

Equilibrium of Heterogeneous Congestion Control: Optimality and Stability

Ao Tang, *member, IEEE*, Xiaoliang Wei, *member, IEEE*, Steven H. Low, *Fellow, IEEE*, and Mung Chiang, *member, IEEE*

Abstract—When heterogeneous congestion control protocols that react to different pricing signals (They could be different types of signals such as packet loss, queueing delay etc. or different values of the same type of signal such as different ECN marking values based on the same actual link congestion level) share the same network, the current theory based on utility maximization fails to predict the network behavior. Unlike in a homogeneous network, the bandwidth allocation now depends on router parameters and flow arrival patterns. It can be non-unique, suboptimal and unstable. In [36], existence and uniqueness of equilibrium of heterogeneous protocols are investigated. This paper extends the study with two objectives: analyze the optimality and stability of such networks and design control schemes to improve them. First, we demonstrate the intricate behavior of a heterogeneous network through simulations and present a framework to help understand its equilibrium properties. Second, we propose a simple source-based algorithm to decouple bandwidth allocation from router parameters and flow arrival patterns by only updating a linear parameter in the sources’ algorithms on a slow timescale. It is used to steer a network to the unique optimal equilibrium. The scheme can be deployed incrementally as the existing protocol needs no change and only the new protocols need to adopt the slow timescale adaptation.

I. INTRODUCTION

Congestion control in TCP (Transmission Control Protocol), first introduced in [11], has enabled the explosive growth of the Internet. The *currently* predominant implementation, referred to as TCP Reno in this paper, uses packet loss as the congestion signal to dynamically adapt its transmission rate, or more precisely, its window size.¹ It has worked remarkably well in the past, but its limitations in wireless networks and in networks with large bandwidth-delay product have motivated various proposals some of which use different congestion signals. For example, in addition to loss based protocols such as HighSpeed TCP [9], STCP [19] and BIC TCP [42], schemes that use queueing delay include the earlier proposals CARD [13], DUAL [39] and Vegas [3], and the recent proposal FAST [40]. Schemes that use one-bit congestion signal include ECN [28], and those that use multi-bit feedback include XCP [15], MaxNet [41], and RCP [6]. Indeed, recently the Linux operating system already allows users to choose from a variety of congestion control algorithms since the kernel version 2.6.13, including TCP-Illinois [22] that uses both packet loss and delay as congestion signals. Recently, compound TCP [33] which also uses multiple congestion signals is deployed

in Windows Vista and Window Server 2008 TCP stack [25]. If explicit feedback is deployed, it will become possible to feed back different signals to different users to implement new applications and services. Note that in this case, the heterogeneous signals can all be loss-based – different users receiving different explicit values based on the actual link loss rate – or all delay-based, or a mix. Clearly, going forward, our network will become more heterogeneous in which protocols that react to *different* congestion signals interact. Yet, our understanding of such a heterogeneous network is rudimentary. For example, a *heterogeneous* network, as shown in an early companion paper [36], may have multiple equilibrium points, and they cannot all be stable unless the equilibrium is globally unique. This still leaves many other important properties open, such as optimality and stability of equilibrium.

In a homogeneous network, even though the sources may control their rates using different algorithms, they all adapt to the *same* congestion signal, e.g., all react to packet loss rate, as in the various variants of Reno and TFRC [8], or all to queueing delay, as in Vegas and FAST. For homogeneous networks, besides various detailed studies (see e.g., [27], [30]), there is already a well-developed theory, based on network utility maximization, e.g. [17], [21], [23], [24], [26], [43], that can help understand and engineer network behaviors. In particular, it is known that a homogeneous network of general topology always has a unique equilibrium (operating point). It maximizes aggregate utility, and the fairness associated with it can be well predicted and controlled. More importantly, the allocation depends only on the congestion control algorithms (equivalently, its underlying utility functions) but not on network parameters (e.g., buffer sizes) or flow arrival patterns, and hence can be designed through the choice of end-to-end TCP algorithms.

In contrast, we demonstrate in Section II of this paper that the bandwidth allocation among heterogeneous flows can depend on both network parameters and flow arrival patterns. It means that in general we cannot predict, nor control, the bandwidth allocation through the current design of end-to-end congestion control algorithms for heterogeneous networks. This implies, for example, the standard “TCP friendly” concept is not well defined anymore. To fully understand heterogeneous networks and develop ways to address these issues, we review our basic model in Section III. By identifying an optimization problem associated with any given equilibrium point, we discuss efficiency in Section IV-A and fairness in Section IV-B. Study of stability then follows in Section V. Finally, we propose a general scheme to steer

¹All our experiments and simulations use NewReno with SACK. These are enhanced versions of the original Tahoe and Reno, but we will refer them generically as TCP Reno.

an arbitrary heterogeneous network to the unique equilibrium that maximizes the standard weighted aggregate utility by updating a linear scaler in the sources' algorithms on a slow timescale (Section VI). The scheme requires only local end-to-end information but *does assume all flows have access to a common price*, which is generally true in practice since the common price can be what the incumbent dominate protocol uses. It can be deployed incrementally as *the existing protocol needs no change and only the new protocols need to adopt the slow timescale adaption*. Packet-level (ns-2) simulation results using TCP Reno and FAST are presented in Section VII and Linux experiments on a realistic testbed are reported in Appendix IX-C to further discuss some issues that are ignored in the mathematical model. We conclude in Section VIII.

We summarize here the main results that we have derived about heterogeneous congestion control in [36] and this paper:

- Existence of equilibrium: Theorem 2 in [36];
- Uniqueness of equilibrium.
 - Local uniqueness: Theorem 3 in [36];
 - Global uniqueness: Theorems 7 and 12 in [36].
- Optimality of equilibrium
 - Efficiency: Theorems 1 and Corollary 3 in this paper;
 - Fairness: Theorems 4 and 5 in this paper.
- Stability of equilibrium:
 - Local stability: Theorem 6 in this paper;
 - Special results: Theorems 12 and 13 in this paper.
- Control of heterogeneous networks: Theorem 11, Algorithms 1 and 2 in this paper.

II. TWO MOTIVATING EXAMPLES

In this section, we describe two simulations to illustrate some peculiar throughput behavior in heterogenous networks. All simulations use TCP Reno, which uses packet loss as congestion signal, and FAST TCP, which uses queueing delay as congestion signal.

The first experiment (Example 1a) shows that when a Reno flow shares a single bottleneck link with a FAST flow, the relative bandwidth allocation depends critically on the link parameter (buffer size): the Reno flow achieves higher bandwidth than FAST when the buffer size is large and smaller bandwidth when it is small. This implies that one cannot control the fairness between Reno and FAST through just the design of end-to-end congestion control algorithms, since fairness is now linked to network parameters, unlike in the case of homogeneous networks.

The second experiment (Example 2a) shows that even on a (multi-link) network with fixed parameters, one cannot control the fairness between Reno and FAST because the relative allocation can change dramatically depending on which flow starts first!

A. Example 1a: dependence of bandwidth allocation on network buffer size

FAST [40] is a high speed TCP variant that uses delay as its main control signal. Periodically, a FAST flow adjusts its

congestion window W according to

$$W \leftarrow \frac{\text{baseRTT}}{RTT} W + \alpha \quad (1)$$

In equilibrium, each FAST flow i achieves a throughput $x_i^* = \alpha/q_i^*$, where q_i^* is the equilibrium queueing delay observed by flow i . Hence, α is the number of packets that each FAST flow maintains in the bottleneck links along its path.

In this example, one FAST flow and one Reno flow share a single bottleneck link with capacity of 8.3 pkts per ms (equivalent to 100Mbps with maximum packet size) and round trip propagation delay 50ms. The topology is shown in Figure 1. The FAST flow fixes its α parameter at 50 packets.

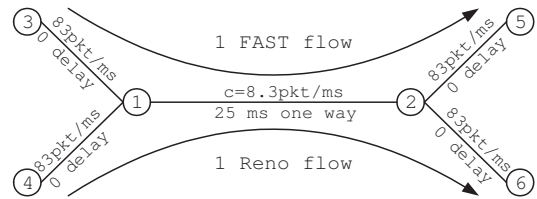


Fig. 1. Single link example.

In all of the ns-2 simulations in this paper, heavy-tail noise traffic is introduced at each link at an average rate of 10% of the link capacity.² Figure 2 shows the result with a bottleneck buffer size $B = 400$ packets. In this case, FAST gets an average of 2.1 pkts per ms while Reno gets 5.4 pkts per ms. Figure 3 shows the result with $B = 80$ packets. Since the bottleneck buffer size is smaller, the average queue is also smaller. Therefore FAST gets a higher throughput of 3.4 pkts per ms and Reno gets a much lower throughput of 0.6 pkt per ms. In this case, the loss rate is fairly high and the aggregate throughput is much lower (53.6 percent utilization) than the bottleneck capacity due to many timeout events.

In summary, contrary to the case of homogeneous network, bandwidth sharing between Reno and FAST depends on network parameters in a heterogeneous network.

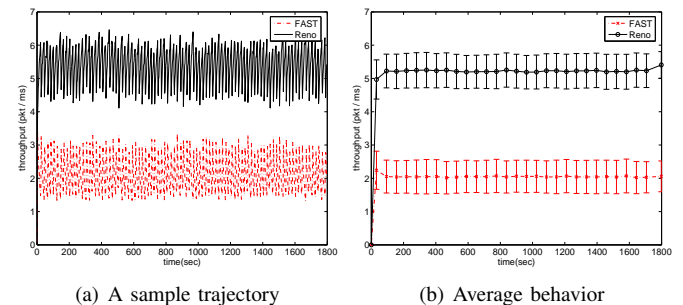


Fig. 2. FAST vs. Reno with a buffer size of 400 pkts.

²We usually present one sample figure on the left and the summary figure on the right. The sample figure shows the rate trajectory in one simulation run. The rate value is measured every 2 seconds. The summary figure presents the rate trajectory averaged over 20 simulation runs with different random seeds. Each point in the summary figure represents the average throughput over a period of one minute. The error bars are also shown in the summary figure.

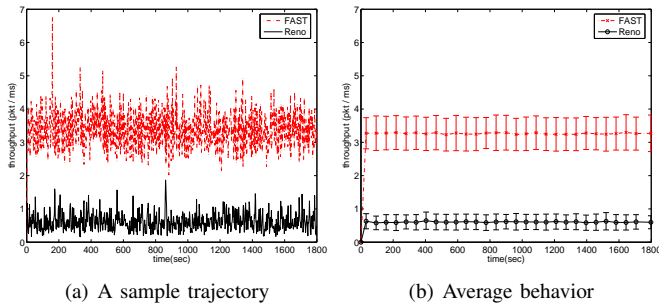


Fig. 3. FAST vs. Reno with a buffer size of 80 pkts.

