An In-Band Easy-to-Deploy Mechanism for Network-to-Transport Signaling

Mayank Sharma	Dina Katabi	Rong Pan	Balaji Prabhakar
Stanford	MIT	Stanford	Stanford
msharma@stanford.edu	dk@mit.edu	rong@stanford.edu	balaji@stanford.edu

Abstract— **Network-to-transport signaling is desirable for ensuring efficient resource usage and timely notice of network status. ICMP is the standard way for signaling, but unfortunately it generates extra load and does not traverse firewalls. In this paper, we develop M-ECN, an in-band network-to-transport signaling mechanism, which does not generate any extra packets and does not require dedicated header bits. The key idea is to sneak messages into the stream of ECN bits, *but without* interfering with ECN congestion signaling. Compared to other alternatives, M-ECN is easy to deploy because routers read/write to the IP header, and the mechanism requires no change to legacy routers along the path which do not participate in the signaling.**

I. Introduction

In this paper, we are interested in network-to-transport signaling, a restricted form of communication in which the network reports to the end-points of a flow the occurrence of a particular event. Signaling is desirable for ensuring efficient resource usage and timely notice of network status. Some example applications are: a router signaling to a sender that a packet loss was due to wireless errors rather than congestion [2], [4], [6], [10], [16], warning a source that it is suspected of misbehaving [17], and informing sources of a drastic change in link capacity [9].

Prior proposals for network-to-transport signaling require either sending out-of-band ICMP messages or using dedicated fields in packet headers. Both approaches have shortcomings: Out-of-band messages generate extra load, do not cross firewalls, and create the potential for misuse (e.g., an attacker may misinform a victim that the path capacity has been drastically reduced). Using dedicated fields in the IP header is problematic given that header real-estate has already been allocated to different functionalities. The use of additional application-dependent headers requires a change to router architecture, interferes with tunnels, and violates end-to-end semantics.

We propose multi-functional header fields which can simultaneously be used by more than one task. Specifically, we develop the Multiplexed ECN (M-ECN) channel as a means for allowing routers and end-systems to communicate low rate information using ECN, but without interfering with ECN's signaling of congestion. The key observation we leverage is this: The stream of ECN bits from a flow creates a communication channel whose capacity can be up to one bit per packet. But because the ECN mark is infrequently set, the rate of information transfer over this channel is much smaller than its capacity. Thus, there is some left-over capacity to communicate other information. To use this spare capacity, the router needs to spread a message across multiple packets of a flow, while ensuring that ECN congestion marks are given priority over other information.

Our contribution is twofold. First, we introduce the M-ECN channel as a way of designing multi-functional header fields. Prior work on using a field in the IP header for a new task (which differs from the field's actual purpose) did not explicitly require the (peaceful) co-existence of different functions on a single header field [23], [25]. Modeling the M-ECN channel as a "**Z**-channel with erasures" allows us to information-theoretically evaluate the capacity of the M-ECN channel and compute the spare capacity available for other information. Some other features of M-ECN are:

1. It is amenable to gradual deployment. Only routers and end-points which participate in communication over the M-ECN channel need to be modified. The channel is transparent to legacy routers.

2. It can be simultaneously used by multiple routers along a path to communicate with the end-points of a flow.

3. Different flows can use it for different applications.

4. It uses ECN-Nonce to prevent a receiver from conveying the wrong M-ECN signal to the sender.

5. It is resilient to packet drops and reordering.

Second, as a concrete application of the M-ECN channel abstraction, we develop and evaluate a novel protocol called WiSE, for signaling wireless error drops to the source. Such signaling is known to significantly improve TCP performance over wireless links. We devise a simple scheme for signaling packet corruption on the M-ECN channel via ECN mark positions. We show through simulation that WiSE achieves a performance gain comparable to previous end-to-end wireless error recovery mechanisms, such as ELN [2] and ETEN [16], *but without* generating extra messages or requiring dedicated bits in packet header. Further, WiSE is fair to non-WiSE sources and reacts properly to ECN congestion marks.

II. Limitations of Prior Signaling Mechanisms

Current mechanisms for network-to-end-points signaling are sometimes inadequate or suboptimal, as argued below.

