Fair End-to-End Window-based Congestion Control

Jeonghoon Mo[a] and Jean Walrand[b]

[a]Department of IEOR,UC Berkeley, Berkeley, CA 94720
[b]Department of EECS,UC Berkeley, Berkeley, CA 94720

ABSTRACT

In this paper, we demonstrate the existence of fair end-to-end window-based congestion control protocols for packet-switched networks with FCFS routers. Our definition of fairness generalizes proportional fairness and includes arbitrarily close approximations of max-min fairness. The protocols use only information that is available to end hosts and are designed to converge reasonably fast.

Our study is based on a multiclass fluid model of the network. The convergence of the protocols is proved using a Lyapunov function. The technical challenge is in the construction of the protocols.

Keywords: congestion control, end to end, fairness, window, max-min fair, proportionally fair

1. INTRODUCTION

We study the existence of fair end-to-end congestion control schemes. More precisely, the question is that of the existence of congestion control protocols that converge to a fair equilibrium without the help of the internal network nodes, or routers. Using such a protocol, end-nodes, or hosts, monitor their connections. By so doing, the hosts get implicit feedback from the network such as round-trip delays and throughput but no explicit signals from the network routers. The hosts implement a window congestion control mechanism. Such end-to-end control schemes do not need any network configuration and therefore could be implemented in the Internet without modifying the existing routers nor the IP protocol.

The Internet congestion control is implemented in end-to-end protocols. The motivation for such protocols is that they place the complex functions in the hosts and not inside the network. Consequently, only the hosts that want to implement different complex functions need to have their software upgraded. Another justification, which is more difficult to make precise, is that by keeping the network simple it can scale more easily.

TCP is the most widely used end-to-end protocol in the Internet. When using TCP,[1] a source host adjusts its windows size, the maximum amount of outstanding packets it can send to the network, to avoid overloading routers in the network and the destination host.

Many researchers have observed that, when using TCP, connections with a long round-trip time that go through many bottlenecks have a smaller transmission rate that the other connections.[2–4] A bottleneck is a node where packets are backlogged so that its transmission rate limits the rate of the connections that go through it. The observed bias can be explained as follows. While a host does not detect congestion, it increases its window size by one unit per round-trip time of the connection. Accordingly, the window size of a connection with a short propagation delay increases faster than that of a connection with a longer propagation delay. Consequently, a long-delay connection looses out when competing with a short-delay connection.

Based on this observation, Floyd and Jacobson[5] proposed a "constant rate adjustment" algorithm. Handerson et al[6] simulated a variation of this scheme. They report that if the rate of increase of the window size is not excessive, then this scheme is not harmful to the other connections that use the original TCP scheme. Moreover, as expected, this scheme results in better performance for connections with longer propagation time. However, choosing the parameters of such algorithms is still an open problem.

Thus, although end-to-end protocols such as those implemented in TCP are very desirable for extensibility and scalability reasons, they are unfair. Roughly, a fair scheme is one that does not penalize some users arbitrarily. Accordingly, the question that arises naturally is the existence of fair end-to-end congestion protocols.

Other author information: (Send correspondence to J. Mo)
J.Mo: E-mail: jhmo@eecs.berkeley.edu; WWW: http://path.berkeley.edu/~jhmo
J.Walrand: E-mail: wlr@eecs.berkeley.edu; WWW: http://path.berkeley.edu/~wlr

In an early paper, Jaffe[7] shows that power cannot be optimized in a distributed manner.

Chiu and Jain[8] show that in a network with N users that share a unique bottleneck node, a linear increase and multiplicative decrease algorithm converges to an efficient and fair equilibrium. Most current implementations of TCP window-based control use a linear increase and multiplicative decrease of the window size, as suggested by Jacobson.[1] However, these implementations control the size of their window and not their transmission rate. Moreover, simple examples show that the result does not hold for networks with multiple bottleneck nodes.

