Effect of Higher Priority EF Traffic on TCP Throughput and Fairness

Jasleen Kaur and **Harrick M. Vin**

Technical Report TR2001-1
Department of Computer Sciences
University of Texas at Austin
TAY 2.124, Austin, TX 78712-1188, USA

{jks,vin}@cs.utexas.edu

July 5, 2000

Abstract

In this paper, we study the effect on TCP of assigning higher-priority to traffic requesting *Expedited Forwarding* (EF) service in a *Differentiated Services* network. We analyze networks in which (1) EF traffic occupies different fractions of link bandwidth and is bursty at different time-scales; and (2) multiple TCP flows with heterogeneous round trip times share the network with the EF traffic. We find that even in the presence of bursty EF traffic, statistical multiplexing gains allow TCP to utilize most of the available bandwidth. Further, the presence of bursty EF traffic improves the fairness of bandwidth allocation among TCP flows; smaller more frequent bursts yield larger improvements in TCP fairness.

1 Introduction

The Internet has traditionally supported the *best-effort* service model in which the network offers no assurance about when, or even if, packets will be delivered. With the commercialization of the Internet and the deployment of inelastic continuous media applications, however, the best-effort service model is increasingly becoming inadequate. To facilitate the co-existence of these emerging applications with conventional elastic applications, the *differentiated services* architecture has been proposed [22]. In this architecture, traffic entering a network is classified and conditioned at the boundaries of the network, and is assigned to a small set of behavior (or flow) aggregates (also referred to as Per Hop Behaviors—PHB). Recently, several PHBs—such as the Expedited Forwarding (EF) and the Assured Forwarding (AF) PHB—and several end-to-end services—such as the Virtual Leased Line service [14, 22]—have been defined. However, very little is known about what end-to-end performance can be expected for flows that utilize a specific PHB, or how do the implementations for providing service differentiation among the PHBs impact the performance of best-effort flows. In this paper, we take a step towards addressing this question.

To formulate precisely the problem we investigate in this paper, consider the proposal for using the EF PHB to implement the Virtual Leased Line (VLL) service [14, 22]. The VLL service desires the network to provide *guaranteed rate* and *low delay* to flows. It is suggested that a differentiated services network can provide VLL service by following three steps: (1) shape the flows requesting the VLL service to *constant bit rate (CBR)*, and mark packets of the flow as belonging to the EF service class [14]) by appropriately setting the Type-of-Service (ToS) byte in the IP header of the packet [21]; (2) employ admission control algorithms at the routers to ensure that the aggregate rate of flows that request the Virtual Leased Line service does not exceed the capacity reserved for the EF PHB; and (3) provide higher priority to packets requesting EF PHB or implement a fair queuing algorithm to arbitrate access to link bandwidth among the different PHBs. It has been shown that providing higher priority to packets requesting EF PHB yields lower end-to-end delay and jitter to EF traffic [22], as required by the VLL service definition. Further, it is well-known that priority schedulers are simpler to implement than more sophisticated fair queuing algorithms. Hence, implementing VLL service by providing higher priority to EF traffic may be desirable – unless such an approach severely affects the performance of other traffic classes sharing the network with the EF traffic.

In this paper, we attempt to answer this very question: *what is the impact of providing higher priority to EF traffic on the throughput and the fairness of best-effort TCP flows sharing the differentiated services Internet?* We study—through simulations—the effect of different levels of burstiness in the EF traffic on the throughput and fairness of best-effort TCP flows. Our experiments show that:

1. The throughput of an isolated TCP flow is affected severely in the presence of higher priority bursty traffic; the loss in TCP throughput is higher when the EF traffic is bursty at short time-scales. However, the aggregate throughput of multiple TCP flows remains roughly unaffected by the higher priority bursty traffic.

2. Presence of bursty EF traffic improves the fairness of bandwidth allocation among TCP flows; smaller more frequent bursts yield larger improvements in TCP fairness.

We observe that these results hold when the EF traffic occupies several different fractions—in the 5%–30% range—of bottleneck link bandwidth.

Based on these observations, we conclude that providing high priority to EF traffic does not adversely affect the aggregate throughput TCP flows and does, in fact, improve the fairness of bandwidth allocation among the best-effort TCP flows.

The rest of the paper is organized as follows. Section 2 describes our simulation setup. In Sections 3 and 4, respectively, we discuss the effect of higher priority EF traffic on the throughput and fairness of TCP. Section 5 discusses the related work, and Section 6 summarizes our contributions.

2 Experimental Methodology

The objective of our study is to evaluate the effect of higher priority EF traffic on the throughput and fairness of best-effort TCP flows. We have conducted an extensive simulation study using the *NS-2* network simulator [1]. In what follows, we describe our simulation environment, the design of our experiments, and the measures for the performance evaluation.

2.1 Simulation Environment

2.1.1 Network Topology

We consider a network topology depicted in Figure 1; similar network topologies have been used in several prior TCP performance studies [9, 16, 20, 26, 27]. The topology contains two core routers, R_1 and R_2. All the network links have a bandwidth of 40Mbps. One of the input links of router R_1 carries the higher priority EF traffic[1]; this traffic occupies a fixed percentage of the bottleneck link bandwidth. Router R_1 is also connected to 8 other input links that carry the best-effort TCP traffic destined to one of more than 100 sinks connected to router R_2. Routers R_1 and R_2 provide higher priority with respect to link scheduling and link buffer occupancy to the packets belonging to the EF service class. In this setup, at the bottleneck link connecting routers R_1 and R_2, the presence of higher priority EF traffic would affect the queuing delay and packet loss rate experienced by TCP flows.

The topology models the heterogeneity in the round-trip propagation latencies for different TCP flows by assigning different deterministic propagation delays to each incoming link of router R_1 and each outgoing link of router R_2.

[1] Observe that this assumption ensures that the higher priority EF traffic does not occupy any link buffers. In practice, however, EF traffic would arrive at a core router on more than one input link, and would occupy link buffers. Hence, the results presented in this paper provide a conservative estimate on the impact of providing higher priority to EF traffic on the performance of best-effort TCP flows.

