A Simple Throughput Model for TCP Veno

Bin Zhou*, Cheng Peng Fu*, Dah-Ming Chiu†, Chiew Tong Lau*, and Lek Heng Ngoh‡

*School of Computer Engineering, Nanyang Technological University, Singapore 639798
Email: {zhou0022, ascpfu, asctlau}@ntu.edu.sg

†Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong
Email: dmchiu@ie.cuhk.edu.hk

‡Networking Department, Institute for Infocomm Research, Singapore 119613
Email: lhn@i2r.a-star.edu.sg

***Abstract*— TCP Veno was proposed to eliminate TCP performance suffering from wireless links. Real network measurements and live Internet results have validated TCP Veno's significant throughput improvement in wireless networks and its harmonious co-existence with TCP Reno connections in wired networks. In this paper, we develop a simple analytic approach to characterize TCP Veno behavior in both wire and wireless situations. Being different from the equation of TCP Reno, a more general close formula is derived, taking into account of the refined multiplicative decrease algorithm in Veno, to model the throughput for a bulk transfer of TCP Veno flow. Our simulation and experimental results demonstrate that such an equation is able to accurately predict TCP Veno throughput over different network scenarios, ranging from very low lossy links to very heavy lossy links.**

I. Introduction

TCP Reno [1] is the dominating version of TCP. Over last decade, many efforts have been made to model its throughput in a bulk transfer TCP flow. Among those work, a equation derived by [2] (hereinafter, called "Reno equation") has been widely used today, e.g., in the TCP-Friendly Rate Control (TFRC) protocol [3].

Recently, wireless communication technology has been making significant progress and will be playing a more and more important role in the future. Such evolution brings new challenges to TCP Reno. For example, Reno assumes packet loss is always induced by congestion, which can lead to significant performance degradation in wireless networks, where random loss is rampant due to environmental noise. To deal with random loss effectively, a novel end-to-end congestion control mechanism called TCP Veno [4][7][9][12][13] was proposed. The key innovation in Veno is the enhancement of Reno congestion control algorithm by making use of the estimated state of a connection based on Vegas [5]. This scheme significantly reduces "blind" reduction of TCP window regardless of the cause of packet loss. Thus Veno is suitable for any heterogeneous networks, particularly when wireless links form part of such networks [6][8][10][11].

In this paper, we develop a simple analytic approach to characterize TCP Veno behavior in both wire and wireless situations. Being different from the equation of TCP Reno, a more general closed form equation (hereinafter, called "Veno equation") is derived, taking into account of the refined multiplicative decrease algorithm in Veno, to model the throughput for a bulk transfer of TCP Veno flow.

The remainder of the paper is organized as follows. In Section 2, we describe TCP Veno mechanism in detail. Then Section 3 gives the derivation of Veno equation. A validation of this equation on NS-2 [14] simulation and wireless LAN experiments is presented in Section 4. Finally, Section 5 concludes our work and gives the future directions.

II. Basic Mechanism of Veno

TCP Veno makes use of the state distinguishing scheme from TCP Vegas, and integrates it into congestion window evolution scheme of TCP Reno. In Vegas, 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:

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

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

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

is in non-congestive state and the packet loss is random loss. Here β is normally set to be 3. Based on the differentiated state, Veno applies different algorithms to Reno's additive increase (AI) and multiplicative decrease (MD) phases, thus makes congestion window ($cwnd$) evolution more efficient:

In AI phase,
IF ($N < \beta$)
THEN set $cwnd = cwnd + \frac{1}{cwnd}$ for every new ACK
ELSE IF ($N \geq \beta$)
THEN set $cwnd = cwnd + \frac{1}{cwnd}$ for every other new ACK
In MD phase,
1)Retransmit the missing packet, and
IF ($N < \beta$)
THEN set $ssthresh = cwnd \times \frac{4}{5}$
ELSE IF ($N \geq \beta$)
THEN set $ssthresh = cwnd \times \frac{1}{2}$, and
set $cwnd = ssthresh + 3$.
2)Each time another dup ACK arrives, increment $cwnd$ by one packet.
3)When the next ACK acknowledging new data arrives, set $cwnd$ to $ssthresh$ (value in step 1).

