Delay and data rate decoupled fair queueing for multimedia services in wireless networks

Suwon Lee\textsuperscript{a,*}, Kiseon Kim\textsuperscript{a}, Aftab Ahmad\textsuperscript{b}

\textsuperscript{a}Multimedia Communication System Laboratory, Department of Information and Communications, Kwang-Ju Institute of Science and Technology (K-JIST), 1 Oryong Dong, Puk Gu, Kwang-ju 500-712, South Korea

\textsuperscript{b}School of Computer Science, Telecommunications and Information Systems, DePaul University, 243 South Wabash Ave., Chicago IL, USA

Received 21 January 2003; accepted 21 January 2003

Abstract

This paper proposes a delay and data rate decoupled fair queueing scheme for wire-less multimedia, where two separate scheduling algorithms are applied respectively for real time and non-real time flows. For real time flows, we propose an explicit-delay-guarantee algorithm to guarantee the delay deadline, while a conventional fair queuing algorithm is adopted for non-real time flows to guarantee fairness and throughput requirements. In erroneous wireless channel, resource swapping is further adopted to guarantee the required quality of service. In addition, a real time flow can easily react to varying channel error condition by controlling the trans-mission order using allocation window and accelerating factor without degrading the performance of other real time flows. Through simulations, it is shown that the proposed scheme provides not only improved performance in terms of delay but also simplified control operation in dynamic channel error condition.

q 2003 Published by Elsevier B.V.

Keywords: Fair queueing; Wireless network; Multimedia services; QoS guarantee

1. Introduction

Third generation (3G) systems such as UMTS and cdma2000, and systems beyond 3G will not only provide high transmission rates over the air inter-face, but also enable the provisioning of IP multimedia services to mobile users [1]. One of the main goals of the 3G and beyond mobile communication systems will be the provision of different kinds of multimedia services with a certain degree of Quality of Service (QoS) guarantee. Packet data services will play a major role in these new multimedia services and packetized transmission over wireless links makes it possible to achieve a high statistical multiplexing gain. Due to the heterogeneous nature of multimedia traffic, the traditional voice-based Medium Access Control (MAC) protocols do not perform well in a multimedia environment. Therefore, a flexible MAC protocol is required to accommodate the multimedia traffic. One important MAC issue is the packet scheduling which determines the order of packet transmissions [2]. Packet scheduling algorithms provide mechanisms for bandwidth allocation and multi-plexing at the packet level.

In wireless networks, many packet scheduling algorithms have been introduced to provide guaranteed QoS as in wireline networks [3–5]. A packet scheduling in wireline networks is a maturing area and fair queueing algorithms are known to be one of the efficient scheduling policies for multimedia traffic [6,7]. However, in wireless networks QoS maintenance through a packet scheduling is not an easy task because of channel error, time-varying link capacity and user mobility, etc. Fair queueing approaches in wireless networks are mainly performed in such a way that a performance as in wireline networks can be achieved in the presence of wireless specific characteristics. There are three representative approaches for wireless fair queueing to resolve the channel error problem: resource swapping, explicit weight changing methods and hybrid of the two. In resource swapping method [8–10], QoS guarantee is accomplished through resource swapping between erroneous flows and error-free flows. In explicit weight
In this paper we propose a delay and data rate decoupled scheduling scheme in fluid version. The packetized version of the proposed scheme for an error-free channel is described in Section 3. We present the packetized version of the proposed scheme for an erroneous channel and evaluate its performance in Section 4. Concluding remarks are given in Section 5.

2. Delay-data rate decoupled scheduling scheme

In this section a proposed delay and data rate decoupled scheduling scheme in fluid version is described. Among the diverse objectives for fairness to incorporate multimedia services, we emphasize different objectives for different traffic types. In conventional fair queueing algorithms, fairness is measured based on a normalized received service in a specific period of time given a set of flows in a system regardless of traffic types [13]. However, the only fairness measure, i.e. the normalized received service, cannot effectively reflect the different QoS requirements of multimedia traffic. Therefore, in this paper we use a different fairness measure for different traffic type: a delay for real time traffic and a normalized received service for non-real time traffic. A fairness guarantee for real time traffic is assumed to be achieved when a delay requirement is satisfied. On the other hand, a fairness guarantee for non-real time traffic is assumed to be achieved when a long-term fairness in terms of normalized received service can be guaranteed. In both cases, a long-term throughput guarantee is a basic condition for fairness.

Our approach for guaranteeing different objectives of different traffic types is to provide independent control for delay and data rate by separating scheduling algorithms for real time and non-real time flows. For real time flows, we propose an explicit-delay-guarantee (EDG) algorithm to guarantee delay deadline, while a GPS-based algorithm is adopted for non-real time flows to guarantee fairness and throughput requirements. In wireless environment, resource swapping between an error-free flow and an erroneous flow is used to combat channel error. In addition, a real time flow’s performance can be improved by dynamically controlling reservation times according to a varying channel condition, which can be performed without degrading the performance of other real time flows. Through simulations, it is shown that the proposed scheme not only explicitly guarantees the delay deadline of real time flows but also improves the performance of non-real time flows in terms of packet delay. In addition, the proposed decoupled feature enables a simplified control in the face of dynamic channel condition.

