Upper-level scheduling supporting multimedia traffic in cellular data networks

Young-June Choi a,*, Jin-Ghoo Choi b, Saewoong Bahk a

a School of EE & INMC, Seoul National University, Sillim-dong, Gwanak-gu, Seoul 151-742, Republic of Korea
b Samsung Electronics, Suwon 443-370, Republic of Korea

Received 27 June 2005; received in revised form 22 February 2006; accepted 22 May 2006
Available online 22 June 2006

Responsible Editor: X.S. Shen

Abstract

Wireless data networks such as cdma2000 1x EV-DO and UMTS HSDPA use downlink scheduling that exploits channel fading to increase the system throughput. As future wireless networks will eventually support multimedia and data traffic together, we need a proper criterion for scheduling that can count various service requirements such as delay and packet loss. Although some previous approaches proposed opportunistic schedulers at the lower layer, it has not been investigated well whether they are able to meet explicit QoS defined at the upper layer. Hence, in this paper, we develop a hierarchical scheduling model that considers QoS provisioning and the time-varying channel feature separately. We focus on the upper-level QoS scheduling that supports various traffic classes in a unified manner. Supposing that a user gets some satisfaction or utility when served, we introduce a novel concept of opportunity cost, which is defined as the maximum utility loss among users incurred by serving a particular user at the current turn. We obtain each user’s net profit by subtracting the opportunity cost from its expected utility, and then select a user with the maximum profit for service. Simulation results reveal that our scheme supports various QoS classes well that are represented by delay and packet loss under various traffic loadings.

Keywords: Scheduling; Quality-of-service; Utility; Opportunity cost

1. Introduction

Recently high data rate systems have been deployed for cellular networks of which examples are cdma2000 1x EV-DO (or HDR) in 3GPP2 [1] and High Speed Downlink Packet Access (HSDPA) in 3GPP [2]. They have some features in common with the next generation cellular system and can...
be good candidates for it. To support a variety of services, it is important to meet each user’s desired QoS in these networks. Especially a cellular data network needs to manage network resources deliberately because it has some different characteristics from a voice network in many aspects. First, in the data network, the traffic volume for downlink is much higher than that for uplink. Second, there are many kinds of services such as HTTP, WAP, VoIP, real time video traffic, and so on, which have their own requirements of delay and loss rate. Last, data traffic is bursty on the whole.

In this paper, we consider a cellular data system that uses transmission rate control instead of power control in the downlink. The base station (BS) transmits at full power [6] and uses a time division multiplexing to maximize the cell throughput like in the EV-DO standard [1]. A lot of works have dealt with downlink scheduling, where all users share a single channel [8–13]. They exploit channel fading instead of overcoming it. By allocating a time slot to a user with the maximum signal-to-interference-noise-ratio (SINR), the system achieves the maximum throughput. It is a simple concept of opportunistic scheduling that has lots of variants [8–13]. The opportunistic scheduling is beneficial for best-effort traffic, but it needs further consideration to be applied for multimedia applications because its effect on packet delay has not been investigated enough.

Some previous works have dealt with the problem of scheduling delay-sensitive traffic [14–17]. They apply opportunistic scheduling for real-time traffic, but cannot guarantee network-level QoS explicitly. Therefore we construct a two-level scheduling framework which comprises of the QoS-aware scheduler at the upper layer and the opportunistic scheduler at the lower layer. This architecture aims at maximizing the system efficiency, while satisfying each user’s QoS requirements. Since there have been a lot of works for lower-level scheduling, we work on the upper-level scheduling that supports diverse classes of traffic in an integrated fashion.

We use the concept of utility as the scheduling metric, which was first introduced in the area of networks by [19]. The utility represents the level of user’s satisfaction for the given service, and many opportunistic schedulers have already taken it as the objective function, i.e., maximizing the utility depending on time-varying SINR [9,10] or average transmission rate [20]. The QoS of multimedia traffic is usually characterized by delay bound and loss rate, so we reflect them in designing the utility function. Our scheduler chooses a user who can achieve the maximum profit from using the opportunity cost that is determined by considering other users’ utilities.

