Scheduling Algorithm based on Sender Buffer Backlog for Real-Time Application in Mobile Packet Networks

Hideyuki Koto†, Masaki Fukushima†, Shinichi Nomoto† and Fumio Takahata‡

†KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Kamifukuoka, Saitama, 356-8502, JAPAN.
TEL: +81 49 278 7625
E-mail: *{koto, fukusima, nomoto}@kddilabs.jp*

‡Graduate School of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku, Tokyo, 169-8555, JAPAN.
TEL: +81 3 5286 3389
E-mail: *fu@takahata.comm.waseda.ac.jp*

Abstract— **The demand for high-speed data communication and the use of real-time applications in mobile networks are becoming increasingly high, and communication systems that specialize in downlink packet transfer are being developed and deployed. In these high-speed downlink packet networks, scheduling at the base station is one of the key technologies to accomplish such high-speed. A scheduling algorithm that is based on the proportional fair criterion is widely used in commercial networks, but the problem with such algorithm is that it does not schedule real-time application's traffic efficiently. This paper proposes a new scheduling algorithm for real-time applications that does not need substantial change in the existing system and coexists efficiently with the proportional fair criterion. We evaluate the characteristics of the proposed algorithm through computer simulation and show that the efficiency of the proposed algorithm is better than that of existing algorithms.**

Keywords; Scheduling, Real-Time Application, Buffer Backlog, Mobile Packet Netowrks, 1x EV-DO

I. INTRODUCTION

In recent years, the demand for high-speed data communications in mobile networks is becoming increasingly high, and communication systems which specialize in downlink packet transfer such as the Internet access are being developed and deployed (HDR for CDMA2000 [1], HSDPA for W-CDMA [2]). In these downlink packet networks, the wireless channel is shared among several users and each user is multiplexed in a time-divisional manner at the base station, where data to be transmitted to the mobile hosts are queued at a buffer allocated to each user. Since several users share one common wireless channel, scheduling which allocates time slot to a specific user is made possible. Furthermore, the channel state of each mobile host varies temporally. So, if the scheduling algorithm takes the channel state into consideration and allocates time slots to a user with good channel state, the system throughput of the mobile packet network could be improved. Hence time slot scheduling at the base station is a key technology for high-speed data communication in mobile networks.

Several scheduling algorithms that take the channel state of mobile hosts into account have been proposed [3]-[5]. One of these scheduling algorithms is the *Proportional Fair* (PF) scheduling rule [3] and is implemented in commercial mobile networks including CDMA2000 1xEV-DO (or HDR [1]). The PF scheduling rule is widely accepted because it is not only efficient but is also simple and it has few parameters that need to be optimized. However, the problem with the PF rule is that it is designed for best effort services, and is not designed to support real-time applications.

Meanwhile, the demand for the use of real-time applications even in mobile networks is increasing [6]. Therefore, enhancement to the current mobile packet systems is needed, and a scheduling algorithm that could efficiently support real-time application's traffic is required. Recently, several researchers have proposed scheduling algorithms for mobile packet systems that support real-time applications [7]-[10]. However, the existing scheduling algorithms for real-time applications have major drawbacks such as that the calculation of the algorithm is complex or the existing system needs to be changed considerably. Furthermore, since the PF scheduling rule is widely accepted in commercial services, scheduling algorithms for real-time applications should coexist efficiently with the PF rule.

This paper proposes a scheduling algorithm for real-time application that does not need substantial change in the existing system and coexists efficiently with the PF criterion. The proposed scheduling algorithm, the *Sender-Buffer-Sensitive* (SB) scheduling rule, schedules time slots based on the sender buffer backlog at the base station, which in general is thought to be correlated with the playout buffer backlog at the receiving terminal that represents the utility of real-time applications. The relation of sender buffer backlog to the amount of transmitted data is also considered. The proposed scheduling algorithm is evaluated through computer simulation.

II. CONVENTIONAL SCHEDULING ALGORITHMS

In this section, we briefly describe the PF scheduling rule, and some scheduling algorithms for mobile packet networks that are designed to support real-time applications.

A. *Proportional Fair Rule*

The *Proportional Fair* (PF) scheduling rule is widely implemented in commercial mobile packet networks including CDMA2000 1xEV-DO [1], [3]. The PF rule assigns time slot to a mobile host whose current channel state is good relative to its mean allocated throughput. Let N be the number of mobile stations, $r_i(n)$ be the feasible rate of user i at slot n and $R_i(n)$ be the mean allocated throughput, then $R_i(n)$ is given by

$$R_i(n+1) = \left(1 - \frac{1}{t_r}\right) R_i(n) + \frac{1}{t_r} 1_{\{i=\text{slot allocated user}\}} r_i(n) \qquad (1)$$

where i denotes *i-th* user among N users, t_r is the smoothing factor in a moving average calculation and $1_{\{\bullet\}}$ is an indicator function. The PF rule allocates time slot to the user with the highest $r_i(n)/R_i(n)$. The PF rule is widely accepted because it is efficient, simple and has only few parameters that need to be optimized. However, the problem with the PF rule is that it is not designed to support real-time applications efficiently.

