Transient Behaviors of TCP-friendly Congestion Control Protocols *

Y. Richard Yang, Min Sik Kim, Simon S. Lam
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712–1188
{yangyang,minskim,lam}@cs.utexas.edu

TR2000–23
July, 2000

Abstract

We investigate the fairness, smoothness, responsiveness, and aggressiveness of TCP and three representative TCP-friendly congestion control protocols: GAIMD, TFRC, and TEAR. The properties are evaluated both analytically and experimentally by studying protocol responses to three network environment changes. The first environment change is the inherent fluctuations in a stationary network environment. We consider three types of sending rate variations: smoothness, short-term fairness, and long-term fairness. For a stationary environment, we observe that smoothness and fairness are positively correlated. We derive an analytical expression for the sending rate coefficient of variation for each of the four protocols. These analytical results match well with experimental results. The other two environment changes we study are a step increase of network congestion and a step increase of available bandwidth. Protocol responses to these changes reflect their responsiveness and aggressiveness, respectively.

*Research sponsored in part by NSF grant no. ANI–9977267 and ONR grant no. N00014–99–1–0402. Experiments were performed on equipment procured with NSF grant no. CDA–9624082.

1 Introduction

In a shared network, such as the Internet, end systems should react to congestion by adapting their transmission rates to avoid congestion collapse and keep network utilization high [8]. The robustness of the current Internet is due in large part to the end-to-end congestion control mechanisms of TCP [14]. However, while TCP congestion control is appropriate for applications such as bulk data transfer, many real-time applications would find halving the sending rate of a flow to be too severe a response to a congestion indication as it can noticeably reduce the flow's user-perceived quality [22].

In the past few years, many unicast congestion control schemes have been proposed and investigated with the objective of finding an alternative to TCP congestion control [13, 16, 24, 21, 6, 19, 22, 17, **?**, 10, 27, 20]. Since the dominant Internet traffic is TCP-based [23], it is important that new congestion control schemes be *TCP-friendly*. By this, we mean that the sending rate of a non-TCP flow should be approximately the same as that of a TCP flow under the same conditions of round-trip time and packet loss rate [16, 4].

However, evaluations of these new protocols have been focused mainly on fairness. Two methods were proposed to establish the fairness of a protocol. The first is the Chiu and Jain's phase space method [7], which can be used to show that a protocol will converge asymptotically to a fair state, ignoring such operational factors as randomness of the loss process and timeouts. The second method is to show that the long-term mean sending rate of a protocol is approximately the same as that of TCP. However, it has been observed in experiments [27, 10, 9] that flows with TCP-friendly long-term mean sending rates can still have large rate variations when loss rate is high.

Furthermore, fairness is only one of several desirable properties of a TCP-friendly congestion control protocol. We identify four desired properties: 1) *fairness*: small variations over the sending rates of competing flows, 2) *smoothness*: small sending rate variations over time for a particular flow in a stationary environment, 3) *responsiveness*: fast deceleration of protocol sending rate when there is a step increase of network congestion, and 4) *aggressiveness*: fast acceleration of protocol sending rate to improve network utilization when there is a step increase of available bandwidth.

The objective of this paper is to evaluate these properties by analytically and experimentally studying the transient behaviors of several TCP-friendly congestion control protocols. Proposed congestion control schemes in the literature fall into two major categories: AIMD-based [13, 21, 6, 19, 27, 20] and formula-based [16, 24, 22, 17, 10]. For our study, we select TCP [14] and GAIMD [27] as representatives of the AIMD-based schemes. GAIMD generalizes TCP by parameterizing the congestion window increase value and decrease ratio. That is, in the congestion avoidance state, the window size is increased by α per window of packets acknowledged and it is decreased to β of the current value whenever there is a triple-duplicate congestion indication. In our experiments, we chose the parameter values $\alpha = 0.31$ and $\beta = 7/8$ for our representative TCP-friendly GAIMD protocol. In what follows, we use GAIMD to refer to GAIMD with these parameter values. We select TFRC [10] as a representative of the formula-based schemes. In addition to these three protocols, we also select TEAR [20] which uses a sliding window to smooth sending rates.

