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 re-

sponses 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.

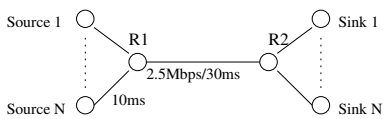

Figure 1: Network topology

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 measure-

ment 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.

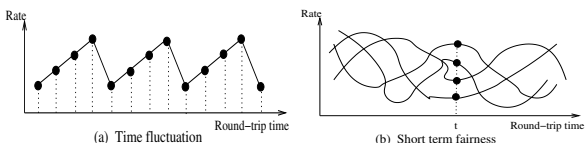

Figure 2: CoV_{time} and CoV_{sf}

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

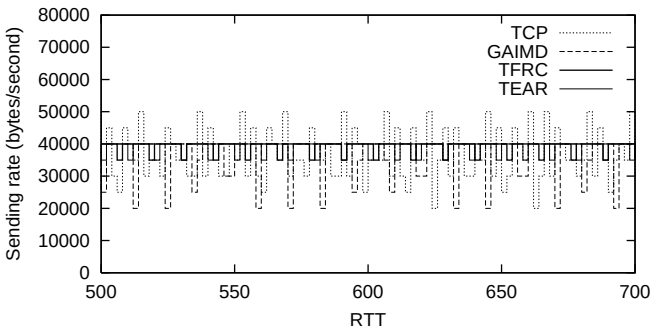

Figure 3: Sending rates (periodic loss, $p = 5\%$)

scaled by round-trip time, and the y-axis is the flow sending rate during a round-trip time. This simple experiment shows that the sending rates of TFRC and TEAR are smoother than those of TCP and GAIMD at low loss rate. Figure 4 shows the sending rate traces under Bernoulli loss of the same loss rate. Comparing Figure 3 to Figure 4, we observe that because of the randomness of Bernoulli loss, all four protocols exhibit much larger fluctuations even at this relatively low loss rate.

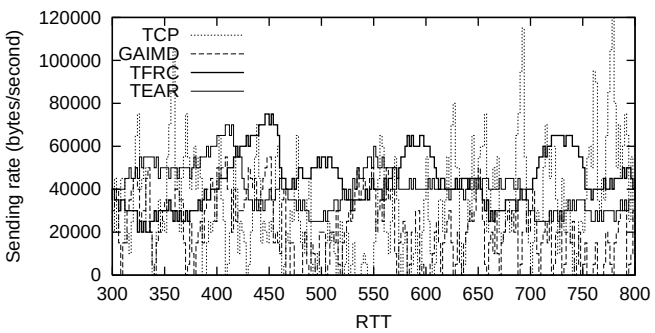

Figure 4: Sending rates (Bernoulli loss, $p = 5\%$)

The result for the high loss rate of 20% is even worse. We observe from Figure 5 that at 20% loss rate, the sending rate of TFRC drops to almost 0 and the average sending rate of TEAR is much higher than those of TCP and GAIMD. Therefore, at high loss rate, the behaviors of neither TFRC nor TEAR are desirable.

Figure 6 summarizes CoV_{time} from simulations for all four protocols when Bernoulli loss rates are varied from 0.5% to 20%. We can make the following observations:

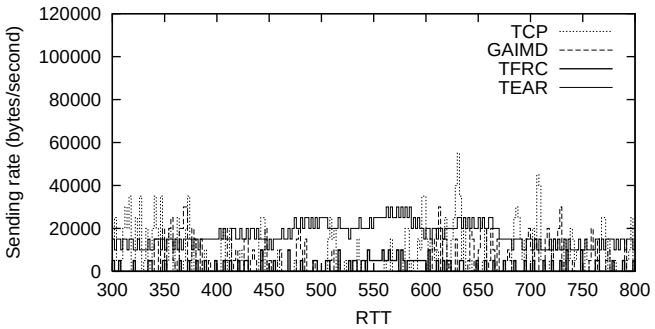

Figure 5: Sending rates (Bernoulli loss, $p = 20\%$)

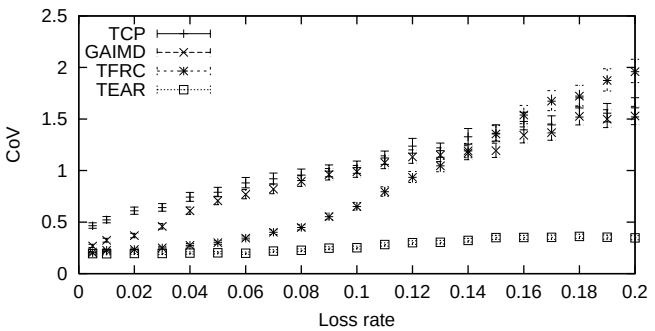

Figure 6: CoV_{time} of sending rates (Bernoulli loss)

- For all protocols, the overall trend is that CoV_{time} increases with increasing loss rates. In other words, the smoothness of the protocols reduces with increasing loss rate.

- At the low loss rate of 1%, TCP has the largest CoV_{time} of 0.51, which indicates that TCP smoothness at low loss rate is the worst. GAIMD CoV_{time} for this loss rate is the second largest with a value of 0.3. TFRC and TEAR have similar CoV_{time} for this loss rate with values of 0.23 and 0.22, respectively. We observe that these experimental values, 0.51, 0.3, 0.23 and 0.22, are close to the analytical predictions of 0.58 from Equation (5), 0.26 from Equation (6), 0.22 from Equation (9), and 0.21 from Equation (8), respectively.

- At 8% loss rate, GAIMD CoV_{time} increases to be the same as that of TCP. This is not surprising since at high loss rate, timeout dominates AIMD approaches. From Equation (7), we anticipate a CoV_{time} of 1.73 at 20% loss rate. We observe that this analytical prediction are very close to the measured value of 1.6.

- TFRC CoV_{time} stays low for up to 4% loss rate. Then it increases very fast and exceeds TCP and GAIMD at 15% loss rate. At 20% loss rate, the CoV_{time} of TFRC increases to 2, which is in the analytical prediction range of 0.8 to 2.4. Since TFRC has the highest CoV_{time} at high loss rate, it indicates that TFRC smoothness and fairness become the worst at high loss rate.

- TEAR keeps a low CoV_{time} of 0.2 to 0.4 across the range of measured loss rates. These experimental values agree with the analytical prediction of 0.21. These results show that TEAR can maintain a relatively stable smoothness and fairness performance over a wide range of loss rates.

Besides Bernoulli loss, for high multiplexing environment, we study protocol responses to the ON/OFF loss model. From Figure 7 we observe that under this loss model protocol smoothness is slightly worse than under Bernoulli loss. To understand the reason for the higher fluctuations, we investigate the bottleneck queue size. Figure 8 shows that ON/OFF traffic causes large fluctuations for bottleneck queue size. Therefore, we can expect large fluctuations in sending rates of responsive competing traffic.

3.3.2 Low multiplexing environments

Again starting from the simplest case, Figure 9 shows the sending rate traces when a single flow is sending across the bottleneck link. Since the bandwidth delay

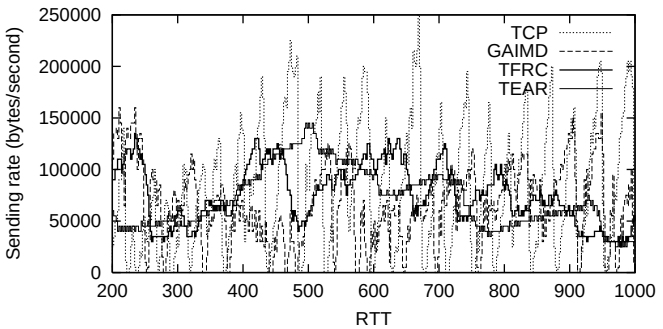

Figure 7: Sending rates (1 flow + 5 ON/OFF)

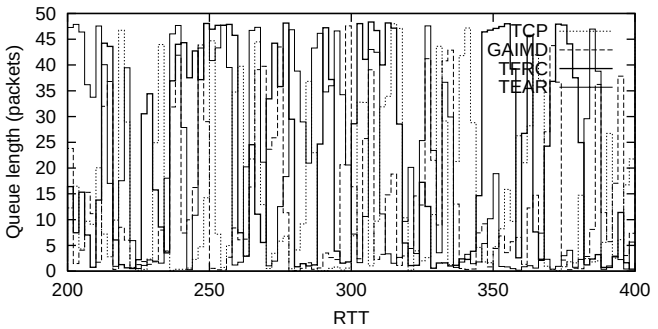

Figure 8: Queue length (1 flow + 5 ON/OFF)