Note that, Veno only refines the AIMD mechanism of Reno. All other parts of Reno, including initial slow start, fast retransmit, fast recovery, computation of the retransmission timeout, and the backoff algorithm remain intact.

III. Derivation of Veno Equation

From the mathematical view, we simplify the above AIMD mechanism as follows:
in AI phase,

$$cwnd = \begin{cases} cwnd + \frac{1}{cwnd}, \text{for every new ACK} & N < \beta \\ cwnd + \frac{1}{cwnd}, \text{for every other new ACK} & N \geq \beta \end{cases}$$

and in MD phase,

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

As shown in Figure 1, although new AI phase in Veno brings certain improvement in throughput, the most enhancement of Veno is contributed by the MD phases at random losses (i.e., cutting down $\frac{1}{5}$ rather than $\frac{1}{2}$ of $cwnd$). To simplify the problem, we ignore the different AI phases of Veno and Reno in the following derivation. As a result, Veno acts quite similarly as Reno, and the only difference between Veno and Reno is how much $cwnd$ value should be cut down when losses occur: Reno always cuts down $\frac{1}{2}$ while Veno sometimes cuts down $\frac{1}{2}$ (congestion loss), and sometimes cuts down $\frac{1}{5}$ (random loss).

In this case, we introduce a random variable λ to represent the proportion of $cwnd$ value *after* cut down, instead of a constant $\frac{1}{2}$. Here λ can be $\frac{1}{2}$ or $\frac{4}{5}$ with certain probability. This is a new parameter introduced by Veno's MD phase characteristic. The following derivation is divided into three steps: firstly we only consider the situation where packet losses

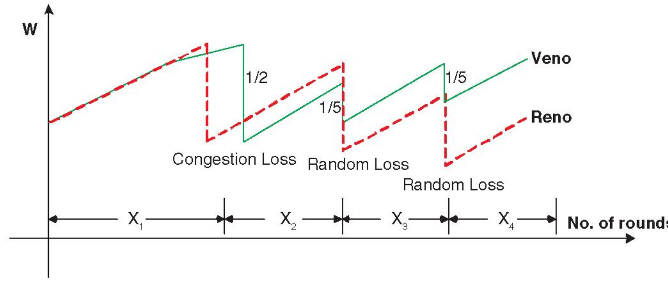

Fig. 1. Congestion window evolution of Veno and Reno.

is indicated by the triple duplicated ACKs. We call these losses triple-duplicated (TD) losses; then we include packet losses indicated by time outs, which are called time-out (TO) losses; finally we extend our model to include the impact of advertisement window limitation.

A. Veno Equation for Only TD Losses

According to equation (7) in [2] we have:

$$W_i = \lambda_i W_{i-1} + \frac{X_i}{b} \quad (1)$$

where, W_{i-1}, W_i are $cwnd$ value before the previous TD loss and the next TD loss respectively, X_i is the number of round between these two losses, and b is the number of packets that are acknowledged by a received ACK. Note that all these parameters have the same meanings as those in [2]. In addition, λ_i is the proportion of $cwnd$ value ($\frac{1}{2}$ or $\frac{4}{5}$) after the previous TD loss.

Then the packets transmitted between these two TD losses are:

$$\begin{aligned} Y_i &= \sum_{k=0}^{X_i/b-1} (\lambda_i W_{i-1} + k)b + \beta_i \\ &= \lambda_i W_{i-1} X_i + \frac{X_i}{2}(\frac{X_i}{b} - 1) + \beta_i \\ &= \frac{X_i}{2}(\lambda_i W_{i-1} + W_{i-1} - 1) + \beta_i \text{ using(1)} \quad (2) \end{aligned}$$

where β_i is the number of packets sent in the last round, as seen in [2]. As same in [2], β_i is considered as a uniformly distributed variable between 1 and W_i.

