A simulation Study of Fair Queueing and Policy Enforcement

James R. Davin and Andrew T. Heybey
MIT Laboratory for Computer Science

1 Introduction

This paper describes the results of research into algorithms for realizing improved control over the distribution of router and link resources among the users of a packet-switched network. It reports the results of simulation experiments designed to evaluate — in the context of the current Internet architecture and protocols — the effectiveness of various router algorithms as tools for enforcing resource allocation policies. Two variants on the fair queueing algorithm described in [1] are compared (in simulation experiments) to the familiar first-come-first-served (FCFS) discipline as mechanisms for enforcing a range of bandwidth allocation policies. The variations on fair queueing studied here represent innovations designed to simplify implementation in typical router systems. The results reported here suggest that a fair queueing service discipline enforces both uniform and non-uniform resource allocation policies better than does the FCFS discipline.

2 Definitions

- The term *resource allocation policy* is used in this discussion to refer to a static apportionment of network bandwidth and buffering capacity among some specified set of network user classes.
- The term *user class* is used in this discussion to refer to a set of human beings, application processes, or hosts among which the relative apportionment of network resources is not addressed by a given resource allocation policy. One example of a user class is the application process at one end of a single TCP connection; another example is the collection of all communicating processes and hosts within a particular corporation or government agency.
- The term *uniform allocation policy* is used in this discussion to refer to a resource allocation policy by which all identified user classes of a network are afforded equal shares of the network capacity.
- The term *non-uniform allocation policy* is used in this discussion to refer to a resource allocation policy by which all identified user classes of a network are not afforded equal shares of the network capacity.

3 Assumptions

The assumptions underlying this study are several.

- Resource allocation policies should be enforced without changes to the current Internet architecture.
- Enforcement of resource allocation policies should afford adequate service to network users (e.g., Slow-Start TCP connections) that adapt in cooperative ways to conditions within the network.
- Effective enforcement of resource allocation policies can not depend upon universally cooperative behavior from network users.

  Indeed, some user classes may be nasty; some user classes may be aggregates of TCP connections and other traffic for which the architecture affords no effective control mechanism.
- Enforcement of resource allocation policies should not entail denial of otherwise unused network resources to any user class.

  Policy enforcement that precludes any use of network resources in excess of the assigned share is realized in a straightforward way by time division multiplexing techniques. Neither these techniques nor this restrictive mode of policy enforcement is addressed by this study.
- The identity of the network user class associated with any particular Internet datagram is efficiently computable in real time from information carried in the datagram itself.

- The space resources of deployed routers are adequate to the largest possible number of simultaneously active user classes.

- Processor resources at an Internet router are more than adequate to any network load.

  This "assumption" may be in fact a logical requirement of effective policy enforcement. Unless a router enjoys sufficient processing capacity both to compute the user class identity for each incoming packet and and to compute the relative entitlement of said user class to occupy additional buffer space, then the (resource allocation) decision to buffer or discard an arriving packet is necessarily independent of the resource allocation policy.

4 Algorithms

This section describes each of the three algorithms evaluated in simulation experiments.

4.1 FCFS Discipline

First come, first served is the service discipline used by the current Internet architecture. Packets are queued for service at an output interface in the order in which they arrive. If the router runs out of buffer space, arriving packets are dropped.

4.2 Fair Queueing Discipline Without Punishment

This service discipline is based on [1], and is abbreviated FQNP (fair queueing, no punishment). FQNP provides an approximation to bit-wise round-robin service. Conceptually, each user class has its own queue. The bits in the queues are sent in a round-robin order. If a user class has no data in its queue, it is skipped during the current round. Such a scheme provides almost perfectly fair service.

Of course, packet fragmentation (especially into individual bits) is not a viable strategy in the real world. Instead, the algorithm that is proposed in [1] computes the bit round during which the last bit of each packet would be sent if the router were actually sending packets bit-wise round-robin. Packets from all user classes are placed on one output queue ordered by this finishing bit round number. We believe that this approximation is never unfair by more than the number of bits in a maximally sized packet.

However, even computation of the finishing bit round in this way may be difficult in real-world router implementations, for it requires knowledge of the bit round in progress at the time of the packet arrival. In many router implementations, precise reckoning of the current bit round while transmission of a packet is in progress is either difficult or impossible. One solution to this dilemma is to approximate the current bit round in progress (required for computation of the finishing bit round for the arriving packet) by the bit round in which the current (or most recent) packet transmission finishes. This modification to the original fair queueing algorithm is the basis for this simulation study, and we believe that the short-term unfairness introduced by this approximation is similar in magnitude to that introduced in the original algorithm by its departure from strictly bit-wise multiplexing of the link.

