M/G/FQ: STOCHASTIC ANALYSIS OF FAIR QUEUEING SYSTEMS

MOHAMMED HAWA AND DAVID W. PETR
Information and Telecommunications Technology Center
University of Kansas, Lawrence, Kansas, 66045
email: {hawa, dwp}@ittc.ku.edu

Fair scheduling or Fair Queueing (FQ) algorithms have received much attention in recent years because of their ability to provide a wide range of Quality-of-Service (QoS) guarantees to end users. In this paper, we present a new analysis method that can statistically model such fair scheduling algorithms under Poisson arrivals and a general packet length distribution. We coin the new term M/G/FQ to describe this analysis method, which fits a broad range of scheduling policies including WFQ, SCFQ, SFQ and SPFQ.

1 Introduction

The main advantage of *fair scheduling* or *Fair Queueing* (FQ) algorithms is the ability to provide minimum Quality-of-Service (QoS) guarantees to end users [1]. Examples of well-known FQ algorithms include the Generalized Processor Sharing (GPS) policy [2], Weighted Fair Queuing (WFQ) [2, 3], Self-Clocked Fair Queuing (SCFQ) [4], Start-Time Fair Queuing (SFQ) [5] and Starting Potential-based Fair Queuing (SPFQ) [6].

A significant volume of work in the literature $[2-10]$ has been concerned with evaluating the deterministic worst-case delay guarantees that these FQ algorithms can provide when the burstiness of the traffic feeding them is bounded (for example, shaped by a leaky bucket).

Little work, though, has been reported on analyzing the delay characteristics of such policies under a general probabilistic traffic model. This has been mainly due to the difficulty of statistically modeling the complex behavior of a FQ algorithm. Indeed an important advantage of statistical modeling of FQ systems as compared to worst-case deterministic analysis is that stochastic analysis takes into account the actual dynamics of the packet arrival process, thus being more accurate in predicting the system status under normal operating conditions.

To the best of our knowledge, the only work on statistical modeling of FQ algorithms is that of [11], in which the author derives stochastic bounds on the delay distribution of GPS-related FQ algorithms fed by a Switched Bernoulli Batch process. The analysis in [11] is quite complex and does not result in explicit analytical equations, thus limiting its usefulness for backof-the-envelope calculations and comparisons. The analysis also makes some limiting assumptions such as the use of fixed packet lengths and the need to set all flows other than the tagged one to be greedy all the time.

The analysis we introduce in this paper, on the other hand, is much simpler than that in [11] and results in upper and lower bounds on mean packet delay and mean buffer occupancy experienced by a FQ algorithm fed by Poisson arrivals. Our analysis follows closely the well-known M/G/1 queueing analysis, thus the name M/G/FQ, and results in well-contained equations that provide significant theoretical value and great insight into the operation of FQ systems.

Our analysis fits a broad range of scheduling policies for which the difference between the normalized service received by any connection and the system potential is bounded at any time (see Section 2.2 for the exact definition of such a fairness bound). This class of FQ algorithms is similar to the one studied in [7] and many scheduling policies belong to this group including WFQ, SCFQ, SFQ and SPFQ. The key to our analysis is to utilize the bounded fairness criterion of FQ algorithms in order to derive the desired bounds on mean packet delay and mean buffer occupancy.

The rest of this paper is organized as follows. In the next section, we provide a quick overview of FQ algorithms and their bounded fairness criterion. We then lay out our stochastic analysis method in Section 3. Sections 4 and 5 provide more details about the properties of the derived delay bounds and some related experimental results. Finally, we provide some concluding remarks in Section 6.

2 Background

2.1 Classification of Fair Queueing Algorithms

In this paper we consider *sorted packet*-based FQ algorithms, which are mainly derived from a packet-by-packet implementation of the GPS policy suggested in [2]. Such FQ algorithms provide minimum bandwidth guarantees to supported flows by assigning certain timestamps to arriving packets and then serving those packets in increasing order of their timestamps. The timestamps (either *virtual finish times* or *virtual start times*) are assigned based on a system-wide function called the *virtual time*, denoted by $v(t)$, which tracks the progress of work in the scheduling system.

