Loss Differentiation Schemes for TCP over Wireless Networks*

Fabio Martignon and Luigi Fratta

DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milan, Italy
{martignon,fratta}@elet.polimi.it

Abstract. The use of loss differentiation schemes within the congestion control mechanism 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 the so-enhanced TCP NewReno source (TCP NewReno-LP) with both extensive simulation and real test bed measurements, and we compare it with that achieved by existing solutions, namely TIBET [1], TCP Westwood [2] and the standard TCP NewReno. For that purpose, Linux implementations of TCP NewReno-LP, TIBET and TCP Westwood have been developed and compared with an implementation of NewReno.
We show that TCP NewReno-LP 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.

1 Introduction

The Transmission Control Protocol (TCP) performs well over the traditional network, that is constructed by purely wired links. However, as wireless access networks (like cellular networks and wireless local area networks) are growing rapidly, a heterogeneous environment will get wide deployment in the next-generation wireless networks, thus posing new challenges to the TCP congestion control scheme.

The performance degradation of existing versions of TCP in wireless and wired-wireless hybrid networks is mainly due to their lack of the ability to differentiate the packet losses caused by network congestions from the losses caused by wireless link errors. Therefore, the standard 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 [3].

* This work has been partially supported under the grant of MURST Tango project.

M. Ajmone Marsan et al. (Eds.): QoS-IP 2005, LNCS 3375, pp. 586–599, 2005.

A possible approach to this problem is to modify the TCP congestion control scheme implementing explicit bandwidth estimation [1, 2] and loss differentiation schemes [4–8]. Note that these two approaches are deeply interwined, as we showed in [4] that the most efficient loss differentiation algorithms base their functioning on bandwidth measurements to estimate the cause of packet losses.

We analyzed and discussed in detail the first approach in [1] where we proposed TIBET, a new bandwidth estimation algorithm that allows to obtain accurate and unbiased estimates of the TCP trasmission rate.

In this paper, we propose a new TCP scheme, called TCP NewReno-LP, which is capable of distinguishing the wireless packet losses from the congestion packet losses, and reacting accordingly. TCP NewReno-LP implements an enhanced error-recovery scheme as proposed in [4–8], based on the Vegas Loss Predictor (LP) [9], and it avoids unnecessary rate reduction caused by packet losses induced by bit corruption on the wireless channel. TCP NewReno-LP can be implemented by modifying the sender-side only of a TCP connection, thus allowing immediate deployment in the Internet.

We evaluate the performance of TCP NewReno-LP with both simulation and real test bed measurements. For that purpose, Linux implementations of TCP NewReno-LP, TIBET and TCP Westwood have been developed and compared with an implementation of NewReno. We compare the performance of TCP NewReno-LP with that achieved by TIBET, TCP Westwood and standard TCP NewReno, showing how the proposed enhanced TCP source achieves higher goodput over wireless networks, while guaranteeing fair share of network resources with current TCP versions over wired links.

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. Sections 5 and 6 measure the performance of TCP NewReno-LP in terms of achieved goodput, friendliness and fairness, using both simulation and real Test bed scenarios, respectively. The performance of TCP NewReno-LP is compared to existing TCP versions, like TCP NewReno, TIBET and TCP Westwood [2, 10], over heterogeneous networks with both wired and wireless links affected by independent and correlated packet losses. Finally, Section 7 concludes this paper.

2 TCP NewReno Enhanced with Vegas Loss Predictor

The Vegas loss predictor [9] 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:

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

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 [11], 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 [12]), evaluating its performance in several scenarios as proposed in [13].

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 [14].

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}.

We considered two different statistical models of packet losses on the wireless link: independent and correlated losses. To model independent packet losses, the

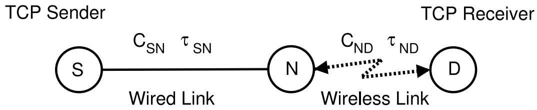


Fig. 1. Network topology in simulations for TCP performance evaluation.

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 [15], 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.

Finally, we considered two different traffic scenarios: in the first one, no cross traffic is transmitted over the wired link $S \longleftrightarrow N$; 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 states, with Pareto-distributed periods having shape parameter equal to 1.5 and mean durations equal to 100 ms and 200 ms, respectively. During the ON state, each source transmits packets with 1500 byte size at constant bit rate equal to R_{UDP} Mbit/s, while in OFF period the UDP sources do not transmit any packet. In every network scenario with cross traffic on the wired link, the value of R_{UDP} is chosen to leave to the TCP source an available bandwidth that varies randomly during the simulation, with an average equal to half the bottleneck capacity.