Shenker[9] considers a limited class of protocols and argues that "no aggregate feedback control is guaranteed fair." This statement suggests that end-to-end control cannot guarantee convergence to a fair equilibrium. Unfortunately, the class of protocols that he considers excludes many implementable end-to-end protocols. Jain and Charny refers to Shenker[9] to justify the necessity of switch-based control for fairness.[10,11]

Recently, Kelly et al[12] exhibited an aggregate feedback algorithm that converges to a *proportionally fair* point. In their scheme, each user is implementing a linear increase and multiplicative decrease of its rate based on an additive feedback from the routers the connection goes through. This protocol requires that the routers can signal the difference between their load and their capacity. In our protocol, each host controls its window size not its rate, based on the total delay. Window-based algorithms are used to control errors. A rate-based control must be augmented with another retransmission protocol for error control. The window-based control algorithm integrates these two functions of error and congestion control. Although the delay is an additive congestion signal, this is less informative than Kelly's. Our protocol can be viewed as a refinement of TCP congestion control algorithms.

In this paper we revisit the fundamental question of the existence of fair end-to-end protocols and we provide a positive answer by constructing explicitly such protocols.

2. MODEL

Window flow control is usually modeled as a closed queuing network.[13–15] For instance, Mitra et al[13] study the window flow control of *a single connection* with fixed propagation delay in a product form network. They derived the optimal window size and an adaptive window-based control scheme based on the analytical model.

In this paper we consider *a closed multiclass fluid network* with M links and N connections. We define that model next. The sender of connection i ($i = 1, \ldots, N$) exercises a window-type flow control and adjusts the window size w_i of the connection. A connection follows a route that is a set of links. Link j ($j = 1, \cdots, M$) has capacity, or transmission rate, c_j. We define the matrix $A = (A_{ij}, i = 1, \ldots, N, j = 1, \cdots, M)$ where $A_{ij} = 1$ if connection i uses link j and $A_{ij} = 0$, otherwise. Let also $A_{i\cdot} := \{j | A_{ij} = 1\}$ be the set of links that connection i uses and $A_{\cdot j} := \{i | A_{ij} = 1\}$ the set of connections that use link j.

Each connection i has a fixed round-trip propagation delay d_i, which is the minimum delay between the sending of a packet by the sender host and the reception of its acknowledgment by the same host. We assume that the processing times are negligible. A typical acknowledgment delay comprises d_i and some additional queuing delay in bottleneck routers. Let x_i be the flow rate of the i-th connection for $i = 1, \ldots, N$. For $j = 1, \cdots, M$, we assume that every link j has an infinite buffer space and we designate by q_j the work to be done by link j. By definition, q_j is the ratio of the queue size in the buffer of link j divided by the capacity c_j. The service discipline of the links is first come - first served (FCFS).

We consider a fluid model of the network where the packets are infinitely divisible and small. This model is represented by following equations:

$$A^T x - c \leq 0 \quad (1)$$
$$Q(A^T x - c) = 0 \quad (2)$$
$$X(Aq + d) = w \quad (3)$$
$$x \geq 0,\ q \geq 0 \quad (4)$$

where

$$\begin{aligned} x &= (x_1, \ldots, x_N)^T,\ c = (c_1, \ldots, c_M)^T, \\ q &= (q_1, \ldots, q_M)^T,\ d = (d_1, \ldots, d_N)^T, \\ X &= diag(x),\ Q = diag(q). \end{aligned}$$

The inequalities (1) express the capacity constraints: the sum of the rates of flows that go through a link cannot exceed the capacity of the link. The identities (2) can be written as

$$q_j[(A^T x)_j - c_j] = 0, \text{ for } j = 1, \ldots, M.$$

The j-th identity means that if the rate $(A^T x)_j$ through link j is less than the capacity c_j of the link, then the queue size q_j at that link is equal to 0. Finally, the identities (3), which can be written as

$$x_i[(Aq)_i + d_i] = w_i, i = 1, \ldots, N,$$

mean that the total number of packets w_i for each connection $i, i = 1, \ldots, N$, is equal to the number $x_i d_i$ of packets in transit in the transmission lines plus the total number $x_i(Aq)_i$ of packets of connection i stored in buffers along the route. To clarify the meaning of $x_i(Aq)_i$, note that

$$x_i(Aq)_i = x_i \sum_j A_{ij} q_j = \sum_j A_{ij} x_i q_j.$$

Now, $c_j q_j$ is the number of packets in the buffer of link j and a fraction x_i/c_j of these packets are of connection i. Thus, $(c_j q_j)(x_i/c_j) = x_i q_j$ is the backlog of packets of connection i in the buffer of link j. Summing over all j such that connection i goes through link j shows that $x_i(Aq)_i$ is the total backlog of packets of connection i.

