Improving ECN-based TCP Performance over Wireless Networks Using A Homogeneous Implementation of EWLN

Oyunchimeg Shagdar
ATR Adaptive Communications Research Laboratories
2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
Email: x-oyunaa@atr.co.jp

Mahdad N. Shirazi
Communications Research Laboratory
2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
Email: mahdad@crl.co.jp

Bing Zhang
ATR Adaptive Communications Research Laboratories
2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
Email: zhang@atr.co.jp

Abstract-**In this paper we present an EWLN for an ECN enabled wired-and-wireless network. ECN is proposed to improve TCP performance by marking instead of dropping packets to indicate an incipient congestion in the wired network. ECN mechanism has many benefits and performs especially well in low-bandwidth delay sensitive TCP connections. However, it is not efficient enough in wireless networks. Since ECN-based TCP has been proven to work very well in traditional wired networks, we propose implementing an EWLN scheme that improves performance of ECN-based TCP over wired-and-wireless networks. The scheme deploys the information from the MAC layer and takes into account the interact with the error recovery mechanism at the link layer. In addition, the scheme does not require different implementations for mobile terminals and base stations and hence can support efficient data transmission over wired-and-wireless networks.**

Keywords: wireless networks, congestion, EWLN, ECN, TCP-Reno, TCP-SACK, DPSK, SNR

I. INTRODUCTION

Over past years, the Transmission Control Protocol (TCP) has been used as the main transport layer protocol to provide a reliable transmission service to a variety of internet applications such as WWW, file transfer, electronic mail, and etc. In current TCP/IP networks, TCP relies on packet drops as the indication of congestion. Future routers, however, are likely to have developed mechanisms for detection of incipient congestion. For networks with mechanisms for detection of incipient congestion, the use of ECN [1] (Explicit Congestion Notification) mechanisms for the notification of congestion to the end nodes prevents unnecessary packet drops and packet retransmissions that can result in noticeable and unnecessary delays for the users. ECN-based TCP, hence, would possibly become the principle protocol upon which most of applications runs in near future.

On the other hand, the increasing activity in recent years in wireless network strongly indicates that wireless links will play an important role in future networks. Communication over wireless links is characterized by limited bandwidth, high bit-error rates, and the change of wireless signal strength due to the mobile terminals.

Although ECN-based TCP works very well in traditional wired networks, it is not robust enough for wired-and-wireless environments. The reason for this is that ECN-based TCP is not able to completely distinguish congestion from wireless link error and it treats an actual packet loss as an indication of congestion, and invokes the congestion control as the regular TCP. To improve the performance of ECN-based TCP in wireless networks, the key point is to completely distinguish packets losses due to wireless link errors from those due to congestion. Explicit Wireless Loss Notification (EWLN) [2] is a method that is proposed to explicitly inform wireless losses to the TCP sender.

EWLN informs the sender that a loss has occurred for a reason unrelated to network congestion. Accordingly, the sender's congestion control mechanism can be decoupled from the retransmission mechanism and set to react only to congestion related losses. EWLN has been proposed for two common scenarios, where information is accessed either from a fixed terminal through a wired-to-wireless connection or from a mobile station through a wireless-to-wired connection. In the first case, if the receiver knows that the loss of a packet was not due to congestion, it sets the EWLN bit in the TCP header and propagates it to the source. In the latter case, a proposed version of the snoop scheme uses EWLN at the base station. The snoop agent running at the base station monitors all TCP segments that arrive over the wireless link and keeps track of holes in the sequence space, where holes correspond to packets that have been lost over the wireless link [3]. When ACKs (acknowledgements) arrive from the receiver, the agent at the base station consults its list of holes and sets the EWLN bit on the ACK if it corresponds to a packet in the list. It also cleans up all the holes with sequence numbers smaller than the current ACK. However, it does not cache any TCP segments because it does not perform any retransmission.

