An Enhancement of TFRC over Wireless Networks

Bin Zhou, Cheng Peng Fu, Chiew Tong Lau, and Chuan Heng Foh
School of Computer Engineering
Nanyang Technological University, Singapore 639798
Email: {zhou0022, ascpfu, asctlau, aschfoh}@ntu.edu.sg

Abstract— **TFRC is TCP-Friendly Rate Control protocol with TCP Reno's throughput equation based. It is designed to mainly provide optimal service for unicast multimedia flow operating in the best-effort Internet environment. However, due to Reno's significant performance degradation in wireless networks, TFRC has also led to unavoidable performance suffering from wireless links due to the poor performance of Reno throughput. This paper proposes an enhancement of TFRC based on the differentiating method used in TCP Veno. Specifically, it utilizes Veno's state differentiator to distinguish congestion losses and non-congestion losses during transmission. By discounting the impact of those non-congestion losses on the throughput calculation, it can effectively alleviate the throughput degradation caused by wireless links. The simulation results have shown that, our proposal can achieve throughput improvement up to 70% as compared to the original TFRC. Meanwhile, it maintains other merits of the original TFRC, such as sending rate smoothness, fairness, and TCP-friendliness.**

I. Introduction

In recent years, multimedia applications such as VoIP and streaming video have experienced rapid growth. These applications used to be built over UDP protocol and implemented with no form of congestion control. Therefore, the widespread deployment of them may harm the performance of the competing TCP applications (e.g., FTP, HTTP, SMTP), and even lead to congestion collapse of the Internet [1]. To avoid the potential threat to the health of the Internet, the TCP-friendly Rate Control (TFRC) protocol is proposed [3]. It uses an equation on the modeling of TCP Reno [2] to adjust the sending rate of the application. Such equation makes its steady state sending rate be roughly equal to that of a TCP Reno connection experiencing similar network conditions, without bias behavior to the competing TCP connections. Meanwhile, it avoids the abrupt fluctuation in the rate, which is especially attractive to the multimedia applications.

However, keeping the steady state sending rate equal to that of a TCP Reno flow also makes TFRC suffer the same problems that TCP Reno has encountered, e.g., the performance degradations over wireless networks. This is because both TFRC and TCP Reno interpret packet loss as an indication of congestion. TFRC will reduce its sending rate for every loss, which is not necessary in wireless networks since a high percentage of losses are not caused by congestion.

Recently, some schemes have been proposed to address this issue. WM-TFRC [13] uses the access point (AP) in wireless LAN to measure the rate of wireless loss events (i.e., loss events caused by bit error) p_w and feeds back its value to the sender periodically. Meanwhile, the receiver also provides feedback about the rate of total loss events (including wireless loss and congestion loss) p to the sender. Therefore, the sender can deduce the rate of congestion loss events (i.e., loss events caused by congestion) p_{con} by subtracting p_w from p. V. Arya proposed Accurate and Explicit Differentiation (AED) [14] by assuming that agents are deployed before and after each wireless link. The agents snoop through each packet and detect a loss by finding a packet with an out-of-order sequence number. Agents at the end of wired networks treat a packet loss as congestion loss and agents after a wireless link treat a packet loss as wireless loss. The agents then mark the packets that are not lost with information about the lost packets (i.e., whether those packets were lost due to congestion or bit error.). Thus, when receiving these marked packets, the receiver can calculate the correct loss event rate and provide feedback to the sender. ECN-based TFRC [15] calculates sending rate based on ECN-marked packet probability rather than packet loss probability, thus effectively eliminating the effect of wireless losses.

All the above proposals try to differentiate non-congestion losses from congestion losses and mask them from the calculation of sending rate. However, all of them require the supports from the intermediate nodes (APs or routers), which are not easy to be deployed in the existing network facilities.

In this paper, we propose an end-to-end enhancement of TFRC based on the differentiating method used in TCP Veno [4]. Specifically, our enhancement applies the state differentiator of TCP Veno to distinguish non-congestion losses and congestion losses. Non-congestion losses will make less contribution to the calculation of the sending rate than congestion losses do. Our simulation results show that, such Veno-based modification can improve TFRC performance up to 70% over wireless networks. Furthermore, it does not require any supports from the intermediate nodes.

