

A fair scheduling algorithm with traffic classification for wireless networks

You-Chiun Wang, Shiang-Rung Ye, Yu-Chee Tseng*

Department of Computer Science and Information Engineering, National Chiao Tung University, Hsin-Chu, 30050 Taiwan, ROC

Received 21 July 2004; accepted 21 July 2004

Available online 11 September 2004

Abstract

Wireless channels are characterized by more serious bursty and location-dependent errors. Many packet scheduling algorithms have been proposed for wireless networks to guarantee fairness and delay bounds. However, most existing schemes do not consider the difference of traffic natures among packet flows. This will cause the *delay-weight coupling* problem. In particular, serious queuing delays may be incurred for real-time flows. To resolve this problem, we propose a traffic-dependent wireless fair queuing (TD-FQ) algorithm that takes traffic types of flows into consideration when scheduling packets. The proposed TD-FQ algorithm not only alleviates queuing delay of real-time flows, but also guarantees bounded delays and fairness for all flows.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Data communication; Fair queuing; Mobile communication system; Scheduling; Wireless network

1. Introduction

To meet QoS requirements, many packet scheduling algorithms [1–6] have been proposed for wireline networks to guarantee fairness and delay bounds. However, it is not a trivial task to directly apply these algorithms to wireless domain. In particular, wireless channels are characterized by more serious bursty and location-dependent errors [7,8]. Bursty errors may break a flow's continuous services, while location-dependent errors are likely to allow error-free flows to receive more services than they deserve, thus violating the fairness and delay bound properties.

To solve these problems, several wireless packet scheduling algorithms have been proposed [9–14]. In idealized wireless fair queuing (IWFQ) [9], each packet is associated with a *finish tag*, which is computed according to the principles of weight fair queuing (WFQ) [2]. The scheduler always selects the error-free packet with

the smallest finish tag to serve. When a flow suffers from channel errors, all its packets will keep their old tags. Therefore, when the flow exits from errors, its packets are likely to have smaller finish tags, thus achieving the compensation purpose. In channel-condition independent fair queuing (CIF-Q) [10], fairness is achieved by transferring the time allocated to those error flows to those error-free flows. Later on, compensation services (CS) will be dispatched to the former proportional to their weights. However, as Ref. [13] shows, an inherent limitation of fluid fair queuing is that the delay observed by a flow is tightly coupled with the fraction of bandwidth given to that flow among all backlogged flows. Since the fraction is in turn coupled with the weight assigned to the flow, we call this the *delay-weight coupling problem*. Both IWFQ and CIF-Q may suffer from this problem.

In this work, we consider the fair scheduling problem in a wireless network whose input includes both real-time and non-real-time traffics. This problem is especially important with the recently emerging multi-media services (MMS) in next-generation wireless networks. Real-time applications are typically delay-sensitive. If wireless fair scheduling is

* Corresponding author. Tel.: +886 3 513 1366; fax: +886 3 572 4176.
E-mail address: yctsen@csie.nctu.edu.tw (Y.-C. Tseng).

supported without special consideration for real-time flows, the delay-weight dilemma would either hurt real-time flows or the system performance. Several wireless scheduling algorithms have been proposed to address this concern [11–14]. However, they still suffer from certain weaknesses (Ref. to Section 2).

In this work, we propose a new algorithm called Traffic-Dependent wireless Fair Queuing (TD-FQ). Traffic arriving at a base station (BS) are mixed with real-time and non-real-time flows. TD-FQ is developed based on CIF-Q [10], but it adds extra mechanisms to reduce queuing delays of real-time flows by giving them higher priorities. Nevertheless, TD-FQ guarantees that the special treatment of real-time flows will not starve non-real-time flows. Thus, it still maintains fairness and bounded delays for all flows.

The rest of this paper is organized as follows. Related work is discussed in Section 2. Section 3 presents our TD-FQ algorithm. Section 4 formally proves several properties of TD-FQ. Simulation results are presented in Section 5. Conclusions are drawn in Section 6.

2. Related work

In server based fairness approach (SBFA) [11], a fraction of bandwidth is reserved particularly for compensation purpose. A number of virtual servers called long-term fairness servers (LTFS) are created for those flows that experienced errors. Then the reserved bandwidth will be used to compensate those LTFS flows. However, since the erroneous flows are compensated in a first-come-first-served manner, real-time lagging flows may still suffer from long queuing delay.

Effort-limited fair (ELF) [12] suggests to adjust each flow's weight in response to the error rate of that flow, up to a maximum defined by that flow's *power factor*. However, since the scheduler does not have immediate knowledge about the error rates of a flow, there could be some delay in adjusting its weight to respond to its channel and queue condition. Besides, when a real-time flow just exits from errors, it is emergent to deliver packets for the flow, or these packets may be dropped. Unfortunately, adjusting weights cannot guarantee higher priorities for such flows.

Wireless fair service (WFS) [13] assigns each flow i with a *rate weight* r_i and a *delay weight* Φ_i , and associates every packet p_i^k with a start tag $S(p_i^k)$ and a finish tag $F(p_i^k)$,

$$S(p_i^k) = \max\{V(A(p_i^k)), S(p_i^{k-1}) + L_i^{k-1}/r_i\},$$

$$F(p_i^k) = S(p_i^k) + L_i^k/\Phi_i,$$

where L_i^k is the length of the k th packet of flow i , $A(p_i^k)$ is the arrival time of the packet, and $V(t)$ is the virtual time at time t . Essentially, flow i is drained into the scheduler according to the rate weight r_i , but served according to the delay weight Φ_i . The flow with the smallest finish tag will be picked by the scheduler. By introducing the delay weight, WFS decouples delay and bandwidth to a certain degree. However, since the computation of start tags is still based on rate weights, real-time flows may not get much benefit. Besides, WFS adopts a compensation mechanism based on a weighted round robin approach, where the lagging degree of a flow is used as its weight. Without distinguishing real-time and non-real-time flows, this algorithm may still cause serious queuing delays for real-time flows.

Lee et al. [14] classify flows into four groups: *poor*, *poorer*, *rich*, and *normal*. A flow is said *poor* if it receives less service than it expects. When a poor flow transmitting real-time traffic is about to drop packets due to long waiting time, this flow is changed to a *poorer* flow. When there are CSs available, the poorer flows always have the highest priority to receive such services. However, this behavior may cause other poor flows to starve if there are many poorer flows.

3. The TD-FQ algorithm

Below, we first introduce the system model and basic operations of TD-FQ, followed by some special designs of TD-FQ, including *graceful degradation*, *compensation*, and *lag redistribution*.

3.1. System model

We consider a packet-cellular network as in Fig. 1. Packets arriving at a BS are classified into real-time traffic

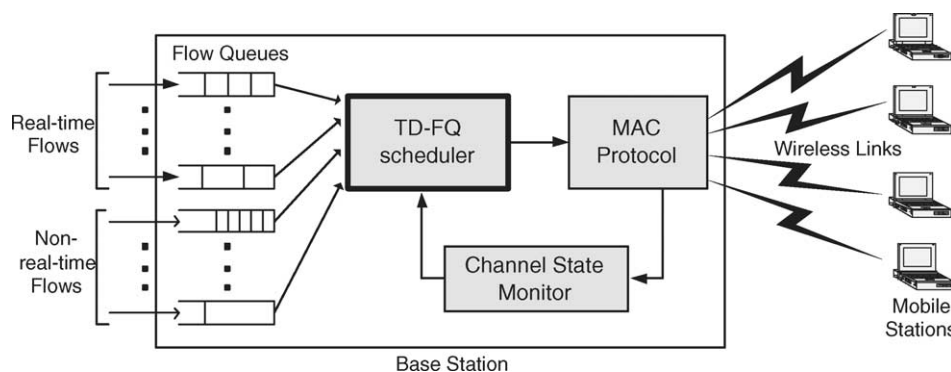


Fig. 1. System architecture of TD-FQ.

and non-real-time traffic and dispatched into different queues depending on their destination mobile stations. These traffic flows are sent to the *TD-FQ scheduler*, which is responsible for scheduling flows and transmitting their head-of-line (HOL) packets via the *MAC protocol*. The *Channel state monitor* provides information about the channel state of each mobile station (there are different alternatives to achieve this, but this is out of the scope of this work). For simplicity, we assume that BS has immediate and accurate knowledge of each channel's state.

In this paper, we focus on the design of TD-FQ scheduler. Mobile stations may suffer from bursty and location-dependent channel errors. However, error periods are assumed to be sporadic and short relative to the whole lifetime of flows so that long-term unfairness would not happen.

3.2. Basic operations

Following most fair queuing works, each flow i is assigned a weight r_i to represent the ideal fraction of bandwidth that the system commits to it. However, the real services received by flow i may not match exactly its assigned weight. So we maintain a virtual time v_i to record the nominal services received by it, and a *lagging level* lag_i

to record its credits/debits. The former is to compete with other flows for services, while the latter is to arrange CSs. The actual normalized service received by flow i is $v_i - lag_i/r_i$. Flow i is called *leading* if $lag_i < 0$, called *lagging* if $lag_i > 0$, and called *satisfied* if $lag_i = 0$. Further, depending on its queue content, a flow is called *backlogged* if its queue is nonempty, called *unbacklogged* if its queue is empty, and called *active* if it is backlogged or unbacklogged but leading. Note that TD-FQ will only choose active flows to serve. When an unbacklogged but leading flow (i.e. an active flow) is chosen, its service will actually be transferred to another flow for compensation purpose. Also, following the principle of CIF-Q, whenever a flow i transits from unbacklogged to backlogged, its virtual time v_i is set to $\max\{v_i, \min_{j \in A}\{v_j\}\}$, where A is the set of all active flows.

