A Generalized FAST TCP Scheme

Cao Yuan [a], Liansheng Tan [a,b], Lachlan L. H. Andrew [c],
Wei Zhang [a], Moshe Zukerman [d,*],

[a] *Department of Computer Science, Central China Normal University, Wuhan 430079, P.R. China*

[b] *The Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT 0200, Australia*

[c] *Department of Computer Science, California Institute of Technology (Caltech), Pasadena, CA 91125, USA*

[d] *ARC Special Research Center for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010, Australia*

Abstract

FAST TCP has been shown to be promising in terms of system stability, throughput and fairness. However, it requires buffering which increases linearly with the number of flows bottlenecked at a link. This paper proposes a new TCP algorithm that extends FAST TCP to achieve (α, n)-proportional fairness in steady state, yielding buffer requirements which grow only as the nth power of the number of flows. We call the new algorithm Generalized FAST TCP. We prove stability for the case of a single bottleneck link with homogeneous sources in the absence of feedback delay. Simulation results verify that the new scheme is stable in the presence of feedback delay, and that its buffering requirements can be made to scale significantly better than standard FAST TCP.

Key words: FAST TCP, fairness, congestion control, (α, n)-proportional fair FAST TCP, stability, scalability.

* Corresponding author.
Email addresses: `yuancao1980@gmail.com` (Cao Yuan), `liansheng.tan@rsise.anu.edu.au` (Liansheng Tan), `l.andrew@ieee.org` (Lachlan L. H. Andrew), `w.zhang@mail.ccnu.edu.cn` (Wei Zhang), `mzu@unimelb.edu.au` (Moshe Zukerman).

1 Introduction

There is much evidence [4] that the loss-based additive increase/multiplicative decrease (AIMD) algorithm used in TCP [6] does not scale well to high capacity networks. Many new improved versions of TCP have been proposed to solve this problem. These include CUBIC [14], H-TCP [15] and FAST TCP [8]. Recent simulation [8] and experimental [9] studies indicate that FAST TCP is a viable alternative to the currently used loss-based TCP versions.

Many modern congestion control algorithms can be understood as algorithms to solve an optimization problem, in which the network seeks to maximize the sum of the users' "utilities" subject to link capacity constraints. A user's utility is the benefit it derives from transmitting at a given rate. The equilibrium rates are determined by the objective of the optimization, while the dynamics are determined by the optimization procedure. In this framework, users pay a "price" for transmitting data on a congested link; typically either in terms of loss or queueing delay, and the equilibrium value of this price depends on the users' utility functions. As these two price mechanisms have adverse effects on users, it is desirable to use a utility function, which achieves a fair rate allocation and imposes low (and fair) prices on users. This paper adapts the dynamics of FAST [8] to allow it to optimize a more general form of utility function. This allows a tradeoff to be made between fairness and low queueing delay.

Unlike AIMD-based TCP schemes, FAST TCP uses queueing delay as the congestion indication, or price. Users' utilities are logarithmic, making the solution to the optimization problem satisfy the proportional fairness criterion [10]. If all users use FAST, the unique equilibrium rate vector is the unique solution of the utility maximization problem. One drawback of this approach is that the queueing delay (and hence buffer requirements) at a node increase in proportion to the number of flows bottlenecked there.

To allow a tradeoff between fairness and network utilization, Mo and Walrand [13] popularized the concept of (α, n)-proportional fairness, which generalizes max-min fairness [1], proportional fairness [10] and minimum potential delay [12]. This corresponds to a simple family of power-law utility functions. We propose an extended version of FAST TCP, termed *Generalized FAST TCP*, whose equilibrium rates are (α, n)-proportional fair. This is achieved by making a slight change to the window update equation, which implicitly optimizes a suitable utility function. As well as allowing increased fairness, corresponding to $n > 1$, Generalized FAST TCP allows the queueing delay to be reduced at nodes carrying many flows by setting $n < 1$.

Our proposed scheme is a generalization of the existing FAST TCP [8]. Specifically, the behavior of FAST TCP is reproduced by the special case of Generalized FAST with $n = 1$, while other modes of Generalized FAST cannot be achieved simply by tuning FAST TCP parameters. We will show that the new scheme inherits the merits of the current FAST TCP regarding stability and throughput for any value of n and not just for $n = 1$. We also provide stability analysis and prove that Generalized FAST TCP achieves (α, n)-proportional fairness.

The remainder of this paper is organized as follows. In Section 2, we clarify the relationship between the mechanism of FAST TCP and the proportional fairness notion. In Section 3, we describe the new Generalized FAST TCP scheme and discuss the effect of the parameters $\alpha^{1/n}$ and n on buffer occupancy and fairness. In Section 4, we analyze and prove the stability of the new scheme. Section 5 investigates the tradeoff between fairness and the queueing delay experienced by users. In Section 6, we verify by simulations that the new scheme is stable and (α, n)-proportionally fair. Finally, the conclusions are drawn in Section 7.

2 Proportional Fairness and FAST TCP

A general network can be described as a set $L = \{1, \cdots, M\}$ of links, shared by a set $I = \{1, \cdots, N\}$ of flows. Each link $l \in L$ has capacity c_l. Flow $i \in I$ traveled a router L_i consisting of a subset of links, i.e., $L_i = \{l \in L \,|\, i \text{ traverses } l\}$. A link l is shared by a subset I_l of flows where $I_l = \{i \in I| \, i \text{ traverses } l\}$. Let x_i be the rate of flow i and let $x = \{x_i, i \in I\}$ be the rate vector. Let $A = (A_{li}, i \in I, l \in L)$ be the routing matrix, where $A_{li} = 1$ if flow i traverses link l, and 0 otherwise. Throughout this paper, the terms "flow", "sources" and "users" are used synonymously.