The rest of this paper is organized as follows: Section II describes the basic mechanism of TFRC. Then Section III modifies TFRC based on Veno's state differentiator. In Section IV, we evaluate the performance of our enhancement based on NS-2 [7] simulation experiments. Finally, Section V summarizes the paper.

II. Basic Mechanism of TFRC

A brief mechanism of TFRC can be illustrated in Fig 1.

Once packet losses occur, the receiver will estimate the packet loss rate p (the method will be detailed later), and acknowledge it to the sender. Then the sender uses the new

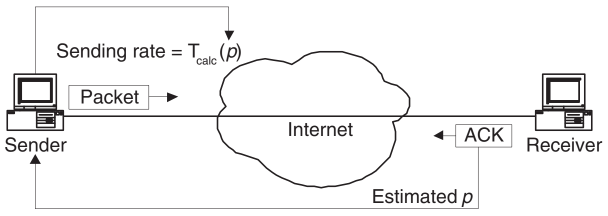


Fig. 1. Basic mechanism of TFRC.

packet loss rate to calculate its sending rate based on the following equation [3]:

$$T_{calc} = \frac{S}{RTT\sqrt{\frac{2bp}{3}} + 3RTO\sqrt{\frac{3bp}{8}}p(1+32p^2)} \tag{1}$$

where T_{calc} is the expected sending rate in bytes/sec, S is the packet size in bytes, RTT is the round-trip time, b is the number of packets that are acknowledged by a received ACK, p is the steady-state rate of loss event, and RTO is the retransmit timeout value.

The essence of TFRC is the measurement of p, which is also called loss event rate. Its principle can be described as follows. The receiver judges a packet is lost when three packets with higher sequence number than it arrive. The term loss event refers to several packets lost within one round-trip time. Note that the subsequent losses following the first loss in the round-trip time are ignored, i.e., at most one loss event in one round-trip time. The term loss interval is defined as the number of packets between loss events. The value of a loss interval is obtained by subtracting the sequence number of the first lost packet in a loss event from the sequence number of the firs lost packet in the subsequent loss event. Figure 2 illustrates the relationship between loss events and loss intervals.

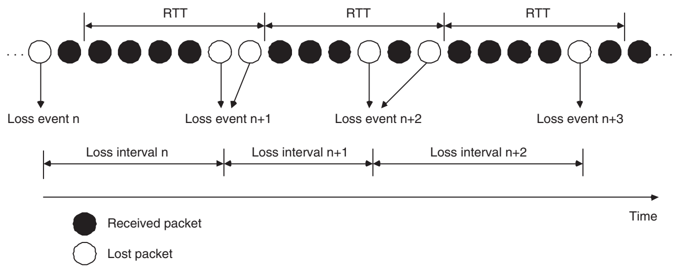


Fig. 2. An example of loss events.

Let A_i be the ith recent loss interval, and let w_i be the weight of A_i. The receiver calculates the average loss interval of the recent n loss intervals as follows:

$$A(1,n) = \frac{\sum_{i=1}^{n} w_i A_i}{\sum_{i=1}^{n} w_i} \tag{2}$$

In TFRC, the default value of n is 8, and $w_1 = w_2 = w_3 = w_4 = 1, w_5 = 0.8, w_6 = 0.6, w_7 = 0.4, w_8 = 0.2$. Let $w = 1/\sum_{i=1}^{n} w_i = \frac{1}{6}$, we can rewrite equation (2) as

$$A(1,n) = w\sum_{i=1}^{n} w_i A_i \tag{3}$$

In the calculation of $A(1,n)$, equation (2) or (3) ignores the most recent loss interval A_0, which is the number of received packets since the latest loss event. If there is no loss event in one round-trip time, A_0 will continue to increase. If A_0 is large enough, it means that there are no packet losses for a long time; therefore the average loss interval should be increased accordingly. Thus, TFRC also calculates $A(0,n-1)$, which covers A_0, according to

