# SCPS-TP, TCP AND RATE-BASED PROTOCOL EVALUATION FOR HIGH DELAY, ERROR PRONE LINKS

Frances J. Lawas-Grodek, Diepchi T. Tran,
Robert P. Dimond, and William D. Ivancic
NASA Glenn Research Center

## ABSTRACT
This paper describes testing performed to validate operation of Space Communications Protocol Suite Transport Protocol (SCPS-TP) relative to the specification and perform a comprehensive comparison of SCPS-TP protocol options to IP based protocols.

Tests were performed at Glenn Research Center to validate the operation of SCPS-TP relative to the Consultative Committee on Space Data Systems (CCSDS) specification, to perform a comprehensive comparison of SCPS-TP protocol options to IP based protocols, and to determine the implementation maturity level of these protocols – particularly for higher speeds. The testing was performed over reasonably high data rates of up to 100 Mbps with delays that are indicative of near planetary environments. The tests were run for a fixed packet size, but for various errored environments. The results indicated that SCPS-TP congestion-friendly options perform slightly better than TCP SACK protocols at moderate and high error-rates. The results also show that existing standard transport protocols and capabilities (drawn from a variety of communities) appear to satisfy all known mission needs.

## INTRODUCTION
In the late 1980's and throughout the 1990's, the Internet has rapidly developed allowing vast improvements in communication and networking. These technologies utilize packet-based communications rather than circuit-based communications. The Consultative Committee on Space Data Systems (CCSDS) foresaw the need to take advantage of this new Internet technology and developed the Space Communications Protocol Suite (SCPS) to address some specific issue related to space systems. Thus, the TCP/IP protocol suite was investigated and modifications to the networking, security and transport protocols were specified. These specifications as know as the SCPS security protocol , network protocol, transport protocol and file transfer protocol (SCPS-SP, SCPS-NP, SCPS-TP and SCPS-FP)

## PURPOSE
There have been numerous debates regarding the actual improvements that SCPS may provide over the ever-evolving TCP/IP protocol suite. In addition, much of the SCPS initially testing and demonstrations often did not provide what many consider to be a valid comparison relative to TCP as known improvements to TCP for long bandwidth-delay networks were often not implemented (i.e. large windows, selective acknowledgements) [1]. Other testing often was performed over simulated links where SCPS would provide little advantage due to the very low bandwidths [2]. Some well documented and thorough testing has been performed at lower rates. These results correlate well with our test results [ 3,4 ].

In order to get a better understanding of the actual improvements, if any, that SCPS could provide relative to TCP and to determine the maturity of the various protocols for higher-rate links, the NASA Space and Data Communications Systems (SCDS) Office requested Glenn Research Center to perform a comprehensive set of tests. This paper is a summary of the comprehensive set of tests that are currently being documented in a NASA Technical Report [5]. This testing was only performed for SCPS Transport Protocol (SCPS-TP). None of the security mechanisms or networking protocol was implemented or tested.

## TESTBED CONFIGURATION

Our testbed environment is shown in Figure 1. It was configured to run either single-flow tests or multi-flow tests. The single flow tests were for baselining congestion friendly TCP and SCPS-TP performance and for evaluation of rate-based protocols.

For the single-flow and rated-based tests the testbed is shown by the solid lines in Figure 1 and consists of the following: Two separate networks, representing a terrestrial and space network, are joined by an Adtech SX/14 channel emulator that allows time delays and random bit errors to be inserted into the network flow. Each side of the emulated space channel services a network consisting of a CISCO 7100 router, connected to the SX/14 via ATM, and a CISCO 2900 Catalyst Ethernet switch, connected to the routers via fast Ethernet. The Catalyst switches then serve as our LANs, connecting to the originator or receiver for our tests. Two Sun machines running Solaris 7 configured in full duplex mode act as either the receiver or the originator of packets. To monitor and capture the network data, two PCs running NetBSD 1.5 are mirrored to a corresponding Sun test machine.