This paper is organized as follows. In Section 2, we introduce a delay and data rate decoupled scheduling scheme in fluid version. The packetized version of the proposed scheme for an error-free channel is described in Section 3. We present the packetized version of the proposed scheme for an erroneous channel and evaluate its performance in Section 4. Concluding remarks are given in Section 5.
while a GPS-based algorithm is adopted for non-real time flows to guarantee fairness and throughput requirements.

2.1. Improved GPS and issues

Assume that there are two flows; one real time and one non-real time. A real time flow with leaky bucket constrained input traffic is assumed to have delay deadline of 8 and service share \( \phi_{rt} = 3R/4 \), where \( R \) is the link capacity. A non-real time flow \( \phi_{nrt} = R/4 \) without explicit delay deadline requirement. Fig. 1 presents an example of packet service in GPS scheme when a real time flow \( i \) has three packet arrivals while non-real time flow is backlogged, i.e. its buffer is non-empty, throughout time 0–12. In the figure, the arrived packets are served according to the given service share and at time 8 all packets have been completely served. Even though the required transmission deadline of the last packet is 12 it is served earlier than expected, which increases the other flow’s packet delay. This situation originates from the characteristic of GPS where the only parameter, service share, determines the performance of each flow. The performance of GPS can be improved in terms of other flows’ packet delay if the service share of a real time flow is appropriately adjusted as far as its delay deadline can be guaranteed.

In Fig. 1 an example solution is shown as a curve named ‘improved GPS’. Following the GPS algorithm we can get a gradient (service share) for the arriving packets as shown in the figure. In the improved GPS algorithm the gradient of the real time flow is calculated based on the delay deadline of the input packets, which is different from GPS algorithm where the gradient is determined based on both a given service share and the state of other flows. At each packet arrival (or departure), real time flow’s gradient is increased (or decreased) by \((2/8)R\). For a non-real time flow, conventional GPS can be used to guarantee the throughput and delay bound. The delay performance of non-real time flows can be enhanced when a real time flow changes its gradient dynamically as in Fig. 1. In summary, Fig. 1 indicates that for those packet arrivals if a gradient in each time period between events can be provided the delay deadline of those packets can be guaranteed. This in turn improves the delay performance of a non-real time flow.

However, the exact implementation of the improved GPS scheme as in Fig. 1 is not possible or is not efficient due to the following reason. When multiple flows are backlogged at time \( t \) with their maximum burst size, i.e. \( \sigma_i \) for a flow \( i \), the gradient for the first period (from time \( t \) to the first event such as next arrival or departure of a flow) is just the sum of gradient of all flows, which might easily exceed the link capacity \( R \). In order to prevent this overload condition, an admission algorithm should be associated with an arrival of a new flow such that a new flow will not be admitted if it could cause overload condition at any time.

This control, however, will result in an inefficient use of resource because a new flow will be blocked even though there is enough available bandwidth. This constraint can be alleviated by admitting a new flow if its backlog can be served within a deadline period with adaptive gradient pattern. That is, the admission is decided based not on the instant gradient (resource) but on the available resource until the delay deadline. In this scheme a real time flow which fails to obtain an expected gradient can be compensated for gradient later when bandwidth is available.

2.2. Explicit-Delay-Guarantee algorithm

The proposed EDG algorithm is concerned with providing dynamic gradient for real time flows. The following describes how the EDG algorithm provides dynamic

![Fig. 1. Service example of a real time flow in GPS and improved GPS schemes (delay deadline = 8, service share = 3R = 4).](image-url)
Gradient for real time flows. Input traffic of a real time flow \(i\) is regulated by a leaky bucket that consists of bucket size \(s_i\), constraining the burst size of input traffic and a token generation rate equal to \(p_i\), representing the sustained data rate. Let \(r_i\) and \(D_i\) be data rate and packet delay deadline respectively for a flow \(i\). For real time flows \(D_i\) is set to the required delay deadline and for non-real time flows \(D_i\) is set to \(\infty\). Let \(g_i(t)\) and \(g_{inc}^i\) be a gradient of a flow \(i\) at time \(t\) and the required gradient increase of a flow \(i\), respectively. Let \(b_i(t)\) and \(\hat{b}_i(t)\) be the actual backlog size of a flow \(i\) at time \(t\) and a backlog size when an expected gradient is used for the same flow, respectively. In this paper the terminology 'expected' is used to represent the situation when ideal gradients as in the previous example are assigned to each flow. A flow is called as conforming at time \(t\) when \(\hat{b}_i(t) - b_i(t) \geq 0\). Otherwise it is called a non-conforming flow. Events in the proposed scheme represent the following four possibilities: a packet arrival from the same (or other) flow and a packet departure from the same (or other) flow. The sum of existing flow’s gradient is less than the link capacity the remaining available resource is proportionally assigned to each flow as an extra gradient.

Gradient changes are based on a preemptive method where a flow with earlier departure time preempts the resource of a flow with later departure time. For an event from a flow \(j\) with packet size \(l_j\) and for a reference flow \(i\), gradient adjustment is executed as follows and Fig. 2 shows the flow diagram of the proposed algorithm.

