A Novel Congestion Control Algorithm for Large BDP Networks with Wireless Links

Tuan-Anh Le, and Choong Seon Hong
Dept. of Computer Engineering, College of Electronics and Information, Kyung Hee University
{letuanh, cshong}@khu.ac.kr

Abstract

A new TCP protocol can succeed for large bandwidth delay product when it meets network bandwidth utilization efficiency and fair sharing. We introduce a novel congestion control algorithm which employs queueing delay information in order to calculate the amount of congestion window increment in increase phase, and reduces congestion window to optimal estimated bound as packet loss occurs. Combination of such methods guarantees that the proposal utilizes fully network bandwidth, recovers quickly from packet loss in wireless link, and preserves fairness for competing flows mixed short RTT and long RTT. Our simulations show that features of the proposed TCP meet the desired requirements.

1. Introduction

Two main challenges are efficiently utilization of large network bandwidth and fair sharing network resources for successfully designing a TCP protocol for large bandwidth delay product (BDP) networks with wireless links.

The standard TCP (TCP Reno) [1] employs a window growth algorithm to be an additive increase and multiplicative decrease mechanism. The data transmission control mechanism, in high BDP networks, requires TCP connections to maintain very large congestion window. However, it is slow for TCP Reno to obtain full utilization in the network resource if BDP of the TCP connection path is large [2] because a congestion control algorithm of TCP Reno increases the congestion window (*cwnd*) by one packet in every round trip time (RTT) in congestion avoidance phase (CA) since no packet loss occurs, and reduces *cwnd* by half if a packet loss is detected by means of three duplicate ACKs [1].

Specially, TCP connection paths include a wireless link. If packet losses occur by random errors of the wireless link, the sending rate is reduced blindly [5]. That is TCP performance to be degrading unreasonably.

In the recent research reports, these issues have been addressed and many solutions have proposed. These can be classified into four categories: loss-based approaches, delay-based approaches, mixed loss-delay-based approaches, and explicit congestion notification-based approaches. Protocols in loss-based approaches employ packet loss event as an indication of network congestion and try to modify the parameters of TCP Reno congestion control algorithm to be more aggressive in bandwidth utilization in high BDP networks. Such protocols consist of High Speed TCP (HS-TCP) [2], Binary Increase Congestion Control (BIC-TCP) [10], and CUBIC-TCP [13] (an improved BIC-TCP version).

Delay-based approaches try to estimate RTT variations or queueing delay to detect an implicit incipient congestion indication, e.g., FAST TCP [11].

This work was supported by the IT R&D program of MKE/KEIT [10035245: Study on Architecture of Future Internet to Support Mobile Environments and Network Diversity]. Dr. CS Hong is corresponding author.

In mixed loss-delay-based approaches, proposals combine both delay-based and loss-based congestion indications. Compound TCP [14] maintains two different windows: *cwnd* of TCP Reno is updated according to TCP Reno algorithm, the delay window (*dwnd*) increases rapidly when more bandwidth is available, but decreases when bottleneck queue is built-up. The actual congestion window is the sum of *cwnd* and the *dwnd*; TCP-Illinois [12] employs packet loss as event in order to reduce cwnd, and senses queueing delay to determine an increase factor and multiplicative decrease factor during the CA.

EXplicit Control Protocol (XCP) [15] belongs to explicit congestion notification-based approaches. It needs intermediate router to signal to XCP about the degree congestion at the bottleneck. XCP maintains aggressiveness according to the available bandwidth and the feedback delay.

In this paper, we are interested in the mixed loss-delay-based approaches and introduce a congestion control algorithm, which based on queueing delay information and packet loss event as congestion indication.

The rest of this paper is organized as follows: in section 2 we address the shortcoming of the existing protocols, and list our objectives for design a good new protocol for high BDP networks. Section 3 articulates details of the congestion control algorithm. Simulation results are presented in section 4. Finally we conclude in section 5.

2. Motivation

To reduce *cwnd*, loss-based approaches employ packet loss after the full link utilization as congestion indication. After that, they are aggressive in *cwnd* increment to achieve efficiency in high BDP network. However, this causes fluctuation in using queue at routers as well as unfairness in competing TCP flows with different RTTs. On the contrary, delay-based approaches try to adjust *cwnd* around the full link utilization, which is based on queueing delay. However, they cannot share the same link with standard TCP [8], [9], and require large buffer size at router for queueing flows [12]. To design a good new protocol for high BDP networks, in our opinion, following requirements should be considered:

Efficiency: The new protocol should efficiently utilize the large available bandwidth of lager BDP networks with both only wired links and mixed wired-wireless links.

