

# END TO END LOSS DIFFERENTIATION SCHEME FOR IDENTIFICATION OF PACKET LOSSES IN TCP OVER WIRED & WIRELESS NETWORKS

Sandeep Goswami<sup>1</sup>, Mridul<sup>2</sup>

<sup>1, 2</sup> Shobhit University, Meerut (India)

## ABSTRACT

*We propose a robust end-to-end loss differentiation scheme to identify the packet losses due to congestion for TCP connections over wired/wireless networks. We use the measured RTT values in determining whether the cause of packet loss is due to congestion over wired path or regular bit errors over wireless paths. The classification should be as accurate as possible to achieve high throughput and maximum fairness for the TCP connections sharing the wired/wireless paths. The accuracies of previous schemes in the literature depends on varying network parameters such as RTT, buffer size, amount of cross traffic, wireless loss rate and congestion loss rate. The proposed scheme is robust in that the accuracy remains rather stable under varying network parameters. The basic idea behind our scheme is to set the threshold for the classification to be a function of the minimum RTT and the current sample RTT, so that it may automatically adapt itself to current congestion level. When the congestion level of the path is estimated to be low, the threshold for a packet loss to be classified as a congestion loss is increased.*

*This avoids unnecessary halving of the congestion window on packet loss due to regular bit errors over wireless path and hence improves the TCP throughput. When the congestion level of the path is estimated to be high, the threshold for a packet loss to be classified as a congestion loss not to miss any congestion loss is decreased and hence improves the TCP fairness.*

**Keywords:** TCP, Wired/Wireless, Packet Loss

## I. INTRODUCTION

TCP congestion control runs under the basic assumption that any packet loss is the indication of congestion. However, the assumption does not hold when the TCP flow path includes wireless part. In such a case the packet loss may not come from congestion but from regular bit errors over wireless path. TCP throughput may be unnecessarily degraded due to the packet loss from bit errors over wireless part even though there is little congestion.

Research on improving performance of TCP over hybrid wired/wireless paths has focused on differentiating packet losses using information readily available to TCP: congestion window size, inter-arrival time between ACK packets, and changes in round-trip time(RTT) [1][11][4][3]. However, correct classification based on these metrics has been found to be difficult [11].

The reason is that nature of losses is weakly correlated with the observable metrics(RTT, congestion window size, inter-arrival of ACKs, and Jitter) [12]. In [2], the timestamp option in TCP is used to measure inter-arrival Jitter between data packets. From our simulations, this scheme tends to regard most of packet loss due to

wireless channel errors. We propose a new end-to-end scheme to precisely infer the nature of packet losses over wired/wireless networks. (From now on, we call packet losses due to congestion congestion losses and the losses due to wireless channel errors wireless losses.) Instead of fixed threshold in estimating the cause of individual packet loss, our scheme employs a moving threshold for relative change of RTT against the minimum RTT. In particular, we use the ratio of a difference between a sample RTT and a mean of RTT over a variance of RTT. We classify a packet loss as congestion loss if this value exceeds the threshold, or classify it as wireless loss, otherwise.

The threshold is defined as a function of minimum and sample RTT, which decreases as congestion level increases. The moving threshold is lowered for the differentiator to be more sensitive to congestion loss when the network is congested while it is increased when the network is unloaded. Our scheme is shown to be more stable in identifying the congestion losses under varying network conditions than previous schemes [1][11][4][3][2]. The accuracy of correctly detecting wireless loss on unloaded path is similar to previous schemes as far as congestion loss rate is kept low.

The rest of this thesis is organized as follows. We describe related work in Section 2, and in Section 3 we propose a new scheme of differentiating the nature of packet loss and its evaluation is described in Section 4. In Section 5 we conclude.

## II. RELATED WORK

Known schemes to improve the TCP throughput over wired and wireless paths can be divided into three categories: First, network-based schemes locate an agent at the access point/base station on the TCP path to locally recover the wireless loss at transport or link layer. Network-based schemes at transport layer do not maintain the end-to-end semantics of TCP and may require states to be maintained and packets to be buffered at the base station [5] [6].

