

Modeling the Transient Rate Behavior of Bandwidth Sharing as a Hybrid Control System

Kang Li[†], Molly H. Shor[‡], Jonathan Walpole[♯], and Calton Pu^{*},

Abstract:

This paper uses hybrid control to model a problem of computer network systems, the dynamic behavior of bandwidth sharing among competing TCP traffic. It has been well known in the computer network community that well-behaved (TCP-friendly) congestion control mechanisms are crucial to the robustness of the Internet. Congestion control determines the transmission rate for each flow. Right now, most TCP-friendly research focuses only on the average throughput behavior without considering how the data is sent out in the short-term (e.g. bursty or smooth). However, recent experimental results show that short-term rate adjustments can change the bandwidth sharing result. Therefore, it is important to study the dynamic behavior of bandwidth sharing, and its impact on the long-term average throughput behavior. While existing network models and simulators for TCP-friendly control work well in general, they can not be used for theoretical proofs of the stability of the bandwidth-sharing behavior by competing TCP-friendly congestion controls. We want models that can capture the system's dynamic behavior to help us understand and predict the impact of short-term rate adjustments.

In this paper, we present a hybrid state-space-based model for the bandwidth sharing among TCP-friendly flows. We explain the hybrid nature of the system, which has a distributed control algorithm and is event-driven. The model helps us to understand the system's behavior in general and we use it to prove the system's stability under certain assumptions about packet loss. We also describe simulations and experiments to explore the cases in which these assumptions do not hold. This paper presents some preliminary results.

1. Introduction

A key issue facing computer network researchers is how to keep Internet bandwidth allocation to users fair and stable. Internet bandwidth is shared by all the competing flows that use it. The bandwidth allocation to each flow depends on the end-host congestion control mechanisms of all competing flows. Congestion control refers to a mechanism that enables the source to match its transmission rate to the currently available bandwidth. When a sender starts transmitting data to a receiver through the Internet, the available bandwidth between the sender and the receiver is often not known *a priori*. If the sender transmits too fast, it results in data accumulation in network routers and eventually data losses due to overflow of limited buffering capacities. If the sender transmits too slowly, the network bandwidth is underutilized. The congestion control mechanism should probe for newly available bandwidth and increase the sender's transmission rate to use it. It should also detect network congestion and reduce the sender's transmission rate at that time.

The allocation of an Internet link's bandwidth is controlled by the distributed congestion control mechanisms of all participating flows. In general, more bandwidth will be allocated to the one that expands its sending rate aggressively and reacts to congestion slowly. As an extreme example, a non-congestion-controlled flow usually can shut out all the congestion-controlled traffic and keep using all the bandwidth by itself. In past decades, Internet bandwidth was about "fairly" shared by competing traffic because almost all the competing traffic used the same congestion control mechanisms, the Transmission Control Protocol (TCP) [1]. Therefore, all the flows probed network bandwidth and reacted to congestion in a similar manner. As a result, competing TCP flows secured the same bandwidth under the same conditions (e.g., round-trip-time (RTT) and packet loss rate).

[♯] Dept. of Computer Science and Engineering, Oregon Graduate Institute, Beaverton, Oregon 97921-1000 USA. {kangli, walpole}@cse.ogi.edu +01-503-748-1267. Research was supported in part by DARPA Grant N66001-00-2-8901, NSF Grant CCR-9988440, and Intel.

[‡] Dept. of Electrical and Computer Engineering, Oregon State University, Corvallis, Oregon 97331-3211 USA. shor@ece.orst.edu +01-541-737-3186. Research was supported in part by NSF Grant ECS-9988435

^{*} College of Computing, CCB Room 261, Georgia Institute of Technology, Atlanta, Georgia 30332-0280 USA. calton@cc.gatech.edu

New “TCP-friendly” protocols have been proposed to serve the emerging class of multimedia applications. While TCP is appropriate for applications such as bulk data transfer, it does not serve well streaming multimedia applications. TCP’s variable delay, caused by retransmission and rate variations, may reduce the user’s perceived quality. Various TCP-friendly congestion controls have been proposed as alternatives to TCP. A flow is called TCP-friendly [3,6,11] if its long-term average transmission rate is equal to that of a normal TCP under the same packet loss rate.

We propose that only maintaining the same long-term average rate under the same average loss rate is not sufficient to keep the bandwidth shared fairly. A flow can perceive different loss rates depending on its short-term rate behavior and that of competing traffic. Existing works on keeping Internet bandwidth sharing fair and stable focus on average rate behavior. However, work on TCP pacing and the interactions between TCP and packet schedulers [4,5] have shown that short-term rate adjustments to a flow significantly affect the flow’s bandwidth share.

To study the impact of short-term rate behavior on a flow’s bandwidth share, we believe it is important to capture the transient behavior of bandwidth sharing among competing traffic. A typical congestion control is an event-driven non-linear system, with a variable, state-dependent sampling frequency. Its state jumps in value when congestion events (e.g. packet losses) happen. Therefore, we choose a hybrid dynamic model to describe the state of a bandwidth sharing system.

The purpose of this paper is to define the challenge of modeling the transient bandwidth sharing behaviors of TCP-friendly flows, and to use our model to demonstrate how hybrid control system models may help computer network research. This work is one of our series of steps to use feedback control theory to understand and attack the network resource allocation problem. This paper summarizes our early work [12, 13] on it and presents new experimental results with our hybrid control models.

The paper is organized as follows. Section 2 describes the bandwidth sharing system and TCP congestion controls in terms of feedback control systems. Section 3 presents our model for the bandwidth sharing system by competing TCP-friendly flows. We describe our simulation experiments based our control model in Section 4. We present related work in Section 5 and conclude the paper in Section 6.

