SCTP vs. TCP Delay and Packet Loss

Muhammad Khalil Afzal[1], Aman-Ullah-Khan[1], Antonio Pescape[2]
Yousaf Bin Zikria[3], Salvatore Loreto[4]
Computer Science Daprtment CIIT,Wah Cantt[1], Universit`a di Napoli "Federico II" Naples (Italy)[2]
Electrical Engineering Department CIIT,Islamabad[3], Salvatore Loreto Ericsson Research Jorvas, Finland[4]
khalil_78_pk@yahoo.com[1], auk_pk@yahoo.com[1], pescape@unina.it[2],
yusi_2@hotmail.com[3], salvatore.loreto@ieee.org[4]

Abstract- **Stream control Transmission protocol (SCTP) is a new transport layer protocol, proposed by IETF in RFC 4960. In this paper we have done a simulation-based comparison of important Quality of Service parameter delay and the impact of packet loss on the throughput using SCTP and TCP Sack as a transport protocol in network simulator (ns-2) in wired network. Our result shows that SCTP and TCP Sack show similar behavior in term of delay in single flow and competing traffic while SCTP achieves higher throughput than TCP when different loss probability is induced in the network.**

Keywords-SCTP; TCP SACK; Cwnd; Rwnd; Ack;

I. INTRODUCTION

In this paper we have investigated a comparison of delays offered by SCTP and TCP SACK, and the impact of packet loss on throughput when carrying FTP traffic with varied loss probability. TCP has been the exclusive dominant reliable transport protocol on the internet even to date. Although a lot of research has been carried out regarding transport protocol redesign and reengineering, but it has not swayed the dominance of TCP to a great extent. Still the transport protocols like SCTP and others are still stirring in the research community and nowadays are going to become a regular part of the TCP/IP protocol suite in popular operating systems.

SCTP was presented by the IETF SIGTRAN Working Group to transport SS7 signaling traffic on the Internet. At present the SCTP standardization work is continuing in the IETF TSV (Transport Area Working group). Even if the original aim was a protocol for transport signaling the research has taken a giant leap in terms of proposing SCTP has a perspective transport for carrying FTP as well as web traffic.

Amer et al. have carried out major research activities in this regard. They have done investigation of SCTP for carrying web and FTP traffic. In [1] they propose it as a state-of-the-art transport protocol for web, whereas in [2] they have talked about concurrent multi-path transfer using multi-homing feature of SCTP, retransmission policies for this scenario [3], and effect of receiver buffer [4] .The stress in this work has been the calculation of one-way delay and the impact of packet loss on the performance of both TCP Sack and SCTP.

The rest of the paper is organized as follow. In section II we have dilated about some of the research efforts carried out in the perspective related to our work. Section III makes conspicuous some of the key features of SCTP. In section IV we summarize the TCP SACK. In Section V we narrated our simulations work. And finally in Section VI we conclude our work.

II. RELATED WORK

In case of the wired scenario, authors in [5] presented results related to the experimentation of the SCTP over high-speed wide area networks. In [6] the authors compare the performance of SCTP and TCP with respect to Web traffic. In [7], using ns-2, the authors study the multi-streaming and the multi-homing SCTP features. They prove that these aspects have advantages over TCP in the given scenarios. In particular, they define the optimal number of streams in multi-streaming and explain how it affects network performance. In the case of wireless networks, in [8] the authors developed an analytical model that takes into account the congestion window, the round trip time, the slow start and congestion avoidance processes to predict the SCTP performance. By comparing numerical results from the analytical model with simulation results, they demonstrate that the proposed model is able to accurately predict SCTP throughput.

And finally in [9] the author measure SCTP throughput and Jitter over heterogeneous networks in a real environment and present a packet level analysis of SCTP in wired, wireless and heterogeneous scenarios. Our work is an extension of [9] to measure the packet delay and impact of packet loss on throughput using TCP Sack and SCTP as a transport protocol.

