Combining Bandwidth Estimate and Explicit Congestion Notification for Improving TCP over Heterogeneous Networks

Le Tuan Anh
Department of Computer Engineering, Kyung Hee University
Giheung-Eup, Yongin-Si, Gyeonggi-Do, 449-701, Korea
Email: letuanh@networking.khu.ac.kr

Choong Seon Hong
Department of Computer Engineering, Kyung Hee University
Giheung-Eup, Yongin-Si, Gyeonggi-Do, 449-701, Korea
Email: cshong@khu.ac.kr

Soyoung Doo
Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
Email: sydoo@etri.re.kr

Abstract – **The challenge for applied TCP over heterogeneous networks is the performance degradation caused by not only network congestion, but also random errors of wireless links. We propose a stable accurate rapid bandwidth estimate (SARBE) algorithm reacting appropriately to Explicit Congestion Notification (ECN), which improves the TCP performance over heterogeneous networks. Our SARBE algorithm takes advantage of estimating the forward bandwidth of connection to improve the estimated bandwidth more accurate than TCP Westwood. By incorporating the estimated bandwidth of SARBE and the incipient congestion notifications of the intermediate routers, our proposal adjusts the packet transmission rate precisely, according to the causes of packet loss.**

Keywords — **Bandwidth estimate, explicit congestion notification, transmission control protocol, wireless.**

1 Introduction

It is well-known that the TCP performance degradation is erroneous in behaviors of the congestion control when the packet loss does not concern the network congestion. In cellular, wireless networks, or combination of wired networks and wireless networks, called heterogeneous networks, the packet losses may be caused by random errors, signal fading or mobile handoff on wireless links, or network congestion.

For the TCP sender, the congestion control probes the available bandwidth of the bottleneck link by continuously increasing the congestion window size (*cwnd*) until reaching the network capacity. When the network congestion is detected by indicating received Duplicate ACKs, the congestion control decreases abundantly to one half of the current *cwnd* setting to the slow start threshold (*ssthresh*). *cwnd* is reset for restarting the slow start phase until retransmission timer is expired. If packet losses occur by random errors of wireless links before *ssthresh* reaches the actual network capacity, *ssthresh* can be haven a smaller value. Therefore the sending rate is reduced blindly. That is the TCP performance is degraded unreasonably.

In this paper, we propose a new bandwidth estimate algorithm reacting appropriately incipient congestion signaling to improve TCP over heterogeneous networks. The simulation results have been shown that when only using SARBE algorithm, SARBE achieves robustness in aspect of stability, accuracy and rapidity of the estimate compared with TCP Westwood, and tolerates ACK compression in the backward path. The combining SARBE and ECN algorithm can improve better performance compared with TCP Reno, TCP Westwood. Furthermore, it is fair in bottleneck sharing and friendly to existing TCP versions.

The rest of this paper is organized as follows. Section 2 summarizes the related works. Section 3 presents our algorithm in detail. Simulation results are presented in section 4. Finally, section 5 is for our conclusion.

2 Related works

To improve TCP performance over wireless networks, several approaches have been proposed and [2] classified into three classes: the link-layer approach, which improves wireless link characteristics or hiding non-congestion caused packet losses from TCP; the split-connection approach, in which a base station separates the wireless connection from the wired connection, and is responsible for retransmission of packet losses on wireless link; the end-to-end approach, which retains TCP semantics, but requires to improve the protocol stack at either the sender side or the receiver side.

Explicit Congestion Notification [1] is an extension to Random Early Detection (RED) [11]. In RED, before router's queue overflows, the end TCP is received incipient congestion signal from the intermediate router by indicating probabilistic packet losses. On the other hand, in ECN, packets are marked instead of being dropped when the

This work was supported by a grant No. R05-2003-12193-0 from Korea Science and Engineering Foundation, and Electronics and Telecommunications Research Institute.

average queue size is between the minimum and maximum thresholds, which are maintained by RED. The intermediate router marks packets via setting congestion experienced (CE) bit in the IP header. When such marked packets are received, the TCP receiver echoes this information to the TCP sender via setting the explicit congestion echo (ECE) bit in the TCP header. Thus the sender is explicitly notified of possible congestion and thereby distinguishing between the losses caused due to congestion and those caused due to random errors.

TCP Westwood scheme [4], [5], [7] is the end-to-end approach. In this scheme, the sender estimates available bandwidth dynamically by monitoring and averaging the rate of ACKs receiving. The sender then updates *cwnd* and *ssthresh* to the estimated bandwidth when fast retransmit or retransmission timeout event occurs. Although, the filter of TCP Westwood is complex, it cannot reflect the rapid changes of the network condition. In addition, as soon as the ACK packets encounter queuing with cross traffic along the backward path, their time spacing is no longer the transmission time upon leaving the queue. These time spacing may be shorter than original time spacing, called ACK compression [9]. In this case, TCP Westwood overestimates the available bandwidth.

