Improving TCP Performance over Wireless Networks using Loss Predictors

Fabio Martignon
Dipartimento Elettronica e Informazione
Politecnico di Milano
P.zza L. Da Vinci 32, 20133 Milano
Email: martignon@elet.polimi.it

***Abstract*— The use of Loss Predictors (LP) in TCP congestion control has been recently proposed to improve performance over heterogeneous networks including wired and wireless links. These algorithms provide TCP with an estimate of the cause of packet losses. In this paper we propose a modification to the Vegas predictor, studying its accuracy under several realistic scenarios. We show that this predictor achieves high accuracy throughout all the network scenarios. We propose to use the modified Vegas predictor in the TCP congestion control scheme to react to packet losses avoiding the overly conservative rate reductions of TCP Reno in the presence of random losses. We study the performance of this new TCP version (TCP NewReno-LP), showing how it achieves higher performance over wireless networks while guaranteeing a good fairness level with existing TCP versions over wired links. We also provide an upper bound on the performance of TCP using Loss Predictors by implementing an ideal scheme having perfect knowledge of the cause of packet losses, showing how our TCP NewReno-LP approaches that bound.**

I. Introduction

The Transmission Control Protocol (TCP) has proved efficient in classical wired networks, showing an ability to adapt to modern, high-speed networks and new scenarios for which it was not originally designed. However, the extraordinary success of modern wireless access networks, such as cellular networks and wireless local area networks, poses new challenges to the TCP congestion control scheme.

The existing versions of TCP, like Reno or NewReno, experience heavy throughput degradation over channels with high error rate, such as wireless channels. The main reason for this poor performance is that the TCP congestion control mechanism cannot distinguish between packet losses occurring randomly in wireless channels and those due to network congestion. Therefore, the TCP congestion control mechanism reduces, even when not necessary, the transmission rate [1].

To solve such limitation and degradation the TCP congestion control scheme needs modifying, by the implementation of more sophisticated algorithms that perform an explicit estimation of the cause of packet losses. This estimate can be used to avoid unnecessary reductions of TCP transmission rate when losses are due to fading on the wireless channel, and not to network congestion.

In this paper we propose to use a Vegas predictor [2] in the TCP congestion control scheme to react to packet losses avoiding the overly conservative rate reductions of TCP NewReno in the presence of random losses. We study the performance of this new scheme, called TCP NewReno-LP, showing how it achieves higher performance over wireless networks with both long-lived and short-lived TCP connections, while guaranteeing a good fairness level with existing TCP versions over wired links. TCP NewReno-LP can be implemented by modifying the sender-side only of a TCP connection, and this implies the possibility of its immediate deployment all over the Internet.

We also provide an upper bound on the performance of TCPs using Loss Predictors by implementing an ideal scheme that always has a perfect knowledge of the cause of packet losses, showing how TCP NewReno-LP approaches reasonably that bound.

The paper is structured as follows: Section II presents TCP NewReno-LP. Section III analyzes the accuracy of TCP NewReno-LP in estimating the cause of packet losses under several realistic network scenarios. Section IV measures the performance of TCP NewReno-LP in therms of achieved goodput and fairness, and its performance is compared to existing TCP versions over heterogeneous networks with both wired and wireless links affected by independent and correlated packet losses. Finally, Section V concludes the paper.

II. TCP NewReno Enhanced with Vegas Loss Predictor

The Vegas loss predictor [2] decides whether the network is congested or uncongested based on rate estimations. This predictor estimates the cause of packet losses based on V_P, calculated as

$$V_P = (Expected\_Rate - Actual\_Rate) \cdot RTT_{min}$$

where the *Expected* and the *Actual* flow rates are defined by $cwnd/RTT_{min}$ and $cwnd/RTT$, respectively. $Cwnd$ is the congestion window and RTT_{min} is the minimum Round Trip Time measured by the TCP source. When $V_P \geq \beta$ segments, the Vegas predictor assumes that network is congested; when $V_P \leq \alpha$ segments, eventual losses will be accounted as due to fading over the wireless link; finally, when $\alpha < V_P < \beta$, the predictor assumes that the network state is the same as in the previous estimation.