At link layer, they can be purely local or aware of the semantics of the TCP protocol [7]. In an explicit loss notification approach, the receivers/network routers mark the acknowledgements with appropriate notification of distinguishing the channel errors from congestion losses [13]. Then the senders respond to the notification. The explicit loss notification approaches require modifications to network infrastructure, the receiver, and the senders. Finally, end-to-end schemes modify the TCP congestion control algorithm to distinguish the losses due to congestion in the network from other random losses over wireless paths. They can be deployed easily via simple modification to the TCP congestion control at sender or receiver side [1] [2] [4] [3] [11].

Our scheme also falls into end-to-end schemes. The Vegas predictor [11][1] estimates the outstanding packets over the network from the difference between the expected and the actual bandwidth. Biaz and Vaidya [11] study the accuracy of the Vegas predictor and show the accuracy is dependent on network parameters. TCP Veno [1] employs the vegas predictor to differentiate the cause of packet losses.

$$N = \left( \frac{cwnd}{BaseRTT} - \frac{cwnd}{RTT} \right) \times BaseRTT \quad (1)$$

where cwnd is the current TCP window size, RTT is the smoothed round-trip time measures, and BaseRTT is the minimum of measured round-trip times. In this scheme, the network is considered in congestion if N exceeds threshold  $\beta$ . In NS-2 simulation, we use the setting for  $\beta$  equal to 3.

The spike [3] is a scheme based on RTT measurement. In this work, we use RTT instead of Relative One-way Trip Time (ROTT). It keeps track of the minimum and maximum RTT values and computes thresholds Bspikestart and Bspikeend. The two thresholds are determined as some values between the minimum and maximum RTT values where Bspikestart > Bspikeend. TCP enters the spike state if RTT exceeds Bspikestart, but it exits the spike state if RTT drops below Bspikeend. In the spike state, all of the packet losses are regarded as due to congestion. Recently, TCP Westwood+ with bulk repeat (TCPW-BR) [15] employs this scheme to differentiate the cause of packet losses. The spike scheme has two thresholds Bspikestart and Bendspike defined as:

$$\begin{aligned} B_{spikestart} &= BaseRTT + \alpha \cdot (MaxRTT - BaseRTT) \\ B_{spikeend} &= BaseRTT + \beta \cdot (MaxRTT - BaseRTT) \end{aligned} \quad (2)$$

where MaxRTT is the maximum of measured round-trip times. The parameters  $\alpha$  and  $\beta$  affect the sensitivity and aggressiveness of Spike. In TCPW-BR, the setting for  $\alpha$  and  $\beta$  in [15] is 0.4 and 0.05, respectively. In our simulations, this setting regards the most of packet losses as due to congestion. i.e. The accuracy of congestion classification is almost 100%, but of wireless classification is about 0%. Therefore we choose the setting in [3].  $\alpha$  and  $\beta$  are set to / and / respectively.

NCPLD [4] is to find the knee point in throughput-load curve where the throughput decreases negligibly as the load increases. The TCP sender estimates the total number of segments in flight over the path to the receiver. This scheme tends to correctly detect the congestion loss, but tends to regard most of packet losses as congestion losses.

The TCP sender estimates the total number of segments in flight over the path to the receiver as follows:

$$TotalPipeSize = \frac{1}{2} \cdot T_k \cdot \frac{cwnd_k - cwnd_{k-1} - 1}{T_k - T_{k-1} - 1} \quad (3)$$

where  $T_k$  is the round-trip time measured on receipt of the k-th ACK. The NCPLD scheme requires also an estimate of the bandwidth-delay product. To this aim, they used an exponentially weighted moving average(EWMA) filtering the ACK reception rate to get an estimate of the transmission rate BTX. Both estimates yield the current value of T at the knee-point as presented by Eq. (3). The NCPLD scheme regards packet loss due to congestion if the current T is greater than  $T_{kp}$ .

$$T_{kp} = BaseRTT + \frac{1}{2} \cdot T \cdot \frac{BTX \cdot BaseRTT}{TotalPipeSize} \quad (4)$$

These loss differentiators estimate the change of the queuing delay to detect the packet loss due to congestion which occurs after build-up of the network buffer. However, assuming that round-trip path and location of the bottleneck do not change, the accuracy depends on network conditions. For accurate loss differentiation, favorable values of network conditions are as follows: round-trip time small, router queue size large, and input bandwidth to the bottleneck small [11].

