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 via simulation by studying protocol responses to three network environment changes. The first environment change is the inherent fluctuations in a stationary network environment. Under this scenario, 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 share bandwidth fairly, to avoid congestion collapse, and to keep network utilization high [8]; the robustness of the Internet is due in large part to the end-to-end congestion control mechanisms of TCP [13]. However, while TCP congestion control is appropriate for applications such as bulk data transfer, other applications such as streaming multimedia 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 last few years, many unicast congestion control protocols have been proposed and investigated [16, 24, 21, 6, 19, 22, 17, 25, 10, 28, 20]. Since the dominant Internet traffic is TCP-based [23], it is important that new congestion control protocols 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].

Evaluations of these protocols, however, have been focused mainly on protocol fairness in stationary environments. Two methods were proposed to establish the fairness of a protocol. The first is 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 [28, 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 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 studying the transient behaviors of several congestion control protocols under three network environment changes. Proposed congestion control protocols in the literature fall into two major categories: AIMD-based [21, 6, 19, 28, 20] and formula-based [16, 24, 22, 17, 10]. For our study, we select TCP [13] and GAIMD [28] as representatives of the first category. 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 evaluation, we choose $\beta = 7/8$ because it reduces a flow's sending rate less rapidly than TCP does. For $\beta = 7/8$, we choose $\alpha = 0.31$ so that the flow is TCP-friendly [28]. 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 protocols. In addition to these three protocols, we 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 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 fairness behaviors at high loss rate, i.e., TFRC sending rate dropping to almost zero and TEAR sending rate being too high compared with 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 its overshot state. We also found that TEAR does not reduce its sending rate under persistent congestion. From this finding, we suggest that the *self-clocking* mechanism of window-based protocols should be included in congestion control protocols to improve protocol responsivess. This shows that our evaluation framework can be a valuable tool for evaluating congestion control protocols 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 newly available bandwidth. Again TCP overshoots. TFRC with history discounting and GAIMD have similar aggressiveness. TEAR is the lest aggressive to utilize newly available bandwidth.

The balance of this paper is organized as follows. In Section 2 we discuss our evaluation methodology. In Section 3 we evaluate protocol responses 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 conclusion and future work are in Section 6.

2 Evaluation methodology

2.1 Loss models

Network loss process is a major factor in determining the performance of a congestion control protocol. In our simulations, 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 a 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 independent and identically distributed (i.i.d.) for all packets. We consider this model as one representative for high multiplexing environments. For example, in today's Internet, packets are dropped by routers without regard to which flows they belong to when buffers overflow. Though packet losses can be correlated, a number of studies [3, 27, 29] 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 for high multiplexing environments 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 [26]. Therefore, we choose ON/OFF UDP flows as our third type of background traffic. In our experiments, we set the mean ON time to be 1 second, and the mean OFF time to be 2 seconds. During ON time each source sends at 500Kbps. The shape parameter of the Pareto distribution is set to be 1.5. The number of ON/OFF sources in our experiments is 5.

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

2.2 Simulation configurations

Our network topology is the well-known single bottleneck ("dumbbell") as shown in Figure 1. In this topology, all access links have a delay of 10 ms, and they are sufficiently provisioned to ensure that packet drops due to congestion occur only at the bottleneck link from R1 ro R2. The bottleneck link is configured to have a bandwidth of 2.5 Mbps and a propagation delay of 30 ms. We repeat each simulation twice by configuring the bottleneck link as either a drop-tail or a RED link. For drop-tail link, we set a buffer size of 50 packets with packet size 1000 bytes. The parameters of RED link are scaled as in [10]. For most cases, the results for drop-tail and RED are similar. Therefore, the reported results are for drop-tail link unless we state otherwise.

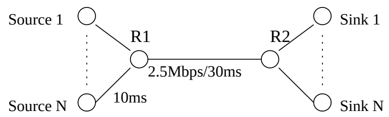

Figure 1: Network topology

We use GAIMD based on TCP/Reno. Most of our reported results on TCP are based on TCP/Reno unless we explicitly point out. TFRC is based on the code from *ns* June 12th, 2000 snapshot. In our initial set of TEAR experiments, we used the code from the authors' web site. However, we found that the timeout mechanism described in their paper [20] was not implemented. 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 those cases, we will point them out.

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

3 Responses to Stationary Fluctuations

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

3.1 Performance metrics

We use coefficient of variation (CoV) to measure protocol smoothness and fairness. First, we clarify three types of coefficient of variation.

3.1.1 Three types of coefficient of variation

The definition of CoV depends on measurement timescale: the longer the timescale, the smaller the CoV is. For our purpose, we measure smoothness and short-term fairness at a timescale of round-trip time; we define long-term fairness at a timescale of multiple round-trip times.

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. We observe that CoV_{time} measures the smoothness of a flow.

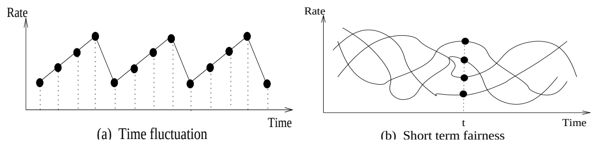

Figure 2: CoV_{time} and CoV_{sf}

2. *Short-term fairness* CoV_{sf}. Consider the solid dots in Figure 2b, which 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.

3. *Long-term fairness* CoV_{lf}. Instead of measuring the sending rates of competing flows during the same round-trip time, we can measure their sending rates during multiple round-trip times. Therefore, we

define long-term fairness CoV_{lf} as the coefficient of variation over the sending rates of competing flows in a longer time period.

With the definitions above, next we discuss their relationships. First consider the relationship between smoothness and fairness at a *given* timescale. Assuming competing flows are i.i.d. (the flows will then have the same mean sending rate if the measurement interval is infinity) and ergodic, we know that time distribution and population distribution are equal, that is,

$$CoV_{\text{time samples}} = CoV_{\text{population samples}} \tag{1}$$

Thus we observe that generating smoother traffic (measured by time samples) improves fairness (measured by population samples).