We propose to use this predictor within the congestion control of a TCP source as follows: when the source detects a packet loss, i.e. when 3 duplicate acknowledgements are received or a retransmission timeout expires, the Vegas predictor estimates the cause of the packet loss.

If the loss is classified as due to congestion, the TCP source reacts exactly as a classical TCP NewReno source [3]. On the contrary, if the loss is classified as due to the wireless channel, the slow start threshold is first updated to the current $cwnd$ value. Then, if the packet loss was detected by the TCP source after a retransmission timeout expiration, the congestion window is reset to 1 segment; if the loss was detected after the receipt of 3 duplicate ACKs, the congestion window value is left unchanged.

III. Accuracy of the Vegas Loss Predictor embedded in TCP NewReno

Accuracy is a key feature for Loss Predictor (LP) schemes, as TCP error recovery schemes using LP estimates can react gently or aggressively depending on the LP estimate. Therefore, accuracy on congestion losses is required when the packet error rate is low, in order to achieve fairness with concurrent TCP flows. Moreover, accuracy on wireless losses is required when the packet error rate is considerable, in order to achieve higher goodput, defined as the bandwidth actually used for successfull transmission of useful data (payload).

We have implemented the TCP NewReno-LP scheme described in the previous Section in the Network Simulator, ns ver.2. The accuracy of the Vegas loss predictor used by TCP NewReno-LP has been tested inseveral network scenarios. The network topology considered is shown in Fig.1: a single TCP NewReno-LP source performs a long-lived FTP transfer. The wired link $S-N$, has capacity and propagation delay equal to 10 Mbit/s and 50 ms, and the wireless link $N-D$ has capacity equal to 10 Mbit/s and delay equal to 0.01 ms.

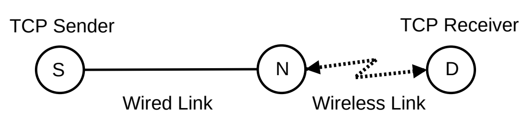

Fig. 1. Network Topology

We considered two different distributions of packet losses on the wireless link: independent and correlated losses. To model independent packet losses, the link drops packets according to a Poisson process, causing a packet error rate (PER) in the 10^{-5} to 10^{-1} range.

To account for the effects of multi-path fading typical of wireless environments, we also considered links affected by correlated errors. From the existing literature [4], we modeled the wireless link state (*Good* or *Bad*) with a two-state Markov chain. The average durations of the *Good* and *Bad* states are equal to 1 and 0.05 seconds, respectively. In the *Good* state no packet loss occurs, while we varied the packet error rate in the *Bad* state, to take into account various levels of fading.

We assume, as in the rest of the paper, that the Maximum Segment Size (MSS) of the TCP source is equal to 1500 bytes, and that all the queues can store a number of packets equal to the bandwidth-delay product of the connection. The TCP receiver always implements the Delayed ACKs algorithm, as recommended in [5].

We considered two different scenarios: in the first one, no cross traffic is transmitted over the wired link $S-N$; in the second scenario, the TCP source shares the wired link with 30 UDP sources having the same priority as the TCP source. Each UDP source switches between ON and OFF periods, whose durations are both Pareto distributed with shape parameter equal to 1.5 and mean durations equal to 100 ms and 200 ms, respectively. During the ON period, each source transmits packets with 1500 byte size at constant bit rate equal to 0.5 Mbit/s, while in OFF period the UDP sources don't transmit any packet. Such cross-traffic configuration leaves to the TCP source an available bandwidth that varies randomly during the simulation, with an average equal to half the bottleneck capacity.

TCP sources detect *loss events* based on the reception of triple duplicate acknowledgements or retransmission timeout expirations. We define wireless loss a packet loss caused by the wireless noisy channel; a congestion loss is defined as a packet loss caused by network congestion.