In jitter-based TCP scheme (JTCP) [2], a TCP sender measures jitter ratio from the packet-by-packet inter-arrival time at the receiver side and the sending interval for one RTT. It requires the timestamp option in TCP header. The jitter ratio(Jr) can be written as following,

$$J_r = \frac{(t_r(i) - t_r(i - cwnd)) - (t_s(i) - t_s(i - cwnd))}{t_r(i) - t_r(i - cwnd)} \dots\dots\dots(5)$$

where  $t_r$  is the arrival time for  $i$ -th packet at a receiver, and  $t_s(i)$  is the sending time for  $i$ -th packet at a sender. JTCP regards a packet loss as due to congestion if the jitter ratio is greater than the congestion window inverse( $k/cwnd$ ). A threshold constant  $k$  is recommended to set to 1 in [2]. This scheme improves the TCP throughput by improving the accuracy of correctly detecting the wireless losses. These end-to-end schemes use the queue build-up for detection of congestion losses in end-to-end manner.

However, their accuracies are unstable due to depending on network conditions [11] [12]. Some of them consider only the improvement in TCP throughput or the achievement of fairness, and then tend to regard most of packet losses as one type between congestion and wireless losses.

In this paper, a simple robust and adaptive scheme is proposed to distinguish the nature of packet loss for TCP using information readily available to TCP. The scheme's accuracy for congestion losses is shown to be stable without favorable network condition. The scheme can capture more wireless losses when the network is unloaded while it can capture more congestion losses when the network is congested. Furthermore, TCP connections using the scheme can improve the throughput over wired and wireless network, and can achieve the intraprotocol fairness and the TCP friendliness.

### III. THE PROPOSED SCHEME

We present a new robust scheme to infer the cause of each packet loss encountered whether it is due to congestion or not. The RTT ( $T$ ) measured immediately before the current packet loss is used in Eq. (6) as an indicator.  $\bar{T}$  and  $T_{dev}$  are an exponentially weighted moving average of RTT( $T$ ) and the deviation, respectively.

They are updated by:

$$\bar{T} = \frac{7}{8}\bar{T} + \frac{1}{8}T \quad \text{and} \quad T_{dev} = \frac{3}{4}T_{dev} + \frac{1}{4}|T - \bar{T}|$$

The current packet loss is determined to be a congestion loss if Eq. (6) is satisfied.

$$T > \bar{T} + T_{dev} \cdot \left( 2 \cdot \left( \frac{T_p}{\bar{T}} \right)^k - 1 \right) \quad (6)$$

This differentiation depends on four parameters:  $T$ ,  $T_{dev}$ ,  $k$ , and  $T_p$ . Statistical values  $T$  and  $T_{dev}$  can indicate the current network condition. A constant  $k$  can be chosen for the proposed differentiation to be accurate.  $T_p$  can be a criteria to infer the cause of packet loss. We will study on effect of constant  $k$  and various network conditions ( $T$  and  $T_{dev}$ ) on the decision rule. We then study the impact of an overestimated  $T_p$  on the decision rule. Finally, we find effective  $k$  for the differentiation to be accurate via simulations.

### IV. THROUGHPUT MODEL

We derive the TCP throughput of the proposed scheme following arguments developed by Kelly [18]. For the simplicity, we do not model the behavior after a timeout.

The cwnd is updated upon ACK reception. Each time an ACK is received back by the sender the cwnd is increased by  $1/\text{cwnd}$ . On the receipt of TDACKs, the proposed scheme involves to infer the cause of the packet loss. The cwnd is reduced by half if the proposed scheme regards the packet loss as due to congestion. Otherwise, the cwnd will be kept. Let  $\Pr(c|l)$  denote the probability of packet losses regarded as congestion losses. It results from the aggregate sum of the probability to correctly detect the congestion losses represented by  $P_c \cdot A_c / P$  and the probability to wrongly infer the wireless loss into the congestion loss represented by  $P_w / P[1 - A_w]$

Therefore change in cwnd is equal to

$$\frac{1}{2} \cdot P_r(c|l) \cdot \text{cwnd}$$

$$P_r(c|l) = \frac{1}{P} [P_c \cdot A_c + p_w \cdot (1 - A_w)]$$

Now [6]

Since the total packet loss rate is  $p$ , it follows that the expected change of cwnd per update step is:

$$E[\Delta \text{cwnd}] = \frac{1-p}{\text{cwnd}} - \frac{1}{2} \cdot P_r(c|l) \cdot \text{cwnd} \cdot p \quad (7)$$