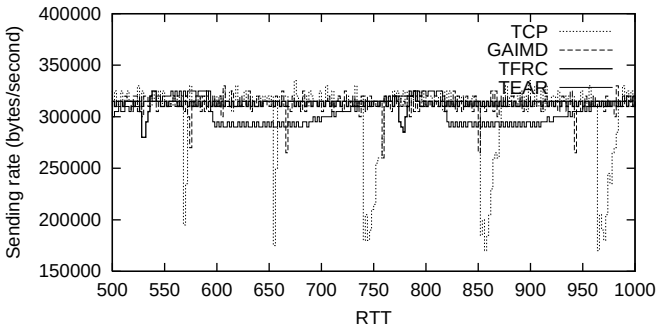

Figure 9: Sending rates (1 flow)

product is about 30 packets and the bottleneck link has a buffer size of 50 packets, when TCP/GAIMD congestion window size exceeds 80 packets, a packet will be dropped and their congestion window size will be cut to half. Since most of the time the congestion window size is larger than the bottleneck sending rate as shown in Figure 10, the actually sending rate is limited by ACK arrivals. Therefore, TCP/GAIMD sending rates are stable at the bottleneck link speed until they experience packet loss and the window size drops below 30 packets. Nevertheless, the sending rates of TCP and GAIMD exhibit more frequent and sometime larger fluctuations than those of TFRC and TEAR.

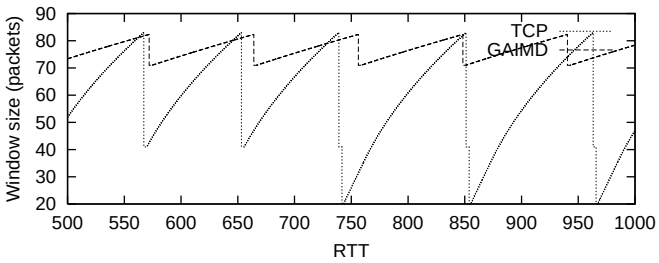

Figure 10: Congestion window sizes (1 flow)

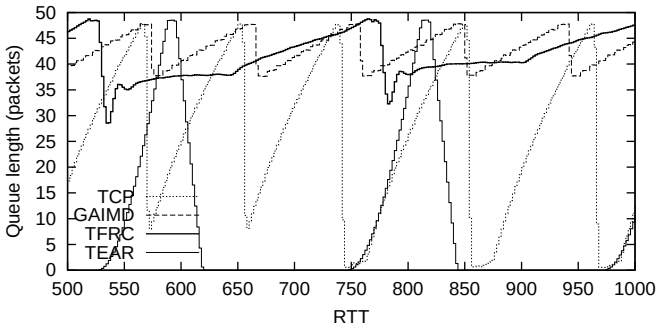

Figure 11: Queue length (1 flow)

As in the previous ON/OFF source case, to understand the reason for the sending rate fluctuations, we plot in Figure 11 the fluctuations of the bottleneck queue size. We make the following observations: 1) The bottleneck queue size under TCP exhibits large variations. TCP builds up the queue very quickly; when the queue is full and a packet is dropped, TCP backs off, and the queue drains to be empty very quickly. 2) GAIMD behavior is similar, but the fluctuations of queue size are much smaller than those of TCP. 3) TEAR queue behavior is similar to TCP — fast ramp

up to the peak and quickly drain out, but the cycle length of TEAR queue fluctuation is about two times of TCP. We notice that during half of the cycle TEAR queue size is almost empty. The reason, as we will see in Section 5, is that TEAR sending rate accelerating speed is very slow. 4) Similar to GAIMD, TFRC does not drain the queue to be empty, and its queue size fluctuations are also smaller. Similar to TEAR, we observe that the cycle of TFRC queue size fluctuation is much longer than that of TCP and GAIMD.

With the understanding of the simplest low multiplexing case, we next consider a loss environment generated by several competing congestion avoidance flows. Figure 12 shows the sending rate traces when 7 TCP flows are competing with

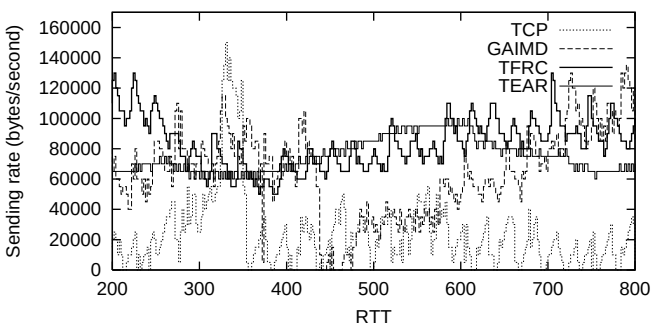

Figure 12: Sending rates (1 flow + 7 TCP)

the flow under study. We observe that TFRC and TEAR can maintain relatively smooth sending rates, while TCP and GAIMD sending rates fluctuate.

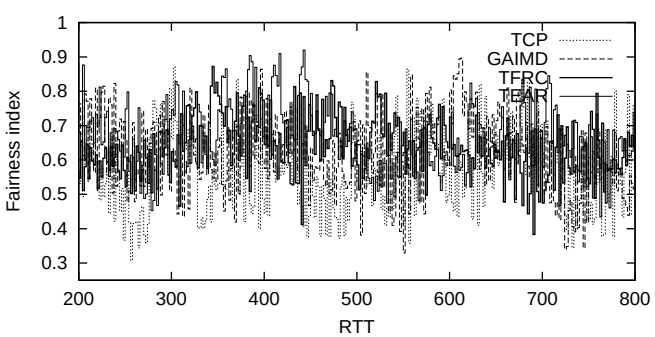

Figure 13: Fairness index (1 flow + 7 TCP)

Since in this case several TCP-friendly flows are competing against each other, we investigate the fluctuations of short-term fairness index. Figure 13 plots fair-

ness index at each round-trip time. As we stated at the beginning of this section, short-term fairness is correlated to sending rate smoothness. We know that TCP smoothness metric CoV_{time} is 0.58, and the 7 TCP flows dominate in this experiment, therefore, plugging $CoV = 0.58$ into Equation (3), we predict a short-term fairness index of $F = \frac{1}{1+0.58^2} \approx 0.7$. From Figure 13, we see that the prediction is close to the experimental result. We have also repeated this experiment with RED link, which has a different loss correlation, the result is very similar.

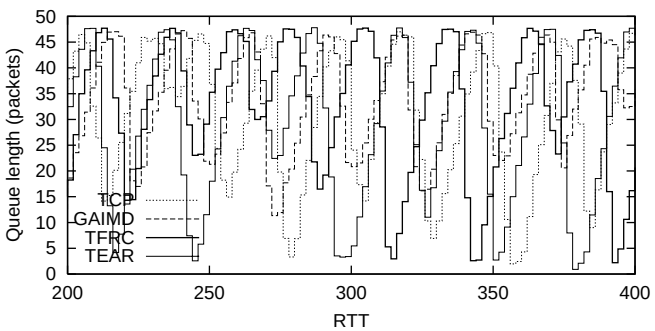

Figure 14: Queue length (1 flow + 7 TCP)

Figure 14 presents this experiment from another perspective: the fluctuations of the bottleneck queue size. Comparing Figure 14 to Figure 11, we observe that the queue behavior in this experiment is similar to the queue behavior of a single TCP flow, but the fluctuations have shorter cycles. Therefore, we get higher fluctuations in sending rates.

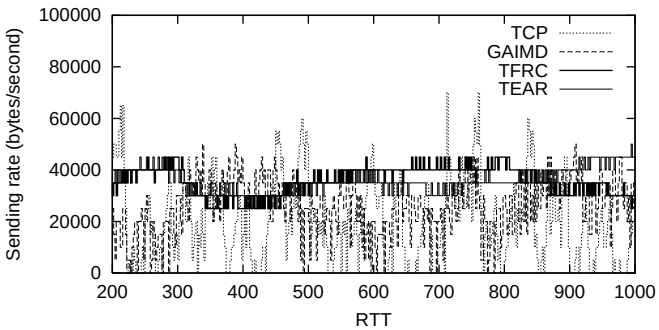

Figure 15: Sending rates (same protocol, 8 flows)

Protocol behaviors are different in a single protocol environment. Figure 15 shows the sending rate traces when the 7 competing flows belong to the same pro-

tocol as the flow under study. We observe that in this case the smoothness of TFRC and TEAR improves.

