

Improving Performance of Delay-Based TCPs with Rerouting

Cheng-Yuan Ho, *Student Member, IEEE*, Yaw-Chung Chen, *Member, IEEE*, and Cheng-Yun Ho

Abstract—Delay-based TCPs detect network congestion in the early stage and successfully prevent periodic packet loss that usually occurs in loss-based mechanisms. It has been demonstrated that delay-based algorithms outperform loss-based schemes in many aspects. However, a delay-based TCP may not prevent unnecessary throughput degradation when rerouting occurs and in mobile IP networks and mobile ad hoc networks because it could not differentiate whether the increased RTT is due to route change or network congestion. This work investigates how to improve the performance of delay-based TCPs with rerouting and proposes a mechanism for delay-based TCPs. The proposed mechanism is able to re-measure the BaseRTT if necessary by detecting the change of TTL value of two end-hosts. Based on the simulation results, we demonstrate the effectiveness and utilities.

Index Terms—Transport protocol, delay-based TCP, congestion control, rerouting, Mobile IP, and mobile ad hoc networks.

I. INTRODUCTION

WITH the fast growth of Internet traffic, how to efficiently utilize network resources is essential to a successful congestion control. Currently, most applications use the Transmission Control Protocol (TCP) to transmit data over the Internet because TCP provides reliable data transmission with embedded congestion control algorithm which effectively removes congestion collapses in the Internet by adjusting the sending rate according to the network's available bandwidth.

TCP has several implementation versions classed as two categories. One is called the loss-based approach that uses packet loss as the only indication of congestion, e.g., TCP Reno [1]. The other is named the delay-based scheme which makes congestion decisions that adjust the transmission rate based on round-trip time (RTT) variations, e.g., TCP Vegas [2] and FAST TCP [3]. Many studies have demonstrated that delay-based TCPs outperform loss-based TCPs in the aspects of overall network utilization, stability, fairness, and throughput [2], [3], [4]. However, rerouting may lead to the change of the end-to-end fixed delay, which is the sum of propagation delay and packet processing time, and therefore bring about inaccurate estimation of the minimum of ever measured RTTs (BaseRTT). This may erroneously affect the delay-based TCP's adjustment of the congestion window size (CWND) [4], [5], [6], [7]. Several works [5], [6], [7] have been proposed to improve TCP performance with rerouting, but they have some limits and drawbacks.

Manuscript received June 20, 2006. The associate editor coordinating the review of this letter and approving it for publication was Prof. Nasir Ghani.

C.-Y. Ho and Y.-C. Chen are with the Department of Computer Science, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu City, 30050, Taiwan (e-mail: {cyho, ycchen}@csie.nctu.edu.tw).

C.-Y. Ho is with the Department of Computer Science, National Cheng Chi University, No. 64, Sec. 2, Zhi Nan Road, Taipei City, 11605, Taiwan (e-mail: s9249@cs.nccu.edu.tw).

Digital Object Identifier 10.1109/LCOMM.2007.060951.

In [5] the authors proposed a modification to TCP Vegas to counteract the rerouting problem by assuming any lasting increase in RTT as a sign of rerouting. Besides the fact that this may not be a valid assumption in all cases, several new parameters K , N , L , δ , and γ were introduced in this scheme and finding appropriate values for these variables remain an unaddressed problem. Depending on the network conditions, TCP Vegas-A [6] uses adaptive values of two thresholds, α and β , instead of 1 and 3 respectively; however, when the route of a connection is changed, TCP Vegas-A may not react this event immediately and may not reach the available bandwidth after rerouting occurs. In order to know whether the end-to-end fixed delay is changed, RoVegas [7] obtains the forward and backward queuing times by performing its accumulate queuing time (AQT) scheme in routers along the round-trip path. However, this mechanism works well only when all bottleneck routers along the round-trip path are AQT-enabled.

In this work, we investigate how to improve the performance of delay-based TCPs with rerouting and propose a mechanism for delay-based TCPs. The proposed mechanism is able to re-measure the BaseRTT if necessary by detecting the change of Time To Live value (TTL) of two end-hosts. Based on the simulation results, the proposed mechanism may improve the TCP performance with rerouting and in Mobile IP (MIP) networks [8] and mobile ad hoc networks (MANET) [9].

The rest of this paper is organized as follows. The proposed mechanism is described in Section II. Section III shows the simulation results. Finally, Section IV summarizes this work.

