SCTP CONGESTION CONTROL PERFORMANCE IN WIRELESS MULTI-HOP NETWORKS*

Guanhua Ye
Tarek Saadawi
Myung Lee

Dept. of Electrical Engineering, City University of New York, City College
CUNY Graduate Center
New York, NY

ABSTRACT

Stream Control Transmission Protocol (SCTP) is originally designed for signaling transport over IP-based networks in the Signaling Transport (SIGTRAN) group of IETF and it has the potential to be an alternative transport protocol that may be better able to satisfy the requirements of future battlefield network than traditional protocols, TCP and UDP. SCTP performance over IEEE 802.11 is being evaluated in this paper. The impact of various IEEE 802.11 parameters on SCTP congestion control parameters is simulated and their effect on throughput as well as its performance is studied. Further more, a new algorithm is proposed to overcome the "small window syndrome" resulting from the MAC layer when the SCTP receiver side window is small. Simulation results demonstrate that SCTP performance is improved using this new algorithm.

1 INTRODUCTION

Stream Control Transmission Protocol (SCTP) [1] is a message based new transport protocol of Internet standards track. It has the potential to be better able to meet the requirements of future battlefield network than traditional protocols, TCP and UDP. Taking its multi-homing feature as an example, which allows multiple source and/or destination addresses in one SCTP connection ("association" is used in SCTP terminology), when one interface/address fails, the traffic can be automatically transferred to another interface without interrupting the on-going association. Typical usages are reliable server pooling and seamless mobility support [2] by using the extension specified in [3]. Because of space limitation, we refer readers to [4, 5] for more information about the multi-streaming feature of SCTP. Besides its diverse applications, we believe it is also important to examine its performance under these applications. Our earlier work [6, 7] has shown the limitations of TCP performance over IEEE 802.11 wireless LAN protocol. In this paper, we examine the interaction of SCTP and the IEEE 802.11 MAC layer. The impact of various IEEE 802.11 parameters on SCTP congestion control parameters is simulated and their effect on throughput as well as its performance is studied.

The next section describes basic concepts of SCTP with emphasis on its congestion control part. A brief introduction of those issues related to IEEE 802.11 MAC layer is given in section 3, in order to facilitate the understanding of the following sections. In section 4, we evaluate and analyze the performance of SCTP over IEEE 802.11 based wireless multi-hop networks. The proposed algorithm for "small window syndrome" and simulation results are presented in section 5. Finally, we conclude the paper.

2 SCTP CONGESTION CONTROL

SCTP bases its congestion control on TCP congestion control principles [8] and uses the SACK extension of TCP [9]. It also includes slow start, congestion avoidance, and fast retransmit. We refer readers to [1] for the details of SCTP congestion control algorithms. The major differences of SCTP congestion control algorithm and TCP congest control algorithm are:

(1) The initial congestion window (*cwnd*) is suggested to be 2*MTU in SCTP, which is usually one MTU in TCP.
(2) In SCTP, the increase of the *cwnd* is controlled by the number of acknowledged bytes; while in TCP, it is controlled by the number of new acknowledgement received.
(3) In SCTP, it is required to be in slow start phase when the slow start threshold (*ssthresh*) is equal to the *cwnd.*

* Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

It is optional in TCP to be either in the slow start phase or in the congestion avoidance phase when the *ssthresh* is equal to the *cwnd*.

(4) In SCTP, Fast retransmit is triggered by the fourth missing report of a chunk. This implies that at least 5*MTU of the receiver side window is required to trigger fast retransmission; while in TCP, the minimum receiver side window for fast retransmission is 4*MTU (three duplicate ACK triggers fast retransmission).

(5) SCTP has no explicit fast recovery algorithm that is used in TCP. In SCTP, the parameter *Max.Burst* is used after the fast retransmit to avoid flooding the network. *Max.Burst* limits the number of SCTP packets that may be sent after processing the SACK, which acknowledges the data chunk that has been fast retransmitted.