Since the time between update steps is about  $T/\text{cwnd}$ , recalling Eq. (9), the expected change in the rate  $x$  per unit time is approximately:

$$\frac{\delta x(t)}{\delta t} = \frac{1-p}{T^2} - \frac{1}{2} \cdot P_r(c|l) \cdot p \cdot x^2(t) \quad (8)$$

$$\frac{\delta x(t)}{x^2(t) - \frac{2(1-p)}{T^2 \cdot P_r(c|l) \cdot p}} = -\frac{1}{2} \cdot P_r(c|l) \cdot p \cdot \delta t \quad (9)$$

and by integrating each member the following solution can be obtained.

$$x(t) = \frac{x_1 \cdot (1 + C_0 \cdot e^{-\frac{1}{2} P_r(c|l) \cdot p \cdot t})}{1 - C_0 \cdot e^{-\frac{1}{2} P_r(c|l) \cdot p \cdot t}}$$

$$x_1 = \frac{1}{T} \sqrt{\frac{2(1-p)}{P_r(c|l) \cdot p}} \quad \text{Where } x_1 = \frac{1}{T} \sqrt{\frac{2(1-p)}{P_r(c|l) \cdot p}} \text{ is the root of equation } x^2(t) - \frac{2(1-p)}{T^2 \cdot P_r(c|l) \cdot p} = 0.$$

The steady state throughput of the proposed scheme is then,

$$x_{\text{ssd}} = \lim_{t \rightarrow \infty} x(t) = \frac{1}{T} \sqrt{\frac{2(1-p)}{P_r(c|l) \cdot p}} \\ = \frac{1}{T} \sqrt{\frac{2(1-p)}{P_c \cdot A_c + p_w \cdot (1 - A_w)}}$$

Therefore, the improvement in TCP throughput by the loss differentiation scheme depends on both  $A_c$  and  $A_w$ . When the path is unloaded,  $A_w$  is important to achieve higher throughput over wireless network. When the path is loaded,  $A_c$  is important to achieve fairness between connections or existing TCP variants.

The proposed scheme can adjust the threshold into the current congestion level: The threshold is increased on the loaded path and is lowered on the unloaded path, automatically. We will show that this property helps the proposed scheme to achieve both the improvement in throughput and the fairness.

## **V. PERFORMANCE EVALUATION AND RESULT**

In this chapter, we evaluate the accuracy and the throughput performance of the proposed and other LDAs on TCP-Sack based on simulations on the ns-2 [9]. The result of simulations is presented in four separate categories:

### **5.1 Accuracies $A_c$ and $A_w$**

We present the result of comparison with accuracies of existing LDAs such as Spike, Jitter-based scheme and Vegas predictor with various wireless packet loss rates coexisting with congestion losses.

### **5.2 Throughput Improvement**

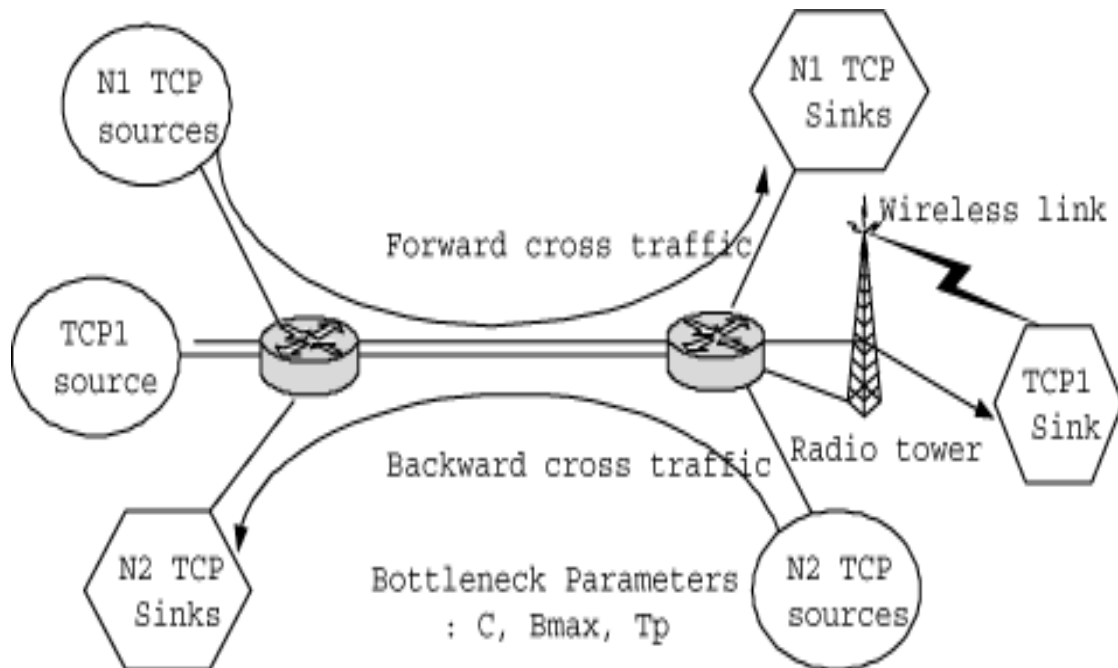
This category of simulations helps us understand the effect of LDAs on the throughput with uniform and correlated loss model. We present the result of comparison with accuracies of existing LDAs such as Spike, Jitter-based scheme and Vegas predictor with various wireless packet loss rates coexisting with congestion losses.

