Equilibrium and Fairness of Networks Shared by TCP Reno and Vegas/FAST

Ao Tang Jiantao Wang Sanjay Hegde Steven H. Low
California Institute of Technology, Pasadena, CA 91125, USA
{aotang, jiantao@cds, hegdesan@cs, slow}@.caltech.edu

September 29, 2005

Abstract

It has been proved theoretically that a network with heterogeneous congestion control algorithms that react to different congestion signals can have multiple equilibrium points. In this paper, we demonstrate this experimentally using TCP Reno and Vegas/FAST. We also show that any desired inter-protocol fairness is *in principle* achievable by an appropriate choice of Vegas/FAST parameter, and that intra-protocol fairness among flows within each protocol is unaffected by the presence of the other protocol except for a reduction in effective link capacities. Dummynet experiments and ns-2 simulations are presented to verify these results.

1 Introduction

The current TCP congestion control algorithms, TCP Reno and its variants, do not scale as the bandwidth-delay product continues to grow, e.g., [Hollot et al., 2002],[Low et al., 2003]. This has motivated several recent proposals on new algorithms for high speed networks, including TCP Westwood [Casetti et al., 2002], HSTCP [Floyd, 2003], STCP [Kelly, 2003], FAST TCP [Jin et al., 2004], and BIC TCP [Xu et al., 2004] (see [Jin et al., 2004] for extensive references). To incrementally deploy these protocols, we must understand networks running heterogeneous protocols. It has been shown in [Low, 2003] that any TCP–AQM can be interpreted as distributed primal-dual algorithms over a network to solve a utility maximization problem defined in [Kelly et al., 1998] and its Lagrangian dual [Low and Lapsley, 1999]; see also, e.g., [Mo and Walrand, 2000], [Yaiche et al., 2000], [Low et al., 2002], [Massoulie and Roberts, 2002], and [Kunniyur and Srikant, 2003]. Moreover, the utility functions that correspond to different TCP algorithms proposed in the literature all turn out to be strictly concave increasing, and hence the underlying maximization problem is a simple concave program. This underlying concavity is responsible for the relatively simple behavior of existing TCP–AQM models, both their equilibrium and dynamic properties. This interpretation holds, however, only when all protocols in the network react to the same kind of congestion signal (e.g., all react to loss or all to delay). When heterogeneous TCP algorithms that use different congestion signals share the same network, the situation becomes much more complicated. Even when equilibrium exists, it may no longer be the solution of a convex program.

There is little study on networks with heterogeneous protocols and most of the existing work has been limited to very simple topologies [Mo et al., 1999], [Kurata et al., 2000], [Low, 2003]. In [Grieco and Mascolo, 2004], inter-protocol fairness is studied through experiments in multi-bottleneck scenarios. In particular, it is showed that Vegas is not able to achieve a high bandwidth share when Reno is present. Recently, a general model is introduced in [Tang et al., 2005b] to systematically study the equilibrium of such networks. It is proved there that equilibrium indeed exists under mild conditions, it is generally non-unique, and virtually

all networks have finite number of (isolated) equilibrium points. The number of equilibrium points associated with the same set of bottleneck links must be odd, and not all of them can be locally stable unless the equilibrium is unique. This paper is motivated by two followup questions.

First, although [Tang et al., 2005b] provides examples that exhibit multiple equilibria, these examples are numerical based on a simple theoretical model and involve carefully designed utility functions. Can this phenomenon happen with real protocols? In this paper, we answer this unambiguously in the affirmative, using real implementations of TCP Reno and Vegas/FAST. Second, how do heterogeneous protocols share bandwidth between them (inter-protocol fairness), and how do flows within each protocol share among themselves (intra-protocol fairness)? We show that any desired degree of fairness between TCP Reno and Vegas/FAST is in principle achievable in general networks by an appropriate choice of Veags/FAST parameter, though it is an open problem how to compute this parameter in practice dynamically using only local information. Within each protocol, the flows would share the bandwidth among themselves *as if* they were in a single-protocol network, except that the link capacities are reduced by the amount consumed by the other protocol. In other words, intra-protocol fairness is unaffected by the presence of other protocols.

The paper is organized as follows. In Section 2, we specialize the general model introduced in [Tang et al., 2005b] to the case of TCP Reno and Vegas/FAST. In Section 3, we present a three-link network shared by these two protocols and derive a sufficient condition for the existence of two equilibrium points. This condition forms the basis of our experiments and simulations in later sections. In Section 4, we show that any desired fairness between TCP Reno and Vegas/FAST is achievable in principle by appropriate choice of Vegas/FAST parameter, and that the presence of the other protocol does not affect the intra-protocol fairness among flows running the same protocol. In Section 5, Dummynet experiments are reported to exhibit multiple equilibrium points in an emulated network. In Section 6, extensive simulations are provided to illustrate our theoretical results on multiple equilibria, and on inter-protocol and intra-protocol fairness. Finally we conclude in Section 7 with possible future directions to extend this work.

2 Model

We use "multiple protocols" and "heterogeneous protocols" interchangeably to denote congestion control algorithms that react to different congestion prices. To make our study concrete, we use two protocols that have been implemented. The first is TCP Reno which is loss-based. The second is TCP Vegas or FAST both of which are delay-based. Since both Vegas and FAST have identical equilibrium structure, we will often use FAST to generically refer to both FAST and Vegas. All networks in our experiments have three (bottleneck) links, for two reasons. First, it is shown in [Tang et al., 2005b] that a network with a full-rank routing matrix (which guarantees a unique price vector associated with each equilibrium) must have at least three links to exhibit more than one equilibrium. Hence a three-link network is the simplest set-up with full-rank routing matrix to allow an interesting behavior. Second, although empirical study shows that more than half of the paths in the Internet have at least one bottleneck link [Akella et al., 2003], very few of them (about 3 percent) experience more than three bottleneck links [Shriram and Kaur, 2003].

We now present a model for the equilibrium of a network shared by TCP Reno (loss-based protocol) and TCP FAST (delay-based protocol).

2.1 Notations

A network consists of a set of L links, indexed by l with finite capacities $c_l > 0$. It is shared by N^r Reno flows with equilibrium rates $x^r = (x^r_i, i = 1, \ldots, N^r)$, and N^f FAST flows with equilibrium rates $x^f = (x^f_i, i = 1, \ldots, N^f)$. The total number of flows is $N := N^r + N^f$. The packet loss probability is p^r_l, and the queueing delay is p^f_l at link l. Throughout this paper, the superscripts ^{r} and ^{f} are associated with

Reno and FAST, respectively.

The routing matrix for Reno flows is defined as R^r, where $R^r_{li} = 1$ if Reno flow i uses link l, and 0 otherwise. The routing matrix R^f is similarly defined for the FAST flows. The overall routing matrix can be expressed as $R = \left[R^r \;\; R^f \right]$. Finally, all quantities are valued at equilibrium.

