A Study on Transport Protocol for High Speed Networking

Yoonjoo Kwon*, Woojin Seok*, Okhwan Byeon*

*Supercomputing Center
*KISTI (Korea Institute of Science and Technology Information)
Email: {yulli, wjseok, ohbyeon}@kisti.re.kr

Abstract *– There are emerging many eScience applications. More and more scientists want to collaborate on their investigation with international partners without space limitation by using these applications. Since these applications have to analyze the massive raw data, scientists need to send and receive the data in short time. So today's network related requirement is high speed networking. The key point of network performance is transport protocol. We can use TCP and UDP as transport protocol but we use TCP due to the data reliability. However, TCP was designed under low bandwidth network, therefore, general TCP, for example Reno, cannot utilize the whole bandwidth of high capacity network. There are several TCP variants to solve TCP problems related to high speed networking. They can be classified into two groups: loss based TCP and delay based TCP. In this paper, I will compare two approaches of TCP variants and propose a hybrid approach for high speed networking*

Keywords: High Speed Networking, Loss based TCP, Delay based TCP

1 Introduction

There are emerging many eScience applications. More and more scientists want to collaborate on their investigation with international partners without space limitation by using these applications. Since these applications have to analyze the massive raw data, scientists need to send and receive the data in short time. So today's network related requirement is high speed networking.

For high speed networking, high capacity network infrastructure and high transferring protocol are necessary. To satisfy high capacity network infrastructure is comparatively easy. Most of international research networks have been upgraded to more than 10Gbps infrastructure. However, only high link speed cannot guarantee high network performance. The key point of network performance is transport protocol. We can use TCP and UDP as transport protocol. In the case of data-intensive applications, we use TCP due to the data reliability. But TCP was designed under low bandwidth network, therefore, general TCP, for example Reno, cannot utilize the whole bandwidth of high capacity network. There are several TCP variants to solve TCP problems related to high speed networking. They can be classified into two groups: loss based TCP and delay based TCP. Loss based TCP, for example BI-TCP, is so aggressive that it can consume the most of available bandwidth in short time but it has some performance degraded problems on loss related overhead. Because loss based TCP increases the congestion window size continuously until loss is occurred, its transferring speed is increased fast but lots of loss cannot help being happened. While delay based TCP, for example FastTCP, has the mechanism to avoid congestion. It monitors the RTT(round trip time) and determines the congestion window size according to RTT variation each time. It does not occur loss frequently but its transferring speed is increased more slowly than loss based TCP.

In this paper, I will compare two approaches of TCP variants and propose a hybrid approach for high speed networking to be aggressive and stable. This paper is organized as follows. In Section 2, we explain some related works about high-bandwidth-required applications and high speed transferring protocols. In Section 3, we introduce our proposed mechanism. Finally we conclude this paper with some future works in Section 4.

2 Related Works

There are many eScience applications requiring high bandwidth. In this section, we describe these applications and TCP variants developed for high speed networking.

2.1 Application Parts requiring for High Network Bandwidth

Year / Application		' 05	' 06 - ' 07	' 08 - ' 09
K-STAR Fusion	Data(TB)	4	200	500
	Link Speed(Gbps)	0.6	5	10
High Energy Physics	Data(TB)	100	800	3000
	Link Speed(Gbps)	5	10	40-100
Visual Astronomy	Data(TB)	6	100	1000
	Link Speed(Gbps)	0.6	5	10

[Table 1] Required Bandwidth each Application Part[4]

According to [Table 1], in HEP(High Energy Physics), Fusion and etc. we can predict they will require more than 10 times present link speed. Therefore, domestic and international research networks are driving forward an increase in network infrastructure bandwidth.

2.2 TCP Variants

These days, we need high throughput transport mechanism. However, TCP was designed under low bandwidth network. In order to increase TCP performance, several TCP variants have been developed. As I mentioned above, they have two approaches : loss-based approach, delay-based approach. In this section we describe BI-TCP of the representative of loss-based TCP and FastTCP of delay-based TCP.

2.2.1 FastTCP

Like TCP Vegas, FastTCP uses queuing delay as its main measure of congestion in its window adjust algorithm. Delay information allows the sources to settle into a steady state when the network is static. Queuing delay also has two advantages as a congestion measure. It provides a finer-grained measure of congestion: The dynamics of delay has the right scaling with respect to link capacity that helps maintain stability as networks scale up in capacity[3, 5,6].

Under normal network conditions, FastTCP periodically updates the congestion window w based on the average RTT according to[3]:

$$w \leftarrow \min\{2w,(1-\gamma)w+\gamma(\frac{baseRTT}{RTT}w+\alpha)\}$$

W : congestion window size

γ : a constant between 0 and 1

RTT : the current average RTT

baseRTT : the minimum RTT observed so far

α : a protocol parameter that controls fairness and the number of packets ecah flow buffered in the network

2.2.2 BI-TCP

BI-TCP uses loss as it main measure of congestion[1]. Unlike Reno, it has three parts of its flow control : binary search increase, additive increase, slow start .

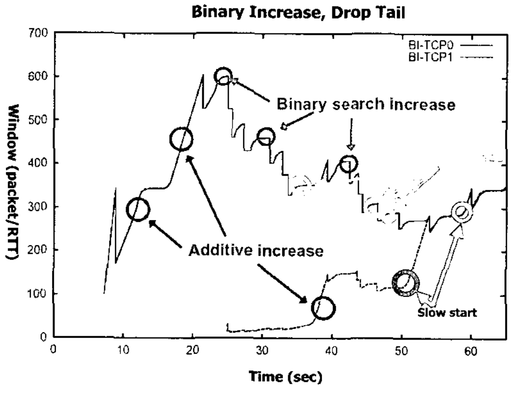
A. Binary Search Increase : If the maximum window size is known, wd can apply a binary search technique to set the target window size to the midpoint of the maximum and minimum.