$$A(0,n-1) = w\sum_{i=0}^{n-1} w_{i+1} A_i \tag{4}$$

The refined value of average loss interval, A, is determined by

$$A = \max(A(1,n), A(0,n-1)) \tag{5}$$

Finally, the loss event rate, p, is given by

$$p = \frac{1}{A} \tag{6}$$

III. Veno-based Modification on TFRC

In our proposal, the loss events are divided into non-congestion losses and congestion losses based on Veno's state differentiator. Non-congestion losses will make less contribution in calculating loss event rate than congestion losses do. Thus the new value of p can indicate the congestion situation of the link more correctly. In this section, we first describe Veno's state differentiator, and then modify the original TFRC mechanism based on it.

A. *Veno's State Differentiator*

TCP Veno innovatively makes use of congestion detection scheme in TCP Vegas [5] to differentiate the state of TCP connection. We named this scheme as Veno's state differentiator. By using such differentiator to distinguish non-congestion or congestion losses, TCP Veno can achieve much higher throughput than TCP Reno does [4] [10] [11]. Specifically, the sender measures the so-called $Expected$ and $Actual$ rates:

$$Expected = \frac{cwnd}{BaseRTT}$$

$$Actual = \frac{cwnd}{RTT}$$

where $cwnd$ is the current congestion window size, $BaseRTT$ is the minimum of measured round-trip time, and RTT is the smoothed round-trip time measured. The difference of the rate is:

$$Diff = Expected - Actual$$

When $RTT > BaseRTT$, there is a bottleneck link where the packets of the connection accumulate. Let the backlog at the queue be denoted by N. We have:

$$RTT = BaseRTT + \frac{N}{Actual}$$

That is, we attribute the extra delay to the bottleneck link in the second term of the right side above. Rearranging, we have:

$$\begin{aligned} N &= Actual \times (RTT - BaseRTT) \\ &= \frac{cwnd}{RTT} \times (RTT - BaseRTT) \end{aligned} \quad (7)$$

The main idea of Veno is to use the measurement of N as an indication of whether the network is in congestive state or non-congestive state. At any time, if $N \geq \beta$, Veno deduces the link is in congestive state and considers the packet loss is congestion loss. Otherwise if $N < \beta$, the link is in non-congestive state and the packet loss is non-congestion loss. In TCP Veno, β is normally set to be 3. However, after a complete evaluation on different values of β in the simulation experiments, we find $\beta = 1$ is a better choice for our enhancement on achieving a good tradeoff between throughput and TCP-friendliness (referring to Section IV).

B. Modification of TFRC

To use Veno's state differentiator to distinguish non-congestion losses from congestion losses, the receiver needs to know whether the network is in non-congestive state ($N < \beta$) or congestive state ($N \geq \beta$). Such information (N) can be obtained from the sender. According to equation (7),

$$N = T_{current} \times (RTT - BaseRTT) \quad (8)$$

Here $T_{current} = \frac{cwnd}{RTT}$, which can be regarded as the current sending rate. The sender measures the $BaseRTT$ and RTT, and then calculates the value of N using $T_{current}$. After that, it sends N to the receiver.

When the receiver detected a new loss event i, it will calculate the loss interval $\overline{A}_i$ based on the current value of N. If $N \geq \beta$, it deduces such loss event is congestion loss event, and calculate the loss interval $\overline{A}_i$ as described in Section II, i.e., $\overline{A}_i = A_i$. If $N < \beta$, however, it deduces the loss event is non-congestion loss event. To reduce the contribute of non-congestion loss event to the whole loss event rate p, we extend the current loss interval A_i by certain times α, i.e., $\overline{A}_i = \alpha A_i$. According to equation (3)–(6), the larger loss interval ($\alpha > 1$) results the smaller loss event rate p.

