Performance evaluation of Westwood+ TCP over WLANs with Local Error Control *

Luigi A. Grieco and Saverio Mascolo
Dipartimento di Elettrotecnica ed Elettronica,
Politecnico di Bari,
Via Orabona, 4 – 70125 BARI, Italy
e-mail{a.grieco, mascolo}@poliba.it

Abstract

Link layer error control is widely adopted in Wireless LANs (WLANs) to hide the unreliability of the wireless channel to higher level protocols. This allows classic end-to-end congestion control algorithms to achieve acceptable throughput even in the presence of radio links. The present work investigates the performance of Westwood+ and NewReno TCP over a noisy wireless LAN channel. In particular, the effect of local error control persistency on the end-to-end performances of Westwood+ and NewReno TCP is addressed. The investigation has been carried out by using ns-*2 computer simulation. Simulation results show that (1) in the presence of uniformly distributed packet losses a well tuned local error control leads both Westwood+ and NewReno TCP to full network utilization; (2) in the presence of bursty losses Westwood+ TCP improves the goodput with respect to NewReno TCP also in the presence of local error control; in particular, Westwood+ TCP requires a smaller number of retransmissions at the link layer than New Reno to achieve full utilization of the wireless channel.*

1 Introduction

The objective of end-to-end TCP congestion control is to regulate the sending rate of the data sources in order to match the network capacity [16, 12]. TCP congestion control is based on the *self-clocking* principle, which states that, in order not to exceed the network capacity, a new segment can be transmitted only when a previous one has left the network. For that purpose the congestion window (*cwnd*), is used to bound the number of released and not yet acknowledged segments (i.e. the outstanding segments) [16, 19]. In order to achieve high network utilization while guaranteeing sudden reactions to network congestion, the *self-clocking* principle is coupled with the *additive increase multiplicative decrease* probing paradigm (AIMD)[7, 8]. In the AIMD mechanism, the *cwnd* is additively inflated until the network capacity is hit and a congestion episode happens. The sender becomes aware of congestion via the reception of duplicate acknowledgments (DUPACKs) or the expiration of a timeout. Then it reacts to light congestion (i.e. 3 DUPACKs) by halving the congestion window (fast recovery) and sending again the missing packet (fast retransmit), and to heavy congestion (i.e. timeout) by reducing the congestion window to one. This is the multiplicative decrease phase that quickly shrinks the input rate after congestion.

While today end-to-end TCP congestion control can insure that network capacity is not exceeded, it cannot insure fair sharing of that capacity [16, 20, 14]. Furthermore, today TCP is not well suited for wireless links since losses due to radio channel problems are misinterpreted as a symptom of congestion by current TCP schemes and lead to an undue transmission rate reduction. Thus, TCP requires supplementary link layer protocols such as reliable link-layer or split-connections approach to efficiently operate over wireless links [5, 17, 4, 10, 6].

Westwood TCP proposes to improve fairness and efficiency of congestion control by introducing an innovative end-to-end bandwidth estimation algorithm. In particular, Westwood TCP leaves unchanged the probing phase of classic TCP but it substitutes the multiplicative decrease phase with an adaptive decrease phase, which sets the control windows by taking into account the bandwidth estimate [20]. Westwood+ TCP is a recent modification to Westwood TCP that employs an improved end-to-end bandwidth estimation scheme, which is robust with respect to ACK compression [15]. It has been shown that Westwood+ TCP increases the TCP throughput over lossy links, increases the fairness in

*This work was supported by the MIUR-FIRB project n. RBNE01BNL5 "Traffic Models and Algorithms for Next Generation IP networks Optimization (TANGO)"

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

bandwidth allocation *wrt* NewReno TCP and is friendly towards NewReno TCP [20, 14, 11].

The performances of Westwdood+ TCP over unreliable links have been investigated in [15] but it hasn't yet been shown if the Westwood+ TCP algorithm still provides performance improvements also in the presence of local error control algorithms.

In order to address this issue, the present work investigates the performance of Westwood+ and NewReno TCP [3, 13] over a noisy wireless LAN channel. In particular, the effect of local error control persistency on the end-to-end performances of Westwood+ and NewReno TCP is addressed. The investigation has been carried out by using *ns*-2 [21] computer simulations.

The work is organized as follows: in Section 2 basic 802.11 WLAN features are described, in Section 3 the NewReno TCP and Westwood+ TCP algorithms are briefly summarized, in Section 4 *ns*-2 computer simulations are reported to assess the impact of local error control on Westwood+ and NewReno TCP. Finally the last section draws the conclusions.