The end-to-end loss probability vector for Reno flows is defined as

$$q^r = (R^r)^T p^r \tag{1}$$

Similarly the end-to-end delay experienced by FAST flows is :

$$q^f = (R^f)^T p^f \tag{2}$$

We assume that at each link l, the relation between loss probability and queueing delay is described by a *price mapping function* m_l:

$$p^r_l = m_l(p^f_l) \tag{3}$$

We assume that m_l are continuous and increasing with $m_l(\infty) = 1$, their inverses m_l^{-1} exist, and both m_l and m_l^{-1} are nonnegative functions. The exact form of m_l depends on the AQM (Active Queue Management) algorithm used at the link; see the mapping function for RED in Section 6.

2.2 Equilibrium model of Reno and FAST

An equilibrium point $(x^r, p^r) \in \Re_+^{N^r+L}$ of Reno is characterized by [Low, 2003]

$$q^r_i \;\; = \;\; \frac{2}{2 + (x^r_i)^2 T_i^2} \tag{4}$$

where the end-to-end loss probability q^r_i is given by (1). Here, we assume that the round-trip time T_i for Reno flow i is a constant.

As shown in [Low, 2003], Reno's rate solves the following problem

$$\max_{x^r_i \geq 0} \; U^r_i(x^r_i) - x^r_i q^r_i \tag{5}$$

with the utility function

$$U^r_i(x^r_i) = \frac{\sqrt{2}}{T_i} \tan^{-1}\left(\frac{x^r_i T_i}{\sqrt{2}} \right) \tag{6}$$

Both TCP Vegas and FAST have the same equation that characterizes their equilibrium $(x^f, p^f) \in \Re_+^{N^f+L}$ [Jin et al., 2004]:

$$x^f_i \;\; = \;\; \frac{\alpha^f_i}{q^f_i} \tag{7}$$

where α^f_i is a protocol parameter for FAST flow i and the end-to-end queueing delay q^f_i is given by (2). Similarly, FAST solves the following problem at equilibrium:

$$\max_{x^f_i \geq 0} \; U^f_i(x^f_i) - x^f_i q^f_i \tag{8}$$

with the utility function

$$U^f_i(x^f_i) = \alpha^f_i \log(x^f_i) \tag{9}$$

We say that the network is at equilibrium, or the link prices and flow rates are in equilibrium, when each flow maximizes its net benefit (utility minus bandwidth cost), and the demand for and supply of bandwidth at each link are balanced. Formally, an $(N+2L)$-dimensional nonnegative vector $(x,p)=(x^r,x^f,p^r,p^f)$ is an *equilibrium* if it satisfies (1)–(3), (4), (7), and

$$Rx - c \le 0, \quad P(Rx-c) = 0 \tag{10}$$

where P =diag$(p_l^f, l=1,\ldots,L)$. A set of links l that attains equality (10), i.e., links l with $\sum_i R_{li}^r x_i^r + \sum_i R_{li}^f x_i^f = c_l$, is called an *active constraint set*. The links in the active constraint set are said to be *saturated*. Note that there can be equilibrium points that have different active constraint sets. Indeed, in the example in this paper, the multiplicity of equilibrium points originates from different active constraint sets in equilibrium, and each constraint set admits a unique equilibrium.

If all flows react to the same congestion signal (i.e., $N^r = 0$ or $N^f = 0$), the equilibrium described above is the unique solution of the following utility maximization problem defined in [Kelly et al., 1998]:

$$\max_{x\ge 0} \sum_i U_i(x_i) \qquad \text{subject to } Rx \le c$$

with all utility functions U_i given by (6) or all by (9). This utility maximization problem provides a simple framework to study the properties of network equilibrium, e.g., the strict concavity of U_i guarantees the existence and uniqueness of the optimal solution. When heterogeneous protocols share the same network, (i.e., $N^r > 0$ and $N^f > 0$), although flows still do local optimization (5) and (8), they in general no longer optimize a social welfare.

3 An example with multiple equilibria

In this section, we derive a simple sufficient condition for the network in Figure 1 to exhibit two equilibrium points when it is shared by both TCP Reno and FAST. It forms the basis of Experiment 1 in Section 5.2 and simulations in Sections 6.1.

Consider the symmetric network in Figure 1 with 3 links l with capacities c_l. There are FAST flows

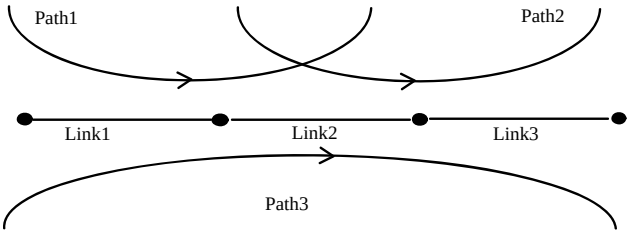


Figure 1: Multiple equilibria scenario.

that use path 1 and path 2. They have a common utility function denoted by U^1 and a common source rate denoted by x^1. There is a Reno flow that use path 3. Its utility function is denoted by U^2 and its rate by x^2. Their routing matrices are respectively

$$R^1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad R^2 = (1,1,1)^T$$

Links 1 and 3 both have capacity c_1 and a price mapping function m_1. Let $p_1 := p_1^f$ denote the queueing delay (price for FAST) at links 1 and 3, and let $m_1(p_1^f)$ be the loss probability (price for Reno) at these links.

Link 2 has capacity c_2 and a price mapping function m_2. Let $p_2 := p_2^f$ and $m_2(p_2)$ be the queueing delay and loss probability at link 2.

The following proposition provides a sufficient condition for multiple equilibria, under the following assumption:

A1: Utility functions U^j are strictly concave increasing, and twice continuously differentiable in their domains. Price mapping functions m_l are continuously differentiable in their domains and strictly increasing with $m_l(0) = 0$.

The key idea is to design two scenarios with different active constraint sets, each of which has a unique equilibrium. In the first equilibrium, only links 1 and 3 are saturated, whereas in the second equilibrium, only link 2 is saturated. Note that the condition applies more generally than just to (the utility functions of) FAST and Reno.

Proposition 1. *Suppose assumption A1 holds. The network shown in Figure 1 has two equilibria provided:*

1. $c_1 < c_2 < 2c_1$*;*
2. *for* $j = 1, 2$, $(U^j)'(x^j) \to \overline{p}^j$ *for some* $\overline{p}^j$ *possibly* ∞*, if and only if* $x^j \to 0$*;*
3. *for* $l = 1, 2$, $m_l(p_l) \to \overline{p}^2$ *as* $p_l \to \overline{p}^1$*, and satisfy*

$$2m_1((U^1)'(c_2 - c_1)) \; < \; (U^2)'(2c_1 - c_2) < \; m_2((U^1)'(c_2 - c_1))$$

Proof: We first claim that, if $c_1 < c_2$ and $(U^2)'(2c_1 - c_2) > 2m_1^2((U^1)'(c_2 - c_1))$, then there is an equilibrium point where only links 1 and 3 are saturated and link 2 is not. In this case the equilibrium price for link 2 is $p_2 = 0$ and, by symmetry, those for links 1 and 3 are both p_1. Such an equilibrium, if exists, is defined by the following equations:

$$\begin{array}{cc} (U^1)'(x^1) = p_1 & (U^2)'(x^2) = 2m_1(p_1) \\ x^1 + x^2 = c_1 & 2x^1 + x^2 < c_2 \end{array}$$

Eliminating x^2 and p_1, the above equations are reduced to:

$$(U^2)'(c_1 - x^1) \;=\; 2m_1((U^1)'(x^1)) \tag{11}$$

$$x^1 \;<\; c_2 - c_1 \tag{12}$$

An equilibrium exists if and only if (11)–(12) has a nonnegative solution for x^1. We now show that (11)–(12) indeed admits a unique solution $x^* > 0$ under the hypothesis of the proposition.

