PAPER

Design and Evaluation of a Weighted Sacrificing Fair Queueing Algorithm for Wireless Packet Networks

Sheng-Tzong CHENG†a, Member and Ming-Hung TAO†, Nonmember

SUMMARY  Fair scheduling algorithms have been proposed to tackle the problem of bursty and location-dependent errors in wireless packet networks. Most of those algorithms ensure the fairness property and guarantee the QoS of all sessions in a large-scale cellular network such as GSM or GPRS. In this paper, we propose the Weighted-Sacrificing Fair Queueing (WSFQ) scheduling algorithm for small-area and device-limited wireless networks. WSFQ slows down the growth of queue length in limited-buffer devices, still maintains the properties of fairness, and guarantees the throughputs of the system. Moreover, WSFQ can easily adapt itself to various kinds of traffic load. We also implement a packet-based scheduling algorithm, the Packetized Weighted Sacrificing Fair Queueing (PWSFQ), to approach the WSFQ. WSFQ and PWSFQ are evaluated by comparing with other algorithms by mathematical analysis and simulations.

key words:  fair scheduling, weighted-sacrificing fair queueing, packetized weighted sacrificing fair queueing

1. Introduction

Many Packet Fair Queueing algorithms (PFQ) have been developed for providing fairness and bounded delay access in wireline networks. PFQ algorithms are first proposed in the context of wired networks to approximate the idealized Generalized Processor Sharing (GPS) policy [1], [2]. GPS has been proven to have two important properties:

- It provides end-to-end delay-bounded services to leaky-bucket constrained sessions; and
- It ensures fair allocation of bandwidth among all backlogged sessions regardless of whether or not their traffic is constrained.

While GPS is a fluid model that cannot be implemented, various packet approximation algorithms are designed to provide services that are nearly identical to GPS. However, it is inappropriate to directly apply GPS and the corresponding algorithms to wireless networks because of bursty channel errors and location-dependent channel capacity and errors. Bursty channel errors make the host unable to receive continuous services. Location-dependent channel capacity and errors make other error-free sessions receive more services. Lu et al. [3] and Ng et al. [4] noticed the unfairness problem and presented the Idealized Wireless Fair-Queueing (IWQF) and the Channel-condition Independent Packet Fair Queueing (CIF-Q) solutions respectively. IWFS and CIF-Q tried to solve the problem by compensating the error-prone flows by using the service share of error-free flows. Thus, if the network becomes heavily loaded when a flow is being compensated, the services of error-free flows may be deteriorated. In order to prevent the service degradation of error-free flows during compensation, Jeong et al. [5] presented the Packetized Wireless General Processor Sharing (PWGPS) algorithm which used pre-allocated service shares for compensation.

In most wireless PFQ algorithms, the amount of compensation services from a leading session is proportional to its weight (i.e. the service share). If the traffic is not heavy and is steady, most of the wireless PFQ algorithms work correctly and efficiently. However, in a small-area WLAN, the traffic is heavy and varied, the queue length of each session that has limited buffer size grows rapidly and hence the packet loss occurs frequently, especially for a high weighted session. Thus the adaptability to the traffic load becomes a new requirement of wireless PFQ algorithms. In this paper we develop a Weighted-SacrificingFair Queueing (WSFQ) algorithm. It dynamically adjusts the sacrificing rates of leading sessions according to their weights and the traffic load. In WSFQ, the ability of adaptation to network traffic is observed. It achieves better performance in packet loss rate and queueing delay.

The rest of the paper is organized as follows. In Sect. 2, we introduce WSFQ, examine its fairness, and compare with other PFQ policies by the queue length of high weighted leading sessions. Section 3 implements PWSFQ that approaches WSFQ. Section 4 discusses the simulation results of PWSFQ and CIF-Q.

2. Weighted Sacrificing Fair Queueing

2.1 Model Definitions

In order to tackle the high variance in traffic, a fair scheduling model with Weighted Sacrificing (WS) is proposed for wireless network in this paper. WSFQ performs well in terms of packet loss, and still retains the following properties that are addressed by the well-known CIF-Q:

- Delay bound and throughput guarantees,
- Long-term fairness,
- Short-term fairness, and
- Graceful degradation in quality of services.