II. THE PROPOSED MECHANISM

Different from loss-based mechanisms, which detect network congestion based on packet losses, delay-based TCPs control the congestion window size based on BaseRTT and the average RTT (or the average queueing delay). For example, TCP Vegas estimates a proper amount of extra data (Δ) to be kept in the network pipe. Moreover, Δ is between two thresholds α and β , as shown in the following:

$$\alpha \leq (Expected - Actual) \times BaseRTT \leq \beta, \quad (1)$$

where *Expected* throughput is the current CWND divided by BaseRTT, and *Actual* throughput represents the CWND divided by the newly measured RTT. The CWND is kept constant when Δ is between α and β . If Δ is greater than β , it is taken as a sign for incipient congestion, thus the CWND will be reduced. Otherwise, the connection may be under utilizing the available bandwidth. Hence, the CWND will be increased. Similarly, FAST adjusts its congestion window according to:

$$W \leftarrow \min\{2W, (1 - \gamma)W + \gamma(\frac{BaseRTT}{RTT}W + \alpha)\}, \quad (2)$$

where W is the CWND, $\gamma \in (0, 1]$, and α is a positive protocol parameter that determines the total number of packets queued

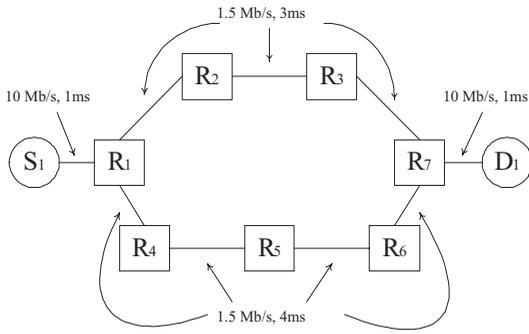


Fig. 1. A single network topology for the rerouting simulation.

in routers in equilibrium along the flow's path. In addition, FAST TCP can be regarded as a scaled version of Vegas [3].

In a wired network, when a link fails or the heavy congestion is encountered at some intermediate switching point in the connection, traffic that was using the failed link must change its path in order to reach its destination. This is so-called rerouting. Similarly, the rerouting may happen when a mobile node (MN) leaves its current network to another network in MIP networks or mobile routers (MR) are free to move randomly and organize themselves arbitrarily in MANETs.

Accordingly, when the route of a connection is changed, if the new route has a shorter fixed delay, it will not cause any serious problem for delay-based TCPS because most likely some packets will experience shorter RTT, and BaseRTT will be updated eventually. On the other hand, if there is a longer fixed delay in the new route, it would be unable to tell whether the increased RTT is due to network congestion or route change. The source host may misinterpret the increased RTT as a signal of congestion in the network and decrease its window size. This is just the opposite of what the source should do.

The Time To Live (TTL) is a timer field (8 bits) in the IP header which indicates how long a packet should be allowed to survive before it is discarded and TTL essentially determines the maximum number of hops permitted. In other words, when the TTL field is decremented down to zero, the packet is discarded. Moreover, if the routing path is not changed, the TTL value of each packet will be the same. Therefore, we modify the end hosts including both the sender and receiver to detect the change of TTL value in our proposed mechanism.

In the sender side, while the sender knows that the TTL value of an acknowledgement (ACK) is changed, it will re-measure the BaseRTT. In the receiver side, a receiver will tell the sender to re-measure BaseRTT when the TTL value of a data packet is varied. Thus, we use a reserved bit in TCP header, called "Re-measure BaseRTT" (RB) bit, to represent the change of the routing path. The sender records the value of 'RB' and the receiver will keep the 'RB' bit value of each ACK unchanged until the TTL value of a data packet is changed again. When the value of 'RB' bit changes from 0 to 1 (or from 1 to 0), it means that the forward routing path has been changed. In the beginning, the value of 'RB' bit is set to 0. Once a receiver detects the change of the forward routing path, it will alter the 'RB' bit value to inform the source to re-measure the BaseRTT. Whenever a sender receives an ACK,

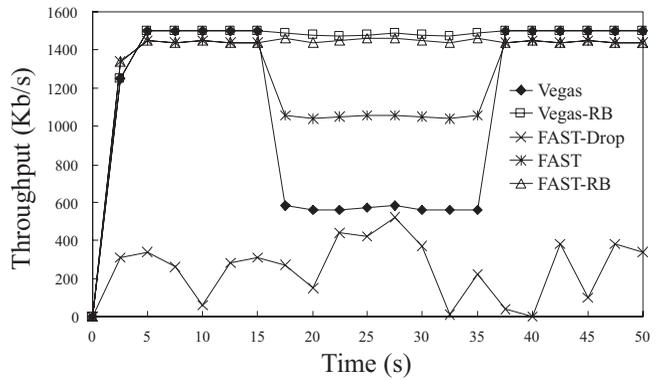


Fig. 2. Throughput comparison between different TCPS with rerouting.

it compares the value of the 'RB' bit with its current value. If they are different, the sender will re-measure the BaseRTT. In addition, a delay-based TCP with the proposed mechanism is called xxx-RB, where xxx is the name of a delay-based TCP, like Vegas-RB and FAST-RB.

