SANDIA REPORT

SAND2005-3132
Unlimited Release
Printed June 2005

Network Protocol Changes Can Improve DisCom WAN Performance: Evaluating TCP Modifications and SCTP in the ASC Tri-lab Environment

Tan Chang Hu and Larwrence F. Tolendino

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2005-3132
Unlimited Release
Printed June 2005

Network Protocol Changes Can Improve DisCom WAN Performance: Evaluating TCP Modifications and SCTP in the ASC Tri-lab Environment

T. C. Hu, Systems Analysis and Trouble Resolution Department
and
L. F. Tolendino, Network Design Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0806

Abstract

The Advanced Simulation and Computing (ASC) Distance Computing (DisCom) Wide Area Network (WAN) is a high performance, long distance network environment that is based on the ubiquitous TCP/IP protocol set. However, the Transmission Control Protocol (TCP) and the algorithms that govern its operation were defined almost two decades ago for a network environment vastly different from the DisCom WAN. In this paper we explore and evaluate possible modifications to TCP that purport to improve TCP performance in environments like the DisCom WAN. We also examine a much newer protocol, SCTP (Stream Control Transmission Protocol) that claims to provide reliable network transport while also implementing multi-streaming, multi-homing capabilities that are appealing in the DisCom high performance network environment.

We provide performance comparisons and recommendations for continued development that will lead to network communications protocol implementations capable of supporting the coming ASC Petaflop computing environments.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockeed Martin Company for the United States Department of Energy's National Nuclear Security administration under contract DE-AC04-94AL85000.

Table of Contents

Table of Figures

Introduction

The Advanced Simulation and Computing (ASC) program was instituted to maintain a safe and effective nuclear weapon stockpile while upholding the Nuclear Test Ban Treaty. In turn, as modeling and simulation became more important in the maintenance of the nuclear stockpile, it encouraged the development of an infrastructure to support the sharing of valuable and limited computational and visualization resources. Distance Computing (DisCom) is a critical infrastructure aspect of the ASC program that makes resource sharing possible and allows users to utilize remote ASC resources as if they were local.

ASC/DisCom

From an infrastructure viewpoint the ASC effort has been focused on the supercomputers, visualization systems, and assorted support systems primarily located at the three main weapons laboratories: Los Alamos National Laboratory (LANL), Lawrence Livermore National Laboratory (LLNL), and Sandia National Laboratories (SNL). Part of the implementation strategy for these expensive hardware systems was to create an environment for sharing the expensive resources throughout the weapons complex. The Distance Computing Wide Area Network (DisCom WAN) is a critical part of the infrastructure that makes resource sharing possible and allows users to utilize remote ASC resources as if they were local. The performance of the DisCom WAN strongly impacts the practicality of such resource sharing.

The DisCom WAN is based on high speed network links that use the ubiquitous TCP/IP protocol set and parallel data streams to transfer copious amounts of data among the participating laboratories. ASC applications that use the DisCom WAN depend on the TCP/IP protocol set to provide reliable data delivery. Parallel data paths and multiple TCP/IP data streams are combined in the WAN to provide high performance over long distances. Such high performance comes at a cost in added complexity and increased maintenance as any design changes or trouble shooting activities are more difficult.

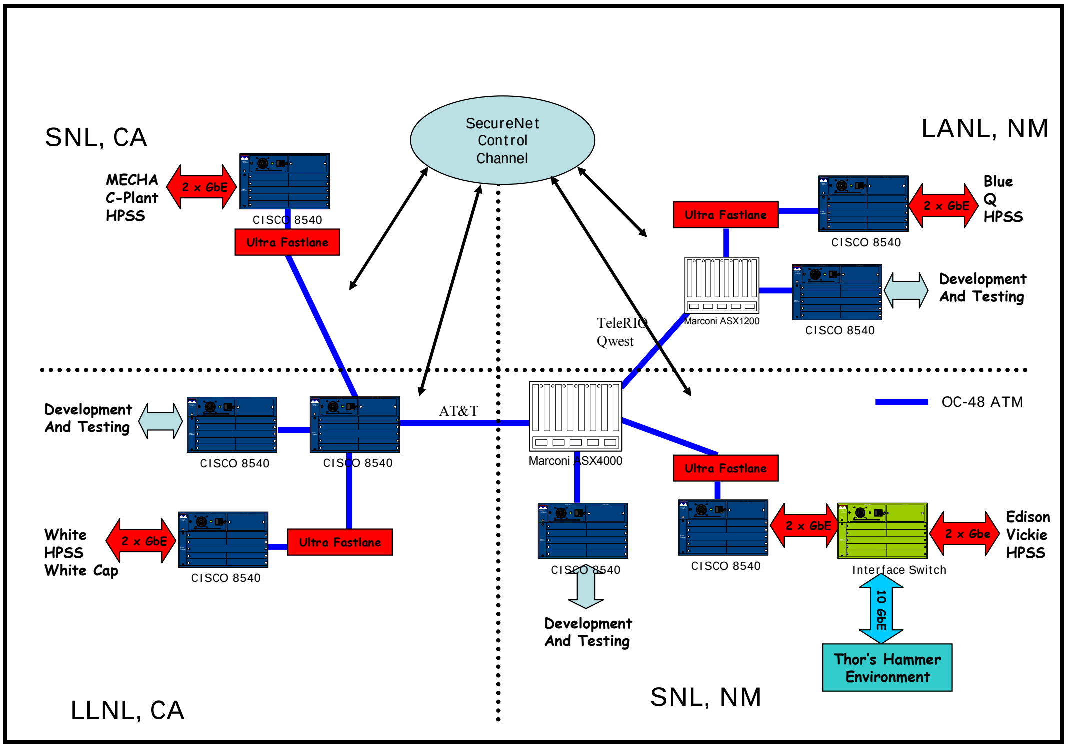

Figure 1: ASCI/DisCom Wide Area Network

Performance Challenges of the DisCom WAN

The TCP protocol was first standardized in 1980 and over time congestion control mechanisms were added to improve performance and avoid network grid lock. These mechanisms have been modified as time went on and fine tuned to improve performance for the relatively low speed, large latency, shared, Internet environment of the 1980's and 1990's. The DisCom WAN is a far different environment. The three laboratories, SNL, LANL, and LLNL, are currently connected using high speed, dedicated data links that allow computing resources to share large amounts of data over long distances. Compared to local access, these long distances mandate a large Round Trip Time (RTT) for any data exchanges. The two circumstances, high bandwidth and large RTT, create a very challenging network environment characterized by network congestion with very poor recovery performance due to the large distances involved. With this high speed, long delay network, an alternative TCP congestion control algorithm or protocol design could provide better utilization of the channels between laboratories.

