Computer Communications 31 (2008) 3242–3249

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

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, PR 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, Grattan Street, Melbourne, Vic. 3010, Australia

ARTICLE INFO

Article history:
Received 11 March 2007
Received in revised form 14 May 2008
Accepted 19 May 2008
Available online 6 June 2008

Keywords:
FAST TCP
Fairness
Congestion control
Stability
Scalability

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.

1. Introduction

There is much evidence [3] that the loss-based additive increase/multiplicative decrease (AIMD) algorithm used in TCP [5] does not scale well to high capacity networks. Many new improved versions of TCP have been proposed to solve this problem. These include CUBIC [13], H-TCP [14] and FAST TCP [19]. Recent simulation [19] and experimental [8] 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 [19] 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 [9]. 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 [12] popularized the concept of (α, n)-proportional fairness, which generalizes max-min fairness [1], proportional fairness [9] and minimum potential delay [11]. 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 $n < 1$.

* Corresponding author. Tel.: +61 398879028.
E-mail addresses: yuancao1980@gmail.com (C. Yuan), liansheng.tan@rsise.anu.edu.au (L. Tan), l.andrew@ieee.org (L.L.H. Andrew), w.zhang@mail.ccnu.edu.cn (W. Zhang), mzu@unimelb.edu.au (M. Zukerman).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.05.028

Our proposed scheme is a generalization of the existing FAST TCP [19]. 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, conclusions are drawn in Section 7.

2. Proportional fairness and FAST TCP

A general network can be described as a set $L = \{1, \ldots, M\}$ of links, shared by a set $I = \{1, \ldots, N\}$ of flows. Each link $l \in L$ has capacity c_l. Flow $i \in I$ follows a route 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", "source" and "user" are used synonymously.

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

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

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. [9] 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,

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

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

Consider the following optimization problem (P):

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

subject to the constraint given by (1), where U_i is the utility function of user i. We follow the standard approach [9] of taking the Lagrangian

$$L(x; p) = \sum_i (U_i(x_i) - x_i q_i) - \sum_l p_l c_l, \tag{4}$$

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, in equilibrium,

$$q_i = \sum_{l=1}^{M} A_{li} p_l \tag{5}$$

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

$$y_l = \sum_{i=1}^{N} A_{li} x_i. \tag{6}$$

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

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

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

$$\min_{p \geqslant 0} \sum_i (U_i(x_i(p)) - q_i x_i(p)) + \sum_l c_l p_l. \tag{8}$$

According to [9], 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, an 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 [19] that the rate allocaton (10) is the unique equilibrium point (x_i^*, q_i^*) of

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

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 [19, 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 [8]). 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 (α, n)-proportional fairness is given by Mo and Walrand in [12], which is described as follows. Note that our notation differs slightly from that of [12], 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,

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

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 [12]

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