We organize the remainder of this paper as follows. Section 2 shows a hierarchical scheduling model and Section 3 describes the design principles of a utility function that concerns the QoS of each traffic class. Section 4 proposes our scheduling scheme that considers the utility and the opportunity cost together. Then we investigate the performance of our scheme through simulation experiments in Section 5. Finally, Section 6 concludes this paper.

2. Hierarchical scheduler model

The transmission rate control enables a scheduler to use opportunistic scheduling that exploits channel fluctuation to achieve high throughput rather than overcomes it. If the scheduler simply chooses a user with the maximum SINR, it unfairly prefers some users with good channel. Thus, various algorithms have been developed in [8–13]. In [8], the current data rate is normalized by the average data rate and the concept of proportional fairness is applied for scheduling. The fairness is generally expressed as a form of utility function in [9,10], and combined with opportunistic scheduling in [11]. Dynamic service demand is considered in [12], and a queue length based opportunistic scheduler is proposed in [13].

While these algorithms have focused on scheduling best-effort traffic, some works have dealt with scheduling real-time traffic [14–17]. In [14], opportunistic scheduling is exploited for serving real-time traffic. In [15], the objective is to reduce the delay for CDMA downlink. In [17], task size is considered for opportunistic scheduling in reducing the delay. In [16], the possibility of supporting VoIP service by opportunistic scheduling is considered.

The existing schemes opportunistically support the real-time service by focusing on physical and/or MAC layer. While the opportunistic scheduling determines the transmission order in a microscopic sense, the upper layers need QoS control in a macroscopic sense. This motivates us to construct a framework of hierarchical scheduling that consists of two steps: upper-level scheduling and lower-level scheduling. The upper-level scheduling is for network layer, so it goes with IP-QoS architecture such as DiffServ and IntServ. Upon arrival of an IP packet, the traffic classifier checks its service level
and puts it into the appropriate queue. The upper-level scheduler selects an IP packet to be served next considering its QoS requirement and passes it down to MAC/PHY layer.

The actual transmission depends on the adopted technologies such as channel coding and modulation scheme as well as the lower-level scheduler. Although the lower-level scheduler behaves stochastically, it does not affect overall QoS performance as much as the upper-level scheduler because of its microscopic scale, which will be confirmed through the simulations.\footnote{The upper-level scheduler needs to provide QoS with some margin to give room for lower layer scheduling, thereby meeting QoS requirements with much higher probability. As this can be another issue, we do not investigate this further.}

Fig. 1 shows a hierarchical scheduler model. When a packet arrives at the BS, it is classified according to its DS field (i.e., TOS field in IPv4) in the IP header or some other rules. The upper-level scheduling deals with the queues according to the predefined QoS attributes that are represented by utility mapping in our case. Then the packet undergoes MAC and PHY processing and enqueues for transmission. The second queue is handled by lower-level scheduling such as opportunistic scheduling. Each scheduler makes use of the channel feedback information and obtains the average data rate for each user by low pass filtering.

This model has an advantage of reusing various existing schemes, which makes the implementation easy. Some existing opportunistic schedulers can work for lower-level scheduling to increase the system efficiency. As the upper-level scheduler, we can exploit Earliest Deadline First (EDF), Shortest Remaining Processing Time first (SRPT), or other fair scheduling algorithms for QoS provisioning. They are not efficient in supporting various traffic classes, however, because they cannot integrate various QoS levels. For example, EDF decides the service order by the earliest deadline from several deadlines, but it does not consider packet loss or other QoS attributes. In conventional IP networks, each queue at a router focuses on achieving fairness by the Weighted Fair Queueing (WFQ) scheduler\cite{22} or its variants. To apply those schedulers to a wireless network, some mechanisms that compensate for user’s location-dependency in bandwidth and burst channel error over the wireless link are
additionally needed [23]. As a way of compensation, they simply classify the wireless channel into good or bad state, and recently have been replaced by opportunistic scheduling that exploits the characteristics of channel fading. In this paper, we propose and discuss a utility-based upper-level scheduler, which manages multiple QoS levels effectively.