(A) Out-of-Band ICMP Messages: ICMP is the standard way for communicating information between network and users. Unfortunately, as a signaling mechanism, ICMP has many limitations. First, ICMP messages are filtered out by most firewalls because they create a security hazard and a potential for a denial-of-service attack [5]. Second, ICMP messages create extra load and hence should not be used too often. Third, ICMP does not deal with IPsec [15] or tunnels. Fourth, ICMP may be inappropriate for certain applications. For example, consider using ICMP messages to signal error drops over a multi-hop wireless network. When fading occurs the quality

0-7803-8794-5/04/$20.00 © 2004 IEEE

of the wireless channel degrades causing a large number of packet losses in a short interval. The router will generate ICMP messages to signal each corrupted packet to its sender. But, in a multi-hop wireless network, the router will need to use the same erroneous wireless channel to send the ICMP messages. So the ICMP messages will compete with the data packets for the low quality wireless channel causing even more errors and drops, and consequently more ICMP messages.

(B) IP options: Information may be exchanged between routers and end-systems in IP options. Currently the use of IP options in packets removes them from the fast path, and causes them to incur substantial delays. Thus, IP options are not desirable for signaling repeated events (e.g., wireless error drops). In general, the problem with IP options is gradual deployment; even if one changes the signaling router to use IPv6 or to implement IP options efficiently, legacy routers along the path will either drop the packet (if it is IPv6) or send it along the slow path.

(C) Dedicated Bits in the IP or TCP Header: Using dedicated fields in the IP header is problematic given that header real-estate has already been allocated to different functionalities. Allowing routers to write to fields in the TCP header requires the router to recompute the TCP checksum for each packet, violates end-to-end semantics, interferes with tunnels, and makes it hard to use the same service with other transport protocols.

(D) Additional Dedicated Headers: Conceptually, it is possible to signal using an additional service-dependent header that intervenes between the TCP and IP headers. In practice, this causes deployment problems, might require changing router architecture, and each new application will define a new header. Currently routers strip the first few bytes of a packet which usually contain the IP and TCP headers. Inserting additional headers between IP and TCP may prevent access to the TCP header, and confuse legacy routers which want to read some TCP fields (e.g. port numbers) but access the new header instead.

III. Background & Terminology

(A) Explicit Congestion Notification (ECN): ECN allows routers that use active queue management (AQM) to mark packets as an indication of imminent congestion. This has the potential of limiting packet drops due to a buffer overflow. ECN uses the ECT and CE bits in the IP header for signaling between routers and connection endpoints, and uses the ECN-Echo, CWR, and NS bits in the TCP header for signaling between sender and receiver. Figure 1 defines the meaning of these bits as presented in RFC 3168 [22].

(B) Terminology: For simplicity, we will always refer to ECN as being one bit. Thus, ECN = 1 means that the CE and ECT flags are "11", whereas ECN = 0 means that the flags are either "01" or "10".

IV. The M-ECN channel

M-ECN is an in-band easy-to-deploy mechanism for network-to-transport signaling. We envision that some routers in the network may be equipped to signal certain events. For example, a subset of the wireless routers may be able of signaling packet corruption; some other routers may be equipped to signal quality of service, etc. Each one of these services will have a unique well-known identifier. Using its first packet, a flow (identified by its sender, receiver, and ports) may subscribe to a particular M-ECN signal by specifying the service identifier in an IP router-alert option [13]. (Subsequent packets of the flow do not use IP options).

ECT CE	Meaning
00	ECN incapable
10	no congestion, Nonce=0
01	no congestion, Nonce=1
11	congestion, Nonce cleared

(a) ECN encoding in the IP header (see §VI-A for nonce definition)

ECN-Echo: Set by the receiver in TCP ACK to signal to the sender the reception of an ECN marked packet.
CWR: Set by the sender to acknowledge its receipt of and reaction to the ECN-Echo flag.
NS: The receiver returns the cumulative sum of the nonce in the NS bit in the TCP ACK.

(b) ECN bits in the TCP header

Fig. 1. The bits used by ECN in both the IP and TCP headers.