Note that our model assumes that, for each link j, the contribution to the queue size of connection i is proportional to its flow rate x_i. This assumption is consistent with the fluid assumption under which the packets are infinitely divisible.

We rewrite the i-th identity of (3) as follows:

$$x_i = \frac{w_i}{D_i} \text{ where } D_i = d_i + (Aq)_i. \tag{5}$$

The identity (5) means that the flow rate x_i of connection i is equal to the ratio of the window size w_i of the connection divided by its total round-trip delay D_i. The total delay D_i consists of fixed propagation delay d_i plus a variable queuing delay which depends on congestion in the network. Accordingly, the flow rate x_i of connection i is a function of not only the window size w_i of the connection but also of the window sizes of the other connections. When the network is not congested, $q = (q_1, \ldots, q_M) = 0$ and the flow rates are proportional to the window sizes. However, as congestion builds up, $q \neq 0$ and the rates are no longer linear in the window sizes.

We prove that the flow rates x are a well-defined function of the window sizes w. This result is intuitively clear and its proof is a confirmation that the model captures the essence of the physical system. Before proving the uniqueness of the rate vector x, we first show the existence of a rate vector x that solves the relations that characterize the fluid model. All the proofs of this paper are given in Mo et al.[16]

THEOREM 1. *For given values of (w, A, d, c), there exists at least one rate vector x which satisfies the relations (1)-(4).*

THEOREM 2. *Given (w, A, d, c), the flow rate $x = (x_1, \cdots, x_M)$ satisfying the equations (1)- (4) is unique.*

Although the rate vector is uniquely determined from the window sizes, the workload vector q generally is not, as the following example shows. Consider a network with two bottleneck links in series with the same capacity c and a single connection with window size w. If $\frac{w}{d} > c$, then the queues build up in the links. For this network, any vector (q_1, q_2) such that $q_1 + q_2 = \frac{w}{c} - d$ is a solution of the equations (1) - (3).

The following corollary shows a sufficient condition for q to be determined uniquely. Let A_B be the sub-matrix of A corresponding to set B of bottlenecks.

COROLLARY 1. *If $rank(A_B)$ is equal to the number $|B|$ of bottleneck links, then (w, A, c, d), uniquely determines the vector q.* The following lemma provides sufficient conditions for links not to be bottlenecks.

LEMMA 2.1. *For any given window size vector w, $0-1$ matrix A, and diagonal matrix D,*

- *(a) if ${A_{\cdot j}}^T D^{-1} w \leq c_j$, then $q_j = 0$.*

- *(b)* $A^T D^{-1} w \le c$ *if and only if* $q = 0$.

The above lemma can be proved if we observe that $\frac{w_i}{d_i}$ is an upper bound on x_i.

The converse of part (a) is not always true, as can be seen from the next example. Let $M = 2, N = 2, C = (5,5)^T, d = (1,1)^T$, and

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

If $w = (10, 20)$, clearly, $q_2 = 0$, the flow rate out of resource $1 \le 5$, but $A^T_{\cdot 2} D^{-1} w = 10 > 5$.

Let $F : W \to X$ be the mapping from the window space W to a flow rate space X defined by (1)-(4). F is a continuous function but is not always differentiable as the next example shows. Consider the network and connections in figure 1(a). Two users are sharing one link and each uses another link. Figure 1(b) is a plot of x_1

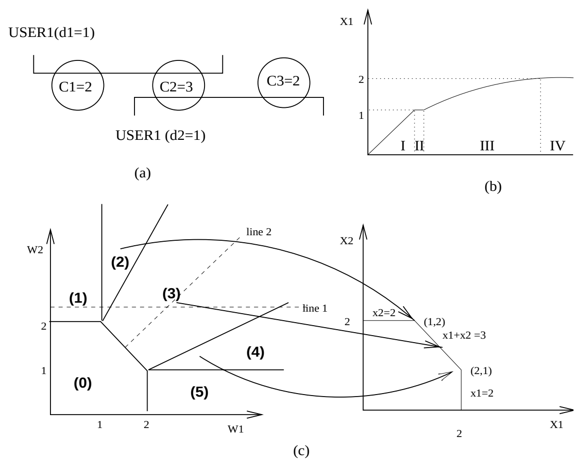

Figure 1. (a)network topology (b) flow rate x vs. window size (c) mapping between x and w

along the horizontal dotted line 1 in figure 1(c). Figure 1(b) shows that x_1 is a continuous nondecreasing function of the window size w_1, but is not differentiable at the points where the set of bottlenecks changes. Each region I,II,III, and IV corresponds to different sets of bottlenecks. For example, in region I, user 1 does not suffer from any bottlenecks, but user 2 does.

