Explicit Transport Error Notification (ETEN) for Error-Prone Wireless and Satellite Networks – Summary

Rajesh Krishnan, Mark Allman, Craig Partridge, and James P.G. Sterbenz
BBN Technologies

William. Ivancic
Glenn Research Center

***Abstract*— This paper is a summary of the BBN Technical Report No. 8333, "Explicit Transport Error Notification for Error-Prone Wireless and Satellite Networks."**

In this study we discuss two types of Explicit Transport Error Notification (ETEN) mechanisms: (i) *per-packet* mechanisms that notify endpoints of each detected corruption; and (ii) *cumulative* mechanisms that notify endpoints of aggregate corruption statistics. We have implemented the proposed mechanisms in the ns-2 simulator. We present simulation results on performance gains achievable for TCP Reno and TCP SACK, using ETEN mechanisms over a wide range of bit error rates and traffic conditions. We compare TCP Reno and TCP SACK enhanced with ETEN mechanisms against TCP Westwood, which uses a bandwidth estimation strategy in place of the traditional AIMD congestion avoidance algorithm. We discuss two issues related to the practical deployment of ETEN mechanisms: corruption detection mechanisms (and their co-operation with ETEN-based recovery in the transport layer) and security aspects. We include recommendations for further work.

***Index Terms*—Congestion Control, Explicit Transport Error Notification, Internet, Protocols, Satellite, TCP/IP**

I. Background

NASA is working to extend the Internet into space in order to improve communications, enable new system capabilities and reduce overall mission costs. As such, NASA is interested in leveraging technologies developed by the commercial communication industry. In particular, NASA is interested in utilizing commodity protocols, the TCP/IP protocol suite, wherever possible.

NASA commissioned BBN Technologies to investigate the potential network performance benefits of ETEN and the practical issues involved in implementing and deploying ETEN. This paper is a summary of the BBN Technical Report No. 8333, "Explicit Transport Error Notification for Error-Prone Wireless and Satellite Networks."

II. Introduction

One obstacle to good performance of internetworks with wireless and satellite components is non-negligible bit-error rates (BER). The most widely used transport protocol in the TCP/IP suite, the transmission Control Protocol (TCP) [1], guarantees that corrupted data will be retransmitted by the data sender, hence providing a reliable byte-stream to applications. However, packet loss is also used by TCP to determine the level of congestion in the network [2] – as traditionally, the bulk of packet loss in networks comes from router queue overflow (i.e. congestion). Therefore, to avoid congestion collapse TCP responds to packet loss by decreasing the congestion window [2] [3], and therefore the sending rate. The reduction of the congestion window is not needed to protect network stability in the case when losses are caused by corruption and therefore these needless reductions in the sending rate have a negative impact on performance with little overall benefit to the network.

If the TCP sender can distinguish packets lost due to congestion from packets lost due to errors, better performance may be achieved. The performance benefit can be realized if TCP can retransmit a packet lost due to corruption without needlessly reducing the transmission rate, while continuing to protect network stability by decreasing the sending rate when loss is caused by network congestion.

TCP Explicit Transport Error Notification (ETEN) is the concept of notifying TCP that packets were lost due to corruption[1]. ETEN mechanisms can aid TCP in distinguishing packets that are lost due to congestion from ones that are lost due to corruption.

The purpose of this study is two-fold:

1. To establish bounds on the performance improvements that can be obtained with the use of ideal ETEN mechanisms under different network conditions – error rates, capacities, delays, topologies, congestion – and thereby determine promising directions for future research, if any.
2. To consider issues related to practical deployment of ETEN mechanisms, to propose suitable architectures and mechanisms, to identify security vulnerabilities, and to identify areas that require further study before an ETEN system is viable.