Over the years researchers had proposed modifications to the TCP protocol such as High Speed TCP or Scalable TCP to address the unique characteristics of high performance networks like the DisCom WAN. However, none of these proposals has been adopted into the main stream TCP implementations. A laboratory version of the DisCom WAN was created so that Sandia could investigate these modifications, evaluate them in great detail and report their performance [1, 2]. In addition to modifications to TCP a new protocol, the Stream Control Transmission Protocol (SCTP), had been developed and become part of the Linux operating system environment. Just recently the performance of SCTP was evaluated in a DisCom WAN like environment. In the remainder of this paper we will summarize the results of all these protocol performance evaluations and compare potential new capabilities to our users' requirements. In addition, we will make recommendations for future protocol implementations and development. But first we have to characterize actual DisCom WAN usage.

Characterizing Usage of the DisCom WAN

When the ASC environment was envisioned about five years ago, it was assumed that large amounts of data would be exchanged among the four laboratory locations using very large files. Large files were considered to be many Gigabytes in size. With that assumption in mind the DisCom WAN was configured to support jumbo Ethernet frames (9,000 bytes) and a special parallel FTP (PFTP) application was developed to insure that the large bandwidth of the DisCom WAN would be used effectively. These two capabilities working together typically achieve file transfer throughputs of 100 Megabytes per second or more. However, data gathered just after the jumbo frame support was implemented showed that most file transfers were not achieving the maximum throughput possible.

File Size Distribution

The reason that some of the measured file transfers did not achieve maximum performance is related to the relatively small size of most of the files transmitted. The massively parallel supercomputers used to model nuclear weapons events typically produce an output file for each compute node. With thousands of compute nodes involved in a calculation, thousands of output files are produced. In some cases our users have elected to simply transfer the thousands of small files individually instead of concatenating them into large files. This can be seen in the following histogram which relates the number of files transferred to the file size and associated throughput achieved on the DisCom WAN. The file transfer activity shown is for data transfers between the ASC White supercomputer at Lawrence Livermore National Laboratory in California and the ASC visualization node, Edison, at Sandia National Laboratories in New Mexico.

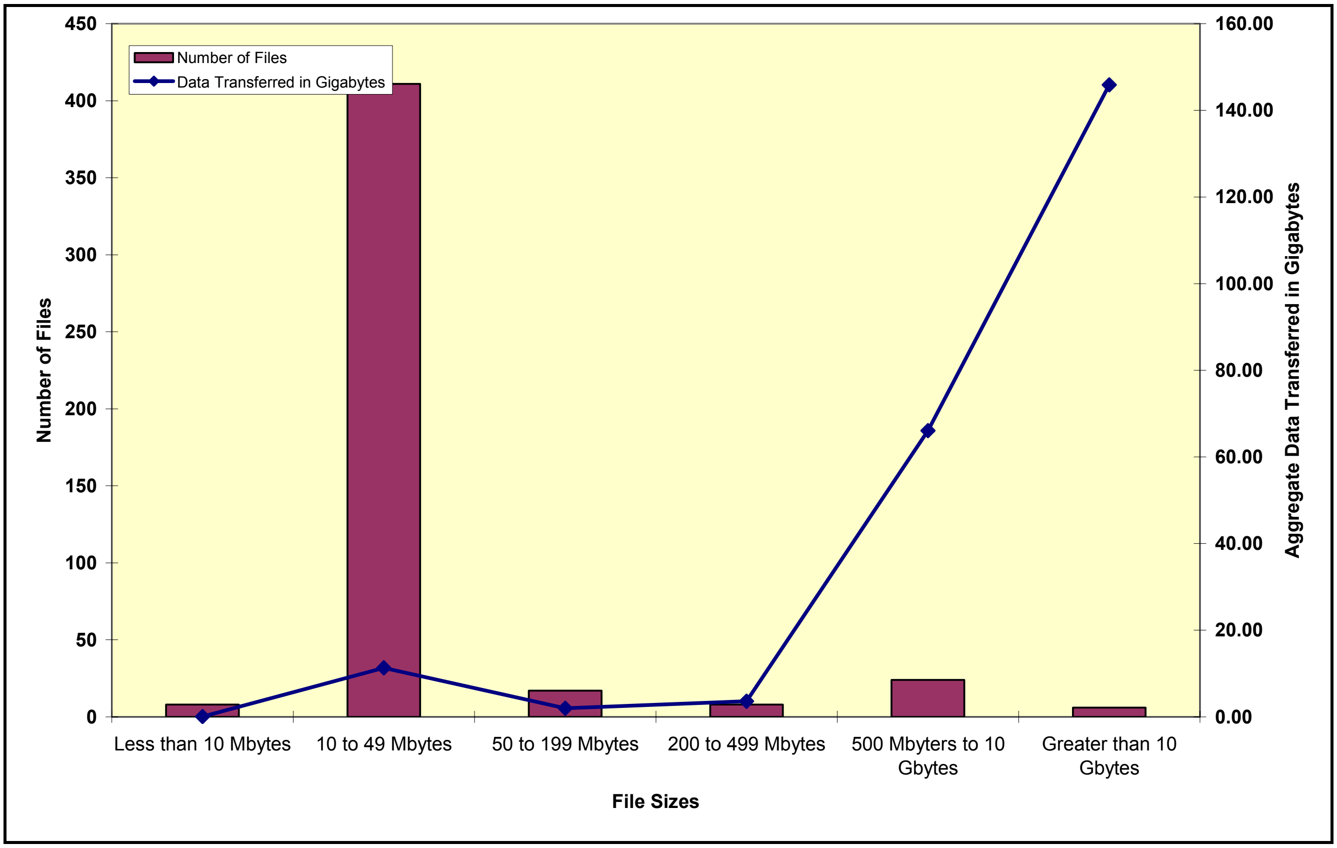

Figure 2: DisCom File Size Distribution and Performance May 2003

As one can see the vast majority of files transferred in the May 2003 time frame were around 50 megabytes, much too small to take full advantage of the capabilities of the parallel FTP application. The parallel FTP application operating over the DisCom WAN can only achieve high throughput for files of a Gigabyte or more. However, it would be wrong to conclude that implementing jumbo frames and the parallel FTP application were activities taken in vain. 80% to 90% of the data transferred during the period recorded above was transferred in large data files of approximately 1 Gigabyte or more.