B. Scheduling Algorithm for Real-Time Applications

Real-time applications in packet switching networks typically have a buffer space called a playout buffer (or jitter buffer) in the receiver's side to mitigate delay variations incurred in the intermediate network. The applications fetch data from the playout buffer at a specific rate. Application QoS (Quality of Service) is degraded if the playout buffer is empty when the application tries to fetch data. Therefore the playout buffer starvation probability is an important performance measure, and hence the urgency of data to be transmitted varies according to the state of the playout buffer backlog; the user with the playout buffer that is going to starve should be preferentially scheduled.

Some scheduling algorithms for mobile packet networks that can support real-time applications have been proposed [7]-[10]. For example, the *Exponential* (EXP) rule [7] equalizes different user's weighted delay when their differences are large and behaves like the PF rule when their differences are small. In [10], utility function based scheduling algorithm called the *Playout-Buffer-Sensitive* (PB) rule is proposed. The PB rule allocates time slot to a user based on the playout buffer backlog. Since playout buffer starvation is a key factor in real-time applications, the PB rule is a sub-optimal resolution for scheduling real-time applications. However the PB rule requires the information of the playout buffer backlog at a certain time; hence a feedback mechanism is needed. So, the problem with the existing algorithms is that the already deployed systems have to be changed substantially. Furthermore, some parameters depending on the real-time application's characteristics (e.g. streaming rate, maximum tolerable delay etc.) have to be configured for the algorithm to operate efficiently However, it is difficult for the scheduling mechanism to know these parameters beforehand because every application have their own original characteristics.

III. PROPOSED SCHEDULING ALGORITHM

In this section, we describe the policy and the details of our proposed scheduling algorithm, the *Sender-Buffer-Sensitive* (SB) scheduling rule.

A. Baseline Policy

First of all, for best effort class applications, it is desirable to use the PF rule as the scheduling algorithm considering the fairness criterion, system capacity, and the degree of system complexity. Furthermore, since the PF rule has already been spread widely and considering implementation simplicity, the modifications in the system introduced by the new scheduling algorithm should be kept minimum.

Next, for real-time class applications, since playout buffer starvation probability is the key factor, the scheduling algorithm should allocate time slots based on the playout buffer backlog, like the PB rule. However, it is difficult for the scheduler to know the amount of the playout buffer backlog without any feedback mechanisms. Generally, there is a correlation between the playout buffer backlog at a mobile host and the sender buffer backlog at the base station. When the sender buffer backlog is small, the playout buffer backlog is generally large, and data does not have to be transmitted immediately. On the other hand, when the sender buffer backlog is large, the playout buffer backlog is generally small, and more data is needed to avoid buffer starvation. Furthermore, urgency of data to be transmitted varies according to previously allocated throughput i.e. for the same amount of sender buffer backlog, urgency after sufficient slot allocation and urgency after a long allocation interval is not the same. So, we propose a scheduling algorithm that allocates time slots based on the relation of the sender buffer backlog to the amount of transmitted data.

Finally, since the proposed scheduling algorithm is for mobile packet networks, the state of the wireless channel needs to be considered.

Overall, the proposed scheduling algorithm is designed considering the following criteria.

- PF rule for best effort class applications
- Modifications in the system are kept minimum
- Relation of the sender buffer backlog at the base station to the amount of transmitted data
- Wireless channel state

B. Introduced Parameters

Considering the above mentioned criteria, we introduce three new parameters in the SB rule, besides mean throughput $R_i(n)$ that is used in the PF rule; the mean wireless channel state $C_i(n)$, the mean sender buffer backlog $S_i(n)$ and the weighting factor $B_i(n)$ based on sender buffer utilization.

The mean wireless channel state $C_i(n)$ is based on $r_i(n)$ and is given by

$$C_i(n+1) = \left(1 - \frac{1}{t_c}\right) C_i(n) + \frac{1}{t_c} r_i(n) \qquad (2)$$

where i denotes *i-th* user among N users, and t_c is the smoothing factor in a moving average calculation. Since $C_i(n)$ is a moving average value updated using $r_i(n)$ irrespective of whether the time slot is allocated to a user or not, $C_i(n)$ represents the mean channel state of each user. In the SB rule, we use the ratio of the current channel state to the average channel state $r_i(n)/C_i(n)$ as a new priority factor for scheduling. Therefore, time slot is preferentially allocated to a user with good instantaneous channel state compared to the mean channel state.

