An Enhancement of TCP Veno over Light-Load Wireless Networks

Bin Zhou, Cheng Peng Fu, *Member, IEEE*, Ke Zhang, Chiew Tong Lau, *Member, IEEE*, and Chuan Heng Foh

***Abstract*— This letter observes that TCP Veno behaves conservatively over light-load wireless networks. A new variable, congestion loss rate, is introduced into Veno's algorithm. It helps Veno deal with random loss more intelligently, by keeping its congestion window increasing if the link load is in light state. The simulation results demonstrate that, such enhancement can improve Veno's throughput up to 60% without any fairness or friendliness sacrificed.**

***Index Terms*— TCP Veno, random loss, congestion loss.**

I. INTRODUCTION

IT is well known that TCP Reno [1] has throughput suffering in wireless networks. Recently, a novel TCP, called TCP Veno [3], was proposed to alleviate such suffering. The key innovation in Veno is to use the estimated state of a connection [2] to differentiate congestion loss from random loss. Specifically, the number of packets on the connection (say N) is estimated until a packet loss occurs. If it is smaller than certain threshold, say β, the packet loss is deduced as random loss, and Veno cuts down its congestion window ($cwnd$) by $\frac{1}{5}$; otherwise, the packet loss is regarded as congestion loss and Veno thus cuts down its congestion window by $\frac{1}{2}$. Equation (1) describes this operation. A detailed Veno algorithm is referred to [3].

$$cwnd = \begin{cases} cwnd \times \frac{4}{5}, & N < \beta \\ cwnd \times \frac{1}{2}, & N \geq \beta \end{cases} \tag{1}$$

Many experiments have proved Veno's better performance over Reno in wireless environments [3] [4] [5]. In this paper, however, we present results showing that Veno still misses much available bandwidth, especially when the network load is light and random losses are pervasive. Further analysis points out that such conservation is due to Veno's "blind" reduction of congestion window when random loss occurs. A new variable, named *congestion loss rate*, is introduced in our proposal. It helps Veno act more appropriately in response to random losses and grab more available bandwidth on the link, without any fairness or friendliness sacrificed.

II. SIMULATION RESULTS OF VENO OVER LIGHT-LOAD WIRELESS NETWORKS

Fig. 1 gives a NS-2 simulation topology [6]. The left side network has wired links with bandwidth of 10Mbps, delay of 1ms and buffer of 50 packets. The right side network has wireless links with bandwidth of 10Mbps, delay of 1ms and buffer of 50 packets. Random loss in wireless links follows exponential distribution. The bottleneck link's bandwidth and delay will change in our experiments. Its buffer is set to be 20 packets. TCP packets are transferred from the wired network to the wireless network. Packet size is 1000Byte.

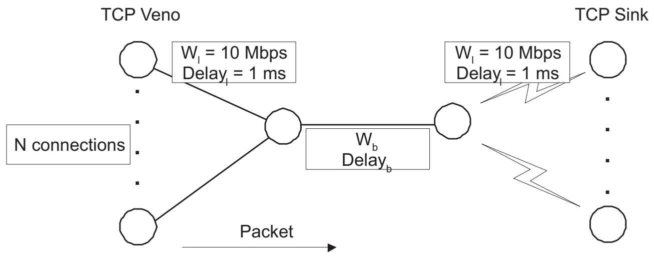

Fig. 1. Topology in NS-2 experiments.

TABLE I
THROUGHPUTS OF VENO OVER 2MBPS AND 8MBPS.

	Pr=0.001	Pr=0.005	Pr=0.01	Pr=0.05	Pr=0.1
Over 2Mbps (packets/s)	192.46	142.65	100.91	27.83	11.49
Over 8Mbps (packets/s)	320.12	153.19	104.95	28.45	11.86

To simulate the light load of the bottleneck link, we let only one TCP Veno connection run over it. The delay of the bottleneck link is set to be 80ms, and the bandwidth is 2Mbps and 8Mbps respectively.