### **5.3 Interoperability**

This part includes 4 types of evaluations of joining the network with persistent congestion, sudden changes in available bandwidth, intraprotocol fairness of injecting multiple LDA connections into the link, and the possibility if LDA connections can coexist with TCP-SACK. We present the result of comparison with accuracies of existing LDAs such as Spike, Jitter-based scheme and Vegas predictor with various wireless packet loss rates coexisting with congestion losses.

### **5.4 Experimental Setup**

We evaluate the performance of the proposed scheme via ns-2 (Ver 2.27) [9] simulation. The network model is shown in Figure 4-1. The bandwidth  $C$  is set to 2 Mbps. The size of the buffer  $B_{max}$  is set to 50 packets. The one way delay is set to 70ms, which is equal to an half of  $T_p$ . We set the packet size equal to 1000 bytes. To generate network traffic, several sets of TCP sources/sinks are used. Among the TCP source/sink pairs, the TCP1 connection is treated as an observable TCP source/sink pair, and all the others are treated as background TCP source/sink pairs used to create the forward( $N1$ ) and backward( $N2$ ) cross traffic. The TCP1 connection goes through a wired path terminated with a last hop wireless link.



**Fig.1: Mixed Wired/Wireless Scenario**

The wireless last hop models a mobile user accessing the Internet via a radio link. RTTs of the N1 TCP forward cross traffic connections and of the N2 TCP backward cross traffic connections are the same as the path of the TCP1 connection. The congestion packet loss rate  $p_c$  increases as N1 increases. We run simulations 30 times. The simulation time is 1000 seconds. In the each of the 30 runs, we estimate the average values of  $A_c$  and  $A_w$ .

### 5.5 Throughput Improvements

In this section, we investigate the effect of the accuracy of the loss differentiation scheme on the TCP throughput improvement as varying environmental variables. The throughput improvement indicates the ratio of the throughput of TCP with the loss differentiation over the throughput of TCP Sack. We write it as following,

$$I_{\text{imp}} = 100 \times \left( \frac{\text{Thgt}_{LDA}}{\text{Thgt}_{TCP}} - 1 \right)$$

### 5.6 Improvement Vs Pw, Tp, C, and Loss Model

We show the improvement the TCP throughput as varying  $p_w$ ,  $T_p$ ,  $C$  and loss model under no congestion. Figure 2(a) shows that the proposed scheme can achieve the highest improvement in TCP throughput for higher  $p_w$  in uniform model. Especially, for  $p_w=10\%$ , the improvement of proposed scheme is shown to be about 30% higher than other schemes.

Figure 2 (b) shows that the Spike can respond to bursty losses in correlated model. The proposed scheme still achieves the second best improvement overall. Figure 2 (c) and 2 (d) show that the improvement of proposed scheme can increase with  $T_p$  and  $C$ . In particular,

for  $T_p > 0.4s$  or  $C > 4Mbps$ , the proposed scheme achieves the best improvements. Improvements of Spike and NCPLD are lowered from  $T_p = 0.5s$  as shown in Figure 2(c). In Figure 2 (d), other schemes except the Jitter ratio fail to increase the improvement for  $C > 6Mbps$ .