Next, we introduce the mean amount of the sender buffer backlog $S_i(n)$, which is a parameter based on the sender buffer backlog of the real-time class user at the base station. Here, the mean buffer backlog $S_i(n)$ is based on the amount of the current sender buffer backlog $b_i(n)$ and is given by

$$S_i(n+1) = \left(1 - \frac{1}{t_s}\right) S_i(n) + \frac{1}{t_s} b_i(n) \tag{3}$$

where t_s denotes the smoothing factor in a moving average calculation. $S_i(n)$ is a moving average value updated using $b_i(n)$ irrespective of whether the time slot is allocated to a user or not. In the SB rule, we use the ratio of the mean buffer backlog to the mean amount of transmitted data $S_i(n)/R'_i(n)$ as a new priority factor for scheduling, where $R'_i(n)$ is the mean amount of transmitted data. Since $R_i(n)$ is the mean allocated throughput of a user, $R'_i(n)$ could be calculated as follows.

$$R_i'(n) = R_i(n)\Delta t \tag{4}$$

where Δt denotes the length of a time slot.

Furthermore, let r_{In} denote the incoming rate and r_{Out} denote the outgoing rate of a flow at a node. Ignoring the prior accumulation of data at such node, the buffered amount of data $b_{Buffered}$ during one time slot could be shown as

$$b_{Buffered} = (r_{In} - r_{Out})\Delta t \tag{5}$$

and if we divide the above equation by $r_{Out}\,\Delta t$, we get

$$\frac{b_{Buffered}}{r_{Out}\Delta t} = \frac{r_{In}}{r_{Out}} - 1. \tag{6}$$

Since $R'_i(n)$ represents the mean amount of data that is transmitted and $S_i(n)$ represents the mean amount of buffer backlog during a certain time period, the parameter $S_i(n)/R'_i(n)$ is an approximation of the above equation, and therefore this parameter also represents the ratio between the incoming rate and the outgoing rate. So, by introducing this parameter, time slot is preferentially allocated to a user when the incoming rate is large compared to the outgoing rate.

Finally, we introduce a weighting factor $B_i(n)$ based on the sender buffer utilization. The weighting factor $B_i(n)$ is given by

$$B_i(n+1) = \begin{cases} (1-\frac{1}{t_b})B_i(n)+\frac{1}{t_b}\times\frac{b_i(n)}{b_i^{high}b_i^{\max}} \quad , & \frac{b_i(n)}{b_i^{\max}} > b_i^{high} \\ B_i(n) \quad , & b_i^{low} \le \frac{b_i(n)}{b_i^{\max}} \le b_i^{high} \\ \max\left\{(1-\frac{1}{t_b})B_i(n), 1\right\} \quad , & \frac{b_i(n)}{b_i^{\max}} < b_i^{low} \end{cases} \tag{7}$$

where b_i^{max} denotes the maximum buffer backlog, b_i^{high} and b_i^{low} are $0 \le b_i^{high}, b_i^{low} \le 1$, and t_b denotes the smoothing factor in a moving average calculation. Here $b_i(n)/b_i^{max}$ is the buffer utilization. $B_i(n)$ is the weighting factor when buffered data exists more than a certain limit, and $B_i(0) = 1$. The parameters b_i^{high} and b_i^{low} determine the range in which the buffer usage is controlled. Furthermore, t_b is set smaller than t_s to respond to a rapid change in the amount of buffer backlog.

The introduced parameters do not need substantial change to the currently deployed system, such as a new feedback mechanism from a receiving terminal, and only the information of sender buffer backlog and its capacity at the base station is needed. Also, the parameters are independent from the characteristics of real-time applications and do not have to be configured for the algorithm to operate efficiently. Furthermore, although $C_i(n)$ is calculated for all users, $S_i(n)$ and $B_i(n)$ is used only for real-time class scheduling, so there is no need to calculate these parameters for best effort class.

C. *Proposed Scheduling Algorithm*

Using the introduced parameters, the proposed scheduling algorithm is as follows. Depending on the priority factors calculated, real-time class mobile hosts are not always preferentially scheduled by the SB rule.

(1) Calculate $\frac{r_i(n)}{R_i(n)}$ for all best effort class mobile hosts.

(2) Calculate $\frac{r_i(n)}{C_i(n)}$ as the priority factor of the hosts that has the highest value calculated in (1).

(3) Calculate $\frac{r_i(n)}{C_i(n)} \times \frac{S_i(n)}{R_i'(n)} \times B_i(n)$ as the priority factor for all real-time class mobile hosts.

(4) Compare the priority factors calculated in (2) and (3), and allocate time slot to the host with the highest value.

