IMPROVING TCP PERFORMANCE OVER WIRELESS NETWORKS USING LOSS DIFFERENTIATION ALGORITHMS

Fabio Martignon

Politecnico di Milano, Dept. of Electronics and Information, Piazza L. Da Vinci 32
20133 Milano, ITALY, Tel.: +39-02-2399-3691 - Fax: +39-02-2399-3413

martignon@elet.polimi.it

Abstract   The use of loss differentiation algorithms within the congestion control scheme of TCP was proposed recently as a way of improving TCP performance over heterogeneous networks including wireless links affected by random loss. Such algorithms provide TCP with an estimate of the cause of packet losses. In this paper, we propose to use the Vegas loss differentiation algorithm to enhance the TCP NewReno error-recovery scheme, thus avoiding unnecessary rate reduction caused by packet losses induced by bit corruption on the wireless channel. We evaluate the performance of this enhanced TCP, showing that it achieves higher goodput over wireless networks, while guaranteeing fair share of network resources with classical TCP versions over wired links. Finally, by studying the TCP behavior with an ideal scheme having perfect knowledge of the cause of packet losses, we provide an upper bound to the performance of all possible schemes based on loss differentiation algorithms. The proposed TCP enhanced with Vegas loss differentiation algorithm well approaches this ideal bound.

Keywords: TCP, Wireless Networks, Internet, Protocols.

1.   Introduction

The Transmission Control Protocol (TCP) has proved efficient in classical wired networks, proving an ability to adapt to modern, high-speed networks and new scenarios for which it was not originally designed. However, modern wireless access networks, such as cellular networks and wireless local area networks, pose 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 random packet losses due to bit corruption in wireless channels and those due to network congestion.

Therefore, the TCP congestion control mechanism reduces, even when not necessary, the transmission rate. To avoid such limitation and degradation, several schemes have been proposed and are classified in [H.Balakrishnan et al., 1997]. A possible way to reduce the throughput degradation due to transmission errors is to use loss differentiation algorithms that try to estimate the cause of packet losses [S.Cen et al., 2003].

In this paper we propose to use the Vegas loss differentiation algorithm, also known in literature as Vegas Loss Predictor (LP) [Brakmo and Peterson, 1995] to enhance the TCP NewReno error-recovery scheme, as proposed in [S.Cen et al., 2003, Fu and S.C.Liew, 2003], thus avoiding unnecessary rate reduction caused by packet losses induced by bit corruption on the wireless channel.

We evaluate the performance of this enhanced TCP (TCP NewReno-LP), showing that it achieves higher goodput over wireless networks, with both long-lived and short-lived TCP connections, while guaranteeing fair share of network resources with current TCP versions over wired links. TCP NewReno-LP can be implemented by modifying the sender-side only of a TCP connection, thus allowing immediate deployment in the Internet.

We also evaluate the behavior of TCP enhanced with ideal loss prediction, assuming perfect knowledge of the cause of packet losses, thus providing an upper bound to the performance of all possible schemes based on different loss differentiation algorithms. The TCP enhanced with Vegas loss predictor well approaches this ideal bound.

The paper is structured as follows: Section 2 presents TCP NewReno-LP. Section 3 presents the simulation network model. Section 4 analyzes the accuracy of TCP NewReno-LP in estimating the cause of packet losses under several realistic network scenarios. Section 5 measures the performance of TCP NewReno-LP in terms of achieved goodput, fairness and friendliness, and its performance is compared to existing TCP versions, like TCP NewReno and TCP Westwood [Wang et al., 2002], over heterogeneous networks with both wired and wireless links affected by independent and correlated packet losses. Finally, Section 6 concludes the paper.

2.   TCP NewReno Enhanced with Vegas Loss Predictor

The Vegas loss predictor [Brakmo and Peterson, 1995] decides whether the network is congested or uncongested based on rate estimations. This predictor estimates the cause of packet losses based on the parameter V_P, calculated as

$$V_P = (\frac{cwnd}{RTT_{min}} - \frac{cwnd}{RTT}) \cdot RTT_{min} \qquad (1)$$

where $cwnd/RTT_{min}$ represents the *expected* flow rate and $cwnd/RTT$ the *actual* flow rate; $cwnd$ is the congestion window and RTT_{min} is the minimum Round Trip Time measured by the TCP source.