3 RELATED ISSUES IN IEEE 802.11

The fundamental access method defined in the IEEE 802.11 is a distributed coordination function (DCF) known as the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The CSMA mechanism works as follows: in order to transmit, a station shall sense the medium first. If the medium is busy, then the station defers its transmission to a later time. If the medium is determined to be free, then the station is allowed to transmit. The chances of collision still exist using this mechanism because multiple stations may determine the medium to be free and transmit simultaneously. In order to minimize such collisions, the virtual carrier sense mechanism is used to distribute channel reservation information by announcing the impending use of the medium. For data frames with sizes greater than the RTS threshold, control frames RTS/CTS (request to send/clear to send) exchange is required before data transmission. All the other stations hearing RTS or CTS shall update their network allocation vectors (NAV), which reflect the current state of the medium usage. Physical carrier sense and NAV are used together to determine whether the medium is idle or not. For data frames whose size is less than or equal to the RTS threshold, no RTS/CTS exchange is required before data transmission.

4 PERFORMANCE EVALUATION AND ANALYSIS

We implement SCTP (sections 6 and 7 of RFC2960 with the modification in [10]) into the commercially available simulation software OPNET. Dynamic Source Routing (DSR) protocol is used at the routing layer. The MAC layer uses the IEEE 802.11 based wireless radio with a bandwidth of 1 Mbps. A string topology (figure 1) is used in the simulation studies. The wireless LAN range is set to be 300m. The distance between any two neighboring nodes is equal to 250m, which allows a node to communicate directly solely to its neighboring nodes. Nodes are static (No mobility), because we want to focus on the interaction between SCTP and the IEEE 802.11 MAC layer. In all the simulation runs, SCTP data chunk size is set to be 512 bytes, and the path MTU is 544 bytes. Delayed SACK is used. The initial *cwnd* is 2*MTU. There is only one SCTP association in network, no background traffic. Node 0 is the source. Nodes 1 through 3 are selected as the destination respectively. Meanwhile, the SCTP receiver side window size and the RTS threshold of IEEE 802.11 are adjusted to see their effect upon the SCTP throughput. Large file transfer from the source to the destination (i.e. the source node always has data to send out) is assumed.

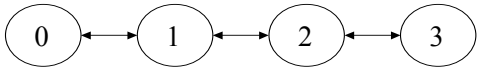


Figure 1. String Topology

Figure 2 shows the results for the simulation run between node 0 and node 1. As is shown in figure 2 (a), when the window size is 2, no matter what the value of the RTS threshold is, SCTP association achieves the highest throughput (about 660Kbps) it can achieve. Although smaller value of the RTS threshold should lead to lower throughput (more control traffic), the difference is negligible in this case because of the short simulation duration. Since the receiver side window is 2*MTU and delayed SACK is used, SCTP actually acts like the stop-and-wait protocol. It starts sending two data chunks, then waits for the acknowledgment; after it receives the acknowledgment, it sends another two data chunks. Two nodes use the wireless channel in turn. When the transmission time dominates the end-to-end delay of the network, stop-and-wait protocol should achieve very good link usage efficiency. This condition is met in our simulation setting. Thus SCTP should achieve maximum throughput in this case. Our simulation results agree with the theory analysis.

In figure 2 (b), we observe that SCTP throughput goes down to about zero several times during the simulation time when the RTS threshold is 256 bytes. And we also observe that each throughput degradation period is associated with a timeout at the SCTP source. By debugging the simulation carefully, we find that RTS frames collide with SACK chunks at the MAC layer, and the collision causes drop of SCTP data chunk. When the RTS threshold is 256 bytes, there is no RTS/CTS exchange

before SACK transmission (SACK chunks are usually less than this threshold). Thus SACK chunks may collide with RTS frames sent by the sender for data chunks when both nodes sense the medium to be free and send simultaneously. When the window size is increased to 4 or more, forwarding SCTP data chunks and reversing SACK chunks will compete for the wireless channel. Whenever the sender has data to send, it sends RTS first to reserve the channel. If this RTS collides with SACK sent by the receiver, both endpoint will back off and retransmit later. The retry limit for SACK chunks is 7 (short retry limit), and it is 4 (long retry limit) for data chunks. Thus SACK chunks get more chances to win the competition. In our simulation setting, we find SCTP data chunks almost always fail the competition. Then link failure is reported to the DSR layer whenever a SCTP data chunk is discarded when the retry counter reaches the limit. The DSR then initiates route discovery procedure again to obtain a route to the destination node. Even the route discovery succeeds before SCTP timer runs out; the source cannot retransmit the lost data chunk in advance when the window size is small (no fast retransmit). Thus SCTP times out, and starts from slow start again. This is the main reason for the throughput degradation.

