Reno Friendly TCP Westwood based on Router Buffer Estimation

Kazumi KANEKO and Jiro KATTO

Graduate School of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjyuku-ku, Tokyo, 169-8555 Japan
E-mail: {kaneko, katto}@katto.comm.waseda.ac.jp

Abstract— TCP Reno versions are widely used in current network, however it has been actualized that their throughput deteriorates in high-speed network and wireless environments. To overcome these problems of TCP Reno versions, a number of protocols have been proposed. In these researches, friendliness with TCP Reno becomes important. TCP Westwood is an example that is based on end-to-end bandwidth estimate, and brings higher efficiency performances. However it will be shown that the friendliness with TCP Reno is deteriorated according to network situations such as the buffer size of a bottleneck link router. In this paper, we quantify the buffer size that TCP Reno and TCP Westwood perform friendly to each other, and then propose an improved version of TCP Westwood that achieves friendliness with TCP Reno by estimating the buffer size of a bottleneck link router by using bandwidth estimation technique, RCE, and by updating congestion control parameters under the constraint of throughput estimation models. We confirm effectiveness of our proposal scheme by extensive simulation experiments. Simulation results show that our proposed scheme performs friendly to TCP Reno in various network situations independently of router buffer sizes.

Index Terms—TCP Westwood, TCP Reno, Friendliness, Router Buffer Size

I. INTRODUCTION

THE TCP(Transmission Control Protocol) protocol is widely used in current network, provides end-to-end, reliable congestion control. The majority of data services in the Internet are carried by TCP, with applications including *FTP* (File Transfer Protocol) and *HTTP* (Hyper Text Transfer Protocol). Recently, while the amount of Internet traffic is explosively increasing with the rapid growth of Internet users, Internet is evolving to high speed network with large bandwidth delay product and wireless environments. Thus even if the network infrastructure may change, TCP has to adapt to these various networks in a scalable manner.

The TCP congestion control mechanisms have evolved over time, from TCP Tahoe to the currently widely used TCP Reno versions (Reno [1], NewReno [10], SACK [9]). However many problems of TCP Reno have been actualized. When TCP Reno are used for high speed network environments with large bandwidth delay product, it can not achieve enough throughputs to fill link bandwidth because of the essential nature of TCP Reno's congestion control mechanism. The main reason is that TCP Reno decreases its congestion window size dramatically when packet losses are detected, but increases it slightly when no packets are lost. Furthermore, when TCP Reno is used for wireless environments, their throughput is degraded. This is because TCP Reno is originally designed for traditional wired environments, where congestion accounts for packet losses. However, unlikely to wired environments, in wireless environments, packet losses are due to bit errors and external interference over wireless links. The congestion control mechanism of current TCP Reno detects packet losses due to not only congestion but also link error such as wireless environments, and reduce its congestion window size.

To overcome these problems of TCP Reno, a number of protocols have been proposed; for example, TCP Vegas[13], FAST TCP [20], Highspeed TCP[14], Scalable TCP[15], and TCP Westwood[2][3]. In such researches, friendliness with legacy protocol becomes important problem.

However, in the situation where TCP Reno and new protocols connections share the same bottleneck link, occurs a problem that throughput of either one protocol is degraded. Among of them, in the case of TCP Westwood, when the buffer size of bottleneck link router is set to bandwidth delay product, TCP Westwood achieves friendliness with TCP Reno. However, if the buffer size is not set to the bandwidth delay product, the friendliness with TCP Reno is deteriorated. For example, when buffer size is set to smaller than bandwidth delay product, throughput of TCP Reno connections is degraded. On the other hand, when buffer size is set to larger than bandwidth delay product, throughput of TCP Westwood connections is degraded by TCP Reno connection

In this paper, we propose an improved version of TCP Westwood to overcome unfriendliness of TCP Westwood according to buffer size of bottleneck link router. We first investigate the friendliness of TCP Westwood through mathematical analysis using throughput model. We then estimate the buffer size of the bottleneck link router by applying a bandwidth estimation technique known as *RCE* (Residual Capacity Estimator) [4], and set the parameter considering difference between actual buffer size and the buffer size in mathematical analysis. Simulation results using ns-2 will show that proposal method achieve friendliness with

Proceedings of the Joint International Conference on Autonomic and Autonomous Systems
and International Conference on Networking and Services (ICAS/ICNS 2005)
0-7695-2450-8/05 $20.00 © 2005 IEEE

TCP Reno all the case of variety buffer size of bottleneck link router.