B. Example 2a: dependence of bandwidth allocation on flow arrival pattern

The topology of this example is shown in Figure 4. We use RED algorithm [7] and packet marking instead of dropping. The marking probability $p(b)$ of RED is a function of queue length b :

$$p(b) = \begin{cases} 0 & b \leq \underline{b} \\ \frac{1}{K} \frac{b - \underline{b}}{\bar{b} - \underline{b}} & \underline{b} \leq b \leq \bar{b} \\ \frac{1}{K} & b \geq \bar{b} \end{cases} \quad (2)$$

where \underline{b} , \bar{b} and K are RED parameters. Links 1-2 and 3-4 are both configured with 9.1pkts per ms capacity (equivalent to 111 Mbps), 30 ms one-way propagation delay, and a buffer of 1500 packets. Their RED parameters are $(\underline{b}, \bar{b}, K) = (300, 1500, 10000)$. Link 2-3 has a capacity of 13.8 pkts per ms (166 Mbps) with 30 ms one-way propagation delay and a buffer size of 1500 packets. Its RED parameters are set to $(0, 1500, 10)$.

There are eight Reno flows on path 1-2-3-4, utilizing all three links, with one-way propagation delay of 90 ms. There are two FAST flows on each of paths 1-2-3 and 2-3-4. Both of them have one-way propagation delay of 60 ms. All FAST flows use a common $\alpha = 50$ packets.

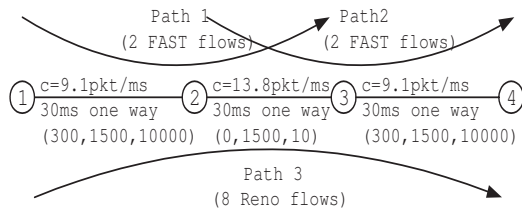


Fig. 4. Multiple equilibria scenario.

In our simulations, one set of flows (Reno or FAST) starts at time zero, and the other set of flows starts at the 100th second. We presents the throughput achieved by one of the FAST flows and one of the Reno flows. Each point in the summary figures represents the average rate over 5 minutes. Figure 5 shows the scenario in which FAST flows start first. Initially, FAST flows occupy most of the buffers in link 2-3. With the steep RED dropping slope in link 2-3, the Reno flows experience heavy loss and have very small throughput when they join the network. Figure 6 shows the scenario in which Reno flows start first. Initially, Reno flows maintain large queues in link

1-2 and link 3-4. FAST flows experience large queuing delays and are never able to fully utilize link 2-3.

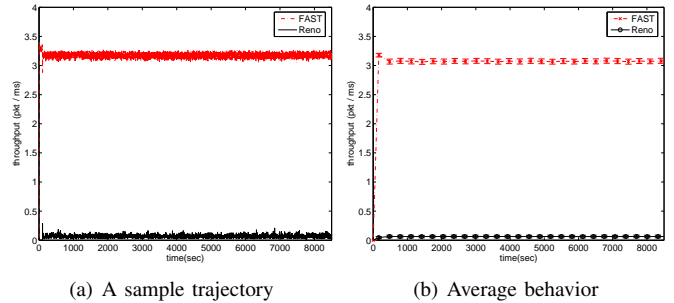


Fig. 5. Bandwidth shares of Reno and FAST when FAST starts first.

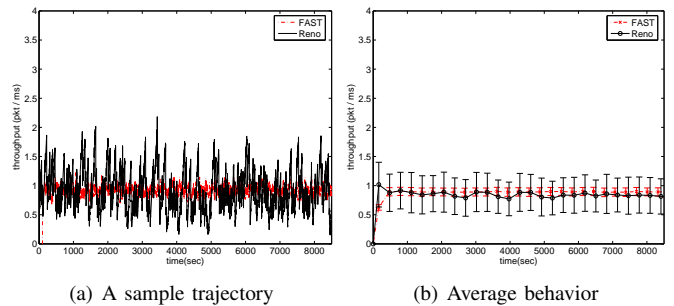


Fig. 6. Bandwidth shares of Reno and FAST when Reno starts first.

In short, bandwidth sharing in heterogeneous networks may depend on which type of TCP starts first and becomes unpredictable.

III. MODEL

A. Notations and Assumptions

Consider a network consisting of a set of L links, indexed by $l = 1, \dots, L$, with fixed finite capacities c_l . We sometimes abuse notation and use L to denote both the number of links and the set $L = \{1, \dots, L\}$ of links. Each link has a price p_l as its congestion measure. There are J different congestion control protocols indexed by superscript j , and N^j sources using protocol j , indexed by (j, i) where $j = 1, \dots, J$ and $i = 1, \dots, N^j$. The set of links used by source (j, i) is denoted by $L(j, i)$, and the total number of sources by $N := \sum_j N^j$.

The $L \times N^j$ routing matrix R^j for type j sources is defined by $R_{li}^j = 1$ if source (j, i) uses link l , and 0 otherwise. The overall routing matrix is denoted by

$$R = [R^1 \quad R^2 \quad \dots \quad R^J]$$

Even though different classes of sources react to different prices, e.g. Reno to packet loss probability and Vegas/FAST to queuing delay, the prices are related. We model this relationship through a price mapping function that maps a common “intrinsic” price (e.g. queue length) at a link to different prices (e.g. loss probability and queuing delay) observed by different sources. Formally, every link l has a price p_l . A type j source reacts to the “effective price” $m_l^j(p_l)$ in its path, where m_l^j is a price mapping function that can

depend on both the link and the protocol type. The exact form of m_l^j depends on the AQM algorithm used at the link; see (2) for links with RED. Let $m^j(p) = (m_l^j(p_l), l = 1, \dots, L)$ and $m(p) = (m^j(p_l), j = 1, \dots, J)$. The aggregate prices for source (j, i) is defined as

$$q_i^j = \sum_l R_{li}^j m_l^j(p_l) \quad (3)$$

Let $q^j = (q_i^j, i = 1, \dots, N^j)$ and $q = (q^j, j = 1, \dots, J)$ be vectors of aggregate prices. Then $q^j = (R^j)^T m^j(p)$ and $q = R^T m(p)$.

Let x^j be a vector with the rate x_i^j of source (j, i) as its i th entry, and x be the vector of x^j :

$$x = [(x^1)^T, (x^2)^T, \dots, (x^J)^T]^T$$

Source (j, i) has a utility function³ $U_i^j(x_i^j)$ that is strictly concave increasing in its rate x_i^j . Let $U = (U_i^j, i = 1, \dots, N^j, j = 1, \dots, J)$.

With the above notation, we refer to (c, m, R, U) as a *network*, where (in general) z denotes the (column) vector $z = (z_k, \forall k)$. The following basic assumptions are adopted, as in [36] that studies the existence and uniqueness of equilibrium for heterogeneous protocols.

- A1: Utility functions U_i^j are strictly concave increasing, and twice continuously differentiable in their domains. Price mapping functions m_l^j are continuously differentiable and strictly increasing with $m_l^j(0) = 0$.
- A2: For any $\epsilon > 0$, there exists a number p_{\max} such that if $p_l > p_{\max}$ for link l , then

$$x_i^j(p) < \epsilon \text{ for all } (j, i) \text{ with } R_{li}^j = 1$$

These are mild assumptions. Concavity and monotonicity of utility functions are often assumed in network pricing for elastic traffic. The assumption on m_l^j means that sources to observe the fluctuation as link congestion (p_l) rises and falls, as they must in order to control congestion. Assumption A2 says that when p_l is high enough, then every source going through link l has a rate less than ϵ , modeling the basic intuition in congestion control.

B. Network Model

As usual, we use $x^j(q^j) = (x_i^j(q_i^j), i = 1, \dots, N^j)$ and $x(q) = (x^j(q^j), j = 1, \dots, J)$ to denote the vector-valued functions composed of x_i^j . Since $q = R^T m(p)$, we often abuse notation and write $x_i^j(p), x^j(p), x(p)$. Define the aggregate source rates $y(p) = (y_l(p), l = 1, \dots, L)$ at links l as:

$$y^j(p) = R^j x^j(p), \quad y(p) = R x(p) \quad (4)$$

We consider the ‘‘dual algorithm’’ [17], [23]⁴ where sources select transmission rates that maximize their utility minus

bandwidth cost, and network links adjust bandwidth prices according to the utilization of the links:

$$x_i^j(q_i^j) = \left[(U_i^j)^{\prime-1} (q_i^j) \right]^+ \quad (5)$$

$$\dot{p}_l(t) = y_l(p(t)) - c_l =: f_l(p(t))$$

Remark: There are different fluid models in the literature. For example, the ‘‘primal algorithm’’ has dynamics at sources while the congestion signal at links depends on the instantaneous arrival rate or even both arrival rate and queue state. One is referred to e.g., [5], [18], [21], [31] for related discussion and justification. The main issues of heterogeneous congestion control (multiple equilibria, optimality loss and asymmetric Jacobian which may lead to instability) remain the same for both the primal and dual models. In other words, the difficulty due to heterogeneity is the same for various dynamical models. For example, if there are two marking functions $p_l^1(t) = g_l^1(y_l(t))$ and $p_l^2(t) = g_l^2(y_l(t))$ at the same link l as in the primal model, then these functions g serve the role of m functions defined above and aggregate rate y_l becomes the intrinsic measure of congestion as p_l defined before. The results and techniques developed in this paper should be useful for analyzing other models.

Under the assumptions in this paper, $(U_i^j)^{\prime-1} (q_i^j) > 0$ for all the prices p that we consider, and hence we can ignore the projection $[\cdot]^+$ and assume, without loss of generality, that

$$x_i^j(q_i^j) = (U_i^j)^{\prime-1} (q_i^j) \quad (6)$$

(6) is nothing but the ‘‘response function’’ of TCP which determines source rate based on its observed end-to-end congestion signal.

In equilibrium, the aggregate rate at each link is no more than the link capacity, and they are equal if the link price is strictly positive. Formally, we call p an *equilibrium price* (or a *network equilibrium* or an *equilibrium*) if it satisfies (from (3), (6), (4))

$$P(y(p) - c) = 0, \quad y(p) \leq c, \quad p \geq 0 \quad (7)$$

where $P := \text{diag}(p_l)$ is a diagonal matrix.

When all sources react to the same price, then the equilibrium described by (3), (4), (6) and (7) is the unique solution of the following utility maximization problem defined in [17] and its Lagrange dual [23]:

$$\max_{x \geq 0} \sum_i U_i(x_i) \quad (8)$$

$$\text{subject to } R x \leq c \quad (9)$$

where we have omitted the superscript $j = 1$. The strict concavity of U_i guarantees the existence and uniqueness of the optimal solution of (8)–(9) as well as the global convergence of the dual algorithm.