(5) Allocate time slot n, and update $R_i(n)$ and $C_i(n)$ using $r_i(n)$, and update $S_i(n)$ and $B_i(n)$ using $b_i(n)$.

(6) Go back to (1) with $n \rightarrow n+1$.

IV. Simulation Model

We evaluate the SB rule by computer simulation using the network simulator ns-2 [11], with wireless channel fluctuation, time slotted link and various scheduling algorithms implemented. By using ns-2, we could evaluate the effects of link-level simulation of the wireless channel and time slot scheduling, and the network level simulation of TCP/IP (Transmission Control Protocol/Internet Protocol) [12] protocols together. The performance of the SB rule is evaluated compared to other existing algorithms. We also compare the efficiency of the algorithm with a theoretical upper bound.

Since scheduling algorithms operate on a relatively smaller time-scale than the shadowing process, the average CIR (Carrier-to-Interference power Ratio) of each host is assumed to be constant. Multipath fading is simulated using the Jakes model [13]. The parameters are determined according to 1xEV-DO. Since 1xEV-DO is a wideband wireless system (carrier frequency is 2 GHz and signal frequency is 1.25 MHz), we apply frequency selective fading with exponential power delay profile for the channel model. Doppler frequency is 10 Hz which corresponds to a velocity of about 5.4

km/h in 1xEV-DO. Time slot duration Δt = 1/600 second. Instantaneous CIR is determined by the above-mentioned fading simulator and feasible rate is determined from the instantaneous CIR according to the data shown in [1]. Sender buffer capacity b_i^{max} = 32 kbyte, playout buffer capacity is set equal to b_i^{max}, and b_i^{high} and b_i^{low} for the SB rule are set as b_i^{high} = 0.5 and b_i^{low} = 0.2, for all hosts. The parameters in the moving average calculations are set as $t_r = t_c = t_s = 1000$ and t_b = 100. For comparison, we also run simulations for the PF rule, the EXP rule (based on token queues [7]), and the PB rule.

The real-time class mobile hosts apply "initial buffering" regardless of the scheduling algorithm; the real-time class mobile hosts do not fetch data from the playout buffer until the buffer usage exceeds 90% of the buffer capacity. When the buffer usage exceeds the threshold, the application starts to consume the buffered data, e.g. playout video streaming.

The network model for evaluating the performances of the scheduling algorithms consists of a wired and wireless mixed scenario with data transmitted from several sources through a wired link toward mobile hosts via a base station and a wireless channel. The TCP variant employed here is TCP SACK [14]. The maximum window size for TCP is set as 32 kbyte. The traffic models used are FTP (File Transfer Protocol) file transfer for the best effort class application and CBR (Constant Bit Rate) streaming using UDP (User Datagram Protocol) for the real-time class application, all of which are of greedy sources i.e. infinite backlog at the data source. The data rate of real-time application is 64 kbit/s. Packets arriving at the base station are fragmented into frames of 1024 bits and sent to each mobile host. When all the frames consisting one packet are received at a mobile host, frames are reassembled to a packet. To focus on the fundamental performance of the scheduling algorithms, no congestion (buffer overflow at intermediate nodes) is induced in the simulation. In addition to that, since the system of 1xEV-DO is designed to keep the PER (Packet Error Rate) at low level (approximately 1%), transmission errors are also not considered. Classification of the best effort and the real-time flows is assumed as ideal.

V. SIMULATION RESULTS

A. *Overall Performance*

Fig. 1 shows the performance comparison of throughput versus the number of real-time class mobile hosts for the various scheduling algorithms. BE and RT in the figure denotes the total throughput for best effort class mobile hosts and real-time class mobile hosts respectively. Total denotes the sum of BE and RT throughput which represents the system throughput. The average CIR of all the hosts is 0 dB, which is the typical value in CDMA systems. The total number of mobile hosts is 8. The number of real-time class mobile hosts is varied from 0, i.e. best effort class only, to 8 i.e. real-time class only. We evaluate the performance using sufficiently long simulations of 6,000,000 time slots.

Although the results are not shown in this paper, we verify that the playout buffer starvation probabilities are negligibly small for all algorithms for the duration of the simulation. Also, from the figure, the throughput of the real-time class is proportional to the number of real-time class mobile hosts, irrespective of the type of the scheduling algorithm. This is because that all the scheduling algorithms achieve the required throughput of the real–time class applications due to sufficiently large average CIR and therefore feasible rates, compared to the CBR rate. Furthermore, when there are only real-time class hosts in the system, the results for all the algorithms converge to the same value. The reason for this is also due to the sufficiently large feasible rates compared to the CBR rate; after the required rate has been achieved, there are no more data to send even if extra time slots are allocated.