From above equation (1), (2) and equation (5) in paper [2], we have:

$$(1 - E[\lambda])E[W] = \frac{E[X]}{b} \quad (3)$$

and,

$$\frac{1-p}{p} + E[W] = \frac{E[X]}{2}((1 + E[\lambda])E[W] - 1) + E[\beta] \quad (4)$$

where, $E[\beta] = \frac{E[W]}{2}$, and p is the probability that a packet is lost, given that either it is the first packet in its round or the preceding packet in its round is not lost. Note that p includes both congestion and random losses.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

Let $\gamma = E[\lambda]$, and from (3) and (4) we can get:

$$E[W] = \frac{b(1-\gamma)+1}{2b(1-\gamma^2)} + \sqrt{\frac{2(1-p)}{b(1-\gamma^2)p} + \frac{\left(\frac{b(1-\gamma)+1}{2}\right)^2}{b^2(1-\gamma^2)^2}} \quad (5)$$

Then from above equation (3) and equation (6) in paper [2], it follows:

$$E[X] = \frac{b(1-\gamma)+1}{2(1+\gamma)} + \sqrt{\frac{2b(1-\gamma)(1-p)}{(1+\gamma)p} + \frac{\left(\frac{b(1-\gamma)+1}{2}\right)^2}{(1+\gamma)^2}} \quad (6)$$

and,

$$E[A] = RTT\left(\frac{b(1-\gamma)+1}{2(1+\gamma)} + \sqrt{\frac{2b(1-\gamma)(1-p)}{(1+\gamma)p} + \frac{\left(\frac{b(1-\gamma)+1}{2}\right)^2}{(1+\gamma)^2}} + 1\right) \quad (7)$$

Finally, substitute $E[A]$ and $E[W]$ into equation (1) and (5) in paper [2] we have:

$$\begin{aligned} B(p) &= \frac{\frac{1-p}{p} + E[W]}{E[A]} \\ &= \frac{\frac{1-p}{p} + \frac{b(1-\gamma)+1}{2b(1-\gamma^2)} + \sqrt{\frac{2(1-p)}{b(1-\gamma^2)p} + \frac{\left(\frac{b(1-\gamma)+1}{2}\right)^2}{b^2(1-\gamma^2)^2}}}{RTT\left(\frac{b(1-\gamma)+1}{2(1+\gamma)} + \sqrt{\frac{2b(1-\gamma)(1-p)}{(1+\gamma)p} + \frac{\left(\frac{b(1-\gamma)+1}{2}\right)^2}{(1+\gamma)^2}} + 1\right)} \end{aligned} \quad (8)$$

For small values of p, it can be approximated by:

$$B(p) \approx \frac{1}{RTT}\sqrt{\frac{(1+\gamma)}{2b(1-\gamma)p}} \quad (9)$$

B. Veno Equation for Both TD and TO Losses

So far equation (9) gives Veno equation when only TD losses occur. Actually there are many TO losses over real communications, and they must be included in the model. According to equation (21) in [2], with TO losses the throughput can be written as follows:

$$B = \frac{E[Y] + Q \times E[R]}{E[A] + Q \times E[Z^{TO}]} \quad (10)$$

where, $E[Y]$ and $E[A]$ have been derived in the previous section. Q is the probability that a loss indication ending a TD period is a TO, $E[R]$ is the mean number of packets sent during TO periods, and $E[Z^{TO}]$ denotes the mean duration of TO periods.

