

Evaluation of SCTP Multistreaming over Satellite Links

Mohammed Atiquzzaman

School of Computer Science
University of Oklahoma,
Norman, OK 73019-6151
Email: atiq@ieee.org

William Ivancic

NASA Glenn Research Center
21000 Brookpark Rd. MS 54-8,
Cleveland, OH 44135.

Abstract—The Stream Control Transmission Protocol has been developed as a reliable transport protocol to carry PSTN signaling messages over an IP network. Multistreaming is one of its powerful features to overcome some of the bottlenecks of single streamed protocols such as TCP. In this paper, we study the impact of multistreaming on the performance of SCTP over error prone satellite networks. We show that multistreaming results in higher goodput and reduces the buffer requirements at the wireless receiver.

I. INTRODUCTION

To enable transfer of signaling traffic over IP networks [1], a family of protocols [2] is being developed based on SCTP [3], a new end to end message-based reliable transport protocol. SCTP provides an important element for the convergence of voice and data networks. SCTP removes some of the restrictions of TCP, such as strict byte ordered delivery which is too limiting in the case of signaling where mutually independent transactions may be included within each packet; TCP’s strict byte order of delivery in such cases is not only unnecessary but also reduces the throughput by introducing head of line blocking. Although initially conceived as a reliable transport protocol to carry PSTN signaling messages over IP, it has emerged into a highly reliable and fault-tolerant transport protocol which has the potential to be an alternative to TCP in the future.

Multistreaming, one of the powerful features of SCTP, allows data from a number of upper layer applications to be multiplexed onto one channel (called association in SCTP) as shown in Fig. 1. Congestion control is applied to the association instead of individual streams. Sequencing of data is done within a stream; if a packet belonging to a certain stream is lost, packets (from that stream) following the lost one will be stored in the receiver’s stream buffer until the lost packet is retransmitted from the source. However, data from other streams can still be passed to upper layer applications. This avoids the head of line blocking found in TCP where only one stream carries data from all the different upper layer applications.

The head of line blocking at the individual streams in shown in Fig. 2 with four applications corresponding to the four streams. Packets are identified by Stream Sequence Numbers (SSN) [3] which are unique within a stream. SSN 11 has been delivered to application number 1 and SSN 12 is arriving at the buffer of stream 1. SSN 9 for the second stream is lost; SSNs 10, 11, 12 are there-

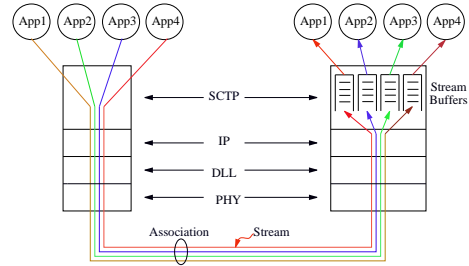


Fig. 1. An SCTP association consisting of four streams carrying data from four upper layer applications.

fore queued in the buffer of the second stream. Arriving SSN 13 will also be queued. Similarly SSN 3 of stream 3 is missing. For application 4, SSN 21 is being delivered to the application while arriving SSN 23 will be queued in the buffer because of missing SSN 22. This illustrates the fact that packets arriving on stream 1 can still be delivered to the upper layer application although stream 2 and 3 are (and stream 4 will be) blocked because of lost packets.

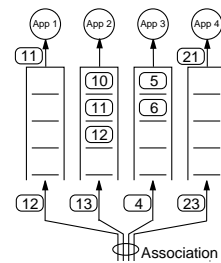


Fig. 2. Illustration showing head of line blocking of individual streams at the receiver.

It is anticipated that multistreaming will allow web like applications, where multiple objects (such as images) are typically downloaded concurrently, to run faster than using TCP. The inherent multistreaming capability of SCTP will also speed up access to image databases [4] and transmission of multimedia traffic [5]. It will even speedup the downloading of a single image when it is compressed using multiresolution schemes (such as EZW or progressive JPEG) [6].

The use of inline images and other associated data often requires a client to make multiple requests of the same server in a short amount of time. The HTTP/1.1 protocol requires implementing persistent connections where

the same TCP connection is used for many URL fetches. Future HTTP protocols could use the multistreaming feature of SCTP to allow fast downloading of web pages by simultaneously downloading different objects over different streams belonging to the *same association*.