The rest of this paper is organized as follows. In Section Ⅱ, we summarized the congestion control mechanism TCP Westwood. In Section Ⅲ, the problem of TCP Westwood is described in detail, and buffer size with friendliness to TCP Reno is quantified by mathematical analysis. In Section Ⅳ, we propose an improved version of TCP Westwood, and show its effectiveness through various simulation experiments in Section Ⅴ. We finally conclude this paper in section Ⅵ.

II. Congestion Control Mechanisms of TCP Westwood

TCP Westwood[2] [3] is a sender-side only modification of TCP. In TCP Westwood, the sender continuously monitors ACK packets from the receiver and computes the current Eligible Rate Estimation (ERE) for this connection. The ERE depends on an adaptive estimation technique applied to ACK packets stream. Several of ERE methods had been proposed, for example, Bandwidth Estimation (BE) [2], Rate Estimation (RE) [3], and Adaptive Bandwidth Share Estimation (ABSE) [16][17]. In this paper, we use a Rate Estimation (RE) method.

In the RE method, the sender considers the amount of data acknowledged during the latest interval of time T, which is typically a RTT (round trip time), and computes the RE sample as the amount of data successfully delivered divided by interval of time T. Then the RE samples are fed into a low-pass filter to get a smoothed estimate. If TCP Westwood sender detects packet losses by duplicate ACK packets, *cwnd* and *ssthresh* are updated as follows;

$$\begin{cases} ssthresh = \dfrac{(ERE * RTT_{\min})}{Packet\_size * 8}; \\ if\,(cwnd > ssthresh) \qquad cwnd = ssthresh, \end{cases} \tag{1}$$

where ERE is estimated bandwidth obtained by the RE method, and $RTT_{\min}$ is the minimum of RTT. On the other hand, if the TCP Westwood sender detects packet losses by retransmission timeout expiration, *cwnd* and *ssthresh* are updated as follows;

$$\begin{cases} cwnd = 1, \\ ssthresh = \dfrac{(ERE * RTT_{\min})}{Packet\_size * 8}; \end{cases} \tag{2}$$

III. The Problem of TCP Westwood

In this section, we firstly describe a problem of TCP Westwood about its sensitiveness to a buffer size of the bottleneck router. We then analyze mathematically its friendliness with Reno using steady state throughput models, and quantify a buffer size which enables TCP Reno and TCP Westwood to perform friendly to each other.

A. Router buffer problem of TCP Westwood

TCP Westwood uses the minimum RTT ($RTT_{\min}$) to set the *ssthresh*, but this means that TCP Westwood does not consider RTT oscillation which happens when network begins to be congested. The fact that *RTT* relies on link delay (approximately $RTT_{\min}$) and buffering delay means that TCP Westwood performance will depend on buffer size of a bottleneck link router.

Sizes of router buffers are determined based on rule-of thumb. A router needs an amount of buffering time in the buffer that is equal to an average round trip time of a connection that passes through the router multiplied by a link capacity as

$$B = \frac{\overline{RTT} * C}{Packet\_size * 8} \tag{3}$$

where B is a buffer size, $\overline{RTT}$ is an average round trip time of a connection, and C is a link capacity. This is a well-known rule and is equal to bandwidth-delay product (BDP). However, the routers can not know $\overline{RTT}$ of a connection that passes through the router, moreover, and $\overline{RTT}$ of each connection that passes through the router is different. Therefore, network router manufactures assigns typical RTT values such as 250ms or more of buffering time [18] [19]. It is impractical for all TCP connections to be assigned a buffer size that is equal to BDP of each connection.

In the situation where TCP Reno and TCP Westwood connections share the bottleneck link, when the buffer size is smaller than BDP, buffer overflows happen frequency, and both TCP Reno and TCP Westwood connections can not increase their window sizes to fill the link capacity before the buffer overflow. TCP Reno connection detects packet losses by duplicate ACK packets, and reduces its window size by half, which would be smaller than half of the link capacity, while TCP Westwood keeps higher window sizes corresponding to the eligible rate estimation (ERE). As a result, throughputs of TCP Reno connections are degraded by TCP Westwood connections, thus it is unfriendly.