0-7803-7661-7/03/$17.00©2003 IEEE

There are shortcomings associated with the above approaches to deploy EWLN. Among others, 1) the treatment of wired-to-wireless and wireless-to-wired cases are heterogeneous, which results in a need for different implementations at receivers and base stations, 2) the computational requirements on base stations imposed by the agent's process is high, 3) these schemes do not fully consider the interact with link-layer error recovery schemes, which are essential for handling high bit-error rates in wireless links, 4) the agent-based scheme does not fit naturally into the layered structure of network protocols.

In this paper, we propose a simple and homogeneous implementation of EWLN that can be deployed with a link-layer error recovery scheme while also fitting seamlessly into the layered structure of network protocols. Furthermore, our EWLN implementation can be used at both mobile terminals and base stations to support efficient ECN-based TCP data transmission for both wired-to-wireless and wireless-to-wired connections.

II. EXPLICIT CONGESTION NOTIFICATION

The Explicit Congestion Notification (ECN) scheme provides a mechanism for routers to send a direct indication of congestion to the source. ECN makes use of active queue management techniques such as RED (Random Early Detection) [4]. According to ECN RFC2481, ECN requires two bits in the IP header and two new flags in the TCP header. The ECN-Capable Transport (ECT) bit is set by the sender to indicate that the end-points of transport protocol are ECN capable. Using RED algorithm, based on an average queue length exceeding a threshold, the Congestion Experienced (CE) bit is set by router as an indication of congestion. When CE data packet is received, the ECN-Echo flag is set by the data receiver to inform the data sender of congestion. To enable the data receiver to determine when to stop setting the ECN-Echo flag, the Congestion Window Reduced (CWR) flag is assigned.

A. The RED algorithm

The RED gateway calculates the average queue size, and compares it to two thresholds, a minimum threshold and a maximum threshold. When the average queue size is less the minimum threshold, no packets are marked. When the average queue size is greater than the maximum threshold, every arriving packet is marked. When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked with a certain probability.

B. Role of Router

In order to detect incipient congestion, a router uses RED queue management algorithm to probabilistically mark packets and signifies the sources about incipient congestion. The router set a CE bit to indicate congestion to the end nodes. Since a router detects and informs congestion before the queue overflows, the main advantage of ECN mechanism is to avoid unnecessary packet drops, and therefore avoiding unnecessary packet delay.

However, in case of a severe congestion, when the router's queue is full, then the router has no choice but to drop packets when a new packet arrives. In this case, TCP-ECN will not be able to completely distinguish congestion from wireless link error.

C. The TCP Receiver

When a CE data packet is received at the destination end-points, the TCP data receiver sets the ECN-Echo flag in the header of the ACK packet. After the TCP receiver sends an ECN-Echo ACK packet, it continues to set the ECN-Echo flag in ACK packets until a CWR packet arrives.

D. The TCP Sender

If the sender receives an ECN-Echo ACK packet, then the sender knows that congestion was encountered in the network. The TCP sender responds to incoming packets that have the ECN-Echo flag set by reducing the congestion window and set CWR flag in the TCP header of the first data packet sent after the window reduction.

III. EWLN IMPLEMENTATION USING MAC LAYER INFORMATION

The Explicit Wireless Loss Notification (EWLN) scheme provides a general concept by which the sender can be informed that a packet loss has occurred due to reasons unrelated to network congestion. If the TCP receiver knows that the packet loss was not due to congestion, it sets the EWLN bit in the TCP header and propagates it to the TCP sender. However, a way for the TCP receiver to learn the reason for packet loss has not been established. Consequently, there is no practical implementation method of EWLN. In this section, we discuss the implementation of EWLN based on MAC layer information. This approach can be used at both mobile terminals and base stations, associate with minor modifications in the sender and receiver of the TCP protocol.