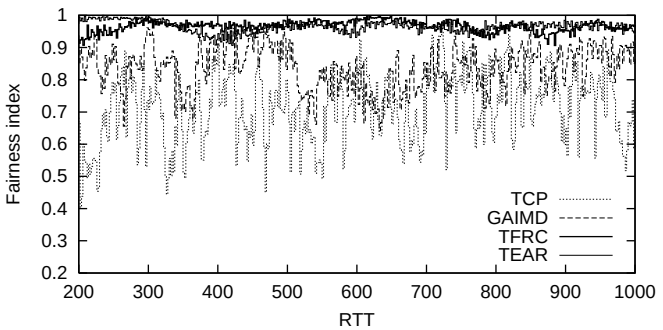

Figure 16: Fairness index (same protocol, 8 flows)

Figure 16 investigates the fluctuations of the short-term fairness for this experiment. It is particularly interesting to notice that the short-term fairness indices of TFRC and TEAR are close to 1, and the indices do not exhibit large variations. Therefore, it shows that TFRC and TEAR have better short-term fairness performance than TCP.

We also notice that the short-term fairness index values of all four protocols can be predicted as shown below. First consider TFRC and TEAR. We know that their CoV_{time} are about 0.21. Plugging this value into Equation (3), we have $F = \frac{1}{(1+0.21^2)} = 0.96$, which is close to 1. As for GAIMD, we know its CoV_{time} is 0.26. Plugging this value into Equation (3), we have $F = \frac{1}{(1+0.26^2)} = 0.93$, which is close to the experimental result in Figure 16. Figure 17 investigates the

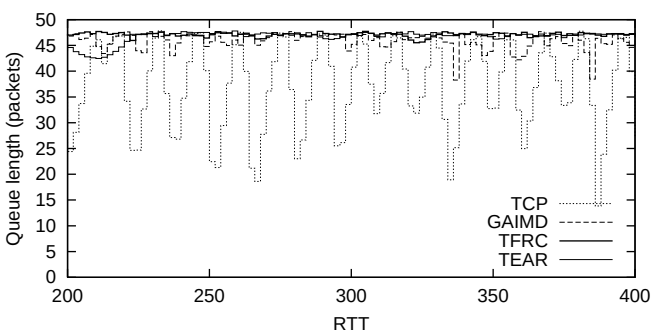

Figure 17: Queue length (same protocol, 8 flows)

fluctuations of the bottleneck queue size for this experiment. The queue behaviors

are very different in Figure 17 and Figure 14. In particular, when all flows belong to TFRC or TEAR, they can maintain a high and stable queue size. The queue size of GAIMD is also relatively high and stable. Therefore, for these three protocols, we can expect smaller delay jitter and smoother sending rates than those of TCP.

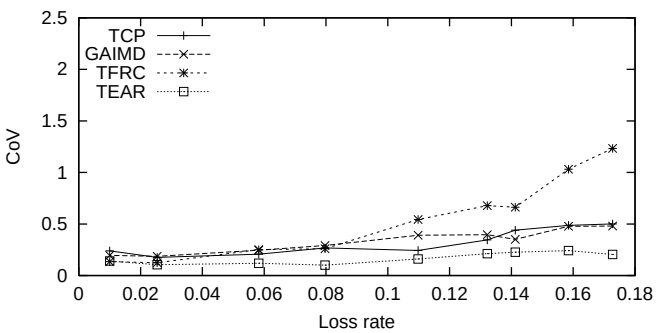

Figure 18: CoV_{lf} (same protocol, 32 flows, 15 second average)

In the last experiments in stationary scenarios, we evaluate the long-term fairness CoV_{lf}. Figure 18 shows the CoV_{lf} of 32 flows belonging to the same protocol when Bernoulli loss rates are varied from 0.5% to 17%. The measurement time interval for this experiment is 15 seconds. We use TCP/SACK in this experiment. Observe that this figure is very similar to Figure 6 in terms of trend and relative orders among the four protocols. This is not surprising given the relationship we observed from Equation (2). In another experiment, we have also evaluated CoV_{lf} when the measure time interval is 60 second, which is 4 times longer. We observe in Figure 19 that, at low loss rate, CoV_{lf} with 60 second measurement interval is about half of that with 15 second measurement interval. These results validate the relationship in Equation (2).

4 Responses to step increase of congestion

In this section, we evaluate protocol responsiveness. In the terminology of control theory, what we study are the protocol responses to a step increase function.

We first define the metrics. Our first metric to measure protocol responsiveness is the number of round-trip times D for a protocol to decrease its sending rate to half of the current rate under a persistent congestion, i.e., one loss indication for each round-trip time. This metric has also been used in [10, 9]. However, this metric represents a special case and does not measure the complete responding process. For the general case, we define protocol adaptation round-trip time, R.

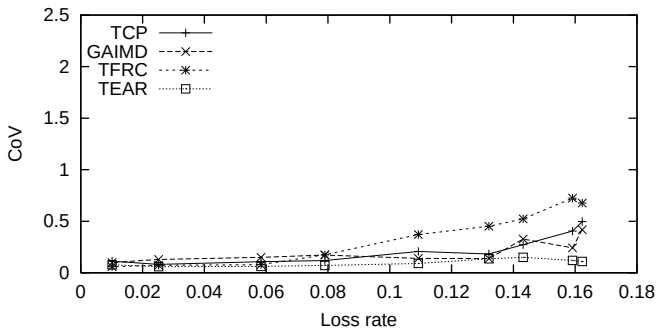

Figure 19: CoV_{lf} of sending rates (16 flows per protocol, 60 second average)

Let T_n denotes the sending rate of a protocol at the nth RTT after the disturbance. We define T_0 as the rate just before the disturbance, and T_∞ as the rate at the new steady state. Then T is the smallest n_0 such that for any $n \geq n_0$,

$$\left| \frac{T_n - T_0}{T_\infty - T_0} \right| \leq \alpha \tag{10}$$

where α is a small threhold, say 5%. We notice that the evaluation of R depends on the protocol sending rates before and after the increase of congestion.

4.1 Analytical results

Since TCP takes only one round-trip time to reduce its sending rate to half, its responsiveness $D_{\text{TCP}} = 1$. For GAIMD, $D_{\text{GAIMD}} = \log_{7/8} 0.5 \approx 5$. As for TFRC, from [10], we have $D_{\text{TFRC}} = 5$. For TEAR, denote W as the steady state window size just before persistent congestion, we know that TEAR old sending rate is $3W/4$ per RTT. After 5 consecutive congestion indications, the history of TEAR sending rates becomes $W/32$, $W/16$, $W/8$, $W/4$, $W/2$, $3W/4$, $3W/4$, and $3W/4$, therefore, TEAR sending rate after 5 loss indications will be reduced to half, and we have $D_{\text{TEAR}} = 5$.

To derive results for R, to avoid the complexity of randomness in states, we only consider the periodic loss model. Also, as we will see that the protocol adaption speed R depends on the loss rates before and after the step increase.

Suppose that the periodic loss rate changes from p_0 to p. t_n denotes the time when the nth packet loss occurs since the loss rate has changed, and T_n the sending rate at $t = t_n-$.

