WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2002; **2**:37–57 (DOI: 10.1002/wcm.32)

Syndrome: a light-weight approach to improving TCP performance in mobile wireless networks

Wei-Peng Chen, Yung-Ching Hsiao,
Jennifer C. Hou,*,† Ye Ge and Michael P. Fitz

Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210-1272,
U.S.A.

Summary

It is well known that the performance of TCP deteriorates in a mobile wireless environment. This is due to the fact that although the majority of packet losses are results of transmission errors over the wireless links, TCP senders still take packet loss as an indication of congestion, and adjust their congestion windows according to the additive increase and multiplicative decrease (AIMD) algorithm. As a result, the throughput attained by TCP connections in the wireless environment is much less than it should be. The key problem that leads to the performance degradation is that TCP senders are unable to distinguish whether packet loss is a result of congestion in the wireline network or transmission errors on the wireless links.

In this paper, we propose a light-weight approach, called *syndrome*, to improving TCP performance in mobile wireless environments. In *syndrome*, the BS simply counts, for each TCP connection, the number of packets that it relays to the destination host so far, and attaches this number in the TCP header. Based on the combination of the TCP sequence number and the BS-attached number and a solid theoretical base, the destination host will be able to tell where (on the wireline or wireless networks) packet loss (if any) occurs, and notify TCP senders (via explicit loss notification, ELN) to take appropriate actions. If packet loss is a result of transmission errors on the wireless link, the sender does not have to reduce its congestion window.

Syndrome is grounded on a rigorous, analytic foundation, does not require the base station to buffer packets or keep an enormous amount of states, and can be easily incorporated into the current protocol stack as a software patch. Through

*Correspondence to: Jennifer C. Hou, Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210-1272, U.S.A.
†E-mail: jhou@ee.eng.ohio-state.edu

Contract/grant sponsor: NSF; contract/grant number: ANI-989018.

simulation studies in *ns-2* (UCB, LBNL, VINT network simulator, http://www-mash.cs.berkeley.edu/ns/), we also show that *syndrome* significantly improves the TCP performance in wireless environments and the performance gain is comparable to the heavy-weight *SNOOP* approach (either with local retransmission or with ELN) that requires the base station to buffer, in the worst case, a window worth of packets or states.

KEY WORDS
TCP performance
mobile wireless networks
syndrome

Published online: 5 December 2001

1. Introduction

Mobile wireless networking has become an important technique for supporting emerging personal communication services. A mobile wireless network is composed of a number of wireless LANs (also referred to as *cells*). Each wireless LAN is composed of a base station (BS) and a variable number of mobile hosts (MHs). BS's are connected to one another to form a wired point-to-point backbone network. Due to the different characteristics between wireline and wireless networks, many protocols originally designed for wireline networks have to be adapted for wireless networks. For example, Mobile IP was proposed, and becomes the current standard, for IP mobility support.

TCP is the transport layer of the Internet protocol suite and is intended to provide reliable byte stream transport over an underlying unreliable network. The TCP congestion control mechanism, in particular the additive increase and multiplicative decrease (AIMD) algorithm, has been the major reason for keeping the Internet from congestion collapse. However, it is well known that the TCP performance in terms of throughput attainable by TCP connections deteriorates in mobile wireless environments. There are two major reasons that account for the performance degradation, both of which are attributed to the characteristics of wireless environments [1]. First, as wireless links are usually subject to higher bit error rates as compared to wireline networks, the majority of packet losses in wireless environments are results of transmission errors over wireless links. However, TCP senders take packet loss as an indication of congestion and reduce their congestion windows according to the AIMD algorithm. Second, during the period of hand-offs, the end-to-end connection cannot be maintained and packets are being dropped. This is again erroneously taken by TCP senders as an indication of congestion.

Many research efforts have been made to adapt TCP to the mobile wireless environment. They can be classified, according to the types of wireless packet losses they intend to dealing with, into (i) approaches that deal with losses caused by hand-off and mobility and (ii) those that deal with losses caused by the high bit error rate (BER). *M-TCP* [2], *Freeze-TCP* [3], and the approach proposed in Reference [1] are perhaps the most notable work in the first category. Approaches in the second category are exemplified by *Indirect-TCP* [4], *MTCP* [5], delayed duplicate ACKs [6], *SNOOP* [7], and *ELN-based SNOOP* [8]. We will provide a detailed summary of, and a comparison among, these approaches in Section 2.

In this paper, we propose an alternative, lightweight approach, called *syndrome*, to deal with wireless packet losses. In *syndrome*, the BS simply counts, for each TCP connection, the number of packets that it relays to the destination host so far, and attaches this number[‡] in the TCP header. Based on

[‡] Similar to the TCP sequence number, this number can be expressed in a mod n manner.

the combination of the TCP sequence number and the BS attached number which we call *syndrome*, and a set of propositions/lemmas analytically derived in the paper, the destination host will be able to tell where (on the wireline or wireless networks) packet loss (if any) occurs, and notify TCP senders (via explicit loss notification, ELN) to take appropriate actions.

Syndrome possesses many desirable features. First, the BS does not have to buffer packets or maintain an enormous amount of states for each TCP connection. Second, the modification at end hosts is minor (a few lines of code change) and can be packaged up as a patch in the software distribution. Moreover, the modification is backward compatible with original TCP; end hosts without *syndrome* cannot benefit from the performance improvement, but can otherwise coexist with *syndrome* hosts. Third, *syndrome* maintains the end-to-end semantics. Fourth, although *syndrome* is not designed to deal with packet losses caused by hand-offs, it can resume operations and take effect immediately after hand-off, when packets arrive at the new BS. Through simulation studies in *ns-2*, we show that *syndrome* significantly improves the TCP performance over wireless links and the performance gain is comparable to the heavy-weight *SNOOP* approach (either with local retransmission or with ELN) that requires the base station to buffer, in the worst case, a window worth of packets or states.

The rest of the paper is organized as follows. We provide a taxonomy of existing approaches and discuss their advantages and disadvantages in Section 2. For clarity of presentation, we present in Section 3 a base approach that operates under the (unrealistic) assumption that packets may be lost but are neither out of order nor duplicate. In Section 4 we extend the base approach to accommodate the cases in which packets may be lost or out of order on the wireline network, and may be lost on the wireless link. Following that, we investigate in Section 5 the impact of duplicate packets on the performance of *syndrome*. We present simulation results in Section 6 and conclude the paper with future work in Section 7.

2. Related Work

As mentioned in Section 1, TCP enhancements in the wireless environment can be classified into (i) approaches that deal with losses caused by hand-off and mobility and (ii) approaches that deal with losses caused by the high bit error rate:

2.1. Approaches in the first category

M-TCP [2], *Freeze-TCP* [3], and the approach proposed in Reference [1] receive the most attention in the first category. In Reference [1], the authors observed that as numerous packets are lost during hand-off, the TCP sender may eventually shut down the window and start a timer. If the timeout period is larger than the hand-off period, the MH does not receive any data until the timeout period is over. To reduce the waiting period after hand-off, the authors in Reference [1] proposed that an MH retransmits three duplicate ACKs immediately after the hand-off for the last data segment it received prior to the disconnection.

In *M-TCP* [2], when the BS detects a disconnection, it relays back an ACK to the sender, but with the receiver advertisement window size set to 0. The consequence is that upon receipt of this ACK, the TCP sender freezes all retransmission timers and enters a persist mode. In the persist mode, the sender does not shrink the congestion window so that the slow-start phase can be avoided. When the connection recovers and/or the hand-off is completed, the BS then relays an ACK with non-zero window size to resume the transmission. *Freeze-TCP* [3] moves the onus of signaling an impending disconnection from the BS to the MH (that is, the MH sends the ACK with a zero receiver advertisement window size). As discussed in Reference [3], with this subtle change, *Freeze-TCP* eliminates several drawbacks inherited from putting intelligence in the BS.

2.2. Approaches in the second category

Approaches in the second category can be further classified into three sub-categories [8, 9]:

1. Split connections (*Indirect-TCP* [4], *MTCP* [5], and those proposed in References [10, 11]): split-connection approach separate each connection into two TCP connections, one between a fixed host (FH) and a BS, and the other between a BS and a MH. *Indirect-TCP* [4] used standard TCP for these two connections while *MTCP* [5] uses, instead of TCP, a selective repeat protocol (SRP) on top of UDP for the wireless link. The major advantage of these approaches is that a customized transport protocol can be used to optimize the performance over the wireless link. On the other hand, these approaches suffer from the following drawbacks: they usually require large memory space

at the BS's in order to store/forward packets and maintain hard states for each connection. When an MH moves from one domain to another, the entire 'state' of the connection (including packets buffered for local retransmission) must also be transferred. Also, dividing each connection into two connections violates the end-to-end semantics.

2. TCP-aware link layer retransmission (*SNOOP* [7] and delayed duplicate ACKs [6]): In Reference [6], duplicate ACKs are delayed at the BS and not relayed back to the FH, in order to allow special local retransmission on the wireless link. *SNOOP* recovers wireless errors by buffering at the BS all the packets destined for MHs and locally retransmitting packets at the link level upon receipt of duplicate acknowledgments or upon timeout (i.e., the *SNOOP* module implements a round-trip timer and retransmits unacknowledged packets based on this timer). In addition, *SNOOP* prevents the sender from fast retransmission by intercepting and discarding duplicate acknowledgments for packets lost on the wireless link. *SNOOP* does not require code changes either at FHs or at MHs; only the protocol stack at BS's has to be changed. It retains the end-to-end semantics. However, the major drawbacks of *SNOOP* are: (i) it requires significant amount of memory space and processing capability at the BS in order to keep states and to buffer packets for local retransmission; (ii) it suffers from the same problem as split-connection approaches in the case of hand-offs. The first drawback is especially serious, as the BS has to keep states and maintain a *window worth* of packets for potentially thousands of TCP connections.
3. Explicit loss notification (ELN) [8, 12–14]: An ELN bit has been proposed in Reference [15], included in the packet header, and set (by certain network entity) to explicitly notify the sender of packet loss. To alleviate the workload at the BS, an ELN-based version of *SNOOP* [8] was proposed in which the BS does not buffer packets, but instead records sequence numbers of packets lost in the wireline network in a TCP congestion window. Upon receipt of duplicate acknowledgments for packets lost on the wireless link, the BS then sets the ELN bit in the packet header before forwarding them to the sender. When a sender receives duplicate acknowledgments with the ELN bit set, it does not reduce its congestion window size. Use of ELN mitigates the BS loading problem, but each BS still has to maintain, in the worst case, a window worth of sequence numbers and monitors duplicate ACKs returned by a MH. Also, if the state kept at the old BS is not available to a new BS after hand-off, the hand-off problem remains. Another ELN-inherited drawback is that both end hosts and BS's have to be modified. In [12–14] the author proposed a similar method in which the sequence numbers of all packets need to be re-ordered.

