A Comparison of Feedback Based and Fair Queuing Mechanisms for Handling Unresponsive Traffic

Costin Iancu, Anurag Acharya
Dept of Computer Science, University of California Santa Barbara
{ciancu,acha}cs.ucsb.edu

Abstract

The stability of the current Internet depends on the end-to-end congestion control mechanism provided by TCP. Recently, popular multimedia applications (RealAudio, RLM) started using more aggressive forms of congestion control. It is not clear yet how wide uncooperation will become in Internet, but it seems dangerous to base the Internet congestion control solely on the assumption of end-host cooperation. In this paper we examine the impact of router based congestion control mechanisms in the presence of uncooperative traffic. We consider two classes of congestion control mechanisms (1) fair queuing mechanisms (FRED, DRR, CSFQ) and (2) feedback based mechanisms (ERAF, NBP). Within the subclass of feedback based mechanisms we distinguish between binary feedback mechanisms (ERAF) and rate feedback mechanisms (NBP). The criteria we use for our comparison are: efficiency, fairness, convergence and scalability. Our results indicate that the feedback based mechanisms are able to shield responsive traffic competing with uncooperative traffic on a single link with the same efficiency as the fair queuing mechanisms. For more complex topologies, the feedback based mechanisms offer better bandwidth allocations than the fair queuing mechanisms alone. In particular, the binary feedback mechanism we examine is able to provide allocations close to proportional fairness.

1 Introduction

In todays Internet, the dominant form of congestion control is the *end-host closed-loop* mechanism provided by TCP and the stability of the whole Internet depends on this mechanism. The efficiency of this approach depends on the assumption that all hosts are cooperative and they implement congestion control algorithms with comparable behavior. Unfortunately, both assumptions can be easily violated in practice. Early multimedia applications (Real Audio, *vat*, *wb*) did not implement any congestion control algorithms, most of the multicast applications use non-TCP forms of congestion control (RLM) and finally, users can cheat and use aggressive TCP implementations. It is not clear yet how wide uncooperation will become but since it can have a significant negative impact on cooperative flows (TCP), in the absence of evidence to the contrary, it seems dangerous to base the Internet's congestion control solely on the assumption of universal end-host cooperation. Assuming that uncooperation will become more widespread in Internet, it is important to understand what are the characteristics and the impact of congestion control mechanisms on today's networks.

As a result of the lack of cooperation from end-hosts, the problem of uncooperative flows has to be addressed by router based congestion control mechanisms. In this paper we compare two classes of router based congestion control mechanisms: *fair queuing* mechanisms and *feedback based* mechanisms. As representatives of fair queuing mechanisms we have chosen DRR [31], FRED [26] and CSFQ [32]. These mechanisms cover a good part of the design spectrum of fair queuing mechanisms: DRR [31] is a packet scheduling oriented fair queuing mechanism, FRED [26] is a "penalty based" queue management algorithm and CSFQ is a "distributed" penalty based fair queuing mechanism. As representatives of the *feedback based* router mechanisms we have chosen two classes of mechanisms (1) binary feedback based mechanisms (ERAF [29]) and (2) rate feedback based mechanisms (NBP [3]).

A second reason for this study stems from the divergence of design paths of congestion control mechanisms for ATM networks and IP (packet switched) networks. While for the ATM networks there seems to be a consensus that congestion control mechanisms based on explicit rate feedback are likely to exhibit the best performance and are viewed by most researchers as the candidates for the ATM ABR (Available Bit Rate) service class, no such consensus exists in the IP world and feedback based mechanisms are not deployed on a large scale. Regardless of the flavor they come in, a common complaint against such mechanisms in packet switched networks is that they usually require routers to perform per-flow operations and are deemed as being too "heavyweight" for deployment in the current Internet routers.

Jain [23] in "Myths About Congestion Management in High-Speed Networks" presents a comprehensive discussion about the characteristics of congestion control mechanisms in modern networks. Our goals in this paper are less ambitious. We compare selected representatives of fair-queuing mechanisms with binary feedback and rate feedback based mechanisms with respect to the following criteria [24]: *efficiency*, *fairness*, *convergence* and *scalability*. We perform our comparison through simulations using the network simulator *ns-2* [19]. We examine what is the impact of the considered mechanisms on traffic sharing one congested link, what is the behavior in the presence of multiple congested gateways, how fair are the bandwidth allocations provided by the considered mechanisms. We also examine the implementation challenges of these mechanisms in packet switched networks.

Our results indicate that with respect to traffic competing for the same link, the feedback based mechanisms are able to shield responsive traffic from unresponsive traffic with the same efficiency as the fair queuing mechanisms. With respect to traffic traversing complex topologies with multiple congested gateways, the feedback based mechanisms offer bandwidth allocations superior to the allocations provided by the fair queuing mechanisms alone. In particular, the binary feedback mechanism we examine is able to provide proportional fair allocations [25]. With respect to the implementation complexity, we found the complexity of the feedback based mechanisms to be similar to the complexity of the fair queuing mechanisms.

2 The Generic Architecture of Router Based Congestion Control Mechanisms

In order to provide a common reference point for the subsequent comparison of the considered congestion control mechanisms, we try first to identify the generic architecture of a router based congestion control mechanism.

Router based congestion control mechanisms (CCM for short), can operate either at the data-link layer of the router or at the network layer or above. Earlier mechanisms were designed to operate at the data-link layer [9] and were generically referred to as back pressure or Xon/Xoff mechanisms. In this approach gateways sharing a direct link pass congestion signals to each other and stop forwarding all traffic upon receiving a congestion signal (Xon). Upon receiving

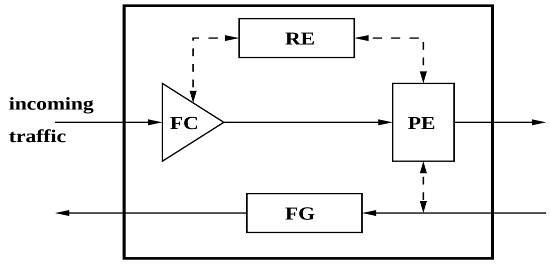

Figure 1. Schematic of a router implementing a congestion control mechanism. FC - flow classifier, RE - rate estimator, PE - policy enforcer, FG - feedback generator.

a Xoff signals routers start forwarding traffic again. Experiences with backpressure showed that it can be highly unfair[1] and unstable[2]. More recent router mechanisms are designed to operate at the network layer and are based on flow isolation, i.e. routers perform per flow operations.

Figure 1 presents the generic architecture of a network layer congestion control mechanism. Upon receiving a packet, the router first determines the flow the packet belongs to. Without going into the architectural details of the packet classier, we note that the classification required by congestion control mechanisms for best effort traffic is a simplification of the generic packet classification problem as presented in [20]. Also, the notion of a flow definition is intentionally left open. For this study we consider a flow as defined by the pair (source, destination, port) and note that all the CCM we evaluate have similar classification criteria. The flow classifier component is labeled FC in Figure 1.

After classification, i.e. identifying the flow a packet belongs to, the packet is forwarded to a *policy enforcer* component, labeled PE in Figure 1. This component, based on the mechanism's particular criteria, decides whether to drop, enque or forward the packet. In addition to these 2 components, i.e. flow classifier and policy enforcer, some mechanisms have a flow *rate estimation component* (RE), used by the policy enforcer component.

The three components presented (flow classifier, policy enforcer, rate estimator) are situated on a flow's forward path and are common to all congestion control mechanisms, regardless of their functional classification, i.e. closed-loop or open-loop. Closed loop mechanisms have an additional feedback generator component, labeled FG in Figure 1, and situated on the flow's backward path.

