Impact of Retransmission Mechanisms on the Performance of SCTP and TCP [*]

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

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

ABSTRACT
The Stream Control Transmission Protocol (SCTP) has been developed as a reliable transport layer protocol for carrying PSTN signalling messages over IP networks. However, its advanced congestion control and fault tolerant features make it a strong competitor to the widely used TCP protocol. In this paper, we investigate the performance of SCTP in an error-prone network, and compared it with that of TCP. We have found that the performance of SCTP is higher than that of TCP for a range of wireless error probabilities. We have shown that differences in the retransmission mechanisms of the two protocols leads to the better performance of SCTP.

KEY WORDS
Stream Control Transmission Protocol, Transport Layer.

1 Introduction

The Internet Engineering Task Force (IETF) has standardized the Stream Control Transmission Protocol (SCTP) [1] as a reliable transport layer protocol for carrying PSTN signalling messages over IP networks. However, its advanced congestion control and fault tolerant features make it a strong competitor to the widely used TCP protocol. Like TCP, SCTP is based on reliable transmission, ordered delivery, flow and congestion control, slowstart, fast retransmit, slow-start threshold variation, Retransmission Timeout (RTO) calculation and congestion window computation. SCTP was developed when it became apparent that TCP or UDP are unsuitable for use in a common core network transporting both signalling and data traffic [2]. Differences, therefore, exist between TCP and SCTP, such as the *multistreaming* and *multihoming* features of SCTP. Multistreaming allows SCTP to have several streams within an association which is analogous to a connection in TCP. Multihoming increases the network-level redundancy by allowing an association to be set up using different IP addresses.

TCP and SCTP are reliable transport protocols. When a TCP or SCTP source detects a packet loss (such as due to errors in a satellite link), it retransmits the lost packet based on the information received in the acknowledgements from the receiver. The performance of the protocol can depend on the retransmission mechanism, such as time of retransmission of the lost packet, whether new packets can also be transmitted along with the lost packet, etc.

The next generation Internet will consist of a combination of wired and wireless links. The *goal* of this paper is to study the performance of SCTP when it coexists with TCP over large delay-bandwidth product links, such as satellite links. The delays in satellite networks are influenced by several factors, the main one being the orbit type [3]. One-way delay of Low Earth Orbit (LEO), Medium Earth Orbit (MEO) and Geostationary Earth Orbit (GEO) satellites are about 25ms, 130ms and 260ms respectively. Their channel bandwidths can vary from a few kb/s to as large as 622 Mb/s. Several problems arise in the performance of TCP and SCTP over high delay-bandwidth paths [4]. We considered the performance when transmission was carried out between end points over a GEO satellite link.

A number of SCTP features have been investigated in previous papers. The suitability of SCTP for satellite networks has been reported by Fu et al. [5]. Performance of SCTP in a satellite network using a backup terrestrial link has been studied by Jungmaier et al. [6]. The effect of SCTP's congestion control on large delay bandwidth product networks has been reported by Alamgir et al. [7]. Experimental studies have been used to show that fault tolerance of an SCTP association is improved through multihoming [8]. The use of unlimited number of SACK blocks in SCTP to improve the performance of Mobile IP has been investigated by Fu et al. [9]. Large sudden delays cause spurious retransmissions and spurious timeouts in TCP connections. Fu et al. [10] studied the impact of large sudden delays on the performance of SCTP and TCP Reno, and have shown that the two protocols have similar performance for large sudden delays. Excellent surveys

[*]This work was supported by NASA grant nos. NAG3-2528 and NAG3-2922.

of SCTP, its features and performance have recently been presented by Fu et al. [11], Stewart et al. [12], and Caro et al. [13]. Most relevant to this paper is the work presented in [14] in which the performance of SCTP, competing for resources with TCP in a loss-free network, is presented. Although a larage body of work on SCTP has been published in the literature, the *authors are not aware of any study on the impact of SCTP's retransmission scheme in a satellite network*.

