

# Effect of Congestion Control on the Performance of TCP and SCTP over Satellite Networks

Rumana Alamgir  
 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**—Stream Control Transmission Protocol (SCTP) is a new transport layer protocol which can be deployed in the Internet along with TCP. In this paper, we investigate the performance issues that arise when SCTP and TCP are used in the same satellite network. We evaluate performance through measurement of throughput, and also fairness of use of networks resources by the two protocols. We observed that SCTP is fair, although SCTP always achieved slightly higher throughput than TCP. We have analyzed the results to show that differences in the congestion control mechanism of TCP and SCTP are responsible for the higher throughput attained by SCTP.

## I. INTRODUCTION

The Internet Protocol suite glues together a large number of different computer systems and networks. The IP protocol can support many different transport layer protocols, with TCP being the most widely used protocol. Recently, the Stream Control Transmission Protocol (SCTP) [1] has been standardized by the IETF as a reliable transport layer protocol for carrying Public Switched Telephone Network (PSTN) signalling messages over IP networks. However, its advanced congestion control and fault tolerant features also make it suitable for carrying data in computer networks, for which it has already been proposed as an alternative to TCP [2]. SCTP is essentially a reliable, message-oriented data transport protocol that supports multiple streams (called *multistreaming* in SCTP) within an association, and hosts with multiple network addresses (called *multihoming*) [3]. SCTP is particularly valuable to applications where monitoring and detection of loss of session is required. For such applications, the SCTP path/session failure detection mechanisms, especially the heartbeat, will actively monitor the connectivity of the session. The following services are also provided to its users [4]:

1. Acknowledged error-free non-duplicated transfer of user data.
2. Data fragmentation to conform to discovered path Maximum Transmission Unit (MTU) size.
3. Sequenced delivery of user messages within multiple streams, with an option for order-of-arrival delivery of individual user messages.
4. Optional bundling of multiple user messages into a single SCTP packet.
5. Network-level fault tolerance through support of multihoming at either or both ends of an association.

6. Resistance to flooding and masquerade attacks.

SCTP has an advanced congestion control mechanism, that is used to recover from segment losses effectively and efficiently. Many aspects of the congestion control of SCTP are similar to those of TCP, though SCTP also offers features such as byte-oriented congestion window (instead of segment-oriented, as in TCP). SCTP can also perform a rapid recovery of lost segments in an error-prone network; it can therefore *transmit segments at a faster rate* than TCP as described in Secs. V-B and V-C. Further differences in congestion control that are advantageous to SCTP are detailed in Sec. II.

The faster rate achieved by SCTP sources than TCP sources raises the issue of *fairness* among the protocols when they are deployed in the same network. Fairness measures the distribution of network resources among the sources using different protocols such as SCTP and TCP. Maintaining fairness among multiple transport protocols in the network is essential in the uniform distribution of network resources, and the widespread acceptance of the network itself [5]. Fairness can be dealt with not only at the transport level by the senders and receivers, but also at the network level. Fairness can be significantly improved by the participation of the network (eg. through some fair queuing strategy at routers). The *aim* of this paper is to study fairness issues resulting at the transport layer, from the protocols SCTP and TCP sharing a common network.

In this paper, we *focus on networks with large delay-bandwidth product links*, such as satellite networks. The delays in satellite networks are influenced by several factors, the main one being the orbit type [6]. 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. Problems arising when TCP is run over satellite links have been studied by many researchers [7], [8], and solutions to the problems have been also been reported in the literature [9]. In this paper, we consider SCTP over GEO satellite links.

The *objective* of this paper is two-fold. First, we determine if one protocol achieves a better throughput than another, and what the level of any resulting unfairness is, for various number of TCP and SCTP sources in the same satellite network. We then aim to determine the impact of the advanced congestion control mechanism of SCTP on

the results obtained from the first part of our objective.

Results of preliminary and in-depth studies on various features of SCTP have been reported in the literature. Jungmaier [10] studies the performance of SCTP in a satellite network using a backup terrestrial link and showed that multihoming allows SCTP to use its secondary link during times of congestion. Experimental studies have been used by Ravier et.al. [11] to show that multihoming increases the fault tolerance of an SCTP association. Brennan et.al. [3] have shown that some features of SCTP congestion control may hamper its performance, and also cause it to be aggressive. The performance of SCTP competing for resources with TCP in a loss-free network has been studied by Jungmaier et.al. [12], and it has been shown that in such a scenario, allocation of network resources between the protocols is fair. The impact of the retransmission mechanism of SCTP on the performance of a long delay network shared with TCP hosts is examined in [13]. The above mentioned work did not carry out any detailed performance and fairness study of a combined TCP and SCTP, error prone satellite network. The study presented in this paper *differs* from previous studies in that it considers the performance of TCP and SCTP in an error prone satellite network and also presents a detailed analysis of the effect of the congestion control mechanisms on the performance of TCP and SCTP in such a network.