- When \(i \neq j\)
  - If the event is arrival, the gradient allocation routine is performed with required gradient increase \(g_{inc}^i = l_j/D_j\).
  - Otherwise, the flow \(i\)'s gradient is decreased by \(l_j/D_j\).
- When \(i = j\)
  - If the event is arrival, an extra gradient is returned to a system.
  - Otherwise, if the flow \(i\) is a non-conforming flow, gradient allocation routine is performed with \(g_{inc}^i = (b_i(t) - \hat{b}_i(t))/ (\hat{t}_n^i - t)\), where \(\hat{t}_n^i\) is the next expected departure time of the flow \(i\).
- Gradient allocation routine
  When a flow \(i\) requests a gradient increase by \(g_{inc}^i\) at time \(t\), if the available resource is not less than \(g_{inc}^i\), \(g_i(t)\) is increased by \(g_{inc}^i\). Otherwise, a preemption procedure is executed as follows, where a flow with earlier departure time preempts the resource of a flow with later departure time.

**Fig. 2.** Flow diagram of the proposed EDG algorithm.
The scheduler continuously looks for a flow with the lowest priority, i.e. with the latest departure time, and deprives the flow of resource until it obtains $g_{i}^{inc}$.

At the end of a searching epoch, if the obtained gradient is still less than $g_{i}^{inc}$, the flow $i$’s gradient is kept until next event occurs.

If there is more than one request the scheduler chooses a flow with an earliest departure time.

After the assignment of gradient for each flow, if resource is still available it is proportionally distributed to all flows as an extra gradient. If $b_{i}(t) = h_{i}(t)$ after a departure event the flow $i$ becomes a conforming flow and follows the expected gradient.

In the proposed algorithm the gradient of each flow might not follow the improved GPS curve in Fig. 1. However, the following principals are preserved to guarantee the delay deadline: (i) the backlog is served throughout the deadline period, (ii) the deadline of each packet is guaranteed and (iii) the control of delay and data rate for real time flows is completely decoupled.

Though in the proposed scheme input traffic of a real time flow is regulated by a leaky bucket other policies such as discussed in Ref. [14] can be adopted as the regulation of input traffic.

2.3. Admission control for delay deadline guarantee

In this section, a call admission control (CAC) algorithm is proposed to guarantee the delay deadline of real time flows in the EDG algorithm. The proposed CAC algorithm is based on the amount of minimum bandwidth, which can guarantee the delay deadline.

Firstly, an example of admission control for two real time flows is presented and then the CAC algorithm is extended to more than three. In the proposed CAC algorithm a new real time flow is accepted only when the acceptance of the flow does not influence the delay deadline of existing real time flows as well as its own worst condition. For a real time flow $i$ with leaky bucket constrained input traffic, which is characterized by a set of parameters $(\sigma_{i}, \rho_{i}, D_{i}, r_{i})$, the following equation should hold

$$\sigma_{i} \leq RD_{i},$$

where $R$ is the link capacity.

Assume that there is a real time flow $i$ with $(\sigma_{i}, \rho_{i}, D_{i}, r_{i})$ in a system at time $t$ and a new real time flow $j$ with $(\sigma_{j}, \rho_{j}, D_{j}, r_{j})$ and $D_{j} > D_{i}$ is waiting for an admission. For a convenience we set $t$ to be 0. For an admission of the flow $j$, the following minimum bandwidth condition should be satisfied. The total bandwidth of flows including the flow $j$ should not exceed the link capacity, i.e. $r_{i} + r_{j} \leq R$. In addition, some resource should be available to serve $\sigma_{j}$ (maximum backlog) of the flow $j$ during $[0, D_{j}]$, which can be calculated by the following equation:

$$\sigma_{j} \leq RD_{j} - \rho_{i}(D_{j} - D_{i}) - \sigma_{i},$$

where $\rho_{i}(D_{j} - D_{i})$ represents the amount of the flow $i$ backlog which should be served during $[0, D_{j} - D_{i}]$ to guarantee the delay deadline of the flow $i$. From Eqs. (1) and (2), we can verify that $\sigma_{i}$ and $\sigma_{j}$ can be served during $[0, D_{j}]$ and $[0, D_{j} - D_{i}+]$, respectively. Besides, the backlog of flow $i$ which could be generated during $[0, D_{j} - D_{i}]$ can be served during $[0, D_{j}]$.

For a generalization of an admission condition for bandwidth, we consider the following two cases. Let $\mathcal{B}$ be a set of existing flows in a system. The basic bandwidth condition for an admission is $\sum_{i \in \mathcal{B}} r_{i} + r_{j} \leq R$. We can divide $\mathcal{B}$ into two groups according to $D_{i}$, $i \in \mathcal{B}$, $\mathcal{B}^{1} = \{ \forall i \in \mathcal{B} | D_{i} \leq D_{j} \}$ and $\mathcal{B}^{2} = \{ \forall i \in \mathcal{B} | D_{i} > D_{j} \}$ for a new flow $j$.