We can divide sorted packet-based FQ algorithms into *Earliest Finish Time First* (EFTF) policies, in which packets are scheduled in the increasing order of their virtual finish times (e.g., WFQ, SCFQ and SPFQ), and

Earliest Start Time First (ESTF) policies, in which packets are scheduled in the increasing order of their virtual start times (e.g. SFQ). The delay analysis presented in this paper is intended for sorted packet-based EFTF FQ algorithms. The analysis can be easily expanded to ESTF algorithms as well.

2.2 The Bounded Fairness Criterion of FQ Algorithms

Let us consider a scheduler served by an access link with a total capacity of C (bits/second). We denote by K the set of flows supported by this scheduler, and by r_k the minimum reserved service rate (in bits/second) associated with each flow k, $k \in K$. Let us also denote by $W_k(t_1, t_2)$ the aggregate service (in bits) received by flow k during the time interval $[t_1, t_2]$. $W_k(t_1, t_2)/r_k$ is then the total *normalized service* provided to flow k during that time interval.

A *scheduling algorithm is said to be fair if the difference in normalized services received by different backlogged flows in the scheduler is bounded* (by a *fairness bound* Ψ) *for all intervals of time* [4], where the value of Ψ is specific to the scheduling algorithm under consideration.

In sorted packet-based schedulers, an alternative definition of fairness is also possible. To arrive at such a definition we notice that such FQ algorithms maintain, in addition to the virtual time (also called the *system potential*), a *connection potential* $v_k(t)$ associated with each flow $k \in K$. The connection potential keeps track of the amount of normalized service *received* by that connection, and is mathematically defined as follows,

$$v_k(t_2) = \begin{cases} v_k(t_1) + W_k(t_1, t_2)/r_k\,, & k \in B(t_1, t_2) \\ v(t_2), & k \notin B(t_1, t_2) \end{cases} \tag{1}$$

where $B(t_1, t_2)$ is the set of flows which are backlogged[a] during the entire time interval $[t_1, t_2]$. Connection potentials can be used to generate timestamps for the packets queued in a FQ system. In an EFTF FQ system, for example, the Nth packet in queue k has the timestamp (virtual finish time) of [10],

$$v_k(t) + \sum_{n=1}^{N} \frac{L_k^n}{r_k} \tag{2}$$

where L_k^n is the length of the nth packet queued in buffer k at time t.

---

[a] It is worth mentioning that in some FQ algorithms, such as WFQ, the condition $k \in B(t_1, t_2)$ in (1) refers to flows being backlogged in the fluid-based reference system maintained by such FQ algorithms rather than the actual packet-by-packet system.

The *service lag* of a flow k, denoted by $\delta_k(t)$, is defined as the difference between the system potential $v(t)$ and the connection potential $v_k(t)$ of flow k at any time t, i.e.,

$$\delta_k(t) = v(t) - v_k(t), \qquad k \in K \tag{3}$$

The fairness bound Ψ for sorted packet-based FQ algorithms can be translated into an equivalent fairness bound in terms of the service lag in (3). This fairness bound can be written as follows [12],

$$0 \leq \delta_k(t) \leq \psi_k(t), \qquad k \in K \tag{4}$$

where $\psi_k(t)$ is a fairness bound specific to the scheduling algorithm under consideration[b]. In SCFQ, for example, the bound on the service lag of flow k is given by $\psi_{k,SCFQ}(t) = L_k/r_k$, where L_k is the length of the packet in queue k that finishes service after time t [4].