When $x^1 = 0$, we have

$$(U^2)'(c_1 - x^1) = (U^2)'(c_1) < \overline{p}^2 \le 2\overline{p}^2 = 2m_1((U^1)'(0))$$

The inequality and the last equality have made multiple use of conditions 2 and 3 of the proposition. On the other hand, when $x^1 = c_2 - c_1$, we have $U_2'(2c_1 - c_2) > 2m_1(U_1'(c_2 - c_1))$ by condition 3. Since all functions here are continuous, $(U^j)'$ are strictly decreasing, and m_l are strictly increasing, there exists a unique $0 < x^* < c_2 - c_1$ such that $(U^2)'(c_1 - x^*) = 2m_1((U^1)'(x^*))$.

We next claim that, if $c_2 < 2c_1$ and $(U^2)'(2c_1 - c_2) < m_2((U^1)'(c_2 - c_1))$, then there is an equilibrium point where only link 2 is saturated and links 1 and 3 are not. In this case $p_1 = 0$, and the following equations determine such an equilibrium:

$$\begin{array}{cc} (U^1)'(x^1) = p_2 & (U^2)'(x^2) = m_2(p_2) \\ x^1 + x^2 < c_1 & 2x^1 + x^2 = c_2 \end{array}$$

Eliminating x^2 and p_2, the equilibrium is specified by

$$(U^2)'(c_2 - 2x^1) = m_2((U^1)'(x^1)) \quad (13)$$

$$x^1 > c_2 - c_1 \quad (14)$$

When $x^1 = c_2 - c_1$, we have

$$(U^2)'(c_2 - 2x^1) = (U^2)'(2c_1 - c_2) < m_2((U^1)'(x^1))$$

by condition 3. When $x^1 = c_2/2$

$$(U^2)'(c_2 - 2x^1) = (U^2)'(0) = \overline{p}^2 > m_2((U^1)'(x^1))$$

where we have used conditions 2 and 3. Hence, again, there is a unique x^* that satisfies (13)–(14). Moreover, from (12) and (14), the two equilibria are distinct. □

Remark: It can be checked that the utility functions of FAST and Reno satisfy A1 and condition 2 of the proposition, with $\overline{p}^1 = \infty$ and $\overline{p}^2 = 1$. Hence the key condition in Proposition 1 is condition 3 on the price mapping functions m_1 and m_2. As mentioned above, the key idea in realizing the equilibrium points is to make links 1 and 3 sustain large delay and small loss, and make link 2 sustain large loss and small delay.

To be more specific, consider linear price mapping function at links $m_l(p) = k_l p_l$ for $p_l \in [0, 1/k_l]$. This can be viewed as an approximate model for RED (see below). The condition in the proposition then translates into the following two inequalities

$$k_1 < \frac{(U^2)'(2c_1 - c_2)}{2(U^1)'(c_2 - c_1)} \quad \text{and} \quad k_2 > \frac{(U^2)'(2c_1 - c_2)}{(U^1)'(c_2 - c_1)}$$

This implies that $k_2/k_1 > 2$ is necessary for the (sufficient) condition in the proposition to hold. This suggests that the AQM algorithms at various links need to be sufficiently different for multiple equilibria. This intuition is made precise in [Tang et al., 2005b].

Although there is no analytical model for the price mapping function for Droptail router, we can conceivably satisfy the requirement by using large buffers for link 1 and link 3 while using a small buffer for link 2. Indeed, this is how we demonstrate the phenomenon of multiple equilibria using Dummynet testbed in Experiment 1 in Section 5.

4 Inter and intra-protocol fairness

In this section, we study fairness in networks shared by TCP Reno and FAST. Two questions we address are: how these two protocols share bandwidth in equilibrium, and how the flows within each protocol share among themselves. We start with the second question.

4.1 Intra-protocol fairness

As indicated above, when the network is shared only by TCP Reno flows or only by FAST flows, the equilibrium flow rates are the unique optimal solution of a utility maximization problem with corresponding utility functions (6) or (9) respectively. In another words, the utility functions describe how the flows share bandwidth among themselves. For instance, the $\log$ utility function of FAST implies that it achieves weighted proportional fairness. When TCP Reno flows and FAST flows share the same network, it turns out that this feature is preserved "locally" within each protocol, as we now show. In particular, it implies that the intra-protocol fairness of FAST is still proportional fairness.

Proposition 2. *Given an equilibrium $(\hat{x}^r, \hat{x}^f, \hat{p}^r, \hat{p}^f) \geq 0$, let $\hat{c}^f := R^f \hat{x}^f$ be the total bandwidth consumed by FAST flows at each link. The FAST flow rates $\hat{x}^f$ are the unique solution of:*

$$\max_{x \geq 0} \sum_{i=1}^{N^f} U_i^f(x_i) \qquad \textit{subject to } \; R^f x \leq \hat{c}^f \tag{15}$$

where $U_i^f(x_i)$ are given by (9). The Reno flow rates $\hat{x}^r$ are the unique solution of:

$$\max_{x \geq 0} \sum_{i=1}^{N^r} U_i^r(x_i) \qquad \textit{subject to } \; R^r x \; \leq \; c - \hat{c}^f$$

where $U_i^r(x_i)$ are given by (6).

Proof: Since $(\hat{x}^r, \hat{x}^f, \hat{p}^r, \hat{p}^f) \geq 0$ is an equilibrium, from (7) and (2), we have

$$\frac{\alpha_i^f}{\hat{x}_i^f} \;\; = \;\; \sum_l R_{li}^f \hat{p}_l^f \quad \text{for } \; i = 1, ..., N^f$$

This, together with (from the definition of $\hat{c}^f$)

$$\sum_i R_{li}^f \hat{x}_i^f \leq \hat{c}_l^f, \qquad \hat{p}_l^f \left(\sum_i R_{li}^f \hat{x}_i^f - \hat{c}_l^f \right) = 0, \quad \forall l$$

form the necessary and sufficient condition for $\hat{x}^f$ and $\hat{p}^f$ to be optimal for (15) and its dual respectively.

The proof for Reno rates $\hat{x}^r$ follows the same argument with loss probabilities $m_l(\hat{p}_l^f)$ as the Lagrange multipliers. □

Note that in Proposition 2, the "effective capacities" $\hat{c}^f$ and $c - \hat{c}^f$ for FAST and Reno are not pre-assigned. They are the outcome of competition between FAST and Reno and are related to inter-protocol fairness, which we now discuss.