Through simulations, we have evaluated possible enhancements to TCP that are based on ETEN notifications from intermediate routers and/or end systems. Emulations in a testbed and live testing over real networks were considered out of scope of this effort. This study included the following tasks:

- Determine bounds on TCP goodput improvements possible from ETEN when a TCP sender is presented with ideal information about the cause of each loss.
- Evaluate via simulations, actual performance achievable over a range of network topologies and traffic conditions with different TCP variants such as Reno and SACK.
- Discuss and evaluate the performance of specific ETEN mechanisms that fall in one or more of the following broad categories:
  - Forward notification – whereby any notification about corrupted packets is sent in the direction of the data packets and then returned to the sender in TCP acknowledgment segments.
  - Backward notification – in which a message is sent from the node (end-host or intermediate router) that detects a corrupted packet to the host that originated the packet.
- Per-packet mechanisms that attempt to determine the root cause of each loss experienced.
- Aggregate notification schemes where the TCP sender is provided with aggregate statistics about the loss patterns experienced in the network path.
- Determine how TCP should best react upon receiving ETEN notification.
- Assess the security implications of introducing various ETEN mechanisms into the Internet architecture. These include:
  - Potential vulnerabilities of the proposed mechanisms to distributed denial-of-service attacks.
  - Operation over encrypted tunnels, VPNs, and MPLS paths, where intermediate nodes may not be able to determine actual source or destination IP addresses and ports, making ETEN notification effectively impossible.
  - Vulnerabilities to misbehaving receivers that attempt to mask congestion-related losses using ETEN mechanisms in an attempt to obtain an unfair share of network resources.

III. Error Notification and Response Mechanisms

For the ETEN mechanisms proposed in this report we assume one of the following two cases holds:

1. The source and destination IP addresses, the source and destination TCP ports, and the TCP sequence number can be correctly obtained from the corrupted packet. In addition, the packet in question must be part of the sender's current window; otherwise, the opportunity to mitigate the performance problems caused by the corrupted packet is lost. For this case, Oracle, Backward and Forward ETEN were considered with Oracle and Backward ETEN simulated.
2. The node detecting errors can only calculate cumulative error rates for each link. In other words, the information in the header of a corrupted packet is considered inaccurate. Both Forward and Backward Cumulative ETEN were considered for this case with only Forward CETEN (FCETEN) simulated.

[1] ETEN is simlar to Explicit Congestion Notification (ECN). In ECN, TCP can be informed of the onset of congestion and adjust its transmissions accordingly thereby improving overall performance.

IV. ORACLE ETEN

Oracle ETEN, illustrated in Figure 1, is a theoretical construct that assumes sufficient knowledge about the corrupted packet (sender and destination IP addresses, sender and destination TCP port numbers, and the TCP sequence number) is available to the intermediate router or the end-system that detects corruption. Furthermore, this mechanism assumes that the source of the flow can be instantaneously notified of the packet corruption. Oracle ETEN provides an upper bound on the performance improvement achievable by ETEN mechanisms that notify the source. While the Oracle ETEN mechanism is an impossibility in the real world, it can be used to distinguish between cases in which *some* ETEN mechanism would be useful and cases when no ETEN scheme would aid performance.

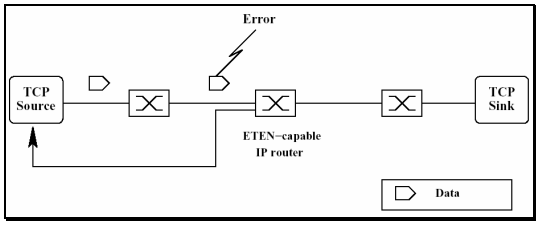

Figure 1 - Oracle ETEN

V. BACKWARD ETEN

The backward ETEN (BETEN) mechanism, illustrated in Figure 2, is analogous to backward explicit congestion notification schemes (e.g., source-quench [4]). This mechanism assumes that the intermediate router can extract or reconstruct (e.g., using FEC) sufficient knowledge about the corrupted packet that is required to notify the sender.

