

## Effect of Delay Spike on SCTP, TCP Reno, and Eifel in a Wireless Mobile Environment

Shaojian Fu

Mohammed Atiquzzaman  
School of Computer Science  
University of Oklahoma,  
Norman, OK 73019-6151.

William Ivancic

Satellite Networks & Architectures Branch  
NASA Glenn Research Center  
21000 Brookpark Rd. MS 54-8,  
Cleveland, OH 44135.

**Abstract**— Eifel has been proposed as a solution to enhance the performance of TCP during delay spikes in a wireless mobile environment. This paper evaluates and compares the performance of SCTP, TCP, and Eifel during delay spikes. We have shown that although Eifel performs better than TCP Reno and SCTP when there are no packet losses, the opposite is true when packets are lost in the presence of delay spikes. Our results also show that a higher link bandwidth does not always increase the data throughput of SCTP, TCP Reno, and Eifel.

### I. INTRODUCTION

The Stream Control Transmission Protocol (SCTP) [1] has been developed as a reliable transport protocol over IP network. SCTP is based on congestion control and retransmission schemes which are similar to those of TCP. SCTP and TCP are both designed with wireline environments in mind, and assume slow and gradual changes in RTT.

Wireless mobile networks encounter delay spikes more frequently than wireline networks. A *delay spike* is defined as a situation where the RTT suddenly increases and then drops sharply back to its previous value [2]. Delay spikes in a wireless mobile environment may occur due to hand-off between cells, physical disconnection of the wireless link, link level recovery by the RLC layer, and preemption of data traffic by higher-priority traffic [3]. Delay spikes, resulting in *Spurious Timeout* (ST) and *Spurious Fast Retransmission* (SFR), can lead to serious end to end performance penalty in TCP [4].

The Eifel algorithm [4] has been proposed to alleviate the performance penalty in the case of TCP. Eifel requires both the sender and receiver to support TCP's timestamp option, which in turn requires an additional 12 bytes in the TCP header. Other alternative proposals to Eifel can be found in [5].

The *objective* of this paper is to evaluate and compare the impact of delay spikes on the performance of SCTP, TCP Reno and Eifel under various wireless scenarios, and suggest the appropriate use of protocols depending on

The work reported in this paper was supported by National Aeronautics and Space Administration (NASA) grant no. NAG3-2528

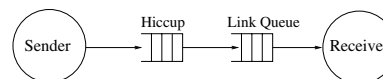


Fig. 1. Simulation Topology.

network conditions. The rest of the paper is organized as follows: The behavior and performance of TCP Reno, Eifel and SCTP in the presence of delay spikes in *lossless* and *lossy* networks are studied in Secs. II and III, respectively. Our recommendations on the use of an appropriate transport layer protocol are presented in Sec. IV, followed by concluding remarks in Sec. V.

### II. EFFECT OF DELAY SPIKE IN LOSSLESS NETWORK

In this section, we use the *ns-2* simulator to illustrate the impact of a delay spike on TCP Reno, Eifel and SCTP in a *loss free network*.

#### A. Simulation setup

The simulation topology is shown in Fig. 1. The link delay is 1.4 seconds [6] for both the uplink and downlink. A delay spike occurs in the uplink, beginning at time  $t = 28.0s$  and lasting for 12 seconds. We use a link bandwidth of 46.8Kbps in this section; results for a range of link bandwidths (9.05Kbps to 1.5Mbps) will be given in Sec. II-E.

#### B. Effect of Delay Spike on TCP Reno

The sender's segment plot for a TCP Reno sender and receiver is shown in Fig. 2. The delay spike causes two timeouts at points Fig2-A and Fig2-B, resulting in a spurious go-back-N retransmission of segments 132-150 (starting at point Fig2-C) and a SFR of segment 151 (point Fig2-D). Detailed description of the ST and SFR can be found in full-sized version of this paper [7]. RFC 2582 [8] proposed a "bug fix" which disables fast retransmissions until all the segments outstanding at timeout are Acked. Fast retransmission at point Fig2-D can be eliminated by using this bug fix.

