

# Enhanced TCP with End-to-End Bandwidth and Loss Differentiation Estimate over Heterogeneous Networks<sup>\*</sup>

Le Tuan Anh and Choong Seon Hong

Computer Engineering Department, Kyung Hee University  
1, Seocheon, Giheung, Yongin, Gyeonggi 449-701, Korea  
letuanh@networking.khu.ac.kr, cshong@khu.ac.kr

**Abstract.** The TCP performance degradation over heterogeneous networks is caused by not only network congestion, but also random errors of wireless links. We propose an end-to-end stable accurate rapid bandwidth estimate (SARBE) algorithm reacting appropriately to the end-to-end loss differentiation estimate algorithm (LDED), which can improve the TCP performance over heterogeneous networks without the incipient congestion notifications of the intermediate routers. LDED detects the signal of incipient congestion to lead the sender to enter the congestion avoidance phase opportunely before router's queue overflows. As well as relying distinguishing ability 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 events.

## 1 Introduction

Wireless and mixed wired-wireless environments are becoming more popular in recent years. The original TCP assumes that every packet loss as an indication of network congestion, which may not apply to heterogeneous networks because packets may be lost by random errors, signal fading or mobile handoff on wireless links. Therefore, in mixed wired and wireless environments, poor performance of TCP is erroneous in behaviors of the congestion avoidance when the packet loss doesn't concern the 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 (SS) until retransmission timer is expired. If packet losses occur by random errors of wireless links before *ssthresh*

---

<sup>\*</sup> This work was supported by University ITRC Project of MIC. Dr. C.S.Hong is corresponding author.

reaches the actual network capacity, *ssthresh* may be obtained a smaller value. Therefore the sending rate is reduced blindly. That is the TCP performance is degraded unreasonably.

In this paper, we are interested in the end-to-end mechanism, in which bandwidth estimate algorithm reacts appropriately incipient congestion signal estimated from the end-to-end loss differentiation algorithm to improve TCP over heterogeneous networks.

The rest of this paper is organized as follows: Section 2 summarizes the related work. Section 3 presents in detail SARBE, and incorporating SARBE and LDED. Simulation results are presented in section 4. Finally, section 5 is for our conclusion.

## 2 Related Work

There are several approaches proposed for improving TCP performance over wireless networks. They was classified into three classes [2]: the link-layer approach improves wireless link characteristics; 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 changes of the protocol stack at either the sender side or the receiver side.

TCP Westwood [4], [5] monitors and averages available bandwidth for every ACK arrived at the sender. The estimated bandwidth is used to control transmission rate via setting *cwnd* and *ssthresh* to the estimated bandwidth after fast retransmission or transmission timeout. Although, the filter of TCP Westwood is complex, it cannot reflect the rapid changes of the network condition. In addition, if ACK packets encounter the network congestion along the backward path, called ACK compression [9], TCP Westwood overestimates the available bandwidth.

The end-to-end loss differentiation proposals categorize the packet losses explicitly through different estimate without any support from the intermediate routers, such as Flip Flop [7], Vegas [11] and Non Congestion Packet Loss Detection (NCPLD) [12]. They are based on the TCP state variables and information of ACKs to estimate the reason of packet losses. NCPLD categorizes the nature of the error by detecting the knee point of the throughput-load bend.

The Vegas predictor measures the lowest Round Trip Time ( $RTT_{min}$ ) during the TCP connection and computes the expected throughput ( $cwnd/RTT_{min}$ ). When the sender receives ACK, it computes the actual throughput ( $cwnd/RTT$ ). [11] defined extra packets between two thresholds  $\alpha$  and  $\beta$  in the network as

$$D_{Vegas} = RTT_{min} \times \left( \frac{cwnd}{RTT_{min}} - \frac{cwnd}{RTT} \right) \quad (1)$$

If  $D_{Vegas} \geq \beta$ , the Vegas predictor detects the network becoming congestion. Otherwise, if  $D_{Vegas} \leq \alpha$ , there are more available bandwidth for connection. In

the other hand, the network state is kept the same as in the last estimate when  $\alpha < D_{Vegas} < \beta$ .