A rate vector $x \geq 0$ is called *feasible* if

$$\sum_{i \in I_l} x_i \leq c_l, \ \forall \, l \in L. \tag{1}$$

The notion of fairness characterizes how competing users should share the bottleneck resources subject to the above constraint. A feasible flow rate vector x is defined to be *max-min* fair if any rate x_i can not be increased without decreasing some x_j which is smaller than or equal to x_i [1]. Kelly *et al.* [10] proposed the so-called *proportional fairness.* A rate vector x^* is α_i-weighted proportional fair if it is feasible, and if for any other feasible vector x_i, the aggregate of proportional change is non-positive,

$$\sum_{i\in I}\alpha_i\frac{x_i-x_i^*}{x_i^*}\leq 0, \tag{2}$$

where α_i is positive numbers, $i=1,2,\cdots$.

Consider the following optimization problem (P):

$$\max_{x\geq 0}\sum_{i\in I}U_i(x_i), \tag{3}$$

subject to the constraint given by (1), where U_i is the utility function of user i. We follow the standard approach [10] of taking the Lagrangian

$$L(x;p)=\sum_i(U_i(x_i)-x_iq_i)-\sum_l p_lc_l, \tag{4}$$

where p_l, called the *price* of link l, is the Lagrange multiplier of the constraint due to the capacity of link l. We assume that

$$q_i(t)=\sum_{l=1}^{M}A_{li}p_l(t-\tau_{li}^b) \tag{5}$$

is the aggregate price observed by source i in its path, and link l observes the aggregate source rate

$$y_l(t)=\sum_{i=1}^{N}A_{li}x_i(t-\tau_{li}^f), \tag{6}$$

where τ_{li}^f is the forward feedback delay from source i to link l, and τ_{li}^b is the backward feedback delay from link l to source i. For simplicity, we assume that the feedback delays τ_{li}^f and τ_{li}^b are constants.

For given link prices, each source i determines its optimal rate as

$$x_i(p)=\arg\max_{x_i}U_i(x_i)-x_iq_i=(U_i')^{-1}(q_i). \tag{7}$$

The primal optimization (P) can then be replaced by its dual (D) given by

$$\min_{p\geq 0}\sum_i(U_i(x_i(p))-q_ix_i(p))+\sum_l c_lp_l. \tag{8}$$

According to [10], α_i-weighted proportional fairness is achieved within a system of social welfare maximization, if all users have utility functions of the following form:

$$f_i(x_i) = \alpha_i \log x_i. \tag{9}$$

That is, a α_i-weighted proportional fair vector solves the above optimization problem (P) by maximizing the sum of all the logarithmic utility functions. In this case, (7) becomes

$$x_i = \frac{\alpha_i}{q_i}. \tag{10}$$

For the existing version of FAST TCP, it is known [8] that the source window updating equation (10) has a unique equilibrium point (x_i^*, q_i^*) satisfying (10).

$$w_i(t+1) = \gamma \left(\frac{d_i w_i(t)}{d_i + q_i(t)} + \alpha_i(w_i(t),\, q_i(t)) \right) + (1-\gamma) w_i(t), \tag{11}$$

where

$$\alpha_i(w_i,\, q_i) = \begin{cases} \alpha_i w_i & \text{if } q_i = 0 \\ \alpha_i & \text{otherwise.} \end{cases}$$

Since this equilibrium point is known ([8], Theorem 1) to be the unique optimal solution of the above problem (P) with the specific utility functions given by (9), FAST TCP maximizes the sum of logarithmic utility functions. This implies in particular that the current FAST TCP achieves α_i-weighted proportional fairness. Note that, the fairness parameter α_i is also the number of flow i's packets that are buffered in the routers in its path at equilibrium. If there are N flows, the total number of packets buffered in the routers at equilibrium is $\sum_{i=1}^{N} \alpha_i$ (see [9]). From this, it is seen that the buffer occupancy increases linearly with the number of flows.

3 The Generalized FAST TCP

As a generalization of proportional fairness and max-min fairness, the definition of $(\alpha,\, n)$-proportional fairness is given by Mo and Walrand in [13], which is described as follows. Note that our notation differs slightly from that of [13], so that it corresponds to its usual meaning in the FAST algorithm. A rate vector x^* is (α, n)-proportionally fair, if it is feasible, and if for any other feasible vector x,

$$\sum_{i \in I_l} \alpha_i \frac{x_i - x_i^*}{(x_i^*)^n} \leq 0, \tag{12}$$

where α_i are positive numbers, for $i \in I$. Note that (12) reduces to (2) when $n = 1$. It is also seen that, when n becomes large, the (α, n)-proportional fair rate vector approaches the max-min fair rate vector. Achieving (α, n)-proportional fairness corresponds to maximizing the sum of users' utilities of the form [13]

$$U_i(x_i) = \begin{cases} \alpha_i \log x_i; & n = 1 \\ \alpha_i (1-n)^{-1} x_i^{1-n}; & \text{otherwise.} \end{cases} \tag{13}$$

Thus, from (7), the optimal rates satisfy

$$x_i^* = \frac{\alpha_i^{1/n}}{(q_i^*)^{1/n}}. \tag{14}$$

Generalized FAST TCP seeks to achieve (α, n)-proportional fairness. This is achieved by modifying the window update equation to be

$$w_i(t+1) = w_i(t) + \gamma_i \left(\alpha_i^{1/n} - \frac{(q_i(t))^{1/n}}{d_i + q_i(t)} w_i(t) \right), \tag{15}$$

where $\gamma_i \in (0, 1]$, in appropriate units. It is easily to see that the equilibrium point (x_i^*, q_i^*) of (15) exactly satisfies (14).

In fact, most of the parameters of Generalized FAST TCP have already existed in the original FAST TCP; hence, we do not need to modify the original implementation significantly. As $\alpha_i^{1/n}$ is a constant, and all we need to do is to choose the power of the queueing delay $q_i(t)$. The FAST algorithm is usually expressed in terms of an estimate, denoted $baseRTT$, of the pure propagation delay of flow i, d_i, and of the measured round trip time $d_i + q_i$, denoted by RTT. The queuing delay $q_i(t)$ can then be calculated by $RTT - baseRTT$. Using that notation, and the observation that

$$x_i(t) = w_i(t)/(d_i + q_i(t)). \tag{16}$$

the window update rule for Generalized FAST can be written as the pseudocode

$$w \leftarrow w + \gamma \left(\alpha^{1/n} - \frac{w}{\text{RTT}} (\text{RTT-baseRTT})^{1/n} \right),$$

where again $\gamma \in (0, 1]$ in suitable units. Note that the equilibrium window size, w_i^*, queueing delay, q_i^* and rate, x_i^* of source i are related by $x_i^* = w_i^*/(d_i + q_i^*)$, where d_i are the propagation delay that flow i experienced. Analogously to the analysis of FAST [8], by using the notion of (α, n)-proportional fairness it is straightforward to prove the following theorem.