Next, although the results for the EXP rule is better than that of the PF rule, the throughput of the best effort class for both rules decreases in proportion to the increase of the real-time class mobile hosts. As a result, the total system throughput also decreases for both rules. This demonstrates that these algorithms cannot efficiently schedule real-time class applications. On the other hand, the decrease of the total throughput under the PB and the SB rule are small compared to the PF and the EXP rule. This result shows that the PB and the SB rule can support real-time class applications efficiently. Furthermore, although the SB rule does not have the information about the playout buffer backlog, buffer starvation probability is negligible, and the system throughput is almost the same and even better than the PB rule when the number of real-time class is 7. The reasons for the obtained results are described in the next section.

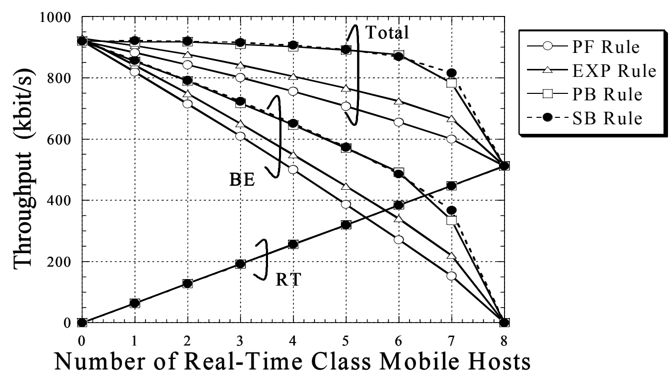


Fig. 1. Throughput of Scheduling Algorithms.

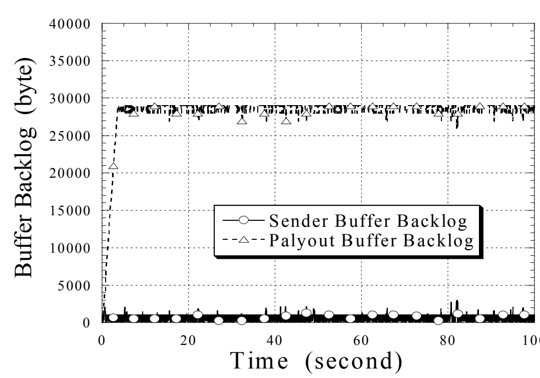


Fig. 2. Example of PF rule.

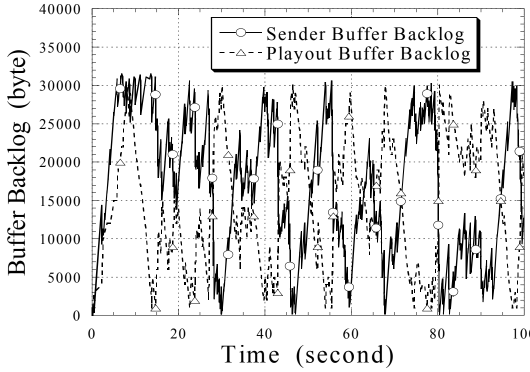


Fig. 3. Example of PB rule.

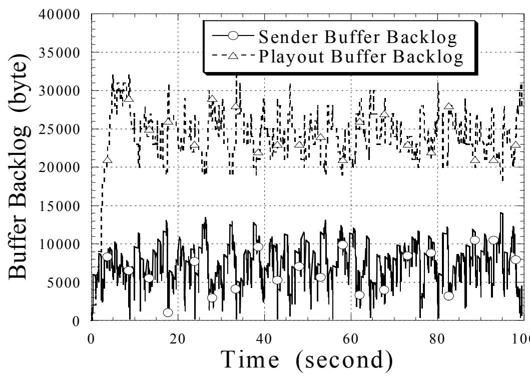
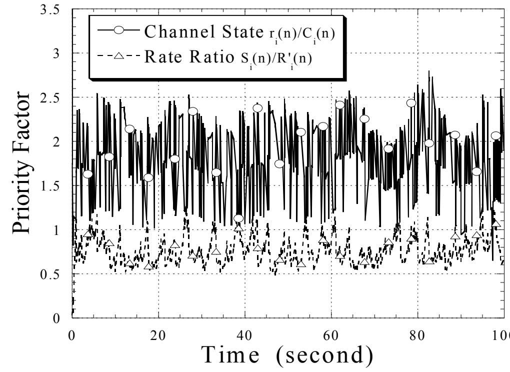
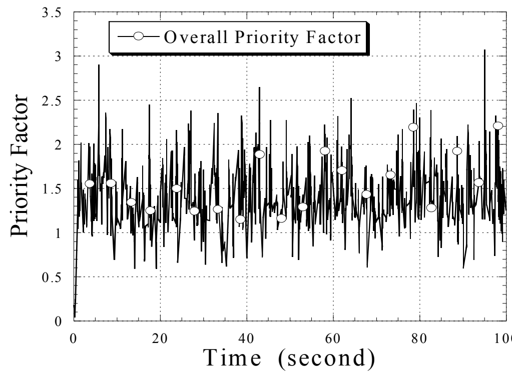


Fig. 4. Example of SB rule.



