Asia-pacific Journal of Multimedia Services Convergence with Art, Humanities and Sociology
Vol.2, No.2 (2012), pp. 47-62
http://dx.doi.org/10.14257/AJMSCAHS.2012.12.01

Enhancement of TCP congestion control based on relative delay and Bandwidth Estimation

Kokula Krishna Hari K, Long CAI[1)]

Abstract

The traditional TCP assumes that all packet losses result from the congestion even in networks with high bit error rates (BER), and this causes a significant performance degradation. In this paper, we propose a novel end-to-end congestion control mechanism called TCP VenoDB, which joins the relative delay and bandwidth estimation together as a judgment condition to distinguish the network states, and refines the traditional fast retransmit and fast recovery phase with partial ACK mechanism. Simulation results by NS-2 show that TCP VenoDB provides more significant performance improvement in throughput, bandwidth utilization and fairness than TCP Veno, TCP Westwood and TCP NewReno in networks with high BER, and has minimal influence of reverse link congestion. Furthermore, TCP VenoDB is friendly towards TCP NewReno for practical purpose.

Keywords : congestion loss, random loss, TCP Veno, relative delay, bandwidth estimation, partial ACK mechanism

1. Introduction

With the emerging modern communication technology, more and more links with BER are used for network transmission, such as power line communications (PLC). Because of its complex impedance characteristics, serious signal attenuation, strong noise interference, randomness and time-varying, frequent random packet losses will occur during the transmission process. Therefore, the traditional TCP protocol based on the wired networks with good link quality has been unable to meet the performance requirements of increasingly complex network environment, especially in networks with high BER. This is because the traditional TCP protocol mistakes random packet losses in the link with high BER for congestion losses and it causes the unnecessary congestion control mechanism which reduces the sender's transmission window size, resulting in low end-to-end throughput and bandwidth utilization. Therefore, a clear and simple adjustment of the congestion window based on loss differentiation between congestion losses and random packet loss becomes the key of study, and attracts a lot of researchers.

Received(June 15, 2012), Review request(June 16, 2012), Review Result(1st: July 04, 2012)
Accepted(December 31, 2012)

[1]Chief Scientist, Techno Forum Research and Development Centre, India
email: k.kokulakrishnahari@technoforum.co.in

*This work is supported by the Fundamental Research Funds for the Central Universities Project No. ZYGX2012J020

ISSN: 2383-5281 AJMSCAHS

In the past years, several TCP enhancements have been presented for the networks with high BER, and they can be mainly classified into three different approaches: split-connection technique [1], localized link layer solution [2] and end-to-end schemes. Especially, the end-to-end schemes have the most widely attention because of their low complexity and intact end-to-end TCP semantics. As an end-to-end scheme, TCP Veno [3] is proved to be more applicable to the networks with high BER than other traditional TCP variants. Though TCP Veno distinguishes the congestion and non-congestion states in congestion avoidance phase, differentiates the random losses from network congestion during fast retransmit phase and eventually improves the link throughput, its performance is still not good enough in the environment of high BER. The main reasons are as follows.

i. TCP Veno cannot accurately estimate the minimum round-trip time (BaseRTT), leading to an inaccurate loss differentiation, and resulting in a decline of bandwidth utilization rate in the networks with high BER.
ii. Its fast retransmit and fast recovery (FF) algorithm cannot be well adapted to the situation that multiple consecutive packets are lost within a window [4].

To overcome the above disadvantages of TCP Veno, we propose a novel end-to-end congestion control mechanism called TCP VenoDB based on relative delay and bandwidth estimation, and expect that it can provide a significant performance improvement of throughput, bandwidth estimation and reduce the influence of the reverse link congestion in the environment with high BER compared with other TCP variants, such as NewReno [5], Veno and Westwood [6].

The organization of the rest of this paper is as follows. Section 2 introduces and analyzes the principle of TCP Veno. In section 3, we propose and discuss the TCP VenoDB. Section 4 descripts the simulation and performance evaluation of TCP VenoDB and other TCP variants. Finally, section 5 concludes with the simulation and observation.