Moving on, it is easy to see that the mean of the service lag $\delta_k(t)$, denoted by $\overline{\delta_k}$, has the following bounds,

$$0 \leq \overline{\delta_k} \leq \overline{\psi_k}, \qquad k \in K \tag{5}$$

where $\overline{\psi_k}$ is the mean of $\psi_k(t)$. As it turns out, however, we can derive tighter bounds on the quantity $\overline{\delta_k}$ by noticing that the connection potential of flow k becomes equal to the system potential (i.e., the service lag becomes zero) when flow k is not backlogged and remains that way until the flow is backlogged again. It can be shown that this results in the following tighter bounds on $\overline{\delta_k}$ [12],

$$0 \leq \overline{\delta_k} \leq \lambda_k \frac{\overline{L_k}}{r_k} \overline{\psi_k} = \rho'_k \overline{\psi_k}, \qquad k \in K \tag{6}$$

where $\overline{L_k}$ is the mean length of flow k data packets and λ_k is the mean arrival rate of that flow. In (6) we used ρ'_k to represent the quantity $\lambda_k \overline{L_k}/r_k$ to express the notion of a new utilization factor of flow k under an equivalent server of capacity r_k.

[b] The results derived in this paper work just as well for FQ algorithms that have a fairness bound of the form $-a_k \leq \delta_k \leq b_k$. We only need to define a new equivalent fairness bound given by $0 \leq \delta'_k \leq a_k + b_k = \psi'_k$.

Another important parameter that we will use in our analysis is the lag between the connection potentials of two different flows in a FQ system, defined as follows,

$$\delta_{kj}(t) = v_k(t) - v_j(t) = [v(t) - v_j(t)] - [v(t) - v_k(t)] = \delta_j(t) - \delta_k(t), \quad k, j \in K \tag{7}$$

The $\delta_{kj}(t)$ parameter is a random variable with a mean that is bounded in a sorted packet-based FQ system by (cf. (6) and (7)),

$$-\rho'_k \overline{\psi_k} \leq \overline{\delta_{kj}} \leq \rho'_j \overline{\psi_j}, \qquad k, j \in K \tag{8}$$

It is worth mentioning that if we introduce a reasonable, although not mathematically rigorous, assumption we can dramatically enhance the upper bound on $\overline{\delta_{kj}}$ in (8). The assumption is that the distributions of $\delta_k(t)$ and $\delta_j(t)$, $k, j \in K$, are similar[c], which is justified by the fact that different flows in a FQ system are treated in the exact same way except for the amount of reserved bandwidth they receive. This assumption is valid in almost all practical cases, and although an approximation, it can be used safely in many situations where tighter bounds are desired.

It can be shown [12] that if such an assumption holds, a new upper bound on $\overline{\delta_{kj}}$ can be found as follows,

$$\overline{\delta_{kj}} \leq \rho'_j \overline{\psi_j} - \rho'_k \overline{\psi_k}, \qquad k, j \in K \tag{9}$$

This new upper bound is tighter than that in (8), and will result in a tighter upper bound on mean packet delay (see Sections 3 and 5).

3 M/G/FQ Stochastic Analysis

Let us consider a single-server EFTF FQ system fed by multiple Poisson streams with arrival rates $\lambda_1, \lambda_2, \ldots, \lambda_K$ as shown in Figure 1. The buffers corresponding to different flows are infinite in length and the packets in each of those buffers are served in the order they arrive. We use L_k^i to denote the length (in bits) of the ith data packet arrival at the kth buffer, $k \in K$. Similarly, we use $X_k^i = L_k^i / C$ to denote the service time (in seconds) of the ith data packet arrival at the kth buffer, where C is the output link capacity. The random variables X_k^i from the multiple Poisson streams are

[c] We refer to two distributions as similar if they have the forms $f_\delta(\delta)$ and $(1/\tau)\, f_\delta(\delta/\tau)$, where τ is a constant.

identically distributed, mutually independent, and independent of the arrival times. Such variables X_k^i can assume any general distribution. We denote the mean service time of arriving data packets by $\overline{X} = E\left[X_k^i\right] = 1/\mu$, where μ is the mean service rate. The second moment of the service time is denoted by $\overline{X^2}$. For convenience, we will refer to the total arrival rate at the FQ system by $\lambda = \sum_{k \in K} \lambda_k$.