3 Fair Queuing Mechanisms

In order to reduce complexity and decrease per-packet service time, IP gateways normally maintain as little state as possible: they usually keep no per-flow state and use a single first-in first-out (FIFO) packet queue shared by all flows. The gateway estimates the overload by monitoring the queue length and starts discarding packets when the queue space is exhausted. This discipline, referred to as DropTail, has been shown to distribute losses among connections arbitrarily [17], to be unfair and to penalize bursty connections.

In order to address the deficiencies of DropTail, alternative queue management algorithms have been proposed. One

[1] Flows can be shut-off by the presence of a very aggressive flow that causes congestion.

[2] Backpressure can have a cascading effect, i.e. traffic can be effectively shut-off in the whole network and the network oscillates enters in a start-stop cycle with a short period.

of the most popular algorithms is RED [18] - Random Early Detection. RED tries to use randomization to ensure that all connection encounter similar loss rates and tries to prevent congestion, rather than just reacting to it, by dropping eagerly, before the gateway's buffers are exhausted.

Even if RED seems to address some of the DropTail queue management algorithm shortcomings, it has been shown by Morris [26] that it can be unfair to competing TCP connections when robust connections are present, i.e. connections with large windows and short round-trip times, or unresponsive traffic is present. In order to address the unfairness intrinsic to either DropTail or RED, researchers proposed algorithms able to perform fair bandwidth allocations to competing flows. These algorithms, besides improving link utilization, have been shown [12] to improve other traffic characteristics, e.g. delay-jitter.

For this study, we considered as representatives three fair-queuing mechanisms, namely DRR [31], FRED [26] and CSFQ [32]. We chose these mechanisms such as they cover a large part of the possible design spectrum. FRED [26] is a queue management mechanism designed to provide fairness by allocating the buffer space available in a fair manner and by dropping packets based on queue occupancy. DRR [31] is a fair-queuing mechanism, i.e. the router performs access control based on a computed per-flow fair share of the link. Finally, the last mechanism we consider is CSFQ [32], which is a "distributed" penalty based mechanism that provides approximately fair queuing by requiring the cooperation of more than one router. In the following we will present each mechanism in greater detail.

FRED (Flow Random Early Drop): This queue management algorithm is based on RED (Random Early Detection). A RED router serves packets in a FIFO order but uses a more complicated buffer management scheme than DropTail. RED maintains two queue occupancy thresholds. When the queue occupancy[3] is smaller than the first threshold no packets are dropped, when queue occupancy is higher than the second threshold all packets are dropped. When the queue occupancy is between the two thresholds packets are dropped with a probability that increases linearly with the queue occupancy. FRED extends RED to provide fair bandwidth allocation. In order to achieve fairness, FRED maintains state for all flows that have at least one packet buffered and drops packets preferentially from flows that either (1) had many packets dropped in the past or (2) have a queue larger than the average queue size.

When receiving a packet, FRED performs flow classification in order to update the internal state maintained by the policy enforcer. The policy enforce component in FRED maintains per-flow state for all flows that have packets buffered. This state consists of the number of packets buffered and a strike variable counting the number of times the flow has been unresponsive to congestion. In addition, FRED computes at each packet arrival and departure an average queue length. Based on these values the policy enforcer component decides whether to drop or enque packets.

DRR (Deficit Round Robin): Traditional fair-queuing mechanisms (FQ [12], W2FQ [6]) have an implementation complexity of $O(logn)$. Shreedhar&Varghese [31] introduce DRR which is an efficient fair queuing mechanism with a $O(1)$ per flow operational complexity. In DRR each flow has associated it's own queue and each queue has associated a deficit counter. The DRR router has a service quantum which is a measure of the minimum service unit. The DRR router examines in a round-robin fashion all active queues and compares the queue deficit counter with the quantum. If the front packet has a size less than the quantum the packet is served, otherwise the quantum is added to the deficit counter. The front packet from a queue is served in the scheduling round when the queue deficit counter plus the quantum exceeds the packet size. After the packet is served the packet size is subtracted from the quantum.

[3] Exponentially averaged

In order to determine the queue associated with a flow, a DRR router performs flow classification. DRR routers do not need to perform per flow rate estimation and the policy enforcer component performs the round-robin packet scheduling described above.

CSFQ (Core-Stateless Fair Queuing): This queue management algorithm has been introduced recently as a lightweight alternative to the other fair queuing mechanisms. CSFQ assumes cooperation of the routers inside an "island" of routers. Routers at the entry to the island (*ingress*) classify flows, compute flow arrival rates and label packets with the estimated arrival rate. Routers inside the island examine the packet labels and drop packets based on an estimated link fair share.

The mapping of the CSFQ operation to the generic architecture presented in Section 2 is straightforward. *Ingress* routers contain the flow classification (FC) and rate estimation (RE) components. The policy enforcer component is distributed in all the routers inside the island.

4 Feedback Based Mechanisms

The class of closed-loop congestion control mechanisms contains those mechanisms that operate based on feedback from the network. We evaluate mechanisms employing two types of feedback: *binary feedback* and *rate feedback*.

In their general form, rate feedback control mechanisms adapt the sending rate of a source using a equation of the form $x_{n+1} = x_n + \alpha(\mu_n - x_n) + f(\gamma)$, where x_n is the sending rate in the n^{th} control interval and μ_n is the network service rate. In the above expression $f(\gamma)$ is a factor usually added for better convergence and takes various forms. Recent work of Altman et. al [4] presents a model for rate feedback mechanisms using a generic rate adaptation control. They show that mechanisms that try to match directly the network service rate, i.e. $f(\gamma) = 0$, are inherently unstable. They also show that when $f(\gamma)$ is a function of queue occupancy the generic control has good convergence and stability.

There is a large body of work on rate feedback based congestion control mechanisms. Jain&Charny [8] introduce a rate feedback mechanism with rate matching that exhibits good performance in terms of convergence and behavior in transience. However, their mechanism is specifically designed for connection oriented networks (ATM) in which the switches have information about the flows and the routes are stable. Mishra&Kanakia introduce a hop-by-hop rate feedback mechanism in which routers adjust service rates based on rate feedback, buffer occupancy and a notion of service setpoint. They compare their proposed control for mechanisms using hop-to-hop(H2H) feedback and end-to-end(E2E) feedback and they show empirically that H2H feedback behaves better under most circumstances.

For this evaluation, as representative of the rate feedback control mechanisms we chose a mechanism introduced by Albuquerque et. al (NBP [3]), a multi-hop rate matching mechanism with a control equation in the steady state of the form $x_{n+1} = x_n + \alpha(\mu_n - x_n)$.

Binary feedback mechanisms usually use window based rate regulators that adapt the sending rate of a source based on congestion signals from the network. Jain [24] shows that from the possible window variation policies, additive-increase – multiplicative-decrease (AIMD) exhibits the best convergence properties.

For this study, as a representative of binary feedback mechanisms with a AIMD window variation policy we chose a mechanism introduced by Acharya et. al (ERAF [29, 22]). A similar mechanism, but using a additive-increase/additive-decrease window discipline (SquId [28]) was proposed by Prue&Postel. Recent work by Floyd [15], examined alternate window variation policies, namely TCP-friendly rate control mechanisms. We do not specifically consider TCP-friendly window variation policies and for a comparison of TRFC and AIMD the reader is referred to [16].

4.1 Rate Feedback Mechanisms With Rate Matching - NBP

The mechanism we evaluate is based on Network Border Patrol (NBP). NBP is a congestion control mechanism for packet-switched networks introduced by Albuquerque et. al [3] and is a mechanism very similar in spirit with the one introduced by Jain&Charny [8] for ATM networks. The goal of NBP is to have the routers situated at the entry point (periphery) of the network allocate bandwidth to a flow as close to the minimum service rate that the network is able to provide (or as close to the flow exit rate as possible).

