WF^2Q : Worst-case Fair Weighted Fair Queueing

Jon C.R. Bennett

FORE Systems

Hui Zhang

School of Computer Science
Carnegie Mellon University

Abstract

The Generalized Processor Sharing (GPS) discipline is proven to have two desirable properties: (a) it can provide an end-to-end bounded-delay service to a session whose traffic is constrained by a leaky bucket; (b) it can ensure fair allocation of bandwidth among all backlogged sessions regardless of whether or not their traffic is constrained. The former property is the basis for supporting guaranteed service traffic while the later property is important for supporting best-effort service traffic. Since GPS uses an idealized fluid model which cannot be realized in the real world, various packet approximation algorithms of GPS have been proposed. Among these, Weighted Fair Queueing (WFQ) also known as Packet Generalized Processor Sharing (PGPS) has been considered to be the best one in terms of accuracy. In particular, it has been proven that the delay bound provided by WFQ is within one packet transmission time of that provided by GPS. In this paper, we will show that, contrary to popular belief, there could be large discrepancies between the services provided by the packet WFQ system and the fluid GPS system. We argue that such a discrepancy will adversely effect many congestion control algorithms that rely on services similar to those provided by GPS. A new packet approximation algorithm of GPS called Worst-case Fair Weighted Fair Queueing (WF^2Q) is proposed. The service provided by WF^2Q is almost identical to that of GPS, differing by no more than one maximum size packet.

1 Introduction

One of the most important issues in designing integrated services networks is the choice of the packet service discipline at queueing points in the network. Recently, disciplines that approximate Generalized Processor Sharing (GPS) have received much attention. GPS is a general form of the head-of-line processor sharing service discipline (PS) [10]. With PS, there is a separate FIFO queue for each session sharing the same link. During any time interval when there are exactly N non-empty queues, the server services the N packets at the head of the queues simultaneously, each at a rate of one N^{th} of the link speed. While a PS server services all non-empty queues at the same rate, GPS allows different sessions to have different service shares and serves the non-empty queues in proportion to the service shares of their corresponding sessions.

There are two independent threads of work that have shown the advantages of the GPS discipline. The first is in the context of designing feedback based congestion control algorithms for use in datagram networks. In most feedback based congestion control algorithms, sources constantly sample the network state using feedback from the receiver or the network, and try to detect symptoms of network congestion. When congestion is detected, sources usually lower the transmission rates to alleviate the congestion. In the case when all sessions share the same FIFO queue at a switch, such a scheme can only work if *all* sessions cooperate. If one session ignores the congestion signal and keeps on sending more data, it can capture an arbitrarily large fraction of the link bandwidth while the performance of other sessions suffer. To address this problem of isolating misbehaving sources, Nagle proposes [12] maintaining a separate FIFO for each session and servicing these queues in round-robin fashion, such that each time a queue is serviced the packet at the head of the queue is transmitted. This scheme provides better protection against misbehaving sources than FCFS, however it favors sessions with larger packet sizes. A misbehaving source can still combine smaller packets into one large packet and send large packets to gain an unfairly large fraction of the bandwidth compared with other sessions. Notice that the PS scheme described above does not suffer from this problem. In PS, all backlogged sessions will receive equal share of bandwidth regardless their packet sizes. However, PS is an ideal discipline where the server can service N sessions simultaneously. In reality, the server has to transmit one packet at a time. Demers et. al. [4] proposed a packet approximation algorithm of PS called Fair Queueing (FQ). They show that Fair Queueing provides fair allocation of bandwidth and offers protection from misbehaving sources. Keshav [9] and Shenker [14] also showed that by having servers approximating PS, sources can measure the network state more accurately. Robust congestion algorithms can be designed based on the more accurate measurement and protection provided by PS like service disciplines.

A separate thread of studying GPS related disciplines is in the context of providing guaranteed bounded delay services in packet-switched networks. Parekh [13] demonstrated that, by employing GPS servers at switches, end-to-end delay bound can be guaranteed to a session provided its traffic is leaky

a_i^k	arrival time of the k^{th} packet on session i
$b_{i,s}^k$	the time the k^{th} packet on session i begins service under the s server
$d_{i,s}^k$	the time the k^{th} packet on session i departs under the s server
$B_s(\tau)$	the set of backlogged sessions at time τ under the s server
$Q_{i,s}(\tau)$	the queue size of session i at time τ under the s server
$W_{i,s}(t_1,t_2)$	the amount of work received by session i during the time interval $[t_1,t_2]$ under the s server
L_i^k	size of the k^{th} packet on session i in number of bits
$L_{i,max}$	maximum packet size of session i
L_{max}	maximum packet size among all sessions
r	link speed
r_i	guaranteed rate for session i

Table 1: Notation Used in this Paper

bucket constrained at the source. He also proposed a packet approximation algorithm for GPS which he called Packet-by-Packet Generalized Processor Sharing or PGPS. It turns out that PGPS is identical to the weighted version of Fair Queueing or WFQ. Parekh has established several important relationships between a fluid GPS system and it's corresponding packet WFQ system:

1. in terms of delay, a packet will finish service in a WFQ system later than in the corresponding GPS system by no more than the transmission time of one maximum size packet;

2. in terms of total number of bits served for each session, a WFQ system does not fall behind a corresponding GPS system by more than one maximum size packet.

The above result can easily be mis-interpreted to say that the packet WFQ discipline and the fluid GPS discipline provide almost identical service except for a difference of one packet. Contrary to this popular (but incorrect) belief, we will demonstrate that there could be *large* discrepancies between the services provided by WFQ and GPS. In fact, what has been proven is that WFQ cannot fall behind GPS by one maximum size packet. However, WFQ can be *far ahead of* GPS in terms of number of bits served for a session. Since many congestion control algorithms [9, 14] were designed with the assumption that WFQ will provide almost identical service with GPS, large discrepancies between the two disciplines and the lack of knowledge that such discrepancies exist will result in unstable and less efficient network control algorithms.