As mentioned in References [3, 8, 9], the criteria that should be used to assess TCP enhancements are (i) whether or not changes are required at intermediate nodes (e.g., BS's) or FHs in the wireline network; (ii) whether or not intermediate nodes have to keep states for each connection (and may hence become the performance bottleneck); (iii) whether or not the end-to-end semantics are maintained; (iv) capability to handle IPSEC-encrypted[§] traffic; (v) capability to deal with packet losses caused by high wireless BER; (vi) capability to handle hand-offs. We give an assessment of all the existing approaches (and *syndrome*) with respect to the above criteria in Table 1.

As shown in Table 1, all existing approaches in the second category require intermediate nodes (e.g., BS's) to buffer state/packets for each TCP connection. In particular, I-TCP, MTCP, and SNOOP requires the BS to buffer, in the worst case, a window worth of packets and states. The advantage of buffering packets and duplicating the functions performed at a TCP source at the BS is that packets lost on the final wireless link can be locally transmitted, instead of being transmitted all the way from the sender. However, as a BS may potentially handle thousands of TCP connections (considering wireless webs are becoming one of the most popular wireless services), this inevitably imposes performance and scalability problems. Also, the resources available in the wireline network are usually more abundant than those on the wireless link, and hence retransmitting all the way from the source may not be a performance concern. Although *syndrome* requires code change at BS's and FHs, the modification is quite minor (a few lines of code change as outlined in Figure 1) and can be packaged up as a patch.

[§] In IPSEC, the entire IP payload is encrypted so the intermediate nodes may not be able to know the type of traffic being carried in the payload, if itself is not made a party to the security association between the FH and MH.

Table 1. A comparison of existing TCP enhancement approaches with respect to different criteria.

	Approaches for pkt losses caused by disconnection		Approaches for pkt losses caused by high wireless BER					
	M-TCP	Freeze-TCP	I-TCP	MTCP	DDA*	SNOOP	ELN-based SNOOP	Syndrome
Code change at BS or FHs?	yes	no	yes	yes	yes	yes	yes	minor
BS keeps state?	yes	no	yes	yes	yes	yes	yes	no‡
Retransmission all the way from the source?	yes	yes	no	no	yes	no	yes	yes
Handle encrypted traffic?	no†	yes	no†	no†	no†	no†	no†	no†
Can ACKs be routed along a different path on which data is routed?	no	yes	yes	yes	no	no	no	yes
Handle high BER?	some what	no	yes	yes	no	yes	yes	yes
Handle hand-offs?	yes	yes	no+	no+	no+	no+	no+	no#

* DDA: delayed duplicate ACKs.
‡ *Syndrome* has to keep, for each TCP connection, the number of packets it has relayed to the destination.
† If the intermediate node is made a party to the security association between the FH and the MH and the IPSEC tunneling mode is terminated at the intermediate node, these approach may handle encrypted traffic.
+ If the entire state (including buffered packets) can be moved to the new BS, these approaches can handle hand-offs (expensively).
\# The performance of *Syndrome* degrades during hand-off, but the operations resume immediately after hand-off when packets start to arrive at the new BS.

3. The Base Algorithm

As mentioned in Section 1, the key issue to improving the TCP performance in a mobile wireless environment is to distinguish packet loss which results from congestion in the wireline network from that which results from transmission errors over the wireless link. We propose the following approach: a BS simply counts, for each TCP connection, the number of packets that it relays to the destination host so far, and attaches this number in the (option field of the) TCP header. The attached number could be made 32 bit long and wrapped around in exactly the same manner TCP sequence numbers do. Also, if a packet is dropped at the BS due to buffer overflow, it is treated as being lost in the wireline network. Now the question is how the destination host infers, based on the *syndrome*—the combination of the TCP sequence number and the BS attached number—the cause of packet loss. In what follows, we establish the theoretical base, under the assumption that packets arriving at the BS are neither out of order nor duplicate. (We will relax these assumptions in Sections 4 and 5.)

Without loss of generality, we describe the proposed approach in the packet mode (as the byte-mode version of the proposed approach can be straightforwardly devised). Also, to ease discussion we denote

- (SN_i, AN_i): as the combination of the TCP sequence number (SN) and the BS-attached number (AN) as observed by a destination host.
- G_n: as a gap of size n in the attached number space if the attached numbers AN_i and AN_{i+1} of two packets consecutively received at the destination host differ by $n+1$ (i.e., if the numbers carried in two consecutive packets differ by 1, there is no gap).
  The definition of a gap in the sequence number space is, however, somewhat different. Let *max_seq_seen* denote the maximum sequence number ever received at the time of receiving the packet with SN_i and $n \triangleq \max\{SN_{i+1} - max\_seq\_seen - 1, 0\}$. If $n > 0$, a gap of size n exists; otherwise, no gap exists. Figure 2 gives an example that illustrates the definition of a gap in the sequence space.
- the *syndrome*, (G_n, G_m), $m \geq 0$: as the gaps in the sequence number space and the attached number space, respectively, of two packets consecutively received by a destination host. Note that the syndrome is defined from the *perspective of a destination host at the time when it receives the second packet of the two consecutive ones.*

The destination host deduces, based on (G_n, G_m), where packet loss (if any) occurs. The following proposition establishes the theoretical base of *syndrome*:

Proposition 1. *Under the assumption that packets that arrive at the BS are neither out of order nor duplicate (but can be lost), for connections from FHs*

Case 1. **FH → BS → MH**

BS: Upon receipt of a data packet $pkt(j)$ of connection i, where $pkt(j)$ is a TCP segment with sequence number j, the BS does the following:

```
if (pkt(j) is the first packet received for connection i) {
    C_i ← 0; // count from zero
else
    C_i ++;
attach C_i to the TCP header of pkt(j), and forward pkt(j, C_i) to the destination MH;
}
```

MH: Upon receipt of a data packet $pkt(j, C_i)$, of connection i, the MH does the following ($credit \leftarrow 0$ at system initialization):

```
compute the syndrome;
if (syndrome = = (G_n, G_m), n ≥ m > 0) {
    credit ← credit + m;
    record the indices k according to Equation (1) ; }
if ((ack ∈ a set of duplicate acks with index k) && (credit > 0))
    send ack with ELN ← 1;
else if ((ack is not a duplicate) && (previous ack ∈ a set of 3 or more duplicate acks with ELN = 1)) {
    credit ← credit −1;
    send ack with ELN ← 0;
}
else
    send ack with ELN ← 0;
```

FH: Upon receipt of three duplicate acks for connection i, the FH does the following:

```
if (ack.ELN is set)
    retransmit the lost packet only, but do not reduce the congestion window;
else
    act as TCP does;
```

Case 2. **MH → BS → fixed host**

BS: The procedure taken by the BS is the same as in Case 1.

FH: Upon receipt of a data packet of $pkt(j, C_i)$, of connection i, the FH does the following ($credit \leftarrow 0$ at system initialization):

```
compute the syndrome;
if (syndrome = = (G_n, G_m), n > m ≥ 0) {
    credit ← credit + (n − m);
    record the indices k for setting ELN bits; }
if ((ack ∈ a set of duplicate acks with index k) && (credit > 0))
    send ack with ELN ← 1;
else if ((ack is not a duplicate) && (previous ack ∈ a set of 3 or more duplicate acks with ELN = 1)) {
    credit ← credit −1;
    send ack with ELN ← 0;
}
else
    send ack with ELN ← 0;
```

MH: The procedure taken by the MH is the same as that taken by the FH in Case 1.

Fig. 1. The base algorithm of *syndrome*.

	Sequence Number	Attached Number	Max SN Seen	SN in Ack
A gap of size 1	11	1	11	12
	13	2	13	12
	14	3	14	12
	15	4	15	12
Not a gap because $12-16=-4<0$	16	5	16	12
	12	6	16	17
Not a gap because max_seq_seen = 16 and $n = 17 - 16 - 1 = 0$	17	7	17	18
	18	8	18	19

Fig. 2. An example that illustrates the definition of a gap in the sequence number space.

to MHs, the syndrome for two packets consecutively received by the MH is

(a) (G_0, G_0)*, if and only if no packet loss occurs.*
(b) (G_m, G_m), $m \geq 0$*, if and only if m packets are lost on the wireless link and no packet is lost in the wireline network.*
(c) (G_n, G_m), $n \geq m \geq 0$*, if and only if* $n - m$ *packets are lost in the wireline network and m packets are lost on the wireless link.*

Proof. Under the assumption of orderly packet delivery on the wireless link (an MH receives packets in the same order in which they are sent at the BS), if a packet is lost on the wireless link, it will necessarily create a gap in the attached number space and vice versa. Hence, the size of the gap, G_m, in the attached number space is equal to the number of packets lost on the wireless link. Similar arguments can be made for inferring, based on the size of G_n, the number of packets lost in the wireline network. □

By Proposition 1, we know that under the simplifying assumption, among the n packet losses observed in a syndrome (G_n, G_m), $n \geq m \geq 0$, m of them occur on the wireless link and should not lead to window reduction. That is, the ELN bits should be set by the MH for m sets of duplicate acknowledgments, one for each packet lost on the wireless link. However, since the $n - m$ wireline packet losses and the m wireless losses may intervene with one another, and the destination host cannot tell (without extra information) the exact order in which they occur, we devise *syndrome* to give 'credits' to the kth observed packet loss, where

$$k = \begin{cases} 1, 3, \ldots, 2m - 1, & \text{if } 2m \leq n \\ 1, 3, \ldots, 2(n - m) - 1, 2(n - m) + 1, & \\ \quad 2(n - m) + 2, \ldots, n, & \text{if } m \leq n < 2m \end{cases} \tag{1}$$

and set the ELN bits for the corresponding duplicate acknowledgments. Similarly, for connections from MHs to FHs, the following proposition establishes the theoretical base.