Next, consider the relationship between short-term and long-term CoV. It is intuitive that long-term CoV will be smaller than short-term CoV. Define an epoch as a time interval long enough such that the sending processes of a flow between epochs are independent and identically distributed. Let S_j denote the flow's average sending rate during the jth epoch, and define $R(n) = \sum_{j=1}^{n} S_j/n$ as its average sending rate in n epochs. Since we assume the random variables $\{S_j\}_{j=1}^{n}$ are i.i.d., by the central limit theorem, we know that the distribution of $R(n)$ can be approximated by normal distribution when n is large:

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

3.1.2 Metrics

In our evaluations, instead of using CoV_{sf} to measure short-term fairness, we follow [14] and use fairness index F, defined as $(\sum x_i)^2/(K \sum x_i^2)$, where $\{x_i\}_{i=1}^{K}$ are the sending rates of competing flows. Let X denote the underlying random variable of samples $\{x_i\}_{i=1}^{K}$. We observe that $F \approx E[X]^2/E[X^2]$. Rearranging, we have

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

In summary, the 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 cannot be fully characterized by these metrics. Moreover, our analytical results are derived for specific loss models. Therefore, to gain intuition, we will also show sending rate traces and the fluctuations of the bottleneck queue length for some simulations.

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

At low loss rate, under Poisson loss arrival, we derive CoV_{time} for AIMD (including GAIMD and TCP Reno as special cases) to be (see Appendix):

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

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

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

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

Plugging $\beta = 7/8$ for GAIMD 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 timeout states most of the time. Modeling timeout as a Markovian process, we derive CoV_{time} to be (see Appendix):

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

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

Plugging $p = 20\%$ 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

At low loss rate, assuming Poisson loss, we derive CoV_{time} for TEAR to be (see Appendix):

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

3.2.3 TFRC

At low loss rate, assuming Bernoulli loss, we derive CoV_{time} for TFRC to be:

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

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

3.3 Simulation results

3.3.1 High multiplexing environments

We start our simulation with periodic loss. Figure 3 shows flow sending rate traces when the loss rate is 5%. For this figure, the horizontal axis is

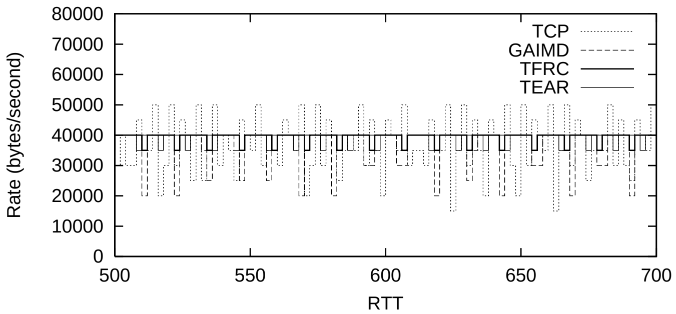

Figure 3: Sending rates (periodic loss, $p = 5\%$)

measured in the number of round-trip times (RTT), and the vertical 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 flow sending rate traces under Bernoulli loss model at the same loss rate. Comparing Figure 3 with 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.

The result for 20% Bernoulli loss is even worse. We observe from Figure 5 that at 20% loss, 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.

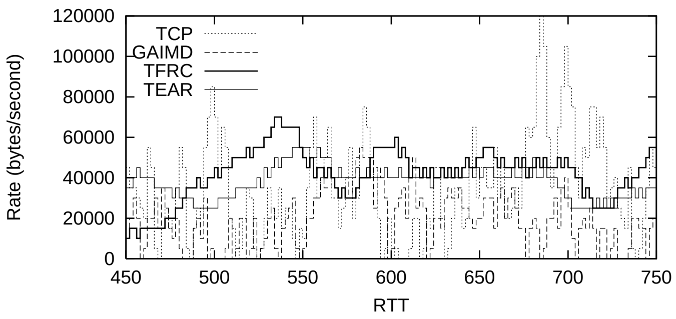

Figure 4: Sending rates (Bernoulli loss, $p = 5\%$)

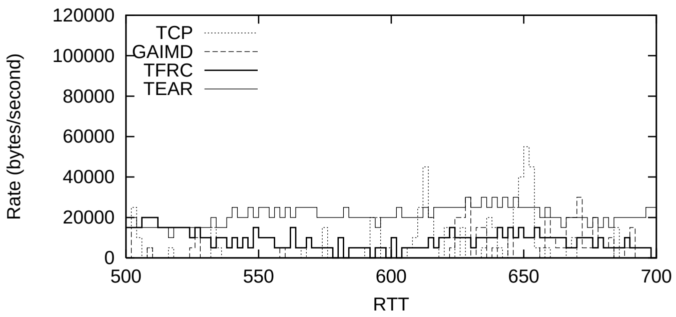

Figure 5: Sending rates (Bernoulli loss, $p = 20\%$)

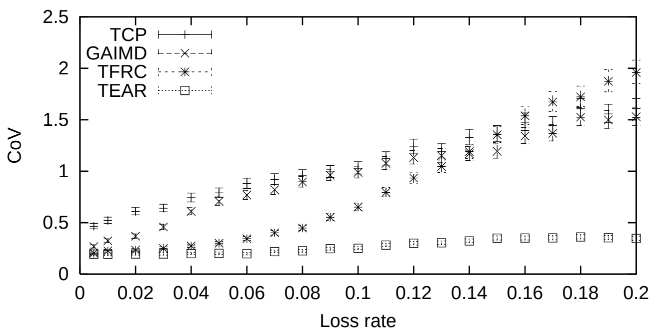

Figure 6: CoV_{time} of sending rates (Bernoulli loss)

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

- 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} at this loss rate is the second largest with a value of 0.3. TFRC and TEAR have similar CoV_{time} at 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 is 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, 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 model, for high multiplexing environment, we also 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 that under Bernoulli loss model. To understand the reason for the higher fluctuations, we investigate the bottleneck queue length. Figure 8 shows that ON/OFF traffic causes large fluctuations for bottleneck queue length.