On the other hand, when the buffer size is larger than BDP, throughputs of TCP Westwood connections are degraded oppositely by TCP Reno connections. TCP Reno connection detects packet losses by duplicate ACK packets, and reduces its window size by half, but the reduced window size is still too large to clear the buffer. On the other contrary, due to the large buffer size, TCP Westwood connections reduce the *cwnd* and *ssthresh* much more than half and clear the buffer much faster. Thus it is unfriendly, too.

B. Mathematical analysis of friendliness with TCP Reno

This subsection investigates the problem of TCP Westwood shown in the previous subsection using a steady state throughput estimation model of TCP Reno and TCP Westwood. We then quantify the buffer size that TCP Reno and TCP Westwood are friendly to each other.

Steady state throughput estimation for TCP Reno is given by

$$S_{Reno} = \frac{1}{RTT}\sqrt{\frac{2(1-p)}{p}} \tag{4}$$

where p is loss probability [7]. Similarly, steady state throughput estimation for TCP Westwood is provided by

Proceedings of the Joint International Conference on Autonomic and Autonomous Systems
and International Conference on Networking and Services (ICAS/ICNS 2005)
0-7695-2450-8/05 $20.00 © 2005 IEEE

$$S_{West} = \frac{1}{\sqrt{RTT}\sqrt{T_q}}\sqrt{\frac{(1-p)}{p}} \quad (5)$$

where T_q is a average buffering time that is equal to the difference between RTT and the minimum round-trip time $RTT_{\min}$, i.e., $RTT - RTT_{\min}$ [8]. With reference to friendliness, by comparing Eq.(4) and Eq.(5), it can be noted that the both throughputs of TCP Reno and TCP Westwood is proportional to $1/\sqrt{p}$. However, the TCP Westwood throughput model has a buffering delay of parameter T_q, which will cause unfriendliness with TCP Reno.

On the other hand, the buffering delay T_q is given by using buffer size *B* by

$$T_q = \frac{B * Packet\_size * 8}{C} \quad (6)$$

where *C* is a link capacity. When Eq.(5) is equal to Eq.(4), TCP Reno and TCP Westwood are friendly to each other, where buffer size *B* corresponds to the case of setting

$$B = \frac{C * RTT_{\min}}{Packet\_size * 8} \quad (7)$$

This is equal to the bandwidth-delay product (BDP) of Eq.(3). When the buffer size of router is set to half of BDP of the TCP Westwood connection, according to Eq.(5), the throughput ratio between TCP Westwood and TCP Reno becomes 1.4,. When the buffer size of router is set to twice of BDP, the ratio becomes 0.7.

IV. IMPROVED VERSION OF TCP WESTWOOD

In this section, we describe Residual Capacity Estimator used by the proposal method. Next, we propose an improved version of TCP Westwood.

A. Residual Capacity Estimator (RCE) [4]

Residual Capacity Estimator (RCE) scheme is able to estimate the bottleneck link capacity deducted by the uniformly distributed traffic present. The RCE scheme is based on a packet train scheme [5] of which expands a packet pair scheme [6]. The RCE scheme eliminates buffering times. The sender counts packets leaving to the receiver in retransmission time-out (*RTO*), and then the sender waits for correspondent returning ACK packets, where the time is set to *ACKs_slot_time*. Here, the sender calculates the *wasted_time* from the *ACKs_slot_time* standpoint. The sender measures an average of the interarrival time between the ACKs of *ACKs_slot_time*. The *wasted_time* is then computed as the sum of time exceeding the average in each interarrival time of *ACKs_slot_time*. The exceeding gap times between ACKs are most likely a result of having periods of buffering. Then the sender computes the bottleneck link capacity by

$$C = \frac{Bits\_ACKed}{ACKs\_slot\_time - Wasted\_time} \quad (8)$$

where *Bits_ACKed* is the total packets size acknowledged by ACK packets in *ACKs_slot_time.*

B. TCP Westwood improvement considering buffer size

1) Buffer size estimation

The buffer size of the bottleneck link router is estimated by the difference between *RTT* and $RTT_{\min}$ multiplied by a link capacity by

$$BS\_estimte = \frac{(RTT - RTT_{\min}) * C}{Packet\_size * 8} \quad (9)$$

where *BS_estimte* is an estimated actual buffer size, and *C* is a bottleneck link capacity estimated using RCE scheme.