Theorem 1. *In the case that the routing matrix, A, has full row rank and given the identities (5) and (16), the unique equilibrium point (x^*, q^*) of the system of window updating equations (15) is such that the rate vector $x^* = (x_1^*, \cdots, x_N^*)^T$ is the unique maximizer of the problem P with the utility function given by (13) and the queueing delay vector $q^* = (q_1^*, \cdots, q_N^*)$ is such that $p^* = (p_1^*, \cdots, p_M^*)$ is the unique minimizer of (8).*

The above theorem implies in particular that the equilibrium rate vector determined by the new Generalized FAST TCP proposal achieves (α, n)-proportional fairness.

Now we study the queue size at equilibrium. According to (14),

$$q_i^* = \frac{\alpha_i}{(x_i^*)^n},$$

the backlog of each source kept in the links is

$$b_i = x_i^* \times q_i^* = \frac{\alpha_i}{(x_i^*)^{n-1}}. \tag{17}$$

Under the new Generalized FAST TCP scheme, we consider a dumbbell network with N flows sharing one bottleneck link with capacity C, the total buffer occupancy at equilibrium denoted by B is

$$B = \sum_{i=1}^{N} b_i = \sum_{i=1}^{N} \frac{\alpha_i}{(x_i^*)^{n-1}}. \tag{18}$$

By setting the same value of $\alpha^{1/n}$ for each flow and noting $x^* = C/N$, we can rewrite (17) as

$$B = \frac{N^n}{C^{n-1}} \cdot \left(\alpha^{\frac{1}{n}} \right)^n. \tag{19}$$

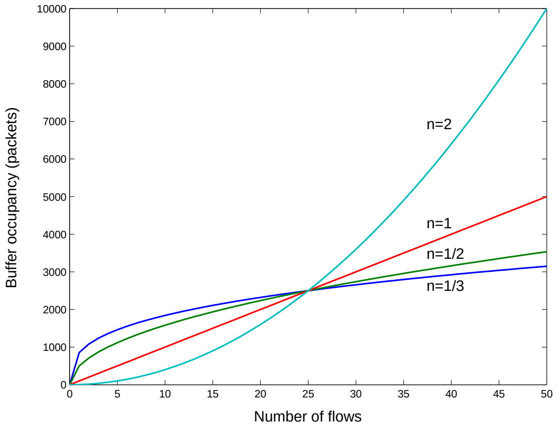

Fig. 1. The relationship between the buffer occupancy and number of flows with $\alpha_i^{1/n} = 100$

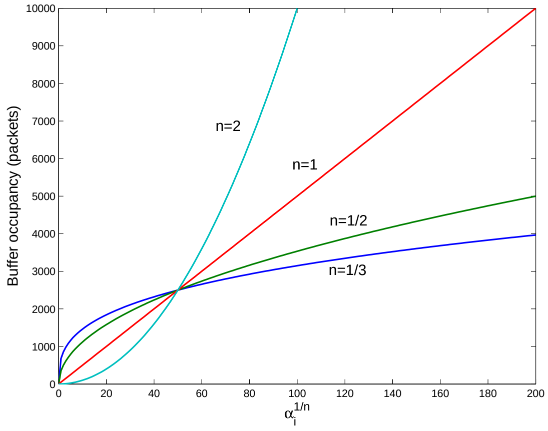

Fig. 2. The relationship between the buffer occupancy and parameter $\alpha_i^{1/n}$ with the number of flows being 50.

When $n = 1$, the summation of (18) is the total number of packets buffered in the router at equilibrium under the original FAST TCP scheme. However, this can be reduced by altering n, which will be discussed further in Section V.

First, let us fix the value of $\alpha_i^{1/n}$ so as to study the relationship between the buffer occupancy and the number of flows under Generalized FAST TCP. To illustrate this relationship, by setting $\alpha_i^{1/n} = 100$ and the router capacity to be 2500 packet/s in a dumbbell network, Figure 1 plots four curves of buffer occupancy changing with the number of flows, which is corresponding to $n = 1/3$, $1/2$, 1, 2, respectively.

Observing Figure 1, one finds that smaller n is, the slower the increase in buffer occupancy as the number of connection increases.

Next, let us fix the number of flows, so as to study the relationship between the buffer occupancy and parameter $\alpha_i^{1/n}$. To illustrate this relationship, by

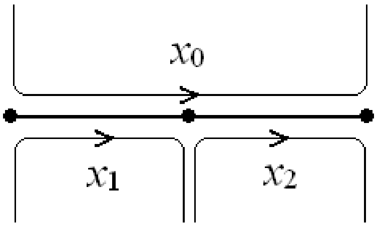

Fig. 3. A linear network

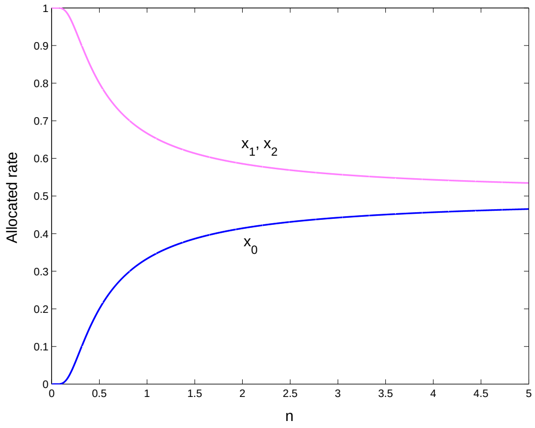

Fig. 4. Bandwidth allocation of linear network

setting the number of flows as 50 and the router capacity as 2500 packet/s in a dumbbell network, we plot Figure 2. In Figure 2, we plot four curves of buffer occupancy changing with the parameter $\alpha_i^{1/n}$, which is corresponding to $n = 1/3$, $1/2$, 1, 2, respectively.

