Fair Queueing with Feedback-Based Policing

Célio Albuquerque[†], Tatsuya Suda[†] and Brett J. Vickers[‡]

[†]Dept. Of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
949-824-3097 (phone)
949-824-2886 (facsimile)

[‡]Department of Computer Science
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08854-8019
732-445-1496 (phone)
732-445-0537 (facsimile)

Technical Report 99-26

Abstract

End-to-end congestion control is an important reason why the Internet is robust, scalable and simple to use. Unfortunately, purely end-to-end congestion control algorithms are incapable of preventing the unfair bandwidth allocations and congestion collapse caused by unresponsive applications, which are becoming increasingly prevalent in the Internet. In this paper, we propose a new mechanism called Fair Queueing with Feedback-based Policing (FQFP) to address unfair bandwidth allocation and congestion collapse in the Internet. We demonstrate the promise of FQFP through simulations and suggest ways in which FQFP may leverage the mechanisms currently being developed in the context of differentiated services. The FQFP mechanism is compliant with the Internet philosophy of keeping router implementations simple and pushing complexity toward the edges of the network.

1 Introduction

The essential philosophy behind the Internet is expressed by the scalability argument: no protocol, algorithm or service should be introduced into the Internet if it does not scale well. A key corollary to the scalability argument is the end-to-end argument: to maintain scalability, algorithmic complexity should be pushed to the edges of the network whenever possible. Perhaps the best example of the Internet philosophy is TCP congestion control, which is achieved primarily through algorithms implemented at end system hosts. Unfortunately, TCP congestion control is also an illustration of some of the shortcomings of the end-to-end argument. As a result of its strict adherence to end-to-end congestion control, the current Internet suffers from *unfair bandwidth allocations* between competing traffic flows and potential *congestion collapse*.

Unfairness in the Internet arises for a variety of reasons. Flows that are adaptive to congestion (e.g., TCP flows) may be restricted to a small fraction of bottleneck link bandwidth when competing with unresponsive[1] or malicious flows, which are becoming disturbingly prevalent in the present Internet. TCP also introduces unfairness by allocating a disproportionately large amount of bandwidth to flows with shorter round-trip times.

Related to the unfairness problem is the problem of congestion collapse due to undelivered packets [8]. This form of congestion collapse occurs when bandwidth is continuously wasted on the transport of packets that are dropped before reaching their ultimate destination. Unresponsive flows are the main cause of this kind of congestion collapse.

In this paper, we present and preliminarily evaluate a mechanism called Fair Queueing with Feedback-based Policing (FQFP), which aims to prevent congestion collapse and achieve fair bandwidth allocation for best effort traffic. The basic idea of FQFP is to perform fair queueing on traffic flows and to use feedback-based traffic policing between edge routers to prevent congestion collapse. The FQFP mechanism does not compromise the Internet design philosophy, because core routers at the interior of the network perform only simple operations al-

* This research is supported by the National Science Foundation through grant NCR-9628109. It has also been supported by grants from the University of California MICRO program, Hitachi Ltd., Hitachi America, Tokyo Electric Power Company, Nippon Telegraph and Telephone Corporation (NTT), Nippon Steel Information and Communication Systems Inc. (ENICOM), Matsushita Electric Industrial Company, and Fundação CAPES/Brazil.

[1] An *unresponsive* flow is any flow that fails to reduce its rate in response to increased packet discarding due to congestion [8].

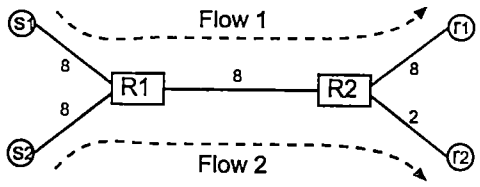


Figure 1: Example of an unfair bandwidth allocation created by congestion collapse.

ready being proposed and implemented in other contexts, and congestion control algorithms at the end systems may remain unchanged.