Proposition 2. *Under the assumption that packets that arrive at the BS are neither out of order nor duplicate (but can be lost), for connections from MHs to FHs, the syndrome for two packets consecutively received by the MH is*

(a) (G_0, G_0)*, if and only if no packet loss occurs.*
(b) (G_m, G_m), $m \geq 0$*, if and only if m packets are lost in the wireline network and no packet is lost on the wireless link.*
(c) (G_n, G_m), $n \geq m \geq 0$*, if and only if* $n - m$ *packets are lost on the wireless link and m packets are lost in the wireline network.*

The procedures taken by the BS, by the MH, and by the FH are listed in Figure 1. In the case that both the sender and the destination hosts are mobile hosts and their BS's are connected via a wireline network, two 'instances' of syndrome can be applied to infer errors from the sender to its base station and errors from the second base station to the destination host, respectively. The following example illustrates how the base algorithm operates.

Example 1 Consider the connection from a FH to a MH. Suppose packets 12 and 13 are lost on the wireless link, and packet 16 is lost in the wireline network. The sequence in which packets arrive at the BS is given in Figure 3(a), and the syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 3(b).

As shown in Figure 3(b), duplicate acknowledgments for packets 12 and 16 are sent with ELN = 1 and hence will not lead to window reduction, while those sent for packet 13 are sent with ELN = 0 and will lead to window reduction. The total number of times the congestion window is halved is 1, accounting for the fact that only one packet is lost on the wireline network.

Note also that as illustrated in Figure 3, as the destination host does not know which packet is lost on the wireless link and should be given the 'credit,' the time at which the sender reduces its congestion window may be slightly offset when the two types of packet losses (wireline or wireless) interleave with each other.

4. The Extended Algorithm

In this section, we discuss how we extend the base approach to accommodate the condition that packets may be lost in the wireline/wireless network and, in addition, re-ordered on the wireline network. We still assume that packets can not be reordered on the wireless link (i.e., packets received at the MH are in the order in which they are sent by the BS). As the wireless link consists of only one hop, the assumption is valid for wireless links with the MAC-level retransmission capability. Also, we assume that

SN_i	AN_i
11	1
12	2
13	3
14	4
15	5
17	6
18	7
12‡	8
19	9
20	10
21	11
22	12
13‡	13
23	14
24	15
25	16
26	17
16‡	18
27	19
...	

(a)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}			0
14	4	A_{12}	(G_2, G_2)	1	2
15	5	A_{12}	(G_0, G_0)	1	2
17	6	A_{12}	(G_1, G_0)	1	2
18	7	A_{12}	(G_0, G_0)	1	2
12‡	8	A_{13}	(G_0, G_0)	0	1
19	9	A_{13}	(G_0, G_0)	0	1
20	10	A_{13}	(G_0, G_0)	0	1
21	11	A_{13}	(G_0, G_0)	0	1
22	12	A_{13}	(G_0, G_0)	0	1
13‡	13	A_{16}	(G_0, G_0)	0	1
23	14	A_{16}	(G_0, G_0)	1	1
24	15	A_{16}	(G_0, G_0)	1	1
25	16	A_{16}	(G_0, G_0)	1	1
26	17	A_{16}	(G_0, G_0)	1	1
16‡	18	A_{27}	(G_0, G_0)	0	0
27	19	A_{28}	(G_0, G_0)	0	0
			...		

(b)

‡ retransmitted packet

Fig. 3. An example that shows how the base approach works.

there exist no duplicate packets, due to, for example, packet retransmission. In Section 5, we will discuss the impact of existence of duplicate packets on the performance of *syndrome*.

4.1. Extended algorithm for $FH \rightarrow BS \rightarrow MH$ connections

We consider connections from FHs to MHs. Let O, L, $-$ denote the out-of-order packet event, the packet loss event, and the no error event, respectively, and (X, Y), *where* $X, Y \in \{O, L, -\}$, denote the combined events (as perceived by the destination host) in the wireline network and on the wireless link. For example, $(O + L, -)$ denotes that packets are re-ordered and lost in the wireline network, but incur no error on the wireless link. Note that error events are defined *in the view of the destination host at the time when the host receives the second packet of two consecutively received packets*. We first establish the theoretical base by analyzing, for each syndrome, the possible events that may cause the syndrome. Then, we determine how *credits* should be adjusted and how ELN bits should be set under each syndrome.

4.1.1. *Cause of syndrome* (G_0, G_0)

LEMMA 1 *If the syndrome for two packets consecutively received by the MH is* (X, G_0)*, where X denotes 'don't care,' then no packet loss occurs (between the two packets) on the wireless link.*

Proof. Under the assumption of orderly packet delivery on the wireless link, if a packet is lost on the wireless link, it will necessarily create a gap in the attached number space. Thus, if there is no gap in the attached number space, no packet loss occurs on the wireless link. □

Proposition 3. *If the syndrome for two packets consecutively received by the MH is* (G_0, G_0)*, then the possible error events are* $(-, -)$ *or* $(O, -)$.

Proof. By Lemma 1, we know that no packet loss occurs on the wireless link. This implies that the syndrome for the two packets consecutively sent by the BS is also (G_0, G_0). The fact that no gap in the sequence number space is observed at the BS implies that (i) no packet loss occurs or (ii) an out of order event occurs in the wireline network. The latter occurs when the SN of the second packet is smaller than that of the first one. □

4.1.2. *Causes of syndrome* $(G_n, G_0), n > 0$

Proposition 4. *If the syndrome for two packets (with sequence numbers* SN_i *and* SN_{i+1}*) consecutively received by the mobile host is* $(G_n, G_0), n > 0$*, the possible error events are* $(L, -)$*,* $(O, -)$*, or* $(L + O, -)$.

Proof. By Lemma 1, we know that no packet loss occurs on the wireless link. This implies that the syndrome for the two packets consecutively sent by the BS is also (G_n, G_0), i.e., there is a gap of size n in the sequence number space. Several possible scenarios may lead to this gap: packets with sequence numbers larger than SN_i and smaller than SN_{i+1} are either lost, or arrive out of order at the BS (before SN_i or after SN_{i+1}), or a combination thereof. Hence, the possible error events are thus $(L, -)$, $(O, -)$, or $(L + O, -)$. □

As indicated in Proposition 4, among all the possible error events that lead to this syndrome, none of them incur packet loss on the wireless link. Hence, no credit is given and the TCP sender should respond to packet loss in the regular AIMD manner.

4.1.3. Cause of syndrome (G_0, G_m), $m > 0$

LEMMA 2 *If the syndrome for two packets consecutively received by the MH is (X, G_m), $m > 0$, where X denotes 'don't care,' exactly m packet losses occur on the wireless link.*

Proof. The proof is similar to Lemma 1, and is thus omitted. □

Proposition 5. *If the syndrome for two packets (with sequence numbers SN_i and SN_{i+1}) consecutively received by the MH is (G_0, G_m), $m > 0$, then the possible error event is (O, L), or $(O + L, L)$.*

Proof. By Lemma 2, we know that there are m packet losses on the wireless link. The fact that there exists no gap in the sequence number space implies that (i) at least $k \geq m$ packets are transmitted out of order over the wireline network, and (ii) exactly m packets with sequence numbers either smaller than SN_i or larger than SN_{i+1} arrive at the BS after the packet with SN_i but before the packet with SN_{i+1}, and are subsequently lost on the wireless link. The other $k - m$ packets are lost on the wireline network and do not arrive at the BS. □

As indicated in Proposition 5, exactly m packet losses occur on the wireless link under this syndrome. Hence, we augment the base approach (in which the syndrome (G_0, G_m), $m > 0$ is not possible) and assign m credits (i.e., set the ELN bits alternatively in m sets of duplicate acknowledgment). The following example illustrates how credits are given and ELN bits are set under the (G_0, G_m), $m > 0$ syndrome.

Example 2 Suppose packet 13 arrives out of order at the BS (the order in which packets arrive at the BS is given in Figure 4(a)) and is lost on the wireless link (the (O, L) event). The syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 4(b).

4.1.4. Causes of syndrome (G_n, G_m), $0 < n < m$

Proposition 6. *If the syndrome for two packets (with sequence numbers SN_i and SN_{i+1}) consecutively received by the MH is (G_n, G_m), $0 < n < m$, then the possible error events are (O, L) or $(O + L, L)$.*

Proof. By Lemma 2, we know there are m packet losses on the wireless link. Also, if no out-of-order events occur in the wireline network, then by Proposition 1, we know that the only possible syndromes are those with $n \geq m \geq 0$. Hence, at least one packet arrives out of order (between SN_i and SN_{i+1}) at the BS.

In addition, the fact that $n > 0$ (there is a gap in the sequence number space) implies that at least one packet whose sequence number falls in (SN_i, SN_{i+1})

SN_i	AN_i
11	1
13	2
12	3
14	4
15	5
16	6
17	7
13‡	8
18	9
19	10
20	11
21	12
22	13
...	