Routers signal to the end point of a flow using messages constructed out of ECN marks without interfering with ECN congestion signaling (see Figure 2). The key idea is to use *opportunistic signaling*; i.e. to sneak messages into seldom-used standardized fields, while giving priority to the original functionality of the field. In §V-C, we discuss in detail how to construct M-ECN messages, but at a high-level opportunistic signaling works as follows:

1. It spreads an M-ECN message across multiple packets. We call the sequence of ECN values (i.e. string of 1's and 0's) in a single M-ECN message a **Codeword**. Spreading a codeword across multiple packets requires a router to maintain per-subscriber state. Although this limits the number of flows to which the router can signal at high speed, the space of possible applications is still substantial. For example, both wireless loss notification and warning a suspected misbehaver are potential applications, since in the former the bottleneck is the link bandwidth not the router, whereas in the latter the router would need to maintain state to detect misbehaving regardless of whether it uses the M-ECN channel [17].

2. It encodes information in patterns (i.e., strings) of 0's & 1's that are unlikely to be generated by ECN congestion marking.

3. It gives priority to ECN congestion signaling over sneaked messages. First, whenever an AQM router along the path marks a packet because of congestion, it results in an illegal M-ECN codeword which allows the end system to recognize the congestion mark. Second, M-ECN codewords have a very low density of 1's, leaving most of the packets available for congestion marking by AQM routers.

4. To construct M-ECN codewords, we use the observations: 1) ECN marks can be swapped between two packets that are closely spaced in time; 2) a cluster of ECN marks is equivalent to a single mark since TCP reacts to only one decrease signal in a congestion window (`cwnd`). Similarly, an ECN mark

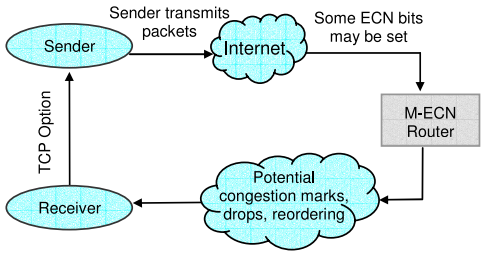

Fig. 2. M-ECN routers signal to the receiver using code-words constructed out of ECN marks. If needed, the receiver can relay the message to the sender using an end-to-end mechanism such as a TCP option.

inserted after a drop is harmless because TCP reacts only to one congestion signal per cwnd.

We believe that the concept of opportunistic signaling may be applicable to fields other than ECN (e.g. the DiffServ bits). We have chosen to focus on ECN because it is standardized, has been widely deployed in both routers and end systems, and being in the IP header, is easily accessible to routers for reading and writing. Further, there are a few proposals for using ECN to signal alternative information such as congestion price [11]. These proposals usually assume that all routers along the path have been updated to understand their alternative meaning of ECN. But if some routers perform conventional ECN congestion marking and others use the alternative meaning, the resulting signal will contain unpredictable errors. Hence, some of these proposals may be able to use M-ECN for gradual deployment to allow them to deliver a signal even when the path traverses legacy ECN routers.

A. The Capacity of the M-ECN channel

Using methods from information theory, we model the M-ECN channel as a binary channel with erasures and compute the left-over capacity after congestion marking. In the canonical formulation of the communication problem, there is a transmitter communicating to a receiver over a noisy channel. Here, the transmitter is an M-ECN router and the receiver is a flow's end-point. The effect of ECN congestion marking on M-ECN messages is modeled as channel noise. This noise flips a 0 to 1 with the congestion marking probability p, but it faithfully transmits a 1 with probability 1, as illustrated in Figure 3(a). The resulting channel is the **Z**-channel of information theory. The capacity of the **Z**-channel is known [12], and asymmetric error correcting codes have been constructed and shown to perform well [21].

While the **Z**-channel models the congestion marking process at a regular AQM router, it does not model the effect of packet drops. Information theoretically, packet drops lead to the "erasure" of a 0 or a 1, which we model by the **Z**-channel with erasures (or the **ZwE**-channel), shown in Figure 3(b). In the **ZwE**-channel a 0 or a 1 is erased with probability ϵ, which is the drop probability along the path. When an erasure (i.e. a packet drop) occurs, we receive an e (i.e. a gap in the sequence number) instead of a 0 or a 1.