The following graphic relates file size distribution to aggregate data transferred.

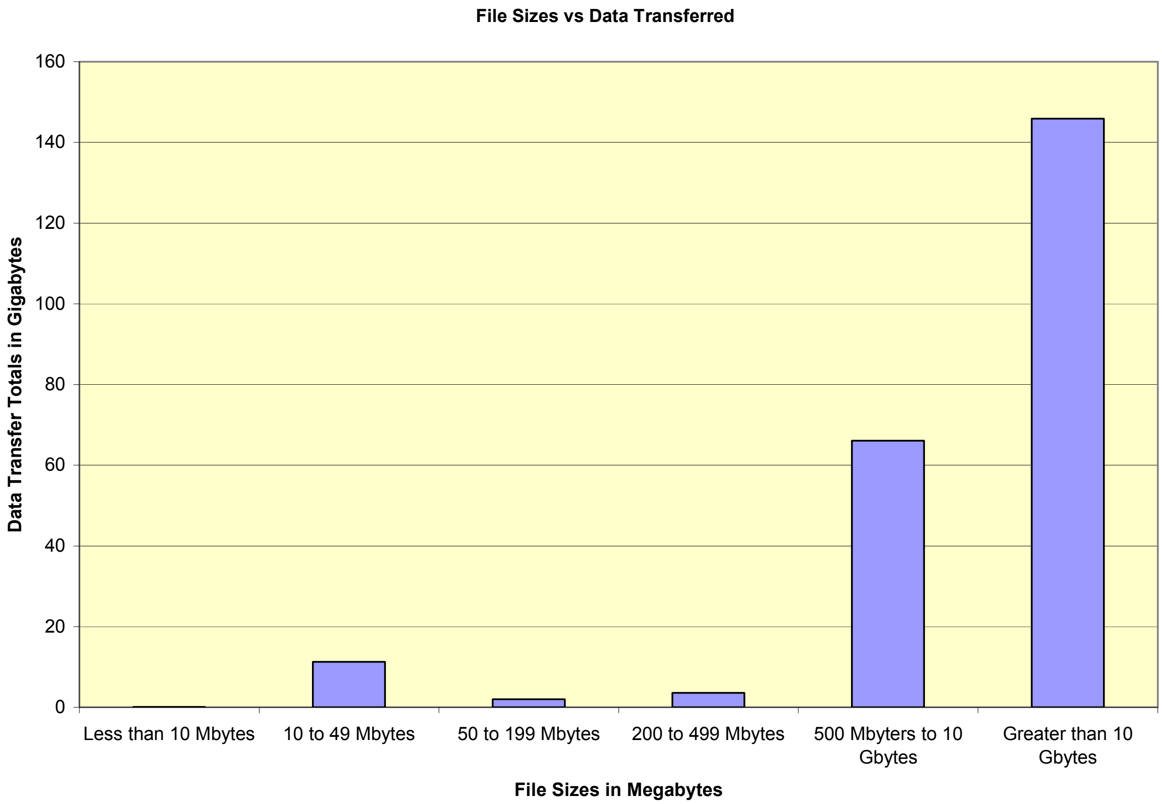

Figure 3: DisCom File Size Distribution vs. Data Transferred

Temporal Distribution of Large File Transfers

Large data files carried that majority of the data transferred, but the number of such files was small. The following graphic illustrates the aggregate bytes transferred between SNL/NM and LLNL/CA each hour for the month of August 2004. Hourly data transfer totals of 50,000 Megabytes or more (color other than blue) indicate a relatively large file transfer took place in that hour.