(a)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
12	3	A_{13}	(G_0, G_1)	0	1
14	4	A_{13}	(G_1, G_0)	1	1
15	5	A_{13}	(G_0, G_0)	1	1
16	6	A_{13}	(G_0, G_0)	1	1
17	7	A_{13}	(G_0, G_0)	1	1
13‡	8	A_{18}	(G_0, G_0)	0	0
18	9	A_{19}	(G_0, G_0)	0	0
19	10	A_{20}	(G_0, G_0)	0	0
20	11	A_{21}	(G_0, G_0)	0	0
21	12	A_{22}	(G_0, G_0)	0	0
22	13	A_{23}	(G_0, G_0)	0	0
			...		

(b)

Fig. 4. Examples that give the (G_0, G_m), $m > 0$ syndrome.

is lost or delayed (but not early[¶]). Hence, the possible error events are (O, L) and $(O + L, L)$. □

Similar to how we handle the $(G_0, G_m), m > 0$ syndrome, we augment the base approach (in which the syndrome (G_n, G_m), $0 < n < m$ is not possible) and assign m credits. The following example illustrates how credits are given and ELN bits are set under the $(G_n, G_m), 0 < n < m$ syndrome.

Example 3 Consider the following two scenarios:

Scenario 1: ((O, L) event) Packet 17 arrives out of order at the BS (the order in which packets arrive at the BS is given in Figure 5(a)) and both packet 12 and 17 are lost on the wireless link. The syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 5(b).

Scenario 2: ($(O + L, L)$ event) Packet 12 is lost in the wireline network, and packets 17 and 18 arrive out of order at the BS (the order in which packets arrive at the BS is given in Figure 5(c)) and are both lost on the wireless link. The syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 5(d).

As shown in Figure 5(b) and (d), the sequence numbers received and the syndromes observed before sequence number 12 are indistinguishable under the two scenarios, and the destination host cannot really tell whether packet 12 is lost in the wireline

[¶] Whose sequence number falls in (SN_i, SN_{i+1}) arrived early (i.e., before SN_i), *max_seq_seen* would be set to the maximum sequence number of these packets, and hence by the definition of a gap earlier, this will not be considered as a gap.

SN_i	AN_i
11	1
12	2
17	3
13	4
14	5
15	6
16	7
12‡	8
18	9
19	10
20	11
21	12
17‡	13
22	14
...	

(a)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
13	4	A_{12}	(G_1, G_2)	1	2
14	5	A_{12}	(G_0, G_0)	1	2
15	6	A_{12}	(G_0, G_0)	1	2
16	7	A_{12}	(G_0, G_0)	1	2
12‡	8	A_{17}	(G_0, G_0)	0	1
18	9	A_{17}	(G_1, G_0)	1	1
19	10	A_{17}	(G_0, G_0)	1	1
20	11	A_{17}	(G_0, G_0)	1	1
21	12	A_{17}	(G_0, G_0)	1	1
17‡	13	A_{22}	(G_0, G_0)	0	0
22	14	A_{23}	(G_0, G_0)	0	0
			...		

(b)

SN_i	AN_i
11	1
17	2
18	3
13	4
14	5
15	6
16	7
12‡	8
19	9
20	10
21	11
22	12
17‡	13
23	14
24	15
25	16
26	17
18‡	18
27	19
...	

(c)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
13	4	A_{12}	(G_1, G_2)	1	2
14	5	A_{12}	(G_0, G_0)	1	2
15	6	A_{12}	(G_0, G_0)	1	2
16	7	A_{12}	(G_0, G_0)	1	2
12‡	8	A_{17}	(G_0, G_0)	0	1
19	9	A_{17}	(G_2, G_0)	0	1
20	10	A_{17}	(G_0, G_0)	0	1
21	11	A_{17}	(G_0, G_0)	0	1
22	12	A_{17}	(G_0, G_0)	0	1
17‡	13	A_{18}	(G_0, G_0)	0	1
23	14	A_{18}	(G_0, G_0)	1	1
24	15	A_{18}	(G_0, G_0)	1	1
25	16	A_{18}	(G_0, G_0)	1	1
26	17	A_{18}	(G_0, G_0)	1	1
18‡	18	A_{27}	(G_0, G_0)	0	0
27	19	A_{28}	(G_0, G_0)	0	0
			...		

(d)

‡ retransmitted packet

Fig. 5. Examples that give the (G_n, G_m), $0 < n < m$ syndrome.

or wireless network. Consequently, the window reduction operation as a result of packet 12 being lost in the wireline network will be slightly delayed under the second scenario. The total number of window reduction, however, equals the number of packets lost on the wireline network under both scenarios.

4.1.5. Causes of syndrome (G_n, G_m), $0 < m < n$

Proposition 7. *If the syndrome for two packets (with sequence numbers SN_i and SN_{i+1}) consecutively received by the MH is (G_n, G_m), $0 < m < n$, then the possible error events are (L, L), (O, L) or $(O + L, L)$.*

Proof. In the case of no out-of-order packet event in the wireline network, by Proposition 1, we know that the only possible error event is (L, L) (i.e., m packet losses occur on the wireless link, and $n - m$ packet losses occur on the wireline network).

In the case that packets may be out-of-order in the wireline network, first by Lemma 2, we know m packet losses occur on the wireless link. Second, as the gap in the sequence number space is of size $n > m$, we know at least $n - m$ packets whose sequence numbers fall in (SN_i, SN_{i+1}) are either lost, does not arrive in sequence (but between SN_i and SN_{i+1}), or both. Hence, the possible error events are (L, L), (O, L) or $(O + L, L)$. □

The same rules used in the base approach here. The following example illustrates how credits are given and ELN bits are set under the (G_n, G_m), $0 < m < n$ syndrome.

Example 4 Consider the following three scenarios:

Scenario 1: ((L, L) event) Both packets 12 and 13 are lost in the wireline network and packet 14 is lost on the wireless link. The order in which packets arrive at the BS is given in Figure 6(a), and the syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 6(b).

Scenario 2: ((O, L) event) Both packets 12 and 13 arrive out of order at the BS (the order in which packets arrive at the BS is given in Figure 6(c)), and packet 14 is lost on the wireless link. The syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 6(d).

Scenario 3: ($(O + L, L)$ event) Packet 12 is lost in the wireline network, packet 13 arrives out of order at the BS (the order in which packets arrive at the BS is given in Figure 6(e)), and packet 14 is lost on the wireless link. The syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 6(f).

As shown in Figure 6(b), (d) and (f), the sequence numbers received and the syndromes observed before sequence number 17 are indistinguishable under the three scenarios and 1 credit is given in all three scenarios. Again the total number of window reduction equals the number of packets lost in the wireline network. Note also that in the second scenario, as packet 12 is significantly delayed (i.e., more than three packets), which in turn triggers the window reduction operation under TCP-Reno. With *syndrome*, the one credit is used here to prevent occurrence of window reduction (i.e., the ELN bits for acknowledgments for packet 12 are set).

4.1.6. Causes of syndrome (G_n, G_m), $0 < m = n$

Proposition 8. *If the syndrome for two packets (with sequence numbers SN_i and SN_{i+1}) consecutively received by the MH is (G_n, G_m), $0 < m = n$, then the possible error events are $(-, L)$, (O, L) or $(O + L, L)$.*

Proof. In the case of no out-of-order packet event in the wireline network, by Proposition 1, we know that the only possible error event is $(-, L)$ (i.e., m packet losses occur on the wireless link).

Now we consider the case in which packets may be re-ordered in the wireline network. By Lemma 2, we know m packet losses occur on the wireless link. Moreover, the fact that a gap of size n exists in the sequence number space implies that the n packets whose sequence numbers fall in (SN_i, SN_{i+1}) either are lost in the wireline network, arrive later than SN_{i+1}, or are lost on the wireless link. In the former two cases, exactly k other packets arrive out of order and are inserted between SN_i and SN_{i+1}. The possible error events are $(-, L)$, (O, L) or $(O + L, L)$. □

The same rule used in the base approach applies here. The following example illustrates how credits are given and ELN bits are set under the (G_n, G_m), $0 < m = n$ syndrome.

SN_i	AN_i
11	1
14	2
15	3
16	4
17	5
18	6
12‡	7
19	8
20	9
21	10
22	11
13‡	12
23	13
24	14
25	15
26	16
14‡	17
27	18
...	

(a)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
15	3	A_{12}	(G_3, G_1)	1	1
16	4	A_{12}	(G_0, G_0)	1	1
17	5	A_{12}	(G_0, G_0)	1	1
18	6	A_{12}	(G_0, G_0)	1	1
12‡	7	A_{13}	(G_0, G_0)	0	0
19	8	A_{13}	(G_0, G_0)	0	0
20	9	A_{13}	(G_0, G_0)	0	0
21	10	A_{13}	(G_0, G_0)	0	0
22	11	A_{13}	(G_0, G_0)	0	0
13‡	12	A_{14}	(G_0, G_0)	0	0
23	13	A_{14}	(G_0, G_0)	0	0
24	14	A_{14}	(G_0, G_0)	0	0
25	15	A_{14}	(G_0, G_0)	0	0
26	16	A_{14}	(G_0, G_0)	0	0
14‡	17	A_{27}	(G_0, G_0)	0	0
27	18	A_{28}	(G_0, G_0)	0	0
			...		

(b)

SN_i	AN_i
11	1
14	2
15	3
16	4
17	5
12	6
13	7
18	8
19	9
20	10
21	11
14‡	12
22	13
...	

(c)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
15	3	A_{12}	(G_3, G_1)	1	1
16	4	A_{12}	(G_0, G_0)	1	1
17	5	A_{12}	(G_0, G_0)	1	1
12	6	A_{13}	(G_0, G_0)	0	0
13	7	A_{14}	(G_0, G_0)	0	0
18	8	A_{14}	(G_0, G_0)	0	0
19	9	A_{14}	(G_0, G_0)	0	0
20	10	A_{14}	(G_0, G_0)	0	0
21	11	A_{14}	(G_0, G_0)	0	0
14‡	12	A_{22}	(G_0, G_0)	0	0
22	13	A_{23}	(G_0, G_0)	0	0
			...		