This mechanism also works well in MIP networks and MANETs because the TTL value of a packet must be decreased by one as it passes through a router. Moreover, the inner IP header is not changed by the encapsulator except the TTL value, although the IP-in-IP encapsulation may be used in MIP networks and MANETs [8], [9]. In other words, no matter what encapsulation scheme is used, the TTL value is decreased by one when a packet passes through a router.

III. PERFORMANCE EVALUATION

In this section, we compare the performance of TCP Vegas-RB with TCP Vegas and FAST-RB with FAST by the network simulator *ns-2* [10]. The FIFO service discipline is assumed. The size of data packet is 1 Kbytes and the size of ACK is 40 bytes for Vegas, FAST, Vegas-RB, and FAST-RB. To ease the comparison, we assume that the sources are backlogged.

A. Rerouting Simulation

Figure 1 shows the first network topology. A source S_1 of Vegas, Vegas-RB, FAST, or FAST-RB sends data packets to its destination D_1 . The bandwidth and propagation delay are 10 Mb/s and 1 ms for each full-duplex access link, 1.5 Mb/s and 3 ms for the full-duplex trunk link from R_1 to R_2 , from R_2 to R_3 and from R_3 to R_7 , and 1.5 Mb/s and 4 ms for the full-duplex trunk link from R_1 to R_4 , from R_4 to R_5 , from R_5 to R_6 and from R_6 to R_7 . In the beginning, the packets are routed through S_1 , R_1 , R_2 , R_3 , R_7 , and D_1 in order. At 15th second, the connection link from R_2 to R_3 is broken and then recovered at 35th second. Therefore, the packets pass through the other path from 15th till 35th second. As shown in Fig. 2, when the packets are routed through the path with shorter RTT, Vegas achieves high throughput and stabilizes at 1.5 Mb/s. However, the performance of Vegas degrades dramatically as the packets are rerouted through the other path. On the other hand, Vegas-RB always maintains a steady throughput regardless of the route change. Similarly, the throughput of FAST-RB is higher than that of FAST when the routing path is changed. In addition, in order to make these

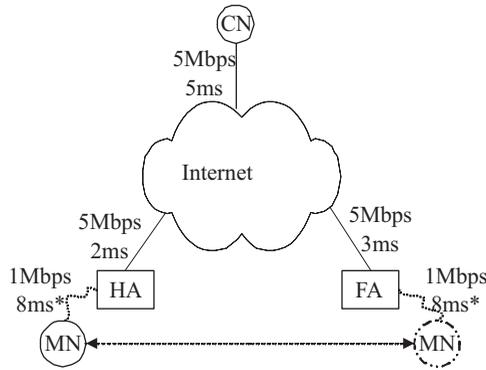


Fig. 3. A single network topology for the Mobile IP simulation.

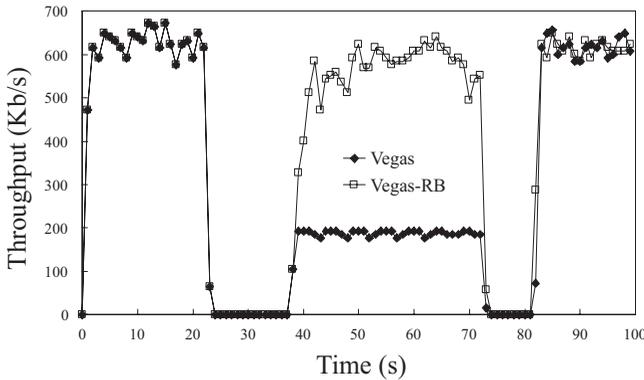


Fig. 4. Throughput comparison between Vegas and Vegas-RB with MIP.

four TCPs be under the steady state, the size of each FIFO queue used in routers is set 110 packets because FAST TCP requires more buffer size (at least 107 packets). If the buffer size is not large enough, the performance of FAST will be bad as “FAST-Drop” in Fig. 2.

