On Maximum Rate Control of Worst-Case Weighted Fair Queueing

Jeng Farn Lee[1] , Yeali Sun[2] , Meng Chang Chen[1]

[1] Institute of Infromation Science,Academia Sinica

[2] Department of Information Management, National Taiwan University

TR-IIS-02-003

June 2002

Institude of Information Science

Academia Sinica

On Maximum Rate Control of Worst-Case Weighted Fair Queueing

Jeng Farn Lee[1] , Yeali Sun[2] , Meng Chang Chen[1]

[1] Institute of Infromation Science,Academia Sinica

[2] Department of Information Management, National Taiwan University

Abstract—While exisiting weighted fair scheduling schemes guarantee minimum bandwidths for classes/sessions in a shared channel, maximum rate control, which is critical to service providers and carriers for resource management and business strategies, was generally enforced by employing policing mechanisms. The previous approaches use either a concatenation of rate controller and scheduler, or a policer in front of scheduler. The concatenation method uses two sets of queues and management apparartus, and thus incurs overhead. The other method allows bursty traffic to pass through that can violate maximum rate constraint or cause a high packet loss rate. In this paper, we present a new weighted fair scheduling scheme, WF^2Q-M, to simultaneously support maximum rate control and minimum service rate guarantee. WF^2Q-M proposes the virtual clock adjustment method to enforce maximum rate control by distributing the excess bandwidths of maximum rate constrained sessions to other sessions without recalculating the virtual starting and finishing times of regular sessionss. In terms of performance metrics, we prove that WF^2Q-M is theoretically bounded by a fluid reference model. A procedural scheduling implementation of WF^2Q-M is proposed and proof of correctness is given. Finally, we conduct extensive experiments to show the performance of WF^2Q-M is just as we claimed.

INTRODUCTION

Many critical Internet applications have urgent performance requirements in terms of throughput, delay, delay jitter and loss rate, or a combination of these items. Current best-effort service models cannot meet these requirements, as it handles all traffics equally and does not provide performance guarantees. A number of service disciplines have endeavored to provide per-connection or per-queue performance guarantees [1], [6], [8], [12] that also provide minimum performance guarantees. However they do not provide maximum rate constraint. Maximum rate constraint, an important management feature for service providers and many applications, limits the maximum rate that some specific sessions or service classes can have. Some scenarios where the maximum rate constraint may be applied:

1. Control of service rates for private lines or VPNs that restrict customers to the bandwidth specified in their contracts.
2. Restriction of outgoing traffic of specific services to enforce management policies (e.g. a limit of at most 1Mbps of bandwidth for non-business related Web browsing).
3. Stabilization of throughput for streaming media operations in order to prevent overflow of receiving buffers or packet drop.
4. Ban of excess bandwidth sharing for fluctuation sensitive services in a link-sharing environment (e.g. a VoD system would be better not use excess bandwidth in a shared link to admit more users in order to avoid a situation in which excess bandwidth becomes unavailable.)
5. Support of multi-services, multi-user resource limited communication context (such as wireless communication [9]) so that QoS can be provided.

Consequently, many services need disciplines that simultaneously provide minimum performance guarantees and enforce maximum service rate constraints. We propose a new service discipline, WF^2Q-M (Worst-case Fair Weighted Fair Queueing with maximum rate control), which provides link-sharing capability of WF^2Q as well as maximum rate constraint enforcement. Compared to traditional approaches, WF^2Q-M is efficient in terms of buffer space, management complexity and computation cost.

This paper is organized as follows. In section 2, we review weighted fair scheduling schemes and approaches for maximum rate constraint. In section 3, we describe the proposed WF^2Q-M and its corresponding GPS-M model. In section 4, we present system properties of WF^2Q-M, and show the performance of WF^2Q-M through simulation in section 5. The conclusion is given in section 6.

RELATED WORK

1.1 *GPS, WFQ and WF^2Q*

In this section, we review GPS (Generalized Processor Sharing)[12], the popular packet approximation algorithms WFQ (Weighted Fair Queueing) [12] and WF^2Q (Worst-case Fair Weighted Fair Queueing) [1]. GPS is a fluid system in which the traffic is infinitely divisible and all traffic streams can receive service simultaneously. Every session[1] i of through traffic is assigned a positive real number ϕ_i indicating its weight in sharing the channel capacity. Let $W_{i,GPS}(t_1,t_2)$ be the amount of work received by session *i* in the time interval $[t_1,t_2]$, then a GPS server guarantees

$$\frac{W_{i,GPS}(t_1,t_2)}{W_{j,GPS}(t_1,t_2)} \geq \frac{\phi_i}{\phi_j} \quad i,j=1, 2, \ldots, N \tag{1}$$

for any session *i* that is continuously backlogged throughout the interval $[t_1,t_2]$. Every backlogged session *i* is allocated with service rate r_i as below:

$$r_i = (\phi_i \Big/ \sum_{j \in B(t)} \phi_j)C \quad , \tag{2}$$

in which *C* is the link capacity and *B(t)* is the set of backlogged sessions.

Unlike the idealized fluid model, realistic packet systems serve only one session at a time and the transmission unit is a packet. WFQ and WF^{2}Q are two popular packet approximation service disciplines of GPS. Let a_i^k be the arrival time of the k^{th} packet of session *i*, and d_i^k be its departure time under GPS. Both WFQ and WF^{2}Q select the packet with the smallest d_i^k as the next packet to be transmitted. The disciplines differ in that WF^{2}Q only considers the sets of packets that have started receiving service in the corresponding GPS system whereas WFQ does not. Parekh and Gallager [9] showed that WFQ has the following properties:

$$\begin{aligned} d_{i,WFQ}^k - d_{i,GPS}^k &\leq \frac{L_{\max}}{C} \\ W_{i,GPS}(0,t) - W_{i,WFQ}(0,t) &\leq L_{\max} \end{aligned} \tag{3}$$

where $d_{i,WFQ}^k$ and $d_{i,GPS}^k$ are the departure times under WFQ and GPS respectively, and $W_{i,WFQ}(0,t)$ and $W_{i,GPS}(0,t)$ are the total amounts of service received by session *i* at time t from WFQ and GPS respectively, and $L_{\max}$ is the maximum packet length. These two properties show that WFQ maintains a service performance close to GPS. However, WFQ has a problem that it may serve far ahead from GPS system. In [1], Bennett and Zhang showed the property of WF^{2}Q that provides a tight bound for this problem, as shown in the formula.

$$W_{i,WF^2Q}(0,t) - W_{i,GPS}(0,t) \leq (1 - \frac{r_i}{C})L_{i,\max} \tag{4}$$

Both WFQ and WF^{2}Q are implemented based on the system virtual clock $V(t)$, which is the

[1] "Session" used in this paper can be replaced with "class", as in class-based scheduling disciplines, without affecting the correctness of this paper.

normalized amount of service that a backlogged session should receive at time *t* in the corresponding GPS system. $V(t)$ evolves as:

$$V(0) = 0$$
$$V(t_{j-1} + \tau) = V(t_{j-1}) + \frac{\tau}{\sum_{i \in B(t)} \phi_i} \tag{5}$$

Each packet p_i^k (i.e. the k^{th} packet in session *i*) is assigned a virtual starting S_i^k and finishing time F_i^k, defined as:

$$S_i^k = \max\{F_i^{k-1}, V(a_i^k)\}$$
$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i * C} \tag{6}$$

As the virtual clock function is monotonically increasing, its reverse function exists, denoted as $V^{-1}(t)$. Thus, $V^{-1}(S_i^k)$ and $V^{-1}(F_i^k)$ are the real clock times when packet p_i^k starts and finishes service in the corresponding GPS system. While WFQ selects the packet with the earliest virtual finishing time, WF^2Q only considers packets whose virtual starting times are earlier than $V(t)$, and selects from among them the packet p_i^k with the smallest F_i^k.

WF^2Q is an accurate approximation algorithm of GPS, however, it can only guarantee minimum service rate for a session, but can not constrain its maximum service rate. In this paper, we propose a new service discipline called WF^2Q–M that has the bounded-delay and fairness properties in respect to the corresponding fluid model, which is similar to WF^2Q. It additionally provides the maximum service rate constraint.

1.2 Rate-Controlled Service Disciplines

Several non-work conserving disciplines have been proposed, including Jitter Earliest-Due-Date (Jitter-EDD) [11], Stop-and-Go [57], Hierarchical Round Robin (HRR) [11], and Rate-Controlled Static Priority (RCSP) [17] aim to provide delay-jitter bounds, end-to-end delay bound or rate control based on either a time-framing strategy, or a sorted priority queue mechanism. In [18], Zhang and Ferrari showed that the general rate-controlled service discipline could represent all of the disciplines. As shown in Figure 1, the general rate-controlled server has two stages: rate controller and scheduler. The rate controller is responsible for shaping input traffic into desired traffic patterns, assigning an eligible time for each packet, and moving packets to the scheduler when eligible. The scheduler multiplexes eligible packets from all connections and determines the service sequence of the packets.

An important operation issue of the rate-controller is deciding when to move packets to the scheduler. A simplest method is to set a timer for the head packet of each queue so that it introduces overhead, which is not acceptable for high-speed routers. Hardware implementation may help to reduce overhead; even so, this method limits the number of timers, and therefore the number of classes. Additionally, each packet needs one interrupt to get into the scheduler. Most solutions in the literature are based on time-framing, event driven strategies or both. There is a tradeoff between system accuracy and time granularity in the framing strategy. A smaller frame period provides more accurate bandwidth allocation and higher operation cost, and a larger frame period results in the opposite. The event-driven strategy is based on the occurrence of driving events, e.g. packet enqueue or dequeue. As the timing of event occurrence is not predictable, high uncertainty is intrinsic in this approach. Our proposed service discipline alleviates the problem by eliminating the eligible time calculation and checking which greatly reduces overhead.

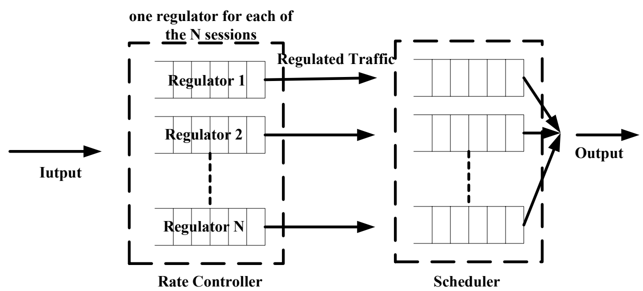

Figure 1. Two-Stage Rate-Control Service Discipline

Another popular rate-controlled model is policer-based rate-control service model, as shown in Figure 2, in which a token bucket or a leaky bucket [10] is used as policer. When an incoming packet obtains enough tokens, it proceeds directly to the scheduler. Otherwise, it is dropped. Note that if a policer is implemented with packet buffer, we consider it is a special case of the two-stage rate-control service discipline. While a token bucket policer can maintain the session's average rate, burst traffic with a rate exceeding the designated maximum rate may be transmitted. If the policer uses a leaky bucket, the maximum rate constraint can be strictly enforced, however bursty packets may be dropped which results in less than expected throughput. Although they do not have the complexity problems of a two-stage rate-control service, their drawbacks prevent them from providing effective maximum rate control.