III. OVERVIEW OF SCTP

The Stream Control Transmission Protocol (SCTP) [10] is a transport protocol designed to transport signaling traffic. SCTP inherits its congestion and flow control mechanisms from TCP, and includes a number of refurbishments intended to make it a more efficacious signaling transport than TCP. In general the main differing facets that SCTP has are: multistreaming (i.e. multiple logical paths per association with optional unordered delivery avoiding one of the primary TCP problems Head-of-Line Blocking), multihoming (multiple network interfaces out of witch one is chosen as primary), four way connection establishment (avoiding SYN Denial-of-Service attacks) and message orientation (making the job of the application parser easier by negating the requirement of application level framing)

1-4244-1553-5/07/$25.00 ©2007 IEEE

IV. TCP SACK

SACK is an extension to TCP that uses selective ACKs in addition to the cumulative ACKs. The cumulative ACK acknowledges the reception of all the data within a connection with a sequence number less than a certain number, whereas the selective ACK acknowledges the reception of non-contiguous ranges of sequence numbers. The cumulative ACK is the main mechanism to detect packet losses in TCP. If a TCP end-point receives packets 1, 2, 3, 5, and 7, it will send ACK (1), ACK (2), ACK (3), ACK (3), and ACK (3) again. When the sender receives ACK (3) multiple times, it knows that packet 4 was lost and thus has to be retransmitted. However, the sender does not know which packets with a higher sequence number than 4 arrived successfully at the peer and which ones were lost as well. If the receiver had used the SACK TCP option it would have returned ACK (3)-SACK (5) and (7). With this information the sender is able to retransmit not only packet 4, but also packet 6. Performance measurements [11] show that TCP SACK recovers better than TCP Reno and New Reno from multiple losses in single TCP window.

V. SIMULATION

All simulation is carried out using the network simulator (ns-2.30) [12]. The ns-2.30 has a built in capability to support the SCTP.

A. TCP and SCTP Source Configurations

We desire to calculate the delay and impact of packet loss on the performance of protocols. The following configuration is used for the TCP and SCTP Sources for this purpose.

1. Since TCP use only one connection between the source and destination, SCTP is configured to use only one stream.
2. The IP payload for both protocols is 1480 byte (without header).
3. As SCTP support both the ordered and unordered delivery of data but here SCTP is configured for ordered delivery of data.
4. The Sack option is mandatory for SCTP, so TCP therefore also uses Sack.

B. Single Flow

1. Simulation Model

Figure 1 show the topology used in a single flow experiment, where N0 acts as the TCP and SCTP sources and N1 acts as the sink for both. The node R1 and R2 are the buffer-limited drop tail (FIFO) routers so that packets arrive at the routers are dropped when the buffer is full. R1-R2 is the bottleneck link. Various simulations are run by changing the link bandwidth of R1-R2. FTP traffic is generated by using the ftp traffic generator so that a continuous stream of packets (bytes) is transferred from the source to destination. The propagation delay between the sender and receiver is 45 ms.

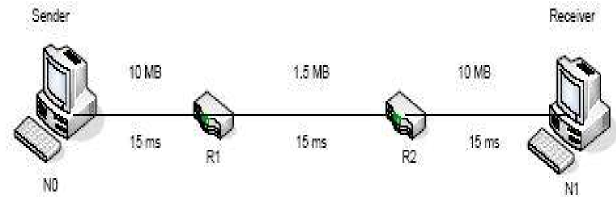

Figure 1. Single Flow Topology

2. Delays with No Packet Loss

Table 1 contains the statistical analysis of delay for both protocols. It can be analyzed from the table that SCTP and TCP show almost similar behavior in one-way delay. As far as the connection setup mechanism is concerned, SCTP takes more time since it has a broader notion of setting up an association in comparison to the connection in TCP and goes through a four-way handshake rather than three ways.

3. Delays with Packet Loss

Loss probability is set to 0.01 and 0.02 to discern the effect of packet loss on delay. In the case of 1 MB when there is no packet loss in the network the TCP sender sends 8225 Packets while when the loss probability is 0.01, TCP sends 7448 packets. As throughput decreases, the mean delay decreases. Likewise in the case of SCTP with no packet drop the sender sends 8217 packets and with 0.01-loss probability, SCTP sender sends 7640 packets. More packet leads to more queuing delay and this plays a significant role in increasing total end-to-end delay. So from the above we can say that higher throughput resulted in higher delay. Similar behavior is observed with loss probability 0.02. Table 2 and Table 3 contain the statistic with loss probability 0.01 and 0.02.