2 Background

Per-flow scheduling mechanisms like Weighted Fair Queueing (WFQ) [2] attempt to offer fair allocations of bandwidth to flows contending for the same link. These mechanisms are typically more complex to implement than traditional FIFO scheduling, because they require the maintenance of per-flow state in every router. Despite its higher implementation complexity, several vendors have implemented WFQ in their routers and switches, and the implementation cost of maintaining state and performing per-flow scheduling is gradually decreasing. We must emphasize, though, that WFQ does not achieve *globally max-min fair* allocations of bandwidth, and thus it cannot prevent congestion collapse by itself.[2] Figure 1 shows an example in which two unresponsive flows with equal weights compete for a shared bottleneck link. The available bandwidth on each link is displayed in the figure, and both flows unresponsively transmit at a fixed rate of 6 Mbps. At the first router, WFQ ensures that each flow receives 4 Mbps of the bottleneck link's available bandwidth. At the second router, packets from flow 2 are dropped. Hence, flow 1 achieves a throughput of 4 Mbps and flow 2 achieves a throughput of 2 Mbps. We say that congestion collapse has occurred, because flow 2 packets, which were ultimately discarded at the second router due to congestion, limited the throughput of flow 1. A globally max-min fair allocation of bandwidth of 6 Mbps for flow 1 and 2 Mbps for flow 2 would have prevented congestion collapse in this case.

Stoica, Shenker and Zhang propose an approximation of WFQ called Core-Stateless Fair Queueing (CSFQ) with the significant advantage that only edge routers are required to maintain per-flow state [1]. Core routers operate only on aggregate flows. While CSFQ achieves an approximately fair allocation of bandwidth between competing flows at each router, like WFQ it suffers from an inability to produce globally max-min fair allocations of bandwidth, and thus it cannot prevent unresponsive flows from creating the congestion collapse problem either.

Jain *et al.* propose several closed loop explicit rate-based approaches to flow control (e.g., ERICA, ERICA+) for the ATM Available Bit Rate (ABR) service in which all network switches compute fair allocations of bandwidth among competing connections [6]. This approach is able to achieve globally max-min fair bandwidth allocations, but it does not enforce the responsiveness of applications. Malicious users may ignore the explicit rates reported by the network, and so these solutions may also fail to prevent the congestion collapse problem. Furthermore, if implemented in the Internet, explicit rate-based algorithms such as ERICA would require that relatively complex rate monitoring algorithms be uniformly present in routers, and this is not likely to happen.

Floyd and Fall [8] identify congestion collapse as a serious problem in the Internet and propose network mechanisms that encourage the use of adaptive or "TCP-friendly" end-to-end congestion control. We agree with the idea of providing applications with incentives to become more responsive to congestion. However, the approach taken in [8] requires that routers perform the complex task of monitoring and filtering unresponsive and malicious flows. Another limitation of this approach is that the procedures used to identify unresponsive flows are somewhat arbitrary and not always successful [1].

Our approach, which we call Fair Queueing with Feedback-based Policing (FQFP), differs significantly from these other approaches in the sense that its goal is to prevent congestion collapse and achieve approximate global max-min fairness by filtering out malicious traffic at edge routers.

[2] At a given router, an allocation of bandwidth is said to be globally max-min fair if all active flows not bottlenecked at another router are allocated an equal share of the available bandwidth [7].

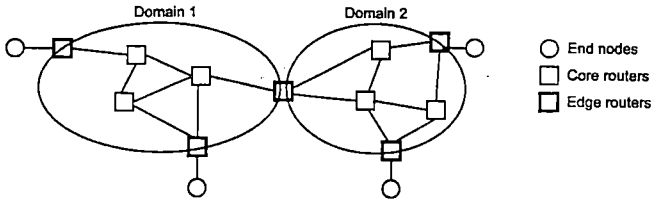


Figure 2: Internet architecture assumed by FQFP.