Figure 1(c) shows the mapping $x = F(w)$. If $w \in (0)$, there is no queue, and w and x are such that $w_i = x_i d_i$, so that $x = F(w)$ is differentiable in that region. If $w \notin (0)$, F is no longer one to one. For instance, $F(w) = (1,2)$ for all $w \in (2)$ and $F(w) = (2,1)$ for all $w \in (4)$.

Let $F^{-1}(x) = \{w | F(w) = x\}$. The dimension of $F^{-1}(x)$ is related to the number of bottlenecks. To be precise, the dimension of $F^{-1}(x)$ is same as the rank of A_B. This property follows from $w = Xd + XAq$. Since $XAq = XA_B q_B$, $F^{-1}(w)$ is a positive cone of XA_B with vertex Xd, as we now illustrate in figure 1(c). When $q = 0$, the inverse image of F is a point, of which the dimension is 0. When $x = (1.5, 1.5)$, $F^{-1}(x)$ is the dotted line 2 in the figure, whose dimension is 1. When $x = (2,1)$ or $(1,2)$, the number of bottlenecks is 2, which is the dimension of $F^{-1}(x)$.

Let $B(w)$ be the set of bottlenecks for the window sizes w. We call w an interior point if there is $\epsilon > 0$ such that $B(\bar{w})$ are same for all $\bar{w} \in$ neighborhood, $N_\epsilon(w)$, of w. Otherwise, w is said to be a boundary point.

Claim 1. *F is a continuous function of w.*

Claim 2. *F is differentiable except at the boundary points.*

Corollary 2. *Let $D_u^+ F = \lim_{\epsilon \downarrow 0} \frac{F(w+\epsilon u) - F(w)}{\epsilon}$. Then $D_u^+ F$ exists for all w.*

3. FAIRNESS

3.1. Fairness

Fairness has been defined in a number of different ways. One of the most common fairness definitions is *max-min* or *bottleneck optimality* criterion.[17–20,10] A feasible flow rate x is defined to be *max-min fair* if any rate x_i cannot be increased without decreasing some x_j which is smaller than or equal to x_i.[18] Many researcher have developed algorithms achieving *max-min fair* rates.[18,20,10] But computing the *max-min fair* vector requires global information,[21] and most of those algorithms require exchange of information between network and hosts. Hahne[19] suggested a simple round-robin scheduling that achieves *max-min fairness*. However, this algorithm requires that all the links perform round-robin scheduling and that packets of all users are always ready for all links.

Kelly[22] proposed *proportionally fairness*. A vector of rates x is *proportionally fair* if it is feasible, that is $x^* \geq 0$ and $A^T x^* \leq c$, and if for any other feasible vector x, the aggregate of proportional changes is negative:

$$\sum_i \frac{x_i - x_i^*}{x_i^*} < 0. \tag{6}$$

Kelly et al[12] suggested a simple linear increase and multiplicative decrease algorithm that converges to the *proportionally fair* point.

Recently, game theory has been applied to flow control.[23–25] These authors model users as players competing for common shared resources. The concept of *Nash Equilibrium* provides a framework for defining fairness and proper operating points for the network. The game can be viewed as non-cooperative[24] or cooperative.[23]

Next, we generalize the concept of *proportional fairness*. Consider the following optimization problem:
(P)

$$\begin{aligned} \text{maximize} \quad & g = \textstyle\sum_i p_i f(x_i) && (7)\\ \text{subject to} \quad & A^T x \leq c && (8)\\ \text{over} \quad & x \geq 0 && (9) \end{aligned}$$

where f is an increasing strictly concave function and the p_i are positive numbers. Since the objective function (7) is strictly concave and the feasible region (8)-(9) is compact, the optimal solution of (P) is exist and unique. Let $L(x,\mu) = g(x) + \mu^T(c - A^T x)$. The Kuhn-Tucker conditions[26] for a solution x^* of (P) are