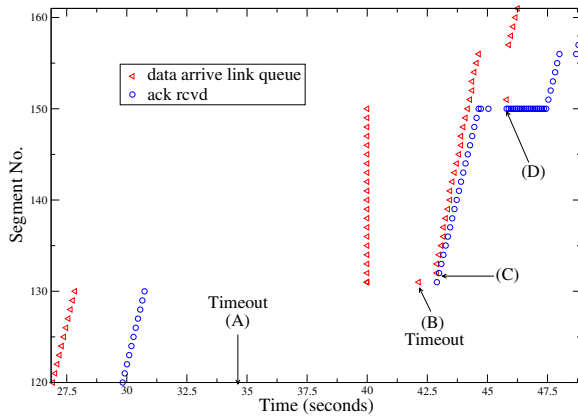


Fig. 2. Spurious Transmission and Spurious Fast Retransmission in TCP Reno.

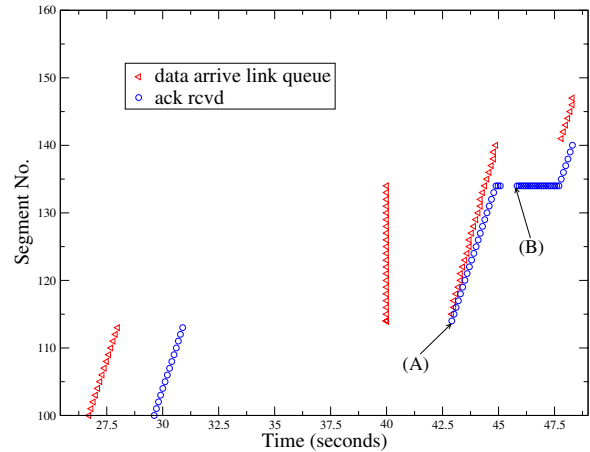


Fig. 4. Spurious timeout in SCTP due to a delay spike.

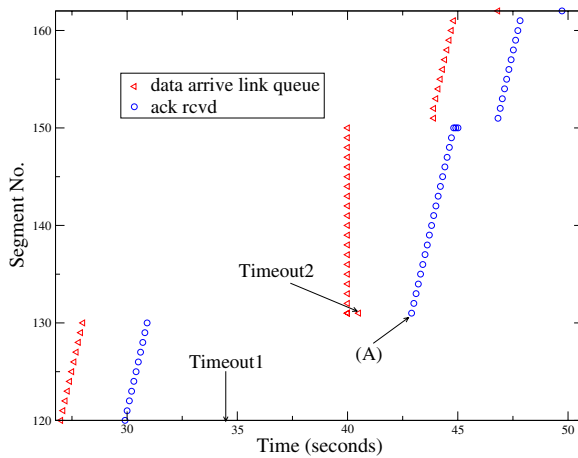


Fig. 3. Detecting Spurious Timeout with Eifel.

### C. Effect of Delay Spike on Eifel

Fig. 3 shows the sender's segment plot for the Eifel algorithm. When the sender gets the ACK for the original segment 131 at  $t = 42.9s$  (point Fig3-A), it detects the spurious timeout using the timestamp option. As a result, contrary to Fig. 2 for TCP Reno, segments 132-150 are *not retransmitted*. Because no DupAcks are generated by the receiver, Spurious Retransmissions are eliminated.

### D. Effect of Delay Spike on SCTP

SCTP and TCP Reno use the same RTO estimation algorithm; *SCTP, therefore, also suffers from Spurious Timeout* and exhibits a go-back-N behavior (point Fig4-A). *The good news is that SCTP can be free from Spurious Fast Retransmission*. Even though the sender receives a series of duplicate SACKs acknowledging segment 134 (point Fig4-B), *it doesn't do a SFR of segment 135*. This is because SACK allows the SCTP sender to determine whether a duplicate SACK acknowledging segment 134 is due to a spurious retransmission or due to a lost segment; this is not possible in TCP Reno.

### E. Performance Comparison in a Lossless Network

In this section, we compare the performance of the three protocols for link bandwidths varying from 9.05Kbps to 1.5Mbps for small and large Receiver Window ( $rwnd$ ) sizes. Bandwidths of 9.05, 13.4, 15.6, and 21.4Kbps correspond to the GPRS data rates for CS-1 to CS-4 coding schemes [6], and higher bandwidths (up to 1.5 Mbps) are available in 3G wireless networks. Our performance criteria is *throughput* which is measured by the number of segments delivered to the receiver during a 150 second FTP session.