In order to measure fairness, Weighted Sacrificing Fair
Queueing (WSFQ) associates to each system a reference error-free system $S_r$, in which every checking points $T_c$, time units, where $T_c$ is the checking period of the WSFQ and assigned a constant weight. The period between the checking points $i-1$ and $i$ is called the steady period $i$ (Fig. 1). At the checking points, a session is leading if it has received more services in $S_r$ than it would have received in $S_r$, lagging if it has received less, and synchronizing if it has received the same amount of services. The reference system of WSFQ satisfies the Start-time Fair Queueing (SFQ) model [6] and is described by referring to the work of Jeong et al. [5]. The details are shown as follows.

Let $F_i(t_0, t)$ be the amount of services for session $i$ in $S_r$ in a time interval $(t_0, t)$. $F_i(t_0, t)$ is given by

$$
\frac{\partial F_i(t_0, t)}{\partial t} = r_i \cdot \frac{\partial V}{\partial t},
$$

(1)

in which $r_i$ is the given weight (as well as the service share in GPS) of session $i$, and $V$ is the virtual time of $S_r$.

Let $F_i(t_0, t)$ be the amount of services for session $i$ in $S_r$ in a time interval $(t_0, t)$. $F_i(t_0, t)$ is given by

$$
\frac{\partial F_i(t_0, t)}{\partial t} = r_i \cdot \frac{\partial V}{\partial t},
$$

(1)

in which $r_i$ is the given weight (as well as the service share in GPS) of session $i$, and $V$ is the virtual time of $S_r$, which is defined as follows.

$$
\frac{\partial V}{\partial t} = \begin{cases} 
\frac{A}{\sum_j r_j} & \text{if } \sum_j r_j \neq 0 \\
0 & \text{otherwise}
\end{cases},
$$

(2)

in which $A$ is the total service rate of the system.

In WSFQ, the lagging services caused by the channel errors are compensated by the weights of all sessions. But it is important that not only the fairness properties need to be satisfied, but also the problems of crowded queue caused by high traffic load to each session need to be overcome, especially for high weighted sessions. Thus, a leading session should postpone by the normalized amount of services according to its weight, compensation services should be distributed to lagging sessions in proportion to the weights of lagging sessions, and the services from suspended sessions should be distributed to all the available sessions in proportion to their weights.

For the reasons described above, WSFQ uses time-varying weight $v(t)$ instead of constant weight $r$ to make services allocation for each session. The time-varying weight $v_i(t)$ of session $i$ is defined as follows.

$$
v_i(t) = \begin{cases} 
    r_i & \text{if } i \in C(t) \\
    r_i \cdot (1 - \Phi_i) & \text{if } i \in N(t) \\
    0 & \text{otherwise}
\end{cases},
$$

(3)

in which $C(t)$ is the set of sessions that are backlogged and

in the lagging state, $N(t)$ is the set of sessions that are backlogged and not in the lagging state (i.e. leading or synchronizing). Let $n = |N(t)|$. Define $\Phi_i$ to be the sacrificing ratio of session $i$. The value of $\Phi_i$ is set according to session $i$’s weight, i.e.

$$
\Phi_i = \begin{cases} 
    \Phi_{\max} & \text{if } \Theta > \Phi_{\max} \\
    0 & \text{if } \Theta < 0 \\
    \Theta & \text{otherwise}
\end{cases},
$$

(4)

in which $\Phi$ is the default ratio of sacrificing services and is set within the range of $0$ to $\Phi_{\max}$, where $\Phi_{\max}$ is the maximum sacrificing ratio. $\Delta$ in (4) is the degree of the influence of the weight. In other words, the larger the value of $\Delta$ is, the more influence $r_i$ has on $\Phi_i$. Therefore WSFQ can be adapted to different traffic density by controlling the value of $\Delta$. For example, if the traffic is heavy, the value of $\Delta$ needs to be increased; if the traffic is low, the value of $\Delta$ needs to be decreased.

With the time-varying weight $v(t)$, the services served for session $i$ in WSFQ in a time interval $(t_0, t)$ can be defined as follows (the state for each session in WSFQ is not changed in $(t_0, t)$).

$$
\frac{\partial W(t_0, t)}{\partial t} = v_i(t) \cdot \frac{\partial V}{\partial t},
$$

(5)

where $V'$ is the virtual time of WSFQ system, and satisfies

$$
\frac{\partial V'}{\partial t} = \begin{cases} 
    \frac{\Delta}{\sum_j v_j(t)} & \text{if } \sum_j v_j(t) \neq 0 \\
    0 & \text{otherwise}
\end{cases},
$$

(6)

Each session in WSFQ is associated with a parameter $delay$ to measure the difference between the services that a session should receive in a referenced error-free network and the services it has received in the WSFQ. If the value of $delay$ is positive, then we say session $i$ is in lagging state. If the value is negative, then session $i$ is in leading state. If the value is exactly zero, then the session is in synchronizing state.