The overall accuracy on packet loss classification of the Vegas predictor is defined as the ratio between the number of correct packet loss classifications and the total number of loss events.

We measured the accuracy of the Vegas predictor in the network scenarios described above. As we said in Section II, the Vegas predictor detects congestion and wireless losses based on two thresholds, α and β. We tested several values for the parameters α and β. We found the best performance for the accuracy of the Vegas predictor for $\alpha = 1$ and $\beta = 3$. Fig.2 shows the accuracy on packet loss classifications of the Vegas predictor with these parameters as a function of the packet error rate in the scenario with no cross traffic. Each accuracy value has been calculated over multiple file tranfers with a 97.5% confidence interval. The vertical lines reported in the Figures represent the confidence interval for each accuracy value.

Fig.3 shows the accuracy for the Vegas predictor in the scenario with cross traffic on the wired link. We observed that the Vegas predictor is very accurate in discriminating the cause of packet losses for the whole range of packet error rates we considered. Moreover, the Vegas predictor shows very low sensitivity to parameter settings, especially if compared to other predictors proposed in the literature [6], [7].

Finally, Fig.4 shows the accuracy of the predictors when transmission errors are correlated and modeled according to the Markov chain described above. The packet error rate in the Bad state varies from 0% to 80% to account for different levels of fading in the wireless link. The Vegas predictor

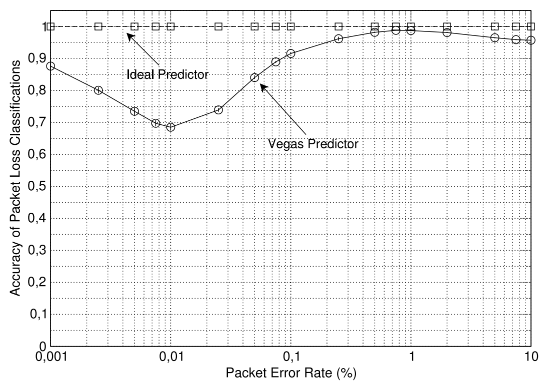

Fig. 2. Accuracy on classifications for the Vegas loss predictor in the scenario with no cross traffic on the wired link

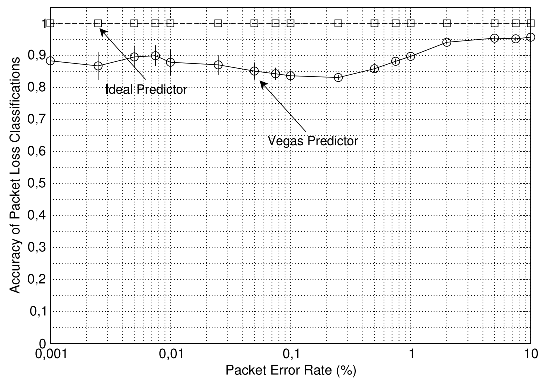

Fig. 3. Accuracy on classifications for the Vegas loss predictor in the scenario with cross traffic on the wired link

provides high accuracy and approaches an ideal estimator for the whole range of packet error rates.

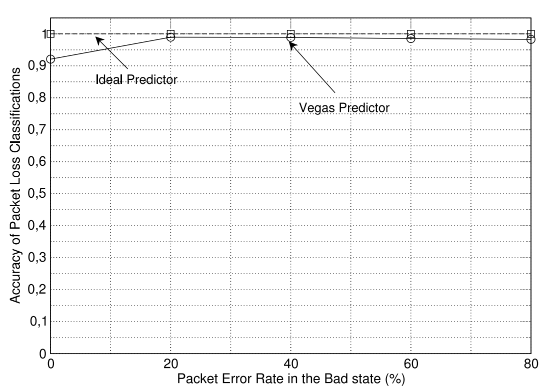

Fig. 4. Accuracy on classification of packet losses for the Vegas loss predictor as a function of the packet error rate in the Bad state in the scenario with no cross traffic

We have also extended our analysis to more complex network scenarios, with a varying number of TCP connections and multiple hops. For the sake of brevity we don't report these results. In all the scenarios we examined, the accuracy of the Vegas predictor has always been higher than 70%.