To overcome the limitation of WFQ, we propose a new and better packet approximation algorithm of GPS called Worst-case Fair Weighted Fair Queueing or WF^{2}Q. We show that WF^{2}Q provides almost identical service to GPS with a maximum difference of one packet size, and it shares both the bounded-delay and fairness properties of GPS.

2 GPS and WFQ

In this section, we first define GPS and its most popular packet approximation algorithm WFQ, then describe the important difference between these two disciplines.

A GPS server serving N sessions is characterized by N positive real numbers, $\phi_1, \phi_2, \cdots, \phi_N$. The server operates at a fixed rate r and is work-conserving [1]. Let $W_i(t_1,t_2)$ be the amount of session i traffic served in the interval $[t_1,t_2]$, then a GPS server is defined as one for which

$$\frac{W_i(t_1,t_2)}{W_j(t_1,t_2)} \geq \frac{\phi_i}{\phi_j} \quad j = 1,2,\cdots,N \tag{1}$$

holds for any session i that is backlogged throughout the interval $[t_1,t_2]$ [13]. From the definition, it immediately follows that if $B_{GPS}(\tau)$, the set of backlogged sessions at time τ, remains unchanged during any time interval $[t_1,t_2]$, the service rate of session i during the interval will be exactly

$$r_i^*(t_1,t_2) = \frac{\phi_i}{\sum_{j \in B_{GPS}(t_1)} \phi_j} r \tag{2}$$

where r is the link speed. Since $B_{GPS}(t_1)$ is a subset of all the sessions at the server, it is easy to see that

$$r_i^*(t_1,t_2) \geq r_i \tag{3}$$

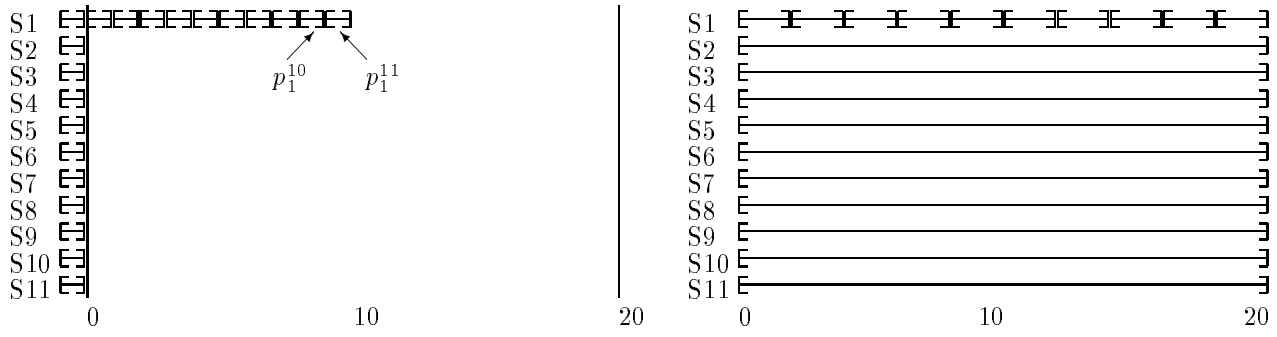
holds where

$$r_i = \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} r \tag{4}$$

Therefore, session i is guaranteed a minimum service rate of r_i during any interval when it is backlogged. Let the time interval length go to zero, we get the instantaneous service rate of the session, $r_i^*(\tau)$.

Notice that GPS is an idealized server that does not transmit packets as entities. It assumes that the server can serve all backlogged sessions simultaneously and that the traffic is infinitely divisible. In a more realistic packet system, only one session can receive service at a time and an entire packet must be served before another packet can be served. There are different ways of emulating GPS service in a packet system. The most popular one is the Weighted Fair Queueing discipline (WFQ) [4], also known as Packet Generalized Processor Sharing or PGPS [13].

[1] A server is work-conserving if it is never idle whenever there are packets to be transmitted. Otherwise, it is non-work-conserving.



(a) Packet Arrivals

(b) GPS Service Order

Figure 1: An Example

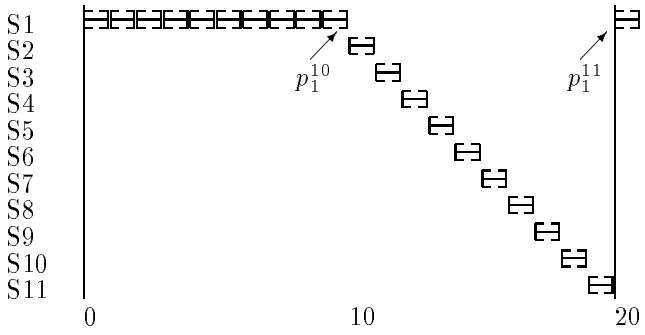


Figure 2: WFQ Service Order

Since WFQ tries to emulate GPS, we need to consider two queueing systems, one using the GPS discipline and one using the packet WFQ discipline.

Definition 1 *Two queueing systems with different service disciplines are called* **corresponding** *systems of each other if they have the same speed, same set of sessions, same arrival pattern, and if applicable, same service share for each session.*

A WFQ system is defined with respect to its corresponding GPS system. Let d^p_{GPS} be the time at which packet p will depart (finish service) under GPS. A good approximation of GPS would be a scheme that serves packet in increasing order of d^p_{GPS}. However this is not always possible without causing the discipline to be non-work-conserving. This is because when the packet system is ready to choose the next packet to transmit, the next packet to depart under GPS *may not have arrived at the packet system yet.* Waiting for it requires the knowledge of the future and also causes the system to be non-work-conserving. In WFQ, when the server is ready to transmit the next packet at time τ, it picks the first packet that would complete service in the corresponding GPS system if no additional packets were to arrive after time τ.