That is to say, buffer occupancy is related with the value of n and is affected by the parameter $\alpha^{1/n}$. However, a certain threshold of $\alpha^{1/n}$ value is necessary for keeping the stability of network [9]. We cannot reduce the buffer occupancy by setting $\alpha^{1/n}$ to 1. Too large value will lead to large buffer occupancy and too small value will result in network instability. So far there is no clear rule for choosing a reasonable value for the parameter $\alpha^{1/n}$.

Although small n can alleviate the increase in buffer occupancy, it also results in unfairness in a multi-bottleneck network. We study the relationship between bandwidth allocation and the value of n using a network with the simplest multi-bottleneck topology as shown in Figure 3. We assume that both multi-bottleneck link capacities are 1. We plot the (α, n) fair bandwidth allocation for this network into Figure 4.

As shown in Figure 4, as n increases, the bandwidth allocation converges to max-min fairness, when $n = 1$, it satisfies the so-called proportional fairness, and as n approaches zero, it becomes the maximum throughput allocation. We will revisit this issue in Section V and provide a more detailed quantitative discussion on the relationship between the fairness and value of n.

4 Stability Analyses

We now analyze the stability of the Generalized FAST TCP under the dumbbell (single bottleneck) topology with N greedy sources, and with equal propagation delays, d. Consider the continuous form of the equation (14),

$$\dot{w}_i(t) = \gamma \left(\alpha_i^{1/n} - x_i(t)(q(t))^{1/n}\right), \ i = 1, \cdots, N. \tag{19a}$$

From $w_i(t) = x_i(t)(d + q(t))$, we have

$$\dot{w}_i(t) = d\dot{x}_i(t) + x_i(t)\dot{q}(t) + \dot{x}_i(t)q(t). \tag{19b}$$

Substituting (19a) into (19b) gives the implicit equation

$$\begin{aligned} \dot{x}_i(t) &= f\left(x_i(t), \dot{x}_i(t), q(t), \dot{q}(t)\right) \\ &= \frac{1}{d}\left[-x_i(t)\dot{q}(t) - \dot{x}_i(t)q(t) + \gamma\alpha_i^{1/n} - \gamma x_i(t)(q(t))^{1/n}\right]. \end{aligned}$$

We now linearize the above equation around the equilibrium point (x_i^*, q^*), by setting

$$\begin{aligned} x_i(t) &= x_i^* + \delta x_i(t) \\ q(t) &= q^* + \delta q(t). \end{aligned}$$

Note that

$$\begin{aligned} \delta\dot{x}_i(t) = \dot{x}_i(t) = {} & \left.\frac{\partial f}{\partial x_i}\right|_* \cdot \delta x_i(t) + \left.\frac{\partial f}{\partial \dot{x}_i}\right|_* \cdot \delta\dot{x}_i(t) \\ & + \left.\frac{\partial f}{\partial q}\right|_* \cdot \delta q(t) + \left.\frac{\partial f}{\partial \dot{q}}\right|_* \cdot \delta\dot{q}(t). \end{aligned}$$

Thus the linearized system becomes

$$\begin{aligned} \delta\dot{x}_i(t) = {} & \frac{1}{d}\left(-\gamma(q^*)^{1/n} \cdot \delta x_i(t) - q^* \cdot \delta\dot{x}_i(t)\right) \\ & -\frac{1}{d}\left(x_i^* \cdot \delta\dot{q}(t) + \frac{\gamma}{n}x_i^*(q^*)^{1/n-1} \cdot \delta q(t)\right). \end{aligned} \tag{20}$$

In the absence of feedback delay, we have

$$\dot{q}(t) = \frac{1}{C}\sum_{i=1}^{N} x_i(t) - 1,$$

giving

$$\delta\dot{q}(t) = \frac{1}{C}\sum_{i=1}^{N}\delta x_i(t) + \frac{1}{C}\sum_{i=1}^{N} x_i^* - 1 = \frac{1}{C}\sum_{i=1}^{N}\delta x_i(t). \tag{21}$$

Taking the Laplace transform of (20) and (21) respectively, we have

$$\begin{aligned}&\left((d+q^*)s+\gamma(q^*)^{1/n}\right)\delta X_i(s)=-x_i^* s\delta Q(s)\\&-\frac{\gamma}{n}x_i^*(q^*)^{\frac{1-n}{n}}\delta Q(s)+(d+q^*)\delta x_i(0)+x_i^*\delta q(0),\end{aligned}\tag{22}$$

and

$$s\delta Q(s)=\frac{1}{C}\sum_{i=1}^{N}\delta X_i(s)+\delta q(0),\tag{23}$$

where $\delta X_i(s)$ and $\delta Q(s)$ denote the Laplace transform of $x_i(t)$ and $q(t)$, respectively.

Considering in (22) and (23),

$$\begin{aligned}&\left((d+q^*)s+\gamma(q^*)^{1/n}\right)\delta X_i(s)\\=&-\frac{x_i^*}{C}\left(1+\frac{\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\right)\sum_{i=1}^{N}\delta X_i(s)\\&+(d+q^*)\delta x_i(0)-\frac{x_i^*\cdot\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\delta q(0).\end{aligned}\tag{24}$$

Denoting $[\delta X(s)]^T=[\delta X_1(s),\delta X_2(s),\cdots,\delta X_N(s)]$ and $F^T(s)=[f_1(s),f_2(s),\cdots,f_N(s)]$, (24) is then rewritten as the following matrix equation:

$$(A+B)\delta X(s)=F(s),\tag{25}$$

where both A and B are N-order square matrices, and

$$A=\begin{bmatrix}(d+q^*)s+\gamma(q^*)^{1/n} & & 0\\ & \ddots & \\ 0 & & (d+q^*)s+\gamma(q^*)^{1/n}\end{bmatrix},$$

$$B=\begin{bmatrix}\frac{x_1^*}{C}\left(1+\frac{\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\right) & \cdots & \frac{x_1^*}{C}\left(1+\frac{\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\right)\\ \vdots & \ddots & \vdots\\ \frac{x_N^*}{C}\left(1+\frac{\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\right) & \cdots & \frac{x_N^*}{C}\left(1+\frac{\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\right)\end{bmatrix},$$

$$f_i(s)=(d+q^*)\delta x_i(0)-\frac{x_i^*\cdot\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\delta q(0),\quad i=1,2,\cdots,N.$$