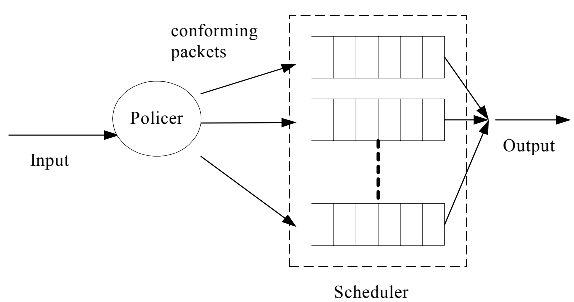

Figure 2. Policer-Based Rate-Control Service Discipline

Here we present simulation results of using NS-2 [7] to show that the policer-based rate-control service discipline is not suitable for maximum rate control. We compare the packet loss rates from our proposed method (WF^{2}Q-M), token bucket, leaky bucket, and a concatenation of token bucket and leaky bucket with different simulation settings. The data source generates UDP traffic based on exponential ON/OFF model. The average times of the ON and OFF period are 312ms and 325ms respectively. The inter-arrival time of packets in the ON period is 2ms, and packet sizes are exponentially distributed with average sizes 1000, 950, and 900 bytes for the three simulations (i.e. the transmission rates of the ON period are 4 Mbps, 3.8 Mbps and 3.6 Mbps respectively). The maximum rate constraint of the session is set as 4Mbps. The token generation rate is 4Mbps for both token bucket and leaky bucket, and the bucket depth of token bucket are 0.05Mb, 0.1Mb, 0.15Mb, 0.2Mb, and 0.25Mb. The buffer size of WF^{2}Q-M is the same as the depth of token bucket in the same simulation.

The *over maximum rate* in the simulation measures the ratio of output rate exceeding the designated maximum rate. Simulation results in Table 1 show that the packet loss rate of WF^{2}Q-M is similar to that of the token bucket scheme with the same simulation settings, but the token bucket scheme suffers from high *over maximum rate* ratio that hampers itself as a good maximum rate constraint server. In addition, the percentage grows much higher when bucket depth becomes slightly larger. For instance, when the ON period rate is 4 Mbps and bucket depth is 0.1 Mbyte, the token bucket policer scheme has a 3.8% over maximum rate ratio. This over maximum rate jumps to 8.8% when bucket depth increases to 0.15 Mbyte. The packet loss rate of the leaky bucket is the same as the concatenation of the token bucket and leaky bucket, since the leaky bucket dominates the process of packet drop. Simulations show that neither the token bucket, nor the leaky bucket is suitable for supporting a maximum rate constraint.

TABLE I PACKET LOSS AND OVER MAXIMUM RATES OF DIFFERENT SERVICE MODELS

	Packet Loss Rate (%)	buffer size (Mbyte)				
		0.05	0.1	0.15	0.2	0.25
ON period rate at 4Mbps	WF^{2}Q-M	7.2	2.9	1.2	0.5	0.2
	Token bucket	7.1	2.8	1.1	0.5	0.2
	Token bucket over maximum rate	0.3	3.8	8.8	11.4	12.9
	Leaky bucket	58.89				
	Token bucket+leaky buckets	58.89				
ON period rate at 3.8Mbps	WF^{2}Q-M	5.2	1.3	0.3	0.07	0.004
	Token bucket	5	1.3	0.3	0.07	0.004
	Token bucket over maximum rate	0.1	2	4.7	5.8	6
	Leaky bucket	55.913				

	Token bucket+leaky buckets	55.913				
ON period rate at 3.6Mbps	WF^{2}Q-M	3.4	0.4	0.04	0	0
	Token bucket	3.2	0.4	0.04	0	0
	Token bucket over maximum rate	0	0.9	1.6	1.7	1.7
	Leaky bucket	52.576				
	Token bucket+leaky buckets	52.576				

WF^{2}Q-M

We propose a new service discipline called WF^{2}Q-M (Worst-case Fair Weighted Fair Queueing with Maximum Rate Control) to enforce the maximum rate constraint. WF^{2}Q-M is more efficient than conventional two-stage rate-control service disciplines in that it uses only one set of queues, and produces more accurate output than the policer-based rate-control service discipline as WF^{2}Q-M strictly enforces maximum rate constraint. Like other weighted fair service disciplines, WF^{2}Q-M users can define a set of sessions and specify a positive real number for each session as its relative link sharing weight on the shared link. In addition, WF^{2}Q-M users can assign maximum rates (also called *peak rates*) for sessions, called *maximum rate constrained* (*MRC*) sessions, as the upper bounds of transmission rates. The MRC sessions that transmit data at their peak rates are *saturated*. Otherwise, the sessions are *non-saturated*. For non-MRC sessions and non-saturated MRC sessions, WF^{2}Q-M allocates bandwidth to the sessions according to their associated weights just like WF^{2}Q. Table 2 lists the notations used in this paper and their descriptions. For the notations associated with packets such as length, time, etc., the superscripts denote packet numbers and the subscripts denote session numbers. The scheduling discipline employed is also signified in the second position of subscript. When without ambiguity, the WF^{2}Q-M subscript may be omitted for simplicity, e.g. d^k_{i,WF^2Q-M} may be represented as d^k_i.

Table II NOTATIONS

a^k_i	Arrival time of the k^{th} packet of session i
$B(t)$	The set of backlogged sessions at time t
$B_p(t)$	The set of saturated sessions at time t
$\overline{B_p}(t)$	The set of non-saturated sessions at time t

C	Link capacity
$d_{i,SD}^{k}$	Departure time of the k^{th} packet of session i under scheduling discipline SD.
e_i^k	Eligible time of the k^{th} packet of session i under the maximum rate constraint
L_i^k	Size of the k^{th} packet of session i
S_i^k	Virtual starting time of the k^{th} packet of session i
SE_i	Session i
F_i^k	Virtual finishing time of the k^{th} packet of session i
$Norm(t)$	Normalization factor at time t.
P_i	The maximum rate of session i
p_i^k	The k^{th} packet of session i
$ratio(t)$	The real clock to virtual clock mapping ratio
$r_i(t)$	The bandwidth of session i at time t under the GPS-M policy
$Q_{i,SD}(t)$	The queue size of session i at time t under scheduling discipline SD.
$W_{i,SD}(0,\tau)$	The work/service received by session i from time 0 to τ under scheduling discipline SD.
ϕ_i	The assigned weight of session i
$\phi_{B(t)}$	The sum of assigned weights of backlogged sessions at time t.
$\phi_{B_p(t)}$	The sum of assigned weights of saturated sessions at time t.

3.1 GPS-M Model

The fluid model GPS-M (Generalized Processor Sharing with Maximum Rate Control) introduced in this section is used as the reference model of WF^2Q-M. GPS-M is an extension of GPS in that every session i is assigned a weight. Sessions may be constrained by their assigned maximum rates P_i, called *maximum rate constrained* (MRC) sessions. If a MRC session receives a higher rate than its assigned maximum rate in the corresponding GPS, we call the session *saturated*, and the set of saturated sessions is denoted as $B_p(t)$. We also denote the set of backlogged sessions not in $B_p(t)$ as $\overline{B_p}(t)$. In GPS-M, the sessions in $B_p(t)$ receive their assigned maximum rates, and sessions in $\overline{B_p}(t)$ share the remaining bandwidth in proportion to their weights as in GPS. The allocated bandwidth of the session is defined as $r_i(t)$:

$$r_i(t) = \begin{cases} P_i & \text{if } SE_i \in B_p(t) \\ \phi_i * Norm(t) & \text{otherwise} \end{cases}$$

$$\text{where } Norm(t) = \frac{(C - \sum_{k \in B_p(t)} P_k)}{\phi_{B(t)} - \phi_{B_p(t)}}. \tag{7}$$

$\phi_{B(t)}$ is the sum of assigned weight of backlogged sessions at time t, and $\phi_{B_p(t)}$ is the sum of assigned weights of sessions in $B_p(t)$ at time t. GPS-M is the same as GPS in many ways; it assumes that the server can serve all backlogged sessions simultaneously, and that the service is infinitely divisible. The difference between GPS and GPS-M is that GPS serves session i with rate $(\phi_i / \phi_{B(t)}) \times C$, while GPS-M serves with $r_i(t)$. In other words, a GPS-M with no sessions in $B_p(t)$ is equivalent to GPS. When all the backlogged sessions are saturated and the sum of the maximum rates of the sessions in $B_p(t)$ is less than the link capacity, GPS-M becomes non-work conserving.

Formula (7) of bandwidth allocation is a declarative definition. To calculate allocated bandwidth, we need a procedural algorithm to distribute the excess bandwidths of saturated sessions (i.e. $C * \frac{\phi_i}{\phi_{B(t)}} - P_i$) to non-saturated sessions. The algorithm that finds $B_p(t)$ and its proof are presented in Appendix A.

3.2 *Virtual Clock Adjustment*

Here we propose a mechanism called *virtual clock adjustment* that distributes the excess bandwidth from sessions in $B_p(t)$ to the sessions in $\overline{B_p}(t)$ in proportion to their assigned weights without recalculating the virtual starting and finishing times of packets of sessions in $\overline{B_p}(t)$. The following example shows how virtual clock adjustment works. Assume that there are four sessions sharing the same link. For simplicity, all packets are of size 1, and link speed is 1. Let the weights of four sessions be 50%, 25%, 12.5% and 12.5%. Assume session 1 is inactive, and each of other sessions receives one packet at the beginning of every second. In Figure 3, a rectangle represents a packet with virtual starting and finishing times in WF^{2}Q. The service order ($p_2^1, p_3^1, p_2^2, p_4^1, p_2^3, p_3^2, p_2^4, p_4^2, \ldots$) is shown in Figure 4. The transmission rates of the sessions 2, 3, 4 are 0.5, 0.25 and 0.25, respectively. Now let session 2 become an MRC session with a maximum rate of 0.4. Excess bandwidth 0.1 is distributed to sessions 3 and 4 evenly. The resulting bandwidth of sessions 2, 3 and 4 are 0.4, 0.3 and 0.3 respectively.