Therefore, we can expect large fluctuations in sending rates of responsive competing traffic.

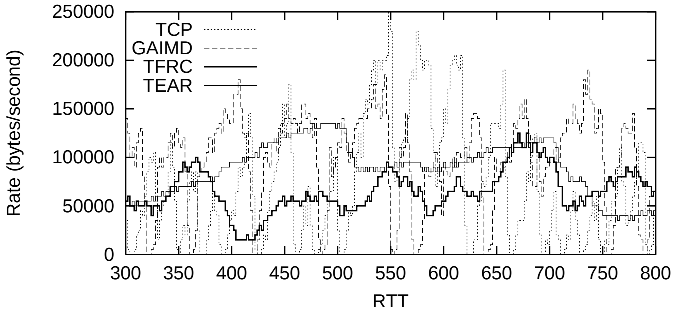

Figure 7: Sending rates (1 flow + 5 ON/OFF)

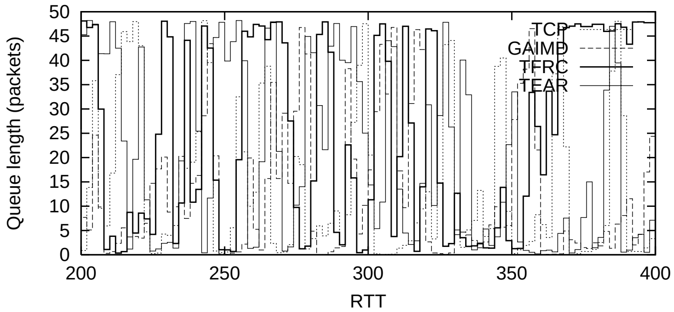

Figure 8: Queue length (1 flow + 5 ON/OFF)

3.3.2 Low multiplexing environments

We again start from the simplest environment. Figure 9 shows flow sending rate traces when a single flow is sending across the bottleneck link. From simulation configuration, we know that the bandwidth delay product is about 30 packets and the bottleneck link has a buffer size of 50 packets, therefore, there can be only 80 outstanding packets. When the congestion window size of a TCP/GAIMD flow exceeds 80 packets, a packet will be dropped and the flow's window size will be reduced. However, since most of the time the congestion window allows sending at a rate that is higher than the bottleneck link speed as shown in Figure 10, the achieved sending rate is limited by ACK arrivals. Therefore, TCP/GAIMD sending rates are

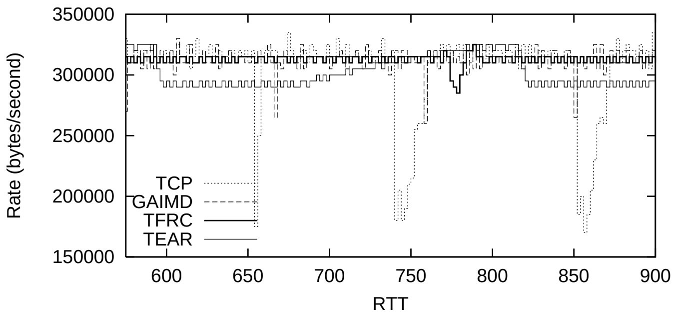

Figure 9: Sending rates (1 flow)

stable at the bottleneck link speed until they experience packet loss and the window size drops below 30 packets.

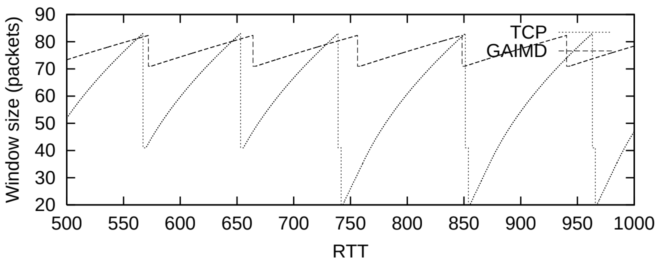

Figure 10: Congestion window sizes (1 flow)

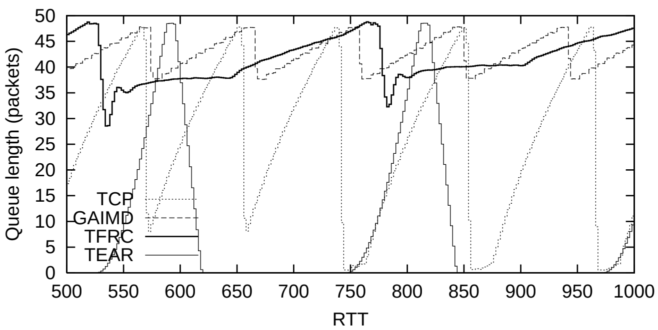

Figure 11: Queue length (1 flow)

As in the previous ON/OFF case, to understand the reason for sending rate fluctuations, we plot in Figure 11 the fluctuations of the bottleneck queue length. We make the following observations: 1) The queue length 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 length 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 of TEAR queue fluctuation is about two times of TCP. We notice that during half of the cycle TEAR queue is almost empty. The reason, as we will see in Section 5, is that TEAR is very slow in accelerating its sending rate. 4) Similar to GAIMD, TFRC does not drain the queue to be empty, and its queue length fluctuations are smaller. Similar to TEAR, the cycle of TFRC queue length fluctuation is much longer than those of TCP and GAIMD.