A. Setting the EWLN bit within MAC protocol

In a wireless LAN MAC protocol, such as IEEE802.11.b, a Cyclic Redundancy Check (CRC) is used to determine if data packet was received correctly. If a data packet is transmitted incorrectly, a local retry at link-level continues for each failing packet until the transmission is successful, or until the relevant retry limit is reached. For those corrupted packets detected by CRC, which have been retransmitted until the retry limit, will be passed to the transport layer as an EWLN message at the receiver instead of discarding them as in IEEE802.11.b. Because most of time, packet corruption occurs in the payload part of a packet, we here assume that the sequence number in the TCP header is available when the packet is corrupted due to bit error. The flow chart of setting EWLN bit within the MAC protocol is shown in Fig. 1.

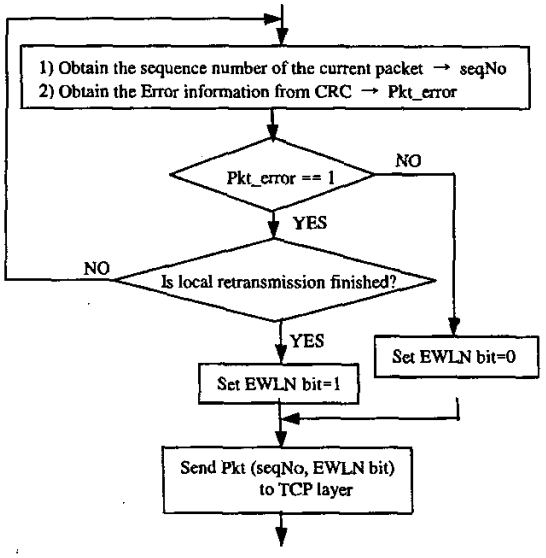

Fig.1. Flow chart of setting EWLN bit within MAC protocol

First, it is verified whether the local retransmission retry is finished or not. Upon the verification, the EWLN bit is set based on the error information provided by CRC. If a packet is erroneous, its payload is dropped, the EWLN bit is set to 1 in its TCP header, and it is handed to the transport-layer at a mobile terminal or routed to the next node at a base station. To verify that the local retransmission retry is finished, a buffering technique is used to keep the previously arrived packet. The sequence number of the current packet is compared with that of the one in buffer. If the sequence number of the current packet and the last packet are different, it means that the local retransmission has finished.

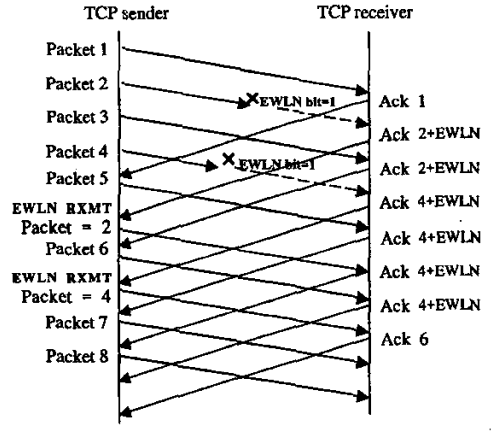

Fig. 2. Transmission of data and acknowledgement packets in EWLN-Reno protocol when two packet losses occur over a wireless link in a single transmission window.

B. The TCP Receiver

When the TCP receiver receives a packet with the EWLN bit, it generates ACK with the EWLN bit set in response to packet corruption due to bit-error over a wireless link. This ACK also contains the sequence number of the corrupted packet, thus allowing the sender to retransmit it immediately. Once the EWLN bit is set on ACK, it will continue to be set by duplicate acknowledgements until the retransmission of the corrupted packet is finished or a new corrupted packet with the EWLN bit set arrives.

Figure 2 shows an example of data transmission and acknowledgement using the EWLN bit, when two packet losses occur over the wireless link in a single transmission window.

C. The TCP Sender

The EWLN bit in the TCP header, which is set at the MAC-level and passed on to the TCP layer, is finally sent to the sender with the ACK. Upon receiving the EWLN message with the ACK, the sender performs retransmission without invoking the associated congestion control procedures. Because EWLN explicitly notifies the sender that packets are lost due to errors on the wireless link, the sender then has no need to wait until three DUPACKS arrive and can retransmit the packet upon receiving the first ACK with EWLN bit set. To avoid transmission duplication, retransmission is carried out only when the first ACK arrives with EWLN indicating a corrupted packet. When the EWLN bit is not set, the normal TCP operation is performed with fast retransmission and fast recovery for the congestion packet loss (see Fig. 3).