$$\nabla g^T - \mu^T A^T = 0 \tag{10}$$
$$\mu_j(c_j - A^T_{\cdot j} x^*) = 0 \text{ for } j = 1, \cdots, M \tag{11}$$
$$A^T x^* \leq c \tag{12}$$
$$x^* \geq 0,\ \mu \geq 0 \tag{13}$$

where $\nabla g^T = (p_1 f'(x_1), \cdots, p_n f'(x_n))$. When there is only one link and N connections, the optimal solution of (P) is $x_i = \frac{c}{N}$ for all i: All the connections have an equal share of the bottleneck capacity, irrespective of the increasing concave f. Indeed, (10) implies $f'(x_i) = \mu$ for all i, so that $x_i = f'^{-1}(\mu)$ for all i. If x is a proportionally fair vector then it solves (P) when $f(x) = \log x$ with $p_i = 1$ for all i. Thus, a proportionally fair vector is one that maximizes the sum of all the logarithmic utility functions. The situation is not the same when there are multiple bottlenecks. Consider the following network with 2 different bottlenecks and 3 connections. The *max-min fair* rate vector of this

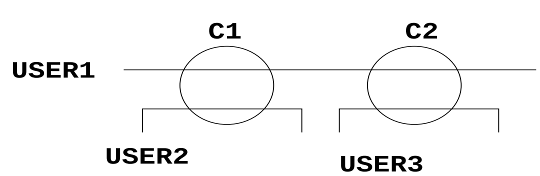

Figure 2. Network with multiple bottlenecks

network is $(\frac{c_1}{2}, \frac{c_1}{2}, c_2 - \frac{c_1}{2})$ if $c_1 < c_2$, while the *proportionally fair* rate vector is different, since by decreasing the rate of user 1, the sum of the utility functions f increases. Hence the optimal vector x depends on the utility function f when there are at least two bottlenecks.

It is the concavity of the function f that forces fairness between users. If f is a convex increasing function instead of concave, then to maximize the objective function g of (P), the larger flow rate x_i should be increased, since the rate of increase of $f(x_i)$ is increasing in x_i. When f is linear, the rate of increase of f is the same for all x. When f is concave, a smaller x_i favored, since $f'(x) > f'(y)$ if $x < y$.

It is a matter of controversy what is a fair rate allocation for the network in figure 2. It can be argued that the *max-min fair* rate is desirable. On the other hand, connection 1 is using more resources than the others under the *max-min fair* rate. Generally, the problem is how to compromise between the fairness to users and the utilization of resources. The max-min definition gives the absolute priority to the fairness.

We generalize the concept of *proportional fairness* as follows.

DEFINITION 1 ((p, α)-PROPORTIONALLY FAIR).

*Let $p = \{p_1, \ldots, p_N\}$ and α be positive numbers. A vector of rates x^* is (p, α)-*proportional fair *if it is feasible and for any other feasible vector x,*

$$\sum_i p_i \frac{x_i - x_i^*}{{x_i^*}^{\alpha}} < 0. \tag{14}$$

Note that (14) reduces to (6) when $p = (1, \cdots, 1)^T$ and $\alpha = 1$.

The following lemma clarifies the relationship between the above definition and the problem (P).

LEMMA 3.1. *Define the function f_α as follows:*

$$f_\alpha(x) := \begin{cases} x^{1-\alpha} & \text{if } 0 < \alpha < 1 \\ \log x & \text{if } \alpha = 1 \\ -x^{1-\alpha} & \text{if } \alpha > 1. \end{cases}$$

Then the rate vector x^* solves the problem (P) with $f = f_\alpha$ if and only if x^* is (p, α)-proportionally fair.

The next lemma explains the relationship between *max-min fair* rate and the parameter α.

LEMMA 3.2. *If h is increasing concave negative function, the solution of (P) with $f_n = -(-h)^n$ approaches the* max-min fair *rates as $n \to \infty$.*

$$\text{Since } \frac{f_n'(x)}{f_n'(x+\epsilon)} = \left(\frac{h(x)}{h(x+\epsilon)}\right)^{n-1} \frac{h'(x)}{h'(x+\epsilon)} \to \infty \text{ as } n \to \infty,$$

the function f_n gives more priority to smaller flows as n increases.

COROLLARY 3. *The (p, α)-* proportionally fair *rate vector approaches the* max-min *fair rate vector as $\alpha \to \infty$.*

3.2. Window Size and Fairness

In this subsection we study the relationship between window sizes and fairness.

