

# TCP-ELN: On the Protocol Aspects and Performance of Explicit Loss Notification for TCP over Wireless Networks

Gergő Buchholz<sup>1</sup>, Thomas Ziegler<sup>2</sup>, Tien Van Do<sup>1</sup>

<sup>1</sup>Department of Telecommunications, Budapest University of Technology and Economics  
Magyar tudósok körútja 2., Budapest, Hungary  
buchholz@hit.bme.hu, do@hit.bme.hu

<sup>2</sup>Telecommunications Research Center Vienna (FTW)  
Tech Gate Vienna, Donau-City-Straße 1/3. Stock A-1220 Wien, Austria  
ziegler@ftw.at

## Abstract

Recently many works have been invested to overcome the low performance of TCP in wireless environments. These proposals are hardly adaptable in real networks due to high resource demand, non scalability and difficulties in their implementation. In this paper we propose a new technique that besides of allowing TCP to improve its performance also takes applicability into account. Our solution is based on the idea of Explicit Loss Notification (ELN) to notify the sender about packet losses on the wireless channel. We evaluate the performance of TCP-ELN with different kind of traffic and error models. Simulation results show the ELN significantly improves TCP efficiency and fairness performance for FTP-like and web-like traffic.

## 1. Introduction

In the past years TCP/IP has been more and more deployed in wireless networks where bit errors on the air interface happen with high probability. However, standard TCP versions do not distinguish between losses due to congestion along a path and losses due to bit-error on the wireless channel and simply reduce the congestion window for both types of packet losses. The reduction of the congestion window without distinguishing losses due to bit error and congestion at the TCP layer makes TCP congestion control ineffective. Consider aggregate TCP flows over a network part having one wireless link with high bit error rate and no error recovery at the link layer. As TCP reduces its congestion window in response to packet losses caused by bit errors on the wireless channel, the throughput of TCP flows would be close to zero although the network may not be congested at all.

Several options exist to optimize TCP performance over wireless networks. A major part of today's operational wireless networks employ an ARQ mechanism at the link layer for IP based data services (e.g. 802.11, UMTS acknowledged mode). Link layer retransmission has been shown to be efficient and scalable in many scenarios. The number of retransmissions attempts before a link layer frame is considered lost, however, is usually rather small

in order to avoid high delay variation and buffering. Thus, even in the presence of link layer retransmission a residual packet loss probability exists from the perspective of the transport layer, as has been reported for instance in [1]. This paper reports an average residual packet loss probability of more than 6% for the case of WLAN in a standard industrial environment. This result shows that there exists a need for additional mechanisms at the TCP layer to improve performance in the case of critical radio conditions even in the presence of link layer ARQ. In case of no link layer ARQ and FEC (e.g. UMTS transparent mode), TCP is obviously unprotected against packet loss due to bit errors, making a clear case for protection at the transport layer.

Several proposals have been worked out for transport layer mechanisms to help TCP distinguish between losses due to bit error and losses due to congestion. The indirect TCP approach [2] splits a TCP connection between a fixed and a mobile host into two separate connections and hides TCP from lossy link by using a protocol optimized for lossy links. The Berkeley SNOOP protocol [3] caches packets at the base station and performs local retransmissions over the lossy link. Note that both approaches require storage of extensive per microflow state and even data packet retransmissions at the base station make them scale badly in terms of link bandwidth and number of flows. Bakshi et al. [4] investigated the performance of TCP over wireless

links varying the MTU size and proposes to send ICMP packets to inform the TCP source about the wireless link experiencing bit-errors.

Balan et al. [8] proposed the TCP header checksum option to make TCP detect whether a bit-error happened in the body of the TCP segment or in the header. In the first case, the congestion window is not reduced.

TCP Westwood [5] has the potential to improve the performance over lossy channels as it adapts the congestion window based on throughput variations rather than packet losses. TCP Westwood is a sender-only modification to TCP thus it scales and is easy to deploy. The mechanism of throughput based window adaptation, however, needs some initial time to become effective and TCP Westwood cannot avoid unnecessary window reductions in the presence of packet losses due to bit-error. Thus Westwood will not show significant improvements of TCP flows over lossy channels.