B. Mobile IP Simulation

A simulation network topology is shown in Fig. 3, where correspondent node (CN) and MN represent end hosts. CN and MN are the two end sides which execute either Vegas (or FAST) or Vegas-RB (or FAST-RB). The application service in our simulation is FTP. The receiver sends an ACK for every data packet received. A home agent (HA) and a foreign agent (FA) represent finite-buffer gateways. The buffer size in each gateway is set to 50 packets. The bandwidth is 5 Mbps for all wired links. The propagation delay is 5 ms from CN to the Internet, 2 ms from the Internet to HA, and 3 ms from the Internet to FA, respectively. From an agent (HA or FA) to MN, the bandwidth is 1 Mbps and wireless transmission delay is a multiple of 8 ms which includes both packet transmission delay and the propagation delay. The former may account the layer 2 retransmission due to unsuccessful frame delivery, while the later can be ignored because the propagation delay is much smaller comparing with the packet transmission delay. In addition, this is a two dimensional plane in topology. The distance of radio coverage for the agent is 75 meters. The position of HA is (200, 325) and FA is (350, 325).

The MN moves with a speed of 10 m/s from (130, 300) to (420, 300) at the 10th second and starts to come back at

the 60th second. When MN is in the foreign network, the datagrams are routed from CN to HA, tunneled from HA to FA, and FA de-tunnels these datagrams to MN. The ACKs are routed directly from MN to CN through the FA. In Vegas, the sender is unable to tell the routing path change from congestion, so the latter is assumed. Therefore, it reduces the congestion window size and thus the throughput decreases. While in Vegas-RB, the sender is able to detect the change of routing path through ‘RB’ bit and consequently re-measures the BaseRTT. The throughput of Vegas and Vegas-RB are shown in Fig. 4, where we can observe that Vegas-RB outperforms Vegas when the MN is in the foreign networks. Since the space is limited, we only show the results of configuration with a fixed sender and a mobile receiver. It is because the diagrams of configuration with a fixed receiver and a mobile sender, or two mobile nodes are just like that with one fixed node and one mobile node. In addition, the outcome of comparing FAST-RB with FAST is similar to the above and the space is limited, so the simulation results of FAST and FAST-RB in MIP networks and MANET are omitted.

IV. CONCLUSIONS

In this research, we investigate how to improve the performance of delay-based TCPs with rerouting and propose a mechanism for them. The proposed mechanism is able to re-measure the BaseRTT if necessary by detecting the change of TTL value of two end-hosts. Comparing with other previous studies, delay-based TCPs with RB provide a more effective way to improve the connection throughput with rerouting and in Mobile IP and mobile ad hoc networks. The simulation results show the effectiveness of our proposed mechanism. Nevertheless, this approach may not work well when the old route and the new route have same hop count and the RTT of the new route is longer than that of the old route. This issue and how to enable compatibility between delay-based and loss-based TCPs will be discussed in our future work.

REFERENCES

- [1] V. Jacobson, “Modified TCP congestion avoidance algorithm,” Tech. Rep., Apr. 1990.
- [2] L. S. Brakmo and L. L. Peterson, “TCP Vegas: end to end congestion avoidance on a global Internet,” *IEEE J. Sel. Areas Commun.*, vol. 13, pp. 1465-1480, 1995.
- [3] C. Jin, D. X. Wei, and S. H. Low, “FAST TCP: motivation, architecture, algorithms, performance,” in *Proc. IEEE INFOCOM 2004*, vol. 4, pp. 2490-2501.
- [4] J. Mo, R. J. La, V. Anantharam, and J. Walrand, “Analysis and comparison of TCP Reno and Vegas,” in *Proc. IEEE INFOCOM 1999*, vol. 3, pp. 1556-1563.
- [5] R. J. La, J. Walrand, and V. Anantharam, “Issues in TCP Vegas,” UCB/ERL memorandum, no. M99/3, Electronics Research Laboratory, University of California, Berkeley, Jan. 1999.
- [6] K. N. Srijith, L. Jacob, and A. L. Ananda, “TCP Vegas-A: improving the performance of TCP Vegas,” *Computer Commun.*, vol. 28, no. 4, pp. 429-440, Mar. 2005.
- [7] Y.-C. Chan, C.-T. Chan, and Y.-C. Chen, “RoVegas: a router-based congestion avoidance mechanism for TCP Vegas,” *Computer Commun.*, vol. 27, no. 16, pp. 1624-1636, Oct. 2004.
- [8] C. Perkins, “IP mobility support for IPv4,” IETF RFC 3344, Aug. 2002.
- [9] S. Corson and J. Macker, “Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations,” IETF RFC 2501, Jan. 1999.
- [10] <http://www.isi.edu/nsnam/ns/>