3. Utility design

3.1. QoS and utility functions

3GPP and 3GPP2 have classified traffic into conversational, streaming, interactive and background classes [3,4]. Their characteristics are shown in Table 1. We represent QoS objective by a utility function that quantifies the satisfaction level of each user, depending on throughput, delay, and loss rate [19]. As shown in Table 1, multimedia traffic is generally sensitive to delay and loss. Hence, we incorporate both delay and data loss into the utility function. For example, class 2 traffic in Fig. 2 has the delay bound $D$ and packets out of this bound will be dropped. There exists a relationship between delay bound and loss rate. Hence, we introduce the offset value $g_{\text{offset}}$ as shown for class 2 to incorporate the delay bound into the utility function.

The utility function can reflect the service priority flexibly. For instance, streaming traffic has priority over interactive or background traffic generally. An interactive packet that approaches its delay bound, however, needs to have priority over other streaming packets that have some remaining time. Considering the delay bound as well as the service priority among the traffic classes, we can design various utility functions as shown in Fig. 2. In this paper, we find a proper $g_{\text{offset}}$ that reflects QoS requirements, and propose a scheduling algorithm that handles total utility functions in an integrative manner.

3.2. Delay

The user delay consists of access and propagation delay as well as queuing delay at the scheduler. As the propagation delay is fixed and the access delay is irrelevant to the downlink scheduling scheme, we leave both delays out of our concern. Other process-

Table 1
Traffic classes [3]

<table>
<thead>
<tr>
<th>Class</th>
<th>Attributes of traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conversational</td>
<td>Low delay/loss rates</td>
</tr>
<tr>
<td>Streaming</td>
<td>Less sensitive to delay, high bandwidth</td>
</tr>
<tr>
<td>Interactive</td>
<td>Bursty, moderate delay/loss rate</td>
</tr>
<tr>
<td>Background</td>
<td>Highly tolerant to delay/loss rate</td>
</tr>
</tbody>
</table>

![Fig. 2. Examples of class k’s utility function $y_k$: $y_1 = 1.05$ ($x < 30$), $y_2 = \exp(-x/20)$ ($x < 100$), $y_3 = \exp(-x/60)$ ($x < 300$), and $y_4 = 0.2$.](image)

For a more comprehensive approach, some other factors such as throughput and fairness can be considered together. Due to the complexity in handling such a more general form of utility function, we leave this issue for future work.

3 A similar assumption is found in [18], and it is acceptable in static channel environments for nomadic users [7].

4 ARQ mechanisms have two types: MAC and PHY ARQs. MAC ARQ is responsive to the upper-level scheduler while PHY ARQ like hybrid ARQ is to the lower-level scheduler.

If the scheduler allocates jobs in a consecutive manner, it can estimate the delay for each job and guarantee the maximum profit, which we will discuss in the following section. The transmission sequence determined by the lower-level scheduling may not follow that by the upper-level scheduling due to the opportunistic transmission or other diversity schemes. However, we assume that the lower-level scheduling does not affect the delay bound in a macroscopic scale. Let $d_j$ be the total estimated delay for job $j$. If job $j$ has been already served for $d_j^{\text{cur}}$, $d_j$ can be written as

$$d_j = d_j^{\text{cur}} + d_j^{\text{res}}. \tag{2}$$

Job $j$ will be able to get the utility for the current job at $d_j$. As the utility is not available at present, we call $U_j(d_j)$ the potential utility.