Explicit Loss Notification (ELN) [6] is a mechanism by which the reason for the loss of a packet can be communicated to the TCP sender. The base station monitors all TCP segments that arrive over the wireless link but does not cache any TCP segments since it does not perform any retransmission. Rather, it keeps track of the segments that have been lost over the wireless link and sets the ELN bit in the corresponding ACKs. This original ELN proposal implies the drawbacks of requiring a list of the lost sequence numbers per microflow at the base station which is impossible in case of IPSEC encryption. Moreover, this proposal does not take ACK loss into account and provides poor signaling information for the TCP sender.

In [7] we have redefined the ELN technique keeping only the fundamental idea of it, informing the TCP source explicitly on packet losses on the wireless channel. We proposed a Reno based TCP congestion control algorithm that can improve the performance of TCP significantly if the source is able to inform the sender about the number of the corrupted TCP segments. However the technique to gain loss information that makes the protocol work is only briefly described in that paper and implemented in the simulator in a course grained manner.

In this paper we give the details of gaining loss information from corrupted packets taking the different configuration cases of wireless environments into account. We also introduce a novel enchanted congestion algorithm that exploits all information from the loss notifications in order to maximize its performance.

The rest of the paper is organized as follows. The ELN based TCP congestion control algorithm is discussed in section 2. In section 3 we describe the details of our loss recovery technique. Section 4 and 5 specifies the simulation environment and shows the results. Finally section 6 concludes the paper.

## 2. The TCP-ELN proposal

In order to improve the performance of TCP by collecting loss information both the server and the client side of the TCP agent must be modified. Note that the approach for ELN taken in this paper does not necessarily require modification of network elements. Therefore, it can be implemented in an end-to-end manner. The TCP receiver must be able to receive loss information from lower layers and notify the sender, who needs to act properly in reaction to the reception of loss and congestion information.

### 2.1. Receiver side modifications

By using a loss recovery technique the TCP receiver may receive not only intact TCP segments but also loss notifications. When the TCP data receiver receives intact TCP segments it operates in compliance with standard TCP behavior i.e. an acknowledgement is generated. If a loss notification is received then it is stored. This loss information is sent to the TCP source embedded into the option field of the acknowledgement. In case of bursty bit errors several TCP segments can get damaged possibly producing a high amount of loss notifications before an ACK is generated. Thus these notifications *must be stored* at the receiver to avoid dropping of loss information.

No negative ACKs are used thus loss notifications can be sent to the source only piggybacked on ACKs. As a consequence the use of *delayed acknowledgements is feasible though not recommended* since it decreases the signaling speed.

The TCP receiver must take also ACK loss into account. Since losing an ACK causes the loss of loss notifications the notifications *must be sent redundantly*. Depending on the implementation it must be defined how many notifications can be stored in an ACK and how many times a notification is resent. In our implementation we have implemented a SACK like solution with sending loss notifications three times.

During the lifetime of a TCP flow timeouts can occur at the sender. After the timeout the sender starts sending packets regardless of the ongoing TCP ACKs. In this case all loss notifications must be cleared by the receiver since these losses occurred before the timeout and sending them to the source can cause inconsistent TCP behavior. Thus the sender must be aware of sending timeout notifications and the receiver must be aware of *handling these signals*. How timeout signals are generated by the source can be found in the next subsection.

### 2.2. Sender side modifications

The flow control algorithm of TCP-ELN is based upon TCP Newreno's standard. If no loss notification is received by the source TCP-ELN behaves exactly the same as Newreno. If the source receives loss notifications

in the ACKs from the receiver then it stores the notifications and resends the lost segments. Until receiving the 3<sup>rd</sup> duplicate ACK TCP-ELN operates according to Slow Start or Congestion Avoidance algorithms. The 3<sup>rd</sup> duplicate ACK can be a sign of congestion loss or loss due to bit-error. To decide which recovery state the TCP source should enter the stored loss notifications are examined. If a notification of the lost segment indicated by the duplicate ACKs was stored previously then TCP-ELN enters the wireless recovery state. If no loss notification is stored for the segment indicated by the triple duplicate ack event then TCP-ELN treats the 3 consecutive duplicate ACKs as a sign of congestion. Thus Fast Retransmit and Fast Recovery are invoked.