For simplicity, we denote

$$g(s)=(d+q^*)s+\gamma(q^*)^{1/n},$$

and

$$h_i(s)=\frac{x_i^*}{C}\left(1+\frac{\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\right).$$

By some manipulations, we have

$$
\begin{aligned}
&|A+B| \\
=&\begin{vmatrix} g(s)+h_1(s) & h_1(s) & \cdots & h_1(s) \\ h_2(s) & g(s)+h_2(s) & \cdots & h_2(s) \\ \vdots & \vdots & \ddots & \vdots \\ h_N(s) & h_N(s) & \cdots & g(s)+h_N(s) \end{vmatrix} \\
=&\begin{vmatrix} g(s) & 0 & \cdots & h_1(s) \\ 0 & g(s) & \cdots & h_2(s) \\ \vdots & \vdots & \ddots & \vdots \\ -g(s) & -g(s) & \cdots & g(s)+h_N(s) \end{vmatrix} \\
=&\begin{vmatrix} g(s) & 0 & \cdots & h_1(s) \\ 0 & g(s) & \cdots & h_2(s) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g(s)+\sum\limits_{i=1}^{N} h_i(s) \end{vmatrix} \\
=&\,(g(s))^{N-1}\cdot\left(g(s)+\sum_{i=1}^{N} h_i(s)\right)=0.
\end{aligned}
$$

Therefore, we obtain the characteristic equation of the close-loop system (20) and (21), given as follows:

$$
\begin{aligned}
\Delta(s) = &\left((d+q^*)s+\gamma(q^*)^{1/n}\right)^{N-1} \\
&\cdot\left((d+q^*)s+\gamma(q^*)^{1/n}+1+\frac{\gamma}{s\cdot n}(q^*)^{\frac{1-n}{n}}\right)=0.
\end{aligned}
\tag{26}
$$

Solving (26), we obtain its roots:

$$
s_j = -\frac{\gamma(q^*)^{1/n}}{d+q^*},\ j=1,2,\dots,N-1, \tag{27}
$$

and

$$
\begin{aligned}
s_k = &\frac{-\gamma(q^*)^{1/n}-1}{2(d+q^*)} \\
&\pm\frac{\sqrt{(\gamma(q^*)^{1/n}+1)^2-4(d+q^*)\gamma(q^*)^{\frac{1-n}{n}}/n}}{2(d+q^*)}, \\
&\qquad\qquad k=N,N+1.
\end{aligned}
\tag{28}
$$

where s_N takes the "+" and s_{N+1} takes the "−" of the "±". It is thus seen that all the roots of (26) are in the left-hand plane. Therefore, the system (20) and (21) is stable. This can be summarized as the following theorem.

Theorem 2. The Generalized FAST TCP is locally stable for a single bottleneck link topology in the absence of feedback delay.

Interestly, the above stability is independent of the fairness parameter n. When $n = 1$, it reduces to the case of the usual FAST TCP, the stability of which was discussed in [18, 20]. The simulation results presented in Section VI also demonstrate that there is almost no oscillation in the dynamics of the queue size and rate allocations under Generalized FAST TCP.

5 Fairness-Scalability Tradeoff

The concept of (α, n)-proportional fairness has often been used to investigate the tradeoff between fairness and total throughput (see for example [17]). In the context of Generalized FAST it also provides a tradeoff between fairness and scalability.

In a network with N flows and a single bottleneck running standard FAST, the mean queue size scales linearly with N. Under Generalized FAST, if all flows have the same α, the mean queue size scales as N^n. Specifically, (18) shows that the mean queue size is $(\alpha/C^{n-1})N^n$ for a capacity C, since $x_i^* = C/n$. Thus, setting $n < 1$ causes the queue size to scale better as the number of flows increases.

However, intuition says that reducing n pushes the equilibrium further from max-min fairness, since max-min fairness is the limiting case for large n. In particular, the demand function, which maps the route price to the flow rate, becomes $D(q_i) = (U')^{-1}(q_i) = q_i^{-1/n}$. To see the impact of this, consider a linear network with $N = M + 1$ flows, in which flow 1 traverses all M links, and for $i = 2, \cdots, M + 1$, flow i traverses only link $i - 1$. Thus $q_1 = Mq_i$ for all $i \neq 1$, giving $x_1 = M^{-1/n}x_i$. Setting increases the disparity between flow 1 and the remaining flows.

Let us first consider a suitable way of measuring the fairness of a set of rates. A common measure, advocated by Jain [7], is

$$J(x_1, \cdots, x_N) = \frac{\left(\sum_{i=1}^N x_i\right)^2}{N \sum_{i=1}^N x_i^2} = \left(\frac{\mathcal{M}_1}{\mathcal{M}_2}\right)^2,$$

where $\mathcal{M}_i = (x_1^i + \cdots + x_N^i)^{1/i}$ denotes the ith power mean of the values x_i.

By the power mean inequality ([5], Theorem 16), this maps vectors into the interval (0,1], with a value of 1 denoting all values being equal, and lower values corresponding to less fairness. Note that users do not observe rate directly; rather, they observe the time required to transfer a given amount of data. The least fair allocation is one in which a user receives zero rate, requiring an infinite download time. The severity of this is significantly understated by J. In particular, $J(0, x, \cdots, x) = (N-1)/N$, suggesting that starving one user from a large group is quite fair. It could be argued that starving one user is analogous to a form of admission control, and is hence not necessarily unfair. However, the unfairness in the above scenario is based on the topology of the path, with the longer flow consistently receiving a lower rate. Thus, the user cannot simply give up and expect to be treated fairly on a new attempt. Such consistent discrimination is unfair.