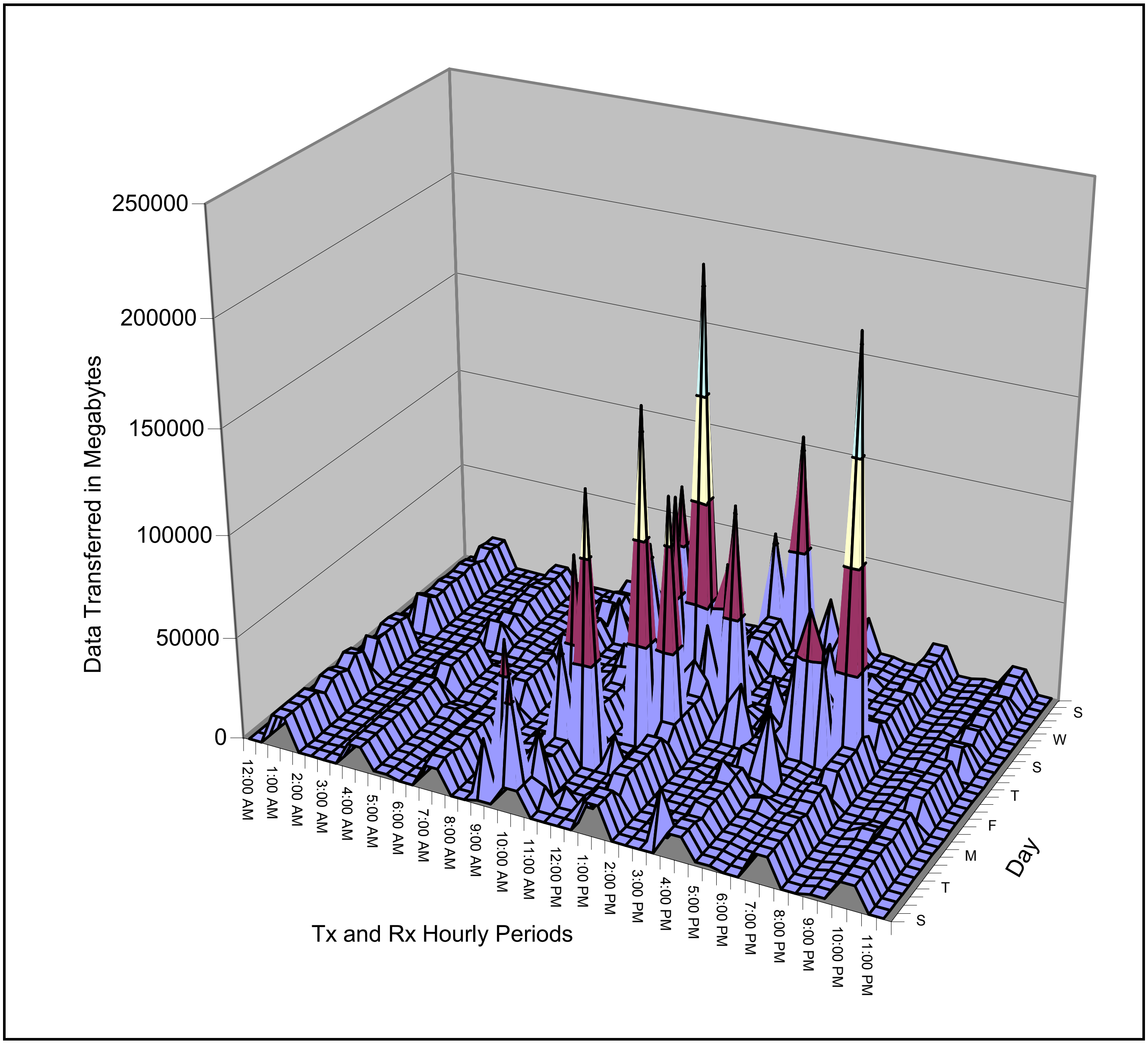

Figure 4: August 2004 - Large File Transfers Are Rare

One can conclude that large file transfers are relatively rare and spread out in the typical month. Therefore, we have identified two significant file transfer characteristics for the DisCom WAN;

- Most data files are small and
- Large data files transfer most of the data but such transfers are rare.

The busiest hour for the month of August transferred 210 Gigabytes of data which implies an average hourly data throughput of 500 Megabits per second (25% of the full line rate available).

Transport Stack Modification

Modifications to TCP

The Transmission Control Protocol (TCP) is a connection oriented protocol that provides applications with reliable, in-order delivery of data packets sent over networks that implement the Internet Protocol. TCP also provides flow control to prevent chronic network congestion as well as efficient multiplexing and demultiplexing of data streams sent from different processes on a single machine.

Currently the TCP congestion control algorithm is based on the algorithms proposed by Van Jacobson in 1988 [4]. This TCP congestion control algorithm was designed for much lower bandwidth speeds, around 32Kbps, than used today. Sandia National Laboratories, Los Alamos National Laboratory, and Lawrence Livermore National Laboratory are currently connected using 2.5Gbps network links which are over 78,000 times faster then the network speeds used in 1988. Alternative congestion control algorithms have been proposed to improve TCP network performance on such network links as these which are characterized by large distances and high bandwidth. The alternative congestion control algorithms tested and reported on here were included in two alternative TCP implementations, HighSpeed TCP and Scalable TCP.

Standard TCP uses the congestion control algorithms described in RFC2581 [5]. The algorithms used are: slow start, congestion avoidance, fast retransmit, and fast recovery. An active TCP connection is always operating according to one of these four algorithms throughout the life of the connection.

Summary of TCP Congestion Control Algorithms

Figure 5 gives an overview of the congestion control algorithms used in various TCP implementations.

	Increase	Decrease	a,b Values	Algorithm
Standard TCP	$a*MSS^2/Cwnd$	b*FlightSize	Constant	AIMD
HighSpeed TCP	$a(w)*MSS^2/Cwnd$	b(w)*FlightSize	Varies	AIMD
Scalable TCP	a*Cwnd	b*FlightSize	Adjustable	MIMD

Figure 5: Summary of Congestion Control Algorithms

The following sections define the performance variables and explain how the alternate algorithms function.

High Speed TCP

HighSpeed TCP was proposed by Sally Floyd as a sender-side alternative congestion control algorithm [6]. HighSpeed TCP uses the same slow start, fast retransmit and recovery algorithms that are implemented in Standard TCP. Modifications were only made to the congestion avoidance algorithm.

HighSpeed TCP attempts to improve the performance of TCP connections by using adjustable congestion windows. With large bandwidth, HighSpeed TCP utilizes large congestion windows to improve performance while at low bandwidth it behaves similarly to Standard TCP using small congestion windows.

Congestion Avoidance

HighSpeed TCP still uses an (Additive Increase Multiplicative Decrease) AIMD congestion avoidance algorithm similar to that of standard TCP. The changes made involved adjusting the increase and decrease parameters which are specifically the "a" and "b" parameters described in the standard TCP description listed in Figure 5 above. The value of the parameters is found in a table as a function of w. "w" is given by equation (1) as the current congestion window expressed in MSS segments.

$$\mathbf{w = Cwnd/MSS} \qquad (1)$$

Cwnd (bytes) is the current congestion window and MSS (bytes) the Maxmimum Segment Size.

The table in Figure 6 is created using equations (2) and (3):

$$\mathbf{a(w) = (HW^2 * HP * 2 * b(w)) / (2 - b(w))} \qquad (2)$$

$$\mathbf{b(w) = (((HD - 0.5) * (\log(w) - \log(HW))) / (\log(LW) - \log(w))) + 0.5} \qquad (3)$$

HW = 83000, HP = 10^{-7}, LW = 38, and HD = 0.1. The first 5 values generated for the HighSpeed TCP table are given below in Figure 6.

w	a(w)	b(w)
38	1	0.50
118	2	0.44
221	3	0.41
347	4	0.38
495	5	0.37

Figure 6: HighSpeed TCP Table

Additive Increase

The congestion window is represented in HighSpeed TCP's additive increase formula.

$$\mathbf{Cwnd = Cwnd + a(w)*MSS^2/Cwnd} \qquad (4)$$

The goal of HighSpeed TCP's additive increase is to open the congestion window by a(w) each RTT. Equation (4) allows for large congestion windows to open faster than the standard TCP additive increase algorithm would have allowed.

Scalable TCP

Tom Kelly proposed Scalable TCP as another alternative sender-side congestion control algorithm [7]. The goal of Scalable TCP is to quickly recover high network throughput when subject to short congestion periods.

Congestion Avoidance

Scalable TCP uses a different congestion avoidance algorithm than Standard TCP and implements Multiplicative Increase, Multiplicative Decrease (MIMD) algorithm rather than the AIMD algorithms of Standard TCP.

Multiplicative Increase

The multiplicative increase algorithm is operational in all the situations where the Standard TCP additive increase algorithm would normally be used. Equation (5) shows the formula used to adjust the congestion window after receiving a new ACK.

$$\mathbf{Cwnd = Cwnd + a*Cwnd} \tag{5}$$

For this implementation of the multiplicative increase algorithm, the parameter "a" is adjustable and was set to 0.02 for testing.