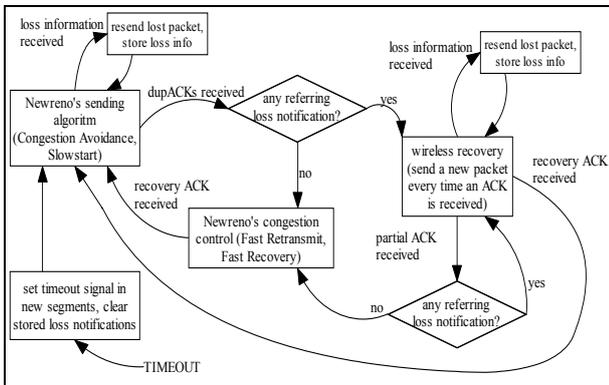


Figure 2-1 The operation of the TCP-ELN sender

In the wireless recovery state the source waits for a recovery ACK that acknowledges all segments that were lost on the wireless channel. During this recovery phase reported segments that are lost due to wireless bit-error are resent. Every time a duplicate ACK arrives the source increases the size of its congestion window with one segment. As a consequence the TCP data flow and self clocking is kept going, except if the effective window size is limited by the maximum value of the send window. If a partial ACK arrives at the source then the source decides whether it should stay in the wireless recovery state or enter the congestion recovery state. To make the decision the stored loss notifications are checked just like after receiving the 3<sup>rd</sup> duplicate ACK. If there is a referring loss notification then the source stays in the wireless recovery state otherwise it enters the congestion recovery state. If the recovery ACK is received then Slow Start or Congestion Avoidance, respectively, continues. On The operation of the TCP-ELN sender can be seen on Figure 2-1.

During the operation of the flow control algorithms timeouts can occur. In this case all flow control parameters are reset like in Newreno, the list of the loss notifications is cleared and a timeout signal is sent to the receiver. Since this timeout signal can be lost it must be sent in a redundant manner. The sender includes the number of

timeouts since the start of the flow in every TCP segment's header option field. Thus even if packets get lost the timeout signal will reach the receiver. It will be received when the first intact TCP segment arrives.

### 3. Loss information recovery techniques

To use the ELN proposal loss information must be collected from the TCP segments that are damaged on the wireless channel. Certain assumptions depending on the network configuration have to come true to use the recovery methods efficiently. To demonstrate the relationship between the ELN techniques and the different network setups the topology of Figure 3-1 is used. This is a widely deployed scenario that contains a wireless link between the client and the access point.

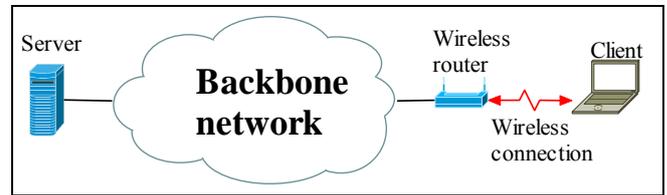


Figure 3-1 Typical wireless access scenario

In the applied protocol layer model TCP segments can be fragmented in 4 different manners depending on the IP and MAC layer settings as shown in Table 1. In all the 4 cases it is examined, how efficiently a technique can separate the causes of packet loss: congestion and radio channel errors.

Case	TCP packets are fragmented at the IP layer	IP packets are fragmented at the MAC layer at the wireless router
1	-	-
2	-	✓
3	✓	-
4	✓	✓

Table 1: The 4 different cases of fragmenting

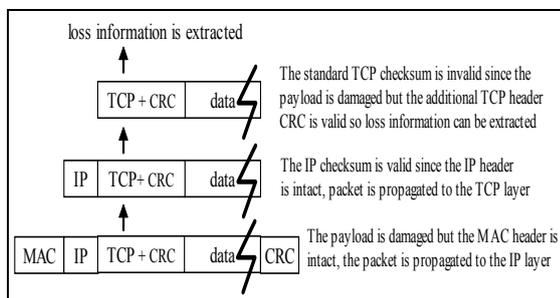
Three parameters the TCP sequence number, port number, and the source IP address of the lost segments must be collected by the wireless client in order to identify a single TCP flow. There are two possibilities to get this loss information, which are explained in the subsequent paragraphs.