As we have pointed out before, Veno does not change Reno's algorithms on dealing with TO losses, such as the computation of the timeout, and the backoff algorithm. We can easily derive the equation following the same routine described in [2], because the expressions of Q, $E[R]$, and $E[Z^{TO}]$ are same in both Veno and Reno. Therefore, we have:

$$B(p) = \frac{\frac{1-p}{p} + E[W] + \hat{Q}(E[W])\frac{1}{1-p}}{RTT(E[X]+1) + \hat{Q}(E[W])T_0\frac{f(p)}{1-p}} \quad (11)$$

where, $E[W]$ and $E[X]$ have been given in the previous section. Note that, both $E[W]$ and $E[X]$ now include a new parameter γ here. T_0 denotes the period the sender will wait for to retransmit non-acknowledged packet,

$$\hat{Q}(w) = \min(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{w-3}))}{1-(1-p)^w}) \quad (12)$$

and,

$$f(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6 \quad (13)$$

For small values of p, equation (11) can be approximated by:

$$B(p) \approx \frac{1}{RTT\sqrt{\frac{2b(1-\gamma)p}{1+\gamma}} + T_0 \min\left(1, 3\sqrt{\frac{b(1-\gamma^2)p}{2}}\right)p(1+32p^2)} \quad (14)$$

C. Veno Equation with Impact of window limitation

Finally, we consider the Veno equation with the impact of the advertisement window limitation. In such scenario, the congestion window will start increase after the previous loss, and will keep constant when it reaches the advertisement window limitation. Let $W_{\max}$ denote the advertisement window limitation, U_i denote the number of rounds where the congestion window increases, and V_i denote the number of rounds where the congestion window keeps constant. Due to the new MD phase in Veno, $W_{\max}$ has following equation, according to [2]:

$$W_{\max} = \lambda_i W_{\max} + \frac{U_i}{b} \quad (15)$$

where, λ_i is the proportion of $cwnd$ value after the previous TD loss ($\frac{1}{2}$ or $\frac{4}{5}$), as defined in Section A. We assume $E[U] = \frac{bW_{\max}}{2}$, same as that in [2].

Then considering the number of packets sent in the ith TD period, we have:

$$Y_i = \frac{U_i}{2}(\lambda_i W_{\max} + W_{\max}) + V_i W_{\max} \quad (16)$$

and then,

$$\begin{aligned} E[Y] &= (\frac{1+\gamma}{2})W_{\max}E[U] + W_{\max}E[V] \\ &= \frac{b(1+\gamma)}{4}W_{\max}^2 + W_{\max}E[V] \end{aligned} \quad (17)$$

After getting $E[Y]$, we can easily derive the final throughput equation when the advertisement window is limited, following the steps of [2]:

$$B(p) = \frac{\frac{1-p}{p} + W_{\max} + \hat{Q}(W_{\max})\frac{1}{1-p}}{RTT(\frac{b(1-\gamma)}{4}W_{\max} + \frac{1-p}{pW_{\max}} + 2) + \hat{Q}(W_{\max})T_0\frac{f(p)}{1-p}} \quad (18)$$

Combine equation (11) and (18) we can get the final approximated result for small values of p, according to equation (31) in paper [2]:

$$B(p) \approx \min\left(\frac{W_{\max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2b(1-\gamma)p}{1+\gamma}} + T_0 \min\left(1, 3\sqrt{\frac{b(1-\gamma^2)p}{2}}\right)p(1+32p^2)}\right) \quad (19)$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

D. Expression of γ

Now we need to determine the value of γ. From our definition:

$$\begin{aligned}\gamma &= E[\lambda] \\ &= \frac{4}{5} \times P(N < \beta) + \frac{1}{2} \times P(N \geq \beta) \\ &= \frac{4}{5} \times P(N < \beta) + \frac{1}{2} \times (1 - P(N < \beta)) \\ &= \frac{1}{2} + 0.3 \times P(N < \beta) \end{aligned} \tag{20}$$

Recall:

$$\begin{aligned} N &= \left(\frac{W}{BaseRTT} - \frac{W}{RTT}\right) \times BaseRTT \\ &= W \times \frac{RTT - BaseRTT}{RTT} \end{aligned} \tag{21}$$