(d)

SN_i	AN_i
11	1
14	2
15	3
16	4
17	5
18	6
13	7
12‡	8
19	9
20	10
21	11
22	12
14‡	13
23	14
...	

(e)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
15	3	A_{12}	(G_3, G_1)	1	1
16	4	A_{12}	(G_0, G_0)	1	1
17	5	A_{12}	(G_0, G_0)	1	1
18	6	A_{12}	(G_0, G_0)	1	1
13	7	A_{12}	(G_0, G_0)	0	0
12‡	8	A_{14}	(G_0, G_0)	0	0
19	9	A_{14}	(G_0, G_0)	0	0
20	10	A_{14}	(G_0, G_0)	0	0
21	11	A_{14}	(G_0, G_0)	0	0
22	12	A_{14}	(G_0, G_0)	0	0
14‡	13	A_{23}	(G_0, G_0)	0	0
23	14	A_{24}	(G_0, G_0)	0	0
			...		

(f)

‡ retransmitted packet

Fig. 6. Examples that give the (G_n, G_m), $0 < m < n$ syndrome.

Example 5 Consider the following three scenarios:

Scenario 1: $((-, L)$ event) Packet 12 is lost on the wireless link. The order in which packets arrive at the BS is given in Figure 7(a), and the syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 7(b).

Scenario 2: $((O, L)$ event) Both packets 12 and 17 arrive out of order at the BS (the order in which packets arrive at the BS is given in Figure 7(c)), and packet 17 is lost on the wireless link. The syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 7(d).

SN_i	AN_i
11	1
12	2
13	3
14	4
15	5
16	6
12‡	7
17	8
...	

(a)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
13	3	A_{12}	(G_1, G_1)	1	1
14	4	A_{12}	(G_0, G_0)	1	1
15	5	A_{12}	(G_0, G_0)	1	1
16	6	A_{12}	(G_0, G_0)	1	1
12‡	7	A_{17}	(G_0, G_0)	0	0
17	8	A_{18}	(G_0, G_0)	0	0
			...		

(b)

SN_i	AN_i
11	1
17	2
13	3
14	4
12	5
15	6
16	7
18	8
19	9
20	10
17	11
21	12
...	

(c)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
13	3	A_{12}	(G_1, G_1)	1	1
14	4	A_{12}	(G_0, G_0)	1	1
12	5	A_{15}	(G_0, G_0)	0	1
15	6	A_{16}	(G_0, G_0)	0	1
16	7	A_{17}	(G_0, G_0)	0	1
18	8	A_{17}	(G_0, G_0)	1	1
19	9	A_{17}	(G_0, G_0)	1	1
20	10	A_{17}	(G_0, G_0)	1	1
17	11	A_{21}	(G_0, G_0)	0	0
21	12	A_{22}	(G_0, G_0)	0	0
			...		

(d)

SN_i	AN_i
11	1
17	2
13	3
14	4
15	5
16	6
18	7
12‡	8
19	9
20	10
21	11
17‡	12
22	13
...	

(e)

SN_i	AN_i	Ack	Syndrome	ELN	Crdt
11	1	A_{12}		0	0
13	3	A_{12}	(G_1, G_1)	1	1
14	4	A_{12}	(G_0, G_0)	1	1
15	5	A_{12}	(G_0, G_0)	1	1
16	6	A_{12}	(G_0, G_0)	1	1
18	7	A_{12}	(G_1, G_0)	1	1
12‡	8	A_{17}	(G_0, G_0)	0	0
19	9	A_{17}	(G_0, G_0)	0	0
20	10	A_{17}	(G_0, G_0)	0	0
21	11	A_{17}	(G_0, G_0)	0	0
17	12	A_{22}	(G_0, G_0)	0	0
22	13	A_{23}	(G_0, G_0)	0	0
			...		

(f)

‡ retransmitted packet

Fig. 7. Examples that give the (G_n, G_m), $0 < m = n$ syndrome.

Scenario 3: ($(O + L, L)$ event) Packet 12 is lost in the wireline network, and packet 17 arrives out of order at the BS (the order in which packets arrive at the BS is given in Figure 7(e)) and is lost on the wireless link. The syndromes observed, the acknowledgment returned, and the action taken, by the MH are given in Figure 7(f).

As shown in Figure 7(b), (d) and (f), the sequence numbers received and the syndromes observed before sequence number 14 are indistinguishable under the three scenarios and 1 credit is given in all three scenarios. Again the total number of window reduction equals the number of packets lost on the wireline network. Note also that in the second scenario, the credit is not deducted when the out-of-order packet 12 arrives because only three duplicate acknowledgments are generated for packet 12.

Table 2 summarizes the possible syndromes that exist for connections from FHs to MHs and the corresponding actions receivers take under each syndrome.

4.2. Extended algorithm for $MH \rightarrow BS \rightarrow FH$ connections

For connections that are initiated by MHs, the BS adopts a similar approach to label transit packets. That is, the BS counts, for each TCP connection, the number of packets that it relays to the destination FH so far, and attaches this number in the (option

Table 2. Causes and the corresponding actions for syndromes that occur for connections FHs → MHs.

Syndrome	Possible events (wireline, wireless)	Action
(G_0, G_0)	$(-,-)$, $(O,-)$	No op
(G_n, G_0), $n > 0$	$(L,-)$, $(O,-)$, $(O+L,-)$	No op
(G_0, G_m)	(O,L), $(O+L,L)$	*credit* ← *credit* + m
(G_n, G_m), $0 < n < m$	(O,L), $(O+L,L)$	*credit* ← *credit* + m
(G_n, G_m), $0 < m < n$	(L,L), (O,L), $(O+L,L)$	*credit* ← *credit* + m
(G_n, G_m) $0 < n = m$	$(-,L)$, (O,L), $(O+L,L)$	*credit* ← *credit* + m

Table 3. Causes and the corresponding actions for syndromes that occur for connections MHs → FHs.

Syndrome	Possible events (wireline, wireless)	Action
(G_0, G_0)	$(-,-)$, $(O,-)$	No op
(G_n, G_0), $n > 0$	$(-,L)$	*credit* ← *credit* $+n$
(G_0, G_m), $m > 0$	Impossible	—
(G_n, G_m), $0 < n < m$	Impossible	—
(G_n, G_m), $0 < m < n$	(L,L), (O,L), $(O+L,L)$	*credit* ← *credit* $+(n-m)$
(G_n, G_m), $0 < n = m$	$(L,-)$, $(O,-)$, $(O+L,-)$	No op

field of the) TCP header. According to the syndromes observed, the FH determines when to set the ELN bit. Again we make the assumptions that no packets are re-ordered on the wireless link and that there exist no duplicate packets (including retransmitted ones), and defer the discussion on the impact of existence of duplicate packets to Section 5.

Let (X, Y), X, $Y \in \{O, L, -\}$ denote the combined events on the wireline network and the wireless link, as perceived by the destination host at the time when it receives the second packet of the two packets consecutively received. The rules used to adjust *credits* and set ELN bits can be derived in a similar way as in Section 4.1. For completeness, we list all the propositions that give the causes for possible syndromes and summarize in Table 3 the actions taken for each possible syndrome. Interested readers are referred to Reference [16] for a detailed account of the proofs.

Proposition 9. *If the syndrome for two packets consecutively received by a fixed host is* (G_0, G_0)*, then no packet loss occurs (between the two packets) both in the wireline network and on the wireless link.*

Proposition 10. *If the syndrome for two packets (with sequence numbers* SN_i *and* SN_{i+1}*) consecutively received by a fixed host is* (G_n, G_0), $n > 0$*, the error event is* $(-, L)$.

Proposition 11. *It is impossible for a FH to observe the* (G_0, G_m)*,* $m > 0$ *syndrome.*

Proposition 12. *It is impossible for a FH to observe the* (G_n, G_m)*,* $0 < n < m$ *syndrome.*

Proposition 13. *If the syndrome for two packets (with sequence numbers* SN_i *and* SN_{i+1}*) consecutively received by a FH is* (G_n, G_m)*,* $0 < m < n$*, then the possible error events are* (L, L)*,* (O, L) *or* $(L + O, L)$.

Proposition 14. *If the syndrome for two packets (with sequence numbers* SN_i *and* SN_{i+1}*) consecutively received by a FH is* (G_n, G_m)*,* $0 < m = n$*, then the possible error events are* $(L, -)$*,* $(O, -)$ *or* $(O + L, -)$.

Table 3 summaries the possible syndromes that exist for connections from MHs to FHs and the corresponding actions receivers take under each syndrome.

5. Impact of Duplicate Packets on *Syndrome*

In this section, we analyze the impact of duplicate packets on the performance of *syndrome*. Duplicate packets may be generated either by the wireline network (as a result of corrupted routing tables) or the sender (the latter as a result of (premature) retransmission).

Without loss of generality, we only consider connections from FHs to MHs (connections for the other direction can be reasoned in a similar way). Let the original packet be denoted as P_i and its duplicate as $\widehat{P}_i$. We consider four possible scenarios for packets P_i and $\widehat{P}_i$: (C1) both packets arrive at the BS; (C2) P_i is lost in the wireline network, but $\widehat{P}_i$ arrives at the BS; (C3) P_i arrives at the BS, but $\widehat{P}_i$ is lost in the wireline network; and (C4) both packets are lost in the wireline network.

As far as *syndrome* (and the BS) is concerned, the effect of case (C2) is the same as that of packet P_i being delayed, and case (C3) can be treated as if the duplicate packet were never generated. Similarly, case (C4) can be treated as if packet P_i were lost in the wireline network and $\widehat{P}_i$ never generated. The only scenario that calls for a careful analysis is case (C1).