We shall now compute the capacity, $C(p, \epsilon)$, of the **ZwE**-channel as a function of the congestion marking probability p and the drop probability ϵ. This allows us to understand how

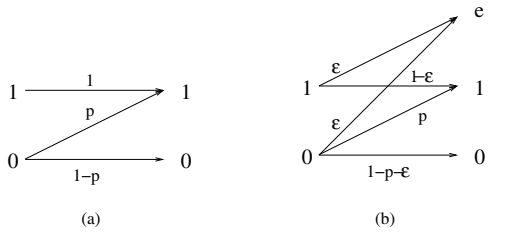

Fig. 3. (a) The **Z**-channel (b) The **ZwE**-channel

much spare capacity remains for transmitting extra information over the M-ECN channel. By definition, (see [7]) the capacity of a discrete memoryless channel is:

$$C = \max_{0 \le q \le 1} I(X;Y) = \max_{0 \le q \le 1} H(Y) - H(Y|X), \quad (1)$$

where $I(X;Y)$ is the Mutual Information between the input X and the output Y, $H(Y)$ is the entropy of the output, $H(Y|X)$ is the conditional entropy of the output given the input and the maximum is over all possible probability distributions on the input. For the **ZwE**-channel, the inputs X are 0 or 1, and the possible outputs Y are 0, 1 or e. We want to express this capacity as a function of the density of 1's in the code q because we want to ensure that M-ECN messages have a low density of ones so that there are enough unmarked packets to allow a regular AQM router to signal congestion through marking. Thus,

$$I(X;Y) = H\left(\epsilon, q(1-\epsilon) + (1-q)p, (1-q)(1-p-\epsilon)\right) - qH\left(\epsilon, 1-\epsilon\right) - (1-q)H\left(\epsilon, p, 1-p-\epsilon\right) \quad (2)$$

and the capacity $C = C(p, \epsilon)$ can be computed as an explicit function of p, ϵ, and q (details of the derivation are in [24]).

To give a feeling for the result, we plot in Figure 4 the capacity for the particular case where $p = 0.01$ and $\epsilon = 0.01$. The figure shows that there is reasonable capacity left even if we constrain our codewords to have a very low density of 1's. For example, if the congestion marking probability $p = 0.01$, the drop probability $\epsilon = 0.01$, and one allows the density of ones in the codewords to be 0.02, then the capacity turns out to be 0.11 bits/packet. Thus, the M-ECN channel provides $0.11 - p = 0.1$ bits/packet to signal extra information. Said differently, on an average, if we had a sequence of 100 packets of which 1 got randomly dropped and 1 randomly marked, and we were allowed to use only 2 ones (i.e., 2 marks) to communicate a message, then we could transmit up to 10 bits of information using this packet sequence.

To construct M-ECN messages, one needs to follow the rules of opportunistic signaling in §IV. Next, we consider a particular application called WiSE and in that context, devise a simple scheme which constructs messages using the position of ECN marks relative to wireless drops.

V. WiSE: Wireless Signaling via ECN

As a concrete application of M-ECN, we propose WiSE, a protocol for signaling wireless losses to a TCP sender over the M-ECN channel.

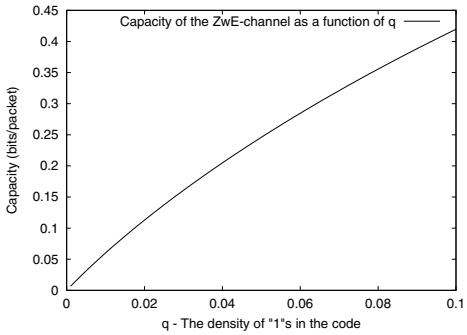

Fig. 4. Capacity of the **ZwE**-channel as a function of the density of 1's for congestion marking probability $p = 0.01$ and drop probability $\epsilon = 0.01$.

A. Problem & Previous Solutions

The wireless channel is well-known for exhibiting error rates substantially higher than those observed in wired environments. Although current access technologies have mitigated this problem to some extent for the last hop, short range, indoor environment, they are still of concern in designing high-rate channels for outdoor or mobile environments [8], [14].

TCP reacts adversely to the high error rate seen on wireless links because it interprets any packet drop as a sign of congestion. Thus, TCP reacts to wireless packet losses by unnecessarily halving its congestion window.