3 Fair Queueing with Feedback-based Policing

In this paper, we follow the lead of Stoica, Shenker and Zhang [1], who identify a contiguous region of the network as an *island* of routers and distinguish between routers at the island's edges and core. We draw a further distinction between two types of edge routers. Depending on which flow it is operating on, an edge router may be viewed as an *ingress* or *egress* router. An edge router operating on a flow passing *into* the island of routers is considered an ingress router, whereas an edge router operating on a flow passing *out* of the island is considered an egress router.[3]

FQFP may be implemented using any variant of fair queueing. For instance, it may be implemented using WFQ in all core routers, or it may just as easily be implemented in conjunction with CSFQ, which maintains per-flow state at edge routers and uses estimates of aggregate traffic at core routers. The primary difference between FQFP and these other fair queueing approaches is in the function that edge routers perform. FQFP achieves approximate global max-min fairness and prevents congestion collapse by implementing feedback-based traffic policing. Egress routers monitor the rate of each flow and periodically report each flow's fair bandwidth allocation to the ingress router. Optionally, the feedback may also contain a congestion indication determined through an explicit congestion notification (ECN) [3][4] mechanism. Upon receiving the congestion feedback, ingress routers police and shape the incoming traffic and filter packets from malicious flows.

With the FQFP mechanism, unresponsive or malicious applications have their traffic regulated and dropped by ingress routers, thereby preventing congestion collapse within the network. Furthermore, the FQFP mechanism allocates bandwidth to competing flows in an approximately max-min fair fashion regardless of whether the applications are adaptive or unresponsive.

3.1 Required behavior at the core routers

To ensure fairness within an island of routers, the FQFP mechanism expects that all core routers will perform some type of fair queueing (e.g., WFQ, CSFQ). Fortunately, the effort to provide differentiated services is leading many network equipment vendors to enhance routers with precisely the type of scheduling mechanisms expected by FQFP. The advent of differentiated services offers us an opportunity to address an increasingly difficult problem in Internet traffic control.

Even if core routers do not implement fair queueing, we expect FQFP to provide relief from congestion collapse and unfair bandwidth allocation. Edge routers can still provide useful congestion feedback to ingress routers without fair queueing in the core routers; the result will simply be less optimal: congestion collapse is prevented but max-min fair allocations are not achieved.

3.2 Monitoring flows and generating feedback at the egress router

Egress routers determine fair bandwidth allocations by monitoring the arrival rate of each flow. When fair queueing is used in core routers and rate adaptation is performed at the ingress router, the arrival rate at the egress router is an indication of the global max-min fair share. If WFQ is used in core routers, egress routers determine the arrival rate by monitoring a flow's packet arrivals and applying a rate estimation algorithm such as Time Sliding Window [10]. In the case of a CSFQ core router, which attaches to each packet a label containing the outgoing flow rate, the process of determining a flow's fair share allocation is even simpler. The egress router simply reads one of the flow's packet labels.

---

[3] Note that a flow may pass through more than one egress (or ingress) router if the end-to-end path crosses multiple *islands* of routers. Typically an island will correspond to an administrative domain.

Once the egress router has determined a flow's fair share allocation, it periodically reports the value (along with an optional congestion indication bit generated by ECN) in a special feedback packet to the flow's ingress router. The simplest way to do this is to send the feedback with the Internet control message protocol (ICMP) toward the flow's ingress router. To lessen feedback overhead, feedback from multiple flows to a single ingress router may be aggregated into a single ICMP packet. Optionally, TCP acknowledgement packets may be used to indicate the fair share allocation to the sender, which can use this information to adapt its transmission rate.