Fig. 2 outlines the scheduling policy of TD-FQ. TD-FQ follows the design principle of CIF-Q. First, the active flow i with the smallest virtual time v_i is selected. If flow i is backlogged and its channel condition is good, the HOL packet of flow i can be served if flow i is non-leading, in which case the service is called a normal service (NS). Then we update the virtual time v_i as $(v_i + l_p/r_i)$, where l_p is the length of the packet. In case that flow i has to give up its service due to an empty queue or a bad channel condition, the service will become an extra service (ES). On the other

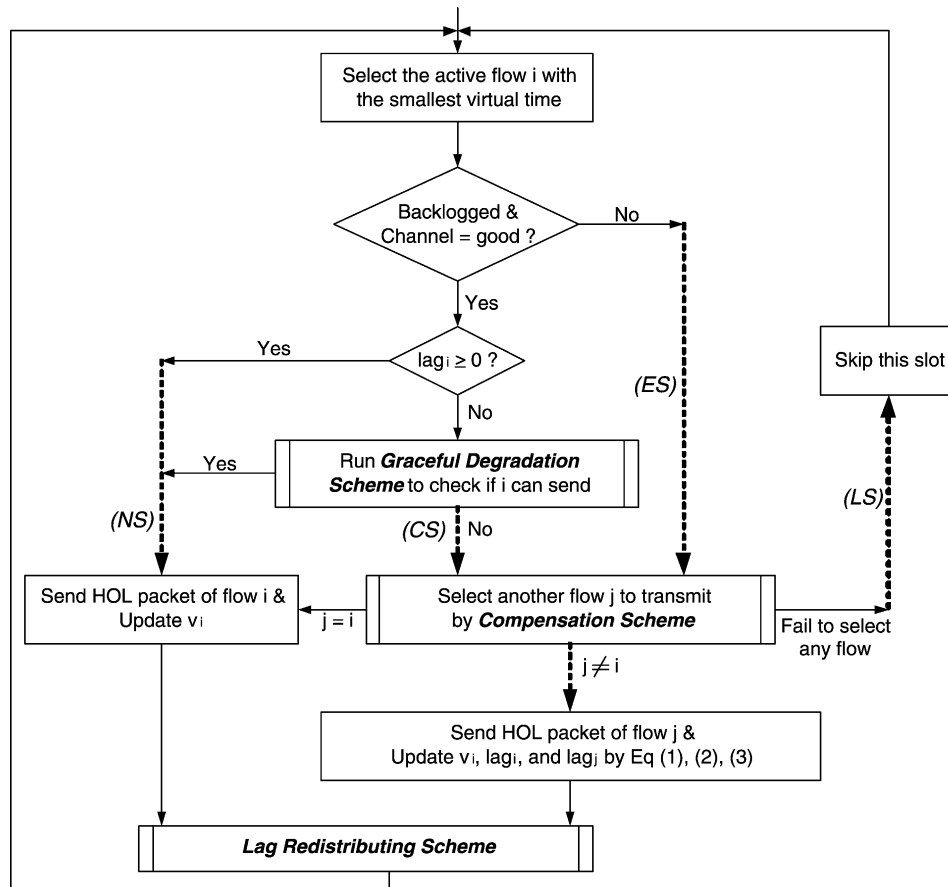


Fig. 2. Scheduling policy of TD-FQ.

hand, if flow i is over-served (i.e. leading), the *Graceful Degradation Scheme* will be activated to check if flow i is still eligible for the service. If flow i has to give up its service, the service will be transferred to a CS. In both cases of CS and ES, the *Compensation Scheme* will be triggered, trying to select another flow j to serve. If the scheme fails to select any flow, this service is wasted, called a lost service (LS). If the scheme still selects flow i to serve, then we update v_i and send its HOL packet. If a flow j ($\neq i$) is selected, flow j 's packet will be sent and the values of v_i , lag_i , and lag_j are updated as follows:

$$v_i = v_i + l_p/r_i, \quad (1)$$

$$\text{lag}_i = \text{lag}_i + l_p, \quad (2)$$

$$\text{lag}_j = \text{lag}_j - l_p, \quad (3)$$

where p' is the packet being sent. Note that in this case we 'charge' to flow i by increasing its virtual time, but 'credit' (resp., debit) to lag_i (resp., lag_j) of flow i (resp., j).

Whenever the scheduler serves the HOL packet of any flow i , it has to check the queue size of flow i . If it finds that flow i 's queue is empty, it will invoke the *Lag Redistributing Scheme* to adjust flow i 's lag, if necessary.

Below, we introduce the three schemes, Graceful Degradation Scheme, Compensation Scheme, and Lag Redistributing Scheme, in TD-FQ. Table 1 summarizes notations used in TD-FQ.

3.3. Graceful degradation scheme

When a leading flow i is selected for service, the Graceful Degradation Scheme will be triggered to check its

Table 1
Summary of symbols used in TD-FQ

Symbols	Definition
v_i	Virtual time of flow i
lag_i	The credits/debits of flow i
r_i	Weight of flow i
s_i	Graceful degradation service index of flow i when $\text{lag}_i < 0$
α_R, α_N	Graceful degradation ratios for real-time and non-real-time flows
δ	The threshold to distinguish seriously/moderately lagging flows
$L_R, L_N, L_R^S, L_R^M, L_N^S, L_N^M$	Lagging flows (defined in CWC)
$W_R, W_N, W_R^S, W_R^M, W_N^S, W_N^M$	Weights of lagging flows $L_R, L_N, L_R^S, L_R^M, L_N^S, L_N^M$, respectively
$G_R, G_N, G_R^S, G_R^M, G_N^S, G_N^M$	Normalized amounts of ES/CS received by $L_R, L_N, L_R^S, L_R^M, L_N^S, L_N^M$, respectively
B	Bound of differences of services (used in CWC)
c_i^S, c_i^M	Normalized amounts of ES/CS received by flow i when $\text{lag}_i/r_i \geq \delta$ and $0 < \text{lag}_i/r_i < \delta$, respectively
f_i	Normalized amount of ES received by flow i when $\text{lag}_i \leq 0$

leading amount. Here we adopt the idea in CIF-Q to limit the amount of such services a leading flow may enjoy. The scheme in CIF-Q works as follows. A leading flow is allowed to receive an amount of additional service proportional to its NSs. Specifically, when a flow i transits from lagging/satisfied to leading, we set up a parameter $s_i = \alpha v_i$, where α ($0 \leq \alpha \leq 1$) is a system-defined constant. Later on, flow i 's virtual time will be increased each time when it is selected by the scheduler (note that 'selected' does not mean that it is actually 'served'). Let v'_i be flow i 's current virtual time when it is selected. We will allow flow i to be served if $s_i \leq \alpha v'_i$. If so, s_i is updated as $s_i + l_p/r_i$, where l_p is the length of the packet. Intuitively, flow i can enjoy approximately $\alpha(v'_i - v_i)$ services, and this is called 'graceful degradation'.

TD-FQ adopts the above idea. Further, to distinguish real-time from non-real time flows, we substitute α by a parameter α_R for real-time flows, and by α_N for non-real-time flows. We set $\alpha_R > \alpha_N$ to distinguish their priorities.

3.4. Compensation scheme

When the selected flow i has a bad channel or fails to satisfy the graceful degradation condition, the Compensation Scheme will be triggered (reflected by ES and CS in Fig. 2). In this case, lagging flows should always have a higher priority over non-lagging flows to receive such additional services. Section 3.4.1 discusses how to choose a lagging flow. Section 3.4.2 deals with the case when all lagging flows are experiencing error.

3.4.1. Dispatching ES and CS to lagging flows

The Compensation Scheme first tries to dispatch ES/CS to lagging flows. We propose a class-based weight compensation (CWC) mechanism, as shown in Fig. 3. CWC first divides lagging flows into a real-time set L_R and a non-real-time set L_N . These sets are each further divided into a seriously lagging set and a moderately lagging set. Individual flows are at the bottom. The concept of weight is used to dispatch services to these sets.

To dispatch ES/CS to L_R and L_N , we assign weights W_R and W_N to them, respectively. (Normally, we would set $W_R \geq W_N$.) Also, a variable G_R (resp., G_N) is used to record the normalized ES/CS received by L_R (resp., L_N). When both L_R and L_N have error-free flows, the service will be given to L_R if $G_R \leq G_N$, and to L_N otherwise. When only one of L_R and L_N has error-free flows, the service will be given to that one, independent of the values of G_R and G_N . When L_R receives the service, G_R is updated as

$$G_R = \min \left\{ G_R + \frac{l_p}{W_R}, \frac{B + G_N W_N}{W_R} \right\}, \quad (4)$$

where l_p is the length of the transmitted packet, and B is a predefined value to bound the difference between G_R and G_N . Similarly, when L_N receives the service, G_N is

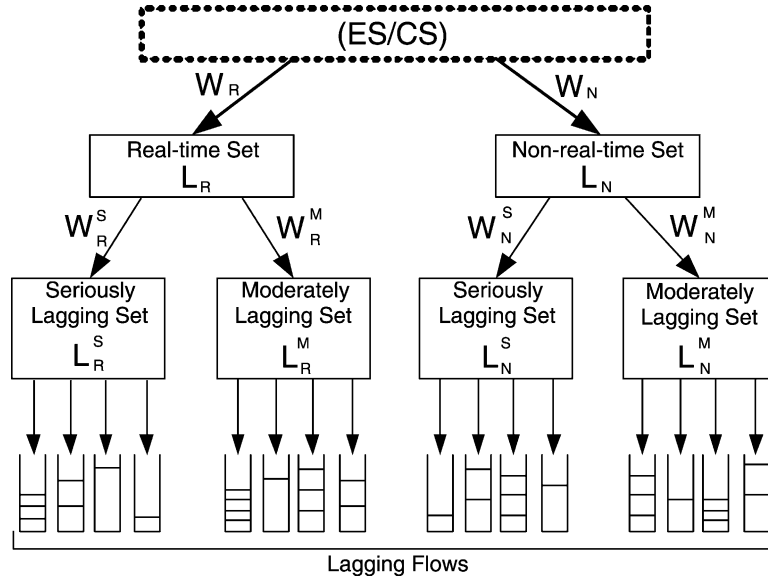


Fig. 3. Structure of the class-based weight compensation (CWC) scheme.

updated as

$$G_N = \min \left\{ G_N + \frac{l_p}{W_N}, \frac{B + G_R W_R}{W_N} \right\}. \quad (5)$$

Note that to avoid the cases of $G_R \gg G_N$ or $G_N \gg G_R$, which may cause L_R or L_N to starve when the other set recovers from error, we set up a bound $|G_R W_R - G_N W_N| \leq B$. This gives the second term in the right-hand side of Eqs. (4) and (5).

The flows in L_R are further divided into a seriously lagging set L_R^S and a moderately lagging set L_R^M . We assign a real-time lagging flow i to L_R^S if $\text{lag}_i/r_i \geq \delta$, where δ is a predefined value. Otherwise, flow i is assigned to L_R^M . Similarly, the flows in L_N are divided into a seriously lagging set L_N^S and a moderately lagging set L_N^M . Again, services are dispatched to sets L_R^S, L_R^M, L_N^S and L_N^M according to their weights W_R^S, W_R^M, W_N^S , and W_N^M , respectively. To favor seriously lagging flows, we suggest that $W_R^S \geq W_R^M$ and $W_N^S \geq W_N^M$. Services are dispatched to these sets similar to the earlier case (i.e. the service distribution to L_R and L_N). We use G_R^S, G_R^M, G_N^S and G_N^M to record the services received by these sets. Again a bound B is set to limit the differences between G_R^S and G_R^M and between G_N^S and G_N^M .