3 The proposed algorithm

3.1 Available Bandwidth Estimate

In comparison with TCP Westwood, we take the advantage of using ACKs sending time interval to achieve a more accurate available bandwidth estimate. SARBE employs the ACKs sending time intervals to compute the available bandwidth of the forward path via the timestamp in ACK. In SARBE, the estimate of the forward path is not be affected by ACK compression that results in overestimate.

When the *k*th ACK arrives, the sender simply uses information of the *k*th ACK to compute an available bandwidth sample, which can be written as

$$Bw_k = \frac{L_k}{ts_k - ts_{k-1}} \quad (1)$$

where L_k is the amount of data acknowledged by the *k*th ACK, ts_k is timestamp of the *k*th ACK; ts_{k-1} is the timestamp of the previous ACK arrived at the sender. It can be seen obviously, sample Bw_k represents the current network condition, which faces noises. So the bandwidth estimator has to eliminate transient noise but responds rapidly to persistent changes.

We used the stability-based filter [8] which is similar to the EWMA filter, except using a measure function of the samples' large variance to dynamically change the gain in the EWMA filter. After computing the bandwidth sample Bw_k from "(1)", the stability-based filter can be expressed in the recursive form

$$U_k = \beta U_{k-1} + (1-\beta)\left|Bw_k - Bw_{k-1}\right| \quad (2)$$

$$U_{\max} = \max(U_{k-N}, \ldots, U_{k-1}, U_k)$$

$$\alpha = \frac{U_k}{U_{\max}} \quad (3)$$

$$eBw_k = \alpha \cdot eBw_{k-1} + (1-\alpha)Bw_k \quad (4)$$

where U_k is the network instability computed in "(2)" by EWMA filter with gainβ, β was found to be 0.8 in our simulations; U_{max} is the largest network instability observed among the last N instabilities (N = 8 in our simulations); and eBw_k is the estimated smoothed bandwidth, eBw_{k-1} is the previous estimate and the gain α is computed as "(3)" when the bandwidth samples vary largely.

3.2 The algorithm of combining SARBE and ECN

As mention in Section 1, *ssthresh* represents the probed network bandwidth; while the above estimated bandwidth value also represents the current available bandwidth of download link. Consequently, we have to transform the estimated value into equivalent size of the congestion window for updating *ssthresh*. [5] proposed the interrelation of estimated bandwidth with the optimal congestion window (*oCwnd*) size as

$$oCwnd = \frac{eBw \cdot RTT_{\min}}{Seg\_size} \quad (5)$$

where $RTT_{\min}$ is the lowest Round Trip Time, *Seg_size* is the length of the TCP segment.

As we know from Section 1, the congestion control maintains the packet transmission rate via variables *cwnd* and *ssthresh*. Therefore, the success of an algorithm mainly depends on updating these variables.

We propose the new algorithm by incorporating SARBE and ECN, in which the congestion control can distinguish the packet losses caused due to congestion from those caused due to random errors. Depending on distinguishing the causes of losses, our algorithm adjusts the packet transmission rate precisely according to the estimated bandwidth after new ACK receiving, fast retransmit or transmission timeout event occurs.

The pseudo code of our algorithm is presented following.

A. Algorithm after receiving ACK or Duplicate ACKs

```
if (ACK is received)
    /* signal of incipient congestion */
    if (ECE == 1 and cwnd < ssthresh)
        ssthresh = oCwnd;
    endif
endif

if (n DupACKs are received)
  ssthresh = oCwnd;
  if (cwnd > ssthresh and ECE == 1)
      cwnd = ssthresh;
  endif
endif
```

Whenever the sender receives a new ACK with the sign of incipient congestion, the congestion control updates *ssthresh* to the optimal congestion window size during the slow start phase. Setting precisely *ssthresh* to the accurate bandwidth of bottleneck link leads the sender to enter the congestion avoidance phase opportunely.

When Duplicate ACKs are received, *ssthresh* is set to *oCwnd*. If the packet loss is caused by the network congestion, the congestion control should enter into the congestion avoidance phase by setting *cwnd* to *ssthresh*. Otherwise, it keeps the current *cwnd*.

B. Algorithm after timeout expiration

```
if (retransmission timer expires)
    ssthresh = oCwnd;
    cwnd = 1;
endif
```

If the sender is triggered by a retransmission timeout event due to heavy network congestion or very high bit-error rate of wireless link, its congestion control sets *ssthresh* to the optimal congestion window, and sets the *cwnd* to one in order to restart the slow start phase.