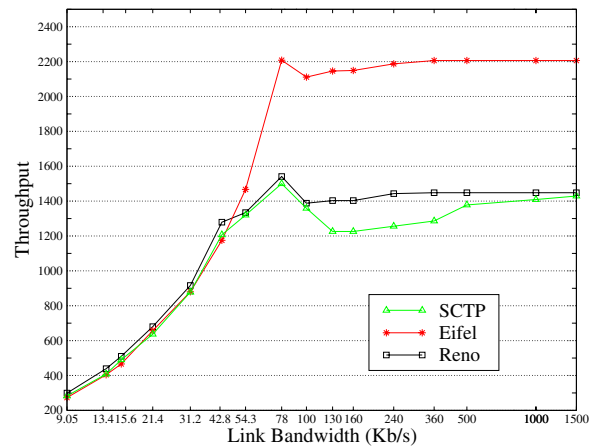


Fig. 5. Performance comparison of three protocols for large  $rwnd$ .

For low bandwidths (upto 42.8 Kbps) and large  $rwnd$ , there are no spurious timeouts because the continuous reception of ACKs during the delay spike prevents the sender from spurious timeouts. Therefore, we present the throughput in two sets: Figs. 5 and 6 for large (60 segments) and small (20 segments)  $rwnd$  respectively.

In Figs. 5 and 6, the slightly lower throughput of SCTP as compared to TCP is due to the fact that data and acknowledgment segments of SCTP use 52 and 68 byte headers respectively. Eifel achieves a higher throughput

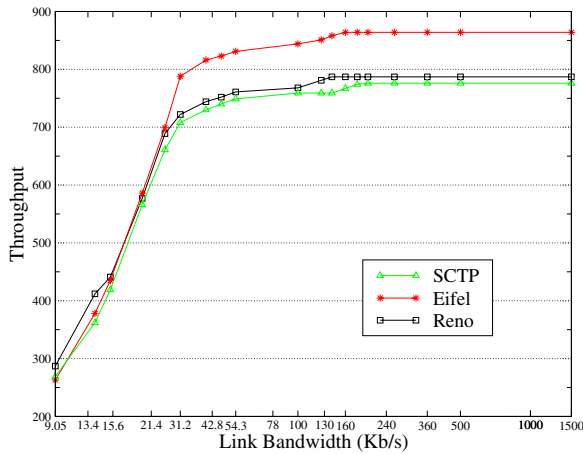


Fig. 6. Performance comparison of three protocols for small  $rwnd$ .

by eliminating the go-back-N retransmission (see Sec. II-C) and reverting the congestion window on the detection of ST.

The reason for a drop in throughput near 78Kbps for large  $rwnd$  (see Fig. 5) is that a higher bandwidth provides a lower RTO estimation and results in a second timeout during the delay spike. Consequently, the sender enters congestion avoidance starting at a *low* value of  $2 * MSS$ , which is the value of  $ssthresh$  after timeout.

### III. EFFECT OF DELAY SPIKE WITH PACKET LOSSES

In this section, **we extend the study of Sec. II to include packet losses due to congestion**<sup>1</sup>, using the same topology and link delays. For large  $rwnd$  of 60 segments, we use a queue size of 50 segments. However, for small  $rwnd$  of 20 segments, we limited the queue size to 13 segments to induce packet losses.

#### A. Comparison of behavior of the three protocols

Figs. 7-9 show the behavior of TCP Reno (with bug-fix [8]), Eifel and SCTP respectively in the presence of a delay spike with packet losses. At the end of delay spike, 60 segments are released from the hiccup queue with 10 segments dropped due to queue overflow (queue size limited to 50). Since the go-back-N retransmission of TCP Reno and SCTP caused by the delay spike effectively recovers the lost segments, we can see that the loss of segments doesn't have much effect on TCP Reno and SCTP.

Eifel detects the spurious timeout from the timestamp, and *resumes transmitting new segments* starting at point Fig8-A. Although, the sender can fast retransmit segment 257 at point Fig8-B, the rest of the lost segments can't be retransmitted until a timeout at point Fig8-C. Because

<sup>1</sup>We have observed that the behavior remains unchanged in the presence of other types of losses, such as link errors.