3.3 Policing flows at the ingress router

The ingress router uses the feedback from egress routers to police and filter each flow. To regulate each flow's transmission rate, the ingress router uses a token bucket operating at the rate specified by the flow's most recent feedback packet. To capture unutilized bandwidth in the network, the ingress router periodically increments the flow's token generation rate in a TCP-friendly fashion. When ECN is available, the congestion notification supplied in a flow's feedback packet can be used to inform the rate-incrementing procedure. Upon receiving a feedback packet with a congestion notification bit set, the ingress router learns that the network is becoming congested and reduces the token generation rate to the fair allocation indicated in the flow's last feedback packet. Only when the network is relieved of congestion is the ingress router allowed to exploit newly available bandwidth by increasing the flow's token generation rate.

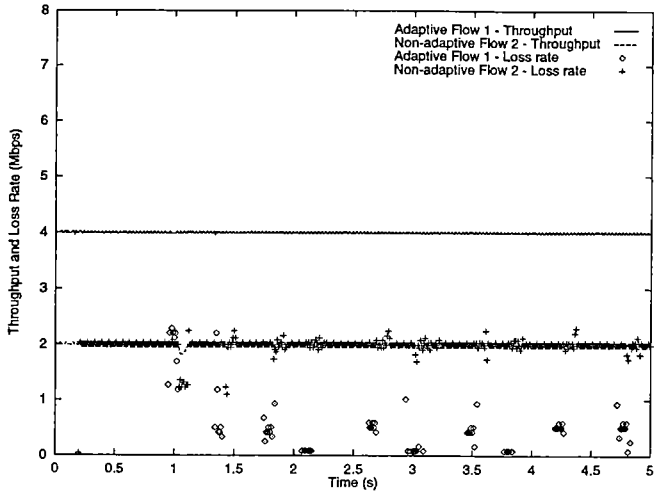
4 Performance

This paper asserts that globally max-min fair bandwidth allocations can be achieved and congestion collapse can be avoided by combining feedback-based policing at edge routers and some form of fair queueing within core routers. In order to provide evidence for this claim, we present the results of several simulations in which the following measures of performance are obtained: (1) bandwidth allocations and link utilization, (2) packet drop rate, and (3) convergence time to a fair state.

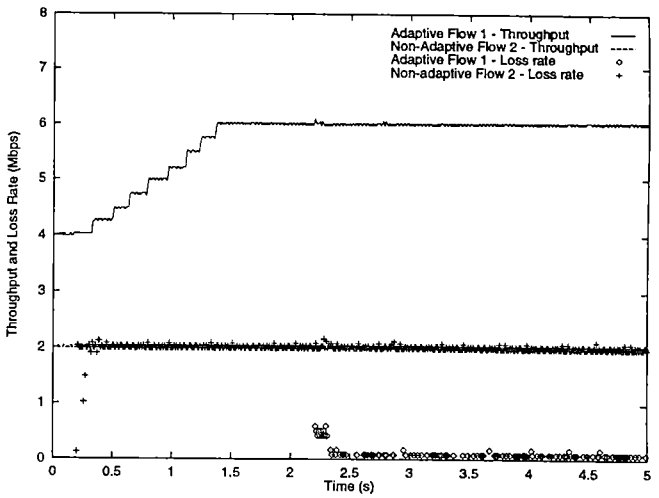
In these simulation experiments, we utilize a combination of adaptive and unresponsive flows. For the purposes of this analysis, we define an adaptive flow as any flow generated by an application that uses an end-to-end congestion control algorithm to adjust its transmission rate. An unresponsive flow is defined as any flow generated by an application that transmits at a fixed rate regardless of packet loss or congestion. The congestion control algorithm used by adaptive end system hosts is an extremely simple rate-based algorithm. It operates as follows: receivers monitor a flow's packet arrival and loss rates and periodically report them back to the sender; the sender transmits at the indicated rate and incrementally increases its transmission rate when no packet loss is being reported. In a sense, the end systems perform the same feedback operations as FQFP edge routers. Furthermore, some experiments are performed with a simple ECN scheme, in which routers mark packets when their buffer occupancies exceed a threshold.