2 Principle and analysis of TCP Veno

2.1 TCP Veno

TCP Veno makes use of the idea of congestion control scheme in TCP Vegas [7] and intelligently integrates it into TCP Reno [8]. It refines Vegas' additive increase and multiplicative decrease (AIMD) algorithm and adds the packet loss differentiation mechanism to Reno after receiving three duplicate ACKs. In the slow-start phase, Veno uses the same exponential increase algorithm as Reno. In the congestion avoidance phase, Veno adopts the improved AI algorithm coming from TCP Vegas to reduce the transmission speed, so as to avoid congestion: Veno uses the Diff [3] value of Vegas to check whether or not the bottleneck link is currently congested and then adjusts the cwnd accordingly. The Diff value can be obtained by subtracting actual throughput from expected throughput [3]. When three duplicate ACKs occur, Veno uses the number of packets queued in the router buffer N [3], which is the product of Diff and BaseRTT, as a criterion of loss differentiation. When N is smaller than the threshold β (generally 3), Veno reasonably concludes that the current network bandwidth is not fully utilized, and regards the packet loss as a random loss. Otherwise, the packet is lost for network congestion. According to this way, Veno performs the loss differentiation, and reduces ssthresh by a smaller amount (1/5) rather than the half of cwnd for random loss in order to avoid unnecessary rate reduction. Veno's fast retransmit and fast recovery (FF) algorithm remains intact compared with the previous Reno.

2.2 Related Works of TCP Veno

Obviously, Veno does not take into account the influence brought by the fact that the minimum round-trip time (BaseRTT) is difficult to accurately estimate in the environment with frequent random bit errors. Therefore, computing the expected throughput and N solely depending on inaccurate BaseRTT will lead to low accuracy rate of differentiating packet losses, which will then result in the low utilization rate of bandwidth and degradation of throughput. To solve this problem, there are a lot of improvements in last years.

TCP Vegas-A [9] added the comparison of two consecutive RTT's throughput on the basis of Vegas, and dynamically adjusted the threshold in order to reduce the influence brought by inaccurate estimation of the BaseRTT. TCP-New Veno [10] also made use of the idea of Vegas-A, and then adjusted the cwnd according to different network states. Based on the two improvements above, TCP Vegas-A+ [11] proposed a new differentiation condition, the relative delay. And then TCP Veno+ [12], joined the throughput estimation and relative delay together as the judgment conditions to realize a more detailed division of the network states. However, little attention has been paid to the following defects still existing in these improved algorithms:

i. The estimated throughput is coarse, gained simply by the ratio of cwnd and round-trip time (RTT). The network status is obviously not precise enough, only judged by the coarse-grained estimation of throughput.
ii. The relative delay, namely the difference of ACK packet arrival interval and packet transmission interval, is susceptible to reverse link congestion. The reverse link congestion shortens the ACK packet arrival interval, which leads to the ACKs compression and an accuracy decline of relative delay estimation.
iii. They only refined the AIMD algorithm, leaving the traditional FF algorithm intact. FF will reduce

ISSN: 2383-5281 AJMSCAHS

congestion window by several times if multiple packets are lost within a window and drive the TCP connection into timeout, finally cause TCP performance degradation.

3 TCP VenoDB

In order to solve the above problems, TCP VenoDB proposes the following two improvements.

i. The one is that a precise bandwidth estimation and refined relative delay are used to improve AIMD algorithm, and differentiate network states synthetically. Firstly, the coarse-grained throughput estimation is replaced with a more precise bandwidth estimation, which is obtained from the amount of data acknowledged by each arrival ACK divided by the corresponding time interval. Secondly, the refined method of calculating the relative delay [13] based on the timestamp is supposed not only to be conducive to the judgment of network states, but also to reduce the interference from the reverse link congestion.
ii. The other improvement is on the FF algorithm. TCP VenoDB adopts the partial ACK mechanism similar to TCP NewReno to deal with the problem of performance degradation caused by multiple packets losses within a window.

3.1 Improved AIMD

TCP VenoDB joins the refined relative delay and available bandwidth estimation together as a judgment condition to distinguish the network states. Through the introduction of relative delay variable D[13], relative delay difference ΔD, bandwidth estimation variables bwe(t) and bwe(t-RTT), the network would be divided into three states: stable states, congestion increasing states and congestion decreasing states. VenoDB would adopt a more reasonable and effective algorithm to adjust cwnd according to different network states respectively.

A. Relative Delay

Relative delay Dn is namely the transmission time difference of two consecutive data packets Packet (n-1) and Packet (n). Firstly, the sender records the transmitting timestamps Sn and Sn-1 of two consecutive packets respectively, and then calculates the transmission time interval Sn-Sn-1. Secondly, the receiver records the two packets' receiving timestamps Rn and Rn-1 when the packets are received, and puts the two timestamps into their respective ACK packet. Next, the sender gets the receiving timestamp from TCP header of the ACK returned to the sender, so the reception time interval between two packets is Rn-Rn-1. Finally, the relative delay between two consecutive data packets Packet(n-1) and Packet(n) is supposed to be

Dn=(Rn-Rn-1)-(Sn-Sn-1) [13], while ΔD means the jitter of two consecutive relative delays Dn and Dn-1. The two parameters above are calculated by the following formulas.