At the bottom of CWC are four groups of individual flows of the same properties (traffic types and lagging degrees). Here the scheduler dispatches ES/CS proportional to flows' weights. Specifically, for each flow i , we maintain two *compensation virtual times* c_i^S and c_i^M , which keep track of the normalized amount of ES/CS received by flow i when $\text{lag}_i/r_i \geq \delta$ and $0 < \text{lag}_i/r_i < \delta$, respectively. When the scheduler chooses the seriously lagging set (L_R^S or L_N^S), it selects the *error-free* flow i with the smallest c_i^S in the set to serve. Similarly, when the scheduler chooses the moderately lagging set (L_R^M or L_N^M), it selects the error-free flow i with the smallest c_i^M in the set to serve. When a lagging

flow i receives such a service, its compensation virtual times are updated as

$$\begin{cases} c_i^S = c_i^S + l_p/r_i, & \text{if } \text{lag}_i/r_i \geq \delta \\ c_i^M = c_i^M + l_p/r_i, & \text{otherwise} \end{cases}$$

When a flow i newly enters one of the sets L_R^S, L_R^M, L_N^S , and L_N^M or transits from one set to another, we have to assign its c_i^S or c_i^M as follows. If flow i is seriously lagging (i.e. $\text{lag}_i/r_i \geq \delta$), we set

$$c_i^S = \begin{cases} \max\{c_i^S, c_{\min}^{SR}\}, & \text{if flow } i \text{ is real-time} \\ \max\{c_i^S, c_{\min}^{SN}\}, & \text{if flow } i \text{ is non-real-time} \end{cases}$$

Otherwise, we set

$$c_i^M = \begin{cases} \max\{c_i^M, c_{\min}^{MR}\}, & \text{if flow } i \text{ is real-time} \\ \max\{c_i^M, c_{\min}^{MN}\}, & \text{if flow } i \text{ is non-real-time} \end{cases}$$

where c_{\min}^{SR} (resp., c_{\min}^{SN}) is the minimum value of c_j^S such that $j \in L_R^S$ (resp., $j \in L_N^S$), and c_{\min}^{MR} (resp., c_{\min}^{MN}) is the minimum value of c_j^M such that $j \in L_R^M$ (resp., $j \in L_N^M$). One exception is when the set $L_R^S/L_N^S/L_R^M/L_N^M$ is empty, in which case $c_{\min}^{SR}/c_{\min}^{SN}/c_{\min}^{MR}/c_{\min}^{MN}$ is undefined. If so, we set $c_{\min}^{SR}/c_{\min}^{SN}/c_{\min}^{MR}/c_{\min}^{MN}$ to the value of c_j^S/c_j^M of the 'last flow' j that left the set $L_R^S/L_N^S/L_R^M/L_N^M$.

The main contribution of CWC is that it compensates more services for real-time flows and for seriously lagging flows, thus alleviating these flows' queuing delays. Besides, CWC does not starve other lagging flows because these flows can still share a fraction of ES/CS.

3.4.2. Dispatching ES to non-lagging flows

If there is no lagging flow selected in the previous stage (due to errors), the service will be dispatched according to its original type. If the service comes from CS, it will be

returned back to the originally selected flow. Otherwise, the (ES) service will be given to a non-lagging flow. Just like CIF-Q, TD-FQ also dispatches ES proportional to those non-lagging flows' weights. That is, each flow i is assigned with an extra virtual time f_i to keep track of the normalized amount of ES received by flow i when it is non-lagging ($\text{lag}_i \leq 0$). Whenever a backlogged flow i becomes error-free and non-lagging, f_i is set to

$$f_i = \max\{f_i, \min\{f_j | \text{flow } j \text{ is error-free,}$$

backlogged, and non-lagging, $j \neq i\}\}$.

The scheduler selects the flow i with the smallest f_i value among all *error-free*, *backlogged*, and *non-lagging* flows to serve. When flow i receives the service, f_i is updated as $(f_i + l_p/r_i)$. An exception occurs when there is no selectable non-lagging flow, in which case this time slot will simply be wasted.

3.5. Lag redistributing scheme

After a flow is served, if its queue state changes to unbacklogged and it is still lagging, we will distribute its credit to other flows that are in debt and reset its credit to zero. This is because the flow does not need the credit any more [15]. This is done by the Lag Redistribution Scheme.

The scheme examines the flow i that is actually served in this round. After the service, if flow i 's queue becomes empty and $\text{lag}_i > 0$, we will give its credit to other flows in debt proportional to their weights, i.e. for each flow k such that $\text{lag}_k < 0$, we set

$$\text{lag}_k = \text{lag}_k + \text{lag}_i \times \frac{r_k}{\sum_{\text{lag}_m < 0} r_m}.$$

Then we reset $\text{lag}_i = 0$. Our redistribution rule is slightly different from CIF-Q (where all flows, including lagging ones, will share the credit). We feel that it makes sense to give these credits to only those flows in need of services.

Table 2 summarizes the major differences between TD-FQ and CIF-Q.

Table 2

The major differences between TD-FQ and CIF-Q

Schemes	TD-FQ	CIF-Q
Graceful degradation scheme	Real-time leading flows can receive more additional services	Each leading flow receives the same ratio of additional services
Compensation scheme	Dispatch ES/CS to lagging flows by CWC, which treats real-time and non-real-time flows in different ways	Dispatch ES/CS to lagging flows proportional to their weights
Lag redistributing scheme	Distribute the lag of an unbacklogged lagging flow to all other leading flows	Distribute the lag of an unbacklogged lagging flow to all other flows

4. Theoretical analyses

In this section, we analyze the fairness and delay properties of TD-FQ. Our proof relies on the following assumptions: (i) $\alpha_R \geq \alpha_N$, (ii) $W_R \geq W_N$, (iii) $W_R^S \geq W_R^M$, (iv) $W_N^S \geq W_N^M$, and (v) $B \geq \hat{L}_{\max}$, where \hat{L}_{\max} is the maximum length of a packet.

4.1. Fairness properties

Lemmas 1–3 give bounds on the differences between virtual times (v_i 's), extra virtual times (f_i 's), and compensation virtual times (c_i^S 's and c_i^M 's) of any two active flows.

Lemma 1. Let $v_i(t)$ be the virtual time of flow i at time t . For any two active flows i and j such that $t \geq 0$, we have

$$-\frac{\hat{L}_{\max}}{r_j} \leq v_i(t) - v_j(t) \leq \frac{\hat{L}_{\max}}{r_i}. \quad (6)$$

Proof. This proof is by induction on t .

Basic step. When $t=0$, all virtual times are 0, so Eq. (6) holds trivially.

Induction step. Suppose that at time t , Eq. (6) holds. Let $t + \Delta_t$ be the nearest time when any flow changes its virtual time. We want to prove Eq. (6) for time $t + \Delta_t$. Observe that a flow's virtual time may be updated in two cases: (1) it is selected by the scheduler and the service does not become a LS, and (2) it becomes active.

In Case (1), let flow i be selected by the scheduler. Then its virtual time becomes

$$v_i(t + \Delta_t) = v_i(t) + \frac{l_p}{r_i},$$

where l_p is the length of the packet being transmitted (not necessarily flow i 's). By TD-FQ, it follows that $v_i(t) \leq v_j(t)$, for all $j \in A$. Since v_i is increased, by induction hypothesis, we have

$$-\frac{\hat{L}_{\max}}{r_j} \leq v_i(t + \Delta_t) - v_j(t) = v_i(t + \Delta_t) - v_j(t + \Delta_t).$$

Further, since $v_i(t) \leq v_j(t)$, we have

$$v_i(t + \Delta_t) - v_j(t + \Delta_t) = \left(v_i(t) + \frac{l_p}{r_i} \right) - v_j(t) \leq \frac{l_p}{r_i} \leq \frac{\hat{L}_{\max}}{r_i}.$$

So Eq. (6) holds at $t + \Delta_t$.

In Eq. (6), if flow j is selected by the scheduler, then $v_i(t + \Delta_t) - v_j(t + \Delta_t) \leq \hat{L}_{\max}/r_i$ holds trivially. Further,

$$\begin{aligned} v_i(t + \Delta_t) - v_j(t + \Delta_t) &= v_i(t) - \left(v_j(t) + \frac{l_p}{r_j} \right) \\ &\geq -\frac{l_p}{r_j} \geq -\frac{\hat{L}_{\max}}{r_j}. \end{aligned}$$

So Eq. (6) still holds at $t + \Delta_t$.

In Case (2), suppose that flow i becomes active at $t + \Delta_t$. By TD-FQ, $v_i(t + \Delta_t)$ is set to $\max\{v_i(t), \min_{k \in A - \{i\}}\{v_k(t + \Delta_t)\}\}$. If $v_i(t + \Delta_t) = \min_{k \in A - \{i\}}\{v_k(t + \Delta_t)\}$, then Eq. (6) holds trivially. Otherwise, $v_i(t + \Delta_t) = v_i(t)$, which means that $v_i(t) \geq \min_{k \in A - \{i\}}\{v_k(t + \Delta_t)\}$. So we have

$$v_i(t + \Delta_t) - v_j(t + \Delta_t) \geq \min_{k \in A - \{i\}}\{v_k(t + \Delta_t)\} - v_j(t + \Delta_t) \geq -\frac{\hat{L}_{\max}}{r_j}.$$

Since the virtual time is non-decreasing, we have

$$v_i(t + \Delta_t) - v_j(t + \Delta_t) \leq v_i(t) - v_j(t) \leq \frac{\hat{L}_{\max}}{r_i}.$$

So Eq. (6) holds at $t + \Delta_t$. When flow j (instead of i) becomes active, the proof is similar, so we can conclude the proof. \square

Because TD-FQ updates f_i , c_i^S , and c_i^M similarly to that of the v_i , proofs of Lemmas 2 and 3 are similar to that of Lemma 1. So we omit the proofs.

Lemma 2. Let $f_i(t)$ be the extra virtual time of flow i at time t . For any two active flows i and j such that $t \geq 0$, we have

$$-\frac{\hat{L}_{\max}}{r_j} \leq f_i(t) - f_j(t) \leq \frac{\hat{L}_{\max}}{r_i}.$$

Lemma 3. Let $c_i^S(t)$ and $c_i^M(t)$ be the compensation virtual times of flow i at time t . For any two active flows i and j which have the same traffic type (real-time or non-real-time) such that $t \geq 0$, we have

$$\begin{cases} -\frac{\hat{L}_{\max}}{r_j} \leq c_i^S(t) - c_j^S(t) \leq \frac{\hat{L}_{\max}}{r_i} & \text{if both flows are seriously lagging} \\ -\frac{\hat{L}_{\max}}{r_j} \leq c_i^M(t) - c_j^M(t) \leq \frac{\hat{L}_{\max}}{r_i} & \text{if both flows are moderately lagging} \end{cases}.$$

Lemma 4 gives bounds on the difference between the normalized services received by a leading flow i (i.e. s_i) and the maximum amount that it can receive (i.e. $\alpha_i v_i$).

Lemma 4. Let $s_i(t)$ be the value of s_i at time t . For any flow i that is error-free, backlogged, and leading during the time interval $t \in [t_1, t_2]$, we have

$$(\alpha - 1) \frac{\hat{L}_{\max}}{r_i} \leq \alpha v_i(t) - s_i(t) \leq \alpha \frac{\hat{L}_{\max}}{r_i}, \quad (7)$$

where $\alpha = \alpha_R$ if flow i is a real-time flow, and $\alpha = \alpha_N$ otherwise.