For heterogeneous case, the utility maximization problem no longer underlies the equilibrium described by (3), (4), (6) and (7). The current theory cannot be directly applied and substantial difficulties had to be overcome when exploring even some basic questions such as existence and uniqueness of equilibrium [36].

³Most TCP variants proposed or deployed can be shown to implicitly maximize some strictly concave increasing utility functions [24]. Here we take this reverse-engineering view and use utility function to represent the exact form of congestion protocol.

⁴Delay is omitted for simplicity.

IV. OPTIMALITY

As we have shown in [36], for heterogeneous congestion control networks, equilibrium cannot be characterized by (8)–(9) anymore. In this section, we further investigate the deviation of optimality in terms of both efficiency and fairness. This analysis provides insights on networks with heterogeneous congestion signals, for example, how to define inter-protocol fairness. It also motivates the algorithm design in section VI.

A. Efficiency

We first make the following key observation, which motivates other results on optimality and algorithm development.

Theorem 1. *Given an equilibrium p^* , there exists a positive vector $\gamma(p)$, such that the equilibrium rate vector $x^*(p)$ is the unique solution of following problem:*

$$\max_{x \geq 0} \sum_{i,j} \gamma_i^j U_i^j(x_i^j) \quad (10)$$

$$\text{subject to} \quad Rx \leq c \quad (11)$$

Proof. The KKT (Karush-Kuhn-Tucker) optimality conditions for (10)-(11) are:

$$\gamma_i^j \left(U_i^j \right)' (x_i^j) = \sum_l R_{il}^j p_l \quad \text{for all } (i, j) \quad (12)$$

$$p^T (Rx - c) = 0 \quad (13)$$

$$Rx - c \leq 0 \quad (14)$$

where the (x, p) are the primal-dual variables. We now claim these conditions are satisfied with equilibrium rates and prices (x^*, p^*) by choosing

$$\gamma_i^j = \frac{\sum_l R_{il}^j p_l^*}{\sum_l R_{il}^j m_l^j(p_l^*)} \quad (15)$$

To see this, note (13) and (14) are conditions for equilibrium. After substituting (15) into (12), we have

$$\left(U_i^j \right)' (x_i^{j*}) = \sum_l R_{il}^j m_l^j(p_l^*) \quad (16)$$

That is consistent with equations (3) and (6) that are used to define equilibrium. \square

Unlike the homogenous case where the equilibrium maximizes aggregate utility $\sum_i U_i(x_i)$, in the heterogeneous case, an equilibrium $x(p^*)$ maximizes a weighted aggregate utility $\sum_{i,j} \gamma_i^j U_i(x_i^j)$, where the weight depends on the equilibrium itself. Theorem 1 characterizes this underlying convex optimization problem that an equilibrium solves. It further motivates the algorithm in section VI. Since this optimization problem itself depends on the equilibrium, it cannot be used to find equilibrium directly, nor does it guarantee existence and uniqueness properties as in the single-protocol case [36].

As stated by the celebrated first fundamental theorem of welfare economics, assuming a homogeneous price signal, any competitive equilibrium is Pareto efficient. As a direct corollary of Theorem 1, the same holds for networks with heterogeneous price signals.

Corollary 2. *All equilibrium points are Pareto efficient.*

Pareto efficiency can be viewed as a necessary requirement for an efficient allocation. An equilibrium is *optimal* if it is Pareto efficient and maximizes (possibly weighted) aggregate utility. As shown in (8)-(9), for the homogeneous case, the equilibrium is indeed optimal. For the heterogeneous case, Theorem 1 implies a bound on the loss in optimality, as the following corollary states.

Corollary 3. *Assume all utility functions are nonnegative, i.e., $U(x) \geq 0$. Suppose the optimal aggregate utility is U^* and \hat{U} is the achieved aggregate utility at an equilibrium (\hat{x}) of a network with heterogeneous protocols. Then*

$$\frac{\hat{U}}{U^*} \geq \frac{\gamma_{\min}}{\gamma_{\max}} \quad (17)$$

where γ_{\min} and γ_{\max} are any lower and upper bounds of γ_i^j ⁵, i.e., $\gamma_{\min} \leq \gamma_i^j \leq \gamma_{\max}$.

Proof. Assume \hat{x} is one of the solutions of (10)-(11), then

$$\max_x \sum_{i,j} \gamma_i^j U_i^j(x_i^j) = \sum_{i,j} \gamma_i^j U_i^j(\hat{x}_i^j) \leq \gamma_{\max} \hat{U} \quad (18)$$

On the other hand,

$$\max_x \sum_{i,j} \gamma_i^j U_i^j(x_i^j) \geq \gamma_{\min} \max_x \sum_{i,j} U_i^j(x_i^j) = \gamma_{\min} U^* \quad (19)$$

Combining the two equalities above, we get $\frac{\hat{U}}{U^*} \geq \frac{\gamma_{\min}}{\gamma_{\max}}$ \square

It has been well known that price can serve as the “invisible hand” to coordinate competing users and realize optimal resource allocation. That however requires two basic assumptions. The first assumption is that users are all price takers. If instead they are noncooperative game players, there will be efficiency loss. Such “price of anarchy” was recently bounded from above for both routing [29] and congestion control [14]. The second assumption is the homogeneity of price, which does not hold in networks with more than one type of congestion control signals. Our result above attempts to quantify the “price of heterogeneity” in congestion control.

B. Fairness

In this subsection, we study fairness in networks shared by heterogeneous congestion control protocols. Two questions we address are: how the flows within each protocol share among themselves (intra-protocol fairness) and how these protocols share bandwidth in equilibrium (inter-protocol fairness). The results here generalize the corresponding theorems in [35].

1) *Intra-protocol fairness:* As indicated by (8)–(9), when a network is shared only by flows using the same congestion signal, the utility functions describe how the flows share bandwidth among themselves. When flows using different congestion signals share the same network, this feature is still preserved “locally” within each protocol.

⁵Both γ_{\min} and γ_{\max} can be bounded using inv_l^j . For example, for a network with both loss based and delay based protocols and assuming RED is used, the slopes of RED at different links can be used to compute γ_{\min} and γ_{\max} .

Theorem 4. Given an equilibrium (\hat{x}, \hat{p}) , let $\hat{c}^j := R^j \hat{x}^j$ be the total bandwidth consumed by flows using protocol j at each link. The corresponding flow rates \hat{x}^j are the unique solution of:

$$\max_{x^j \geq 0} \sum_{i=1}^{N^j} U_i^j(x_i^j) \quad \text{subject to} \quad R^j x^j \leq \hat{c}^j \quad (20)$$

Proof: Since $(\hat{x}^j, \hat{p}^j) \geq 0$ is an equilibrium, from (3) to (7), we have

$$(U_i^j)'(\hat{x}_i^j) = \sum_l R_{li}^j \hat{p}_l^j \quad \text{for } i = 1, \dots, N^j$$

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

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

forms the necessary and sufficient condition for \hat{x}^j and \hat{p}^j to be optimal for (20) and its dual respectively. \square

Note that in Theorem 4, the ‘‘effective capacities’’ \hat{c}^j are not preassigned. They are the outcome of competition among flows using different congestion prices and are related to inter-protocol fairness, which we now discuss.

2) *Inter-protocol fairness:* Even though flows using different congestion signals individually solve a utility maximization problem to determine their intra-protocol fairness, they in general do not jointly solve any predefined convex utility maximization problem. Here we provide a feasibility result, which says any reasonable inter-protocol fairness is achievable by linearly scaling congestion control algorithms.

Assume flow (j, i) has a parameter μ_i^j with which it chooses its rate in the following way:

$$x_i^j(q_i^j) = (U_i^j)'^{-1} \left(\frac{1}{\mu_i^j} q_i^j \right) \quad (21)$$

Our main result here says that for a network with J protocols, given any desirable bandwidth allocation across protocols, there exists a μ vector such that one of the resulting equilibria achieves the given bandwidth partition. Before stating the theorem, we first characterize the feasible set of predefined bandwidth allocation.

Assume that except for $j = J$, flow (j, i) has parameter μ_i^j . Or equivalently, we can define $\mu_i^J = 1$. The equilibrium rates x^j clearly depend on parameter μ . For $j = 1, 2, \dots, J-1$, let $\bar{x}^j(\mu)$ be the unique rate vector of flows using protocol j if there were no other protocols in the network, i.e., $\bar{x}^j(\mu)$ solves the following problem.

$$\max_{x^j \geq 0} \sum_{i=1}^{N^j} \mu_i^j U_i^j(x_i^j) \quad \text{subject to} \quad R^j x^j \leq c$$

Let $\underline{x}^j(\mu)$ be the unique rates of type j flows if network capacity were $(c - \sum_{k \neq j} R^k \bar{x}^k)^+$ and no other protocols are in the network, i.e., $\underline{x}^j(\mu)$ solves the following problem.

$$\max_{x^j \geq 0} \sum_{i=1}^{N^j} \mu_i^j U_i^j(x_i^j) \quad \text{subject to} \quad R^j x^j \leq (c - \sum_{k \neq j} R^k \bar{x}^k)^+$$

Let $X := \{ x | \underline{x}^j(\mu) \leq x^j \leq \bar{x}^j(\mu), \mu \geq 0, R x \leq c \}$. X includes all possible rates of flows using protocol j if they were given strict priority over other flows or if others were given strict priority over them, and all rates in between. In this sense X contains the entire spectrum of inter-protocol fairness among different protocols. The next result says that every point in this spectrum is achievable by an appropriate choice of parameter μ .

Let $S(\mu)$ denote the set of equilibrium rates of flows when the protocol parameter is μ . Clearly, equilibrium is characterized by (3), (4), (7) and (21).

Theorem 5. For every link l , assume there is at least one type J flow that only uses that link. Given any $x \in X$, there exists an $\mu \geq 0$ such that $x \in S(\mu)$.