TABLE 1.
DELAY WITH NO PACKET LOSS

Bandwidth	TCP		SCTP	
	Delay (ms)	Variance (ms)	Delay (ms)	Variance (ms)
1 MB	491.92	242.37	480	230.72
2 MB	224.49	50.23	218.18	47.57
3 MB	134.64	18.10	130.62	17.03
4 MB	89.79	8.04	86.77	7.51
5 MB	62.88	3.94	60.46	3.46

TABLE 2.
DELAY WITH LOSS PROBABILITY 0.01

Bandwidth	TCP		SCTP	
	Delay (ms)	Variance (ms)	Delay (ms)	Variance (ms)
1 MB	122.75	19.71	131.56	22.14
2 MB	63.62	4.59	64.89	4.89
3 MB	54.71	3.22	54.94	3.30
4 MB	52.83	2.97	52.95	3.08
5 MB	52.83	2.97	52.03	2.97

TABLE .3.
DELAY WITH LOSS PROBABILITY 0.02

Bandwidth	TCP		SCTP	
	Delay (ms)	Variance (ms)	Delay (ms)	Variance (ms)
1 MB	89.39	9.90	100	14.12
2 MB	59.50	4.20	61.47	5.51
3 MB	55.89	3.74	57.06	4.85
4 MB	54.59	3.58	55.71	4.84
5 MB	53.81	3.50	54.79	4.56

C. Competing Traffic

This experiment is designed to simulate the effect of competing traffic generated by the node N0 and N2. This provide competition for bandwidth on the bottleneck link between the Router R1 and R2.The topology used is shown in Figure 2 where N0 acts as SCTP source and N1 acts as SCTP receiver whereas N2 act as a TCP source and N3 acts as a TCP receiver. R1 and R2 act as a router. Using the FTP traffic generator, so that a continuous stream of packets (bytes) is transferred from the source to destination. The mean delay of SCTP is 269.617 ms and that of TCP is 268.811 ms, which is approximately the same.

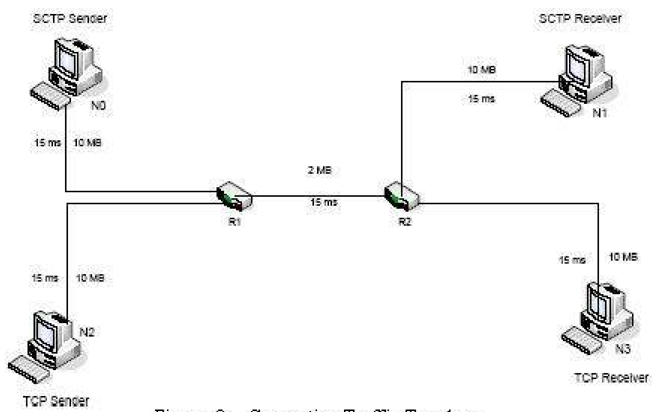

Figure 2. Competing Traffic Topology

D. Effect of Packet Loss on the Performance of TCP and SCTP

TCP and SCTP are reliable transport protocols. When TCP or SCTP sources detect a packet loss it retransmits the lost packet based on information received by the receiver in the acknowledgement. In this section we present the effect of packet loss on the performance of TCP and SCTP. Fig 2 shows the simulation model used for our experiment to analyze the packet loss behavior of protocols. The Percentage throughput is calculated by the following formula (1) as described in the [13]

$$\Gamma_k = \frac{\lambda_k}{\sum_{i=1}^{n} \lambda_i} * 100 \quad \ldots\ldots\ldots\ldots\ldots (1)$$

TABLE 4.
PERFORMANCE COMPARISON OF SCTP AND TCP

Loss Probability	Percentage Throughput of TCP	Percentage Throughput of SCTP
0	51.57	48.43
0.005	51.59	48.41
0.01	45.76	54.23
0.02	47.69	52.31
0.03	45.28	54.71
0.04	43.50	56.50

Where λ_i denote the throughput of source i. λ_i is measured by the total number of bytes sent by source i during a period of time, excluding retransmitted packets.