Proof. The proof is by induction on time $t \in [t_1, t_2]$.

Basic step. When $t = t_1$, flow i just becomes leading, and the Graceful Degradation Scheme will set $s_i(t) = \alpha v_i(t)$, so the lemma is trivially true.

Induction step. Suppose that at time t , the lemma holds. Observe that v_i and/or s_i change only when flow i is selected. So we consider two cases: (1) flow i is actually served,

and (2) another flow $j \neq i$ is served. Let $t + \Delta_t \leq t_2$ be the nearest time that v_i and/or s_i are updated. We want to prove that the lemma still holds at $t + \Delta_t$.

According to TD-FQ, Case (1) occurs only when $s_i(t) \leq \alpha v_i(t)$, so we have

$$\begin{aligned} \alpha v_i(t + \Delta_t) - s_i(t + \Delta_t) &= \alpha \left(v_i(t) + \frac{l_p}{r_i} \right) - \left(s_i(t) + \frac{l_p}{r_i} \right) \\ &= (\alpha - 1) \frac{l_p}{r_i} + \alpha v_i(t) - s_i(t) \geq (\alpha - 1) \frac{\hat{L}_{\max}}{r_i}, \end{aligned}$$

where l_p represents the length of the packet being transmitted.

Case (2) implies $s_i(t) > \alpha v_i(t)$. Also, v_i is updated but s_i is not. So we have

$$\begin{aligned} \alpha v_i(t + \Delta_t) - s_i(t + \Delta_t) &= \alpha \left(v_i(t) + \frac{l_p}{r_i} \right) \\ &\quad - s_i(t) < \alpha \frac{l_p}{r_i} \leq \alpha \frac{\hat{L}_{\max}}{r_i}. \end{aligned} \quad \square$$

Theorems 1–3 show the fairness property guaranteed by TD-FQ. Theorem 1 is for flows of the same traffic type, while Theorem 2 is for flows of different types. Theorem 3 provides some bounds on differences of services received by L_R , L_N , L_R^S , L_R^M , L_N^S and L_N^M .

Theorem 1. For any two active flows i and j of the same traffic type, the difference between the normalized services received by flows i and j in any time interval $[t_1, t_2]$ during which both flows are continuously backlogged, error-free, and remain in the same state (leading, seriously lagging, moderately lagging, or satisfied) satisfies the inequality:

$$\left| \frac{\Phi_i(t_1, t_2)}{r_i} - \frac{\Phi_j(t_1, t_2)}{r_j} \right| \leq \varepsilon \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right),$$

where $\Phi_i(t_1, t_2)$ represents the services received by flow i during $[t_1, t_2]$, $\varepsilon = 3$ if both flows belong to the same lagging set (L_R^S , L_R^M , L_N^S , or L_N^M) or both flows are satisfied, $\varepsilon = 3 + \alpha_R$ if both flows are real-time leading flows, and $\varepsilon = 3 + \alpha_N$ if both flows are non-real-time leading flows.

Proof. We consider the four cases: flows i and j are both (1) seriously lagging, (2) moderately lagging, (3) satisfied, and (4) leading and backlogged during the entire time interval $[t_1, t_2]$.

Case (1): In this case, any flow i that is seriously lagging can receive services each time when it is selected (by v_i), or when it receives ES/CS from another flow (by c_i^S). Since v_i and c_i^S are updated *before* a packet is transmitted, the services received by flow i may deviate from what really reflects by its virtual times by one packet, so

$$\begin{aligned} v_i(t_2) - v_i(t_1) + c_i^S(t_2) - c_i^S(t_1) - \frac{\hat{L}_{\max}}{r_i} &\leq \frac{\Phi_i(t_1, t_2)}{r_i} \\ &\leq v_i(t_2) - v_i(t_1) + c_i^S(t_2) - c_i^S(t_1) + \frac{\hat{L}_{\max}}{r_i}. \end{aligned} \quad (8)$$

Applying Eq. (8) to flows i and j , we have

$$\begin{aligned} & v_i(t_2) - v_i(t_1) + c_i^S(t_2) - c_i^S(t_1) - \frac{\hat{L}_{\max}}{r_i} \\ & - \left(v_j(t_2) - v_j(t_1) + c_j^S(t_2) - c_j^S(t_1) + \frac{\hat{L}_{\max}}{r_j} \right) \leq \frac{\Phi_i(t_1, t_2)}{r_i} \\ & - \frac{\Phi_j(t_1, t_2)}{r_j} \leq v_i(t_2) - v_i(t_1) + c_i^S(t_2) - c_i^S(t_1) + \frac{\hat{L}_{\max}}{r_i} \\ & - \left(v_j(t_2) - v_j(t_1) + c_j^S(t_2) - c_j^S(t_1) - \frac{\hat{L}_{\max}}{r_j} \right). \end{aligned}$$

By Lemmas 1 and 3, the leftmost term can be reduced to

$$\begin{aligned} & v_i(t_2) - v_j(t_2) - (v_i(t_1) - v_j(t_1)) + c_i^S(t_2) - c_j^S(t_2) - (c_i^S(t_1) \\ & - c_j^S(t_1)) - \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right) \geq -3 \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right). \end{aligned}$$

Similarly, the right-most term would be less than or equal to $3(\hat{L}_{\max}/r_i + \hat{L}_{\max}/r_j)$, which leads to

$$\left| \frac{\Phi_i(t_1, t_2)}{r_i} - \frac{\Phi_j(t_1, t_2)}{r_j} \right| \leq 3 \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right).$$

Case (2): This case is similar to Case (1). So we can replace c_i^S and c_j^S by c_i^M and c_j^M , respectively, and obtain an inequality similar to Eq. (8). This will lead to a $\varepsilon=3$ too.

Case (3): In this case, both flows can receive services each time when they are selected (by v_i), or when they receive ES from another flow (by f_i). So we have

$$\begin{aligned} & v_i(t_2) - v_i(t_1) + f_i(t_2) - f_i(t_1) - \frac{\hat{L}_{\max}}{r_i} \leq \frac{\Phi_i(t_1, t_2)}{r_i} \leq v_i(t_2) \\ & - v_i(t_1) + f_i(t_2) - f_i(t_1) + \frac{\hat{L}_{\max}}{r_i}. \end{aligned}$$

Consequently, similar to Case (1), by Lemmas 1 and 2, we can obtain

$$\left| \frac{\Phi_i(t_1, t_2)}{r_i} - \frac{\Phi_j(t_1, t_2)}{r_j} \right| \leq 3 \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right).$$

Case (4): An error-free, backlogged, and leading flow i can receive NS (by s_i) and ES from other flows (by f_i). So the total services received by the flow i during $[t_1, t_2]$ is bounded as

$$\begin{aligned} & s_i(t_2) - s_i(t_1) + f_i(t_2) - f_i(t_1) - \frac{\hat{L}_{\max}}{r_i} \leq \frac{\Phi_i(t_1, t_2)}{r_i} \\ & \leq s_i(t_2) - s_i(t_1) + f_i(t_2) - f_i(t_1) + \frac{\hat{L}_{\max}}{r_i}. \end{aligned} \quad (9)$$

Applying Lemma 4 twice to flows i and j and subtracting one by the other, we have

$$\begin{aligned} & \alpha(v_i(t) - v_j(t)) + \alpha \left(\frac{\hat{L}_{\max}}{r_j} - \frac{\hat{L}_{\max}}{r_i} \right) - \frac{\hat{L}_{\max}}{r_j} \leq s_i(t) - s_j(t) \\ & \leq \alpha(v_i(t) - v_j(t)) + \alpha \left(\frac{\hat{L}_{\max}}{r_j} - \frac{\hat{L}_{\max}}{r_i} \right) + \frac{\hat{L}_{\max}}{r_i}. \end{aligned}$$

By Lemma 1, we can rewrite the inequality as

$$-\alpha \frac{\hat{L}_{\max}}{r_i} - \frac{\hat{L}_{\max}}{r_j} \leq s_i(t) - s_j(t) \leq \alpha \frac{\hat{L}_{\max}}{r_j} + \frac{\hat{L}_{\max}}{r_i}. \quad (10)$$

Applying Eq. (10) and Lemma 2 to Eq. (9), we have

$$\left| \frac{\Phi_i(t_1, t_2)}{r_i} - \frac{\Phi_j(t_1, t_2)}{r_j} \right| \leq (3 + \alpha) \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right),$$

where $\alpha = \alpha_R$ if these flows are real-time, and $\alpha = \alpha_N$ if they are non-real-time. \square

Theorem 2. For any real-time flow i and non-real-time flow j , the difference between the normalized services received by flows i and j in any time interval $[t_1, t_2]$ during which both flows are continuously backlogged, error-free, and remain leading satisfies the inequality:

$$\left| \frac{\Phi_i(t_1, t_2)}{r_i} - \frac{\Phi_j(t_1, t_2)}{r_j} \right| \leq 3 \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right) + 2\alpha_N \frac{\hat{L}_{\max}}{r_j}. \quad (11)$$

Proof. Applying Lemma 4 to flows i and j and taking a subtract leads to

$$\begin{aligned} & \alpha_R v_i(t) - \alpha_R \frac{\hat{L}_{\max}}{r_i} - \left(\alpha_N v_j(t) - (\alpha_N - 1) \frac{\hat{L}_{\max}}{r_j} \right) \leq s_i(t) \\ & - s_j(t) \leq \alpha_R v_i(t) - (\alpha_R - 1) \frac{\hat{L}_{\max}}{r_i} \\ & - \left(\alpha_N v_j(t) - \alpha_N \frac{\hat{L}_{\max}}{r_j} \right) = T. \end{aligned} \quad (12)$$

By Lemma 1 and the $\alpha_R \geq \alpha_N$ principle, the left-hand side of Eq. (12) becomes

$$\begin{aligned} & \alpha_R v_i(t) - \alpha_N v_j(t) + \alpha_N \frac{\hat{L}_{\max}}{r_j} - \alpha_R \frac{\hat{L}_{\max}}{r_i} - \frac{\hat{L}_{\max}}{r_j} \\ & \geq \alpha_N (v_i(t) - v_j(t)) + \alpha_N \frac{\hat{L}_{\max}}{r_j} - \alpha_R \frac{\hat{L}_{\max}}{r_i} - \frac{\hat{L}_{\max}}{r_j} \\ & \geq -\alpha_R \frac{\hat{L}_{\max}}{r_i} - \frac{\hat{L}_{\max}}{r_j}. \end{aligned}$$

Consider the right-hand side of Eq. (12). There are two cases for the term $\alpha_R v_i(t) - \alpha_N v_j(t)$. If $\alpha_R v_i(t) - \alpha_N v_j(t) \geq 0$, we have $v_i(t) \geq \alpha_N / \alpha_R v_j(t)$. By Lemma 1,

$$\begin{aligned} T &\leq \alpha_N (v_j(t) - v_i(t)) + \alpha_N \frac{\hat{L}_{\max}}{r_j} - \alpha_R \frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_i} \\ &\leq 2\alpha_N \frac{\hat{L}_{\max}}{r_j} - \alpha_R \frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_i}. \end{aligned}$$

