Supporting Best-Effort Traffic With Fair Service Curve*

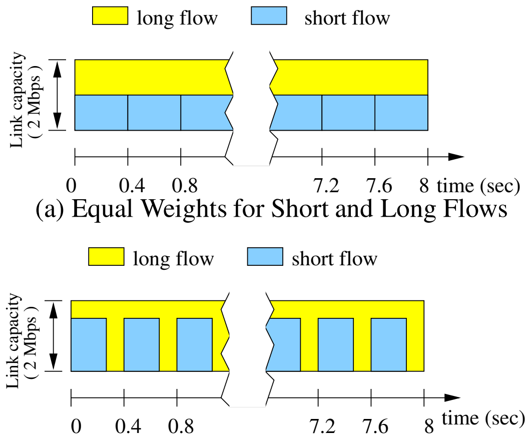
T.S. Eugene Ng, Donpaul C. Stephens, Ion Stoica, Hui Zhang
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
{eugeneng,donpaul,istoica,hzhang}@cs.cmu.edu

1 Introduction

Packet Fair Queueing (PFQ) algorithms are the most popular and well studied scheduling algorithms for integrated services networks for two reasons: (1) With reservation, they can provide per-flow end-to-end delay guarantees for real-time traffic flows. (2) Without reservation, they can provide protection among competing best-effort flows while allowing dynamic bandwidth sharing. However, PFQ algorithms have two important limitations. The first and more well known limitation is that, since only one parameter (a weight) is used to allocate resource for each flow, there is a coupling between delay and bandwidth allocation. This can result in network under-utilization when real-time flows have diverse delay and bandwidth requirements. The second and less well known limitation is that, when used for best-effort traffic, PFQ algorithms favor throughput-oriented applications such as FTP over delay-sensitive bursty applications such as WWW, and telnet. This is due to the instantaneous fairness property of PFQ algorithms, which are memory-less to any recent activity of the traffic. In a previous study [3], we proposed the Fair Service Curve (FSC) algorithm which enables more flexible delay and bandwidth allocation for real-time traffic through the use of non-linear service curves. In this paper [2], we show that, when used for best-effort traffic, FSC can eliminate the bias against delay-sensitive bursty applications without negatively affecting the performance of throughput-oriented applications.

2 Timescales for Best Effort Traffic

Different types of best-effort data traffic, such as Telnet, FTP, and WWW have different characteristics and thus performance objectives. For example, while the burst delay is the performance index for interactive services, the average throughput is the performance index for bulk transfer applications such as FTP. The key observation is that, since the performance index of bulk-transfer applications is determined over relatively long timescales, we may be able



(a) Equal Weights for Short and Long Flows

(b) Short Flows with $3\times$ the Long Flow's weight

Figure 1: Improving burst delays

to exploit these applications' insensitivity to short term service variations to improve the performance of delay sensitive bursty applications.

To illustrate how this may be realized, consider a 2 Mbps link shared by one long flow that transfers 1 MB, and several short flows that transfer 50 KB each. Assume that the link is managed by PFQ and each flow has a weight of one. For simplicity, assume that all flows are continuously backlogged, and that once a short flow finishes, another short flow starts immediately. Thus, there are exactly two flows, the long flow and a short flow, backlogged at any given time. As a result each backlogged flow is allocated 1 Mbps. Therefore, as shown in Figure 2 (a), the long flow takes 8 seconds to finish, while a short flow takes 0.4 seconds to complete. Now consider the case where all short flows are assigned three times the weight of the long flow. Each short flow now receives 1.5 Mbps, which consequently reduces its latency by 33% to 0.27 seconds. At the same time, the transfer time of the long flow does *not* change. Thus, by assigning different weights, it is possible to significantly speed-up short transfers without affecting the longer flow.

In order to achieve this performance, a system would either need to estimate the length of a flow when it becomes backlogged, or dynamically reduce the flow's weight after the length of the transfer exceeds a certain threshold. While it is unclear how this could be implemented in a system based on PFQ, we can achieve this in a system based on FSC by properly setting the service curves to allow a flow to burst at a higher rate for a short period if it was previously idle.

*This research was sponsored by DARPA under contract numbers N66001-96-C-8528 and E30602-97-2-0287, and by NSF under grant numbers Career Award NCR-9624979 and ANI-9814929. Additional support was provided by Intel Corp. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA, NSF, Intel, or the U.S. government.

3 Fair Service Curve for Best-Effort

PFQ algorithms provide only a rate based guarantee, which results in a coupling of bandwidth and delay allocation. By using non-linear service curves, FSC has the ability to decouple the allocation of bandwidth and delay. In the context of best effort traffic, we use FSC to improve the performance of delay sensitive bursty applications without affecting throughput oriented applications. For simplicity, in this study we consider two-piece linear concave service curves, as shown in Figure 2. By using concave service curves, short bursts will primarily be served according to the first slope (m_1), while longer transmissions will be served mostly according to the second slope (m_2).