The first environment change we study is the inherent network fluctuations in a stationary environment. We evaluate three types of sending rate variations: smoothness, short-term fairness, and long-term fairness. For a stationary environment, we observe that smoothness and fairness are positively correlated. To quantify the smoothness of a flow, we derived an analytical expression for the sending rate coefficient of variation (CoV) for each of the four protocols. We found that our analytical results match with experimental results very well. We observe that with increasing loss rate, smoothness and fairness become worse for all four protocols. However, their deteriorating speeds are different. In particular, at 20% loss rate, TFRC CoV increases to be the highest. TEAR maintains a relatively stable smoothness and fairness performance, but it scores the lowest in experiments on responsiveness and aggressiveness (see below). Also, while TFRC and TEAR have smoother sending rates than those of TCP and GAIMD, they have undesirable behaviors at high loss rate, i.e., TFRC sending rate dropping to almost zero and TEAR sending rate being too high compared to TCP.

The second environment change we study is a step increase of network congestion. Protocol responses to this change reflect their responsiveness. In our experiments, TCP is the most responsive of the four protocols. However, TCP overshoots and has to recover from the overshot state. TFRC and TEAR can gradually change to a new state. We also found a potential protocol misbehavior with TEAR. This shows that our evaluation framework can be a valuable tool for studying protocol responses and detecting undesirable protocol behaviors.

The third environment change we study is a step increase of available bandwidth. Protocol responses to this change reflect their aggressiveness. In our experiments, we found that TCP is the most aggressive of the four protocols to use extra bandwidth. Again TCP overshoots. GAIMD and TFRC with history discounting have similar aggressiveness. TEAR is the slowest to utilize extra bandwidth.

The balance of this paper is organized as follows. In Section 2 our evaluation methodology is discussed. In Section 3 we evaluate protocol responses to fluctuations of the network loss process in stationary environments. In Section 4, we evaluate protocol responses to a step increase of network congestion. Protocol responses to a step increase in available bandwidth are shown in Section 5. Our conclusions are in Section 6.

2 Evaluation methodology

2.1 Loss models

The loss process is a major factor determining the performance of a responsive congestion control protocol. In our experimental evaluation, we use four simple and representative loss models. We distinguish between loss models for high multiplexing environments and low multiplexing environments. By high multiplexing environment, we mean that loss is relatively insensitive to the sending rate of the flow under study. This is intended to be a model for backbone routers. By low multiplexing environment, we mean that loss is somewhat sensitive to the sending rate of the flow.

Our first loss model is deterministic periodic loss. Though this model may be unrealistic, it is simple and protocol responses for this model are representative and clear.

The second loss model is Bernoulli loss. In this model, each packet is lost with probability p, which is identical and independent for all packets. We consider this model to be representative for some high multiplexing environments. In today's Internet, packets are dropped by routers without regard to which flows they belong to when buffers overflow. For drop-tail routers, packet losses can be correlated. However, a number of studies [3, 26, 28] show that loss bursts in the Internet are short and any loss correlation does not span long, typically less than one RTT.

The third loss model of a high multiplexing environment is the loss process when background traffic consists of ON/OFF sources. Since the dominant traffic in the Internet is web-like traffic, we believe that it is important to model the effects of competing web-like traffic (short TCP connections and some UDP flows). It has been reported that WWW-related traffic tends to be self-similar in nature [18]. Willinger et al. shows that self-similar traffic can be generated by using several ON/OFF UDP sources whose ON/OFF times are drawn from a heavy-tailed distribution such as the Pareto distribution [25].

The fourth loss model is the loss process when N flows are competing with each other. We consider this loss model as representative of a low multiplexing environment.