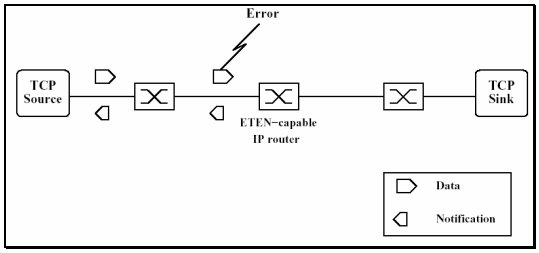

Figure 2 - Backward ETEN

VI. FORWARD ETEN

The forward ETEN (FETEN) mechanism illustrated in Figure 3 is analogous to forward explicit congestion notification schemes (e.g., [6] [7]). This mechanism also assumes that the intermediate router can extract (or reconstruct using FEC) complete and correct knowledge of the IP addresses, TCP ports, and TCP sequence number corresponding to the corrupted packet. Upon detection of a corrupted packet, the intermediate router transmits a FETEN message to the destination host, which then conveys the information to the sender on a subsequent acknowledgment.

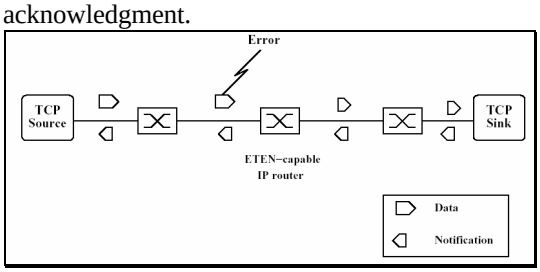

Figure 3 - Foreward ETEN

VII. CUMULATIVE ETEN

In practice, we cannot always accurately retrieve the source and destination IP address, source and destination TCP port numbers, and TCP sequence number from a corrupted packet or link-layer frame. For such cases we consider ETEN mechanisms that work on the basis of cumulative error rates (for example, error rates that are averaged over an interval of time and across various flows), rather than attempting to make notifications on a per-packet basis.

The cumulative ETEN (CETEN) information conveyed to the end-hosts can be in one of several different forms:

- An *absolute* bit error rate, byte error rate, or packet error rate observed within a moving window in time.
- The error rate may be quantized into a small number of steps (for example, *high*, *medium*, and *low*).
- A binary feedback scheme [7] (see also [5] [6]) is a special case that provides indication that the bit/byte/packet error rate exceeds some threshold.
- A *relative* error rate that simply indicates that the quantized error rate has increased or decreased from the previous value.
- An estimate of the probability that a packet survives corruption.

CETEN information can be delivered to a sender via forward or backward signaling,

analogous to a FETEN-based or a BETEN-based strategy. Also, CETEN can be piggybacked on data and acknowledgment packets, rather than using additional distinct messages.

CETEN information can be collected on a per-hop basis or aggregated over the end-to-end path. Due to the difficulty in correctly assigning corrupted packets to their corresponding flows, any per-flow CETEN information has to be estimated, for example from what is observed across all flows using a given link. CETEN strategies that rely purely on statistics collected within the lifetime of a particular flow are of limited use for short flows. For example, a short flow may have terminated before we obtain a good estimate of the packet corruption probability.

VIII. Sender response to ETEN

The sender's response to an ETEN notification depends on the type of the notification. If the sender receives timely and reliable information about the corrupted packet that identifies the TCP flow and the sequence number within the flow, then the sender can retransmit the corrupted packet without adjusting the congestion state. However, if the information contained in the ETEN notification is only partially reliable, or if only a cumulative error rate is available, then the sender has to apply a heuristic to determine what action is appropriate. When a transport endpoint infers a packet loss, it cannot exactly determine from the CETEN information if the packet loss occurred due to corruption or congestion. At best, the CETEN information provides a recent estimate of the fraction of the losses that are due to corruption. The decision to be made by the sender includes whether an outstanding segment should be retransmitted and whether the congestion state should be altered in response.