For simplicity, we assume the value of congestion window at losses (including congestion and random losses), W, is uniformly distributed between 0 and $W_{\max}$, where $W_{\max}$ is the maximum congestion window size during the transmission. Then we have:

$$\begin{aligned} P(N < \beta) &= P\left(W < \beta \times \frac{RTT}{RTT - BaseRTT}\right) \\ &= \min\left(1, \frac{\beta \times RTT}{(RTT - BaseRTT) \times W_{\max}}\right) \end{aligned} \tag{22}$$

From (20) and (22) we have:

$$\gamma = \min\left(\frac{4}{5}, \frac{1}{2} + 0.3 \times \frac{\beta \times RTT}{(RTT - BaseRTT) \times W_{\max}}\right) \tag{23}$$

Equation (19) with (23) is the derived Veno equation.

E. Discussion

Being different from the Reno equation, the Veno equation uses a variable $\gamma \in [\frac{1}{2}, \frac{4}{5}]$ to replace the original constant $\frac{1}{2}$. Here we call it "Veno parameter". To verify the boundary of this parameter, consider two extreme scenarios: 1) if all losses are estimated to be congestion losses during transmission, then Veno performs same as Reno by halving the congestion window every time, and the result leads to $\gamma = \frac{1}{2}$ in the equation; 2) if all losses are estimated to be random losses, then Veno performs as "super Reno" by cutting down only $\frac{1}{5}$ of the congestion window every time. The result leads to $\gamma = \frac{4}{5}$ in the equation. Because Veno equation is a none-decreasing function of γ, for a normal Veno flow γ should be a value between $\frac{1}{2}$ and $\frac{4}{5}$. To determine an accurate expression for γ is still an ongoing work. We assumes W value is uniformly distributed. The validation on NS-2 simulation and wireless LAN experiments shows that, our assumption can approximate the experiment results quite well. Moreover, we assume that the refined AI phase in Veno has negligible improvement in TCP throughput as compared to the refined MD phase. This has been illustrated at the beginning of this section.

Some other simplifying assumptions have also been made during our derivation: 1) our model does not include the slow start and fast recovery algorithm of Veno. We believe the impacts of both phases are small, and can be ignored in the model; 2) we assume that packet losses within a round are correlated. This is caused by the drop-tail policy of intermediate routers: if one packet is dropped by the full buffer, it is likely the following packets in the round will be dropped too; 3) we assume losses in one round are independent of losses in another round, because packet in different rounds are separated by one RTT and more, and thus they are likely to encounter different buffer states; 4) we assume the round trip time is independent of the window size. Note that, all the above assumptions are the same as those in [2].

IV. Validation of Veno Equation

The validation of Veno equation is divided into two parts: the simulation results on NS-2 and the experimental results on a wireless LAN.

A. Simulation Results

The topology of our experiment on NS-2 [14] is shown in Figure 2.

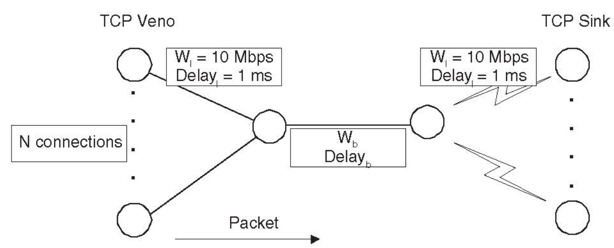

Fig. 2. Topology of NS-2 experiments.

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. Its bandwidth (W_b) and delay ($Delay_b$) will be changed during our experiments. Its packet buffer size is set to be 20. TCP packets are transferred from the wired network to the wireless network. Packet size is 1000Byte.