The utilization of each connection k is denoted by $\rho_k = \lambda_k \overline{X} = \lambda_k/\mu$, while the utilization of the output link is given by $\rho = \lambda \overline{X}$. We also previously defined $\rho'_k = \lambda_k \overline{L_k}/r_k$. In this analysis we maintain $\rho'_k < 1$ for all $k \in K$. This will ensure that we maintain $\rho < 1$, which keeps the system from being overloaded (in an average sense)[d].

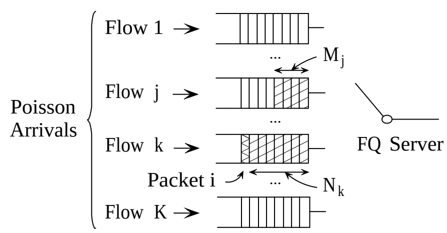

Figure 1. Arrivals at the Fair Queueing System.

We denote the *mean* packet waiting time in queue k by W_k, and the *expected* number of packets in such a queue (*not* including any packet that may be in service) by N_k. We assume *ergodicity* of the queueing system (which is true provided that $\rho < 1$) and note that, in our system, the values of N_k and W_k seen by an outside observer at a random time are the same as seen by an arriving customer. This is due to the Poisson character of the arrival process, which implies that the occupancy distribution upon arrival is typical [13].

Now, let us consider the ith data packet arrival at the kth queue of the FQ system. This packet must wait in queue for a mean residual time R until the end of the current packet transmission and must also wait for the transmission of the mean number of packets N_k currently in the kth queue ahead of it. In addition, the ith packet must also wait for the transmission of all packets in

[d] For the analysis presented in this paper to be valid we only need to maintain $\rho < 1$. However, the extra requirement that $\rho'_k < 1$, $k \in K$, adds to the convenience of the derivation without any loss of generality.

the system (not in queue k) with timestamps (virtual finish times) that are smaller than the timestamp (virtual finish time) assigned to the ith packet. The mean number of such packets in each queue j in the system is denoted by M_j (see Figure 1).

Thus, the mean queuing delay (waiting time in queue k) for the ith packet is given by,

$$W_k = R + \frac{1}{\mu}\left(N_k + \sum_{j\in J} M_j\right) \tag{10}$$

where we used the fact that buffer occupancy is independent of individual packet service times, and we also used J to represent the set of all flows supported by the scheduler *but* flow k. In other words, $J = \{j \in K : j \neq k\}$.

We can evaluate the mean residual time R by a graphical argument as in [13] to obtain $R = \lambda\overline{X^2}\big/2$. Using Little's law, $N_k = \lambda_k W_k$, we get,

$$W_k = \frac{1}{2}\overline{X^2}\lambda + \rho_k W_k + \frac{1}{\mu}\sum_{j\in J} M_j \tag{11}$$

The only unknown quantities in (11) now are the values of M_j. To find such values, let us consider a single queue $j \in J$ in the system. Assume that this queue has a connection potential $v_j(a_k^i)$ at the time a_k^i of packet i arrival. Assume also that the connection potential of queue k was $v_k(a_k^i)$ at that time. Using the result in (2) and noting that the M_jth packet in the jth queue should have a timestamp that is smaller than the ith packet (in queue k) so that the FQ scheduler can serve it first, we get,

$$E\left[v_j(a_k^i)\right] + M_j\frac{\overline{L_j}}{r_j} \leq E\left[v_k(a_k^i)\right] + (N_k+1)\frac{\overline{L_k}}{r_k}$$

Also, packet M_j+1 in the jth queue should have a timestamp that is larger than the ith packet in queue k by construction. Hence, we can write,

$$E\left[v_j(a_k^i)\right] + (M_j+1)\frac{\overline{L_j}}{r_j} \geq E\left[v_k(a_k^i)\right] + (N_k+1)\frac{\overline{L_k}}{r_k}$$