In case (C1), as both duplicate packets arrive at the BS, they will be transmitted over the wireless link. Again we consider four cases:

C1.a If both packets are successfully received by the mobile host, a (G_0, G_0) syndrome will be observed by the MH both when P_i and $\widehat{P}_i$ arrive.

C1.b If packet P_i is lost on the wireless link but packet $\widehat{P}_i$ arrives, the MH will observe a (G_1, G_1) syndrome when packet P_{i+1} arrives, and give one credit (Proposition 8). Later when packet $\widehat{P}_i$ arrives, the MH will treat it as if it were the retransmission. In the rare situation that packet $\widehat{P}_i$ arrives before the MH sends three consecutive duplicate acknowledgments, this one credit will not be used. This implies the sender will be allowed not to reduce its congestion window one time when it should have.

C1.c If packet P_i arrives, but packet $\widehat{P}_i$ is lost on the wireless link, the MH will observe a (G_0, G_0) syndrome when packet P_i arrives, but a (G_0, G_1) syndrome upon arrival of the packet that is labeled right after packet $\widehat{P}_i$ at the BS. The MH will give one credit (Proposition 5). This, again, implies the sender will be allowed not to reduce its congestion window one time when it should have.

C1.d If both packets are lost on the wireless link, the MH will observe (G_1, G_1) and (G_0, G_1) syndromes, respectively, as the packets following P_i and $\widehat{P}_i$ arrive. Totally two credits will be given: one is used to allow the sender not to reduce its congestion window when it receives three duplicate acknowledgments for P_i, but the other may be misused.

Table 4 summarizes the impact of duplicate packets under the various scenarios. As shown in Table 4, *syndrome* may err optimistically (i.e., it allows the sender not to reduce its congestion window when the sender should have) in three rare cases (**C1.b–C1.d**). However, as will be verified in Section 6, the impact on the performance of *syndrome* is not very significant, as these cases rarely occur. For example, **C1.b** occurs when a duplicate, $\widehat{P}_i$, of packet P_i is generated, both packets arrive at the BS, but only $\widehat{P}_i$ arrives at the MH, and moreover $\widehat{P}_i$ arrives at the MH before the MH sends three duplicate acknowledgments asking for P_i.

6. Simulation Results

We have implemented *syndrome* in *ns-2*, and conducted a simulation study to validate the proposed design and compared the performance against *SNOOP* —the approach known to give the 'best' performance in the second category. (By 'best,' we mean *SNOOP* perfectly detects lost/out-of-order/duplicate packet that occur in the wireline/wireless network, as it keeps the state in an entire congestion window.) All algorithms used in the simulation, except *syndrome*, were part of the standard *ns-2* distribution.

Parameter setting: In the simulation study, we have considered networks of arbitrary topology and used an assortment of traffic sources (mainly infinite-duration TCP and finite-duration TCP). To control the packet loss rate, all packet losses in the wireline/wireless network are artificially generated, and the packet loss probability over the wireless link is x times larger than that of the wireline link, where x varies from 10 to 1000. We use two types of packet loss models: uniform loss model (specified by the packet loss probability, p, over the wireless link) and burst loss model (specified by a two-state Markovian model shown in Figure 8).

Due to the space limitation, we only report on a small set of the simulations that we believe is the most representative. In particular, the results reported below are obtained from the network topology depicted in Figure 9 in which all the links have a bandwidth of 2 Mbps, a link delay of 5 ms, and are equipped with buffers of reasonably large sizes. S_1 and S_2 are two FHs, each establishing a TCP

Table 4. The impact of duplicate packets on *syndrome* for connections from FHs to MHs. P_i and $\widehat{P}_i$ are the original and duplicate packets, respectively.

Scenarios	P_i lost in wireline?	$\widehat{P}_i$ lost in wireline?	P_i lost on wireless?	$\widehat{P}_i$ lost on wireless?	Action	Impact
C2	Yes	No	—	don't care	treated as if P_i is delayed	No
C3	No	Yes	don't care	—	treated as if $\widehat{P}_i$ is never generated	No
C4	Yes	Yes	—	—	treated as if P_i is lost in the wireline network and $\widehat{P}_i$ is never generated	No
C1.a	No	No	No	No	MH observes (G_0, G_0) syndromes as P_i and $\widehat{P}_i$ arrive and identifies the duplicate	No
C1.b	No	No	Yes	No	MH observes a (G_1, G_1) syndrome as P_{i+1} arrives, and gives 1 credit	if $\widehat{P}_i$ arrives before 3 duplicate acknowledgments are sent, the algorithm errs on the optimistic side (1 time).
C1.c	No	No	No	Yes	MH observes a (G_0, G_0) syndrome as P_i arrives, and a (G_0, G_1) syndrome as the packet following $\widehat{P}_i$ arrives, and gives 1 credit	the algorithm errs on the optimistic side (1 time).
C1.d	No	No	Yes	Yes	MH observes (G_1, G_1) and (G_0, G_1) syndromes, and gives 2 credits	the algorithm errs on the optimistic side (1 time).

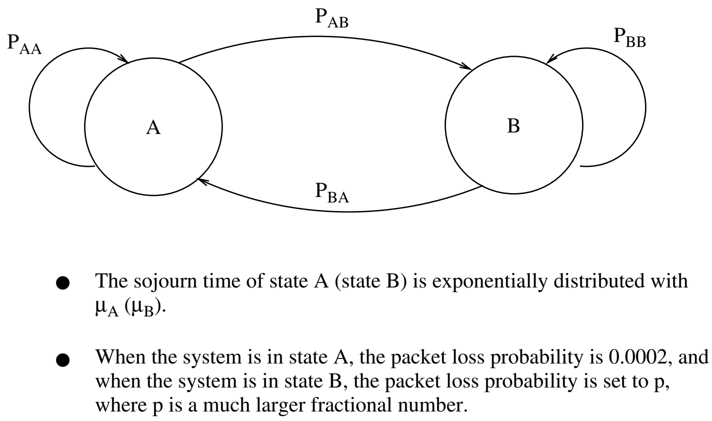


Fig. 8. The burst loss model used in the simulation. The sojourn time of state A and state B is exponentially distributed with mean $\mu_A = 0.075$ s and $\mu_B = 0.00375$ s. When the system is in state A, the packet loss probability over wireless links is set to 0.0002. When the system is in state B, the packet loss probability over wireless links set to p, where p varies over a wide range.

connection with the MH. The two hexagons denote the two base stations that serve as the home agent (HA) and the foreign agent (FA), respectively. Mobile IP is used as the underlying network protocol. To simulate packet losses, two instances of the loss model are inserted into the wireless link (HA $\leftrightarrow$ MH) and the link between node 8 and S_1. The parameters of the bursty loss model are set as follows: $p_{AA} = 0.95$, $p_{AB} = 0.05$, $p_{BB} = 0.95$, $p_{BA} = 0.05$, $\mu_A = 0.075$ s, $\mu_B = 0.00375$ s. In both (uniform and burst) models, the ratio of packet loss probability, p, on the wireless link over that on the wireline link is $x = 50$. In spite of quite a number of system parameters (topology, link capacity, buffer size, and packet size) involved, the results are found to be quite robust in the sense that the conclusion drawn from the performance curves for a representative set of parameters is valid over a wide range of parameter values.

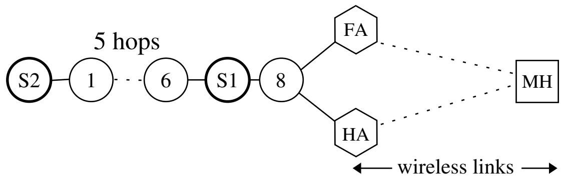


Fig. 9. Network topology used in the simulation.

Comparison of the base version of syndrome with TCP-Reno and SNOOP: We first evaluate the performance of the base approach of *syndrome* under the scenario that packets can be lost in the wireline/wireless network, but are neither out-of-order nor duplicate. Figure 10 gives the attainable throughput under *syndrome*, *SNOOP*, and TCP-Reno for the two connections established between $S_1 \leftrightarrow$ MH and between $S_2 \leftrightarrow$ MH under the uniform loss model. Several observations are in order: first, the throughput attained by the connection $S_1 \leftrightarrow MH$ is consistently higher than that by the connection $S_2 \leftrightarrow MH$, verifying the well-known fact that TCP is in favor of small-RTT connections. Second, the performance of *syndrome* is very close, and comparable, to that of *SNOOP* for a wide range of packet loss probabilities. The throughput attained by both *SNOOP* and *syndrome* senders is 30–80 per cent more than TCP senders in the case of high packet loss rates. As the performance in both directions ($FH \rightarrow MH$ and $MH \rightarrow FH$) exhibits similar behaviors, in what follows we present only performance results for connections $FH \rightarrow MH$.