Case I (admission condition for $\mathcal{B}^{1}$): For a new flow $j$ to be admitted, all $\mathcal{B}^{2}$ flows should be served within their deadline as well as its own $\sigma_{j}$. Then a condition for an admission is as follows:

$$\sigma_{j} \leq RD_{j} - \sum_{k \in \mathcal{B}^{2}} \{ \sigma_{k} + \rho_{k}(D_{j} - D_{k}) \}.$$  (3)

For the set of $\mathcal{B}^{2}$ only one calculation is required because, since the new flow does not affect the performance of existing flows in $\mathcal{B}^{2}$, only the calculation of available resource for its own backlog is required.

Case II (admission condition for $\mathcal{B}^{2}$): For $\mathcal{B}^{2}$, the admission decision becomes more complex because the new flow affects the delay deadline of existing flows. In order to guarantee the delay deadline of all existing flows, the calculation for admission should be performed for each flow in $\mathcal{B}^{2}$. The condition for admission can be derived from the modification of Eq. (3) as follows. Let $L(k)$ be a flow index with the $k$th smallest value of deadline in $\mathcal{B}^{2}$, where $k = 1, 2, ..., N(\mathcal{B}^{2})$ and $N(x)$ is the number of element of $x$. For all $k$’s, the following equation should hold to admit a new flow $j$

$$\sigma_{L(k)} \leq RD_{L(k)} - \sum_{i \in \mathcal{B}^{2}} \{ \sigma_{i} + \rho_{i}(D_{L(k)} - D_{i}) \} - \sum_{j = 1}^{k-1} \{ \sigma_{L(j)} + \rho_{L(j)}(D_{L(k)} - D_{L(j)}) \}.  \quad (4)$$

On the other hand, an admission control for non-real time flows is simply accomplished by examining an available bandwidth, i.e. a new non-real time flow $j$ is admitted when $\sum_{i \in \mathcal{B}} r_{i} + r_{j} \leq R$.

Lemma 1. When a real time flow is accepted in a system based on the proposed admission condition for bandwidth and it is scheduled according to the proposed EDG algorithm, its delay deadline can be guaranteed.

Proof. The proof is given by showing that a series of gradient can be provided to a flow $i$ to guarantee the delay
deadline. Assume that a packet from a flow \(i\) with size \(l_i\) arrives at time \(t_x\) and its deadline is \(t_y(=t_x+D_i)\). Let \(t_x\) be the time of arrival or departure event in \([t_a,t_d]\) in ascending order, where \(x = 1, 2, \ldots, d\) and \(t_0 = t_a\). Let \(\mathcal{N}\) be a set of flows whose next departure is included in \([t_a,t_d]\).

Since when the flow \(i\) is a conforming flow the proof is straightforward we consider only a non-conforming flow \(i\). In this proof we take into account flows only in \(\mathcal{N}\) because other flows do not influence the delay deadline of the flow \(i\) as the following fact. Assume that the flow \(j\) whose next departure time is greater than \(t_y\) and has a gradient \(g_j > 0\) at a departure time \(t_x\). \(t_x < t_y\). Assume that a flow \(i\) is a non-conforming at \(t_x\) and tries to increase its gradient. Following the EDG algorithm, for a gradient compensation the scheduler selects a flow \(k\) with the latest departure time and \(g_k > 0\). Thus the flow \(j\) would be selected eventually and its gradient would be decreased to 0 because the flow \(i\) is still non-conforming. Thus the flow \(j\) keeps its gradient to 0 after time \(t_x\) as far as the flow \(i\) is non-conforming, which indicates that the existence of the flow \(j\) does not affect the assignment of flows \(i\)'s gradient.

Let \(R_i\) be \(\sum_{x \in \mathcal{N}} g^i_x\), where \(g^i_x\) represents a flow \(j\)'s gradient at time \([t_{x-1}, t_x]\). Since during \([t_{x-1}, t_x]\), only flows in \(\mathcal{N}\) and the flow \(i\) can participate in a scheduling as far as the flow \(i\) is non-conforming, \(R_i\) further can be represented by \(R_i = R - \sum_{x \in \mathcal{N}} g^i_x\). Let \(S_i\) be the total amount of service received by flows in \(\mathcal{N}\) during \([t_{x-1}, t_x]\). Then we can get the following received service equation:

\[
S_i = R_i(t_x - t_{x-1}) = (R - \sum_{x \in \mathcal{N}} g^i_x)(t_x - t_{x-1}). 
\]  

The maximum total service which can be received by flows in \(\mathcal{N}\) while the flow \(i\) is non-conforming is \(\sum_{x \in \mathcal{N}} R\times \{\sigma_j + \rho_j(t_x - (t_x + D_j))\}\). Further by summing Eq. (5) for all intervals in \([t_a,t_d]\) and using the above result we can get

\[
\sum_x S_i = \sum_x (R - \sum_{x \in \mathcal{N}} g^i_x)(t_x - t_{x-1})
\]

\[
= R(t_y - t_x) - \sum_x \sum_{x \in \mathcal{N}} g^i_x(t_x - t_{x-1})
\]

\[
\sum_x S_i \leq \sum_{x \in \mathcal{N}} \{\sigma_j + \rho_j(D_j - D_i)\}. 
\]  