Following the simplest low multiplexing environment, we next consider an environment with several competing congestion avoidance flows. Figure 12

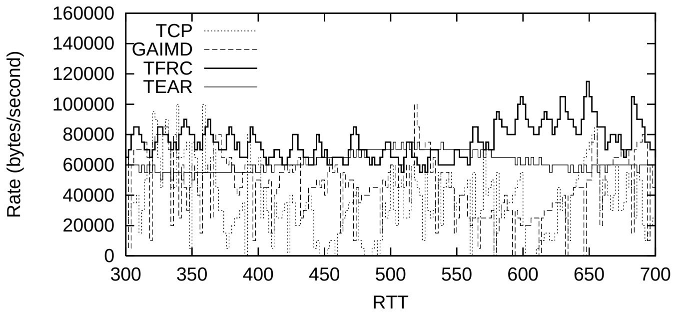

Figure 12: Sending rates (1 flow + 7 TCP)

shows flow sending rate traces when 7 TCP flows are competing with the flow under study. We observe that TFRC and TEAR can maintain relatively smooth sending rates, while the sending rates of TCP and GAIMD fluctuate.

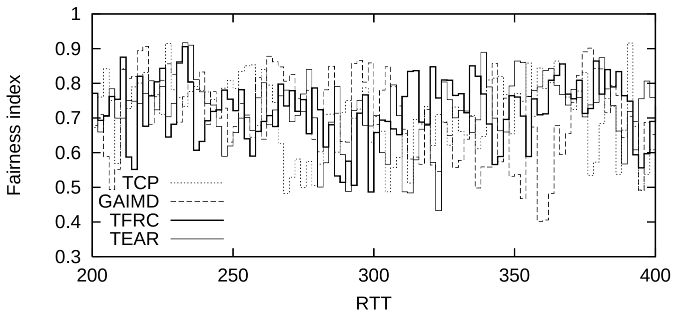

Figure 13: Fairness index (1 flow + 7 TCP)

Since in this environment 8 TCP-friendly flows are competing against each other, we investigate the fluctuations of short-term fairness index. Figure 13 plots fairness index at each round-trip time. As we stated in Section 3.1.1, short-term fairness is correlated to sending rate smoothness. We know that TCP smoothness metric $CoV^{\text{TCP}}_{\text{time}}$ 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 $F = 1/(1 + 0.58^2) = 0.7$. From Figure 13, we see that the simulation results fluctuate around the analytical value. We have also repeated this simulation with RED link, which has a different loss model; the result is similar.

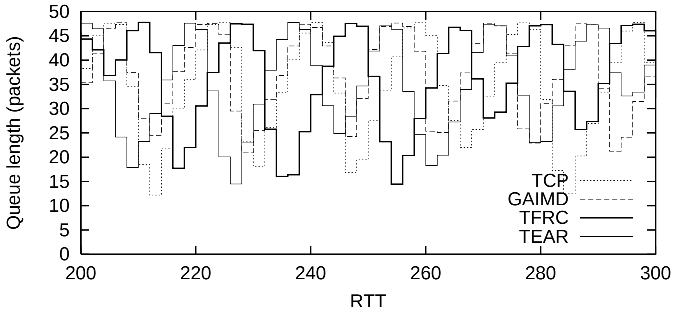

Figure 14: Queue length (1 flow + 7 TCP)

Figure 14 presents this experiment from another perspective: the fluctuations of the bottleneck queue length. Comparing Figure 14 with 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.

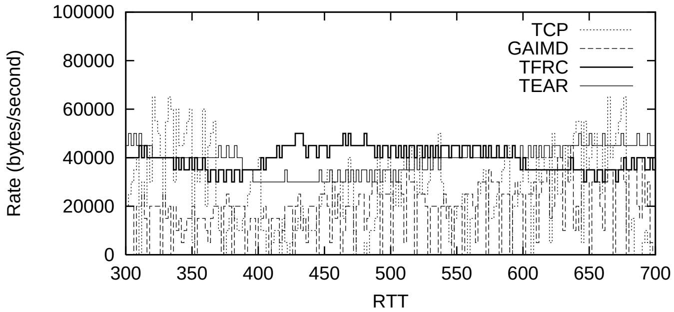

Figure 15: Sending rates (same protocol, 8 flows)

Protocol behaviors are different in an environment consisting of flows that belong to the same protocol. Figure 15 shows flow sending rate traces when the 7 competing flows belong to the same protocol as the flow under study. Comparing with Figure 12, we observe that in single protocol environment the smoothness of GAIMD, TFRC, and TEAR improves.

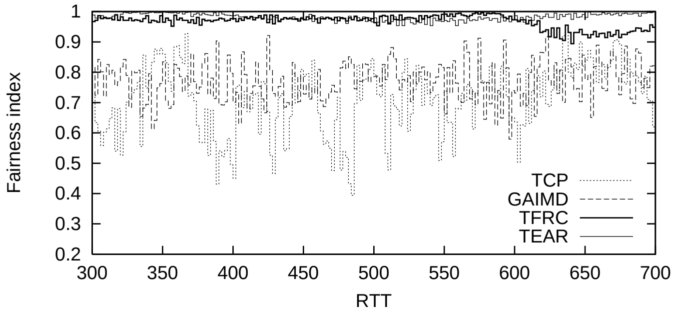

Figure 16: Fairness index (same protocol, 8 flows)

Figure 16 investigates the fluctuations of short-term fairness. 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 that of TCP.

We next compare our analytical results of short-term fairness index with those from simulations. First consider TFRC and TEAR. We know that their CoV_{time} are about 0.21. Plugging this value into Equation (3), we have $F = 1/(1 + 0.21^2) = 0.96$, which is close to 1. As for GAIMD, we know its CoV_{time} is 0.26. Plugging this value into Equation (3), we have $F = 1/(1 + 0.26^2) = 0.93$, which is slightly higher than the experimental result in Figure 16.