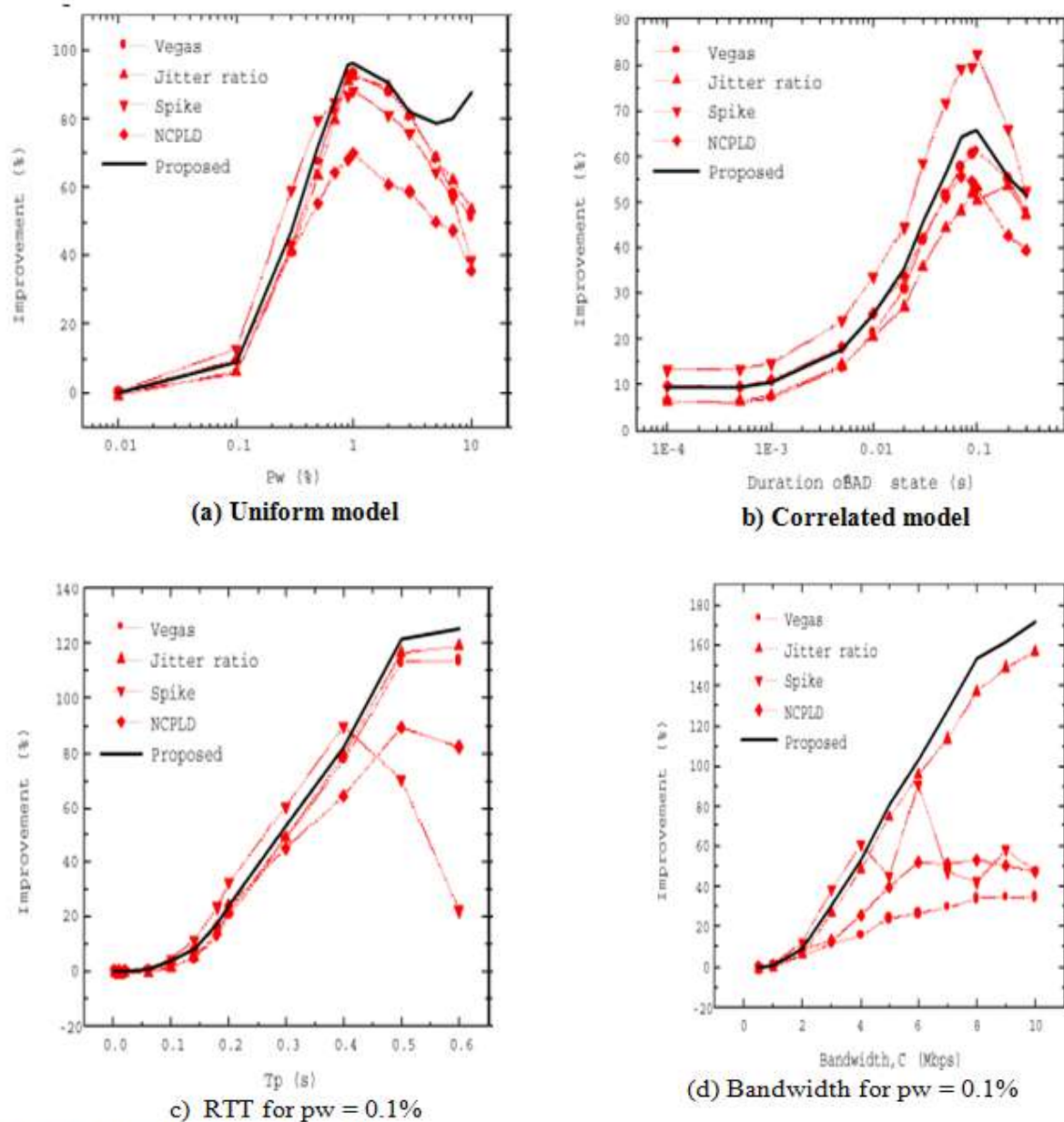


Fig. 2: Improvement in TCP throughput as varying pw, Tp and C

## VI. CONCLUSIONS

We propose a simple loss differentiation scheme to improve the accuracy through precise inference of the cause of packet loss. The proposed scheme is an end-to-end scheme which is easy to implement because it uses information readily available to TCP. We evaluated our differentiation scheme under various network conditions, including different network configurations, workload, error rates of the wireless medium, and different wireless error models. Accuracies of loss differentiation of other schemes proposed previously depend on network parameters.