Given the two parameters α and β [segments], when $V_P \geq \beta$, the Vegas loss predictor assumes that the network is congested; when $V_P \leq \alpha$, possible losses will be ascribed to transmission random errors. 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 is asked to estimate 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, setting the slow start threshold ($ssthresh$) to half the current flight size. This allows TCP NewReno-LP to behave as fairly as the standard TCP protocol in congested network environments.

On the contrary, if the loss is classified as due to random bit corruption on the wireless channel, the $ssthresh$ is first updated to the current flight size value.

Then, if the packet loss has been detected by the TCP source after the receipt of 3 duplicate ACKs, the TCP sender updates the $cwnd$ to $ssthresh + 3$ Maximum Segment Sizes (MSS) and enters the fast retransmit phase as the standard TCP NewReno. This allows the source to achieve higher transmission rates upon the occurrence of wireless losses, if compared to the blind halving of the transmission rate performed by current TCP implementations.

If the packet loss has been detected by the TCP source after a retransmission timeout expiration, the congestion window is reset to 1 segment, thus enforcing a friendly behavior of the TCP source toward current TCP implementations.

3. Simulation Network Model

The TCP NewReno-LP scheme described in the previous Section was simulated using the Network Simulator package (ns v.2 [ns-2 network simulator (ver.2).LBL,]), evaluating its performance in several scenarios as proposed in [S.Floyd and V.Paxson, 2001].

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. The TCP receiver always implements the Delayed ACKs algorithm, as recommended in [M.Allman et al., 1999].

The network topology considered in this work is shown in Fig. 1. A single TCP NewReno-LP source performs a file transfer. The wired link $S \longleftrightarrow N$ has capacity C_{SN} and propagation delay τ_{SN}. The wireless link $N \longleftrightarrow D$ has capacity C_{ND} and propagation delay τ_{ND}.
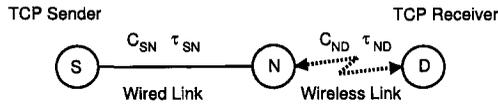
Figure 1.   Network topology in simulations for TCP performance evaluation.

We considered two different statistical models 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 [A.A.Abouzeid et al., 2000], 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 from 0% to 100%, to take into account different levels of fading.

4.   Accuracy Evaluation

The key feature of Loss Predictor schemes (LP) is to be accurate in estimating the cause of packet losses, as the TCP error-recovery algorithm we introduced in Section 2, based on the Vegas Predictor, reacts more gently or more aggressively than existing TCP sources depending on the LP estimate. Evidently, when the packet error rate is low and most of packet losses are due to congestion, LP accuracy in ascribing losses is necessary to achieve fairness and friendliness with concurrent TCP flows. On the other hand, when the packet error rate is high such as on wireless links, LP accuracy is necessary to achieve higher goodput, defined as the bandwidth actually used for successful transmission of data segments (payload).

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.
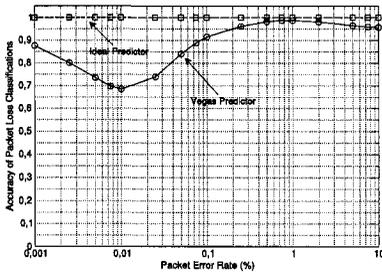
The overall *accuracy* of packet loss classification achieved by a loss predictor is thus defined as the ratio between the number of correct packet loss classifications and the total number of loss events.

We measured the accuracy of the Vegas predictor in the network topology of Fig. 1, with $C_{SN} = 10$ Mbit/s, $\tau_{SN} = 50$ ms and $C_{ND} = 10$ Mbit/s, $\tau_{ND} = 0.01$ ms. We considered both the scenarios with and without cross
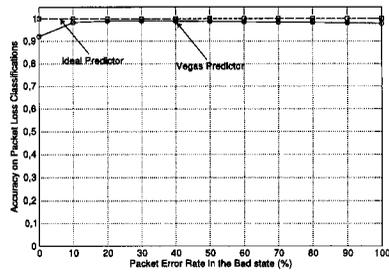
traffic on the wired link and both uncorrelated and correlated errors on the wireless link.