2 WLAN Background

The basic building block of a WLAN is referred as BSS (Basic Service Set), i.e., a group of stations that communicate through the wireless medium. There are two WLAN topology [22, 9, 18]: the IBSS (Independent BSS) in which a group of independent stations directly communicate with each other and the ESS (Extended Service Set), where many BSS are interconnected through a Distribution System (i.e, a network backbone typically wired) forming an infrastructure network. In the latter, all traffic is channelled through an Access Point (AP).

The primary 802.11 MAC protocol is the Distributed Coordination Function (DCF), which supports asynchronous data transfer on a best effort basis. DCF is based on CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance): all stations contend for access to the radio channel and listen if channel is idle before transmitting data. Each successful transmitted frame is acknowledged by the receiving station with an ACK frame. If there is an unsuccessful transmission (i.e., no ACK frame is received by the sending station), the station restarts sensing the channel. Each delivered packet contains a duration field indicating the time required for the transmission of the MAC Protocol Data Unit (MPDU). The duration field is used by all the stations in the BSS to adjust their Network Allocation Vector (NAV), which define the amount of time that must elapse until the current transmission session is finished and the channel can be tested again for the access.

The above described access mechanism is subject to the hidden station problem, i.e., a single receiver station (generally the AP) can hear two different transmitters, but the two transmitters each other because they are out of radio range. Given that each source cannot detect the collision, the two transmitting stations will continue to send their MPDUs. With large MPDU (e.g., 2300 bytes) a lot of channel bandwidth will be wasted due to corrupt data [9]. To reduce this effect, the 802.11 standard defines an optional Request-to-Send/Clear-to-Send (RTS/CTS) mechanism. RTS and CTS control frames (relatively small, they are 20 and 14 bytes long, respectively) are used by stations to reserve bandwidth before data transmission. It is worth noticing that the RTS/CTS handshake is used only if the MPDU to be transmitted is larger than a threshold, which is generally referred as *RTSthreshold*.

2.1 WLAN local error control

The WLAN local error control is used when a transmitted MAC frame gets lost due to interference or channel unreliability. In this case, the lost frame is retransmitted up to a maximum number of times. The retransmission persistency varies depending on the size of the MAC frame. In particular, if the transmitted frame is shorter than the *RTSthreshold*, then it is retransmitted up to the maximum number of times indicated by the *Short Retry Limit* parameter. Otherwise, the *Long Retry Limit* parameter is used as upper bound to the number of retransmissions. When an *RTS* packet gets lost, *the Short Retry Limit* is used before discarding the data.

3 Westwood+ TCP outline

A TCP connection is characterized by the following variables: (1) congestion window ($cwnd$); (2) slow start threshold ($ssthresh$); (3) round trip time of the connection (RTT); (4): minimum round trip time measured by the sender (RTT_{min}); (5) size of the delivered segments (seg_size).

The pseudo-code of the NewReno algorithm is reported below:

```
- When 3 DUPACKs are received by the
sender (fast recovery):
    ssthresh =cwnd/2;
    if ssthresh<2 then ssthresh=2;
    cwnd = ssthresh;
- When a coarse timeout expires:
    ssthresh =cwnd/2;
    if ssthresh<2 then ssthresh=2;
    cwnd = 1;
- When ACKs are successfully received:
    if (cwnd<ssthresh) cwnd=cwnd+1
    else cwnd=cwnd+1/cwnd.
```

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

When 3DUPACKs are received, the Reno TCP algorithm enters the *fast recovery phase* and retransmits the segment that has been acked 3 times. This phase is leaved when the retransmitted segment is successfully acknowledged. NewReno differs from Reno TCP in that it does not exit the *fast recovery* phase until all the segments within the current window are acknowledged. This feature, which is known as NewReno feature, improves the performance of Reno TCP when several segments within the same window get lost [13].

The key idea of TCP Westwood+ is to exploit the stream of returning acknowledgment packets to estimate the bandwidth B that is available for the TCP connection. When a congestion episode happens at the end of the TCP probing phase, the used bandwidth corresponds to the definition of best effort available bandwidth in a connectionless packet network. The bandwidth estimate is used to adaptively decrease the congestion window and the slow-start threshold after a timeout or three duplicate ACKs as it is described below:

```
- When 3 DUPACKs are received by the
sender:
     ssthresh =(B*RTTmin)/seg_size;
    if ssthresh<2 then ssthresh=2;
    cwnd = ssthresh;
- When a coarse timeout expires:
    ssthresh =(B*RTTmin)/seg_size;
    if ssthresh<2 then ssthresh=2;
    cwnd = 1;
- When ACKs are successfully received:
    cwnd increases following the New
Reno algorithm.
```

It is worth noting that the adaptive decrease mechanism employed by Westwood+ TCP improves the stability of the standard TCP multiplicative decrease algorithm. In fact, the adaptive window shrinking provides a congestion window that is decreased enough in the presence of heavy congestion and not too much in the presence of light congestion or losses that are not due to congestion, such as in the case of unreliable radio links [15].