1) Single Flow: At first a single TCP Veno connection is established. The bandwidth of the bottleneck link is set to be 2Mbps and the delay is 80ms. Figure 3 shows the validation result under different packet loss rates from 0.0001 to 0.1. We also plot the minimum and maximum boundaries curves when $\gamma = \frac{1}{2}$ and $\gamma = \frac{4}{5}$ respectively. Note that during experiments, we collect the dynamic values of RTT and $(RTT - BaseRTT)$ at every TD loss, to calculate the Veno parameter γ_i using equation (23). Finally we use the average of γ_i to calculate the throughput using equation (19). As shown in Figure 3, most experiment results fall between the maximum and minimum boundaries, and they can be accurately predicted by our Veno equation. At the same time, we study whether the

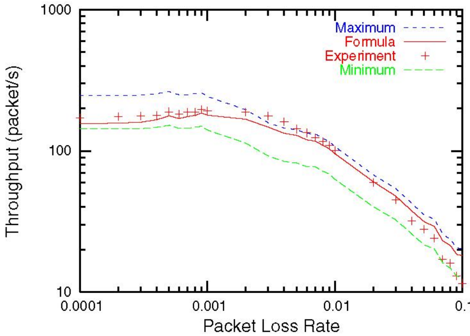

Fig. 3. Single flow, $W_b = 2Mbps$, $Delay_b = 80ms$.

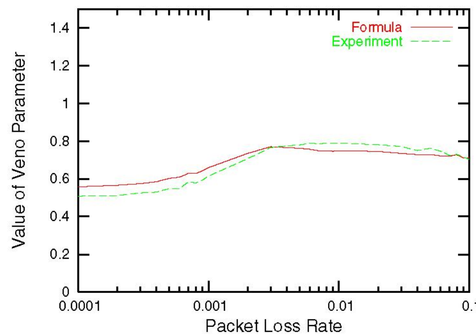

Fig. 4. Single flow, comparison of γ and γ_e.

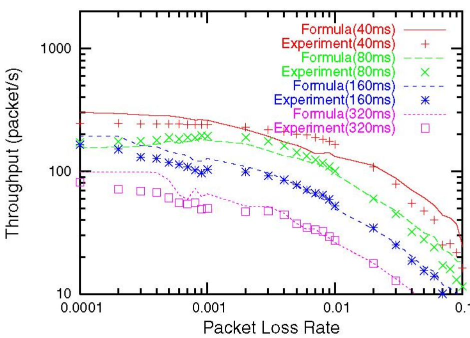

Fig. 5. Single flow, validation under different bottleneck link delays ($Delay_b$).

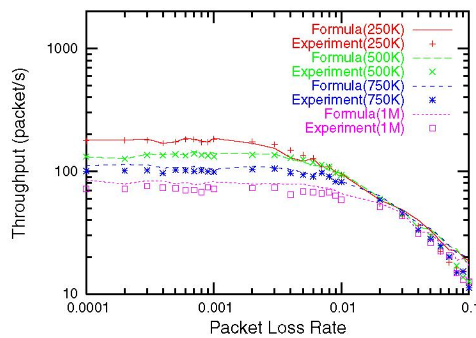

Fig. 6. Single flow, validation under different background traffic.

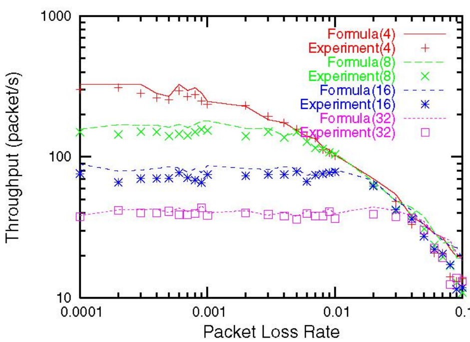

Fig. 7. Multiple flows, $W_b = 12Mbps$, $Delay_b = 80ms$.

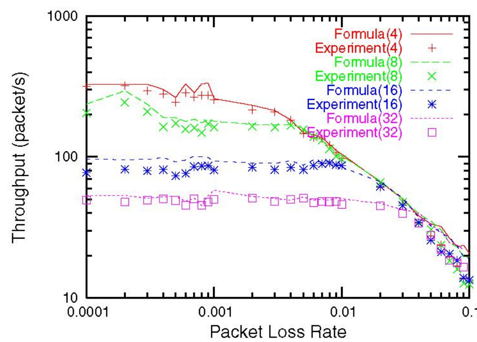

Fig. 8. Mix flows of Veno and Sack, $W_b = 12Mbps$, $Delay_b = 80ms$.