Table I presents Veno's throughputs over 2Mbps and 8Mbps bottleneck link respectively, under different packet loss rates (Pr). It shows that when the packet loss rate is larger than 0.005, Veno running over 8Mbps bottleneck link has almost the same throughput as that over 2Mbps one. In other words, additional 6Mbps bottleneck link is not used by Veno even when there are no other competing connections.

The reason why Veno behaves so conservatively in bandwidth utilization can be easily found in its congestion window evolution, as shown in Fig. 3 (the dotted line). In light-load wireless networks, congestion losses are rare and random losses are dominating. These random losses make Veno frequently cut down its congestion window by $\frac{1}{5}$, and keep $cwnd$ value at a low level. In other words, such frequent random losses prevent Veno touching the equilibrium point during transmission. As a result, Veno wastes much available bandwidth during transmission.

According to the above analysis, a "blind" reduction of congestion window for random loss regardless of its congestion

Accepted
IEEE Communications Letters 2006.

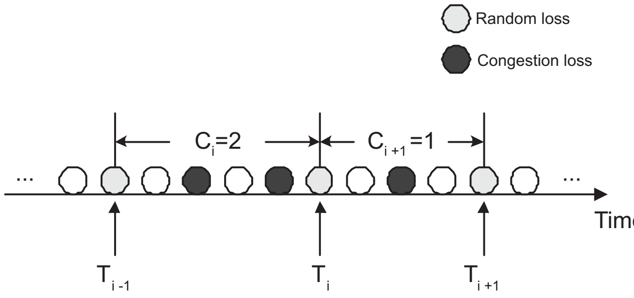

Fig. 2. Calculation of congestion loss rate.

context is harmful to the improvement of Veno's throughput. A more "intelligent" algorithm is needed to deal with these random losses, rather than always cutting down $\frac{1}{5}$ congestion window.

III. Congestion Loss Rate

We keep Veno's method to estimate the backlog (N) at the link and deduce a packet loss is congestion loss ($N \geq \beta$) or random loss ($N < \beta$). However, a new variable, called congestion loss rate, is introduced to help Veno adjust its congestion window in more smart way. Note that congestion loss rate is calculated as follows:

Considering a sequence of random losses during transmission $\{T_i\}$, where T_i is the time at which random loss occurs, we count the number of congestion losses occurring between two consecutive random losses T_{i-1} and T_i, called C_i. Then congestion loss rate at this moment con_r_i can be calculated as follows:

$$con\_r_i = \frac{C_i}{T_i - T_{i-1}} \tag{2}$$

If $con\_r_i > con\_r_{i-1}$, which means the rate of congestion loss occurrence increases since the last time, then we assume the network state is becoming congestive, and cut down $\frac{1}{5}$ of congestion window at random loss T_i. Otherwise if $con\_r_i \leq con\_r_{i-1}$, which means the congestion of networks is not becoming worse, then we keep the value of congestion window unchanged at T_i.

In summary, our enhancement of Veno (hereinafter, called "Enhanced Veno") acts as follows:

$$cwnd = \begin{cases} cwnd, & N < \beta \text{ and } con\_r_i \leq con\_r_{i-1} \\ cwnd \times \frac{4}{5}, & N < \beta \text{ and } con\_r_i > con\_r_{i-1} \\ cwnd \times \frac{1}{2}, & N \geq \beta \end{cases} \tag{3}$$

As shown in Fig. 3 (the solid line), over a light-load network con_r_i is always 0 during transmission, except of occasional pulses. Thus $cwnd$ value will keep increasing at most random losses and help Enhanced Veno absorb as much available bandwidth as possible.

It is worthy to emphasize that, as the network becomes more and more congestive, and congestion becomes the main reason of packet losses, our proposal turns to behave similarly as original Veno does. This can be observed in the next section, where more detailed experiments are conducted.