As explained in Section 2, 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 and we found the best performance for the accuracy of the Vegas predictor for $\alpha = 1$ and $\beta = 3$. We presented a detailed analysis of the accuracy of the Vegas predictor and other loss differentiation algorithms in [S.Bregni et al., 2003]. In this paper, we summarize only some of the most significant results.

Fig. 2(a) shows the accuracy of 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 and independent packet losses. Each accuracy value has been calculated over multiple file transfers, with very narrow 97.5% confidence intervals [K.Pawlikowski et al., 2002]. The vertical lines reported in all Figures represent such confidence intervals for each accuracy value.

(a) Independent losses   (b) Correlated losses

Figure 2.   Accuracy of classification of packet losses for the Vegas loss predictor as a function of PER in two scenarios: (a) independent packet losses (b) correlated packet losses

Fig. 2(b) shows the accuracy of the Vegas predictor when transmission errors are correlated and modeled as described in Section 3. The Vegas predictor provides high accuracy and approaches an ideal estimator for the whole range of packet error rates.

We have also extended our analysis to more complex network scenarios, with a varying number of TCP connections, multiple hops and various patterns of cross traffic on the wired link. For the sake of brevity we do not report these results. In all the scenarios we examined, the accuracy of the Vegas predictor has always been higher than 70%.

5.   TCP Performance over Wireless Links

So far, this paper has shown that TCP NewReno-LP performs an accurate estimation of the cause of packet losses in various network scenarios. However, as this algorithm is mainly designed to achieve high goodput in the presence of links affected by random errors, a study was made of the performance of this algorithm over wireless links.

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

Uncorrelated Losses

Following the guidelines proposed in [S.Floyd and V.Paxson, 2001], we considered the topology shown in Fig. 1. We analyzed three network scenarios with different capacity of the wired and wireless link: $C_{SN} = 2, 5$ or 10 Mbit/s and $C_{ND} = 10$Mbit/s. The Round Trip Time (RTT) is always equal to 100 ms and the queue can contain a number of packets equal to the bandwidth-delay product. We considered independent packet losses, modeled as described in Section 3. For each scenario we measured the steady state goodput obtained by TCP NewReno-LP (the bold line), TCP Westwood with NewReno extensions [High Performance Internet Research Group,] and TCP NewReno. All goodput values presented in this Section were calculated over multiple file transfers with a 97.5% confidence level [K.Pawlikowski et al., 2002]. The results are shown in Figures 3(a), 3(b) and 4, where the vertical lines represent, as in all the other Figures, the confidence interval for each goodput value.
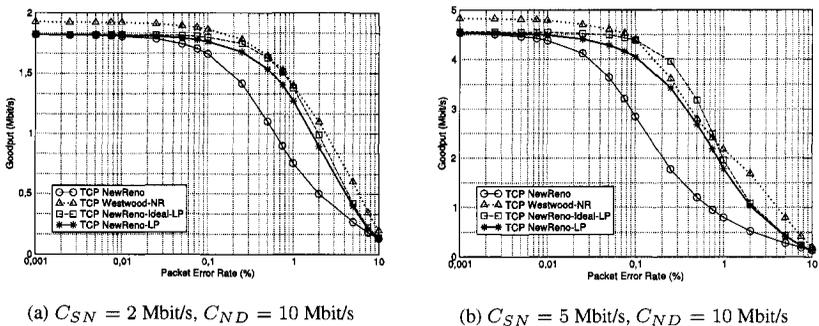

(a) $C_{SN} = 2$ Mbit/s, $C_{ND} = 10$ Mbit/s   (b) $C_{SN} = 5$ Mbit/s, $C_{ND} = 10$ Mbit/s

Figure 3.   Goodput achieved by various TCP versions in the topology of Fig. 1 as a function of PER
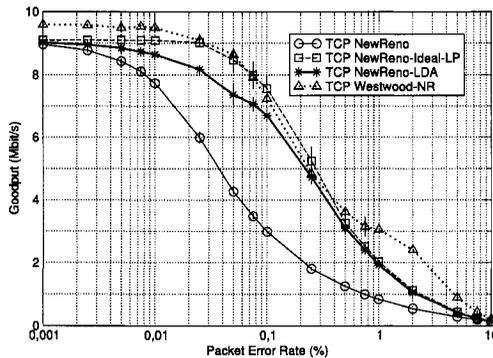
Figure 4.   Goodput achieved by various TCP versions in the topology of Fig. 1 with $C_{SN} = 2$ Mbit/s and $C_{ND} = 10$ Mbit/s as a function of PER

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.