Comparison of the full version of syndrome with TCP-Reno and SNOOP: We now evaluate the performance of the full version of *syndrome* under the scenario that packets can be lost in the wireline/wireless network, and in addition, can be out-of-order or duplicate. In this set of simulation runs, a packet may become out of order with probability 0.05, and the 'distance' between the in-order position of a packet and its out-of-order position is uniformly distributed between [1, 10] packets. Similarly, a packet may be duplicated by

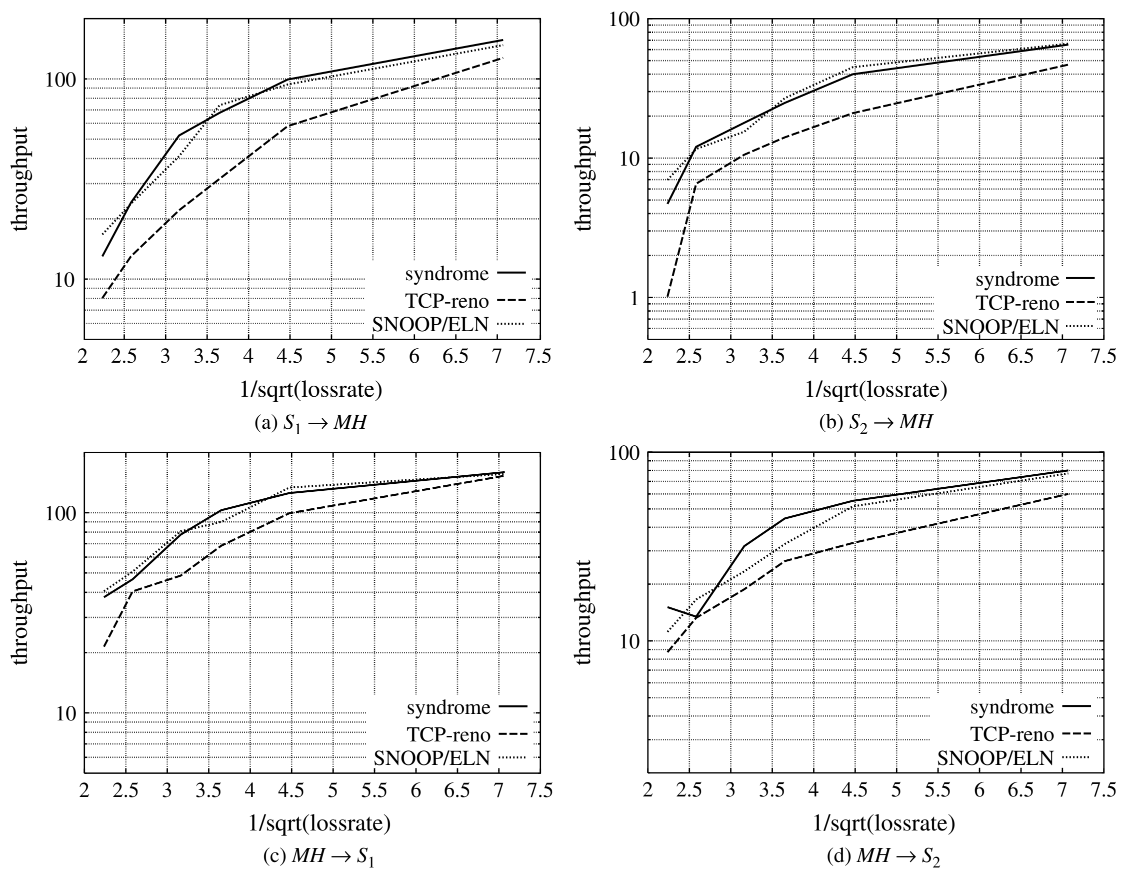


Fig. 10. Throughput attained by the base version of *syndrome*, *SNOOP*, and TCP-Reno under the uniform loss model. The x-axis is $1/\sqrt{p}$, where p is the packet loss probability. The y-axis is the attainable throughput (in kilobytes per second).

the network with probability 0.02 (which excludes duplicates generated by premature retransmission), and the 'distance' between the original and the duplicate packets is uniformly distributed between [1, 20] packets. Figure 11 gives the attainable throughput under *syndrome*, *SNOOP*, and TCP-Reno for connections $S_1 \rightarrow$ MH and $S_2 \rightarrow$ MH under the uniform loss model. *Syndrome* achieves comparable, and sometimes *slightly* better, throughput to/than *SNOOP*. This accounts for the fact that in the existence of duplicate packets *syndrome* may err on the optimistic side (**C1.b–C1.d** in Table 4).

A closer look of the impact of duplicate packets on the performance of syndrome: To further investigate the impact of duplicate packets on the performance of *syndrome*, we vary the probability of packets being duplicated from 0.02 to 0.2, under a fixed packet loss probability of 0.02. (This represents the unusual scenarios in which packets are highly likely to be duplicated.) Figure 12 gives the attainable throughput for connections $S_1 \rightarrow$ MH and $S_2 \rightarrow$ MH under *syndrome* and *SNOOP* versus the probability of packets being duplicated. As shown in Figure 12, only when the probability of packets being duplicated exceeds 0.05 (i.e., 2.5 times larger than the packet loss probability) and 0.1 (i.e., 5 times larger than the packet loss probability), respectively, for connections $S_1 \rightarrow MH$ and $S_2 \rightarrow MH$, will a notable performance discrepancy between *syndrome* and *SNOOP* be observed. Even in those cases in which *syndrome* does err (**C1.b–C1.d** in Table 4), *syndrome* performs, in the worst case (when the probability of packet

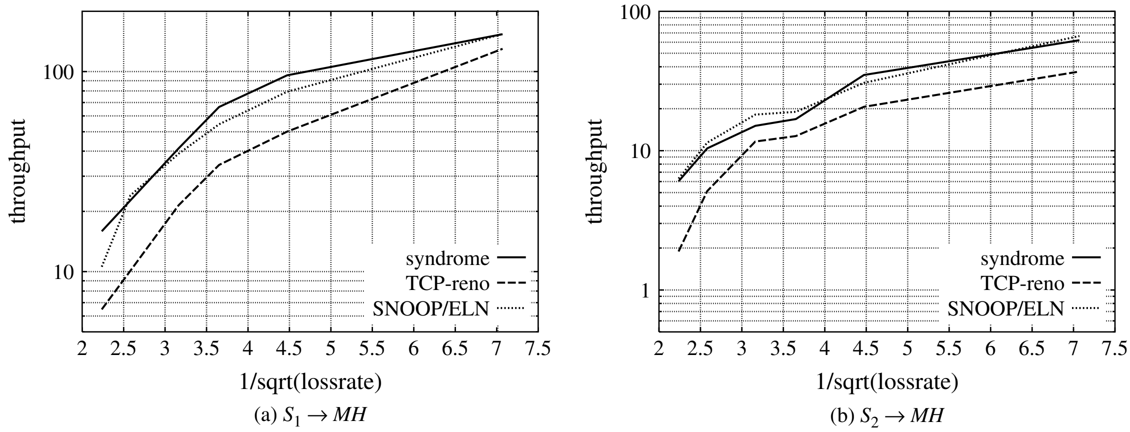


Fig. 11. Throughput attained by the full version of *syndrome*, *SNOOP*, and TCP-Reno under the uniform loss model. The x-axis is $1/\sqrt{p}$, where p is the packet loss probability. The y-axis is the attainable throughput (in kilobytes per second).

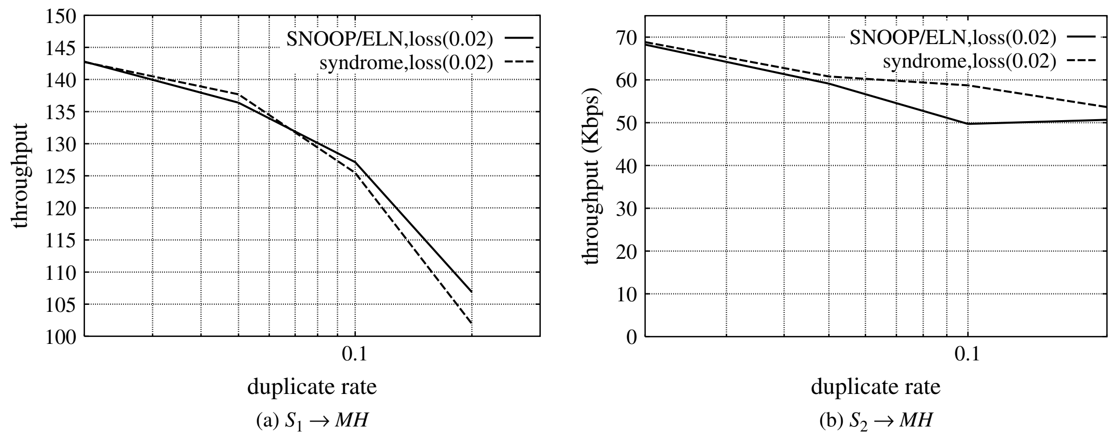


Fig. 12. Throughput attained by the full version of *syndrome* and *SNOOP* versus the probability of packets being duplicated. The x-axis is the probability of packets being duplicated. The y-axis is the attainable throughput (in kilobytes per second).

being duplicated is 0.1 for connection $S_2 \rightarrow MH$), 20 per cent more aggressively than *SNOOP*.

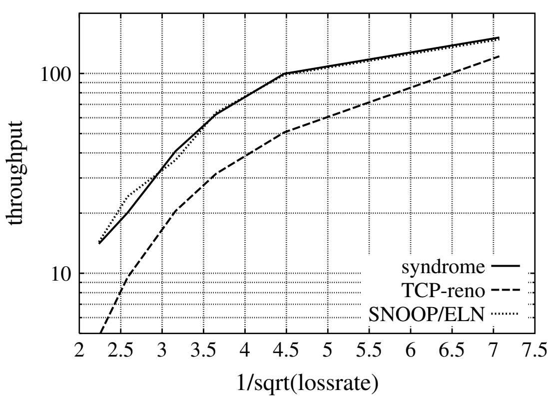
Performance under the burst loss model: To study whether or not the packet loss pattern may affect the performance of *syndrome*, we repeat the second set of simulations under the burst loss model. Figure 13 gives the attainable throughput under *syndrome*, *SNOOP*, and TCP-Reno for connections $S_1 \rightarrow$ MH and $S_2 \rightarrow$ MH under the burst loss model. Again *syndrome* achieves comparable throughput performance to *SNOOP*. This implies that the performance of *syndrome* is not subject to the packet loss pattern. This is anticipated, as we did not make any assumption on the packet loss pattern in establishing the theoretical base of *syndrome*.