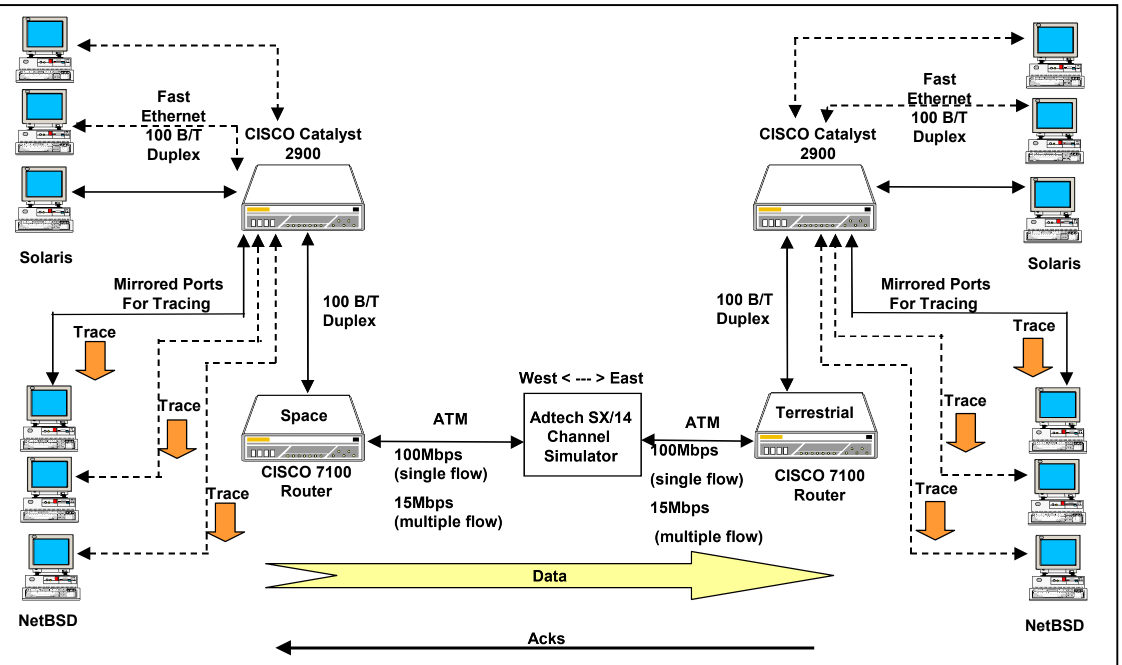


**Figure 1: Testbed Configuration**

The network architecture for multiple flow testing is similar to that of the single flow testing, with the exception that there are now three sender/receiver machines on our aforementioned terrestrial-space Internet. Three monitor/capture machines are also required, as a machine must be dedicated to monitor a mirrored port in a switched environment. In addition, the data transfer rate between the to networks comprising our Internet was decreased to 15 Mbps by rate limited VCs on the ATM ports of our CISCO 7100 routers. The rate limiting was enabled to assure congestion would occur during the multiple flow transmission tests. The dashed lines in Figure 1 connect those additional machines needed for the multiple flow tests.

### *Hardware*

The following workstations were used for our testing. The TCP and SCPS single flow testing was performed using the Sun Ultra II machines.

**Sending Machines**

1. Sun Ultra II, 200 Mhz, 512 Mb RAM, Solaris 7
2. Sun Ultra I, 143 Mhz, 64 Mb RAM, Solaris 7
3. Sun Ultra 10, 440 Mhz, 132 Mb RAM, Solaris 8

**Receiving Machines**

1. Sun Ultra II, 200 Mhz, 256 Mb RAM, Solaris 7
2. Sun Ultra 10, 440 Mhz, 132 Mb RAM, Solaris 8
3. Sun Ultra 5, 270 Mhz, 256 Mb RAM, Solaris 7

### *Software*

We utilized TCP-SACK because SACK is becoming widely deployed standard throughout the Internet along with fast retransmit. The TCP protocol that came with the 7.0 Solaris kernel with the SACK option enabled was used for these tests.