4.2 Inter-protocol fairness

Even though TCP Reno and FAST individually solve a utility maximization problem to determine their intra-protocol fairness, they in generally do not jointly solve any convex utility maximization problem. This makes the study of inter-protocol fairness hard.

The equilibrium rates x^f and x^r of FAST and Reno flows, respectively, depend on FAST parameter $\alpha = (\alpha_i, i = 1, \ldots, N^f) \geq 0$. Let $\overline{x}^f(\alpha)$ be the unique FAST rates if there were no Reno flows, and let $\overline{x}^r$ be the unique Reno rates if there were no FAST flows. Let $\underline{x}^f(\alpha)$ be the unique FAST rates if network capacity is $c - R^r \overline{x}^r$. Let

$$X^* \;\; := \;\; \{ \, x^f \, | \, \underline{x}^f(\alpha) \leq x^f \leq \overline{x}^f(\alpha), \; \alpha \geq 0 \, \}$$

X^* includes all possible FAST rates if FAST were given strict priority over Reno or if Reno were given strict priority over FAST, and all rates in between. In this sense X^* contains the entire spectrum of inter-protocol fairness between TCP Reno and FAST. The next result says that every point in this spectrum is achievable by an appropriate choice of FAST parameter α.

Let $x^f(\alpha)$ denote the unique equilibrium rates of FAST flows sharing the same network (R, c) with Reno flows when the protocol parameter is α. It is determined by (1)–(3), (4), (7), and (10).

Proposition 3. *Given any $x^* \in X^*$, there exists an $\alpha^* \geq 0$ such that $x^f(\alpha^*) = x^*$.*

Proof: Given any $x^* \in X^*$, the capacity for all Reno flows is $c - R^f x^*$. Since $x^* \leq \overline{x}(\alpha)$ (for all coordinates), we have $c - R^f x^* \geq c - R_f \overline{x}(\alpha)$, which is greater than or equal to 0 by the construction of $\overline{x}(\alpha)$. Hence the following utility maximization problem solved by TCP Reno is feasible:

$$\begin{aligned} \max_{x \geq 0} \quad & \sum_i U_i^r(x_i) \\ \text{subject to} \quad & R^r x \ \leq \ c - R^f x^* \end{aligned}$$

From Proposition 2, the unique optimal solution is Reno flow rates in equilibrium. Let p^r be an associated Lagrange multiplier vector. Choose α^* with $\alpha_i^* = x_i^* \sum_l R_{li}^f m_l^{-1}(p_l^r)$. It can be checked that all equilibrium equations are satisfied. □

Remark: Proposition 3 implies that given any target fairness between TCP Reno and FAST, in terms of a desirable rate allocation x^* for FAST, there exists a protocol parameter vector α^* that achieves it. It is however an open problem how to compute α^* in practice dynamically using local information.

5 Experiments

5.1 Testbed Setup

We set up a Dummynet testbed with seven Linux servers as senders and receivers and three BSD servers to emulate software routers; see Figure 2. The Linux senders and receivers run TCP Reno or FAST. The three

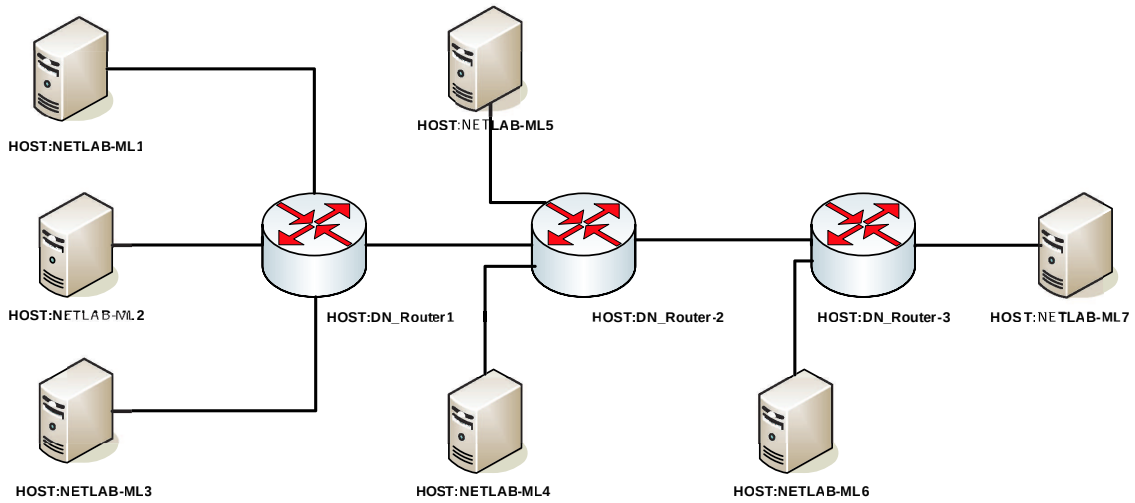


Figure 2: Dummynet setup for Experiments 1 and 2.

emulated routers run FreeBSD 5.2.1. Each testbed machine has dual Xeon 2.66GHz, 2GB of main memory, and dual on-board Intel PRO/1000 Gigabit Ethernet interfaces. The test machines are interconnected through a Cisco 3750 Gigabit switch. The network is fully configurable, and the link delay and capacity can be modified on the emulated router. The queueing discipline is Droptail. We have programmed the Dummynet router to capture various state variables to compute queue trajectories, loss and utilization. The sender and receiver hosts have been instrumented using kernel instrumentation tools to monitor different TCP state variables. We use 2.4.22 modified FAST kernel and Linux kernel. In order to minimize host limitations and

accommodate large bursts, we have increased the Linux transmission queue length to 5000 and ring buffer to 4096. Iperf is used to generate TCP traffic for each protocol.

We make the following remarks before presenting our results in detail:

- We modified FAST implementation so that it does not halve its window after a loss. Therefore it only reacts to queueing delay, as in [Tang et al., 2005b].
- Standard 1500-byte MTU (Maximum Transmission Unit) is used. Then, e.g., 100 Mbps = 8.33 pkts/ms.
- All the queue sizes reported below are exponential moving average of instantaneous queue trajectories. Averaging does not affect the equilibrium value, which is our primary interest. Note however that even though the averaged trajectory may not reach buffer capacity, the instantaneous trajectory often does.

5.2 Experiment 1: multiple equilibria

The goal of this experiment is to demonstrate on our Dummynet testbed the two equilibrium points guaranteed by Proposition 1. The topology of the network is shown in Figure 1. Links 1 and 3 (which correspond to the outgoing links of routers 1 and 3) are each configured with 110 Mbps capacity, 50 ms one-way propagation delay and a buffer of 800 packets. Link 2 (router 2) has a capacity of 150 Mbps with 10 ms one-way propagation delay and buffer size of 150 packets. There are 8 Reno flows on path 3 utilizing all the three links, with one-way propagation delay of 110 ms. There are two FAST flows on each of paths 1 and 2. Both of them have one-way propagation delay of 60 ms. All FAST flows use a common $\alpha = 50$ packets.