The amount of data that can be sent by a TCP (or SCTP) sender depends on the congestion window and the receiver buffer size. When a packet is lost in the network, subsequent packets are queued up in the receiver buffer for resequencing until the lost packet is retransmitted from the source and arrives at the receiver. In the case of limited receiver buffer size, the receiver could run out of buffer space due to head of line blocking resulting in the sender being unable to send data. In the case of multistreaming, some of the streams could be delivering packets to the upper layer when a particular stream is blocked while waiting for a lost packet to arrive. *Multistreaming could therefore result in higher throughput and reduced buffer requirements than single streaming.*

The *objective* of this paper is to study the performance of SCTP in a satellite network which is characterized by errors and long propagation delays. In particular, we investigate the effect of multistreaming on the goodput of SCTP, and the buffer requirements at the receiver which is connected using a satellite link. The *contributions* of this work are as follows:

- Determine the goodput of SCTP over satellite networks for various link error rates and receiver buffer sizes.
- Demonstrate that multistreaming can significantly reduce the receiver buffer size requirements which would be helpful in designing wireless handheld devices.
- Determine the optimal buffer requirement at the receiver which can be used to dimension the receiver buffer.

The rest of the paper is organized as follows. Differences between TCP and SCTP congestion control are discussed in Sec. II followed by definitions and notations used in this paper in Sec. III. The simulation topology and the assumptions used are described in Sec. IV followed by results in Sec. V. Finally, summary of our findings and conclusions of this work are presented in Sec. VI.

II. CONGESTION CONTROL OF TCP AND SCTP

In this section, we describe the TCP and SCTP congestion control schemes. Differences between them can be found in [7].

A. TCP Congestion control

The rate-adaptive window-based congestion control scheme of TCP throttles back its transmission rate in response to packet loss which can be detected by timeout or duplicate acknowledgements (DUP ACK) from the receiver. Slow start [8] is invoked when packet loss is detected by timeout, and duplicate acknowledgement results in Fast Retransmit [8]

Multiple packet losses within the same RTT can not be recovered by DUP ACKs. To solve this problem, SACK [9] can be used to determine all packets within an RTT which

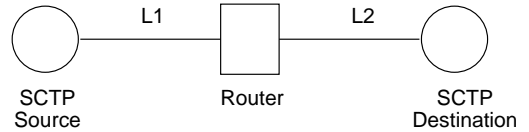


Fig. 3. Network simulation topology.

are missing. Fast Retransmit can then be used to recover these multiple packets lost within an RTT.

B. SCTP Congestion control

The congestion control of SCTP is "based" on schemes similar to those of TCP such as slow start, congestion avoidance, etc. This ensures fairness for both protocols as they work together in the Internet. SCTP provides reliable transmission and detects lost, reordered, duplicated or corrupt packets. It provides reliability by retransmitting lost or corrupt packets. In the next section, we describe the differences between TCP and SCTP congestion control mechanisms.

III. DEFINITIONS AND NOTATIONS

We define the following terms which will be used to describe the results in the following sections:

- *Goodput* is defined as the total number of packets (without considering the retransmitted packets) reaching the destination during the simulation period. In our experiments, we measure the goodput by the highest numbered TSN reaching the destination during the simulation time.
- *Optimal receiver buffer size* is the smallest amount of receiver buffer for which the SCTP goodput is independent of the receiver buffer size.

We use the following notations in the rest of the paper.

- s = number of streams per SCTP association.
- r_i = transmission rate of link i in Mbps.
- d_i = propagation delay of link i in msec.
- ϵ_i = error rate of link i . It is the probability that a packet is lost in the link due to errors.
- B = Receiver buffer size in bytes.
- a_rwnd = Advertised receiver window size in bytes.
- $cwnd$ = Congestion window size in bytes.

IV. SIMULATION SETUP AND ASSUMPTIONS

We have used *ns* simulator [10] with an SCTP patch from the University of Delaware to evaluate the performance of SCTP multistreaming over a satellite link. The network topology for the simulation is shown in Figure 3 where an SCTP source sends one way traffic to an SCTP sink through a router, possibly located in the satellite. $L_i, 1 \leq i \leq 2$ are satellite links whose transmission rate, propagation delay and random error rate are expressed by the tuple (r_i, d_i, ϵ_i) .

We make the following assumptions regarding the simulation setup:

- Data transfers are long. The SCTP source has an infinite supply of data which is being transferred to the destination using *ftp*.