IV. TCP Performance over Wireless Links

TCP NewReno-LP performs an efficient estimation of the cause of packet losses. However, as this algorithm is mainly designed to achieve high throughput in the presence of links affected by random errors, we studied the performance of this algorithm over wireless links.

In order to measure TCP NewReno-LP performance, and compare it with other TCP versions, we considered several scenarios with two different types of connection: the long-lived TCP connections, typical of FTP file transfers, and short-lived connection, typical of HTTP connections. In the following we present, and discuss, the results obtained by simulation.

A. *Uncorrelated Losses*

In accordance with existing literature [8], we considered the topology shown in Fig.1. We analyzed three scenarios with different capacity of the wired and the wireless link: 2, 5 or 10 Mbit/s. The Round Trip Time is always equal to 100 ms and the queue can contain a number of packets equal to the bandwidth-delay product. Independent errors occur at random, causing a packet error rate in the 10^{-5} to 10^{-1} range. For each scenario we measured the steady state goodput obtained by TCP NewReno-LP and TCP NewReno. All goodput values presented in this Section were calculated over multiple file transfers with a 97.5% confidence interval. The results are shown in Figures 5,6 and 7, where the vertical lines represent, as in all the other Figures, the confidence interval for each goodput value.

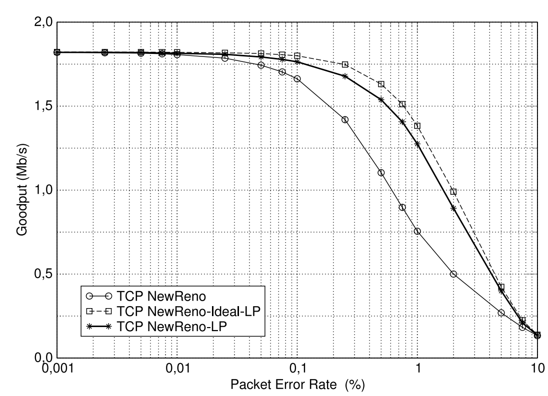

Fig. 5. Goodput achieved in the topology shown in Fig.1 with 2 Mb/s link capacity by various TCP versions as a function of the packet error rate

It can be seen that for all packet error rates and at all link speeds TCP NewReno-LP achieves higher goodput than TCP NewReno. This is due to the Vegas loss predictor that prevents, most of the time, confusion between real network congestion signals, due to queue overflow, and signals due to link errors.

To provide a comparison, Figures 5,6 and 7 also report the performance achieved by a TCP NewReno based on an ideal estimator that always knows the exact cause of packet losses (TCP NewReno-Ideal-LP). This scheme provides an upper bound on the performance achievable by every scheme based on loss predictor. Note that our scheme approaches this bound for all the considered scenarios.

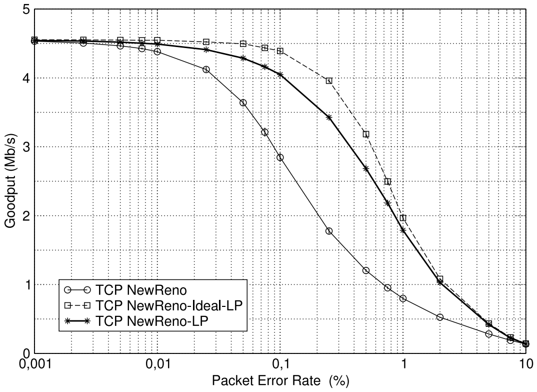

Fig. 6. Goodput achieved in the topology shown in Fig.1 with 5 Mb/s link capacity by various TCP versions as a function of the packet error rate