#### 3.1. Method 1: Cross layer Communication

The first method is built upon the fact that MAC frames are not simply lost but received in a damaged manner by the destination node in the case of radio link error. Sensitive information can be found in the TCP/IP headers so extracting the intact headers from a damaged MAC

frame solves the identification of a single TCP flow. To check whether the TCP header is damaged or not an additional header checksum is needed since TCP header has no dedicated protection [8].

This checksum can be inserted in the TCP header's option field. When the modified MAC layer of the client receives a corrupted frame instead of dropping it, it propagates the payload of the frame to the upper layers. If all the headers belonging to the different protocol levels are undamaged then the TCP segment reaches the TCP layer where loss information is extracted and sent to the source of the data flow. As the payload is much longer than the headers, bit errors are likely to occur in the payload leaving the headers intact. So this recovery method can be very efficient to collect loss information



**Figure 3-2: Recovery of damaged packets**

Figure 3-2 shows how a damaged packet gets to the TCP layer.

The MAC modification at the receiver is very sensitive to packet fragmentation. In the different fragmentation cases the assumptions alter as follows.

1. *No fragmentation at the IP and MAC layer*

A TCP segment encapsulated in an IP packet and a MAC frame is received by the wireless client as shown on

Figure 3-2. If the MAC frame is damaged the IP packet is extracted despite of the invalid MAC checksum. Then the IP packet header checksum is inspected and if it is valid – the IP header is intact – the TCP segment is extracted. If the TCP checksum is invalid while the TCP header checksum is valid then loss information can be collected.

If the MAC protocol supports the MAC header verification it is easy to extract the damaged IP packet since the integrity of MAC level flow control information can be checked. However this is not crucial. IP packets may be extracted also from MAC frames with fixed header length or partial header protection. In this case MAC control information that is included in a damaged frame must be neglected. Congestion is well

separated from radio link errors since in case of congestion the drop of an IP packet means the drop of a whole TCP segment. So no explicit loss information can reach the receiver.

2. *No fragmentation at the IP layer, fragmentation at the MAC layer*

When the MAC frames are received by the client, the MAC layer tries to reassemble them. If all MAC frames are received and all MAC headers are intact then the reassembly is possible provided that the MAC header integrity can be checked. After the reassembly the recovery proceeds the same way as in the “*No fragmentation at the IP and MAC layer*” case. Recovery may work even if some of the MAC headers are damaged since only the TCP/IP headers are needed to collect loss information and these headers can be found in the first few MAC frames. If these parts are received error free then they can be partially compiled. The damaged IP packet can be propagated to the IP layer with an indication that the packet is not intact. The IP layer extends the packet to the right length according to the IP header's total length field and the recovery proceeds the same way as in the “*No fragmentation at the IP and MAC layer*” case. This method works even if there is no option to check whether the MAC header is intact or not.

Since there is no fragmentation at the IP level every TCP segment crosses the backbone embedded in a single IP packet. As a consequence losing a congested IP packet causes the loss of a whole TCP segment. Thus in case of congestion at the IP level no loss information reaches the receiver.

3. *Fragmentation at the IP layer, no fragmentation at the MAC layer*

Since a TCP segment crosses the network in more than one IP packet it is crucial to receive all IP packets – even in a damaged manner - by the client in order to be sure that no IP packet was lost due to congestion.

To collect every IP packet the 1<sup>st</sup> case method can be used. The MAC layer has to simply extract the damaged MAC frame's payload and the damaged IP packet. If all the IP packets with an intact header can be extracted then the TCP segment can be reassembled. The TCP layer after receiving the reassembled segment verifies the header checksum and collects the loss information if the checksum is valid.

In case of losing a congested IP packet the TCP segment cannot be reassembled so no loss information can be extracted.

#### 4. Fragmentation at the IP and MAC layer

In this case both the 2<sup>nd</sup> and the 3<sup>rd</sup> point's restriction prevail. All IP packets have to be received by the client in order to gain loss information.