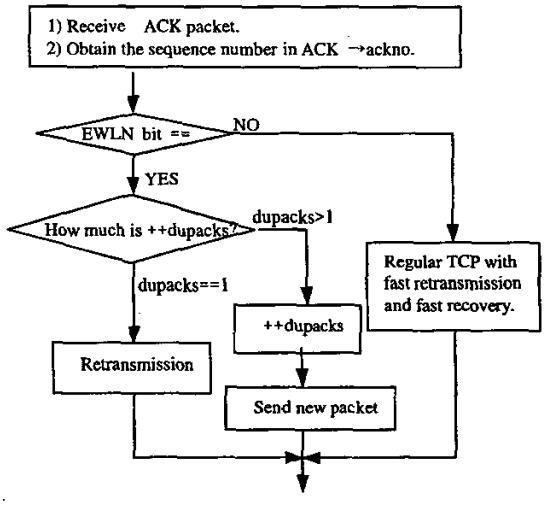

Fig. 3. Flow chart of retransmission strategy based on EWLN message.

IV. SIMULATION RESULTS

We performed the simulations with ns-2 network simulator of Berkeley [5]. Extensions include an implementation of an 802.11.b Mac layer and a radio propagation model. In addition, the average probability of error for DPSK (Differential Phase Shift Keying) is included as an error model to measure the performance in a near real-world situation. The average probability of error for DPSK (Differential Phase Shift Keying) is given by

$$P_{e,DPSK} = \frac{1}{2}\exp(-10^{\frac{SNR}{10}}), \qquad (1)$$

which is function of SNR [6].

A. *Behavior of EWLN-ECN over a simple wireless-to-wired networks*

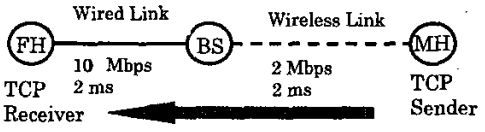

Fig. 4. Simulation topology 1.

Figure 4 depicts the network topology used in the experiment. The link between the TCP sender and the base station is wireless, whereas the link between the base station and TCP receiver is wired. In our simulation, the maximum bandwidth of the wireless link was 2 Mbits/s. The TCP data packets contain 1 Kbytes, while the TCP ACK contain 40 bytes. Maximum congestion window size for the TCP sender is set to 20Kbytes.

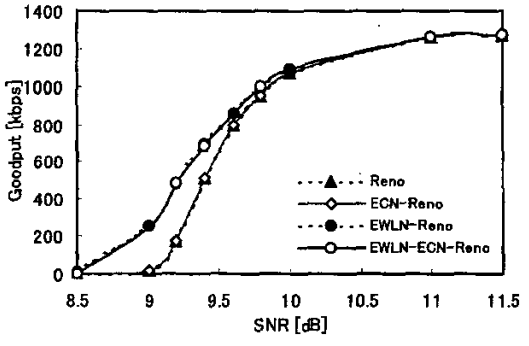

Fig. 5. Performance improvement of TCP-Reno in response to SNR over wired-to-wireless network

Since TCP-Reno is a standard version for TCP implementation, we use it as the standard basis for performance comparison. We compared performances between original TCP-Reno, ECN-based TCP-Reno (ECN-Reno), EWLN-capable TCP-Reno (EWLN-Reno) and EWLN-capable ECN-based TCP-Reno (EWLN-ECN-Reno). When a packet is dropped on the wireless link based on the DPSK error model, in EWLN-Reno and EWLN-ECN-Reno connections, the ACK is marked with an EWLN bit to indicate that a non-congestion related loss has occurred.