If $\alpha_R v_i(t) - \alpha_N v_j(t) < 0$, we have

$$T \leq \alpha_N \frac{\hat{L}_{\max}}{r_j} - \alpha_R \frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_i}.$$

These two cases together imply

$$T \leq 2\alpha_N \frac{\hat{L}_{\max}}{r_j} - \alpha_R \frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_i}.$$

So we have

$$-\alpha_R \frac{\hat{L}_{\max}}{r_i} - \frac{\hat{L}_{\max}}{r_j} \leq s_i(t) - s_j(t) \leq 2\alpha_N \frac{\hat{L}_{\max}}{r_j} + (1 - \alpha_R) \frac{\hat{L}_{\max}}{r_i}.$$

Similar to the proof of Theorem 1, the service received by any leading flow i during $[t_1, t_2]$ satisfies Eq. (9). Subtracting Eq. (9) of flow i by Eq. (9) of flow j leads to

$$\begin{aligned} s_i(t_2) - s_i(t_1) + f_i(t_2) - f_i(t_1) - \frac{\hat{L}_{\max}}{r_i} \\ - \left(s_j(t_2) - s_j(t_1) + f_j(t_2) - f_j(t_1) + \frac{\hat{L}_{\max}}{r_j} \right) &\leq \frac{\Phi_i(t_1, t_2)}{r_i} \\ - \frac{\Phi_j(t_1, t_2)}{r_j} \leq s_i(t_2) - s_i(t_1) + f_i(t_2) - f_i(t_1) + \frac{\hat{L}_{\max}}{r_i} \\ - \left(s_j(t_2) - s_j(t_1) + f_j(t_2) - f_j(t_1) - \frac{\hat{L}_{\max}}{r_j} \right), \end{aligned}$$

The leftmost term can be reduced to

$$\begin{aligned} s_i(t_2) - s_j(t_2) - (s_i(t_1) - s_j(t_1)) + f_i(t_2) - f_j(t_2) - (f_i(t_1) \\ - f_j(t_1)) - \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right) &\geq -\alpha_R \frac{\hat{L}_{\max}}{r_i} - \frac{\hat{L}_{\max}}{r_j} \\ - 2\alpha_N \frac{\hat{L}_{\max}}{r_j} + (\alpha_R - 1) \frac{\hat{L}_{\max}}{r_i} - 2 \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right) \\ &= -3 \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right) - 2\alpha_N \frac{\hat{L}_{\max}}{r_j}. \end{aligned}$$

Similarly, the right-most term would be less than or equal to

$$3 \left(\frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right) + 2\alpha_N \frac{\hat{L}_{\max}}{r_j}.$$

Thus, Eq. (11) holds. \square

Lemma 5. Let $G_R(t)$, $G_N(t)$, $G_R^S(t)$, $G_R^M(t)$, $G_N^S(t)$ and $G_N^M(t)$ be the value of G_R , G_N , G_R^S , G_R^M , G_N^S , and G_N^M at time t , respectively. For $t \geq 0$, we have

$$\begin{cases} -\frac{B}{W_N} \leq G_R(t) - G_N(t) \leq \frac{B}{W_R} \\ -\frac{B}{W_R^M} \leq G_R^S(t) - G_R^M(t) \leq \frac{B}{W_R^S} \\ -\frac{B}{W_N^M} \leq G_N^S(t) - G_N^M(t) \leq \frac{B}{W_N^S} \end{cases}$$

Proof. This proof is by induction on time $t \geq 0$.

Basic step. When $t=0$, $G_R(t)=G_N(t)=0$, so the lemma is trivially true.

Induction step. Assume that the lemma holds at time t . G_R (resp., G_N) is updated only when L_R or L_N is non-empty. We consider two cases: (1) only one set is non-empty, and (2) two sets are non-empty. Let $t + \Delta_t$ be the nearest time that G_R or G_N is updated. We want to prove the lemma to be true at time $t + \Delta_t$.

In Case (1), if L_R is active, then ES/CS will be given to L_R . In TD-FQ, we bound the total difference of ES/CS received by L_R and L_N at any time by $|W_R G_R - W_N G_N| \leq B$. So at time $t + \Delta_t$, $W_R G_R(t + \Delta_t) - W_N G_N(t + \Delta_t) \leq B$. Since $W_R \geq W_N$, we have

$$\begin{aligned} W_R G_R(t + \Delta_t) - W_N G_N(t + \Delta_t) &\leq W_R G_R(t + \Delta_t) \\ - W_N G_N(t + \Delta_t) \leq B &\Rightarrow G_R(t + \Delta_t) - G_N(t + \Delta_t) \leq \frac{B}{W_R}. \end{aligned}$$

On the other hand, if L_N is active, we can similarly derive that

$$G_R(t + \Delta_t) - G_N(t + \Delta_t) \geq -\frac{B}{W_N}.$$

So the first inequality in the lemma holds at $t + \Delta_t$.

In Case (2), since both sets are non-empty, the scheduler gives ES/CS to L_R if $G_R(t) \leq G_N(t)$. Let l_p represent the length of the packet being transmitted. We have

$$\begin{aligned} G_R(t + \Delta_t) - G_N(t + \Delta_t) &= \left(G_R(t) + \frac{l_p}{W_R} \right) \\ - G_N(t) &\leq \frac{l_p}{W_R} \leq \frac{\hat{L}_{\max}}{W_R} \leq \frac{B}{W_R}. \end{aligned}$$

Note that it is trivially true that $-B/W_N \leq G_R(t + \Delta_t) - G_N(t + \Delta_t)$. Similarly, if $G_R(t) > G_N(t)$, the service is given to L_N , so we have

$$\begin{aligned} G_R(t + \Delta_t) - G_N(t + \Delta_t) &= G_R(t) - \left(G_N(t) + \frac{l_p}{W_N} \right) \\ &> -\frac{l_p}{W_N} \geq -\frac{\hat{L}_{\max}}{W_N} \geq -\frac{B}{W_N}. \end{aligned}$$

Note that it is trivially true that $G_R(t + \Delta_t) - G_N(t + \Delta_t) \leq B/W_R$. Therefore, the first inequality in this lemma still holds at $t + \Delta_t$. The other two inequalities in this lemma can be proved in a similar way. \square

Theorem 3. *The difference between normalized ES/CS received by any two lagging sets in any time interval $[t_1, t_2]$ during which both sets remain active satisfies the inequalities:*

$$\begin{aligned} (1) \text{ for } L_R \text{ and } L_N : & \left| \frac{\Phi_R(t_1, t_2)}{W_R} - \frac{\Phi_N(t_1, t_2)}{W_N} \right| \leq \frac{B + \hat{L}_{\max}}{W_R} + \frac{B + \hat{L}_{\max}}{W_N}, \\ (2) \text{ for } L_R^S \text{ and } L_R^M : & \left| \frac{\Phi_R^S(t_1, t_2)}{W_R^S} - \frac{\Phi_R^M(t_1, t_2)}{W_R^M} \right| \leq \frac{B + \hat{L}_{\max}}{W_R^S} + \frac{B + \hat{L}_{\max}}{W_R^M}, \\ (3) \text{ for } L_N^S \text{ and } L_N^M : & \left| \frac{\Phi_N^S(t_1, t_2)}{W_N^S} - \frac{\Phi_N^M(t_1, t_2)}{W_N^M} \right| \leq \frac{B + \hat{L}_{\max}}{W_N^S} + \frac{B + \hat{L}_{\max}}{W_N^M}, \end{aligned}$$

where $\Phi_R(t_1, t_2)$, $\Phi_N(t_1, t_2)$, $\Phi_R^S(t_1, t_2)$, $\Phi_R^M(t_1, t_2)$, $\Phi_N^S(t_1, t_2)$, and $\Phi_N^M(t_1, t_2)$ represents ES/CS received by L_R , L_N , L_R^S , L_R^M , L_N^S and L_N^M during $[t_1, t_2]$, respectively.

Proof. Since G_R is updated before a packet is transmitted, it follows that the total ES/CS received by L_R during $[t_1, t_2]$ is bounded by

$$G_R(t_2) - G_R(t_1) - \frac{\hat{L}_{\max}}{W_R} \leq \frac{\Phi_R(t_1, t_2)}{W_R} \leq G_R(t_2) - G_R(t_1) + \frac{\hat{L}_{\max}}{W_R}.$$

Similarly, for G_N , we have

$$G_N(t_2) - G_N(t_1) - \frac{\hat{L}_{\max}}{W_N} \leq \frac{\Phi_N(t_1, t_2)}{W_N} \leq G_N(t_2) - G_N(t_1) + \frac{\hat{L}_{\max}}{W_N}.$$

Therefore, we have

$$\begin{aligned} G_R(t_2) - G_R(t_1) - \frac{\hat{L}_{\max}}{W_R} - \left(G_N(t_2) - G_N(t_1) + \frac{\hat{L}_{\max}}{W_N} \right) \\ \leq \frac{\Phi_R(t_1, t_2)}{W_R} - \frac{\Phi_N(t_1, t_2)}{W_N} \leq G_R(t_2) - G_R(t_1) + \frac{\hat{L}_{\max}}{W_R} \\ - \left(G_N(t_2) - G_N(t_1) - \frac{\hat{L}_{\max}}{W_N} \right). \end{aligned}$$

By Lemma 5, we can rewrite the inequality as

$$\begin{aligned} - \left(\frac{B + \hat{L}_{\max}}{W_R} + \frac{B + \hat{L}_{\max}}{W_N} \right) & \leq \frac{\Phi_R(t_1, t_2)}{W_R} - \frac{\Phi_N(t_1, t_2)}{W_N} \\ & \leq \frac{B + \hat{L}_{\max}}{W_R} + \frac{B + \hat{L}_{\max}}{W_N} \Rightarrow \left| \frac{\Phi_R(t_1, t_2)}{W_R} - \frac{\Phi_N(t_1, t_2)}{W_N} \right| \\ & \leq \frac{B + \hat{L}_{\max}}{W_R} + \frac{B + \hat{L}_{\max}}{W_N}. \end{aligned}$$

This concludes the first inequality. The other two inequalities in this theorem can be proved similarly. \square

4.2. Delay bounds

When a backlogged flow suffers from errors, it becomes lagging. Theorem 4 shows that if a lagging flow becomes error-free and has sufficient service demand, it can get back all its lagging services within bounded time.