In the first set of experiments, we use the network model depicted in Figure 1, where flow 1 is an adaptive flow, and flow 2 is unresponsive. We first consider the case in which routers perform only weighted fair queueing; no edge router policing or ECN is performed. The result is shown in Figure 3(a). Weighted fair queueing is clearly not sufficient to avoid congestion collapse caused by the unresponsive flow (flow 2). The adaptive flow (flow 1) achieves only 4 Mbps throughput due to flow 2's wasteful use of the bottleneck link. Moreover, the adaptive flow experiences a significant amount of loss due to attempts by source 1 to increase its transmission rate.

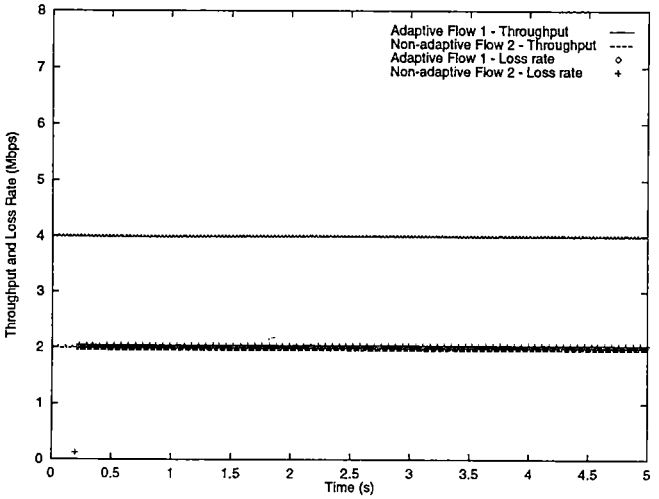
Next, we introduce FQFP and perform policing at the ingress router using bandwidth allocations reported by the receivers. In Figure 3(b), we see that this allows the adaptive flow to capture its globally max-min fair allocation of bandwidth (6 Mbps). The loss rate for the adaptive flow is also reduced, although it does not remain at zero.