Fairness: In network with shared resources, the new protocol should fairly share the network bandwidth with the competing with TCP flows in the same protocol. Additionally, average TCP flows with different RTTs, their average throughput can be achieved in inverse proportion to their RTTs. The flows with shorter RTT should not obtain shared bandwidth more compared with longer RTT flows.

Our proposal falls in with the same idea of TCP-Illinois and Compound TCP, which are based on packet loss event and queueing delay information. TCP-Illinois and BTBE TCP [3] estimate the maximum RTT and buffer size respectively at bottleneck link. They will be more aggressive when either new shorter path alternates or some packs pass through longer path during TCP connection. In our proposal, queueing delay is employed to calculate the gain increment function. The proposal is simpler in calculation and overcomes wrong aggressiveness estimation.

3. Congestion Control Algorithm

To achieve high utilization in large BDP networks, a TCP congestion control mechanism should be more aggressive to obtain available bandwidth more by increasing *cwnd* large when bottleneck link's bandwidth is under-utilized. However, once the bottleneck link's utilization is becoming full, *cwnd* increment should be smaller. By using queuing delay information allows congestion control to adjust the amount of *cwnd* increment according to the workload of path. And then using congestion signal (e.g. packet loss or ECN) as primary response to reduce *cwnd* can overcome the major weakness of queuing delay-based approaches which are competitive adversely to loss-base approaches. We implement adaptive additive increase (AAI) algorithm via the gain function α(.) instead of fixed value of the gain parameter as in TCP Reno as following

$$Cwnd(t+1) \leftarrow Cwnd(t) + \frac{\alpha(q(t),d)}{Cwnd(t)}$$

$$\alpha(q(t),d) \leftarrow \lambda\left(\frac{d}{q(t)}\right)^{\gamma}$$

where $q(t)$ is round trip queuing delay measured at the sender at time t; d is round trip propagation delay measured as the queue is empty. Parameters λ and γ determine smoothness, aggressiveness and scalability of protocol. Maximum increment value is given by λ. To avoid slowly data rate convergence to fairness, we need to limit on minimum increment that is identical to TCP Reno's increment (i.e. 1.0). In our simulations, we set λ to 20.0 and γ to 4.5.

We employ forward path estimation technique that based on ACK inter-sending time interval compute the available bandwidth (*eBW*) as in [5], [6]. The optimal window (W_{Op}) is defined as

$$W_{Op} = \frac{eBW \cdot d}{Seg\_size}$$

where *Seg_size* is the length of the TCP segment.

Whenever the packet loss is detected through receiving triple duplicate ACKs, the congestion avoidance controller (CAC) updates *ssthresh* and *cwnd*, as follows, and starts a new CA phase.

$$ssthresh \leftarrow \max(W_{Op}, 2)$$

$$cwnd \leftarrow ssthresh.$$

If CAC is triggered by a retransmission timeout event due to the heavy network congestion or very high bit-error rate of wireless link, it sets *ssthresh* to W_{Op} and then sets *cwnd* to one for restarting the slow start phase.

4. Performance Evaluation

We evaluate the proposed TCP, Compound TCP, and TCP-Illinois using the NS-2 [16] with TCP-Linux in [4]. A dumbbell scenario was employed in order to obtain and compare the fundamental features of the protocols such as congestion window evolution, goodput, and fairness.

Such scenario configuration is shown in Fig. 1 with 1500-byte TCP packets, the one way link delay of 50ms, and the bottleneck link speed of 200Mbps, and the router's buffer size of 1600 packets (with the drop tail queue). The simulations were run for 200 seconds.

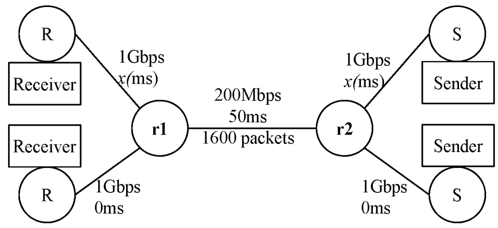

Fig. 1 Dumbbell scenario.