Table 4 shows Percentage Throughput achieved by TCP and SCTP. An inspection of the Table show that when there is no packet loss in the network TCP show better throughput than SCTP whereas when loss probability in induced in the network then SCTP show higher throughput than TCP. Figure 3 shows the number of packets sent with different loss probability where axis of abscissa displays the loss probability and the ordinate axis represents the Number of packets sent.

As the probability of packet loss is increased to 0.01 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 window of both the sources frequently goes down to the minimum. The same behavior is also observed in [13]. The SCTP average congestion window is 16.87, 14.05, 9.32, 8.19 and 6.87 with loss probability 0.005, 0.01, 0.02, 0.03 and 0.04 respectively, While in case of TCP these values stay at 22, 14.92, 11.02, 8.49 and 6.78 respectively. As SCTP congestion window is traced in bytes while TCP is in packets to show similarity bytes are converted to packets.

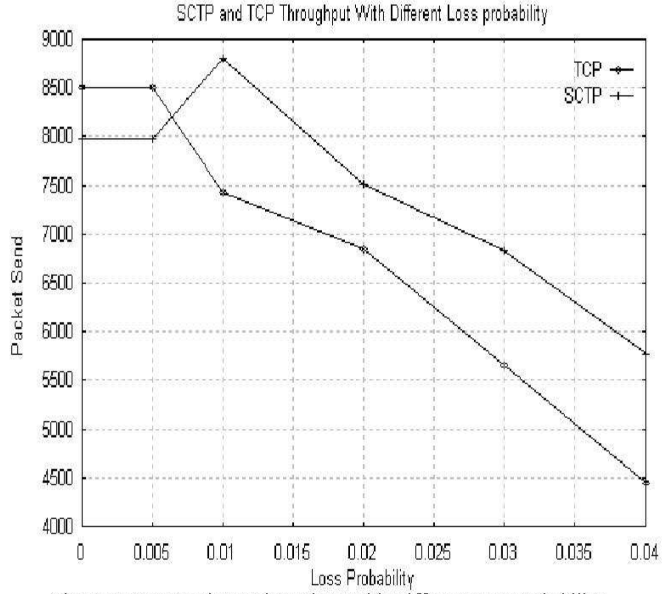

Figure 3: SCTP and TCP throughput with Different Loss Probability

For a loss probability of 0.01 the amount of packet loss is not too small and not too great, at this loss rate SCTP achieves a higher advantage over TCP, it is thus ideal for demonstrating the difference in performance between SCTP and TCP. So we choose the 0.01 loss probability to show the impact of retransmission on the performance of protocols in the next section.

E. Lost Packet retransmission Mechanism

Figure 4 and Figure 5 show the retransmission of SCTP and TCP with single packet loss. First we turn our attention to SCTP. Packet sequence number 678 that is dropped during transmission at time 7.544 sec at that time cwnd was 13.84 packets. Packet 679-691 received at the receiver and triggered the duplicate ack. The first three duplicate acks for the packet decrease the out standing packet by one and clock out new data packets 692, 693 and 694. Upon receiving the fourth duplicate ack for the packet 678, the sender fast retransmits the lost packet. The sender then halves its congestion window and sets it to 6. 92 packets. At that time there are 12 more packets unacknowledged in the network (683- 694). Packets 695 to 699 are clocked out with out the modification of the congestion window. Because an end point congestion window is not tied to its cumulative TSN ack point, as dup Sacks come in, even though, they may not advance the cumulative TSN ack point an end point can still used them to clock out new data [10]. That is the data newly acknowledged by the Sack diminishes the amount of data now in flight to less than the congestion window and so the current unchanged values of the congestion window now allows new data to be sent. On the other hand, the increase of the congestion window must be tied to the cumulative TSN ack point advancement as specified above. Otherwise the duplicate Sacks will not only clock out new data, but also will adversely clock out more new data then what has just left the network, during a time of possible congestion.

Now in the TCP we consider the packet sequence number 323 is dropped at time 6.054142 sec and cwnd was 11 packets at that time. Packets 324 to 333 arrived at receiver and trigger out dup ack. The first two duplicate acks for the packet number 323 also clock out new data that is packet number 334 and 335 that is against the [13], where they stated that TCP wasted its two dupacks by not transmitting any data. But in our case TCP also clocked out new data on the reception of dup ack. Upon receiving the third dup ack for the packet 323, sender fast retransmit the packet 323 and set its congestion window 5 packets. As depicted in [14] that ns-2 implementation of TCP performs cwnd in whole packets, whereas the ns2-implementation of SCTP reduce these variables in bytes. For example, if a cwnd of 15 is to be halved, TCP would set to 7, but SCTP would set it 7.5. So in the above case TCP would set congestion window to 5 on detection packet loss. There are 10 more packets which are unacknowledged in the network at the time of packet loss detection and cwnd is 5 packets.