Since most link level technologies require corrupted packets to be discarded even before it reaches the IP layer, per-packet ETEN mechanisms (at the IP and TCP layers) cannot see the corrupted packets. Although the sender response to per-packet ETEN is more straightforward than the response to CETEN, it must be noted that the corruption link layer counters of errors are readily available; these counters can be used to generate CETEN.

IX. Performance of ETEN Mechanisms

In this section, we describe results of simulations on the performance of Oracle ETEN, BETEN and FCETEN. Various types of links (e.g., terrestrial LAN, WAN, and satellite), modeled by their respective latencies, are simulated over a wide range of bit error rates. ETEN performance is compared against conventional Reno [2] and SACK [8] variants of TCP. Each simulation consists of a bulk TCP flow (FTP application) of 120 seconds duration with unlimited data to send. The actual values and variable ranges used in the study are listed in Table 1. All simulations were performed using the ns-2 simulator [9] (version 2.1b7a) with extensions.

Table 1 - Parameters Values

Parameter	Value
Link capacity	1.5 Mb/s, 10 Mb/s, 100 Mb/s
Forward link bit error rate	1.56×10^{-10} – 1.56×10^{-5}
Backward link bit error rate	0 or same as forward bit error rate
Link propagation delay	10ms, 100ms, 320ms
TCP variants	Reno, SACK
ETEN mechanisms	none, Oracle ETEN, BETEN
MSS	536 bytes
Receiver window	20 segments
Router buffers	50 packets shared FIFO queue

Oracle ETEN represents the ideal, yet impossible, baseline that provides an upper bound on the performance achievable by any practical per-packet ETEN scheme. One design goal is that the addition of any ETEN scheme (to any given TCP congestion avoidance strategy) should not make the performance worse; therefore, the case with no ETEN is expected to provide a useful lower bound (and, this is shown in our simulation results).

The BETEN strategy represents an implementable per-packet ETEN strategy (assuming that we can extract sufficient information from corrupted packets). In the absence of congestion, we can expect that the goodput when using BETEN will lie between the goodputs using Oracle ETEN and no ETEN.

The CETEN strategy represents an implementable cumulative ETEN strategy that is potentially more robust in terms of security than per-packet ETEN strategies, but theoretically provides less performance gains. In our strategy the CETEN flows in the forward direction and gets copied over on to the acknowledgments going back.

We consider eight sets of simulations, as follows:

A. Baseline – no cross traffic over a single-hop topology

This set of simulations is aimed at evaluating the gains possible over a single uncongested link using Oracle ETEN and BETEN with TCP Reno and TCP SACK.

B. Multi-hop topology with no cross-traffic

In this set of simulations, we use a 3-hop linear topology of identical links, while varying the other parameters outlined above. These simulations serve the purpose of validating our implementation in a more complex topology with multiple links and routers. The results are expected to match those of the first set.

C. Multi-hop topology with competing UDP flows:

In this set of simulations, we use a 3-hop linear topology to provide insight into the performance of ETEN mechanisms in the face of congestion from constant-bit-rate UDP traffic. The intensity of cross-traffic is varied across simulation runs. The competing traffic in these simulations does not use a congestion avoidance strategy.

D. Multi-hop topology with competing TCP flows:

This set of simulations offers competing TCP traffic (instead of UDP traffic) and is otherwise identical to the third set. This provides insight into the performance of ETEN when the competing traffic flows also use a congestion avoidance strategy.

E. Comparison of ETEN to TCP Westwood:

This set of simulations provides performance comparison of our ETEN mechanisms with TCP Westwood [10] in the absence of cross traffic. Recently proposed modifications to TCP congestion avoidance include using bandwidth estimation techniques. TCP Westwood [10] is a representative congestion avoidance strategy based on bandwidth estimation. TCP Westwood has been shown to perform well under high error rates in simulated comparisons to TCP Reno and SACK TCP. Here, we compare via simulations the performance of ETEN with Reno and SACK against TCP Westwood.