2) RTT_{min} *compensation*

As explained in subsection Ⅲ-B, only when the buffer size of a bottleneck link router is equal to BDP, TCP Westwood can achieve friendliness with TCP Reno. Then the buffer delay difference between the actual buffer size, *BS_estimate* and the BDP are calculated by

$$diff\_delay = \frac{(BS\_estimate - BDP) * Packet\_size * 8}{C} \quad (10)$$

where the BDP is calculated by Eq(7). The steady state throughput estimation model of TCP Westwood which is friendly with TCP Reno can be modified by substituting *diff_delay* in RTT of Eq,(5), and a *fair_response*, that is the parts of a friendliness throughput in Eq.(5) except a loss probability part, is achieved by

$$frindly\_response = \frac{\sqrt{RTT - diff\_delay}}{\sqrt{(RTT - diff\_delay) - RTT_{\min}}} \quad (11)$$

Therefore TCP Westwood can achieve the friendliness with TCP Reno regardless of the buffer size of a bottleneck link router by updating the throughput estimation model of TCP Westwood using *diff_delay* which is buffer delay difference between the actual buffer size and the BDP. Furthermore $RTT_{\min}$ is compensated by reflecting Eq (11) in $RTT_{\min}$ of Eq.(5).

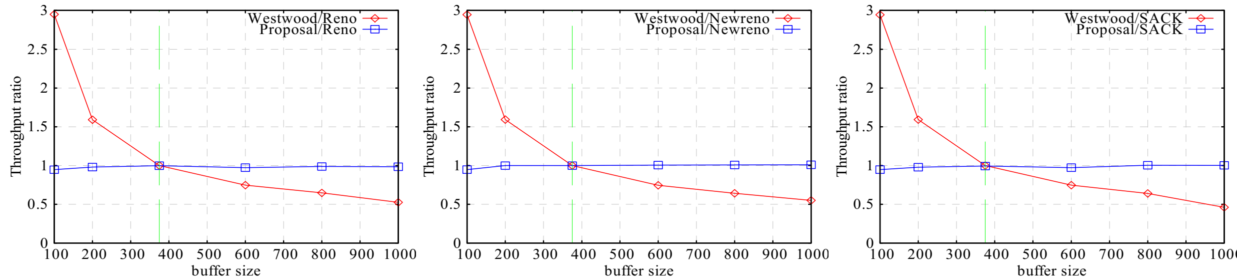
$$\begin{aligned} compensated\_RTT_{\min} &= RTT - \frac{RTT}{fair\_response^2} \\ &= \frac{RTT}{RTT - diff\_delay} RTT_{\min} \end{aligned} \quad (12)$$

Finally, when packet losses are detected, the *ssthresh* is updated by applying $compensated\_RTT_{\min}$ by

$$ssthresh = \frac{ERE * compensated\_RTT_{\min}}{Packet\_size * 8} \quad (13)$$

V. PERFORMANCE EVALUATION

In this section, we compare the performance of TCP Westwood and our proposal scheme under various network conditions. All results are obtained using ns2 simulator [12]. We first evaluate the throughput ratio between legacy protocols and TCP Westwood or our proposal scheme with variant buffer



(a) TCP Reno (b) TCP NewReno (c) TCP SACK

Fig. 1 Throughput ratio between legacy protocols and TCP Westwood/our proposal scheme with variant buffer size

sizes of a bottleneck link router in section V-A. Second, in section V-B, we prove effectiveness of our proposal scheme using an efficiency/friendliness tradeoff graph. In section V-C, we evaluate the fairness between flows with different round trip propagation delay connections. .