We have carried out simulations to accomplish our objective. TCP and SCTP sources shared the same long-delay network, and the throughput and fairness were obtained from the simulations. Network simulator *ns 2.1*, with a SCTP patch from the University of Delaware, was used to conduct the simulation experiments. We have found that SCTP achieves a considerably higher throughput than TCP, although a high degree of fairness is exhibited by the protocols. We have examined the congestion control mechanisms of the two protocols in detail, and found that SCTP has inherent congestion control properties that allow it to achieve higher throughput.

The *contributions* of this paper can be summarized as follows.

1. We have shown that TCP and SCTP are fair in sharing resources in a satellite network.
2. Our results show that although the fairness is high, SCTP achieves a higher throughput as compared to TCP.
3. We have demonstrated that SCTP congestion control results in a larger average congestion window and faster recovery after segment loss, which is responsible for the higher throughput of SCTP.

The remainder of this paper is organized as follows. SCTP association and its congestion control mechanisms are described in Sec. II. The network topology, parameters and configurations used in our simulations are detailed in Sec. III. Sec. IV provides the metrics used to measure the performance of TCP and SCTP. Results on throughput and fairness of SCTP along with a detailed interpretation of the results are presented in Sec. V, while Sec. VI discusses the case when the satellite link is fully utilized. Finally, section VII summarizes the conclusions that have been drawn

from this study.

## II. SCTP ASSOCIATION AND CONGESTION CONTROL

An *association* in SCTP is analogous to a TCP *connection*. An SCTP data source is able to transfer segments to the destination in the context of an association. Data is transmitted between the source and destination in the form of segments, that contain a common header and a sequence of structures called chunks [14]. During setup of association, various information is exchanged between the two participating nodes. Each data chunk transmitted is assigned a unique 32-bit Transmission Sequence Number (TSN), which is one larger than that of the previous data chunk sent.

SCTP uses an end-to-end window based flow and congestion control mechanism similar to the one that is well known from TCP [15]. Certain extensions of the TCP congestion control mechanism have been incorporated to accommodate the *multihoming* aspect of SCTP, and the message-based (rather than the stream-based) nature of the protocol. The data receiver may control the rate at which the sender is sending by returning a receiver window size (*rwnd*) along with all SACK chunks. The sender itself keeps a variable known as the congestion window (*cwnd*) that controls the maximum number of outstanding bytes (i.e. bytes that may be sent before they are acknowledged). The receiver must acknowledge all data chunks; the receiver may however, wait (maximum of 200 ms) before sending the acknowledgement.

### A. Overview of SCTP Congestion Control

As in TCP, congestion control of SCTP has two modes, slow-start and congestion-avoidance. The mode is determined by the set of congestion control variables, and these are path specific. A path of an SCTP multihoming [10] association is one of the connections set up between a source and a destination. So, while the transmission to the primary path may be in the congestion-avoidance mode, the implementation may still use slow-start for the backup path(s).

For successfully delivered and acknowledged data, the *cwnd* is steadily increased, and once it exceeds a certain boundary (called slow-start-threshold, *ssthresh*), the mode changes from slow-start to congestion-avoidance. In slow-start, the *cwnd* is increased faster (roughly one MTU per received SACK chunk), and in congestion-avoidance mode, it is only increased by roughly one MTU per round-trip-time, (*rtt*). During both the modes, a variable called *flight-size* keeps track of the total number of bytes that have not yet been acknowledged. During congestion-avoidance, the variable called Partial Bytes Acknowledged (*pba*), keeps count of the number of bytes in the current *cwnd* that have been acknowledged so far, and this variable decides whether the *cwnd* should be increased with the next transmission. Events that trigger retransmission (timeouts or Fast Retransmission [15]) cause the *ssthresh* to be cut down drastically, and reset the *cwnd*. A timeout causes a new slow-start with *cwnd*=1 MTU, and a Fast Retransmit halves

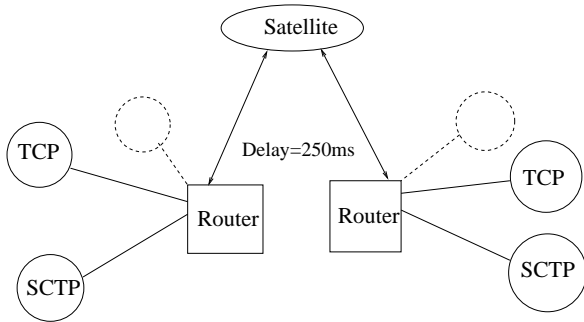


Fig. 1. Satellite Network.

the  $cwnd$  and sets  $ssthresh=cwnd$ . An SCTP source determines that a Fast Retransmit of a lost segment is required by the arrival of four consecutive duplicate acknowledgements (*dupack*) of the last received segment.