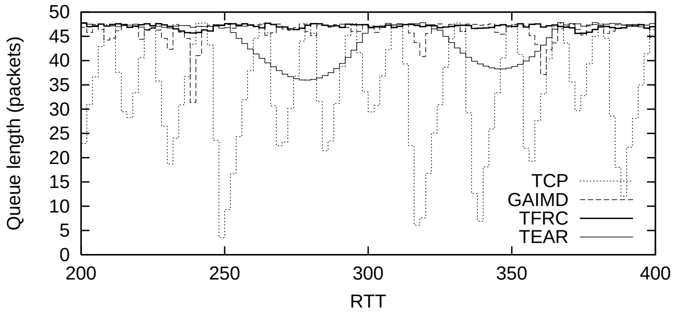

Figure 17: Queue length (same protocol, 8 flows)

Figure 17 investigates the fluctuations of the bottleneck queue length for this experiment. The queue behaviors are different between Figure 17 and

Figure 14. In particular, when all flows belong to TFRC or TEAR, they can maintain a high and stable queue length. The queue length 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.

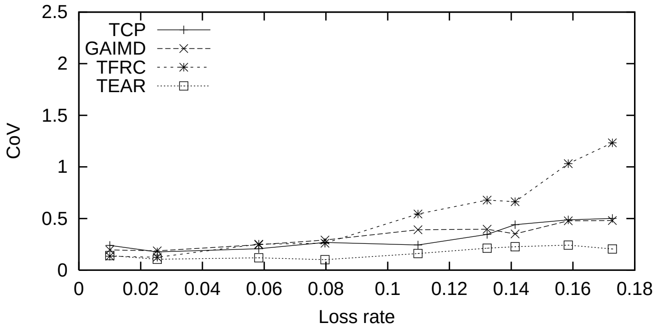

Figure 18: CoV_{lf} (same protocol, 32 flows, 15 second average)

In the last experiments in stationary environments, we evaluate the long-term fairness CoV_{lf}. Figure 18 shows the simulation results for CoV_{lf} of 32 flows belonging to the same protocol when Bernoulli loss rates are varied from 0.5% to 17%. In this simulation, TCP is based on TCP/SACK, and the measurement interval is 15 seconds. We 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 measurement interval is 60 seconds, 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).

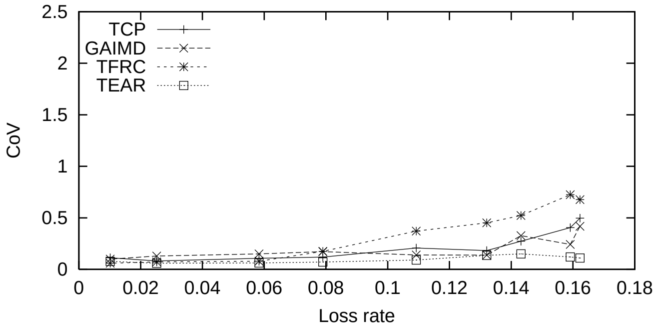

Figure 19: CoV_{lf} of sending rates (16 flows per protocol, 60 second average)

4 Responses to step increase of congestion

In this section, we evaluate protocol responsiveness. In the terminology of control theory, what we study are 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 under a persistent congestion, i.e., one loss indication for each round-trip time. This metric has also been used in [10, 9]. We notice that this metric does not measure the complete responding process. We have also defined protocol adaptation round-trip time, R. 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 $n \geq n_0$,

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

where α is a small threshold, 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 sending rate is $3W/4$ per RTT before persistent congestion, and that all of the 8 entries in its history window are $3W/4$. After 5 consecutive congestion indications, the 8 entries in its history window become $\{W/32, W/16, W/8, W/4, W/2, 3W/4, 3W/4, 3W/4\}$. 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

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

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.

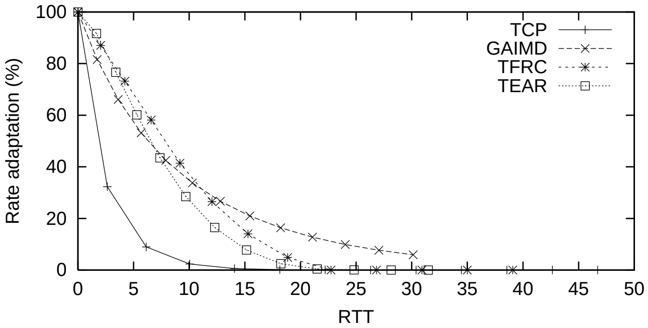

Figure 20: Rate decrease

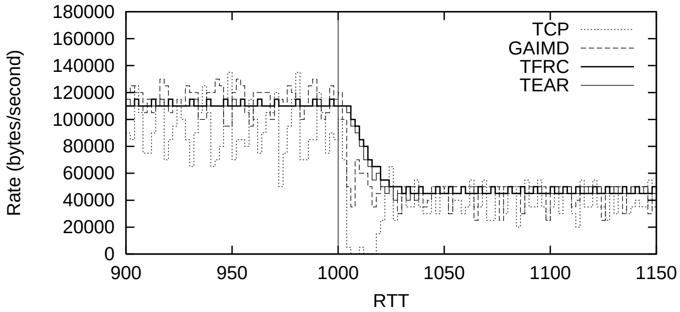

Figure 21: Sending rates (periodic loss, p=1% → p=4%)

4.2 Simulation results

We start our simulation with periodic loss model. Figure 21 shows protocol responses when loss rate is increased from 1% to 4% at round-trip time 1000, which is indicated as a vertical line in the figure. Clearly, TCP is the fastest of the four protocols to respond to loss rate increase; GAIMD follows; TFRC, and TEAR have similar responding speed and are obviously slower than TCP and GAIMD. 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 its new stable state as the other three protocols.*

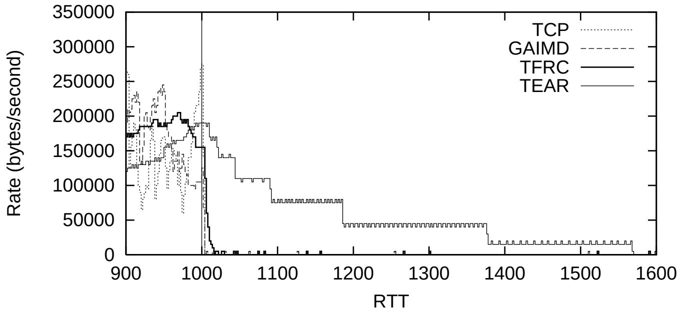

Figure 22: Sending rates (Bernoulli loss, p=0.5% → p=100%)

Next we consider protocol responses to Bernoulli loss rate change. Figure 22 shows protocol responses when at round-trip time 1000 Bernoulli loss rate is increased 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. The behaviors of TCP and GAIMD are expected because they are window-based protocols and the *self-clocking* mechanism provided by acknowledgement prevents the protocols to send any further packets until timeout. 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. To rectify similar problems and improve the responsiveness of rate-based congestion control protocols, we suggest that the self-clocking mechanisms of window-based protocols should be included.