In WSFQ, if a non-lagging session $i$ has a high weight (higher than the average weight of all non-lagging sessions), the value of $(\sum_{j \in N(t)} r_j/n - r_i) \cdot \Delta$ is negative, thus the value of $\Phi_i$ is smaller than the default ratio $\Phi$. On the other hand, if a non-lagging session has a low weight (lower than the average weight of all non-lagging sessions), the value of $\Phi_i$ is larger than the default ratio $\Phi$. We can easily find that the sacrificing ratio of a high weighted non-lagging session is lower than that of a low weighted non-lagging session. Thus a high weighted non-lagging session has better service rate at the low cost of less service rate associated to low weighted sessions. The problem of crowded queue in high weighted sessions is improved with little influence on other sessions. Moreover, since the sacrificing ratio is limited to be smaller than $\Phi_{\max}$, all non-lagging sessions have graceful degradation in quality of services while it is sacrificed for compensation.
2.2 Fairness and Delay Analysis

In this subsection, we investigate the fairness property of WSFQ in Theorem 1 by proving the bounded difference in the services served to arbitrary two sessions. The arbitrary two sessions can be both in non-lagging states or one is in lagging state and the other is not. In Theorem 2 we show that within this fairness property, high weighted sessions have shorter queue length and smaller queueing delay in WSFQ than in other PFQ algorithms. At last in Theorem 3, the growth speed of queue length in WSFQ is proved to be slower than that in other PFQ algorithms.

**Theorem 1:** Let \( t_0 \) and \( t_1 \) be any two instants in the same steady period and \( t_0 < t_1 \). Assume that session \( i \) and session \( j \) are two servable sessions of the WSFQ. \( \forall t \in (t_0, t_1) \), the value of \( \frac{W_i(t_0, t)}{r_i} - \frac{W_j(t_0, t)}{r_j} \) satisfies:

- If \( i \in C(t) \) or \( j \in N(t) \), then
  \[
  \frac{W_i(t_0, t)}{r_i} - \frac{W_j(t_0, t)}{r_j} \leq \Phi_{\text{max}} \cdot (V(t_1) - V(t_0)).
  \]  

- If \( i, j \in N(t) \), then
  \[
  \frac{W_i(t_0, t)}{r_i} - \frac{W_j(t_0, t)}{r_j} \leq |\Delta \cdot (r_i - r_j) \cdot (V(t_1) - V(t_0)).
  \]  

- If \( i, j \in C(t) \), then
  \[
  \frac{W_i(t_0, t)}{r_i} - \frac{W_j(t_0, t)}{r_j} = 0.
  \]

Theorem 1 shows that WSFQ is fair between two sessions if both of them are being compensated. If both of the sessions are in the non-lagging state, or only a session is being compensated, the services difference is bounded in (7) and (8) respectively. The details of the proof are shown in Appendix A.

**Theorem 2:** In WSFQ, if a non-lagging session whose weight is higher than the average weight of all non-lagging sessions, then it has shorter queue length and smaller queueing delay compared to CIF-Q-like queueing policies.

The briefs about the proof of Theorem 2 are described as follows. Let session \( i \) be the high weighted session, \( E_i^w[N] \) be the expected queue length of session \( i \) in WSFQ, and \( E_j^w[N] \) be the expected queue length of session \( j \) in CIF-Q. Then a M/D/1 queueing model is employed to prove that \( E_i^w[N] < E_j^w[N] \). The details of the proof are shown in Appendix B.

**Theorem 3:** In WSFQ, if a non-lagging session whose weight is higher than the average weight of all non-lagging sessions, then it grows more gently in queue length comparing to CIF-Q-like queueing policies while increasing the arrival rate of this session.

This proof follows the proof of Theorem 2. We denote with \( E_i^w[N] \) as the derivative of \( E_i^w[N] \) and \( E_j^w[N] \) as the derivative of \( E_j^w[N] \). After that we prove that \( E_i^w[N] < E_j^w[N] \). The details of the proof are shown in Appendix C.