F. Comparison of ETEN to TCPWestwood with UDP cross-traffic:

This set of simulations provides performance comparison of our ETEN mechanisms with TCP Westwood [10] in the presence of cross traffic.

G. Cumulative ETEN performance with UDP cross traffic:

In this set of simulations, we use a 3-hop linear topology of identical links. The performance of CETEN is evaluated in the presence of UDP cross traffic.

H. Cumulative ETEN performance with TCP cross traffic

In this set of simulations, we use a 3-hop linear topology of identical links. The performance of CETEN is evaluated in the presence of TCP cross traffic.

X. PERFORMANCE

The following are three sample results of the various tests that were performed in this study. For a detailed description of all the tests and results, refer to the complete BBN report.

A. Basline

In the baseline set of simulations, we investigate a single TCP flow over a single link with channel errors that result in packet corruption. In this set of simulations, there is no cross-traffic competing with the TCP flow. Examining ETEN in isolation provides an empirical upper bound on the gain in TCP goodput that is achievable using ETEN mechanisms. The baseline for the simulations is the performance of TCP Reno and SACK under various error rates. We consider two near-ideal conditions for the error detection and notification:

1. Oracle ETEN – complete knowledge of the corrupted packet and instantaneous notification to the source.
2. BETEN – complete knowledge of the corrupted packet with real BETEN messages propagating back to the source.

The results in Figure 4 show the goodput using Reno with Oracle ETEN over a long-thin network (at a BER of 10^{-5}) is almost seven times the baseline goodput using Reno alone. The goodput using BETEN with SACK is more than three times the SACK baseline, and the goodput using BETEN with Reno is about two and one

half times the Reno baseline. The figure also illustrates that when the errors are not as prevalent on the link the ETEN mechanisms have a relatively small impact because errors have only a small impact on stock TCP.

From the simple simulations presented in this section we can derive several conclusions:

- The performance using BETEN with SACK is close to that of Oracle ETEN at low error rates.
- As the BER increases, the chances of losing a notification also increases and we see that gains from BETEN begin to diminish.
- Using BETEN with SACK outperforms BETEN with Reno; this may be because the ability of SACK to correct multiple losses complements ETEN.
- In general, TCP SACK performs better than TCP Reno due to the ability of TCP SACK to mostly decouple loss recovery from congestion control.

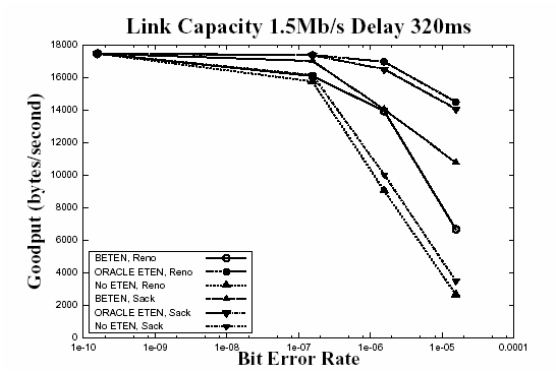

Figure 4 - TCP with ETEN over an uncongested long thin network (LTN)

B. TCP Westwood versus SACK BETEN

For the simulation results in figure 5, we compare the performance of TCP Westwood when both congestion and corruption losses are present. Figure 5 shows the performance of TCP Westwood and BETEN over a 3-hop linear topology with 1.5 Mb/s links each with a one-way delay of 320 ms. We use competing UDP traffic for these simulations. The plot shows that at high error rates and moderate congestion, BETEN's ability to distinguish between corruption and congestion losses provides performance improvements over the TCP Westwood strategy that relies on intelligent bandwidth estimation alone. The Westwood strategy, however, shows an advantage under heavy congestion (competing flows) with low to moderate error rates.