From Eqs. (6) and (7) and an admission condition in Eq. (2) we can get

\[
\sum_x g^i_x(t_x - t_{x-1}) \geq R(t_y - t_x) - \sum_{x \in \mathcal{N}} \{\sigma_j + \rho_j(D_j - D_i)\} \geq \sigma_i. 
\]  

Eq. (8) indicates that there exists a series of \(g^i_x\) throughout the period \([t_a,t_d]\) and with those gradients the amount of service which can be received during \([t_a,t_d]\) is at least \(\sigma_i\). Since \(\sum_x g^i_x\) is the maximum gradient possible during \([t_a,t_d]\), we can consider an actual gradient \(\tilde{g}^i_x\) such that \(\sum_x g^i_x \leq \sum_x \tilde{g}^i_x\) for the same period. Finally, we can get an actual series of gradient \(\tilde{g}^i_x\), \(x = 1, 2, \ldots, d\) for a flow \(i\).

### 2.4. Performance analysis

In this section we analyze the performance of the proposed scheme for real time and non-real time flows.

We can easily get a delay, \(d_i\), of a real time flow \(i\) as follows:

\[
d_i \leq D_i. 
\]

From the above results we can verify that the backlog of real time flows is always cleared, which indicates that if \(\sigma\) and \(\rho\) are appropriately assigned according to an actual data rate that amount of bandwidth is always provided in the proposed scheme.

Let \(B^r\) and \(B^n\) be respectively the set of real time and non-real time flows. The throughput, \(c_i\), that a non-real time flow \(i\) can achieve in the proposed scheme is

\[
c_i \geq \frac{R}{\sum \rho_j} - \frac{\sum \rho_j}{\sum \rho_j} \frac{\sum \sigma_j}{R}. 
\]

for a long time period because a GPS scheduler is used for non-real time flows. Let us consider a non-real time flow \(i\) with data rate \(r_i\). When the flow has an arrival of packet with maximum size \(r^\text{max}\) at time \(t_0\), we can get the time the packet can be completely served in the proposed scheme. Let \(t_d\) be an expected service finishing time for the packet. We can get \(t_d\) based on the amount of service that can be received by the flow in the system. We can get the following equation for \(t_d\) in an EDG scheme in the worst condition when \(t_0\) is assumed to be 0:

\[
\left\{ \frac{R_i^{\text{EDG}} - \sum_{j \in B^r} \{\sigma_j + \rho_j(D_j - D_i)\}}{\sum \rho_j} \right\} \cdot \frac{r_i}{\sum \rho_j} = r^\text{max}. 
\]

By arranging the above equation for \(t_d\) we can get

\[
t_d = \frac{\sum \rho_j}{R - \sum \rho_j} \frac{r^\text{max}}{\sum \rho_j} + \frac{\sum \rho_j}{R - \sum \rho_j} \frac{\sum \sigma_j}{R}. 
\]

where \(t_d\) for a GPS scheme can be calculated by eliminating \(D_i\) and re-placing \(t_d^{\text{EDG}}\) by \(t_d^{\text{GPS}}\) in Eq. (9). The second term of the right side of Eq. (10) is always larger than zero because \(R > \sum \rho_j\). Therefore, in the proposed EDG scheme non-real time flows can always be served earlier at least by \(\sum \rho_j D_i/(R - \sum \rho_j)\) than in GPS.

### 3. Packetized scheme of the proposed algorithm in error-free channel

In this section, we propose a packetized implementation of the EDG scheme discussed in Section 2 in error-free
channel, where real time flows are scheduled based on the proposed packetized EDG algorithm and non-real time flows are scheduled based on WFQ algorithm [12]. Since real time flows need to be served within deadline, all packets from real time flows are explicitly assigned transmission slots (time). Non-real time flows are scheduled by WFQ using the remaining slots. In the proposed scheme other conventional fair queuing algorithms can be used for the scheduling of non-real time flows.

We assume that a transmission link is composed of time slots with fixed duration. Let \( L \) and \( l_i \) be a slot size and the packet size of a flow \( i \) in bits, respectively. Let us assume \( l_i = nL \) for a flow \( i \); where \( n \) is a positive integer. Without loss of generality we assume that a time instant \( t \) indicates a slot which lasts during \( [t, t + d) \), where \( d \) is a slot duration.

A new real time flow joins a system based on the admission control described in Section 2. Since unlike a fluid system a packet cannot be served throughout the deadline period, it is served through one or more slots placed within a deadline. That is, as soon as packets (single or burst of packets) from real time flows arrive, one or more time slots which are placed within a deadline from current time slot are assigned. For this purpose we propose an allocation window to designate the owner of future slots. Let \( W(t) \) be an allocation window at time \( t \):

\[
W(t) = \begin{cases} \text{marked} & \text{if} \; W(t) \neq \emptyset \\ \text{un-marked} & \text{if} \; W(t) = \emptyset \end{cases}
\]

In this paper, we use an incremental probability scheme to assign a slot in an allocation window for real time flows, where the packet allocation is decided by a probability whose value is increased when the previous allocation fails. The allocation probability of a real time flow \( i \) at \( t \) is calculated by

\[
p_i(t) = \frac{L_i(t)}{E(t)}, \tag{11}
\]

where \( L_i(t) \) and \( E(t) \) are the lag size of the flow \( i \) in terms of the number of packets in its buffer and the number of un-marked slots in the allocation window at time \( t \), respectively. \( E(t) \) is determined by counting the number of un-marked slots from time \( t \) to the deadline of the flow \( i \). Eq. (11) shows that when the allocation at time \( t \) fails the probability for the next slot is increased to

\[
p_i(t + 1) = \frac{L_i(t + 1)}{E(t + 1)} = \frac{L_i(t)}{E(t) - 1},
\]

which verifies that the lag can be cleared at least in the last slot. This procedure conforms to the fluid EDG algorithm discussed in Section 2 even if it cannot exactly implement the fluid EDG algorithm.