Dn = Packet reception interval - Packet transmission interval

$= (Rn-Rn_{-1})- (Sn-Sn_{-1}= (Rn-Sn)-(Rn_{-1}-Sn_{-1})$ (1)

$\Delta D = Dn - Dn_{-1}$ (2)

B. Bandwidth Estimation

To estimate available bandwidth, VenoDB adopts the following method. Firstly, the amount of data acknowledged by each arrival ACK packet will be divided by the corresponding time interval to get the bandwidth sample; Secondly, VenoDB takes advantage of an exponentially weighted moving average smoothing filter to deal with the sample in order to obtain available bandwidth bwe(t). This method uses the idea of the famous bandwidth estimation in Westwood algorithm detailed in reference [6]. Here the variable bwe(t) is defined as the available bandwidth of the bottleneck link in the present moment, while bwe(t-RTT) indicates the available bandwidth in the moment of last RTT. Then, VenoDB would predict the network states further in congestion avoidance phase based on the comparison of the bandwidth estimation bwe(t) and bwe(t-RTT).

C. Network States Division

Firstly, when the relative delay D=0, namely that the transmission time interval of two packets is equal to the reception time interval, which means that the packets does not experience queuing delay, the network is stable; When D>0, in other words, the reception time interval is greater than the transmission time interval, which means that the queuing delay between intermediate nodes is increasing, the degree of network congestion is increasing; Similarly, D<0 means the queue length between the intermediate nodes is becoming shorter, i.e., queuing delay is reduced and in this case, the network is in the congestion decreasing state.

Secondly, for further differentiation of network states, relative delay difference ΔD is introduced. ΔD=0 means the network state is stable. ΔD>0 indicates increased network congestion, and ΔD<0 indicates the state of decreased network congestion.

Thirdly, when bwe(t)<bwe(t-RTT), which means the network congestion state is increasing. In the case of bwe(t)>bwe(t-RTT), the network is supposed to be in the state of congestion decreasing. When bwe(t)=bwe (t-RTT), the network state is stable.

D. Comprehensive Judgment

In the refined AI algorithm of congestion avoidance phase, when N>β, the bandwidth is close to saturation state, according to the variables D, ΔD, bwe(t) and bwe(t-RTT) ,VenoDB divides the network into three states:

ISSN: 2383-5281 AJMSCAHS

stable state, increasing congestion state and decreasing congestion state. When $N<\beta$, the bandwidth is not yet saturated, VenoDB fine-tunes the cwnd according to ΔD. If the congestion degree has been alleviated, VenoDB will increase the cwnd appropriately. Otherwise if the congestion state is intensifying, VenoDB will adjust the cwnd to maintain a linear growth as the original Veno does. In the improved algorithm MD algorithm, i.e., when more than three duplicate ACKs occurs. If $N>\beta$, VenoDB will regard the packet loss as a congestion loss, mean while, $D<0$ means the degree of congestion has been alleviated, in this case, there is no necessity to halve the cwnd as the original congestion control mechanism of Veno, cutting the cwnd down to eighty percent of the current cwnd is enough. When $N<\beta$, VenoDB deals with the cwnd the same way with Veno. The pseudocode of these improvements is shown below.

```
//The refined AI--Differentiate network states
if(N>=β) //The network is close to saturation
if(D<0 || bwe(t)>bwe(t-RTT)) //State1: decreased congestion state
cwnd = cwnd + 1; //maintain a linear growth
if(D=0 || bwe(t)=bwe(t-RTT)) //State2: stable state
if(ΔD<0) //fine-tuned
cwnd = cwnd + 1 //maintain a linear growth
if(ΔD>=0) //fine-tuned
cwnd = cwnd +1/2 //add 1 every other RTT
if(D>0 || bwe(t)<bwe(t-RTT)) //State3:increased congestion state
if(ΔD<=0)
cwnd = cwnd + 1/4 //add 1 every 4 RTTs
if(ΔD>0)
cwnd = cwnd //cwnd no change
if(N<β) //The network is not yet saturated
if(ΔD<0)
cwnd = cwnd + 2 //add 2 each RTT
if(ΔD>=0)
cwnd = cwnd +1 //maintain a linear growth
//The refined MD after 3 dup-ACKs, loss differentiation
if(N>=β) //congestion loss
if(D>=0) //congestion state increasing
cwnd = cwnd×1/2
if(D<0) //congestion state decreasing
```