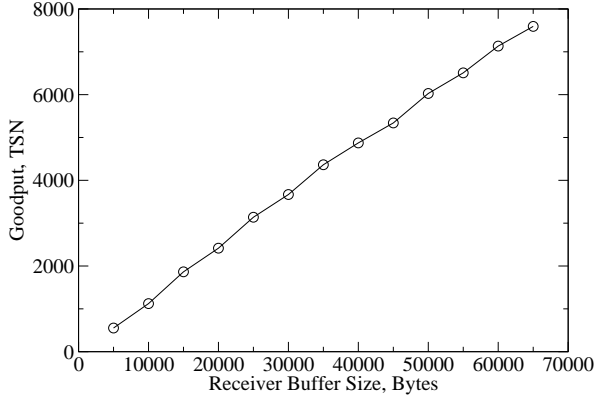


Fig. 4. Goodput as a function of receiver buffer size for $\epsilon_2 = 0$ and $s = 4$.

- Errors in the satellite links are lumped at L_2 , i.e. $\epsilon_1 = 0$ and $\epsilon_2 \geq 0$.
- Packets are of fixed length, and are of one MTU.
- The upper layer at the destination is always ready to accept data.

V. RESULTS

In this section, we first present the results for the performance of SCTP in the absence and presence of errors. We then show the improvement of SCTP goodput due to multistreaming for various error rates and receiver buffer sizes. Finally, we determine the optimal receiver buffer size for different error rates and number of streams. Throughout this study, we have used $r_1 = 5$, $r_2 = 10$, $d_1 = d_2 = 130$, and $\epsilon_1 = 0$. The values of r_1 and r_2 were chosen to avoid any congestion loss in the network, thereby leaving all possible losses due to satellite link errors.

A. Effect of Zero Errors

Fig. 4 shows SCTP goodput as a function of B for $s = 4$ and $\epsilon_2 = 0$. In the absence of packet losses, there is no blocking at the destination. $cwnd$ initially increases and then becomes constant when it reaches B . After this point, the source sends a burst of $\lfloor \frac{B}{MTU} \rfloor$ packets every RTT as seen in Fig. 5 which plots the TSNs of the packets leaving and Cumulative TSN of the ACKs arriving at the source. The y-axis shows (TSN mod 100). The goodput therefore, depends directly on the receiver buffer size as seen in Fig. 4.

In the absence of packet losses, there is no blocking at the destination buffer. The goodput is therefore independent of the number of streams, i.e. although Fig. 4 is shown for $s = 4$, it applies for any value of s .

B. Effect of Errors

Fig. 6 shows the goodput as a function of the receiver buffer size for $s = 4$ and $\epsilon_2 = 0.01, 0.03$ and 0.05 . For a particular error rate, the goodput initially increases as

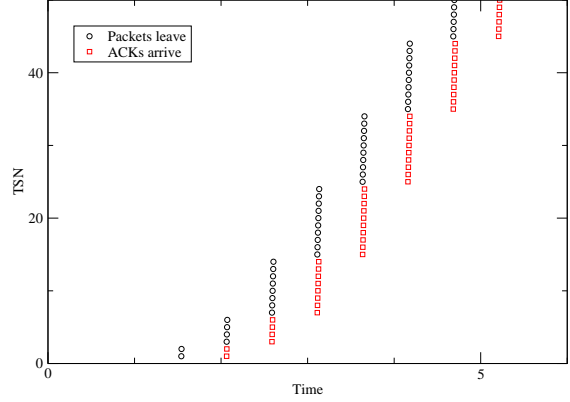


Fig. 5. Packets sent and acknowledgements received at the source for $\epsilon_2 = 0$, $s = 4$ and $B = 15K$.

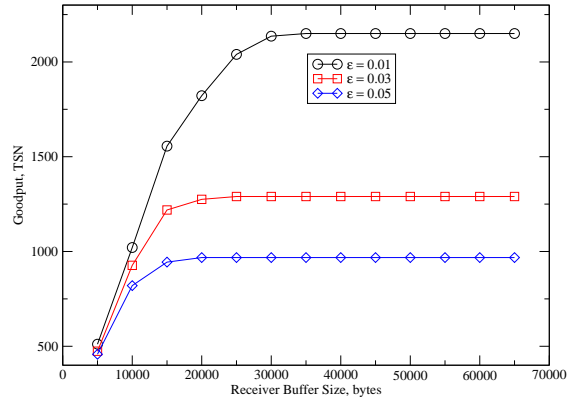


Fig. 6. Goodput as function of receiver buffer size for different error rates and $s = 4$.