Rearranging, we get the following upper and lower bounds on M_j, respectively,

$$M_j \leq \min\left((N_k+1)\frac{r_j}{r_k} + \overline{\delta_{kj}}\frac{r_j}{\overline{X}C},\ N_j\right),\quad j \in J \tag{12}$$

$$M_j \geq \min\left(\max\left((N_k+1)\frac{r_j}{r_k} + \overline{\delta_{kj}}\frac{r_j}{\overline{X}C} - 1,\ 0\right),\ N_j\right),\quad j \in J \tag{13}$$

where $\delta_{kj} = v_k - v_j$ is the difference in the connection potentials of flows k and j. Notice that in (13) we limited the lower bound on M_j to a minimum value of zero. Otherwise, such a lower bound may turn out to be negative, which happens when N_k is small and the conversion factor $r_j/r_k \ll 1$. Such a negative value of M_j is not practically acceptable. Also notice that we have limited the M_j upper and lower bounds in (12) and (13) to a maximum value equal to the mean number of packets N_j in the jth buffer, which is another practical limit we have to maintain.

3.1 The Upper Bound on Mean Packet Delay

The challenge we face in trying to solve for the upper bound on mean packet delay is that we need to choose the minimum of two quantities in (12) for each flow $j \in J$ before being able to substitute it into (11). Such a decision cannot be made without prior knowledge of the actual N_j (or W_j) values, which are the unknowns we are seeking to find.

To avoid such a problem we notice that as far as the upper bound on mean packet delay is concerned, using the first expression on the right hand side of (12) instead of the minimum does not actually affect the correctness of the upper bound on M_j, although it might slightly weaken its tightness. Since such an expression is not dependent on the value of N_j, we can drastically simplify the derivation process, which gives the following upper bound on mean packet delay based on (11), (12) and the upper bound on $\overline{\delta_{kj}}$ from (8),

$$W_k \leq \frac{\frac{1}{2}\overline{X^2}\lambda + \sum\limits_{j\in J}\left[\frac{1}{\mu}\frac{r_j}{r_k} + \rho'_j\overline{\psi_j}\frac{r_j}{C}\right]}{1 - \rho_k\frac{\sum\limits_{j\in K} r_j}{r_k}} \tag{14}$$

On the other hand, using the improved upper bound on $\overline{\delta_{kj}}$ from (9) transforms (14) into,

$$W_k \leq \frac{\frac{1}{2}\overline{X^2}\lambda + \sum\limits_{j\in J}\left[\frac{1}{\mu}\frac{r_j}{r_k} + \left(\rho'_j\overline{\psi_j} - \rho'_k\overline{\psi_k}\right)\frac{r_j}{C}\right]}{1 - \rho_k\frac{\sum\limits_{j\in K} r_j}{r_k}} \tag{15}$$

3.2 The Lower Bound on Mean Packet Delay

Finding the lower bound on the mean packet delay W_k requires a similar approach to that of finding the upper bound. For the lower bound, however, we cannot just substitute the first expression in (13) instead of the required minimum since this is mathematically incorrect. However, we can still derive a simple equation for the lower bound similar to that of (14) by setting the minimum in (13) to zero all the time, which gives the following simple lower bound on mean packet delay,

$$W_k \geq \frac{\frac{1}{2}\overline{X^2}\lambda}{1 - \rho_k} \tag{16}$$

Obviously, this is not the best possible lower bound on mean packet delay. However, as will be apparent in Section 5, this lower bound is reasonably tight in almost all practical cases one might encounter.

Now, using the upper and lower bounds on mean packet delay in (14), (15) and (16) we can derive the corresponding upper and lower bounds on mean buffer occupancy using Little's law, which states that $N_k = \lambda_k W_k$.

4 Properties of the Delay Bounds

An important observation we can make about the M/G/FQ delay bounds derived earlier in Section 3 is that both the upper and lower bounds increase in inverse proportion to $1-\rho_k$ (or $1-\rho'_k$). This means that the mean packet delay in our system is expected to dramatically increase as the utilization factor approaches unity, or at least that would be the behavior of the delay bounds in such a condition. Comparing this to an M/G/1 queueing system, we notice the same exact behavior for the mean packet delay versus utilization. This behavior is actually a general characteristic of almost any queueing system one might encounter.