Multiplicative Decrease

The multiplicative decrease algorithm is the same as that implemented in Standard TCP except that the value of "b" in equation (6) is adjustable. The value of b used for testing was 0.125.

$$\mathbf{ssthresh = (1 - b)*FlightSize} \tag{6}$$

FlightSize is equal to the amount of data that has been sent but not yet acknowledged by the receiver. The slow start threshold, ssthresh, is a variable used to determine when the TCP connection should change from operating under the slow start algorithm to operating under the congestion avoidance algorithm.

SCTP Protocol

Stream Control Transmission Protocol (SCTP) is a message oriented, reliable transport protocol with direct support for multi-homing that runs on top of Internet Protocol (IP).

SCTP essentially combines the following features, which are either TCP or UDP like, into a single protocol:

- Reliable data transfer
- Congestion control
- Message boundary conservation
- Path MTU discovery and message fragmentation
- Ordered and unordered data delivery

Compared to TCP and UDP, the following new capabilities are also built into the SCTP protocol:

- **Multi-streaming** - This feature provides a user-controlled multiple stream data transfer mechanism within a single association between two endpoints. It is designed to solve the head-of-the-line blocking problem of TCP; but more importantly, it provides users the capability of delivering multiple independent data flows, either ordered or unordered, between two endpoints without the overhead of invoking multiple connections.
- **Multi-homing** - This feature allows a single SCTP association to run across multiple links or paths. With built-in heartbeat and failover capabilities, it provides transparent link/path redundancy to applications. Moreover, being in the transport layer protocol, SCTP Multi-homing does not poll the core routing tables the way today's most popular multi-homing solutions do.
- **Security and authentication** - The protocol is designed with a security cookie mechanism to prevent the SYN-flood attacks known for TCP. It also has checksum and tagging information in the protocol headers to ensure the data integrity and to provide protection against potential security offenders.

SCTP is described in RFC2960 [8] and RFC3286 [9]. Both the multi-streaming and multi-homing aspects of SCTP are of interest to DisCom as they would minimize efforts to modify applications to take advantage of parallel streams and simplify network design. A table from the University of Delaware CIS department shows a comparison of the three transport protocols [11].

SCTP vs TCP vs UDP

Services/Features	SCTP	TCP	UDP
Connection-oriented	yes	yes	no
Full duplex	yes	yes	yes
Reliable data transfer	yes	yes	no
Partial-reliable data transfer	optional	no	no
Ordered data delivery	yes	yes	no
Unordered data delivery	yes	no	yes
Flow control	yes	yes	no
Congestion control	yes	yes	no
ECN capable	yes	yes	no
Selective ACKs	yes	optional	no
Preservation of message boundaries	yes	no	yes
Path MTU discovery	yes	yes	no
Application PDU fragmentation	yes	yes	no
Application PDU bundling	yes	yes	no
Multistreaming	yes	no	no
Multihoming	yes	no	no
Protection against SYN flooding attacks	yes	no	n/a
Allows half-closed connections	no	yes	n/a
Reachability check	yes	yes	no
Psuedo-header for checksum	no (uses vtags)	yes	yes
Time wait state	for vtags	for 4-tuple	n/a

Figure 7: SCTP vs TCP vs UDP

The SCTP standard was published by the IETF in 2000 and the IETF Transport Area Working Group is currently continuing the standards process for SCTP [14].

Conclusions

TCP Modifications

Testing with High speed and Scalable TCP showed improved performance for single stream traffic. The performance gain with multiple, parallel streams was not as dramatic. Parallel, simultaneous streams effectively mask the impact of single packet drops well enough that Standard TCP performed within 7% of HighSpeed TCP when standard TCP was configured with two parallel streams. Using parallel streams, numbering more than two streams, with the alternative congestion control algorithms resulted in little, if any, performance increase compared to the standard congestion algorithm. For large numbers of parallel streams a decrease in performance was observed when using large window sizes. Standard TCP outperforms both HighSpeed and Scalable TCP when using large window sizes for 8 (not shown) and 16 parallel streams.

The figures below, extracted from previous works [1,2], show that the increased (best case of 12%-to 5%) from standard TCP currently does not warrant the modification to the TCP stack.