3. Packized WSFQ Algorithm

In this section, we investigate a packetized weighted sacrificing fair queueing algorithm (PWSFQ) that realized WSFQ by practical packet-by-packet scheduling.

The procedure of the PWSFQ algorithm is described as follows:

- Initially, all sessions are synchronized. But soon some of them change their states to become leading, lagging, or synchronizing due to the channel failures.
- The SFQ scheduler that is implemented by the reference system \( S_r \) schedules the packets from each session and outputs the sequence of the packets. This procedure is shown in Fig. 2.
- With the inputs of the sequence of backlogged sessions and the information of the states of all sessions, PWSFQ perform the fair scheduling. The sequence of backlogged sessions is decided by the sequence of the packets in the previous step.
- The fair scheduling of the PWSFQ includes the following procedures: the procedure of non-lagging sessions, the procedure of compensation, the procedure of extra services, and the procedure of lagging sessions. The details are described in the subsections of 3.1 and 3.2.

In order to account for the service lost or gained by a session due to errors, we associate with PWSFQ a reference error-free system \( S_r \) that has been described in the previous section. PWSFQ determines the states of all sessions according to the service difference of each session between the real system and the reference system. All parameters used in PWSFQ algorithm are summarized in Table 1.

The parameter \( \text{Lead}_i \) represents the amount of services that a leading session \( i \) leads by comparing it to \( S_r \).

This proof follows the proof of Theorem 2. We denote with \( E_i^w[N] \) as the derivative of \( E_i^w[N] \) and \( E_j^w[N] \) as the derivative of \( E_j^w[N] \). After that we prove that \( E_i^w[N] < E_j^w[N] \). The details of the proof are shown in Appendix C.
Table 1: Parameters used in PWSFQ.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_i$</td>
<td>Service count of a leading session $i$.</td>
</tr>
<tr>
<td>$C_i$</td>
<td>Sacrificing testing of a non-lagging session $i$.</td>
</tr>
<tr>
<td>$C_{\text{min}}$</td>
<td>Lower bound of $C_i$.</td>
</tr>
<tr>
<td>$\text{Lead}_i$</td>
<td>Amount of leading services of leading session $i$.</td>
</tr>
<tr>
<td>$\text{Lag}_i$</td>
<td>Amount of lagging services of lagging session $i$.</td>
</tr>
<tr>
<td>$O_i$</td>
<td>The priority of a lagging session $i$.</td>
</tr>
<tr>
<td>$O_{\text{max}}$</td>
<td>Upper bound of $O_i$.</td>
</tr>
<tr>
<td>$L_i$</td>
<td>The packet length of the head-of-line packet in a session $i$.</td>
</tr>
</tbody>
</table>

The parameter $\text{Lag}_i$ represents the amount of services that a lagging session $i$ lags by comparing it to $S_r$. At any time, if the value of $\text{Lead}_i$ becomes a negative integer, session $i$ becomes lagging and the parameter $\text{Lag}_i$ is assigned the value of $|\text{Lead}_i|$. On the other hand, if the value of $\text{Lag}_i$ becomes a negative integer, session $i$ becomes leading and the parameter $\text{Lead}_i$ is assigned the value of $|\text{Lag}_i|$. Parameters $C_i$ and $R_i$ are both configured for leading sessions. The parameter $C_i$ is the checking flag to determine the time when a non-lagging session $i$ is about to do the sacrificing; $R_i$ is a count which keeps the amount of services that the session $i$ has already received. If the value of $R_i$ exceeds $C_i$, the sacrificing process takes place and the value of $R_i$ is reset to be zero. The $C_i$ is configured according to

$$C_i = k_1 + \left( r_1 - \sum_{j \in N(i)} r_j/n \right) \cdot k_2. \tag{10}$$

$C_i$ is bounded by $C_{\text{min}}$, $k_1$ and $k_2$ are constant values that implement the values of $\Delta$ and $\Phi$ in (4).

For a lagging session $i$, we use parameter $O_i$ to indicate its priority to receive compensation services. Let $O_i = \text{Lag}_i \cdot r_i$. The value of $O_i$ is bounded by $O_{\text{max}}$.