TCP Vegas[27] uses the estimated total backlog of a connection as a decision function. In our notation, the total backlog of connection i is $w_i - x_i d_i$. In TCP Vegas, a host increases its window size if the estimated total backlog is smaller than a target value and decreases it otherwise.

We now establish the relationship between the total backlogs and fairness. Let $p_i > 0$ for $i = 1, \cdots, N$. Define

$$s_i = w_i - x_i d_i - p_i, \text{ for } i = 1, \cdots, N. \tag{15}$$

The next theorem shows that any window vector w such that $s_i = 0$ for all i corresponds to a $(p, 1)$-proportionally fair rate vector x.

THEOREM 3. *There is a unique window vector w such that $s_i = 0$ for $i = 1, \cdots, N$. Moreover, the corresponding rate vector $x(w)$ defined by the equations (1)-(4) is a $(p, 1)$-proportionally fair rate vector.*

Proof. The key observation is q plays the role of Lagrange multiplier of (P). Refer to [16] for a detailed proof. ■

This theorem implies that by controlling the total backlogs of the network, we can operate the network at the $(p,1)$-proportionally fair point.

This theorem can be extended to the (p,α)-proportionally fair case. Let $p_i > 0$ for $i = 1, \cdots, N$ and $\alpha > 1$. Define

$$s_i^\alpha = w_i - x_i d_i - \frac{p_i}{x_i^{\alpha-1}}, \text{ for } i = 1, \cdots, N. \tag{16}$$

THEOREM 4. *There is a unique window vector w such that $s_i^\alpha = 0$ for all i. Moreover, the corresponding rate vector $x(w)$ defined by the equations (1)-(4) is a (p,α)- proportionally fair rate vector.*

4. FAIR END-TO-END ALGORITHM

4.1. (p,1)-Proportionally Fair Algorithm

In this section we construct an end-to-end control that converges to the *proportionally fair* point. Define

$$\bar{d}_i = d_i + A_{i\cdot} q, \text{ for } i = 1, \cdots, N.$$

That is, $\bar{d}_i$ is the measured round-trip delay of connection i. Fix $\kappa > 0$.

Consider the following system of differential equations:

$$\frac{d}{dt} w_i(t) = -\kappa \frac{d_i}{\bar{d}_i} \frac{s_i}{w_i} \tag{17}$$

$$s_i = w_i - x_i d_i - p_i \text{ for } i = 1, \cdots, N. \tag{18}$$

THEOREM 5. *Let $V(w) = \sum_{i=1}^N \left(\frac{s_i}{w_i}\right)^2$. Then V is a Lyapunov function for the system of differential equations (17)-(18). The unique value minimizing V is a stable point of this system, to which all trajectories converge.*

Proof. We show in [16] that $\frac{d}{dt} V(w(t)) = -\sum_j \sum_i \frac{s_i}{w_i} \frac{dr_i}{dw_j} \dot{w}_j < 0$ for all t. ■

Kelly et al[12] proposed a rate control algorithm that converges to the *proportionally fair* point. The algorithm changes the rate as follows:

$$\frac{d}{dt} x_i(t) = \kappa(p_i - x_i(t) A_{i\cdot} \mu(t))$$

where

$$\mu_j(t) = ((A^T x)_j - C_j + \epsilon)/\epsilon^2.$$

The source i gets the explicit feedback $\sum_j \mu_j(t)$, sum of residual capacities, from the links and changes its rate accordingly. The increase is linear and the decrease is multiplicative. Each $\mu_j(t)$ play the role of a Lagrange multipliers of the problem P as $\epsilon \rightarrow 0$.

Our algorithm, however, controls the window size instead of the rate explicitly. The rate is a function of all windows. The algorithm (17)-(18) can be written as follows:

$$\frac{d}{dt} w_i(t) = \kappa\left(\frac{p_i}{w_i} + \frac{d_i}{\bar{d}_i} - 1\right) \frac{d_i}{\bar{d}_i}$$

where

$$\bar{d}_i = d_i + \sum_{j \in A_{\cdot i}} q_j.$$

Here, the measured delay $\bar{d}_i$ plays the role of the feedback. This delay is the summation of q_j plus d_i. Thus, q_j in our algorithm is comparable to μ_j in Kelly's. They are both Lagrange multipliers of (P). However, we do not linearly increase and multiplicatively decrease the window. When the network is not congested, $q = 0$, $\dot{w} = \kappa \frac{p_i}{w_i}$. and the increasing rate is a decreasing function of w.