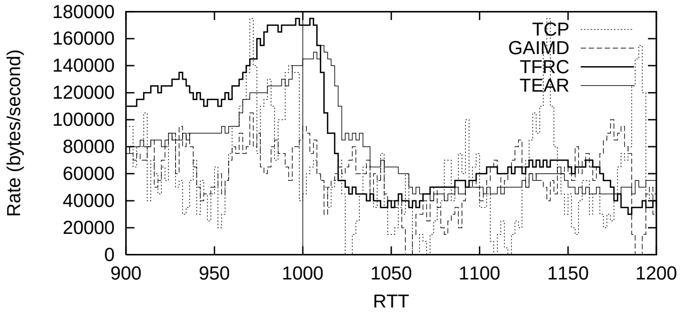

Figure 23: Sending rates (Bernoulli loss, p=1% → p=4%)

Protocol responses in the previous experiment are mainly determined by their timeout and self-clocking mechanisms (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 Bernoulli loss rate is increased from 1% to 4% at round-trip time 1000. Since these loss 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, TCP drops below the new target state and recovers slowly from its over-reaction. On the other hand, GAIMD, TFRC, and TEAR have slower response speed than that of TCP, but none of them has over-reaction.

Instead of controlling loss rate directly, next we test the protocols by introducing new flows into a steady environment. 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.

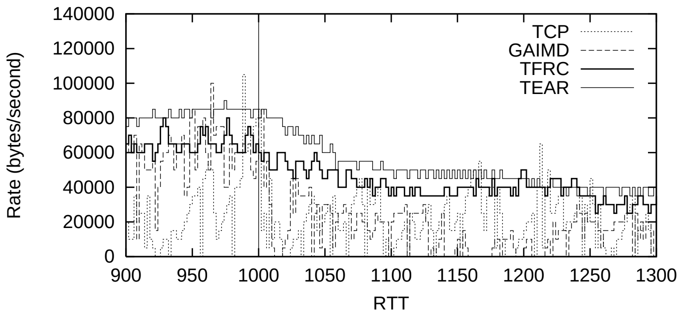

Figure 24: Sending rate (1 flow + 7 TCP → 1 flow + 15 TCP)

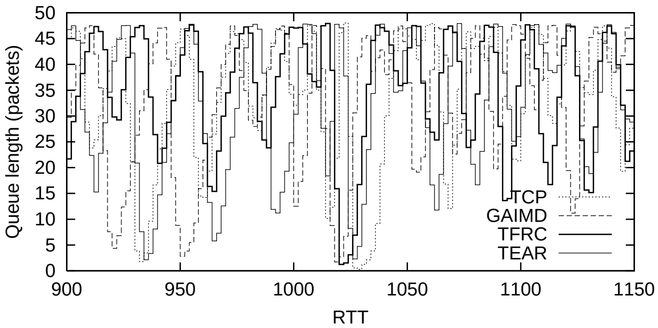

Figure 25: Queue length (1 flow + 7 TCP → 1 flow + 15 TCP)

In this experiment, we find the queue behavior is particularly interesting. Figure 25 shows queue length traces at the bottleneck link. At about round-trip time 1025, the queue lengths for all four protocols exhibit large dips. This suggests that the old flows back off due to the introduction of new flows. Thus, the bottleneck queue length decreases.

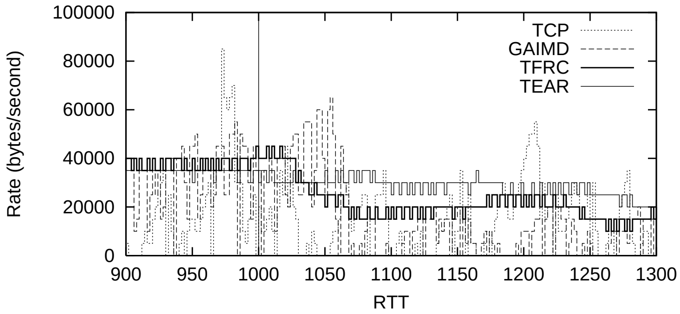

Figure 26: Sending rates (same protocol, 8 → 16 flows)

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.

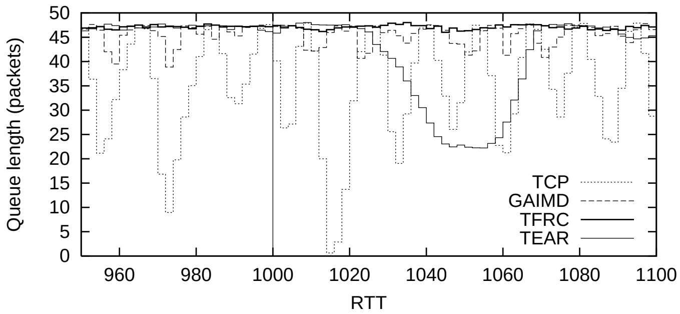

Figure 27: Queue length (same protocol, 8 → 16 flows)

We next study the behavior of the bottleneck queue. We expect that the queue will be in overload state for a longer period of time for a less responsive protocol. Figure 27 investigates queue lengths before and after we increase the number of flows. In this figure, increasing the number of flows generates a large dip of queue length for TEAR around round-trip time 1050. This dip indicates that the responses of TEAR flows are much longer delayed.