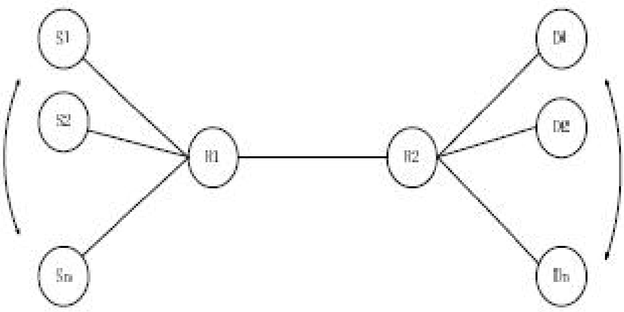

cwnd = cwnd×4/5
if(N<β) //random loss
cwnd = cwnd×4/5

3.2 Improved FF Algorithm

TCP VenoDB makes use of the partial ACK mechanism inherited from TCP NewReno to deal with the problem of performance degradation caused by multiple packets losses within a window in the traditional FF algorithm. TCP NewReno modifies TCP Reno's sender-side behavior for the partial ACK, which acknowledges the part of but not all of the outstanding packets, when multiple packets are lost in one recovery phase. Upon receipt of the partial ACK, TCP NewReno retransmits the lost packets but fixes the congestion window until all the lost packets are retransmitted. As a result, TCP NewReno avoids multiple window reductions in one recovery phase. The detailed description is presented in reference [5]. Therefore, VenoDB is to provide more significant performance improvement than TCP Veno, especially in networks with high BER.

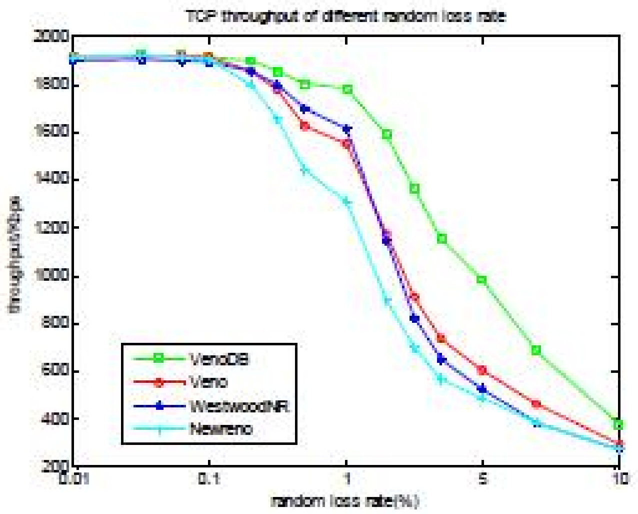
4 Simulation and Performance Analysis

Several experiments in NS2 are performed to evaluate the efficiency of VenoDB among a wide range of options in networks with high BER. In terms of throughput, bandwidth utilization, fairness, friendliness and the influence of inverse network link congestion, we compare the performance of VenoDB to the following three TCP variants: TCP Veno, TCP WestwoodNR and TCP NewReno.



[Fig. 1] Simulation topology

The traditional dumbbell network topology as shown in Figure 1 is used for performance evaluation. Related parameters for the network are as follows: The TCP senders S1 to Sn connect to the router R1 via a 10 Mb/s link with 10 ms one-way delay. The TCP receivers D1 to Dn connect to the router R2 via a 10 Mb/s link with 10ms one-way delay. The link between R1 and R2 is the bottleneck link with 10ms one-way delay, 2Mb/s bandwidth unless otherwise specified. The maximum segment size of TCP is set to 1000 bytes. The range of random packet loss rate of bottleneck link is 0.01%~10%. The router buffers are all set to a constant value of 60 packets. The bottleneck router uses a DropTail scheduling policy.