3.1 Non-Lagging Sessions

The procedure that tackles the packets of a non-lagging session is shown in Fig. 3. At first the parameter $R_i$ is increased by the value of $L_i$, then the value of $R_i$ is checked to see if it exceeds $C_i$. If it exceeds, $R_i$ is reset to be zero, $\text{Lead}_i$ is subtracted by $L_i$, and the procedure of compensation is performed. If it does not exceed, this packet is allowed to be transmitted. However, if the packet cannot be sent because of the channel errors, PWSFQ subtracts $L_i$ from $\text{Lead}_i$ and the procedure of compensation is performed. Since the value of $C_i$ is bounded by $C_{\text{min}}$, even if a session leads by a large amount of services, the penalty it would pay is bounded. With the upper bound, the property of graceful degradation is observed.

Figure 4 depicts the procedure of compensation. At first the PWSFQ checks the numbers of lagging sessions in the system. If there is at least one lagging session, PWSFQ continues with the current procedure. After that PWSFQ picks the session $k$ with the maximum $O_k$ among all lagging sessions and checks whether the channel state is good enough for this session. If it is good, the head-of-line (HOL) packet of the session is transferred and $\text{Lag}_k$ is subtracted by $L_k$, but if the session $k$ is not servable due to the channel errors, PWSFQ picks the next lagging session that has maximum priority except the session $k$ until there is no lagging session. This procedure terminates and another procedure called “the procedure of extra services” starts if no lagging session passes the check.

The procedure of extra services is almost the same as the procedure of compensation. This time PWSFQ picks the session $m$ whose $\text{Lead}_m$ is the minimum among all available sessions, transfers the packet of session $m$, and adds the value of $L_m$ to $\text{Lead}_m$. Figure 5 depicts the whole procedure of extra services.
3.2 Lagging Sessions

The procedure of lagging sessions is depicted in Fig. 6. PWSFQ only needs to check whether the current session \( i \) is servable. If it is, the PWSFQ transfers the HOL packet of the session \( i \); if it is not, PWSFQ adds \( \text{Lag}_i \) the value of \( L_i \) and starts the procedure of extra services.

### 4. Performance Analysis

In this section, we present the simulation results obtained by the Stochastic Petri Net (SPN) to demonstrate the queueing behaviors of various kinds of sessions thorough PWSFQ and CIF-Q. We discuss two scenarios of the simulation in the following. One is the full-length simulation and the other one is focusing on the period of an error-free circumstance in which all sessions are backlogged.

#### 4.1 Scenario A: Full-Length Simulation

Consider the scenario with three sessions: one text session and two image sessions in the system. The default properties of each session are listed in Table 2. We assign the text session with the weight of 1 and the image sessions with the weight of 2. This implies that the amount of services needs to be served in the image sessions is twice as much as that in the text session. For the purpose of simulation, we assume the channel of the image session B has the probability of 0.3 to have errors and the other channels have no error, and the packet sizes of all packets are identical. The SPN model of each session in this scenario is depicted in Fig. 7. In scenario A, three sessions are considered. Therefore, three SPN models are required. Notice that the three models are identical except the identifiers of the places and transitions. The complete legends of places and transitions in three sessions are given in Table 3. The multiplicity inhibitor arc from \( p_1 \) to \( t_0 \) defines the buffer capacity of a session. The value \( m \) is the maximum numbers of packets that a buffer can restore, if the numbers of packets exceed the value of \( m \), the packet

### Table 2 Properties of three sessions in scenario A.

<table>
<thead>
<tr>
<th>Session Type</th>
<th>Weight</th>
<th>Session Mode</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text session</td>
<td>1</td>
<td>Poisson</td>
<td>None</td>
</tr>
<tr>
<td>Image session A</td>
<td>2</td>
<td>Poisson</td>
<td>None</td>
</tr>
<tr>
<td>Image session B</td>
<td>2</td>
<td>Poisson</td>
<td>30% of channel error</td>
</tr>
</tbody>
</table>