The parameters  $\alpha = 1$  and  $\beta = 1$  is not accurate that proved in [11]. The authors of [10] then showed that the predictor achieves the highest accuracy if  $\alpha = 1$  and  $\beta = 3$ .

### 3 Proposal

#### 3.1 Available Bandwidth Estimate

In stable accurate rapid bandwidth estimate (SARBE) algorithm. The ACKs sending time intervals are used to compute the available bandwidth of the forward path via the timestamp of ACK. The estimate of the forward path is not be affected by ACK compression that results in overestimate.

To estimate the current bandwidth by observing the pattern of bandwidth for consecutive packets, this can be written as

$$Bw_k = \frac{L_k}{ts_k - ts_{k-1}} \quad (2)$$

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.

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 (2), the stability-based filter can be expressed in the recursive form

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

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

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

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

where  $U_k$  is the network instability computed in (2) by EWMA filter with gain  $\beta$ ,  $\beta$  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  $\alpha$  is computed as (3) when the bandwidth samples vary largely.

We 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(a).

The result is shown in Fig. 2(a); 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.

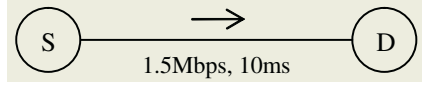


Fig. 1. Single bottleneck link

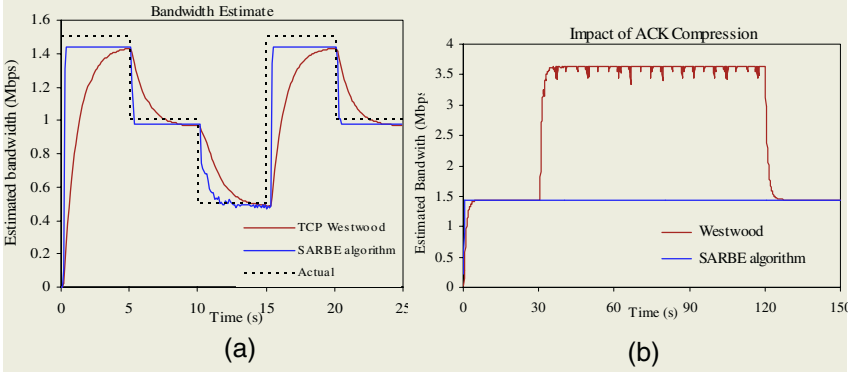


Fig. 2. (a) Comparison of Bandwidth estimate algorithms, (b) Overestimated bandwidth of TCP Westwood

This is due to adaptability of dynamic changes of gain when the bandwidth samples vary largely.

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 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 2(b).

The TCP’s *ssthresh* represents the probed network bandwidth; while the above estimated bandwidth value also represents the current available bandwidth of forward path. Consequently, we have to transform the estimated value into equivalent size of the congestion window for updating *ssthresh*. [5] proposed the interrelation of the estimated bandwidth with the optimal congestion window size (*oCwnd*) as

$$oCwnd = \frac{eBw \cdot RTT_{min}}{Seg\_size}$$

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

### 3.2 Enhanced TCP with Incorporating SARBE and LDEA

In our design, we propose a new scheme by incorporating SARBE and LDEA. For LDEA, we apply the equation (1) to detect the network becoming congestion for every ACK arrived at the sender. Accordingly, the sender can distinguish the packet losses caused due to congestion from those caused due to random

errors of wireless links. And then, relying distinguishing the causes of losses, our scheme 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)
  /* calling the loss differentiation estimate algorithm */
  if (cwnd < ssthresh and isIncipientCongestion== true)
    ssthresh = oCwnd;
  endif
endif
if ( n DupACKs are received)
  ssthresh = oCwnd;
  /* the packet loss is caused by congestion */
  if (isIncipientCongestion == true)
    if (cwnd > ssthresh )
      cwnd = ssthresh;
    endif
  else /* the packet loss is not caused by congestion */
    /* keeping the current cwnd */
  endif
endif
endif

```

Whenever the sender receives a new ACK with incipient congestion, the congestion control updates *ssthresh* to *oCwnd* during the slow start phase (SS). Setting precisely *ssthresh* to the available bandwidth of bottleneck link leads the sender to enter the congestion avoidance phase (CA) opportunely before router's buffer overflow.

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

*B. Algorithm after timeout expiration*