Various strategies have been suggested in the literature to combat this problem [1], [2], [3], [4], [6], [10], [16], [18], and they can be classified into two main classes. The first class shields the sender from the wireless link by using local link-layer retransmissions. The second class, to which WiSE belongs, exposes the reasons for the drops to the end-user thus allowing the sender to recover appropriately.

WiSE Contributions: In addition to serving as a case study of the M-ECN channel, WiSE provides a performance gain higher than ETEN [16] and comparable to ELN [2], without using additional message packets or changing the header structure. In contrast to the current version of ELN, WiSE does not assume route symmetry and works even if a flow traverses multiple wireless hops. Finally, WiSE signaling is simple and can be implemented at a router using 3 lines of C code [24].

B. Overview of the WiSE protocol

The WiSE protocol is implemented through two components: the WiSE-Agent and WiSE-TCP. Located at the router, the WiSE-Agent is a data-link layer module that detects wireless packet errors and signals them to WiSE-TCP over the M-ECN channel. WiSE-TCP is an extension to standard TCP which allows it to decode the "packet-drop type" information sent by the WiSE-Agents along a flow's path. We assume 802.11, where a corrupted packet is discarded at the sending MAC after zero or more retrials. Thus, before dropping a packet because of wireless errors, the router can read the ECN value in the packet.

Below is an overview of how WiSE works (the implementation details are in [24]).

1. The first packet subscribes the flow to the WiSE service using a router alert IP option [13].

2. Whenever 802.11 discards a packet because of a wireless error, the WiSE agent signals this information in a message sent over the M-ECN channel (as described in §V-C). The agent preserves ECN marks from upstream.

3. Receiver decodes the WiSE signal and interprets illegal codewords as congestion. It informs the sender about error losses using a TCP option. (Routers do not access this option).

4. Sender recovers from errors without reducing its rate.

C. Legal & Illegal Codewords

Our objective is to make corruption drops look very different from congestion drops by associating them with a very structured pattern of ECN bits that is unlikely to be created randomly. Whenever a WiSE router (i.e. agent) drops a packet because of corruption, it inserts a 1 in the stream of ECN bits from the flow. Further, the WiSE router preserves the 1's in the ECN field of packets coming from upstream. Thus, if a WiSE router drops a marked packet because of wireless errors, then, in addition to inserting a new mark, the router remembers the ECN mark on the dropped packet and moves it to a later packet from the same flow. The same happens if the router sets a WiSE mark on an already marked packet.

The resulting codewords used by WiSE start with the occurrence of a drop and end with the occurrence of ECN=0, and are characterized by having an equal number of drops and ECN marks. For example, let X represents a drop then X10, X1X10, and XX110 are all legal WiSE codewords. Further, at any point through a legal codeword, the number of drops equals or exceeds the number of ECN marks; e.g. X11X0 is not a legal codeword. Any sequence of drops/marks that doesn't satisfy the above properties is a sign of congestion.

D. Example Scenarios

In the examples, we will only show a flow's ECN bits and represent them as a stream. The ECN field of a dropped packet will be marked with an X. Each packet, and thus the ECN bit associated with it, will be referenced by its *sequence number*.

Example 1: Composability of WiSE Messages
This example (Figure 5) shows that WiSE encoding is accumulative in nature, and thus multiple routers in an ad-hoc network can compose their signals. Packet 5 is corrupted at router WiSE-1 leading to the ECN mark of packet 6 being set to 1. But, packet 6 is corrupted at WiSE-2. So, WiSE-2 has to set ECN=1 on the next two packets to account for the corruption of packet 6 and the fact that packet 6 was already marked. The receiver gets the valid WiSE word XX110, indicating that packets 5 and 6 were dropped because of errors.

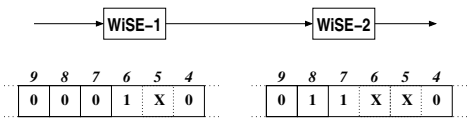

Fig. 5. Composability of WiSE messages

Example 2: WiSE interaction with AQM routers
This example demonstrates that WiSE signaling does not hamper the congestion signals sent by AQM routers. As in

Figure 6, the AQM router marks packet 6 with a 1 to signal congestion. But packet 6 is discarded at the wireless link because of errors. Then the WiSE router not only marks a subsequent packet (packet 7) to signal a wireless drop, but also reads the "ECN" mark on the dropped packet and preserves it by setting the ECN bit on packet 8 to 1. The receiver will recognize X110 as an illegal WiSE codeword (since, marks $\neq$ drops) and convey the congestion signal to the sender.