To determine the value of α, consider the following extreme scenario: if all loss events are non-congestion loss events, then

$$\overline{A}_i = \alpha A_i \qquad \forall i \quad (9)$$

According to equation (3)–(6) we have

$$\overline{A}(1,n) = w\sum_{i=1}^{n} w_i \overline{A}_i = \alpha w \sum_{i=1}^{n} w_i A_i = \alpha A(1,n) \quad (10)$$

$$\overline{A}(0,n-1) = w\sum_{i=0}^{n-1} w_{i+1} \overline{A}_i = \alpha w \sum_{i=0}^{n-1} w_{i+1} A_i = \alpha A(0,n-1) \quad (11)$$

$$\overline{A} = \max(\overline{A}(1,n), \overline{A}(0,n-1)) = \alpha \max(A(1,n), A(0,n-1)) = \alpha A \quad (12)$$

and,

$$\overline{p} = \frac{1}{\overline{A}} = \frac{1}{\alpha A} = \frac{1}{\alpha} p \quad (13)$$

According to [6], the throughput of TCP Veno is $\sqrt{3}$ times of that of TCP Reno if all the losses are non-congestion losses and all the time-outs are ignored. Then we have:

$$\overline{T} = \frac{1}{RTT}\sqrt{\frac{3}{2b\overline{p}}} = \frac{1}{RTT}\sqrt{\frac{3\alpha}{2bp}} = \sqrt{3}T = \frac{1}{RTT}\frac{3}{\sqrt{2bp}} \quad (14)$$

From equation (14) we get:

$$\alpha = 3 \quad (15)$$

In summary, the loss interval $\overline{A}_i$ is calculated as follows:

$$\overline{A}_i = \begin{cases} A_i, & N \geq \beta \\ 3A_i, & N < \beta \end{cases}$$

IV. Performance Evaluation

The evaluation of our enhancement (hereinafter, called "Enhancement") is based on NS-2 simulation experiments. Different aspects of performance such as throughput, receiving ratio, sending rate variation, fairness, and TCP-friendliness, have been extensively studied in the following section.

The topology of our experiment on NS-2 is shown in Figure 3.

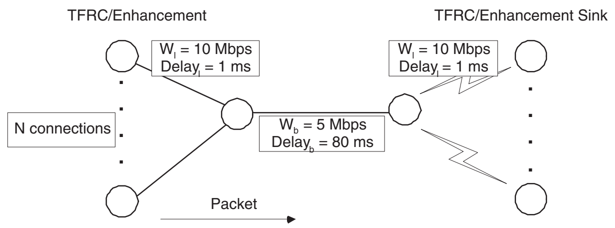
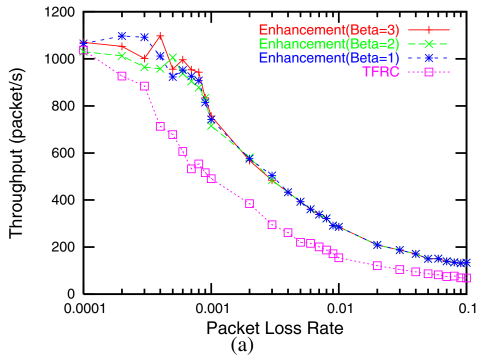


Fig. 3. Topology of NS-2 experiments.

The left side network has wired links with bandwidth of 10Mbps, delay of 1ms and packet buffer size of 50. The right side network has wireless links with bandwidth of 10Mbps, delay of 1ms and packet buffer size of 50. Random losses in wireless links follow exponential distribution and the packet loss rate ranges from 0.0001 to 0.1. The bottleneck link between these two networks is a wired link. The bandwidth is 5Mbps and delay is 80ms. Its packet buffer size is set to be 20. TFRC packets are transferred from the wired network to the wireless network. Packet size is 1000Byte.