Proof: Given any $x \in X$, the capacity for all type J flows is $c - \sum_{k \neq J} R^k x^k$. Since $R x \leq c$ (for all coordinates), we have $c - \sum_{k \neq J} R^k x^k \geq R^J x^J$, which is greater than or equal to 0. Hence the following utility maximization problem solved by flows of type J is feasible:

$$\begin{aligned} \max_{x^J \geq 0} \quad & \sum_i U_i^J(x_i^J) \\ \text{subject to} \quad & R^J x^J \leq c - \sum_{k \neq J} R^k x^k \end{aligned}$$

Let p^J be the associated Lagrange multiplier vector. By the assumption that every link has at least one single-link type J flow, we know $p_l^J > 0$ for all l . Choose $\mu_i^j = \frac{\sum_l R_{li}^J m_l^j ((m^J)_l^{-1} (p_l^J))}{(U_i^j)'(x_i^j)}$. It can be checked that all equations that characterize an equilibrium (3), (4), (7) and (21) are satisfied. \square

In general, one can view Theorem 1 as defining fairness of flows using heterogeneous protocols and can conclude that price mapping functions (router parameters) affect fairness (supported by Example 1a). Clearly, if one can choose price mapping functions, one can achieve any predefined fairness. More interestingly, Theorem 5 implies that given any reasonable fairness among flows using different congestion signals, in terms of a desirable rate allocation x , there exists a protocol parameter vector μ that can achieve it without changing parameters inside the network. In section VI, we will discuss distributed algorithms to compute a particular μ , which will result in the optimal bandwidth allocation.

V. STABILITY

For general dynamical systems, a globally unique equilibrium point may not even be locally stable [16], [32]. In this section, we focus on the stability of heterogeneous congestion control protocols, which dictates whether an equilibrium can manifest itself experimentally or not [35]. For general networks, it is shown that once the degree of heterogeneity is properly bounded, the equilibrium is not only unique as shown in [36] but also locally stable. Stronger results for some special cases can be found in the Appendix IX-A.

We now state the general result on local stability. It essentially says that if the similarity condition on price mapping

functions that guarantees uniqueness [36] is satisfied, the unique equilibrium is also locally stable. In particular, if for any l all m_l^j are the same, then (22) is satisfied and the equilibrium is locally stable. This certainly agrees with our knowledge on the homogeneous case.

We call a vector $\sigma = (\sigma_1, \dots, \sigma_L)$ a *permutation* if each σ_l is distinct and takes value in $\{1, \dots, L\}$. Treating σ as a mapping $\sigma : \{1, \dots, L\} \rightarrow \{1, \dots, L\}$, we let σ^{-1} denote its unique inverse permutation. For any vector $a \in \mathbb{R}^L$, $\sigma(a)$ denotes the permutation of a under σ , i.e., $[\sigma(a)]_l = a_{\sigma_l}$. If $a \in \{1, \dots, L\}^L$ is a permutation, then $\sigma(a)$ is also a permutation and we often write σa instead. Let $\mathbf{l} = (1, \dots, L)$ denote the identity permutation. Then $\sigma \mathbf{l} = \sigma$. Finally, denote dm_l^j/dp_l by \dot{m}_l^j .

Theorem 6. *If for any vector $\mathbf{j} \in \{1, \dots, J\}^L$ and any permutations $\sigma, \mathbf{k}, \mathbf{n}$ in $\{1, \dots, L\}^L$,*

$$\prod_{l=1}^L \dot{m}_l^{[\mathbf{k}(\mathbf{j})]_l} + \prod_{l=1}^L \dot{m}_l^{[\mathbf{n}(\mathbf{j})]_l} \geq \prod_{l=1}^L \dot{m}_l^{[\sigma(\mathbf{j})]_l} \quad (22)$$

then the equilibrium of a regular network is locally stable.

Proof. For a real matrix A , if all its principle minors are positive, A is called a P -matrix [34]. If $a_{ii} \geq 0$, $a_{ij} \leq 0$, then A is called an M -matrix. Clearly if a P -matrix is symmetric, then it is positive definite and hence stable. However, the Jacobian matrix in our problem is not symmetric when multiple protocols exist, which is the main difficulty in proving stability. Before getting into the main proof, we state three lemmas. One is referred to [1] for other related results.

Lemma 7. *If A is a P -matrix and also an M -matrix, then all its eigenvalues have positive real parts.*

Let e be the column vector $e = [1, 1, \dots, 1]^T$.

Lemma 8. *If A is an M -matrix and all its eigenvalues have positive real parts, then there is an $D = \text{diag}[d_1, \dots, d_n]$, $d_i > 0$ for all i , such that $D^{-1}ADe > 0$. In other words, A is strictly diagonally dominant.*

For a matrix A , we define its comparison matrix $M(A) = (m_{ij})$ by setting $m_{ii} = |a_{ii}|$, and $m_{ij} = -|a_{ij}|$ if $i \neq j$. Clearly $M(A)$ is an M -matrix. The following lemma points out a simple yet important fact that relates diagonal dominance property of A with positive diagonal entries and that of $M(A)$.

Lemma 9. *Suppose all diagonal entries of A are positive. If there is an $D = \text{diag}[d_1, \dots, d_n]$, $d_i > 0$ for all i , such that $D^{-1}M(A)De > 0$, then $D^{-1}ADe > 0$, i.e., A is also strictly diagonally dominant.*

We now state the proof of Theorem 6. We need to show all eigenvalues of $-\mathbf{J}$ have positive real parts, where \mathbf{J} is the Jacobian of equilibrium equations ($\mathbf{J} = \partial y / \partial p$) evaluated at equilibrium. It is enough to show $-\mathbf{J}$ is strictly diagonally dominant and by Lemma 9 we only need to show $M(-\mathbf{J})$ is strictly diagonally dominant since all diagonal entries of $-\mathbf{J}$ are positive (each link has at least one flow using it). Using Lemma 8, it suffices to show that $M(-\mathbf{J})$ is positive stable, which then can be reduced to check whether $M(-\mathbf{J})$ is a

P -matrix by Lemma 7. By similar arguments in [36], it is enough to show $\det(M(-\mathbf{J})) > 0$, which will be done in the remainder of the proof.

Following [36], let π denote an L -bit binary sequence that represents the path consisting of exactly those links k for which the k th entries of π are 1, i.e., $\pi_k = 1$. Let $\Pi(k, l) := \{\pi | \pi_k = \pi_l = 1\}$ be the set of paths that contain both links k and l . Let $I_\pi^j = \{i | R_{i_i}^j = 1 \text{ if and only if } \pi_l = 1\}$ be the set of type j sources on path π , possibly empty. Let

$$r_\pi^j = r_\pi^j(p) = \sum_{i \in I_\pi^j} \left(-\frac{\partial^2 U_i^j}{\partial (x_i^j)^2} \right)^{-1} \quad (23)$$

where r_π^j is zero if I_π^j is empty. Denote by $\mathbf{1}(a)$ the indicator function that is 1 if the assertion a is true and 0 otherwise. Define

$$\mu(\mathbf{j}) := \prod_{l=1}^L \dot{m}_l^{j_l} \quad (24)$$

$$\rho(\mathbf{j}, \pi) := \prod_{l=1}^L \dot{m}_l^{j_l} \pi_l \quad (25)$$

For any permutation \mathbf{k} , Define $L_{\mathbf{k}}^+ = \{l | k_l = l\}$ and $L_{\mathbf{k}}^- = \{l | k_l \neq l\}$. We then have

$$\det(M(-\mathbf{J})) = \sum_{\mathbf{j}} \sum_{\pi} G(\mathbf{j}, \pi) \rho(\mathbf{j}, \pi) \quad (26)$$

where the last summation in (26) is over the vector index $\pi = (\pi^1, \dots, \pi^L)$ that takes value in the set $\{\text{all } L\text{-bit binary sequences}\}^L$. $\mathbf{l} = (1, \dots, L)$ denotes the identity permutation, and " $\pi \in \Pi(\mathbf{k}, \mathbf{l})$ " is a shorthand for " $\pi^l \in \Pi(k_l, l), l = 1, \dots, L$ ". and

$$G(\mathbf{j}, \pi) := \sum_{\mathbf{k}} \mathbf{1}(\pi \in \Pi(\mathbf{k}, \mathbf{l})) \text{sgn} \mathbf{k} (-1)^{|L_{\mathbf{k}}^-|} \mu(\mathbf{j}) \quad (27)$$

Then let Θ_0 be the largest subset of the set of all possible (\mathbf{j}, π) 's that is *permutationally distinct*, i.e., no vector in Θ_0 is a permutation of another vector in Θ_0 . We then have

$$\det(M(-\mathbf{J}(p))) = \sum_{(\mathbf{j}, \pi) \in \Theta_0} H(\mathbf{j}, \pi) \rho(\mathbf{j}, \pi) \quad (28)$$

$$H(\mathbf{j}, \pi) = \sum_{\sigma \in \Sigma(\mathbf{j}, \pi)} \sum_{\mathbf{k}} \mathbf{1}(\sigma(\pi) \in \Pi(\mathbf{k}, \mathbf{l})) T \quad (29)$$

where

$$T = \text{sgn} \mathbf{k} (-1)^{|L_{\mathbf{k}}^-|} \mu(\sigma(\mathbf{j}))$$

and $\Sigma(\mathbf{j}, \pi)$ is the largest subset of the set of all permutations σ that generates distinct $\sigma(\mathbf{j}, \pi)$.

We now use (29) to derive a sufficient condition under which $H(\mathbf{j}, \pi)$ are nonnegative for all permutationally distinct (\mathbf{j}, π) . The main idea is to show that for every negative term in the summation in (29), either it can be exactly cancelled by a positive term, or we can find two positive terms whose sum has a larger or equal magnitude under the given condition. Theorem 6 is then directly implied by the following Lemma, whose proof is provided in the Appendix IX-B.

Lemma 10. Suppose for any $j \in \{1, \dots, J\}^L$ and permutations $\sigma, \mathbf{k}, \mathbf{n}$ in $\{1, \dots, L\}^L$, we have for a regular network

$$\mu(\mathbf{k}(j)) + \mu(\mathbf{n}(j)) \geq \mu(\sigma(j))$$

Then, for all $(j, \pi) \in \Theta_0$, $H(j, \pi) \geq 0$.

VI. SLOW TIMESCALE UPDATE

A. Motivation

As pointed out in Corollary 2, all equilibria are Pareto efficient. However, based on analysis in section IV, large efficiency loss may occur and no guarantee on fairness can be provided. This motivates us to turn from analysis to design, and develop a readily implementable control mechanism that “drives” any network with heterogeneous congestion control protocols to a target operating point with a fair and efficient bandwidth allocation. Our target equilibrium is the maximizer of some weighted aggregate utility. The first step is to set up the existence and uniqueness of such a solution.