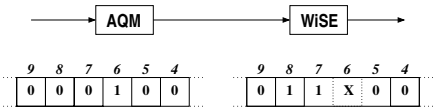

Fig. 6. WiSE interaction with an AQM router

E. WiSE Evaluation

(A) Simulation Environment: Our simulations use *ns-2* [20], which we have extended to include a WiSE-Agent (derived from RED queue) and a WiSE-TCP (a modified NewReno) module. (We have also compared WiSE-SACK with standard SACK, which lead to results similar in nature to those below.) For wireless errors, we use a two-state Markov error model. As per data cited in studies of the wireless channel [19], we set the average burst size of errors in the Markov model to 3 packets. Finally, in all simulations, wired routers use RED queues (max= 2×min = 1/3 buffer), the buffer size is a bandwidth-delay product, packets are 576 bytes, the round trip time (RTT) is 50 ms, and the ECN option is on.

(B) WiSE boosts performance over wireless links: In this experiment, we have a 40 Mb/s wireless link traversed by 700 web sessions with Pareto distributed file sizes having an average of 12 pkts/file; 10 long-lived FTPs and reverse traffic. Average packet error rate is 0.01. Figure 7 shows the boost in total throughput due to WiSE when the traffic mix is similar to that expected in the Internet. Simulations over an ad-hoc network with many wireless links showed similar results [24].

(C) Robustness & TCP-Friendliness: In this experiment, the bottleneck is *wired*. Two flows share a congested 2 Mb/s RED+ECN router. One of these flows also traverses a 10 Mb/s wireless link with error rate 0.01. Figure 8(top) shows that when both flows are NewReno the wireless flow suffers. In contrast, when the wireless flow uses WiSE, as in Figure 8(bottom), its performance significantly improves. The experiment reveals: 1) WiSE works correctly when there are non-WiSE routers along the path; 2) The fact that WiSE is fair to the non-WiSE flow shows that WiSE reacts properly to ECN marks from the congested RED+ECN router.

(D) Comparison of WiSE, NewReno, ELN, and ETEN: Figure 9 shows the average combined throughput of 5 long-lived TCPs traversing a 10 Mb/s wireless link as a function of the packet error rate. It shows that WiSE achieves a performance gain comparable to ELN and better than ETEN, without generating extra messages or requiring dedicated header space. ETEN [16] sends out-of-band ICMP messages to the sender to report the packet corruption rate at the wireless link. ELN [2] assumes route symmetry and uses a dedicated field in the TCP header to report the corruption of a certain packet.

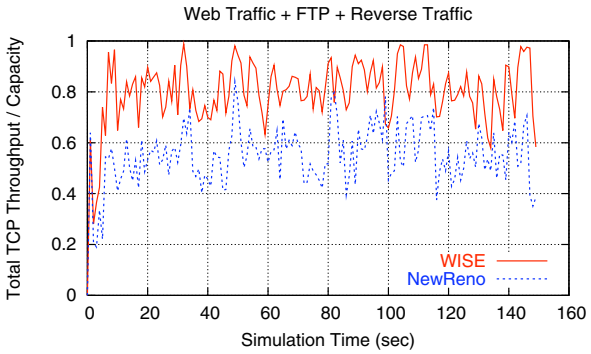

Fig. 7. WiSE boosts performance over wireless links

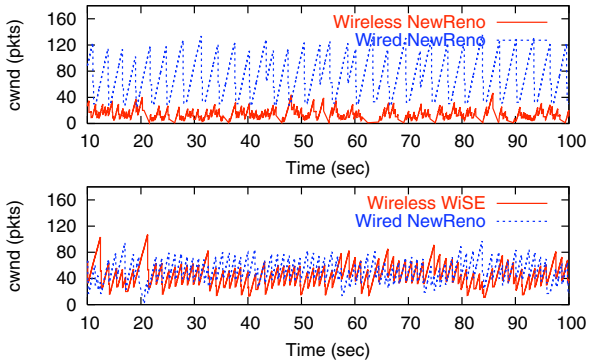

Fig. 8. A wired & wireless flows sharing a congested RED+ECN router; (top) both flows are NewReno; (bottom) the wireless flow is WiSE.