When the RTS threshold is 20 bytes, the virtual carrier sense mechanism greatly reduces collision between the source and destination nodes, thus SCTP association achieves smooth throughput as shown in figure 2 (b) and figure 2 (c) for RTS threshold of 20 bytes.

In figure 2 (c), we observe that the throughput of SCTP association (for the RTS threshold of 256 bytes) is zero for about 30 seconds. This is because of multiple packets loss in one window because of collision with reverse SACK chunks, and thus SCTP association times out several times in order to retransmit lost packets. This dramatic throughput degradation is very rare when there is only one hop between source and destination nodes, and this does not always happen when we change the simulation seed. We purposely list it here to emphasize the importance of the RTS threshold to the performance of SCTP association when the number of hops is small.

Simulation results for the SCTP association covering two hops (from node 0 to node 2) are reported in figure 3. When the window is 2, similar as in the one hop case, there is only one-way traffic: either data chunks or SACK chunks are in transmission. The only difference is that node 0 and node 1 may compete for the channel when both of them have data to send. But since they are in the transmission range of each other, and they only compete with each other when both have SCTP data chunks to send, the chances of collision is greatly reduced because RTS/CTS exchange is used before data chunk transmission. So, SCTP association achieves rather smooth throughput when window size is 2*MTU. But with the number of nodes increases, more time is spent for control traffic in MAC layer with smaller RTS threshold. This is the reason that SCTP throughput for RTS threshold of 20 bytes is a bit lower than that for RTS threshold of 256 bytes when window size is 2*MTU.

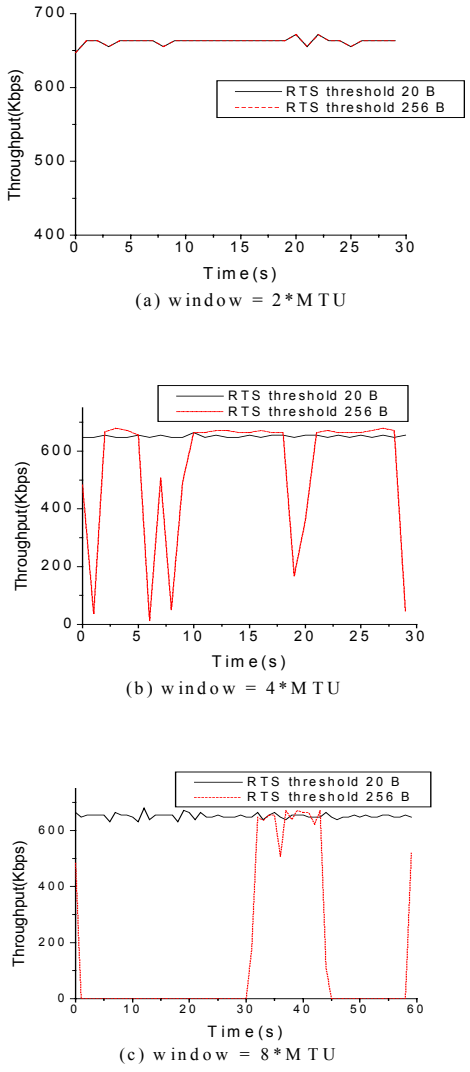


Figure 2. Throughput of one hop SCTP association

When the window goes to 4*MTU, the hidden node problem arise. Node 0 and node 2 are hidden node to each other as shown in figure 1, so it is possible that node 0 and node 1 may send frames simultaneously and that collisions occur at node 1. For small RTS threshold (20 bytes), the virtual carrier sense mechanism almost eliminates this problem, because all the nodes are either in the source's transmission range or in the destination's transmission range. Thus the throughput is rather smooth for smaller

RTS threshold as shown in figure 3(b). But for larger RTS threshold (256 bytes), there is no RTS/CTS exchange before the SACK transmission, thus data chunks may be lost after failing the competition with reverse SACK chunks. Thus the explanation for the throughput degradation for figure 2(b) can also be applied for figure 3(b) with RTS threshold of 256 bytes. But when the network load increases (window goes up to 8*MTU), the hidden node problem becomes more serious, and more link failure happens at the MAC layer, thus SCTP throughput decreases from time to time as shown in figure 3(c).