This paper takes an in-depth look at the congestion control and retransmission mechanisms of TCP and SCTP [15, 1]. TCP uses the Fast Retransmission algorithm [16] following a packet loss. Although SCTP uses a similar algorithm, it differs slightly from the one used by TCP it terms of when to start fast retransmission (see Sec. 2) and the behavior when duplicate acknowledgements are received as described in Sec. 5.1. The *objective of this paper is to study the impact of SCTP's retransmission mechanism on its performance in terms of goodput and link utilization*.

Our *contributions* in this paper are as follows:

- We determined the *impact of retransmission schemes* on the performance of SCTP and TCP as a function of error rate in a satellite network.
- We showed that *SCTP achieves a better performance than TCP* when both share a long delay error-prone network.
- We took an in-depth look into the retransmission mechanisms of the two protocols and *identified differences which lead to a better performance of SCTP*.

The rest of the paper is organized as follows. Retransmission schemes of TCP and SCTP are outlined in Sec. 2. Our experimental setup and measures of performance are described in Secs. 3 and 4 respectively. Results are presented in Sec. 5, followed by concluding remarks in Sec. 6.

2 Congestion Control and Retransmission

This paper takes a closer look at the congestion control and retransmission mechanisms of TCP [15] and SCTP [1]. Terms such as congestion window (*cwnd*), slow-start thresh (*ssthresh*) and receiver window (*rwnd*) are used when describing congestion control in this paper. SCTP uses a variable called *flightsize* to denote the packet still unacknowledged by the receiver. For TCP, we used *number of unacknowledged packets* to represent TCP's *flightsize*.

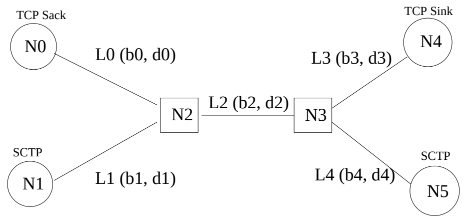

Figure 1: Simulation model.

The Fast Retransmit algorithm [16] is used by TCP, while SCTP implements a TCP-like Fast Retransmit [1]. In Fast Retransmit of both the protocols, duplicate acknowledgements (*dupack*s) are generated by the receiver whenever a packet is received out-of-order. The *dupack*s inform the sender which packet is expected by the receiver. When the sender receives a series of *dupack*s, it retransmits the lost packet, and at the same time, decreases both the *ssthresh* and the *cwnd* down to half the previous value of the *cwnd*. When the retransmitted packet is received by the receiver, it sends out an ack that acknowledges all the intermediate packets received between the time when the lost packet was identified, and when the retransmitted packet is required to be acknowledged. A slight difference exists between TCP and SCTP retransmission; TCP retransmits the lost packet on receipt of the third *dupack* while SCTP retransmits on receipt of the fourth *dupack* [1]. Further differences are discussed later in Sec. 5.1

3 Experimental Setup

All the simulation results presented in this paper were obtained from implementations of TCP and SCTP in the network simulator, *ns 2.1* [17], with an SCTP patch from the University of Delaware.

3.1 Simulation Model

Figure 1 shows the simulation model used for all our experiments. It consists of a TCP source/sink pair (N0,N4) and an SCTP source/sink pair (N1,N5) sharing a common bottleneck satellite link (L2) (with a high delay-bandwidth product) connected by a pair of routers (N2,N3). For each link, the tuple $(b*, d*)$ indicates the bandwidth and delay of that link.

The simulations in this paper all consisted of one-way traffic. The conventional drop-tail queueing mechanism was employed at the routers to implement FIFO. FTP was used as traffic generators so that a continuous stream of packets (or bytes) were transferred from the sources to the destinations, modeling large file transfers. Fast retransmit

	L0	L1	L2	L3	L4
Bandwidth, b Mbps	10	10	3	10	10
Delay, d ms	2	2	250	2	2

Table 1: Bandwidths and delays of links.

was used by TCP and SCTP sources. Acknowledgements were never dropped.