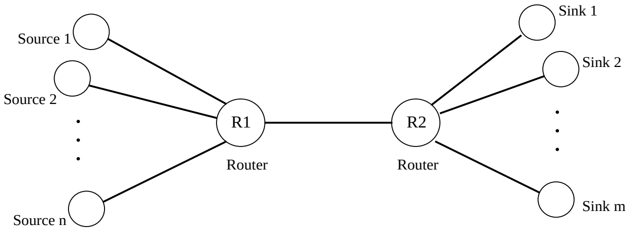


Figure 1: Network Topology

This allows us to model over 800 different TCP round-trip propagation latencies (RTPs)—ranging from around 6ms to 40ms. To maintain the network pipeline full, we provision link buffers in accordance with the *delay-bandwidth product* (i.e., the product of the link bandwidth and the maximum RTP) [6]. Finally, we assume *drop-tail* routers; drop-tail is the most widely deployed buffer management policy in today's Internet routers.

2.1.2 Network Traffic

Flows requesting the EF PHB are shaped to CBR at the ingress routers. However, it has been shown that the aggregation of CBR traffic entering a core router is bursty due to at least two reasons [8, 13, 24, 25]: (1) superpositioning of heterogeneous CBR flows yields inherently bursty traffic; and (2) CBR traffic gets distorted as it traverses through a multi-hop network. The burstiness of the EF traffic depends on several parameters including the heterogeneity in the bit rates and the packet sizes of the individual CBR flows being aggregated; the percentage of the link bandwidth available to the EF class; and the number of input ports in a core router.

We have generated and experimented with several traces of EF traffic for specific network configurations; however, to explore the design space thoroughly and for ease of parameterizing the traffic burstiness, in this paper, we present results obtained by modeling the EF traffic (i.e., the aggregation of CBR flows at core routers) as an *on-off* source, with exponentially distributed on- and off-durations. We experiment with different levels of burstiness in the EF traffic by selecting a wide range of values for the average on- and off-durations. During the on-durations, such a source transmits packets of size 1500B at the link speed. The fraction f of the bottleneck link bandwidth occupied by the EF traffic is given by:

$$f = \frac{T_{on}}{T_{on} + T_{off}}$$

where T_{on} and T_{off}, respectively, are the average on- and off-durations of the on-off source.

As for the TCP traffic, we use *TCP-Reno* [2], the most popular and widely deployed version of TCP in the Internet. TCP-Reno employs the *slow-start*, *congestion-avoidance*, *fast retransmit*, and *fast recovery* algorithms for congestion control [2]. The throughput achieved by a TCP flow is governed by the available network bandwidth, the maximum receiver-advertised window size, and the rate at which data is generated at the source for transmission. To isolate the effects on TCP performance of fluctuations in the available bandwidth and link buffers caused by higher priority EF traffic, we assume: (1) a large value for the receiver-advertised window size; and (2) back-logged TCP data sources, characterizing the long file transfers resulting from ftp and http on the Internet. This ensures that the TCP throughput is limited only by the available network bandwidth. Finally, we assume that the TCP flows send packets of size 536B, which is representative of a large number of TCP flows in the Internet [4].

2.2 Experimental Design

We conduct experiments to study the impact of higher priority EF traffic on the *throughput* and *fairness* of TCP.

1. **Throughput**: To gain basic understanding on the effect of higher priority EF traffic on the throughput of TCP, we first consider a simple network in which the EF traffic shares the bottleneck link with a single TCP flow.

We study the impact on the throughput of the TCP flow of different levels of burstiness in the EF traffic and different round-trip propagation latencies for the TCP flow (see Section 3.1).

We then consider a more realistic case where multiple TCP flows share the bottleneck link with the EF traffic (see Section 3.2). We consider TCP flows with equal and unequal round-trip propagation latencies. For both the settings, we measure the effect on the *aggregate* TCP throughput of (1) different levels of burstiness in the EF traffic and (2) increasing the number of TCP flows sharing the bottleneck link with the EF traffic.

2. **Fairness**: We study the impact of the higher priority EF traffic on TCP fairness in two network settings. First, we consider a network setting in which all best-effort traffic is carried by TCP flows. Second, we consider a network setting, similar to the current Internet, in which the best-effort traffic consists of a mixture of TCP and UDP flows, with UDP traffic occupying roughly 5% of the total best-effort traffic [5, 7]. In both settings, we compare the fairness of bandwidth allocation among the TCP flows in the presence and absence of EF traffic.

We have conducted these experiments in network environments where the EF traffic occupies different percentages—5%, 10%, 20%, and 30%—of the link bandwidth. Due to space constraints, in the following sections, we report only the results obtained from the set of experiments where the EF traffic occupies 30% of the link bandwidth; the conclusions and observations we report hold for all of the percentages.

2.3 Performance Measures

We measure throughput and fairness of TCP flows as follows.

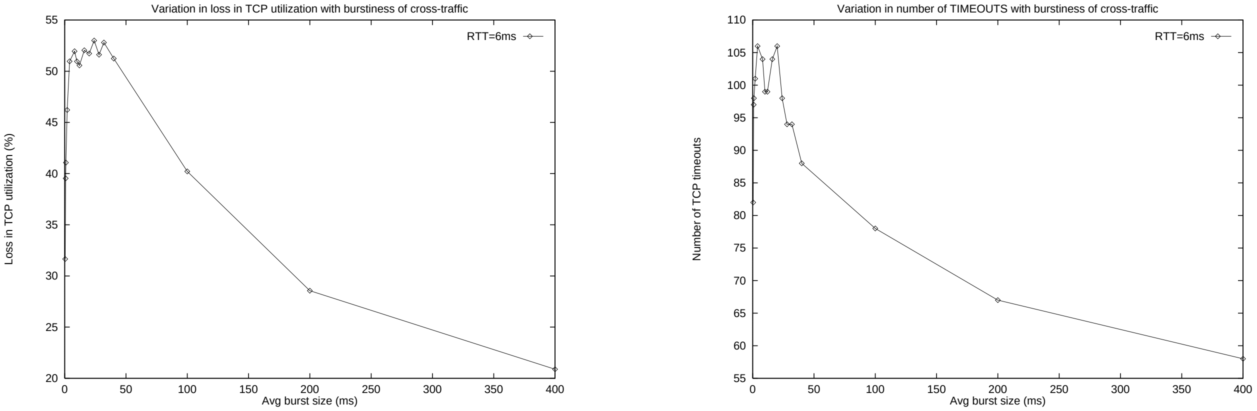
1. *Throughput*: For an individual TCP flow, throughput is defined as the ratio of the total number of bytes received to the total time required for transmission. We compute the network utilization U achieved by TCP flows as the ratio of the aggregate throughput of TCP flows to the bottleneck link bandwidth available to the TCP flows.