### Table 3 Definitions of places and transitions.

<table>
<thead>
<tr>
<th>Place/Transition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 )</td>
<td>Numbers of packets queued in buffer.</td>
</tr>
<tr>
<td>( p_2 )</td>
<td>Lagging count.</td>
</tr>
<tr>
<td>( p_3 )</td>
<td>Leading count.</td>
</tr>
<tr>
<td>( t_0 )</td>
<td>Arrival of a packet.</td>
</tr>
<tr>
<td>( t_1 )</td>
<td>An occurrence of channel errors.</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>Transmitting a packet.</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>Transmitting a packet by the procedure of extra services.</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>Transmitting a packet by the procedure of compensation.</td>
</tr>
<tr>
<td>( t_5 )</td>
<td>Sacrificing a transmission for compensation.</td>
</tr>
<tr>
<td>( t_6 )</td>
<td>An occurrence of channel errors.</td>
</tr>
</tbody>
</table>
To compare the performance of PWSFQ with that of CIF-Q, we approximate the CIF-Q algorithm by using the same SPN model. The major difference is the firing rates of the transitions $t_2$, $t_3$, and $t_5$. Thus we reassign these firing rates according to the policy of CIF-Q. Moreover, the value of the system parameter $\alpha$ in full version CIF-Q is set to be 0.5 in order to approximate the configurations of the PWSFQ.

We run the SPN to solve the corresponding Markov chain with the minimum precision of $10^{-6}$, and analyze the transient results of the simulation. The performance of queue length and packet loss (for $m=6$) for each session under different packet arrival rates is shown from Figs. 8 to 11. Here the packet loss is caused only by the limited buffer size. The performance of queue length of all sessions in the PWSFQ when the network-adaptive parameter $k_2$ (in (10)) is assigned with the value from 0.1 to 0.7 is shown in Fig. 12.

Our main conclusions from the simulation results are described as follows.

By comparing PWSFQ with approximated CIF-Q by the increasing arrival rate of the image session $A$, the PWSFQ performs better on the queue length and the packet loss than the approximated CIF-Q does. Oppositely, the approximated CIF-Q has shorter queue length and smaller packet-loss rate than the PWSFQ does in the case of text session. However, the difference between PWSFQ and CIFQ of the high weighted sessions (Image $A$) is more obvious than the difference of the low weighted sessions (the Text session). This is because that the traffic circumstance has more influence on high weighted sessions than on low weighted sessions, and PWSFQ has the agency for adapting the heavy traffic, especially for high weighted sessions.

In the case of lagging sessions, the performance of queue length and packet loss in CIF-Q is better than that in PWSFQ. This is because that there is only one low weighted session in the scenario. We will discuss this issue father at the next scenario.

4.2 Scenario B: Simulation of an Error-Free Period

In this scenario, we would like to focus on the period in which no channel error occurs to any session, and to analyze the simulation results about the queuing behaviors of the leading sessions. Seven sessions are considered for simulation: four are leading sessions and the others are lagging sessions. Among these leading sessions, one is an image session and the other three are text sessions. Among lagging sessions, two are image sessions and one is a text session.

The SPN model of each leading session in this scenario is depicted in Fig. 13. In scenario B, four leading sessions...
are considered. Therefore, four SPN models are required. The complete legends of places and transitions in the four leading sessions are given in Table 4. The multiplicity inhibitor arc from $p_1$ to $t_0$ defines the buffer capacity of a session.

We run the SPN of each leading session independently to solve the corresponding Markov chain with the minimum precision of $10^{-6}$, and analyze the transient results of the simulation. Each leading session is assumed to have 20 leading packets. The performance of queue length and packet loss (for $m=12$) for each leading session under different packet arrival rates is shown from Figs. 14 to 17. The total amount of sacrificed services per time unit of all leading sessions with various values of $k_2$ is shown in Fig. 18.

The simulation results shown from Figs. 14 to 17 are similar to the results in the scenario A, but more obviously the difference between high weighted sessions and low weighted sessions is. Figure 18 shows that the compensation services per time unit in PWSFQ increase when the value of $k_2$ becomes bigger.

5. Conclusions

In this paper, we have developed the WSFQ algorithm to improve the performance of queue length and queueing delay of the high weighted sessions exposed to high traffic density. We also implement the packetized fair scheduling algorithm that approaches the WSFQ. The mathematical analysis shows the fairness property and within this property, the
high weighted sessions have better performance in WSFQ than in other PFQ algorithms. The simulation results show that the PWSFQ performs well on the high weighted sessions at low cost of queueing delay of other low weighted leading sessions. Moreover, WSFQ can easily adapt itself to various kinds of traffic load. Thus applying the WSFQ into the device-limited wireless networks is more suitable than applying the others.