3.2 Simulation Configuration and Parameters

The link bandwidth and delay values are summarized in Table 1. To ensure fairness among the two sources, the parameters for links L0 and L1 were kept the same. Buffer sizes for N2 and N3 were set to a low value of 10 packets (about 15000 bytes). The router buffer sizes did not need to be set to a higher value because, due to the long propagation delay in l_2, few packets were queued up at N2. To take advantage of the large delay-bandwidth product, the buffer sizes of both the TCP and SCTP receivers was set to a large value of 64KB.

To isolate the impact of the differences in retransmission mechanisms between the two protocols, the TCP and SCTP host configurations were otherwise kept similar as given below:

1. Selective Acknowledgement (SACK) is mandatory for SCTP; TCP therefore used the SACK option.

2. Delayed acknowledgement was not used in either TCP or SCTP.

3. SCTP used one stream per association.

4. The payload of each packet (that is, without headers) was 1448 bytes for both protocols.

5. The initial receiver windows (*rwnd*) for both was set to the maximum allowed (64 KB).

6. The initial slow-start threshold (*ssthresh*) was made equal to the *rwnd* for both.

7. For both the protocols, throughput is inversely proportional to packet loss. Packet losses result in retransmissions which result in low throughput. To ensure fairness among the protocols, we set the error module in such a way that both TCP and SCTP sources drop approximately the same number of packets, with random packet loss probability in the range 0.008 - 0.03.

Loss Prob.	Perc. Thpt. (Γ_k)		Perc. Inc. in Thpt. (δ)	Link Util (ψ)
	TCP	SCTP		
0.008	49.8	50.2	6.1	20.9
0.009	46.7	53.3	13.3	19
0.01	43.7	56.3	27.6	17.8
0.015	44.1	55.9	25.6	14.4
0.02	45.8	54.2	17.0	12.4
0.025	46.7	53.3	13.3	10.8
0.03	46.8	53.2	12.7	9.8

Table 2: Performance comparison for TCP and SCTP.

4 Performance Metrics

In this section, we define the metrics that we have used to determine the impact of retransmission strategy on the performance of TCP and SCTP.

Percentage Throughput (Γ_k) (or goodput) of source $k, 1 \leq k \leq n$, where n is the total number of sources in the network, was obtained as:

$$\Gamma_k = \frac{\lambda_k}{\sum_{i=1}^{n} \lambda_i} \times 100 \qquad (1)$$

where λ_i denotes the throughput of source i. λ_i was measured by the total number of bytes sent by source i during a period of time, excluding retransmitted packets.

For our two sources case, the *Percentage Increase in Throughput* (δ) of source 1 over source 2 was calculated as:

$$\delta = \frac{\lambda_1 - \lambda_2}{\lambda_2} \times 100 \qquad (2)$$

Link utilization ψ of the bottleneck link (L2) was calculated as [18]

$$\psi = \frac{\sum_{i=1}^{n} \lambda_i}{b2} \times 100 \qquad (3)$$

where $b2$ is the bottleneck link bandwidth (see Fig. 1).

5 Results

This section presents the results from our simulations described in Sec. 3. Each simulation was run for 1000 seconds.

Table 2 shows *Percentage Throughput* ($\Gamma_k, 1 \leq k \leq 2$) achieved by TCP and SCTP, *Link Utilization* (ψ) of the bottleneck link (see Eqn. (3)) and *Percentage Increase in Throughput* (δ) of SCTP over TCP (see Eqn. (2)) for a range of loss probabilities.

A close inspection of Table 2 reveals that when the packet loss probability is 0.01, SCTP achieves the highest Percentage Increase in Throughput and Percentage Throughput. Figure 2 shows the variation of Percentage Throughput of SCTP as a function of loss probability.