A more appropriate measure may be obtained by applying Jain's measure to the download times instead of the rates, giving $J(1/x_1, \cdots, 1/x_N)$. If one rate tends to zero, this measure tends to $1/N$, which is better indicates the unfairness, but still does not reflect the complete starvation of one source.

An alternative quantitative measure of fairness of rates is the ratio of the harmonic mean to the arithmetic mean,

$$F(x_1, \cdots, x_N) = \frac{N^2}{\left(\sum_{i=1}^N 1/x_i\right)\left(\sum_{i=1}^N x_i\right)} = \frac{\mathcal{M}_{-1}}{\mathcal{M}_1}.$$

By the harmonic-arithmetic mean inequality (see Theorem 16 in [5]), this is again a number in the interval [0,1], and equal to 1 only if all rates are equal. It is more suitable than J because it is 0 if and only if a flow is entirely starved of bandwidth, $x_i = 0$ for some i. Note that this measure is independent of the network topology. For complex topologies, it may be desirable to consider a more sophisticated measure, such as the ratio of the harmonic mean of the rate vector to the harmonic mean of the max-min fair rate vector. However, the function F is sufficient for this example.

In the linear network scenario, the fairness ratio F of the equilibrium rates can be shown to be

$$\frac{M^2 + 2M + 1}{M^2 + M^{1+1/n} + M^{1-1/n} + 1}$$

This is $1 - O\left((\log M)^2/Mn^2\right)$ for large M and large n but $O(M^{1-1/n})$ for large M and small n. This shows that the fairness goes to 0 for small n, indicating that the long flow would be starved of bandwidth if too small a value of n were used.

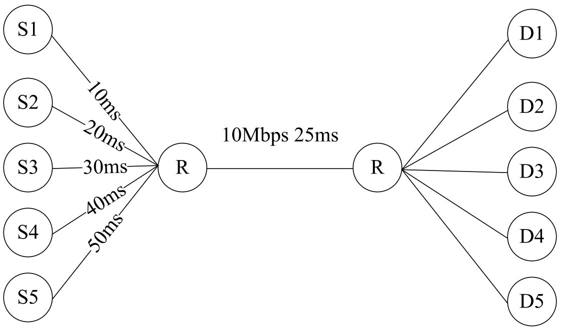

Fig. 5. The simulation model of a dumbbell topology

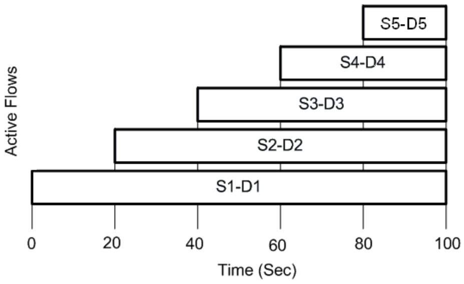

Fig. 6. The active periods of the five flows

Generalized FAST allows the protocol designer to trade fairness (large n) against queue scalability (small n). This may be particularly useful for private networks where knowledge of the topology and expected load is available; if the expected number of flows at any router is expected to be small, then scalability can be traded for increased fairness, while if the diameter of the network is small, then scalability can be improved.

6 Simulation Results

We perform three sets of ns2 [2,19] simulations. The main objective of the first set of simulations is to verify that the buffer occupancy increases more slowly as the number of flows increases for smaller values of n. It also demonstrates that feedback delay does not effect bandwidth allocation. The second set of simulations quantify the reduction in fairness between flows with different numbers of bottleneck links as n decreases. The the third set demonstrate that feedback delay does not effect the queue size.

The first set is for the dumbbell network topology shown in Figure 5 involving the five sender-receiver pairs. In order to study the effect of different feedback delay on sending rate, we set different propagation delay for each flow. The link capacity is set to 1250 packets/s and the one-way propagation delays of the flows are 35, 45, 55, 65 and 75 ms, respectively. The size of every packet is set at 1000 bytes in all the simulations. As shown in Figure 6, we add a new flow

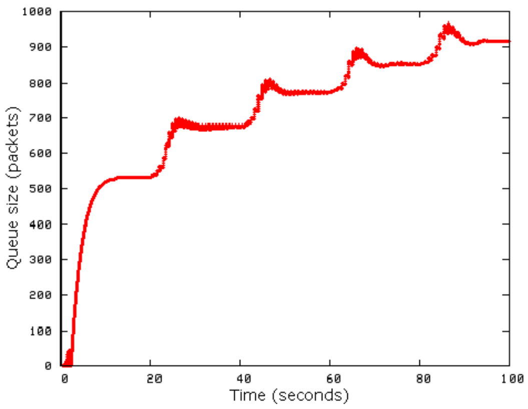

Fig. 7. Queue size for the Generalized FAST TCP with $n = 1/3$, $\alpha^3 = 100$

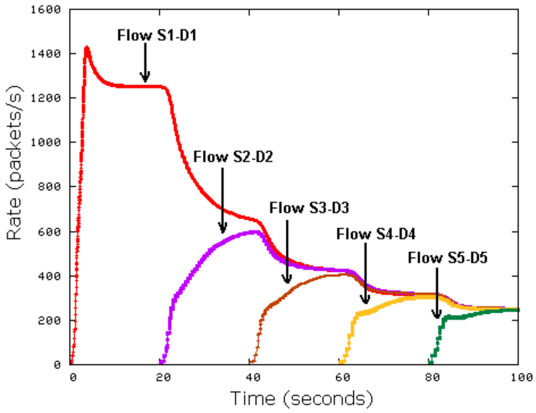

Fig. 8. Sending rate for the Generalized FAST TCP with $n = 1/3$, $\alpha^3 = 100$

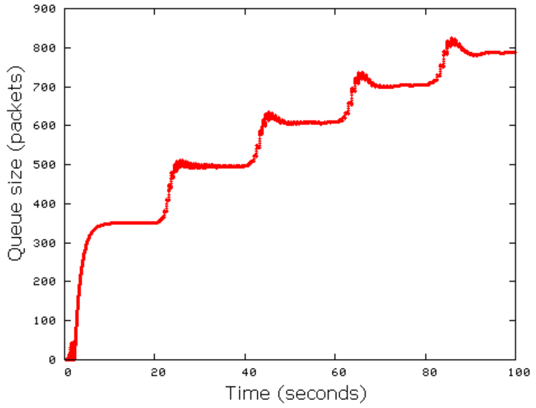

Fig. 9. Queue size for the Generalized FAST TCP with $n = 1/2$, $\alpha^2 = 100$

into the network after each 20 seconds. To avoid the problem of unfairness due to over-estimating *baseRTT* in the presence of persistent congestion [8, 11], we use the priority queue (PQ) method [16] in this scheme. Each flow has the same values of $\alpha^{1/n} = 100$ in the individual simulation. The parameter n for each simulation is set to 1/3, 1/2, and 1, respectively.