Figure 5 compares the throughput of the connection from the Mobile Host (MH) to Fixed Host (FH) using the Reno, ECN-Reno, EWLN-Reno and EWLN-ECN-Reno. The simulation time is 80 seconds. The final result is an average of 100 simulations runs were for each approach. It is clear from the figure that EWLN mechanism significantly improves the throughput when the range of SNR is [8.5 dB, 11.5 dB]. But there is almost no difference in throughputs of EWLN-Reno and EWLN-ECN-Reno. It can be explained that since in topology 1, packet losses are only caused by transmission error but not congestion, ECN bits almost will not be set, and therefore it leads to the same throughput with the throughput of EWLN-Reno.

Furthermore, we have also used the NS implementation of ECN, EWLN and EWLN-ECN mechanisms to TCP-SACK option. TCP-SACK [7] (Selective Acknowledgements) was proposed to alleviate TCP's inefficiency in handling multiple drops in a single window of data. In contrast with the standard cumulative TCP ACKs, a SACK can convey information to the sender about up to three noncontiguous blocs of data that have been received lately and successfully by receiver.

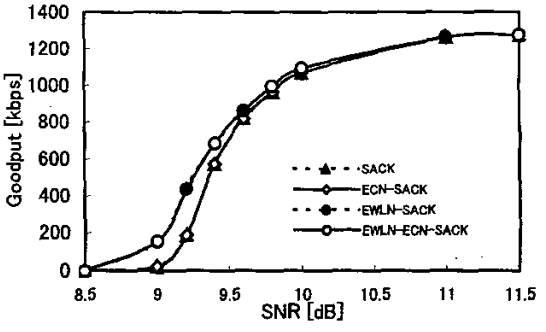

Fig. 6. Performance improvement of TCP-SACK in response to SNR over wireless-to-wired network

Figure 6 shows the throughput comparison between SACK, ECN-SACK, EWLN-SACK and EWLN-ECN-SACK. It is obvious that even in case of SACK, EWLN mechanism significantly improves the throughput when the range of SNR is [8.5 dB, 11dB]. If we compare the Fig 5 and Fig 6, in the best case, throughput of EWLN-Reno is 12 times higher than that of Reno. However, because of the non-congestion network

environment we could not see a performance improvement caused by ECN mechanism.

B. *Behavior of EWLN-ECN over Wired-to-Wireless Networks with a Bottleneck Connection*

In the previous subsection, we have showed the performance comparisons related with EWLN and ECN over a simple wired-to-wireless network where no congestion related losses were involved. From the results of topology 1, we could see the performance improvement of EWLN mechanism. However, the topology was too simple that packet losses occur only due to transmission error on the wireless link.

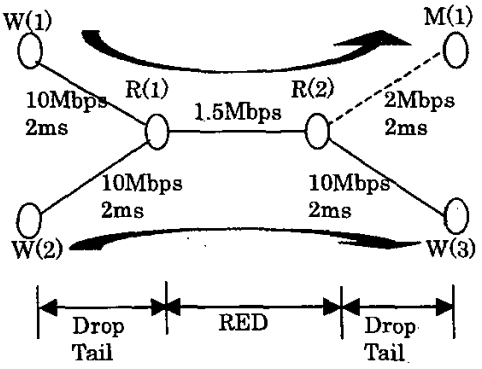

Fig. 7. Simulation topology 2

But in the real wired-and-wireless networks, both congestion or transmission error exist. To confirm the effectiveness of the implemented EWLN-ECN, we have chosen a network model, which has packet losses due to either congestion and transmission error. Figure 7 depicts the network topology with a bottleneck at the wired-to-wireless junction used in the experiment. Two TCP connections are set up, from W(1) to M(1) and from W(2) to W(3). The link between R(2) and M(1) is a wireless link, and the same error model of topology 1 given in (1) is used for this link.