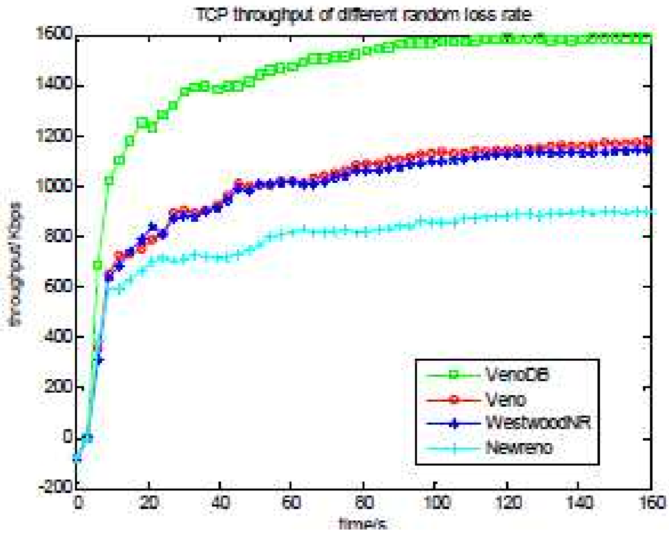


[Fig. 2] Average throughput of different random error rate

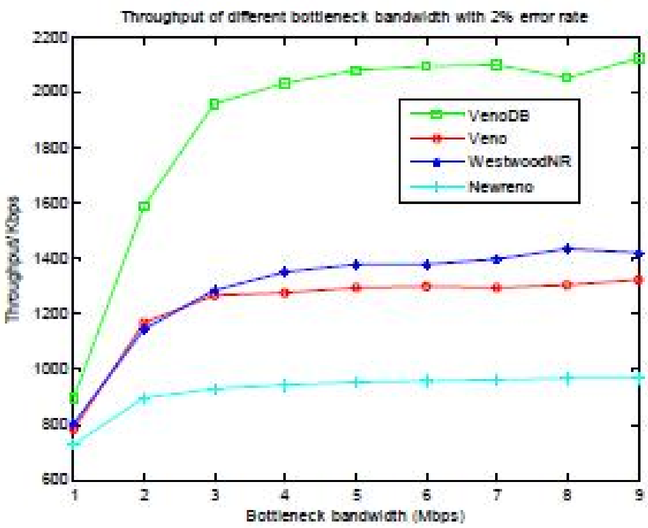
4.1 Performance Simulation under Single Flow

In Figure 1, the node S1 sends a single FTP flow to the receiver node D1 from 0s to 100s using the VenoDB, Veno, WestwoodNR and NewReno respectively, via a bottleneck link with 2 Mb/s bandwidth, and the random error rate ranges from 0.01% to 10%.

In Figure 2, when the BER is less than 0.1%, which means nearly all lost packets result from the congestion, the average throughput curves of the four algorithms are almost the same. The advantages of VenoDB are more and more obvious than the other three algorithms with the increasing of BER starting from 0.1%. Furthermore, when the BER is higher than 1%, namely under the environment of high bit error rate, the throughput of VenoDB is especially superior to others. When the random error rate is 2%, for example, the throughput of VenoDB exceeds that of the Veno, WestwoodNR and NewReno for 35.7%, 38.6% and 76.8% respectively. It can be seen from Figure 3 that the VenoDB provides not only significant improvement of throughput, but also better stability compared with the other three algorithms under high BER. The advantages of VenoDB still exist in various bottleneck bandwidths as shown in Figure 4.



[Fig. 3] Actual throughput under the BER of 0.02



[Fig. 4] Throughput in different bottleneck bandwidth

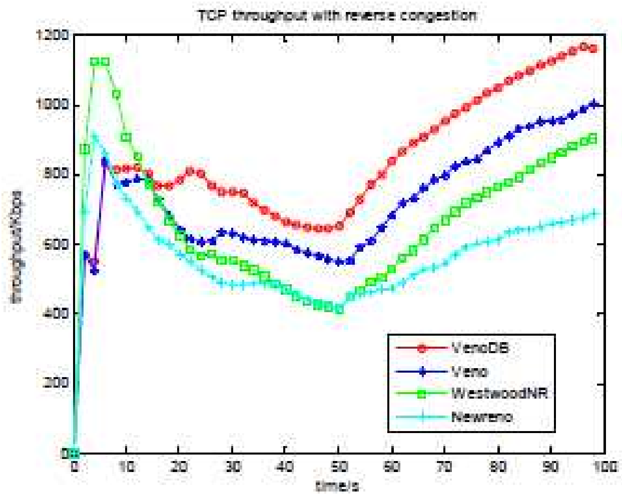
4.2 Performance Simulations under Multi-Flows

A. Influence of Reverse Congestion Interference

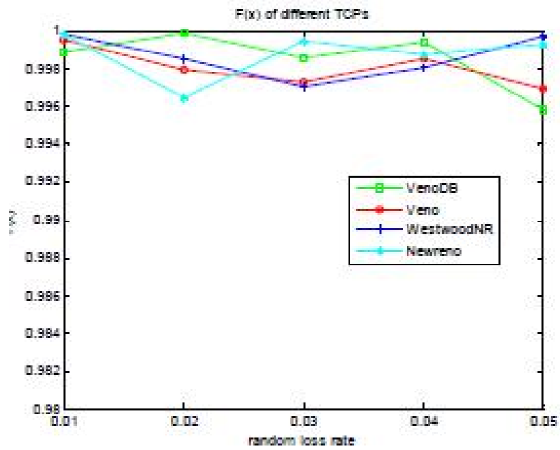
To evaluate the performance of the TCP VenoDB in the minimum influence from the reverse link congestion, a reverse UDP connection running CBR flow from node D2 to S2 between 5s~50s is added in our experiment. The maximum segment size of UDP is set to 1000 bytes and the transmission rate of UDP flow is

4 Mb/s. The node S1 sends FTP flow to node D1 from 0s to 100s using the protocol of TCP VenoDB, TCP Veno, TCP WestwoodNR and TCP NewReno respectively via a 2Mb/s bottleneck with 1% random error rate.