Two sets of experiments have been carried out with different starting times for Reno and FAST flows. The intuition is that if FAST flows start first, link 2 will be saturated and links 1 and 3 will not. Since the buffer size for link 2 is small, when Reno flows join, they will experience so many losses that links 1 and 3 will remain unsaturated. This corresponds to an equilibrium with an active constraint set consisting of link 2 only. If Reno start first, on the other hand, links 1 and 3 are saturated while link 2 is not because link 2 has a higher capacity. Since the buffer size at links 1 and 3 are large, they can generate enough queueing delay to squeeze FAST flows when they join and keep link 2 unsaturated. This corresponds to an equilibrium with an active constraint set consisting of links 1 and 3. We repeat the experiments 30 times for both scenarios. Now we report the results.

The average aggregate rates and the standard deviation over the 30 experiments of all the flows on each of paths 1, 2, 3 are shown in Table 1 when FAST flows start first and when Reno flows start first. Since

	Path 1 (FAST)	Path 2 (FAST)	Path 3 (Reno)
FAST start first	(52.0, 2.0) Mbps	(61.1, 3.3) Mbps	(26.6, 4.8) Mbps
Reno start first	(13.3, 0.8) Mbps	(13.4, 0.8) Mbps	(92.7, 0.7) Mbps

Table 1: Average aggregate rates and their standard deviations of all flows on paths 1, 2, 3.

the difference of the aggregate rates for each path is far more than the standard deviation, it is clear that the network has reached very different equilibria depending on which flows start first. This is further confirmed by queue and throughput measurements shown in Figure 3 for link 1 and in Figure 4 for link 2 for one of the thirty experiments. The results for link 3 are similar to those for link 1 and are omitted. These figures show that when FAST flows start first, link 2 queue remains nonzero while link 1 (and hence link 3) queue remains empty throughout the experiment, as expected. As a consequence, the aggregate throughput at link 2 is close to capacity while that at link 1 remains low most of the time. When Reno flows start first, the queue and throughput behaviors are exactly opposite.

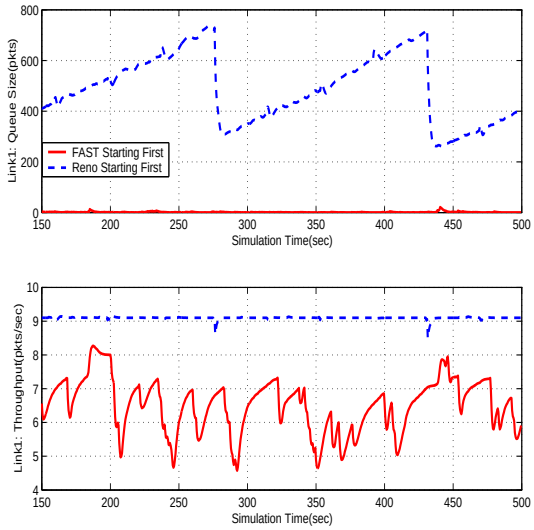


Figure 3: Experiment 1: queue size and aggregate throughput at link 1.

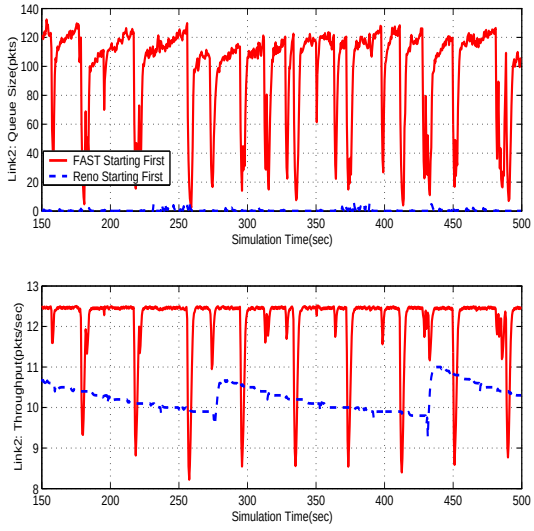


Figure 4: Experiment 1: queue size and aggregate throughput at link 2.

To make sure that the above behavior is indeed due to multi-protocol rather than different flow arrival patterns, we repeated the experiment with the same network setup, but using all Reno or all FAST flows. When we used FAST flows along the long path, the α is set to 30. The average throughput results are summarized in the Table 2 and 3. They confirm that the network admits a unique equilibrium when a single protocol is used, regardless of flow arrival patterns.

	Path 1	Path 2	Path 3
Short flows start first	(47.7, 1.3) Mbps	(70.1, 1.8) Mbps	(13.4, 1.0) Mbps
Long flow starts first	(40.7, 1.5) Mbps	(64.9, 2.0) Mbps	(21.4, 1.2) Mbps

Table 2: Average aggregate rates and their standard deviations of all flows on paths 1, 2, 3 (All flows are Reno).

	Path 1	Path 2	Path 3
Short flows start first	(47.2, 1.1) Mbps	(72.3, 1.6) Mbps	(15.6, 1.2) Mbps
Long flow starts first	(46.8, 1.3) Mbps	(72.0, 1.7) Mbps	(16.3, 1.0) Mbps

Table 3: Average aggregate rates and their standard deviations of all flows on paths 1, 2, 3 (All flows are FAST).

5.3 Experiment 2: unique equilibrium

The linear network in Figure 5 is proved in [Tang et al., 2005b] to admit a unique equilibrium. Experiment 2 verifies this. Each Dummynet router is configured to have 40 ms one-way propagation delay and 200-packet buffer. The link bandwidth is 100 Mbps for link 1, 150 Mbps for link 2, and 120 Mbps for link 3. There are three FAST TCP flows using the paths 1, 2 and 3 with one flow on each path. There are eight Reno flows using path 4. 30 experiments are done for each scenario.

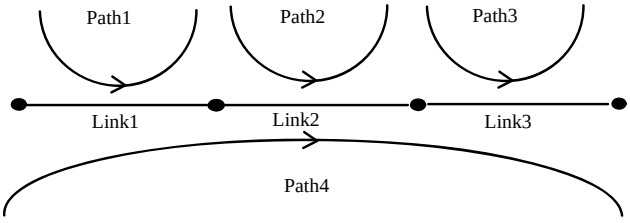


Figure 5: Experiment 2: unique equilibrium.

The average aggregate flow rates and their standard deviations on each of paths 1, 2, 3, 4 are shown in Table 4. They suggest that the network has reached the same equilibrium regardless of which flows start first. This is further confirmed by the queue and throughput trajectories at links 1–3 in Figures 6–8. At each link, the queue and throughput behaviors are very similar whether FAST or Reno flows start first.

6 Simulations

The Dummynet experiments provide qualitative evidence of multiple equilibria with practical protocols. We could not have verified the experimental results with quantitative predictions because Droptail router does

	Path 1 (FAST)	Path 2 (FAST)	Path 3 (FAST)	Path 4 (Reno)
FAST start first	(47.8, 2.7) Mbps	(96.2, 2.8) Mbps	(67.2, 2.8) Mbps	(47.9, 2.7) Mbps
Reno start first	(46.1, 0.8) Mbps	(94.2, 0.8) Mbps	(64.6, 3.7) Mbps	(43.7, 1.9) Mbps

Table 4: Average aggregate rates and their standard deviations of all flows on paths 1, 2, 3, 4.