AIMD We model the window size in the interval (t_{i-1}, t_i) as $W(t) = \beta W(t_{i-1}-) + \frac{\alpha}{R}(t - t_{i-1})$, where R is round-trip time. Then the number of packets sent in the interval, which is the integration of the sending rate, should be $\frac{1}{p}$.

$$\int_{t_{i-1}}^{t_i} \frac{1}{R}\left(\beta W(t_{i-1}-) + \frac{\alpha}{R}(t - t_{i-1})\right) dt = \frac{1}{p}$$

From this equation we get t_i and $W(t_i-)$.

$$t_i = \frac{-\sqrt{p}R\beta W(t_{i-1}-) + R\sqrt{2\alpha + p\beta^2 W(t_{i-1}-)^2}}{\alpha\sqrt{p}}$$

$$W(t_i-) = \sqrt{\frac{2\alpha}{p} + \beta^2 W(t_{i-1}-)^2}$$

Then the sending rate is $\frac{W(t_i)-}{R}$.

$$T_i = \sqrt{\frac{2\alpha}{pR^2} + \beta^2 T_{i-1}^2}$$

TFRC Since the length of a loss interval is $\frac{1}{p_0}$ before the loss rate change, and $\frac{1}{p}$ after the change, the average loss interval is

$$\hat{s}_i = a_i \frac{1}{p_0} + (1 - a_i)\frac{1}{p}$$

where $a_0 = 1$, $a_1 = \frac{5}{6}$, $a_2 = \frac{2}{3}$, $a_3 = \frac{1}{2}$, $a_4 = \frac{1}{3}$, $a_5 = \frac{1}{5}$, $a_6 = \frac{1}{10}$, $a_7 = \frac{1}{30}$, and $a_8 = 0$. Then we get the following throughput of TFRC ignoring timeout.

$$T_i = \frac{1}{R}\sqrt{3\hat{s}_i}2 = \frac{1}{R}\sqrt{\frac{3}{2}\left(a_i\frac{1}{p_0} + (1 - a_i)\frac{1}{p}\right)}$$

When the loss rate becomes higher, TFRC keeps its rate constant until it sees a new packet loss event. Thus the time length of the i-th loss interval is $\frac{1}{pT_{i-1}}$, and we get $t_i = t_{i-1} + \frac{1}{pT_{i-1}}$.

TEAR We model the window size of TEAR in the interval (t_{i-1}, t_i) as follows:

$$W(t) = \frac{1}{2}W(t_{i-1}-) + \frac{2}{pW(t_{i-1}-)}$$

Then the average window size in the i-th interval $\overline{W}_i$ is

$$\overline{W}_i = \begin{cases} \frac{3}{4}W_0 & \text{if } i \leq 0 \\ \frac{W_i}{2} + \frac{W_{i-1}}{4} & \text{if } i > 0 \end{cases}$$

Then the throughput is the weighted average of these average window sizes.

$$T_i = \frac{1}{R}\sum_{k=1}^{8} w_k \overline{W}_{k+i-8}$$

where $w_1 = \frac{1}{6}$, $w_2 = \frac{1}{6}$, $w_3 = \frac{1}{6}$, $w_4 = \frac{1}{6}$, $w_5 = \frac{2}{15}$, $w_6 = \frac{1}{10}$, $w_7 = \frac{1}{15}$, and $w_8 = \frac{1}{30}$.

The equation for the length of the i-th interval is the same as that of TFRC.

$$t_i = t_{i-1} + \frac{1}{pT_{i-1}}$$

We plot the rate adaption $\left|\frac{T_n - T_0}{T_\infty - T_0}\right|$ in Figure 20 to show the rate decrease speed of each congestion control scheme using the results of the previous section, when the loss rate changes from 1% to 4%. TCP is the fastest one and the other three look similar except that GAIMD becomes slower after 10 RTT's. But in practice, GAIMD decreases faster than shown in this figure because the window size is rounded down when a packet loss occurs.

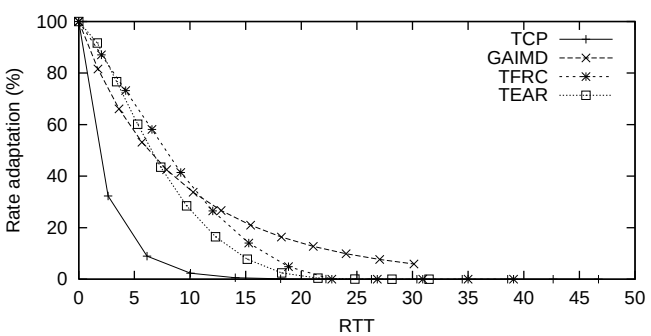

Figure 20: Rate decrease

4.2 Experimental results

We start our experiments with the periodic loss model. Figure 21 shows the protocol responses when the loss rate increases from 1% to 4% at round-trip time

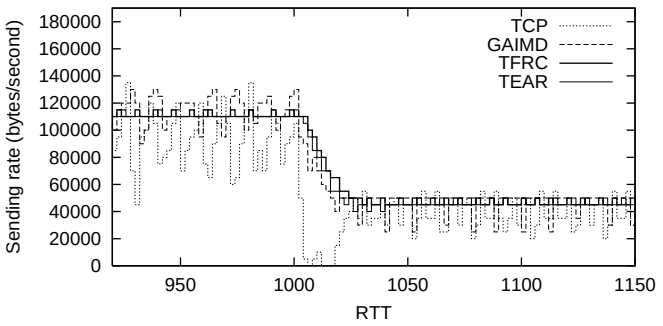

Figure 21: Sending rates (periodic loss, p=1% → p=4%)

1000. Clearly, TCP is the fastest of the four protocols to respond to loss rate increase; GAIMD, TFRC, and TEAR have similar responding speed and are obviously slower than TCP. However, we observe that TCP over-reacts and drops its sending rate to almost 0. Due to this behavior, TCP takes as long to reach the new stable state as the other three protocols.

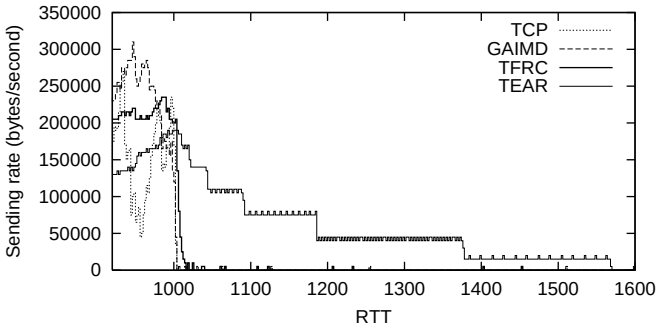

Figure 22: Sending rates (Bernoulli loss, p=0.5% → p=100%)

Next we consider protocol responses to Bernoulli loss rate change. Figure 22 shows the protocol responses when at round-trip time 1000 the loss rate increases from 0.5% to 100%, i.e., all data packets are dropped at the bottleneck link from the sender to the receiver. Since no data packet can go through, we expect that a responsive protocol would reduce its sending rate to almost 0. Among the four protocols, TCP responds at the highest speed and reduces its sending rate to almost 0. GAIMD is the second; it also reduces its rate to almost 0. TFRC is the third with a reasonable responding speed. The responding speed of TEAR is very slow. Furthermore, what we show here is the behavior of TEAR with timeout mechanism

added. In our initial experiments, where TEAR timeout mechanism is not implemented, TEAR does not reduce its sending rate at all. Another potential problem with TEAR is that TEAR puts all control functionality in the receiver. Therefore, if the feedback channel from the receiver to the sender is totally congested, the sender will keep on sending at the old rate. To rectify this potential problem, we suggest that the TEAR sender should reduce its sending rate if no feedback has arrived for a certain amount of time.

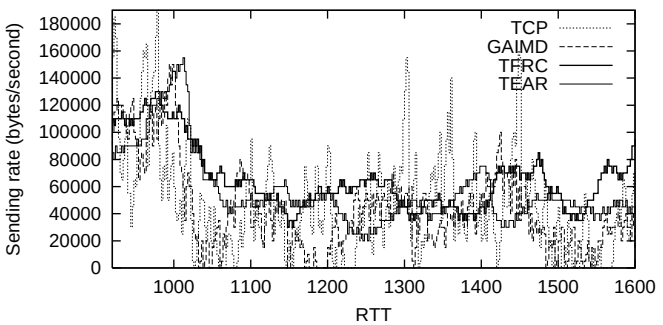

Figure 23: Sending rates (Bernoulli loss, p=1% $\rightarrow$ p=4%)

Protocol responses in the previous experiment are mainly determined by their timeout mechenisms (either at the sender or at the receiver); they show different responses when data packets can still go through the bottleneck link. Figure 23 tests the protocols when the loss rate increases from 1% to 4% at rount-trip time 1000. Since these rates are relatively low, we expect a TCP-friendly protocol to reduce its sending rate to half. From Figure 23 we observe that TCP is the fastest to respond, and GAIMD follows. However, both TCP and GAIMD drop below the new target state and recover slowly from the over-reaction. On the other hand, TFRC and TEAR have slower response speed than those of TCP and GAIMD, but neither TFRC nor TEAR has such over-reactions.