As shown in Figure 5, during the period of reverse congestion, the throughput of VenoDB is the most stable among the other three algorithms. When the reverse congestion just appeared, VenoDB's throughput curve not only sharply declined like other algorithms, but also increased. At the end of the reverse congestion phase, VenoDB reacted first and achieved higher bandwidth shares with the fastest rate. Therefore, the VenoDB is only slightly influenced by the reverse link congestion, compared with other TCP variants.



[Fig. 5] Throughput comparison under reverse link congestion

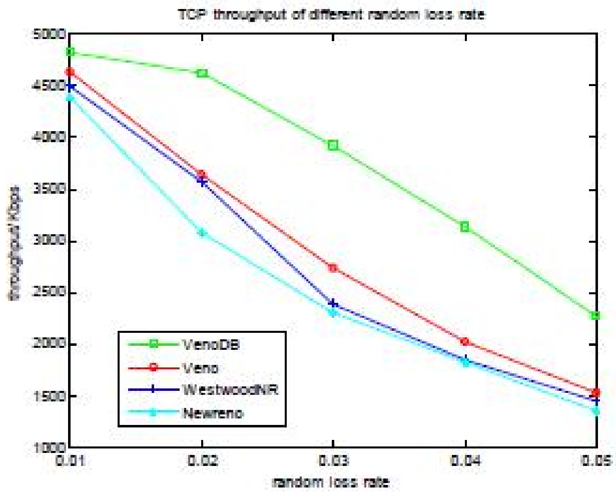


[Fig. 6] Fairness Index comparison of different error rate

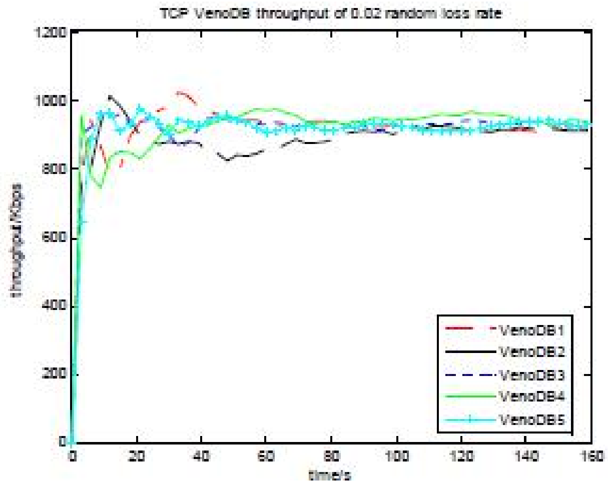
B. Intra-Protocol Fairness

The intra-protocol fairness is generally evaluated with the fairness index[14]. To verify whether the fairness index of VenoDB meets the requirements of intra-protocol fairness or not, we adopts the Jain formula to calculate the fairness index F(x) [15]. In our tests, five senders S1 to S5 are connected to five receivers D1 to D5 via a bottle neck link of 5Mb/s bandwidth with a range of BER 1%~5%, and five FTP flows are ran from 0s to 160s based on the same protocol of TCP VenoDB, TCP Veno, TCP WestwoodNR and TCP NewReno respectively. The fairness index F(x) and the total average throughput of four different TCP algorithms are shown in Figure 6 and Figure 7.

In Fig. 6, the fairness index of four algorithms are all close to 1 on the condition of high BER, except that the fairness index of VenoDB is slightly higher than the others, which means the four algorithms are all of good fairness and VenoDB has a better fairness than the others. Besides that, from Fig. 7, we can see clearly that the total average throughput of VenoDB is significantly higher than that of other algorithms, namely that VenoDB greatly improve the bottleneck link bandwidth utilization compared to the others. Fig. 8 shows the throughput curve of five VenoDB flows from 0s to 160s with 2% BER, apparently, the five flows almost equally allocate the bottleneck bandwidth, which shows a good fairness and stability of TCP VenoDB. In conclusion, VenoDB does not only have a better fairness, higher bandwidth utilization compared with other algorithms in high BER situation, but also has a good stability.