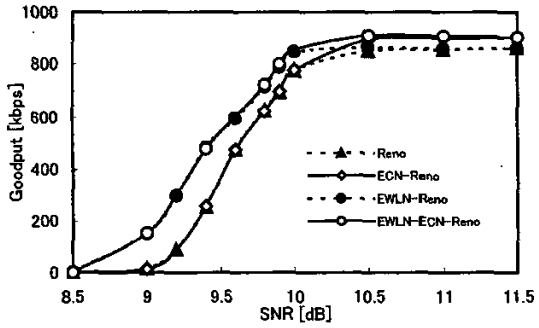

Fig. 8. Performance improvement of TCP-Reno in response to SNR over a congested wired-to-wireless network

Here, we also have chosen TCP-Reno and TCP-SACK to see the throughput improvement caused by EWLN and EWLN-ECN. The simulation time is also 80 seconds. The results of 100 simulation runs are averaged and shown in Fig 8 and Fig 9.

Figure 8 compares the throughput of the connection from W(1) to M(1) using original Reno, ECN-Reno, EWLN-Reno, EWLN-ECN-Reno. Figure 9 compares the throughputs in case of TCP-SACK.

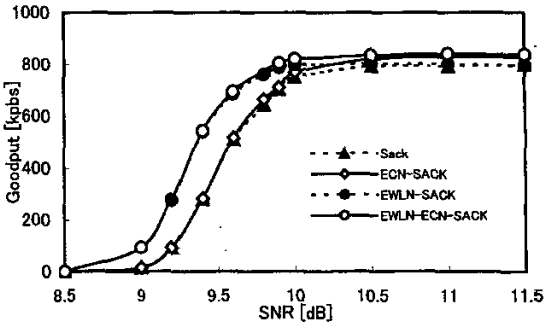

Fig. 9. Performance improvement of TCP-SACK in response to SNR over a congested wired-to-wireless network

The figures clearly show that both EWLN-capable TCP-Reno and EWLN-capable TCP-SACK significantly improve the throughput over the range of [8.5 dB, 10 dB]. However, when SNR increases over 10 dB, throughput of EWLN-Reno and EWLN-SACK decreases and when SNR reaches 11.5 dB it shows the same throughput with that of TCP-Reno and TCP-SACK. The reason of this is that in higher range of SNR, packet loss occurs mainly due to congestion, and EWLN-capable TCP will behave the same as the original TCP. The throughput differences between EWLN-Reno and EWLN-ECN-Reno or EWLN-SACK and EWLN-ECN-SACK is because unnecessary packet drops are avoided due to ECN mechanism.

From the figures it is obvious that EWLN-ECN mechanism gives the highest throughput over all ranges of SNR.

V. CONCLUSIONS

The main advantage of ECN mechanisms is in avoiding unnecessary packet drops, and unnecessary delay for packets from low bandwidth delay sensitive TCP connection. Although ECN mechanism significantly improves TCP performance over wired network, its inability to distinguish wireless link errors from congestion make it unsuitable for wired-and-wireless networks. In this paper, we proposed implementing EWLN mechanism in ECN-based TCP to improve its performance over wired-and-wireless networks. EWLN is implemented on a receiver and sender TCP sides by using the information

from the MAC layer. Through using two kinds of simulation topologies, we confirmed the utility of the implemented EWLN in the ECN-based TCP. EWLN significantly improves performance of TCP in networks where packet losses are caused by either transmission error or congestion and transmission error.

REFERENCES

[1] K Ramakrishnan, and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," RFC2481, January 1999.

[2] H. Balakrishnan, and R.H. Katz, "Explicit Loss Notificaion and Wireless Web Performance," Proc. IEE Globecom Internet Mini-Conference, Nov. 1998..

[3] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving Reliable transport and handoff performance in cellular wireless networks," ACM Wireless Networks, Vol.1, Dec. 1995.

[4] S. Floyd, and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transactions on Networking, V.1 N.4, August 1993.

[5] UCB/LBNL/VINT Network Simulator ns (version 2), http://www.isi.edu/nsnam/ns/

[6] Theodore S. Rappaport, "Wireless Communications-Principles & Practice," Prentice Hall, 1996.

[7] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "Selective acknowledgement options," REC2018, 1996