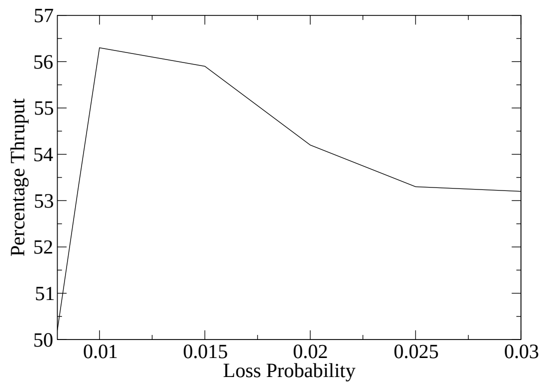

Figure 2: Percentage throughput of SCTP.

This can be explained as follows. When the loss probability is low, fewer packets are dropped. Ignoring timeouts, the number of retransmissions required for a packet is consequently low. Since SCTP has an inherent advantage over TCP during retransmissions as discussed in Sec. 5.1, SCTP is not able to achieve a much higher throughput than TCP when there are not many retransmissions because of relatively few packet losses.

As the probability of packet loss increases past 0.01, the percentage increase in the throughput of SCTP deteriorates again. This is because as the frequency of drops increases, both TCP and SCTP suffer from numerous drops, and timeouts which occur every time a retransmitted packet is lost. The congestion windows of both the sources frequently go down to the minimum, and slow-start initiated each time.

The Link Utilization column of Table 2 shows how utilization deteriorates as the number of drops increases. It also shows that SCTP always uses more than 50% of the utilized bandwidth.

For a loss probability of 0.01, the number of packet losses is not too few, and not too large. This loss rate being that in which SCTP achieves the highest advantage over TCP, it is thus ideal for demonstrating the difference in performance between SCTP and TCP. In the rest of this paper, we therefore use a loss probability of 0.01.

5.1 Analysis

Figure 3 shows the *Total Throughput* ($\lambda = \lambda_1 + \lambda_2$) as a function of time between time $t = 100$ and 140 seconds; λ was measured every 0.1 sec. The notable feature of this graph is that the throughput reaches zero periodically, indicating under-utilization of the shared link L2.

The time-sequence diagram of Figure 4 shows the sequence numbers of packets sent by the sources as a function of time. *It depicts the fact that the SCTP source allows more packets to be transmitted than the TCP during the same time period.* To rule out the possibility of other factors contributing to SCTP's higher performance,

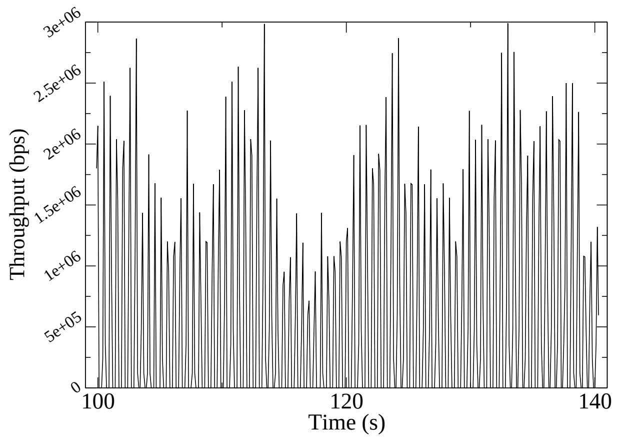

Figure 3: Total Throughput of the two sources.

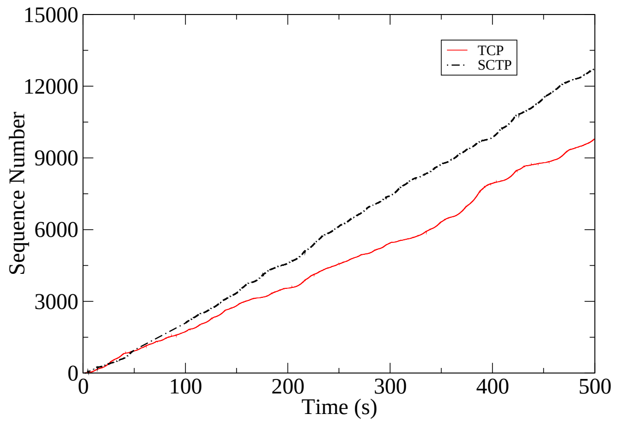

Figure 4: Sequence plot for loss prob of 0.01.