Note that for packet error rates close to zero, when congestion is the main cause of packet losses, TCP NewReno-LP achieves practically the same good-put as TCP NewReno. This allows TCP NewReno-LP sources to share friendly network resources in mixed scenarios with standard TCP implementations, as it will be shown in Section 5.

In all the considered scenarios, we also measured the goodput achieved by a TCP Westwood source with NewReno extensions (TCP Westwood-NR). In all simulations this source achieved higher goodput than the other TCP versions, especially when the packet error rate was high. However, we believe that there is a trade-off between achieving goodput gain in wireless scenarios and being friendly toward existing TCP versions in mixed scenarios where the sources use different TCPs. In fact, if a TCP source is too aggressive and achieves a goodput higher than its fair share over a wired, congested link, its behavior is not friendly toward the other competing connections. This behavior will be analyzed in Section 5.

To provide a comparison, Figures 3(a), 3(b) and 4 also report the perfor-mance 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 predictors. Note that our scheme approaches this bound for all the con-sidered scenarios.

Correlated Losses

To account for the effects of multi-path fading typical of wireless environ-
ments, we also investigated the behavior of TCP NewReno-LP in the presence
of links affected by correlated errors, modeled as described in Section 3. We
considered two different scenarios with wireless link capacities equal to 2 and
5 Mbit/s, and a Round Trip Time equal to 100 ms. Fig. 5(a) 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
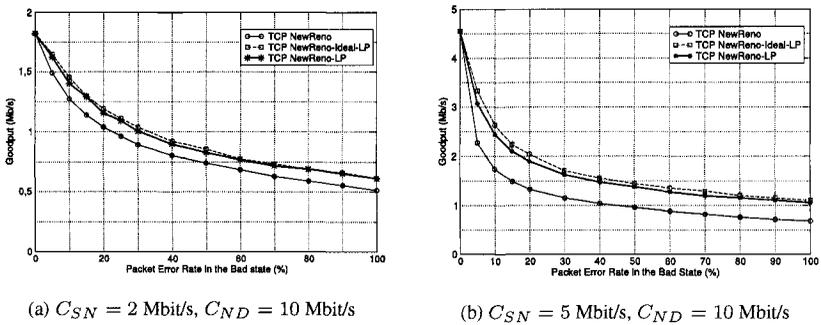bound achieved by the ideal scheme TCP NewReno-Ideal-LP.

(a) $C_{SN} = 2$ Mbit/s, $C_{ND} = 10$ Mbit/s   (b) $C_{SN} = 5$ Mbit/s, $C_{ND} = 10$ Mbit/s

Figure 5.   Goodput achieved by various TCP versions in the topology of Fig. 1 as a function of PER in
the *Bad* state

A similar behavior was observed in Fig. 5(b) where we reported the goodput
achieved by the analyzed TCP versions in the topology shown in Fig. 1 with
a 5 Mbit/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 Mbit/s scenario, as wireless losses
affect more heavily TCP NewReno goodput when the bandwidth-delay product
of the connection is higher [T.V.Lakshman and U.Madhow, 1997].

Impact of Round Trip Time

Packet losses are not the only cause of TCP throughput degradation. Many
studies [J.Padhye et al., 1998] have pointed out that TCP performance also de-
grades when the Round Trip Time (RTT) of the connection increases. TCP
NewReno-LP allows to alleviate this degradation to improve performance.
Fig. 6(a) and 6(b) report the goodput achieved by TCP NewReno and TCP
NewReno-LP sources transmitting over a single link with capacity equal to 2
Mbit/s and a 5 Mbit/s, respectively, as a function of the Round Trip Time of