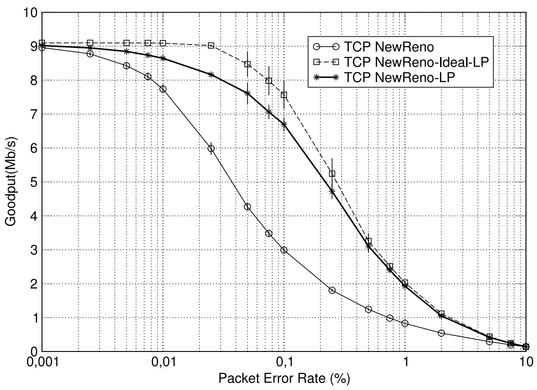

Fig. 7. Goodput achieved in the topology shown in Fig.1 with 10 Mb/s link capacity by various TCP versions as a function of the packet error rate

B. Correlated Losses

To account for the effects of multi-path fading typical of wireless environments, we also investigated the behavior of TCP NewReno-LP in the presence of links affected by correlated errors.

From the existing literature [4], we modeled the wireless link state with a two-state Markov chain as described in Section III. We considered two different scenarios with wireless link capacities equal to 2 and 5 Mb/s, a Round Trip Time equal to 100 ms and an average duration of $Good$ and Bad states equal to 1 and 0.05 seconds, respectively. In the $Good$ state no packet loss occurs, while, in the Bad state, the packet error rate varies from 0 to 80% to take into account various levels of fading. Figure 8 shows the steady-state goodput achieved by the TCP versions analyzed in this paper as a function of the packet error rate in the Bad state. TCP NewReno-LP achieves higher goodput than TCP NewReno and practically overlaps to the goodput upper bound achieved by the ideal scheme TCP NewReno-Ideal-LP.

A similar behavior was observed in Fig.9 where we reported the goodput achieved by the analyzed TCP versions in the topology shown in Fig.1 with a 5 Mb/s link capacity as a function of the packet error rate in the Bad state. Note that in this scenario the performance improvement of TCP NewReno-LP over TCP NewReno is higher than in the 2 Mb/s scenario, as wireless losses affect more heavily TCP NewReno goodput when the bandwidth-delay product of the connection is higher [1].

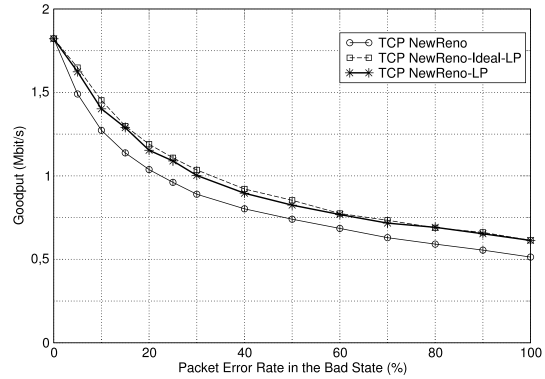

Fig. 8. Goodput achieved in the topology shown in Fig.1 with 2 Mb/s link capacity by various TCP versions as a function of the packet error rate in the Bad state

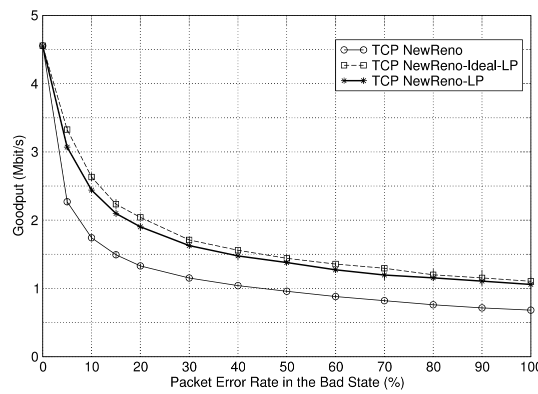

Fig. 9. Goodput achieved in the topology shown in Fig.1 with 5 Mb/s link capacity by various TCP versions as a function of the packet error rate in the Bad state

C. Impact of Varying Round Trip Time

Packet losses are not the only cause of TCP throughput degradation. Many studies proposed in the literature [9] have pointed out that TCP performance also degrades when the Round Trip Time of the connection increases. TCP NewReno-LP allows to alleviate this degradation and obtains better performance. Fig. 10 and 11 report the goodput achieved by TCP NewReno and TCP NewReno-LP sources transmitting over a 5 Mb/s and a 10 Mb/s link, respectively, as a function of the Round Trip Time of the connection. The link drops packets independently with a loss probability constantly equal to 0.5%.