A. Throughput

At first we let a single TFRC or Enhancement flow run over the network. Meanwhile, we set up three UDP connections with pareto distribution [8] over the bottleneck link as the background traffic. The packet size of UDP is 512Byte. The burst time and idle time are both 100ms. The rate of each connection is set to 500Kbps. The experiment time is 300s. Figure 4(a) plots the throughput results of TFRC and Enhancements with different values of β (1, 2, and 3). As shown in the figure, different values of β bring a little difference in the



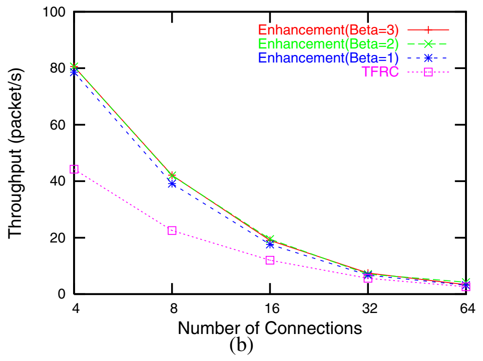
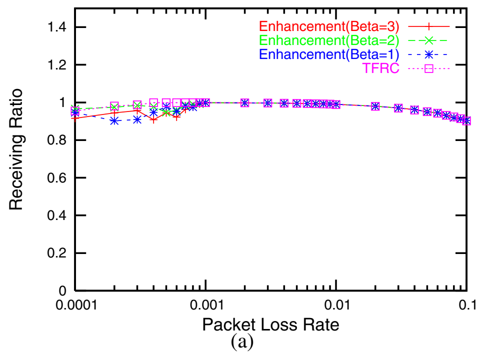


Fig. 4. Throughput comparison. (a) Single flow, $W_b = 5Mbps$, $Delay_b = 80ms$. (b)Multiple flows, $W_b = 5Mbps$, $Delay_b = 80ms$, random loss rate is 0.01.



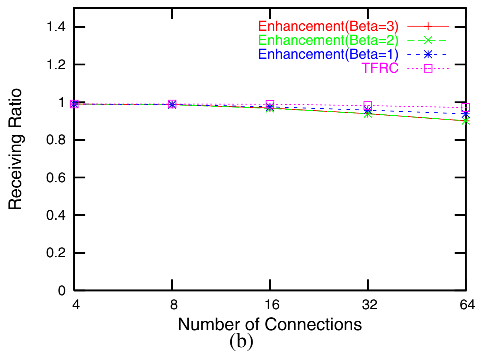


Fig. 5. Receiving ratio comparison. (a) Single flow, $W_b = 5Mbps$, $Delay_b = 80ms$. (b)Multiple flows, $W_b = 5Mbps$, $Delay_b = 80ms$, random loss rate is 0.01.

throughput for each Enhancement. However, all of these three Enhancements improve much throughput (up to 70%) over the original TFRC.

Then we increase the number of flows from 4 to 64 and compare the throughputs of TFRC and Enhancement when random loss rate is 0.01. Figure 4(b) observes that, as the number of connections increases, the network is dominated by congestion and Enhancement performs similarly as TFRC does finally.

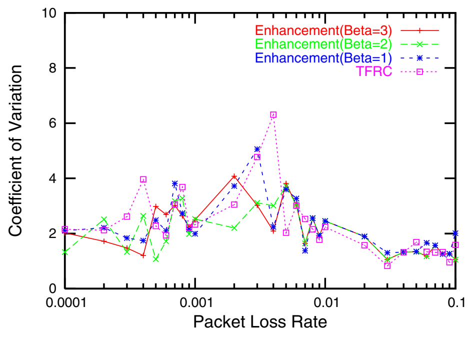


Fig. 6. CoV comparison.

B. Receiving ratio

We also compare the receiving ratio between our enhancement and TFRC during the experiments in Section IV-A. Receiving ratio is defined as follows:

$$\text{Receiving Ratio} = \frac{\text{Number of packets received}}{\text{Number of packets sent}}$$

which indicates how many packets are lost during transmission.

From Figure 5(a) and 5(b) we can see, Enhancement almost keeps the same receiving ration as that of TFRC under single connection. However, as the number of connections increase, Enhancement becomes losing a few more packets than TFRC does. This is because of the accuracy of Veno's state differentiator. As stated in [12], Veno's state differentiator works well under light load networks, but will misinterpret some congestion losses as non-congestion losses when the network load is heavy. The larger the threshold β is, the more misinterpretations Veno's differentiator has. In this case, our proposal based on Veno's state differentiator will overestimate the sending rate and thus causes a few more packet losses during transmission. Furthermore, the larger the threshold β is, the higher the overestimated sending rate is, and the more packet losses our enhancement has.