4. Scheduling strategy

4.1. Algorithm

Packet scheduling can select the job with the maximum utility. In this case, some other jobs with low utility may starve. Therefore we introduce the concept of opportunity cost. It is incurred by making a decision and defined as the benefit that the system gives up while carrying out that decision. Here we simply define it as the maximum utility that should be given up among the other users because of their losing the current scheduling turn. For example, consider a case of three jobs; jobs a, b, and c with the potential utility of 3, 5, and 9, respectively. If the scheduler serves job ‘b’ first, the other jobs ‘a’ and ‘c’ will lose their opportunities to achieve the utilities of 3 and 9. Then the opportunity cost is 9 ($=\max(3,9)$) according to the definition. As the time division multiplexing system can give the transmission right to only a job, other jobs should wait for the current job to be complete. If job $j$ is scheduled at the current turn, another backlogged job $i$ with delay $d_i$ experiences utility loss due to job $j$’s residual service time of $d_j^{\text{res}}$. Then the opportunity cost caused by job $j$ can be expressed as

$$C_j = \max_{i \neq j} \left\{ U_i(d_i) - U_i(d_i + d_j^{\text{res}}) \right\} \quad \forall i. \tag{3}$$

We can represent profit $\Phi_j$ by subtracting the opportunity cost from the utility of job $j$

$$\Phi_j = U_j(d_j) - \max_{i \neq j} \left\{ U_i(d_i) - U_i(d_i + d_j^{\text{res}}) \right\}. \tag{4}$$

Our scheduling scheme chooses a job with the maximum profit, that is,

$$Q^* = \arg \max_j \Phi_j. \tag{5}$$

The algorithm is summarized in Fig. 3.

4.2. Scheduling sequences

We now prove that the policy of successive transmission of packets from each job ensures the maximum profit. This is because the utility as a function of delay decreases monotonically. We see the effect of successive transmission in Fig. 4 and simply states it in the following lemma.

**Lemma 1.** Assume that the average data rate for a job is constant and the utility function decreases monotonically. The scheduler can maximize the sum of potential utilities by assigning slots for the job consecutively.

**Proof.** To simplify our proof, we consider a case of two jobs $i$ and $j$. Assume that job $i$ is currently being served. Then the sum of potential utilities is given by $U_i(d_i) + U_j(d_j + d_j^{\text{res}})$. If job $j$ uses one slot during $i$’s service time, the utility sum is given by $U_i(d_i + T_{\text{slot}}) + U_j(d_j + d_j^{\text{res}})$ as job $i$ will be finished one slot later. This sum is smaller than the previous one since the utility function is monotonically decreasing. Similarly, we can easily show that any mixed slot allocation has smaller utility sum than the continuous allocation. □

Fig. 3. A scheduling algorithm based on the opportunity cost.

```plaintext
1 : If (number of users == 0)  
2 : slot idle  
3 : Else if (number of users == 1)  
4 : select  
5 : Else {  
6 : For each user j  
7 : For each user i(\neq j)  
8 : \quad c_i = U_i(d_i) - U_i(d_i + d_j^{\text{res}})  
9 : \quad C_j = \max(c_1, \ldots, c_n)  
10 : \quad \Phi_j = U_j(d_j) - C_j  
11 : select Q^* = \arg \max \Phi_j  
12 : }
```

Fig. 3. A scheduling algorithm based on the opportunity cost.
Lemma 2. The consecutive slot allocation for a job minimizes the sum of opportunity costs.