As another measure of protocol transient behaviors when network congestion is increased, we consider 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. Afterwards, 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 increased loss rate.

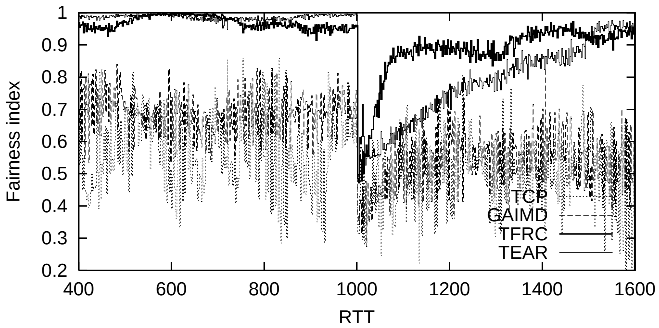

Figure 28: Fairness index (same protocol, 8 → 16 flows)

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 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. However, we do observe that the metric I can be used to optimize application performance [15]. 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 its estimation of the sending rate 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.

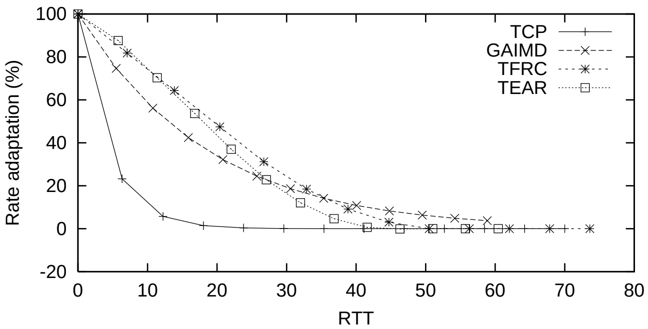

Figure 29: Rate increase

5.2 Simulation results

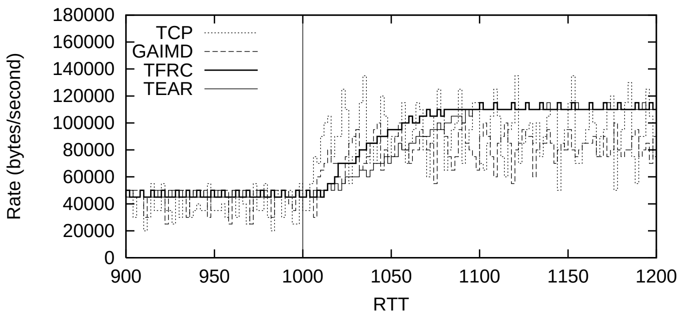

Figure 30: Sending rates (periodic loss, p=4% → p=1%)

We again start with periodic loss. Figure 30 shows protocol responses when loss rate is decreased from 4% to 1% at round-trip time 1000. 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. TEAR is the slowest to increase its sending rate. For this experiment, we observe that the relative order of the protocols to increase their sending rates conforms to our analytical result.

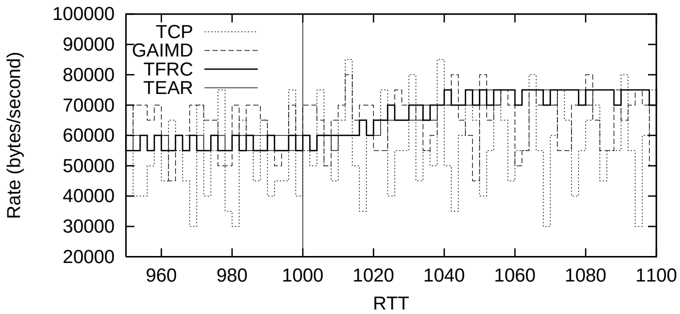

Figure 31: Sending rates (periodic loss, p=3% → p=2%)

Figure 31 shows another experiment where periodic loss rate is decreased from 3% to 2%. In this experiment, the history discounting mechanism of TFRC is not activated, and we observe that TEAR and TFRC become similar. Comparing the aggressiveness of TFRC with and without history discounting, we suggest that such mechanisms need to be considered by other protocols, such as TEAR.

Next, we consider Bernoulli loss model. In Figure 32, we reduce 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. We observe that TCP is the fastest, and GAIMD is faster than TFRC. TEAR is very slow compared with the other three protocols.

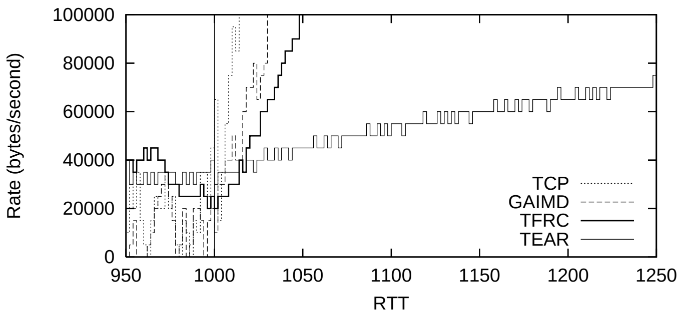

Figure 32: Sending rates (Bernoulli loss, p=10% → p=0%)

Instead of testing the extreme case when there is no loss at all, Figure 33 shows protocol responses when we reduce 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.