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).

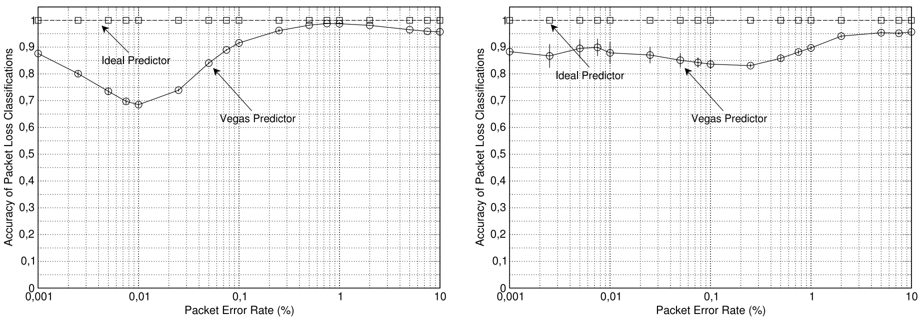
TCP sources detect *loss events* based on the reception of triple duplicate acknowledgements or retransmission timeout expirations. We define *wireless loss* a packet loss caused by the wireless noisy channel; a *congestion loss* is defined as a packet loss caused by network congestion.

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 [4]. 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 [16]. The vertical lines reported in all Figures represent such confidence intervals for each accuracy value.



(a) No cross traffic. (b) With cross traffic.

Fig. 2. Accuracy of classification of packet losses for the Vegas loss predictor as a function of PER in two scenarios: (a) no cross traffic on the wired link (b) with cross traffic on the wired link.

Fig. 2(b) shows the accuracy for the Vegas predictor in the scenario with cross traffic on the wired link ($R_{UDP} = 0.5$ Mbit/s). 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.

Finally, Fig. 3 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 and multiple hops. 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%.

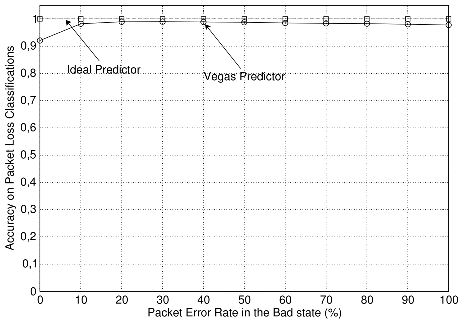


Fig. 3. Accuracy of classification of packet losses for the Vegas loss predictor as a function of PER in the *Bad* state in the scenario with no cross traffic on the wired link.

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.

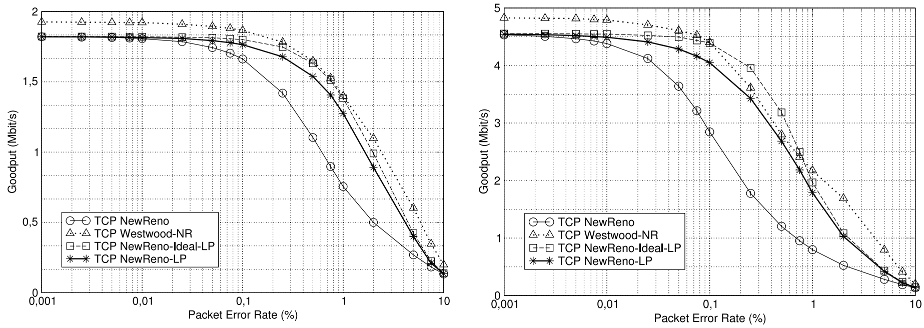
To measure TCP NewReno-LP performance, and compare it with other TCP versions, we first considered several simulated network scenarios with long-lived TCP connections, typical of FTP file transfers. In the following we discuss the results obtained by simulation.

5.1 Uncorrelated Losses

Following the guidelines proposed in [13], 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 [17] and TCP NewReno. All goodput values presented in this Section were calculated over multiple file transfers with a 97.5% confidence level [16]. The results are shown in Figures 4(a), 4(b) and 5, where the vertical lines represent, as in all the other Figures, the confidence interval for each goodput value.

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-



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

Fig. 4. Goodput achieved by various TCP versions in the topology of Fig. 1 as a function of PER.