NBP operates over an island of routers (as CSFQ [32], ERAF [22]) and distinguishes between network entry points - *ingress* routers and network exit points - *egress* routers. In NBP, ingress routers monitor flows, compute flow arrival rates and periodically send forward feedback (FF) packets to egress routers. These packets are sent spaced at intervals equal to the round-trip delay (RTT) between the ingress router and the egress router and contain a list of flow specifiers and a time stamp. Upon receiving a FF packet, the egress router replies with a backward feedback packet (BF) containing the same time stamp, flow specifiers and a list of flow departure rates corresponding to the specifiers. BF packets are used by the ingress to update the RTT between the ingress router and the egress router (by examining the time stamp) and to determine if a flow experiences congestion.

A flow is determined to have been experienced congestion if

$$ingress\_rate * \Delta RTT > hops * MSS \tag{1}$$

where $ingress\_rate$ is the flow arrival rate at the ingress router, ΔRTT is the difference between the current RTT (as computed from BF packets) and the minimum RTT observed and *hops* is the number of hops between the ingress router and the egress router. If the flow experienced congestion, its rate is set to the egress rate reported in the BF packet, i.e. $new\_rate = egress\_rate - MRC$[4], where $MRC = min(MSS/RTT, \alpha * egress\_rate)$, with $MSS = 1500 bytes$ and $\alpha = 0.1$. If the flow does not experience congestion the rate is increased with MRC, i.e. $new\_rate = ingress\_rate + MRC$.

Flows enter the network in a slow start phase. The rate of flow in slow start phase is doubled at every control interval until the flow experiences congestion. After that the flow enters a congestion avoidance phase. While in the congestion avoidance phase, the flow bandwidth is controlled as previously described.

The efficiency of the mechanism described in [3] depends on its ability to identify flows that experience congestion. The left hand side term of the congestion test, i.e. $ingress\_rate * \Delta RTT$, is intended to provide an estimate of the amount of flow data that is buffered somewhere in the network. In particular, for flows with low bandwidth, i.e. $ingress\_rate \downarrow\downarrow$, or for network topologies where ΔRTT is low (regardless of the $ingress\_rate$), we found this test to fail to identify congestion. We found similar shortcomings of this test for flows with small packet sizes ($<< MSS$).

In order to address this problem, the implementation we evaluate uses a congestion test of the from $ingress\_rate * \Delta RTT > hops * packet\_size$.

[4] MRC - Minimum Rate Change.

4.2 Binary Feedback Mechanisms - ERAF

As a representative of window based binary feedback congestion control mechanisms we chose for this evaluation ERAF [22], a mechanism introduced by Acharya et.al.

ERAF is a congestion control mechanism design to prevent congestion collapse from undeliverable packets by dropping these packets as close to their network entrance point as possible. Similar approaches, i.e. flow regulation at the network periphery are employed by NBP [3] and CSFQ [32].

In ERAF routers at the network periphery (ingress) monitor flow arrival rates. Routers inside the network (core) generate binary feedback signals when congestion occurs, i.e. on packet drops. These signals are sent back to the originator of the dropped packet. Ingress routers monitor traffic, identify the congestion signals and regulate flows based on these signals.

The regulation process works as follows. Flows are initially allowed to enter the network without any bandwidth restrictions. When an ingress router detects a congestion signal for a flow, it starts regulating the flow bandwidth using an AIMD window based discipline. Under this discipline, when congestion is detected, the flow' allowed bandwidth is halved and all subsequent congestion signals for the flow within a time interval are ignored. Ideally, in order to allow TCP flows to proceed unpenalized, this time interval is equal to the flow' end-to-end delay. Since ingress routers have no good way of determining the round-trip delay for a flow, they use an easy to compute approximation equal to twice the delay of to the next downstream router. After the expiration of this interval the flow bandwidth is increased by one average packet size per interval until a new congestion signal is received for the flow. If the flow' bandwidth reaches a certain threshold, the flow is allowed to proceed unregulated until a new congestion signal is received for it.

In order to lower the impact on the network during congestion, in ERAF, the feedback generator component situated in the core routers is extended with bandwidth limiters. These limiters are associated with links and forward on the associated link only a static fraction of the incoming feedback traffic.

5 Simulations

In this section we evaluate the performance of the considered mechanisms through simulation using *ns-2* [19] under a variety of conditions. From the implementations we evaluate, RED and DRR are contained in the standard *ns-2* distribution, CSFQ and FRED are obtained from Stoica [1], NBP is obtained from Albuquerque [2] and ERAF is implemented by ourselves. The traffic load we use in our simulations is composed of constant-bit-rate UDP flows and a mixture of "robust" TCP flows that have an unlimited amount of data to send and large windows ("ftp-like" flows) and "thin"[5] TCP flows (smaller windows, 100 bytes packets). For selected experiments we also considered bursty traffic sources (on-off). All simulations, unless indicated otherwise, were run for 100s. The packet size for the UDP flows, unless indicated otherwise, was set to 100 bytes. The packet size for the TCP flows was set to 1500 bytes.

As a general rule, in a packet switched network, sources should limit their sending rate by taking into consideration the state of the network. Ignoring this principle may put the network into congestion collapse[6]. Accordingly, one

[5] TCP is biased in favor of connections with high values of the delay-bandwidth product. Demers [12] showed that fair queuing algorithms are able to improve the behavior of thin TCP connections competing with robust TCP connections. Since thin TCP connections recover slower from packet losses, we are interested in determining the impact, if any, of the feedback based mechanisms on these connections.

[6] Congestion collapse occurs when network resources are consumed by traffic that will be later discarded.

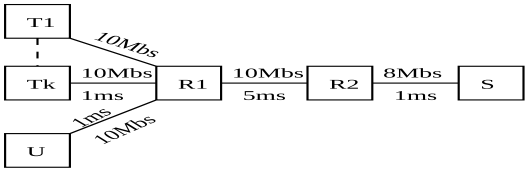

Figure 2. A single congested link scenario. Traffic consists of N=20 TCP flows and one UDP flow. The UDP sending rate is varied from 10Kbs to the R1-R2 link capacity 10Mbs.

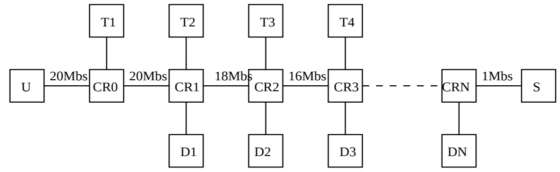

Figure 3. Multiple congested links topology: one UDP flow between U-S, TCP flows between $T_i - D_i$. The bandwidth of the $CR_i - CR_{i+1}$ link is (20-2i)Mbs.

goal of congestion control mechanisms, i.e. *efficiency*, is to avoid congestion collapse while maximizing network usage (throughput). Unfortunately, maximizing network throughput does not necessarily imply that all traffic participants will get a share of the network resources. Because of this, a secondary goal of congestion control mechanisms is *fairness*, i.e. all traffic participants are allowed by the mechanism to make progress.

Due to space limitations we report here only results for a selected subset of experiments. To evaluate the efficiency of the considered mechanisms we run experiments designed to illustrate the impact on various traffic sources competing for one congested link and multiple congested links. We evaluate the fairness of the bandwidth allocations provided by these mechanisms on topologies with one shared congested link and on more complex network topologies specifically designed for the fairness evaluations (GFC-2 [34]) of traffic control mechanisms.

As a primary metric we report the *goodput* of a flow . Floyd&Fall [13] define *goodput* of a flow as the bandwidth delivered to the receiver, excluding duplicate packets. The goodput metric shows the performance improvements achieved by the network by dropping undeliverable packets as early as possible. It also shows how well a congestion control mechanism is able to shield responsive flows from the impact of the unresponsive ones.