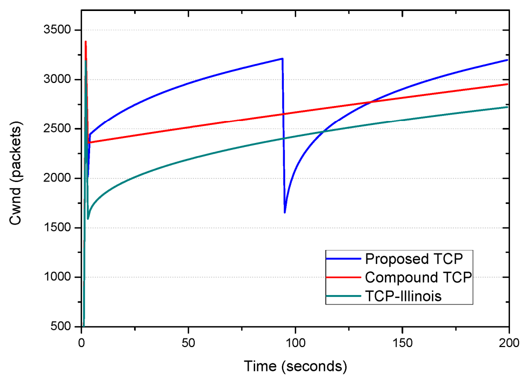

Fig. 2 Congestion window evolution.

The congestion window evolution of single flow is visualized as in Fig. 2. We see that both Compound TCP and TCP-Illinois are slow to utilize fully network bandwidth because they couldn't reach maximum bandwidth at least once within 200 seconds of simulation in CA phase. By the contrary, *cwnd* of the proposed TCP is fluctuated between the optimal estimated window (equals to bandwidth delay product in such single flow) and maximum buffer size at the bottleneck link. Evolution curve of *cwnd* between two consecutive packet loss events is concave. Therefore, combination of such working zone and concave *cwnd* evolution helps us to explain how the proposal can achieve high network bandwidth utilization, fast recovery from packet loss in wireless link while preserving fair sharing as in the rest later.

To investigate TCP performance with wireless links, we compared the goodput of all protocols under random packet loss rate varying from 0% to 10%. In Fig. 3, for any random packet loss rate, the average goodput of the proposed TCP is higher than other TCP protocols. Practically, at 0.01% random packet loss rate, the proposal TCP outperforms Compound TCP and TCP-Illinois by 3.6 times and 5.2 times, respectively. This is interpreted that the proposed TCP can quickly recover from the random packet loss, which is not related to the queueing drop.

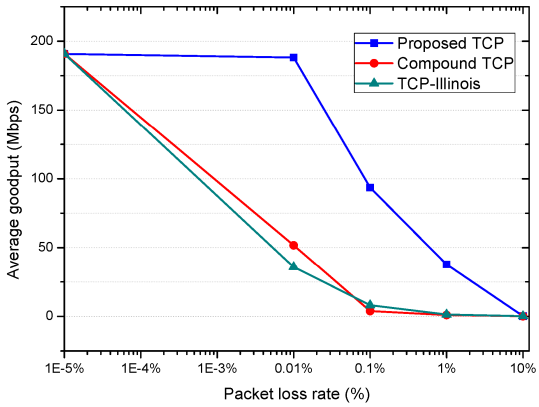

Fig. 3 Goodput vs. random packet loss rates.

We consider fairness as nice sharing the network bandwidth with competing TCP flows in the same protocol. To measure of fairness, we employ Jain's fairness index [7]. The numbers of TCP flows are varied from 2 flows to 8 flows in our simulations. Fig. 4 shows that the proposed TCP can achieve highest fairness indexes in all cases. These values are greater 0.98 over the perfect index of 1.0.

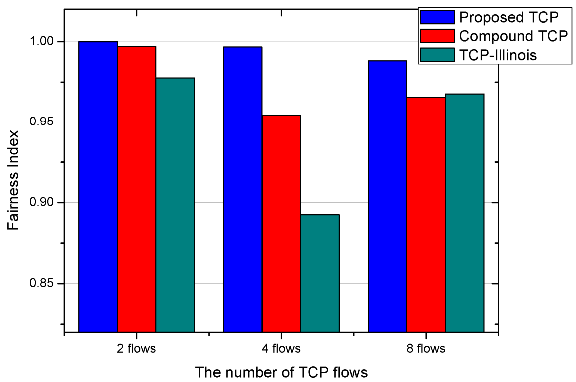

Fig. 4 Fairness index in bottleneck sharing.

TABLE I
THE AVERAGE THROUGHPUT RATIO UNDER DIFFERENT RTTS.