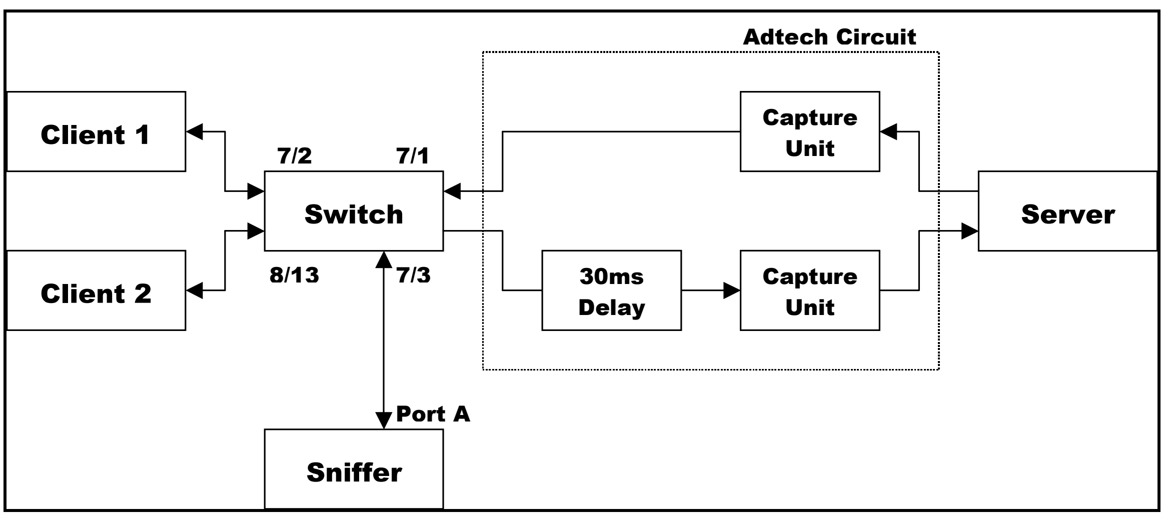

Figure 8: Network Diagram - Two-to-One 1Gpbs test setup with Delay

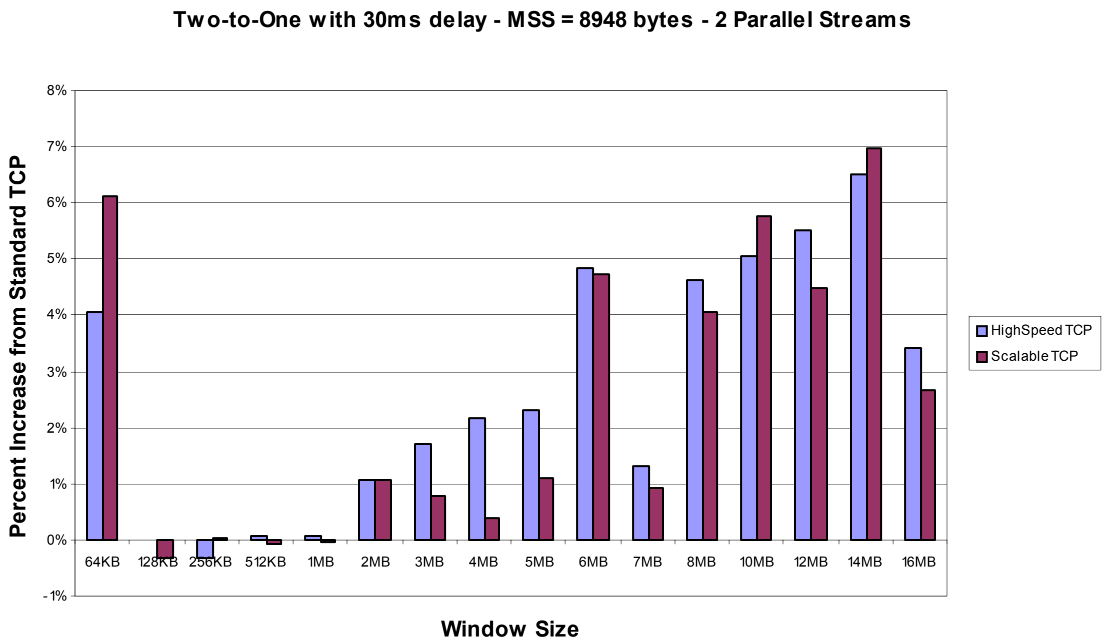

Figure 9: Percent Increase for Two Parallel Streams

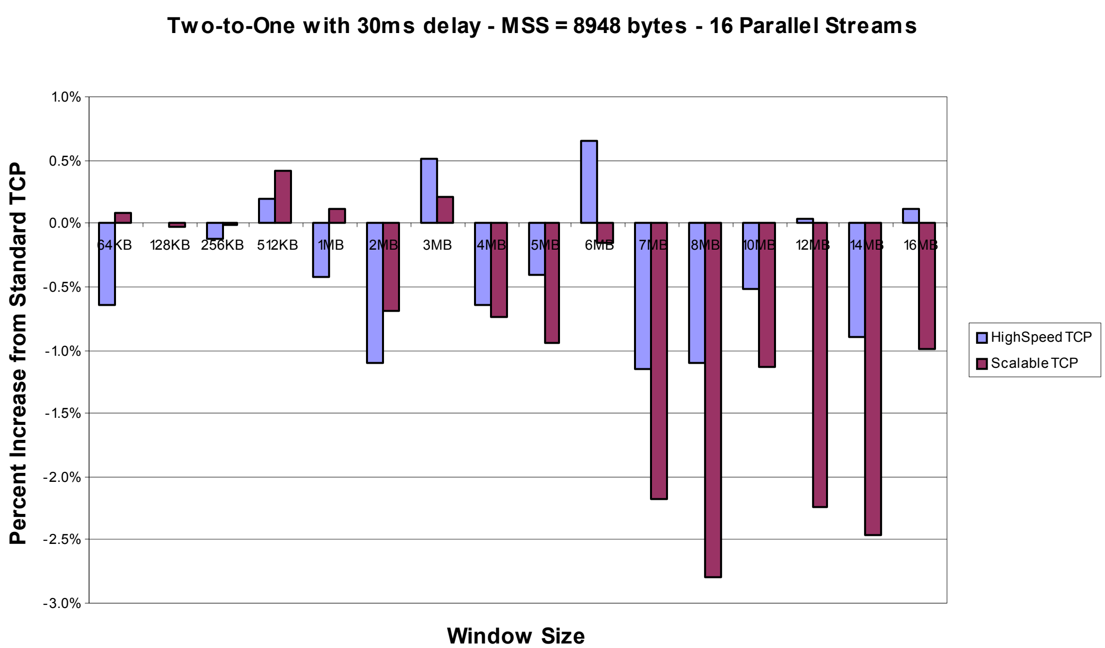

Figure 10: Percent Increase for Sixteen Parallel Streams

Increasing the client network link bandwidth from 1Gbps to 10Gbps did not significantly change the performance difference. Using 2 parallel streams the largest difference between the modified TCP and Standard TCP was around 12% and dropped to 5% at 8 parallel streams. The main bottleneck is the 10Gpbs-to-1Gbps (2.5Gbps in actual operations) transition going from the LAN to the WAN.