The volume of data presented is rather large and each experiment involves complicated dynamics. As a base case for our comparison we considered RED routers. Since the fair queuing mechanisms have been closely studied, and due to space limitations, we will focus our discussion mostly on the behavior of the feedback based mechanisms.

5.1 A Single Congested Link

We examine first what is the effect of the considered congestion control mechanisms on traffic sharing a single congested link. The simulated network topology is presented in Figure 2. The bandwidth of the shared link (R1-R2) is 10Mbs. The buffer size in all routers is set to 64KB.

The experiments we present are designed to illustrate the behavior of the evaluated mechanisms under various traffic mixes: we are interested to determine how well is a particular mechanism able to shield the TCP traffic from the UDP traffic and how "fair" is the bandwidth distribution between competing TCP flows.

The traffic consists of a total of N=20 TCP flows competing with one UDP flow whose bandwidth is varied from 10% to 100% of the capacity of the R1-R2 link. The TCP traffic consists of a mixture of K "robust" flows with a large window and a packet size of 1500 bytes and "thin" flows with a small window and a packet size of 100 bytes. Under a fair allocation the UDP flow should receive a bandwidth share of $\approx 500Kbs$.

In the first experiment we set the UDP packet size to 100 bytes and have a low number (K=3) of robust TCP flows.

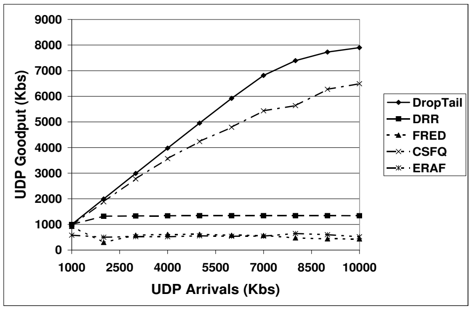

Figure 4. UDP goodput for the link sharing scenario with N=3 robust TCP flows and 17 thin TCP flows. The UDP packet size is 100 bytes. The ideal UDP bandwidth share is $\approx 500Kbs$.

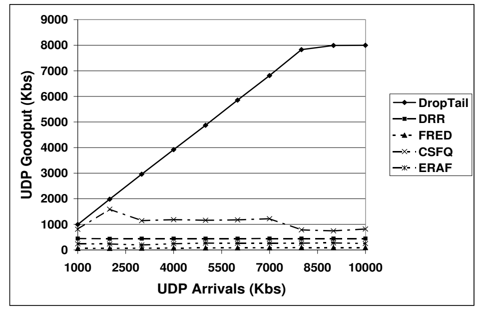

Figure 5. UDP goodput for the link sharing scenario with N=20 robust TCP flows. The UDP packet size is 100 bytes. The ideal UDP bandwidth share is $\approx 500Kbs$.

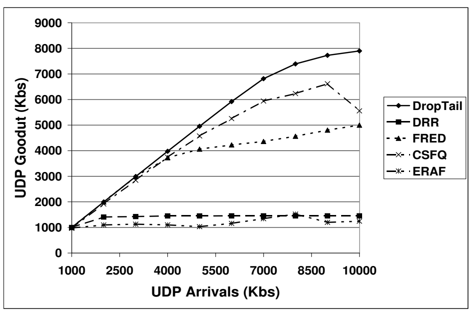

Figure 6. UDP goodput for the link sharing scenario with N=3 robust TCP flows and 17 thin TCP flows. The UDP packet size is 1500 bytes. The ideal UDP bandwidth share is $\approx 500Kbs$.

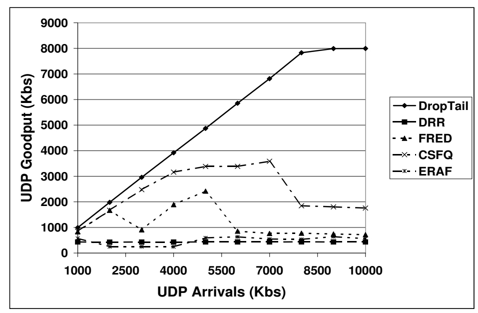

Figure 7. UDP goodput for the link sharing scenario with N=20 robust TCP flows. The UDP packet size is 1500 bytes. The ideal UDP bandwidth share is $\approx 500Kbs$.

As shown in Figure 4, both ERAF and FRED keep the UDP bandwidth share close to the ideal allocation, followed by DRR and CSFQ. When increasing the packet size of the aggressive UDP traffic to 1500 bytes, Figure 6, we find that ERAF is best at containing the aggressive UDP flow, followed by DRR, FRED and CSFQ respectively. When increasing the number K of robust TCP flows from 3 to 20, all mechanisms considered offer a better bandwidth repartition than in the previous case (Figure 5 and Figure 7), regardless of the packet size. These experiments indicate that ERAF is less sensitive to the packet size of the regulated traffic than CSFQ and FRED are.

To understand how well the available bandwidth is partitioned between the TCP traffic we present in Figure 8 the average and the standard deviation of the number of TCP packets delivered. The UDP sending rate is set to 5Mbs. For K=20, i.e. all TCP flows are robust, ERAF is close to DRR, followed by the other mechanisms. For a traffic mixture of robust and thin flows the fair queuing mechanisms are able to improve the goodput of thin flows in a larger degree than ERAF. ERAF still performs better than RED.

	AVG	STD DEV
RED	2910	194
DRR	3145	17
FRED	3305	117
CSFQ	2810	226
BF	3223	181

Figure 8. Average number of packets delivered for a TCP flow and the standard deviation for N=20 robust TCP flows.

	Robust		Thin	
	AVG	STD DEV	AVG	STD DEV
RED	16874	738	6733	38
DRR	11556	1	20810	1
FRED	15547	71	10750	159
CSFQ	12855	690	13074	284
BF	17591	403	7890	48

Figure 9. Average number of packets delivered for a TCP flow and the standard deviation for N=3 robust TCP flows and 17 thin TCP flows.

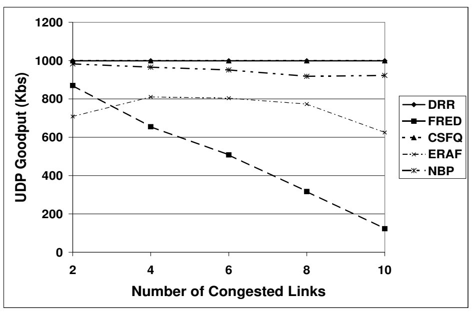

Figure 10. Goodput variation of an UDP flow with packet size of 100 bytes on the multiple congested links topology.

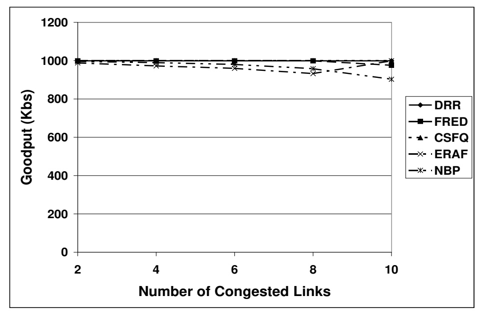

Figure 11. Goodput variation of an UDP flow with packet size of 1000 bytes on the multiple congested links topology.

5.2 Multiple Congested Links

We now examine how the goodput of a flow is affected when the flow traverses more than one congested link. The experiment performed is based on the topology in Figure 3, with N, the number of congested links varied from two to ten. The topology is adapted after a network topology used by Floyd&Fall [14] to study the behavior of TCP connections in the presence of multiple congested gateways. The UDP flow, between U-S, has a sending rate of 20Mbs and its goodput is limited to 1Mbs by the last link it traverses.