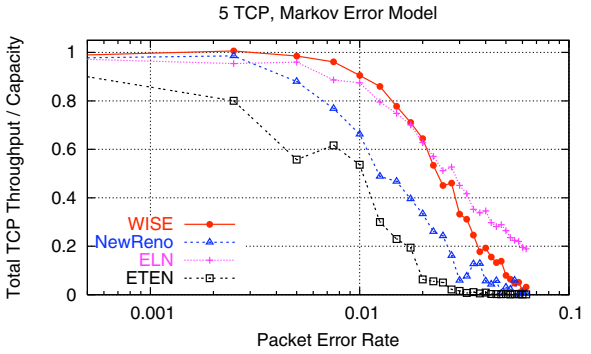

Fig. 9. Comparison of WiSE, NewReno, ELN and ETEN

VI. Some Characteristics of M-ECN and WiSE

A. Security & Interaction with Nonce

To prevent a receiver from ignoring ECN congestion marks, the sender sets a one bit random nonce in each packet. The receiver returns the nonce sum to the sender in the Acks. But when a router marks a packet it clears the nonce, and the receiver would not know the correct nonce sum [22].

Communication on the M-ECN channel works properly when nonce are used. Further, WiSE can use the nonce to prevent a receiver from cheating by making a congestion drop look as if it were a corruption drop. Since the WiSE router can read the value of nonce before marking a packet, it can help the receiver adjust its nonce sum, and prove to the sender the correctness of a WiSE mark. In particular, the WiSE router maintains a single bit called Δ-nonce in which it stores the

cumulative sum of the nonce in all packets that got corrupted or marked during the encoding of a single WiSE codeword. The Δ-nonce is added to the nonce value in the first unmarked packet after the codeword. This causes the receiver's nonce sum to match that of the sender as long as there are no congestion drops/marks. Said differently, the receiver cannot claim that a congestion drop is a corruption drop because, without the help of a WiSE agent, the receiver doesn't know the nonce value of the dropped packet.

B. Robustness to Packet Reordering

Packet reordering does not flip 1's to 0's and thus does not lead to the loss of congestion marks. Reordering is unlikely to create legal M-ECN codewords because codewords are well-structures patterns, which are hard to generate by random reordering. Reordering may destroy an M-ECN codeword causing the loss of the M-ECN signal. For example, in WiSE, reordering could cause X10 $\rightarrow$ X01; thus TCP will mistake the error drop to be a congestion drop, which is what would have happened anyways if we didn't use WiSE. Further, since reordering occurs between outstanding packets, the reordered WiSE marks don't do any harm because TCP reacts to only one congestion signal per congestion window.

C. Partial Deployment

The M-ECN channel is amenable to partial deployment because it is completely transparent to routers and flows which do not participate in the communication. For example, in §V-E, we have shown that WiSE works well when the path traverses both WiSE and non-WiSE routers. We have also shown that WiSE functions correctly independently of whether the bottleneck is a wired or wireless link. Finally, we have also shown that WiSE interacts properly with non-WiSE TCPs.

VII. Conclusion and future work

We presented M-ECN, a novel mechanism for network-to-transport signaling, which generates no extra packets and requires no specific bits in the packet header. M-ECN relies on opportunistic signaling which sneaks messages in the stream of ECN bits, while giving priority to ECN congestion signaling over other information. As an example application of M-ECN, we have developed WiSE (for Wireless Signaling via ECN) and shown that it boosts TCP performance over wireless channels while reacting properly to ECN congestion marks.

Currently we are looking at using M-ECN to help flows with a very large end-to-end pipe to quickly acquire spare bandwidth and refine their decrease signal. We believe that the number of gigabit flows over a path will be small and thus, a per-gigabit-flow state might be acceptable. In contrast to WiSE codewords, which rely on the position of ECN marks, the codewords we devise for this application use fixed length frames with a preamble [24]. M-ECN is attractive for this application because it doesn't require all routers along the path to understand the new signaling. When legacy routers along the path generate ECN congestion marks/drops, they create an illegal M-ECN codeword. This causes the sender to revert to standard TCP and to halve its congestion window. More work is needed to study potential M-ECN applications and the applicability of opportunistic signaling to header fields other than ECN.