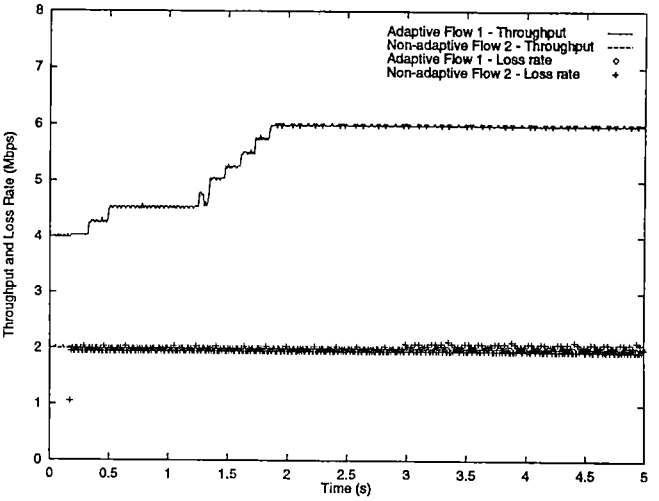
(a) WFQ only



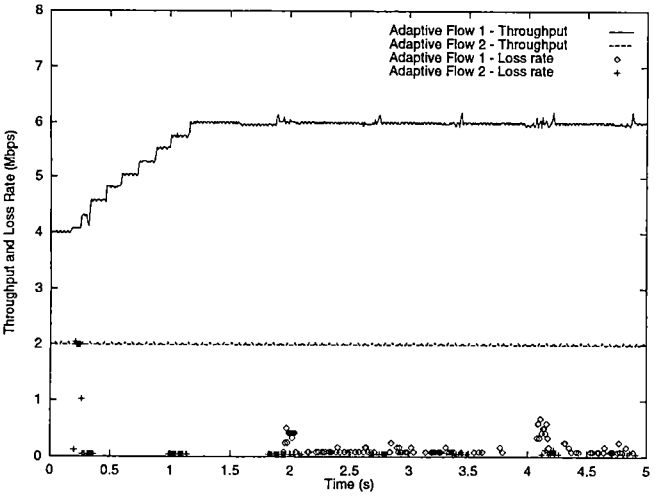
(b) FQFP only



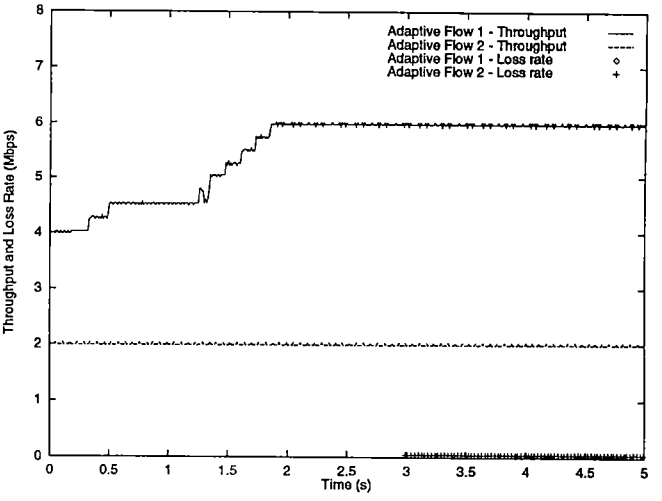
(c) WFQ with ECN



(d) FQFP with ECN



(e) FQFP only



(f) FQFP with ECN

Figure 3: Throughput and Loss Rate

Figures 3(c) and (d) show the result of adding ECN to WFQ and FQFP, respectively. Because ECN does not require a sender or an ingress router to wait for a packet loss to occur before determining that there is congestion, the loss rate for the adaptive flow falls to zero in both cases. As Figure 3(d) shows, the combination of FQFP and ECN provides the best performance in this set of simulations; global max-min fairness is achieved and packet loss for adaptive flows is minimized.

Figures 3(e) and (f) show the impact of FQFP with and without ECN when both flows 1 and 2 are adaptive. Again, global max-min fairness is achieved, and loss rates for both flows are near zero.

In another set of experiments, we use the second General Fairness Configuration (GFC-2) model [11][12] to further evaluate the ability of the FQFP mechanism to achieve global max-min fairness. (See Figure 4.) In this model, there are 22 competing sources, 22 receivers and 7 routers, and all links serve as bottlenecks for at least one of the 22 flows. We set the link propagation delay factor D to 5 msec and set all link capacities to a multiple of 50 Mbps. All entry and exit links have propagation delays of D and a capacity of 150 Mbps.

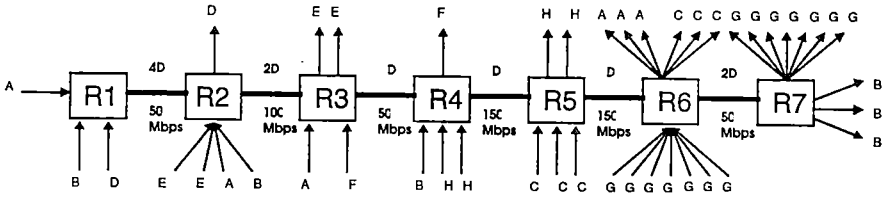


Figure 4: General Fairness Configuration 2 (GFC-2).

The first column of Table 1 lists the ideal max-min fair allocation of bandwidth for each type of flow in the GFC-2 configuration. Also listed are the observed bandwidth allocations for each flow after they have converged to a state of equilibrium. The table reflects the transmission rate to which a sample flow from each flow type converges. The architecture proved to be able to achieve bandwidth allocations that are very close to globally max-min fair.

Flow Type	Ideal max-min Fair share	FQFP Bandwidth Allocation
A	10	10.04
B	5	5.01
C	35	35.21
D	35	34.79
E	35	34.93
F	10	10.18
G	5	5.26
H	52.5	52.81

Table 1: Global max-min fair share and flows' actual bandwidth allocation.