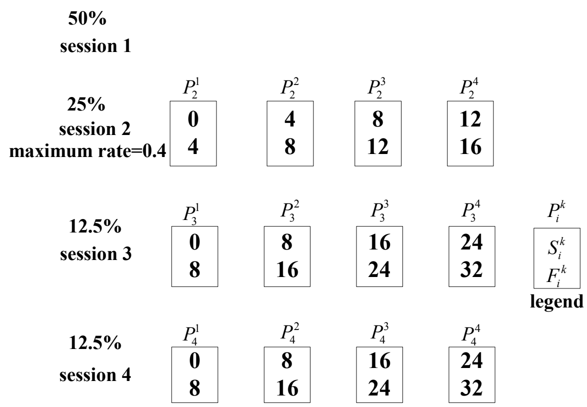

Figure 3. Virtual starting and virtual finishing times of packets in WF^{2}Q

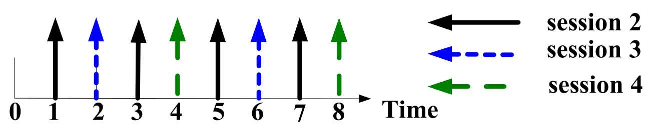

Figure 4. WF^{2}Q service order of Figure 3

For a virtual time based scheduler, when the virtual times of packets are modified, the scheduling sequence is changed, and thus, the allocated bandwidths of sessions are changed. A simplest approach of time adjustment is to recalculate the virtual starting and finishing times of every packet in the system when backlog changes. This method is infeasible due to high overhead. The proposed virtual clock adjustment method aims to alleviate high computational complexity by adjusting the ticking rate of the virtual clock. The virtual times of packets of the saturated sessions in $B_p(t)$ are adjusted so that saturated sessions are constrained by their peak rates. In other words, the virtual times of sessions in $\overline{B_p}(t)$ are relatively ahead to gain higher transmission rate. The virtual clock $V(t)$ is :

$$V(0) = 0$$

$$V(t+\tau) = V(t) + \frac{\tau}{ratio(t)} \tag{8}$$

Assume that the backlog of server does not change during the time period $(t, t+\tau)$.

The real clock to virtual clock mapping ratio is $\phi_{B(t)} * \dfrac{\dfrac{\phi_{B(t)} - \phi_{B_P(t)}}{\phi_{B(t)}} * C}{C - \sum_{k \in B_p(t)} P_k}$ where $\phi_{B(t)}$ is the original ratio, the numerator is the original total bandwidth shared by $\overline{B_p}(t)$ sessions, and the denominator is the total bandwidth shared by $\overline{B_p}(t)$ sessions after receiving excess bandwidth from $B_p(t)$ sessions. For instance, the virtual finish time of p_3^1 is 8, which maps to real clock time 4 in WF^{2}Q, while in WF^{2}Q-M, the real clock is 10/3 (i.e. 8*(0.5*(0.5/0.6))). To summarize, the system virtual clock *V(t)* of WF^{2}Q-M is defined as:

$$ratio(t) = \begin{cases} \dfrac{C}{Norm(t)} & \text{if } \phi_{B(t)} \neq \phi_{B_p(t)}. \\ \phi_{B(t)} & \text{Otherwise} \end{cases} \tag{9}$$

Note that when there are no saturated sessions, $V(t)$ of WF^{2}Q-M is the same as in WF^{2}Q.

3.3 Maximum Rate Control Models

We present a fluid and a packet maximum rate control model and show their correspondence in this section. To enforce maximum rate control for sessions in $B_p(t)$, the eligible time for each packet is introduced and only those packets whose eligible times have been exceeded are considered for receiving service. The eligible time of a packet is defined as $e_i^k = \max(a_i^k, e_i^{k-1} + L_i^{k-1}/P_i)$. With the eligible time constraint, it is obvious that the sessions cannot transmit packets higher than their maximum rates. Incorporating the eligible time, the packet starting time $S_{i,GPS-M}^k$ and finishing time $F_{i,GPS-M}^k$ of the packets of sessions in $B_p(t)$ in GPS-M are defined as:

$$\begin{aligned} e_i^k &= \max(a_i^k, e_i^{k-1} + \frac{L_i^{k-1}}{P_i}) \\ S_{i,GPS-M}^k &= \max\{e_i^k, F_{i,GPS-M}^{k-1}\} \\ F_{i,GPS-M}^k &= S_{i,GPS-M}^k + \frac{L_i^k}{P_i} \end{aligned} \tag{10}$$

To obtain the packet virtual starting time S_i^k and virtual finishing time F_i^k of the $B_p(t)$ packets in WF^{2}Q-M, the ratio function in Formula (9) is applied to map the real clock times to virtual clock times of saturated sessions. The virtual starting and finishing times of a $B_p(t)$ packet, if *ratio(t)* maintains the same value during the transmission of L_i^k in GPS-M, are specified as:

$$
\begin{aligned}
e_i^k &= \max(a_i^k, e_i^{k-1} + \frac{L_i^{k-1}}{P_i}) \\
S_i^k &= \max\{V(e_i^k), F_i^{k-1}\}\} \\
F_i^k &= S_i^k + \frac{L_i^k}{P_i} * \frac{1}{ratio(t)}
\end{aligned} \tag{11}
$$

For $\overline{B_p}(t)$ sessions, packet starting and finishing times in GPS-M, if $r_i(t)$ maintains the same value during the transmission of L_i^k in GPS-M, are defined as:

$$
\begin{aligned}
S_{i,GPS-M}^k &= \max\{a_i^k, F_{i,GPS-M}^{k-1}\}\} \\
F_{i,GPS-M}^k &= S_{i,GPS-M}^k + \frac{L_i^k}{r_i(t)}
\end{aligned} \tag{12}
$$

The virtual starting and finishing times of $\overline{B_p}(t)$ packets in WF^{2}Q-M are defined as:

$$
\begin{aligned}
S_i^k &= \max\{V(a_i^k), F_i^{k-1}\} \\
F_i^k &= S_i^k + \frac{L_i^k}{\phi_i * C}
\end{aligned} \tag{13}
$$

which are the same as WF^{2}Q. Note that F_i^k is unchanged when system backlog changes.

We call a session continuously backlogged in WF^{2}Q-M, if the session is serviced continuously in the corresponding GPS-M. The following theorem shows the correspondence (i.e. $S_i^k = V(S_{i,GPS-M}^k)$ and $F_i^k = V(F_{i,GPS-M}^k)$) between GPS-M and WF^{2}Q-M systems.

Lemma 1. For two times t_1 and t_2 that $t_1 < t_2$, assume backlogged sessions change at time $l_1, l_2, \ldots, l_q$, then $V(t_2)-V(t_1) = \sum_{j=1,q-1} \frac{l_{j+1} - l_j}{ratio(l_j)}$.

Proof : This lemma is a direct result of virtual time function in Formula (8) of WF^{2}Q-M. ■

Since there may be backlogged session changes during the service of a packet, *ratio(t)* and $r_i(t)$ can not maintain constancy. Assume backlog changes at time $l_1, l_2, \ldots, l_q$, and $\hat{L}_j$ is the portion of packet transmitted during (l_j, l_{j+1}). When the session SE_i maintains saturation, the virtual finishing times in Formula (11) can be further revised as:

$$
F_i^k = S_i^k + \sum_{j=1,q} \frac{\hat{L}_j}{P_i} * \frac{1}{ratio(l_j)} \tag{14}
$$

The finish time in Formula (12) can be revised as:

$$F_{i,GPS-M}^{k} = S_{i,GPS-M}^{k} + \sum_{j=1,q} \frac{\hat{L}_j}{r_i(l_j)} \tag{15}$$

Note that if during transmission of a packet, the session status changes from saturated to non-saturated, or vice versa, its virtual finishing time can be obtained by using different formulas according to its status at the time.

Lemma 2. For a continuously backlogged session SE_i, $F_i^k - S_i^k = V(F_{i,GPS-M}^k) - V(S_{i,GPS-M}^k)$.

Proof :

We show the lemma is true for the following cases.

Case 1: $SE_i \in \overline{B_p}(t)$.

From Formula (13) and (15), the transmission time of a packet measured in real time is $F_{i,GPS-M}^k - S_{i,GPS-M}^k = \sum_{j=1,q} \frac{\hat{L}_j}{r_i(l_j)}$, and $F_i^k - S_i^k = \frac{L_i^k}{\phi_i * C}$ in virtual time. According to Lemma 1,

$$V(F_{i,GPS-M}^k) - V(S_{i,GPS-M}^k) = \sum_{j=1,q-1} \frac{l_{j+1} - l_j}{ratio(l_j)} = \sum_{j=1,q} \frac{\hat{L}_j}{r_i(l_j)} * \frac{1}{ratio(l_j)}.$$

After replacing $r_i(l_j)$ and $ratio(l_j)$ into the above equation, it is reduced to

$$V(F_{i,GPS-M}^k) - V(S_{i,GPS-M}^k) = \sum_{j=1,q} \frac{\hat{L}_j}{\phi_i * C} = \frac{L_i^k}{\phi_i * C}.$$

Case 2: $SE_i \in B_p(t)$.

From Formula (10) and (14), the transmission time of a packet measured in real time is $F_{i,GPS-M}^k - S_{i,GPS-M}^k = \frac{L_i^k}{P_i}$, and $F_i^k - S_i^k = \sum_{j=1,q} \frac{\hat{L}_j}{P_i} * \frac{1}{ratio(l_j)}$ in virtual time. According to Lemma 1,

$$V(F_{i,GPS-M}^k) - V(S_{i,GPS-M}^k) = \sum_{j=1,q} \frac{l_{j+1} - l}{ratio(l_j)} = \sum_{j=1,q} \frac{\hat{L}_j}{P_i} * \frac{1}{ratio(l_j)}.$$

If the status of a session flip-flops between $B_p(t)$ and $\overline{B}_p(t)$ during transmission of a packet, case 1 or case 2 can be applied according to its status, and this lemma still holds.■

Theorem 1. For a continuously backlogged session SE_i, $S_i^k = V(S_{i,GPS-M}^k)$ and $F_i^k = V(F_{i,GPS-M}^k)$, for all backlogged packets p_i^k.

Proof : We prove this theorem by induction.

Step N=1. From the definitions of Formulas (11) and (13), $S_i^1 = V(S_{i,GPS-M}^1)$, and from Lemma 2,

$F_i^1 - S_i^1 = V(F_{i,GPS-M}^1) - V(S_{i,GPS-M}^1)$. Together, it implies $F_i^1 = V(F_{i,GPS-M}^1)$.

Step N=m. Assume when N=m, this theorem is true, i.e. $S_i^m = V(S_{i,GPS-M}^m)$ and $F_i^m = V(F_{i,GPS-M}^m)$.