4. Simulation results

Our simulations were run by the NS-2 simulation network tool [6]. We used the recent Westwood module NS-2 [3] for comparison.

4.1 Accuracy of the bandwidth estimator

We first evaluate the stability, accurateness and rapidity of SARBE. The simulation network scenario is depicted in Fig. 1. We used an FTP over TCP and an UDP-based CBR background load with the same packet size of 1000 bytes. The CBR rate varies according to time as the dotted line in Fig 2.

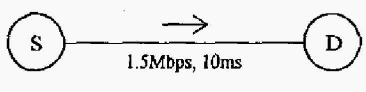

Fig. 1. The single bottleneck link.

The result is shown in Fig. 2; TCP Westwood is very slow to obtain the available bandwidth changes. By contrast, SARBE reaches the persistent bandwidth changes rapidly, which closely follow the available bandwidth changes. This is due to adaptability of dynamic changes of gain α when the bandwidth samples vary largely.

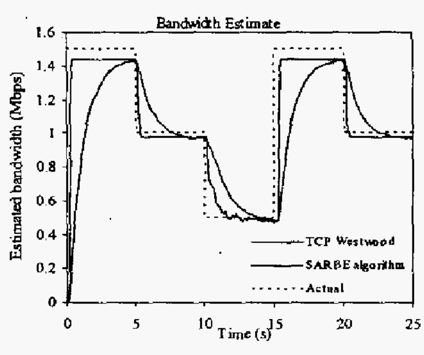

Fig. 2. Comparison of Bandwidth estimate algorithms.

4.2 Impact of ACK compression

To investigate the impact of ACK compression on estimate, we used the network scenario as Fig. 1 and supplemented a traffic load FTP in the reverse direction. The traffic load FTP was started at time 30s and ended at time 120s for 150s simulation time. In this interval, Westwood estimates over 2 Mbps more than SARBE, which is quite near the actual available bandwidth, as in Fig 3.

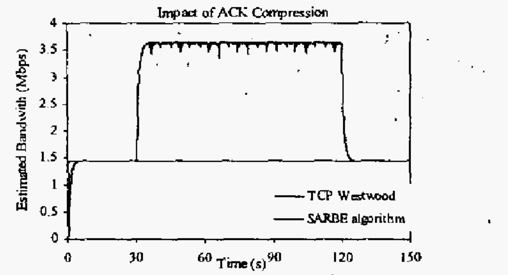

Fig. 3. Overestimated bandwidth of TCP Westwood in the presence of the ACK compression.

4.3 Effectiveness

The simulation was run in a hybrid environment, shown in Fig. 4. The topology includes two RED gateways with an option to set the ECN bit, the bottleneck capacity of 2.5 Mbps, one-way propagation delay of 40 ms, the buffer capacity equal to the pipe size.

We evaluate TCP performance in the lossy link environment. The simulation was performed on one FTP in 150s with the packet size of 1000 bytes, the wireless link random errors ranging from 0.001% to 10% packet loss. In Fig 5, the proposal outperforms other versions for random error rate greater than 0.01%. Particularly, at 1% packet loss rate of wireless link, the proposal achieves better performance than TCP Reno and Westwood by 71.1% and 20.1%, respectively.

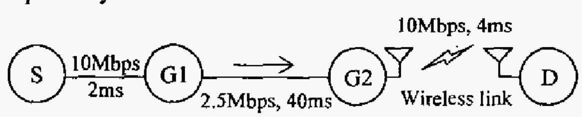

Fig. 4. The heterogeneous network scenario.

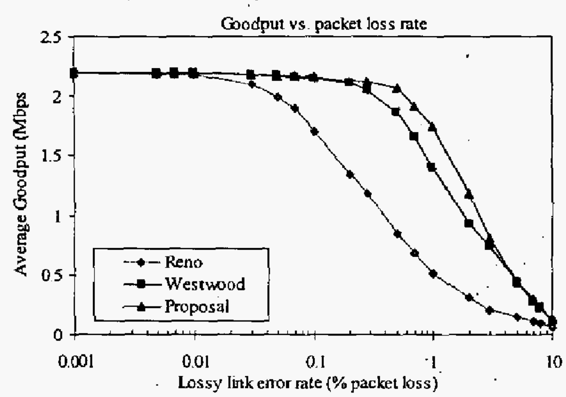

Fig. 5. TCP goodput vs. lossy link error rate.

4.4 Fairness

Another evaluation for TCP is fairness that a set of connections of the same TCP, which can share fairly the bottleneck bandwidth. The index of fairness was defined in [10] as

$$fi = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\left(\sum_{i=1}^{n} x_i^2\right)}, \quad 1/n \le fi \le 1.0$$

where x_i is the throughput of the ith TCP connection, n is the number TCP connections considered in simulation. The fairness index has a range from $1/n$ to 1.0, with 1.0 indicating fair bandwidth allocation.