2. TCP and the Bandwidth Sharing System

In this section, we first describe an individual TCP congestion control as a feedback-based control system. Then we describe the bandwidth sharing system with multiple competing TCP flows.

2.1 Single TCP as a non-linear feedback-based control system

TCP is an adaptive protocol, using the additive-increase multiplicative-decrease (AIMD) algorithm. TCP adjusts the sender’s data output rate according to the network path characteristics and the receiver side behaviors.

We can represent TCP congestion control as a feedback control system that outputs a signal onto the network to probe the network state, which is then used to control the data output rate. This feedback control system is illustrated in Figure 2.1. The feedback loop is composed of the rate controller, the probing signal that goes across the network, and the feedback monitor that monitors the sampling results and sends them to the rate controller.

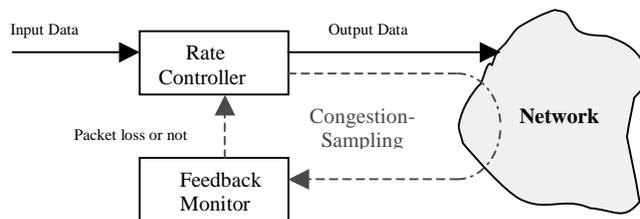


Figure 2.1: TCP Congestion Control System

TCP probes the network's state with the data it sends. Data packets travel from the sender to the receiver, and acknowledgments for each packet travel back from the receiver to the sender.

The time from sending a packet to receiving its acknowledgment is the round-trip time. The RTT is an important state variable in this system. This is the delay around the feedback loop, as well. The RTT varies primarily as a function of the buffer fill levels in the network path along which the data travels. The longer the packets must wait in buffers, the longer it takes them to traverse that path. A significant increase in RTT may be a useful indicator of network congestion.

If a packet arrives somewhere and the buffer is full, then it is lost. Each time the rate controller probes the network, the feedback monitor determines if the packet's acknowledgment returns or not. TCP detects this packet loss by looking at out-of-order acknowledgments. If acknowledgments have arrived for three packets that are sent out later than a certain packet that has not been acknowledged, then TCP decides that that packet is lost.

TCP controls the rate at which data is sent out on the network by using a congestion window. The congestion window size defines the maximum amount of outstanding data, data that has been sent but not yet acknowledged; hence, the amount that is sent out in one round-trip-time. The rate controller uses an AIMD algorithm to control the congestion window size, or out-going data rate. If acknowledgments are received for all packets that are sent during one RTT, then TCP increases its congestion window size by α (default is one packet); otherwise TCP decreases its congestion window to β times (default is half) the current window size.

In our early work [12], we claim that the state of TCP's congestion control converges to an attractor (stable limit cycle) for any given service rate (outgoing link transmission rate) and buffer size. TCP's state will transition from one attractor to another when the service rate or the buffer's size changes. Figure 2.2 shows the attractor of an individual TCP, based on a Simulink model we built for the system.

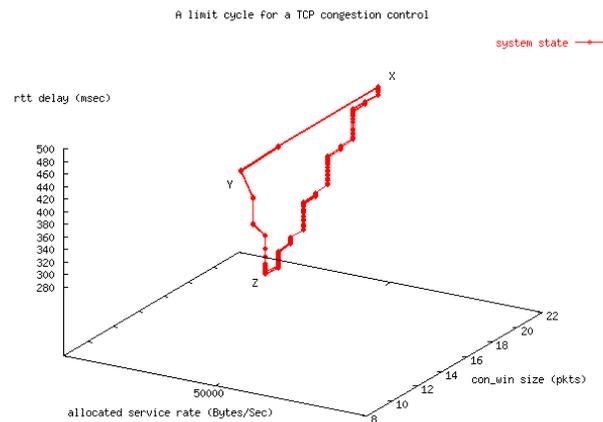


Figure 2.2 Individual TCP's system state as an attractor

2.2 Bandwidth Sharing System with Multiple AIMD TCPs

A bandwidth sharing system is composed of a network link resource and several competing flows. A network link resource is described as a leaky bucket that has a certain buffering capacity and a maximum transmission rate. The packet streams of all flows are multiplexed and read into the buffer, and the outgoing link transmits packets up to the maximum transmission rate. When the total of the rates of all incoming flows is higher than the outgoing link's maximum transmission rate, data is accumulated in the buffer. When the buffer reaches its capacity, some incoming packets will be dropped.

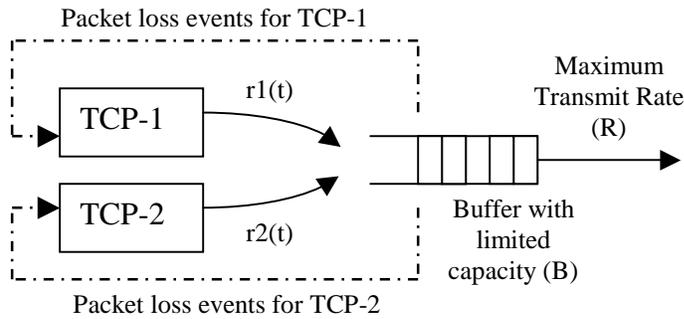


Figure 2.3 Two competing TCPs share one network link

Figure 2.3 shows two TCPs competing for bandwidth. The bandwidth share each one gets depends on both flows' congestion control mechanisms, such as how they react to packet loss events. Figure 2.4 is the system trajectory from our Simulink model when two TCPs use identical congestion control mechanisms but start with different initial rates.