### B. Distinctions Between TCP and SCTP Congestion Control

There are subtle differences between the congestion control mechanisms of TCP and SCTP. The congestion control properties of SCTP that are different from those of TCP are as follows [3]:

1. Congestion window is increased according to the number of bytes acknowledged, not the number of acknowledgements received. Similarly, *flightsize* is decreased by the number of bytes acknowledged.
2. Without gap block generation in the SACK, there is no way of determining segment losses other than a timeout.
3. No fast recovery phase is required, since no artificial inflation of the  $cwnd$  is required for throughput to be maintained.
4. When  $cwnd=ssthresh$ , slow-start is followed.
5. During congestion-avoidance,  $cwnd$  may only be increased if the full  $cwnd$  is currently being used. This allows an SCTP source to limit its  $cwnd$  to the initial  $rwnd$ .
6. An unlimited number of GAP ACK blocks are allowed in SCTP. TCP can allow a maximum of three SACK blocks.

The list is not complete, though the *items listed above contribute significantly to the results presented in this paper*. The following section describes the setup used to conduct the experiments.

## III. EXPERIMENTAL SETUP

In this paper, we assume a number of TCP and SCTP sources connected via a satellite link to the corresponding destinations as shown in Fig. 1. The satellite link has a one-way propagation delay of 250ms which corresponds to the delay of a GEO satellite link.

### A. Network Topology

The network topology of Fig. 1 was modelled using the *ns 2.1* network simulator [16] as shown in Fig. 2. All the results were obtained from implementations of TCP and SCTP in the network simulator with an SCTP patch from the University of Delaware.

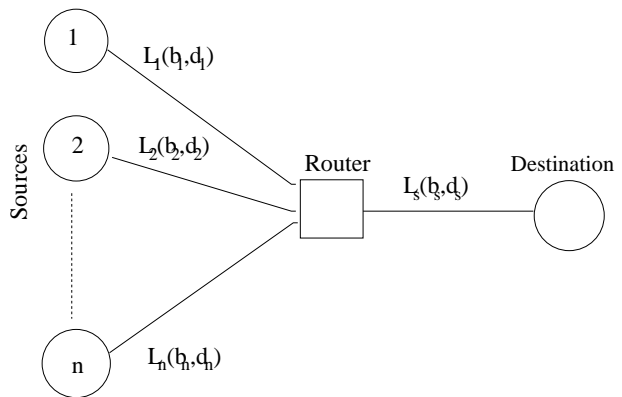


Fig. 2. Simulation Model.

No of Sources	Router Buffer Size (segments)	$L_s$ Bandwidth (Mbps)
2	10	2
4	40	4
6	80	7
8	160	11

TABLE I

ROUTER BUFFER SIZES AND BOTTLENECK BANDWIDTHS.

TCP and SCTP sources (labelled 1 to  $n$ , where  $n \geq 2$  and  $n$  is an even number) send segments through links  $L_1$  to  $L_n$  to a destination node, a constant distance away. Each link has a bandwidth and delay, shown in the diagram by the tuple  $(b_x, d_x)$ , where  $1 \leq x \leq n$ . A router between the sources and satellite link queues incoming segments from links  $L_1$  to  $L_n$ , and then transmits them along the bottleneck link  $L_s$ . Though the topology uses one destination node, it has  $n$  separate agents to establish a connection with each of the sources 1 to  $n$ .

### B. Network Parameters

Several simulations were run, starting from one TCP and one SCTP source to four TCP and four SCTP sources, with an equal number of TCP and SCTP sources for every run. The propagation delays of links  $L_1$  to  $L_n$  were set at 2ms, while the delay of  $L_s$  was 250ms. To ensure fairness to the sources before the segments arrived at the router, the parameters of links  $L_1$  to  $L_n$  were kept the same.

As the number of sources were increased, the router buffer size and the bottleneck bandwidth were also increased to accommodate the increased segment arrival rate at the router. The router buffer size and satellite link bandwidth combinations used for different number of sources are listed in Table I.

Data corruption in the satellite link was simulated by error-module at the link which dropped packets with 1.5% probability. To take advantage of the large delay-bandwidth product of the satellite link, the buffer sizes of both the TCP and SCTP receivers was set to a large value of 64KB. One-way large file transfer traffic was generated

by using *ftp* traffic generator at the sources.

### C. TCP and SCTP Source Configurations

We wanted to study the congestion control schemes of TCP and SCTP in this study. We therefore, configured the TCP and SCTP hosts as similar to each other as possible as given below.

1. Selective Acknowledgement (SACK) is mandatory for SCTP; The SACK option of TCP was therefore enabled.
2. Delayed acknowledgement was not used in either TCP or SCTP.
3. Since TCP uses only one connection between the source and destination, SCTP was configured to use one stream per association.
4. The payload of each segment (that is, without headers) was 1488 bytes for both protocols.
5. The initial *rwnd* for both was set to 64KB.
6. The initial *ssthresh* was made equal to *rwnd* for both protocols.