Step N=m+1. As SE_i is continuously backlogged, $S_{i,GPS-M}^{m+1} = F_{i,GPS-M}^m$ and $S_i^{m+1} = F_i^m$. From Step N=m that $F_i^m = V(F_{i,GPS-M}^m)$, $S_i^{m+1} = V(F_{i,GPS-M}^m)$, which is equivalent to $V(S_{i,GPS-M}^{m+1})$. From Lemma 2 that $F_i^{m+1} - S_i^{m+1} = V(F_{i,GPS-M}^{m+1}) - V(S_{i,GPS-M}^{m+1})$, after removing S_i^{m+1} and $V(S_{i,GPS-M}^{m+1})$, we have $F_i^{m+1} = V(F_{i,GPS-M}^{m+1})$.■

In the previous example, the virtual starting and finishing times of packets in WF^{2}Q-M are shown in Figure 5. Following Formula (11), $F_2^1 = \frac{1}{0.4} / (\frac{0.25}{0.6} * 1) = 6$ in WF^{2}Q-M, while F_2^1 is 4 in WF^{2}Q. The service order in WF^{2}Q-M is ($p_2^1, p_3^1, p_4^1, p_2^2, p_3^2, p_4^2, p_2^3, p_3^3, p_2^4, p_4^3, \ldots$), as shown in Figure 6. As shown in Figure 5 and Figure 6, the service rate of the three sessions is 4:3:3, which conforms to the bandwidth allocation, polices of GPS-M.

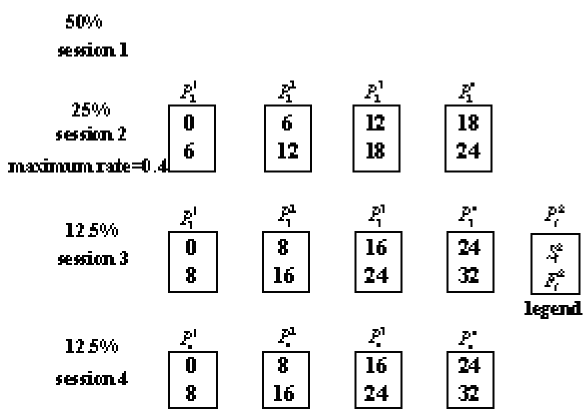

Figure 5. Virtual starting and finishing times of packets in WF^{2}Q-M

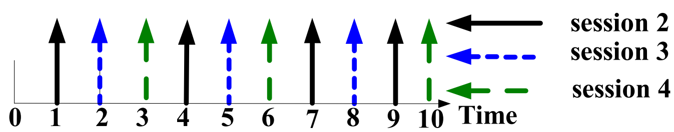

Figure 6. WF^{2}Q-M service order of Figure 5

3.4 *Packet Eligibility*

While the introduction of eligible times of packets allows WF^{2}Q-M to enforce maximum rate constraint, it incurs overhead in maintaining the value of eligible times. In Theorem 2, we show that the eligible time in Formula (11) can be eliminated and Formula (11) can be reduced to the following formula. For $B_p(t)$ packets,

$$S_i^k = \max\{V(a_i^k), F_i^{k-1}\}$$
$$F_i^k = S_i^k + \frac{L_i^k}{P_i} / ratio(t) \tag{16}$$

To prove the theorem, we present the following two lemmas.

Lemma 3. If $a_i^k \le e_i^{k-1} + L_i^{k-1}/P_i$, then $S_i^k = F_i^{k-1}$ in Formula (11) and Formula (16).

Proof. The given condition $a_i^k \le e_i^{k-1} + L_i^{k-1}/P_i$ and Formula (10) imply $e_i^k = e_i^{k-1} + L_i^{k-1}/P_i$. From formula (10), $F_{i,GPS-M}^{k-1} = \max(e_i^{k-1}, F_{i,GPS-M}^{k-1}) + L_i^k/P_i \ge e_i^{k-1} + L_i^{k-1}/P_i$ derives $F_{i,GPS-M}^{k-1} \ge e_i^k$. Since the real clock to virtual clock mapping function is monotonic, $F_{i,GPS-M}^{k-1} \ge e_i^k$ implies $V(F_{i,GPS-M}^{k-1}) > V(e_i^k)$. Since $F_i^{k-1} = V(F_{i,GPS-M}^{k-1})$, then $F_i^{k-1} > V(e_i^k)$ which can be further derived to $S_i^k = F_i^{k-1}$ in Formula (11). For Formula (16), since $F_i^{k-1} > V(e_i^k)$ and $V(e_i^k) > V(a_i^k)$, it implies $F_i^{k-1} > V(a_i^k)$. Thus, $S_i^k = F_i^{k-1}$.■

Lemma 4. If $a_i^k > e_i^{k-1} + L_i^{k-1}/P_i$, then S_i^k has the same value in Formula (11) and Formula (16).

Proof. Given $a_i^k > e_i^{k-1} + L_i^{k-1}/P_i$, two cases are possible:

Case 1: $V(a_i^k) > F_i^{k-1}$. $S_i^k = V(a_i^k)$ in both Formula (11) and Formula (16).

Case 2: $V(a_i^k) \le F_i^{k-1}$. $S_i^k = F_i^{k-1}$ in both Formula (11) and Formula (16).

As a result, S_i^k has the same value in Formula (11) and Formula (16).■

Theorem 2. Formula (11) can be reduced to Formula (16).

Proof. According to Lemma 3, when $a_i^k \le e_i^{k-1} + L_i^{k-1}/P_i$, $S_i^k = F_i^{k-1}$ in Formula (11) and Formula (16). Together with Lemma 4, S_i^k has the same value in Formula (11) and Formula (16) when $a_i^k > e_i^{k-1} + L_i^{k-1}/P_i$. Since F_i^k can be computed from S_i^k, it has the same value in both Formula (11) and Formula (16) providing. Formula (11) can be reduced to Formula (16).■

The elimination of eligible time e_i^k reduces time complexity in calculating virtual times and makes WF^{2}Q-M similar to WF^{2}Q in terms of representation and proofs of properties.

3.5 Packet Processing Algorithms

After presenting the models and their relationships, WF^{2}Q-M packet processing algorithms are introduced along with major functions such as maintaining virtual clock, calculating and adjusting virtual starting and finishing times, and scheduling packet service order. In order to reduce computation cost, the virtual clock advances (i.e. calculates $V(a_i^k)$) only when a packet p_i^k enters the system or the system backlog changes. For the same reason, only when a packet p_i^k reaches the head of the session, S_i^k and F_i^k are calculated according to Formula (13) and (16). If the backlogged sessions change, $r_i(t)$ needs to be re-computed, as well as the virtual starting and finishing times of the head packets of sessions in $B_p(t)$.

Figure 7 shows the algorithm *Enqueue()*, which is activated when a packet arrives. The virtual clock is updated to the time of packet arrival. If the session is empty, the arrival of the packet backlogs the session and the procedure *Time_Adjust()* is used to adjust the virtual starting and finishing times . The packet virtual starting and finishing times of the arriving packet are then calculated. Next, the procedure checks if the timer is set which indicates the system has no eligible packet to service. If the timer is set, it cancels the timer. At the end of the *Enqueue()*, *Dequeue()* is used to select a packet to service. Statement 3.1 can be done using algorithm $findB_p(t)$ shown in Appendix A.

```
procedure Enqueue( p_i^k ) {
 1. Advance virtual clock via calculation of V(a_i^k ) using Formula (8)
 2. Insert the packet at the end of session SE_i
 3. if SE_i is empty {
     3.1 Calculate the r_i(t) for all SE_i ∈ MRC and determine
           the B_p(t)
     3.2 Calculate ratio(t) using Formula (9)
     3.3 Call Time_Adjust()
     3.5 Calculate S_i^k and F_i^k are using Formula (13) or (16)
     3.4 if timer is set, cancel the timer
     }
 4. if (the system if idle)
    4.1 Call Dequeue()
}
```

Figure 7. Algorithm of Packet Enqueue

procedure *Time_Adjust()* {

1. for head packet p_i^k of $SE_i \in B_p(t^-)$

   1.1 if $V(t) < F_i^{k\text{-}1}${

   1.1.1 $S_i^k = V(t) + \frac{(F_i^{k-1} - V(t))*ratio(t^-)}{ratio(t)} * \frac{r_i(t^-)}{r_i(t)}$

   1.1.2 $F_i^k = S_i^k + \frac{L_i^k}{r_i(t)} * \frac{1}{ratio(t)}$

   }

   1.2 else

   1.2.1 $F_i^k = V(t) + \frac{(F_i^k - V(t))*ratio(t^-)}{ratio(t)} * \frac{r_i(t^-)}{r_i(t)}$

}

Figure 8. Algorithm of virtual starting and finishing times recalculation

The algorithm *Time_Adjust()* shown in Figure 8 recalculates the virtual starting and finishing times of head packets of the sessions in $B_p(t)$. (Note that virtual times of sessions in $\overline{B_p}(t)$ do not need to be re-calculated.) Let t^- denote the time immediately before time t when *Time_Adjust()* is called. There are two cases considered in the algorithm. The first case is when $V(t) < F_i^{k\text{-}1}$, the virtual starting time of current packet needs to be adjusted. Otherwise, only the virtual finishing time is adjusted.

procedure *Dequeue*() {

1. Let $idx = \underset{i \in B(t)}{\text{AUGMIN}}\{F_i^h | \; p_i^h = \; Head\_Packet(SE_i) \;\&\; S_i^h \le V(t)\}$

2. if $idx == null$ {

   2.1 Let idx = AUGMIN$\{S_i^h \mid p_i^h = \; Head_Packet(SE_i)\}$

   2.2 $idle\_time = (S_{idx}^h - V(t)) * ratio(t)$

   2.3 $Set\_Timer(idle\_time, Dequeue())$

   2.4 return

   }

3 Serve the packet p_{idx}^h

4. if $SE_{idx} \neq null$

   4.1 Calculate $\mathrm{S_{idx}^h}$ and $\mathrm{F_{idx}^h}$ using Formula (11) or (16)

5. else {

   5.1 Calculate the $r_i(t)$ for all $SE_i \in MRC$
   and determine the $B_p(t)$

   5.2 Calculate *ratio(t)* using Formula (9)

   5.3 Call *Adjust_Time*()

   }

6. Call *Dequeue()*

}

Figure 9. Algorithm of Packet Dequeue

Algorithm *Dequeue()* in Figure 9 is activated when the system is ready to serve the next available packet. Like WF^2Q, WF^2Q-M only considers the set of packets that have started receiving service in the corresponding GPS-M system and selects the packet that will complete service first in GPS-M for service (i.e. with the smallest virtual finishing time). If the system has no eligible packet at time *t*, WF^2Q-M sets a timer (as in Statement 2.2) for the packet with the smallest virtual starting time, and the system becomes idle. The timer expires after *idle_time*, then the procedure calls itself *Dequeue()* and starts another busy period. If, after serving a packet, the session becomes empty which incurs a backlogged sessions change, the system will call *Adjust_Time()* to adjust virtual times. Note that WF^2Q-M needs only one timer and the timer is set only when the system is idle; the overhead of using the timer is almost negligible. When a packet arrives at an idle system, the timer will be cancelled as described in Statement 3.5 of *Enqueue()*.