We shall see, the main benefit of binary search is that it gives a concave response function.

B. Additive Increase : When the distance to the midpoint from the current minimum is too large, increasing the window size directly to that midpoint might add too much stress to the network. When the distance to the current window size to the target in binary search increase is larger than a prescribed maximum step, called the maximum increment(S_{max}) instead of increasing window directly to that midpoint in the next RTT, we increase it by S_{max} until the distance becomes less than S_{max}.

C. Slow Start : After the window grows past the current maximum, the maximum is unknown. At this time, BI-TCP runs a "slow start" strategy to probe for a new maximum. After slow start, it switches to binary increase.

BI-TCP is working like Figure 1.



[Figure 1] BI-TCP in working[1]

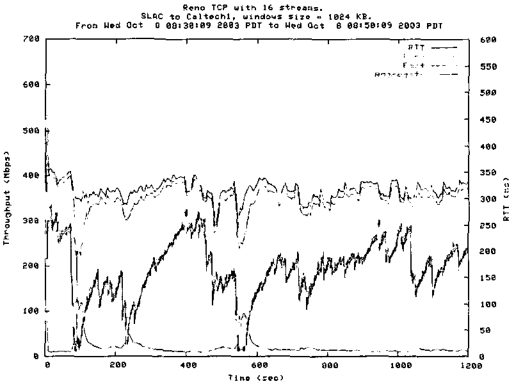
3 The Proposed Mechanism for New TCP variant

The maximum of cwnd size can be increased buffer size the kernel has. However, because appropriate window size depends on both ends situation(ex. Delay, bandwidth..). It is difficult to use the appropriate window size in every case. Therefore, for high speed networking, the modification of transport protocol is more required than system tuning.

As I mentioned above, TCP has two approaches to maintain the cwnd size: loss-based mechanism, delay-based mechanism.

Loss based TCP, for example BI-TCP, uses loss as its measure of congestion and is so aggressive that it can consume the most of available bandwidth in short time. However it has some performance degraded problems on loss related overhead. Because loss based TCP increases the congestion window size continuously until loss is occurred, its transferring speed is increased fast but lots of loss cannot help being happened.

Delay-based mechanism uses queuing delay as a measure of congestion. So it responds more sensitive to network situation than loss-based mechanism. However, if the flow of loss-based TCP and the flow of delay-based TCP exist in same link at same time, the flow of delay-based TCP consumes smaller network bandwidth than that of loss-based TCP, as you can see the figure 2.



[Figure 2] FastTCP & Reno TCP[2]

Proposed TCP mechanism is the hybrid approach of loss-based mechanism and delay-based mechanism. So we propose the method to exhaust network bandwidth aggressively and keep the cwnd size in maximum speed. We choose BI-TCP as the most aggressive mechanism and FastTCP as the most stable mechanism.

Our mechanism uses both delay and loss as a measure of congestion. And it has three steps.

First step : in the case of "no loss", it follows FastTCP mechanism

$$\boxed{cwnd \leftarrow \min\{2cwnd, (1-\gamma)cwnd + \gamma(\frac{baseRTT}{RTT}cwnd - \alpha\}}$$

W : congestion window size

γ : a constant between 0 and 1

RTT : the current average RTT

baseRTT : the minimum RTT observed so far

α : a protocol parameter that controls fairness and the number of packets ecah flow buffered in the network

Second step : in the case that a loss occurs, it follows FastTCP and BI-TCP mechanism. After a loss occurs, it sets the maximum cwnd to be last cwnd before a loss occurs.

$$\boxed{\begin{array}{l} cwnd \leftarrow \min\{binarySearch, additiveIncrease, \\ (1-\gamma)cwnd + \gamma(\frac{baseRTT}{RTT}cwnd + \alpha)\} \end{array}}$$

$binarySearch$: BI-TCP's method.

$additiveIncrease$: BI-TCP's method

$$(1-\gamma)cwnd + \gamma(\frac{baseRTT}{RTT}cwnd + \alpha)$$

FastTCP's method

Third step : after cwnd past target point(maximum cwnd size) and it has no loss. It follows FastTCP.

$$\boxed{cwnd \leftarrow \min\{2cwnd, (1-\gamma)cwnd + \gamma(\frac{baseRTT}{RTT}cwnd + \alpha)\}}$$

Using this mechanism, we can achieve high speed networking.

4 Conclusions

In this paper, we explained eScience applications requiring high bandwidth and TCP variants developed for these applications. And then we described the proposed TCP mechanism to merging loss-based TCP mechanism and delay-based TCP mechanism.

In the future, we will simulate the mechanism and evaluate network performance of the proposed TCP mechanism. And we will complement this mechanism.

References

[1] Lisong Xu, Khaled Harfoush, and Injong Rhee, "Binary Increase Congestion Control for Fast, Long Distance Networks", *INFOCOM 2004* , March 2004, Hong Kong.

[2] TCP Stack testbed :

http://www-iepm.slac.stanford.edu/bw/tcp-eval/caltech/crossInfo.html

[3] C. Jin, D. X. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, S. Singh. "FAST TCP: From Theory to Experiments", IEEE Network, 19(1):4-11, January/February 2005

[4] Technical Report, "User Analysis and Promotion Strategy of Global S&T Collaborative Research Network ", KISTI

[5] Fernando Paganini, Zhikui Wang, John C. Doyle, and Steven H. Low, "Congestion control for high performance, stability and fairness in general networks", IEEE/ACM Transactions on Networking, 2004

[6] Hyojeong Choe and Steven H. Low, "Stabilized Vegas", In Proc. Of IEEE infocom, April 2003