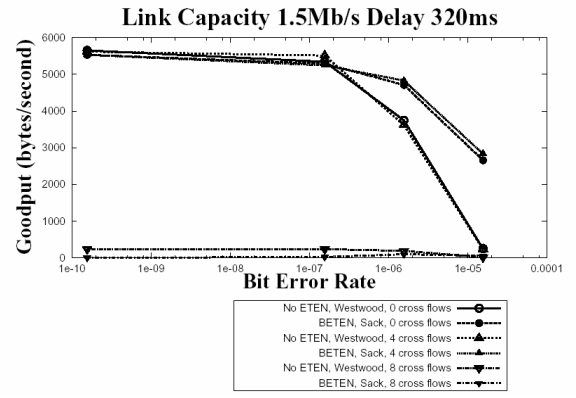

Figure 5 - TCP Westwood versus SACK TCP with ETEN over a long thin network (LTN)

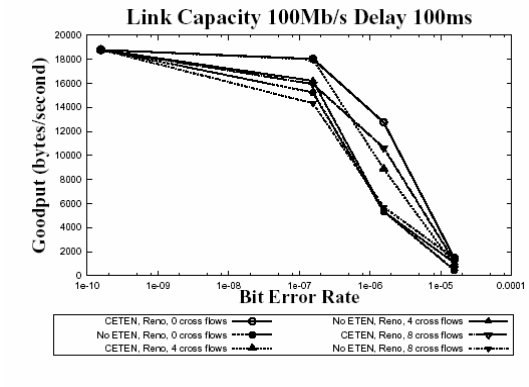

Figure 6 - CETEN Performance with TCP Reno and TCP cross traffic

C. Cumulative ETEN versus TCP RENO

The simulation results in figure 6 show CETEN with TCP cross traffic[2]. The results indicate that under all congestion levels, CETEN offers moderate goodput gains over TCP Reno, except at high BER (10^{-5}). The CETEN simulations we conducted as part of this investigation show CETEN to be a promising approach in some situations. In other situations, CETEN offers worse performance than TCP Reno. We feel that further investigation into additional CETEN mechanisms is warranted before making conclusions on the feasibility of CETEN in general. For instance, an investigation

[2] It is important to note that the competing traffic in our simulation did not use any ETEN mechanism. Thus, the competing traffic needlessly reduce their transmission rates when they experience corruption losses. This allows the flow of interest to use more of the bottleneck bandwidth.

into how well the end system can estimate the total loss rate and use that for determining the fraction of losses caused by congestion may shed additional light on CETEN (and make it more feasible to deploy).

XI. Security Considerations

ETEN techniques (such as BETEN, for example) that require out-of-band messages are vulnerable to distributed denial of service (DDOS) attacks because networks that plan to use this form of ETEN will have to allow such messages to enter or leave their networks. This makes it possible for an adversary to launch a DOS attack by bombarding a host (or a network) with ETEN messages. This can minimally overwhelm the victim host, but if launched as a distributed denial of service attack from a large number of hosts (that have been compromised by an Internet worm, for instance), an attack can overwhelm the capacity of entire networks [11].

ETEN mechanisms may be vulnerable to another more subtle and indirect attack. A malicious adversary can send false notifications corresponding to packets that are either not dropped or were dropped due to congestion. This can induce the sender into retransmitting packets unnecessarily or into bypassing congestion avoidance and continue transmitting at a higher rate than appropriate for the given network conditions. This attack in isolation (on a single flow) can cause limited damage. However, if a coordinated attack were launched on many TCP flows on a heavily loaded network, the attack can potentially drive the network into congestion collapse [12].

The use of encryption can prevent deep header inspection. For example, IPsec [13] hides TCP port information; IPsec tunnels also hide the original source address. This makes it difficult for intermediate routers to determine the correct TCP endpoints to which ETEN messages should be delivered.