They favor long length of buffer, short propagation delay, low wireless loss rate and small number of connections on the bottleneck. However, the moving threshold enables proposed scheme to be accurate in correctly detecting congestion loss without favorable parameters. We found that our differentiation scheme is more stable under varying network conditions than vegas predictor, jitter ratio, NCPLD and spike scheme



## REFERENCES

- [1] C. P. Fu and S. C . Liew, "TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks," IEEE Journal on Selected Areas in Communications, vol. 21, pp216-228, Feb. 2003.
- [2] Eric H.K. Wu and Mei-Zhen Chen, "JTCP: Jitter-Based TCP for Heterogeneous Wireless Networks", IEEE Journal on Selected Areas in Communications, Vol. 22, No. 4, pp757-766, May 2004.
- [3] S. Cen, P. C. Cosman and G. M. Voelker, "End-to-end Differentiation of Congestion and Wireless Losses", IEEE/ACM Transactions on Networking, Vol. 11, No. 5, Oct. 2003, pp703-717.
- [4] N. Samaraweera, "Non-congestion packet loss detection for TCP error recovery using wireless links", In IEEE Proceedings of Communications, Aug. 1999.
- [5] Kevin Brown and Suresh Singh, "M-TCP: TCP for Mobile Cellular Networks", ACM SIGCOMM 97, Cannes, France, July. 1997.
- [6] Ajay Bakre and B. R. Bandrinath, "I-TCP: Indirect TCP for Mobile Hosts", Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS '95), pp 136~143, Vancouver, May, 1995.
- [7] Hari Balakrishnan, Srinivasan Seshan, and Randy H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", In ACM Wireless Networks Journal, 1(4), Dec. 1995.
- [8] Sumitha Bhandarkar, Nauzad Erach Sadry, A.L. Narasimha Reddy, and Nitin H. Vaidya, "TCP-DCR: A Novel Protocol for Tolerating Wireless Channel Errors", IEEE Transactions on Mobile Computing, Vol. 4, No. 5, pp517-529, Sep. 2005.
- [9] NS-2 network simulator (ver 2). LBL, <http://www-mash.cs.berkeley.edu/ns>
- [10] R. Jain, D.M.Chiu and W.Hawe, "A Quantitative Measure of Fairness and Discrimination of Resource Allocation in Shared Systems", Technical Report, Digital Equipment Coporation, DEC-TR-301, 1984.
- [11] Saad Biaz and Nitin H. Vaidya, "Distinguishing Congestion Losses from Wireless Transmission Losses : A Negative Result", In IEEE 7th Int'l Conf. on Computer Communications and Networks, Lafayette, LA, pp390-397, October 1998.
- [12] Saad Biaz and Nitin H. Vaidya, "Is the Round-trip Time Correlated with the Number of Packets in Flight?", Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, pp273-278, 2003.
- [13] Saad Biaz and Nitin H. Vaidya, "'De-randomizing Congestion Losses' to Improve TCP Performance over Wired-Wireless Networks", ACM/IEEE Transactions on Networking, Vol. 13, No. 3, June 2005, pp596-608.
- [14] C. Barakat and E. Altman, "Bandwidth Tradeoff between TCP and Link- level FEC", Computer Networks, 39, 2002, pp. 133-150.
- [15] Guang Yang, Ren Wang, Mario Gerla and M. Y. Sanadidi "TCPW Bulk Repeat", Computer Communications(Elsvier), Vol. 28, Issue 5, Mar. 2005, pp507-518
- [16] D. Aguayo, J. Bicket, S. Biswas, G. Judd and R. Morris, "Link-level Measurement from an 802.11b Mesh Network", ACM SIGCOMM 2004, Aug. 2004, Portland, Oregon, USA
- [17] A. Gurtov and S. Floyd, "Modeling Wireless Links for Transport Protocols", ACM SIGCOMM Computer Communications Review, Vol. 34, Issue 2, April 2004, pp85-96
- [18] Frank Kelly, "Mathematical Modeling of the Internet", Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, 1999, pp105-116.