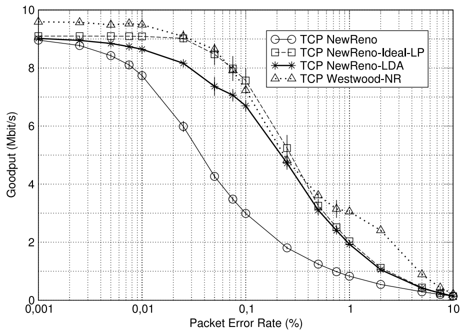
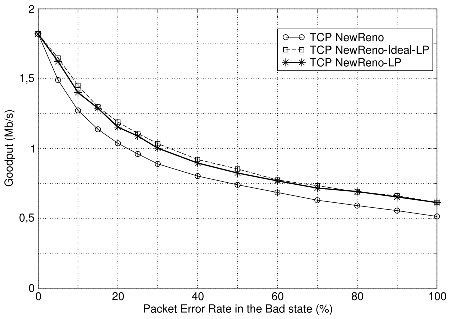


Fig. 5. Goodput achieved by various TCP versions in the topology of Fig. 1 with $C_{SN} = 2$ Mbit/s and $C_{ND} = 10$ Mbit/s as a function of PER.

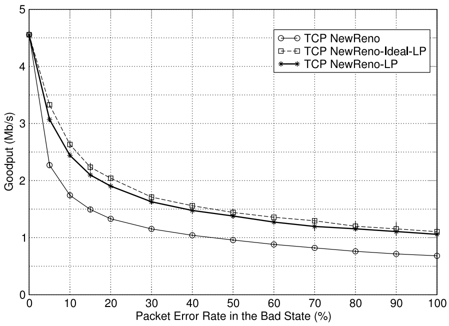
put as TCP NewReno. This allows TCP NewReno-LP sources to share friendly network resources in mixed scenarios with standard TCP implementations, as it will be shown in Section 5.5.

In all the considered scenarios, we also measured the goodput achieved by a TCP Westwood source with NewReno extensions (TCP Westwood-NR), using the *ns* modules available at [17]. 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, again, in Section 5.5.

To provide a comparison, Figures 4(a), 4(b) and 5 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



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



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

Fig. 6. Goodput achieved by various TCP versions in the topology of Fig. 1 as a function of PER in the *Bad* state.

an upper bound on the performance achievable by every scheme based on loss predictors. Note that our scheme approaches this bound for all the considered scenarios.

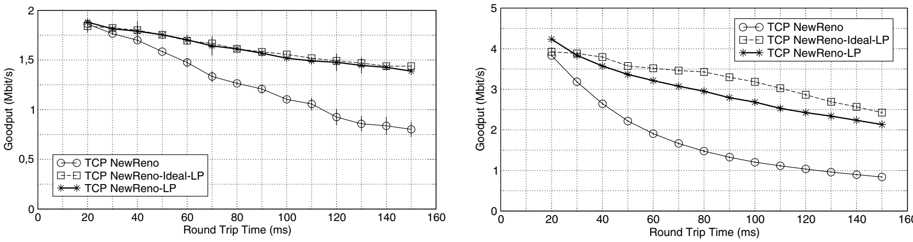
5.2 Correlated Losses

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, 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. 6(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 similar behavior was observed in Fig. 6(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 [18].

5.3 Impact of Round Trip Time

Packet losses are not the only cause of TCP throughput degradation. Many studies [19] have pointed out that TCP performance also degrades when the Round Trip Time (RTT) of the connection increases. TCP NewReno-LP allows to alleviate this degradation to improve performance. Fig. 7(a) and 7(b) report the goodput achieved by TCP NewReno, TCP NewReno-LP and TCP NewReno-Ideal-LP sources transmitting over a single link with capacity equal to 2 Mbit/s



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

Fig. 7. Goodput achieved by TCP NewReno-LP, TCP NewReno-LP with Ideal Predictos and TCP NewReno over a single link as a function of the RTT of the connection.

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%.

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. Note that, even in this scenario, TCP NewReno-LP practically overlaps to the goodput upper bound achieved by the ideal scheme TCP NewReno-Ideal-LP.

5.4 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 [10], 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.