Veno parameter γ estimated by equation (23) can accurately predict the real one. The real Veno parameter γ_e can be calculated in the experiments as follows:

$$\gamma_e = \frac{N \times \frac{4}{5} + M \times \frac{1}{2}}{N + M} \tag{24}$$

where, N and M are the times of congestion window being cut down $\frac{1}{5}$ or $\frac{1}{2}$ respectively during transmission. Figure 4 shows these two values are close, regardless of different packet loss rates.

Figure 5 demonstrates the validation under different bottleneck link delays from 40ms to 320ms, and Figure 6 is the result under different bottleneck link background traffic. Noted that, to simulate the background traffic, we set up two UDP connections with pareto distribution [15] over the bottleneck link. The packet size of UDP is 512Byte. The burst time and idle time are both 100ms. The rate of each connection is set to be from 250Kbps to 1Mbps. As we see in these figures, the theoretical results and the simulation results match quite well, regardless of different loss rates.

2) Multiple Flows: To further verify the correction of Veno equation in complicated scenarios, we test multiple TCP connections (ranging from 4 connections to 32 connections). Firstly, we let all connections are TCP Veno flows, and thereafter, we replace half of them with TCP Sack flows. The bandwidth of the bottleneck link is changed to be 12Mbps and the delay is 80ms. The results are observed in Figure 7 and 8 respectively. Seeing these two figures, our equation continues to keep high match with the simulation results either for multiple connections or for mix connections.

B. Wireless LAN Experiments

Figure 9 depicts the environments of our wireless LAN experiments. In wireless side, the client is a Compaq laptop

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

with W200D wireless card. The wireless Access Point is Cisco Linksys WRT54G. They are separately placed at two corners and the distance between them is about 8m. The bandwidth of the wireless network is 11Mbps and the frequency is 2.4GHz. In the wired side, the server is a PC with FreeBSD 4.3, where TCP Veno is implemented. The gateway between wireless and wired networks is installed with FreeBSD 4.9.

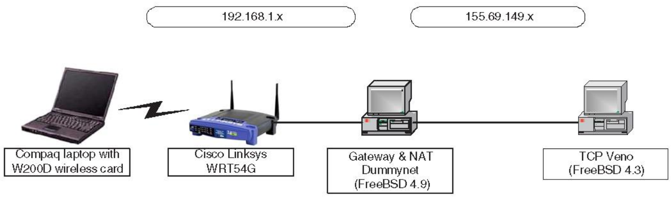

Fig. 9. Topology of the wireless LAN.

To verify our Veno equation, we use the method similar to that in [2]. One object Veno flow runs from the server to the client for 1hr. The other four Veno flows run in the same direction for 2800s, 2000s, 1200s, and 400s sequentially, which are regarded as the background traffic. The 1hr flow trace is divided into 36 consecutive 100s intervals, and we count the number of packets transmitted, RTT, $BaseRTT$, and the packet loss rate p in every interval. Note that p is given as the total number of loss indications (TD and TO) divided by the number of packets transmitted. Then, we put the average values of RTT and $BaseRTT$ into the Veno equation to calculate the theoretical result. Note that the packet size transmitted is 512Byte.

Two scenarios are studied here: 1) with Dummynet [16] configured at the gateway; 2) without Dummynet configured at the gateway. When Dummynet is set, it shapes the wired link bandwidth to 2Mbps, the delay to 80ms and the buffer to 20 packets. Because the general trends of the results in each experiment are similar, we only give one sample in each case for the length limitation. In each figure, three curves represent the maximum boundary (when $\gamma = \frac{4}{5}$), the minimum boundary (when $\gamma = \frac{1}{2}$), and the actual results of Veno equation respectively, and the points represent the experiment results. As shown in Figure 10, the Veno equation can model TCP Veno's throughputs fairly well in each case.