we have used one stream per association in SCTP with no multistreaming or multihoming, no delayed acknowledgement, and same round-trip times (RTT) and Retransmission Timeout (RTO). Moreover, the same initial slow-start threshold (*ssthresh*) for both of the sources result in the same slow start and congestion avoidance procedures. This is evident from Figs. 5 and 6 which show the variation of *cwnd* and *ssthresh* for TCP and SCTP respectively for the first 300 seconds of simulation. We see that the congestion windows of both TCP and SCTP fluctuate within approximately the same range (1-19 packets). We also calculated the same average *cwnd* and found it to be approximately eight for both the sources. We therefore conclude that, in our experimental setup (which is to show the impact of SCTP's retransmission mechanism), SCTP is achieving higher throughput because of its retransmission mechanism and not because of differences in other congestion control parameters which could result in a larger average *cwnd* than that of TCP.

Fig. 7 depicts how the TCP source, on detecting the loss of packet number 7, carries out Fast Retransmit. At the time the packet was lost, *cwnd* was eight packets.

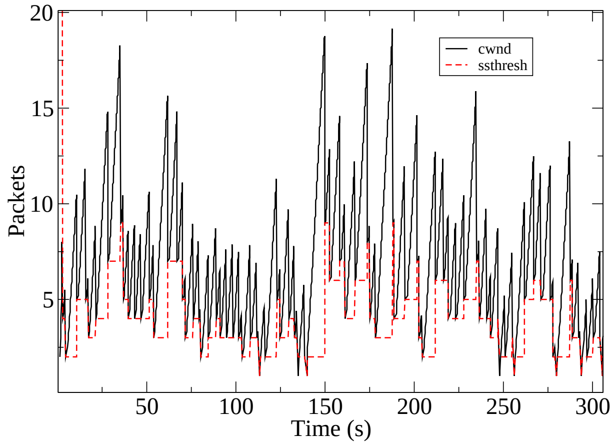

Figure 5: $cwnd$ and $ssthresh$ vs time for TCP.

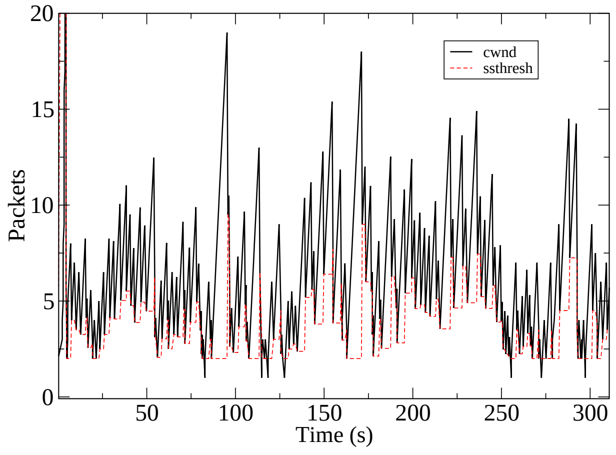

Figure 6: $cwnd$ and $ssthresh$ vs time for SCTP.

Seven *dupack*s were received between times 2.8624 and 2.8862. These are due to the seven packets, following the lost packet, which were received correctly.

Packet loss was detected on arrival of the third *dupack* at $t = 2.8703$ when packet number 7 was retransmitted using Fast Retransmit, and *cwnd* was reduced to four (half its previous value). At this point, there were five packets still unacknowledged in the network, that is, packet number 7 and four packets from the previous *cwnd*. On receipt of the fourth *dupack*, no new packet transmission could be done because the new *cwnd* of four did not allow another transmission, since the number of packets still unacknowledged was four. When the fifth *dupack* arrived, a new packet (no. 15) was transmitted because the number of unacknowledged packets had dropped down to three. Similarly, on arrival of the sixth and seventh *dupack*s, packets 16 and 17 were transmitted. Even though the first two *dupack*s received at $t = 2.8624$ and 2.8664 indicated that two packets had left the network, new packets were not clocked out with them. This was the case even though the *cwnd* allowed for transmission of new data when the