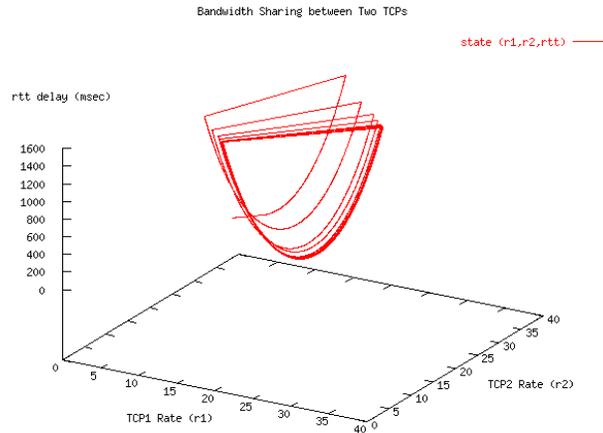


Figure 2.4 Two TCPs share bandwidth

Two important control objectives for bandwidth allocation among competing traffic are the link utilization and the fairness among competing flows. From the network resource's perspective, keeping the link fully utilized is important as long as the network is not congested. The fairness aspect of bandwidth sharing relates to preventing one flow from using up all the bandwidth and starving other users, given that there is no information available about the relative importance of competing flows. When n flows with large transmission requirements compete for a maximum rate R , for fairness, each of them should send at a rate R/n .

The trajectory shown in Figure 2.4 occurs under the condition that the two flows use exactly the same AIMD congestion control algorithm. They have the same short-term rate behavior and therefore are assumed to experience the same congestion signal. As a result they converge to exactly equal bandwidth share. However, experiments [5,8,12] have shown that different short-term rate behaviors could cause flows to experience different packet loss rates. If the two flows use different congestion control algorithms, or the same AIMD algorithm but with different increment/decrement parameters, this could lead to uneven bandwidth share. In the next section, we present a model for bandwidth sharing that can take into account short-term rate behavior. Then we make simulations based on the model for a system with flows that have different AIMD parameters.

3. A Hybrid State Space Model

To study the impact of short-term rate behavior on packet loss behaviors, which determine long-term throughput, we must capture the system dynamics of bandwidth sharing among multiple competing flows. The important states of this system include every competing flow's instantaneous transmission rate and the buffer fill-level of the bottleneck link. We assume a single, common, bottleneck for all the competing flows.

Our state-space based model includes all the important states that determine the bandwidth-sharing behavior. The essential feature of the concept of state for a dynamical system is that it should contain all information about the past history of the system that is relevant to its future behavior. That is to say, if the state at a given instance is known, then its subsequent evolution can be predicted without any other knowledge of what has previously happened to the system.

A state space model contains two parts: the system states that define the status of the system, and state equations that describe the migration of system states. Our model has the following system state definitions and state equations:

- System State

In the state-space model, the state of a system with N competing flows at time t is a vector $S(t) = [r_1(t), r_2(t), \dots, r_N(t), fl(t)]^T$, where $r_i(t)$ is the instantaneous transmission rate of flow i and $fl(t)$ is the buffer fill-level at time t . The system states need to be just enough to describe the system and predict future system states. Since we model a flow as a continuous fluid, its rate is a continuous value in time instead of a series of discrete packet sending events. Although using a continuous rate to represent the discrete packet sending events loses some states of the real system, we reduce the dimension of the system states to a manageable level.

In our model, these continuous rates plus the bottleneck buffer fill-level are sufficient to describe the system, and its subsequent evolution can be predicted without any other knowledge of what has previously happened to the system.

- State Equations

For each individual flow, we choose the following state equations to describe the dynamic states of a TCP-friendly flow that uses the AIMD algorithm. We use a continuous model for the increase to approximate the rate increment by one packet per RTT. Since the AIMD algorithm switches its rate control according to the existence of a congestion signal, we use two equations to describe the system behavior:

When no congestion happens (i.e. the additive increase phase):

$$\frac{dr(t)}{dt} = \frac{\alpha M}{RTT(t)^2} \quad \text{Equation (3.1),}$$

$$\frac{dRTT(t)}{dt} = \frac{r(t) - R}{R} \quad \text{Equation (3.2)}$$

When congestion happens (i.e. the multiplicative decrease phase):

$$r(t) \leftarrow \beta \times r(t) \quad \text{Equation (3.3),}$$

$$RTT(t) = C + B / R \quad \text{Equation (3.4).}$$

In the above equations, α and β are AIMD congestion control parameters, $r(t)$ is the flow's transmission rate, C is the round-trip propagation delay between sender and receiver, and \leftarrow indicates a state transition. The bottleneck link is described by a leaky bucket (R, B) , where R is the leak rate and B is the bucket size. The input rate to the bucket is the total rate of all competing traffic. Therefore, the bucket fill-level is controlled by

$$\frac{df(t)}{dt} = R - \sum r(t) \quad \text{Equation (3.5)}$$

When $f(t) = B$, a congestion signal is produced.

With this model, we proved the following theorem. Details of the model and the proof are presented in [12].

Theorem 1:

Assume RTT is approximately constant for a given flow. When multiple AIMD-based TCP-friendly flows compete for a constant available bandwidth, the system states converge to a stable limit cycle that passes through the point $P = [r_1, r_2, \dots, r_N, B]^T$, in which

$$r_i = \frac{2\beta}{1+\beta} \times \frac{1}{RTT_i^2} \times \frac{R}{\sum_{j=1}^N \frac{1}{RTT_j^2}} \quad \text{Equation (3.6)}$$

The above theorem shows that our state-space model is consistent with early work on the AIMD algorithm by Jain, et al. [2], and existing TCP-friendly congestion control work [3, 8, 9 and 10]. We have presented elsewhere that the trajectory of a system composed by a single TCP is a limit cycle [13]. The above theorem indicates that a system with multiple flows converges to a stable limit cycle when the flows’ average throughputs are equal to a normal TCP’s throughput, assuming that all the flows receive the congestion signal and “back off” simultaneously. This result indicates that AIMD-based or Equation-based TCP-friendly flows converge to a fair and stable state when the conditions are satisfied to guarantee that the flows’ average throughputs are equal to that of a normal TCP’s. This indicates that the TCP-friendliness definition based on long-term throughput is sufficient to guarantee fair bandwidth share among competing traffic under a synchronized feedback.