## IV. PERFORMANCE METRICS

This section provides details of the measures used to study the performance of the simulated networks. Though there exist many metrics for quantifying fairness, a standard does not exist. This paper uses a *fair share per link* metric [17], as given by Jain in [18]. Fairness, given by  $\omega$ , is computed as follows:

$$\omega = \frac{(\sum_{i=1}^n b_i)^2}{n * (\sum_{i=1}^n b_i^2)} \quad (1)$$

where,  $n$  is the number of flows through the bottleneck link, and  $b_i$  is the fraction of the bottleneck link bandwidth obtained by flow  $i$ . The value of fairness obtained through this method ranges from  $1/n$  to 1, with 1 indicating equal allocation to all sources.

For a two sources case, we define the *Percentage Increase in Throughput* ( $\delta$ ) of source 1 over source 2 by:

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

where  $\lambda_i$  is the throughput of source  $i$ . *Link utilization*  $\psi$  of the bottleneck link  $L_2$  was calculated as [17]:

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

where  $b_s$  is the bottleneck link bandwidth (see Fig. 2).

## V. RESULTS

This section presents the results obtained from simulations and the analysis of the results. Most of the graphs presented depict a steady-state of the simulation (approximately after the first 50 seconds of simulation).

Four scenarios were studied and each simulated for 500 seconds. In the first scenario,  $n = 2$ , i.e. one TCP and one SCTP source. For each consecutive scenario,  $n$  was increased by two, so in scenario four,  $n = 8$ . As mentioned in Sec. III-B, there was an equal number of TCP and SCTP sources.

No of Sources	% Incr in Thruput	Link Utlzn (%)	Fairness
2	23.9	21.5	0.988
4	26.1	21.6	0.986
6	19.24	18.4	0.992
8	21.6	15.8	0.989

TABLE II  
PERCENTAGE INCREASE IN THROUGHPUT, LINK UTILIZATION AND FAIRNESS FOR THE FOUR SCENARIOS.

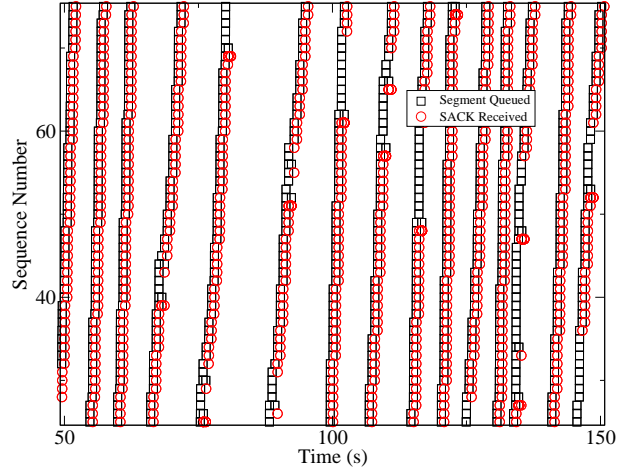


Fig. 3. Time-Sequence for TCP.

### A. Percentage Increase in Throughput, Link Utilization and Fairness

Table II shows the *percentage increase in throughput* of SCTP over TCP, the link utilization and fairness for each scenario. Percentage increase in throughput was obtained using Eqn. 2, link utilization from Eqn. 3 and fairness from Eqn. 1. The table shows that there is a positive percentage increase ranging from 19.24% to 26.1% for all the four scenarios. This indicates that SCTP is able to transmit more data than TCP in the same time range.

Due to space constraints, the remainder of this *Results* section presents the results obtained from the two-source scenario. Figs. 3 and 4 show plots of sequence numbers of segments sent and SACKs received (both mod 100) by the TCP and SCTP sources, respectively, with respect to time. A count of the number of segments in the plots show that in the time frame of 50 to 150 sec, TCP transmits about 1500 segments, while SCTP transmits about 2300 segments. The reason for SCTP being able to send more segments than TCP will be discussed in Sec. V-B

*Link utilization* is poor for all the cases, as seen in Table II. This can be attributed to the fact that the bottleneck satellite link has a long delay, and when losses are detected at multiple hosts, few packets are sent along the bottleneck link because the sources are in the congestion avoidance mode. So the *combined throughput* is low most of the time, as can be seen in Fig. 5 that shows the total

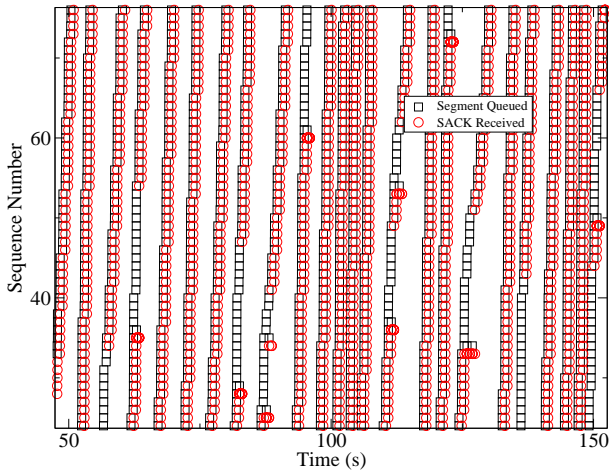


Fig. 4. Time-Sequence for SCTP.