The recovery integrates the 2<sup>nd</sup> and 3<sup>rd</sup> points' recovery methods. The MAC frames that carry the TCP and IP headers have to be received intact so the receiver's MAC layer can partially reassemble the IP packets. After extracting the IP headers from the damaged MAC frames the MAC layer propagates them to the IP layer indicating that the packets are damaged. The IP layer fills the gaps in the damaged IP packets according to the IP header's total length field. If the headers of all IP packets constituting one TCP segment reach the IP layer intact then the TCP segment can be partially reassembled. The TCP layer receives this damaged segment with invalid payload checksum but valid header checksum so it extracts the loss information.

The advantage of the MAC modification method is that only the destination peer needs to be modified in order to collect loss information. All other network nodes remain untouched making this method absolutely end-to-end compliant. Since the MAC layer is modified only at the wireless client only the last wireless link can be monitored and loss information can be extracted only from the download flows.

If modifying the MAC layer or recovering damaged packets at the MAC level is impossible, the ELN proposal may remain applicable by using the IP fragmentation method.

### 3.2. Method 2: IP Fragmentation

This method solves the loss information recovery without modifying the MAC layer; it uses a specialized IP fragmentation technique.

The fundamental idea of the IP fragmentation method is that the probability of losing small packets on the wireless channel is smaller than the probability of losing large packets. So if every TCP segment reaches the access point encapsulated in a single IP packet then fragmenting these IP packets at the access point into at least 2 packets where the first small part contains only the TCP header and the 2<sup>nd</sup> large part contains the payload can be used to collect loss information efficiently.

The 1<sup>st</sup> and 2<sup>nd</sup> case of network configuration (see section 3.1) doesn't affect this recovery method. The wireless router sends the IP packets embedded into MAC frames through the wireless channel. Since the MAC layer of the receiver is untouched, the damaged frames get dropped as usual. The IP layer tries to reassemble the received IP packets, but if any of them is lost then the original TCP segment cannot be reassembled. This means that at least

one IP fragment was lost due to wireless bit error. But if the first IP packet with the TCP header included has arrived then the receiver's IP layer can partially reassemble the damaged TCP segment neglecting the missing parts. These missing IP packets contain the payload that is not used for loss information recovery. The damaged TCP segment is propagated then to the TCP layer where loss information can be extracted after verifying the TCP header checksum. The operation of this technique can be seen on Figure 3-3.

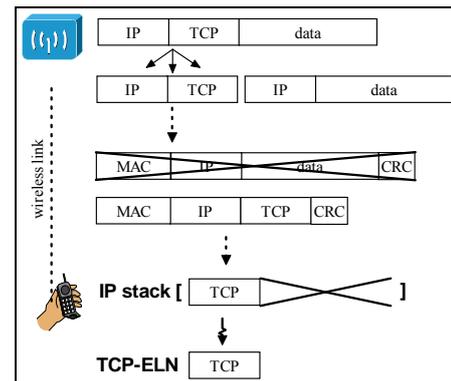


Figure 3-3: The IP fragmentation technique

If an entire IP packet was lost in the backbone network then the whole TCP segment would be lost so the receiver cannot gain loss information. As a consequence the detection of loss reasons are well separated.

In the 3<sup>rd</sup> and 4<sup>th</sup> case the network configuration is in conflict with the technique's assumptions since the TCP segment is fragmented in the network. This makes the IP fragmentation proposal unsuitable to use.

The drawback of the IP fragmentation method is that it violates the end-to-end semantics since it needs the wireless router to be slightly modified. However using this technique can be very efficient since it is *scalable* in contrast to other non-end-to-end proposals like Snoop, the needed modifications at IP layer are in accordance with the *normal IP level operation* (fragmenting) and it is *independent* of the MAC layer. This is a trade-off between the possibility of MAC level modifications and the end-to-end semantics.

## 4. Simulation environment

To measure the performance of TCP-ELN a wide variety of simulations were run.

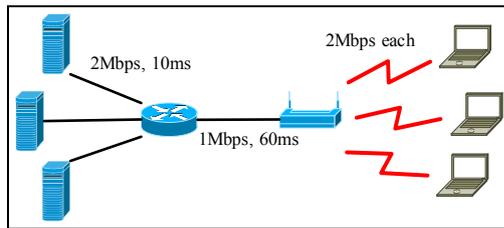


Figure 4-1 Simulation topology

The simulation topology that is shown on Figure 4-1 consists of 3 servers, 3 wireless clients, a router and an access point. The bandwidth of the links between the servers and the router are 2Mbps with 10ms of delay. The router is connected to the access point with a 1Mbps bandwidth link having a latency of 60ms. The access point offers 2Mbps for each client. All links are full-duplex. No specific MAC layer is applied on the wireless links, since the proposed techniques are general and do not depend on a specific wireless technology as strongly as on the topology. Due to the bandwidth setting congestion can occur only at the router in the wired part of the network.

