
23. end if
24. $sid$ transmits;
25. if ($sid$ is a best-effort flow) then
26.   $g_{sid} = g_{sid} + \frac{data\ sent}{r_{sid}}$;
27. end if
28. $t = t + \frac{s}{B_w}$;
end.
```

Fig. 1. Pseudo code of the scheduling procedure

V. Scheduling DS and BE Flows Together

The complete algorithm for scheduling both DS and BE flows is presented in Fig. 1, where the flow sets $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ are defined as follows: $\mathcal{A} = \{\text{all flows}\}$; $\mathcal{B} = \{\text{backlogged BE flows}\}$; $\mathcal{D} = \{\text{backlogged DS flows}\}$; $\mathcal{C} = \mathcal{B} \cup \mathcal{D}$.

The scheduling process is performed in two phases. In the first phase, the scheduling decision is made on an idealistic full-load error-free system. Each real flow, BE or DS, has a corresponding *virtual flow* in the error-free system. The virtual flows are assumed to be always backlogged. Each virtual flow J is assigned bandwidth B_J. We have $\sum_{J=1}^{N} B_J \leq B_w$, where B_w is the total available bandwidth, and N is the total number of users. Each virtual flow J maintains a deadline Δ_J. When the scheduler starts to work, each flow's deadline is initialized as $\Delta_J = s/B_J$, where s is the total number of bits that can be transmitted in one time slot when it is error-free. In the first phase, the scheduler always serves the virtual flow with the smallest deadline. To avoid unwanted synchronization, ties of deadlines are broken randomly. Regardless of which real flow is served in the second-phase scheduling decision, after a virtual flow is picked in the first phase, its deadline Δ_J is updated as $\Delta_J = \Delta_J + \theta_J$, where $\theta_J = s/B_J$. Without loss of generality, suppose the virtual flows become backlogged at time $t = 0$. Since $\sum_{J=1}^{N} B_J \leq B_w$ and the service unit is fixed, it is proven [6] that in the above system each flow J will be served at least once for every θ_J, and the total amount of service received by the flow is at least $m\theta_J B_J$ at $t = m\theta_J$, where $m = 1, 2, \ldots$. Therefore, each virtual flow is guaranteed a service rate B_J.

For a DS flow I, the service requirement is mapped to M_I. In the above idealistic system, if we set $\theta_I = M_I\tau$, where τ is the duration of a slot, virtual flow I will be scheduled at least once for very M_I slots, which is exactly what is required by a DS flow. Accordingly, the bandwidth B_I assigned to the virtual flow corresponding to the DS flow I is $B_I = s/(\tau M_I)$.

For each virtual flow corresponding to a BE flow K, the bandwidth received is $B_K = B_b \cdot w_K / (\sum_{I=1}^{N_b} w_I)$, where w_K, N_b and B_b are flow K's bandwidth weight, the total number of BE flows, and the bandwidth available to BE flows, respectively.

Fig. 1 describes the scheduling procedure performed for each time slot when there is at least one backlogged flow. Lines 1 and 2 are the first-phase operations, where the virtual flow (id) with the smallest deadline is picked.

In the second phase, if id is a backlogged DS flow, its corresponding real flow is always scheduled to transmit (lines 3-4). Thus, each DS flow is guaranteed to have the same service effort as its corresponding virtual flow. Therefore, following the above bandwidth assignments and the scheduling policy, the delay violation probability bound of each DS flow can be guaranteed, and such a guarantee is independent of other flows' behavior or link status.

Each BE flow K has a *goodput target* r_K, which is defined in (17). The scheduler keeps track of the service progress of each BE flow by using a parameter g_K called *normalized service time*. g_K is defined as

$$g_K = g'_K + \frac{W_K}{r_K} \tag{19}$$

where W_K is the total effective output the flow has achieved in the current backlogged period, and g'_K is the *normalized service time* at the beginning of the flow's current backlogged period. When a flow is continuously backlogged and $g'_K = 0$, g_K represents the equivalent service time the flow would have received if its goodput rate were maintained at r_K. To maintain LT-LQW outcome-fair is to ensure that g_K's do not differ greatly.