   To measure the impact of burstiness in EF traffic on TCP throughput, we consider two types of networks: (1) a *TCP-CBR Network*, in which the TCP flows share the bottleneck link with a higher-priority CBR traffic; and (2) a *TCP-EF Network*, in which the TCP flows share the bottleneck link with a higher-priority bursty EF traffic. The CBR and the EF traffic impose the same average load; hence, in both networks, the bottleneck link bandwidth available to the TCP flows is the same. We then measure the loss in TCP throughput as the difference $(U_{CBR} - U_{EF})$, where U_{CBR} and U_{EF}, respectively, denote the network utilization achieved by TCP flows in the TCP-CBR and TCP-EF networks.

2. *Fairness*: The literature contains two measures—the *fairness index* F [15] and the *min-max ratio* M [18]—for measuring fairness. If $x_i (x_i \geq 0)$ denotes the throughput received by flow i, then:

$$F = \frac{(\sum_{i=1}^{n} x_i)^2}{n * \sum_{i=1}^{n} x_i^2}; \quad M = min_{i,j} \left\{ \frac{x_i}{x_j} \right\}$$

   Observe that F varies from $\frac{1}{n}$ (total unfairness) to 1 (total fairness), whereas M takes values between 0 (total unfairness) and 1 (total fairness). While the fairness index represents the fairness of resource allocation in general, the min-max ratio reflects fairness as perceived by individual users. For instance, if the throughput received by flow k is zero, and all other flows receive equal throughput $x > 0$, then $M = 0$ and $F = 1 - \frac{1}{n}$. When $n \to \infty, F \to 1$. Thus, the fairness index can be infinitely close to its optimal value even though from the perspective of flow k, the network is extremely unfair. Since our objective is to study the impact of higher priority EF traffic on the fairness perceived by individual flow, in the rest of this paper, we will use the *min-max* ratio as the fairness measure.



(a) Loss in TCP throughput

(b) Number of TCP timeouts

Figure 2: Effect of EF burstiness on TCP

3 Throughput

In this section, we first analyze a network in which EF traffic shares the bottleneck link with a single TCP flow, and then generalize the analysis to a more realistic case where multiple TCP flows share the bottleneck link with EF traffic.

3.1 Throughput of a Single TCP Flow

In these experiments, we analyze the impact of higher priority EF traffic on the throughput of a single TCP flow for (1) different levels of burstiness in the EF traffic and (2) different round-trip propagation latencies for the TCP flow.

3.1.1 Effect of Burstiness in EF Traffic

To evaluate the effect of different levels of burstiness in the EF traffic, we vary the average on-duration—in the range 0.4ms (i.e., burst size $\approx$ 1 packet) to 400ms (burst size of $\approx$ 1300 packets) of the on-off source; we maintain the fraction of the link bandwidth occupied by EF traffic at 30% by appropriately changing the average off-duration. We consider a TCP flow with round-trip propagation latency of 8ms.

Figure 2(a) shows that burstiness in the EF traffic can cause significant degradation in TCP throughput; in our experiment, the loss in throughput is as high as 50%. We observe that, in a tail-drop router, arrival of packet bursts for the EF traffic causes bursty packet losses for TCP. In TCP-Reno, if the sender window size is W and if d *consecutive* packets are dropped, then it can be shown that the sender will experience a retransmission timeout if $(d \geq 3) \vee ((W < 10) \wedge (d \geq 2)) \vee ((W < 4) \wedge (d \geq 1))$ — which implies that a loss of 3 (or in some cases even 1 or 2) consecutive packets would result in a retransmission timeout [9]. For every such occurrence, the data source waits for the retransmission timer—which typically corresponds to several hundred milliseconds—to expire before re-sending lost data. Figure 2(b) shows the variation in the total number of timeouts experienced by the TCP source, with the increase in average on-duration of the EF traffic. The similarity in the shape of Figures 2(a) and 2(b) illustrates that the loss in throughput is proportional to the number of retransmission timeouts, implying that the loss is predominantly due to periods of inactivity of the TCP source preceding the expiration of retransmission timers.

Figure 2 illustrates that the number of timeouts experienced by the TCP flow, and hence the loss in TCP throughput, increases initially with increase in the EF burst sizes, and then decreases with further increase in burst sizes. This can be explained using the two factors that affect the number of timeouts experienced by the TCP flow: (1) the frequency of bursts in the EF traffic and (2) the probability of the occurrence of a timeout during each individual EF burst.

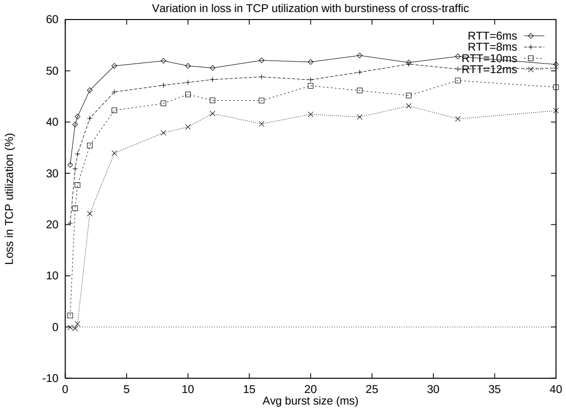


Figure 3: Effect of RTT on loss in TCP throughput

- In our experiments, to maintain the fraction of the link bandwidth occupied by EF traffic at 30%, an increase in T_{on} is coupled with a corresponding increase in T_{off}, which in turn increases $(T_{on} + T_{off})$. Hence, the burst frequency and the total number of EF bursts $N_{bursts} = T_{sim}/(T_{on} + T_{off})$ where T_{sim} is the simulation duration, decreases with increase in T_{on}.