B increases indicating that the goodput is constrained by the receiver buffer size. This is evident from Fig. 7 which shows $cwnd$ and a_rwnd for $s = 4$, $\epsilon_2 = 0.01$ and $B = 15K$. $B = 15K$ was chosen to make the goodput of the connection constrained by the receiver buffer size as seen by frequent dropping of a_rwnd below one MTU which in turn restricts the increase of $cwnd$ beyond 15K.

As B increases (Fig. 6), a point is reached after which any further increase of B does not have any effect on the goodput; at this point, the goodput is limited by the congestion control mechanism of SCTP invoked by packet losses due to link errors. Further increase in goodput can only be achieved by lowering the link error rate as seen in Fig. 6. The fact that the goodput is limited by $cwnd$ for large values of B is evident from Fig. 8 which shows $cwnd$ and a_rwnd for $s = 4$, $\epsilon_2 = 0.01$ and $B = 35K$. It is seen that a_rwnd falls below one MTU only once, while at other times, the decrease of $cwnd$ (which is a measure of goodput) is governed by SCTP's congestion control mechanisms

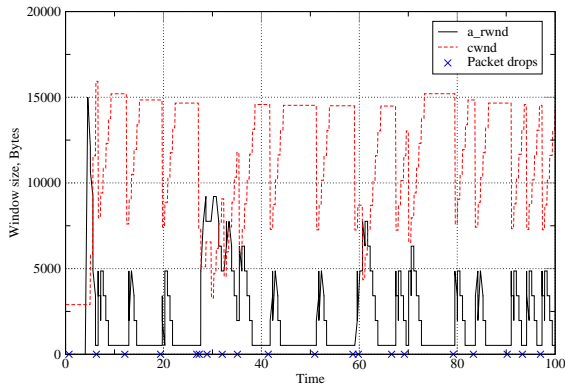


Fig. 7. Advertised receiver window and congestion window with $B = 15K$, $\epsilon_2 = 0.01$, and $s = 4$.

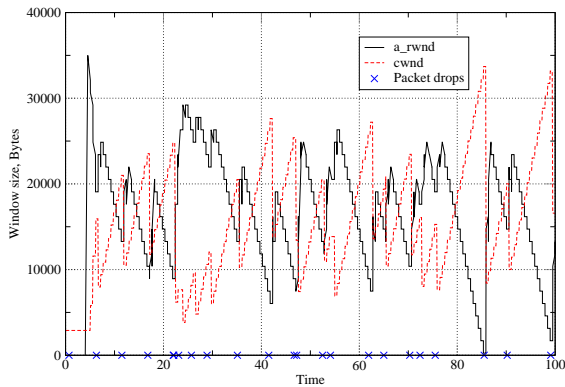


Fig. 8. Advertised receiver window and congestion window with $B = 35K$, $\epsilon_2 = 0.01$, and $s = 4$.

(slow start, congestion avoidance, fast retransmit, etc.) after packet losses. Our observation that the goodput at the flat region of the curves in Fig. 6 is limited by the congestion control mechanism of SCTP is validated by the fact that a_rwnd never fell below one MTU in the flat region of the goodput curve; i.e. the receiver buffer size was not a limiting factor in the source's ability to send data.

Fig. 9 shows packets sent and ACKs received at the source. We can see long delays in retransmitting lost packets while waiting for DUP ACKs resulting from the lost packets. These long delays and the drop of $cwnd$ result in reduced goodput when receiver buffer is not a constraint.

C. Effect of Multistreaming

We have shown in the previous sections that the receiver buffer size could be a limiting factor in the case of small buffers. In this section, we show how multistreaming reduces the receiver buffer requirements and increases the goodput by avoiding HOL blocking at the receiver.