Theorem 11. For any given network (c, m, R, U) , for any positive vector w , there exists a unique positive vector μ such that, if every source scales their own prices by μ_i^j , i.e.,

$$x_i^j = \left(U_i^j\right)^{\prime-1} \left(\frac{1}{\mu_i^j} \sum m_i^j(p_l)\right) \quad (30)$$

then, at equilibrium (x, p) , x solves

$$\max_{x \geq 0} \sum_{(i,j)} \frac{1}{w_i^j} U_i^j(x_i^j) \quad (31)$$

$$\text{subject to } Rx \leq c \quad (32)$$

Moreover,

$$\mu_i^j = \frac{1}{w_i^j} \frac{\sum_{l \in L(j,i)} m_l^j(p_l)}{\sum_{l \in L(j,i)} p_l}$$

Proof. We claim that the optimality conditions of (31) and (32) are the same as equations that characterize the equilibrium of the above system ((3), (30), (4) and (7)). Capacity constraints, nonnegativity, and complementary slackness are obviously the same. We only need to check the relation between rates and prices at equilibrium. Those are

$$\mu_i^j \left(U_i^j\right)'(x_i^j) = \sum_{l \in L(j,i)} m_l^j(p_l) \quad (33)$$

and

$$\mu_i^j = \frac{1}{w_i^j} \frac{\sum_{l \in L(j,i)} m_l^j(p_l)}{\sum_{l \in L(j,i)} p_l} \quad (34)$$

Combining them, we get

$$\frac{1}{w_i^j} \left(U_i^j\right)'(x_i^j) = \sum_{l \in L(j,i)} p_l \quad (35)$$

which is the relation between x and p specified by the optimality conditions of problem (31)-(32). On the other hand, given x and p that satisfy (35), one can always define μ by (34), and (33) will also be satisfied. \square

Algorithm 1 Two timescale control scheme

1) Every source chooses its rate by

$$x_i^j(t) = (U^j)^{-1} \left(\frac{q_i^j(t)}{\mu_i^j(t)} \right);$$

2) Every source updates its μ_i^j by

$$\mu_i^j(t+T) = \mu_i^j(t) + \kappa_i^j \left(\frac{\sum_{l \in L(j,i)} m_l^j(p_l(t+T))}{\sum_{l \in L(j,i)} p_l(t+T)} - \mu_i^j(t) \right)$$

where κ_i^j is the stepsize for flow (j, i) and T is large enough so that the fast timescale dynamics among x and p can reach steady state.

Parameter w enables us to control fairness and to achieve any desired fair bandwidth allocation. Moreover, Theorem 11 suggests Algorithm 1 as a two-timescale scheme to control the operating point of networks with heterogenous congestion control protocols. The essential idea in Algorithm 1 is that by reacting to the same price $(p_l(t))$ on slow timescale, it is guaranteed to reach the optimal equilibrium in the long run. Yet the algorithm allows sources to react to their own effective prices $m_i^j(p_l(t))$ on fast timescale. This flexibility on timescales is important in practice when, for example, the link prices p_l are loss probability that are hard to reliably estimate on the fast timescale. The slow timescale algorithm only updates a linear scaler (μ_i^j) , which is readily implementable, e.g., this corresponds to updating a parameter α in FAST; see Section VII. In general, one can always choose $m_l^j(p_l) = p_l$ for a particular j , say $j = 1$. Then $\mu_i^1 = 1$. This is desirable for incremental deployment as only new protocols need to adapt while the current Reno ($j = 1$) does not.

B. Numerical Examples

Throughout this subsection, we provide some numerical results to further validate the effectiveness of Algorithm 1. For simplicity we choose $w = 1$, i.e., we attempt to maximize the aggregate utility.

Example 3: L=3 with multiple equilibria

We use the following example that has multiple equilibria [36]. The network is shown in Figure 7 with three unit-capacity links, $c_l = 1$. There are three different protocols with the corresponding routing matrices

$$R^1 = I, \quad R^2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}^T, \quad R^3 = (1, 1, 1)^T$$

The price mapping functions are linear: $m_l^j(p_l) = k_l^j p_l$ where

$$K^1 = I, \quad K^2 = \text{diag}(5, 1, 5), \quad K^3 = \text{diag}(1, 3, 1)$$

Utility functions of sources (j, i) are

$$U_i^j(x_i^j, \alpha_i^j) = \begin{cases} \beta_i^j (x_i^j)^{1-\alpha_i^j} / (1-\alpha_i^j) & \text{if } \alpha_i^j \neq 1 \\ \beta_i^j \log x_i^j & \text{if } \alpha_i^j = 1 \end{cases}$$

with appropriately chosen positive constants α_i^j and β_i^j [36]. These utility functions can be viewed as a weighted version of the α -fairness utility functions proposed in [26]. Parameters μ_i^j are updated every 20 time units. We show that starting from

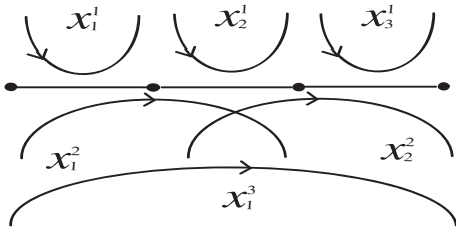


Fig. 7. A three-link network with three equilibria

different initial conditions, although the system reaches different equilibria after the first iteration, it nevertheless finally reaches the unique optimal equilibrium with $p_1^* = 0.222$.

Case 1: We start with initial point $p_1(0) = p_2(0) = p_3(0) = 0.3$. After the first iteration, the network goes to an equilibrium ($p_1 = p_3 = 0.165$, $p_2 = 0.170$). Price $p_1(t)$ with different stepsize κ_i^j is shown in Figures 8.

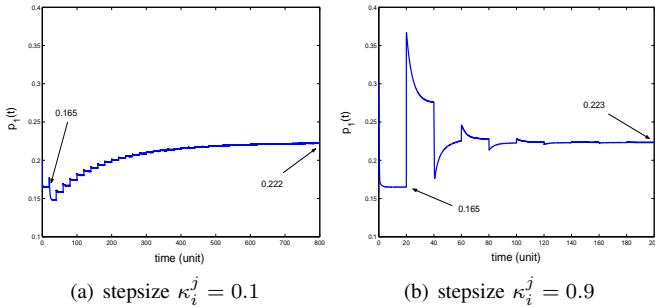


Fig. 8. Case 1: $p_1(t)$ with different κ_i^j

Case 2: We choose another initial point $p_1(0) = p_3(0) = 0.1$, $p_2(0) = 0.3$. As shown in Figure 9. After the first iteration, the system reaches another equilibrium, $p_1 = p_3 = 0.135$ and $p_2 = 0.230$. However finally, the system still reaches the same steady state as in Figure 8.

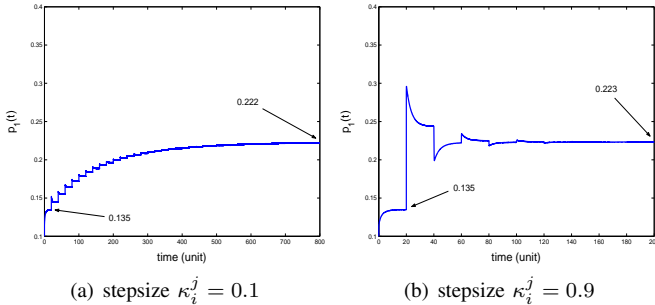


Fig. 9. Case 2: $p_1(t)$ with different κ_i^j

Example 4: L=5 with asynchronous update

In this example, the network has five links and 15 flows. Algorithm 1 is tested in an asynchronous environment. We assume that every five time units, flows can update their μ_i^j and they do so with certain probability. Hence every five time units, only a portion of flows update their μ_i^j . We set link capacities uniformly between 1 to 10, price mapping functions are $m^1(p) = p$ and $m^2(p) = p^\alpha$, where α is chosen between

0.5 to 5 with uniform distribution. Flows 1 to 5 use links 1 to 5 correspondingly while a random routing matrix with entries 0 or 1 with equal probability is used to define routes for other flows. Finally each flow chooses to use price 1 or 2 with equal probability.

All of the 1000 trials converge to the right operating point. Some typical convergence patterns are shown in Figure 10 where the five curves correspond to the p value of the five links. It shows clearly that although asynchronism causes longer convergence time, the system still converges to the target equilibrium.

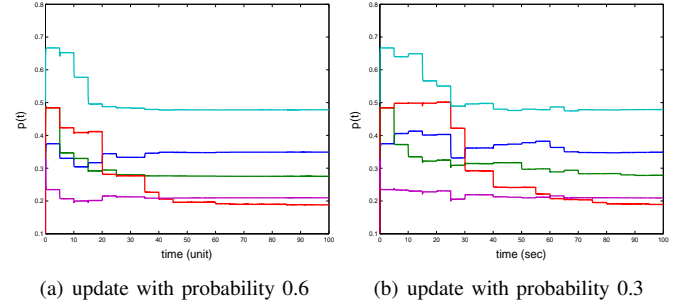


Fig. 10. $p(t)$ with different probability of updating

VII. SIMULATION RESULT: RENO AND FAST

In this section, we apply Algorithm 1 to the case of Reno and FAST to resolve the issues illustrated in Section II. It demonstrates how the algorithm can be deployed incrementally where the existing protocol (Reno in this case) needs no change and only the new protocols (FAST in this case) need to adopt slow timescale adaptation for the whole network to converge to the unique equilibrium that maximizes (weighted) aggregate utility. Experiments in this section were conducted in ns-2; Section IX-C presents results in a real testbed, Caltech's WAN-in-Lab.

We take Reno's loss probability as the link price, i.e., $m_i^1(p_i) = p_i$ for Reno. Algorithm 1 then reduces to an α adaptation scheme for FAST that uses only end-to-end local information that is available to each flow. This algorithm, displayed as Algorithm 2, tunes the value of α according to the signals of queue delay and loss on a large timescale. The basic idea is that FAST should adjust its aggressiveness (parameter α) to the proper level by looking at the ratio of end-to-end queuing delay and end-to-end loss. Therefore FAST also reacts to loss in a slow timescale. We apply Algorithm 2 to the examples illustrated in Section II.

A. Example 1b: independence of bandwidth allocation on buffer size

We repeat the simulations in Example 1a with Algorithm 2, with w set to 125s⁶. Figure 11, Figure 12, should be compared with Figure 2, Figure 3, respectively.

With Algorithm 2, FAST achieves 3.4 pkts per ms with buffer size of 400 and 3.2 pkts per ms with buffer size of

⁶The parameter w determines the equilibrium bandwidth share. Formally, it is stated in (31)-(32). Here w is chosen so that Reno and FAST get equal rates.

Algorithm 2 α adaptation algorithm

1) Every α update interval (2 minutes by default), calculate:

$$\alpha^* = \frac{q}{lw} \alpha_0$$

α_0 is the initial α value; q and l are average queuing delay and average packet loss rate over the interval; w is a parameter with the same unit of q/l . It determines the relative fairness between delay-based and loss-based protocols. Then

$$\alpha = \begin{cases} \min \{(1 + \delta)\alpha, \alpha^*\} & \text{if } \alpha < \alpha^* \\ \max \{(1 - \delta)\alpha, \alpha^*\} & \text{if } \alpha > \alpha^* \end{cases}$$

where δ determines the responsiveness and is 0.1 by default.