Now let us consider which operating conditions would result in tighter bounds on mean packet delay. It is easy to see from (14) and (16) that the difference between the upper and lower bounds is mainly dependent on a summation factor including the fairness bounds $\rho'_j\overline{\psi_j}$ for all $j \in J$. This means that tighter bounds are expected in the following situations: (1) when the number of flows K supported by the scheduler is smaller, (2) when the fairness bound of the FQ system is tighter and (3) when the load on queue k, measured by ρ_k, is smaller.

As a final note, we redirect the reader's attention to the fact that the upper and lower bounds in (14) and (16) are applicable not only to one specific FQ algorithm, but rather to the whole class of FQ policies that exhibit a specific fairness bound. This illustrates the flexibility and generality of our analysis method, which requires only partial information (the fairness bound) about the FQ algorithm itself to be able to produce bounding criterion for its mean packet delay and mean buffer occupancy. Such ability is quite important in many situations where the complexity of the FQ algorithm under consideration may prohibit mathematical tractability of its exact properties.

On the other side of the coin, deriving delay bounds based on partial information (the fairness bound) of FQ algorithms means that the obtained bounds need to be as wide apart as possible to accommodate all scheduling policies with the same fairness bound irrespective of their internal structure. The power of our analysis method is that we can further tailor its delay bounds to a specific scheduling algorithm when needed by considering the exact forms of (11) – (13) and using tighter bounds on the service lag distributions of such algorithms. Of course, this requires more information about the specific scheduling policy under consideration to be known ahead of time to carry out such an analysis.

5 Experimental Results and Discussion

We move now to perform several simulations to validate the results of the M/G/FQ analysis presented in Section 3. A word of caution is essential at this point. Because it is quite difficult to find many FQ algorithms with the exact same fairness bound, we decided to study three EFTF FQ algorithms (namely SCFQ, WFQ and SPFQ) using the fairness bound $\overline{\psi_k} = \overline{L_k}/r_k$ even though this bound is exact only for SCFQ. These three algorithms have close enough fairness bounds that we can safely use only one of them to illustrate the points we are trying to make.

5.1 Experiment: Bounding the Delay of Multiple FQ Algorithms

In this experiment, a FQ server with a total output link capacity of 1 Mb/s supports four incoming Poisson streams under SCFQ, WFQ or SPFQ. The reserved rates for the different connections are: $r_1 = r_2 = r_3 = r_4 = 0.25$ Mb/s. The tagged source is the $k = 3$ one, and its mean arrival rate varies between 0.025 and 0.225 Mb/s, while all other sources transmit at a fixed mean rate of 0.25 Mb/s. The packet length distribution is uniform and ranges between 500 bits and 1500 bits per packet.

The results of the experiment are shown in Figure 2(a), in which the simulation-generated mean packet delay for the tagged flow is displayed versus load under the different FQ policies. We also show in this figure the analytical upper and lower bounds derived for this case based on (15) and (16).

We notice from the results that the mean packet delay values generated by SCFQ, WFQ and SPFQ are all bounded by the upper and lower bounds of the M/G/FQ analysis irrespective of the incoming load value. This emphasizes the fact that the delay bounds derived here actually accommodate all FQ algorithms that exhibit the same fairness bound irrespective of their internal operations, and as such the delay bounds in (15) and (16) are reasonably tight.

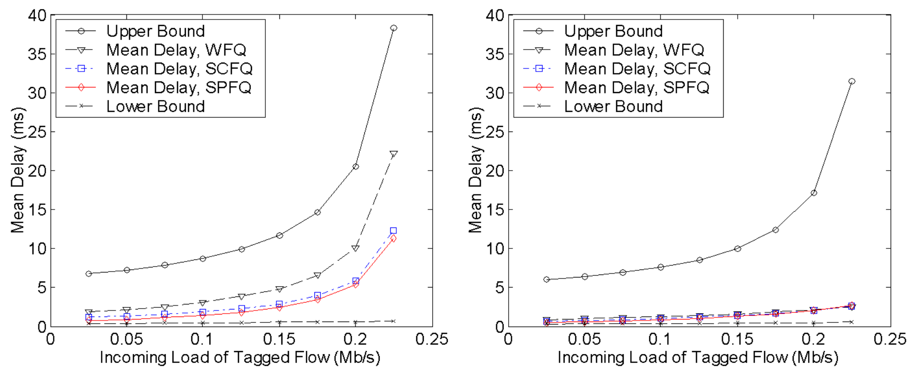

Figure 2. Mean Delay versus Incoming Load for four Poisson streams with a uniform packet length distribution under (a) a heavily loaded system; and (b) a partially loaded system.