4.2. (p,α)-Proportionally Fair Algorithm

In this subsection, we consider an algorithm that converges to an (p,α)- *proportionally fair* rate vector. We know that if $s_i^\alpha = w_i - x_i d_i - \frac{p_i}{{x_i}^{\alpha-1}} = 0$ for all i, then the rate vector is (p,α)-*proportionally fair*. We call $\frac{p_i}{{x_i}^{\alpha-1}}$ the "target queue length," since $w_i - x_i d_i$ is the estimated queue length in the network. Note that target queue length goes to infinity when the rate is very small. When $\alpha = 1$, the target queue length is constant regardless of the rate. On the other hand, when $\alpha > 1$, the target queue length is a function of x, which is varying and is a decreasing function of the rate. Hence, when the flow rate is large, the algorithm tries to maintain smaller queue and vice versa.

One unfavorable property of the target queue length function $\frac{p_i}{{x_i}^{\alpha-1}}$ is that when $x_i < 1$, this function becomes very large and the target queue length fluctuates and makes the control unstable. Consequently, we consider $\frac{p_i}{(x_i+1)^{\alpha-1}}$ instead of $\frac{p_i}{{x_i}^{\alpha-1}}$, since the variation of the former is smaller than that of the latter.

The objective function h_α such that the solution of (P) corresponds to $\bar{s}^\alpha = w_i - x_i d_i - \frac{p_i}{x_i+1^{\alpha-1}} = 0$ is

$$h_\alpha(x) = \begin{cases} \log x & \text{if } \alpha = 1; \\ \log(\frac{x}{x+1}) & \text{if } \alpha = 2; \\ \log(\frac{x}{x+1}) + \sum_{i=1}^{\alpha-2} \frac{1}{i(x+1)^i} & \text{if } \alpha \geq 3. \end{cases}$$

Note that $h'_\alpha = \frac{1}{x(x+1)^{\alpha-1}}$ and $\lim_{x\to\infty} h_\alpha = 0$. These observations show that $h_a p$ is increasing concave and nonnegative, and by the claim 3.2, the solution of (P) with objective function h_α converges to max-min rate vector.

Consider the system of differential equations

$$\frac{d}{dt} w_i = -\kappa x_i s_i u_i \tag{19}$$

where

$$s_i = w_i - x_i d_i - \frac{p_i}{(x_i+1)^{\alpha-1}} \text{ and } u_i = d_i - (\alpha-1)\frac{p_i}{(x_i+1)^\alpha}. \tag{20}$$

THEOREM 6. *If $p_i < \frac{d_i}{\alpha-1}$ for all i, the function $V(w) = \frac{1}{2}\sum_i s_i^2$ is a Lyapunov function for the system of equations (19)-(20). The unique value w minimizing $V(w)$ is a stable point of the system to which all trajectories converge.*

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper we have addressed the fundamental question of the existence of fair end-to-end window-based congestion control. We have shown the existence of window-based fair end-to-end congestion control using multiclass closed fluid model. We showed that the flow rates are a well defined function of the window sizes and characterized this function. We generalized the proportional fairness and related the fairness to the optimization problem. Our definition of fairness addresses the compromise between user fairness and resource utilization. With the help of optimization problem, we have related window sizes and the fair point. We have developed an algorithm which converges to the fair point and proved its convergence using a Lyapunov function.

Our algorithm uses the propagation delay d_i, measured delay $\bar{d}_i$, and window size w_i. Unfortunately, the end user cannot know the exact value of propagation delay. Furthermore, the value of propagation delay could change in the case of rerouting in packet-switched networks. TCP Vegas uses the minimum of delays observed so far as an estimated propagation delay. TCP-Vegas fails to adapt to the route change when the changed route is longer than original route. Refer to La et al[28] for more detail. The challenging question that remains is the implementability of this protocol.

REFERENCES

1. V. Jacobson, "Congestion avoidance and control ," *Computer Communication Review* **18**, pp. 314–29, August 1988.
2. S. Floyd and V. Jacobson, "Connection with multiple Congested Gateways in packet-Switched Networks,Part1: One-way Traffic"," *ACM Computer Communication Review* **21**, pp. 30–47, August 1991.