4 Performance evaluation

This section evaluates the interaction between the WLAN local error control and the end-to-end TCP congestion control algorithms New Reno and Westwood+ TCP.

Computer simulations using *ns*-2 [21] have been employed to assess the impact of the retry limit value on the end-to-end performances in the presence of random packet losses.

The scenario considered in this section is illustrated in Fig. 1. It consists of a 11Mbps WLAN, which is accessed by a TCP sender to upload data towards a remote destination. The WLAN AP is connected to the destination by a 2Mbps duplex link with 100ms delay propagation. The queue feeding the 2Mbps bottleneck link is set equal to the bandwidth delay product. TCP segments are 1500Bytes long and the *RTSThreshold* parameter has been set equal to 0 in order to keep always enabled the RTS/CTS handshake. The TCP sender sets the initial window equal to 3 as suggested in [2]. The wireless channel is affected by random packet losses. Both uniformly distributed random losses and bursty periodic random losses have been considered. NewReno TCP ha been compared with a version of Westwood+ TCP implementing the NewReno feature [1]. Each simulation lasts 1000s during which the TCP sender always sends data. The *Short Retry Limit* and the *Long Retry Limit* parameters have been set equal to a common value that is referred as *Retry Limit*.

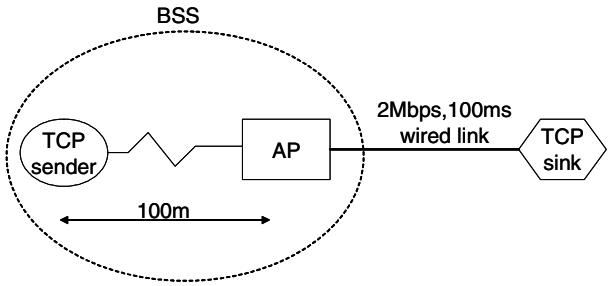

Figure 1. Simulated WLAN scenario.

Fig. 2 show the goodput achieved by Westwood+ and New Reno in the case of uniformly distributed random losses with loss rate p ranging from 0.001 to 1.

It shows that when the *Retry Limit* is larger than 6 both Westwood+ and NewReno achieve high link utilization, whereas, when the *Retry Limit* is less than 6 Westwood+ performs better than New Reno TCP. In particular, it can be noted that, for *Retry Limit* equal to 2, Westwood+ goodput starts to roll off for $p = 0.04$ whereas the goodput of New Reno for $p = 0.02$. Fig. 3 shows that the end-to-end retransmission ratio, computed as the number of retransmitted segments over the total number of transmitted segments, quickly drops to small values when the *Retry Limit* increases.

In order to study the behavior of Westwood+ and NewReno TCP in strenuous conditions, a bursty error model is now considered. In particular, we assume that every 1000 packets *N* are lost, so that the resulting dropping probability is $p = N/1000$, and p is varied from 0.001 to 0.1.

Fig. 4 show the goodput achieved by Westwood+ and New Reno in the case of bursty losses. It is worth noting that, when the *Retry Limit* is larger than 6 and $p < 0.01$, both the goodputs of NewReno and Westwood+ TCP satu-

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

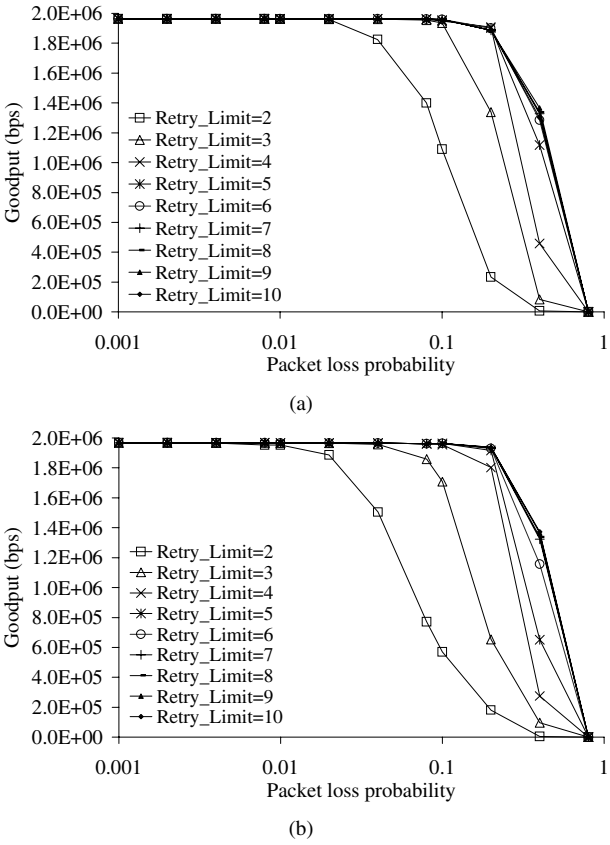

Figure 2. Goodputs of the TCP connection in the presence of uniformly distributed packet losses: (a) Westwood+; (b) New Reno.