In [13], Parekh establishes the following relationships between the GPS system and its corresponding packet WFQ system:

$$d^k_{i,WFQ} - d^k_{i,GPS} \leq \frac{L_{max}}{r} \quad \forall i,k \tag{5}$$

$$W_{i,GPS}(0,\tau) - W_{i,WFQ}(0,\tau) \leq L_{max} \quad \forall i,\tau \tag{6}$$

where $d^k_{i,WFQ}$ and $d^k_{i,GPS}$ are the times at which the k^{th} packet on session i departs under WFQ and GPS respectively, $W_{i,WFQ}(0,\tau)$ and $W_{i,GPS}(0,\tau)$ are the total amounts of service received by session i (the number of session i bits transmitted) by time τ under WFQ and GPS respectively, and L_{max} is the maximum packet length.

The results given by (5) and (6) can be easily misinterpreted to be that WFQ and GPS provide almost identical service except the difference of one packet. What Parekh has proven is that WFQ cannot fall behind GPS with respect to service given to a session by one maximum size packet. However, packets can leave much *earlier* in a WFQ system than in a GPS system, which means that WFQ can be far *ahead of* GPS in terms of number of bits served for a session. Consider the example illustrated in Figure 1 (a) where there are 11 sessions sharing the same link. The horizontal axis shows the time line and the vertical axis shows the sample path of each session. For simplicity, assume all packets have the same size of 1 and the link speed is 1. Also, let the guaranteed rate for session 1 be 0.5, and the guaranteed rate for each of the other 10 sessions be 0.05.

In the example, session 1 sends 11 back-to-back packets starting at time 0 while each of the other 10 sessions sends only one packet at time 0. If the server is GPS, it will take 2 time units to service a session 1 packet and 20 time units to service a packet from another session. This is illustrated in Figure 1 (b). If the server is WFQ, at time 0, all 11 sessions have packets backlogged. Since packet p_1^1 finishes at time 2 while all other p_i^1 $(i = 2 \ldots 11)$ packets finish at time 20 in the GPS system, WFQ will service p_1^1 first. In fact, the first ten packets on session 1 all have finishing times smaller than packets belonging to any other session, which means that 10 packets on session 1 will

be serviced back to back before packets on other sessions can be transmitted. This is shown in Figure 2. After the burst the next packet on session 1, p_1^{11}, will have a larger finishing time in the GPS system than the 10 packets at the head of other sessions' queues, therefore, it will not be serviced until all the other 10 packets are transmitted, at which time, another 10 packets from session 1 will be serviced back to back. This cycle of bursting 10 packets and going silent for 10 packet times can continue indefinitely. With more sessions, the length of the period between bursting and silence can be larger.

Such oscillation is undesirable for feedback-based congestion control algorithms used in data communication networks. Within the framework of feedback-based congestion control, a data source has to balance between two considerations: on the one hand, it wants to send data to the network as fast as possible, on the other hand, it does not want to send data so fast that causes network congestion. To achieve the best performance, the source needs to detect the amount of bandwidth available to itself and match its sending rate to the available bandwidth. How to estimate bandwidth available for a given source in a dynamic network environment has been the subject of much research [2, 3, 9, 14, 11].

In [9], Keshav proposes an algorithm called Packet-Pair for estimating the available bandwidth for a source. In the Packet-Pair algorithm, the source sends two back-to-back probe packets and the receiver sends an acknowledgement packet immediately upon receiving each packet. The source then uses the spacing between the two acknowledgement packets to calculate an estimate of the bottleneck server rate available to the session. The Packet-Pair algorithm works only if the queueing algorithms at switches achieves fair bandwidth allocation on a fine time granularity. While GPS is the ideal service policy, it cannot be realized. Therefore, Keshav's algorithm assumes that switches implement a packet service algorithm that approximates GPS, such as WFQ or round robin. In the example illustrated in Figure 2, if the Packet Pair algorithm is used, the estimated available rate to session 1 will oscillate between full link speed and zero link speed. This is likely to cause instability of the source control algorithm.

To address the problem of measurement errors by the Packet-Pair algorithm, Bernstein proposes an enhancement of inserting data packets between the packet-pairs [1]. Even with such an enhancement, the measurement error still persists as shown in Figure 3. Depending whether the two probe packets are sent during the burst period or the silent period, the estimate of the server rate may range from r to $\frac{r}{N+1}$ where N is the number of other sessions. In particular, the bound on the measurement error is *not* a strictly decreasing function of the number of packets in the measurement interval. That simply increasing the measurement interval does not always reduce the measurement error may be a significant complication in some instances. The difficulty of determining the appropriate measurement interval in a network of

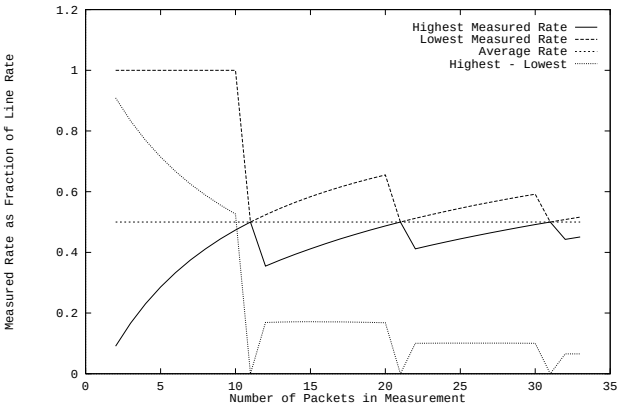


Figure 3: Upper and Lower bounds of measured rate for session 1

WFQ servers may effect not only source based measurement algorithms like Packet-Pair, but also switch based measurement algorithms.