However, the proof [12] depends on the assumption that flows back off simultaneously. We believe this assumption to be unrealistic, since if congestion occurs randomly at the bottleneck, flows with more packets in the bottleneck are more likely to experience a congestion event. When we introduce a model of congestion that distributes packet losses to flows based on their transient transmission rate, we can show that the average throughput can be affected by short-term rate behaviors.

4. Simulation with our model

This section shows some of our experiments [14] based on the hybrid state space model. The target of these experiments is to demonstrate the effect of short-term rate adjustments on the long-term bandwidth share among competing flows.

We design our experiments as follows. We use two TCPs (called *TCP1* and *TCP2*) with different AIMD parameters as two flows with different end-to-end congestion controls. The AIMD parameters follow Equation 4.1, so in the long term the two flows get equal bandwidth share when they have the same packet loss rate [10].

$$\alpha = \frac{3(1-\beta)}{1+\beta} \quad \text{Equation (4.1)}$$

In our experiments, we adjust *TCP1*’s AIMD parameters, and keep *TCP2*’s at the default values as a reference for comparison. We assume that the bandwidth sharing system has other unknown competing traffic besides the two TCP flows. In our experiments, we aggregate all the other competing traffic as one flow, and simulate its rate as a Gaussian distributed random process with a mean of $R/2$ and variance $R/2$, where R is the link bandwidth

Real network measurement [7] indicates that packet loss events are often not evenly distributed to all competing flows. In reality, flows with high transmission rates are more likely to get packet losses when congestion happens. Furthermore, the loss distribution is not necessarily proportional to the transmission rate [8].

In our experiments, packet losses happen when the buffer starts to overflow, and which flow will experience packet losses is controlled by the function of $p = 1 - a * \text{Exp}(-b * r)$. With this function, flows with higher

transmission rates are more likely experience congestion every time the congestion process produces a congestion signal.

This simulation setup is described in Figure 4.1

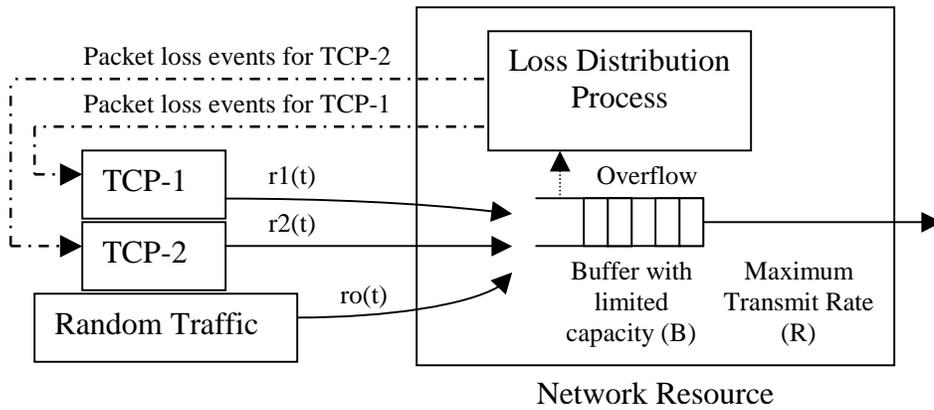


Figure 4.1 Simulation Setup

Each AIMD congestion control mechanism is simulated based the hybrid model in Section 3. Figure 4.2 shows the detail of one AIMD TCP control in Simulink.