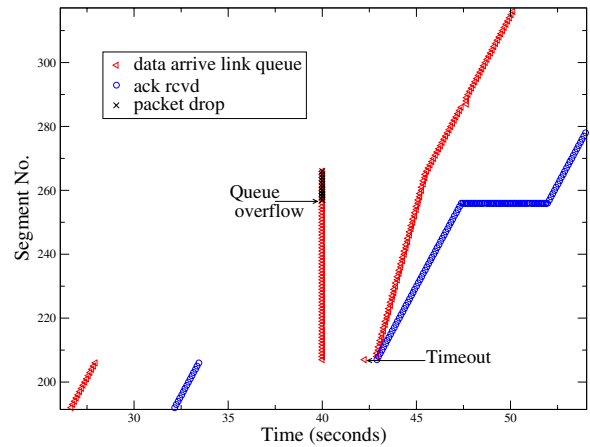


Fig. 7. TCP Reno in a delay spike with packet loss.

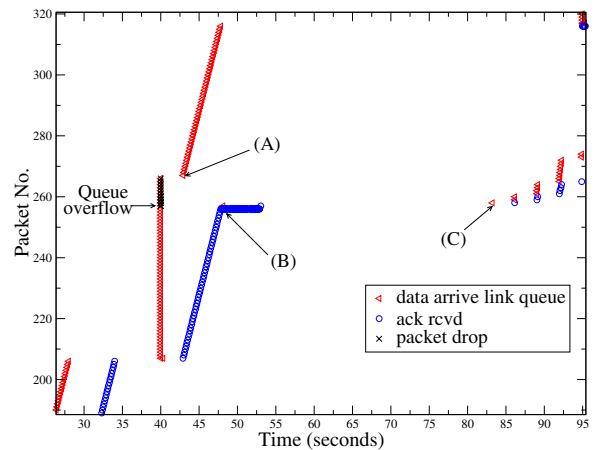


Fig. 8. Eifel in a delay spike with packet loss (bug fix *disabled*).

of the timeout, the RTO is increased to 30.2 seconds at  $t = 47.8s$ . resulting in a *transmission stall for about 30 seconds*.

#### B. Performance Comparison in a Lossy Network

Figs. 10 and 11 show the throughput of the three protocols for large and small  $rwnd$  respectively in a *lossy environment*. All the three protocol experience a drop in throughput near 78Kbps for large  $rwnd$ , for the same reason as described in Sec. II-E.

Fig. 10 shows that for large  $rwnd$  and low bandwidth, SCTP performs better than TCP Reno and Eifel because of the support provided by SCTP SACK in early detection of the lost segments to be retransmitted, instead of waiting for a timeout. For large  $rwnd$  and high bandwidth, spurious timeouts are caused by the delay spike. DupAcks arrive at the sender acknowledging the *spuriously retransmitted segments which clock out new data for TCP Reno but not for SCTP*. The above phenomenon is also observed at all bandwidths for small  $rwnd$ ; this is because spurious timeouts always happen for small  $rwnd$ .

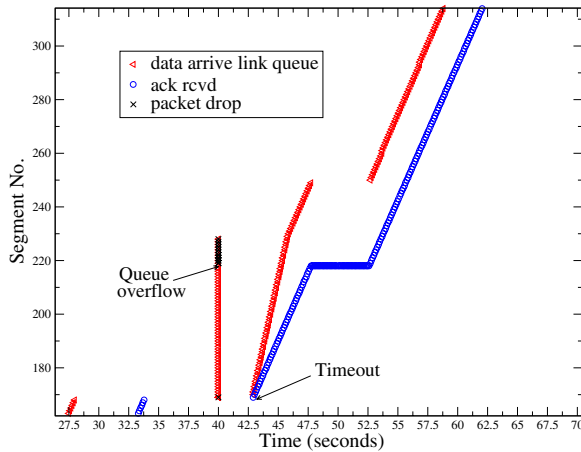


Fig. 9. Sctp in a delay spike with packet loss.

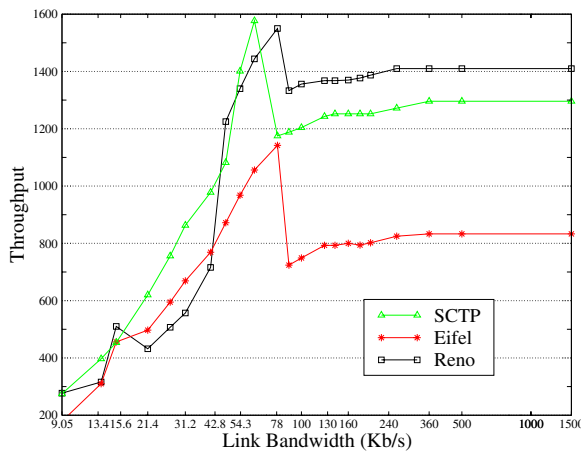


Fig. 10. Performance in the presence of packet losses for large  $rwnd$ .