The upper bound shown in Figure 2(a) is actually the improved upper bound as per (15). As we can see, it still bounds the mean packet delay for all tested FQ algorithms. This strengthens the validity of the assumption we made earlier in Section 2.2 to improve the upper bound on $\overline{\delta_{kj}}$. To appreciate the advantage of using this improved upper bound, we notice that while such an improved bound will result in a mean delay value of 20.6 ms when the incoming load of the tagged flow is equal to 0.2 Mb/s, the more rigorous upper bound results in a larger value of 32.6 ms. This is why in all the following experiments we will only display the improved upper bound on mean delay.

The lower bound in Figure 2(a) is also reasonably tight for low to moderate incoming load values. However, it does not show the same dramatic behavior as the incoming load approaches its limits when compared to the actual mean packet delay curves. This is the price we have to pay to obtain a reasonable mathematical formula as that in (16) for the lower bound on mean packet delay.

5.2 Experiment: Reducing the Load on the FQ System

In the previous experiment, we saw the dramatic increase of the mean packet delay as the utilization factor ρ'_k of the tagged flow approached unity (see Figure 2(a)). This was the behavior we would expect by looking at (15), which describes the upper bound on mean packet delay.

However, a key part contributing to this behavior is actually the fact that all flows other than the tagged one are transmitting at their nominal reserved rate, i.e., they are greedy. The aim of this experiment is to investigate the effects of reducing the load of the non-tagged flows on the results of the system.

For this experiment, we use the exact same parameters as in Section 5.1 with the exception that all flows other than the tagged one are set to transmit at a fixed mean rate of 0.20 Mb/s instead of 0.25 Mb/s. The results are shown in Figure 2(b), the scale of which is set to match Figure 2(a).

In this experiment the load on the FQ system is smaller than that in Section 5.1. Under such circumstances, all three studied FQ policies give comparable performance because the FQ server is free for a considerable amount of time. The lower bound is reasonably tight. The upper bound, on the other hand, cannot completely adjust at higher input load values. The reason for this is that when the load on the non-tagged flows is reduced, the mean packet count $N_j, j \in J$, drops dramatically. This means that neglecting the N_j terms in (12) in such a case results in a weaker upper bound. In such a situation, a more elaborate (and more complex) analysis is preferred to better describe the system under consideration. However, for simpler results, the lower bound gives a good quick estimate of the mean packet delay when the system is partially loaded.

5.3 Experiment: Different Reservations and Packet Length Distributions

In this experiment, we set up a FQ server with an output link capacity of 1 Mb/s that supports two incoming Poisson streams under SCFQ, WFQ or SPFQ. The reserved rates for the two flows are: $r_1 = 0.6$ Mb/s and $r_2 = 0.4$ Mb/s. The tagged flow is the $k = 2$ one, and its mean arrival rate varies between 0.05 and 0.35 Mb/s, while the other flow transmits at a fixed mean rate of 0.6 Mb/s. The packet length distribution is first set to a uniform distribution that ranges between 500 bits and 1500 bits per packet, and then set to an exponential distribution with an average packet length of 1000 bits per packet.

The results of the experiment are shown in Figures 3 (a) and (b) for the uniform and exponential packet length distributions, respectively, which clearly confirms that the delay bounds are working correctly for both cases.