Proof. Following the approach similar to the proof of Lemma 1, we consider a case of two jobs again. Assume that job $i$ is currently being served and job $j$ is queued. When there remains one slot to complete job $i$, we can represent the opportunity cost $C_A$ for job $j$ in case of continuous slot assignment as

$$C_A = U_j\left(d_{i\text{cur}} + T_{\text{slot}} + d_{j\text{res}}\right) - U_j\left(d_{i\text{cur}} + 2T_{\text{slot}} + d_{j\text{res}}\right),$$

where $d_{j\text{res}}$ indicates the delay to transmit its residual packets minus one slot as shown in Fig. 5(a). If job $i$ swaps a slot with job $j$ like in Fig. 5(b), we obtain the opportunity cost $C_B$ for jobs $i$ and $j$ as

$$C_B = U_j\left(d_{i\text{cur}} + T_{\text{slot}}\right) - U_j\left(d_{i\text{cur}} + 2T_{\text{slot}}\right) + U_j\left(d_{i\text{cur}} + T_{\text{slot}} + d_{j\text{res}}\right) - U_j\left(d_{i\text{cur}} + 2T_{\text{slot}} + d_{j\text{res}}\right).$$

Clearly $C_A \leq C_B$. Therefore more opportunity cost incurs by swapping any two packets like in Fig. 5.

From these lemmas, we derive the following proposition.

Proposition 1. The consecutive slot assignment for a job maximizes the total profit.

According to the proposition, the upper-level scheduler targets completing the current job first.\(^5\)

4.3. Delay bound

Fig. 2 introduced $g_{\text{offset}}$ to reflect the delay bound in the utility function. It is desirable to give priority to a job that is likely to experience dropping due to its delay bound. Then the question is how we set $g_{\text{offset}}$ at a proper value in order to avoid the loss. We can obtain a lower bound of $g_{\text{offset}}$ by using the following proposition.

Proposition 2. If job $j$ that will be dropped at the next turn due to the delay bound has priority over other jobs, the condition of $g_{\text{offset}}$ larger than $2\{\sup_{x} U_x(0) - \inf_{x} U_x(D_x)\}$ is sufficient to guarantee the delay bound $D_j$ for job $j$.

Proof. Assume that job $j$ is the closest to the delay bound. Regarding another job $i$ that has the maximum utility, $g_{\text{offset}}$ satisfies the following inequality according to the design principles of a utility function:

$$U_i(d_i) - U_j(d_i + d_{i\text{res}}) \geq g_{\text{offset}}.$$

In order to give priority to job $j$, the profit of job $j$ should be larger than that of any other job. If the opportunity cost of job $j$ has been incurred by job $k$, the relationship between the profits of $j$ and $i$ is given by

$$U_j(d_j) - \left\{U_k(d_k) - U_k\left(d_k + d_{i\text{res}}\right)\right\} \geq U_j(d_j) - \left\{U_j(d_j) - U_j\left(d_j + d_{i\text{res}}\right)\right\}.\quad (9)$$

Then we need to find a minimum value of $g_{\text{offset}}$ that satisfies the following condition:

$$g_{\text{offset}} \geq U_i(d_i) - U_j(d_j) + U_k(d_k) - U_k\left(d_k + d_{i\text{res}}\right).\quad (10)$$

As the utility decreases with the increase of delay in the range $[0, D]$, a maximal value of the right-hand term is given by

$$2\left\{\sup_{x} U_x(0) - \inf_{x} U_x(D_x)\right\} \geq U_i(d_i) - U_j(d_j) + U_k(d_k) - U_k\left(d_k + d_{i\text{res}}\right).\quad (11)$$

\(^5\) Some jobs with long completion time would starve some other traffic. Therefore, in real applications, the maximum job size that can be served at a time needs to be given in advance.
Hence we obtain
\[ g_{\text{offset}} \geq 2G, \quad (12) \]
where
\[ G \triangleq \sup_x U_x(0) - \inf_x U_x(D_x). \quad (13) \]

In the above proposition, we can interpret \( G \) as the largest opportunity cost possibly incurred. To give priority to a job that is likely to experience dropping due to its delay bound, we set the offset at a value of \( 2G \). Note that this does not account for the loss that more than two jobs approach their delay bounds at the same time under heavy loading.