In the first experiment we set the UDP packet size to 100 bytes. As shown in Figure 10, CSFQ, DRR and NBP allow the UDP flow to achieve close to the available bandwidth of 1Mbs. ERAF allows the UDP flow to achieve a bandwidth of $\approx$ 700Kbs. When increasing the UDP packet size to 1500 bytes, all mechanisms allow the UDP flow to achieve a bandwidth close to 1Mbs.

To illustrate how well the considered mechanisms shield the competing TCP traffic from the UDP traffic we present data for the TCP flows at the edge of the network and inside the core of the network for a network with N=10 congested links. As shown in Figures 12 and 13 and 14, the feedback based mechanisms offer consistently better goodput for all

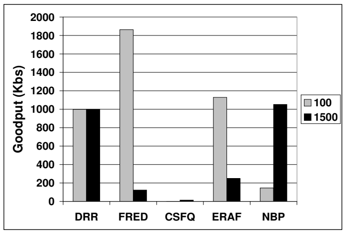

Figure 12. Goodput for the TCP flow traversing the CR9-CR10 link in the multiple congested links scenario. Under a max-min fair allocation the goodput is 1Mbs. The TCP flow competes with one UDP flow whose packet size is set to 100 bytes or 1500 bytes.

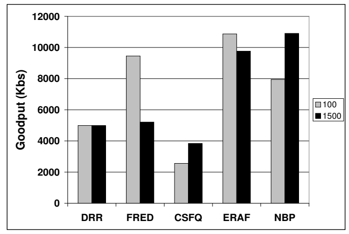

Figure 13. Goodput for the TCP flow traversing the CR5-CR6 link in the multiple congested links scenario. Under a max-min fair allocation the goodput is 10Mbs. The TCP flow competes with one UDP flow whose packet size is set to 100 bytes or 1500 bytes.

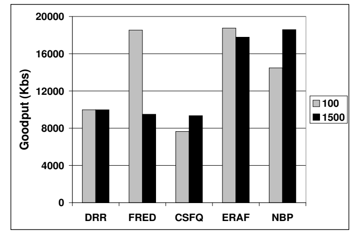

Figure 14. Goodput for the TCP flow traversing the CR1-CR2 link in the multiple congested links scenario. Under a max-min fair allocation the goodput is 19Mbs. The TCP flow competes with one UDP flow whose packet size is set to 100 bytes or 1500 bytes.

	BURST	TCP AVG	TCP STD DEV	Bandwidth
RED	37365	2781	377	871Kbs
DRR	8586	3268	22	989Kbs
FRED	4159	3293	125	992Kbs
CSFQ	33766	2923	265	911Kbs
BF	15044	3021	274	947Kbs

Figure 15. Number of packets delivered for a bursty source competing with N=20 robust TCP flows.

TCP flows. This is because the fair queuing mechanisms alone, for this topology, are not able to prevent congestion collapse. The fact that fair queuing mechanisms are not able to prevent congestion collapse has been also illustrated by Floyd&Fall [14].

5.3 Bursty Traffic Patterns

So far we have examined only the influence on constant-bit-rate UDP and TCP traffic. We now examine the impact on sources with a greater degree of burstiness. The topology considered is the one in Figure 2. The R1-R2 link is shared by one ON-OFF source and 20 TCPs. The ON and OFF periods of the ON-OFF source are drawn from exponential distributions with means of 100ms and 1000ms. During the active phase the ON-OFF source sends at 10Mbs.

Figure 15 shows the average number of packets received for the TCP traffic and the ON-OFF source (column labeled BURST). As shown, with respect to the TCP traffic, the impact of ERAF is similar with the impact of DRR and FRED. With respect to the ON-OFF source, ERAF allows more bandwidth to the ON-OFF source than DRR and FRED do. In both cases ERAF is superior to CSFQ. These results indicate that ERAF reacts reasonably well over short time scales.

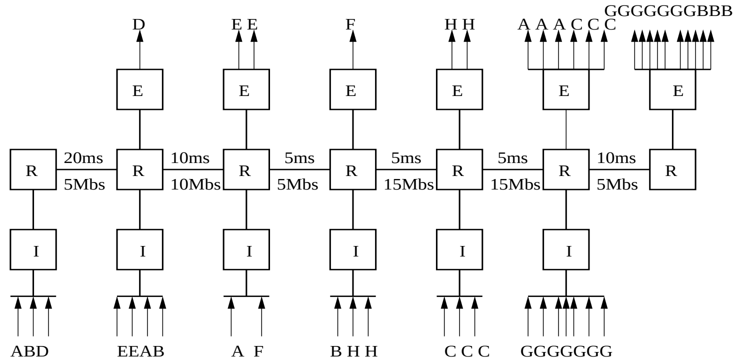

Figure 16. General Fairness Configuration - GFC 2

6 Fairness

There is no generally-agreed-upon definition for fairness in a computer network. While in the case of traffic competing for the same link the generally accepted notion of fairness allocates bandwidth to flows in equal shares, this is not the case for more complicated network topologies.

One of the fairness criteria widely accepted in the research community is *max-min* fairness. Max-min fairness is a direct extension of the "equal share" principle accepted for the one link bandwidth allocation with the additional constraint that the smallest throughput rate must be as large as possible. Informally, a max-min fair allocation for a network topology is computed by a progressive filling algorithm as follows. For each link the equal share for the flows traversing the link is computed, subject to link capacity constraints and bandwidth consumed by the flows for whom the rates are already determined. At each step of the algorithm the minimum of the shares computed is chosen as the final allocation for the flow it corresponds to. Max-min fairness can be achieved by explicit rate calculation algorithms such as those studied in the context of the available bit rate (ABR) service class in ATM networks [21, 8].

Recently, the appropriateness of the max-min fair allocation has been questioned by Kelly [25] who argues that bandwidth should rather be shared such as to maximize an objective function representing the overall utility of the flows in progress. Informally, the rates x_i under a proportional fair allocation maximize the $\Sigma log(x_i)$ subject to link capacity constraints. In particular, according to the utility principle, because they consume more network resources, proportional fairness penalizes long routes more severely than max-min fairness in the interest of greater overall throughput. Kelly [25] also shows that the AIMD congestion avoidance mechanism implemented in TCP tends to realize proportional rather than max-min fair allocations.

We examine the fairness of the considered mechanisms on the topology described in Figure 16. The topology is an adaptation of the General Fairness Configuration (GFC-2) [34], a topology specifically designed to test the max-min fairness of traffic control algorithms. The traffic consists of 22 UDP flows belonging to groups labeled from A to H. Each UDP flow sends data at 10Mbs with a packet size of 100 bytes. The network is designed in such a way that members of each group receive the same max-min bandwidth allocation.

In Figure 17 we present the simulation results. Due to space limitations we present results only for the topologies with RED, DRR, ERAF and NBP with either RED (NBP-RED) or DRR (NBP-DRR). From our selection of results, the topology with DRR routers illustrates the case where fair queuing is uniformly deployed within the network and the

Flow	Max-Min	Prop. Fair.	RED	DRR	ERAF	NBP-RED	NBP-DRR
A	1	0.85	0.15	0.83	0.79	0.85	0.82
B	0.5	0.9	0.04	0.45	0.61	0.33	0.49
C	3.5	3.23	4.15	2.72	2.9	3.3	3.2
D	3.5	3.75	1.57	1.57	3.42	2.82	2.58
E	3.5	3.62	2.1	1.61	3.23	2.91	2.56
F	1	1.25	1.96	0.93	0.44	1.1	0.82
G	0.5	0.32	0.65	0.48	0.41	0.56	0.49
H	5.2	4.85	4.5	3.62	4.25	4.8	4.85
	Max-Min Distance		0.35	0.39	0.16	0.12	0.17
	Proportional Distance		0.39	0.41	0.15	0.16	0.21

Figure 17. Flow rates in Mbs for the GFC-2 configuration. The last two rows present the distance between the allocation as provided by a mechanism and the ideal allocation (either max-min or proportional).

choice of NBP with DRR is motivated by results presented in [3][7].