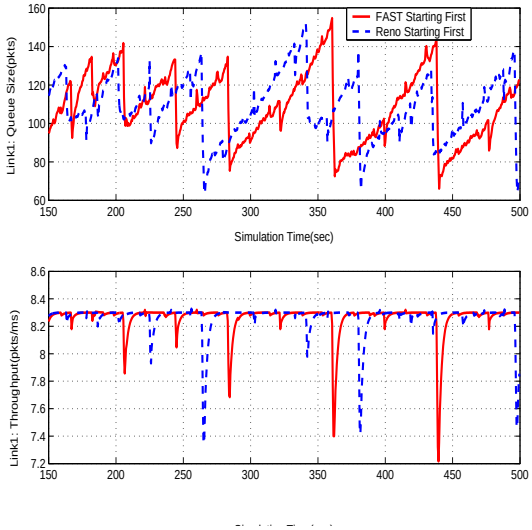


Figure 6: Experiment 2: queue size and aggregate throughput at link 1.

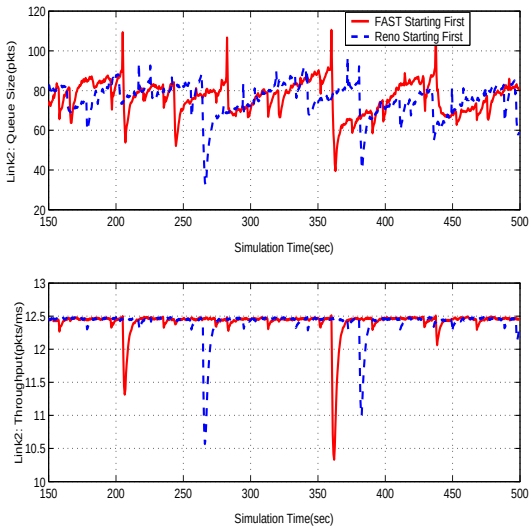


Figure 7: Experiment 2: queue size and aggregate throughput at link 2.

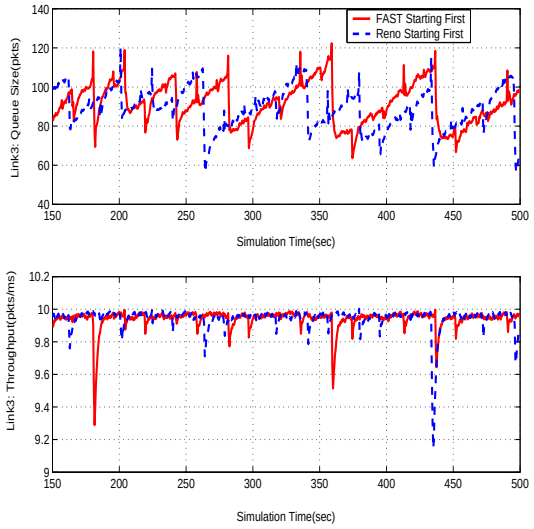


Figure 8: Experiment 2: queue size and aggregate throughput at link 3.

not admit an accurate mathematical model for the price mapping function m_l. In this section, we present simulation results using ns-2 on multiple equilibria and fairness. The network simulator ns-2 allows us to use RED router for which the price mapping function m_l is known. We can thus compare simulation measurements with our theoretical predictions. As there is not a mature ns-2 implementation of FAST yet, we use TCP Vegas for all the simulations. Since Vegas has the same equilibrium structure as FAST, it does not affect our study of equilibrium properties.

The network simulator ns-2 version 2.1b9a is used here. We use RED algorithm and packet marking instead of dropping. The marking probability $p(b)$ of RED is a function of queue length b:

$$p(b) = \begin{cases} 0 & b \le \underline{b} \\ \frac{1}{K}\frac{b-\underline{b}}{\overline{b}-\underline{b}} & \underline{b} \le b \le \overline{b} \\ \frac{1}{K} & b \ge \overline{b} \end{cases} \tag{16}$$

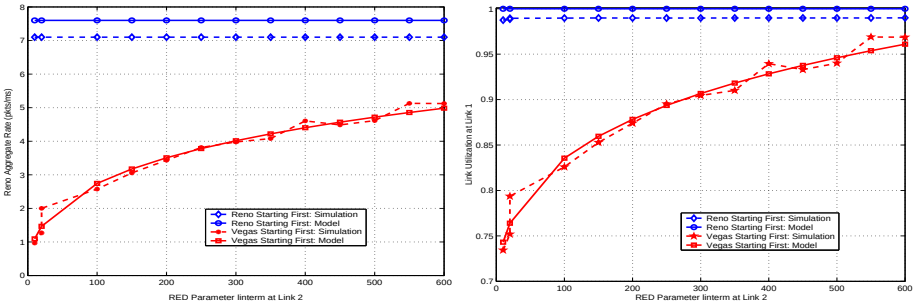
where $\underline{b}$, $\overline{b}$ and K are RED parameters. The price mapping function m_l in (3) which relates loss and delay can now be explicitly expressed as:

$$p_l^r = m_l(p_l^f) = \begin{cases} 0 & p_l^f \le \frac{\underline{b}}{c_l} \\ \frac{1}{K}\frac{p_l^f c_l - \underline{b}}{\overline{b}-\underline{b}} & \frac{\underline{b}}{c_l} \le p_l^f \le \frac{\overline{b}}{c_l} \\ \frac{1}{K} & p_l^f \ge \frac{\overline{b}}{c_l} \end{cases} \tag{17}$$

6.1 Multiple equilibria

The network topology is as shown in Figure 1. The link capacities of link 1 and link 3 are set to be 100 Mbps (8.33pkts/ms) and the one way propagation delay to be 50 ms. For link 2, the capacity is 150 Mbps (12.5pkts/ms) and one way propagation delay is 5 ms. There are 10 Vegas flows on each of paths 1 and 2, and 20 Reno flows on path 3. As in ns simulations, αd is the number of packets the flow maintains along its path, which is called α before by convention. Hence every flow tries to put 5.5 packets along its path as we set $\alpha = 50$.

Experiment 3: varying K_2. We set $(\underline{b}_1, \overline{b}_1, K_1)$ to be (0, 1000, 10000) at link 1 and link 3. Set $(\underline{b}_2, \overline{b}_2)$ to be (100, 1500) at link 2, and vary the slope K_2 at link 2 from 10 to 600. Figure 9 shows the aggregate throughput of all Reno flows and the link utilization at link 1 for different values of K_2. Theoretical predictions are calculated by solving equations (1)–(2), (4), (7), (10), and the price mapping function 16 for RED.[1] As can be seen, the prediction matches the measured curve very well.