Using the same scenario as Fig. 4 with ten same TCP connections, we simulated the different TCP versions individually. The comparison result is shown in Fig. 6. TCP Reno is always manifested the best fair sharing. The proposal can achieve quite high fairness index.

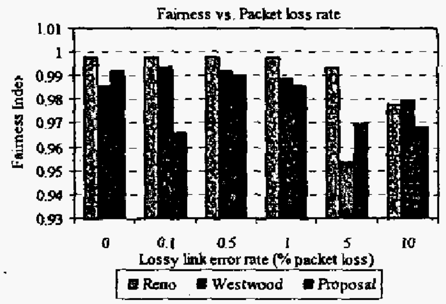

Fig.6. Fairness vs. packet loss rate.

4.5 Friendliness

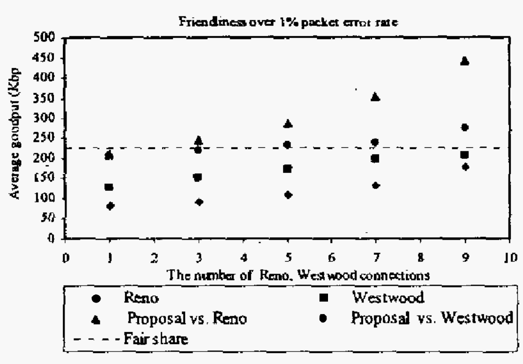

Fig. 7. Friendliness of TCP Reno and Westwood compared with the proposal, respectively, over 1% packet loss of wireless link.

The friendliness of TCP implies fair bandwidth sharing with the existing TCP versions. Our experiments were run on the scenario of Fig. 4. We considered a total of ten connections mixing the proposed TCP with TCP Reno and Westwood at 1% packet loss rate of the wireless link. The x-axis of Fig. 7 represents the number of TCP Reno, Westwood connections; the remaining connections used in the proposed TCP. The dotted line is the fair share. In Fig. 7, the proposal still preserves friendliness with the TCP Reno and Westwood, but outdoes in goodput. This result accords with the above performance evaluation result with the presence of 1% packet loss rate.

5 Conclusion

In this paper, we propose an improved scheme, incorporating our stable accurate rapid bandwidth estimator and congestion notifications of the intermediate router. The scheme can react appropriately to the packet losses in heterogeneous networks, where the losses are caused by either network congestion or random errors of wireless links. Depending on distinguishing the causes of loss, our algorithm adjusts the packet transmission rate precisely according to the estimated bandwidth after new ACK receiving, fast retransmit or transmission timeout event.

In simulation results, the proposed TCP can achieve robustness in aspect of stability, accuracy and rapidity of the estimate in comparison with TCP Westwood, and tolerates ACK compression. It also improves better performance compared with TCP Reno, TCP Westwood. Furthermore, it is fair in bottleneck sharing and friendly to existing TCP versions.

References

[1] S. Floyd, "TCP and explicit congestion notification," ACM Comput.Commun. Rev., vol. 24, pp. 10–23, Oct. 1994.

[2] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," IEEE/ACM Trans. Networking, vol. 5, no. 6, pp. 756–769, 1997.

[3] TCP Westwood - Modules for NS-2 [Online]. Available: http://www.cs.ucla.edu/NRL/hpi/tcpw/tcpw_ns2/tcp-westwood-ns2.html, 2004.

[4] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi,and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," in Proc. ACM MobiCom 2001, Roma, Italy, pp. 287–297, July 2001.

[5] S. Mascolo, C. Casetti, M. Gerla, and S.S. Lee, M. Sanadidi, "TCP Westwood: Congestion Control with Faster Recovery," UCLA CS Technical Report #200017, 2000.

[6] NS-2 network simulator [Online]. Available: http://www.isi.edu/nsnam/, 2004.

[7] S. Mascolo, M.Y. Sanadidi, C. Casetti, M. Gerla, and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," Wireless Networks J., vol. 8, pp. 467-479, 2002.

[8] M. Kim and B. D. Noble, "SANE: stable agile network estimation," Technical Report CSE-TR-432-00, University of Michigan, Department of Electrical Engineering and Computer Science, Ann Arbor, MI, August 2000.

[9] L. Zhang, S. Shenker, and D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic," Proc. SIGCOMM Symp. Comm. Architectures and Protocols, pp. 133-147, Sept. 1991.

[10] 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.

[11] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, V.1 N.4, , p. 397-413, August 1993.