C. Sending rate variation

As a protocol for transmitting streaming multimedia, the sending rate of TFRC should not be too much variable. Following [3], we use coefficient of variation (CoV) to measure this characteristic. CoV is defined as the standard deviation

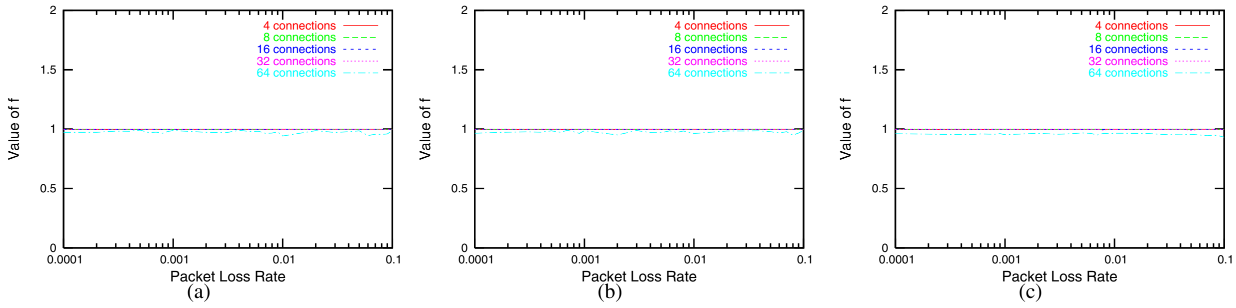


Fig. 7. Fairness of Enhancement with (a) $\beta = 3$, (b) $\beta = 2$, and (c) $\beta = 1$.

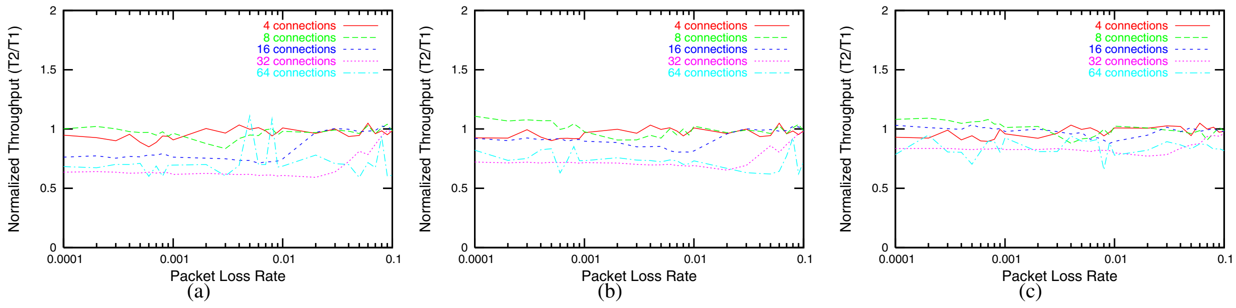


Fig. 8. Friendliness of Enhancement with (a) $\beta = 3$, (b) $\beta = 2$, and (c) $\beta = 1$.

divided by the mean. The smaller CoV is, the smoother sending rate is.

In the experiment, at first 4 TFRC flows run with 4 TCP SACK flows for 300s. Then we replace 4 TFRC flows with Enhancement flows and redo the experiment. We count the number of packets sent within every 0.1s as samples, and finally calculate CoV of all these samples. Figure 6 plots the results. As we see, our proposal does not sacrifice the smoothness of the sending rate.

D. *Fairness*

Fairness means the same kind of flows should share the total bandwidth fairly. To provide a single numerical measure reflecting the fair share distribution across the various connections, we use the Jain's Fairness Index f which is define in [9]:

$$f = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2}$$

where, n is the number of connections, x_i is the throughput of the ith connection. The closer f is to 1, the more fairness the flow has. Here we let 4 to 64 Enhancement connections run over the bottleneck link. As shown in Figure 7, Enhancement can keep good fairness under different packet loss rates, no matter which threshold β is used.