2.2 Evaluation configurations

The network topology for our experiments is the well-known single bottleneck ("dumbbell") as shown in Figure 1. The bottleneck link from R1 to R2 is configured to have a bandwidth of 2.5Mbps, propagation delay 30ms and buffer size of 50 packets with packet size 1000 bytes. We conduct all experiments with both drop-tail and RED bottleneck link. However, we only report drop-tail based results. The results for RED are similar unless we explicitly point them out. All access links have a delay of 10ms. In all experiments, the access links are sufficiently provisioned to ensure that packet drops due to congestion occur only at the bottleneck link.

To implement the deterministic and Bernoulli loss models, we insert a loss module into the link from R1 to R2.

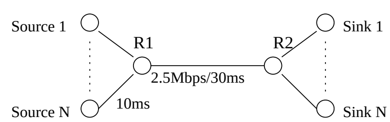

Figure 1: Network topology

In our experiments we use TCP/Reno and GAIMD based on TCP/Reno. TFRC is based on the code from NS June 12th, 2000 snapshot. In our initial set of experiments, we used the TEAR code from the authors' web site. However, we found that the timeout mechanism described in their paper [20] was not implemented in the code. Therefore, we modified their code to implement timeout. For most of the experiments, differences between the modified and unmodified versions are small. However, there are big differences in some experiments; in these cases, we will point them out in experiment descriptions.

To avoid phase effects [11] that mask underlying dynamics of the protocols, we set the *overhead* parameter of TCP, GAIMD, and TFRC to a small non-zero value to introduce randomizations.

3 Responses to stationary loss process

We first investigate protocol responses in stationary environments. The protocol properties we study in this section are smoothness and fairness.

3.1 Performance metrics

Both smoothness and fairness are measures of sending rate variations. The classic measure of variations is coefficient of variation (CoV). CoV depends on measurement time scale. Generally, the longer the time scale is, the smaller the CoV is. We consider three types of CoVs. Two of them have a measurement time scale of round-trip time; the third has a time scale of multiple round-trip times.

3.1.1 Three types of coefficient of variation

1. *Smoothness* CoV_{time}. Consider any solid dot in Figure 2a, which represents the sending rate during a round-trip time of a specific flow. We define CoV_{time} as the coefficient of variation of this time series. CoV_{time} measures the smoothness of a flow.

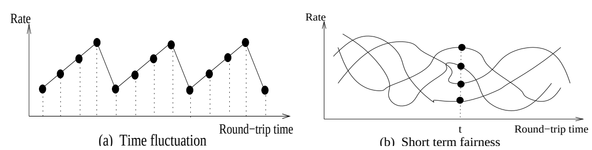

Figure 2: CoV_{time} and CoV_{sf}

2. *Short-term fairness* CoV_{sf}. The solid dots in Figure 2b are samples of the sending rates of several competing flows during the same round-trip time. The coefficient of variation CoV_{sf} of this data series measures short-term fairness among competing flows. For a stationary process the time distribution and sample distribution are equal (assuming that competing flows are i.i.d. processes), and we have

$$CoV_{\text{time}} = CoV_{\text{sf}} \tag{1}$$

   The implication of Equation (1) is that *smoothness and short-term fairness are positively correlated*.

3. *Long-term fairness* CoV_{lf}. Define an epoch as a time interval long enough such that sending process of a flow between epochs are independent and identically distributed. For flow i, let S_{ij} denote its average sending rate during the jth epoch, and define $R_i(n) = \frac{\sum_{j=1}^{n} S_{ij}}{n}$ as its average sending rate in n epochs. Since we assume the random variables $\{S_{ij}\}_{j=1}^{n}$ are i.i.d., by the central limit theorem, the distribution of $R_i(n)$ can be approximated by normal distribution for a large n, and we have

$$CoV[R_i(n)] \approx \frac{CoV[\{S_{ij}\}_{j=1}^{n}]}{\sqrt{n}} \tag{2}$$

Consider K i.i.d. competing flows. Then the coefficient of variation CoV_{lf} of the data series $\{R_i(n)\}_{i=1}^K$ reflects flow long-term fairness. We know that $CoV_{\text{lf}} = CoV[R_i(n)]$. Therefore, from Equation (2) we see again a positive correlation between sample coefficient of variation CoV_{lf}, which reflects long-term fairness, and time coefficient of variation, $CoV[\{S_{ij}\}_{j=1}^n]$, which measures flow smoothness in a time scale of epoch.