Two different types of traffic: FTP and web traffic were used to examine the behavior of TCP-ELN. While FTP traffic is used to investigate the steady state behavior of the protocol web traffic is used to examine the dynamic behavior. In the first case 1 FTP session is started between each server-client pair. All servers used the same TCP version - TCP-ELN or TCP-Newreno. The throughput of the two TCP versions were measured and compared.

In the web traffic case each client simulates a web browsing user that connects to one of the servers. The parameters of the web traffic are in accordance with the SURGE model [9] that is based upon real network traces and widely used to generate realistic workloads. The download speed (page size/download time) of the web pages with the different TCP versions are measured and compared.

To simulate wireless channel errors two error models were used. Uniform and Markov error models are two different approaches of modeling the lossy link behavior. Both of them are used to simulate packet loss at the TCP level. The mean value of the uniform distribution is varied between 0 and 0.2.

The Markov model presented [10] is applied in the simulation study, which integrates channel fading and radio link parameters to take into account the characteristics of wireless channels.

Model number	User speed	Average error rate	Average error burst length
1	Pedestrian	0.001	1.4913
2		0.01	4.0701
3		0.1	13.6708
4	Middle	0.001	1.0083
5		0.01	1.0838
6	Vehicular	0.1	1.8629
7		0.001	1.0024

8		0.01	1.012
9		0.1	1.1317

Table 2 The parameters of the Markov error model

The Markov error model simulates how the radio links with different drop probabilities are experienced by users with different speed (Table 2).

Since the efficiency of the loss recovery algorithms depends on how TCP segments are damaged it is assumed that the probability of recovering a damaged segment is equal to the ratio of the payload size and the total packet size. In real environments this ratio is about 95% so in the simulations 5% of the damaged TCP segments cannot be recovered.

To investigate the interaction between TCP-ELN and TCP Newreno fairness measurements are used. On the same topology with 5 server-client pairs 5 FTP sessions are started. Three sessions use TCP-Newreno without experiencing radio channel errors. Additionally, two Newreno or ELN based sessions are started in order to compute the fairness index [11] of the throughput measured with the Markov and the uniform error model.

## 5. Performance results

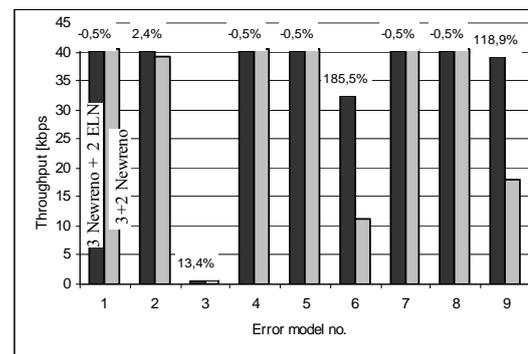
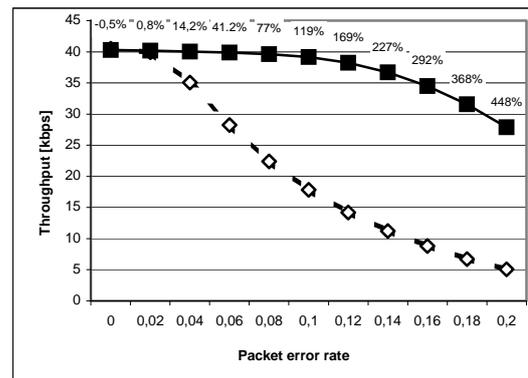


Figure 5-1 The relative improvement and the throughput over FTP traffic with the Markov and uniform error model. Inspecting the throughput over FTP traffic TCP-ELN shows significant improvements in most cases, reaching as high as 400% depending on the particular error rate. As

compared to Newreno, TCP-ELN is much better in the presence of error on the wireless link. In the error free case, i.e. without any packet loss on the wireless link, the performance of TCP-ELN is insignificantly lower (0.5%) than the performance of Newreno. This is due to the overhead, the larger header size of TCP-ELN segments. However this phenomenon also appears when using TCP-SACK.