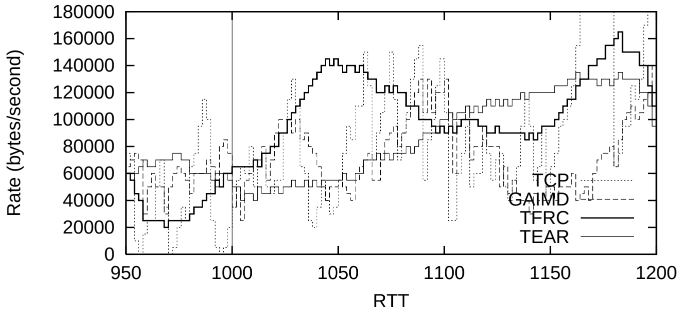

Figure 33: Sending rates (Bernoulli loss, p=4% → p=1%)

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

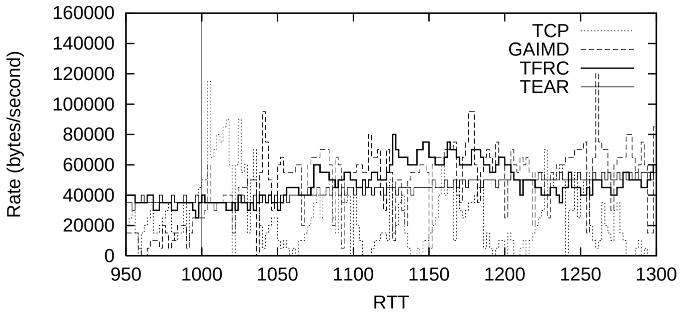

Figure 34: Sending rates (1 flow + 15 TCP → 1 flow + 7 TCP)

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 much longer to utilize the newly available bandwidth. From a control theory perspective, it appears TEAR is over-damped.

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.

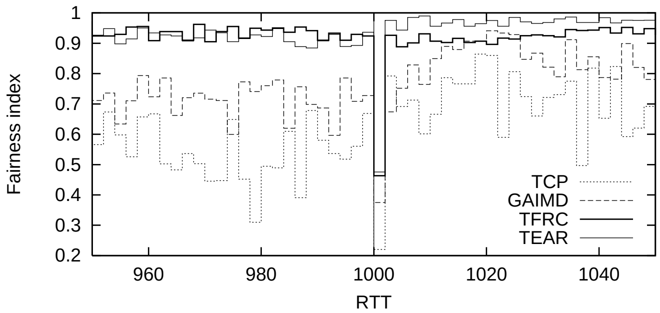

Figure 35: Fairness index (same protocol, 16 → 8 flows)

Another point to notice is how fast each protocol occupies newly available bandwidth. Figure 36 shows queue length traces for this experiment. For all four protocols, queue lengthes drop to zero when the 8 flows stop. TCP takes only 20 RTT's to fill the queue again. GAIMD and TFRC takes 42 and 45 RTT's, respectively. TEAR takes more than 200 RTT's because its sending rate increasing speed is very slow.

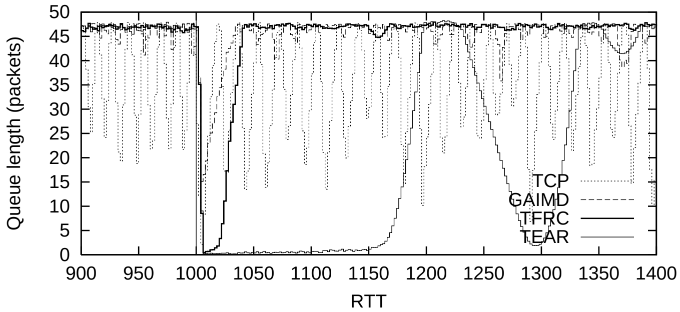

Figure 36: Queue length (same protocol, 16 → 8 flows)

6 Conclusion and Future Work

We studied analytically and via simulation the transient behaviors of TCP, GAIMD, TFRC, and TEAR. Table 1 summarizes our quantitative 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

	TCP	GAIMD	TFRC	TEAR
Fairness	F = 0.7 at low loss	F = 0.9 at low loss	F $\approx$ 1 at low loss; sending rate drops to 0 at high loss	F $\approx$ 1 at low loss; sending 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 discounting 0.22/RTT w/ history discounting	0.08/RTT, slower with delayed feedback

Table 1: Summary of quantitative results

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. While TEAR can maintain a stable smoothness performance up to 20% loss rate, 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.

Some issues that we have not studied include the impact of different round-trip times, and the transient behaviors of congestion control protocols in diffserv environments. Also, it will be interesting to investigate the impact of variations of sending rates (especially for TCP and GAIMD) on protocol responsiveness and aggressiveness. We defer these issues to a future study.

Acknowledgments

The authors thank the suggestions of Yanbin Liu, Yong Liu, Xincheng Zhang, and the anonymous reviewers.

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

[14] 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.

[15] Min Sik Kim, Nishanth R. Sastry, Yang Richard Yang, and Simon S. Lam. Is TCP-friendly viable? Technical Report TR–00–27, The University of Texas at Austin, November 2000.

[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, Department of Computer Science, North Carolina State University, Raleigh, North Carolina, U.S.A., 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] Lorenzo Vicisano, Luigi Rizzo, and Jon Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of IEEE INFOCOM '99*, volume 3, March 1999.

[26] 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.

[27] 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.

[28] Yang Richard Yang and Simon S. Lam. General AIMD congestion control. In *Proceedings of the 8th International Conference on Network Protocols*, Osaka, Japan, November 2000.

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

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 before 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.

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

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

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

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

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

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

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 [28], we can see that this will happen when loss rate is above 20%.

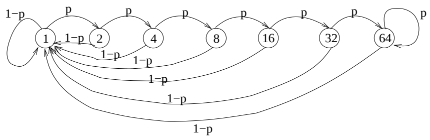

Figure 37: Timeout state transition diagram

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

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

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,

$$
\begin{aligned}
Var_{time}[T_{TEAR}] &= \sum_{i=1}^{8} w_i Y_i && (45)\\
&= 0.07708 Var[W_i] && (46)
\end{aligned}
$$

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

Therefore,

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

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 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:

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

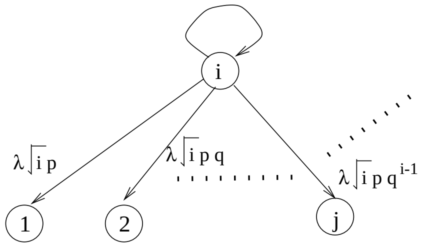

Figure 38: TFRC low loss rate state model

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

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

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

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.