A. Throughput Ratio for Variants Buffer Sizes

In this subsection, we evaluate the impact of buffer sizes on friendliness. The network topology consists of two sender hosts (S1 and S2), two receiver hosts (D1 and D2) and two routers (R1 and R2). Host S1 uses TCP Westwood or our proposal scheme for data transmission, and host S2 using TCP Reno versions shares the same link between routers R1 and R2. That is, one connection of TCP Westwood/our proposal scheme and another connection of TCP Reno versions compete on the bottleneck link. The bandwidth and the propagation delay of each link between the routers and sender/receiver hosts is 100[Mbps] and 5[ms]. The bandwidth and the propagation delay of the link between R1 and R2 is 50[Mbps] and 35[ms]. The total round trip delay between the sender hosts and the receiver hosts is 90[ms]. When we assume the packet size is 1500 byte, the bandwidth delay product becomes 375[packets]. We use a *TailDrop* discipline for buffer management of router R1.

We then investigate a throughput ratio in the steady state between TCP Reno versions and TCP Westwood/our proposal scheme with variant buffer sizes. The buffer size is set to four different values. The throughput ratio is calculated by

$$Throughput\ ratio = \frac{Westwood\_throughput}{legacyprotocol\_throughput} \tag{14}$$

Fig. 1 (a), (b) and (c) shows throughput ratios for TCP Reno, TCP NewReno, and TCP SACK, respectively. In these figures, it can be noted that TCP Westwood achieves friendliness to TCP Reno versions only when the buffer size of a bottleneck link router is set to 375[packets], that is the exact BDP in our simulation condition. However, when the buffer size is not set to 375[packets], friendliness with TCP Reno versions is deteriorated as already explained in subsection Ⅲ-A. On the other hand, it should be emphasized that our proposal scheme achieves friendliness with TCP Reno versions for variant buffer sizes, because our proposal scheme adapts the $RTT_{\min}$ according to the buffer size of a bottleneck link router.

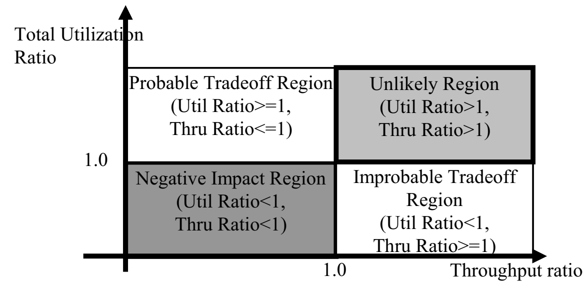


Fig. 2 Efficiency/Friendliness Tradeoff Graph

B. Efficiency/Friendliness Tradeoff graph

To prove effectiveness of our proposal scheme more strictly, we use an efficiency/friendliness tradeoff graph. This tradeoff graph represents how the total link utilization (efficiency) and throughput of TCP Reno (friendliness) are impacted by our proposal scheme. Following experiments are carried out to produce a point on the graph;

1) A simulation with multiple TCP Reno flows is carried out as the base case, where *N* is the number of TCP Reno flows. Throughput of each flow and total link utilization are measured.

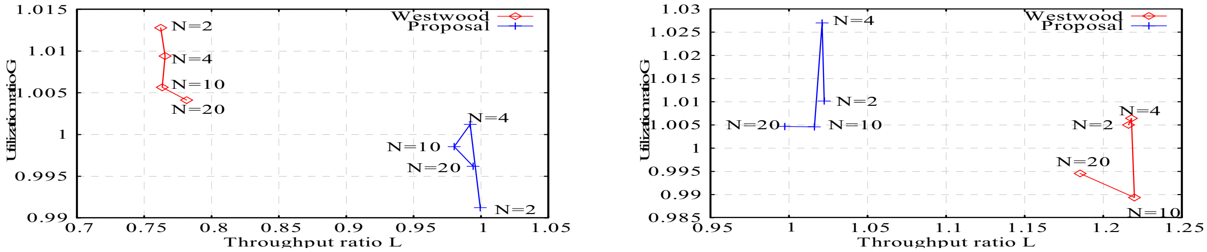
2) Another simulation with half of the flows replaced by either TCP Westwood or our proposal scheme is executed. New throughput and link utilization are measured.

Let t_{Reno1} be the average throughput of flows in the first simulation, and U_1 be the total link utilization. Similarly, let t_{Reno2} be the average throughput of TCP Reno flows in the second simulation, and U_2 be the total link utilization.