To quantify the discrepancy between the services provided by a packet discipline and the fluid GPS discipline, we consider the notion of worst-case packet fair as defined in [13].

Definition 2 *A service discipline s is called worst-case fair for session i if for any time τ, the delay of a packet arriving at τ is bounded above by $\frac{1}{r_i}Q_{i,s}(\tau) + C_{i,s}$, i.e.,*

$$d_{i,s}^k \quad < \quad a_i^k + \frac{Q_{i,s}(a_i^k)}{r_i} + C_{i,s} \tag{7}$$

where r_i is the throughput guarantee to session i, $Q_{i,s}(a_i^k)$ is the queue size of session i at time a_i^k and $C_{i,s}$ is a constant independent of the queues of the other sessions sharing the multiplexer.

A service discipline is called worst-case fair if it is worst-case fair for all sessions.

We call $C_{i,s}$ the Worst-case Fair Index for session i at server s. Since $C_{i,s}$ is measured in absolute time, it is not suitable for comparing $C_{i,s}$'s of sessions with different r_i's. To perform such a comparison, we define the *Normalized* Worst-case Fair Index for session i at server s to be:

$$c_{i,s} = \frac{r_i C_{i,s}}{r} \tag{8}$$

For a server that is worst-case fair, we define its Normalized Worst-case Fair Index to be:

$$c_s = \max_i \{c_{i,s}\} \tag{9}$$

Notice that GPS is worst-case fair with $c_{GPS} = 0$. In this paper, we use c_s as the metric to quantify the service discrepancy between a packet discipline s and GPS.

We now show that c_{WFQ} may increase linearly as a function of number of sessions N. Consider again

the example shown in Figure 2. The delay between the arrival of packet p_1^{11} at time 10 and its departure (not shown) at time 21 represents a delay of one packet per other session. If there were N other sessions with $\phi_i = \frac{1}{2*N}$, and for simplicity assuming all session are sending packets with maximum length L_{max}, then session 1 will transmit N packets in the interval $(0, \frac{N*L_{max}}{r})$ before any other session receives service. Therefore, despite arriving at an empty queue at time $\frac{N*L_{max}}{r}$, packet p_1^{N+1} will not depart until time $\frac{(2N+1)*L_{max}}{r}$, thus

$$\begin{aligned} C_{1,WFQ} &\geq d_{i,WFQ}^{N+1} - a_{i,WFQ}^{N+1} - \frac{Q(a_{i,WFQ}^{N+1})}{r_i} \\ &= (2N+1)\frac{L_{max}}{r} - N\frac{L_{max}}{r} - \frac{L_{max}}{r/2} \\ &= (N-1)\frac{L_{max}}{r} \end{aligned}$$

we have

$$c_{WFQ} \geq C_{1,WFQ}\frac{r_1}{r} = \frac{N-1}{2}\frac{L_{max}}{r}$$

3 WF^{2}Q

In Section 2, we have shown that the services provided by GPS and WFQ can be quite different. In particular, the worst-case fairness property of WFQ is much weaker than that of GPS. In this section, we define a new and better packet approximation policy of GPS called Worst-case Fair Weighted Fair Queueing or WF^{2}Q, and show that WF^{2}Q shares both the bounded-delay and worst-case fairness properties of GPS.

We want to design a packet system that emulates a fluid GPS system as closely as possible. The difference between a fluid system and a packet system is that, at any given time, there can be multiple packets being serviced simultaneously in a fluid system while there can be only one packet being serviced in the packet system. In fact, in a GPS system, every backlogged session has exactly one packet being serviced and the instantaneous service rate for the packet on any backlogged session i is $\frac{\phi_i}{\sum_{j\in B(\tau)}\phi_j}r$ where $B(\tau)$ is the set of backlogged sessions at time τ. While the the service time of a packet with L bits in a packet system is $\frac{L}{r}$, it can be much longer in the GPS system depending on the guaranteed fraction of bandwidth for the session and the number of backlogged sessions during its service period. Therefore, even though a packet may start service later in a packet system than in the GPS system, it may still finish earlier in the packet system than in the GPS system. If a second packet from the session starts service in the packet system before the first packet finishes service in the GPS system, we run into a situation where the second packet starts earlier in the packet system than the GPS system. When such a situation continues, the difference between a packet system and a fluid system can become quite large. Since in the long run, the service received by a session will be the same under both the GPS system and the packet system, when the packet system receives more service than the GPS system during one time period, it will receive less service in some future period. When the discrepancy between the packet system and the GPS system becomes large, the packet system may run into a situation where it alternates between two states. In one state the session receives much more service than in the GPS system, and in the other state the session receives much less service than in the GPS system. This is the root of the problem illustrated in Figure 2.

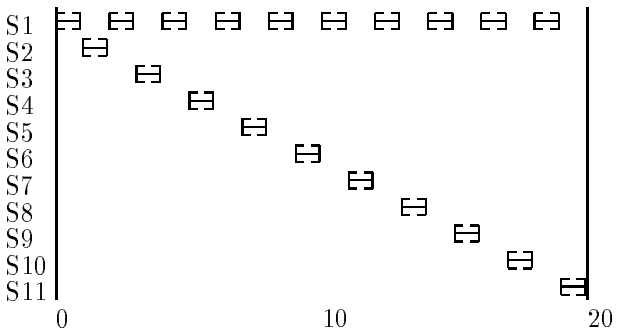


Figure 4: WF^{2}Q Service Order

To minimize the difference between a packet system and the fluid GPS system, we propose a new packet policy called Worst-case Fair Weighted Fair Queueing or WF^{2}Q. Recall that in a WFQ system, when the server chooses the next packet for transmission at time τ, it selects, among all the packets that are backlogged at τ, the first packet that would complete service in the corresponding GPS system. In a WF^{2}Q system, when the next packet is chosen for service at time τ, rather than selecting it from among all the packets at the server as in WFQ, the server only considers the set of packets that have started (and possibly finished) receiving service in the corresponding GPS system at time τ, or formally, $\{p_i^k \mid b_{i,GPS}^k \leq \tau \leq b_{i,WFQ}^k\}$, and selects the packet among them that would complete service first in the corresponding GPS system.