We also evaluate how rapidly the flows converge to their equilibrium allocations. Figure 5 shows how the rate of a selected flow from each flow type converges over time. Even in a complex network model with propagation delays as large as this one's is, all flows converge to their optimal bandwidth allocation within a few seconds.

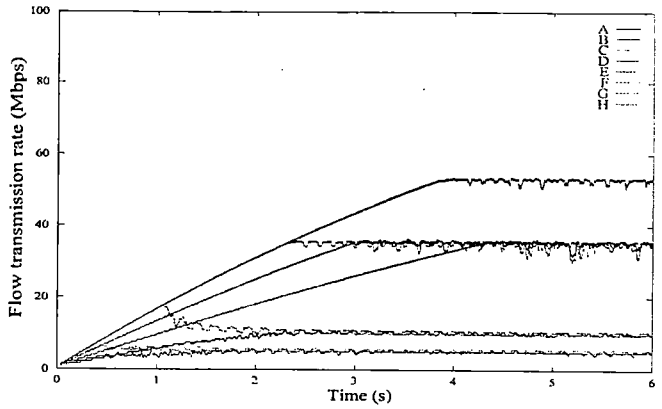


Figure 5 : Flow rates for GFC-2 configuration.

5 Conclusion

In this paper, we have proposed a comprehensive but simple solution to the problems of Internet congestion collapse, unresponsive flows, and unfair bandwidth allocation. Our solution relies on a combination of fair queueing in core routers and feedback-based traffic policing between edge routers.

We hasten to point out that introducing FQFP into the Internet can be facilitated by leveraging some of the efforts currently underway to evolve the Internet architecture toward a differentiated services model [9]. Under the differentiated services model, the Internet becomes capable of offering multiple classes of network service to customers and applications. Achieving differentiated services requires relatively new router functions, such as per-flow scheduling at edge routers and aggregate flow-based "per hop behaviors" at core routers. Fortunately, many of these functions can also be exploited to implement FQFP.

Although the simulations presented in this paper demonstrate the promise of the FQFP mechanism, further investigation using more complex and realistic network models are warranted. In future work, we intend to expand our models to include multi-domain internetworks, TCP hosts, and background traffic generated by non-best-effort network applications.

References

[1] I. Stoica, S. Shenker and H. Zhang, "Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks", in *Proc. of ACM SIGCOMM*, 1998.

[2] A. Demers, S. Keshav and S. Shenker, "Analysis and Simulations of a Fair Queueing Algorithm", in *Journal of Internetworking Research and Experience*, October 1990.

[3] S. Floyd, "TCP and Explicit Congestion Notification", in *Proc. of ACM SIGCOMM*, 1994.

[4] K. Ramakrishnan, "A Proposal to Add Explicit Congestion Notification (ECN) to IP", Internet RFC 2481, January 1999.

[5] "ATM Forum Traffic Management Specification, Version 4.0", ATM Forum Traffic Management Group, April 1996.

[6] R. Jain et al, "ERICA Switch Algorithm: A Complete Description", ATM Forum/96-1172, August 1996.

[7] D. Bartsekas and R. Gallagher, "Data Networks", second edition, Prentice Hall, 1987.

[8] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", submitted to *IEEE/ACM Transactions on Networking*, 1998.

[9] S. Blake et al, "An Architecture for Differentiated Services", Internet RFC 2475, December 1998.

[10] D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service", *IEEE/ACM Transactions on Networking*, August 1998.

[11] R. Simcoe, "Test Configurations for Fairness and Other Tests", ATM Forum 94-0557, July 1994.

[12] B. Vandalore, S. Fahmy, R. Jain , R. Goyal and M. Goyal, "A Definition of Generalized Fairness and its Support in Switch Algorithms", ATM Forum 98-0151, February 1998.