Impact of hand-offs/disconnections on the performance of syndrome: As mentioned in Section 2, the performance of *syndrome* does degrade during hand-offs as it is not designed to deal with packet losses caused by hand-offs. However, *syndrome* can resume operations and take effect immediately after packets start to arrive at the new BS. To demonstrate this, we let the MH to move between the HA and the FA, and perform hand-off at time instant 8, 13, 18, 23 s. The hand-off delay (i.e., the interval between the instant when the MH leaves the old BS till the instant the new BS registers itself with the HA) is approximately 1.2 s. Figure 14 gives the instantaneous throughput and the TCP sequence number versus time under *syndrome*, *SNOOP*, and TCP-Reno during the simulation period. As shown in Figure 14, *syndrome* outperforms *SNOOP* and TCP-Reno in terms of instantaneous throughput and advance in TCP sequence number, and suffers least during hand-off.

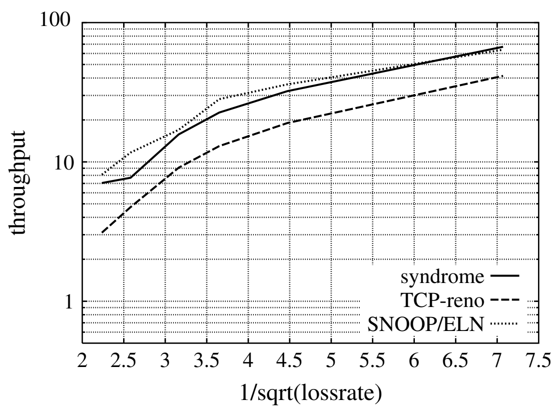
7. Conclusion

In this paper, we propose a light-weight, ELN-based approach, called *syndrome*, that enables receivers to tell the reason of packet losses and notify the sender in the acknowledgment, so that the TCP senders may adjust their congestion windows accordingly. *Syndrome* is grounded on a rigorous, analytic foundation, and does not require the BS to buffer packets or keep an enormous amount of states. The BS simply counts the number of packets for each TCP connection, and attaches this number in the TCP header. Although *syndrome* requires code change at both the BS and the end hosts, the change made to TCP is minor, can be packaged up as a software patch, and is backward compatible with original TCP. Through simulation in *ns-2* [17], we show that *syndrome* significantly improves the TCP performance (30–80 per cent improvement in the case of high packet loss rates) in mobile wireless environments and the performance gain is comparable to the more heavy-weight version of *SNOOP*. Also shown in the simulation is that *syndrome* resumes its operation immediately after hand-off when packets start to arrive at the BS, and hence suffers least from packet losses during hand-off.

We have identified several avenues for future work. Following the same line of derivation, we will establish the theoretical base of *syndrome* when it interfaces with other variations of TCP, e.g., TCP Sack. We are incorporating *syndrome* into TCP (with the extension of ECN) in FreeBSD, and will conduct an empirical study on a Lucent's WaveLAN-based testbed. We will also investigate how to combine

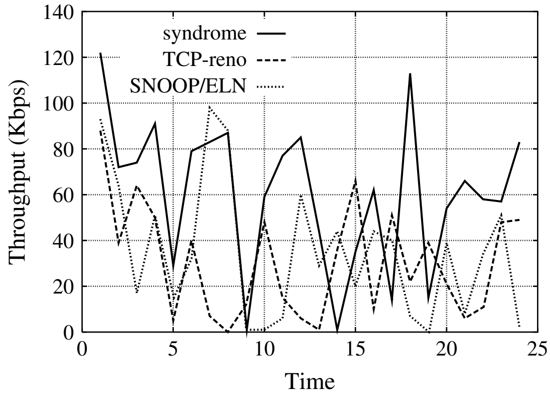


(a) $S_1 \rightarrow MH$

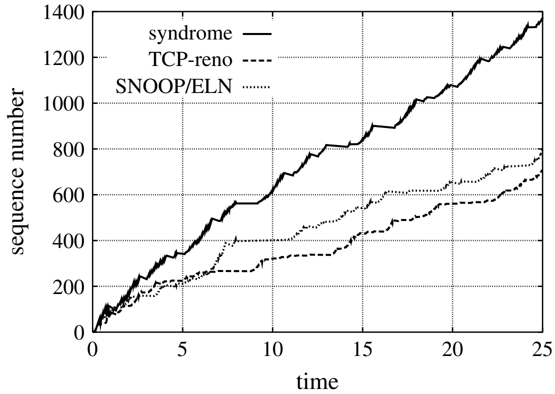


(b) $S_2 \rightarrow MH$

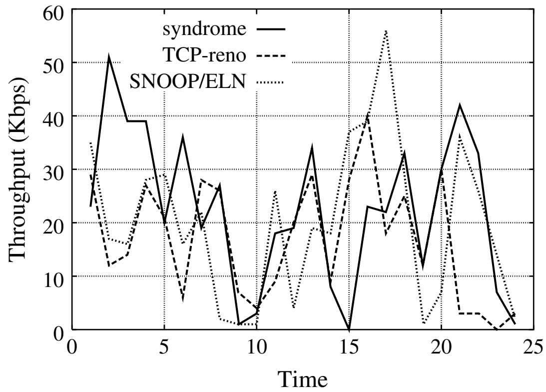
Fig. 13. Throughput attained by the full version of *syndrome*, *SNOOP*, and TCP-Reno under the burst loss model. The x-axis is $1/\sqrt{p}$, where p is the packet loss probability. The y-axis is the attainable throughput (in kilobytes per second).



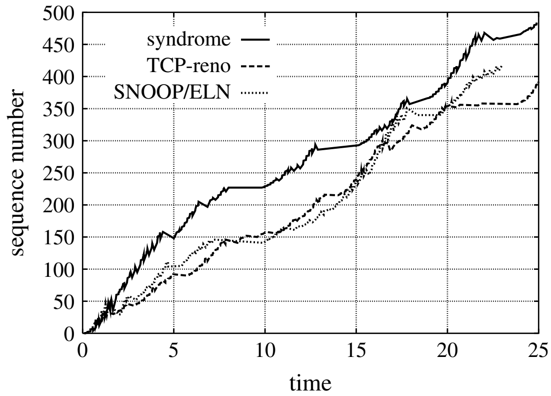
(a) $S_1 \rightarrow MH$ (instantaneous throughput)



(b) $S_1 \rightarrow MH$ (sequence number)



(c) $S_2 \rightarrow MH$ (instantaneous throughput)



(d) $S_2 \rightarrow MH$ (sequence number)

Fig. 14. The instantaneous throughput and the sequence number under *syndrome*, *SNOOP*, and TCP-Reno during the simulation period. The x-axis is the probability of packets being duplicated, and the y-axis is the instantaneous throughput (in kilobytes per second, obtained by using the moving window average with a window size of 1 s) in (a) and (c), and TCP sequence number in (b) and (d).

syndrome with *Freeze-TCP* to further improve its performance during hand-offs/disconnections.

Acknowledgment

The work reported in this paper was supported in part by NSF under Grant No. ANI-989018. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency.

References

1. Cáceres R, Iftode L. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE Journal on Selected Areas in Communications* 1995; **13**(5).
2. Brown K, Singh S. M-TCP: TCP for mobile cellular networks. *ACM Computer Communications Review (CCR)* 1997; **27**(5).
3. Goff T, Moronski J, Phatak DS. Freeze-TCP: A true end-to-end tcp enhancement mechanism for mobile environment. In *Proc. IEEE INFOCOM 2000*, March 2000.
4. Bakre A, Badrinath B. ITCP: Indirect TCP for mobile hosts. In *Proc. 15th Int'l Conf. on Distributed Computing Systems*, May 1995.
5. Yavatkar R, Bhagwat N. Improving end-to-end performance of TCP over mobile internetworks. In *Proc. of Workshop on Mobile Computing Systems and Applications*, December 1994.
6. Mehta M, Vaidya NH. Delayed Duplicate Acknowledgments: A Proposal to Improve Performance of TCP on Wireless Links. Technical Report, Texas A&M University, December 1997; Available at http//www.cs.tamu.edu/faculty/vaidya/Vaidya-mobile.html.
7. Balakrishnan H, Seshan S, Katz RH. Improving reliable transport and handoff performance in cellular wireless networks. *ACM Wireless Networks* 1995; **1**(4).
8. Balakrishnan H, Katz RH. Explicit loss notification and wireless web performance. In *Proc. IEEE Globecom Internet Mini-Conference*, November 1998.
9. Vaidya NH. TCP for wireless and MHs. In *Proc. 5th Annual Int'l Conf. on Mobile Computing and Networking (MOBICOM)*, August 1999. <http//www.cs.tamu.edu/faculty/vaidya/>.

10. Stadler JS. A link layer protocol for efficient transmission of TCP/IP via satellite. In *Proc. of MILCOM* 1997; **2**: 723–727.
11. Wang K, Tripathi SK. Mobile-end transport protocol: an alternative to TCP/IP over wireless links. In *Proc. of INFOCOM'98*, March 1998: 1046–1053.
12. Biaz S, Vaidya NH. Discriminating congestion losses from wireless losses using inter-arrival times at the receiver. In *Proc. of IEEE ASSET'99*, March 1999.
13. Biaz S, Vaidya NH. Distinguishing congestion losses from wireless transmission losses. In *Proc. of 7th Conf. on Computer Communications and Networks*, October 1998.
14. Biaz S. Heterogeneous Data Networks: Congestion or Corruption? PhD Thesis, Texas A&M University, College Station, August 1999.
15. Floyd S. TCP and explicit congestion notification. *ACM Computer Communications Review* 1994; **24**: 10–24.
16. Chen WP, Hsiao YC, Ge Y, Hou JC. Syndrome: a light-weight approach to improving TCP performance in mobile wireless networks. Technical Report, Ohio State University, Columbus, OH, December 2000.
17. UCB, LBNL, VINT network simulator. http//www-mash.cs.-berkeley.edu/ns/.
18. Balakrishnan H, Padmanabhan VN, Seshan S, Katz RH. A comparison of mechanisms for improving TCP performance over wireless links. *IEEEACM Trans. on Networking* 1997; **5**: 756–769.