Now consider again the example discussed in Figure 1 but with WF^{2}Q policy. at time 0, all packets at the head of each session's queue, p_i^1, $i = 1, \cdots, 11$, have started service in the GPS system (Figure 1 (a)). Among them, p_1^1 has the smallest finish time in GPS, so it will be served first in WF^{2}Q. At time 1, there are still 11 packets at the head of the queues: p_1^2 and p_i^1, $i = 2, \cdots, 11$. Although p_1^2 has the smallest finish time, it will not start service in the GPS system until time 2, therefore, it won't be eligible for transmission at time 1. The other 10 packets have all started service at time 0 at the GPS system, thus are eligible. Since they all finish at the same time in the GPS system, the tie-breaking rule of giving highest priority to the session with the smallest number will yield p_2^1 as the next packet for service. In contrast, if a WFQ

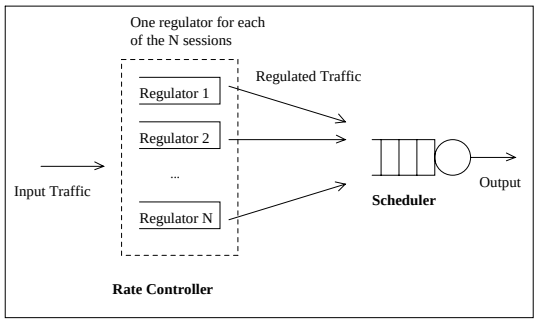


Figure 5: Rate-Controlled Service Discipline

server is used, rather than selecting the next packet from among the 10 packets that have started service in the GPS system, it would pick the packet among all 11 packets, which will result in packet p_1^2 . At time 3, p_1^2 becomes eligible and has the smallest finish time among all backlogged packets, thus it will start service next. The rest of the sample path for the WF^{2}Q system is shown in Figure 4.

Therefore, even in the case when session 1 is sending back-to-back packets, its output from the WF^{2}Q system is rather smooth as opposed to the bursty output under a WFQ system. For sources that measure available link bandwidth in a feedback based flow control environment, more accurate estimates can be obtained in a shorter period of time when the switch uses WF^{2}Q as opposed to WFQ.

The following theorem summarizes some of the most important properties of WF^{2}Q.

Theorem 1 *Given a WF^{2}Q system and a corresponding GPS system, the following properties hold for any i, k, τ:*

$$d^k_{i,WF^2Q} - d^k_{i,GPS} \leq \frac{L_{max}}{r} \quad (10)$$

$$W_{i,GPS}(0,\tau) - W_{i,W^2FQ}(0,\tau) \leq L_{max} \quad (11)$$

$$W_{i,WF^2Q}(0,\tau) - W_{i,GPS}(0,\tau) \leq (1-\frac{r_i}{r})L_{i,max} \quad (12)$$

Before we give the proof, we briefly discuss the implications of the result. First, (10) and (11) correspond respectively to (5) and (6), the results Parekh proves for WFQ. (10) establishes the relationship between the worst-case delay bounds for the fluid GPS server and the packet WF^{2}Q server. In [13], Parekh has shown that the end-to-end network delay can be bounded for a session if (a) the session's traffic is leaky-bucket constrained, (b) GPS servers are used along the path, and (c) the guaranteed rate for the session at each server is no less than the average rate of the session. By applying (5), the equivalence of (10) for WFQ, he has shown that the result also holds if WFQ servers are used along the path. We can easily adopt the same approach by applying (10) and show that the end-to-end-bounded-delay property holds if WF^{2}Q servers are used along the path. A second observation is as follows: (6) and (11) show that from a session's point of view, at any given time, both packet systems of WFQ and WF^{2}Q will not fall far behind the fluid GPS system in terms of bits served. However, as illustrated in the example in Section 2, WFQ system can be quite far *ahead of* the GPS system, which results in large discrepancy between WFQ and GPS. This is not the case with WF^{2}Q. In fact, (12), which holds only for WF^{2}Q but not for WFQ, states that the service provided to a session by a WF^{2}Q system can not be ahead of the corresponding GPS system by more than a fraction of the maximum packet size. Since the service provided by WF^{2}Q can be neither too far behind, nor too far ahead, when compared to that by GPS, it must be that WF^{2}Q provides almost identical service with GPS. Since the maximum difference is less than one packet size, one would expect that WF^{2}Q is an *optimal* packet algorithm in approximating GPS.

To prove the theorem, we consider the following conceptual implementation of WF^{2}Q using a rate-controlled server [17]. As shown in Figure 5, a rate-controlled server has two components: a set of regulators and a scheduler. Packets are held in the regulators until their eligibility time before they are passed to the scheduler. The scheduler only schedules eligible packets. Different policies of assigning eligibility times result in different regulators. Various combinations of regulators and schedulers result in a class of service policies. In this paper, we consider two rate-controlled service disciplines: R-WFQ and R-GPS, which have the same regulators but different schedulers. The schedulers for R-WFQ and R-GPS are WFQ and GPS respectively. Therefore, R-WFQ is a packet algorithm and R-GPS is a fluid algorithm. The eligibility time for the k^{th} packet on session i is defined to be:

$$e^k_i = b^k_{i,GPS} \quad (13)$$

where $b^k_{i,GPS}$ is the time the packet starts service in the corresponding GPS system.