TCP NewReno-LP achieves a high goodput gain over TCP NewReno, and this behavior is more evident when the Round Trip Time of the connection increases.

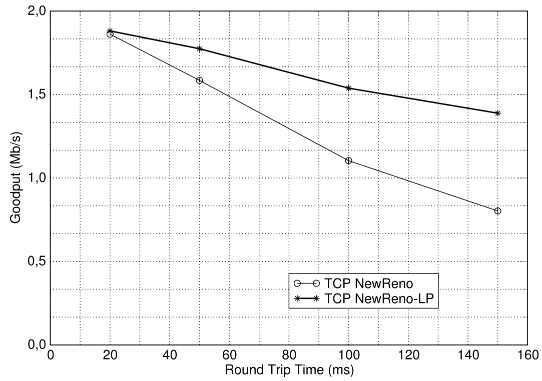

Fig. 10. Goodput achieved by TCP NewReno-LP and TCP NewReno over a 2 Mb/s link as a function of the Round Trip Time of the connection

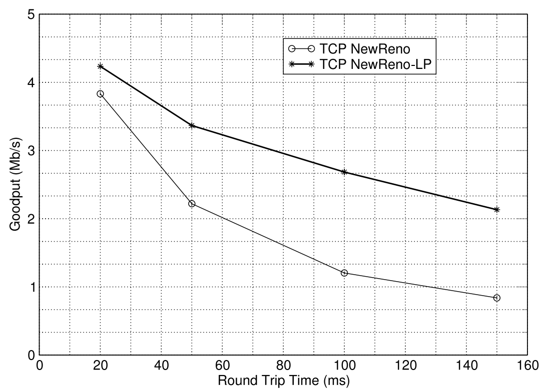

Fig. 11. Goodput achieved by TCP NewReno-LP and TCP NewReno over a 5 Mb/s link as a function of the Round Trip Time of the connection

D. Short-Lived TCP Connections

We also studied the performance of TCP NewReno-LP with short-lived TCP connections. We considered, in line with the literature [10], a typical HTTP connection involving the transfer of a 10 kbyte file over a 5 Mb/s link affected by a 5% random packet loss, with a 100 ms Round Trip Time. We simulated 500 transfers and measured the duration of each file transfer.

The average time to complete the transfer was 0.79 s for TCP NewReno-LP and 0.81 s for TCP NewReno. Hence, also for short file transfers, TCP NewReno-LP achieves a slight improvement over the current TCP version.

E. Fairness

So far we have shown that the TCP NewReno-LP scheme can actually perform an accurate estimate of the cause of packet losses, and that achieves higher goodput than existing TCP versions over wireless links with both uncorrelated and correlated losses. This section shows how the proposed scheme is able to fairly share network resources in mixed scenarios where the sources use different TCPs.

To this purpose, we considered a mixed scenario where 5 TCP connections using either TCP NewReno-LP or TCP NewReno share a 5 Mb/s link. By simulation we measured, for each connection, the goodput. The average goodputs of TCP NewReno-LP and TCP NewReno connections are shown in Figure 12.

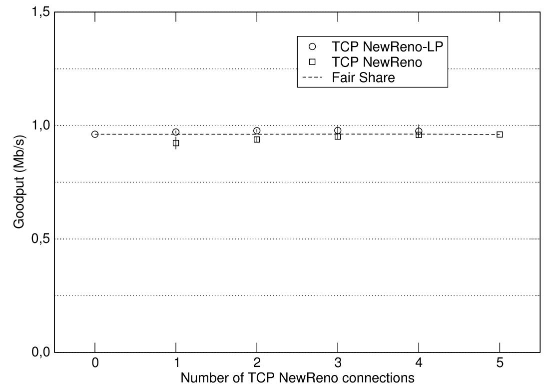

Fig. 12. TCP NewReno-LP fairness towards TCP NewReno over a 5 Mb/s link

The goodput achieved by both algorithms is very close to the fair share for the full range of sources.