Lines 5-15 describe the second-phase operations when the flow (id) picked in the first-phase is a backlogged BE flow. First, g_{id} is compared with t, which records the real time passed after the scheduler starts to work. When $g_{id} < t$, the average goodput the flow has achieved is smaller than r_{id}, its goodput target. In this case, flow id will be scheduled to transmit regardless of the link status (lines 6-7), because we would like to guarantee a minimal goodput level (goodput target) for each BE flow. Otherwise, if flow id's average goodput is already above its target, the scheduler will try to find a backlogged flow with the best instantaneous quality among all the BE flows that has a smaller *normalized service time*, and an instantaneous link quality no worse than flow id's instantaneous link quality (line 9). Note that O_i is the effective output that can be achieved if flow i is scheduled to transmit in the current slot. If such a flow exists, flow id will give up the current slot to this flow. Otherwise, flow id will transmit in the current slot. Such operations incorporate considerations for both fairness and bandwidth efficiency. Flows with smaller *normalized service times* are flows receiving less service than what LT-LQW outcome-fairness requires; therefore, having flow id give up its service opportunity to such flows helps maintain fairness. On the other hand, to improve bandwidth efficiency, flow id only gives up its service to the lagging flow with the best instantaneous link quality.

As each virtual flow K is guaranteed to be served at rate B_K, and its corresponding BE flow does not give up its service whenever its average goodput is below its target, a backlogged flow can be guaranteed a minimal long-term goodput as in (17).

If the virtual flow (id) picked in the first-phase corresponds to an unbacklogged flow, the flow does not have enough traffic to make full use of its bandwidth share. Such excess bandwidth

TABLE I

STATE PROBABILITIES OF THE THREE TYPES OF LINKS

	state 1	state 2	state 3	state 4
type I	0.05	0.55	0.35	0.05
type II	0	0	0.55	0.45
type III	0.3	0.6	0.05	0.05

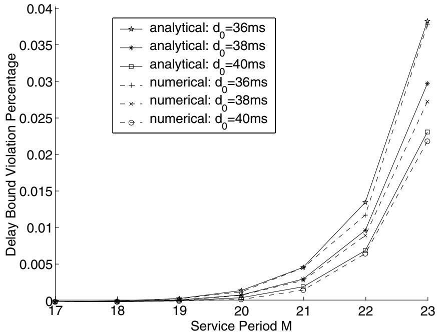

Fig. 2. Delay violation bounds for a single flow

is used to first compensate the BE flow with the least *normalized service time* (lines 17-19). If no BE flow is backlogged, the DS flow with the best instantaneous link quality gets the service (lines 20-22).

At the beginning of each backlogged period of a BE flow K, its g'_K is set as $g'_K = \max(g_K, \bar{g}_K)$, where $\bar{g_K}$ is the average value of the currently backlogged BE flows' g_K's. The reason for such an operation is to let an idle flow forgo its "false goodput credit" accumulated while it is idle.

VI. NUMERICAL EXAMPLES

Simulations are conducted to study the performance of our proposed algorithm. In the simulations, the system bandwidth is $1Mbps$. Each time slot lasts $1ms$. Hence, 250 bytes can be sent in one slot. Of the 250 bytes, at least 50 bytes are used for control, header, and minimum error-correction coding. Each link has four possible states with effective outputs of 200 bytes, 150 bytes, 100 bytes, 50 bytes, respectively. The state probabilities of the three types of links used in the simulations are listed in Table I.

A. Delay Violation Bound for a Single DS Flow

First we test the analytical results in section III. In this test, there is only one DS flow with type I link in the system. The token bucket parameters are $\rho = 40Kbps$, and $\alpha = 50bytes$. The flow is continuously backlogged with fluid traffic to maximize the traffic load. However, the flow does not use the whole available bandwidth. The flow is provided service of one slot every M slots. For *delay bounds* of $36ms$, $38ms$ and $40ms$, the analytical results are compared with simulation results in Fig. 2. The figure verifies that the delay violation percentages are smaller than the analytical bounds.