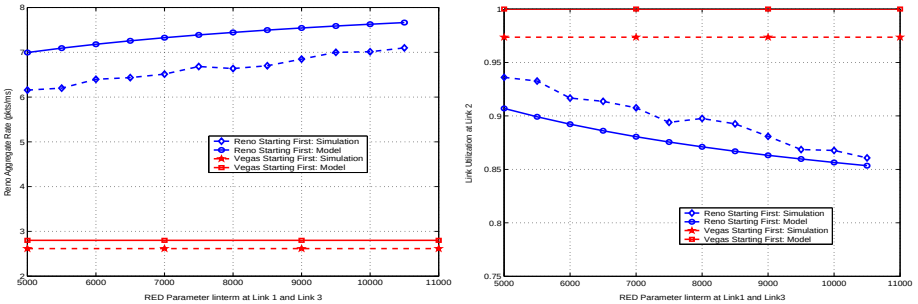
(a) Aggregate Reno throughput (b) Link utilization at link 1

Figure 9: Experiment 3: Aggregate Reno throughput and link utilization at link 1.

From Figure 9, the aggregate throughput and utilization at link 1 are independent of K_2 if Reno flows start first. This is because link 2 is not saturated in this scenario, as explained earlier, and hence varying its parameter does not affect the equilibrium. When Vegas flows start first, on the other hand, link 2 is the bottleneck link, and hence as K_2 increases, Reno achieves more and more bandwidth since the mapping function penalizes Reno less and less.

As K_2 increases, one may expect that the Reno throughput curve in Figure 9 that correspond to Vegas starting first will converge to the same value for the case when Reno starts first. It is not possible to exhibit this beyond $K_2 = 600$ at link 2. As shown in Figure 9, the utilization at link 1 is more than 95% when $K_2 = 600$. Even though link 1 is not saturated yet, it is so close to being saturated that random fluctuations in the queue can readily shift the system from the current equilibrium where only link 2 is saturated to the other equilibrium where links 1 and 3 are saturated (while link 2 is not). See a clear demonstration of this phenomenon in Experiment 5.

Experiment 4: varying K_1. In this experiment, we fix $K_2 = 100$ at link 2 and vary K_1 at link 1 and link 3 simultaneously from 5,000 to 11,000. The results are summarized in Figure 10. When Vegas flows



(a) Aggregate Reno throughput (b) Link utilization at link 2

Figure 10: Experiment 4: Aggregate Reno throughput and link utilization at link 2.

[1] For a more accurate prediction, T_i in Reno utility function should include equilibrium queueing delay.

start first, the bottleneck link is link 2 and therefore both the aggregate Reno throughput and the utilization at link 2 are independent of K_1. When Reno flows start first, on the other hand, links 1 and 3 become saturated and varying K_1 affect both the aggregate Reno throughput and link 2's utilization. The theoretical predictions track the measured data, but are generally larger than the data. The main reason is that Vegas flows overestimated base RTT when Reno flows start first and maintain a nonzero queue. Then Vegas flows become more aggressive and suppress Reno flows more than they should; see [Low et al., 2002] for more discussion on the effect of error in base RTT estimation.

As K_1 decreases at links 1 and 3, Reno flows see more losses and the system may shift to the other equilibrium where only link 2 is saturated. For instance, from Figure 10, the utilization at link 2 is close to 95% when $K_1 = 5000$.

Experiment 5: shifting equilibria. This experiment shows that the system can shift back and forth between the two equilibria when the utilization of the unsaturated link(s) is sufficiently close to 100% so that the system can readily jump between two disjoint active constraint sets due to random fluctuation. The slopes $K_1 = 3500$ at link 1 and link 3 and $K_2 = 500$ at link 2. The simulation duration is 1000 sec. The queues at link 1 and link 2 are shown in Figure 11. This result unambiguously exhibits that there are two equilibria and they are both achieved.

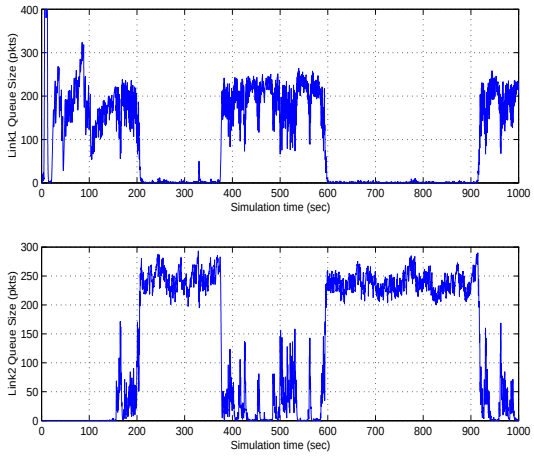


Figure 11: Experiment 5: queue sizes at link 1 and link 2. The system shifts between the two equilibria with disjoint active constraint sets.

6.2 Experiment 6: Intra and inter-protocol fairness

In this subsection, we present simulation results to illustrate Propositions 2 and 3. In all of the following simulations, we set RED parameters $(\underline{b}, \overline{b}, K)$ to be (20, 220, 20) at all links. The network topology is shown in Figure 12. The capacities for links 1, 2, and 3 are 3000 pkts/sec, 4000 pkts/sec, and 2000 pkts/sec respectively. All links have identical round-trip propagation delay of 60 ms. There are five flows on each path and they are labelled as a group in the figure. We vary the parameter value α_2 of Vegas flow3 and maintain the parameter value α_1 of Vegas flow1 and flow2 to be 1.5 times that of α_2.

The rate allocation among the three Vegas flows is shown in Figure 13 and agrees well with the prediction from Proposition 2. Similar results hold for bandwidth allocation among Reno flows and are omitted.

We now take link 2 as an example to show the bandwidth partition between Reno and Vegas and compare them with solutions from the model. The aggregate throughput of all the 15 Vegas flows and that of 5 Reno

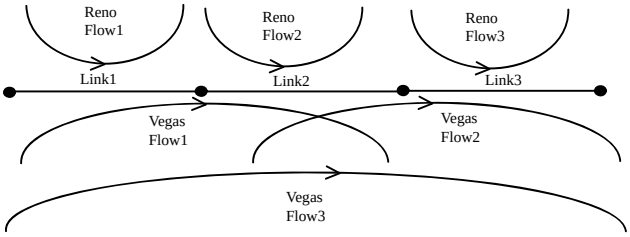


Figure 12: Experiment 6: network topology. There are 5 flows on each path.

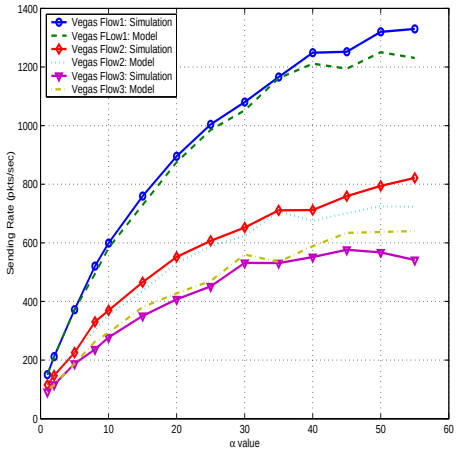


Figure 13: Experiment 6: bandwidth allocation among 3 Vegas flows (intra-protocol fairness)

flows at link 2 are shown in Figure 14 as α_2 is varied. The calculated values agree well with the measured values and therefore verify the model and Proposition 3.