Instead of controlling the loss rate directly, next we test the protocols by introducing new flows into a steady network. We first consider the case when 8 new TCP flows start at time 1000 when one flow of the protocol under study and 7 competing TCP flows are in steady state. Figure 24 shows the sending rate traces for this experiment. Comparing Figure 24 to Figure 26, we do not see substantial differences.

In this experiment, we find the queue behavior is particularly interesting. Figure 25 shows the queue traces at the bottleneck link. At about round-trip time 1025, the queue sizes for all four protocols exhibit large dips. This suggests that all old flows back off by the new flows and that the bottleneck queue size decreases

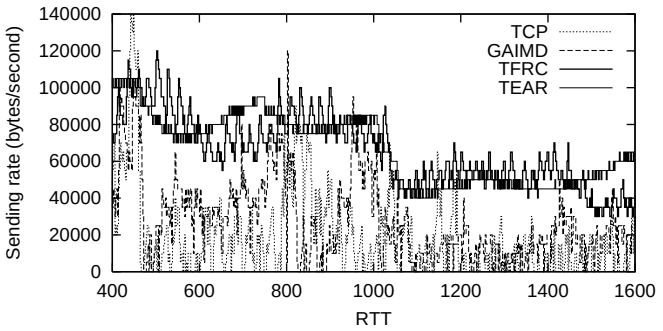

Figure 24: Sending rate (1 flow + 7 TCP → 1 flow + 15 TCP)

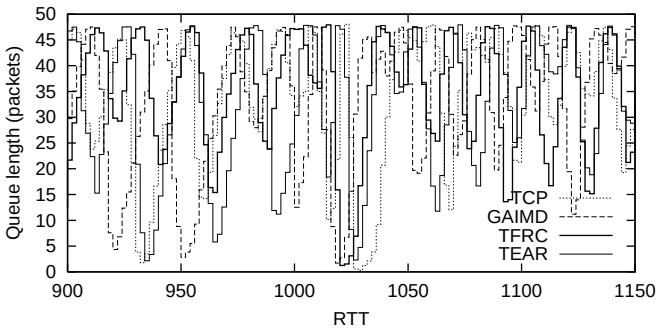

Figure 25: Queue length (1 flow + 7 TCP → 1 flow + 15 TCP)

consequently.

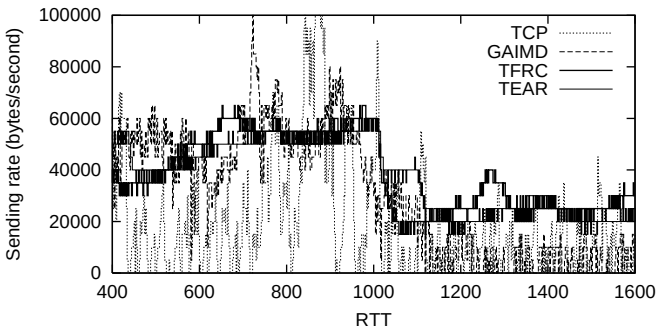

Figure 26: Sending rates (same protocol, 8 → 16 flows)

Next, we study protocol responses in a single protocol environment. Figure 26 shows the response of a flow as 8 new flows start at time 1000 when the flow and the 7 competing flows are in steady state. It is clear from this figure that TCP and GAIMD respond very fast, but both protocols overshoot to 0. TFRC and TEAR reduce their speeds slower. However, they do manage to reduce gradually to the new states.

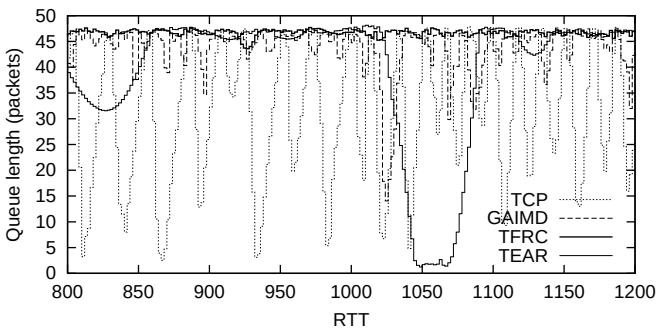

Figure 27: Queue length (same protocol, 8 → 16 flows)

Of particular interest when network traffic is increased is the behavior of the bottleneck queue. We expect the queue overload state will last longer for an irresponsive protocol. Figure 27 investigates the queue size before and after we increase the number of flows. In this figure, increasing the number of flows generates a large dip of queue size for GAIMD and TEAR. This dip is particularly obvious for TEAR, which means the new TEAR flows increase their sending rates very slowly.

As another measure of the protocol transient behavior when network congestion is increased, we consider the fairness indices before and after the disturbance. Figure 28 shows the fluctuations of fairness indices when the number of flows belonging to the same protocol is doubled. The fairness indices of TFRC and TEAR reduce from close to 1 to 0.5 right after the increase. Then their short-term fairness indices gradually increase to 1 and become stable at a value close to 1. The slow increase of TEAR's index indicates that the new TEAR flows are slow in increasing their sending rates, which is also observed in Figure 27. As we have already observed in Section 3, the fairness indices fluctuate for both TCP and GAIMD. We also notice that their fairness indices reduce after the number of flows is doubled. This decrease of fairness index is a result of loss rate increasing.

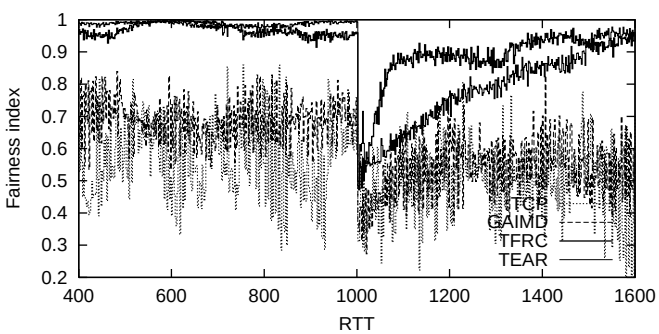

Figure 28: Fairness index (same protocol, 8 → 16 flows)

5 Responses to step increase of bandwidth

We evaluate protocol aggressiveness in this section. In the terminology of control theory, what we will study are the protocol responses to a step increase function of available bandwidth.

As in the previous section, we first define our metric. Our single-number-of-merit metric to measure protocol aggressiveness is a protocol's increasing speed I per RTT. Since protocol increasing speed depends on other factors such as feedback interval, what we derive is an upper bound. To evaluate the complete adaptation speed of a protocol, we again should use the protocol adaptation speed as defined in the previous section.

5.1 Analytical results

TCP increases its rate by 1 per RTT in congestion avoidance state. Thus its aggressiveness metric I_{TCP} equals to 1; likewise $I_{\mathrm{GAIMD}} = \alpha = 0.31$. As for TFRC, from [10], we have $I_{\mathrm{TFRC}} = 0.12$ without history discounting and $I_{\mathrm{TFRC}} = 0.22$ with history discounting. For TEAR, we know that, when there is no loss, the receiver increases the estimate of the sending rate for the most recent history from $\sum_{i=0}^{t} \frac{W+i}{t+1} = W + \frac{t}{2}$ at round-trip time t to $W + \frac{t+1}{2}$ at round-trip time $t+1$. Since the weight of the most recent epoch is $1/6$, the upper bound of I_{TEAR} is $1/12$.

Figure 29 shows the analytical responding speed using the derivations in Section 4.1.