- The probability that an EF burst causes a TCP timeout increases with the increase in average burst size; the larger the burst duration, the greater is the probability that it would coincide with the transmission of a TCP window long enough to fill up the available buffers and cause bursty losses. In fact, the probability of the occurrence of a timeout during an EF burst can be closely approximated by the probability that the duration of the burst exceeds a threshold, τ, that is sufficient to fill up link buffers with the arriving TCP packets and thereby cause a burst of TCP packet losses. Note that τ depends on the rate at which TCP packets arrive at the bottleneck link; the packet arrival rate, in turn, is determined by the number of TCP flows being multiplexed on the link, their round-trip propagation latencies, and TCP window dynamics. For the on-off model of the EF traffic, the probability $P_{timeout}$ that an EF burst causes a TCP timeout is thus given by:

  $$P_{timeout} = P(X_{on} \geq \tau) = e^{-\frac{\tau}{T_{on}}}$$

  where X_{on} is an exponentially-distributed random variable with mean T_{on}.

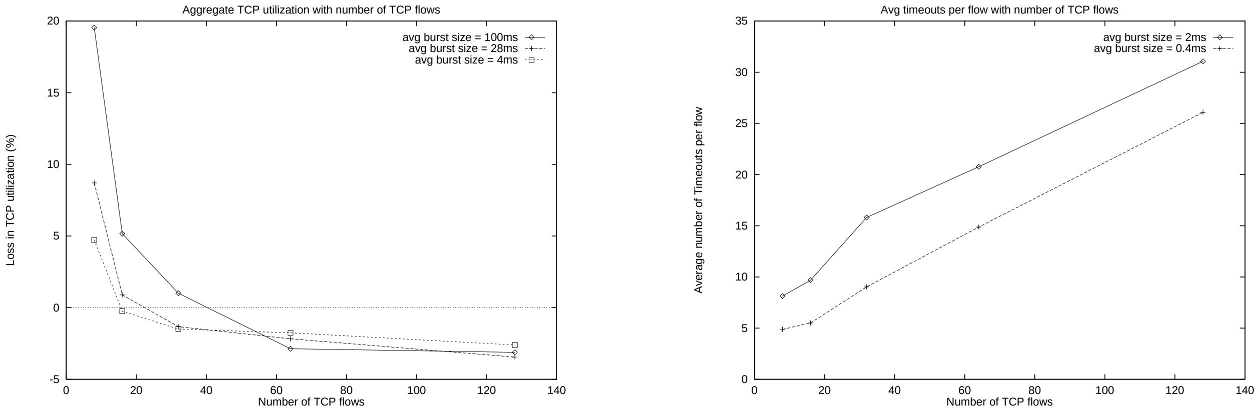
The expected number of TCP timeouts is the product $N_{bursts} * P_{timeout}$; this value increases initially with increase in T_{on}, and then decreases with further increase in T_{on}. This explains the graph in Figure 2(b). In fact, the maximum number of timeouts are observed with $T_{on} = \tau$. From the above discussion, we derive the following conclusion.

Conclusion 1 *The throughput of an isolated TCP flow is severely affected in the presence of higher priority bursty EF traffic; the loss in TCP throughput first increases and then decreases with increase in the average burst size.*

3.1.2 Effect of TCP RTP

We repeat the above experiment with different values of TCP round-trip propagation latencies—ranging from 6ms to 12ms. Figure 3 illustrates that the loss in TCP throughput reduces with increase in the TCP round-trip propagation latency.

To understand this, observe that for a fixed value of average burst size (T_{on}) the value of N_{bursts} remains fixed. However, the larger the round-trip propagation latency, the slower the TCP window dynamics, and hence the smaller the probability that a burst of a given size coincides with the transmission of a sufficient number of TCP packets that would result in a TCP timeout. Hence, with increase in TCP RTP, the expected value of τ—the burst duration



(a) Aggregate TCP throughput

(b) Number of Timeouts per flow

Figure 4: Effect of bursty EF traffic on TCP throughput with large number of TCP flows

required to cause a TCP timeout—increases, which yields smaller values of $P_{timeout}$ for the same level of burstiness (T_{on}) in EF traffic.

Conclusion 2 *The larger the round-trip propagation latency, the smaller is the degradation in TCP throughput in the presence of higher priority bursty traffic.*

3.2 Aggregate Throughput of TCP Flows

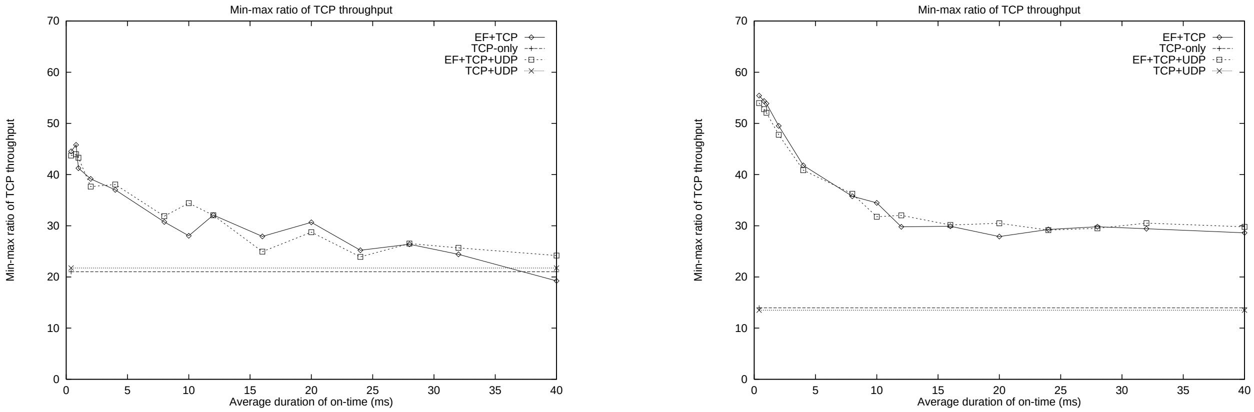
We now consider a more realistic case where multiple TCP flows share the bottleneck link with the EF traffic, and repeat the set of experiments described in Section 3.1.1 with 8, 16, 32, 64, and 128 TCP flows. The round-trip propagation latencies for the TCP flows are chosen randomly from 8 adjacent equal-sized intervals spanning the range from 12ms to 26ms.