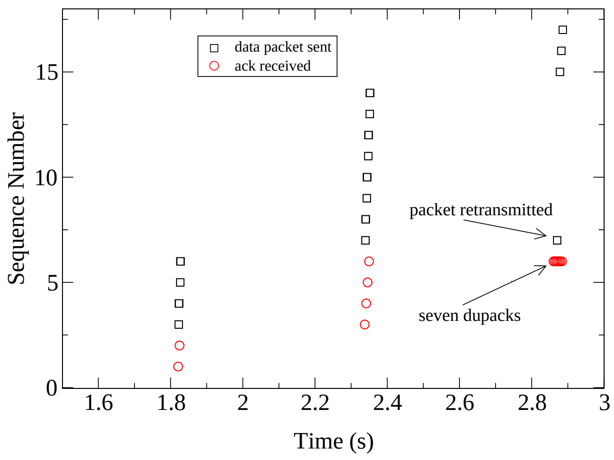

Figure 7: TCP retransmission of packet number 7.

two *dupacks* arrived. *This phenomenon occurred whenever packet losses were detected by the TCP source, and hampered its performance substantially.*

Fig. 8 depicts a comparable situation at the SCTP source. Packet number 63 was lost at $t = 141.6715$, when *cwnd* was 17 packets. Between $t = 142.1858$ and 142.2098, seven *dupack*s arrived at the source. As the first three *dupack*s arrived, three new packets (number 81, 82 and 83) were clocked out without any change to the *cwnd*, since the arrival of the *dupack*s indicated that three packets had left the network. When the fourth *dupack* arrived, the lost packet (number 63) was retransmitted and *cwnd* was reduced to eight (half the previous value). The lost packet was retransmitted even though the *flightsize* was equal to the *cwnd* at that time, leaving no room for another packet transmission or retransmission. Two points are worth noting here. First of all, *SCTP was able to retransmit a lost packet by ignoring the* flightsize *and* cwnd *at the moment of retransmission*. Secondly, *SCTP was able to clock out new data when the first three* dupacks *arrived*; the acks were not wasted. TCP, on the other hand, wasted the first two *dupacks* by not transmitting any data. For TCP, only when the third *dupack* arrived, it retransmitted its lost packet.

6 Conclusion

To better understand potential benefits and/or problems of incorporating SCTP into existing TCP networks, we have conducted an investigation on the behavior of networks in presence of TCP and SCTP sources. We have studied the impact of retransmission schemes on the performance of SCTP and TCP when they share the same satellite link.

We have shown that, for certain loss probabilities, SCTP can achieve higher throughput than TCP. Moreover, SCTP can take advantage of unused network capacity in a

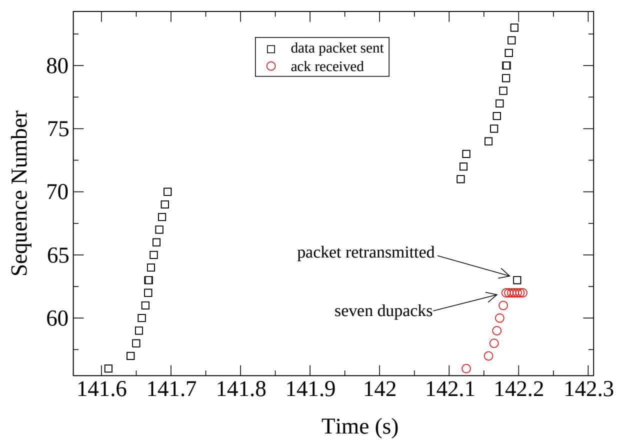

Figure 8: SCTP retransmission of packet number 63.

shared network. *It has been demonstrated that differences in the retransmission mechanisms of the two protocols are responsible for the higher throughput achieved by SCTP*. The difference can be attributed to the fact that SCTP can clock out new packets on receipt of the first three duplicate acknowledgements due to a lost packet. On the contrary, TCP stops transmitting any data between the arrival of the first duplicate acknowledgement and the retransmission of the lost packet. We *conclude that SCTP's retransmission scheme performs better than that of TCP in the case of links with errors, such as satellite links*. After retransmission TCP clocks out new data only if allowed by the congestion window.