Fig. 3 describes an example of assignment of slot in an allocation window. The size of an allocation window, \( n + 1 \), is determined by the largest delay deadline of a real time flow. When a reference flow \( i \) with \( D_i = k + 1 \) has an arrival of two packets at time \( t \), two slots located from index \( t \) to \( t + k \) in the allocation window are reserved for these two packets according to the probability given in Eq. (11). Fig. 3(b) shows that the allocation window of index \( t + 1 \) and \( t + k - 3 \) are assigned for these packets.

A scheduling at time \( t \) in the packetized scheme is performed as follows.

- If \( W(t) \) is marked a real time flow with index \( W(t) \) is served.
- Otherwise,
  - If there is at least one backlogged non-real time flow, a packet from the flow with the least virtual finishing time is served.
  - Otherwise, the first marked slot in the allocation window is selected and the associated real time flow is served.

![Real time packet allocation window](image-url)

Fig. 3. Example of packet assignment for a real time flow \( i \) in an allocation window (a) before allocation (b) after allocation.
served while the selected slot in the allocation window is un-marked.

3.1 Simulation results

In this section we show the performance of the packetized EDG scheme in error-free channel.

In the simulation, a voice flow is modeled as ON–OFF signal with ON and OFF durations having exponential distributions. A video flow is modeled by modified MPEG source, where there are three types of frame, i.e. I, B and P frames. Each frame size is determined by a Lognormal distribution with a specified mean and standard deviation. A video source generates 24 frames per second. Data flow is modeled by Poisson arrival with truncated Pareto distributed burst size. Table 1 shows a traffic model and parameters used in the simulation, where voice and video traffic are assumed to be real time flow while data traffic is assumed to be non-real time flow.

Fig. 4 shows the packet delay of the proposed EDG scheme, a priority scheme, and WFQ scheme [12]. In the simulation result, all three scheduling policies guarantee the required delay bound of real time flows. Though the EDG scheme has the largest voice and video packet delay, it has the smallest guaranteed packet delay of data flows. Since the delay of real time flows is all conforming to the required delay bound, smaller packet delay for real time flow has no significance from a performance point of view. In the figure the result of data delay in WFQ is drawn just for a reference because the result does not give a guaranteed performance. In WFQ, the guaranteed throughput is as little as 5% in the given offered load condition of data flows because a large service share is allocated for real time flows to guarantee their delay bounds. Therefore the guaranteed throughput for data flow is far less than its actual load and the figure is drawn based on the combination of best effort and guaranteed policy.

From the results of the simulation we can understand that by delaying the transmission of real time flows as far as their delay deadline can be guaranteed, the performance of non-real time flows can be improved.

4. Packetized scheme of the proposed algorithm in erroneous channel

In this section, the proposed algorithm is extended to erroneous channel condition. When there is a channel error the algorithm described in Section 3 should be changed to combat the channel error. The allocation window also has to

<table>
<thead>
<tr>
<th>Model</th>
<th>Voice (ON–OFF)</th>
<th>Video (modified MPEG)</th>
<th>Data (Poisson and truncated pareto)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>20</td>
<td>1000/24</td>
<td>∞</td>
</tr>
<tr>
<td>$\sigma_1$</td>
<td>1</td>
<td>30</td>
<td>–</td>
</tr>
<tr>
<td>$r_t$</td>
<td>0.016</td>
<td>0.057</td>
<td>0.12–0.15</td>
</tr>
</tbody>
</table>

Fig. 4. Average packet delay of the proposed EDG, priority, and WFQ schemes without channel error.
be changed to maintain a residual time of each packet in its slot.

4.1. Compensation for channel error in real time flows

When a real time flow \(i\), marked in an allocation window at time \(t\) and selected for transmission, is experiencing channel error, it defers its transmission by resource swapping. The resource swapping in the proposed scheme is performed as follows:

- Select a flow \(j \in S(t)\) which is marked at time \(t' > t\) in an allocation window and has the smallest time instant satisfying the condition \(t' - t \leq \text{Res}_{i}(t)\), where \(S(t)\) and \(\text{Res}_{i}(t)\) are a set of marked real time flows in error-free channel and the residual time of the head-of-line packet of a flow \(i\) at time \(t\), respectively.
- If the flow \(j\) exists then transmit a packet of the flow \(j\) at current slot and mark \(W(t')\) as \(i\) with new residual time.
- Otherwise select an un-marked slot at \(t'\) from an allocation window, which is closest from time \(t\).
  - If \(W(t')\) exists then mark \(W(t')\) as \(i\) with new residual time while un-marking \(W(t)\). Resume the scheduling including the newly un-marked slot \(W(t)\).
  - Otherwise, a packet of the flow \(i\) is dropped because of the violation of delay deadline.