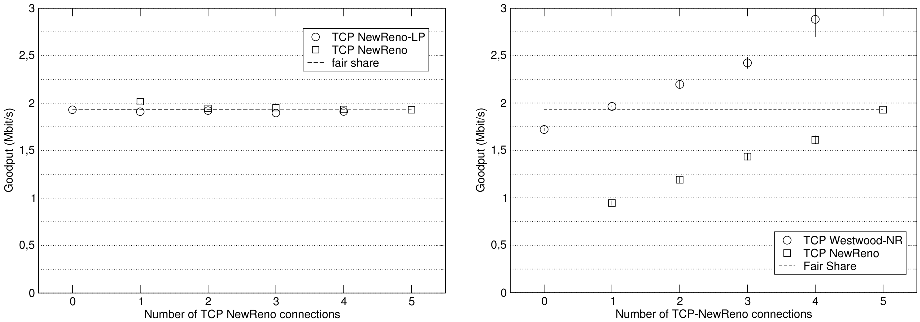
This section shows how the proposed scheme is able to share friendly and fairly network resources in mixed scenarios where the sources use different TCPs.

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. 8(a).

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. 8(b).



(a) TCP NewReno-LP and TCP NewReno.

(b) TCP Westwood-NR and TCP NewReno.

Fig. 8. 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.

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 [21] 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 Implementation and Test Bed

To get more details on the TCP NewReno-LP implementation we have built a test bed, shown in Fig. 9 that consists of a PC server, a client and a PC router, all connected by 10 Mb/s LAN cables. The PC router emulates a wireless link with the desired delay and packet loss rate using the NIST Net software [22], thus allowing to control and tune the features of the wireless link.

In the PC server, besides the TCP NewReno that is the current TCP implementation in the Linux kernel version 2.2-20, we have implemented TCP NewReno-LP, TIBET and TCP Westwood. The choice to implement the TCP variants detailed above in the Linux kernel version 2.2-20 was motivated by the

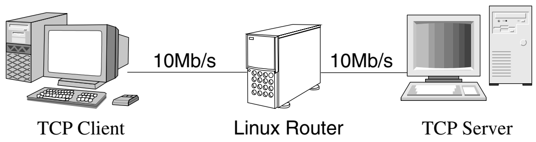


Fig. 9. Test bed Topology for TCP performance evaluation.

observation that this version is fully compliant with the standard TCP implementation as recommended in [14, 11]. Successive versions of the Linux kernel, starting from 2.4, introduced improved features as the Rate-Halving algorithm and the so-called *undo procedures* that are not yet considered standard and can have a deep impact on TCP performance, thus masking the advantages introduced by bandwidth estimation and loss differentiation techniques.

6.1 Uncorrelated Losses

Running the test bed we measured the goodputs achieved by the four TCP versions. Fig. 10 compares the steady-state goodput achieved by TCP NewReno-LP, TIBET, TCP Westwood and TCP NewReno connections transmitting data between the server and the client, with an emulated round trip time equal to 100 ms versus packet loss rates.

The measures on this real scenario validate the results obtained by simulation (see Fig. 4(a) and 4(b)) and provide a further support on the advantages of TCP NewReno-LP over TCP NewReno. Fig. 10 also shows the improvement achieved by TCP NewReno-LP over TIBET, more evident for PER values in the 1% to 4% range.

Note that in this scenario, as well as in all the simulated scenarios presented in this Section, TCP Westwood obtained a higher goodput than any other TCP

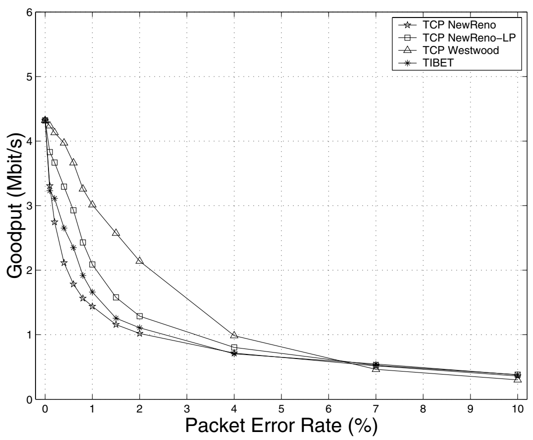


Fig. 10. Goodput achieved by TCP NewReno-LP; TIBET, TCP Westwood and TCP NewReno in the Test Bed.

version. This behavior is due to its overestimate of the available bandwidth, that leads to aggressive behavior and unfair sharing of network resources, as we showed in the previous Section and as we discussed in detail in [1].