2) Every window update interval (20ms by default), run FAST algorithm (1).

80, while Reno gets 4.2 pkts per ms and 4.1 pkts per ms, respectively. The fairness is greatly improved and essentially independent of buffer size now. This is summarized in table I by listing the ratio of Reno's bandwidth to FAST's. We also note that the utilization of the link for $B = 80$ increases significantly from 53.6% to 97.7%. This point will be further discussed in Example 5 in section IX-C.

	B=400	B=80
Without Algorithm 2	5.4/2.1=2.6	0.6/3.4=0.18
With Algorithm 2	4.2/3.1=1.4	4.1/3.2=1.3

TABLE I
RATIO OF RENO'S RATE AND FAST'S RATE

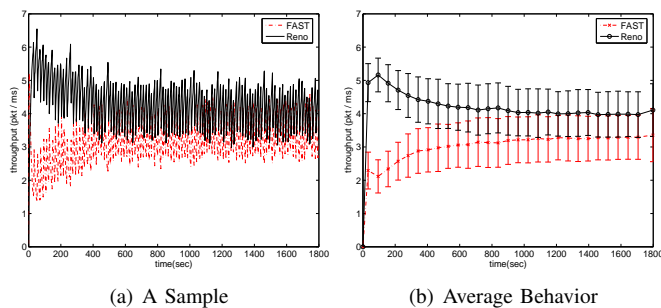


Fig. 11. FAST vs. Reno, with buffer size of 400 pkts.

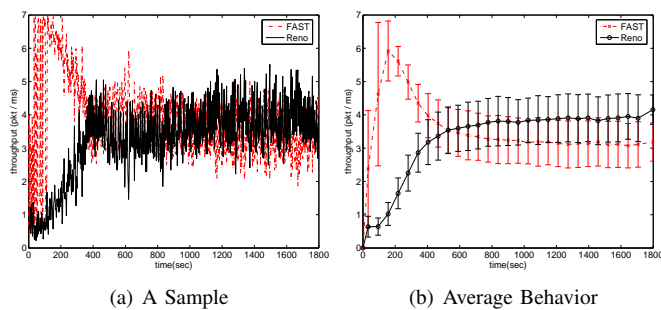


Fig. 12. FAST vs. Reno, with buffer size of 80 pkts.

The trajectories of α with different buffer sizes are presented in Figure 13. It is clear that although FAST starts with $\alpha = 50$ in both cases, it finally ends up with a much larger α in the scenario where $B = 400$, as it experiences much higher equilibrium queuing delay with the large buffer.

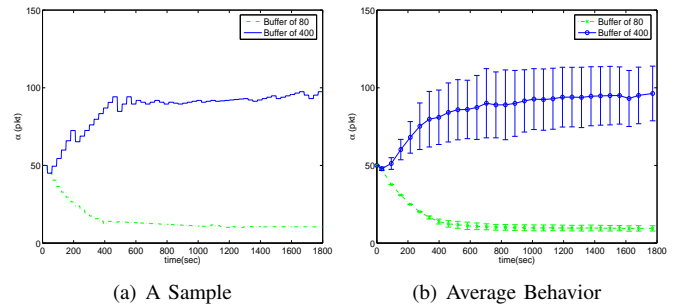


Fig. 13. α trajectory of example 1b

B. Example 2b: independence of bandwidth allocation on flow arrival pattern

We repeat the simulations in Example 2a with Algorithm 2, with w set to 1,820s. Figure 14 and Figure 15 show the effect of α adaptation in the multiple-bottleneck case and should be compared with Figure 5 and Figure 6 respectively. Theorem 11 guarantees a unique equilibrium when we adapt α according to Algorithm 2. In this particular case, this single equilibrium is around the point where each Reno flow gets a throughput of 0.6 pkts per ms and each FAST flow gets 1.5 pkts per ms. At this single equilibrium, link 1 and link 3 are the bottleneck links. In Figure 14, FAST flows start at time zero and link 2 becomes the bottleneck. When Reno flows join at the 100th second, the ratio of queue delay and loss at link 2 is much higher than the target value. The FAST flows hence reduce their α values gradually and the set of bottleneck links switches from link 2 to links 1 and 3 around the 2000th second. After that, FAST flows and Reno flows converge to the unique equilibrium.

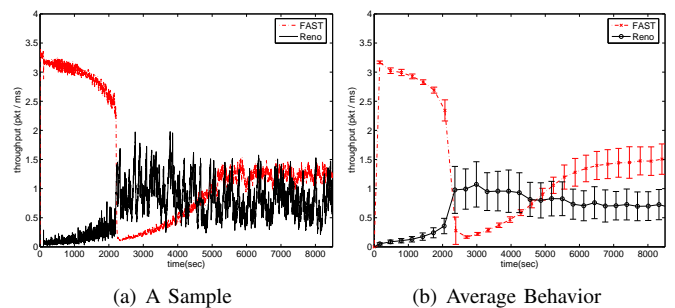


Fig. 14. FAST starts first.

The trajectory of α is presented in Figure 16. Although the α values converge to the same equilibrium value with different starting sequence, the trajectories are very different: When the Reno flows start first, the value of α gradually increases from the initial value of 50 to the equilibrium value of around 96. However, when FAST flows start first, the value of α first decreases, and then increases to the equilibrium value.

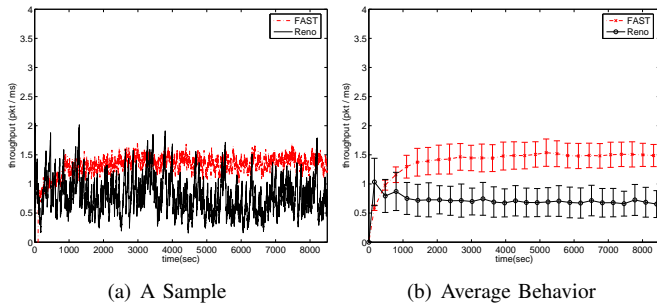


Fig. 15. Reno starts first.

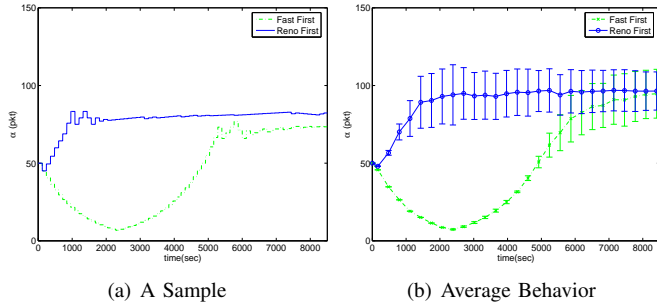


Fig. 16. α trajectory of example 2b

Queue trajectories of the links help better understand this process. Figure 17 presents the queue trajectories of the two cases. When the system converges, the α value is around 96 and the bottleneck links are link 1-2 and link 3-4. If Reno flows start first, the initial bottleneck set is the same as the bottleneck set in equilibrium. With the correct bottleneck set, the α adaptation algorithm adjusts the α value to reach a fair share defined by the w parameter as in Example 1. If FAST flows start first, the initial bottleneck is link 2-3, with which there is no (x, p) that solves the optimization problem defined in Theorem 1. Hence, the α adaptation algorithm keeps decreasing the value of α due to the small delay-to-loss ratio in link 2-3 until the bottleneck link set switches to be link 1-2 and link 3-4. The α adaptation algorithm then works as the scenario when Reno flows start first and finds the right equilibrium point.

We note that the bottleneck switching point has the α value with which the two stable equilibria are very close to each other. Without the α adaptation algorithm, the equilibrium bottleneck set can vary due to random noise, even with the

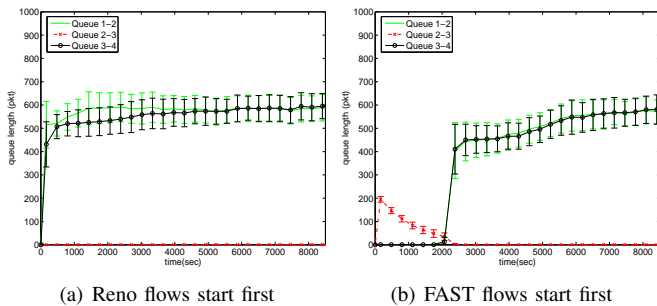


Fig. 17. Trajectories of queues of example 2b

same setup and same starting order. To intuitively illustrate this transition point, Figure 18 presents two individual results of the same scenario of Example 2a, with FAST flows starting first with fixed α value, and with different random seeds in the simulation. Although in both cases FAST flows start first, Reno flows may or may not take over link 1-2 and link 3-4, depending on the randomness of the noise traffic when Reno flows join.

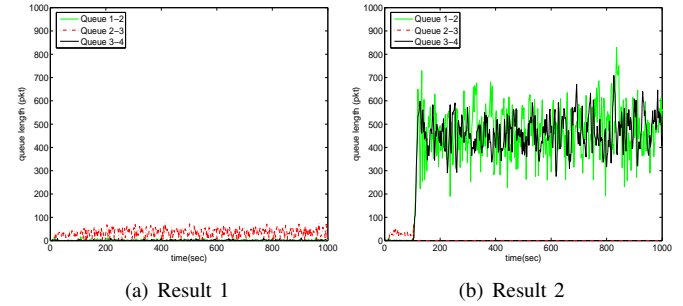


Fig. 18. Trajectories of queues of example 2a, with α fixed at 8.2 and using different random seeds (FAST flows start first)

VIII. CONCLUSION

Congestion control has been extensively studied for networks running a single protocol. However, when sources sharing the same network react to different congestion signals, the existing duality model no longer explains the behavior of bandwidth allocation. The existence and uniqueness properties of equilibrium in heterogeneous protocol case are examined in [36]. In this paper, we study optimality and stability properties. In particular, it is shown that equilibrium is still Pareto efficient, but there is efficiency loss. On fairness, intra-protocol fairness is still determined by utility maximization problem, while inter-protocol fairness is the part which we do not have control on. However, we can achieve any desired inter-protocol fairness by properly choosing protocol parameters. Motivated by the analytical results, we further propose a distributed scheme to steer the whole network to the unique optimal equilibrium. The scheme only needs to update a linear scaler in the source algorithm on a slow timescale. It can be deployed incrementally as the existing protocol needs no change and only the new protocols need to adapt on the slow timescale.