From the simulation results (Figures 7, 8, 9, 10, 11, 12), we can see that there are various ways the buffer increases for different values of n. As shown in Figures 7, 9 and 11, buffer requirements grow as the nth power of the number of flows. For $n = 1$, buffer increases linearly with the number of flows bottlenecked at a link, while buffer increases as the square root of the number of flows when $n = 1/2$, and the cube root for $n = 1/3$. Therefore, we can reduce the trend of buffer increment by setting small value of n. However, all flows obtain equal rates, regardless of their RTTs.

The second set of simulations is performed by using a parking-lot topology, which topology is depicted in Figure 13, where there are n long flows which traverse all m links, and m single hop short flows. We assume that the bandwidth of each link is 12500 packets/s. In order to study the variations of the ratio between the rate of short flows and the rate of long flows, we change the hop count under the different values of the parameter n. The simulation results plotted in Figure 14 suggest that the bandwidth allocation in the parking-lot topology is governed by two factors. One is the value of the fairness parameter n and the other is the number of bottleneck hops traversed by the long flow. Figure 14 shows that a smaller value of n will lead to larger difference of rate allocation between the short flows and the long flows if we fix the hop count, while as n increases and goes to infinity, the bandwidth allocation converges to the max-min fair allocation. Figure 14 also shows that the degree of unfairness between short flows and long flows increases with increasing hop count. Specifically if we fix the fairness parameter n, Figure 14 demonstrates that as the hop count becomes larger, the ratio of bandwidth allocation between short flows and long flows also becomes larger.

The third set of results is for dumbbell network topology shown in Figure 15 involving two sender-receiver pairs. In order to study the effect of different feedback delay on queue size in route, we fix one flow's propagation delay at 10 ms, and set the propagation delay for the other flow to 10, 30, 100 and 300 ms. The other settings are the same as for the first set of simulations. From the simulation results in Figure 16, we can see that different feedback delay does not effect the queue size.

7 Conclusion

This paper generalizes the current FAST TCP scheme in such a way that the parameter n and $\alpha^{1/n}$ in the new window update equation can be set to achieve (α, n)-proportional fairness and control the rate of buffer increase. We derived a stability condition for a single bottleneck link in the absence of feedback delay, and we have discussed the tradeoff between fairness and buffer increment.

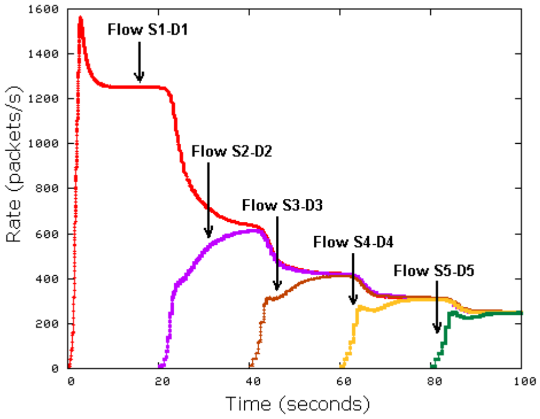

Fig. 10. Sending rate for the Generalized FAST TCP with $n = 1/2$, $\alpha^2 = 100$

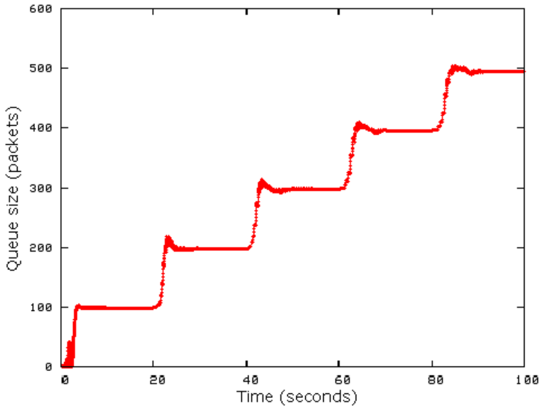

Fig. 11. Queue size for the original FAST TCP $(n = 1)$, $\alpha = 100$

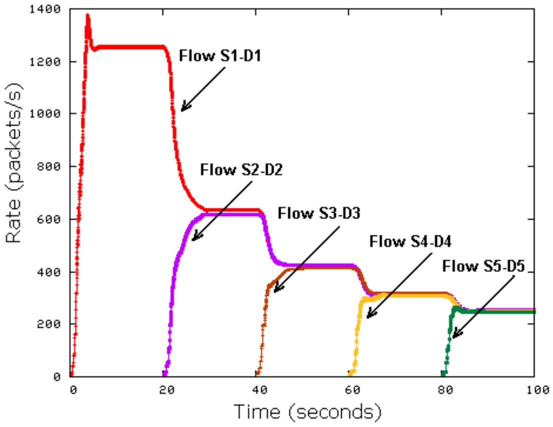

Fig. 12. Sending rate for the original FAST TCP $(n = 1)$, $\alpha = 100$

Future research will investigate the performance of networks with general topology and a variety of traffic scenarios including other TCP versions and UDP traffic under the Generalized FAST TCP with different fairness parameters from design to implementation, and the impact of feedback delay on stability.

Acknowledgements

This research is partially supported by the Program NCET-05-0673, the key project (no. 108166) from Chinese Ministry of Education and partially by the Australian Research Council (ARC).