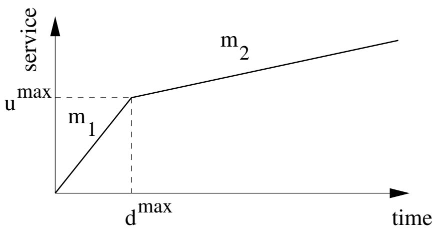


Figure 2: Concave Service Curve

For best effort traffic, the absolute values of m_1 and m_2 are not important, as they specify only the *relative* service priorities between bursts of size less than u^{max} and the continuously backlogged traffic in the system. We denote the ratio m_1/m_2 as the *Burst Preference Ratio* (BPR) and u^{max} as the *Preferred Burst Size* (PBS). Whenever a flow has been idle for a relatively long period of time, it will have its first PBS of traffic served at a rate that is BPR times higher than the existing backlogged traffic. In effect, FSC has "memory" to differentiate between delay sensitive bursty flows from continuously active, throughput oriented flows.

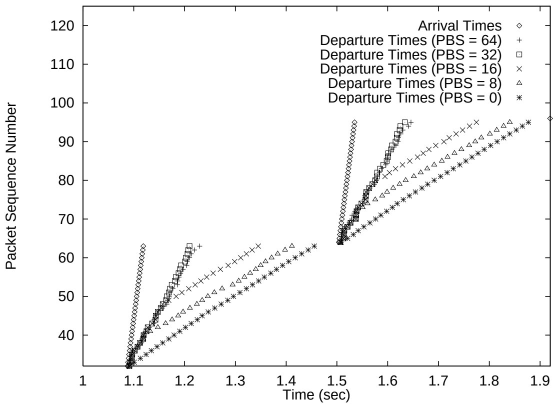


Figure 3: The packet arrival and departure times of a bursty flow for various service curves.

To give some intuition of FSC's behavior, consider a link shared by 15 constant-bit-rate UDP flows and one ON-OFF flow with a burst size of 32 packets. Figure 3 plots the arrival and departure times for each packet belonging to two consecutive burst periods of the ON-OFF flow. For a fixed BPR of 5, the plot shows the impact of the PBS (measured in the number of packets) on the departure times, and implicitly on the packet queueing delay, which is given by the horizontal distance between a packet's arrival time and its departure time. Note that the packet departure times follow accurately the shape of the service curve associated with the flow, decreasing as PBS increases.

To illustrate the performance of FSC with realistic traffic, we generate a synthetic workload of FTP traffic, whose flow lengths are chosen to model Internet flow lengths [1], for a 1 minute simulation over a 10Mbps link with a flow arrival rate corresponding to 95% of the link capacity. We begin with a cumulative distribution of Internet traffic based on flow lengths, wherein, given a flow length l, the distribution indicates the corresponding percentage of Internet traffic contributed by all flows of length less than l. From this distribution, we compute the average flow length of each 10 percentile group (in terms of traffic contribution) and generate flows of these lengths. Figure 4 plots the mean flow transfer time for the 8th flow length group (length = 10,910 Bytes) across a range of PBS and BPR values.

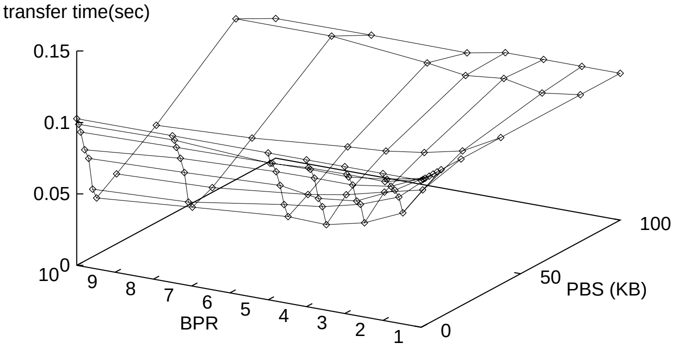


Figure 4: Transfer time for 70 to 80th Percentile Flows

Note that all points with PBS = 0 and/or BPR = 1 correspond to PFQ. While a flow's delay is minimal when its length corresponds to the PBS, minimal improvements are seen with BPR greater than 4. The larger the PBS, the higher the percentage of flows that are entirely covered by the first slope and the performance returns to that of PFQ. For this simulation set, setting BPR = 4 and PBS = 6000 Bytes reduces transfer times of most groups (some by over 50%) while only increasing the transfer time of the largest group by 1%.

References

[1] A. Feldmann, J. Rexford, and R. Caceres. Efficient policies for carrying web traffic over flow-switched networks. *IEEE/ACM Transactions on Networking*. To Appear.

[2] T.S.E. Ng, D.C. Stephens, I. Stoica, and H. Zhang. Supporting Best-Effort Traffic with Fair Service Curve, 1999. http://www.cs.cmu.edu/~hzhang/fsc/.

[3] I. Stoica, H. Zhang, and T.S.E. Ng. A Hierarchical Fair Service Curve algorithm for link-sharing, real-time and priority services. In *Proceedings of the ACM-SIGCOMM 97*, Cannes, France, August 1997.