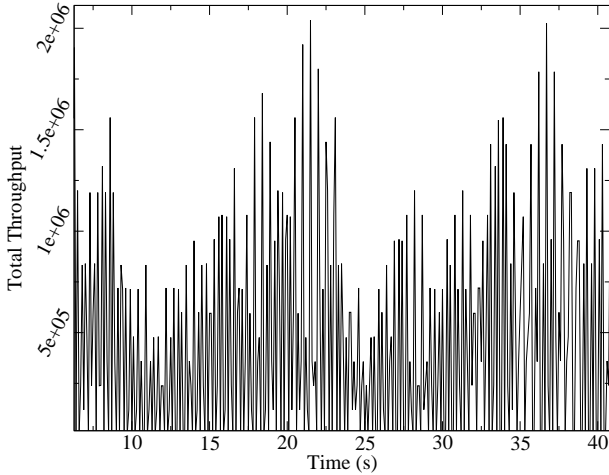


Fig. 5. Total throughput vs Time for two-source simulation.

throughput in  $L_s$  with respect to time for the two-source scenario during the first 40 seconds of simulation.

Though there is a considerable percentage increase in throughput of SCTP over TCP in all the simulations, the fairness values (see Table II) obtained show that each of the networks is nearly always fair. This is because they both underutilize link  $L_s$  due to its long propagation delay; the fraction of the total bottleneck bandwidth available that is used by both the sources is small. So even though SCTP yields a higher throughput when compared to TCP, the degree of unfairness in the network is low. The next subsection looks into how differences in the  $cwnd$  of TCP and SCTP contribute to the higher throughput of SCTP.

### B. Congestion Windows of TCP and SCTP

Figs. 6 and 7 show the congestion windows of TCP and SCTP in the time range 50 - 150 sec. The plots show that on average, the  $cwnd$  of SCTP is higher than that of TCP. A calculation of the average  $cwnd$  of TCP and SCTP over the entire simulation was found to be approximately seven and eight for TCP and SCTP, respectively. The larger

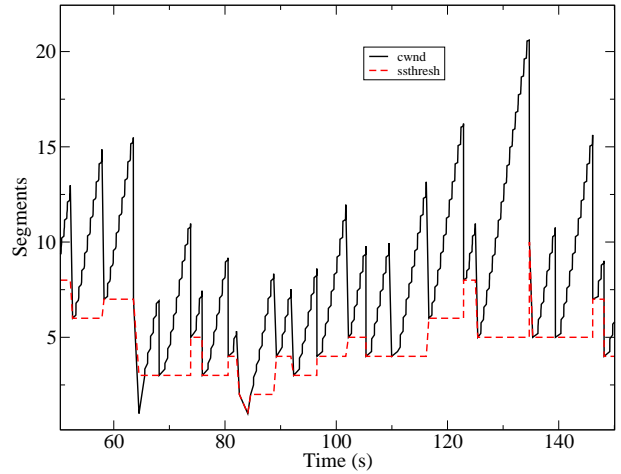


Fig. 6. Congestion Window and Ssthresh Variation for TCP.

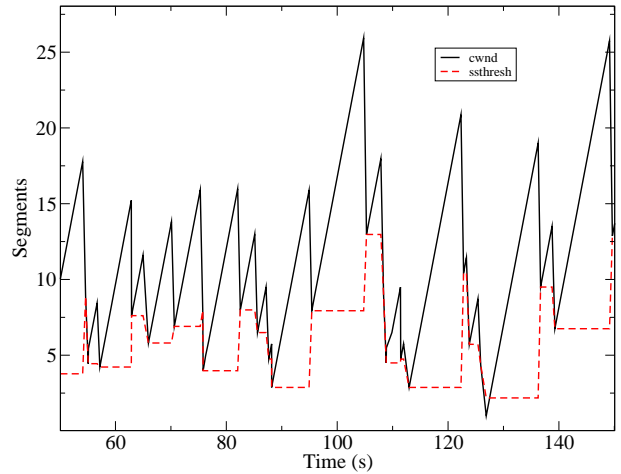


Fig. 7. Congestion Window and Ssthresh Variation for SCTP.

average congestion window of SCTP partially accounts for its improved throughput (see Table II), since a larger  $cwnd$  allows more segments to be transmitted. The remainder of this section discusses the variation of  $cwnd$  for TCP and SCTP.

Figs. 8 and 9 show variation of  $cwnd$  of TCP and SCTP sources due to loss of segment 8 by both the sources. At that point, they were both ramping up in slow start, and  $cwnd$  was eight when the segments were dropped. The loss of segment 8 was detected when  $cwnd$  was nine for both TCP and SCTP. This point is indicated by the point M in Fig. 8 and point Q in Fig. 9.