Theorem 4. *If an active but lagging flow i becomes error-free at time t and remains backlogged continuously after time t , it is guaranteed that flow i will become non-lagging (i.e. $\text{lag}_i \leq 0$) within time Δ_t , where*

$$\Delta_t \leq \frac{\varphi(\Psi + 2\hat{L}_{\max})}{r_{\min}(1 - \alpha_R)\hat{C}} + \left(n + 1 + \frac{\varphi}{r_{\min}} \right) \frac{\hat{L}_{\max}}{\hat{C}},$$

n is the number of active flows, \hat{C} is the channel capacity, φ is the aggregate weight of all flows, φ_R is the aggregate weight of all real-time flows, φ_N is the aggregate weight of all non-real-time flows, r_{\min} is the minimum weight of all flows, and

$$\begin{aligned} \psi = & \frac{(W_R + W_N)(W_R^S + W_R^M)}{W_R W_R^S} \left(\frac{\text{lag}_i(t)}{r_i} \varphi_R + \left(\frac{\varphi_R}{r_i} + n - 2 \right) \hat{L}_{\max} + B \right) \\ & + \frac{W_R + W_N}{W_R} \left(\delta \varphi_R + \left(\frac{2\varphi_R}{r_i} + n - 1 \right) \hat{L}_{\max} + B \right) \end{aligned}$$

if flow i is real-time, and

$$\begin{aligned} \psi = & \frac{(W_R + W_N)(W_N^S + W_N^M)}{W_N W_N^S} \left(\frac{\text{lag}_i(t)}{r_i} \varphi_N + \left(\frac{\varphi_N}{r_i} + n - 2 \right) \hat{L}_{\max} + B \right) \\ & + \frac{W_R + W_N}{W_N} \left(\delta \varphi_N + \left(\frac{2\varphi_N}{r_i} + n - 1 \right) \hat{L}_{\max} + B \right) \end{aligned}$$

if flow i is non-real-time.

Proof. Assume that flow i is a real-time flow. Consider the worst case: flow i has the maximum lag among all flows and $\text{lag}_i/r_i \geq \delta$ at time t . Since flow i becomes error-free after time t , lag_i is decreased each time when it receives CS. Now let flow i becomes moderately lagging at time t_M , and further become non-lagging at time t_N , $t < t_M < t_N$, i.e. $i \in L_R^S$ during $[t, t_M)$ and $i \in L_R^M$ during $[t_M, t_N)$. Also, let $\Phi_C(t, t_N)$ be the total CS received by all lagging flows during $[t, t_N)$.

To prove Theorem 4, observe that Δ_t should be an upper bound of $t_N - t$. The largest value of t_N occurs when all flows in the system are error-free (i.e. no ES) and there is only one leading flow, say k , who provides CS such that flow k is a real-time flow and $r_k = r_{\min}$. Since flow k can still receive a fraction α_R of its NS when it is leading and flow k uses s_k to keep track of the amount of such NS when it is leading, this leads to

$$\Phi_C(t, t_N) \geq r_{\min}(v_k(t_N) - v_k(t)) - r_{\min}(s_k(t_N) - s_k(t)) - \hat{L}_{\max}. \tag{13}$$

By Lemma 1, for any active flow j during $[t, t_N]$, we have

$$v_j(t_N) - v_j(t) \leq v_k(t_N) - v_k(t) + \frac{\hat{L}_{\max}}{r_j} + \frac{\hat{L}_{\max}}{r_{\min}}.$$

This inequality helps to derive the total amount of services provided by the system during $[t, t_N]$:

$$\begin{aligned} \hat{C}(t_N - t) &\leq \left(\sum_{j \in A} r_j (v_j(t_N) - v_j(t)) \right) + \hat{L}_{\max} \\ &\leq \left(\sum_{j \in A} r_j \left(v_k(t_N) - v_k(t) + \frac{\hat{L}_{\max}}{r_j} + \frac{\hat{L}_{\max}}{r_{\min}} \right) \right) + \hat{L}_{\max} \\ &\leq (v_k(t_N) - v_k(t)) \sum_{j \in A} r_j + n \hat{L}_{\max} + \frac{\hat{L}_{\max}}{r_{\min}} \sum_{j \in A} r_j + \hat{L}_{\max} \\ &\leq (v_k(t_N) - v_k(t)) \varphi + \left(n + 1 + \frac{\varphi}{r_{\min}} \right) \hat{L}_{\max} \Rightarrow v_k(t_N) - v_k(t) \\ &\geq \frac{1}{\varphi} \left(\hat{C}(t_N - t) - \left(n + 1 + \frac{\varphi}{r_{\min}} \right) \hat{L}_{\max} \right). \end{aligned} \quad (14)$$

Applying Lemma 4 to flow k at times t and t_N and taking a subtract, we obtain

$$s_k(t_N) - s_k(t) \leq \alpha_R v_k(t_N) - \alpha_R v_k(t) + \frac{\hat{L}_{\max}}{r_{\min}}. \quad (15)$$

By combining Eqs. (14) and (15) into Eq. (13), we can obtain

$$\begin{aligned} \Phi_C(t, t_N) &\geq r_{\min} (v_k(t_N) - v_k(t) - (s_k(t_N) - s_k(t))) - \hat{L}_{\max} \\ &\geq r_{\min} \left(v_k(t_N) - v_k(t) - \alpha_R v_k(t_N) + \alpha_R v_k(t) - \frac{\hat{L}_{\max}}{r_{\min}} \right) - \hat{L}_{\max} \\ &= r_{\min} (1 - \alpha_R) (v_k(t_N) - v_k(t)) - 2 \hat{L}_{\max} \\ &\geq \frac{r_{\min} (1 - \alpha_R)}{\varphi} \left(\hat{C}(t_N - t) - \left(n + 1 + \frac{\varphi}{r_{\min}} \right) \hat{L}_{\max} \right) - 2 \hat{L}_{\max} \\ &\Rightarrow t_N - t \leq \frac{\varphi (\Phi_C(t, t_N) + 2 \hat{L}_{\max})}{r_{\min} (1 - \alpha_R) \hat{C}} + \left(n + 1 + \frac{\varphi}{r_{\min}} \right) \frac{\hat{L}_{\max}}{\hat{C}}. \end{aligned} \quad (16)$$

It remains to derive an upper bound for $\Phi_C(t, t_N)$ in Eq. (16). Note that there are $n - 1$ lagging flows who are allowed to share the $\Phi_C(t, t_N)$ services. The worst case happens when (1) exactly one of the $n - 1$ flows remains in L_N during $[t, t_N]$, (2) exactly $n - 3$ flows remain in L_R^S and 1 flow remains in L_R^M during $[t, t_M]$, and (3) no flow remains in L_R^S and exactly $n - 2$ flows remain in L_R^M during $[t_M, t_N]$. Note that in this case L_R can share at most a fraction $W_R / (W_R + W_N)$ of $\Phi_C(t, t_N)$ during $[t, t_N]$, and L_R^S can share at most a fraction $W_R^S / (W_R^S + W_R^M)$ of CS received by L_R during $[t, t_M]$.

Let $\Phi_R(t, t_N)$ and $\Phi_N(t, t_N)$ be CS received by L_R and L_N during $[t, t_N]$, respectively, $\Phi_C(t, t_N) = \Phi_R(t, t_N) + \Phi_N(t, t_N)$.

According to the first inequality of Theorem 3, we have

$$\begin{aligned} \Phi_N(t, t_N) &\leq W_N \left(\frac{\Phi_R(t, t_N)}{W_R} + \frac{B + \hat{L}_{\max}}{W_R} + \frac{B + \hat{L}_{\max}}{W_N} \right) \\ &\Rightarrow \Phi_C(t, t_N) \leq \frac{W_R + W_N}{W_R} (\Phi_R(t, t_N) + B + \hat{L}_{\max}). \end{aligned} \quad (17)$$

Next we derive the $\Phi_R(t, t_N)$ in Eq. (17). It can be divided into two terms,

$$\Phi_R(t, t_N) = \Phi_R(t, t_M) + \Phi_R(t_M, t_N). \quad (18)$$

Let $\Phi_R^S(t, t_M)$ and $\Phi_R^M(t, t_M)$ be CS received by L_R^S and L_R^M during $[t, t_M]$, respectively. Again, by Theorem 3, we have

$$\begin{aligned} \Phi_R(t, t_M) &= \Phi_R^S(t, t_M) + \Phi_R^M(t, t_M) \leq \Phi_R^S(t, t_M) \\ &\quad + W_R^M \left(\frac{\Phi_R^S(t, t_M)}{W_R^S} + \frac{B + \hat{L}_{\max}}{W_R^S} + \frac{B + \hat{L}_{\max}}{W_R^M} \right) \\ &= \frac{W_R^M + W_R^S}{W_R^S} (\Phi_R^S(t, t_M) + B + \hat{L}_{\max}). \end{aligned} \quad (19)$$

We further expand the term $\Phi_R^S(t, t_M)$ in Eq. (19) as follows:

$$\begin{aligned} \Phi_R^S(t, t_M) &\leq \sum_{j \in L_R^S(t, t_M)} r_j (c_j^S(t_M) - c_j^S(t)) \\ &\leq \sum_{j \in L_R^S(t, t_M)} r_j \left(c_i^S(t_M) - c_i^S(t) + \frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j} \right) \\ &= (c_i^S(t_M) - c_i^S(t)) \sum_{j \in L_R^S(t, t_M)} r_j + \frac{\hat{L}_{\max}}{r_i} \sum_{j \in L_R^S(t, t_M)} r_j \\ &\quad + \sum_{j \in L_R^S(t, t_M)} \hat{L}_{\max} < \varphi_R (c_i^S(t_M) - c_i^S(t)) \\ &\quad + \left(\frac{\varphi_R}{r_i} + n - 3 \right) \hat{L}_{\max}. \end{aligned} \quad (20)$$

Note that the fourth term in Eq. (20) is obtained by applying Lemma 3 twice on flow i and any flow $j \in L_R^S$

$$c_j^S(t_M) - c_j^S(t) \leq c_i^S(t_M) - c_i^S(t) + \frac{\hat{L}_{\max}}{r_i} + \frac{\hat{L}_{\max}}{r_j}.$$

Since L_R^S is empty during $[t_M, t_N]$, $\Phi_R(t_M, t_N) = \Phi_R^M(t_M, t_N)$. Similarly to the derivation of Eq. (20), we have

$$\begin{aligned} \Phi_R(t_M, t_N) &= \Phi_R^M(t_M, t_N) \leq \sum_{j \in L_R^M(t_M, t_N)} r_j (c_j^M(t_N) - c_j^M(t_M)) \\ &\leq \varphi_R (c_i^M(t_N) - c_i^M(t_M)) + \left(\frac{\varphi_R}{r_i} + n - 2 \right) \hat{L}_{\max}. \end{aligned} \quad (21)$$

By Eqs. (19) and (20), we have

$$\begin{aligned} \Phi_R(t, t_M) &< \frac{W_R^M + W_R^S}{W_R^S} \left(\varphi_R (c_i^S(t_M) - c_i^S(t)) + \left(\frac{\varphi_R}{r_i} + n - 2 \right) \hat{L}_{\max} + B \right). \end{aligned} \quad (22)$$