Acknowledgment

Dina Katabi was supported by NSF under award No. ANI-0335256. Balaji Prabhakar was supported in part by the NSF grant ANI-9985446.

References

[1] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. In *Proc. 15th International Conference on Distributed Computing Systems (ICDCS)*, May 1995.

[2] H. Balakrishnan and R. Katz. Explicit loss notification and wireless web performance. In *nms.lcs.mit.edu/papers/globecom98/*, Nov. 1998.

[3] H. Balakrishnan, S. Seshan, and R. Katz. Improving reliable transport and handoff performance in cellular wireless networks. In *ACM Wireless Networks*, Dec. 1995.

[4] D. Barman and I. Matta. Effectiveness of loss labeling in improving TCP performance in wired/wireless networks. Technical report, Boston University, 2002.

[5] S. M. Bellovin. Security problems in the TCP/IP protocol suite. In *Computer Communications Review 2:19, pp. 32-48*, Apr. 1989.

[6] S. Biaz and N. H. Vaidya. Distinguishing congestion losses from wireless transmission losses: A negative result. *Seventh International Conference on Computer Communications and Networks (IC3N)*, 1998.

[7] T. M. Cover and J. A. Thomas. Elements of information theory. In *Wiley Series in Telecommunications*, 1991.

[8] D. C. Cox. Wireless personal communications: what is it? In *IEEE Personal Communications*, volume 2 Issue:2, pp 20-35, Apr. 1995.

[9] S. Dawkins, C. E. Williams, and A. E. Yegin. Problem statement for Triggers for Transport(TRIGTRAN). In *http://www.ietf.org/internet-drafts/draft-dawkins-trigtran-probstmt-00.txt*, Oct. 2002.

[10] D. A. Eckhardt and P. Steenkiste. Improving wireless LAN performance via adaptive local error control. *Proceedings of IEEE ICNP '98*, 1998.

[11] Alternate proposals for the ECN field in IP, or for other ECN-like semantics. http://www.icir.org/floyd/ecn.html.

[12] S. W. Golomb. The limiting behavior of the z-channel. In *IEEE Transactions on Information Theory*, volume IT-26 No. 3, May 1980.

[13] D. Katz. IP Router Alert Option. *RFC 2113*, Feb. 1997.

[14] R. H. Katz. Adaptation and mobility in wireless information systems. In *IEEE Personal Communications*, volume 1 Issue:1, pp 6-17, Apr. 1994.

[15] S. Kent and R. Atkinson. Security architecture for the internet protocol. *RFC 2401*, Nov. 1998.

[16] R. Krishnan, M. Allman, C. Partridge, and J. P. G. Sterbenz. Explicit transport error notification for error-prone wireless and satellite networks. In *BBN Technical Report No. 8333*, Mar. 2002.

[17] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. In *Computer Communications Review 32:3, pp. 62-73*, July 2002.

[18] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang. TCP Westwood: Bandwidth estimation for enhanced transport over wireless links. In *Mobile Computing and Networking*, pages 287–297, 2001.

[19] G. T. Nguyen, B. Noble, R. H. Katz, and M. Satyanarayanan. A trace-based approach for modeling wireless channel behavior. In *Proc. of the 1996 Winter Simulation Conference*, Dec. 1996.

[20] The network simulator ns-2. http://www.isi.edu/nsnam/ns.

[21] P. Oprisan and B. Bose. ARQ in Optical Networks. In *Proc. Pacific Rim International Symposium on Dependable Computing*, Dec. 2001.

[22] K. K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. *RFC 3168*, Sept. 2001.

[23] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proc. of ACM SIGCOMM*, 2000.

[24] M. Sharma, D. Katabi, R. Pan, and B. Prabhakar. A General Multiplexed ECN Channel and its use for Wireless Loss Notification. Technical Report 896, MIT, 2003, http://nms.lcs.mit.edu/projects/mecn/.

[25] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queuing: A scalable architecture to approximate fair bandwidth allocations in high speed networks. In *Proc. of ACM SIGCOMM*, Aug. 1998.