To better characterize an allocation we use as a measure how much a given rate allocation deviates from the desired fair allocation. We measure the unfairness of an allocation as the distance between the allocation $\vec{x}$ as provided by a mechanism and the nearest desired fair allocation $\Pi(\vec{x})$ $d(\vec{x}) = \frac{\|\vec{x}-\Pi(\vec{x})\|}{\|\Pi(\vec{x})\|}$, where $\Pi(\vec{x}) = \frac{\sum_j^I x_j}{I}$ and $\| * \|$ is the standard Euclidean norm [7].

ERAF uses a AIMD window adaptation algorithm similar to the one used by TCP. This window adaptation algorithm allows ERAF to allocate[8] rates closer to the allocation offered by proportional fairness. This is illustrated by the unfairness measure presented in Figure 17. Albuquerque [3] shows that NBP when combined with fair queuing mechanisms is able to provide max-min fair allocations. We also note that, because of their inability to prevent congestion collapse, the allocations provided by a topology that implements fair queuing in all links are more unfair than the allocations provided by the feedback based mechanisms. Our results are consistent with the data presented in [3].

7 Convergence of the Feedback Based Mechanisms

The convergence of a mechanism is generally measured by the speed with which the system approaches the goal state from any starting state. Due to the nondeterministic nature of the various entities, the network will not generally converge to a single steady state. However, the network usually reaches a "dynamic" equilibrium state in which it oscillates around an optimal state. The time taken to reach this "equilibrium" and the size of the oscillations jointly provide a measure of convergence [24]. The time determines the *responsiveness* and the size of the oscillations determines the *smoothness* of the control. Ideally, both these measures are small.

Smoothness: To illustrate the smoothness of the examined mechanisms we ran an experiment on the topology presented in Figure 2. The traffic consists of 20 robust TCP flows and an UDP flow sending at half the bandwidth of the link R1-R2 (5Mbs). Figure 18 presents the variation of the bandwidth of the UDP flow under ERAF for the first 20 seconds of the simulation. The bandwidth variation is measured over intervals of 5ms. Figure 19 presents the behavior of the

[7] Albuquerque [3] shows that NBP when combined with fair queuing is able to provide max-min fair allocations.

[8] For the proportional fairness case the rates are for the class A flows 0.70, 0.87, 0.994 and for B class flows 0.54, 0.63 and 1.55. The figures reported are the average rate for the class.

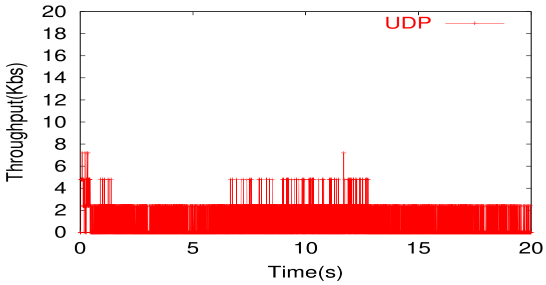

Figure 18. Bandwidth variation of the UDP flow as determined by ERAF. The UDP flows competes with N=20 robust TCP flows in the topology presented in Figure 2.

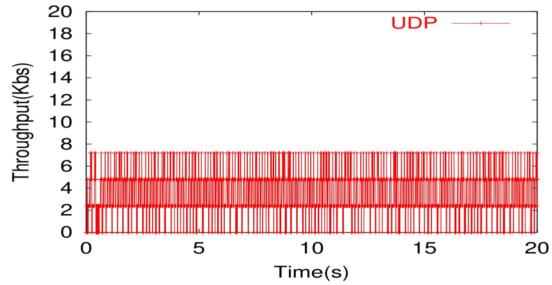

Figure 19. Bandwidth variation of the UDP flow as determined by NBP. The UDP flows competes with N=20 robust TCP flows in the topology presented in Figure 2.

NBP mechanism with DRR. Both mechanisms are able to keep the bandwidth variation of the UDP flow at a low value, with the ERAF offering a better behavior. The amplitude of bandwidth oscillations for ERAF is for most intervals around 2KB, while the amplitude of the oscillations for NBP is around 8KB.

Responsiveness: Fair queuing mechanisms start scheduling packets immediately after a flow becomes active at a gateway. Since they manage the bandwidth only for one link, their responsiveness can be considered as instantaneous. This is not the case for the feedback based mechanisms. The routers implementing feedback based congestion control mechanisms can be viewed as implementing a distributed algorithm whose goal is reaching a dynamic equilibrium state in the network.

To evaluate the responsiveness of the rate feedback mechanism we draw on results presented by Charny et al [8] for a mechanism for ATM networks whose operation is very similar to the operation of the mechanism we examine in this paper. In the mechanism presented by Charny routers compute the local fair rates for the flows traversing them and send these rates to the sources of the flows. Sources adjust sending rates to the minimum rate indicated by the feedback information. They show that for a network with an upper bound for the round trip delay D and N congested links, the upper bound on the convergence time of the mechanism is 4ND. However, they point out that after two round-trip delays after all flows are known at all routers, the rates as approximated by their algorithm are close to the equilibrium values. In NBP, the feedback is generated only by egress routers.In addition, egress routers do not compute exact flow fair shares, instead they communicate instantaneous flow departure rates and assume an underlying fair queuing mechanism. We were able to derive a theoretical upper bound on the convergence time of the mechanism of 8ND, where N is the number of egress routers communicating feedback. The complete derivation is omitted for lack of space. In practice, Albuquerque et al [3] found that their mechanism allocates rates close to the equilibrium rates after a time interval equal to four round-trip delays.

While the fair queuing mechanisms and the rate feedback mechanism operate continuously, our binary feedback mechanism is activated only during congestion when packets are dropped. Because of this we cannot talk about a convergence interval for our mechanism. The feedback generated for the first packet dropped from a flow will cause a halving of the bandwidth for the flow. This allows our mechanism to respond fairly rapidly to congestion.

8 Scalability

Scalability is perhaps the most important performance measure of any traffic control mechanism and one of the main reasons behind the lack of large scale deployment of these mechanisms. For a large scale network, the overhead of maintaining per flow state, performing per flow rate control and rate monitoring may become inordinately expensive. In this section we will examine what are the characteristics of the considered mechanisms in terms of implementation complexity and network traffic.

Packet Classification: The first function performed by a router that implements a congestion control mechanism is packet classification. The mechanisms considered are congestion control mechanisms for best-effort traffic and the packet classification they use is a simplified form of the more general packet classification (e.g. as presented by Gupta&McKeown [20]). In this context packet classification is done to identify the flow a packet belongs to[9]. All mechanisms examined, with the exception of FRED[10], have to maintain state and classify all active flows passing through the gateway, having similar classification overheads.

Rate Monitoring and State Information: From the mechanisms examined, only CSFQ, NBP and ERAF perform rate monitoring. CSFQ and ERAF perform rate monitoring at the ingress routers, while NBP performs it at both ingress and egress routers. The rate monitoring algorithms used by these mechanisms, i.e. exponential averaging (CSFQ) [32] and Time Sliding Window (TSW) [11], maintain two per flow state variables, having similar memory requirements. Exponential averaging uses a more complicated update formula for the estimation of the arrival rate.

Traffic measurements on the MCI backbone [10] reported a number of 150 million active flows over a period of twenty-four hours. Assuming an uniform time distribution of these flows, i.e. 105K active flows per minute, the memory used to maintain the state for a congestion control mechanism is around 900KB.