```

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

```

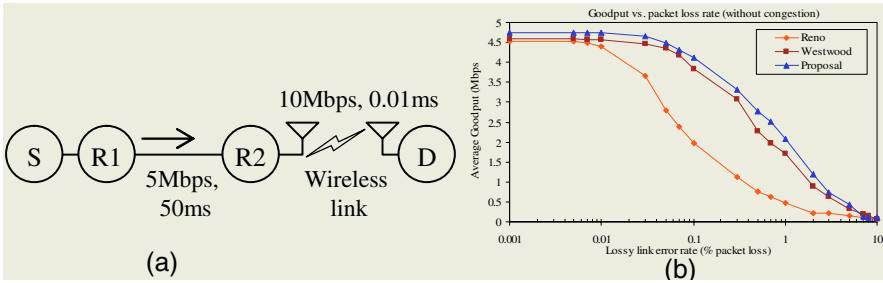
If the sender is triggered by the retransmission timeout event due to the heavy network congestion or very high bit-error rate of wireless link, the congestion control sets *ssthresh* to *oCwnd* and then sets *cwnd* to one for restarting the SS phase.

## 4 Simulation Results

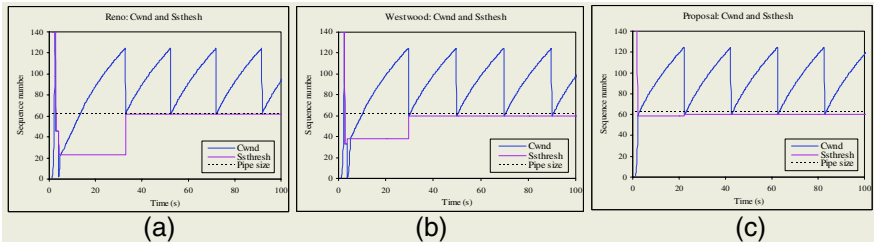
All of our simulations were run by the NS-2 simulation network tool [6]. We used the recent Westwood module NS-2 [3] for comparison.

### 4.1 Effectiveness

The simulation was run in a simple hybrid environment, shown in Fig. 3(a). The topology includes the bottleneck capacity of 5 Mbps, one-way propagation delay of 50 ms, the buffer capacity equal to the pipe size, and a wireless link.



**Fig. 3.** (a) Single bottleneck link; (b) Comparison of Bandwidth estimate algorithms



**Fig. 4.** (a) *Cwnd* and *ssthresh* of Reno, (b) *Cwnd* and *ssthresh* of TCP Westwood, (c) *Cwnd* and *ssthresh* of the proposed TCP in the absence of random errors

We evaluate TCP performance in the lossy link environment. The simulation was performed on one FTP in 100s with the packet size of 1000 bytes, the wireless link random errors ranging from 0.001% to 10% packet loss. In Fig 3(b), for any random error rate, the goodput of the proposed TCP is better than other versions. Particularly, at 1% wireless link packet loss rate, the proposal can achieve better performance than TCP Reno and Westwood by 76.6% and 17.9%, respectively.

Outperforming of the proposal at any random error rate less than 0.001% can be explained by the different behavior of the three protocols shown in Fig 4. In Fig 4(a), (b), at the beginning of the TCP connections, TCP Reno and TCP

Westwood increase exponentially their *cwnds* to probe the network capacity. Until the router’s buffer was overflowed, and then the retransmission timeout events occur, they set *ssthresh* to one-half of the current *cwnd* for TCP Reno, to the estimated bandwidth for TCP Westwood, and restart the SS phase. In contrary, relying the incipient congestion signal of LDEA, the proposed TCP can update *ssthresh* to the estimated bandwidth in the SS phase. This leads the sender to enter the CA phase opportunely before router’s queue overflows, shown in Fig 4(c).

### 4.2 Fairness

The fairness of TCP depicts the fair share ability of the bottleneck bandwidth with multiple connections of the same TCP version. The fairness index is used to assess the convergence of TCP. It was proposed in [1] as following

$$f_i = \frac{(\sum_{i=1}^n x_i)^2}{n(\sum_{i=1}^n x_i^2)}$$