Figure 4(a) plots the loss in *aggregate* TCP throughput as a function of (1) the number of TCP flows in the network and (2) the average burst-durations of the EF traffic. It illustrates that the loss in aggregate TCP throughput due to burstiness in the EF traffic becomes insignificant in networks that multiplex a large number (≥ 20) of flows. Figure 4(b), on the other hand, shows that the number of timeouts per TCP flow increases with increase in the number of TCP flows.

The increase in the number of timeouts per TCP flow is caused by an increase in the probability $P_{timeout}$ that a burst results in a retransmission timeout for a TCP flow. This is because, with increase in the number of TCP flows competing for the bottleneck link, the available link buffers at the router fill up faster. Consequently, the EF burst-duration τ necessary to fill up link buffers and cause bursty packet losses for TCP reduces, leading to a greater value of $P_{timeout}$ for a TCP flow.

The insignificance of the loss in aggregate throughput, on the other hand, can be attributed to the gains due to *statistical multiplexing*. Increasing the number of TCP flows increases the number of flows that are able to utilize the link bandwidth while some other TCP flows are silent waiting for their retransmission timers to expire. This ensures that the aggregate TCP throughput is unaffected.

Conclusion 3 *The aggregate throughput of a large number of TCP flows remains roughly unaffected in the presence of higher priority EF traffic.*



(a) Equal RTPs

(b) Unequal RTPs

Figure 5: Effect of burstiness in EF traffic on TCP fairness

4 Fairness

In this section, we study the impact of the higher priority EF traffic on TCP fairness in two network settings: (1) a network (denoted as TCP-only) in which all best-effort traffic is carried by TCP flows, and (2) a network (denoted by TCP+UDP), similar to the current Internet, where the best-effort traffic consists of a mixture of TCP and UDP flows, with UDP traffic occupying roughly 5% of the total best-effort traffic [5, 7]. In both settings, we compare the fairness of bandwidth allocation among the TCP flows in the presence and absence of higher priority EF traffic.

For these experiments, we consider two scenarios for TCP flows: (1) all TCP flows have the same round-trip propagation latency (8ms), and (2) TCP flows have unequal round-trip propagation latencies, chosen randomly from 8 equal-sized adjacent intervals (ranging from 12ms to 26ms), such that all flows have different round-trip propagation latencies and equal number of samples are chosen from each interval. We repeat the experiment with different combinations of round-trip propagation latencies, and report the average of the observed *min-max* fairness. As for the TCP+UDP network, we assume that the UDP traffic occupies 5% of the link bandwidth [5, 7]; the UDP traffic aggregate is modeled as an *on-off* source with an average burst size of $4ms$[2].

In what follows, we first discuss the effect of different levels of burstiness in the EF traffic on TCP fairness (Section 4.1) and then discuss how these observations are affected with changes in the number of TCP flows multiplexed with the EF traffic (Section 4.2).

4.1 Effect of Burstiness in EF Traffic

To evaluate the effect of different levels of burstiness in the EF traffic, we vary the average on-duration—in the range 0.4ms (i.e., burst size $\approx$ 1 packet) to 400ms (burst size of $\approx$ 1300 packets)—of the on-off source; we maintain the fraction of the link bandwidth occupied by EF traffic at 30% by appropriately changing the average off-duration. We consider a network with 64 TCP flows sharing the bottleneck link with the EF traffic.

Figures 5 plots the min-max fairness for TCP flows observed in the TCP-only, TCP+UDP, TCP-only+EF, and TCP+UDP+EF networks; Figures 5(a) and 5(b), respectively, represent the results for the TCP flows with equal and unequal round-trip propagation latencies. The graphs illustrate that:

1. Both TCP-only and TCP+UDP networks can be quite unfair (with respect to the min-max fairness measure). In our experiments, the observed throughput for TCP flows differed by a factor as large as 30. Even for the setting where TCP flows have equal round-trip propagation latencies, the observed throughput for TCP flows differed by a factor as large as 5-10.

[2] We have experimented with values of average burst duration ranging from $0.4ms$ to $400ms$, and the results do not change significantly

There are two factors that contribute to this unfairness. First, the throughput achieved by a TCP flow is inversely proportional to the square-root of its round-trip propagation latency [17]; consequently, TCP flows with smaller round-trip propagation latencies achieve larger throughput. Second, in our experiments, we observe the existence of *phase effects* in TCP; with phase effects, throughput of TCP flows can differ considerably even for flows with equal round-trip propagation latencies. Phase effects arise because packets leaving the bottleneck link induce correlation between the packet arrival and the freeing up of buffers at the bottleneck link. This causes some TCP sources to operate in a synchronized manner; these sources synchronously re-initiate data transfer after every retransmission timeout period, increase their window sizes, cause congestion to each other, and synchronously experience a timeout again. This allows other TCP flows to obtain an unfair share of the bottleneck link bandwidth.

Similar effects in TCP-Tahoe were observed and reported in [11]. In [11], it was shown that for TCP-Tahoe, the phase effects could yield throughput for flows that differ by a factor of 5-10. In our experiments, with TCP-Reno, we observed larger differences in the throughput of TCP flows, with some flows getting virtually no throughput at all. We believe that the observed increase in unfairness over TCP-Tahoe is due to the larger number of retransmission timeouts experienced by TCP-Reno flows on experiencing bursty packet losses.

2. For both the equal and unequal TCP round-trip propagation latency settings, min-max fairness for TCP flows improves considerably in the presence of EF traffic.

   The smaller the burst-duration in the EF traffic, the larger the improvement in the min-max fairness for TCP flows. This is because, the smaller the average burst-duration, the larger is the frequency of bursts in the EF traffic, and hence the greater is the likelihood that TCP flows utilizing excessive bandwidth experience packet losses and reduce their transmission rates.