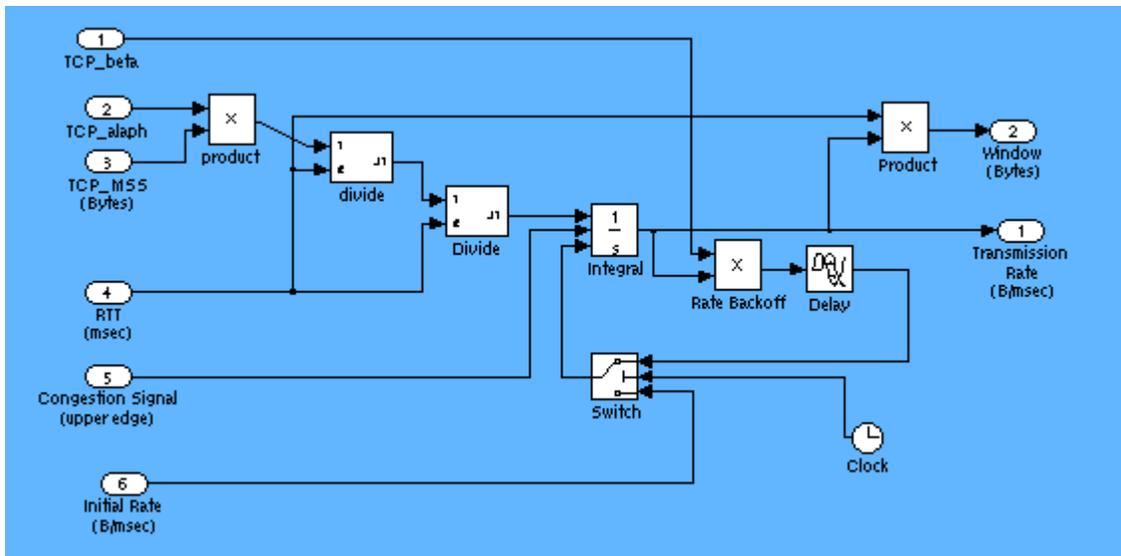


Figure 4.2 Simulation blocks for an AIMD TCP

In the experiments, two sets of parameters are adjusted. One is the AIMD control parameters (α, β), which control the short-term rate behaviors. We adjust α in a range from $1/5$ to $7/3$, while selecting β to satisfy Equation (4.1). The larger α is, the greater the rate variations in a flow's short-term rate behavior. The other set of parameters we adjust is the loss distribution parameters (a, b). In general, when b increases, the loss event distributions are approaching the uniform distribution, which indicates that all flows experience the same congestion signal. As b decreases, the loss event distributions are approaching the proportional distribution, which indicates the number of loss events experienced by a flow is proportional to its rate when congestion happens.

Figure 4.3 shows the average bandwidth ratio between the two TCPs as our experimental results. Each line in Figure 4.3 maps to a certain loss distribution function that is marked by its b value. For each loss distribution, we looked at the bandwidth share ratio when various AIMD TCP flows compete for bandwidth with a normal TCP flow. We run our experiments with 10 different seeds in the random generator for every b (controls the loss distribution) and α

(controls the TCP2's AIMD behavior) combination. We fixed a as a constant 1. Each point on a line in Figure 4.3 shows the average bandwidth share ratio of the two TCP flows, and their maximum and minimum bandwidth share ratio-measurements in every 10 experiments are shown with the error bar around the average point.

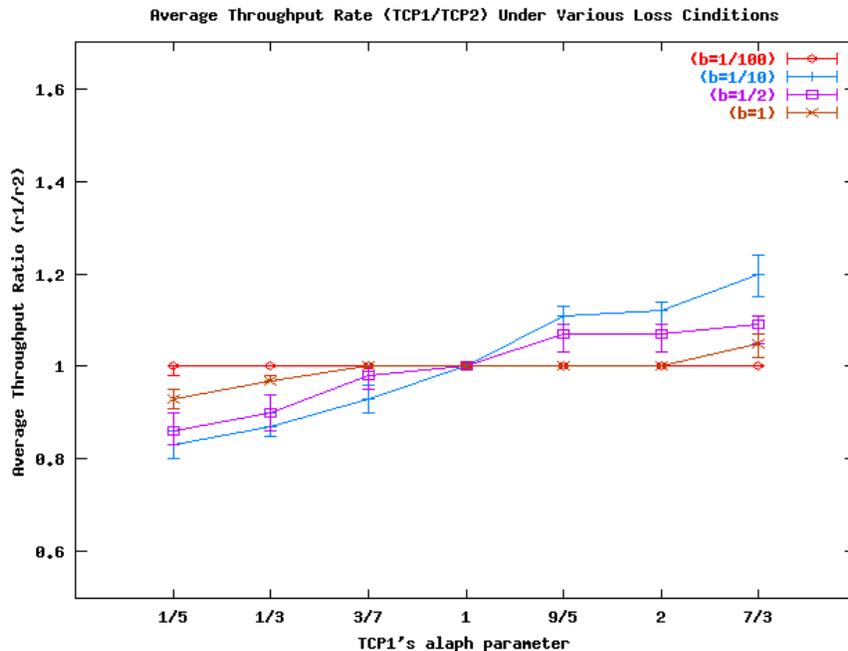


Figure 4.3 Bandwidth Sharing Ratio for two TCPs with various AIMD parameters and loss distributions.

The result shows that the loss distribution affects bandwidth share ratio, and short-term rate variations also contribute to bandwidth share difference.

When the two flows' congestion control behaviors are close (e.g., when TCP1's increment parameter α is close to the default value 1), the two flows are closer to even bandwidth share. This is because the two flows will not have very different transmission rates. When the two flows' short-term behaviors diverge, the bandwidth share diverges from even share. Our experimental results also indicate that an AIMD TCP with a larger increment parameter α (with larger variation in short-term rate behavior) tends to be more aggressive than an AIMD TCP with a smaller α .

When the loss event distribution is close to a uniform distribution, the two flows are close to having synchronized backing off. Therefore, their bandwidth shares do not have much difference. When the loss event distribution is proportional to the transmission rate, the bandwidth share ratio is also close to 1. When the distribution is in between, short-term bursts make the two flows share bandwidth differently. We plan to quantify the effect of bursts on bandwidth share ratio in the future.

5. Related Work

5.1 TCP-friendly

The term "TCP-friendliness" has been used to define the desired bandwidth sharing behavior of such protocols. For a protocol to be TCP-friendly it should consume the same bandwidth, over the long-term, as TCP under the same network conditions.

Various criteria for TCP-friendliness have been proposed in the literature. These can be divided into two classes: *equation-based* and *AIMD-based*. Both classes attempt to model the long-term average bandwidth consumption of a protocol. Equation-based criteria model the protocol's response to packet-loss and define it as TCP-friendly if, given the same loss rate, it consumes the same bandwidth as TCP. In contrast, AIMD-based criteria constrain the

relationship between the additive increase and multiplicative decrease parameters of TCP's bandwidth probing and congestion avoidance algorithms so that the long-term bandwidth share is the same as TCP's. A key assumption underlying both approaches is that the short-term rate variation behavior of the protocol does not influence the rate at which packet loss events¹ occur. By using a dynamic model of protocol behavior we show that this assumption is not valid, and that differences in short-term rate sharing behavior can lead to differences in packet-loss event rates and ultimately to differences in long-term bandwidth share. In other words, protocols that are TCP-friendly according to the equation-based and AIMD-based criteria can share bandwidth unfairly.