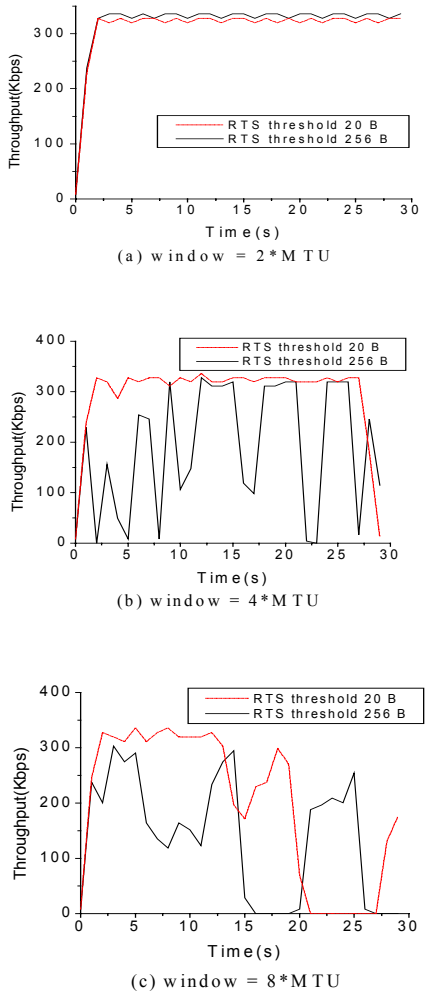


Figure 3. Throughput of two-hop SCTP association

When the number of hops goes to three, the exposed node problem also arises in addition to the hidden node problem. In figure 1, when node 2 is sending a frame to node 3, if node 0 has data to send at this time, it will determine that the medium is free, and send out the RTS frame. In this situation, node 1 cannot reply with the CTS frame (node 0 will enter into link failure if it reaches its retry limit), because it can sense the transmission of node 2. This is the exposed node problem. In figure 4 (a), the SCTP association performance degradation is mostly caused by the exposed node problem. When the window size becomes larger, the hidden node problem and exposed node problem are more serious, leading to poor throughput. And moreover, different RTS threshold values have little effect on the overall throughput, which is shown in figure 4 (b) and 4(c).

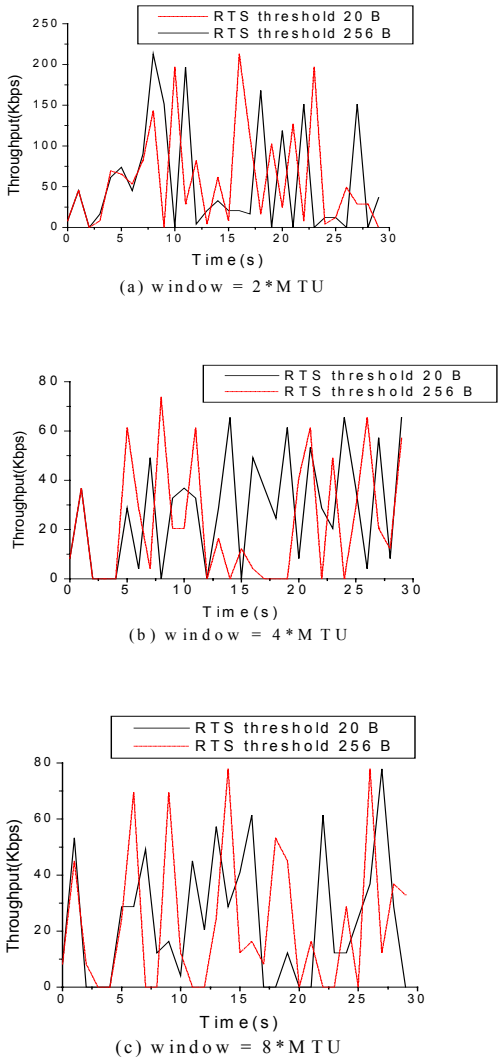


Figure 4. Throughput of three-hop SCTP association

5 SMALL WINDOW SYNDROME

We have mentioned in section 4, that SCTP association cannot retransmit lost packets before timeout when the window size is small. A simple example is given in figure 5 to illustrate this. Suppose the receiver side window size is 4*MTU (the *cwnd* is 4*MTU too), and the data chunk size is MTU. At certain time, the sender transmits data chunks with Transmission Sequence Number (*TSN*) of 21, 22, 23 and 24. If *TSN* 21 is lost for some reason, then *TSN*