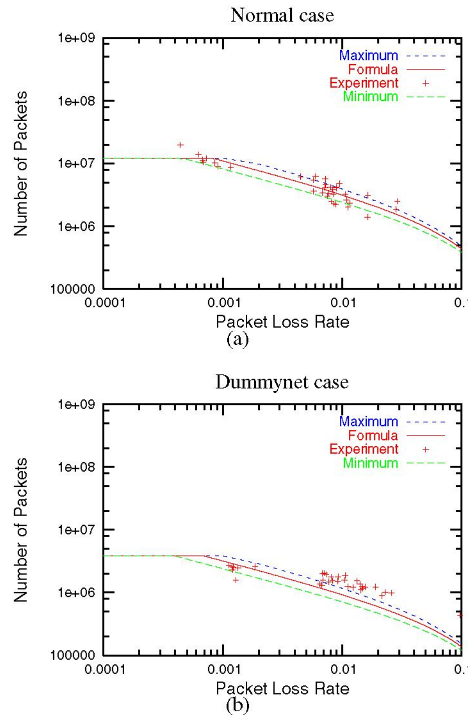

Fig. 10. Experiment results on the wireless LAN in (a) normal case and (b) Dummynet case.

V. Conclusion and Future Work

In this paper, a simple throughput equation for TCP Veno is derived. We introduce a Veno parameter γ into previous work on Reno [2]. Such parameter accurately characterizes the refined MD phase of Veno. Our extensive experiments on both simulation and wireless LAN have validated the correctness of the equation.

In the future work, further efforts are needed to refine the simple assumption about the uniformly distributed W in the deduction, and a more accurate expression of the Veno parameter γ will be studied. Furthermore, we will apply such derived closed form equation of TCP Veno into TFRC protocol.

References

[1] Van Jacobson. "Congestion Avoidance and Control." SIGCOMM 88, ACM, Aug 1988.
[2] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. "Modeling TCP Throughput: A Simple Model and its Empirical Validation." SIGCOMM 98, ACM, 1998.
[3] Sally Floyd, Mark Handley, Jitendra Padhye, and Joerg Widmer. "Equation-Based Congestion Control for Unicast Applications." SIGCOMM 2000, ACM, August 2000.
[4] Cheng Peng Fu, Soung 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." SIGCOMM 94, ACM, 1994.
[6] C. L. Zhang, C.P.Fu, M.T.Yap, C.H.Foh et al. "Dynamics Comparison of TCP Reno and Veno." IEEE Globecom, 2004.
[7] C. P. Fu, W. Lu, B. S. Lee. "TCP Veno Revisited." IEEE Globecom, 2003.
[8] Q. X. Pan, S. C. Liew, C. P. Fu, W. Wang. "Performance Study of TCP Veno over wireless LAN and RED router." IEEE Globecom, 2003.
[9] Z. X. Zou, C. P. Fu et al. "A Modification for Improving TCP Veno Performance with Bursty Congestion." IEEE Globecom, 2005.
[10] Z. X. Zou, B. S. Lee, C. P. Fu. "Packet Loss and Congestion State in TCP Veno." IEEE ICON, 2004.
[11] C. B. Fu, J. L. Wang , C. P. Fu, K. Zhang. "Performance Study of TCP Veno in Wireless/Asymmetric Links." International conference on cyberworlds, Japan 2004.
[12] Ling-chi Chung, C. P. Fu, Song C. Liew. "Improvements Achieved by SAK Employing TCP Veno Equilibrium-Oriented Mechanism over Lossy Networks." IEEE EUROCOM, 2001.
[13] Y. Liu, C. P. Fu et al. "An Enhancement of Multicast Congestion Control over Hybrid Wired/Wireless Networks." IEEE WCNC , 2004.
[14] S. Bajij, L. Breslau, D. Estrin, K. Fall et al. "Improving simulation for network research." Univ. Southern California, Los Angeles, Tech. Rep. 99-702b, 1999
[15] http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf
[16] http://info.iet.unipi.it/~luigi/ip_dummynet

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.