The simulated algorithm enforces a policy giving different user classes unequal shares of the output bandwidth by dividing the length of each packet by the number of shares of its user class before performing the above computations. Thus a user class with twice as many shares as another could send twice as many bits in a given time interval.

If the router runs out of buffers, the last packet in the output queue from the user class with the most packets in the queue is dropped. When a packet is dropped, the original algorithm of [1] enqueues future packets from that user class as if the dropped packet had actually been transmitted. A user class that consistently sends packets faster than they are serviced finds its packets placed farther and farther toward the end of the queue as punishment. In constrast, the FQNP algorithm orders future packets from that user class in the queue as if the dropped packet had never arrived (thus the "no punishment" name). Punishment is inappropriate in the situation being studied for the following reasons:

- Punishment for excessive burstiness can preclude forward progress for protocols (such as the Sun NFS protocols) that involve multi-packet operations.

- If many individual conversations are aggregated to constitute a single user class, then bad behavior on the part of one such conversation can result in denial of service for all the flows in the aggregate.

4.3 Fair Queueing Discipline with Fixed Quota

The fair queueing, fixed quota (FQFQ) service discipline is the fair queueing algorithm described above (including the approximate reckoning of the current bit round) with a different method of buffer management. Each user class is allocated a fixed, equal number of the buffers available in the router. An arriving packet from a network user class is discarded if that user class is already occupying all of its buffers, regardless of the actual number of free buffers in the router. The motivation for this algorithm is to decouple the user classes from one another (a burst of packets from one user class will not cause packets from another user class to be discarded) while at the same time simplifying implementation of the buffer management scheme. The implementation of the buffer management is much simpler because the decision whether to accept or drop a packet is reduced to a simple comparison of the number of packets belonging to the user class against a fixed limit, and the packet dropped (if any) is the one that just arrived. Without the fixed quota mechanism, significant effort is required in order to identify the currently buffered packet that is to be discarded and to detach it from the various data structures that may refer to it. Because efficient operation may require that packets appear in a number of ordered data structures, the cost of packet discard in the canonical algorithm is significantly more than in the FQFQ algorithm.

5 Simulation Environment

The simulated network environment in which the algorithms are studied comprises a pair of IP routers connected by a single, megabit-per-second serial link. In addition, four similar links connected to each router convey the packets of one of four network user classes into and out of the network, to and from a variety of traffic sources and sinks. In each of the experiments reported here, all interfaces in all routers of the simulated network are managed identically according to one of the three considered algorithms. The routers are modeled as having infinite processing capacity.

An example of the simulated topology is represented in Figure 1. Infinite processing resources are attributed to each of the "host" components in the figure, so that each of the TCP conversations in the simulation runs in parallel. Unless otherwise specified, all links in the simulated network enjoy zero propagation delay and zero probability of packet corruption.

Two kinds of traffic source are modelled: a "well-behaved," simplex TCP connection that behaves according to the slow-start congestion control discipline described in [2], and an "ill-behaved," simplex TCP connection that retransmits its entire window in the event of packet loss. Unless otherwise specified, per-segment processing time for all TCP entities is modeled as 400 microseconds, with uniformly distributed random perturbations of plus-or-minus 40 microseconds. All TCPs generate 1000 octet segments.

6 Experimental Design

The behavior of each of the three algorithms studied is observed in each of twelve experimental scenarios. In each scenario, each of four user classes competes for network resources by generating traffic in the form of multiple TCP transfers. The traffic generated by each user class is observed for 75 seconds, as is the utilization of shared network resources for each user class. These eight quantities are sampled at 4 millisecond intervals, and the value of these parameters at each sampling interval is smoothed by averaging with all corresponding samples from the preceding 800 milliseconds.

Each of the twelve experimental scenarios is characterized according to type of allocation policy, type of demand, and variety of delay.

6.1 Policy Characterization

The resource allocation policy in effect for each scenario is identified by one of following terms:

- The term *uniform allocation policy* is applied to scenarios in which all user classes are afforded equal shares of network resources. In this discussion, such a policy is sometimes expressed as "1 1 1 1."

- The term *non-uniform allocation policy* is applied to scenarios in which the most privileged user class is afforded three times the network resources afforded to the least privileged user classes, and another user class is afforded twice the share of the least privileged user classes. In this discussion, such a policy is sometimes expressed as "3 2 1 1."