There are several interesting directions in this relatively open area. For example, more efforts are still needed to fully clarify the global dynamics of the two timescale system. The main technical difficulty here is that the fast timescale system may have multiple equilibria and therefore the usual two timescale argument (e.g. singular perturbation) is not applicable. Our current model assumes each protocol only reacts to one particular price on the fast timescale, even when they have access to multiple types of prices. It'd be interesting to generalize the analysis where a protocol can react to a combination of price types, as new protocols such as TCP Westwood [4], CTCP [33] and TCP-Illinois [22] do. Preliminary steps along this direction can be found in [37]. Finally, the current results should be extended from static to dynamic setting where flows come and go [2], [20].

Acknowledgments: The authors thank Dr. Cheng Jin of Caltech for help on some WAN-in-Lab experiments. We also would like to thank S. Simsek and A. Ozdaglar of M.I.T., D. Palomar of HKUST, and J. Lui of CUHK for useful discussions. The research is supported by NSF, ARO, Caltech Lee Center for Advanced Networking. We also thank the support from NSF CCF-0835706.

REFERENCES

[1] A. Berman and R. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.

[2] T. Bonald and L. Massoulié. Impact of fairness on Internet performance. In *Proceedings of ACM Sigmetrics*, June 2001.

[3] L. Brakmo and L. Peterson. TCP Vegas: end-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–80, October 1995.

[4] C. Casetti, M. Gerla, S. Mascolo, M. Sansadidi and R. Wang. TCP Westwood: end to end congestion control for wired/wireless networks. *Wireless Networks Journal*, vol.8, pp.467-479, 2002

[5] S. Deb and R. Srikant. Rate-Based Versus Queue-Based Models of Congestion Control. *IEEE Transactions on Automatic Control*, 51(4): 606-618, April 2006

[6] N. Dukkipati, M. Kobayashi, R. Z. Shen and N. McKeown. Processor Sharing Flows in the Internet. Thirteenth International Workshop on Quality of Service (IWQoS), June 2005.

[7] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, August 1993.

[8] S. Floyd, M. Handley, J. Padhye and J. Widmer. Equation-Based Congestion Control for Unicast Applications. Sigcomm Symposium on Communications Architectures and Protocols, 2000.

[9] S. Floyd. HighSpeed TCP for large congestion windows. Internet draft draft-floyd-tcp-highspeed-02.txt work in progress, <http://www.icir.org/floyd/hstcp.html>

[10] F. Gantmacher. *The Theory of Matrices*. Chelsea Publishing Company, 1959.

[11] V. Jacobson. Congestion Avoidance and Control. *Proceedings of ACM Sigcomm*, 1988.

[12] R. Jain, W. Hawe, and D. Chiu. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC TR-301, Digital Equipment Corporation, 1984.

[13] R. Jain. A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks. *ACM Computer Communication Review*, 19(5):56–71, Oct. 1989.

[14] R. Johari and J. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3): 407-435, 2004

[15] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high-bandwidth delay product networks. In *Proceedings of ACM Sigcomm*, August 2002.

[16] A. Katok and B. Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press, 1995.

[17] F. Kelly, A. Maoullou, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[18] F. Kelly. Models for a self-managed Internet. *Philosophical Transactions of the Royal Society A358*: 2335-2348, 2000.

[19] T. Kelly. Scalable TCP: Improving Performance in Highspeed Wide Area Networks. *Computer Communication Review* 32(2), April 2003

[20] A. Kherani and A. Kumar. Stochastic models for throughput analysis of randomly arriving elastic flows in the Internet. In *Proceedings of IEEE Infocom*, 2002.

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

[22] S. Liu, T. Basar and R. Srikant. TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks. In *Proc. First Int. Conf. on Perform. Eval. Methodol. Tools (VALUETOOLS)*, 2006.

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

[24] S. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Trans. on Networking*, 11(4):525-536, August 2003.

[25] Microsoft Windows Vista Networking: TCP/IP Stack. <http://technet.microsoft.com/en-us/windowsvista/aa905087.aspx>

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

[27] J. Padhye, V. Firoiu, D. Towsley and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation. *ACM SIGCOMM Computer Communication Review*, 28(4):303-314, October 1998.

[28] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168, Internet Engineering Task Force, September 2001.

[29] T. Roughgarden and E. Tardos. How Bad Is Selfish Routing. *Journal of the ACM*, 49(2):236-259, 2002.

[30] S. Shakkottai, A. Kumar, A. Karnik and A. Anvekar. TCP performance over end-to-end rate control and stochastic available capacity. *IEEE/ACM Transactions on Networking*, 9(4):377-391, August 2001.

[31] S. Shakkottai and R. Srikant. Mean FDE Models for Internet Congestion Control Under a Many-Flows Regime. *IEEE Transactions on Information Theory*, 50(6): 1050-1072, June 2004.

[32] H. Scarf. Some examples of global instability of the competitive equilibrium. *International Economic Review*, 1(3):157–172, September 1960.

[33] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A compound TCP approach for high-speed and long distance networks. In *Proceedings of IEEE Infocom*, April 2006.

[34] A. Tang, A. Simsek, A. Ozdaglar and D. Acemoglu. On the Stability of P -Matrices. *Linear Algebra and Its Applications*, 426(1): 22-32, October 2007.

[35] A. Tang, J. Wang, S. Hedge and S. Low. Equilibrium and fairness of networks shared by TCP Reno and Vegas/FAST. *Telecommunication Systems*, 30(4):417-439, December 2005.

[36] A. Tang, J. Wang, S. Low and M. Chiang. Equilibrium of Heterogeneous Congestion Control: Existence and Uniqueness. *IEEE/ACM Transactions on Networking*, 15(4): 824-837, August 2007.

[37] A. Tang, L. Andrew. Game Theory for Heterogeneous Flow Control. In *Proceedings of 42th CISS*, March 2008.

[38] WAN-in-Lab. <http://wil.cs.caltech.edu>.

[39] Z. Wang and J. Crowcroft. Eliminating periodic packet losses in the 4.3-Tahoe BSD TCP congestion control algorithm. *ACM Computer Communication Review*, April 1992.

[40] D. Wei, C. Jin, S. Low and S. Hegde. FAST TCP: motivation, architecture, algorithms, performance. *IEEE/ACM Transactions on Networking*, 14(6): 1246-1259, December 2006.

[41] B. Wyrowski, L. H. Andrew, M. Zukerman. MaxNet: A Congestion Control Architecture for Scalable Networks. *IEEE Communications Letters*, 2003

[42] L. Xu, K. Harfoush, and I. Rhee. Binary Increase Congestion Control for Fast Long-Distance Networks. *Proceedings of IEEE Infocom*, 2004.

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

IX. APPENDIX

A. Stability: Special Cases

Theorem 12. *For a network with $L \leq 3$, if there is only one equilibrium, it is also locally stable.*

Proof. We want to prove all eigenvalues of $\mathbf{J}(p)$ lie in the left half plane. By the index theorem, we have $\det(-\mathbf{J}) > 0$ for the unique equilibrium.

When L equals 1 or 2, it is obvious as $J_{ii} < J_{ij} < 0$ for $j \neq i$. Let's consider the case with $L = 3$. Suppose $\lambda^3 + \rho_1\lambda^2 + \rho_2\lambda + \rho_3 = 0$ is the characteristic equation for \mathbf{J} . Then ρ_1 is the trace of $-\mathbf{J}$, ρ_2 is sum of all 2×2 principle minors of $-\mathbf{J}$ and $\rho_3 = \det(-\mathbf{J})$.

The Routh array for the equation is

$$\begin{bmatrix} 1 & \rho_2 \\ \rho_1 & \rho_3 \\ (\rho_1\rho_2 - \rho_3)/\rho_1 & 0 \\ \rho_3 & 0 \end{bmatrix}$$

Applying Routh stability criterion [10], we need all quantities in the left column to be positive to guarantee all roots lie in the left half plane. Clearly $\rho_1 > 0$. Global uniqueness implies $\rho_3 > 0$. Hence we only need to check $\rho_1\rho_2 > \rho_3$. We have

$$\begin{aligned} \det(\mathbf{J}) &= J_{11}(J_{22}J_{33} - J_{23}J_{32}) - J_{12}(J_{21}J_{33} - J_{23}J_{31}) \\ &\quad + J_{13}(J_{21}J_{32} - J_{22}J_{31}) \\ &> J_{11}(J_{22}J_{33} - J_{23}J_{32}) + J_{22}(J_{11}J_{33} - J_{13}J_{31}) \\ &\quad + J_{33}(J_{11}J_{22} - J_{12}J_{21}) \\ &> (J_{11} + J_{22} + J_{33})((J_{11}J_{22} - J_{12}J_{21}) \\ &\quad + (J_{11}J_{33} - J_{13}J_{31}) + (J_{22}J_{33} - J_{23}J_{32})) \\ &= -\rho_1\rho_2 \end{aligned}$$

The first inequality follows from $J_{ii} < J_{jj} < 0$ for $j \neq i$. The second one follows from $J_{ii}J_{jj} - J_{ij}J_{ji} > 0$ for $j \neq i$. One is referred to [36] for detail properties of \mathbf{J} . Therefore

$$\rho_3 = \det(-\mathbf{J}) = -\det(\mathbf{J}) < \rho_1\rho_2 \quad \square$$

As reviewed in III-B, when there is only one kind of price, global stability is proved by using the objective function of the dual of the system problem as a Lyapunov function. For heterogeneous protocols, we have the following.

Theorem 13. *For a network with $L \leq 2$, the equilibrium is globally asymptotically stable.*

Proof sketch. Assume the equilibrium price is p^* . When $L=1$, consider the simple quadratic Lyapunov function

$$L(p(t)) = (p(t) - p^*)^2$$

When $L=2$, consider

$$L(p(t)) = |p_1(t) - p_1^*| + |p_2(t) - p_2^*|$$

B. Proof of Lemma 10

Proof. Fix any $(j, \pi) \in \Theta_0$. Each term in (29) is indexed by a pair (σ, \mathbf{k}) . Fix also a permutation σ in (29). Suppose there is only one permutation \mathbf{k} for which the term indexed by (σ, \mathbf{k}) has a negative sign given by $\mathbf{1}(\sigma(\pi) \in \Pi(\mathbf{k}, \mathbf{l}))\text{sgn}(\mathbf{k})(-1)^{|L_{\mathbf{k}}^-|} = -1$. This term is then $-\mu(\sigma(j)) < 0$. Since the summation over \mathbf{k} ranges over all permutations, we can find a positive term, indexed by $(\sigma, \hat{\mathbf{k}})$ with $\hat{\mathbf{k}} = \mathbf{l}$, that exactly cancels this negative term. This is because $\mathbf{1}(\sigma(\pi) \in \Pi(\mathbf{l}, \mathbf{l}))$ is always 1 and $\text{sgn}(\mathbf{l})(-1)^{|L_{\mathbf{l}}^-|} = 1$, yielding the term $\mu(\sigma(j))$. Hence we have shown that, given σ , if there is only one \mathbf{k} that yields a negative term, then it is always cancelled by another positive term indexed by $(\sigma, \hat{\mathbf{k}})$ with $\hat{\mathbf{k}} = \mathbf{l}$.