References

[1] R. Stewart, Q. Xie, K. Morneault, and C. Sharp et. al., "Stream control transmission protocol." RFC 2960, Oct 2000.

[2] R. Brennan and T. Curran, "SCTP congestion control: Initial simulation studies," *International Teletraffic Congress (ITC 17)*, Brazil, 2001.

[3] N. Ghani and S. Dixit, "TCP/IP enhancements for satellite networks," *IEEE Communications Magazine*, vol. 37, no. 7, pp. 64–72, July 1999.

[4] C. Partridge and T. Shepard, "TCP/IP performance over satellite links," *IEEE Network*, vol. 11, no. 5, pp. 44–49, Sep-Oct 1997.

[5] S. Fu, M. Atiquzzaman, and W. Ivancic, "SCTP over satellite networks," *IEEE Computer Communications Workshop (CCW 2003)*, Dana Point, CA, pp. 112–116, Oct 2003.

[6] A. Jungmaier, E.P. Rathgeb, M. Schoop, and M. Tuxen, "SCTP - A multi-link end-to-end protocol for IP-based networks," *International Journal of Electronics and Communications*, vol. 55, no. 1, pp. 46–54, January 2001.

[7] R. Alamgir, M. Atiquzzaman, and W. Ivancic, "Impact of retransmission mechanisms on the performance of SCTP and TCP in a satellite network," *NASA Earth Science Technology Conference*, Pasadena, CA, June 2002.

[8] T. Ravier, R. Brennan, and T. Curran, "Experimental studies of SCTP multihoming," *First Joint IEI/IEE Symposium on Telecommunications Systems Research*, Dublin, Ireland, Nov 27, 2001.

[9] S. Fu and M. Atiquzzaman, "Improving end-to-end throughput of Mobile IP using SCTP," *2003 Workshop on High Performance Switching and Routing*, Torino, Italy, pp. 171–176, June 24-28, 2003.

[10] S. Fu, M. Atiquzzaman, and W. Ivancic, "Effect of delay spike on SCTP, TCP Reno, and Eifel in a wireless mobile environment," *International Conference on Computer Communications and Networks*, Miami, FL, pp. 575–578, October 2002.

[11] S. Fu and M. Atiquzzaman, "SCTP: State of the art in research, products, and technical challenges," *IEEE Communications Magazine*, vol. 42, no. 4, pp. 64–76, April 2004.

[12] R. Stewart and C. Metz, "New transport protocol for TCP/IP," *IEEE Internet Computing*, vol. 5, no. 6, pp. 64–69, Nov/Dec 2001.

[13] A.L. Caro, J.R. Iyengar, and P.D. Amer et al., "SCTP: A proposed standard for robust internet data transport," *IEEE Computer*, vol. 36, no. 11, pp. 56–63, Nov 2003.

[14] A. Jungmaier, M. Schopp, and M. Tuxen, "Performance evaluation of the Stream Control Transmission Protocol," *High Performance Switching and Routing*, Germany, pp. 141–148, June 26-29, 2000.

[15] M. Allman, V. Paxon, and W. Stevens, "TCP congestion control." RFC 2581, April 1999.

[16] W.Stevens, "TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms." RFC 2001, January 1997.

[17] "ns-2 network simulator." www.isi.edu/nsnam/ns/.

[18] T. Henderson, E. Sahouria, S. McCanne, and R.H. Katz, "On improving the fairness of TCP congestion avoidance," *IEEE Globecom*, Sydney, Australia, pp. 539–544, Nov 8-12, 1998.