Thus we conclude that generating smoother traffic (small CoV_{time}) will improve both short-term fairness and long-term fairness. In our experimental evaluations, instead of using CoV_{sf} to measure short-term fairness, we follow [15] and use fairness index F, defined as $\frac{(\sum x_i)^2}{K \sum x_i^2}$, where $\{x_i\}_{i=1}^K$ are the sending rates of competing flows. Let X denote the underlying random variable of samples $\{x_i\}_{i=1}^K$. Observe that $F = \frac{E[X]^2}{E[X^2]}$. Rearranging, we have

$$F(X) = \frac{1}{(1 + CoV(X)^2)} \tag{3}$$

Summarizing, the main performance metrics we use in this section are CoV_{time}, which measures smoothness; F, which measures short-term fairness; and CoV_{lf}, which measures long-term fairness. However, the detailed behavior of a flow is too rich to be fully characterized by these metrics. Therefore, we will also show sending rate traces for some experiments to gain intuition. Furthermore, we will study fluctuations of the bottleneck queue size whenever we can gain more insights.

3.2 Analytical results

We present our analytical results on CoV_{time} for TCP, GAIMD, TFRC, and TEAR. The derivations of these results are put in the Appendix.

3.2.1 AIMD

For a low loss rate, CoV_{time} for AIMD (including GAIMD and TCP Reno as special cases) has been derived in the Appendix to be:

$$CoV_{\text{time}}^{AIMD} = \sqrt{\frac{1-\beta}{1+\beta}} \tag{4}$$

where β is the reduction ratio of congestion window size when there is a congestion indication.

Plugging $\beta = 1/2$ into Equation (4), we have

$$CoV_{time}^{TCP} = \sqrt{\frac{1}{3}} \approx 0.58 \tag{5}$$

Plugging $\beta = 7/8$ into Equation (4), we have

$$CoV_{time}^{GAIMD} = \sqrt{\frac{1}{15}} \approx 0.26 \tag{6}$$

When loss rate is high, both GAIMD and TCP Reno will be in a timeout state most of the time. CoV_{time} for timeout state has been derived in the Appendix to be:

$$CoV_{\text{time}}^{AIMD} = \sqrt{\frac{64(t-1)+32p+16p^2+8p^3+4p^4+2p^5+p^6}{64-32p-16p^2-8p^3-4p^4-2p^5-p^6}}$$

where p is packet loss rate, and t is the ratio of timeout value to round-trip time.

Plugging $p = 0.2$ and $t = 4$ into the expression above, for GAIMD and TCP Reno, we have

$$CoV_{time}^{AIMD} \approx 1.7 \tag{7}$$

3.2.2 TEAR

For a low loss rate, CoV_{time} of TEAR has been derived in Appendix to be:

$$CoV_{time}^{TEAR} \approx 0.21 \tag{8}$$

3.2.3 TFRC

For a low loss rate, CoV_{time} of TFRC has been derived in the Appendix to be:

$$CoV_{time}^{TFRC} \approx 0.22 \tag{9}$$

At a high loss rate (about 20%), we derived in the Appendix that CoV_{time} of TFRC will be between 0.8 and 2.4.

3.3 Experimental results

3.3.1 High multiplexing environments

We start our experimental evaluation with periodic loss. Figure 3 shows the sending rate traces when the loss rate is 5% periodic loss. For this figure, the x-axis is