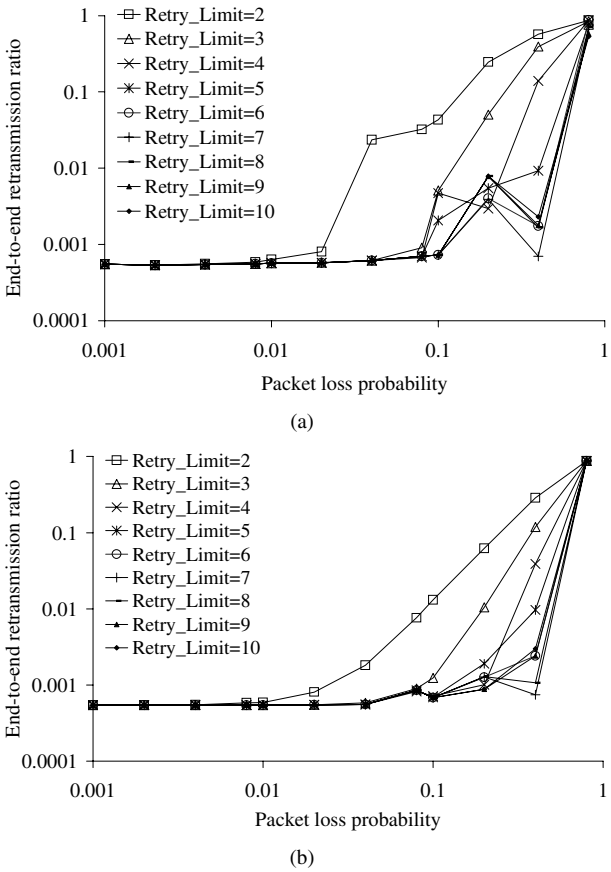

Figure 3. End-to-end retransmission ratio in the presence of uniformly distributed packet losses: (a) Westwood+; (b) New Reno.

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

rate to a maximum value. The difference with respect to the case of uniformly distributed losses is that even a small p heavily affects the goodput of NewReno.

In particular, Fig. 4 shows that Westwood+ already achieves the maximum goodput curve using a *Retry Limit* equal to 2. On the other hand New Reno requires a *Retry Limit* number at least equal to 6 in order to reach the curve with the maximum goodput.

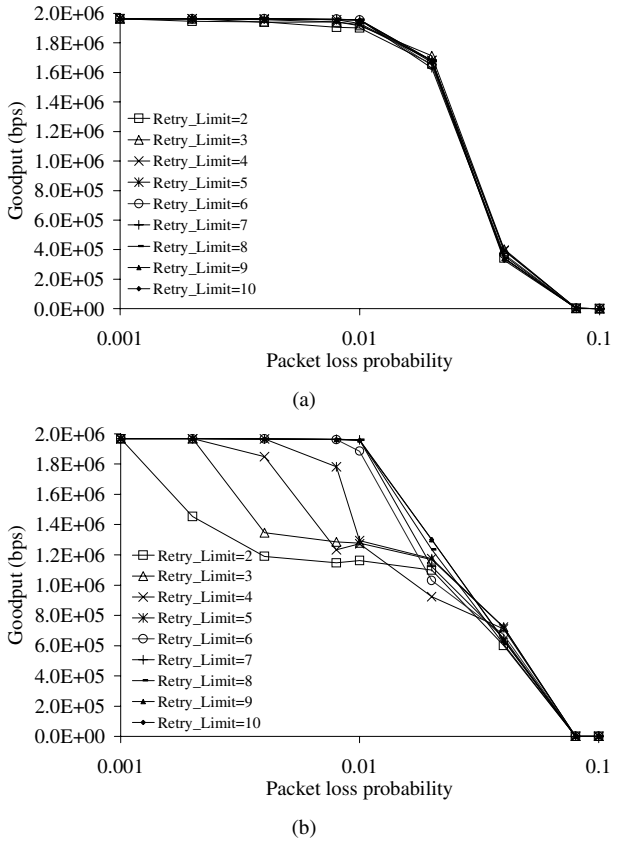

Figure 4. Goodputs of the TCP connection in the presence of bursty packet losses: (a) Westwood+; (b) New Reno.

Fig. 5 shows that for $p < 0.02$ the *Retry Limit* has an effect on the retransmission ratio. In the case of $p \geq 0.02$, even if we set *Retry Limit*=10, the end-to-end retransmission ratio drastically increases with p. This means that the WLAN error control algorithm is not able to fully compensate packet losses that are bursty.