Figure 10 shows the goodput of an SCTP connection us-

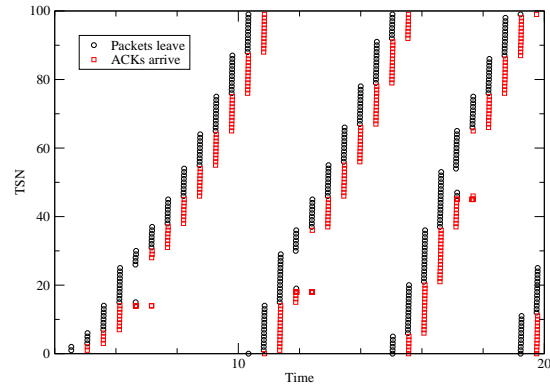


Fig. 9. Packets sent and acknowledgements received at the source for $\epsilon_2 = 0.01$, $s = 4$ and $B = 35K$.

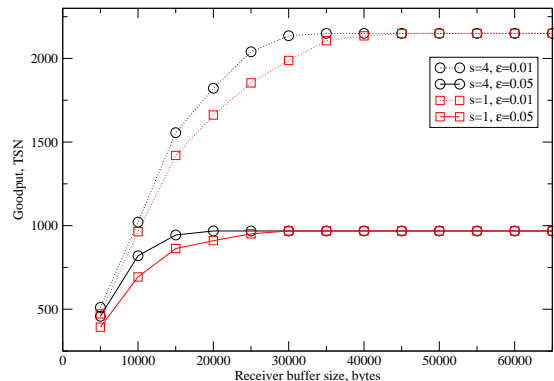


Fig. 10. Goodput as function of receiver buffer size for various wireless error rates with one and four streams.

ing one and four streams for error rates of 1% and 5% in the bottleneck link. The goodput of SCTP with four streams per connection is better than one stream per connection for small receiver buffer sizes. This is because, for small receiver buffer sizes and a single-stream per association, the sender is often prevented from sending packets because of lack of receiver buffer space arising due to HOL blocking. However, in the case of multiple streams per association, some of the streams may be able to pass packets to the upper layers even though other streams are blocked due to lost packets belonging to those streams. Fewer packets are dropped when the error rate is low. This results in higher goodput due to a smaller chance of blocking at the receiver.

Figure 11 compares the goodput of SCTP for one and four streams as a function of the error rate for $B = 15K$. The higher goodput of multistreaming is shown for various error rates. As an example, four streams per association results in about 10% increase in throughput over one stream at

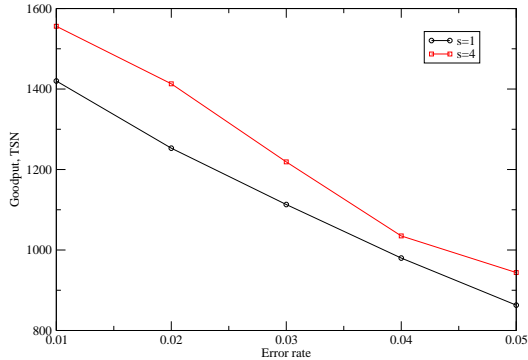


Fig. 11. Goodput as function of error for one and four streams for

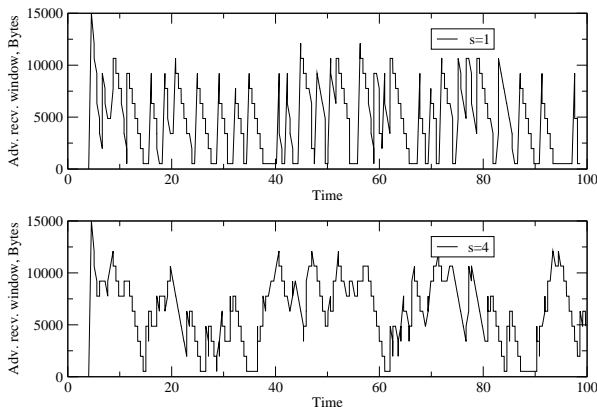


Fig. 12. Advertised receiver window for one and four streams. $\epsilon_2 = 0.05$, $B = 15K$.

3% error rate. A small receiver buffer size (15K) was chosen to demonstrate the *advantage of multistreaming when the buffer size is limited*.

The advertised receiver window for one and four streams is shown in Fig. 12 for $\epsilon_2 = 5\%$ and $B = 15K$. It is seen that, for a single stream, *awnd* frequently falls below 1500 bytes (one MTU) thereby restricting the sender from sending packets. In the case of four streams, the advertised receiver buffer size falling below one MTU is less frequent than the case of a single stream. This results in higher goodput for the case of multistreaming. Note that one MTU is the minimum size required by the sender to send data.