5. Simulation results

5.1. Simulation environments

For simulations, we consider a cluster of seven hexagonal cells as shown in Fig. 6. The six neighboring cells generate signals interfering with the center cell of which performance we are interested in. The cell radius is 500 m and users are grouped according to the channel condition. The channel experiences path loss as \( d^\alpha \), where \( d \) is the distance of the user from the base-station and \( \alpha \) is the exponent that we set equal to 4 approximately for our simulation. The channel also experiences Rayleigh fading that changes slowly for nomadic users. For simplicity, we consider delay and loss at the wireless link only.

We use WAP traffic which has the packet size distribution of truncated Pareto with the mean 256 bytes and the maximum 1400 bytes whose cumulative density function is given by
\[ F(x) = \begin{cases} 1 - \left( \frac{x}{k} \right)^2 & x \geq k \geq 0, \\ 0 & x < k, \end{cases} \quad (14) \]

\[ \text{Loading factor} = \frac{\text{total number of transmitted packets}}{\text{total number of generated packets}}. \quad (15) \]

We introduce the loading factor in order to measure the loading condition in a Pareto distribution. A loading factor of greater than one is considered congestion generally. In some cases, however, packets can be dropped due to the delay bound at a loading factor of smaller than one.

We consider a single broadband link that is shared by all users in a cell. The transmission rate for each user is selected according to the physical layer technology and the channel condition. A BS estimates the channel condition by using the measurement feedback from each mobile terminal. We do not consider the effect of lower-level scheduling in utility and opportunity cost. To do so, we assume that the lower-level scheduling simply follows the service sequence determined by the upper-level scheduling. This assumption is acceptable under nomadic environments where channels are varying slowly. Channel coding, hybrid ARQ, 4-slot interlacing, and various diversity schemes are combined for successful transmission in the error-prone wireless channel [1], but we did not consider any diversity or retransmission schemes in the simulation to measure the performance of upper-level scheduling intensively.\(^7\)

The data rate can be appropriately selected by the well-known relationship between transmission power and data rate [21],
\[ \frac{S}{I} = \frac{E_b}{I_0} \frac{R}{W}. \quad (16) \]

where \( S \) and \( I \) represent the signal power and the noise power, respectively. \( E_b \) is the received energy per bit, \( I_0 \) is the total interference per hertz, \( R \) is the transmission rate, and \( W \) is the system bandwidth of the downlink channel. \( E_b/I_0 \) is determined by a desired frame error rate (FER), which depends on the modulation method. The system can control either \( S \) or \( R \): controlling \( S \) to obtain its desired rate

\( ^6 \) WAP traffic model includes some reading time for user reaction and uplink access.

\( ^7 \) Those schemes with lower-level scheduling could enhance the performance. It is beyond the scope of this work.
or adjusting $R$ according to the measured SINR. The latter one, we used here, can be implemented using the adaptive modulation and coding (AMC) scheme which is a key technique for opportunistic scheduling.

In our simulation, each BS transmits at the full power of 20 W and the slot time is 1.67 ms. Table 2 summarizes the emulated modulation parameters according to the EV-DO specification [1]. The parameter $S$ indicates the number of consecutive slots to transmit $B$ bits. The data rate is determined by (16), but falls into one of discrete values as shown in Table 2 in real implementation. As stated before, a data rate that is known at the start of transmission of a job remains constant until the job is complete.

We evaluated our algorithms with respect to various utility functions through simulations. Our finding is that all the tendencies are the same only if the utility functions are convex and monotonically decreasing. Thus we only present the results for the utility function of $\exp(-x/D) \ (0 < x < D)$.

5.2. Performance evaluations

For performance comparison, we compared our scheme with round robin (RR) and EDF schedulers. The loading factor is 1.2, the delay bound is 300 slots, and all users have the same conditions. Since EDF is aware of each job’s deadline, it serves most packets near the deadline as shown in Fig. 7(a). In contrast, our scheduler, labeled by “Proposed (U)”, completes most of jobs in a short time. Fig. 7(b) demonstrates that, in terms of loss rate, our scheme performs best and EDF performs slightly better than RR.