The computational costs of processing one packet in WF^2Q-M include one call to *Enqueue()* when the packet arrives, and one call to *Dequeue()* when it is scheduled for service. While a packet arrives, the procedure takes constant time to advance the virtual clock and insert the packet to its associated session queue. If the session is empty, *ratio(t)* is executed with time complexity *O(MlogM)*, discussed in Appendix A, which dominates the worst-case time complexity of *Enqueue()*. (Note M is the size of *MRC*.)

The time complexity of *Dequeue()* is dominated by the maintaining a heap data structure to store the virtual starting and finishing times of head packets, and selecting the packet with the smallest virtual finishing time from the heap in Statement 1. Maintaining the set of eligible sessions sorted by virtual finishing time can be accomplished with O(logN) complexity [15]., where N is the number of sessions. *Adjust_Time()* is executed when backlogged sessions change such that its complexity is *O(M)*. In adding up all computational costs, the worst-case time complexity of WF^2Q-M is *O(log(N) + MlogM)*.

WF^2Q-M SYSTEM PROPERTIES

In this section, we discuss the properties of WF^2Q-M including its conformity of bandwidth allocation to GPS-M, delay and work bounds, and work conserving characteristics. In GPS-M, sessions in $B_p(t)$ receive their peak rates and the remaining capacity of the system is shared by sessions in $\overline{B_p}(t)$ in proportion to their assigned weights. Theorem 3 shows that WF^2Q-M achieves the same rate allocation as GPS-M. Delay and work bounds of WF^2Q-M in respect to GPS-M, which are analogous to those of WF^2Q with respect to GPS, are given in Theorem 4. While GPS-M by its rate allocation definition is not always work conserving, Theorem 5 discusses the work conserving property of WF^2Q-M.

Due to the service granularity difference of a packet system and a fluid system, the bandwidth allocation to sessions of WF^2Q-M and GPS-M cannot be equivalent at all time points. Following, we will show the bandwidth allocation conformity of WF^2Q-M to GPS-M under certain condition. Lemma 5 shows the relation of services received by a saturated session and a non-saturated session. Theorem 3 applies the results of Lemma 5 to conclude that WF^2Q-M conforms to GPS-M in bandwidth allocation under the given assumption.

Lemma 5. In WF^2Q-M, if $SE_i \in B_p(t)$ and $SE_j \in \overline{B_p}(t)$ become backlogged at time *0*, and the last serviced packets of both sessions p_i^k and p_j^l have the same virtual finishing time τ, then the proportion of their received services is as follows.

$$W_{i,WF^2Q-M}(0,\tau) : W_{j,WF^2Q-M}(0,\tau) = P_i : \phi_j * Norm(t). \tag{17}$$

Proof. As both sessions are continuously backlogged, the virtual finishing times of p_i^k and p_j^l are $F_i^k = \sum_{m=1,k} (L_i^m / P_i) / ratio(t)$ and $F_j^l = \sum_{m=1,l} L_j^m / (\phi_j * C)$, respectively. Since $F_i^m = F_j^l = \tau$, we have

$$\sum_{m=1,k} L_i^m / (P_i * ratio(t)) = \sum_{m=1,l} L_j^m / (\phi_j * C). \tag{18}$$

As $W_{i,WF^2Q-M}(0,\tau) = \sum_{m=1,k} L_i^m$ and $W_{j,WF^2Q-M}(0,\tau) = \sum_{m=1,k} L_j^m$, we replace them in (18), and obtain

$$W_{i,WF^2Q-M}(0,\tau):W_{j,WF^2Q-M}(0,\tau)=P_i*ratio(t):\phi_j*C. \tag{19}$$

Extending *ratio(t)* using Formula (9), we have

$$W_{i,WF^2Q-M}(0,\tau):W_{j,WF^2Q-M}(0,\tau)=P_i:\phi_j*C\Bigg/\left(\frac{(\phi_{B(t)}-\phi_{B_P(t)})*C}{(C-\sum_{k\in B_p(t)}P_k)}\right) \tag{20}$$

which can be further reduced to

$$W_{i,WF^2Q-M}(0,\tau):W_{j,WF^2Q-M}(0,\tau)=P_i:\phi_j*Norm(t). \quad \blacksquare$$

Theorem 3. In WF^{2}Q-M, if all $SE_k \in B_p(t)$ and $SE_j \in \overline{B_p}(t)$ become backlogged at time *0*, and the last serviced packets of both sessions have the same virtual finishing time τ, then the transmission rate of sessions $SE_k \in B_p(t)$ is P_k and that of $SE_j \in \overline{B_p}(t)$ is $\phi_j*Norm(t)$.

Proof. Let tr_i be the average transmission rate of SE_i from time 0 to $V^{-1}(\tau)$. Then $tr_i = W_{i,WF^2Q-M}(0,\tau)/V^{-1}(\tau)$. As WF^{2}Q-M is work conserving, $\sum_{i\in B(t)} tr_i = C$. Derived from Lemma 5, we have

$$tr_k:tr_j=P_k:\phi_j*Norm(t), \tag{21}$$

where $SE_k \in B_p(t)$ and $SE_j \in \overline{B_p}(\tau)$. Totaling all tr_i in *B(t)*, we have

$$\sum_{i\in B_P(t)}\alpha P_i+\sum_{j\in\overline{B}_P(t)}\alpha\left(\frac{\phi_j}{\phi_{B(t)}-\phi_{B_P(t)}}\right)(C-\sum_{k\in B_p(t)}P_k)=C \tag{22}$$

where α is the constant.

Since $\sum_{j\in\overline{B}_P(t)}\alpha\left(\frac{\phi_j}{\phi_{B(t)}-\phi_{B_P(t)}}\right)=1$, Formula (22) can be reduced to

$$\alpha(\sum_{k\in B_p(t)}P_k+1*(C-\sum_{k\in B_p(t)}P_k)=C$$
$$=>\alpha C=C$$
$$=>\alpha=1$$

Therefore, $tr_k=P_k$ if $SE_k\in B_p(t)$ and $tr_j=\phi_j*Norm(t)$ if $SE_j\in\overline{B}_p(t)$. $\blacksquare$

The above theorem shows that the resource allocation of WF^{2}Q-M conforms to GPS-M service discipline (i.e. coincide with Formula (9)) under the given condition. However, the condition is too strict to be applied for general cases. Theorem 4 shows the delay and work bounds of WF^{2}Q-M with respect to the corresponding model GPS-M.

Theorem 4. The following inequalities hold for WF^{2}Q-M.

$$d^k_{i,WF^2Q-M} - d^k_{i,GPS-M} \le \frac{L_{\max}}{C} \tag{23}$$

$$W_{i,GPS-M}(0,\tau) - W_{i,WF^2Q-M}(0,\tau) \le L_{\max} \tag{24}$$

$$W_{i,WF^2Q-M}(0,\tau) - W_{i,GPS-M}(0,\tau) \le (1 - \frac{r_i(t)}{C})L_{i,\max} \tag{25}$$

$$d^k_{i,WF^2Q-M} - a^k_i \le \frac{Q_{i,WF^2Q-M}(a^k_i)}{r_i} + \frac{L_{i,\max}}{r_i} - \frac{L_{i,\max}}{C} + \frac{L_{\max}}{C} \tag{26}$$

where $Q_{i,SD}(t)$ is the queue size of session *i*, at time *t* under *SD*.

Proof. Proofs can be found in Appendix B.■

When there exists at least one non-MRC backlogged session or the sum of peak rates of saturated session is higher than link capacity, the system is called *ingress-busy.* If a system is not ingress-busy, the sum of peak rates of saturated sessions is less than the link capacity, and obviously, the system is not work conserving. For an ingress-busy GPS-M system, it is obviously work conserving since its channel capacity is fully utilized during busy periods. The following theorem shows that if a WF^2Q-M system is ingress-busy, it is work conserving, i.e. there is at least one packet whose virtual starting time is less than or equal to the system virtual clock when WF^2Q-M selects the next packet for service, as written in Statement 1 of *dequeue()* in Figure 9.

To prove the work conserving property, we employ the concept of rate-controlled service discipline. As introduced in section 2, a rate-controlled service discipline has two stages: rate controller and scheduler. We consider two rate-controlled service disciplines, R-WFQ-M and R-GPS-M, which have the same rate controllers but different schedulers. The schedulers for R-WFQ-M and R-GPS-M are WFQ-M and GPS-M respectively. WFQ-M has the same scheduling algorithm as WF^2Q-M, but WFQ-M selects packets just with the smallest virtual finishing time. We refer to the embedded scheduler in R-GPS-M as GPS-M*, and the embedded WFQ-M server in R-WFQ-M as WFQ-M*. The eligible time for p^k_i in the rate controller is defined as $e^k_i = S^k_{i,GPS-M}$, where $S^k_{i,GPS-M}$ is the time the packet starts service in the corresponding GPS-M system. To prove the theorem, we first present two lemmas borrowed from [1].

Lemma 6. An R-GPS-M system is equivalent to its corresponding GPS-M system, i.e., for any arrival sequence, the instantaneous service rates for each session at any given time are exactly the same as either service discipline, and $d^k_{i,GPS-M} = d^k_{i,R-GPS-M}$ holds.

Lemma 7. An R-WFQ-M system is equivalent to the corresponding WF^2Q-M system, i.e., for any arrival sequence, packets are serviced in exactly the same order with either service discipline and

$d^k_{i,WF^2Q-M} = d^k_{i,R-WFQ-M}$ holds.

Theorem 5. If WF^2Q-M is ingress-busy, it is work conserving.