We then define next two parameters by

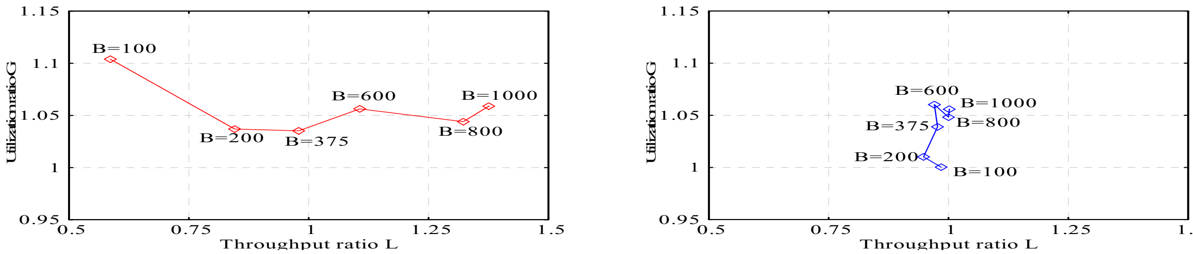
$$\begin{aligned} &\text{"Utilization Gain Ratio"}\ \ G = U_2/U_1 \\ &\text{"Legacy Protocol Throughput"}\ \ L = t_{Reno2}/t_{Reno1} \end{aligned} \tag{15}$$

For each simulation scenario, we get *G* and *L*, and plot a point *(L, G)* on the efficiency/friendliness tradeoff graph. A point falls in one of four regions in the graph as shown in Fig. 2. In the "Negative Impact Region", both *L* and *G* are less than 1, which is undesirable because the new protocol causes degradation of not only link utilization but also legacy protocol throughput. In the "Unlikely Region" both *L* and *G* are greater



(a) BDP/2 (b) 2*BDP

Fig. 3 Multi flows with two variant buffer size on Efficiency/Friendliness graph



(a) Westwood (b) Proposal

Fig. 4 Buffer size impact on Efficiency/Friendliness graph

than first simulation, which would be desirable because the new protocol helps the efficiency of link utilization and the legacy protocol throughput, but as the name suggests, it is not expected for the new protocol. In most cases, points will fall in the two tradeoff regions called "Probable Tradeoff Region" and "Improbable Tradeoff Region", in which an increase in G(or L) is compensated for by a decrease in L(or G). Therefore, we expect to see the points in the region where $G>1$ and $L<1$ ("Probable Tradeoff Region"). We consider that the target points in the graph are anywhere on the line $L=1$, $G>1$, with G as large as possible.

In the first simulation scenario, the bottleneck link bandwidth is 20[Mbps], round trip propagation delay is 80[ms], and then *BDP* is 133[packets]. The corresponding efficiency/friendliness tradeoff graphs are shown in Fig. 3 (a) and (b). For each experiment we show the points for different number of competing flows, *N*. Fig. 3 (a) and (b) show the results in which the buffer size of a bottleneck link router is set to *BDP/2* (67[packets]) and 2**BDP* (266[packets]) respectively. In Fig. 3 (a), when *N* (the number of total connections) is small, TCP Westwood increases the link utilization gain, which is an advantage of TCP Westwood with small buffer sizes but TCP Reno experiences the performance deterioration (L<<1). On the other hand, in our proposal scheme, we can be noted that link utilization gain is less than TCP Westwood because our proposal scheme is set to the adapted $RTT_{\min}$ which is less than $RTT_{\min}$ of TCP Westwood to control friendliness to TCP Reno. Note that, however, our proposal scheme in this case shares the bottleneck link friendly to TCP Reno (L=1). Note that, in Fig. 3 (b), the points of our proposal scheme are closer to the target points (L=1, G>1). When *N* is large, which means serious congestion, the points of our proposal scheme are closer to the points (1, 1). On the other hand, TCP Westwood settles in points where L>>1, G=1, which is unfriendly.