References


Appendix A:  Proof of Theorem 1

[Proof] If \( i \in C(t), \ j \in N(t) \), the time-varying weight \( v_i(t) = r_i \) and \( v_j(t) = r_j(1 - \Phi'_j) \). By applying the time-varying weight into (5),

\[
\begin{align*}
W_i(t_0, t) &= r_i \cdot (V(t) - V(t_0)) \\
W_j(t_0, t) &= r_j \cdot (1 - \Phi'_j)(V(t) - V(t_0))
\end{align*}
\]  
(A-1)

Normalizing \( W_i \) and \( W_j \) by the life-time weight \( r_i \) and \( r_j \), we have

\[
\begin{align*}
W_i(t_0, t) &= r_i \cdot (V(t) - V(t_0)) \\
W_j(t_0, t) &= r_j \cdot (1 - \Phi'_j)(V(t) - V(t_0))
\end{align*}
\]  
(A-2)

Subtracting \( W_i(t_0, t)/r_i \) from \( W_j(t_0, t)/r_j \), then we have

\[
\frac{W_i(t_0, t)}{r_i} - \frac{W_j(t_0, t)}{r_j} = \Phi'_j \cdot (V(t) - V(t_0)) = \left( \Phi + \sum_{m \in N(t)} \frac{r_m}{t - r_j} \cdot \Delta \right) \cdot (V(t) - V(t_0)) 
\]  

\[
\leq \Phi_{\text{max}} \cdot (V(t) - V(t_0)) 
\]  

the case of \( i \in N(t), \ j \in C(t) \), we get (7). If \( i, j \in N(t) \), the (A-3) is modified as follows:

\[
\frac{W_i(t_0, t)}{r_i} - \frac{W_j(t_0, t)}{r_j} = \Phi'_j - \Phi'_{j'} \cdot (V(t) - V(t_0)) 
\]  

\[
\leq |r_i - r_j| \cdot \Delta \cdot (V(t_1) - V(t_0)).
\]  

If \( i, j \in C(t) \), there is no difference between \( W_i(t_0, t)/r_i \) and \( W_j(t_0, t)/r_j \), therefore we get (9).

Appendix B:  Proof of Theorem 2

[Proof] Each session is considered to be a single server with individual arrival rate and service rate. All sessions connect to a main server (Fig. A-1). The arrival rate of the main server is equal to the summation of all sessions’ service rates, and the service rate of the main server is equal to the system capacity. The total arrival rates and system capacity are limited to satisfy

\[
\sum_i \lambda_i = \lambda_{\text{total}} \leq A,
\]  

in which \( A \) is the system capacity (total service rate).

In an error-free system, neither compensation nor sacrificing takes place. Thus the service rate of each session depends on its own weight. That is, if the system capacity is presented as \( A \), the service rate of session \( i \) is equal to \( A \cdot r_i / \sum_j r_j \). However, in an error-prone system, the lagging session \( k \) is compensated by increasing its service rate to be more than \( A \cdot r_k / \sum_j r_j \) when it becomes servable, and the non-lagging session \( l \) is sacrificed by decreasing its service rate to be less than \( A \cdot r_l / \sum_j r_j \). Assume that the service rate of the non-lagging session \( l \) is \((A \cdot r_l / \sum_j r_j) - \alpha_l \). The value of \( \alpha_l \) is determined according to the PFQ algorithm employed. In IWFQ and CIF-Q, due to the bound of leading services of each session, the value of \( \alpha_l \) of non-lagging session \( l \) is proportional to its weight. Thus \( \alpha_l \) is presented as follows in the CIF-Q or in the IWFQ.

![Fig. A-1](image-url)  Illustration of the queueing model in fair scheduling policies.
\[ \alpha_l = r_l \cdot \alpha'_l \left( \alpha'_l < A/ \sum r_i \right) . \]  
(A-6)

In the WSFQ, the denotation of \( \alpha_l \) is different from (A-6) and is expressed as follows:

\[ \alpha_l = r_l \cdot (\alpha'_l + \beta_l) , \]  
(A-7)

where \( \beta_l \) is expressed as follows:

\[ \beta_l = \left( \sum_{i \in N(l)} r_i(n - r_i) \right) \cdot \Delta . \]  
(A-8)

If a non-lagging session whose weight is higher than the average weight of all non-lagging sessions (\( r_l > \sum_{i \in N(l)} r_i/n \)), the value of \( \beta_l \) is negative. By comparing \( r_l \cdot \alpha'_l \) with \( r_l \cdot (\alpha'_l + \beta_l) \), we find that \( r_l \cdot (\alpha'_l + \beta_l) < r_l \cdot \alpha'_l \). Thus the following relationship is derived.