XII. Conclusions

Our conclusions from this study are:

- Per-packet ETEN mechanisms offer substantial gains in bulk TCP goodput in the absence of congestion; however, in the presence of congestion TCP congestion avoidance mechanisms dominate resulting in insignificant gains from ETEN.
- The proposed per-packet mechanisms provide useful upper bounds on performance that can be used to evaluate future proposals of per-packet and cumulative ETEN techniques.
- Per-packet mechanisms present significant challenges to practical implementation by providing a new opportunity to exploit Internet security vulnerabilities and by requiring intermediate nodes to reliably extract information from the headers of corrupted packets
- Cumulative ETEN techniques are more attractive to implementation; however, the particular mechanism we evaluated did not realize the potential gains of per-packet techniques
- Security vulnerabilities include not only denial-of-service attacks but also more subtle attacks with effects ranging from unfair bandwidth sharing to total congestion collapse of the network.
- Future work in this area should focus on alternative cumulative ETEN mechanisms, accurate loss inference at endpoints to avoid tracking congestion losses at every hop, interactions with forward error correction, and cross-layer co-operation for ETEN.

XIII. Recommendations for Future Work

The results of this initial broad study are intriguing; they lead us to recommend further work focused on specific aspects of ETEN. On the one hand, our work demonstrates tremendous potential from ETEN if reliable information extraction from headers were possible and congestion can somehow be controlled. On the other hand, it uncovers a number of practical challenges coupled with achieving only limited success with the particular cumulative ETEN scheme we implemented.

The primary thrust that we recommend is to explore cumulative ETEN alternatives that do not rely on congestion feedback from intermediate routers (since this would implicitly demand global deployment and render the scheme less practical). We believe that the biggest challenge to realizing CETEN schemes is the inability of a TCP endpoint to accurately estimate the total loss at a fine resolution (of a few packets) and in a timely manner (within an

RTT to enable quick recovery). Research is needed to develop this capability.

Given this capability, we recommend that our proposed cumulative ETEN scheme should be refined to make use of it and then re-evaluated. The interactions of ECN with the refined cumulative ETEN scheme also remain to be studied in this context.

Our current effort focused on quantifying throughput improvements achievable using ETEN and was therefore limited to long-lived TCP flows. Further work is needed to isolate the effects of loss during the slow start phase and quantify the benefits of ETEN for short-lived flows. We also recommend that the mechanisms be evaluated using real network topologies and traffic traces including other workloads, for example, HTTP transactions. Under high error rates, TCP connection establishment can be delayed or can fail completely. We believe that increasing the connection establishment rate under high error rates could be a key benefit of ETEN. We recommend that future work address this issue.

REFERENCES

[1] J. Postel (editor), "Transmission Control Protocol," *Request for Comments: 793*, September 1981.

[2] V. Jacobson, "Congestion Avoidance and Control," *Proceedings of ACM SIGCOMM '88*, Stanford, CA, USA, August 1988.

[3] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," *Request for Comments: 2581*, April 1999.

[4] ISI, "Internet Control Message Protocol," *Request for Comments: 792*, September 1981.

[5] K. Ramakrishnan, and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP," *Request for Comments: 2481*, January 1999.

[6] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," *Request for Comments: 3168*, September 2001.

[7] K.K. Ramakrishnan, and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance," *ACM Transactions on Computer Systems*, Volume 8, Number 2, May 1990, pp. 158–181.

[8] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options," *Request for Comments: 2018*, October 1996.

[9] ns-2 simulator, http://www.isi.edu/nsnam/ns/index.html

[10] 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," P*roceedings of MOBICOM 2001*, Rome, Italy, July 2001.

[11] S. Gibson, "The Strange Tale of the Attacks Against GRC.COM," h*ttp://grc.com/dos/grcdos.htm.*

[12] S. Floyd, and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, August 1999, pp. 458–472.

[13] S. Kent, and R. Atkinson, "Security Architecture for the Internet Protocol," *Request for Comments: 2401*, November 1998.