In the next simulation scenario, 2 connections (= N) sharing a bottleneck link are run. The bottleneck link bandwidth is set to 50[Mbps], round trip propagation delay is 90[ms], and then BDP is 375[packets]. Similar to subsection V-A, buffer sizes are set to four different values. Simulation results are shown in Fig. 4 (a) and (b) of TCP Westwood and our proposal scheme, respectively. When the buffer size is set to BDP (375[packets]), both TCP Westwood and our proposal scheme are close to the point (1,1). TCP Westwood achieves larger utilization gain *G*, but its throughput ratio *L* changes greatly. On the other hand, our proposal scheme achieves larger utilization gain *G* less than TCP Westwood without much effect on TCP Reno.

C. Fairness between flows with different RTT

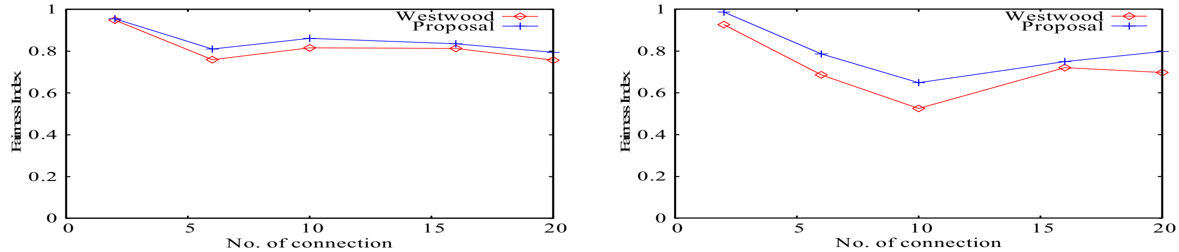
In previous subsections V-A and V-B, the round trip propagation delay of each connection is same. We then evaluate the fairness between flows with different round trip propagation delays. We consider *N*/2 TCP Reno connections and *N*/2 TCP Westwood or our proposal scheme connections; namely, *N* connections sharing the same bottleneck link with *N*=2, 6, 10, 16, 20 and round trip propagation delays ranging uniformly from 10ms+240/*N* to 250ms. The bottleneck link bandwidth is set to 10[Mbps] or 100[Mbps], and the buffer size of a bottleneck link router is set to BDP of the maximum round trip propagation delay connection; that is , the buffer size is set to 200[packets] or 2000[packets], respectively.

As simulation results, we use Jain's Fairness Index [11]. Jain's Fairness Index is calculated by

$$\text{Fairness Index} = \frac{(\sum_{i=1}^{N} b_i)^2}{N\sum_{i=1}^{N} b_i^2} \qquad (16)$$

where b_i is the throughput of each connection and N is the number of connections. If Fairness Index is equal to 1, throughputs of all connections are same. Fig. 5 (a) shows the Fairness Index as a function of the number of connections when the bottleneck link bandwidth is set to 10[Mbps]. Fig. 5 (b)

Proceedings of the Joint International Conference on Autonomic and Autonomous Systems
and International Conference on Networking and Services (ICAS/ICNS 2005)
0-7695-2450-8/05 $20.00 © 2005 IEEE



(a) 10[Mbps] (b) 100[Mbps]

Fig. 5 Jain's Fairness Index: TCP Reno and Westwood/Proposal connections sharing the same bottleneck link

shows the Fairness Index when the bottleneck link bandwidth is set to 100[Mbps]. Fig. 5 shows that our proposal scheme improves the Fairness Index than TCP Westwood. This is because, the buffer size of a bottleneck link router is set to the BDP of the connection with maximum round trip propagation delay (250[ms]), which is larger than BDP of the connections with shorter round trip propagation delay (< 250[ms]). In this case, as shown in subsection V -A, throughputs of TCP Westwood connections are degraded when the buffer size of a bottleneck link router is larger than BDP of each connection. As a consequence, when the router buffer size is too large, TCP Westwood and TCP Reno connections having the same round trip propagation delay become unfriendly, which cause TCP Westwood throughput degradation. This performance deterioration becomes remarkable when the round trip propagation delay is shorter. On the other hand, in our proposal scheme, if the round trip propagation delay of TCP Reno connections and our proposal scheme connections are equal, both connections achieve the same throughput, which is friendliness.

VI. Conclusion

In this paper, we investigated the friendliness between TCP Reno and TCP Westwood when they share the same bottleneck link. We firstly pointed out a problem through mathematical analysis and simulations; the friendliness between TCP Reno and TCP Westwood is deteriorated according to buffer sizes of a bottleneck link router. That is, when the buffer size is smaller than the bandwidth delay product, throughput of TCP Reno is degraded. On the contrary, when the buffer size is larger than the bandwidth delay product, throughput of TCP Westwood is degraded by the TCP Reno connection.