Proof. For the same input pattern, we have (1) R-GPS-M and GPS-M* have identical system busy time because GPS-M* is the embedded scheduler in R-GPS-M, (2) GPS-M* and WFQ-M* have identical busy time because WFQ-M* is work-conserving service discipline and GPS-M* is always busy in this situation, and (3) R-WFQ-M and WFQ-M* have identical busy time because WFQ-M* is the embedded scheduling algorithm of R-WFQ-M. It follows that R-GPS-M and R-WFQ-M have identical busy times so that GPS-M and WF^2Q-M have the same busy period according to lemma 6 and lemma 7 when the system is ingress-busy. Hence, WF^2Q-M is work conserving since GPS-M is always busy when the system is ingress-busy. Note this proof is similar to section 3 in [1].■

SIMULATIONS

In this section, we present simulations to illustrate the performance of WF^2Q-M. We implemented WF^2Q, GPS-M and WF^2Q-M modules on the ns-2 [7] simulator. The simulation topology is shown in Figure 10. Each of the senders S1, S2, S3 and S4 generated 5Mbps CBR UDP traffic to the receivers R1, R2, R3 and R4, respectively. A sender and a receiver pair formed a session such that the transmission periods are from 1sec to 10sec, 1sec to 12sec, 1sec to 14sec, and 1sec to 5sec for sessions 1 (S1-R1), 2 (S2-R2), 3 (S3-R3) and 4 (S4-R4), respectively. Packet sizes are uniformly distributed from 100 to 1500 bytes. The assigned weights at the output port of router n2 are 10%, 15%, 25% and 50% for sessions 1, 2, 3 and 4, respectively, while S3-R3 is assigned a maximum rate of 3Mbps.

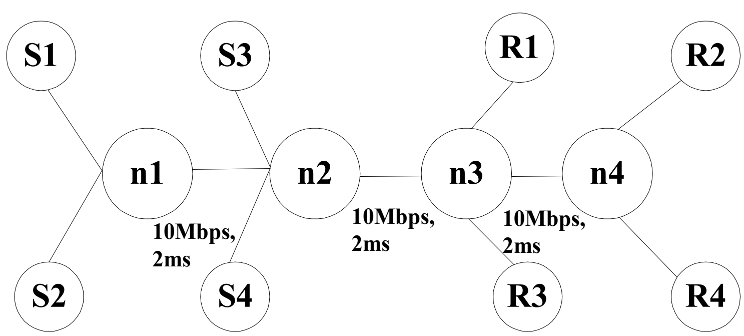

Figure 10. Simulation Topology

We measured the throughput of each session on the bottleneck link between n2 and n3. Figure 11-a shows the results when WF^2Q is used as the scheduling policy of the bottleneck link and Figure 11-b shows the same simulation context using WF^2Q-M. As shown in Figure 11, when there was no session in $B_p(t)$ from 1sec to 5sec, WF^2Q and WF^2Q-M produced same result. After 5sec, since the allocated bandwidth of session 3 was more than 3Mbps in GPS-M, the transmission rate of the session was restricted to 3Mbps in WF^2Q-M, and the excess bandwidth (2 Mbps in this case) was distributed to session 1 and 2 in the ratio of their assigned weights. At time 10sec, S1 stopped transmitting data, and its bandwidth was shared by session 2 and session 3 in WF^2Q, while in WF^2Q-M, because the transmission rate of session 3 was restricted to 3Mbps, the system became non-work conserving.

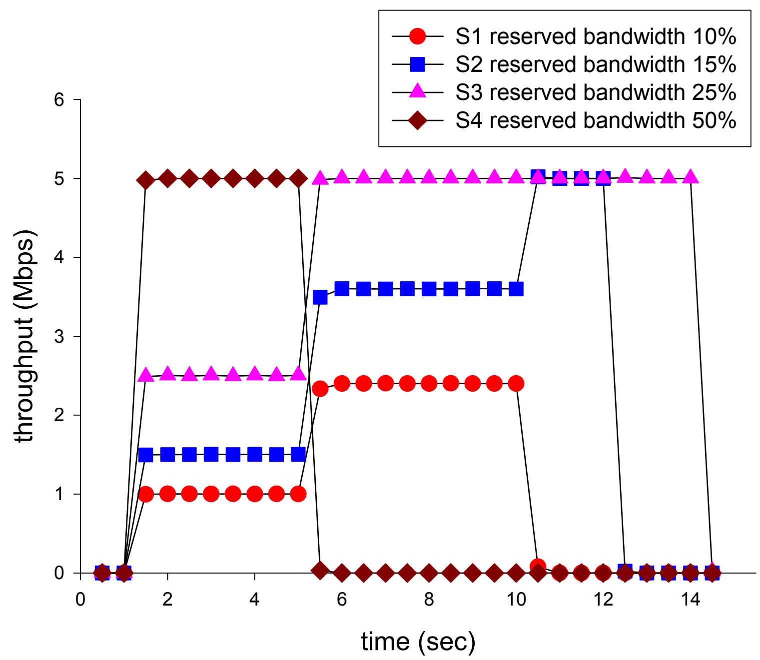

Figure 11-a WF^2Q Scheduling

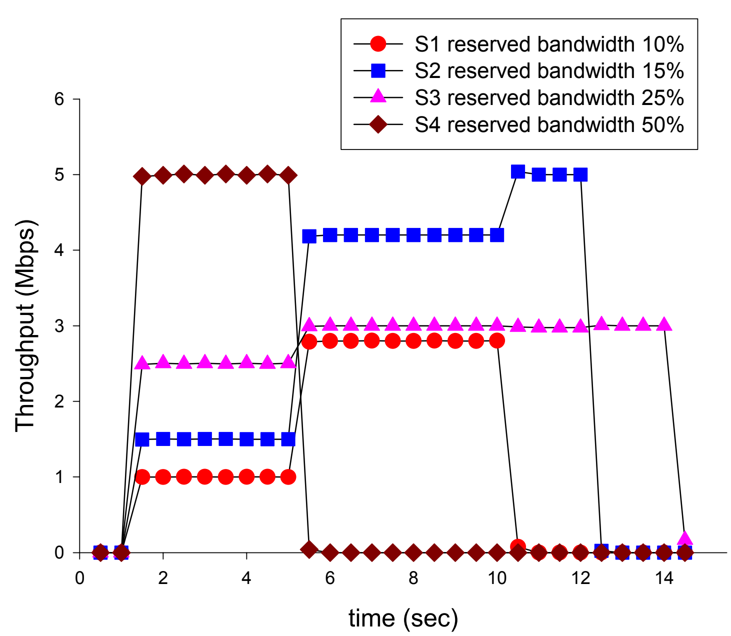

Figure 11-b WF^2Q-M Scheduling

Another measurement shows the relationships between the amount of services received of sessions under WF^{2}Q-M and GPS-M. Let $W_{i,WF^2Q-M}(0,t)$ and $W_{i,GPS-M}(0,t)$ be the amount of service received by session i from time 0 to t under WF^{2}Q-M and GPS-M, respectively. To show the relations between $W_{i,WF^2Q-M}(0,t)$ and $W_{i,GPS-M}(0,t)$, let $W_{i,upper-bound}$ be $W_{i,GPS-M}+(1-\frac{r_i}{C})L_{i,\max}$, and $W_{i,lower-bound}$ be $W_{i,GPS-M}-L_{\max}$. Figure 12 shows the services received near time 5sec. It can be shown that the properties $W_{i,upper-bound}(0,t) \geq W_{i,WF^2Q-M}(0,t) \geq W_{i,lower-bound}(0,t)$ hold for both non-saturated sessions (session 1 and 2 in Figure 12-a and 12-b) and saturated sessions (session 3 in Figure 12-c). In other words, inequalities (23) and (24) of Theorem 4 are verified.

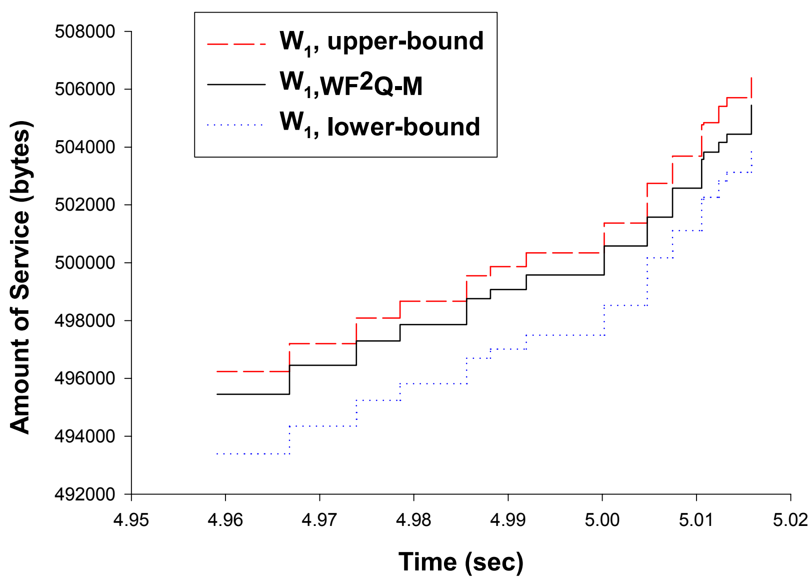

Figure 12-a. Session 1: a $\overline{\mathrm{B}_{\mathrm{p}}}(\tau)$ session.

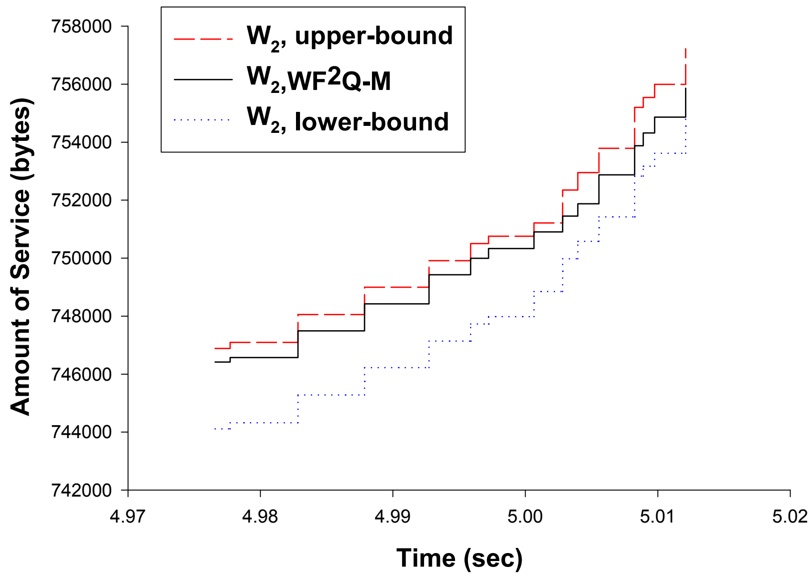

Figure 12-b. Session 2: a $\overline{\mathrm{B}_{\mathrm{p}}}(\tau)$ session.

