Available online at www.sciencedirect.com

ScienceDirect

Computer Standards & Interfaces 30 (2008) 52–61

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Enhancement of fairness in a DiffServ network using a novel queuing algorithm

Yi-Hung Huang [a], Kuan-Cheng Lin [b,*], Chao-Yu Kuo [c], Chin-Hsing Chen [d], Yen-Ping Chu [e]

[a] Hsiuping Institute of Technology, Department of Information Networking Technology, Taichung, Taiwan
[b] National Chung Hsing University, Department of Management Information Systems, No. 250, Kuo Kuang RD., Taichung, Taiwan
[c] National Chung Hsing University, Institute of Computer Science, Taichung, Taiwan
[d] Central Taiwan University of Sciences and Technology, Department of Management Information System, Taichung, Taiwan
[e] Tunghai University, Department of Computer Science and Information Engineer, Taichung, Taiwan

Received 19 February 2006; received in revised form 21 December 2006; accepted 2 July 2007
Available online 10 July 2007

Abstract

Core-Stateless Fair Queuing (CSFQ) algorithm was proposed to allocate bandwidth fairly at routers in the Differentiated Services Model (DiffServ Model). However, CSFQ can not effectively provide the fairness between TCP connections and UDP connections. Besides, the setting of buffer size strongly affects the efficiency of CSFQ. In this paper, we present a novel CSFQ-based queuing algorithm. Based on the estimated value of fair share, the probability to drop packet is determined to ensure fair share of bandwidth. The simulated experimental results demonstrate that the proposed scheme outperforms CSFQ.

Keywords: Differentiated Services; CSFQ; Fair Share; DRR

1. Introduction

The Fair Queuing Algorithm can fairly allocate bandwidth to each connection, and effectively protect well-behaved connections from misbehaved connections in the Integrated Service (Interserv) Model [6]. With the help of the Fair Queuing Algorithm, congestion control increases efficiency. However, such mechanisms usually need to maintain state and perform packet scheduling on a per-connection basis, as in Deficit Round Robin, (DRR) [15]; Weighted Fair Queuing (WFQ) [6]; Flow Random Early Drop (FRED) [7], and others. They raise complexity and cause serious problems in implementation, deployment, and scalability.

In the Differentiated Service (DiffServ) model, routers connected to the end systems are called the edge routers; the ones without connecting to the end systems are called the core routers [2,4,8,10,12]. The state information for each connection is stored in edge router and some measured value is labeled in each packet to support the core router to determine whether the packet is to drop or not. Core routers need not store state information for each connection, reducing their operation loads, as in Core-stateless Fair Queuing (CSFQ) [13], Rainbow Fair Queuing (RFQ) [3], Self-Verifying (SV-CSFQ), Adaptive CSFQ (ACSFQ) [14], and others.

Based on the concept of DiffServ model, CSFQ fairly allocates bandwidth. However, it can not protect the bandwidth shared by TCP connections from UDP connections [1]. In this paper, we propose a novel method based on CSFQ. First, the number of connections was estimated to figure out the approximate fairly allocated bandwidth. Then, the output queue arrival rate of the routers was applied to turn the fairly allocated bandwidth to provide the fairness among connections. NS2 simulator [11] was used to verify the proposed method's efficiency in improving the allocation of bandwidth.

The differences between the proposed method and CSFQ are indicated as follows.

(1) Labeling of packets: CSFQ uses exponential average to estimate the connection arrival rate. The proposed method directly measures the connection arrival rate.

* Corresponding author. Tel.: +886 4 22840864x615; fax: +886 4 22857173.
E-mail addresses: ehhwang@mail.hit.edu.tw (Y.-H. Huang), kclin@nchu.edu.tw (K.-C. Lin), eva@cs.nchu.edu.tw (C.-Y. Kuo), chchen@chtai.ctc.edu.tw (C.-H. Chen).

0920-5489/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2007.07.002

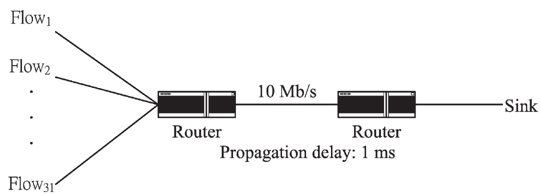

Fig. 1. The topology of the single congested link.

(2) Congestion Control Capability: CSFQ utilizes buffer status to adjust the fair share values. The proposed method utilizes bandwidth utilization to modify the fair share values.

(3) Ability to share bandwidth fairly: CSFQ sets the maximum labeled value out of all packets within current period to be the fair share value in the subsequent period. In this paper, labeled value and connection arrival rate are used to estimate the number of connections for a particular period. Then, the link capacity divided by the number of connections is used to request fair share during the next period for each connection.

Simulated experimental results demonstrate that the proposed scheme is better than CSFQ in terms of fairness. Moreover, the mathematical references and simulated experiments show that the estimated average number of connections approaches the real number of connections.

Section 2 presents a brief outline of the CSFQ algorithm and related issues. Section 3 describes the estimation of the number of connections, and the proposed scheme implemented in the DiffServ service model in detail. Section 4 provides the results of the simulated experiments and analyzes the experimental results to demonstrate that the proposed method is more efficient than CSFQ. Finally, conclusions are drawn, and the paper is summarized.

2. Related research

Stoica, Shenker and Zhang [13] use the exponential average method to measure the average connection arrival rate for each connection in an edge router. For every packet that belongs to the same connection, the connection arrival rate is calculated and labeled in each packet as follows.

$$r_i^{\text{new}} = \left(1 - e^{-T_i^k/K}\right)\frac{l_i^k}{T_i^k} + e^{-T_i^k/K} r_{i,\text{old}}, \qquad (2-1)$$

where r_i^{new} is the current calculated rate for the ith connection; T_i^k represents the interval between the arrival of packet k and that of packet $k-1$ of the ith connection. K is a constant; l_i^k represents the length of the kth packet associated with the i^{th} connection, and $r_{i,\text{old}}$ is the calculated rate in the preceding interval of the ith connection. Then the router will measure the packet arrival rate using another exponential average method. Whenever a packet arrives, the packet arrival rate calculated at the router is as follows:

$$A_{\text{new}} = \left(1 - e^{-T/K_\alpha}\right)\frac{l}{T} + e^{-T/K_\alpha} A_{\text{old}}. \qquad (2-2)$$

T is the interval between the arrival of the current packet and that of the preceding packet; l is the length of the current packet; K_α is a constant, and A_{old} is the value of preceding packet. Similarly, the average packet arrival rate of output queue, F_{new}, from the same router is calculated as follow:

$$F_{\text{new}} = \left(1 - e^{-T/K_\alpha}\right)\frac{l}{T} + e^{-T/K_\alpha} F_{\text{old}}. \qquad (2-3)$$

The router records the maximum packet value r_i^{new} for each measured interval. This value is used in the next period as the *fair-share* bandwidth. However, if A_{new} exceeds the link capacity C for the K_c interval, then congestion is identified and the *fair-share* must be tuned as follows.

$$\text{fair\_share} = \text{fair\_share} \times \frac{C}{F_{\text{new}}}. \qquad (2-4)$$

Then, CSFQ applies the following equation to enqueue or drop a packet;

$$\begin{cases} \text{fair\_share} \geq r_i & \text{Enquence} \\ \text{fair\_share} < r_i & \text{with the probability of} \left(1 - \dfrac{\text{fair\_share}_j}{r_i}\right) \text{to drop packet}' \end{cases} \qquad (2-5)$$

where r_i represents the value of the label of the current packet. In the core router, CSFQ uses Eqs. (2-2), (2-3), (2-4) and (2-5) to enqueue or drop packets.

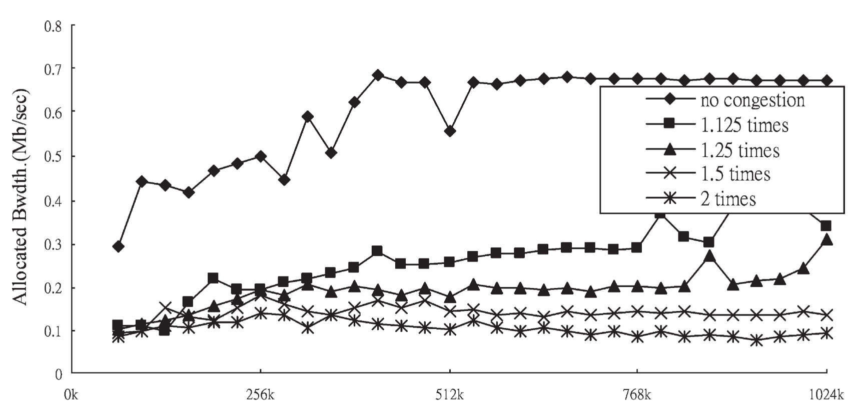

Fig. 2. Using 1 TCP and 31 UDP connection.