We then proposed an improved version of TCP Westwood that achieves friendliness to TCP Reno. Key points are as follows: (1) applying a bandwidth estimation technique, RCE, along with the original rate estimation technique, (2) estimating the buffer size of a bottleneck link router and deriving compensation parameters to force friendliness based on TCP throughput estimation models, and (3) updating the *ssthresh* parameter with the compensated RTT_{min} value. We have evaluated performances of the proposal scheme through various simulation experiments. Simulation results show that our proposal scheme indeed achieves friendliness with TCP Reno versions without impact of router buffer sizes.

As future work, we will try to extend our proposal scheme for high speed network. We will also consider extension of our proposal scheme to be applied to other rate based approaches such as TCP Vegas [13] and FAST TCP [20].

References

[1] W. Richard Stevens: "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, January 1997.

[2] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang: "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", *ACM Mobicom 2001*, Rome, Italy, pp.287-297, July 2001.

[3] R. Wang, M. Valla, M.Y. Sanadidi, B. K. F. Ng, and M. Gerla: "Efficiency/Friendliness Tradeoffs in TCP Westwood", *Seventh IEEE Symposium on Computers and Communications*, Taormina, Italy, July 2002.

[4] Claudio E. Palazzi: "Residual Capacity Estimator for TCP on Wired/Wireless Links," *IFIP WCC 2004*, August 2004.

[5] Robert Carter and Mark Crovella: "Measuring bottleneck link speed in packet-switched networks," Tech. Rep. 1996-006, Boston University Computer Science Department, Boston, MA, USA, March 1996.

[6] Ningning Hu and Peter Steenkiste: "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communication*, 2003.

[7] Kelly, F: "Mathematical modeling of the Internet," *Fourth International Congress on Industrial and Applied Mathematics*, July 1999.

[8] L. A. Grieco and S. Mascolo: "TCP Westwood and Easy RED to Improve Fairness in High-Speed Networks", *Seventh International Workshop on Protocols For High-Speed Networks* (PfHSN'2002), April 2002.

[9] Matthew Mathis. Jamshid Mahdavi, Sally Floyd and Allyn Romanow: "TCP Selective Acknowledgment Options," RFC 2018, October 1996.

[10] Janey C, Hoe: "Improving the Start-up Behavior of a Congestion Control Scheme for TCP," *ACM SIGCOMM 1996*, August 1996.

[11] Chiu, D.-M., and Jain, R: "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, 1989.

[12] "ns-2 network simulator(ver.2)," http://www.mash.cs.berkley.edu/ns.

[13] L.S. Brakmo and L.L. Perterson: "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet," *IEEE Journal on Selected Areas in Communication,* Vol. 13, Nov. 8, October 1995.

[14] S.Floyd: "Highspeed TCP for Large Congestion Window," RFC3649, December 2003.

[15] Tom Kelly: "Scalable TCP: Improving Perfomance in Highspeed Wide Area Networks." *First International Workshop on Protocols for Fast Long-Distance Networks*, February 2003.

[16] Ren Wang, Massimo Valla, M. Y. Sanadidi, and Mario Gerla: "Adaptive Bandwidth Share Estimation in TCP Westwood", *IEEE Globecom 2002*, November 2002

[17] Ren Wang, Giovanni Pau, Kenshin Yamada, M. Y. Sanadidi and Mario Gerla: "TCP Startup Performance in Large Bandwidth Delay Networks," *IEEE INFOCOM 2004*, March 2004.

[18] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing Router Buffers" *in Proc of ACM SIGCOMM,* October 2004.

[19] Cisco line cards, http://www.cisco.com/en/US/products/hw/modules/ ps2710/products data sheets list.html.

[20] C.Jin, D.X.Wei, S.H. Low, "FAST TCP: motivation, architecture, algorithms, performance", *in Proc of IEEE INFOCOM 2004*, March 2004.

Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS/ICNS 2005)
0-7695-2450-8/05 $20.00 © 2005 IEEE