5.2 TCP Rate Pacing and Smoothing

According to queuing theory, smoothed traffic is good for global throughput. Rate pacing for TCP is proposed to improve TCP's behavior by changing its short-term rate behavior. However, recent previous work [5] on TCP pacing (TCP with smoothed output rate but maintains the same average rate) has shown a sophisticated behavior of bandwidth sharing result, and indicates that paced TCP flows experience different packet loss event rates than bursty TCP flows. The results from the TCP pacing work [5] motivate us to try to understand the dynamic behavior of bandwidth sharing.

6. Conclusion

In this paper, we focused on presenting a problem in the computer network domain that can be viewed as a hybrid control system. Our work is to study the effect of short-term rate variations on bandwidth sharing behavior. Certain aspects of bandwidth sharing behavior such as fairness and stability are important to the robustness of the Internet.

Currently, TCP-friendliness is the well-accepted notion used to build new congestion-control protocols that keep bandwidth fair share. At the same time, several research works have shown that the notion itself is not enough to guarantee bandwidth fair share. In our work, we built a hybrid state-space-based model and proved the stability of bandwidth sharing behavior under the assumption of synchronized backing off.

We also seek to understand the unfair share of bandwidth under asynchronous backing off and to propose some possible factors to study in the future. As we have seen, understanding bandwidth sharing among competing traffic is complicated. We believe that the problem of predicting and controlling the bandwidth sharing among competing flows is an important task for both the current and future Internet as long as the network resources are shared among users. To fully understand the problem requires more research work from both the network community and the control community.

Reference

- [1] Van Jacobson, and Michael J. Karels. "Congestion Avoidance and Control". *Proceeding of ACM SIGCOMM'88*, pp. 79-88, August 1988.
- [2] D. Chiu and R. Jain. "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks". *Computer Networks and ISDN Systems* 17, 1989
- [3] Jitendra Padhye, Jim Kurose, Don Towsley, and Rajeev Koodli. "A Model Based TCP-Friendly Rate Control Protocol". *UMass-CMPSCI Technical Report TR 98-04*, October 1998.
- [4] Kang Li, Jonathan Walpole, Dylan McNamee, Calton Pu and David Steere. "A Rate-Matching Packet Scheduler for Real-Rate Applications". To be appear in *MMCN'2001*
- [5] A. Aggarwal, S. Savage, T. Anderson. "Understanding the performance of TCP Pacing". *IEEE Infocom2000*.
- [6] Jitendra Padhye, Victor Firoiu, Don Towsley and Jim Kurose. "Modeling TCP Throughput: A Simple Model and its Empirical Validation". *ACM SICOMM'98*.
- [7] V. Paxson. "End-to-End Internet Packet Dynamics." *IEEE/ACM Transactions on Networking*, 7(3), pp. 277-292, June 1999.

¹ The packet loss event rate differs from the packet loss rate by counting consecutive packet losses in one RTT as one packet loss event.

- [8] S. Floyd, M. Handley, and J. Padhye. "Equation-Based Congestion Control for Unicast Applications". *SIGCOMM'2000*.
- [9] Sally Floyd, and Kevin Fall. "Promoting the Use of End-to-End Congestion Control in the Internet" *IEEE/ACM Transactions on Networking*, August 1999. Available at <http://www.aciri.org/floyd/papers.html>
- [10] J. Crowcroft and P. Oechslin. "Differentiated End-to-end Internet Services using a Weighted Proportional Fair Sharing TCP", *ACM Computer Communication Review*, Jul, 1998.
- [11] Matthew Mathis, Jeffrey Semke, and Jamshid Mahdavi. "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm". *ACM Computer Communication Review*, vol 27 num 3, July 1997. Available at http://www.psc.edu/networking/papers/model_abstract.html
- [12] K. Li, M. Shor, and J. Walpole. "OGI Tech-Report: Modeling the Effect of Short-term Rate Variations on TCP-Friendly Congestion Behavior". Submitted to the *2001 American Control Conference*, Sept. 2000. Available at <http://www.cse.ogi.edu/~kangli/>
- [13] M. Shor, K. Li, J. Walpole, D. C. Steere, and C. Pu. "Application of Control Theory to Modeling and Analysis of Computer Systems". *Proceedings of the Japan-USA-Vietnam Workshop on Research and Education in Systems, Computation and Control Engineering*, HoChiMinh City, Vietnam, June 7-9, 2000.
- [14] Simulation files for the State-Space model. <http://www.cse.ogi.edu/~kangli/simulink.html>

Appendix:

A prove of multiple TCP's behavior under the assumption of synchronous backing off:
Multiple competing TCPs compose a non-linear orbitally stable system with an attractor.

Assumption:

We assume the network bottleneck link follows a leaky bucket model, which is defined by a bucket size B and a leaky rate R . We concern the TCP congestion avoidance stage only, so we limit our discussion of TCP on its Additive Increase and Multiplicative Decrease (AIMD) algorithm.

Our assumption is: when the bucket fill-level reaches its maximum limit, all the competing TCPs lose packets and backoff at the same time. {In reality, it's possible only some of the TCPs get packet losses, and the time they backed off is related to their RTT.}

Our hypothesis is: In TCP's state space (phase portrait), where state variables are bucket fill-level and TCP's transmission rate, TCP has one attractor (a stable limit cycle). We draw the state plan in Figure-1.