   Figure 6 shows, for EF traffic with $T_{on} = 1$ms, the frequency distribution of the normalized throughput of TCP flows; throughput of a flow is normalized by its fair-share of the bottleneck link bandwidth. In particular, if x_i denotes the throughput of flow i, and if the bandwidth C available to TCP is shared by n flows, then the normalized throughput for flow i is defined as $x_i/(C/n)$. Figure 6 illustrates that, both for the equal and unequal round-trip propagation latency settings, the presence of higher priority EF traffic not only reduces the maximum difference in the throughput obtained by TCP flows but also increases the percentage of flows that receive bandwidth close to their fair share.

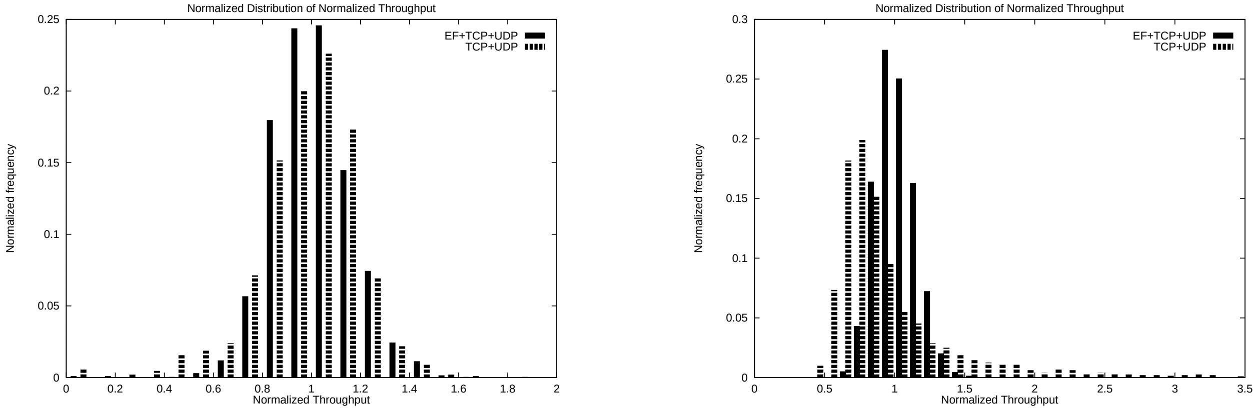
These observations lead us to the following conclusion:

Conclusion 4 *Higher priority EF traffic significantly improves the fairness of TCP flows. The smaller the average burst size, the greater is the improvement in the fairness.*

4.2 Effect of Number of TCP Flows

In this section, we discuss how the observations discussed in Section 4.1 are affected with changes in the number of TCP flows multiplexed with the EF traffic. For these experiments, we consider EF traffic with the average on-duration of 1 ms. Further, we analyze environments with 8, 16, 32, 64, or 128 TCP flows sharing the bottleneck link with the EF traffic.

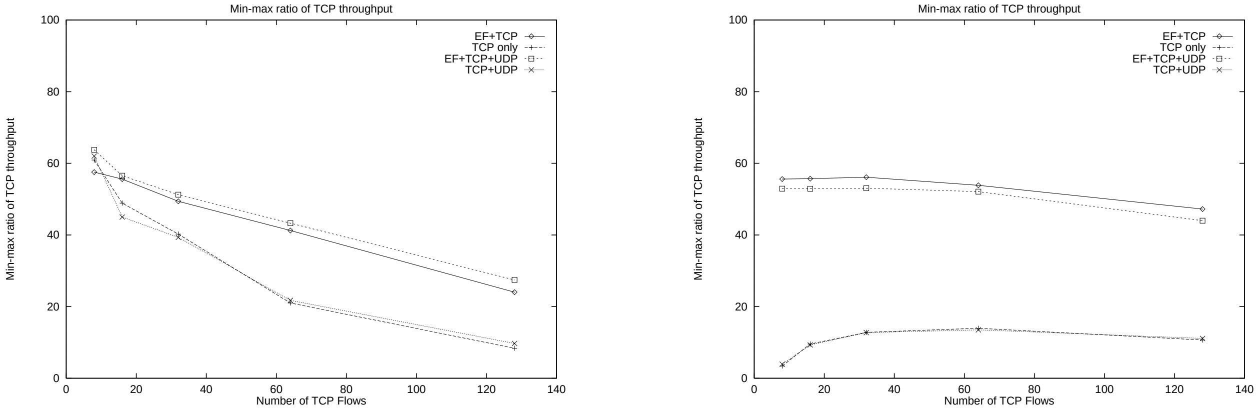
Figures 7 plots the min-max fairness for TCP flows observed in the TCP-only, TCP+UDP, TCP-only+EF, and TCP+UDP+EF networks. Figure 7(a) illustrates that, for TCP flows with equal round-trip propagation latency, in TCP-only and TCP+UDP networks, the min-max fairness of bandwidth allocation decreases quite significantly with increase in the number of TCP flows. Even in the presence of higher priority EF traffic, the min-max fairness decreases with increase in the number of TCP flows; however, the rate of reduction in the min-max fairness is smaller. Figure 7(b), on the other hand, illustrates that for TCP flows with unequal round-trip propagation latencies,



(a) Equal RTPs

(b) Unequal RTPs

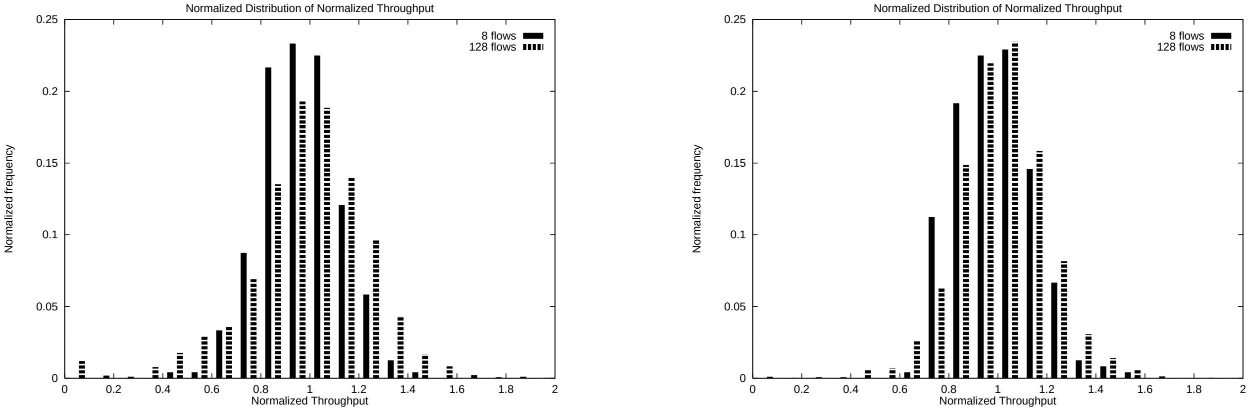
Figure 6: Frequency distribution of the normalized throughput of TCP flows (with average on-duration of 1ms for EF traffic)