In order to highlight how the behaviors of New Reno and Westwood+ TCP change when p varies from 0.01 to 0.02, Figs. 6 (a) and (b) show the $cwnd$ and $ssthresh$ of New Reno TCP when the *Retry Limit* is equal to 10 and bursty losses affect the channel with rate 0.01 and 0.02, respectively. Analogous results, obtained with Westwood+, are reported in Fig. 7.

Figs. 6 and 7 show that when the loss rate is 0.02, the $cwnd$ and $ssthresh$ of New Reno assume lower values than those of Westwood+ TCP. This behavior is due to the adaptive setting of Westwood+ TCP, which allows the $ssthresh$ to assume a value that is very close to the bandwidth delay product, thus ensuring high utilization also in the presence of bursty losses that are not masked by the local recovery mechanism.

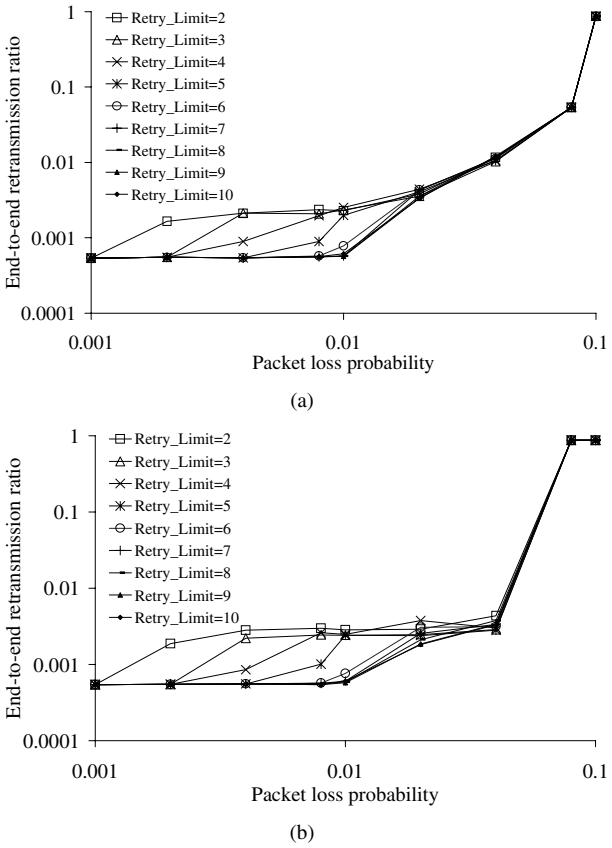

Figure 5. End-to-end retransmission ratio in the presence of bursty packet losses: (a) Westwood+; (b) New Reno.

To give a further insight into the behaviour of Westwood+ and New Reno TCP, Figs. 8 and 9 plot the RTTs experienced by New Reno and Westwood+ TCP during the time interval $[400s, 450s]$, for $p = 0.01$ and $p = 0.02$. They highlight that the RTT exhibits spikes due to data retransmissions at link layer. Delay spikes obtained with $p = 0.02$ are larger than those obtained with $p = 0.01$, due to the larger number of retransmissions at the link layer when p is greater.

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

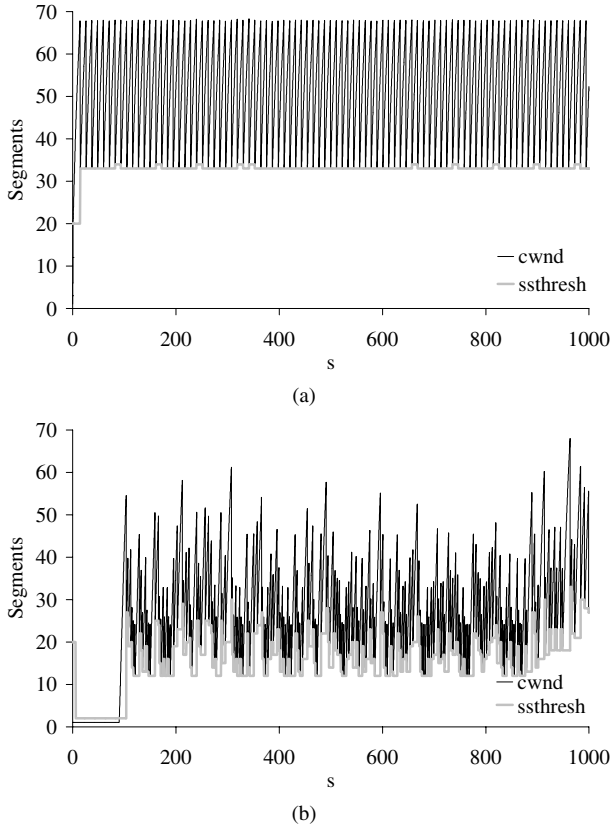

Figure 6. Cwnd and ssthresh of New Reno TCP in the presence of bursty packet losses: (a) p=0.01; (b) p=0.02.