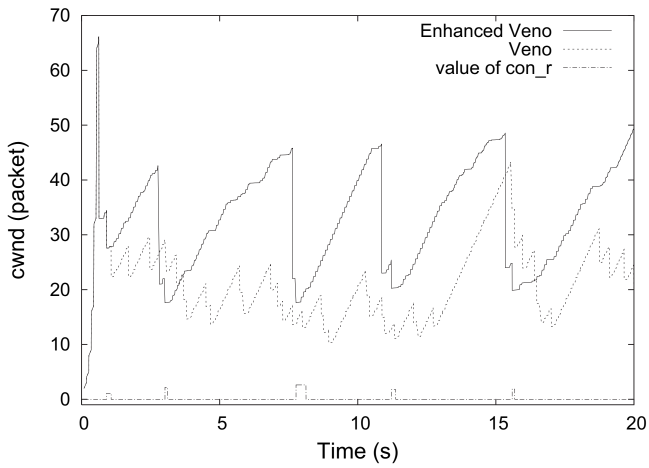

Fig. 3. Window evolution of Enhanced Veno and Veno, packet loss rate = 0.01, bottleneck bandwidth = 8Mbps.

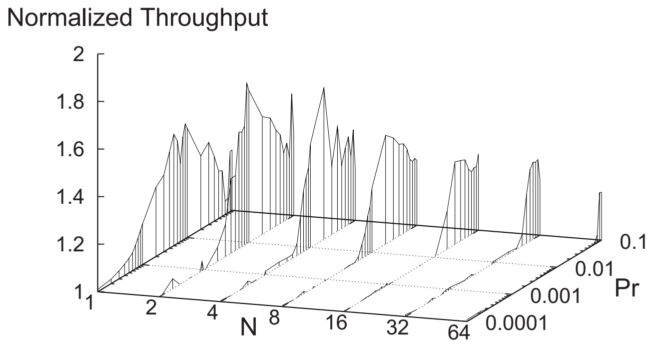

Fig. 4. Throughput improvement of Enhanced Veno under different number of connections (N) and different random loss rates (Pr).

IV. Performance Evaluation

In this section, we evaluate Enhanced Veno's performance in throughput, fairness, and friendliness, as compared to Veno. The topology and settings are same as those in Fig. 1. The bandwidth of bottleneck link is set to be 12Mbps and the delay is 80ms. Random loss rate in wireless links changes from 0.0001 to 0.1 in packet unit.

A. Throughput

A number of connections of Veno and Enhanced Veno, ranging from 1 to 64, are set up respectively to make the bottleneck link more and more congestive. In order to better compare the difference of their throughput, we define the normalized throughput TH_n as following:

$$TH_n = \frac{TH_{Enhanced}}{TH_{Veno}} \tag{4}$$

where, $TH_{Enhanced}$ and TH_{Veno} are the average throughput of Enhanced Veno flows and Veno flows respectively. This metric illustrates the percentage of improvement Enhanced Veno can obtain as compared to Veno. The higher TH_n is, the greater the improvement is.

As shown in Fig. 4, our proposal can effectively improve Veno's throughput when connection number is small (the improvement can be up to 60% when random loss is around

(3)

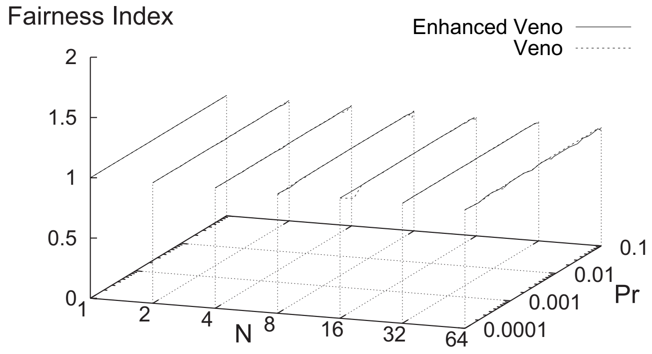

Fig. 5. Fairness comparison between Enhanced Veno and Veno.