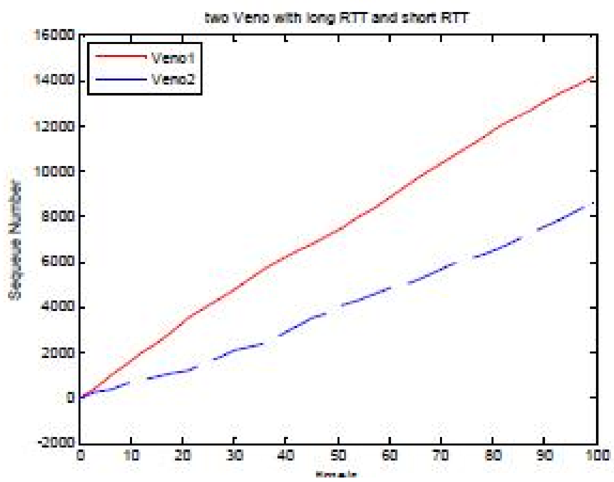
[Fig. 7] Total throughput of different BER Fig.



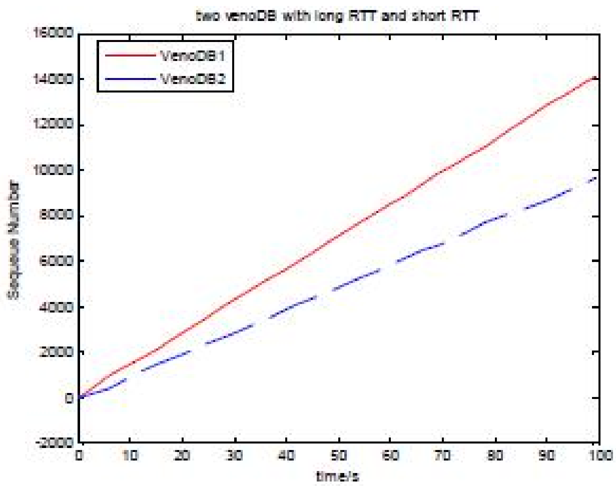
[Fig. 8] Throughput of 5 VenoDB flows under 0.02 error rate

C. RTT Fairness

To evaluate the RTT fairness of different length of RTT flows between VenoDB and Veno, our test sets two senders S1 and S2 connected to two receivers D1 and D2, running the same TCP algorithm with two different length of RTT, a short one with 50ms and a long one with 140ms, via a bottleneck of 2Mb/s bandwidth from 0s to 100s with 1% BER. Figure 9 shows the throughputs (sequence number) obtained from two Veno flows and Figure 10 shows the throughputs obtained from two VenoDB flows on the same condition.



[Fig. 9] Long RTT vs short RTT's throughput of Veno

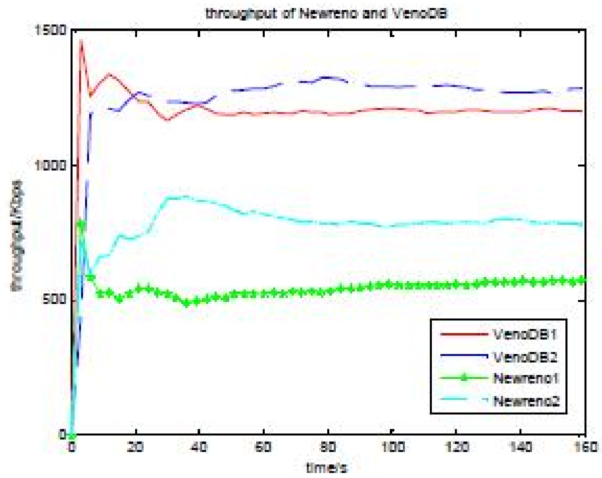


[Fig. 10] Long RTT vs short RTT's throughput of VenoDB

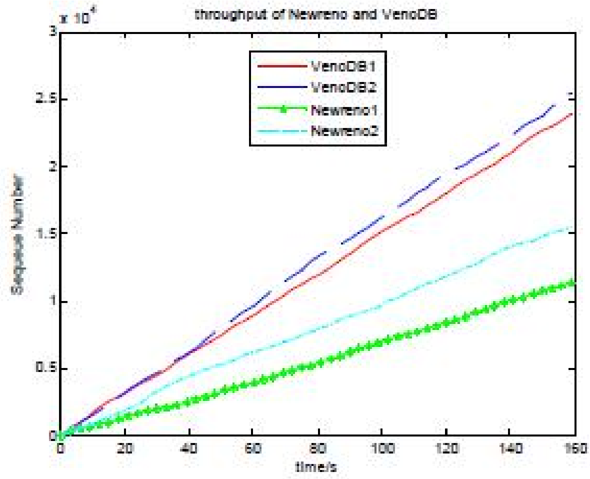
As shown in Figure 9 and Figure 10, the packets' sequence number of the short RTT flow is growing faster than that of long RTT flow. Besides, the difference of growth rate between long RTT flow and short RTT flow of VenoDB in Figure 10 is much smaller than that of Veno in Figure 9. Furthermore, from the perspective of fairness index, the values of F(x) obtained from two VenoDB flows are also higher than that of two Veno flows. In summary, after joining the relative delay, bandwidth estimation and partial ACK mechanism into VenoDB, the improved algorithm achieves a better RTT fairness and more equitable distribution of network bandwidth.