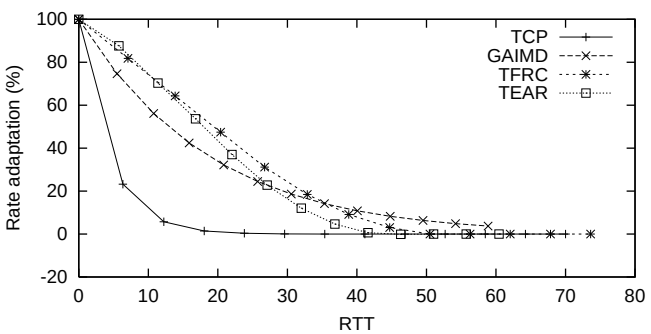

Figure 29: Rate increase

5.2 Experimental results

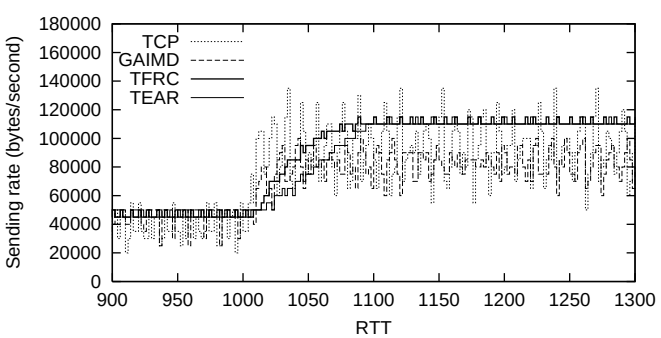

Figure 30: Sending rates (periodic loss, p=4% → p=1%)

We again start with the periodic loss model. Figure 30 shows the protocol responses when the loss rate decreases from 4% to 1%. It is obvious from this figure that TCP is the fastest to utilize new bandwidth. GAIMD and TFRC with history discounting have similar increasing speed but GAIMD is slightly faster than TFRC. TEAR is the slowest to increase its sending rate. The order of the protocol increasing speed for this experiment conforms to our analytical result.

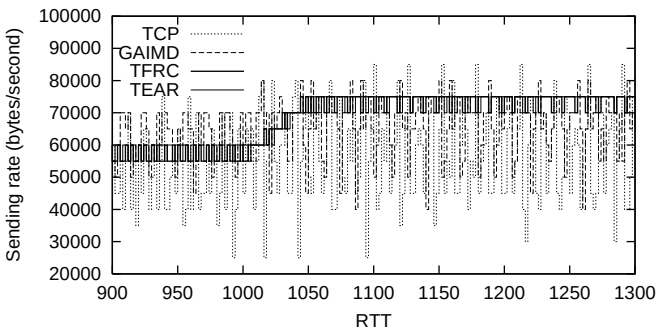

Figure 31: Sending rates (periodic loss, p=3% → p=2%)

Figure 31 shows another experiment where the loss rate decreases from 3% to 2%. In this experiment, TFRC history discounting is not activated, and we observe that TEAR and TFRC become similar.

Next, we consider the Bernoulli loss model. In Figure 32, we reduce the Bernoulli loss rate from 10% to 0% at $t = 1000$. Since no loss event occurs when $p = 0$, TFRC uses history discounting and increases its sending rate faster than usual. It is obvious that TCP is the fastest, and GAIMD is slightly faster than TFRC. TEAR is very slow compared to the other three protocols.

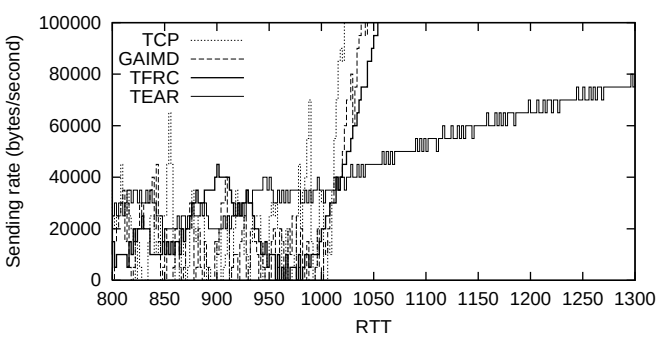

Figure 32: Sending rates (Bernoulli loss, p=10% → p=0%)

Instead of testing the extreme case when there is no loss at all, Figure 33 shows protocol responses when we reduce the Bernoulli loss rate from 4% to 1%. As is shown in the interval $(1000, 1050)$, TCP is the fastest in increasing its sending rate. GAIMD, TFRC, and TEAR follow it.

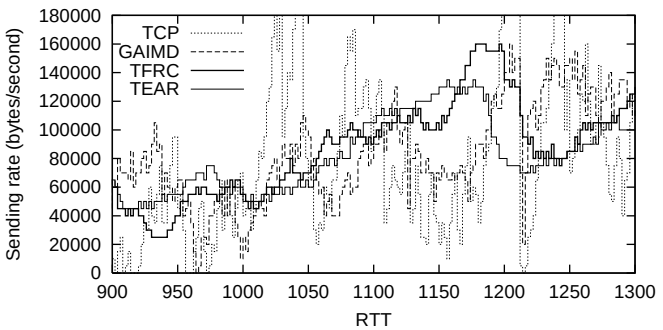

Figure 33: Sending rates (Bernoulli loss, p=4% $\rightarrow$ p=1%)

Instead of controlling the loss rate directly, next we test the protocols by stopping flows in a steady network to increase available bandwidth.

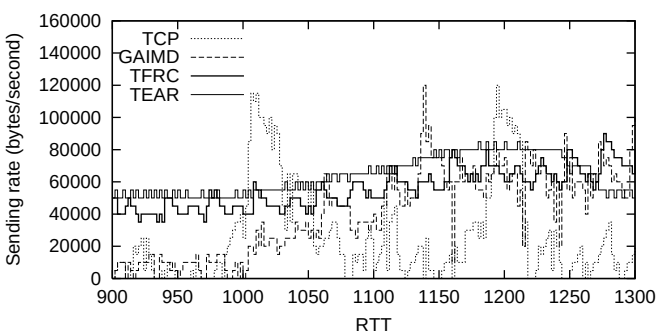

Figure 34: Sending rates (1 flow + 15 TCP $\rightarrow$ 1 flow + 7 TCP)

We first consider the case when 8 of the 15 TCP flows stop at time 1000 in Figure 34. Since we decrease the total number of flows from 16 to 8, the sending rates of remaining flows should be doubled. In this figure, TCP and GAIMD respond almost instantaneously, while TFRC and TEAR take about two hundred RTT's to utilize the newly available bandwidth.

Next we study protocol responses in a single protocol environment. In this experiment, 8 of the 16 flows stop at time 1000. Figure 35 shows the adaptation of fairness indices. While fairness indices of TFRC and TEAR do not change

when the number of flows decreases, those of TCP and GAIMD increase almost instantaneously. This conforms to the behaviors in Figure 28, where TCP and GAIMD fairness indices decrease when the number of flows is increased.

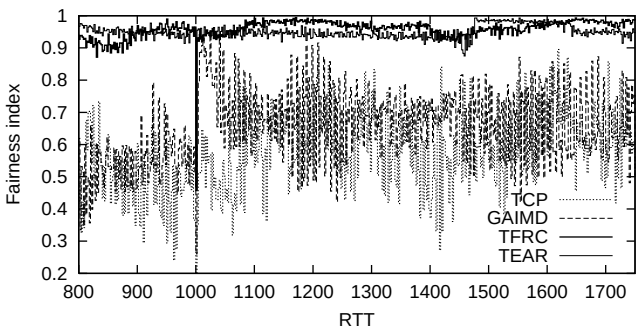

Figure 35: Fairness index (same protocol, 16 → 8 flows)

Another point to notice is how fast each protocol occupies the newly available bandwidth. Figure 36 shows queue size traces for this experiment. For all four protocols, queue sizes drop to zero when the 8 flows stop. TCP takes only 20 RTT's to fill the queue again. GAIMD and TFRC takes 50 and 60 RTT's, respectively. TEAR takes more than 400 RTT's because its sending rate increasing speed is very slow.