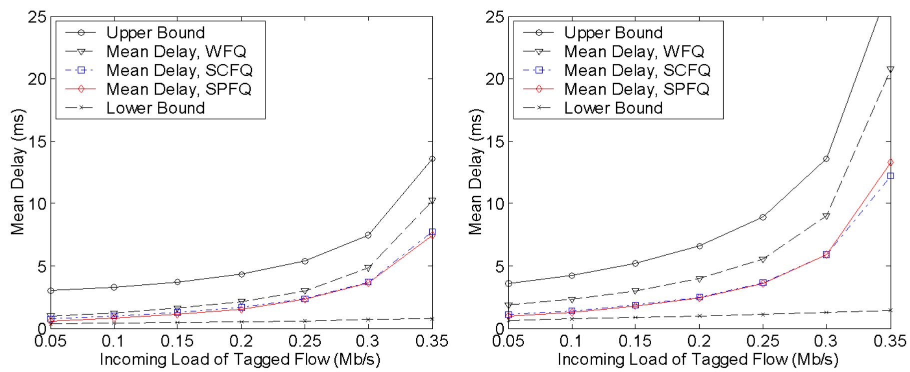

Figure 3. Mean Delay versus Incoming Load for two Poisson streams with (a) a uniform packet length distribution; and (b) an exponential packet length distribution.

6 Concluding Remarks

In this paper we introduced a new analysis method that produces simple and reasonably tight upper and lower bounds on mean packet delay of FQ algorithms under Poisson arrivals. The analysis method uses the bounded fairness criterion of FQ algorithms (represented by the bounds on mean service lag distribution) in order to derive the desired bounds on mean packet delay and mean buffer occupancy.

We showed several experiments that illustrate the validity of such delay bounds and how they accommodate different FQ policies with the same fairness criterion.

We are currently investigating the possibility of deriving tighter bounds on the mean service lag for some of the well-known FQ policies and using those values to derive tighter bounds on mean packet delay for such scheduling algorithms. We are also investigating the possibility of including the N_j factor discussed earlier in Section 3.1 in a more elaborate analysis to provide better results for the upper and lower bounds on mean packet delay under various loading conditions.

Acknowledgements

We thank Sprint Corporation for their support of this work.

References

1. Mohammed Hawa and David W. Petr, "Quality of Service Scheduling in Cable and Broadband Wireless Access Systems," Tenth International Workshop on Quality of Service, pp. 247–255, May 2002.
2. A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," INFOCOM'92, pp. 915–924, May 1992.
3. A. Demers, S. Keshav and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," Journal of Internetworking Research and Experience, pp. 3–26, October 1990. Also in ACM SIGCOMM'89, pp. 3–12, 1989.
4. S. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Applications," INFOCOM'94, pp. 636–646, April 1994.
5. P. Goyal, H. M. Vin and H. Chen, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," SIGCOMM'96, pp.157–169, September 1996.
6. D. Stiliadis and A. Varma, "Efficient Fair Queueing Algorithms for Packet-Switched Networks," IEEE/ACM Transactions on Networking, Vol. 6, No. 2, pp. 175–185, April 1998.
7. S. J. Golestani, "Network Delay Analysis of a Class of Fair Queueing Algorithms," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, pp. 1057–1070, August 1995.
8. P. Goyal, S. S. Lam and H. M. Vin, "Determining End-to-End Delay Bounds in Heterogeneous Networks," 5th International Workshop on Network and Operating System Support for Digital Audio and Video, pp. 287–298, April 1995.
9. D. Stiliadis and A. Varma, "Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms," INFOCOM '96, pp. 111–119, March 1996.
10. F. M. Chiussi and A. Francini, "Implementing Fair Queueing in ATM switches - Part 1: A Practical Methodology for the Analysis of Delay Bounds," GLOBECOM 97, Vol. 1, pp. 509–518, November 1997.
11. N. Pekergin, "Stochastic Bounds on Delays of Fair Queueing Algorithms," INFOCOM '99, pp. 1212–1219, 1999.
12. Mohammed Hawa, "Quality of Service in Broadband Wireless Networks," Ph.D. Thesis, University of Kansas, In Preparation.
13. Dimitri P. Bertsekas and Robert Gallager, "Data Networks," Prentice Hall, Second Edition, 1991.