In the case of TCP, the segment loss was detected at time 2.880, and  $cwnd$  fell to four (indicated by point N in Fig. 8). The next change of  $cwnd$  was at time 3.413, when  $cwnd$  increased to 4.25, according to the congestion avoidance algorithm [15]. At times 3.897, 3.909, and 3.921, the  $cwnd$  increased to 4.485, 4.708, 4.921, respectively. The  $cwnd$  finally increased to 5.124 at time 3.927 (indicated by point P in Fig. 8)

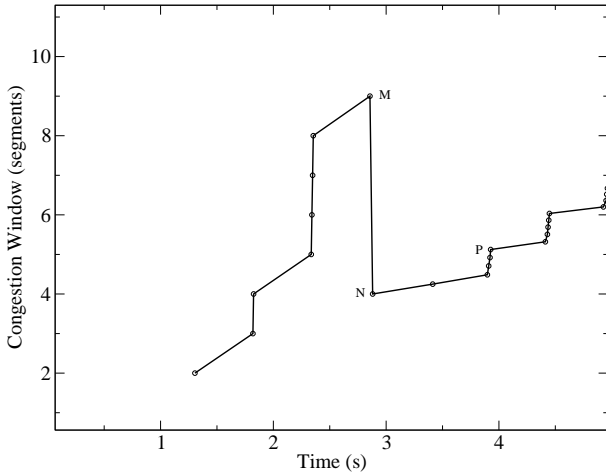


Fig. 8. Congestion Window of TCP during Segment Loss and Re-transmission.

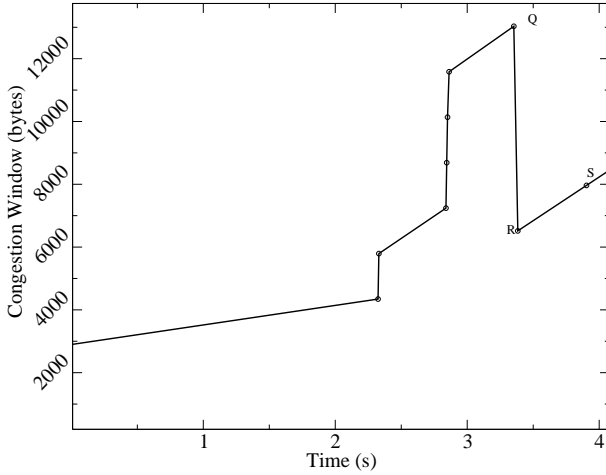


Fig. 9. Congestion Window of SCTP during Segment Loss and Re-transmission.

In the case of SCTP, *cwnd* fell from 13032 bytes<sup>1</sup> (about 9 segments) to 6516 (4.5 segments), indicated by point R in Fig. 9. At time 3.904, the *cwnd* increased to 7964 bytes [14] given by point S, then at time 4.424 to 9412 bytes and so on.

TCP's *cwnd* increased from 4 segments to 5.124 segments in 1.05 seconds. On the other hand, it took only 0.52 seconds for the *cwnd* of SCTP to increase from 6516 bytes (about 4.5 segments) to 7964 bytes (about 5.5 segments). Details of how SCTP is able to increase its *cwnd* faster while recovering from a segment loss are presented below.

### C. Analysis of Congestion Control and Retransmission of TCP and SCTP

This section individually looks into how TCP and SCTP sources handle segment losses. To allow for a comparable scenario, we configured our simulation to drop the packet with the same sequence number for both TCP and SCTP.

<sup>1</sup>Note that *cwnd* of an SCTP association is measured in bytes

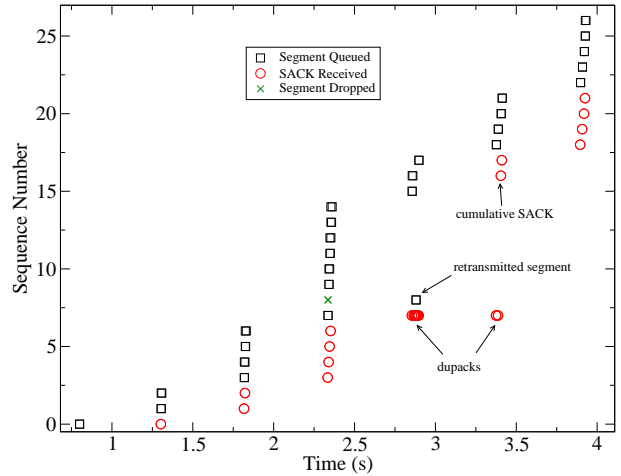


Fig. 10. Congestion Control of TCP for Loss of Segment 8.