As far as there exists a slot in an allocation window, which can be swapped within the delay deadline, the performance of the real time flow is not affected by the channel error. When packet dropping occurs, the performance of the real time flows cannot be guaranteed in terms of zero packet dropping probability. However, since real time flows can tolerate a packet loss to some extent, the performance of real time flows in terms of the required packet dropping probability can be guaranteed by an appropriate scheduling.

Making use of the characteristic of real time flows, we introduce an *accelerating factor* \(A_i \geq 1\) for a flow \(i\), which is used in calculating the allocation probability as follows:

\[
p_i(t) = \frac{L_i(t)}{E_i(t)} A_i.
\]

The packet dropping probability of a flow \(i\) is decreased by increasing the accelerating factor \(A_i\) because the probability of being placed at the front part of the allocation window is increased. This is possible because when a flow experiencing channel error is selected for transmission far earlier than its deadline, it has more chance to swap transmission with others. However, this operation could violate the admission control discussed in Section 2 such that other real time flows might be fallen short of the available slots in the allocation window even though they are accepted based on the admission control. The violation leads to the increase of the packet dropping probability of other real time flows. Therefore, we need to find an appropriate value for the accelerating factor to get a tradeoff point between the packet dropping probabilities of reference flow and other flows in order to minimize the total packet dropping probability.

If other regulation policy which is less strict than a leaky bucket is used for the constraint of input traffic, the resource management could be more flexible in channel error condition.

4.2. Compensation for channel error in non-real time flows

We have proposed a channel error and handoff compensation scheme in wire-less networks in Ref. [15]. In this paper, we adopt part of Ref. [15] for a compensation of non-real time flows. To properly describe the operation of the proposed scheme we present some definitions. An ideal flow is defined as a flow in error-free channel using the same scheduling algorithm as the reference flow. If there is no channel error all flows will be ideal flows. However in wireless network because of channel error each flow may receive more or less service at various times. As the result of this, flows are classified into three types based on the amount of received service: poor, rich and normal flows. Further, to keep track of the service received by an active flow, a flow \(i\) is associated with a parameter \(\Delta_i\) that represents the difference between actual and expected service.

When a flow experiences a burst channel error it cannot transmit packets for a while. Then, the flow receives less service than an ideal (expected) one and is defined as a *poor flow* \((\Delta_i < 0)\). When a flow experiences channel error its service opportunity is transferred to one of flows in error-free channel. A flow receiving the transferred resource becomes a *rich flow* \((\Delta_i > 0)\) if its received service is larger than that of ideal one. If a flow has received exactly the same amount of service as the ideal one the flow is defined as a *normal flow* \((\Delta_i = 0)\). Since \(\Delta_i\) of a rich flow \(i\) is obtained from poor flows, the rich flow has to release its resource later. In the proposed scheme we use a probability for releasing the resource of rich flows.

Each flow is assigned a priority according to its state, i.e. a flow type and a received normalized service. In the proposed scheme, a poor flow has a priority over normal and rich flows. The priority among flows with the same priority is determined based on \(b_i(=\Delta_i r_i)\).

When a flow \(i\) scheduled to be served in the next slot is experiencing channel error, the scheduler changes the service order by delaying the flow \(i\)'s transmission. The scheduler searches an alternative flow \(j\) using a scheduling algorithm and if the flow \(j\) exists \(\Delta_i\) is decreased by 1 while increasing \(\Delta_j\) by 1. At this moment the virtual time of the flow \(i\) advances according to the associated scheduling algorithm. However, the flow \(j\)'s virtual time does not advance even though it receives additional service. These operations not only prevent the flow \(i\) from being selected
repeatedly but give the flow $j$ more chance of releasing resource by giving more chance of being selected. With the proposed mechanism, as long as there exists an active flow that can send, the swapped resource by a flow is always reflected as a richness of other active flows. Therefore, if all flows are in error-free channel for a long enough period of time, $\Delta_i$’s of all flows will approach to zero, which could achieve the performance of the associated fair queueing algorithm. If there is no active flow that can transmit in place of the flow $i$, this slot is used by a marked real time flow that is in error-free channel.

Two different situations have to be considered when a flow wants to leave a system. When the flow is a rich one the remaining poor flows should be compensated for the swapped resource. Whereas, when the departure flow is a poor one the poorness should be reflected in the remaining flows to satisfy a fairness condition. In order to address these situations, we propose a compensation server having $\Delta_{CS}$ as a global discrepancy between actual service and expected service. When a departure event occurs from a flow $i$, $\Delta_{CS}$ is increased by $\Delta_j$ such that the sum of $\Delta$’s including $\Delta_{CS}$ in a system remains same, i.e. zero. When a new burst of packets arrives at a flow $j$ with empty buffer, $\Delta_j$ is set to a value of $\Delta_{CS}/r_j$ and $\Delta_{CS}$ is decreased by $\Delta_{CS}/r_j$.