Flow Rate Control: We compare the complexity of the rate control algorithms for CSFQ, NBP and ERAF. From these mechanisms, CSFQ and NBP have a continuous mode of operation, while ERAF is activated only during congestion. The flow rate control algorithm performed by CSFQ requires core routers to examine packet headers and drop packets according to the information contained in the packet header. In CSFQ core routers also relabel packets. The flow control algorithm used by NBP is executed at the ingress router. These routers perform a congestion test and, according to the result of this test, adjust rates. ERAF uses a AIMD window variation policy at intervals determined by the arrival, or non arrival, of feedback packets. The processing overheads determined by both NBP and ERAF at the ingress router can be considered similar.

Communication Overhead: The only two mechanisms that have a communication overhead are NBP and ERAF. The bandwidth overhead imposed by the rate feedback mechanism is proportional with the number of active flows and bandwidth is consumed both on the flow forward route and on the flow backward route. Periodically, ingress routers send toward egress routers forward feedback packets. These packets contain information about the flows active at the ingress routers. After receiving a forward feedback packet, egress routers reply with backward feedback packets containing the flow departure rates.

In ERAF feedback is generated on packet drops and bandwidth is consumed only on the flow backward route. The

[9] The generic packet classification matches the incoming packets against a database of patterns and actions.

[10] FRED maintains state only for the flows that have packets buffered at the gateway.

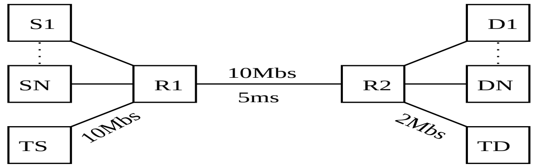

Figure 20. One congested gateway - many flows scenario. One TCP flow from TS to TD. N UDP flows between $S_i - D_i$. Number of UDP flows varied from 10 to 160. Total UDP sending rate 9Mbs.

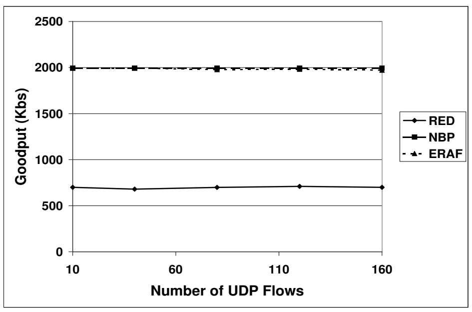

Figure 21. Goodput of a TCP flow competing against a large number of UDP flows on the topology in Figure 20.

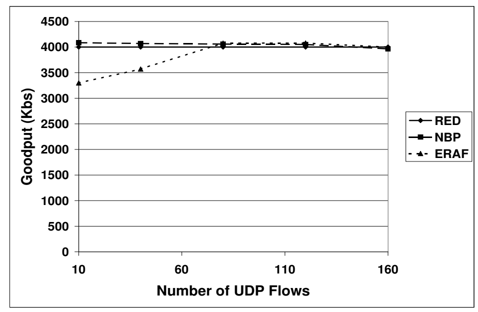

Figure 22. Combined goodput of the UDP flows on the topology in Figure 20.

bandwidth consumed is potentially proportional with the packet drop rate at the congested router. In practice, as recommended also by RFC 1812 [5], this bandwidth should be limited and we present techniques for limiting the bandwidth consumed by the feedback traffic in [22].

To illustrate the bandwidth overhead of both mechanisms we run an experiment on the topology presented in Figure 20. The topology has one congested link R1-R2 with a bandwidth of 10Mbs. The R1-R2 link is shared by one TCP flow between TS-TD and a variable number of UDP flows. The number of UDP flows is varied between 10 and 160. The goodput of the TCP flow is constrained at 2Mbs by the R2-TD link. The goodput of the UDP flows is constrained at 4Mbs by the R2-D_i links.

Figures 21 and 22 present the variation of the TCP flow and the variation of the combined UDP goodput respectively. As shown, both mechanisms are able to improve the goodput of the TCP flow in a similar manner. We measured also the bandwidth consumed by the feedback packets for both mechanisms during the first 10 seconds of simulation. For NBP, the forward feedback packets consumed 180Kbs[11] and the backward feedback packets consumed 1Mbs, for a total of 1.2MBs. The results reported for ERAF are for 1% of the R2-R1 link available for feedback packets. This amounts to 100Kbs[12] bandwidth consumed by the feedback packets.

[11] The ns-2 implementation uses for flow identification IPV6 style flow identifiers. If source-destination address pairs were to be used, the bandwidth consumed would be two or four times larger.

[12] In the current implementation of ERAF, the generated feedback is carried in ICMP Source Quench messages. The payload of these messages contains the header of the dropped packet and its first 64 bits of payload. A simple optimization that will lower the bandwidth requirements will be to carry in the payload of the feedback packets only the information necessary to identify a flow.

9 Discussion

It is generally agreed that in order to achieve effective congestion management in modern high speed networks the deployment of more than one congestion control mechanism is required. The combination of these mechanisms needs to be able to prevent congestion collapse, achieve approximately fair bandwidth allocations and accommodate traffic sources with different requirements[13].

Shenker [30] has proven that router based congestion control mechanisms are necessary to ensure fairness. While fairness is an important goal of a congestion control mechanism, router based fairness mechanisms, such as fair queuing, are not able to prevent congestion collapse [14, 3, 22]. In order to prevent congestion collapse, additional router based mechanisms are required.

Two alternative, a window-based mechanism with binary feedback (ERAF) and a rate matching mechanism (NBP), have been examined in this paper. Both mechanisms are able to provide approximately fair allocations; ERAF provides proportional fair allocations, NBP when combined with fair queuing is able to provide max-min fair allocations. Another alternative explored in literature is the identification based approach presented by Floyd&Fall [14]. In their approach routers try to identify aggressive flows or high bandwidth flows who do not behave in a TCP-friendly manner. Routers use the packet drop history for a flow, combined with a model of the behavior of a TCP flow, to determine if a flow needs to be punished. The efficiency of this approach depends in a large measure on the accuracy of the identification mechanism, and it is not clear yet how accurate this mechanism is [32]. We experimented also with an implementation of this mechanism provided by Floyd, but we were not able to obtain a behavior comparable with the behavior offered by the mechanisms examined in this paper. Recently Floyd&Mahajan [27] presented a new realization of the identification based mechanism that has a greater accuracy in identifying aggressive flows. We have not experimented with this mechanism.

It seems that while the identification based approach still awaits for a viable realization, the feedback based mechanisms are the best candidates for mechanisms able to prevent congestion collapse. In the following we will examine what are the challenges the adoption of these mechanisms will face.

Deployment: Feedback based congestion control mechanisms require modifications to core routers as well as to edge routers. Because of this, the preferred deployment model is per administrative domain or Autonomous System (AS).

Connectionless Environment: Perhaps the biggest challenge that needs to be addressed in the design of feedback based congestion control mechanisms for packet switched networks is the inherent connectionless nature of these networks. The connection oriented nature of the ATM networks is one of the reasons behind the wider acceptance of feedback based mechanisms.

Depending on the nature of the feedback employed by a mechanism, this problem manifests itself in different forms:

- Binary Feedback - ERAF: Asymmetric routes are common enough in the current Internet. In ERAF feedback packets travel from the congested router towards sources. In the presence of asymmetric routes, the feedback packets may travel towards a different ingress router than the ingress router for the forward flow. Currently ERAF is not able to deal with this situation. However, the preferred deployment model for ERAF is per administrative

[13] Some applications, e.g voice and video, are delay sensitive but loss insensitive. Other applications, e.g. file transfer and web traffic, are delay insensitive but loss sensitive.

domain or AS (autonomous system) basis. In these situations ingress routers that see stray feedback packets might be able to forward them to the correct ingress router.