(a). Elements of Priority Factor.



(b). Overall Priority Factor.

Fig. 5. Priority Factor of SB Rule.

B. Detailed Observation

Fig. 2 shows the example of buffer backlog of a real-time class mobile host for the PF rule with 8 mobile hosts, of which 4 are best effort class and 4 are real-time class. From the figure, we could see that the variation of both the sender and the playout buffer backlog are small, and the playout buffer backlog remain around the buffering threshold of real-time applications. Since the average CIR is sufficiently high, time slots are allocated to the real-time class mobile hosts as soon as data arrive at the base station. Consequently, slot allocations with the PF rule are almost channel-insensitive. The result for the EXP rule shows nearly the same overall characteristic as the PF rule because the EXP rule behaves like the PF rule when the differences of the weighted delay among hosts are small.

Next, Figs. 3 and 4 show examples of buffer backlog for the PB and the SB rule respectively, with the same simulation conditions. From the results, we could see that the buffer backlog fluctuate within a certain range, and this means that time slots are efficiently allocated to real-time class mobile hosts in a channel-sensitive manner. That is, both rules efficiently utilize the backlog fluctuation and the capacity of the buffer, therefore increasing the number of time slots allocated to best effort class mobile hosts. However, although the backlog fluctuation for the SB rule is smaller than that of the PB rule, the best system throughput is obtained by the SB rule. The reasons for the obtained results are as follows.

The algorithm of the PB rule is described in detail in [10], and it operates to keep the playout buffer backlog in a prescribed range. Fig. 3 shows this characteristic, and here the backlog fluctuates within a range of $0.1b_i^{max}$ and $0.9b_i^{max}$. When the backlog is in such a range, the PB rule allocates time slots depending on the wireless channel state and the priority factor based on the amount of the playout buffer backlog. However, the value of this factor is not varied unless the backlog exceeds the prescribed range. When the backlog of a host overflows the upper threshold, the priority factor is decreased. But when the backlog underflows the lower threshold, time slots are preferentially allocated virtually irrespective of the wireless channel state. As a result, the fluctuation of the backlog tends to become large, and in such a case the algorithm operates somewhat inefficiently.

Figs. 5(a) and (b) show the fluctuation of the priority factors of a real-time class for the SB rule. Here, the weighting factor $B_i(n)$ is 1 for the duration of the simulation and is not plotted on the graph. The reason for this is because the sender buffer utilization at the base station does not exceed the configured range, and therefore $B_i(n)$ is not increased. From Fig. 5(a), we could see that the priority factor for channel state, $r_i(n)/C_i(n)$, and the priority factor for rate ratio, $S_i(n)/R'_i(n)$, fluctuate with time. The factor $r_i(n)/C_i(n)$ fluctuates in response to the state of the wireless channel that is determined by the value of the average CIR and fading level. The factor $S_i(n)/R'_i(n)$ fluctuates in response to the ratio between the incoming rate and the outgoing rate at the base station. The product of these three factors is the overall priority factor for the real-time class mobile hosts in the SB rule. Fig. 5(b) shows the overall priority factor of a real-time class for the SB rule. As shown in the figure, when $r_i(n)/C_i(n)$ is large i.e. when the current wireless channel state is good compared to its mean, the priority becomes large and time slots are allocated preferentially. But when $S_i(n)/R'_i(n)$ is small i.e. when the allocated outgoing throughput is large compared to the amount of buffered data, the priority of such host decreases even if the channel state is relatively good. So, the SB rule schedules time slots based not only on the actual amount of the buffer backlog, but also on the relation with the amount of transmitted data. Consequently, the SB rule operates more delicate scheduling than the PB rule, therefore resulting in the improvement of throughput and efficiency.

Furthermore, although the results are not shown here, the SB rule showed the best results when the data rate of real-time application is increased to 128 kbit/s or when the sender buffer capacity is decreased to 16 kbyte. At these conditions, the weighting factor $B_i(n)$ plays an important role. Since $B_i(n)$ is based on sender buffer utilization, when data rate of real-time application is increased or when sender buffer capacity is decreased, it fluctuates accordingly, resulting in the preferential allocation of time slots to real-time class. We also confirmed that the SB rule operates well even with variable rate real-time application using actual traffic traces as traffic models.

C. Theoretical Upper Bound

As described in [10], it is difficult to predict the performance of the optimal scheduling algorithm, because it is the optimal solution of a large combinatorial optimization problem. Therefore, the performance of the algorithms is compared to a theoretical upper bound.