### C.1 TCP

Fig. 10 shows the time-sequence diagram for the TCP connection when segment 8 was dropped in the bottleneck link. The segment was dropped at time 2.336 when the *cwnd* was eight. Since the connection was in the slow-start phase, the arrival of the SACK for segment 7 at time 2.856 resulting in clocking out of segments 15 and 16. *Dupacks* started arriving at time 2.869. On arrival of the first two *dupacks*, no new segments were clocked out. When the third *dupack* arrived at the source at time 2.880, segment 8 was Fast Retransmitted, and *cwnd* dropped from nine segments to four segments. The *flightsize* at this point was six. So when the next two *dupacks* arrived, the TCP source did not send out new segments. On arrival of the sixth *dupack* at time 2.898, *flightsize* came down to three, so the *cwnd* allowed the TCP source to clock out a new segment (number 17). From the time segment 8 was retransmitted, the *cwnd* remained constant at four.

Only after the arrival of cumulative SACK (acknowledging the retransmitted segment as well as all other intermediate segments) acknowledging a new segment at time 3.413, the *cwnd* was increased by a fraction to 4.25. As illustrated earlier, the *cwnd* reached 5.124 after arrival of another four SACKs at the source.

### C.2 SCTP

Fig. 11 shows the time-sequence diagram of SCTP, when segment 8 in the SCTP association was dropped at time 2.839. SCTP follows a four-way handshake, while TCP follows a three-way handshake. This explains why the SCTP segment was dropped at a later time than the TCP segment although the same segments in both the TCP connection and SCTP association were dropped.

The value of *cwnd* was 11584 bytes (eight segments) when segment 8 was dropped. Just as in TCP, when the SACK for segment 7 arrived at time 3.353, segments 15 and 16 were clocked out because the association was in slow-start. *Dupacks* started to arrive at time 3.359, and with each of the first three *dupacks*, new segments (17, 18 and



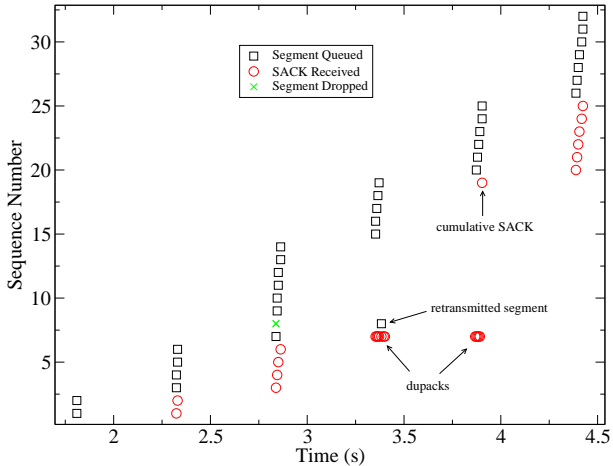


Fig. 11. Congestion Control of SCTP for Loss of Segment 8.

19) were clocked out, since the arrival of the *dupacks* indicated that three segments had left the network. When the fourth *dupack* arrived at time 3.383, the lost segment was Fast Retransmitted. Because of Fast Retransmission, the *cwnd* was also halved from 13032 bytes (nine segments) to 6516 bytes (4.5 segments). At that point, the *flight-size* was 11584 bytes (eight segments). As more *dupacks* arrived, the *flight-size* gradually decreased by 1448 bytes until at time 3.874, the *flight-size* and *cwnd* allowed a new segment (number 20) to be clocked out. Three more *dupacks* arrived after this, and segments 21, 22 and 23 were sent on each arrival.

The cumulative SACK acknowledging the retransmitted segment as well as all the intermediate segments arrived at time 3.904, at which point the  $cwnd = ssthresh$ . SCTP therefore, followed slow start to increase its *cwnd* to 7964 bytes (5.5 segments), and also clocked out two new segments. Then, when the SACK for segment 20 arrived at the source at time 4.388, SCTP went into congestion-avoidance and increased its *pba* by the number of bytes acknowledged.

#### D. Result Summary

SCTP has several advantages over TCP while recovering from a segment loss.

- Firstly, when the *dupacks* start arriving, SCTP is able to clock out new segments, whereas TCP cannot. This contributes significantly towards SCTP’s improved throughput [13].
- Secondly, since SCTP measures *flight-size* in bytes, and is decreased by the number of bytes acknowledged (instead of the number of segments), it can fall below *cwnd* faster, so as to allow quicker transmission of new segments. Since TCP keeps track of *flight-size* and *cwnd* in number of segments, it takes more SACKs to arrive before a new segment can be transmitted.
- Thirdly, SCTP is able to increase its *cwnd* from its current value (instead of from  $cwnd = 1$  MTU) using slow-start on arrival of the cumulative SACK after Fast Retransmit<sup>2</sup>.