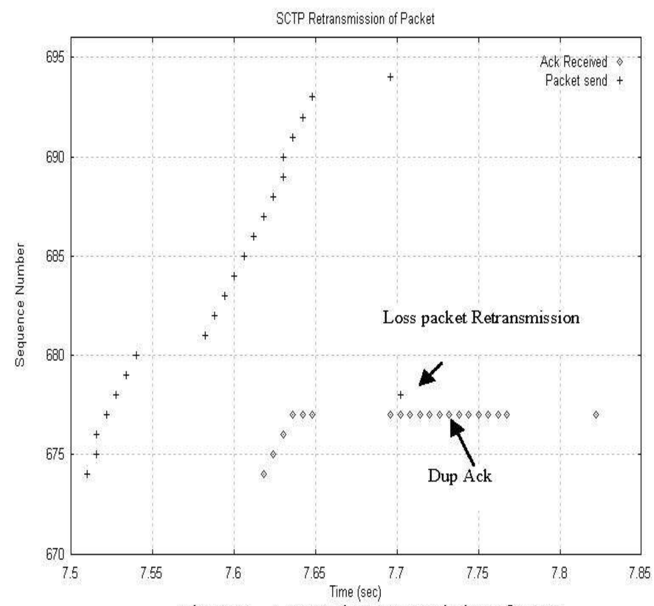

Figure 4. Lost Packet retransmission of SCTP

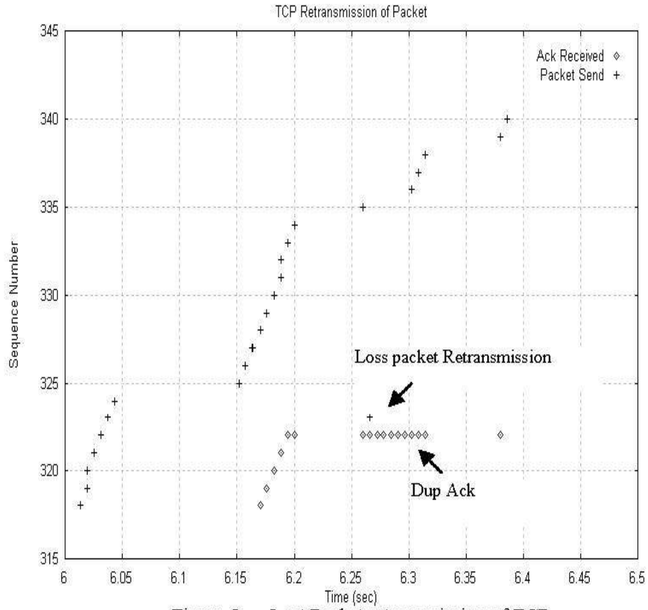

Figure 5. Lost Packet retransmission of TCP

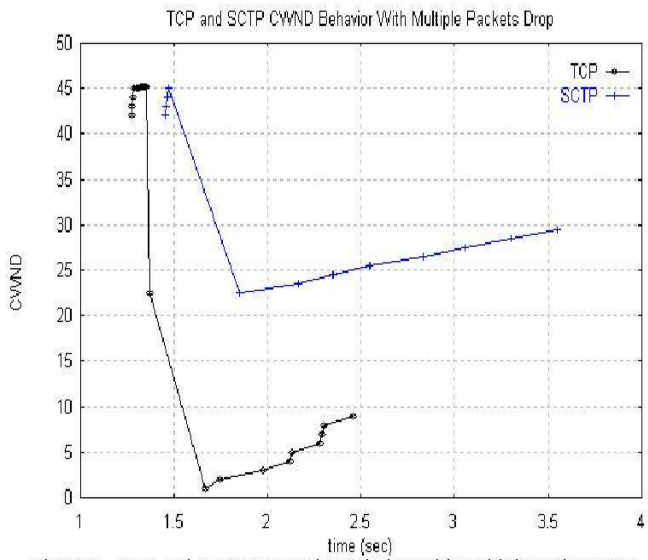

Figure 6. TCP and SCTP Congestion Window with multiple Packet Drops

At time 6.2660 sec fast retransmission occurs but sender waits from 6.2660 sec to 6.3020 sec and does not clock out new data, as there is no room in the congestion window. Upon reception of five more duplicated acks during this time then sender clocks out new data and sends packet sequence number 336. This different behavior on detecting packet loss allows SCTP to send more data without modification of its congestion window.

F. Congestion Window with Multiple Packet Drop

The Figure 6 shows the behavior of the congestion window when there are multiple packets dropped in the same window. The packet sequence number 76, 78, 80, 82, 84 and 86 dropped in TCP during the simulation time 1.2705 sec to 1.3005 sec, at time 1.2705 the cwnd was 45 packets. Upon detection of multiple packet losses the TCP sender sets its congestion window to slow start that is 1*MTU. In the case of SCTP the same sequence number packet drop during the time 1.4505 sec to 1.4805 sec at time 1.4508 congestion windows was 45 packets but in SCTP sender just halves its congestion window on detection of packet loss. This SCTP congestion control behavior allows SCTP sender to send more packets during the same simulation time.

VI. Conclusions and Future Work

This paper provides an initial study of two important QoS parameters in wired networks using a single and ordered stream. We performed a simulation based analysis of SCTP and TCP in terms of delay and impact of packet loss on the throughput. The result shows that the TCP and SCTP almost show similar behavior in delay in single flow and competing traffic. When loss probability is induced in the network throughput is badly effected and causes a decrease in delay. So from the above we can say that higher throughput also resulted in higher delay. In the