Inverse RTT ratio	**1.5**	**2.0**	**3.0**
Proposed TCP	1.44	1.64	1.92
Compound TCP	6.22	10.8	18.6
TCP-Illinois	3.58	5.06	9.66

RTT unfairness is considered as competing TCP flows with different RTTs, where *cwnd* of shorter RTT TCP flows will increase faster than that of longer RTT TCP flows. Hence the average throughput ratio between shorter RTT flows and longer RTT flows is inversely proportional to their RTT ratio [10]. We employ four TCP flows with the same protocol divided into two 2-flow groups. Group 1 has RTT of 100ms. RTT of group 2 is varied among 150ms, 200ms, and 300ms. Table I shows that Compound TCP has high RTT unfairness as the inverse RTT ratio increase. The proposed TCP is fairest to competing TCP flows with different RTTs.

5. Conclusion

In this paper, we present a novel congestion control algorithm for large BDP networks with wireless link, which meets two desired requirements. The protocol is featured by a gain function and updating *cwnd* to optimal estimated window. Using the gain function based on queueing delay is more aggressive in *cwnd* increment when network bandwidth is more available. Reduction *cwnd* to optimal estimated window facilitates fair sharing and utilization efficiency. The simulation results show that the proposed TCP can achieve bandwidth utilization higher than both Compound TCP and TCP-Illinois, as well as can improve fairness in competing flows mixed short RTT and long RTT.

REFERENCES

[1] V. Jacobson and M. J. Karels, *Congestion Avoidance and Control*, SigCOMM 88, 1988.
[2] S. Floyd, *HighSpeed TCP for Large Congestion Windows*, RFC 3649, December 2003.
[3] Tuan-Anh Le, Choong Seon Hong, "A Bottleneck and Target Bandwidth Estimates-Based Congestion Control Algorithm for High BDP Networks," The International Conference on Information Networking 2010(ICOIN2010), Busan, Korea, January 27-29, 2010.
[4] David X. Wei, Pie Cao, A Linux TCP implementation for NS2 [Online]. Available http://netlab.caltech.edu/projects/ns2tcplinux/ns2linux-2.29-linux-2.6.16/. May 2006.
[5] Tuan-Anh Le, Choong Seon Hong, *TCP BaLDE for Improving TCP Performance over Heterogeneous Networks*, IEICE Transactions on Communications, Vol.E89-B, No.4, pp.1127-1135, April 2006.
[6] Le Tuan Anh, Choong Seon Hong , *Stable Accurate Rapid Bandwidth Estimate for Improving TCP over Wireless Networks*, Lecture Notes in Computer Science 3421(ICN 2005), pp. 141-148, April 2005.
[7] R. Jain, D. Chiu, and W. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer systems*, DEC, Res. Rep.TR-301, 1984.
[8] J. Mo, R. J. La, V. Anantharam, and J. C. *Walrand. Analysis and comparison of TCP Reno and Vegas*. In Proceedings of IEEE INFOCOM, 1999.
[9] A. Tang, J. Wang, S. Low, and M. Chiang. *Equilibrium of heterogeneous congestion control protocols*. In Proceedings of IEEE Infocom, Miami, FL, March 2005.
[10] Lisong Xu, Khaled Harfoush, and Injong Rhee, *Binary Increase Congestion Control for Fast, Long Distance Networks*, IEEE INFOCOM, 2004.
[11] David X. Wei, Cheng Jin, Steven H. Low and Sanjay Hegde, F*AST TCP: motivation, architecture, algorithms, performance* , IEEE INFOCOM, March 2004.
[12] S. Liu, T. Başar, and R. Srikant, *TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks*, Proc. First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), Pisa, Italy, October 11-13, 2006.
[13] Sangtae Ha, Injong Rhee and Lisong Xu, *CUBIC: A New TCP-Friendly High-Speed TCP Variant*, ACM SIGOPS Operating System Review, Vol. 42, No. 5, July 2008, pp:64-74, 2008.
[14] Kun Tan, Jingmin Song, Qian Zhang, and Murari Sridharan, *A Compound TCP Approach for High-speed and Long Distance Networks*, Microsoft Research TechReport, 2005.
[15] Dina Katabi, Mark H, Ley Charlie Rohrs, Mit-lcs Icsi Tellabs, *Congestion control for high bandwidth-delay product networks*, In Proceedings of ACM SIGCOMM 2002, 2002.
[16] NS-2 network simulator [Online]. Available: http://www.isi.edu/nsnam/ns/