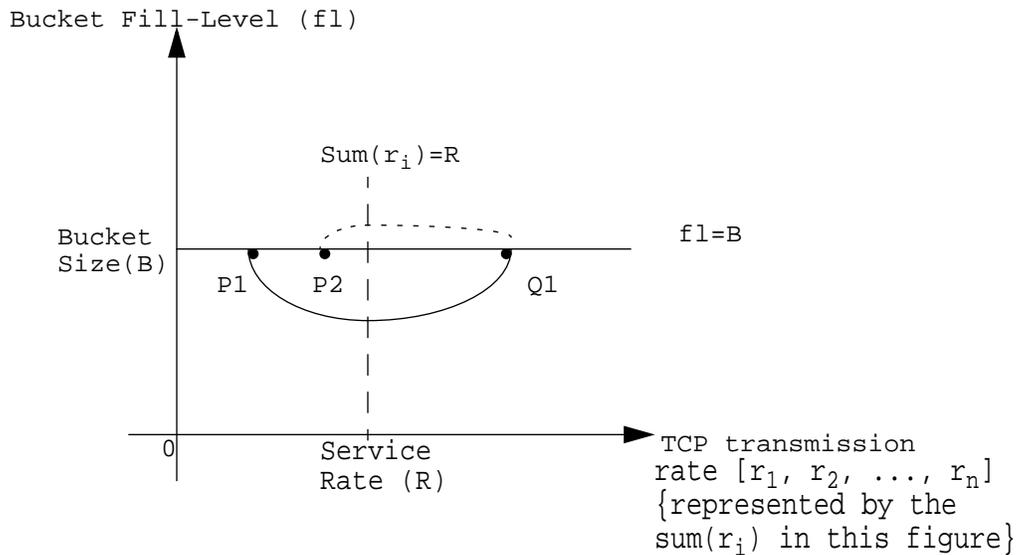


Figure-1: System State Space. System state can only stay in the region $\{0 \leq fl \leq B \text{ and } r_i \geq 0\}$. A trajectory is the path that indicates how the system state migrates in the state space. To make the trajectory clear, we use the dotted line to indicate the transition between two points in $fl=B$, which is otherwise covered by the line $fl=B$.

The TCP AIMD control's state equation are as follows:

When no congestion happen (uncongested state) $\{r \leq R \text{ or } FL < B\}$

$$\frac{dr_i(t)}{dt} = \frac{\alpha \times M}{RTT_i^2} , \quad (1)$$

$$\frac{d(fl(t))}{dt} = \sum_{i=1}^N r_i - R , \quad (2)$$

When congestion happen $\{ \sum_{i=1}^N r_i > R \text{ and } FL=B \}$

$$r_i \leftarrow \beta \times r_i \quad 0 < \beta < 1 , \quad (3)$$

The terms in the above equations is defined in Table-1:

$r_i(t)$:	The i-th TCP's transmission Rate at time t.
$fl(t)$:	Network link bucket fill-level at time t.
α :	TCP's linear increament parameter, and $\alpha \geq 0$
β :	TCP's exponential decreament parameter, and $0 < \beta < 1$
M :	TCP's maximum segment size. (We assume all TCP packets are this size)
RTT_i :	The i-th TCP's round trip time. (We assume RTTs are all constants).
R :	Network leaky bucket's leaking rate.
B :	Network leaky bucket's bucket size.
N :	Number of competing TCPs.

Table-1: Terms definitions

Now we start the prove. We divide the whole precEDURE of proving our hypothesis to prove 3 separate theorems about the system.

Lemma 1: All of the system's trajectories reach some point in the region $\{ fl=B, \text{ where}$

$0 \leq \sum_{i=1}^N r_i \leq R \}$. The region is shown as a line segment when project it to a two dimesion plot

with states $fl(t)$ and $\sum_{i=1}^N r_i(t)$, as shown in Figure-1.

We know the system state is bounded in the range of $\{ 0 \leq fl \leq B \text{ and } 0 \leq r_i \}$. However, this statement itself does not guarantee that all the trajectories will go accross the region $\{ fl=B,$

where $0 \leq \sum_{i=1}^N r_i \leq R \}$. We are going to prove Theorem 1 by first proving that a trajectory that

starts at a random point in the valid region $\{0 \leq fl \leq B, \text{ and } r_i \geq 0\}$ will eventually reach a point on line $fl=B$.

Assume a system state trajectory starts at some point $P(fl_0, r_{i0})$ in the valid range but not on line $fl=B$. Therefore, we have $0 \leq fl_0 < B$ and $r_{i0} \geq 0$. From equation (1) and (2), the bucket fill-level can be expressed as:

$$fl(t) = fl_0 + \int_0^t \left(\sum_{i=1}^N r_i(t) - R \right) dt \quad (4),$$

$$r_i(t) = r_{i0} + \int_0^t \frac{\alpha M}{RTT_i^2} dt = r_{i0} + \frac{\alpha M}{RTT_i^2} \times t \quad (5),$$

Replace $r(t)$ in Equation (4) with (5), we have

$$0 = [fl_0 - fl(t)] + \left\langle \sum_{i=1}^N r_{i0} - R \right\rangle t + \left\langle \sum_{i=1}^N \frac{\alpha M}{2RTT_i^2} \right\rangle t^2 \quad (6).$$

To know whether the trajectory reach $fl=B$, we let $fl(t)=B$ and get the solution of t in Equation(6) as:

$$t1 = \frac{-\left(\sum_{i=1}^N r_{i0} - R \right) + \sqrt{\left(\sum_{i=1}^N r_{i0} - R \right)^2 - 4(fl_0 - B) \times \sum_{i=1}^N \frac{\alpha M}{2RTT_i^2}}}{\sum_{i=1}^N \frac{\alpha M}{RTT_i^2}}, \text{ and .}$$

$$t2 = \frac{-\left(\sum_{i=1}^N r_{i0} - R \right) - \sqrt{\left(\sum_{i=1}^N r_{i0} - R \right)^2 - 4(fl_0 - B) \times \sum_{i=1}^N \frac{\alpha M}{2RTT_i^2}}}{\sum_{i=1}^N \frac{\alpha M}{RTT_i^2}}$$