7 Conclusion

In this paper we have demonstrated experimentally the existence of multiple equilibria in networks shared by TCP Reno and FAST, as predicted theoretically in [Tang et al., 2005b]. It is worthwhile to note that exhibition of multiple equilibria in this paper requires deliberate choices of buffer sizes or RED parameters, corresponding to careful design of the price mapping function. In [Tang et al., 2005a], it is proved that if the price mapping functions do not differ too much, global uniqueness is guaranteed. For instance if all links have their RED slopes inversely proportional to their capacities, then equilibrium is unique. We have also shown that any target inter-protocol fairness between Reno and FAST can be achieved in principle by an appropriate choice of FAST parameter, though it is unclear how to compute this parameter value in practice. Within each protocol, the flows share bandwidth among themselves as if the flows of the other protocol are absent, i.e., intra-protocol fairness is unchanged by the presence of the other protocol except for a reduction in effective link capacities.

This preliminary work can be extended in several ways. First, it would be interesting to study the local dynamics of these equilibria. If an equilibrium is unstable, it cannot be reached in a real network. Second, we have exhibited example networks with unique or multiple equilibria. A necessary and sufficient condition for the uniqueness of equilibrium is still missing. Third, consider the aggregate utility of flows within each protocol. Can one equilibrium point dominate in that the aggregate utilities are higher for all protocols at that equilibrium than at any other equilibrium? Finally, can routers help the network reach a unique and

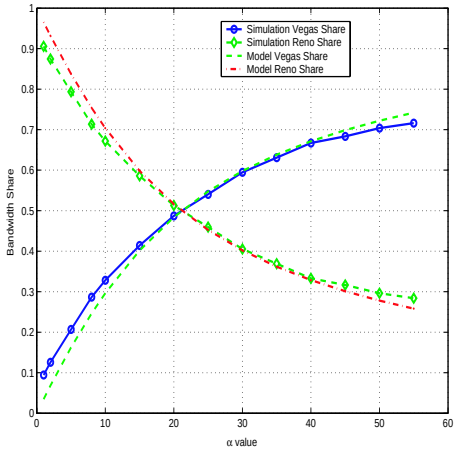


Figure 14: Experiment 6: bandwidth allocation between Reno and Vegas at link 2 (inter-protocol fairness)

stable equilibrium by modifying their price mapping functions in a distributed manner? If so, how to set up the right incentives for routers (or ISP) to do so?

8 Acknowledgments

We thank Cheng Jin for useful discussions, and Raj Jayaraman and George Lee for help on network setup. This is part of the Caltech FAST Project supported by NSF, Lee Center for Advanced Networking, ARO, AFOSR, Cisco, and a SISL Fellowship.

References

[Akella et al., 2003] Akella, A., Seshan, S., and Shaikh, A. (2003). An empirical evaluation of wide-area internet bottlenecks. In *Proceedings of ACM SIGMETRICS*, pages 316–317.

[Casetti et al., 2002] Casetti, C., Gerla, M., Mascolo, S., Sansadidi, M., and Wang, R. (2002). TCP Westwood: end to end congestion control for wired/wireless networks. *Wireless Networks Journal*, 8:467–479.

[Floyd, 2003] Floyd, S. (2003). Highspeed TCP for large congestion windows. RFC 3649, IETF Experimental. http://www.faqs .org/rfcs/rfc3649.html.

[Grieco and Mascolo, 2004] Grieco, L. A. and Mascolo, S. (2004). Performance evaluation and comparison of westwood+, new reno, and vegas tcp congestion control. *SIGCOMM Comput. Commun. Rev.*, 34(2):25–38.

[Hollot et al., 2002] Hollot, C., Misra, V., and Gong, W. (2002). Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*, 47(6).

[Jin et al., 2004] Jin, C., Wei, D. X., and Low, S. H. (2004). FAST TCP: motivation, architecture, algorithms, performance. In *Proceedings of IEEE Infocom*. http://netlab. caltech .edu.

[Kelly et al., 1998] Kelly, F. P., Maulloo, A., and Tan, D. (1998). Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3):237–252.

[Kelly, 2003] Kelly, T. (2003). Scalable TCP: improving performance in highspeed wide area networks. *ACM SIGCOMM Computer Communication Review.*, 33(2):83–91.

[Kunniyur and Srikant, 2003] Kunniyur, S. and Srikant, R. (2003). End-to-end congestion control: utility functions, random losses and ECN marks. *IEEE/ACM Transactions on Networking*, 11(5):689 – 702.

[Kurata et al., 2000] Kurata, K., Hasegawa, G., and Murata, M. (2000). Fairness comparisons between TCP Reno and TCP Vegas for future deployment of TCP Vegas. In *Proceedings of INET*.

[Low, 2003] Low, S. H. (2003). A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4):525–536. http://netlab.caltech.edu.

[Low and Lapsley, 1999] Low, S. H. and Lapsley, D. E. (1999). Optimization flow control I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874.

[Low et al., 2003] Low, S. H., Paganini, F., Wang, J., and Doyle, J. C. (2003). Linear stability of TCP/RED and a scalable control. *Computer Networks Journal*, 43(5):633–647. http://netlab.caltech.edu.

[Low et al., 2002] Low, S. H., Peterson, L., and Wang, L. (2002). Understanding Vegas: a duality model. *Journal of ACM*, 49(2):207–235. http://netlab.caltech.edu.

[Massoulie and Roberts, 2002] Massoulie, L. and Roberts, J. (2002). Bandwidth sharing: objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10(3):320–328.

[Mo et al., 1999] Mo, J., La, R., Anantharam, V., and Walrand, J. (1999). Analysis and comparison of TCP Reno and Vegas. In *Proceedings of IEEE Infocom*.

[Mo and Walrand, 2000] Mo, J. and Walrand, J. (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567.

[Rizzo,] Rizzo, L. IP dummynet. http://http://info.iet.unipi.it/~luigi/ip_dummynet/ .

[Shriram and Kaur, 2003] Shriram, A. and Kaur, J. (2003). Identifying bottleneck links using distributed end-to-end available bandwidth measurements. In *First ISMA Bandwidth Estimation Workshop*.

[Tang et al., 2005a] Tang, A., Wang, J., Low, S. H., and Chiang, M. (2005a). Equilibrium of heterogeneous congestion control protocols. Technical Report CaltechCSTR:2005.005, Caltech.

[Tang et al., 2005b] Tang, A., Wang, J., Low, S. H., and Chiang, M. (2005b). Network equilibrium of heterogeneous congestion control protocols. In *Proceedings of IEEE Infocom*, Miami, FL.

[Xu et al., 2004] Xu, L., Harfoush, K., and Rhee, I. (2004). Binary increase congestion control for fast long-distance networks. In *Proceedings of IEEE Infocom*.

[Yaiche et al., 2000] Yaiche, H., Mazumdar, R. R., and Rosenberg, C. (2000). A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networking*, 8(5):667–678.