Real-time class mobile hosts require fixed target throughput and the throughput of the best effort class is an increasing function of time slots allocated for the best effort class. Accordingly, we adopt the fraction of time slots allocated for the best effort class as a performance measure.

We assume $r_i(n)$ is a stationary process. Let $F_i(r)$ be the stationary distribution function of $r_i(n)$. We define a lower bound of fraction of time slots allocated for a real-time class mobile host i as p_i, which satisfies

$$\text{Data rate of RT application} = \int_0^\infty 1_{\{F_i(r)>1-p_i\}} r dF_i(r). \quad (8)$$

An upper bound of fraction of time slots allocated for best effort class mobile can be defined as

$$1-\sum_{i=1}^{N_{RT}} p_i \quad (9)$$

where N_{RT} is the number of real-time class mobile hosts.

Fig. 6 shows the upper bound computed numerically and the fraction of time slots actually allocated for the best effort class mobile hosts in the simulation for each scheduling algorithm. From the result, we could see that the efficiency of the

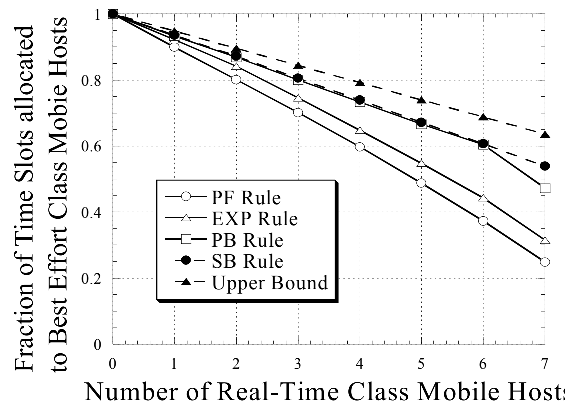


Fig. 6. Fraction of Time Slots allocated to Best Effort Class

SB rule is closer to the theoretical upper bound and is even better than the PB rule that is based on the amount of the playout buffer backlog. The reason for this result is the same as the reason for the performance of system throughput; the SB rule is based not only on the actual amount of the buffer backlog but also on the relation with the amount of transmitted data and operates more delicate scheduling than the PB rule, therefore resulting in the improvement of throughput and efficiency.

D. Interaction with TCP Congestion Control

Here, we investigate whether TCP congestion control has any effects on the performances of the scheduling algorithms.

Fig. 7 shows the performance comparison of throughput versus the number of real-time class mobile hosts for the various scheduling algorithms, when the maximum window size of TCP is set as 64 kbyte. At this value, since the capacity of the sender buffer at the base station is set smaller at 32 kbyte, congestion occurs and sender buffer overflows. From the figure, we could see that the SB rule achieves the best result. In fact, the result of Fig. 7 is almost the same as the result of Fig. 1 i.e. when the maximum window size is set as 32 kbyte. This result shows that the SB rule and other scheduling algorithms are not affected by the congestion control of the TCP protocol. The reason for this is as follows.

Fig. 8 shows the example of sender buffer backlog and TCP congestion window size of a best effort class mobile host for the SB rule with 8 mobile hosts, of which 4 are best effort class and 4 are real-time class. Although the figure shows the example for the SB rule, the results for other algorithms show almost the same characteristics. From the figure, we could see that the sender buffer backlog and the congestion window size fluctuate. Since the maximum window size is set larger than the capacity of the buffer at the base station, buffer overflows and TCP segments are lost due to congestion. TCP protocol tries to avoid congestion by decreasing its congestion window size. When congestion window size is decreased, the amount of data that is transmitted from the FTP source is de creased, resulting in the fluctuation of the sender buffer backlog.

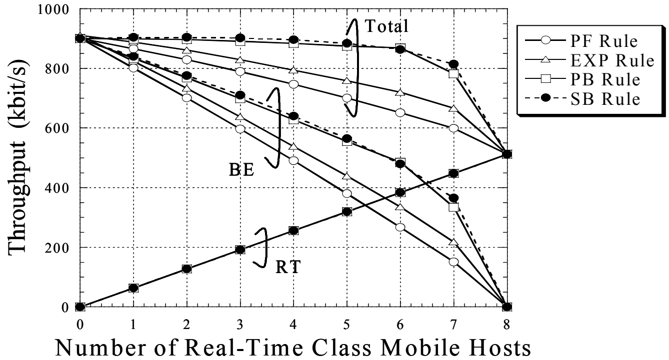


Fig. 7. Throughput of Scheduling Algorithms when Congestion is Induced.