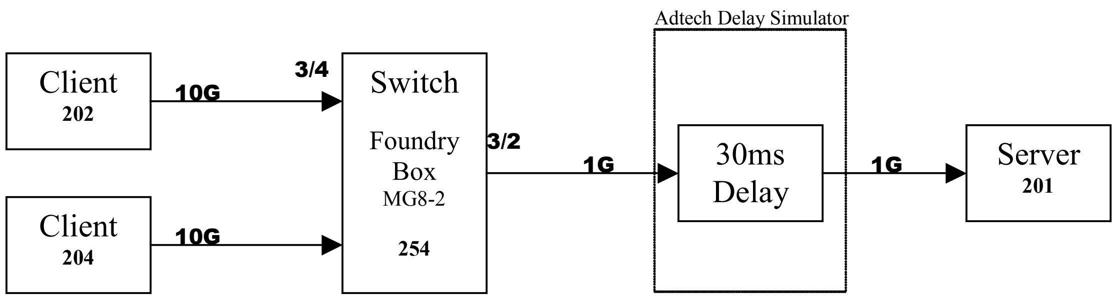

Figure 11: Network Diagram 2: 10Gbps to 1 Gbps test setup with 30 msec Delay

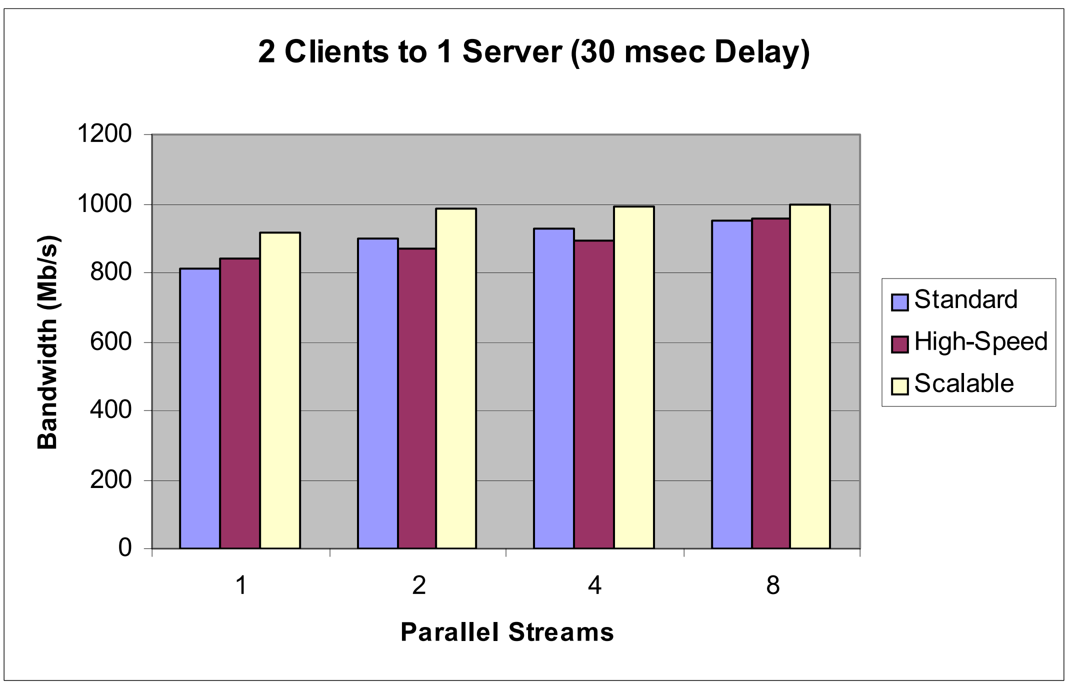

Figure 12: 2 Clients to 1 Server: With 30 msec Delay

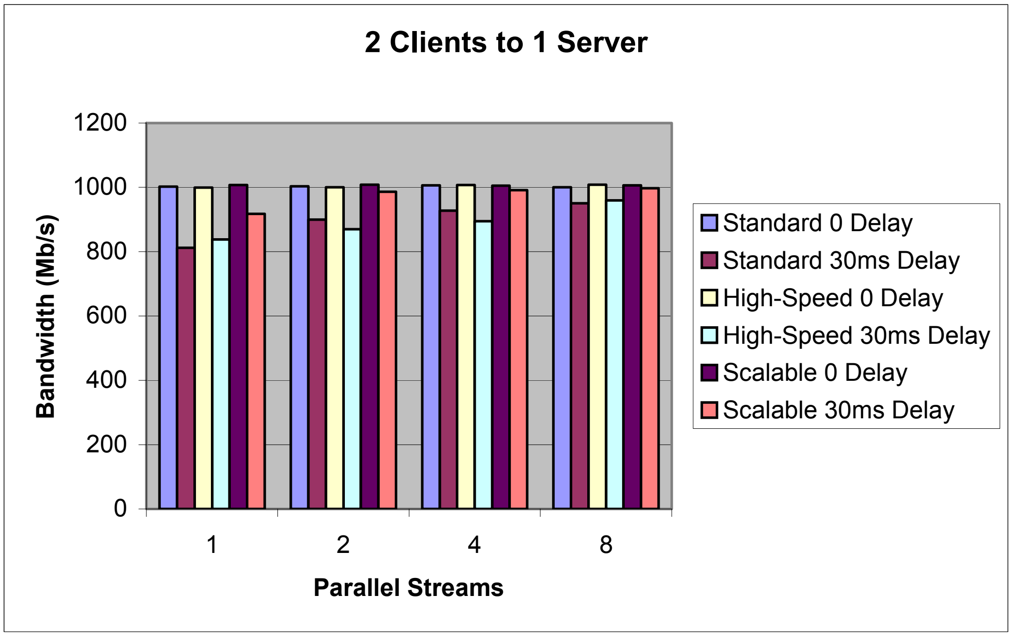

Figure 13: 2 Clients to 1 Server: Overall

SCTP

As the SCTP standard was released in 2000, the maturity and performance of SCTP is of concern. After a search for available SCTP performance data, the analysis of the information collected infers that implementations of SCTP lag behind TCP in terms of raw throughput [12, 13, 17] in environments with low error rate. Two systems were set up in the test laboratory using Linux kernel version 2.6.11 with the built in the lksctp package. Initial testing with a modified version of Iperf from the http://www.openss7.org site indicated that SCTP performance was about ~30% less compared to TCP during memory-to-memory transfer. Although the results were not on par with standard TCP, they were quite an improvement compared to the results previously obtained [18]. Surprisingly, SCTP throughput did not change with larger MTU's. Upon closer examination of Ethereal network traces, it was found that the Openss7's Iperf SCTP test did not take advantage of the larger MTU. Using another performance tool, sctpperf written by Pawel Hadam, it was found that SCTP could perform markedly better in a jumbo frame, minimum network delay network environment. Figure 14 below shows the test layout.