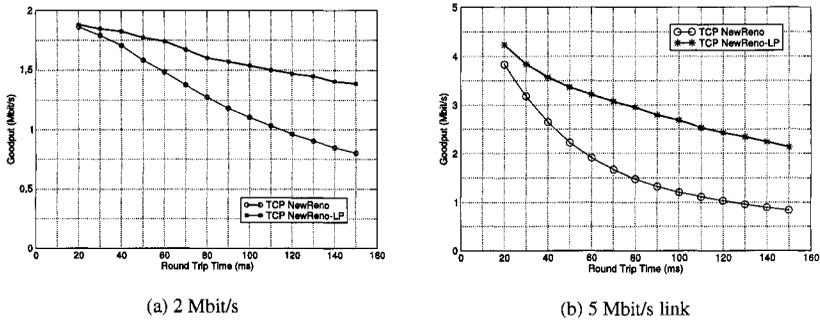the connection. The link drops packets independently with a loss probability constantly equal to 0.5%.

(a) 2 Mbit/s   (b) 5 Mbit/s link

Figure 6. Goodput achieved by TCP NewReno-LP and TCP NewReno over a single link as a function of the RTT of the connection

We point out the high goodput gain of TCP NewReno-LP over TCP NewReno. This behavior is more evident when the Round Trip Time of the connection increases.

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 [N.Cardwell et al., 2000], a typical HTTP connection involving the transfer of a 10 kbyte file over a 5 Mbit/s link affected by a 5% random packet loss, with 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 seconds for TCP NewReno-LP and 0.81 seconds for TCP NewReno. Hence, also for short file transfers, TCP NewReno-LP achieves the same results as the current TCP version.

Friendliness and Fairness

So far we have shown that the TCP NewReno-LP scheme estimates accurately the cause of packet losses and that achieves higher goodput than existing TCP versions over wireless links with both uncorrelated and correlated losses.

Following the methodology proposed in [Wang et al., 2002], we evaluated friendliness and fairness of TCP NewReno-LP in a variety of network scenarios and we compared them by those achieved by TCP Westwood-NR. The term *friendliness* relates to the performance of a set of connections using different

TCP flavors, while the term *fairness* relates to the performance of a set of TCP connections implementing the same algorithms.

This section shows how the proposed scheme is able to share friendly and fairly network resources in mixed scenarios where the sources use different TCPs.

To this purpose, we first evaluated TCP NewReno-LP *friendliness* by considering two mixed scenarios: in the first one 5 TCP connections using either TCP NewReno-LP or TCP NewReno share an error-free link with capacity equal to 10 Mbit/s and RTT equal to 100 ms; in the second one the TCP NewReno-LP sources were replaced by TCP Westwood-NR sources.

By simulation we measured the goodput, for each connection, and for all cases. The average goodput of n TCP NewReno-LP and of m TCP NewReno connections, with $n + m = 5$, is shown in Fig. 7(a).
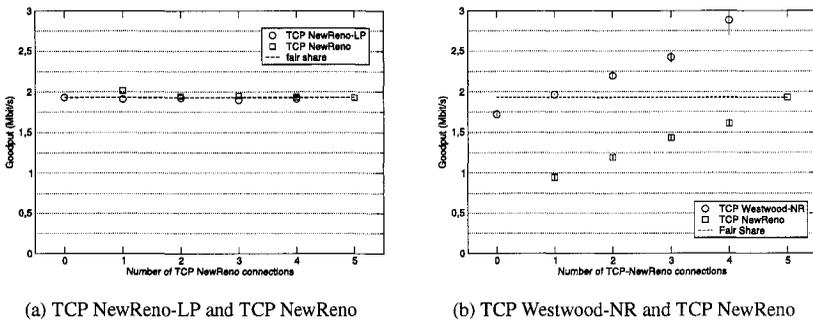

(a) TCP NewReno-LP and TCP NewReno   (b) TCP Westwood-NR and TCP NewReno

Figure 7.   Average goodput of (a) n TCP NewReno-LP and m TCP NewReno connections and (b) n TCP Westwood-NR and m TCP NewReno connections, with $n + m = 5$, over a 10 Mbit/s link with RTT equal to 100 ms

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 TCP connections using either TCP Westwood-NR or TCP NewReno, and the results are shown in Fig. 7(b). In this scenario TCP Westwood-NR sources proved more aggressive toward TCP NewReno sources than TCP NewReno-LP, and achieved a goodput higher than the fair share practically in every case. This behavior evidences the trade off that exists between achieving high goodput gain in wireless scenarios and being friendly in mixed network scenarios.