Figure 5-1 shows that increasing the error rate, the proposed flow control algorithm makes TCP-ELN throughput decline more gently than the rapid decrease observed for Newreno. For instance, while the throughput of TCP-ELN drops to 90% at the error rate of 14% Newreno's throughput drops down to 90% at the error rate of 4%. The results with the Markov error model show that in the high loss scenarios (0.1 mean PER) model 3, 6 and 9 the improvement depending on the burstiness can reach 185%. In the low loss scenarios (0.001, 0.01 mean PER) model 1, 4, 5, 7 and 8 an insignificant (0.5%) decrease can be observed, in model 2 the improvement is 2%.

Comparing the relative improvement measured over the uniform error with the ones measured for the Markov error model an interesting correlation can be recognized. As the speed parameter of the Markov error model increases the average length of error bursts decreases. Thus for the considered scenarios at higher speeds – as packet loss becomes sporadic – the Markov error model converges to the uniform error model.

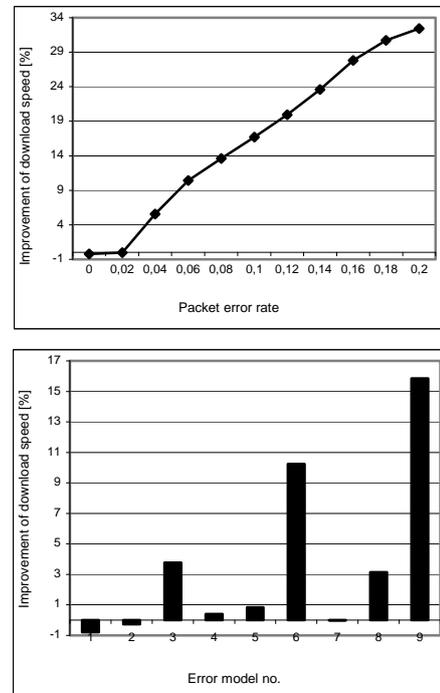
In case of web traffic, less dramatic, but still significant improvements can be observed as shown in Figure 5-2. The improvement can reach 33% with the uniform error model at high error rates. When no packet loss occurs on the wireless channel then the performance decreases 0.1% due to the larger header size. Up to an error rate of 0.02 the performance of Newreno is almost identical to ELN performance. In case of the Markov model the improvement remains moderate 4% to 16% for the high error models (3, 6 and 9) and lower values for the low error models.

Comparing the web and FTP traffic results it can be seen that in case of FTP traffic the improvement is significantly higher. The reason for this is that in case of the used HTTP 1.1-like model for web traffic the majority of flows only has a few packets to send. When the TCP flows are very short Newreno's congestion window remains small. Thus ELN's flow control algorithm cannot improve the performance with avoiding window halving as much as in the case of longer flows and larger congestion windows.

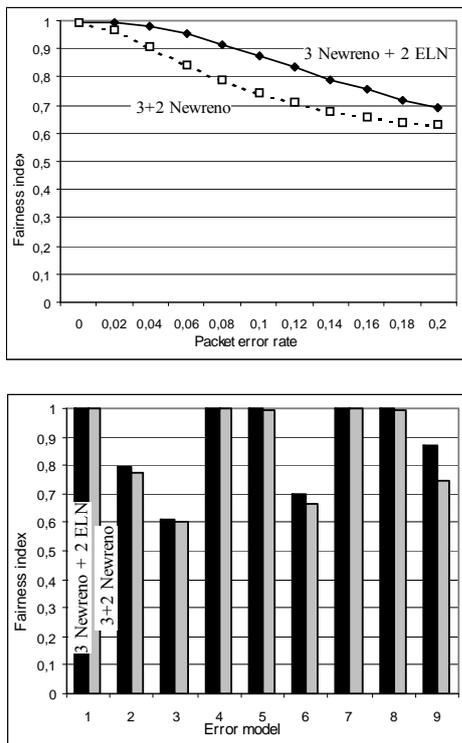
Thus with flows having peer-to-peer traffic, it can be expected that the performance improvement of ELN compared to NewReno is between the FTP case as an upper bound, and the HTTP 1.1 case as a lower bound.