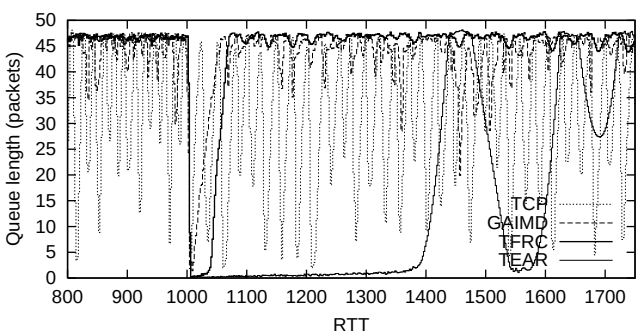

Figure 36: Queue length (same protocol, 16 → 8 flows)

6 Conclusion

We studied analytically and experimentally the transient behaviors of TCP and three TCP-friendly congestion control protocols. Table 1 summarizes our quan-

	TCP	GAIMD	TFRC	TEAR
Fairness	F = 0.7 at low loss	F = 0.9 at low loss	F $\approx$ 1 at low loss; rate drops to 0 at high loss	F $\approx$ 1 at low loss; rate too high at high loss
Smoothness (CoV_{time})	0.58 at 2% loss, 1.7 at 20% loss	0.26 at 2% loss, 1.7 at 20% loss	0.22 at 2% loss, 2 at 20% loss	0.2 at 2% loss, 0.4 at 20% loss
Responsiveness (D)	1 RTT	5 RTTs	$5-6$ RTTs	$5-6$ RTTs, slower if no feedback
Aggressiveness (I)	1.0/RTT	0.31/RTT	0.12/RTT wo/ history disc. 0.22/RTT w/ history disc.	0.08/RTT, slower with delayed feedback

Table 1: Summary of quantitative results

titative results. The first row shows fairness measured by the short-term fairness index F discussed in Section 3. From this metric, we infer that TFRC and TEAR have better fairness performance at low loss rate than TCP and GAIMD. However, 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 row summarizes protocol smoothness measured by CoV_{time}. From this metric, we observe that TFRC and TEAR have better smoothness performance at low loss rate. TEAR can maintain a stable smoothness performance up to 20% loss rate, while TFRC becomes the worst of the four protocols at 20% loss rate. The third row summarizes protocol responsiveness measured by D defined in Section 4. From this metric, we see that TCP is the most responsive among the four protocols. The other three have similar responsive speed. The last row summarizes protocol aggressiveness measured by I defined in Section 5. This metric shows that TCP is the fastest protocol in utilizing extra bandwidth. TFRC, with history discounting, is slightly slower than GAIMD. TEAR is the slowest to increase sending rate.

This study shows that our evaluation framework can be a valuable tool for studying protocol responses and detecting undesirable protocol behaviors. However, a thorough evaluation of all TCP-friendly protocols is a daunting task. Some interesting factors that we have not studied include different round-trip times, and SACK-based TCP and GAIMD instead of Reno-based. We defer these issues to a future study.

Acknowledgments

The authors would like to thank Xincheng Zhang for discussions and suggestions.

A AIMD mean and CoV derivations when loss rate is small

A.1 AIMD sending rate mean and variance derivation using stochastic difference equation and Palm inversion

Consider the window size W_n just after a packet loss. We can express the congestion window update rule as:

$$W_{n+1} = \beta W_n + \alpha X_n \tag{11}$$

Notice that we use α instead of $\frac{\alpha}{RTT}$ for ease of typing.

From [5, 12, 1], we know that the above stochastic difference equation has a stationary solution,

$$W_n^* = \alpha \sum_{k=0}^{\infty} \beta^k X_{n-1-k} \tag{12}$$

Moreover, even if W_0 starts from an arbitrary value, it will converge almost surely to the above stationary distribution, i.e.,

$$\lim_{n\to\infty} |W_n - W_n^*| = 0 \tag{13}$$

Now, we can evaluate the mean and variance of W_n. Follow the convention, denote $m_1 = E[X_i]$, $m_2 = E[X_i^2]$ and $R(k) = E[X_n X_{n+k}]$, then

$$E[W_n^*] = m_1 \frac{\alpha}{(1-\beta)} \tag{14}$$

As for variance, we have

$$E[(W_n^*)^2] = \frac{\alpha^2}{1-\beta^2}(R(0) + 2\sum_{k=1}^{\infty} \beta^k R(k)) \tag{15}$$

However, the above expectations are event average, we are more interested in time average. Through Palm inverse formula [2], we have

$$E[g(W(t))] = \frac{1}{m_1} E^0[\int_0^{X_0} g(W(t))dt] \tag{16}$$

First we consider $E[W(t)]$, and we have

$$
\begin{aligned}
E[W(t)] &= \frac{1}{m_1} E^0[\int_0^{X_0} (\beta W_0 + \alpha t) dt] && (17)\\
&= \frac{1}{m_1} E^0[\beta W_0 X_0 + \frac{\alpha}{2} X_0^2] && (18)\\
&= \frac{1}{m_1} E^0[\alpha\beta \sum_{k=0}^{\infty} \beta^k X_{-1-k} X_0] && (19)\\
&\quad + \frac{\alpha}{2m_1} E^0[X_0^2] && (20)\\
&= \frac{\alpha}{m_1}[\frac{1}{2} R(0) + \sum_{k=1}^{\infty} \beta^k R(k)] && (21)
\end{aligned}
$$

Next, we consider $E[W(t)^2]$, we have

$$
\begin{aligned}
E[W(t)^2] &= \frac{1}{m_1} E^0[\int_0^{X_0} (\beta W_0 + \alpha t)^2 dt] && (22)\\
&= \frac{E^0[\beta^2 W_0^2 X_0 + \alpha\beta W_0 X_0^2 + \frac{\alpha^2}{3} X_0^3]}{m_1} && (23)\\
&= \frac{1}{m_1} E^0[\alpha^2\beta^2 (\sum_{k=0}^{\infty} \beta^k X_{-1-k})^2 X_0 && (24)\\
&\quad + \alpha^2\beta (\sum_{k=0}^{\infty} \beta^k X_{-1-k}) X_0^2 && (25)\\
&\quad + \frac{\alpha^2}{3} X_0^3] && (26)\\
& && (27)
\end{aligned}
$$

Next, we consider two special cases.

A.1.1 The $\{X_i\}$ are I.I.D.

In this case, we assume the loss interval follow an exponential arrival with parameter λ.

$$
E[(W_n^*)] = \frac{\alpha m_1}{1-\beta} \tag{28}
$$

$$
E[(W_n^*)^2] = \frac{\alpha^2}{1-\beta^2} \left[m_2 + \frac{2\beta m_1^2}{1-\beta} \right] \tag{29}
$$

$$CoV[W_n^*] = \frac{1-\beta}{1+\beta} \tag{30}$$

$$\begin{aligned} E[W(t)] &= \frac{\alpha}{m_1}\left[\frac{m_2}{2} + \frac{\beta m_1^2}{1-\beta}\right] && (31) \\ &= \frac{\alpha}{\lambda}\frac{1}{1-\beta} && (32) \end{aligned}$$

and the average sending rate is:

$$\begin{aligned} T &= \frac{E[W(t)]}{RTT} && (33) \\ &= \frac{\alpha}{\lambda RTT}\frac{1}{1-\beta} && (34) \end{aligned}$$

Next, we consider $E[W(t)^2]$. From the above, we know that

$$\begin{aligned} E[W(t)^2] &= \alpha^2\beta^2(\frac{m_2}{1-\beta^2} + \frac{2\beta}{1-\beta^2}\frac{m_1^2}{1-\beta}) && (35) \\ &+\alpha^2\beta\frac{m_2}{1-\beta} && (36) \\ &+\frac{\alpha^2}{3}\frac{m_3}{m_1} && (37) \end{aligned}$$

Plug in the parameters, and we have

$$Var[W(t)] = \frac{\alpha^2}{(1-\beta^2)\lambda^2} \tag{38}$$

And

$$CoV[W(t)] = \sqrt{\frac{1-\beta}{1+\beta}} \tag{39}$$

B AIMD CoV in timeout states

In this section, we derive the CoV formula for AIMD when timeout loss indications dominate. From [27], we can see that this will happen when loss rate is above 20%.

Figure 37 shows a Markovian state transition diagram. Each state represents one back-off factor. We have also analyzed a model considering slowstart effect,

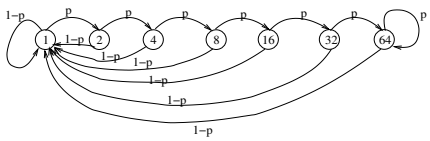

Figure 37: Timeout state transition diagram

but the result is similar. Finding the steady state distribution for this Markovian chain, we have

$$\begin{aligned} p_{2^i} &= (1-p)p^i \quad 0 \leq \mathrm{i} \leq 5 \\ p_{64} &= p^6 \end{aligned} \tag{40}$$

Denote t as the ratio of the timeout value and round-trip time. We know that in state 1, the sender will send one packet in t round-trip times, in state 2, the sender will send one packet in one of the $2t$ round-trip times, etc. Therefore, the probability of sending a packet in a round-trip time will be $\pi_1 = \sum_{k=0}^{k=6} p_{2^k} \frac{1}{2^k t}$.