(a) Equal RTPs

(b) Unequal RTPs

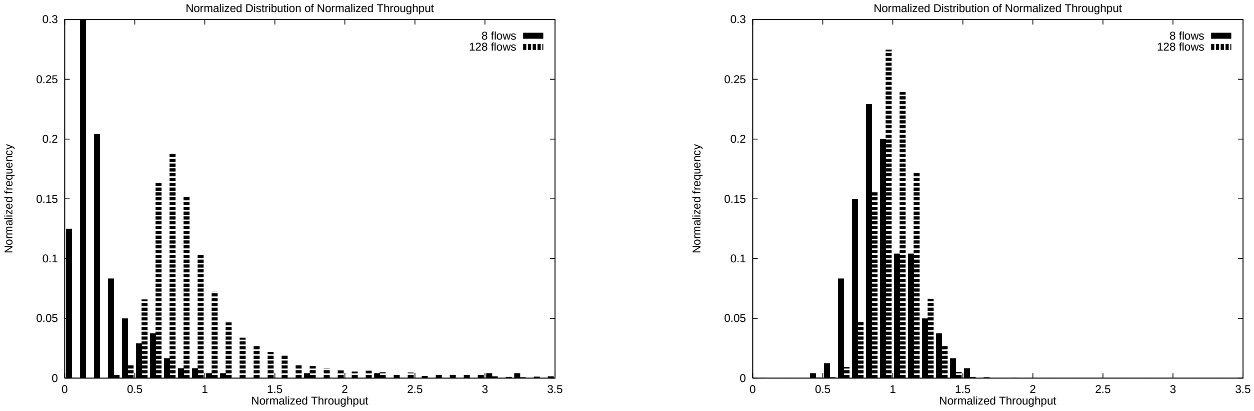
Figure 7: Effect of increasing the number of TCP flows on fairness



(a) TCP+UDP

(b) TCP+UDP+EF

Figure 8: Frequency distribution of the normalized throughput of TCP flows with equal round-trip propagation latency



(a) TCP+UDP

(b) TCP+UDP+EF

Figure 9: Frequency distribution of the normalized throughput of TCP flows with unequal round-trip propagation latencies

both in the presence and absence of EF traffic, the effect on the min-max fairness of increasing the number of TCP flows is marginal.

Figures 8 and 9, respectively, show the frequency distribution of the normalized throughput of TCP flows with equal and unequal round-trip propagation latencies. Figures 8(a) and 9(a) represent the results obtained from the TCP+UDP network, while Figures 8(b) and 9(b) represent the results from the TCP+UDP+EF network.

Figure 8(a) illustrates that, in the TCP+UDP network, increase in the number of TCP flows not only increases the maximum difference in the throughput obtained by TCP flows, but also reduces the percentage of flows that receive bandwidth close to their fair share. For the TCP+UDP+EF network, on the other hand, although the the maximum difference in the throughput obtained by TCP flows increases with increase in the number of TCP flows, the percentage of the flows that receive bandwidth close to their fair share does not change appreciably (see Figure 8(b)).

Figure 9(a) illustrates that, in the TCP+UDP network, increase in the number of TCP flows significantly improves the percentage of flows that receive bandwidth close to their fair share, even though the maximum difference in the throughput obtained by TCP flows remains large. Figure 9(b), on the other hand, demonstrates that in the presence of EF traffic, the fairness of bandwidth allocation among TCP flows changes marginally with increase in number of TCP flows.

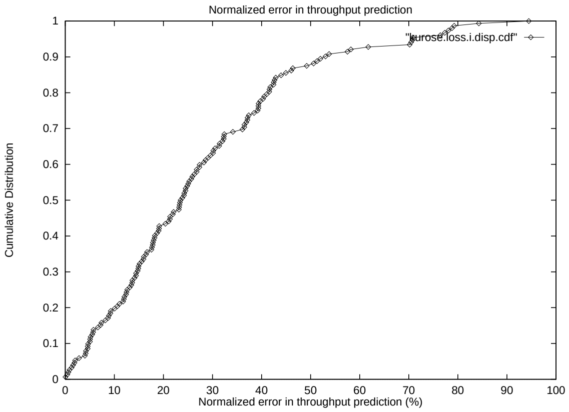


Figure 10: Error in prediction by model in [23] in our scenarios

5 Related Work

The prior research related to the work presented in this paper can be grouped into three categories: (1) studies describing different versions of the TCP protocol, (2) performance evaluation studies of TCP, and (3) studies that evaluate performance of services in differentiated services networks.

Over the past decade, several different versions of TCP—such as TCP-Vegas, TCP-Tahoe, TCP-Reno, TCP-SACK—have been proposed [2, 3, 9, 19]. For our evaluation, we have used the TCP-Reno, the most popular and widely deployed version of TCP in the Internet. TCP-Reno is known to yield poor performance under bursty packet losses, since bursty losses lead to retransmission timeouts in Reno. TCP-SACK has been proposed to address this limitation [19]. We have repeated our experiments with TCP-SACK; our results indicate that the observations presented in this paper hold even under TCP-SACK.

Phase effects have been reported and analyzed in [11]. The authors observe that this effect can be considerably reduced if sufficient amount of randomization is added into the network; such randomness can be introduced either by using RED [12] instead of the drop-tail policy for buffer management in routers, or by ensuring that 15% or more of the link bandwidth is occupied by *telnet* traffic with characteristically different packet sizes than *ftp* traffic. In this paper, we have presented results for drop-tail routers, since drop-tail is the most widely deployed buffer management policy in today's Internet routers. Evaluating the effects of assigning higher priority to EF traffic on TCP throughput and fairness with RED routers is a topic for future research.