For all values of link bandwidths and  $rwnd$  (see Figs. 10 and 11), Eifel has the worst performance when packets are lost during delay spikes. This is because of transmission stalls as discussed in Sec. III-A.

#### IV. RESULTS AND RECOMMENDATIONS

Based on the results presented in Secs II and III, we have ranked the performance of the three protocols in *descending order* for different cases of link bandwidths, Receiver Window, and packet loss in Table I. In a wireless mobile environment with delay spikes, we *recommend* using Eifel in a lossless network. For a lossy network with small link bandwidth and large Receiver Window, use Sctp; in all other cases of a lossy network, use TCP Reno.

#### V. CONCLUSION

We have shown that in the presence of delay spikes without packet loss, Sctp and TCP Reno have similar performance, while Eifel has a higher performance. In the

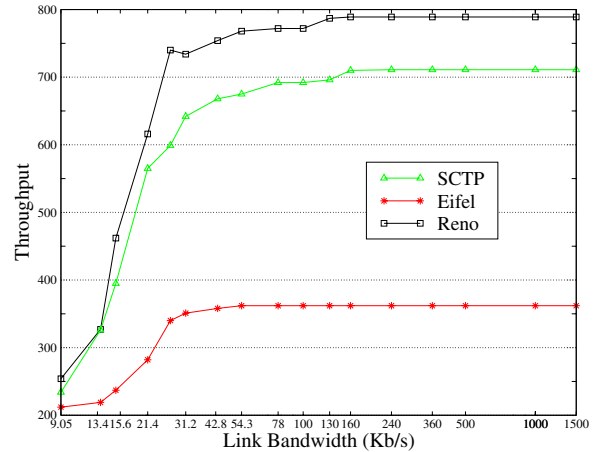


Fig. 11. Performance during delay spike with packet losses for small  $rwnd$ .

TABLE I

RELATIVE PERFORMANCE OF PROTOCOLS IN DESCENDING ORDER.

S.  $rwnd$  = SMALL  $rwnd$ , L.  $rwnd$  = LARGE  $rwnd$ .

	Low link BW		High link BW	
	S. $rwnd$	L. $rwnd$	S. $rwnd$	L. $rwnd$
Delay Spike	Eifel Reno Sctp (Fig.6)	Eifel Reno Sctp (Fig.5)	Eifel Reno Sctp (Fig.6)	Eifel Reno Sctp (Fig.5)
Delay Spike + Loss	Reno Sctp Eifel (Fig.11)	Sctp Reno Eifel (Fig.10)	Reno Sctp Eifel (Fig.11)	Reno Sctp Eifel (Fig.10)

case of delay spikes with packet losses, Eifel suffers from long transmission stalls, and TCP Reno and Sctp have better performance than Eifel. Our recommendations on the use of protocols can be used by networks engineers to switch between protocols depending on network conditions.

#### REFERENCES

- [1] R. Stewart and Q. Xie et. al., "Stream control transmission protocol." IETF RFC 2960, October 2000.
- [2] A. Gurtov and R. Ludwig, "Making TCP robust against delay spikes." Internet Draft, draft-gurtov-tsvwg-tcp-delay-spikes-00.txt, February 2002.
- [3] A. Gurtov, "Effect of delays on TCP performance," *IFIP Personal Wireless Communications*, August 2001.
- [4] R. Ludwig and R. H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmission," *ACM Computer Communications Review*, vol. 30, no. 1, pp. 30–36, January 2000.
- [5] R. Ludwig, "The TCP retransmit (rxt) flag." Internet Draft, draft-ludwig-tsvwg-tcp-rxt-flag-02.txt, November 2001.
- [6] R. Ludwig and D. Turina, "Link Layer Analysis of the General Packet Radio Service for GSM," *ICUPC*, San Diego, pp. 525–530, April 1997.
- [7] S. Fu M. Atiquzzaman and W. Ivancic, "Effect of delay spike on sctp, tcp reno and eifel in a wireless mobile environment," tech. rep., Computer Science, University of Oklahoma, www.cs.ou.edu/~atiq, July 2002.
- [8] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm." IETF RFC 2582, April 1999.