### 4.3. Simulation results

Through a computer simulation, the impact of the accelerating factor for real time flows and the fairness of non-real time flows are evaluated and the performance of the proposed scheme is compared with other schemes.

The same traffic model as in Section 3.1 is assumed in this section. The channel error evolves according to a two-state discrete Markov Chain. Let $p_e$ be the probability that the next time slot is good given that the current slot is in error, and $p_e$ be the probability that the next time slot is in error given that the current slot is good. Then the steady state probabilities $P_G$ and $P_E$ of being in the good and bad states, respectively, are given by $P_G = p_g/(p_g + p_e)$, $P_E = p_g/(p_g + p_e)$, where $p_g + p_e$ is set to 0.1 in the simulation. The releasing probability of non-real time rich flows is assumed to 0.5. The compensation in a priority scheme is assumed to follow the same procedure discussed in Section 4.2.

Fig. 5 depicts the effects of different accelerating factor on the packet dropping probability and delay when $P_E = 0.3$. When an accelerating factor is increased the packet dropping probabilities of voice and video flows are decreased. Whereas, the delay of data flows increases because real time flows are reserving resource earlier than expected. From this result an appropriate accelerating factor can be obtained to guarantee the required packet dropping probability.

Fig. 6 depicts the influence of the change of an accelerating factor. When $A_{video} = 3.1$ and $P_E = 0.3$ if $A_{voice}$ increases from 1.0 to 3.0 the packet dropping probability of voice flows decreases while that of video flows is almost constant. Fig. 6 indicates that even though an accelerating factor of a flow changes the other flows’ (real time flows’) performance is not much influenced as far as those flows maintain enough accelerating factors for guaranteeing the required QoS. This characteristic of the proposed scheme enables a simple control operation for real time flows in varying channel condition. From Figs. 5 and 6 we can conclude that an optimum accelerating factor can be obtained for real time flow in a certain channel condition.

Fig. 7 shows the packet delay of data flows in three schemes when voice and video flow’s packet dropping probabilities are kept to be 0.01 and 0.001, respectively. In the figure as the channel condition becomes worse the packet delays of the conventional CIF-Q [8] scheme and the priority scheme increases abruptly. However in the EDG
scheme the packet delay increases relatively smoothly compared with others. Like Fig. 4 the delay performance of CIF-Q scheme is not guaranteed but it is drawn for a comparison. The delay performance of EDG scheme is much better than CIF-Q scheme especially in high channel error condition because the EDG scheme can adjust the control parameter dynamically according to a channel condition.

Fig. 8 presents the fairness of data flows when one of data flows is experiencing severe channel error. A data flow 1 is assumed to experience channel error with $P_E = 0.9$ for 3 s from 50 s and $P_E = 0.3$ during the rest of the time. As depicted in Fig. 8 data flow 1 can hardly receive service when $P_E = 0.9$ while other data flows can receive more than expected. The value of compensation server is increased with fluctuations during the 3 s.

The increase of $\Delta_{CS}$ indicates that the buffers of data flow 2, 3, and 4 get emptied with positive $\Delta$'s. The decrease indicates the event of arrival of new packet burst to data flows with empty buffer resulting in the assignment of positive $\Delta$'s, which decreases the $\Delta_{CS}$.

The abrupt increase of $\Delta_1$ at time 54 s indicates that the buffer of the data flow 1 becomes empty so that the ‘poorness’ is transferred to the compensation server, which sees an abrupt decrease of $\Delta_{CS}$.
With some fluctuations for a few seconds Δ's of data flows converge to zero such that the fairness among data flows can be guaranteed in the long term in the presence of the severe channel error in a certain flow.

From the simulation results in erroneous channel we can notice that the proposed EDG scheme provides independent control for delay and data rate for different types of traffic. Real time flows can easily react to a varying channel error condition by controlling transmission order through accelerating factor without much affecting other real time flows. Among non-real time flows the fairness in terms of normalized received service can be maintained through resource swapping associated with a fair queuing algorithm in the presence of channel error.

5. Conclusion

In this paper, we have proposed a delay and data rate decoupled fair queuing scheme for wireless multimedia, where two separate scheduling algorithms are applied respectively for real time and non-real time flows. For real time flows, we have proposed an EDG algorithm to guarantee the delay deadline, while a conventional fair queuing algorithm is adopted for non-real time flows to guarantee fairness and throughput requirements. The proposed scheme not only explicitly guarantees the delay deadline of real time flows but also improves the performance of non-real time flows in terms of packet delay. In wireless environment, resource swapping between an error-free flow and an erroneous flow is used to combat channel error. In addition, a real time flow’s performance can be improved by controlling transmission order using accelerating factor according to a varying channel condition, which can be performed without degrading the performance of other real time flows. Through simulations, it is shown that the proposed scheme provides not only improved performance in terms of delay but also simplified control operation in dynamic channel error condition.

Our future work includes an extension of the proposed EDG scheme to CDMA system, where channel condition and resource management are complicated due to the dependence of capacity on interference (lemma and proof delete this paginator, this is dummy citation).

Acknowledgements

This work is supported in part by ERC-UFON, and in part by University Support Program of MIC.

References