Notice that there are two GPS servers under consideration, the corresponding GPS server that is standalone, and the GPS server that is embedded within the R-GPS server. To distinguish between them, we refer to the embedded one as GPS^*. Likewise, we refer to the embedded WFQ server in R-WFQ as WFQ^*.

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

Lemma 1 *An R-GPS system is equivalent to its corresponding GPS system, i.e., for any arrival sequence, the instantaneous service rates for each connection at any given time are exactly the same with either service discipline, and $d^k_{i,GPS} = d^k_{i,R-GPS}$ holds.*

From the point of view of the GPS^* server, what the regulator does is to delay the arrival of packets at the GPS^* server until the latest time that the packet could arrive and still start at the same time as in the GPS server. Figure 6 depicts the arrival and service of packets belonging to session i in a GPS systems and

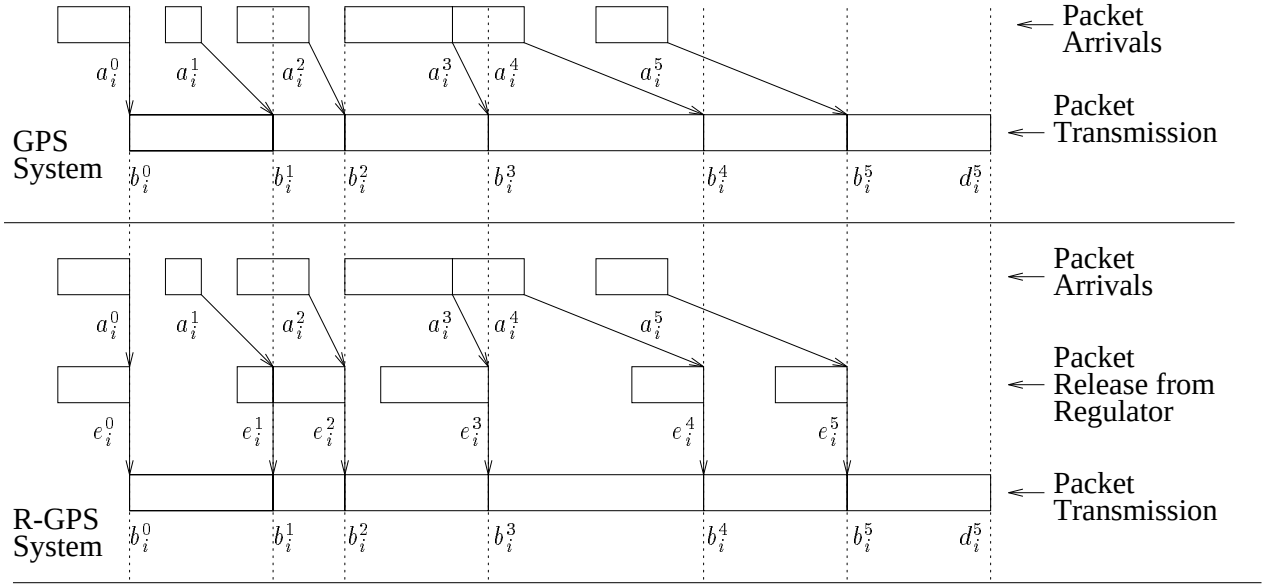


Figure 6: Comparison between GPS System and R-GPS System.

the arrival, regulation and service of the same packets in the corresponding R-GPS system. In the example, $r_i = \frac{r}{2}$.

Lemma 2 *An R-WFQ system is equivalent to the corresponding WF^{2}Q system, i.e., for any arrival sequence, packets are serviced in exactly the same order with either service discipline and* $d^k_{i,WF^2Q} = d^k_{i,R-WFQ}$ *holds.*

Now that we have the equivalence of R-GPS and GPS, and the equivalence of R-WFQ and WF^{2}Q, we are ready to present the proof of Theorem 1.

Proof of Theorem 1.

(1). We first prove (10) and (11).

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

$$d^k_{i,R-WFQ} - d^k_{i,R-GPS} \leq \frac{L_{max}}{r} \quad (14)$$

$$W_{i,R-GPS}(0,\tau) - W_{i,R-WFQ}(0,\tau) \leq L_{max} \quad (15)$$

Since the input traffic pattern, the regulators, and service shares allocated to each session are identical for the two corresponding R-WFQ and R-GPS systems, the input traffic pattern and the per session service shares for the two imbedded WFQ^* and GPS^* systems are also identical. Therefore, if R-WFQ and R-GPS are corresponding systems, the embedded WFQ^* and GPS^* are also corresponding systems.

In addition, we have:

$$d^k_{i,WFQ^*} = d^k_{i,R-WFQ} \quad (16)$$

$$d^k_{i,GPS^*} = d^k_{i,R-GPS} \quad (17)$$

$$W_{i,WFQ^*}(0,\tau) = W_{i,R-WFQ}(0,\tau) \quad (18)$$

$$W_{i,GPS^*}(0,\tau) = W_{i,R-GPS}(0,\tau) \quad (19)$$

(14) follows directly from (16),(17), and (5). Also, (15) follows directly from (18),(19), and (6).

(2). We now prove (12).