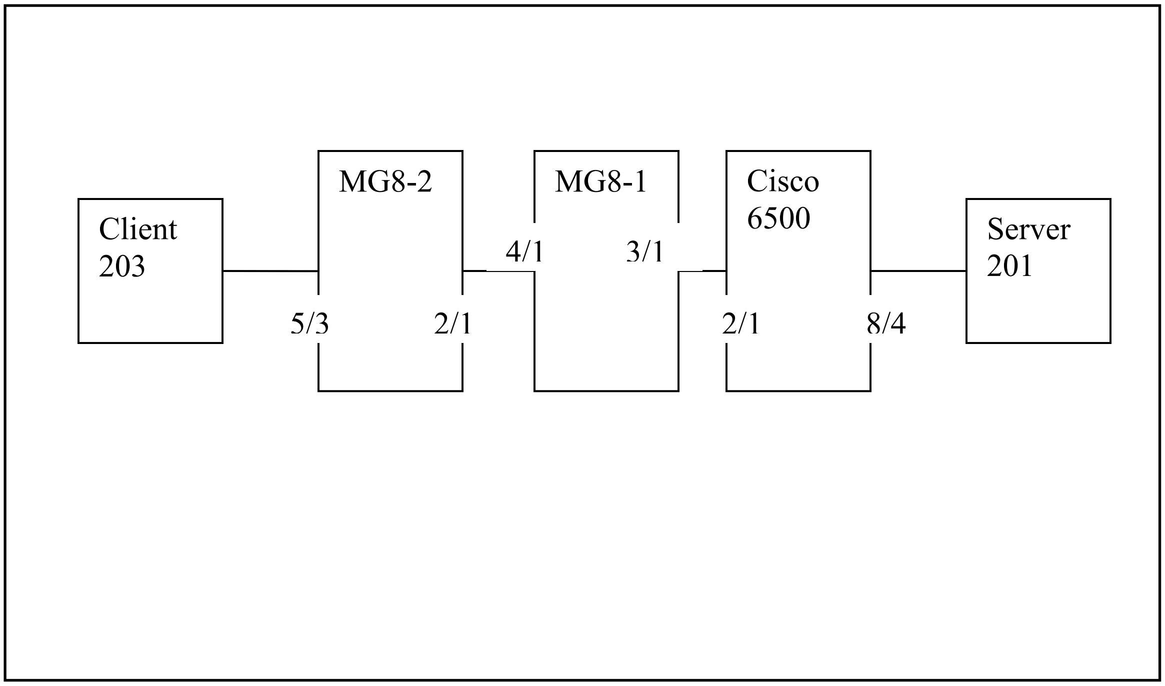

Figure 14: SCTP client-server setup via 1Gbps links

Recommendations

Based on historical user data, there does not appear to be a driving reason to modify the currently installed standard TCP implementations or migrate to an alternative. However, we should continue to monitor the WAN usage patterns to see if the usage models change.

There is ongoing research to improve SCTP performance over large bandwidth, long delay, networks [15, 16] including the incorporation of alternative congestion algorithms. We recommend that DisCom development efforts continue to track the progress of SCTP development and develop better tools for performance testing. Implementation of SCTP could achieve the advantages of parallel FTP within the scope of a standard protocol without the necessity of supporting custom applications. In addition, SCTP could provide automatic failover for encrypted circuits that utilize encryptors in parallel to achieve high data rates.

References

[1] Robert M. Long, "Evaluation of TCP Congestion Control Algorithms, SAND2003-4404, Sandia National Laboratories, December 2003.

[2] Randy L. Romero Jr., "Evaluation of Replacement Protocols and Modifications to TCP to Enhance ASC Wide Area Network Performance", SAND2004-4207, Sandia National Laboratories, December 2004.

[3] Asim Iqbal, "SCTP Performance Tests", WP3, task 3.3 of the EU-funded DataTAG Project, 2003.

[4] V. Jacobson. Congestion Avoidance and Control. In *SIGCOMM 1988*.

[5] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control, *Internet RFC 2581*, April 1999.

[6] S. Floyd. HighSpeed TCP for Large Congestion Windows. *Internet Draft* < draft-ietf-tsvwg-highspeed-01.txt>, August 2003. Work in progress.

[7] T. Kelly. Scalable TCP: Improving Performance in Highspeed Wide Area Networks, URL http://www-lce.eng.cam.ac.uk/~ctk21/scalable/, December 2002

[8] IETF RFC 2960, "Stream Control Transmission Protocol", October 2000

[9] IETF RFC 3286, "An Introduction to the Stream Control Transmission Protocol (SCTP)", May 2002

[10] IETF RFC 3309. "Stream Control Transmission Protocol (SCTP) Checksum Change", September 2002.

[11] URL http://www.cis.udel.edu/~amer/PEL/sctp.vs.tcp.vs.udp.html .

[12] Sourabh Ladha, Paul D. Amer, "Improving File Transfer Using SCTP Multihoming", IPCCC 2004, Phoenix, Arizona, April 2004.

[13] Elvis Pfutzenreuter, "Applicability and Performance of the SCTP transport Protocol", Master dissertation, Santa Catarina Federal University, Brazil

[14] IETF Transport Area Working Group, URL http://www.ietf.org/html.charters/tsvwg-charter.html .

[15] SCTP research and simulation paper page, URL
http://www.sctp.be/sctpresearch.htm .

[16] IshtiaQ Ahmed, Okabe Yasuo, Kanazawa Masanori, "Improving Performance of SCTP over Broadband High Latency Networks", IEEE International Conference on Local Computer Networks 2003.

[17] R. Rajamani, S. Kumar, and N. Gupta. "SCTP versus TCP: Comparing the performance of transport protocols for web traffic," University of Wisconsin-Madison, May 2002. http://www.cs.wisc.edu/~raj/sctp/report.pdf (March 28, 2003)

[18] SCTP performance results, URL
http://datatag.web.cern.ch/datatag/WP3/sctp/tests.htm

Distribution

MS 0321	R.W. Leland, 9300
MS 0622	D.S. Rarick, 9310
MS 0823	J.D. Zepper, 9320
MS 0801	J.A. Larson, 9330
MS 0806	G.K. Rogers, 9312
MS 0822	C. Pavlakos, 9326
MS 0813	R.M. Cahoon, 9311 (2)
MS 0805	W.D. Swartz, 9329 (2)
MS 0788	M.J. Benson, 9334 (4)
MS 0788	P.A. Manke, 9338 (4)
MS 0806	Len Stans, 9336
MS 0806	J.M. Eldridge, 9336
MS 0806	S.A. Gossage, 9336
MS 0806	T.C. Hu, 9338 (6)
MS 0806	B.R. Kellogg, 9336
MS 0806	L.G. Martinez, 9336
MS 0806	J.H. Naegle, 9336
MS 0806	T.J. Pratt, 9338
MS 0806	L.F. Tolendino, 9334 (6)
MS 0806	J.S. Wertz, 9336
MS 9151	K.E. Washington, 8900
MS 9012	R.D. Gay, 8949
MS 9158	H.Y. Chen, 8961
MS 9018	Central Technical Files, 8945-1
MS 0899	Technical Library, 9616 (2)