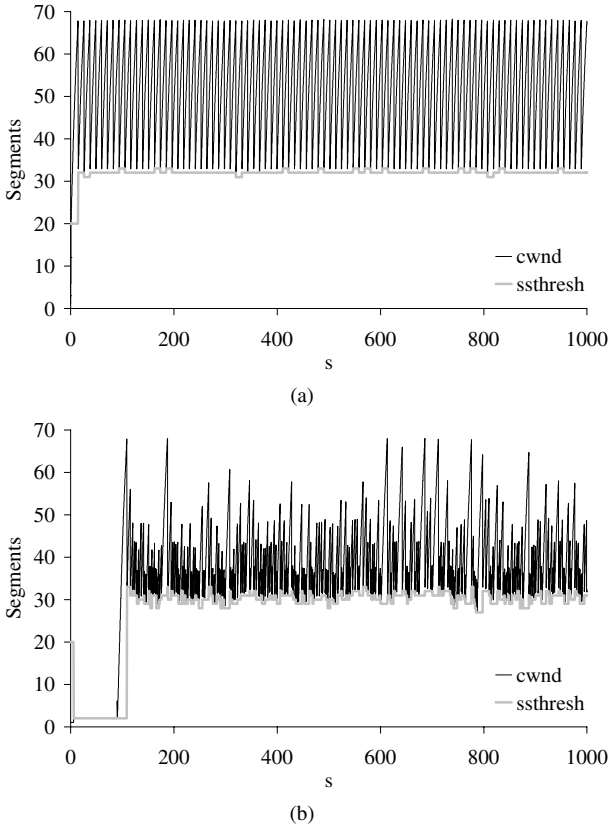

Figure 7. Cwnd and ssthresh of Westwood+ TCP in the presence of bursty packet losses: (a) p=0.01; (b) p=0.02.

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

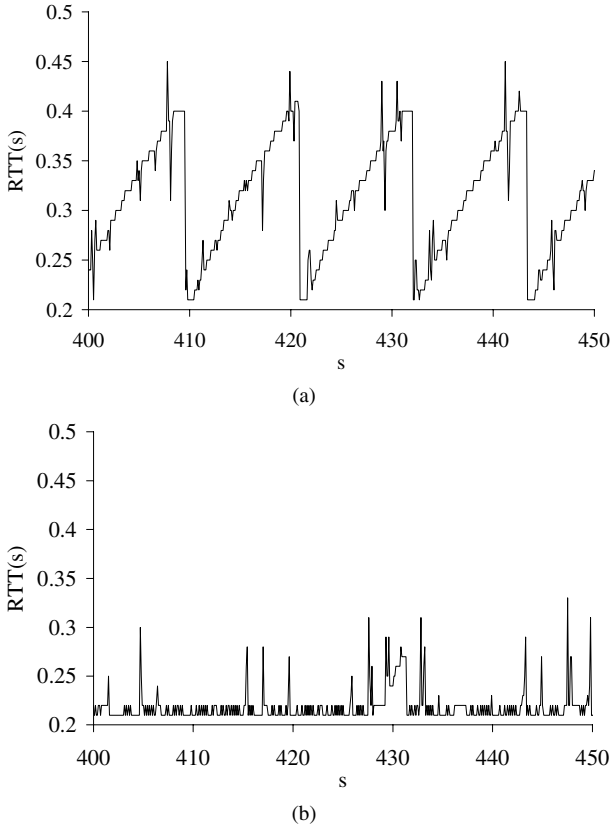

Figure 8. RTT measured by the New Reno TCP sender in the presence of bursty packet losses: (a) p=0.01; (b) p=0.02.