\[ A \cdot r_l/ \sum_{i \in N(l)} r_i - r_l \cdot (\alpha'_l + \beta_l) = A \cdot r_l/ \sum_{i \in N(l)} r_i - r_l \cdot \alpha'_l \cdot \Delta . \]  
(A-9)

Let \( \mu^w_l \) be the service rate of session \( l \) in WSFQ and \( \mu^t_l \) be the service rate of session \( l \) in CIF-Q, we have

\[ \mu^w_l > \mu^t_l . \]  
(A-10)

Considering the session \( l \) as a single server with M/D/1 queueing model, its arrival rate is \( \lambda_l \). Let \( \rho^w_l \) and \( \rho^t_l \) be the traffic intensity in WSFQ and in CIF-Q such that \( \rho^w_l = \lambda_l/\mu^w_l \) and \( \rho^t_l = \lambda_l/\mu^t_l \). By applying (A-10), we have

\[ \rho^w_l < \rho^t_l . \]  
(A-11)

Notice that the expected value of queue length for M/D/1 queue is given by

\[ E[N] = \rho + \frac{\rho^2}{2(1 - \rho)} . \]  
(A-12)

Therefore, let \( E^w_l[N] \) be the expected queue length of session \( l \) in WSFQ and \( E^t_l[N] \) be the expected queue length of session \( l \) in CIF-Q. Then \( E^w_l[N] \) and \( E^t_l[N] \) satisfy:

\[ \begin{cases} 
E^w_l[N] = \rho^w_l + \frac{\rho^w_l^2}{2(1 - \rho^w_l)} , \\
E^t_l[N] = \rho^t_l + \frac{\rho^t_l^2}{2(1 - \rho^t_l)} . 
\end{cases} \]  
(A-13)

By applying (A-11) into (A-13), we have

\[ E^w_l[N] < E^t_l[N] . \]  
(A-14)

Thus queue length of session \( l \) (high weighted session) in WSFQ is shorter than that in CIF-Q. According to Little’s Formulas, if the mean of the queue length in WSFQ is smaller than that in CIF-Q, the mean of the queuing delay in WSFQ is also smaller than which in CIF-Q.

**Appendix C: Proof of Theorem 3**

**[Proof]** The derivative of \( E[N] \) is presented as \( E'[N] \), which represents the growth-gradient of the queue length in a queue and is derived as follows.

\[ E'[N] = \frac{dE[N]}{d\rho} = 1 + \frac{\rho}{2} (2 - \rho) (1 - \rho)^2 . \]  
(A-15)

Let \( E^w[N] \) and \( E^t[N] \) be the growth-gradient of the queue length in the session \( l \) by applying WSFQ and CIF-Q respectively. In order to get the relationship between \( E^w[N] \) and \( E^t[N] \), we denote with the derivative of \( E'[N] \) with \( E''[N] \) as follows.

\[ E''[N] = \frac{dE'[N]}{d\rho} = (1 - \rho)^{-1} + (2 - \rho^2)(1 - \rho)^{-3} . \]  
(A-16)

Since the value of \( \rho \) is less than 1 (0 < \( \rho \) < 1), the value of \( E''[N] \) is always positive. It implies that \( E'[N] \) is an incremental function. Since \( E'[N] \) is an incremental function and \( \rho^w_l < \rho^t_l \), we find the relationship that \( E^w[N] < E^t[N] \) by substituting (A-11) to (A-16). Which means WSFQ grows gently in queue length by comparing that in CIF-Q while the arrival rate of the high weighted sessions are increased.

Sheng-Tzong Cheng received the B.S. (1985) and M.S. (1987) in Electrical Engineering form the National Taiwan University, Taipei, Taiwan. He received the M.S. (1993) and Ph.D. (1995) in Computer Science from the University of Maryland, College Park, Md. He was an Assistant Professor of Computer Science and Information Engineering at National Dong Hwa University, Hualien, Taiwan, in 1995, and became an Associate Professor in 1996. He is currently a Professor in the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. His research interests are in design and performance analysis of mobile computing, wireless communications, multimedia, and real-time systems.

Ming-Hung Tao received the B.S. degree from National Cheng Kung University in 2001. He is working toward the Ph.D. degree in the Department of Computer Science and Information Engineering at the National Cheng Kung University. His current research is focus on the QoS issues of wireless networks and quantum information theory.