E. *TCP-friendliness*

TCP-friendliness measures whether Enhancement flows are aggressive or not when competing currently dominating TCP flows. It can be studied as follows: we first set up certain number (ranging from 4 to 64) of TCP SACK flows over the link, and calculate their average throughput T_1. Then we replace half of them with Enhancement flows and recalculate the average throughput of the left TCP SACK flows T_2. If $\frac{T_2}{T_1}$ equals to 1, it means Sack flows are not affected by Enhancement flows, and thus Enhancement flows are totally friendly. The closer $\frac{T_2}{T_1}$ is to 1, the more friendliness Enhancement flows have. As illustrated in Figure 8, as the threshold value β increases, Enhancement flows becomes more and more aggressive, especially over heavy load networks. The reason is the same as that described in Section IV-B – the overestimated throughput makes Enhancement aggressive. However, the friendliness of Enhancement when $\beta = 1$ is acceptable since $\frac{T_2}{T_1}$ is always around 85% even under heavy load.

V. Conclusion

This paper modifies TFRC protocol based on Veno's state differentiator. Such differentiator is used here to distinguish congestion loss events and non-congestion loss events. Our proposal discounts the impact of non-congestion loss events by extending their loss intervals by 3 times. Simulation results have shown the better performance our enhancement achieves. Meanwhile, we also study how the threshold β used in Veno's state differentiator can influence the performance of our proposal. After a complete evaluation, we find $\beta = 1$ is a better choice on achieving a good tradeoff between throughput and TCP-friendliness.

References

[1] S. Floyd, and K. Fall. "Promoting the Use of End-to-End Congestion Control in the Internet." IEEE/ACM Transactions on Networking, Aug. 1999.

[2] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. "Modeling TCP Throughput: A Simple Model and its Empirical Validation." ACM SIGCOMM 98, 1998.

[3] S. Floyd, M. Handley, J. Padhye, and J. Widmer. "Equation-Based Congestion Control for Unicast Applications." ACM SIGCOMM 2000, August 2000.

[4] C. P. Fu, and S. C. Liew. "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks." IEEE Journal of Selected Areas in Communications, Feb 2003.

[5] L. Brakmo, S. O'Malley, and L. Peterson. "TCP Vegas: New techniques for congestion detection and avoidance." ACM SIGCOMM 94, 1994.

[6] B. Zhou, C. P. Fu, D. M. Chiu, C. T. Lau, and L. H. Ngoh. "A Simple Throughput Model for TCP Veno." IEEE ICC 06, June 2006.

[7] S. Bajij, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmey, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, and D. Zappala. "Improving simulation for network research." Univ. Southern California, Los Angeles, Tech. Rep. 99-702b, 1999.

[8] http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

[9] R. Jain. "The art of computer systems performance analysis." Wiley, New York, 1991.

[10] C. L. Zhang, C. P. Fu, M. T. Yap, C. H. Foh, K. K. Wong, C. T. Lau, and M. K. Lai. "Dynamics Comparison of TCP Reno and Veno." IEEE Globecom 04, 2004.

[11] C. P. Fu, W. Lu, and B. S. Lee. "TCP Veno Revisited." IEEE Globecom 03, 2003.

[12] Z. X. Zou, B. S. Lee, and C. P. Fu. "Packet Loss and Congestion state in TCP Veno." IEEE ICON 04, 2004.

[13] J. Y. Pyun, Y. Kim, K. H. Jang, J. A. park, and S. J. Ko. "Wireless Measurement Based Resource Allocation for QoS Provisioning over IEEE 802.11 Wireless LAN." IEEE Trans. on Consumer Electronics, vol. 49, pp. 1103-1127, 2003.

[14] V. Arya, and T. Turletti. "Accurate and Explicit Differentiation of Wireless and Congestion Losses." in Proceedings of International Conference on Distributed Computing Systems Workshops (ICDCSW'03), May 2003.

[15] S. J. Bae, and S. Chong. "TCP-Friendly Flow Control of Wireless Multimedia using ECN Marking." Signal Processing: Image Communication, vol. 19, pp. 405-419, 2004.