Table 2

<table>
<thead>
<tr>
<th>Data rate (kbps)</th>
<th>$S$</th>
<th>$B$</th>
<th>Modulation type</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.4</td>
<td>16</td>
<td>1024</td>
<td>QPSK</td>
</tr>
<tr>
<td>76.8</td>
<td>8</td>
<td>1024</td>
<td>QPSK</td>
</tr>
<tr>
<td>153.6</td>
<td>4</td>
<td>1024</td>
<td>QPSK</td>
</tr>
<tr>
<td>307.2</td>
<td>2</td>
<td>1024</td>
<td>QPSK</td>
</tr>
<tr>
<td>614.4</td>
<td>1</td>
<td>1024</td>
<td>QPSK</td>
</tr>
<tr>
<td>307.2</td>
<td>4</td>
<td>2048</td>
<td>QPSK</td>
</tr>
<tr>
<td>614.4</td>
<td>2</td>
<td>2048</td>
<td>QPSK</td>
</tr>
<tr>
<td>1228.8</td>
<td>1</td>
<td>2048</td>
<td>QPSK</td>
</tr>
<tr>
<td>921.7</td>
<td>2</td>
<td>3072</td>
<td>8-PSK</td>
</tr>
<tr>
<td>1843.2</td>
<td>1</td>
<td>3072</td>
<td>8-PSK</td>
</tr>
<tr>
<td>1228.8</td>
<td>2</td>
<td>4096</td>
<td>16-QAM</td>
</tr>
<tr>
<td>2457.8</td>
<td>1</td>
<td>4096</td>
<td>16-QAM</td>
</tr>
</tbody>
</table>

Fig. 7(a) and (b) also show the impact of lower-level scheduling by labeling “Proposed (U + L)”. Since the lower-level scheduling may not follow the service sequence that has been determined by the upper-level scheduling, we rearranged the actual service sequence randomly every eight consecutive jobs to reflect time varying environments. However, the results show that the impact of rearrangement on the overall performances of delay and loss rate is minor. This verifies our assumption that the lower-level scheduler behaves in a microscopic scale without hurting the overall performance significantly. We leave extensive investigation about this issue for future work and focus on the upper-level scheduling.

We investigated the performance according to the delay bound under the similar channel condition and the loading factor of 1.2. In the considered utility function of $\exp(-x/D) \ (0 < x < D)$, the delay bound $D$ is set at 20, 40, 60, 80, and 100 (slots),
respectively, for groups A, B, C, D, and E. For various delay bounds, Fig. 8 shows the delay distribution and loss rate of each group. In Fig. 8(a), the number of jobs is saturated after each job reaches its delay bound, decreases from groups A to C, and increases from groups C to E. This trend is the same in the loss rate.

This is because our scheduling algorithm considers the factors of utility and opportunity cost together. Group A generates the largest opportunity cost compared to the other groups while achieving the smallest utility. Conversely, group E achieves high utility and results in low opportunity cost, so it has more chances to transmit its packets. Hence, the utility factor dominates overall performance towards group E while the opportunity cost factor does towards group A. These results verify that the overall performance heavily depends on the utility function.

In this case, we compared our scheme with RR and EDF as shown in Fig. 8(b). An interesting result is that EDF experiences unfair loss rate according to the delay bound under heavy loading. This is because jobs with long delay bound lose their chances to be served while others with short delay bound are continuously being served. On the other hand, our scheme adaptively works according to the utility functions. Groups A and E have small loss rates due to the utility factor and the opportunity cost factor, respectively.