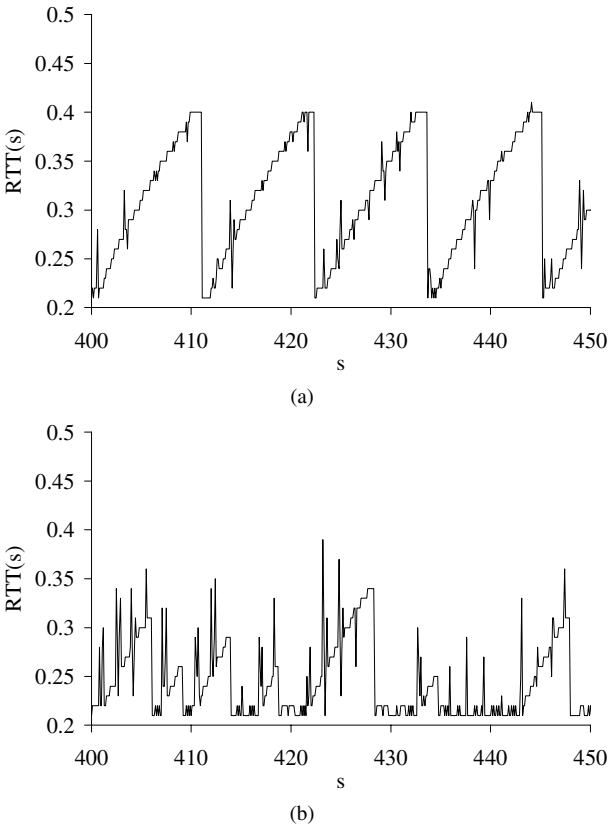

Figure 9. RTT measured by the Westwood+ TCP sender in the presence of bursty packet losses: (a) p=0.01; (b) p=0.02.

Fig. 10 shows the sequence numbers obtained using New Reno and Westwood+ in the case of $p = 0.02$.

Finally, in order to give a further insight into the behavior of Westwood+, Fig. 11 shows the bandwidth estimate of Westwood+ TCP for various loss rate. It is worth noticing that when the loss rate increases, the bandwidth estimate becomes more oscillating, due to the more oscillating behavior of *cwnd* and because a fraction of the bandwidth is used for packet retransmissions at the link layer.

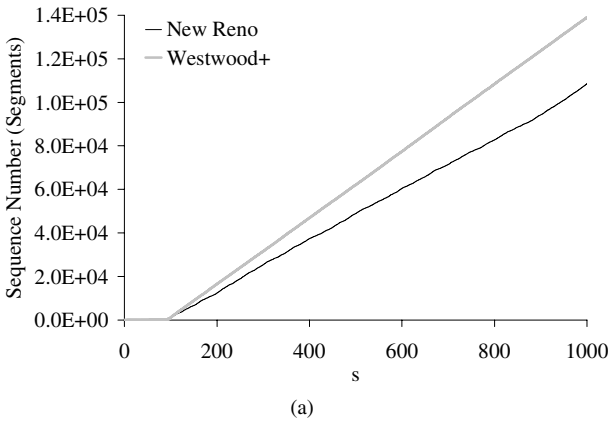

Figure 10. Sequence numbers of New Reno and Westwood+ when bursty losses affect the wireless channel with loss rate 0.02.

5 Conclusion

A comparison of Westwood+ and NewReno TCP over a noisy WLAN channel in the presence of local error control has been investigated. For that purpose *ns*-2 simulation results have been reported. They have shown that Westwood+ TCP improves the goodput *wrt* NewReno TCP when bursty losses affect the wireless channel. In particular Westwood+ TCP requires a smaller number of retransmissions at the link layer than New Reno TCP to achieve full utilization