The fairness index is higher in all the cases when using TCP-ELN as shown in Figure 5-3. At typical error rates that can be experienced by wireless clients the fairness

index can be more than 10% higher with TCP-ELN. This is in accordance with the aim of ELN that is to upgrade the performance of TCP on lossy wireless channels to the level where TCP would operate in error free case.



**Figure 5-2 The relative improvement of the download speed over web traffic with the uniform and the Markov error model**



**Figure 5-3 The fairness index of 3 Newreno with 2 TCP-ELN and 3 Newreno with 2 additional Newreno TCP flows over the same topology**

## 6. Conclusions

In this paper we proposed a new technique to improve the performance of TCP over wireless channels. We based our protocol on the idea of Explicit Loss Notification (ELN). ELN allows TCP to differentiate between packet losses due to congestion and packet losses due to radio channel error. Thus unnecessary window reduction can be reduced substantially in the presence of radio channel error. The resulting performance of the modified TCP protocol is greatly enhanced compared to TCP-Newreno.

We developed two solutions to gain loss information in order to use ELN. The first solution is based on modifying the MAC layer; the second solution uses a special IP fragmentation technique. These methods can be used in a wide range of network environments depending on the topology and the setup of cross layer communication.

We developed a new TCP variant, TCP-ELN that integrates the loss recovery methods to improve its performance. The proposed receiver and sender side algorithms and processes were discussed in detail. Security issues concerning TCP-ELN were also considered.

Our simulation experiments have shown that TCP-ELN can improve the performance of TCP over the radio channel substantially in a wide range of environments. The protocol was tested over random and bursty erroneous

radio links with two different types of traffic, static FTP bulk load and dynamic web traffic. Results proved that TCP-ELN produces better performance and fairness than TCP-Newreno in all the cases. The performance improvement can reach up to 400% when using FTP traffic and 30% when using an HTTP 1.1-like model for web traffic with high error probabilities.

## 7. Reference

- [1] A. Willig, M. Kubisch, C. Hoene and A. Wolisz, Measurements of a Wireless Link in an Industrial Environment Using an IEEE 802.11-Compliant Physical Layer. IEEE Transactions on industrial electronics, Vol. 49, No. 6, December 2002
- [2] A. Bakre and B. R. Badrinath, I-TCP: Indirect TCP for Mobile Hosts. In Proc. 15<sup>th</sup> International Conf. on Distributed Computing Systems (ICDCS), May 1995
- [3] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R.H. Katz, A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, IEEE/ACM Trans. on Networking, 5(6), December 1997.
- [4] B. Bakshi, P. Krishna, N. H. Vaidya, D. K. Pradhan, Improving Performance of TCP over Wireless Networks, 17th Int. Conf. Distributed Computing Systems, Baltimore, May 1997.
- [5] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links, In Proc. ACM Mobicom 2001, pp 287-297, Rome, Italy, July 16-21 2001
- [6] Hari Balakrishnan and Randy Katz, Explicit Loss Notification and Wireless Web Performance, Proc. IEEE GLOBECOM Global Internet Conf., Sydney, Australia, November 1998
- [7] G. Buchholz, A. Grieser, T. Ziegler, T. Van Do, Explicit Loss Notification to Improve TCP Performance over Wireless Networks, 6th IEEE High-Speed Networks and Multimedia Communications (HSNMC), 2003
- [8] R. K. Balan et. al, TCP Hack: TCP Header Checksum Option to Improve Performance over Lossy Links, IEEE Infocom 2001
- [9] P. Barford, M.E. Crovella, Generating Representative Web Workloads for Network and Server Performance Evaluation, in Proc. of Performance '98/ACM SIGMETRICS'98, pp. 151-160, Madison WI.
- [10] A. Chockalingam, M. Zorzi, Rames R. Rao, Performance of TCP on Wireless Fading Links with Memory, ICC 1998
- [11] R. Jain and D. Chiu and W. Haweu, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. Research Report, TR-301, September, 1984.