In the recent past, several analytical models for predicting the performance of TCP flows have been developed [10, 16, 20, 23]. These models attempt to characterize the throughput an individual TCP flow in terms of packet loss probability, average round-trip time, and packet-size. All of these studies assume a loss distribution model, which limit the applicability of the models to certain types of networks. For instance, models that assume independent, uncorrelated losses are not applicable to our environment; in the presence of bursty EF traffic, losses are correlated and lead to retransmission timeouts in TCP. The throughput prediction model in [23] does incorporate the effect of retransmission timeouts. However, when applied to the TCP flows in our setting, 80% of the predictions were off from the observed throughput by at least 10%. Figure 10 plots the cumulative distribution of the prediction error, normalized with respect to the observed throughput.

[14] describes a preliminary study on the performance of flows (including TCP flows) that subscribe to the *Virtual Leased Line* service (implemented using the EF PHB). While the use of priority scheduling for providing low delay and jitter to EF flows is advocated in this study, it does not provide any evaluation of the effect of such priority scheduling mechanism on the performance of non-EF TCP flows.

6 Concluding Remarks

In this paper, we study the effect on TCP of assigning higher-priority to packets belonging to the *Expedited Forwarding* (EF) class in a *Differentiated Services* networks. We study—through simulations—the effect of different levels of burstiness in the EF traffic on the throughput and fairness of best-effort TCP flows. Our experiments show that:

1. The throughput of an isolated TCP flow is severely affected in the presence of higher priority bursty traffic; the loss in TCP throughput is higher when the EF traffic is bursty at short time-scales. However, the aggregate throughput of multiple TCP flows remains roughly unaffected by the higher priority bursty traffic.

2. The presence of bursty EF traffic improves the fairness of bandwidth allocation among TCP flows; smaller more frequent bursts yield larger improvements in TCP fairness.

Based on these observations, we conclude that providing high priority to EF traffic does not adversely affect the aggregate throughput TCP flows and does, in fact, improve the fairness of bandwidth allocation among the best-effort TCP flows.

References

[1] NS (Network Simulator). 1999. http://www-mash.cs.berkeley.edu/ns.

[2] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. April 1999. Internet RFC 2581.

[3] L.S. Brakmo, S.W. O'Malley, and L.L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In *Proceedings of ACM Sigcomm*, August 1994.

[4] CAIDA. Packet sizes and Sequencing. March 1998. http://www.caida.org/Learn/Size.

[5] CAIDA. Traffic WorkLoad Overview. June 1999. http://www.caida.org/Learn/Flow/tcpudp.html.

[6] H. M. Chaskar, T. V. Lakshman, and U. Madhow. TCP Over Wireless with Link Level Error Control: Analysis and Design Methodology. In *IEEE/ACM Transactions on Networking*, October 1999.

[7] K.C. Claffy. Internet Measurement and Data Analysis: Topology, Workload, Performance and Routing Statistics. In *Proceedings of NAE'99*, 1999. http://www.caida.org/outreach/papers/Nae/.

[8] A. DeSimone. Generating Burstiness in Networks: A simulation Study of Correlation Effects in Networks of Queues. *ACM Computer Communication Review*, pages 24–31, 1991.

[9] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and Sack TCP. *ACM Computer Communication Review*, 26(3), July 1996.

[10] S. Floyd. Connections with Multiple Congested Gateways in Packet-Switched Networks, Part1: One-way Traffic. *ACM Computer Communication Review*, 21(5), October 1991.

[11] S. Floyd and V. Jacobson. On Traffic Phase Effects in Packet-Switched Gateways. In *Internetworking: Research and Experience*, volume 3, pages 115–156, September 1992.

[12] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. In *IEEE/ACM Transactions on Networking*, August 1993.

[13] M. Grossglauser and S. Keshav. On CBR Service. In *Proceedings of INFOCOM'96*, pages 129–137, March 1996.

[14] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. June 1999. Internet RFC 2598.

[15] R. Jain, D. W. Chiu, and W. R. Haew. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. September 1984. DEC Research Report TR-301.

[16] T. V. Lakshman and U. Madhow. The Performance of TCP/IP for Networks with High Bandwidth-delay Products and Random Loss. In *IEEE/ACM Transactions on Networking*, June 1997.

[17] J. Mahdavi and S. Floyd. TCP-friendly Unicast Rate-based Flow Control. January 1997. Note sent to end2end-interest mailing list.

[18] M. Marsan and M. Gerla. Fairness in Local Computing Networks. In *Proceedings of IEEE International Conference on Communication ICC82*, June 1982.

[19] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgement Options. October 1996. Internet RFC 2018.

[20] M. Mathis, J. Semske, J. Mahdavi, and T. Ott. The macroscopic behaviour of the TCP congestion avoidance algorithm. *ACM Computer Communication Review*, 27(3), July 1997.

[21] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. December 1998. Internet RFC 2474.

[22] K. Nichols, V. Jacobson, and L. Zhang. A Two-bit Differentiated Services Architecture for the Internet. November 1997. ftp://ftp.ee.lbl.gov/papers/dsarch.pdf.

[23] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughout: A Simple Model and its Emperical Validation. In *Proceedings of ACM SIGCOMM'98*, pages 303–314, 1998.

[24] J.W. Roberts and J.T. Virtamo. The Superposition of Periodic Cell Arrival Streams in an ATM Multiplexer. *IEEE Transactions on Communications*, 39(2):298–303, February 1991.

[25] J. Sahni, P. Goyal, and H. M. Vin. Scheduling CBR Flows: FIFO or Fair-queueing? In *Proceedings of the Workshop on Network and Operating System Support for Digital Audio and Video, June 1999*, 1999.

[26] S. Shenker, L. Zhang, and D. D. Clark. Some observations on the dynamics of a congestion control algorithm. *ACM Computer Communication Review*, pages 30–39, October 1990.

[27] L. Zhang, S. Shenker, and D. D. Clark. Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic. *Proceedings of ACM SIGCOMM'91*, pages 133–147, 1991.