22, 23 and 24 each will trigger a SACK at the receiver. After the sender receives the third SACK, the sender's view of receiver side window size is zero. In fact, there is one data chunk space available at receiver's buffer. The sender cannot send data any more because of zero window size (sender's view). Since no fast retransmission can be triggered when the receiver side window is less than 5*MTU, thus SCTP sender can only resort to timeout to retransmit the lost packet. Timeout is very "expensive", which is visualized in figure 2(b), 3(b) and 4(b) where throughput goes down to about zero from time to time. As is shown in figure 5, there is no more traffic of this association in the network during the "idle period". The network is underused during this period. We call this the "small window syndrome". The small window problem reported here is different from the one reported in [11], which is caused by small *cwnd*. But the problem we identified here is because of small receiver side window size.

As we have demonstrated in section 4, the hidden node problem and exposed node problem can cause packet losses even with small window size in wireless multi-hop environment. It is the major reason of performance degradation for scenarios in section 4. Since this kind of non-congestion loss is not rare any more, so it is not wise to use costly retransmission timeout to recover lost packets. Based on this discussion, we propose a new algorithm to alleviate this problem to recover lost packets caused by MAC layer problems.

For generality, we assume:

1. Link level reliability (e.g. M-ACK) is provided, and *TSN*s arrive at the destination sequentially. This assumption holds when the routing protocol supports only one path between the source and the destination, and link level reliability is provided. Results in [12] show that link level protection improves the TCP throughput performance in wireless multi-hop networks. It has been widely accepted that link loss protection should be supported by the MAC layer protocol for ad hoc networks.
2. Data losses are mostly caused by the hidden node or exposed node problems (MAC layer), and the network is lightly loaded (small window). In other words, network congestion is not the reason for packet losses.

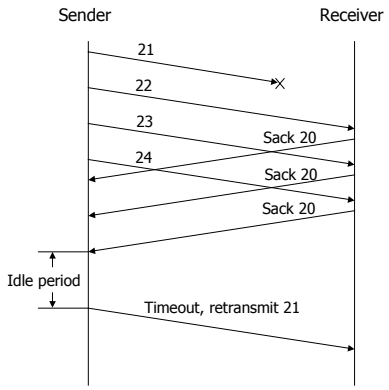
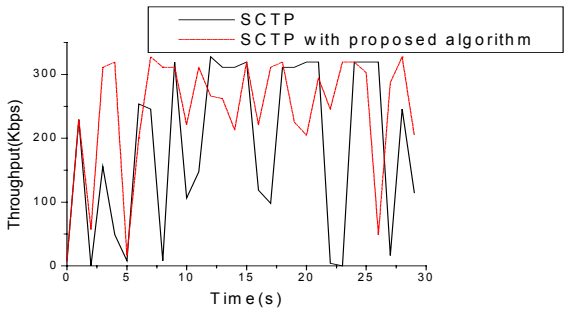


Figure 5. Example of views of receiver's window size

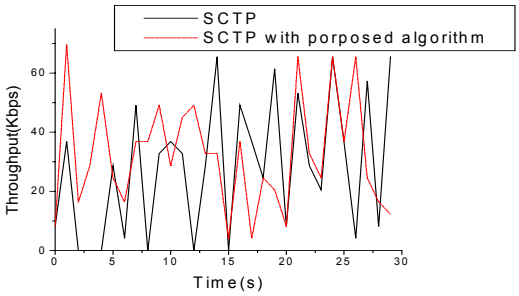
Base on these two assumptions, when the source node receives a SACK and the gap block acknowledges the outstanding data chunk with highest *TSN*, then we can conclude that the outstanding data chunk with lowest *TSN* must be lost somewhere in the network. If the packet loss is because of MAC layer problems instead of network congestion, we may want to retransmit the lost packet before timeout. Because the network may already recover before the retransmission timer runs out at the SCTP layer. In order to recover sooner from this kind of network error, we propose to retransmit the lost *TSN* (lowest *TSN*) during the "idle period" before timeout. Actually, we can determine that a *TSN* is lost whenever a higher *TSN* is gap acknowledged based on our assumption. Thus, it is possible to retransmit the lost *TSN* when a SACK with a gap block is received. But this retransmitted *TSN* may still encounter hidden node problem or exposed node problem at that time. So, we suggest retransmit a lost *TSN* during the "idle period", because MAC layer problems causing packet losses can be avoided during this period. Figure 6 shows that SCTP performs better with this algorithm than without it. The same topology as in figure 1 is used. We keep all the parameters as they are in section 4 except the window size is fixed at 4*MTU. The first run is from node 0 to node 2. The second run is from node 0 to node 3. In figure 6 (a), the total throughput in 30s is improved by 30%. In figure 6 (b), the total throughput increases by 15% in 30s. The improved throughput also proves the correctness of our analysis in section 4 for reasons of throughput degradation of SCTP associations.