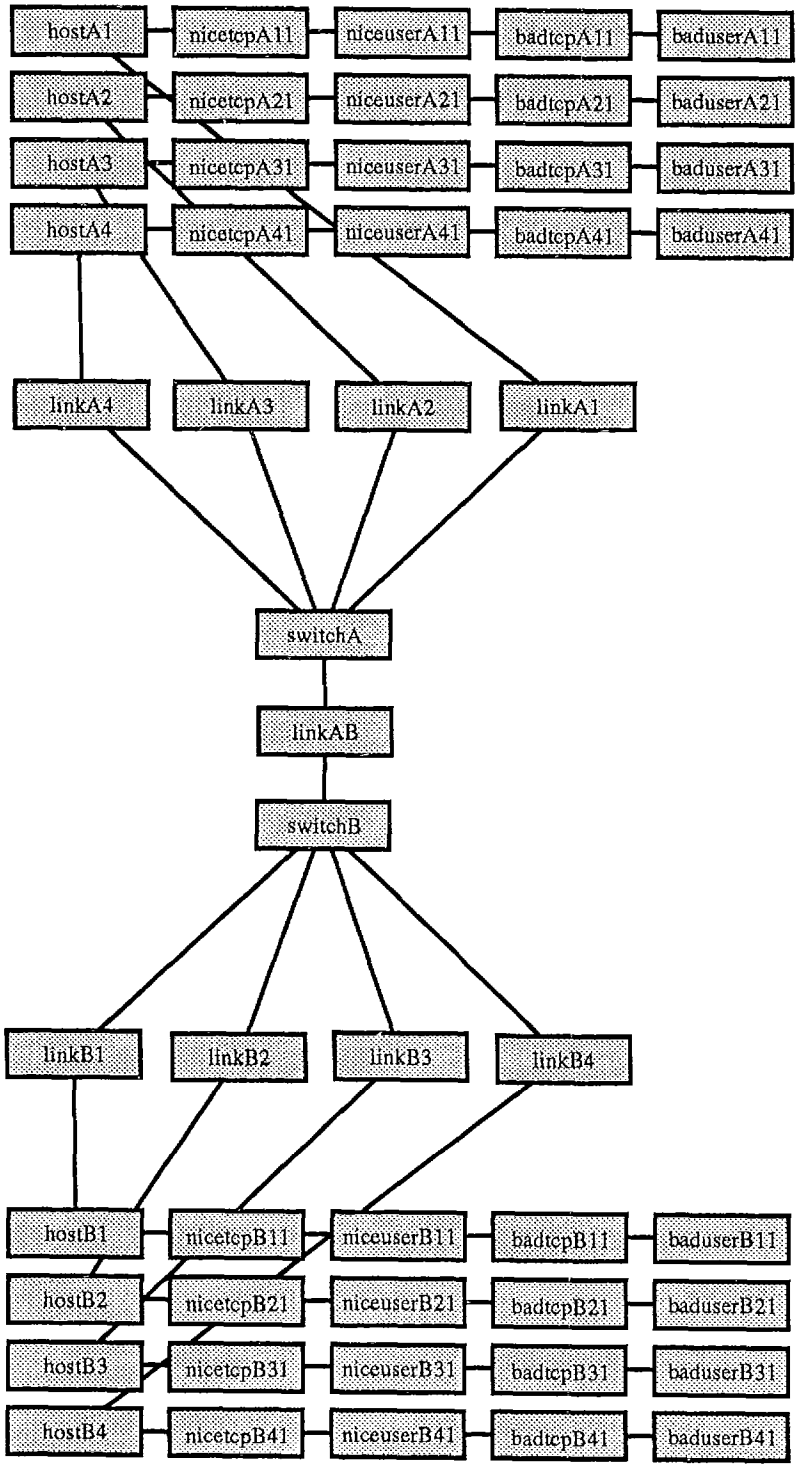

Figure 1: Example Topology

6.2 Demand Characterization

The generated traffic load for each scenario is identified by one of the following terms:

- The term *homogeneous dynamic demand* is applied to scenarios in which, for each user class, 18 well-behaved TCPs and 6 ill-behaved TCPs begin transmitting at random moments in the first 60 seconds of the simulation. Each such TCP transfers 80000 octets of data as quickly as possible and then ceases transmission.
- The term *conforming static demand* is applied to scenarios in which, for each user class, both the number of well-behaved TCPs and the number of ill-behaved TCPs is commensurate with the relative resource share afforded to that user class by the specified resource allocation policy. In scenarios with a uniform allocation policy (1 1 1 1), 6 well-behaved TCPs and 2 ill-behaved TCPs generate the traffic for each user class. In scenarios with a non-uniform allocation policy (3 2 1 1), 18 well-behaved TCPs and 6 ill-behaved TCPs generate traffic for the most privileged user class. Similarly, 12 well-behaved and 4 ill-behaved TCPs generate traffic for the next most privileged user class, and 6 well-behaved and 2 ill-behaved TCPs generate traffic for the least privileged user classes. In scenarios with this type of demand, all TCPs continually transfer data as quickly as possible.
- The term *non-conforming static demand* is applied to scenarios in which, for each user class, both the number of well-behaved TCPs and the number of ill-behaved TCPs is *not* commensurate with the relative resource share afforded to that user class by the specified resource allocation policy. In scenarios with a uniform allocation policy (1 1 1 1), 18 well-behaved TCPs and 6 ill-behaved TCPs generate traffic for one user class, and 6 well-behaved and 2 ill-behaved TCPs generate traffic for the remaining user classes. In scenarios with a non-uniform allocation policy (3 2 1 1), 18 well-behaved TCPs and 6 ill-behaved TCPs generate the traffic for each user class. In scenarios with this type of demand, all TCPs continually transfer data as quickly as possible.

6.3 Delay Characterization

The distribution of network delay in each scenario is identified by one of the following terms:

- The term *homogeneous delay* is applied to scenarios in which the traffic generated for each user class suffers propagation delay identical to that suffered by any other user class. In particular, traffic from all user classes suffers zero propagation delay.
- The term *heterogeneous delay* is applied to scenarios in which the traffic generated by each user class suffers propagation delay that differs from that suffered by any other user class. In particular, traffic for one user class experiences no propagation delay; a second user class experiences 200 milliseconds of delay; a third user class experiences 400 milliseconds of delay; and the fourth user class experiences 600 milliseconds of delay.

7 Discussion of Results

Long-term utilization of shared resources for each user class is the obvious measure of policy enforcement in the context of a long-term enforcement model. These figures are presented for each studied algorithm in each experimental scenario in Tables 1 and 2. The figures represent utilization over the entire 75 second period of each simulation, and, thus, in general, they represent average measures across periods of network underload and overload.

7.1 Conforming Demand Case

In the case of a demand that conforms to the specified allocation policy, the FQNP and FQFQ algorithms enforce the specified policy over the long term slightly better than does the FCFS algorithm. The divergence in the FCFS utilization figures is attributable to the positive feedback effect by which a fortuitously bursty TCP (e.g. an ill-behaved TCP) may acquire a continually inordinate share of the bandwidth at the expense of other TCPs. To the extent that the frustrated TCPs tend to synchronize (by virtue of the shared buffer resources), the fortuitous TCP may enjoy relatively long periods in which there is little competition for its dominant position. For the FCFS algorithm in the case of heterogeneous delay, the divergence in utilizations tends to be patterned according to the distribution of network delay among the user classes. This effect is explained by the ability of TCPs who enjoy lower network delay (and, accordingly, lower control delay) to be more aggressive in exploiting transient excesses of network resources.

Policy	Demand	Measurement	FQNP	FQFQ	FCFS
Uniform 1 1 1 1	Homogeneous Dynamic	User 1 Util %	20.8	20.8	21.3
		User 2 Util %	21.0	20.9	21.3
		User 3 Util %	20.6	20.7	21.6
		User 4 Util %	20.9	20.9	21.5
	Conforming Static 1 1 1 1	User 1 Util %	24.9	24.9	25.4
		User 2 Util %	24.8	24.9	23.8
		User 3 Util %	24.9	24.9	23.1
		User 4 Util %	24.9	24.9	27.1
	Non-Conforming Static 3 1 1 1	User 1 Util %	24.8	24.7	48.3
		User 2 Util %	24.9	24.9	19.0
		User 3 Util %	24.9	24.9	16.5
		User 4 Util %	24.9	24.9	15.7
Non-Uniform 3 2 1 1	Homogeneous Dynamic	User 1 Util %	20.6	20.6	21.3
		User 2 Util %	21.0	21.1	21.3
		User 3 Util %	20.6	20.7	21.6
		User 4 Util %	20.8	21.2	21.5
	Conforming Static 3 2 1 1	User 1 Util %	42.6	42.6	40.7
		User 2 Util %	28.2	28.1	28.0
		User 3 Util %	14.3	14.3	14.3
		User 4 Util %	14.3	14.3	15.5
	Non-Conforming Static 3 3 3 3	User 1 Util %	42.4	42.8	24.3
		User 2 Util %	28.5	28.4	24.2
		User 3 Util %	14.3	14.1	25.0
		User 4 Util %	14.3	14.1	25.4

Table 1: Results for Homogeneous Delay

The analysis in the case of a non-uniform allocation policy is the same except that the disparity among user classes in the conforming, non-uniform demand tends to magnify the divergence in the utilization measurements.