To measure the level of *fairness* achieved by TCP NewReno-LP we considered the same scenario described above first with 5 TCP NewReno-LP connections and then with 5 TCP NewReno sources sharing a 10 Mbit/s link with RTT equal to 100 ms. In this scenarios congestion is the only cause of packet

losses. The Jain's fairness index of 5 TCP NewReno-LP connections was equal to 0.9987, and that achieved by 5 TCP NewReno sources was equal to 0.9995. These results confirm that TCP NewReno-LP achieves the same level of fairness of TCP NewReno.

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 an high level of friendliness toward TCP NewReno, thus allowing its smooth introduction into the Internet.

6.   Conclusions

In this paper, we have discussed and analyzed issues related to the use of Loss Differentiation Algorithms for TCP congestion control. We proposed to use the Vegas loss predictor to enhance the TCP NewReno error-recovery scheme, thus avoiding unnecessary rate reductions caused by packet losses induced by bit corruption on the wireless channel. The performance of this enhanced TCP (TCP NewReno-LP) was evaluated by extensive simulations, examining various network scenarios. Two types of TCP connections were considered, namely long-lived connections, typical of file transfers, and short-lived connections, typical of HTTP traffic. Moreover, we considered two different statistical models of packet losses on the wireless link: independent and correlated losses. We found that TCP NewReno-LP achieves higher goodput over wireless networks, while guaranteeing good friendliness with classical TCP versions over wired links. Moreover, we found that the Vegas loss predictor, embedded in TCP NewReno-LP, proved very accurate in classifying packet losses. Finally, we also defined an ideal scheme that assumes the exact knowledge of packet losses and provides an upper bound to the performance of all possible schemes based on loss differentiation algorithms. The TCP enhanced with Vegas loss predictor well approaches this ideal bound.

References

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

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

[Fu and S.C.Liew, 2003] Fu, C. P. and S.C.Liew (Feb. 2003). TCP Veno: TCP enhancement for transmission over wireless access networks. *IEEE Journal on Selected Areas in Communications*, 21(2):216–228.

[H.Balakrishnan et al., 1997] H.Balakrishnan, V.N.Padmanabhan, S.Seshan, and R.H.Katz (December 1997). A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Transactions on Networking*, 5(6):759–769.

[High Performance Internet Research Group,] High Performance Internet Research Group, U. TCP Westwood Home Page, URL: http://www.cs.ucla.edu/NRL/hpi/tcpw.

[J.Padhye et al., 1998] J.Padhye, V.Firoiu, D.Towsley, and J.Kurose (1998). Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings of ACM SIG-COMM '98*.

[K.Pawlikowski et al., 2002] K.Pawlikowski, Jeong, H., and Lee, J. (Jan. 2002). On Credibility of Simulation Studies of Telecommunication Networks. *IEEE Communications Magazine*, pages 132–139.

[M.Allman et al., 1999] M.Allman, V.Paxson, and W.Stevens (April 1999). TCP Congestion Control. *RFC 2581*.

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

[ns-2 network simulator (ver.2).LBL,] ns-2 network simulator (ver.2).LBL. URL: http://www.isi.edu/nsnam.

[S.Bregni et al., 2003] S.Bregni, D.Caratti, and F.Martignon (San Francisco, 1-5 Dec. 2003). Enhanced Loss Differentiation Algorithms for Use in TCP Sources over Heterogeneous Wireless Networks. In *Proceedings of IEEE GLOBECOM'03*.

[S.Cen et al., 2003] S.Cen, P.C.Cosman, and G.M.Voelker (Oct. 2003). End-to-end Differentiation of Congestion and Wireless Losses. *IEEE/ACM Transactions on Networking*, 11(5):703–717.

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

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

[Wang et al., 2002] Wang, R., Valla, M., Sanadidi, M., and Gerla, M. (2002). Adaptive Bandwidth Share Estimation in TCP Westwood. In *Proceedings of Globecom'02*.