case of packet loss SCTP clock out new data with out modification of its congestion window while this behavior is not found in TCP on detection of packet loss. In addition SCTP show higher throughput than TCP in the case of packet loss. In a future work we have planned to study these parameters in wireless and heterogeneous network.

REFERENCES

[1] Preethi Natarajan, Paul. D. Amer and Randall Stewart "SCTP: An Innovative Transport layer Protocol for the Web"
[2] Janardhan R. Iyengar, Keyur C. Shah, Paul D. Amer "Concurrent Multipath Transfer Using SCTP Multihoming"
[3] Armando L. Caro, Paul D. Amer, and R. Stewart "Retransmission Policies for multihomed Transport protocols"
[4] Janardhan R. Iyengar Paul D. Amer Randall Stewart ""Receive Buffer Blocking In Concurrent Multipath Transfer"
[5] D. Nagamalai and J. -K. Lee." Performance of SCTP over high-speed wide area networks, Dec. '04."
[6] R. Rajamani, S. Kumar, and N. Gupta. "SCTP versus: Comparing the Performance of Transport protocols for web traffic".
[7] S. Kang and M. Fields. "Experimental study of the SCTP compared to TCP". Technical report, Department of Electrical Engineering of Texas A& M University, '03.
[8] L. Ma, F.Yu and V. C. M. Leung."Modeling SCTP throughput in Integrated Wlan / cellular networks, May'05."
[9] Salvatore Loreto, Antonio Pescape and Giorgio Ventre "Measuring SCTP Through put and Jitter over Heterogeneous Networks"
[10] R. Stewart et al., "Stream Control Transmission Protocol," RFC 4960, IETF, Sep. 2007.
[11] K. Fall and S. Floyd, "Simulation Based Comparisons of Tahoe, Reno, and SACK TCP"
[12] http://www.isi.edu/nsnam/ns/index.html
[13] Armando L. Caro Jr., Keyur Shah, Janardhan R. Iyengar, Paul D. Amer "SCTP and TCP Variants: Congestion Control Under Multiple Losses"
[14] Rumana Alamgir, Mohammed Atiquzzaman, William Ivanic "Effect of Congestion Control on the Performance of SCTP and TCP in a Wireless Environment" Wseas Transactions on Communications Issue 6, Volume 4, June 2005.
[15] M. Allman, V. Paxon, and W. Stevens, "TCP congestion control." RFC 581, April 1999.