7.2 Non-Conforming Demand Case

In the case of demand that does not conform to the specified allocation policy, the FQNP and FQFQ algorithms enforce the specified policy over the long term much better than does the FCFS algorithm. Utilization measures for the latter reflect the prevailing demand much more than they reflect the desired policy. Moreover, in the case of a non-uniform policy, the utilization measures for the FCFS algorithms suffer from the "divergence" effect described above — here compounded by the significant number of ill-behaved TCPs. The utilization measures are thus particularly skewed by the domination of a few TCPs.

Again, too, in the case of heterogeneous delay, the divergence is shaped by the delay distribution: TCPs that enjoy lower delay more aggressively exploit shared network resources.

The long-term effectiveness of the FQNP and FQFQ algorithms is roughly comparable.

7.3 Dynamic Demand Case

In the case of homogeneous dynamic demand and a uniform allocation policy, all three algorithms appear, over the long term, to enforce the specified policy. FCFS does as well as the others because the statistically uniform demand produces long-term uniform utilizations. FCFS is successful even in the case of a non-uniform policy because the light network load admits satisfaction of all demands in the long term.

8 Conclusion

The simulation results reported here suggest that the introduction of a fair queueing service discipline into Internet routers enforces both uniform and non-uniform resource allocation policies better than does the tradi-

Policy	Demand	Measurement	FQNP	FQFQ	FCFS
Uniform 1 1 1 1	Homogeneous Dynamic	User 1 Util %	21.4	20.9	21.6
		User 2 Util %	21.0	20.8	21.1
		User 3 Util %	21.0	21.2	20.8
		User 4 Util %	21.4	21.2	20.5
	Conforming Static 1 1 1 1	User 1 Util %	25.0	25.2	42.9
		User 2 Util %	24.9	25.2	23.1
		User 3 Util %	25.0	24.7	18.4
		User 4 Util %	24.5	24.4	15.1
	Non-Conforming Static 3 1 1 1	User 1 Util %	25.1	25.0	70.0
		User 2 Util %	24.8	25.2	11.4
		User 3 Util %	24.9	24.9	10.5
		User 4 Util %	24.7	24.4	7.6
Non-Uniform 3 2 1 1	Homogeneous Dynamic	User 1 Util %	21.0	20.7	21.6
		User 2 Util %	20.9	21.1	21.1
		User 3 Util %	22.1	21.3	20.8
		User 4 Util %	21.0	20.4	20.5
	Conforming Static 3 2 1 1	User 1 Util %	42.7	42.5	63.0
		User 2 Util %	28.4	28.5	20.7
		User 3 Util %	14.2	14.3	8.7
		User 4 Util %	14.2	14.2	7.0
	Non-Conforming Static 3 3 3 3	User 1 Util %	42.7	42.4	42.3
		User 2 Util %	28.4	28.5	24.1
		User 3 Util %	14.2	14.3	20.2
		User 4 Util %	14.2	14.3	12.8

Table 2: Results for Heterogeneous Delay

tional FCFS service discipline. Moreover, the two fair queueing strategies enforce specified bandwidth allocation policies with comparable effectiveness.

This study does not address the effect of these algorithms upon the distribution of the delay experienced by user classes of the network. A compelling intuitive argument suggests that the mean delay afforded by each of the three algorithms is likely to be identical because the average delay is not influenced by reorderings of the service queue. The variance of the delay distribution afforded by each of these algorithms is an area for further study.

This study also does not address the behavior of these algorithms in topologically complex networks, nor does it examine the relative performance of individual TCP connections within a single user class aggregate.

Finally, while the FQFQ algorithm is appealing in its simplicity and in its capacity for effective policy enforcement, its isolation of the buffer pools for each user class may have the effect of reducing overall link utilization when a lone user wishes to make heavy demands on an otherwise idle network. A complete evaluation of the FQFQ algorithm should properly examine this aspect of its behavior.

References

[1] Alan Demers, Srinivasan Keshav, and Scott Shenker. *Analysis and Simulation of a Fair Queueing Algorithm*, In **Proceedings of SIGCOMM '89**, pages 1–12, September 1989.

[2] Van Jacobson. *Congestion Avoidance and Control*, In **Proceedings of SIGCOMM '88**, August 1988.