D. Friendliness

To evaluate the friendliness between TCP VenoDB and the most widely used protocol TCP NewReno, our test sets two senders S1, S2 running two NewReno flows and two senders S3, S4 running two VenoDB flows via a bottleneck of 4Mb/s bandwidth together with 1% BER. As shown in Figure 11 and Figure 12, although the two VenoDB flows occupy more bandwidth than the two Veno flows do, which is because the VenoDB utilizes not only its fair share, i.e., half of the link bandwidth, but also the bandwidth left by the coexisting NewReno, the two VenoDB flows and two Veno flows allocates the bottleneck bandwidth reasonable, which means that, for practical purpose, VenoDB is friendly towards NewReno.



[Fig. 11] Throughput of 2 NewReno and 2 VenoDB flows



[Fig. 12] Sequeue number of 2 NewReno and 2 VenoDB flows

5 Conclusions

Although TCP Veno has achieved remarkable success in network with heavy random losses, its potential has not been fully utilized, this is because Veno can not accurately estimate the minimum round-trip time (BaseRTT), leading to an inaccurate loss differentiation and a decline of bandwidth utilization rate. In this paper, we propose an end-to-end improved TCP algorithm called TCP VenoDB for the networks with high BER. TCP VenoDB incorporates the novel relative delay based on timestamp, bandwidth estimation and partial ACK mechanism of FF phase in order to improve its performance in the heavy random loss environment.

Simulation results reveal that TCP VenoDB provides more significant performance improvement in throughput, bandwidth utilization and fairness than TCP Veno, TCP Westwood and TCP NewReno in high BER networks, and has the minimum influence from the reverse link congestion. Further more, for practical purpose, TCP VenoDB is friendly towards TCP NewReno.

ISSN: 2383-5281 AJMSCAHS

Reference

[1] E. Ayanoglu, S. Paul, T. F. LaPorta et al., "AIRMAIL: A Link-layer Protocol for Wireless Networks", Wireless Networks, vol. 1, no. 1, **(1995)**, pp. 47-60.

[2] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, et al., "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", Networking, IEEE/ACM Transactions on, vol. 5, no. 6, **(1997)**, pp. 756-769.

[3] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks", Selected Areas in Communications, IEEE Journal, **(2003)**, vol. 21, no. 2, pp. 216-228.

[4] K. Fall, S. Floyd, Simulation-based Comparisons of Tahoe, Reno and SACK TCP", ACM SIGCOMM Computer Communication Review, vol. 26, no. 3, **(1996)**, pp. 5-21.

[5] S. Floyd and T. Henderson, "RFC2582: The NewReno Modification to TCP's Fast Recovery Algorithm", **(1999)** April.

[6] M. Gerla, M. Y. Sanadidi, R. Wang, et al., "TCP Westwood: Congestion Window Control using Bandwidth Estimation", Global Telecommunications Conference, GLOBECOM'01. IEEE. IEEE, **(2001)**, pp. 1698 1702.

[7] L.S. Brakmo, S.W. O'Malley and L. L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance", Proc. ACM SIGCOMM, **(1994)** October.

[8] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", RFC2581, **(1999)** April.

[9] K. N. Srijith, L. Jacob, A. L. Ananda, "TCP Vegas-A: Improving the performance of TCP Vegas", Computer Communications, vol. 4, **(2005)**, pp. 429-440.

[10] N. Cho and K. Chung, "TCP-new Veno: The Energy Efficient Congestion Control in Mobile Ad-Hoc Network", Embedded and Ubiquitous Computing, Springer Berlin Heidelberg, **(2006)**, pp. 254-263.

[11] W J H. Guo-ce and C C Z H U Meng, "Enhanced TCP Vegas Congestion Control Algorithm Based on Congestion Status Differentiation over GEO Satellite Networks", Computer Science, vol. 11, **(2008)**, p. 020.

[12] Y. Hu, Y. Chen, W. Feng, et al., "Improvement of TCP-Veno over Hybrid Wired/Wireless Network", Modern Computer, vol.10, **(2010)**, p. 6.

[13] J J C Ryan, "Information Security Tools and Practices: What Works", Computers, IEEE Transactions, vol. 53, no. 8, **(2004)**, pp. 1060-1063.

[14] R, Chun, J. M. Hellerstein, et al., "The Architecture of Pier: an Internet-scale Query Processor", Departmental Papers (CIS), **(2005)**, p. 305.

[15] R. Jain, D. M. Chiu and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System", Eastern Research Laboratory, Digital Equipment Corporation, **(1984)**.