B. Scheduling Both Types of Flows Together

In this example, there are eight flows in the network. Flows 1 and 2 are BE flows with type II links. Flows 3, 4, 5 and 6 are BE flows with type III links. Flows 7 and 8 are DS flows with type I links. Flows 7 and 8 have the same token bucket parameters, which are $\rho = 40Kbps$ and $\alpha = 50bytes$. The required *delay bounds* for flows 7 and 8 are both $40ms$. All the BE flows have the same bandwidth weight.

Fig. 3 and Table II show the results when all flows have greedy traffic, which is the worst case for DS flows. From Table II we see that when multiple flows with both types exist, the analytical delay violation bounds can still be guaranteed.

TABLE II

RESULTS OF DELAY-SENSITIVE FLOWS

	service period M	analytical delay violation bound	simulation delay violation percentage
flow 7	21	2.06×10^{-3}	1.86×10^{-3}
flow 8	20	5.67×10^{-4}	5.03×10^{-4}

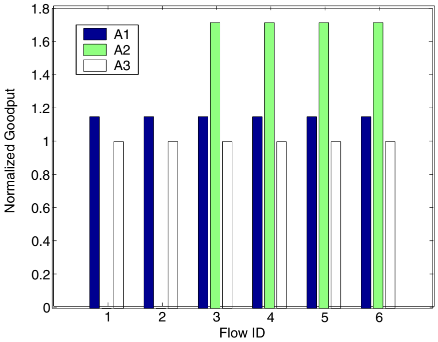

Fig. 3. Comparison of normalized goodput under different schemes

Fig. 3 compares the *target normalized goodputs* of three scheduling schemes. The *target normalized goodput* is defined as a flow's goodput divided by its goodput target. Achieving LT-LQW outcome-fair means that each flow has the same *target normalized goodput*. The three schemes being compared are the algorithm we propose (A1), classical fair-queueing with no service swapping (A3), and one which always schedules the flow with the best instantaneous link qualityn (A3). We note that both our algorithm and classical fair-queueing achieve LT-LQW. In fact, classical fair-queueing also achieves effort-fair. However, as it does not exploit the benefit of service swapping, bandwidth efficiency is lower. Bandwidth efficiency is calculated as the total goodput divided by the total goodput achievable if all the links stay in the best link state. The bandwidth efficiencies of the three schemes are 75.21%, 65.4%, and 89.9%, respectively. On the other hand, the scheme that maximizes the bandwidth efficiency result in extremely unfair share of bandwidth. We also simulated the scenarios when the BE flows are not greedy (Poisson and on-off traffics). After discounting the traffic idle periods when calculating the average goodputs, we find similar goodput distribution as that in Fig. 3. Therefore, the results show that the scheme we propose balances the requirements of bandwidth efficiency and fairness.

REFERENCES

[1] D. A. Eckhardt and P. Steenkiste, "Effort-limited fair (ELF) scheduling for wireless networks," Proc. of INFOCOM 2000, pp. 1097-1106, 2000.

[2] P. Lin, B. Benssou, Q. L. Ding, and K.C. Chua, "CS-WFQ: a wireless fair scheduling algorithm for error-prone wireless channels," Proc. Computer Communications and Networks 2000, pp. 276 -281, 2000.

[3] Y. Cao and V. O. K. Li, "Scheduling algorithms for broadband wireless networks," Proc. of the IEEE, Vol. 89, No. 1, pp. 76 - 87, Jan. 2001.

[4] Z. Jiang and N. K. Shankaranarayana, "Channel Quality Dependent scheduling for flexible wireless resource management," Proc. of GLOBECOM'01, pp. 3633-3638, 2001.

[5] D. Stiliadis and A. Varma, "Latency-rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms," IEEE Trans. on Networking, vol. 6, no. 5, pp. 611-624, Oct. 1998.

[6] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," IEEE JSAC, vol. 13, no. 6, pp. 1048-1056, Aug. 1995.

[7] J. Qiu and E. W. Knightly, "Inter-class resource sharing using statistical service envelopes," Proc. INFOCOM'99, pp. 1404-1411, 1999.

[8] R. G. Gallager, Discrete Stochastic Processes, Kluwer Academic Publisher, 1996.