Therefore, the CoV will be

$$CoV = \sqrt{\frac{64(t-1)+32p+16p^2+8p^3+4p^4+2p^5+p^6}{64-32p-16p^2-8p^3-4p^4-2p^5-p^6}} \tag{41}$$

C TEAR CoV

TEAR is based on TCP with a sliding window to smooth sending rate. Be specific, the per round-trip time sending rate of TEAR will be:

$$T_{TEAR} = \sum_{i=1}^{8} w_i \frac{\sum_{k=0}^{k=W_i-\frac{1}{2}W_{i-1}} \frac{1}{2}W_{i-1}+k}{W_i - \frac{1}{2}W_{i-1}+1} \tag{42}$$

Consider any term in the above summation:

$$Y_i = \frac{\sum_{k=0}^{k=W_i-\frac{1}{2}W_{i-1}} \frac{1}{2}W_{i-1}+k}{W_i - \frac{1}{2}W_{i-1}+1} \tag{43}$$

$$= \frac{1}{2}W_i + \frac{1}{4}W_{i-1} \tag{44}$$

Therefore,

$$Var_{time}[T_{TEAR}] = \sum_{i=1}^{8} w_i Y_i \tag{45}$$

$$= 0.07708 Var[W_i] \tag{46}$$

where $Var[W_i]$ is the variance of the window sizes just before a loss indication arrives.

Therefore,

$$CoV_{time}[T_{TEAR}] = 0.37CoV[W_i] \tag{47}$$

From Equation (30), we have

$$CoV_{time}[T_{TEAR}] = 0.37 * 0.58 = 0.21 \tag{48}$$

D TFRC CoV

At low loss rate, the sending rate of TFRC per round-trip time becomes

$$R(t) = \sqrt{\frac{3s}{2}} \tag{49}$$

where s is the weighted average of 8 loss intervals $\{s_i\}_{i=1}^{8}$,

$$s = \sum_{k=1}^{8} w_i s_i \tag{50}$$

First, without considering the smooth effect, we consider the CoV of the sending process

$$R'(t) = \sqrt{\frac{3s_i}{2}} \tag{51}$$

Denote p as the per packet loss probability, and $q = 1 - p$. Consider the Markovian model in Figure 38. The reason that the departure rate at state i is

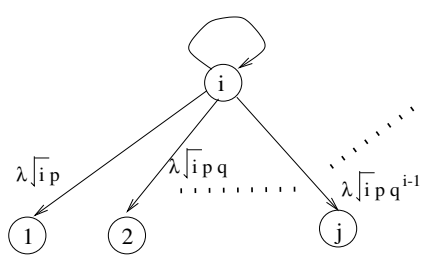

Figure 38: TFRC low loss rate state model

proportional to $\sqrt{i}$ is that we assume Bernoulli loss model, therefore, the higher the sending rate is, the faster the rate to leave the state.

Solve the local balance equation:

$$P_i \lambda \sqrt{i} p q^{i-1} = P_j \lambda \sqrt{j} q^{j-1} \tag{52}$$

We have that the probability in each state is given by:

$$P_i = c \frac{q^i}{\sqrt{i}} \tag{53}$$

where c is the normalizing factor so that the summation of P_i is 1.

With the distribution, we can find $E[R']$ as

$$E[R'] = \sum_{k=1}^{\infty} \lambda \sqrt{i} c \frac{q_i}{\sqrt{i}} \tag{54}$$

$$E[(R')^2] = \sum_{k=1}^{\infty} \lambda^2 i c \frac{q_i}{\sqrt{i}} \tag{55}$$

From the above two expressions, we can derive the expression for $CoV[R']$. Numerically calculating the value, we have $CoV[R']$ have values between 0.66 to 0.54 when the loss rate varies from 0.5% to 9%.

Approximate the smoothed version of TFRC CoV by

$$CoV[R] = (\sum_{k=1}^{8} w_i^2) CoV[R'] \tag{56}$$

We know that $CoV[R]$ varies between 0.25 to 0.21.

Under high loss rate, as a first approximation, the sending rate will be proportional to the inverse of $p^{\frac{3}{2}}$. Similar to the above approach, we can calculate the $CoV[R']$ at about 2.2 when loss rate is about 20%, and the smoothed version will have CoV of about 0.8.

As another approximation, we consider the case when the sending rate is proportional to the inverse of $p^{\frac{5}{2}}$ (which is the middle of 3/2 and 7/2), we can calculate the $CoV[R']$ at about 6.4 when loss rate is about 20%, and the smoothed version will have CoV of about 2.4.

References

[1] Eitan Altman, Kostia Avrachenkov, and Chadi Barakat. A stochastic model of TCP/IP with stationary random losses. In *Proceedings of ACM SIGCOMM 2000*, August 2000.

[2] Francois Baccelli and Pierre Bremaud. *Elements of queueing theory : Palm-Martingale calculus and stochastic recurrences*. Springer-Verlag, 1994.

[3] Jean-Chrysostome Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proceedings of ACM SIGCOMM '93*, September 1993.

[4] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. *Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309*, April 1998.

[5] A. Brandt. The stochastic equation $Y_{n+1} = A_n Y_n + B_n$ with stationary coefficients. *Adv. Appl. Prob.*, 18:211–220, 1986.

[6] Shanwei Cen, Calton Pu, and Jonathan Walpole. Flow and congestion control for Internet streaming applicaitons. In *Proceedings of Multimedia Computing and Networking 1998*, January 1998.

[7] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17, 1989.

[8] Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4), August 1999.

[9] Sally Floyd, Mark Handley, and Jitendra Padhye. A comparison of equation-based congestion control and AIMD-based congestion control. Work-in-progress, available at http://www.aciri.org/tfrc, 2000.

[10] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM 2000*, August 2000.

[11] Sally Floyd and Van Jacobson. On traffic phase effects in packet-switched gateways. *Internetworking: Research and Experience*, 3(3), September 1992.

[12] P. Glasserman and D. D. Yao. Stochastic vector difference equations with stationary coefficients. *J. Appl. Prob.*, 32:851–866, 1995.

[13] Stephen Jacobs and Alexandros Eleftheriadis. Providing video services over networks without quality of service guarantees. In *Proceedings of World Wide Web Consortium Workshop on Real-time Multimedia and the Web*, October 1996.

[14] Van Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM '88*, August 1988.

[15] Raj Jain, K. K. Ramakrishnan, and Dah-Ming Chiu. Congestion avoidance in computer networks with a connectionless network layer. Technical Report DEC–TR–506, DEC, August 1987.

[16] J. Mahdavi and S. Floyd. TCP-friendly unicast rate-based flow control. Note sent to the end2end-interest mailing list, 1997.

[17] Jitendra Padhye, Jim Kurose, Don Towsley, and Rajeev Koodli. A model based TCP-friendly rate control protocol. In *Proceedings of NOSSDAV '99*, June 1999.

[18] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols and self-similar network traffic. In *ICNP '96*, 1996.

[19] Reza Rejaie, Mark Handley, and Deborah Estrin. RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet. In *Proceedings of IEEE INFOCOM '99*, volume 3, March 1999.

[20] Injong Rhee, Volkan Ozdemir, and Yung Yi. TEAR: TCP emulation at receivers — flow control for multimedia streaming. Technical report, North Carolina State University, April 2000.

[21] Dorgham Sisalem and Henning Schulzrinne. The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme. In *Proceedings of NOSSDAV '98*, July 1998.

[22] Wai-Tian Tan and Avideh Zakhor. Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol. *IEEE Trans. on Multimedia*, 1, June 1999.

[23] Kevin Thompson, Gregory J. Miller, and Rick Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network*, 11(6), November 1997.

[24] Thierry Turletti, Sacha Fosse Parisis, and Jean-Chrysostome Bolot. Experiments with a layered transmission scheme over the Internet. Research Report No 3296, INRIA, November 1997.

[25] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level. In *Proceedings of ACM SIGCOMM '95*, 1995.

[26] Maya Yajnik, Sue Moon, Jim Kurose, and Don Towsley. Measurement and modelling of the temporal dependence in packet loss. In *Proceedings of IEEE INFOCOM '99*, March 1999.

[27] Yang Richard Yang and Simon S. Lam. General AIMD congestion control. In *Proceedings of ICNP 2000*, November 2000.

[28] Yin Zhang, Vern Paxson, and Scott Shenker. The stationarity of Internet path properties: Routing, loss, and throughput. Technical report, ACIRI, May 2000.