Furthermore, by combining Eqs. (21) and (22) into Eq. (18), we have

$$\begin{aligned} \Phi_R(t, t_N) &\leq \frac{W_R^M + W_R^S}{W_R^S} \left(\varphi_R(c_i^S(t_M) - c_i^S(t)) + \left(\frac{\varphi_R}{r_i} + n - 2 \right) \hat{L}_{\max} + B \right) \\ &\quad + \varphi_R(c_i^M(t_N) - c_i^M(t_M)) + \left(\frac{\varphi_R}{r_i} + n - 2 \right) \hat{L}_{\max} \\ &= \varphi_R \left(\frac{W_R^S + W_R^M}{W_R^S} (c_i^S(t_M) - c_i^S(t)) + c_i^M(t_N) - c_i^M(t_M) \right) \\ &\quad + \frac{2W_R^S + W_R^M}{W_R^S} \left(\frac{\varphi_R}{r_i} + n - 2 \right) \hat{L}_{\max} + \frac{(W_R^S + W_R^M)B}{W_R^S}. \end{aligned} \quad (23)$$

By combining Eqs. (17) and (23), we have

$$\begin{aligned} \Phi_C(t, t_N) &\leq \frac{W_R + W_N}{W_R W_R^S} (\varphi_R((W_R^S + W_R^M)(c_i^S(t_M) - c_i^S(t)) \\ &\quad + W_R^S(c_i^M(t_N) - c_i^M(t_M))) \\ &\quad + \left((2W_R^S + W_R^M) \left(\frac{\varphi_R}{r_i} + n - 2 \right) + W_R^S \right) \hat{L}_{\max} \\ &\quad + (2W_R^S + W_R^M)B). \end{aligned} \quad (24)$$

Since flow i is still lagging after time t_M , it means that $0 < \text{lag}_i(t_M) < \text{lag}_i(t)$. So

$$\begin{aligned} c_i^S(t_M) - c_i^S(t) &= \frac{|\text{lag}_i(t_M) - \text{lag}_i(t)|}{r_i} \\ &= \frac{\text{lag}_i(t) - \text{lag}_i(t_M)}{r_i} < \frac{\text{lag}_i(t)}{r_i}. \end{aligned} \quad (25)$$

After time t_N , flow i becomes non-lagging, so $-\hat{L}_{\max} < \text{lag}_i(t_N) \leq 0$. Besides, $0 < \text{lag}_i(t_M) < r_i \delta$ since flow i becomes moderately lagging after time t_M , so we have

$$\begin{aligned} c_i^M(t_N) - c_i^M(t_M) &= \frac{|\text{lag}_i(t_N) - \text{lag}_i(t_M)|}{r_i} \\ &= \frac{\text{lag}_i(t_M) - \text{lag}_i(t_N)}{r_i} < \delta + \frac{\hat{L}_{\max}}{r_i}. \end{aligned} \quad (26)$$

By combining Eqs. (25) and (26) into Eq. (24), we have

$$\begin{aligned} \Phi_C(t, t_N) &< \frac{(W_R + W_N)(W_R^S + W_R^M)}{W_R W_R^S} \left(\frac{\text{lag}_i(t)}{r_i} \varphi_R + \left(\frac{\varphi_R}{r_i} + n - 2 \right) \hat{L}_{\max} + B \right) \\ &\quad + \frac{W_R + W_N}{W_R} \left(\delta \varphi_R + \left(\frac{2\varphi_R}{r_i} + n - 1 \right) \hat{L}_{\max} + B \right). \end{aligned} \quad (27)$$

By combining Eqs. (16) and (27), the first part of this theorem is proved. When flow i is a non-real-time flow, the proof is similar and we omit the details. \square

5. Simulation results

In this section, we present some experimental results to verify the effectiveness of the proposed algorithm. The first one observes the packet dropping ratios and queuing delays of real-time flows in TD-FQ and CIF-Q, respectively. The second one compares the throughput of flows in these two algorithms. The last one gives a comparison on different compensation strategies for lagging flows.

5.1. Experiment 1: dropping ratios and delays for real-time flows

In this experiment, we mix real-time and non-real-time traffics together. We observe the packet dropping ratios and queuing delays of real-time flows in TD-FQ and CIF-Q, respectively. Eight flows are used, as shown in Table 3. The first six flows are real-time flows, which have two traffic models: constant-bit-rate (CBR) and ON-OFF model. The latter is to model voice communication. The average durations of ON and OFF states are set to 2.5 and 0.5 s, respectively. During ON period, packets are generated with fixed intervals. No packet is generated during OFF period. The last two flows are non-real-time FTP flows, and their traffics are modeled as greedy sources whose queues are never empty. As for error scenarios, we use two parameters P_{good} and P_{bad} to control the average time when the channel stays in error-free and error states, respectively. The total channel capacity is set to 5 MB/s. The total simulation time in this experiment is 100 s.

For CIF-Q, we set $\alpha=0.5$, while for TD-FQ we set $\alpha_R=0.8$ and $\alpha_N=0.2$, respectively. The weights assigned to lagging sets are $W_R:W_N=3:1$, $W_R^S:W_R^M=3:1$, and $W_N^S:W_N^M=3:1$. The packet dropping ratios and queuing delays of real-time flows are shown in Figs. 4 and 5, respectively, where the packet dropping ratio is defined as

$$\frac{\text{Number of packets dropped due to exceeding deadline}}{\text{Number of packet generated}},$$

where the deadline of a packet is set to twice of the packet interarrival time. From Figs. 4 and 5, we can observe that the packet dropping ratios and queuing delays of real-time flows in TD-FQ are smaller than those in CIF-Q, especially

Table 3

Traffic specification of the flows used in experiment 1

Flow	Guaranteed bandwidth	Packet size (kB)	Error scenario
Voice1	64 kB/s	2	No error occurs
Voice2	32 kB/s	1	$P_{\text{good}}=6$ s, $P_{\text{bad}}=1.5$ s
Voice3	32 kB/s	1	$P_{\text{good}}=5$ s, $P_{\text{bad}}=0.5$ s
CBR1	512 kB/s	2	No error occurs
CBR2	256 kB/s	1	$P_{\text{good}}=6$ s, $P_{\text{bad}}=1.5$ s
CBR3	256 kB/s	1	$P_{\text{good}}=5$ s, $P_{\text{bad}}=0.5$ s
FTP1	2 MB/s	4	No error occurs
FTP2	2 MB/s	4	$P_{\text{good}}=6$ s, $P_{\text{bad}}=1.5$ s

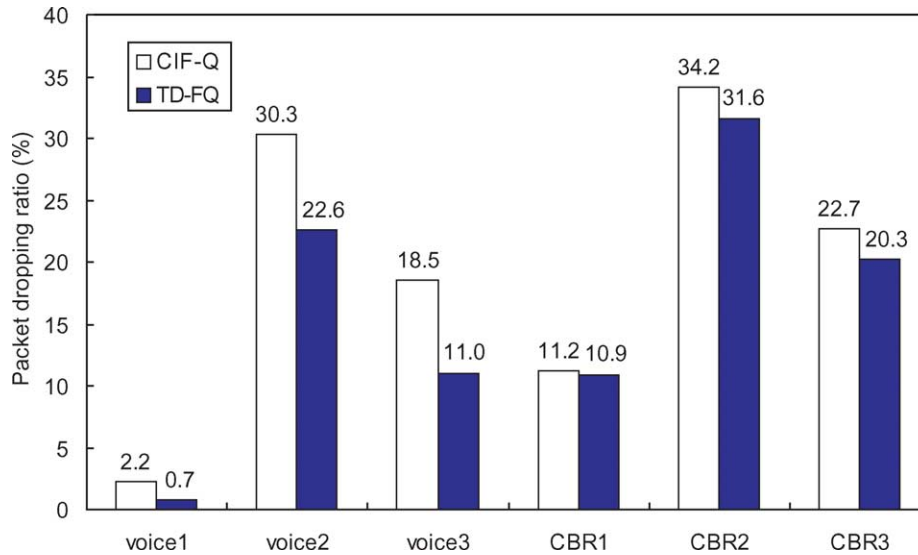


Fig. 4. Packet dropping ratios of real-time flows.

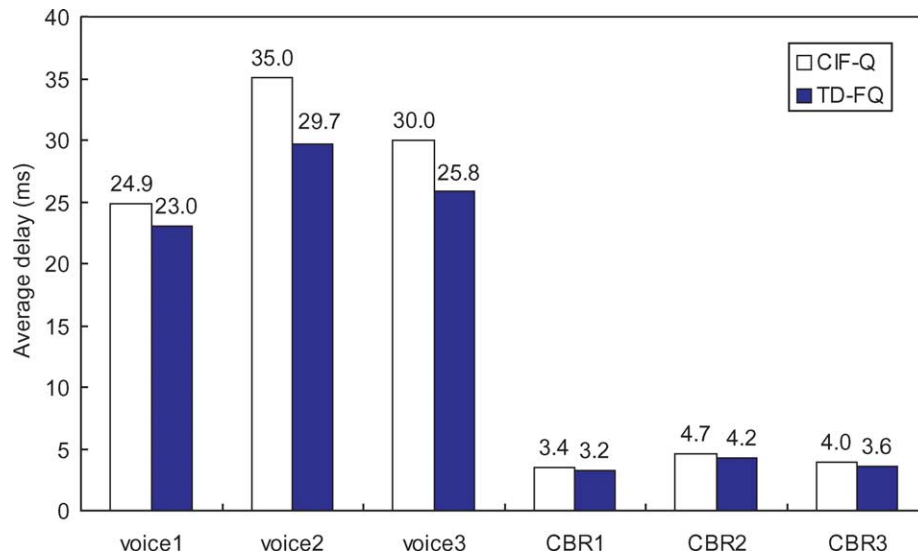


Fig. 5. Average queuing delays of real-time flows.

when the flows are voice traffic. This is because TD-FQ not only lets real-time flows give up less services to compensate other lagging flows, but also gives more services to real-time lagging flows for compensation. From this observation, we conclude that TD-FQ can alleviate the packet dropping ratios and queuing delays of real-time flows as compared to CIF-Q.