Given a σ , suppose now there are two permutations \mathbf{k}, \mathbf{n} for which

$$\sigma(\pi) \in \Pi(\mathbf{k}, \mathbf{l}) \quad \text{and} \quad \sigma(\pi) \in \Pi(\mathbf{n}, \mathbf{l}) \quad (36)$$

and

$$\text{sgn}(\mathbf{k})(-1)^{|L_{\mathbf{k}}^-|} = \text{sgn}(\mathbf{n})(-1)^{|L_{\mathbf{n}}^-|} = -1 \quad (37)$$

Each of (σ, \mathbf{k}) and (σ, \mathbf{n}) yields a negative term $-\mu(\sigma(j))$ in the summation in (29). Notice that (36) says that, for all

$l = 1, \dots, L$, paths $\sigma(\pi)^l$ contains link pairs (k_l, l) and (n_l, l) . Hence $\sigma(\pi)^l$ also pass through link pairs (l, l) , (k_l, n_l) and (n_l, k_l) , i.e.,

$$\sigma(\pi) \in \Pi(\mathbf{l}, \mathbf{l}) \quad (38)$$

$$\sigma(\pi) \in \Pi(\mathbf{k}, \mathbf{n}), \quad \sigma(\pi) \in \Pi(\mathbf{n}, \mathbf{k}) \quad (39)$$

(38) implies that there is a positive term in the summation in (29) indexed by $(\sigma, \hat{\mathbf{k}})$ with $\hat{\mathbf{k}} = \mathbf{l}$:

$$\text{sgn}(\mathbf{l})(-1)^{|L_{\mathbf{l}}^-|}\mu(\sigma(j)) = \mu(\sigma(j)) > 0$$

It cancels the negative term $-\mu(\sigma(j))$ in the summation indexed by (σ, \mathbf{k}) .

To deal with the negative term $-\mu(\sigma(j))$ indexed by (σ, \mathbf{n}) , note that (39) implies that there are two nonzero terms in the summation, indexed by $(\mathbf{n}^{-1}\sigma, \mathbf{n}^{-1}\mathbf{k})$ and $(\mathbf{k}^{-1}\sigma, \mathbf{k}^{-1}\mathbf{n})$, that we now argue are positive. Indeed the term indexed by $(\mathbf{n}^{-1}\sigma, \mathbf{n}^{-1}\mathbf{k})$ is $\text{sgn}(\mathbf{n}^{-1}\mathbf{k})(-1)^{|L_{\mathbf{n}^{-1}\mathbf{k}}^-|}\mu(\mathbf{n}^{-1}(j))$. We further have

$$\begin{aligned} |L_{\mathbf{n}^{-1}\mathbf{k}}^-| &= |L_{\mathbf{k}}^- \cup L_{\mathbf{n}}^-| - |(L_{\mathbf{k}}^- \cap L_{\mathbf{n}}^-)| \\ &= |L_{\mathbf{k}}^-| + |L_{\mathbf{n}}^-| - 2|(L_{\mathbf{k}}^- \cap L_{\mathbf{n}}^-)| \end{aligned} \quad (40)$$

Hence

$$\begin{aligned} \text{sgn}(\mathbf{n}^{-1}\mathbf{k})(-1)^{|L_{\mathbf{n}^{-1}\mathbf{k}}^-|} &= \text{sgn}(\mathbf{n})\text{sgn}(\mathbf{k})(-1)^{|L_{\mathbf{k}}^-|}(-1)^{|L_{\mathbf{n}}^-|} \\ &= 1 \end{aligned}$$

The last equality follows from (37). Similarly, the term with index $(\mathbf{k}^{-1}\sigma, \mathbf{k}^{-1}\mathbf{n})$ is $\mu(\mathbf{k}^{-1}(j))$. The hypothesis of the lemma implies that

$$\mu(\mathbf{n}^{-1}(j)) + \mu(\mathbf{k}^{-1}(j)) - \mu(\sigma(j)) \geq 0$$

Hence, given σ , if there are two negative terms in the summation in (29) indexed by (σ, \mathbf{k}) and (σ, \mathbf{n}) , then we can always find three positive terms, indexed by (σ, \mathbf{l}) , $(\mathbf{n}^{-1}\sigma, \mathbf{n}^{-1}\mathbf{k})$ and $(\mathbf{k}^{-1}\sigma, \mathbf{k}^{-1}\mathbf{n})$, so that the sum of these five terms are nonnegative.

If there are more than two negative terms, take any *additional* negative term, indexed by, say, $(\sigma, \hat{\mathbf{n}})$. The same argument shows that it will be compensated by the two (unique) positive terms indexed by $(\hat{\mathbf{n}}^{-1}\sigma, \hat{\mathbf{n}}^{-1}\mathbf{k})$ and $(\mathbf{k}^{-1}\sigma, \mathbf{k}^{-1}\hat{\mathbf{n}})$. This completes the proof. \square

C. WAN-in-Lab Experiments

In this section, we present two experimental results to illustrate the behavior of Algorithm 2 in new scenarios: when the bottleneck buffer is small and when all flows are FAST. The experiments were conducted on a realistic testbed, Caltech's WAN-in-Lab, which is a wide area network consisting of 2,400 km of long haul optical fiber, a reconfigurable array of Cisco 7609 routers and ONS 15454 high speed switches, servers, clients, interconnected via OC-48, GbE and 10GbE links, using a Calient MEMS optical switch. considering some previously ignored scenarios (e.g., small buffer size, only FAST flows). We test our algorithm with a single bottleneck link with 1Gbps capacity.

Example 5: small buffer size

As we have seen in Example 1a and Example 1b, Algorithm 2 can significantly increase link utilization when buffer size B is not too larger than α . In this experiment, we have $B < \alpha$. Without Algorithm 2, every FAST flow tries to maintain α packets in the queues along its path leading to high packet loss rate and poor throughput for both Reno and FAST. With Algorithm 2, α is automatically adjusted to a proper value with respect to the network parameter B .

One FAST and one Reno compete for bandwidth at a bottleneck link of 1Gbps (80pkts/ms) capacity. The buffer capacity B is 480pkts. The initial α is set to be $\alpha=800$ pkts. The results are summarized in Figure 19. As the left part of the figure shows, both Reno and FAST get very low throughput due to the high packet loss rate (FAST: 135Mbps; Reno: 22Mbps). However, using Algorithm 2, FAST decreases its α as it sees high loss and finally both flows get high throughput (FAST: 593Mbps; Reno: 246Mbps). The utilization is increased significantly from 15.7 percent to 83.9 percent.

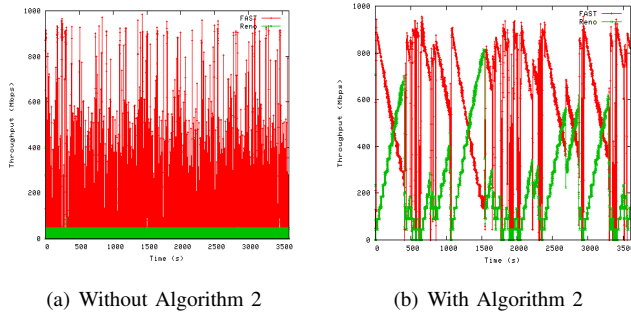


Fig. 19. Bandwidth partition between Reno and FAST

Example 6: only FAST flows

Although the slow timescale update shows desirable properties in various tests we have discussed so far, there is a problem we have not touched, namely the case when there are only FAST flows in a network. As FAST is designed to achieve a steady state with no loss, flows will keep increasing their α according to Algorithm 2 until the buffer is filled and loss is generated. This is undesirable and we propose to turn off the slow timescale update when a FAST flow has not seen any loss for a certain amount of time (ten seconds by default). We conduct a test using three FAST flows all with initial $\alpha=200$ pkts sharing the common 1Gbps link. The throughput trajectories are shown in Figure 20. We can see that after a period of adjustment, all flows are stabilized. The steady state throughputs are 128Mbps, 234Mbps and 566Mbps, which result in a high utilization of 92.8 percent even though the initial sum of α (600 pkts) exceeds the buffer capacity (480 pkts). However, this introduces potential fairness problem as we cannot make sure the individual α values are equal when the update algorithm stops. For example, instead of achieving perfect fairness with a Jain index [12] of 1, we have 0.733 in this experiment.

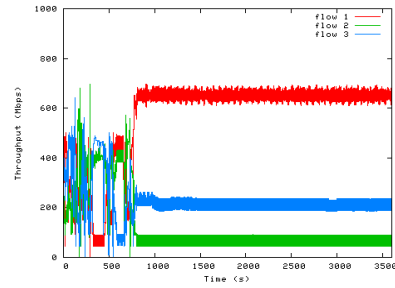


Fig. 20. Bandwidth sharing among FAST flows

PLACE PHOTO HERE

Ao Tang (S'01-M'07) received the B.E. (Hon.) in electronics engineering from Tsinghua University, Beijing, China, and the Ph.D. in electrical engineering with a minor in applied mathematics from Caltech, Pasadena, CA.

He is currently an Assistant Professor in the School of Electrical and Computer Engineering at Cornell University, where his research interests include communication networks, dynamical systems, and network multi-resolution analysis.

PLACE PHOTO HERE

Xiaoliang (David) Wei received his B.E. degree in Computer Science from Tsinghua University, Beijing, China, in 2001. He then obtained his Ph.D. degree in Computer Science in Caltech in 2007. He is currently working for Facebook Inc, where his research interests include network congestion control, active queue management, and routing.

PLACE PHOTO HERE

Steven H. Low (M'92-SM'99-F'08) received his B.S. from Cornell University, and PhD from Berkeley, both in electrical engineering.

He is a Professor of the Computer Science and Electrical Engineering Departments at Caltech. He was with AT&T Bell Laboratories, Murray Hill, from 1992 to 1996, the University of Melbourne, Australia from 1996 to 2000, and was a Senior Fellow of the University of Melbourne from 2000 to 2004.

PLACE PHOTO HERE

Mung Chiang received the B.S. (Hon.) degree in electrical engineering and mathematics, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1999, 2000, and 2003, respectively.

He is an Associate Professor of Electrical Engineering at Princeton University. He conducts research in the areas of optimization of communication systems, analytic foundations of network architectures, algorithms in broadband access, and information theory.