This algorithm overcomes data chunk losses caused by link failure (rooted in MAC layer). It makes use of the "idle period", thus it has a minor effect upon the normal SCTP transactions. It helps to recover sooner from packet loss caused by link failure, thus improving the throughput performance. One disadvantage of this algorithm is that it cannot overcome SACK chunk losses. If the last SACK chunk is lost (the third SACK chunk in our example),

SCTP association still cannot retransmit the lost packet until timeout. Another concern is that this algorithm may make network congestion become worse if the data chunk losses are actually caused by network congestion. However, the network should not be severely congested in this case, because data chunks with larger *TSN*s arrive at the receiver (or no SACK will arrive) and window size is small, and because the lost data chunk is retransmitted only once using this algorithm. Thus, the algorithm should have only a minor effect on the network congestion level.



(a) Two hops between source and destination



(b) Three hops between source and destination

Figure 6. SCTP throughput with and without proposed algorithm

6 CONCLUSION

In this paper, we evaluate SCTP throughput performance over IEEE 802.11 wireless LAN protocol using different SCTP receiver side window sizes and the different number of hops between source and destination. We find the throughput of SCTP degrades when the number of hops increases. Increasing the window size does not help to increase the throughput. On the contrary, it amplifies the hidden node problem and the exposed node problem, worsening the throughput. The virtual carrier sense does help to improve throughput when the number of hops is small, but this is not true when the number of hops increases. Packet loss may cause SCTP timeout and degrade its performance. This is rather unfair if the packet loss is not caused by network congestion and the window size of SCTP is small. We call this phenomenon "small window syndrome". An algorithm is proposed to overcome this problem, and the simulation results show that the SCTP throughput is improved. More work need to be done to recover packet losses caused by MAC layer when a large receiver side window is used. A collision free MAC layer for wireless multi-hop networks would be desirable.

7 DISCLARIMER

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research laboratory or the U.S. Government.

8 REFERENCES

[1] R. Stewart, Q. Xie et al., "Stream Control Transmission Protocol", RFC 2960, 2000

[2] M. Riegel, M. Tuexen, "Mobile SCTP", draft-riegel-tuexen-sctp-00.txt, Feb. 2002

[3] R. Stewart, Q. Xie et al., "SCTP Dynamic Addition of IP Addresses", draft-ietf-tsvwg-addip-sctp-0.4.txt, Jan. 2002

[4] T. Connolly, P. Amer et al., "An Extension to TCP: Partial Order Service", RFC 1693(experimental), 1994

[5] P. Conrad, G. Heinz et al., "SCTP in Battlefield Networks", MILCOM'2001

[6] S. Xu, T. Saadawi. "Revealing TCP Incompatibility Problem in 802.11-based Wireless Multi-hop Networks", GlobeCom'2001, San Antonio, TX 2001

[7] S. Xu, T. Saadawi, "Revealing the Problems with 802.11 MAC Protocol in Multi-hop Wireless Ad Hoc Networks", Journal of Computer Networks. Vol. 38, No. 4, March 2002

[8] M. Allman, V. Paxson et al., "TCP Congestion Control", RFC 2581, 1999

[9] M. Mathis, J. Mahdavi et al., "TCP Selective Acknowledgement Options", RFC 2018, 1996

[10] R. Stewart, L. Ong et al., "SCTP Implementors Guide" draft-ietf-tsvwg-sctpimpguide-02.txt, Nov. 2001

[11] M. Allman, H. Balakrishnan et al., "Enhancing TCP's Loss Recovery Using Limitted Transmit" RFC 3042, January 2001

[12] M. Gerla, K. Tang et al., "TCP Performance in Wireless Multi-hop Networks", Proceeding of IEEE WMCSA'99, Feb. 1999