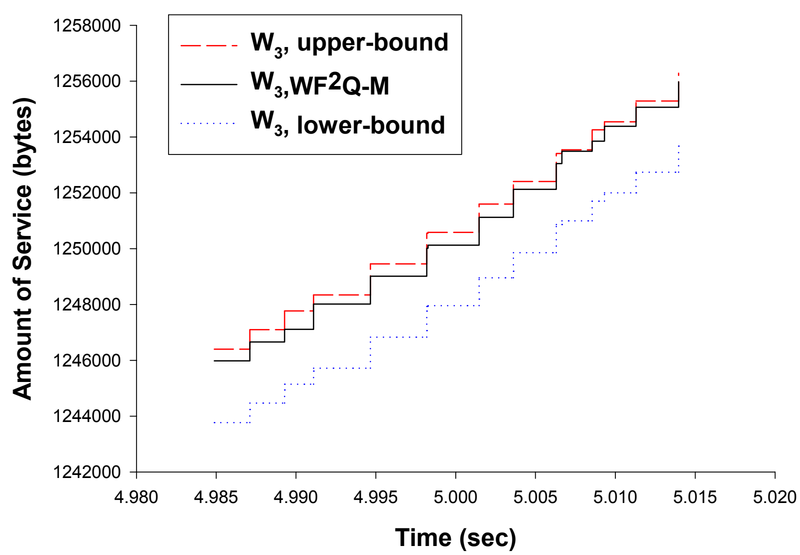

Figure 12-c. session 3: a $\mathrm{B}_{\mathrm{p}}(\tau)$ session.

Figure 12. The relationships between $W_{i,WF^2Q-M}(0,t)$ and $W_{i,GPS-M}(0,t)$

Discussions and Conclusions

In this paper, we propose a new service discipline WF^{2}Q-M to simultaneously provide minimum service rate guarantees and a maximum service rate constraint. We show that packet's eligible time can be merged into its virtual starting time to reduce complexity and make WF^{2}Q-M similar to WF^{2}Q. The virtual clock adjustment allows the sharing of excess bandwidth without recalculating packet virtual starting and finishing times. Additionally, WF^2Q-M takes advantage of the good properties of WF^2Q for efficient, and weighted fair queuing. We also show that a fluid reference model, similar to WF2Q bounded by GPS model, theoretically bound the performance of WF2Q-M.

This paper shows WF^2Q-M can efficiently enforce maximum transmission rate of specific sessions, but not the average rate. In some situations, users want to control the both average and maximum rate. WF^2Q-M service discipline combination with a token bucket policer can achieve the purpose that the architecture is similar to Figure 2. The token bucket controls the average rate of a session, but the bucket size of token bucket allows burst traffic to be transmitted with a rate exceeding the maximum rate of control. By using WF^2Q-M as the packet scheduler, we can control the maximum transmission rate. As a result, the combination of token bucket and WF^2Q-M along with a careful selection of control parameters can support average and maximum transmission rate constraint.

REFERENCES

[1] J. C. R. Bennett and H. Zhang, "WF^2Q: worst-case fair weighted fair queueing," in Proc. IEEE INFOCOM'96, San Francisco, CA, Mar. 1996

[2] J. C. R. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," IEEE/ACM Trans. Networking, vol. 5, pp. 675-689, Oct. 1997.

[3] H. Cormen, E. Leiserson, L. Rivest and Clifford Stein, "Introduction to Algorithms," Second Edition, MIT Inc. 2001.

[4] D. Ferrari and D. Verma. "A scheme for real-time channel establishment in wide-area networks." IEEE Journal on Selected Areas in Communications, 8(3):368-379, April 1990

[5] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," IEEE/ACM Trans. Networking, vol. 3 pp. 365-386, Aug. 1995.

[6] P. Goyal, H.M. Vin, and H. Chen. "Start-time Fair Queuing: A scheduling algorithm for integrated services," In Proceedings of the ACM-SIGCOMM 96, pages 157-168, Palo Alto, CA, August 1996.

[7] S. Golestani. "Congestion-free transmission of real time traffic in packet networks." In Proceedings of IEEE INFOCOM'90, pages 527-542, San Francisco, California, June 1990. IEEE Computer and Communication Societies.

[8] S. Golestani. "A self-clocked fair queueing scheme for broadband applications." In Proceedings of IEEE INFOCOM'94, page 636-646, Toronto, CA, Jane 1994

[9] S. Lu, V. Bharghavan, and R. Srikant. "Fair scheduling in wireless packet networks," IEEE/ACM Trans. On Networking, Vol. 7, No. 4, August 1999.

[10] ns-2 http://www.isi.edu/nsnam/ns/

[11] C. Kalmanek, H. Kanakia, and S. Keshav. "Rate controlled servers for very high-speed networks." In IEEE Global Telecommunications Conference, pages 300.3.1 – 300.3.9, San Diego, California, December 1990.

[12] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case", IEEE/ACM Transactions on Networking, Vol. 1,

No. 3, pp.344-357, June 1993.

[13] J. F. Lee, Y. Sun and M. C. Chen. On maximum rate control of worst-case weighted fair queueing. Technical Report, Institute of Information Science, Academia Sinica, Taiwan ROC, TR-IIS-02-003, 2002

[14] A. S. Tanenbaum, "Computer Networks", 3rd edition, Prentice-Hall, Inc. p380-381, 1996

[15] I. Stoica and H. Abdel-Wahab, "Earliest eligible virtual deadline first: A flexible and accurate mechanism for proportional share resource allocation," Tech. Rep. TR-95-22, Old Dominion Univ., Nov, 1995.

[16] D. Verma, H. Zhang, and D. Ferrari. " Delay Jitter Control for Real-Time Communication in a Packet Switching Network." In Proceedings of Tricomm'91, pages 35-46, Chapel Hill, North Carolina, April 1991

[17] H. Zhang and D. Ferrari. "Rate-controlled static priority queueing." In Proceedings of IEEE INFOCOM'93, pages 227-236, San Francisco, California, April 1993.

[18] H. Zhang and D. Ferrari. "Rate-controlled service disciplines." Journal of High Speed Networks, 3(4):389-412, 1994

[19] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Network", Proc. IEEE, Vol. 83, October 1995, pp. 1374-1396

APPENDIX A: ALGORITHM FINDB$_P$(T)

This appendix present algorithm $findB_p(t)$ to determine $B_p(t)$. Statement 2 performs the first scan to find sessions in $B_p(t)$, and calculate $\sigma's$. Before executing Statement 3, sessions not in $B_p(t)$ are sorted in the descending order of σ_i, i.e. for SE_i and $SE_j \in MRC\text{-}B_p(t)$, $i < j$, if $\sigma_i > \sigma_j$. Statement 3.2 is a termination test such that the algorithm stops if for a session SE_i, the condition $r_i(t) > P_i$ is not satisfied.

Algorithm $findB_p(t)$ (){

1. Set $B_p(t) = null$
2. for $SE_i \in MRC$ {
   - 2.1 $ar_i = (\phi_i / \phi_{B(t)}) * C$
   - 2.2 if $(ar_i > P_i)$ $B_p(t) = B_p(t) \cup SE_i$
   - 2.3 else $\sigma_i = \phi_i / (P_i - ar_i)$

   }
3. for $SE_i \in \{MRC - B_p(t)\}$ in the descending order of σ_i {
   - 3.1 $r_i(t) = (C - \sum_{k \in B_p(t)} P_k) * \phi_i / (\phi_{B(t)} - \phi_{B_p(t)})$
   - 3.2 if $(r_i(t) > P_i)$ $B_p(t) = B_p(t) \cup SE_i$

   }}

Figure A.1. The algorithm finds $B_p(t)$.

Lemma A.1: For two sessions SE_i and $SE_j \in MRC$ and $\sigma_i > \sigma_j$ (as defined in Statement 2.3), if $SE_i \notin B_p(t)$, then $SE_j \notin B_p(t)$ after the execution of *find*$B_p(t)$.

Proof: From the given conditions, SE_i and $SE_j \in MRC$, $\sigma_i > \sigma_j$ and $SE_i \notin B_p(t)$ after the algorithm terminates.

Assume there exists SE_o , j>o>i such that SE_o *is the first session after* SE_i *and* $(C - \sum_{k\in B_p(t)} P_k) * \phi_o /(\phi_{B(t)} - \phi_{B_P(t)}) > P_o$.. Since $\sigma_i > \sigma_j$, by Statement 2.3, $\frac{\phi_i}{P_i - ar_i} > \frac{\phi_j}{P_j - ar_j}$. Note that $(C - \sum_{k\in B_p(t)} P_k) * \phi_i /(\phi_{B(t)} - \phi_{B_P(t)})$ in Statement 3.1 can be rewritten as $ar_i + \frac{\phi_i}{\phi_{B(t)} - \phi_{B_p(t)}} * \sum_{k\in B_p(t)} (ar_k - P_k)$. Let $C' = \sum_{k\in B_p(t)} (ar_k - P_k)$. From given, $SE_i \notin B_p(t)$,

$$\frac{\phi_i}{\phi_{B(t)} - \phi_{B_P(t)}} * C' + ar_i < P_i. \tag{A.1}$$

Since SE_o is the first session after SE_i such that $(C - \sum_{k\in B_p(t)} P_k) * \phi_o /(\phi_{B(t)} - \phi_{B_P(t)}) > P_o$, $B_P(t)$ maintains the same and the following condition holds.

$$\frac{\phi_o}{\phi_{B(t)} - \phi_{B_P(t)}} * C' + ar_o > P_o. \tag{A.2}$$

From (A.1), we have

$$\frac{C'}{\phi_{B(t)} - \phi_{B_P(t)}} < (P_i - ar_i)/\phi_i. \tag{A.3}$$

Similarly, from (A.2), we have

$$\frac{C'}{\phi_{B(t)} - \phi_{B_P(t)}} > (P_o - ar_o)/\phi_o. \tag{A.4}$$

From (A.3) and (A.4),

$$(P_i - ar_i)/\phi_i > (P_o - ar_o)/\phi_o,$$

which can be rewritten as

$$\phi_i /(P_i - ar_i) < \phi_o /(P_o - ar_o). \tag{A.5}$$

Inequality (A.5) violates the assumption that $\sigma_i > \sigma_o$. Therefore there cannot exist such SE_o.

Since no SE_o exists, j>o>i such that $(C-\sum_{k\in B_p(t)} P_k)*\phi_o/(\phi_{B(t)}-\phi_{B_P(t)})>P_o$, SE_i and SE_j have the same $B_p(t)$ when Statement 3.1 is executed. The previous proof shows that $(C-\sum_{k\in B_p(t)} P_k)*\phi_j/(\phi_{B(t)}-\phi_{B_P(t)})$ must less than P_j. Therefore, SE_j cannot be in $B_p(t)$.■

Lemma A.1 implies that not all sessions need to execute Statement 3.1 once there is a session that does not satisfy the condition of Statement 3.2. In order to save computation time, Statement 3.3 as in Figure A.2 can be added and the correctness of $findB_p(t)$ is still maintained.

```
.......
   3.3 else  break
.......
```

Figure A.2. Escape Statement of $findB_p(t)$.