6.2 Correlated Losses

We then considered the same two-state Markov model described in Section 3 to model correlated losses, and we measured the goodput achieved by TCP sources as a function of the packet error rate in the Bad state, to take into account various levels of fading. The results are reported in Figure 11.

These results confirm the improved performance achieved by TCP NewReno-LP even in this network scenario that models very closely real wireless link conditions.

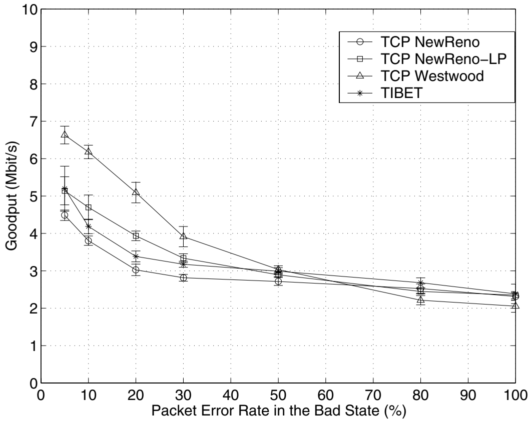


Fig. 11. Goodput Achieved by various TCP versions in the presence of correlated losses.

7 Conclusions

In this work 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 and real testbeds, 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

1. A. Capone, L. Fratta, and F. Martignon. Bandwidth Estimation Schemes for TCP over Wireless Networks. *IEEE Transactions on Mobile Computing*, 3(2):129–143, 2004.
2. C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang. TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks. In *Wireless Networks Journal*, volume 8, pages 467–479, 2002.
3. H.Balakrishnan, V.N.Padmanabhan, S.Seshan, and R.H.Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. In *CIEEE/ACM Transactions on Networking*, volume 5(6), pages 759–769, December 1997.
4. S. Bregni, D. Caratti, and F. Martignon. Enhanced Loss Differentiation Algorithms for Use in TCP Sources over Heterogeneous Wireless Networks. In *Proceedings of GLOBECOM'03*, San Francisco, 1-5 Dec. 2003.
5. S.Cen, P.C.Cosman, and G.M.Voelker. End-to-end Differentiation of Congestion and Wireless Losses. *IEEE/ACM Transactions on Networking*, 11(5):703–717, Oct. 2003.
6. Cheng Peng Fu and S. C. Liew. TCP Veno: TCP enhancement for transmission over wireless access networks. In *IEEE Journal on Selected Areas in Communications*, volume 21(2), pages 216–228, Feb. 2003.
7. K. Xu, Y. Tian, and N. Ansari. TCP-Jersey for Wireless IP Communications. In *IEEE Journal on Selected Areas in Communication*, volume 22(4), May 2004.
8. Eric Hsiao-Kuang Wu and Mei-Zhen Chen. JTCP: Jitter-Based TCP for Heterogeneous Wireless Networks. In *IEEE Journal on Selected Areas in Communication*, volume 22(4), May 2004.
9. 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.
10. R. Wang, M. Valla, M.Y. Sanadidi, and M. Gerla. Adaptive Bandwidth Share Estimation in TCP Westwood. In *Proceedings of Globecom'02*, 2002.
11. S. Floyd and T. R. Henderson. The NewReno Modifications to TCP's Fast Recovery Algorithm. *IETF RFC 2582*, 26(4), April 1999.
12. ns-2 network simulator (ver.2).LBL. URL: http://www.isi.edu/nsnam.
13. S. Floyd and V. Paxson. Difficulties in Simulating the Internet. In *IEEE/ACM Transactions on Networking*, volume 9, pages 392–403, August, 2001.
14. M.Allman, V.Paxson, and W.Stevens. TCP Congestion Control. *RFC 2581*, April 1999.
15. 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.
16. K. Pawlikowski, H.-D.Joshua Jeong, and J.-S.Ruth Lee. On Credibility of Simulation Studies of Telecommunication Networks. *IEEE Communications Magazine*, pages 132–139, Jan. 2002.

17. UCLA High Performance Internet Research Group. Tcp westwood home page, URL: http://www.cs.ucla.edu/nrl/hpi/tcpw.
18. 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.
19. 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.
20. N. Cardwell, S. Savage, and T. Anderson. Modeling TCP Latency. In *Proceedings of INFOCOM 2000*, pages 1742–1751, 2000.
21. R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling.* Wiley, New York, 1991.
22. *NIST Net Home Page.* Available at http://snad.ncsl.nist.gov/itg/nistnet/.