D. Optimal Receiver Buffer Size

It has been shown in Sec. V-B that as the receiver buffer size is increased, the goodput of SCTP in the presence of errors (see Fig. 6) becomes independent of the receiver buffer size. In this section, we determine the *optimal receiver buffer size* (see Sec. III for definition) for various error rates. Receiver buffer size larger than the optimal

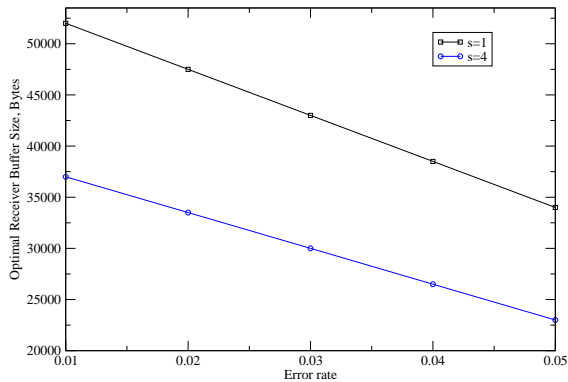


Fig. 13. Optimal receiver buffer size vs. error rate for one and four streams.

receiver buffer size does not contribute to increasing the goodput, and hence is wasted. The optimal receiver buffer size can be used to dimension the buffer at the receiver for portable mobile devices (such as PDAs and next generation wireless phones) where it is desirable to reduce the memory size because of weight and power restrictions.

Figure 13 shows the optimal receiver buffer size as a function of the error rate for one and four streams. The buffer size required for multistreaming is significantly lower than for single streams. For example, four streams result in a saving of about 35% of receiver buffer space over a single stream for an error rate of 1%. Similar saving is also evident for other error rates. The optimal receiver buffer size was determined by the receiver buffer size for which the goodput became constant (i.e. independent of the receiver buffer size) in Fig. 6.

Figure 13 also shows that the optimal receiver buffer size is smaller for higher error rates. This is consistent with our observation in Sec. V-B that *as the error rate increases, the goodput is increasingly dominated by the congestion control mechanism, and to a lesser extent by the receiver buffer size*.

VI. CONCLUSIONS

Multistreaming allows multiple applications to transmit data over an association using streams. We have shown that multistreaming improves the goodput of SCTP for the case of limited receiver buffer size in the presence of errors in a satellite link. With the proliferation of handheld devices operating over wireless (or satellite/radio) links having limited receiver buffer capacity, SCTP will perform better than TCP (which uses a single stream) for such devices.

The results presented in this paper also demonstrate that multistreaming can reduce the buffer requirements at the receiver, and hence would be attractive for wireless handheld devices where the reduction of weight and power requirements is of utmost importance.

REFERENCES

- [1] M. Hassan, A. Nayandoro, and M. Atiquzzaman, "Internet telephony: Services, technical challenges and products," *IEEE Communications Magazine*, vol. 38, no. 4, pp. 96–103, April 2000.
- [2] L. Ong, I. Rytina, M. Garcia, and H. Schwarzbauer et. al., "Framework architecture for signaling transport." RFC 2719, Oct 1999.
- [3] R. Stewart, Q. Xie, K. Morneault, and C. Sharp et. al., "Stream control transmission protocol." RFC 2960, Oct 2000.
- [4] E.A. Heredia, C. Teng, and M. Ozkan, "Using multiresolution and multistreaming for faster access in image database broadcast," *1998 International Conference on Image Processing*, Chicago, IL, pp. 784–788, Oct 4-7, 1998.
- [5] R. Simon, T. Znati, and R. Sciabassi, "XCAP: a multistream routing algorithm for multimedia traffic," *Third IEEE International Conference on Multimedia Computing and Systems*, Hiroshima, Japan, pp. 104–107, June 17-23 1996.
- [6] P.T. Conrad and G.J. Heinz, "SCTP in battlefield networks," *MILCOM 2001*, Washington, DC, pp. 289–295, Oct 2001.
- [7] R. Brennan and T. Curran, "SCTP congestion control: Initial simulation studies," *International Teletraffic Congress (ITC 17)*, Brazil, 2001.
- [8] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control." RFC 2581, April 1999.
- [9] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options." RFC 2018, Oct 1996.
- [10] "ns-2 network simulator." www.isi.edu/nsnam/ns/.