The same experiment was performed with a 10 Mb/s link, and the results shown in Figure 13 confirm that TCP NewReno-LP achieves the same level of fairness of TCP NewReno.

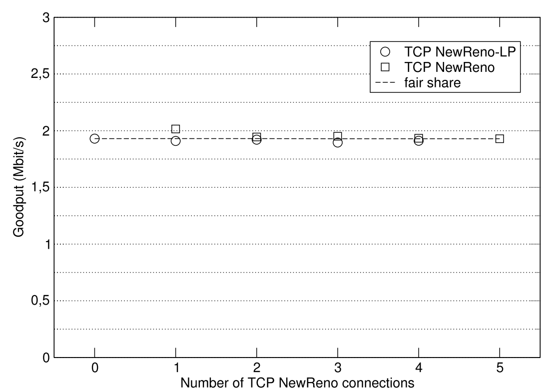

Fig. 13. TCP NewReno-LP fairness towards TCP NewReno over a 10 Mb/s link

We also extended our simulation campaign to more complex scenarios with a varying number of competing connections. The results obtained confirm that TCP NewReno-LP achieves the same level of fairness as TCP NewReno, thus allowing its smooth introduction into the Internet.

V. Conclusion

In this paper we have discussed the issues related to the use of Loss Predictors (LP) in TCP congestion control. These algorithms can potentially improve the throughput of TCP over wireless links affected by independent or correlated errors. We have proposed a modification to the Vegas predictor, studying its accuracy under several realistic scenarios. We have shown that this predictor achieves high accuracy in packet loss classification throughout all the network scenarios. We have proposed TCP NewReno-LP, an enhanced TCP version that uses the Vegas loss predictor in the TCP congestion control scheme to react to packet losses avoiding the overly conservative rate reductions of TCP Reno in the presence of random losses. We have studied the performance of TCP NewReno-LP, showing how it achieves higher performance over wireless networks while guaranteeing a good fairness level with existing TCP versions over wired links.

We have also provided an upper bound on the performance of TCPs using Loss Predictors by implementing an ideal scheme having perfect knowledge of the cause of packet losses, showing how TCP NewReno-LP approaches that bound.

REFERENCES

[1] T.V.Lakshman and U.Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking*, 5(3):336–350, 1997.

[2] L.S. Brakmo and L.L. Peterson. TCP Vegas: End-to-End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480, October 1995.

[3] S.Floyd and T.R.Henderson. The NewReno Modifications to TCP's Fast Recovery Algorithm. *IETF RFC 2582*, 26(4), April 1999.

[4] A.A.Abouzeid, S.Roy, and M.Azizoglu. Stochastic Modeling of TCP over Lossy Links. In *Proceedings of INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[5] M.Allman, V.Paxson, and W.Stevens. TCP Congestion Control. *RFC 2581*, April 1999.

[6] V.Tsaoussidis and I.Matta. Open Issues on TCP for Mobile Computing. In *Journal of Wireless Communications and Mobile Computing*, volume 2(2), February 2002.

[7] S.Cen, P.Cosman, and G.Voelker. End-to-End Differentiation of Congestion and Wireless Losses. In *Proceedings of ACM Multimedia Computing and Networking 2002*, volume 4673, pages 1–15, San Jose, CA, January 2002.

[8] S.Floyd and V.Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, 9:392–403, August 2001.

[9] J.Padhye, V.Firoiu, D.Towsley, and J.Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings of ACM SIGCOMM '98*, 1998.

[10] N.Cardwell, S.Savage, and T.Anderson. Modeling TCP Latency. In *Proceedings of INFOCOM 2000*, pages 1742–1751, 2000.