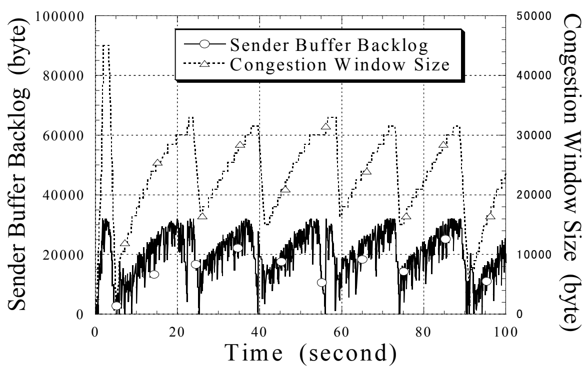


Fig. 8. Example of TCP Congestion Control.

However, the algorithms for the PF rule, the EXP rule, and the PB rule are not based on the amount of the sender buffer backlog, and hence their performances are not affected by the congestion control of the TCP protocol. Furthermore, although the algorithm of the SB rule for the real-time class is based on the amount of the sender buffer backlog, the algorithm for the best effort class is not, and the same algorithm as the PF rule is applied i.e. only the state of the wireless channel and the mean allocated throughput are considered. Therefore, although the sender buffer backlog for the best effort class fluctuates due to congestion, the performance for the SB rule is not affected. As a result, the SB rule is not affected by the congestion control of the TCP protocol.

VI. CONCLUSION

This paper proposes a scheduling algorithm, the *Sender-Buffer-Sensitive* (SB) scheduling rule, for real-time applications in mobile packet networks. The SB rule allocates time slots depending on the wireless channel state, the mean allocated throughput of real-time class mobile hosts, and the amount of the sender buffer backlog at the base station, which in general is thought to be correlated with the playout buffer backlog at the receiving terminal that represents the utility of real-time applications. The results of computer simulations, including fading channel fluctuations and network protocol interactions, show that the SB rule efficiently allocates time slots and achieves high system throughput, even though the information of the playout buffer backlog is not known.

The SB rule is practical and easy to implement because it does not require substantial change in the existing commercial systems such as a feedback mechanism, and co-exists efficiently with the PF criterion. Furthermore, the SB rule is effective because the algorithm is not affected by the congestion control of the TCP protocol, and because the parameters introduced by the SB rule are independent from real-time application's characteristics.

REFERENCES

[1] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: A Bandwidth-Efficient High Speed Wireless Data Service for Nomadic Users," IEEE Communications Magazine, July 2000.

[2] 3GPP, "UTRA High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2," Technical Specification 25.308, Version 6.2.0, September 2004.

[3] A. Jalali, R. Padovani and R Pankaj, "Data Throughput of CDMA-HDR a High Efficiency High Data Rate Personal Communication Wireless System," In Proceedings of IEEE VTC 2000 Spring, May 2000.

[4] S. Borst and P. Whiting, "Dynamic Channel-Sensitive Scheduling Algorithms for Wireless Data Throughput Optimization," IEEE Transactions on Vehicular Technology, Vol. 52, No. 3, May 2003.

[5] M. Jeong, H. Morikawa and T. Aoyama, "A Fair Scheduling Algorithms for Wireless Packet Networks," IEICE Transactions on Fundamental Theory, Vol. E84-A, No. 7, July 2001.

[6] S. Ohmori, Y. Yamao and N. Nakajima, "The Future Generations of Mobile Communications Based on Braodband Access Technologies," IEEE Communications Magazine, December 2000.

[7] S. Shakkottai and A. L. Stoylar, "Scheduling Algorithms for a Mixture of Real-Time and Non-Real-Time Data in HDR," Bell Laboratories Technical Report, 2000.

[8] M. Andrews, K. Kumaran, K. Ramanan, A. Stoylar and P. Whiting, "Providing Quality of Service over a Shared Wireless Link," IEEE Communications Magazine, February 2001.

[9] F. H. P. Fitzek and M. Reisslein, "A Prefetching Protocol for Continuous Media Streaming in Wireless Environments," IEEE Journal on Selected Areas in Communications, Vol. 19, No. 10, October 2001.

[10] M. Fukushima, H. Nakamura and S. Nomoto, "A Playout-Buffer-Sensitive Time Slot Scheduling for Integration of Real-Time and Elastic Traffic in Wireless Networks," In Proceedings of IEEE PIMRC 2004, September 2004.

[11] http://www.isi.edu/nsnam, "The Network Simulator – ns-2".

[12] W.R.Stevens, "TCP/IP Illustrated, Volume 1, The Protocols," Addison-Wesley Publishing Company, 1994.

[13] W.C.Jakes, "Microwave mobile communications," Wiley, 1974.

[14] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and SACK TCP," Computer Communication Review, Vol. 26, pp.5-21, July 1996.