We validate Proposition 2 by observing the performance of our scheme according to $g_{\text{offset}}/G$. We fix the delay bound at 300 slots, and the other conditions are the same as before. In Fig. 9(a), we obtain improved performance by setting $g_{\text{offset}}/G$ high. The number of served jobs increases sharply near the delay bound because of the effect of the opportunity cost, and the magnitude of increase grows larger with $g_{\text{offset}}/G$. However, the loss rate and the throughput are saturated when $g_{\text{offset}}/G \geq 1$ as shown in Fig. 9(b). The gap comes from that the right term in (11) approximately approaches one in the average sense but we assumed a worst case in proving Proposition 2.

We obtain simulation results for various channel conditions under the same delay bound $D = 300$ (slots) and $g_{\text{offset}}/G = 2$. We classify users into four groups in order to represent group 1 as the users of best channel condition (i.e., SINR) and group 2.
has the second best condition and so on. Fig. 10 shows the performances of four groups under different channel conditions. Under the good channel, the number of completed jobs increases while it decreases under the bad condition. Particularly in the bad condition, the delay bound affects overall performance very much. Similarly, the good channel user group has lower loss rate than the bad one.

We examine the feasibility of supporting different service quality according to the traffic class. We simply consider a case of two traffics, video streaming and WAP. The video streaming traffic model has the following characteristics [5]: (1) frame inter-arrival time is fixed at 100 ms; (2) number of packets in a frame is 8; (3) packet size follows a truncated Pareto with $k = 20$ and $\alpha = 1.2$; (4) packet inter-arrival time within a frame follows a truncated Pareto with $k = 2.5$ and $\alpha = 1.2$. We fixed the other parameters at the same values as before except the utility function and the delay bound. We used the utility functions in Fig. 11(a) where WAP has a larger delay bound than video streaming. The video streaming has the utility function of $3\exp(-x/D)$ $(0 < x < D)$, where $D$ is 60, 90, and 120 (slots), respectively, for video1, 2, and 3. Fig. 11(b) shows that our scheme operates well regardless of the traffic type. Like in Fig. 8(a), the performance of video1 is mainly affected by the opportunity cost while that of video3 by the utility.

6. Conclusions

In this paper, we investigated a new downlink scheduling scheme for wireless data networks. We decoupled the scheduling function into two levels: upper-level and lower-level. Each level of scheduling plays the role of QoS guarantee and opportunistic transmission, respectively, in the network and physical layers. Our upper-level scheduler supports multimedia traffic by an integrated approach that uses the concept of opportunity cost. It used utility functions that consider the delay bound and loss rate as the cost metrics to meet various QoS requirements. The simulation results confirmed that our algorithm performs well under various conditions. The scheduler was also able to provide prioritized service...
according to the delay bound. The interaction between upper-level and lower-level schedulings to enhance overall performance is left for future work. It is an open issue to design more comprehensive utility functions that can support best-effort traffic with some other metrics.

References


Young-June Choi is currently a post-doctoral researcher in the School of Electrical Engineering and Computer Science, Seoul National University. He received his B.S., M.S., and Ph.D. degrees from Seoul National University, in 2000, 2002, and 2006, respectively. His research interests include fourth generation wireless networks, wireless resource management, and cross layer system design.

Jin-Ghoo Choi received the B.S., M.S., and Ph.D. degree in the School of Electrical Engineering & Computer Science, Seoul National University, in 1998, 2000, and 2005, respectively. He is a senior engineer in Samsung Electronics. His research interests include resource management and packet scheduling in wireless networks.

Saewoong Bahk received B.S. and M.B. degrees in Electrical Engineering from Seoul National University in 1984 and 1986, respectively, and the Ph.D. degree from the University of Pennsylvania in 1991. From 1991 through 1994 he was with the Department of Network Operations Systems at AT&T Bell Laboratories as an MTS where he worked for AT&T network management. In 1994, he joined the school of electrical engineering at Seoul National University and currently serves as a professor. His areas of interests include performance analysis of communication networks and network security.