References

[1] D. Bertsekas, and R. Gallager, Data Networks, Englewood Cliffs, NJ: Prentice-Hall, 1992.

[2] T. Cui, and L. Andrew, FAST TCP simulator module for ns-2, version 1.1. Available from: <http://www.cubinlab.ee.mu.oz.au/ns2fasttcp>.

[3] C.A. Desoer, and Y.T. Yang, On the generalized Nyquist stability criterion, IEEE Transactions on Automatic Control 25 (1980) 187-196.

[4] S. Floyd, HighSpeed TCP for large congestion windows, IETF, Internet Draft 2002. Available from: <http://www.ietf.org/internet-drafts/draft-floyd-tcp-highspeed-00.txt>.

[5] G.H. Hardy, J.E. Littlewood, and G. Pólya, Inequalities, Cambridge University Press, 1934.

[6] V. Jacobson, Congestion avoidance and control, in: Proceedings of ACM SIGCOMM, 1988, pp. 314-329.

[7] R. Jain, The Art of Computer Systems Performance Analysis: techniques for Experimental Design, Measurement, Simulation and Modeling, John Wiley and Sons, Inc., 1991.

[8] C. Jin, D. Wei, and S.H. Low, FAST TCP: motivation, architecture, algorithms, performance, in: Proceedings of IEEE INFOCOM 2004, vol. 4, Hong Kong, 7-11 March 2004, pp. 2490-2501.

[9] C. Jin et al., FAST TCP: from theory to experiments, IEEE Network 19 (1) (2005) 4-11.

[10] F.P. Kelly, A. Mauloo, and D. Tan, Rate control for communication networks: shadow prices, proportional fairness and stability, Journal of Operations Research Society 49 (3) (1998) 237-252.

[11] S.H. Low, L.L. Peterson, and L. Wang, Understanding Vegas: a duality model, Journal of the ACM 49 (2) (2002) 207-235.

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

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

[14] I. Rhee, L. Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant. In *Proc. PFLDnet*, 2005.

[15] R. N. Shorten and D. J. Leith. H-TCP: TCP for high-speed and long-distance networks. in *Proc. PFLDnet*, Argonne, 2004. http://hamilton.ie/net/htcp3.pdf.

[16] L. Tan, C. Yuan, and M. Zukerman, FAST TCP: fairness and queueing issues, IEEE Communications Letters 9 (8) (2005) 762-764.

[17] A. Tang, J. Wang and S. Low, Counter-intuitive throughput behaviors in networks under end-to-end Control, IEEE/ACM Transactions on Networking, 14(2) (2006) 355-368.

[18] K. Jacobsson, L. L. H. Andrew, A. Tang, S. H. Low and H. Hjalmarsson. An Improved Link Model for Window Flow Control and Its Application to FAST TCP *IEEE Transactions on Automatic Control*, 2008

[19] USC/ISI, Los Angeles, CA, The Network Simulator - ns-2. Available from: <http://www.isi.edu/nsnam/ns/>.

[20] J. Wang, D.X. Wei, and S.H. Low, Modeling and stability of FAST TCP, in: Proceedings of IEEE INFOCOM 2005, Miami, FL, March 2005.

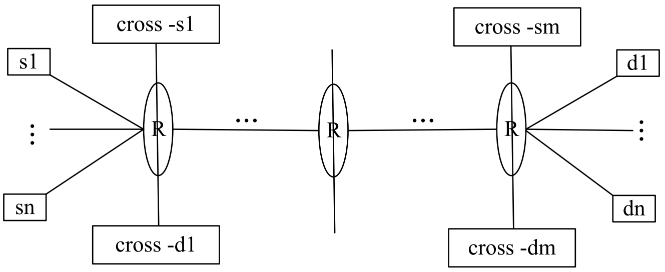

Fig. 13. The simulation model of a parking-lot topology

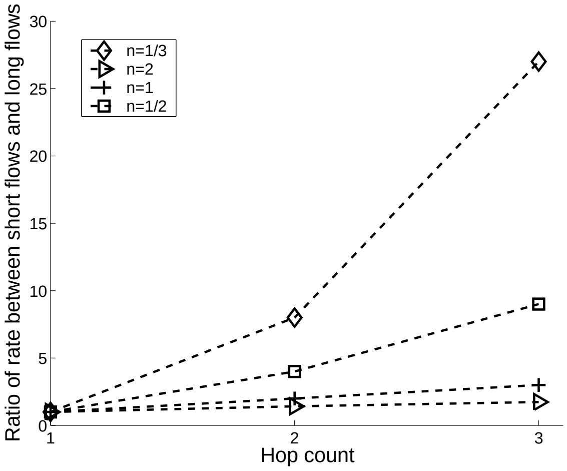

Fig. 14. Ratio of sending rate between short flows and long flows under the parking-lot topology

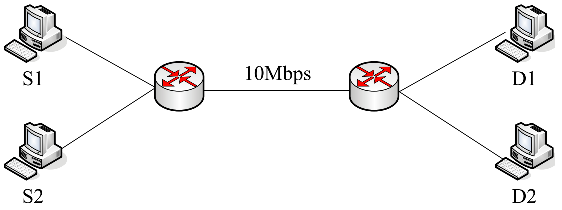

Fig. 15. The simulation model of a dumbbell topology with two flows

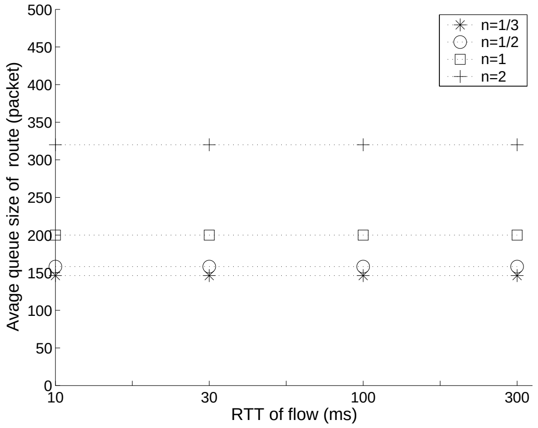

Fig. 16. Ratio of queue size