- Rate Feedback - NBP: It has been argued that in a connectionless environment rate feedback mechanisms are inherently hop-by-hop[14] mechanisms. Rate feedback mechanisms operating at the network periphery, such as NBP, face two challenges: route discovery and feedback implosion. In NBP ingress routers periodically send forward feedback packets towards egress routers. In a connectionless environment, ingress routers have no good way of determining what is the egress router for a flow out of an NBP domain. One possible solution might be to have the ingress routers multicast the forward feedback packets to all periphery routers in the domain. However, due to the periodic nature of the feedback traffic and the potential large number of periphery routers, this approach is likely to be prohibitive in terms of the bandwidth required for the feedback traffic. Another possible solution is to enforce static routes inside an NBP domain.

Feedback Limitation: Feedback based congestion control mechanisms inject additional traffic in the network. Care should be taken [5] that the amount of traffic generated by these mechanisms does not degrade the network performance and does not impact other traffic participants. Under these circumstances, feedback based mechanisms need to be designed in such a way that the generation of feedback traffic is limited[15].

However, the introduction of feedback rate limitation might potentially lower the efficiency of periphery based mechanisms such as ERAF and NBP. This problem can be addressed either by modifying ingress routers to perform flow aggregation or by modifying feedback generating routers to generate feedback preferentially. Since flow aggregation is a hard problem and has the drawback that adaptive flows aggregated with unresponsive flows can be indiscriminately punished by an ingress router, we favor the second solution.

Traffic measurements on the MCI backbone performed in 97 - 98 [10] indicate that 10% of the flows carry approximatively 80% of the traffic. This data is also confirmed by results presented by Floyd [27] for traffic at UC Berkeley. These traffic characteristics can be leveraged in the design of the feedback generator component deployed in routers. Floyd [27] presents a lightweight mechanism that monitors queue drop history and is able to identify high bandwidth flows. Having the routers participating in a feedback based congestion control mechanism generate feedback only for high bandwidth flows is likely to reduce the bandwidth consumed by this traffic while maintaining the efficiency of the mechanism.

10 Conclusion

In this paper we examine the characteristics of router based congestion control mechanisms in the presence of unresponsive traffic. As representatives we evaluate fair queuing mechanisms (FRED, DRR, CSFQ) and feedback based mechanisms (ERAF, NBP).

[14] We have also experimented with a hop-by-hop rate feedback mechanism - NBQ [33]. In NBQ routers send rate feedback information towards the next upstream router. Routers drop packets from incoming flows in an attempt to match the flow departure rate with the flow service rate at the downstream router. We found that this approach severely penalizes connections with multiple congested gateways and does not contain very well bursty traffic sources.

[15] For ERAF we explore feedback traffic limitation mechanisms and their implications in [22].

We found both classes of mechanisms able to shield responsive traffic competing with unresponsive traffic on a single link. We found that the feedback based mechanisms are able to provide better allocations on complex network topologies with multiple congested gateways. In particular, we found the binary feedback mechanisms we evaluate to be able to provide allocations close to proportional fairness.

ERAF, the binary feedback mechanism we examine, is more biased against connections with multiple congested gateways than the rate matching mechanism we examine. Also, ERAF is biased in favor of connections with large packets. This bias in favor of connections with large packets is manifested by all the mechanisms we examine. We also found that the feedback bandwidth required by ERAF to be able to function properly is much smaller than the feedback bandwidth required by a rate matching mechanism.

References

[1] Csfq and fred implementations for ns-2. Available at http://www.cs.cmu.edu/ stoica.

[2] Nbp ns-2 implementation. Available at http://netresearch.ics.uci.edu/nbp.

[3] C. Albuquerque, B. Vickers, and T. Suda. Network border patrol. *Proc. of Infocomm'00*, 2000.

[4] E. Altman, J. Bolot, and F. Bacelli. Discrete-time analysis of rate-based feedback control mechanisms. 2000.

[5] F. Baker. Requirements for IP version 4 routers. RFC 1812, Network Working Group, Jun 1995.

[6] J. Bennett and H. Zhang. WF^2Q: Worst-case fair weighted fair queuing. In *Proc. of INFOCOM'96*, pages 120–8, 1996.

[7] J. Y. L. Boudec. Rate adaptation, congestion control and fairness: a tutorial. Available at http://www.statslab.cam.ac.uk/ frank/pf.

[8] A. Charny, D. Clark, and R. Jain. Congestion control with explicit rate indication.

[9] D. Cheriton. Sirpent: A high performance internetworking approach. *Proc. ACM SIGCOMM Symposium on Commnications Architectures and Protocols*, pages 158–169, September 1989.

[10] K. Claffy, G. Miller, and K. Thompson. The nature of the beast: Recent traffic mesurements from an internet backbone. Available at http://www.caida.org/outreach/papers/Inet98, 1998.

[11] D. D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Transactions on Networking*, 6(4):362–373, 1998.

[12] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queuing algorithm. In *Proc. of SIGCOMM'89*, pages 3–12, 1989.

[13] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. Submitted for publication, 1998.

[14] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, 7(4):458–72, 1999.

[15] S. Floyd, M. Handley, and J. P. et. al. Equation-based congestion control for unicast applications. Technical Report TR-00-03, International Computer Science Institute, March 2000.

[16] S. Floyd, M. Handley, and J. Padhye. A comparison of equation-based and aimd congestion control. *????*, page ?????, 2000.

[17] S. Floyd and V. Jacobson. On traffic phase effects in packet-switched gateways. *Computer Communication Review*, 1991.

[18] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, Aug 1993.

[19] L. N. R. Group. Ucb/lbnl/vint network simulator - ns(version 2). http://wwwmash.cs.berkeley.edu/ns, September 1997.

[20] P. Gupta and N. McKeown. Packet classification on multiple fields, 2000.

[21] E. J. Hernandez-Valencia, L. Benmohamed, S. Chong, and R. Nagarajan. Rate control algorithms for the atm abr service. In *European Transactions on Telecommunications*, pages 7–20, 1997.

[22] C. Iancu and A. Acharya. Eraf - a source quench based mechanism for the regulation of aggressive traffic. Unpublished manuscript., 2000.

[23] R. Jain. Myths about congestion management in high-speed networks.

[24] R. Jain and D. Chiu. Analysisf the increase/decrease algorithms for congestion avoidance in computer networks. *Journal of Computer Networks*, 17(1):1–14, 1989.

[25] F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society 49*, pages 237–252, 1998.

[26] D. Lin and R. Morris. Dynamics of random early detection. In *Proc. of ACM SIGCOMM*, pages 127–137, 1997.

[27] R. Mahajan and S. Floyd. Controlling high bandwidth flows at the congested router. Available at http://www.aciri.org/floyd/papers/red-pd.ps, 2000.

[28] W. Prue and J. Postel. Something a host could do with a source quench. RFC 1016, Jul 1987.

[29] A. Rangarajan and A. Acharya. Eruf: Early regulation of unresponsive traffic. *Proc. of ICNP'99*, pages 117–27, 1999.

[30] S. Shenker. A theoretical analysis of feedback flow control. *Proc. SIGCOMM*, pages 156–165, 1990.

[31] M. Shreedhar and G. Varghese. Efficient fair queuing using deficit round robin. In *Proc. of SIGCOMM'95*, pages 47–57.

[32] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high speed networks. In *Proc. of SIGCOMM'98*, 1998. 118-30.

[33] Y. Tamura, Y. Tobe, and H. Tokuda. Nbq: Neighor-state based queuing for adaptive bandwidth sharing. In *Proc. of IEEE ICNP*, pages 135–144, 1999.

[34] B. Vandalore, S. Fahmy, R. Jain, R. Goyal, and M. Goyal. A definition of generalized fairness and its support in switch algorithms. ATM Forum Document 98-0151, Traffic Management WG, 1998.