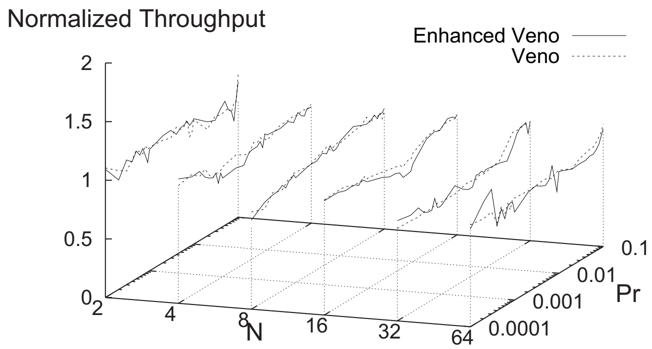

Fig. 6. Friendliness comparison between Enhanced Veno and Veno.

0.01). However, as the connection number increases, which means the network load increases, our proposal performs more and more similarly as Veno (TH_n is around 0) except under very heavy random losses. In other words, our proposal is a feasible enhancement of Veno over light-load wireless networks.

B. Fairness

Fairness means the same kind of flows should share the total bandwidth fairly. In the experiments, multiple Enhanced Veno and Veno connections (ranging from 1 to 64) are established respectively. To reflect the fairness of Enhanced Veno and Veno, we use the Jain's Fairness Index f which is define in [7]:

$$f = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2} \tag{5}$$

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 that kink of flows has. Fig. 5 plots the Fairness Index of Enhanced Veno and Veno respectively.

The values f of Enhanced Veno and Veno in the figure are almost same in each scenario, which proves that our enhancement does not harm original Veno's fairness.

C. Friendliness

Veno has shown its compatibility to the dominating version of TCP today [3]. Here we study whether our proposal has the same TCP compatibility as Veno. We first set up certain numbers (ranging from 2 to 64) of TCP Sack flows over the link, and calculate their average throughput T_1. Then we replace half of them with the objective flows (Enhanced Veno or Veno) and recalculate the average throughput of the left TCP Sack flows T_2. We define the normalized throughput F as follows:

$$F = \frac{T_2}{T_1} \tag{6}$$

If F is 1, it means Sack flows are not affected by the objective flows, and thus the objective flows are totally friendly. The closer F is to 1, the more friendliness the objective flows have.

As illustrated in Fig. 6, the value F of Enhanced Veno is always around that of Veno in different scenarios, which means our proposal does not sacrifice Veno's friendliness when improving its throughput.

V. Conclusion

This paper introduces a new variable called congestion loss rate into TCP Veno. Congestion loss rate can indicate the trend of congestion occurrence on the link and avoid "blind" reduction of congestion window at random losses, especially when the network load is light. Simulation results demonstrate such enhanced TCP Veno can obtain significant throughput improvement over TCP Veno without any fairness or friendliness sacrificed.

References

[1] V. Jacobson, "Congestion avoidance and control," SIGCOMM 88, ACM, Aug. 1988.

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

[3] C. P. Fu and S. C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE J. Select. Areas Commun.*, Feb. 2003.

[4] C. L. Zhang, C. P. Fu, M. T. Yap, C. H. Foh, K. K. Wong, C. T. Lau, and E. M. K. Lai, "Dynamics comparison of TCP Reno and Veno," in *Proc. IEEE Globecom 2004*.

[5] C. P. Fu, W. Lu, and B. S. Lee, "TCP Veno revisited," in *Proc. IEEE Globecom 2003*.

[6] S. Bajij *et al.*, "Improving simulation for network research," Univ. Southern California, Los Angeles, Tech. Rep. 99-702b, 1999.

[7] R. Jain, *The Art of Computer Systems Performance Analysis*. New York: Wiley, 1991.