where  $x_i$  is the throughput of the  $i$ th 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. 3(a) with ten same TCP connections, we simulated the different TCP versions individually. The buffer capacity of bottleneck link is equal to the pipe size. The comparison result is shown in Fig. 5(a). The proposed TCP, TCP Reno and TCP Westwood can achieve high fairness index.

### 4.3 Friendliness

The friendliness of TCP implies fair bandwidth sharing with the existing TCP versions. 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. 5(b) represents the number of TCP Reno, Westwood connections; the remaining connections used in the proposed TCP. In Fig. 5(b), the proposal proves the coexistent ability with the TCP Reno, but outdoes in goodput.

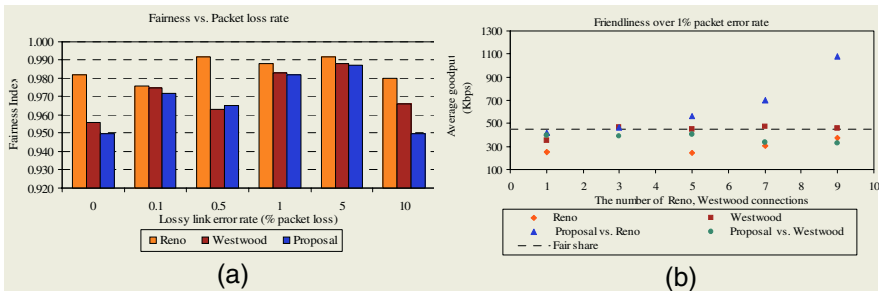


Fig. 5. (a) Fairness vs. packet loss rate; (b) Friendliness of TCP Reno and Westwood compared with the proposal, respectively, over 1% packet loss of wireless link

## 5 Conclusion

By incorporating the stable accurate rapid bandwidth estimator and the loss differentiation estimator, our proposal 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. LDED detects the network becoming congestion to lead the sender to enter the CA phase opportunely before router's queue overflows. As well as relying on distinguishing ability the causes of loss, our algorithm adjusts the packet transmission rate precisely according to the estimated bandwidth obtained from SARBE, after new ACK receiving, fast retransmit or transmission timeout events.

## References

1. R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC, Rep.TR-301, 1984.
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. 756769, 1997.
3. TCP Westwood - Modules for NS-2 [Online]. Available: [http://www.cs.ucla.edu/NRL/hpi/tcpw/tcpw\\_ns2/tcp-westwood-ns2.html](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. 287297, July 2001.
5. S. Mascolo, C. Casetti, M. Gerla, and S.S. Lee, M. Sanadidi, "TCP Westwood: Congestion Control with Faster Recovery," UCLA CS Tech. Report. #200017, 2000.
6. NS-2 network simulator [Online]. Available: <http://www.isi.edu/nsnam/>, 2004.
7. D. Barman and I. Matta, "Effectiveness of Loss Labeling in Improving TCP Performance in Wired/Wireless Networks," Boston University Technical Report, 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. S. Bregni, D. Caratti, and F. Martigon, "Enhanced Loss Differentiation Algorithms for Use in TCP Sources over Heterogeneous Wireless Networks," in *IEEE Global Communications Conference, Globecom 2003*, Dec. 2003.
11. S. Biaz and N. H. Vaidya, "Distinguishing Congestion Losses from Wireless Transmission Losses: A Negative Result," *Seventh International Conference on Computer Communications and Networks (IC3N)*, New Orleans, Oct. 1998.
12. N.K.G. Samaraweera, "Non-Congestion Packet Loss Detection for TCP Error Recovery using Wireless Links," *IEE Proceedings Communications*, volume 146 (4), pages 222-230, August 1999.