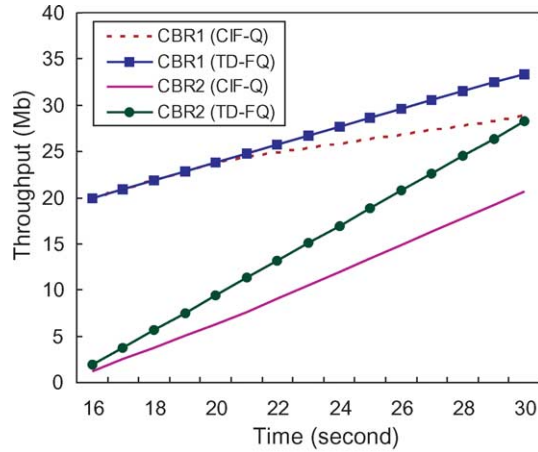
5.2. Experiment 2: throughputs of flows

In this experiment, we observe the throughputs of flows in TD-FQ and CIF-Q. Four flows are used, as shown in Table 4. The first two flows are real-time CBR flows, and the last two flows are non-real-time FTP flows. Suffering from channel errors during [0, 15) period, flows

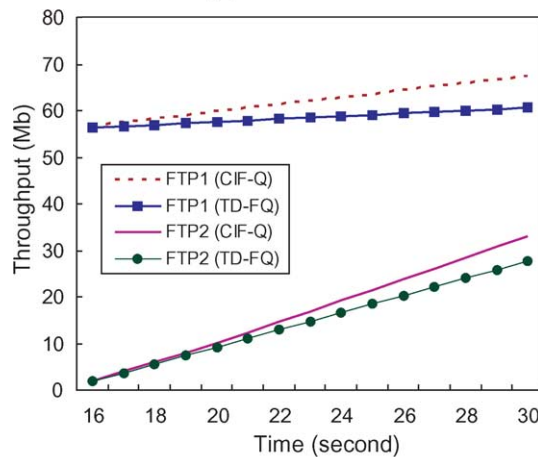
CBR2 and FTP2 will become active but lagging after the 15th second. The other flows are all leading in this experiment. For CIF-Q, we set $\alpha=0.5$, while for TD-FQ we set $\alpha_R=0.8$, $\alpha_N=0.2$, $W_R=3$, and $W_N=1$. The channel capacity in this experiment is set to 2 MB/s.

Table 4
Traffic specification of the flows used in experiment 2

Flow	Guaranteed bandwidth (MB/s)	Packet size (kB)	Error scenario
CBR1	1.25	4	No error occurs
CBR2	1.25	4	Error occurs during [0,15) s
FTP1	2	8	No error occurs
FTP2	2	8	Error occurs during [10,15) s



(a) Real-time flows



(b) Non-real-time flows

Fig. 6. Throughputs of (a) real-time flows CBR1 and CBR2 and (b) non-real-time flows FTP1 and FTP2.

Fig. 6 shows the throughput of flows after the 16th second. We see that real-time flows can receive more services in TD-FQ as compared to CIF-Q. This is because TD-FQ favors real-time flows over non-real-time flows. However, the cost, as shown in Fig. 6(b), is at lower throughputs for non-real-time flows.

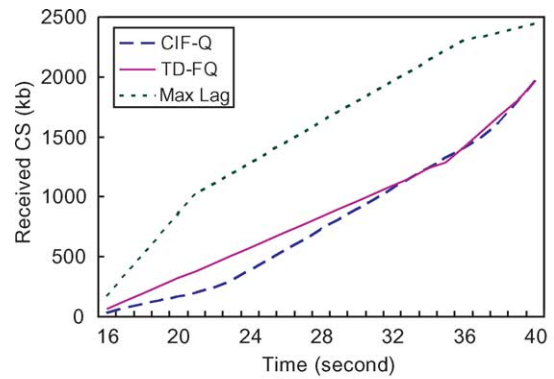
5.3. Experiment 3: effect of compensation

We compare three compensation strategies for lagging flows: (1) TD-FQ, (2) CIF-Q (which dispatches services proportional to flows’ weights), and (3) Max-lag (which always selects the error-free flow with the maximum normalized lag to serve).

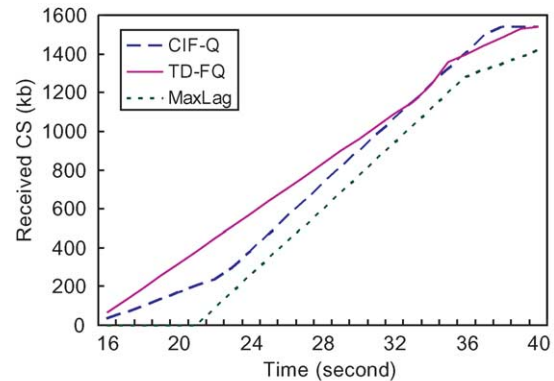
Six FTP flows are used. Table 5 shows the traffic specification of these flows. Each flows has unlimited data to transmit and each packet is of size 1 kB. The bandwidth of the BS is set to 1 MB/s. From Table 5, it is clear that flow 1 will become a leading flow after the 15th second. Flow 2 and 3 are treated as seriously lagging flows in TD-FQ, while other flows are treated moderately lagging. To let

Table 5
Traffic specification of the flows used in experiment 3

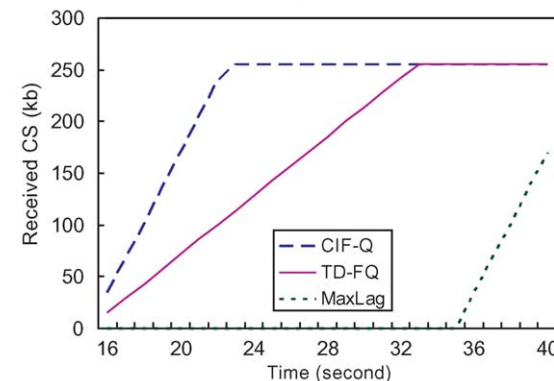
Flow no.	Traffic type	Guaranteed bandwidth (MB/s)	Error scenario
1	FTP	1	No error occurs
2	FTP	1	Error occurs during [0,15) s
3	FTP	1	Error occurs during [5,15) s
4–6	FTP	1	Error occurs during [10,15) s



(a) flow 2



(b) flow 3



(c) flows 4-6

Fig. 7. Received compensation services (CS) by (a) the seriously lagging flow 2, (b) seriously lagging flow 3, and (c) moderately lagging flows 4–6.

lagging flows receive the maximum CSs, we set $\alpha=0$ for both CIF-Q and TD-FQ. In TD-FQ, we assign weights $W_N^S=2$ and $W_N^M=1$.

Fig. 7(a) shows the total CSs that flow 2 receives after the 15th second. We see that flow 2 enjoys the most CSs in the Max-lag scheme. This remains true until flow 2's lag lowers down to the lags of other lagging flows. On the contrary, CIF-Q gives the least CSs to flow 2 because it dispatches CSs proportional to flows' weights. So flow 2 may suffer from more serious queuing delays during this period. TD-FQ performs in between what CIF-Q and Max-lag perform because it separates seriously lagging flows from moderately lagging flows. Note that after the 32th second, the behavior of flow 2 in TD-FQ is similar to that in CIF-Q. This is because after the 32th second, flows 4–6 have become non-lagging both in TD-FQ and CIF-Q (Ref. to Fig. 7(c)), and flows 2 and 3 become moderately lagging in TD-FQ. So in this case, TD-FQ works similarly to CIF-Q. Fig. 7(b) shows the behavior of flow 3, which is also seriously lagging but has less lag compared to flow 2. From Fig. 7(b), we can observe that even flow 3 is seriously lagging, it is starved until the 20th second in the Max-lag scheme. Fig. 7(c) shows our observation for flows 4–6. The result does verify that CIF-Q favors moderately lagging flows over seriously lagging flows. Besides, Fig. 7(c) shows that moderately lagging flows will be starved for longer time when Max-lag is used.

From this experiment, we conclude that CIF-Q addresses the fairness issues purely based on weights to dispatch CSs. So it may incur higher queuing delays for seriously lagging flows. The Max-lag scheme can alleviate the queuing delays of seriously lagging flows, but it violates the fairness principle and may starve other lagging flows when compensating the former. The proposed TD-FQ not only provides fairness in dispatching CSs, but also alleviates the queuing delays of seriously lagging flows.

6. Conclusions

We have addressed the delay-weight coupling problem that exists in many existing fair-queuing schemes. A new algorithm, TD-FQ, is proposed to solve this problem. By taking traffic types of flows into consideration when scheduling packets, TD-FQ not only alleviates queuing delay of real-time flows, but also guarantees bounded delays and fairness for all flows. We have derived analytically the fairness properties and delay bounds of TD-FQ. Simulation results have also shown that TD-FQ incurs less packet dropping and queuing delay for real-time flows when compared to CIF-Q.

Acknowledgements

Y. C. Tseng's research is co-sponsored by the NSC Program for Promoting Academic Excellence of Universities, by Computer and Communications Research Labs., ITRI, Taiwan, by Intel Inc., by the Institute for Information Industry and MOEA, R.O.C, under the Handheld Device Embedded System Software Technology Development Project, and by Chung-Shan Institute of Science and Technology under contract number BC93B12P.

References

- [1] A.K. Parekh, R.G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the single-node case, *IEEE/ACM Transactions on Networking* 1 (1993) 344–357.
- [2] A. Demers, S. Keshav, S. Shenker, Analysis and simulation of a fair queueing algorithm, *Journal of Internetworking Research and Experience* 1 (1990) 3–26.
- [3] P. Goyal, H.M. Vin, H. Chen, Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks, *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* 1996; 157–168.
- [4] S.J. Golestani, A self-clocked fair queueing scheme for broadband applications, *INFOCOM* 1994; 12–16.
- [5] J.C.R. Bennett, H. Zhang, WF²Q: worst-case fair weighted fair queueing, *INFOCOM* 1996; 120–128.
- [6] J.C.R. Bennett, H. Zhang, Hierarchical packet fair queueing algorithms, *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* 1996; 143–156.
- [7] V. Bharghavan, S. Lu, T. Nandagopal, Fair queuing in wireless networks: issues and approaches, *IEEE Personal Communications* 6 (1999) 44–53.
- [8] Y. Cao, V.O.K. Li, Scheduling algorithms in broadband wireless networks, *IEEE Proceedings of the IEEE* 89 (2001) 76–87.
- [9] S. Lu, V. Bharghavan, R. Srikant, Fair scheduling in wireless packet networks, *IEEE/ACM Transactions on Networking* 7 (1999) 473–489.
- [10] T.S.E. Ng, I. Stoica, H. Zhang, Packet fair queueing algorithms for wireless networks with location-dependent errors, *INFOCOM* 3 (1998) 1103–1111.
- [11] P. Ramanathan, P. Agrawal, Adapting packet fair queueing algorithms to wireless networks, *ACM/IEEE International Conference on Mobile Computing and Networking* 1998; 1–9.
- [12] D.A. Eckhardt, P. Steenkiste, Effort-limited fair (ELF) scheduling for wireless networks, *INFOCOM* 3 (2000) 1097–1106.
- [13] S. Lu, T. Nandagopal, V. Bharghavan, A wireless fair service algorithm for packet cellular networks, *ACM/IEEE International Conference on Mobile Computing and Networking* 1998; 10–20.
- [14] S. Lee, K. Kim, A. Ahmad, Channel error and handoff compensation scheme for fair queueing algorithms in wireless networks, *IEEE ICC* 5 (2002) 3128–3132.
- [15] Y. Yi, Y. Seok, T. Kwon, Y. Choi, J. Park, W²F²Q packet fair queueing in wireless packet networks, *Proceedings of the Third ACM International Workshop on Wireless Mobile Multimedia* 2000; 2–10.