3. S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking* **1**, pp. 397–413, August 1993.
4. A. Mankin, “Random Drop Congestion Control,” in *Proc. SIGCOMM'90*, pp. 1–7, (Philadelphia,PA), September 1990.
5. S. Floyd and V. Jacobson, “On Traffic Phase Effects in packet Switched Gateways,” *Internetworking:Research and Experience* **3**, pp. 115–156, September 1993.
6. T. Henderson, E. Sahouria, S. McCanne, and R. Katz, “Improving Fairness of TCP Congestion Avoidance ,” in *Globecom '98*, 1998.
7. J. Jaffe, “Flow Control Power is Nondecentralizable,” *IEEE Transactions on Communications* **COM-29**, pp. 1301–1306, September 1981.
8. D. Chiu and R. Jain, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” *Computer Networks and ISDN Systems* **17**, pp. 1–14, 1989.
9. S. Shenker, “A theoretical analysis of feedback flow control,” in *ACM SIGCOM '90 Symposium*, (Philadelphia, PA), 24-27 Sept 1990.
10. A. Charny, “An algorithm for rate allocation in a packet-switching network with feedback,” Master's thesis, MIT, Cambridge,MA, 1994.
11. R.Jain, “Myths about Congestion Management in High Speed Networks,” *Internetworking:Research and Experience* **3**, pp. 101–113, 1992.
12. F. Kelly, A. Maulloo, and D. Tan, “Rate control for communication networks: shadow price proportional fairness and stability,” *Journal of the Operational Research Society* **49**, pp. 237–252, 1998.
13. D.Mitra, “Asymptotically optimal design of congestion control for high speed data network,” *IEEE Transaction on Communication* **40**, pp. 301–311, February 1992.
14. D.Mitra and I.Mitrani, “Efficient window flow control for high speed data networks with small buffers,” *Annals of Operations Research* **49**, pp. 1–24, March 1994.
15. K. Fendick, D.Mitra, I. Mitrani, M. Rodrigues, J. Seery, and A. Weiss, “An approach to high-performance, high-speed data networks,” *IEEE Communications Magazine* **29**, pp. 74–82, October 1991.
16. J. Mo and J. Walrand, “Fair end to end window-based congestion control,” Tech. Rep. M98/46, UCB/ERL, July 1998. available at http://www.path.berkeley.edu/˜ jhmo.
17. J. Jaffe, “Bottleneck flow control,” *IEEE Transaction on Communications* **COM-29**, July 1981.
18. D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Englwood Cliffs,N.J., 1992.
19. E. Hahne, “Round-robin scheduling for maxmin fairness in data network,” *IEEE Journal on Selected Areas in Communications* **9**, pp. 1024–39, September 1991.
20. R. Jain, “Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey ,” *Computer Networks and ISDN Systems* **28**, pp. 1723–38, October 1996.
21. A. Mayer, Y. Ofek, and M. Yung, “Approximating Max-Min Fair Rates via Distributed Local Scheduling with Partial Information,” in *IEEE Infocom '96*, pp. 926–936, (San Francisco, CA, USA), March 1996.
22. F. Kelly, “Charging and rate control for elastic traffic,” *European Transactions on Telecommunications* **8**, pp. 33–37, Jan-Feb 1997.
23. R. Mazumdar, L. Mason, and C. Douligeris, “Fairness in network optimal flow control:optimality of product forms,” *IEEE Transactions on Communications* **39**, pp. 775–782, May 1991.
24. C. Douligeris and R. Mazumdar, “A Game Theoretic Perspective to Flow Control in Telecommunication Netowk,” *Journal of the Franklin Institute* **329**, pp. 383–402, 1992.
25. Z. Zhang and C. Douligeris, “Convergence of Synchronous and Asynchronous Greedy Algorithms in a Multiclass Telecommunications Environment,” *IEEE Transactions on Communications* **40**, pp. 1277–1281, August 1992.
26. M.Avriel, *Nonlinear Programming*, Prentice-Hall, Englwood Cliffs,N.J., 1976.
27. L. Brakmo and L. Peterson, “Tcp vegas: end to end congestion avoidance on a global internet,” *IEEE Journal on Selected Areas in Communications* **13**, pp. 1465–80, October 1995.
28. R. J. La, J. Walrand, and V. Anantharam, “ Issues in TCP Vegas .” available at http://www.path.berkeley.edu/˜ hyongla, July 1998.