Since a packet will not start service in a WF^{2}Q system until it starts service in the corresponding GPS system, the following must hold

$$W_{i,WF^2Q}(0, b^k_{i,GPS}) \leq W_{i,GPS}(0, b^k_{i,GPS}) \quad \forall i,k \quad (20)$$

Without losing generality, let $b^{k_0}_{i,GPS} \leq \tau < b^{k_0+1}_{i,GPS}$. Since the maximum number of bits that can be served during the interval $[b^{k_0}_{i,GPS}, \tau]$ by WF^{2}Q is limited by both the link speed and the packet size, we have

$$W_{i,WF^2Q}(b^{k_0}_{i,GPS}, \tau) \leq min\{L^{k_0}_i, r(\tau - b^{k_0}_{i,GPS})\} \quad (21)$$

Also, since GPS guarantees a service rate r_i to a backlogged session, we have:

$$W_{i,GPS}(b^{k_0}_{i,GPS}, \tau) \geq min\{L^{k_0}_i, r_i(\tau - b^{k_0}_{i,GPS})\} \quad (22)$$

Combining (21) and (22), we have

$$W_{i,WF^2Q}(b^{k_0}_{i,GPS}, \tau) - W_{i,GPS}(b^{k_0}_{i,GPS}, \tau) \leq min\{L^{k_0}_i, r(\tau - b^{k_0}_{i,GPS})\} - min\{L^{k_0}_i, r_i(\tau - b^{k_0}_{i,GPS})\} \quad (23)$$

The right hand side of (23) is maximized when $L^{k_0}_i = r(\tau - b^{k_0}_{i,GPS})$ or $\tau = b^{k_0}_{i,GPS} + \frac{L^{k_0}_i}{r}$. This corresponds to the case where WF^{2}Q services the k^{th}_0

packet immediately when it becomes eligible, and the maximum difference in service between the packet system and the fluid system is achieved when the packet system finishes serving the packet. Plugging in (23), we have

$$W_{i,WF^2Q}(b_{i,GPS}^{k_0},\tau) - W_{i,GPS}(b_{i,GPS}^{k_0},\tau) \leq (1-\frac{r_i}{r})L_i^{k_0} \quad (24)$$

Combining (20) and (24), we have

$$W_{i,WF^2Q}(0,\tau) - W_{i,GPS}(0,\tau) \quad (25)$$
$$\leq (1-\tfrac{r_i}{r})L_i^{k_0} \quad (26)$$
$$\leq (1-\tfrac{r_i}{r})L_{i,max} \quad (27)$$

Q.E.D.

Now that we have shown that the service provided by WF^{2}Q and GPS are almost identical, we are ready to establish the worst-case fair property of WF^{2}Q. Before we do that, we make the following observation. Since the backlog function is the difference between the cumulative service function and the cumulative arrival function, the fact that the service functions of WF^{2}Q and GPS are close implies that their backlog functions are also very close. This is stated in the following lemma.

Corollary 1 *For two corresponding WF^{2}Q and GPS systems,*

$$Q_{i,WF^2Q}(\tau) - Q_{i,GPS}(\tau) \leq L_{max} \quad (28)$$
$$Q_{i,GPS}(\tau) - Q_{i,WF^2Q}(\tau) \leq (1-\frac{r_i}{r})L_{i,max} \quad (29)$$

So far we have shown that WF^{2}Q not only maintains the bounded delay property of GPS as WFQ does, but also keeps very close track of the GPS service. In the next theorem, we will show that by doing so, WF^{2}Q also maintains the worst-case fair property of GPS.

Theorem 2 *WF^{2}Q is worst-case fair for session i with $C_{i,WF^2Q} = \frac{L_{i,max}}{r_i} - \frac{L_{i,max}}{r} + \frac{L_{max}}{r}$, i.e.,*

$$d_{i,WF^2Q}^k - a_i^k \leq \frac{Q_{i,WF^2Q}(a_i^k)}{r_i} + \frac{L_{i,max}}{r_i} - \frac{L_{i,max}}{r} + \frac{L_{max}}{r} \quad \forall i,k \quad (30)$$

The theorem follows directly from (10) (29), and the following worst-case fair property of GPS:

$$d_{i,GPS}^k - a_i^k \leq \frac{Q_{i,GPS}(a_i^k)}{r_i} \quad (31)$$

Corollary 2 *In a network with all packets having the same size L, such as an ATM network, WF^{2}Q is worst-case fair for session i with the Worst-case Fair Index $C_{i,WF^2Q} = \frac{L}{r_i}$. WF^{2}Q is worst-case fair with the Normalized Worst-case Fair Index $c_{WF^2Q} = \frac{L_{max}}{r}$.*

Since the Normalized Worst-case Fair Index for a packet system is at least one packet transmission time, the above corollary shows that WF^{2}Q is an optimal packet policy with respect to the worst-case fair property. Therefore, we name the policy **Worst-case Fair** Weighted Fair Queueing.

Before we conclude this section, we would like to establish another important property of WF^{2}Q: the work-conserving property. Earlier in the section, we have shown that WF^{2}Q is equivalent to a rate-controlled server R-WFQ. In a rate-controlled server, it is possible that when a server is ready to transmit next packet, there are packets in the regulators but not in the scheduler, in which case the server would be idle even though there are packets in the server. This will result in a non-work-conserving policy. In fact, most recently proposed rate-controlled service disciplines are non-work-conserving [17]. While non-work-conserving policies have exhibited some unique advantages in providing guaranteed performance services [6, 17, 18], work-conserving policies are more efficient in providing best-effort service. As will be proven in the following theorem, unlike most other rate-controlled disciplines, WF^{2}Q is work-conserving.

Theorem 3 *WF^{2}Q is a work-conserving discipline*

*Proof.*From Lemma 1 and 2, a pair of corresponding GPS and WF^{2}Q systems are equivalent to the corresponding R-GPS and R-WFQ systems respectively. Also, the two embedded GPS^* and WFQ^* are corresponding systems. Since for the same input pattern, (1) R-GPS and GPS^* have identical system busy and idle periods, (2) GPS^* and WFQ^* have identical system busy and idle periods, and (3) R-WFQ and WFQ^* have identical system busy and idle period, it follows that R-GPS and R-WFQ, i.e., GPS and WF^{2}Q, also have identical busy and idle periods. Since GPS is work-conserving, we have WF^{2}Q is also work-conserving.