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

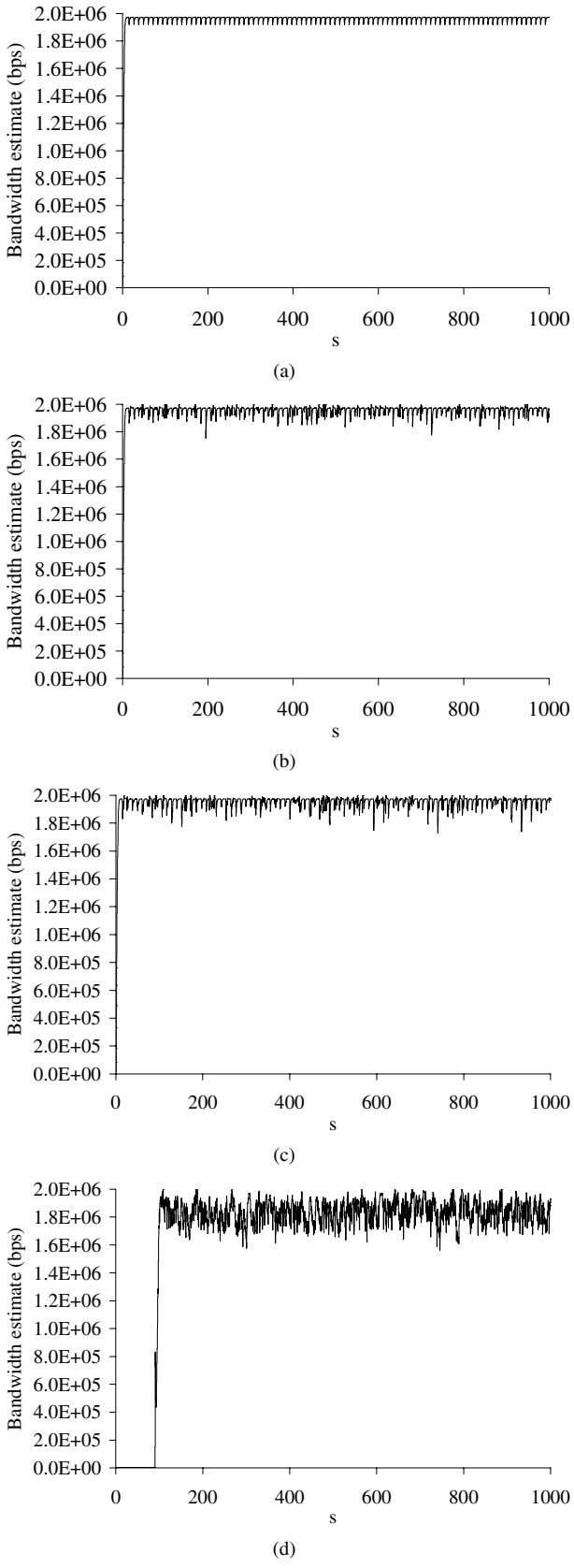

Figure 11. Bandwidth estimation of Westwood+ TCP: (a) p=0.001; (b) p=0.008; (c) p=0.01; (d) p=0.02.

of the wireless link.

References

[1] Westwood+ TCP with the New Reno feature, ns-2 modules. Available at http://www-ictserv.poliba.it/mascolo/tcp%20westwood/modules.htm.

[2] M. Allman, S. Floyd, and C. Partridge. Increasing initial TCP's initial window. RFC 3390, October 2002.

[3] M. Allman, V. Paxson, and W. R. Stevens. TCP congestion control. RFC 2581, April 1999.

[4] H. Balakrishnan, V. N. Padmanabhan, and S. S. andR. H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, December 1997.

[5] C. Barakat and E. Altman. Bandwidth tradeoff between TCP and link-level FEC. *Computer Networks*, 39(2):133–150, June 2002.

[6] H. M. Chaskar, T. V. Lakshman, and U. Madhow. TCP over wireless with link level error control: Analysis and design methodology. *IEEE/ACM Transactions on Networking*, 7(5):605–615, October 1999.

[7] D. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17(1):1–14, June 1989.

[8] D. Clark. The design philosophy of the DARPA Internet protocols. In *ACM Sigcomm '88*, pages 106–114, Stanford, CA, USA, August 1988.

[9] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai. IEEE 802.11 wireless local area networks. *IEEE Commun. Mag.*, pages 116–126, Sep. 1997.

[10] D. A. Eckhardt and P. Steenkiste. Improving Wireless LAN performance via Adaptive Local Error Control. In *International Conference on Network Protocols (ICNP)*, pages 327–338, Austin, Texas, October 1998.

[11] R. Ferorelli, L. A. Grieco, S. Mascolo, G. Piscitelli, and P. Camarda. Live internet measurements using Westwood+ TCP congestion control. In *IEEE Globecom 2002*, Taipei, Taiwan, November 2002.

[12] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, 1999.

[13] S. Floyd and T. Henderson. NewReno modification to TCP's fast recovery. RFC 2582, April 1999.

[14] L. A. Grieco and S. Mascolo. Westwood TCP and easy RED to improve fairness in high speed networks. In *IFIP/IEEE Seventh International Workshop on Protocols For High-Speed Networks*, pages 130–146, Berlin, Germany, April 2002.

[15] L. A. Grieco and S. Mascolo. End-to-end bandwidth estimation for congestion control in packet networks. In *Second International Workshop, QoS-IP 2003*, pages 645–658, Milano, Italy, February 2003.

[16] V. Jacobson. Congestion avoidance and control. In *ACM Sigcomm '88*, pages 314–329, Stanford, CA, USA, August 1988.

[17] R. Krishnan, M. Allman, C. Partridge, and J. P. G. Sterbenz. Explicit transport error notification (ETEN) for error prone wireless and satellite networks. Technical Report 8333, BBN Technologies, March 2002.

[18] R. LaMaire, A. Krishna, and P. Bhagwat. Wireless LANs amd mobile networking: Standards and future directions. *IEEE Commun. Mag.*, pages 86–94, Aug. 1996.

[19] S. Mascolo. Congestion control in high-speed communication networks using the smith principle. *Automatica, Special Issue on Control methods for communication networks*, 35:1921–1935, December 1999.

[20] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang. TCP westwood: End-to-end bandwidth estimation for efficient transport over wired and wireless networks. In *ACM Mobicom 2001*, Rome, Italy, July 2001.

[21] Ns-2. network simulator, 2002.

[22] N. Prasad and A. Prasad, editors. *WLAN Systems and Wireless IP*. Artech House universal personal communications series. Artech House, 2002.

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE