

A Fair Bulk Data Transmission Protocol in Grid Environments

Fanjun Su¹, Xuezheng Pan¹, Yonglv², and Lingdi Ping¹

¹ College of Computer Science, Zhejiang University,
Hangzhou 310027, China
suwang@zju.edu.cn

² College of Electrical Engineering, Zhejiang University,
Hangzhou 310027, China

Abstract. In this paper, we propose FHSTCP (Fair High-Speed TCP) as an improvement of HSTCP, which adds a fair factor to eliminate the difference of congestion window caused by different RTT and adopts block-pacing to reduce the burstiness. Simulation results show that FHSTCP can alleviate the RTT unfairness meanwhile keeping advantages of HSTCP.

1 Introduction

Recently there appear many high-speed networks with bandwidth larger than 1Gbps, even than 10Gbps. Through high-speed networks, data intensive grid application can transfer high-bandwidth real time data, images, and video. TCP [1] performs badly in high-speed networks [2]. Some improvements have been made, such as HighSpeed TCP (HSTCP) [2], Scalable TCP (STCP) [3], BIC [4]. However, in [4] the author points out that HSTCP has very severe RTT (Round Trip Time) unfairness. We define the RTT unfairness of two competing flows to be the throughput ratio. In this paper, a fair protocol named FHSTCP (Fair High-Speed TCP) is proposed, and we give a relative fair criterion to evaluate the protocol. FHSTCP adds a fair factor to eliminate the difference of congestion window between flows with different RTT. Block-pacing scheme is adopted to reduce the burstiness caused by fair factor of long RTT flows. The performance of FHSTCP is evaluated using ns2 [5].

2 FHSTCP

TCP and HSTCP use the following algorithm to adjust their congestion window:

$$\text{ACK: } w \leftarrow w + a(w)/w$$

$$\text{Drop: } w \leftarrow w - w \times b(w)$$

Where w denotes congestion window size. For standard TCP, $a(w)=1$, $b(w)=0.5$, which is not sufficient for high-speed networks, so HSTCP makes $a(w)$ and $b(w)$ become the function of current congestion window size [2].

The main idea of FHSTCP is adding a factor to compensate the congestion window increment difference caused by different RTT. Let $a(w)' = \eta \times a(w)$. The fair factor is based on the value of RTT, calculating as $\eta = c \times RTT$. For example, when $c = 10$ and $RTT = 200\text{ms} = 0.2\text{s}$, $\eta = 2$. After adding fair factor, different flows can get the same $w(t)$. Because throughput $V(t) = w(t)/RTT$, FHSTCP can keep RTT unfairness to be inversely proportional to RTT ratio. After fair factor is added, long RTT flows will have a sharp congestion window increment after an RTT. To solve this problem, a block-pacing method is adopted. The congestion window will be divided into several "blocks". After the packets in one block have been sent out, other packets in another block will be sent after a time interval. This can counteract the negative effects caused by adding fair factor. We set the number of the block based on fair factor η . The details of the algorithm are shown in figure 1.

Setting initial value:

```
k=1; // the number of blocks
block=0; // the packets number in a block
number=0; // number of packets that has been sent
```

On receiving a new ack in congestion avoidance state:

```
increment=c*RTT*a(w)/cwnd;
// cwnd is the size of congestion window
if(increment>1)
    increment=1;
    // avoid the increment larger than slow start
cwnd=cwnd+increment;
k=(int)c*RTT+1;
block=cwnd/k;
```

On congestion happening:

```
cwnd=cwnd*(1-b(w));
number=0;
time_0=now; // now is current time
```

On Sending date:

```
if(number>block)
    time_1=now; // now is current time
    if(k>cwnd)
        k=cwnd;
    delay=RTT/k-(time_1-time_0);
    // delay is the time interval between two blocks
    // (time_1-time_0)is sending time of one block
    output(delay); // send data after a time of delay
    number=0; // begin a new count
    time_0=time_1+delay;
else
    output( ); // send data directly
    number++;
```

Fig. 1. Pseudo-codes of FHSTCP

3 Performance Evaluation

We present a relative fairness criterion: (1) The bandwidth requirement of standard TCP should be met. (2) The fairness between different high-speed TCP flows should be guaranteed. Satisfying condition (1) also means the protocol has the TCP friendliness, because standard TCP only works well in high loss rate environment. The loss rate we choose is 10^{-3} , which is corresponding to the value of Low_P in HSTCP. To evaluate the fairness of high-speed TCP flows, fair index [6] is used as follows:

$$f(x) = \left(\sum_{i=1}^n x_i \right)^2 / \left(n \sum_{i=1}^n x_i^2 \right)$$

where x_i is the link utilization of the flow i . We adopt ns2 simulator (version 2.26) [5]. The topology and configuration are shown in figure 2. The buffer size of the router is set to be the product of bandwidth and the delay of bottleneck link. We use TCP SACK for the simulation, and packet size is set to 1000 byte. The maximal congestion window is set to 1000000. FTP is the application used to transmit data through the TCP connections. To avoid phase effect [7], some web flows and short-lived TCP flows are used, together with 3~5 standard long-lived TCP flows. They act as background traffics for the simulation. DT (Drop Tail) queue management policy is used.

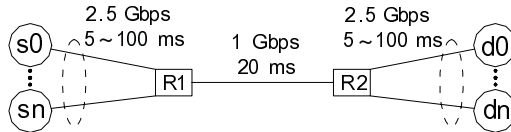


Fig. 2. Simulation topology and configuration

Some simulation results show effect of parameter c . we list our simulation results in table 1(F1, F2, F3 are FHSTCP flows whose RTT are 80, 140, 200ms, and F4 denotes background flows). We can find when $c=5$, the per-flow bandwidth utilization and total bandwidth utilization of FHSTCP flows decrease together. This means that a too small value of c will limit the scalability of FHSTCP. However, when $c=20$, short RTT FHSTCP flows will grasp more bandwidth and fairness are decreased too.

Table 1. The effect of c and the TCP friendliness

| c | Bandwidth Utilization (%) | | | | The fairness index of FHSTCP | Loss rate of background flows |
|-----|---------------------------|-------|-------|-------|------------------------------|--|
| | F1 | F2 | F3 | F4 | | |
| 5 | 29.79 | 16.80 | 14.38 | 26.96 | 0.90 | $7 \times 10^{-5} \sim 1 \times 10^{-4}$ |
| 10 | 32.22 | 26.61 | 19.36 | 16.35 | 0.96 | $7 \times 10^{-5} \sim 8 \times 10^{-4}$ |
| 20 | 44.85 | 24.69 | 19.92 | 5.55 | 0.88 | $5 \times 10^{-4} \sim 1 \times 10^{-3}$ |

What’s more, background flows can get less bandwidth. In other words, the friendliness of FHSTCP decreases. Therefore, $c=10$ is optimal. The packet loss rate of background flows are below 10^{-3} in 3 cases, so we say FHSTCP is TCP friendly.

We also give fairness comparison of FHSTCP, HSTCP, STCP and standard TCP. Three flows with RTT being 80ms, 140ms, and 200ms respectively are used, and different algorithms, such as FHSTCP ($c=10$), HSTCP, STCP, and standard TCP are adopted respectively. The fairness index is calculated at different time scales. As shown in figure 3, we can find FHSTCP has a better fairness property.

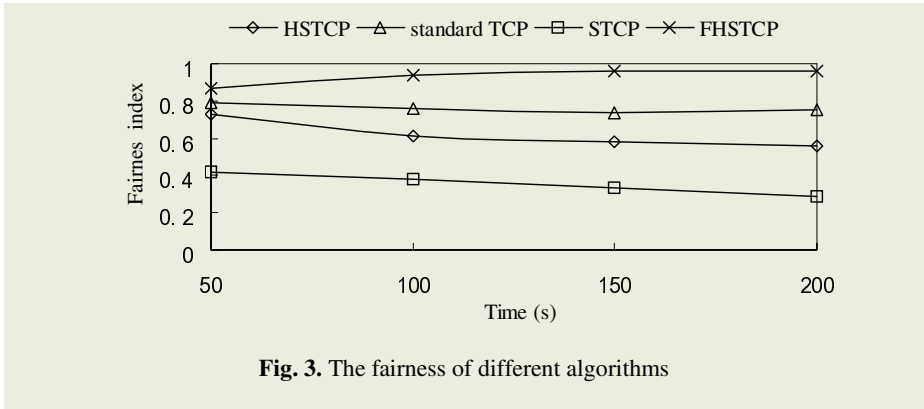


Fig. 3. The fairness of different algorithms

4 Conclusion

As an improvement of HSTCP, FHSTCP adds a fair factor to eliminate the difference of congestion window caused by different RTT and adopts block-pacing to counteract the negative effects caused by fair factor. Simulation results show that FHSTCP has better fairness, and is TCP friendliness while keeping the advantages of HSTCP.

References

1. Stevens, W.R.: TCP/IP Illustrated: The Protocols. Volume 1. Addison-Wesley (1994)
2. Floyd S.: HighSpeed TCP for large congestion windows. RFC 3649 (2003)
3. Kelly T.: Scalable TCP: Improving Performance in High-Speed Wide Area Networks. ACM Computer Communications Review 33 (2003) 83-91
4. Xu L., Harfoush K., Rhee I.: Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks. Proceedings of INFOCOM (2004) 2514 -2524
5. Network simulation-ns2. <http://www.isi.edu/nsnam/ns>
6. Chiu D., Jain R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Journal of Computer Networks and ISDN 17 (1989) 1-14
7. Floyd S., Jacobson V.: On traffic phase effects in packet-switched gateways. Internetworking: Research and Experience 3 (1992) 115-156