Theorem A.1: Algorithm $findB_p$ correctly determines $B_p(t)$.

Proof: Assume Algorithm $findB_p$ terminates when i=k at Statement 3.3. From Lemma 1, sessions $s_k, s_{k+1},..., s_{|MRC|}$ are not saturated. For sessions $s_1, s_{2,}..., s_{k-1}$, as they pass the condition $r_i(t) > P_i$ in Statement 3.2, the sessions must be saturated by definition. In other case, the algorithm terminates with the all tests at Statement 3.2 being true such that all sessions are in $B_p(t)$. Therefore Algorithm find_B_p correctly finds $B_p(t)$.■

The worst-case time complexity of $findB_p(t)$ is O(*MlogM*)[3], which is dominated by Statement 3 when sorting sessions in the descending order of $\sigma's$.

APPENDIX B: Performance Bounds of WF²Q-M

Lemma B1: Let WFQ-M be the scheduling algorithm the same as WF²Q-M, but WFQ-M selects packets just with the smallest virtual finishing time. Therefore, WFQ-M has the following properties.

$$d^k_{i,WFQ-M} - d^k_{i,GPS-M} \le \frac{L_{\max}}{C} \tag{B.1}$$

$$W_{i,GPS-M}(0,\tau) - W_{i,WFQ-M}(0,\tau) \le L_{\max} \tag{B.2}$$

Proof : The proof is referenced to [12].

We first prove (B.1). Consider any busy period and let the busy period begins at time 0. Let p_k be the k^{th} packet with packet size L^k in the busy period depart under WFQ-M, and its departure times under WFQ-M and GPS-M are d^k_{WFQ-M} and d^k_{GPS-M}. Let m be the largest integer that satisfies both $0 < m \le k-1$, and $d^m_{GPS-M} > d^k_{GPS-M}$. If no such integer m exists, then set $m=0$. Then packet p_m is transmitted before packets $p_{m+1}..., p_k$ under WFQ-M but after all these packets under GPS-M. Since

WFQ-M selects packets with minimum virtual finishing time for transmitting, packets $p_{m+1} \dots, p_k$ must arrive after $d^k_{WFQ-M} - \frac{L_m}{C}$ which is the time that packet p_m begins transmission. Thus

$$\min\{a^{m+1},\dots,a^k\} > d^m_{WFQ-M} - \frac{L^m}{C} \tag{B.3}$$

Since $p_{m+1} \dots, p_{k-1}$ arrive after $d^k_{WFQ-M} - \frac{L_m}{C}$ and depart before p_k does under GPS-M, the following statement holds

$$d^k_{GPS-M} \geq \frac{1}{C}(L^k + L^{k-1} + L^{k-2} + \dots + L^{m+1}) + d^m_{WFQ-M} - \frac{L^m}{C}$$

And under WFQ-M,

$$d^k_{WFQ-M} = d^m_{WFQ-M} + \frac{1}{C}(L^k + L^{k-1} + L^{k-2} + \dots + L^{m+1})$$

Therefore

$$d^k_{GPS-M} \geq d^k_{WFQ-M} - \frac{L^m}{C} \quad => \quad d^k_{WFQ-M} \leq d^k_{GPS-M} + \frac{L\max}{C}$$

Hence

$$d^k_{i,WFQ-M} - d^k_{i,GPS-M} \leq \frac{L_{\max}}{C}$$

if m = 0

$$d^k_{i,WFQ-M} \leq d^k_{i,GPS-M} \tag{B.4}$$

We now prove (B.2): The difference $W_{i,GPS-M} - W_{i,WFQ-M}$ reaches its maximum value when session i packets begin transmission under WFQ-M. Let α be some such time, and L be the packet size of the packing going into service. Hence the packet completes service at time $\alpha + \frac{L}{C}$ under WFQ-M. Assume τ be the time at which the given packet completes service under GPS-M. Then

$$W_{i,GPS-M}(0,\tau) = W_{i,WFQ-M}(0,\alpha + \frac{L}{C}) \tag{B.5}$$

According to (B.1)

$$\tau \geq (\alpha + \frac{L}{C}) - \frac{L_{\max}}{C}$$

$$=> \mathrm{W}_{\mathrm{i,GPS\text{-}M}}(0,\tau) \geq W_{i,GPS-M}(0,\alpha + \frac{L - L_{\max}}{C})$$

$$=> W_{i,GPS-M}(0,\alpha + \frac{L - L_{\max}}{C}) \leq W_{i,WFQ-M}(0,\alpha + \frac{L}{C})$$

$$= W_{i,WFQ}(0,\alpha) + L$$

when $L = L_{\max}$, the difference will reach its maximum value. □

Theorem B1: Let d^k_{i,WF^2Q-M} and $d^k_{i,GPS-M}$ be the departure time under WF^{2}Q-M and GPS-M respectively, and $L_{\max}$ and $L_{i,\max}$ be the maximum packet size of all packets and the session *i*. r_i denotes the minimum guaranteed rate under GPS. WF^{2}Q-M has the following properties.

$$d^k_{i,WF^2Q-M} - d^k_{i,GPS-M} \leq \frac{L_{\max}}{C} \tag{B.6}$$

$$W_{i,GPS-M}(0,\tau)-W_{i,WF^2Q-M}(0,\tau)\le L_{\max} \qquad (B.7)$$

$$W_{i,WF^2Q-M}(0,\tau)-W_{i,GPS-M}(0,\tau)\le (1-\frac{r_i}{C})L_{i,\max} \qquad (B.8)$$

1. we first prove (B.6) and (B.7).

Since a WF^2Q-M system is equivalent to corresponding R-WFQ-M and a GPS-M system is equivalent to the corresponding R-GPS-M system, it suffices to show that

$$d^k_{i,R-WFQ-M}-d^k_{i,R-GPS-M}\le \frac{L_{\max}}{C} \qquad (B.9)$$

$$W_{i,R-GPS-M}(0,\tau)-W_{i,R-WFQ-M}(0,\tau)\le L_{\max} \qquad (B.10)$$

we have

$$d^k_{i,WFQ-M*}=d^k_{i,R-WFQ-M} \qquad (B.11)$$

$$d^k_{i,GPS-M*}=d^k_{i,R-GPS-M} \qquad (B.12)$$

$$W_{i.WFQ-M*}(0,\tau)=W_{i,R-WFQ-M}(0,\tau) \qquad (B.13)$$

$$W_{i.GPS-M*}(0,\tau)=W_{i,R-GPS-M}(0,\tau) \qquad (B.14)$$

Additional, according to (B.4) and (B.5):

$$d^k_{i,R-WFQ-M}-d^k_{i,R-GPS-M}=d^k_{i,WFQ-M*}-d^k_{i,GPS-M*}\le \frac{L_{\max}}{C} \qquad (B.15)$$

$$W_{i,R-GPS-M}(0,\tau)-W_{i,R-WFQ-M}(0,\tau)=W_{i,GPS-M*}-W_{i,WFQ-M*}\le L_{\max} \qquad (B.16)$$

2. Since WF^2Q-M only selects packets which has started service under the corresponding GPS-M system, the following statement must hold

$$W_{i,WF^2Q-M}(0,b^k_{i,GPS-M})\le W_{i,GPS-M}(0,b^k_{i,GPS-M}) \quad \forall i,k \qquad (B.17)$$

Let τ be the time that $b^k_{i,GPS-M}\le\tau<b^{k+1}_{i,GPS-M}$. The maximum service that WF^2Q-M can provides during the interval $[b^k_{i,GPS-M},\tau]$ is limited by both the link capacity and the packet size, thus

$$W_{i,WF^2Q-M}(b^k_{i,GPS-M},\tau)\le \min\{L^k_i, C(\tau-b^k_{i,GPS-M})\} \qquad (B.18)$$

Also, Since GPS-M guarantees a minimum service rate r_i to a backlogged session, we have

$$W_{i,GPS-M}(b^k_{i,GPS-M},\tau)\ge \min\{L^k_i, r_i(\tau-b^k_{i,GPS-M})\} \qquad (B.19)$$

According to (B.18) and (B.19), we have

$$\begin{aligned}&W_{i,WF^2Q-M}(b^k_{i,GPS-M},\tau)-W_{i,GPS-M}(b^k_{i,GPS-M},\tau)\le\\ &\min\{L^k_i, r(\tau-b^k_{i,GPS-M})\}-\min(L^k_i, r_i(\tau-b^k_{i,GPS-M}))\end{aligned} \qquad (B.20)$$

If WF^2Q-M services the packet immediately when it becomes eligible, the maximum difference is reached when WF^2Q-M finishes serving the packet that $L^k_i=r(\tau-b^k_{i,GPS-M})$ or $\tau=b^k_{i,GPS-M}+\frac{L^k_i}{C}$. Plugging in (B.19), we have

$$W_{i,WF^2Q-M}(b^k_{i,GPS-M},\tau)-W_{i,GPS-M}(b^k_{i,GPS-M},\tau)\le(1-\frac{r_i}{C})L^k_i \qquad (B.21)$$

Combining (B.17) and (B.21), we have

$$W_{i,WF^2Q-M}(0,\tau) - W_{i,GPS-M}(0,\tau)$$
$$\leq (1-\frac{r_i}{C})L_i^k \qquad (B.22)\square$$
$$\leq (1-\frac{r_i}{C})L_{i,\max}$$

Corollary B.1: For two corresponding WF^{2}Q-M and GPS-M systems,

$$Q_{i,WF2Q-M}(\tau) - Q_{i,GPS-M}(\tau) \leq L_{\max} \qquad (B.23)$$

$$Q_{i,GPS-M}(\tau) - Q_{i,WF2Q-M}(\tau) \leq (1-\frac{r_i}{C})L_{i,\max} \qquad (B.24)$$

where $Q_{i,s}(\tau)$ is the queue size of session *i* at time τ under the s server.

Theorem 3: WF^{2}Q-M has the following properties:

$$d_{i,WF^2Q-M}^k - a_i^k \leq \frac{Q_{i,WF^2Q-M}(a_i^k)}{r_i} + \frac{L_{i,\max}}{r_i} - \frac{L_{i,\max}}{C} + \frac{L_{\max}}{C} \qquad (B.25)$$

Proof: the theorem follows from (B.6) and (B.25) and the property of GPS-M:

$$d_{i,GPS-M}^k - a_i^k \leq \frac{Q_{i,GPS-M}(a_i^k)}{r_i} \qquad (B.26)$$