Because of $fl_0 < B$, we are sure $t1$ is a positive value. Therefore, we know the trajectories start with

any valid point in the state space will reach line $fl=B$ after time $t1$. If at time $t1$, $0 \leq \sum_{i=1}^N r_i \leq R$,

then we prove the trajectory reaches the region $\{fl=B, \text{ and } 0 \leq \sum_{i=1}^N r_i \leq R\}$. If at time $t1$,

$\sum_{i=1}^N r_i > R$. Then according to our definition of congestion ($fl=B$ and $\sum_{i=1}^N r_i > R$), the system

enters the congestion state. According to Equation (3), rate r_i will reduce to βr_i ($0 < \beta < 1$) until the system enters uncongested state. Therefore, the trajectory will still reach the line segment

$\{fl=B, \text{ and } 0 \leq \sum_{i=1}^N r_i \leq R\}$. Lemma-1 is proved.

Lemma 2: The system state has one limit cycle.

We know any trajectory of the system reaches region $\{fl=B, 0 \leq \sum_{i=1}^N r_i \leq R\}$. Assume a trajectory passes a point P1 (B, r_{i1}) in Figure-1. We will derive the next cross point Q1(B, r_{i2}) between this trajectory and line $fl=B$.

With Equation (6), we have

$$\left(\left\langle \sum_{i=1}^N r_{i1} - R \right\rangle t + \left\langle \sum_{i=1}^N \frac{\alpha M}{2RTT_i^2} \right\rangle t^2 \right) = 0,$$

we can get the time that takes the system state goes from P1 to Q1:

$$t = \frac{R - \sum_{i=1}^N r_{i1}}{\frac{\alpha M}{2} \sum_{i=1}^N \frac{1}{RTT_i^2}} \quad (7).$$

Combine Equation (5) and (7), we know

$$r_{i2} = r_{i1} + \frac{2 \left(R - \sum_{j=1}^N r_{j1} \right)}{RTT_i^2 \sum_{j=1}^N \frac{1}{RTT_j^2}} \quad (8). \text{ \{If all TCPs have the same RTT, then}$$

$$r_{i2} = r_{i1} + 2N \left(R - \sum_{j=1}^N r_{j1} \right) \quad (8-1).$$

Since at point Q1 must be a congestion state according to discussion in Theorem 1, the next state after Q1, denoted by P2 (B, r_{i3}), has a simple relation with Q1:

$$r_{i3} = \beta \times r_{i2} \quad (9).$$

Simply according to Figure-1, if P1 and P2 is the same point, then we get a limit cycle for the system state. When we combine $r_{i3}=r_{i1}$ with Equation (9), we get

$$\sum_{j=1}^N r_{j1} = \frac{2\beta R}{1+\beta}$$

$$r_{i3} = r_{i1} = \frac{1}{RTT_i^2 \sum_{j=1}^N \frac{1}{RTT_j^2}} \times \frac{2\beta}{1+\beta} R \quad (10).$$

{If all TCPs have the same RTT, Equation (10) becomes

$$r_{i3} = r_{i1} = \frac{1}{N} \times \frac{2\beta}{1+\beta} R \quad \}$$

Therefore, we know if the system starts with P1(B, $r_i = \frac{1}{RTT_i^2 \sum_{j=1}^N \frac{1}{RTT_j^2}} \times \frac{2\beta}{1+\beta} R$), the state

trajectory will go back to P1. This result proves the existence of limit cycle in the system state space.

{Note: The result is consistent with the average throughput equation $r = \frac{a}{RTT \sqrt{p}}$, because of

$$\sqrt{\frac{1}{p}} = \sqrt{\bar{r} \times t} = \frac{1}{RTT_i^2 \sum_{j=1}^N \frac{1}{RTT_j^2}} \times \sqrt{\frac{1-\beta}{1+\beta} \times \frac{2}{\alpha M}} \times R \quad \}.$$

Theorem-1: The states of the system composed by multiple competing TCPs converge to an attractor. (In other words, the limit cycle in Lemma 2 is stable.)

From Equation (8) and (9), we can get the relation:

$$\Delta r_{i3} = \beta \left\langle 1 - \frac{2}{RTT_i^2 \sum_{j=1}^N \frac{1}{RTT_j^2}} \right\rangle \Delta r_{i1} \quad (11), \text{ in which } \Delta r_3 \text{ and } \Delta r_1 \text{ are some}$$

small deviations from the point (B, $r_i = \frac{1}{RTT_i^2 \sum_{j=1}^N \frac{1}{RTT_j^2}} \times \frac{2\beta}{1+\beta} R$) respectively.

Because of $0 < \beta < 1$, and $0 \leq \left\langle 1 - \frac{2}{N} \frac{RTT_i^2}{\sum_{j=1}^N RTT_j^2} \right\rangle \leq 1$, we know

$$|\Delta r_3| < |\Delta r_1|$$

This relationship indicates that the distance between any start point on line fl=B and P

(B, $r_i = \frac{1}{N} \frac{RTT_i^2}{\sum_{j=1}^N RTT_j^2} \times \frac{2\beta}{1+\beta} R$) will become smaller at the next time the system state reach

region $\{ \text{fl=B and } 0 \leq \sum_{i=1}^N r_i \leq R \}$ after a congestion backoff. The magnitude of this small deviation

will decrease with successive iterations. Therefore, any trajectory will approach the limit cycle in Lemma-2 as time goes to infinity, which is to say, the limit cycle will be stable.