Q.E.D

4 Related Work

Since both WFQ and WF^{2}Q are defined with respect to the corresponding GPS system, both of them need to emulate GPS. However, maintaining the reference GPS server is computationally expensive. One simplier packet approximation algorithm of GPS is *Self-Clocked Fair Queueing* (SCFQ) [7] also known informally as "Chuck's Approximation" [3]. While simpler than WFQ and WF^{2}Q, SCFQ provides a much larger delay bounded than that by WF^{2}Q and WFQ [8]. It can be shown that SCFQ has the same worst-case fair index as WFQ. However, unlike in a WFQ server where a session has to transmit packets faster than its allocated rate in order to experience worse case service, in a SCFQ server, a sessions transmitting no faster than its reserved rate may still receive its worst-case service.

Another related discipline is the Virtual Clock algorithm [19]. While Virtual Clock can provide the

identical delay bound to a session whose source is constrained by a leaky bucket [5, 8, 15], its normalized Worst-case Fair Index can be arbitrarily large even when there are only two sessions [13, 16].

5 Summary and Future Work

We have made two contributions in this paper. First, we demonstrated that, contrary to popular belief, there can be a large discrepancy between the service provided by the packet WFQ system and the fluid GPS system. We use a metric called Worst-case Fair Index to quantitatively measure this discrepancy. We argued that such discrepancy may adversely effect many congestion control algorithms that rely on services similar to that provided by GPS. Second, we propose a new packet approximation algorithm of GPS called Worst-case Fair Weighted Fair Queueing or WF^2Q, and show that WF^2Q provides almost identical service to GPS differing by no more than one maximum size packet.

One of the issues that we didn't discuss in this paper is the feasibility of implementing WF^2Q at high speeds. In [13], Parekh proposed an implementation of WFQ based on a virtual time function, $V_{GPS}(\cdot)$. As observed in [7], the major obstacle of efficiently implementing WFQ, which is shared by WF^2Q, is the high complexity for the computation of $V_{GPS}(\cdot)$. Golestani introduces a new and simpler virtual time function, however, due to the large discrepancy between the new virtual time function and $V_{GPS}(\cdot)$, both the delay and the fairness properties of the resulted SCFQ discipline are worse than WF^2Q. In a follow-up paper, we will present another virtual time function that has a lower complexity than $V_{GPS}(\cdot)$ and approximates $V_{GPS}(\cdot)$ more accurately than $V_{SCFQ}(\cdot)$. We will show that the resulting discipline provides the same delay bound and has the same worst-case fair index as WF^2Q.

References

[1] Greg M. Bernstein. Reserved bandwidth and reservationless traffic in rate allocating servers. *Computer Communication Review*, 23(3), July 1993.

[2] D. Clark, M. Lambert, L. Zhang . NETBLT: A high throughput transport protocol. In *Proceedings of the ACM-SIGCOMM 87*, pages 353–359, Stowe, VT, August 1987.

[3] J. Davin and A. Heybey. A simulation study of fair queueing and policy enforcement. *Computer Communication Review*, 20(5):23–29, October 1990.

[4] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Journal of Internetworking Research and Experience*, pages 3–26, October 1990. Also in Proceedings of ACM SIGCOMM'89, pp 3-12.

[5] N. Figueira and J. Pasquale. An upper bound on delay for the virtualclock service discipline. *IEEE/ACM Transactions on Networking*, 3(4), April 1995.

[6] L. Georgiadis, R. Guérin, and V. Peris. Efficient network QoS provisioning based on per node traffic shaping. In *IEEE INFOCOM'96*, San Francisco, CA, March 1996.

[7] S.J. Golestani. A self-clocked fair queueing scheme for broadband applications. In *Proceedings of IEEE INFOCOM'94*, pages 636–646, Toronto, CA, April 1994.

[8] P. Goyal, S. Lam, and H. Vin. Determining end-to-end delay bounds in heterogeneous networks. In *Proceedings of the 5th International Workshop on Network and Operating System Support For Digital Audio and Video*, pages 287–298, Durham, New Hampshire, April 1995.

[9] S. Keshav. A control-theoretic approach to flow control. In *Proceedings of ACM SIGCOMM'91*, pages 3–15, Zurich, Switzerland, September 1991.

[10] L. Kleinrock. *Queueing Systems, Volume 2: Computer Applications*. Wiley, 1976.

[11] B. Lyles and A. Lin. A class-y mechanism and preliminary simulations, July 1994. #94-207, ANSI T1S1.5.

[12] J Nagle. On packet switches with infinite storage. *IEEE Trans. On Communications*, 35(4):435–438, April 1987.

[13] A. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD dissertation, Massachusetts Institute of Technology, February 1992.

[14] S. Shenker. Making greed work in networks: A game theoretical analysis of switch service disciplines. In *Proceedings of ACM SIGCOMM'94*, pages 47–57, London, UK, August 1994.

[15] G. Xie and S. Lam. Delay guarantee of virtual clock server. *IEEE/ACM Transactions on Networking*, 3(4):683–689, December 1995.

[16] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10):1374–1399, October 1995.

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

[18] H. Zhang and S. Keshav. Comparison of rate-based service disciplines. In *Proceedings of ACM SIGCOMM'91*, pages 113–122, Zurich, Switzerland, September 1991.

[19] L. Zhang. Virtual clock: A new traffic control algorithm for packet switching networks. In *Proceedings of ACM SIGCOMM'90*, pages 19–29, Philadelphia, PA, September 1990.