<sup>2</sup>Note that at this point  $cwnd = ssthresh$

No of Sources	% Incr in Throughput	Link Utilzn (%)	Fairness
2	17.3	96	0.994
4	15.8	90.5	0.995
6	30.6	99.7	0.983
8	14.9	96.1	0.995

TABLE III  
PERCENTAGE INCREASE IN THROUGHPUT, LINK UTILIZATION AND FAIRNESS FOR SECOND SET OF EXPERIMENTS.

As the case above has shown, the *cwnd* had dropped to 6516 bytes during Fast Retransmit, and it had remained at this value until the cumulative SACK arrived at time 3.904, when the *cwnd* jumped to 7964 bytes. This gives the SCTP source a significant advantage, since the *cwnd* is immediately increased by the number of data bytes acknowledged by the cumulative SACK. TCP, on the other hand, does not change its *cwnd* with the arrival of the cumulative SACK. Instead, it increases by only a fraction of the current *cwnd* when the next SACK arrives.

#### VI. HIGH BOTTLENECK LINK UTILIZATION

To observe if there is any difference in fairness in the case when the bottleneck (satellite) link is utilized almost completely, we have conducted a separate set of experiments in which the bandwidth of  $L_s$  is set to a very low value. The topology of Fig. 2 was used. Four simulations were run as described in Sec. V. The only difference from the previous set of experiments was that the bandwidth value of  $L_s$  was set at 0.2 Mbps for all the simulations.

Table III shows the percentage increase in throughput, link utilization and fairness for the four new experiments carried out. The results show that the shared link utilization is improved considerably. The SCTP sources once again achieve a positive percentage increase in throughput, ranging from 14.9% to 30.6%. It can also be observed that the fairness index is high, indicating *fairness when TCP and SCTP sources share the same satellite network*.

#### VII. CONCLUSION

Through experimentations presented in this paper, we have established that there is almost complete fairness in a long-delay network in which TCP and SCTP sources compete for common resources. Four simulations were run, and in each simulation, the total number of sources were increases by two, while maintaining the same number of TCP and SCTP sources. Though the degree of unfairness is low in each case, it persists since SCTP always achieves an improvement in throughput over TCP, with a maximum of 30.6% when there are three TCP and three SCTP sources, and the bottleneck is almost fully utilized. Plots of *cwnd* reveal that SCTP has a larger average *cwnd* than TCP. We have shown that this accounts for the increase in throughput obtained for SCTP. We have also observed the following through detailed analysis of the experimental

results:

1. The retransmission mechanism of SCTP contributes to the increased throughput.
2. The byte-oriented computation of *cwnd* and *flightsize* helps SCTP recover from a low *cwnd* faster after Fast Retransmit.
3. When the  $cwnd = ssthresh$  in SCTP, it initiates slow-start from the current *cwnd*. A faster increase in *cwnd* is thereby achieved.

The observations mentioned above are concluded to be responsible for the better performance of SCTP in the shared, long-delay network. Though SCTP performs better, it does not bring about significant unfairness for TCP sources sharing the same network.

#### REFERENCES

- [1] R. Stewart, Q. Xie, K. Morneault, and C. Sharp et. al., "Stream control transmission protocol." RFC 2960, Oct 2000.
- [2] M. Atiquzzaman and W. Ivancic, "Evaluation of SCTP multi-streaming over satellite links," *Submitted for publication*, February 2002.
- [3] R. Brennan and T. Curran, "SCTP congestion control: Initial simulation studies," *International Teletraffic Congress (ITC 17)*, Brazil, 2001.
- [4] [www.networksorcery.com/enp/protocol/sctp.htm](http://www.networksorcery.com/enp/protocol/sctp.htm), 2002.
- [5] G. Hasegawa and M. Murata, "Survey on fairness issues in TCP congestion control mechanisms," *IEICE Transactions on Communications*, vol. E84-B, no. 6, pp. 1461–1472, June 2001.
- [6] N. Ghani and S. Dixit, "TCP/IP enhancements for satellite networks," *IEEE Communications Magazine*, vol. 37, no. 7, pp. 64–72, July 1999.
- [7] C. Partridge and T. Shepard, "TCP/IP performance over satellite links," *IEEE Network*, vol. 11, no. 5, pp. 44–49, Sep-Oct 1997.
- [8] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saarinen, "TCP performance issues over wireless links," *IEEE Communications Magazine*, April 2001.
- [9] K. Pentikousis, "TCP in wired-cum-wireless environments," *IEEE Communication Surveys*, Fourth Quarter 2000.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] R. Alamgir, M. Atiquzzaman, and W. Ivancic, "Impact of retransmission mechanisms on the performance of SCTP and TCP in a satellite network," *Submitted for publication*, March 2002.
- [14] R. R. Stewart and Q. Xie, *Stream Control Transmission Protocol*, Boston, MA: Pearson Education, Inc., 2002.
- [15] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control." RFC 2581, April 1999.
- [16] "ns-2 network simulator." [www.isi.edu/nsnam/ns/](http://www.isi.edu/nsnam/ns/).
- [17] 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.
- [18] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, Sep-Oct 1989.