The SCPS Reference Implementation (SCPS-RI) versions 1.1.51, 1.1.62, and 1.1.66 were used, as provided by the MITRE Corporation. This is an out-of-kernel implementation and the only implementation available to us at the time we began testing. Some new, in-kernel implementations of SCPS are now becoming available [6,7 ] and may provide significant improvements for the high-rate testing.

Additional software tools: tcpdump [8], tcptrace, and xplot[9] used for the TCP-SACK tests. Modified tcpdump and tcptrace by MITRE, renamed "scps_tcpdump" and "scps_tcptrace", were used for the SCPS tests.

For rate-based protocol testing, we used the Multicast Filed Transport Protocol (MFTP) from StarBurst, the Multicast Dissemination Protocol (MDP) from the Navel Research Laboratory and various rate-based options in the SCPS-TP protocol. Both MFDP and MDP were developed for multicast applications, but can be utilized for unicast operation. The MFTP Version 3.05 software was used but is no longer commercially available as a stand-alone product. In addition, as developed, it has an upper limit transmit rate setting of 50 Mbps. However, since we had previously used this software and were familiar with its operation we felt this would provide a meaningful set of data points for one implementation of a commercial rate-based protocol. The MDP version we used was 1.9a4 compiled with in house modifications optimized for higher data rates.

## TESTING PHILOSOPHY AND PROCEDURES

Both TCP and SCPS appeared to more closely match the theoretical characteristics of the protocol in Solaris than in BSD even though for some configuration BSD gave better performance[i]. Thus, we decided to perform all test with Solaris.

During our initial investigations and baselining, we noted that none of the protocols, TCP or SCPS, were performing consistently according to specifications and theory. For example,

---

[i] See the detailed report [5] for documentation of the anomalies noted in the BSD implementations for both TCP-SACK and SCPS-TP.

occasionally a packet would be lost in a TCP transfer somewhere in the operating system (not in the link) even for an error-free links. Thus, TCP would enter congestion control. We saw unusually and inconsistent discrepancies in the SCPS-TP performance also. For example, SCPS would suddenly halve its transmission rate even though no packets were lost. We concluded that this data is valid and should not be discarded as such inconsistencies are common in operational implementations. Thus, our testing philosophy evolved to the following:

- Tune and baseline protocol on error-free link for each bandwidth-delay product.
  - Both SCPS-TP and TCP where tuned for best performance over the given delay.
- Record all measurements, not just optimal runs!
- Perform 30 runs for TCP-SACK and all SCPS-TP protocols and 20 runs for MDP and MFDP protocols.
- Measurement time is from SYN to FIN (or start to finish for MDP and MFDP).
- Run single flows and multi-flows (3 connections) to ensure accurate reporting and application of results.
- Capture and save some complete trace files – particularly when the unexpected is occurring.

Testing was performed in an automated fashion using script files. The producer was as follows:
1) Remotely set the Bit Error Rate (BER) and transmission delay on the Adtech channel simulator for each test.
2) Login to the ttcp receiver (Terrestrial) and the ttcp sender (Space) through secure shell login (ssh), invoking 30 file transfers for different file sizes using input files under various BER and delay conditions.
3) Using tcpdump/scps_tcpdump to capture the SYN and FIN packets for all 30 runs, and keeping only two dump files for each file size on both the sender and receiver sides.
4) From the output of the tcpdump/scps_tcpdump captured on sender side, calculate the transfer time of each test run by the time stamp of SYN packet sent from the sender, to the time stamp of the FIN ACK packet sent from the receiver.

Once the data was obtained, we calculate the average throughput of 30 runs using the transfer time previously defined above for each test run as well as the standard deviation of each of the 30 throughputs for each file size transferred.

In the multi-flow testing, there are three pairs of sending and receiving machines, there are six possible combinations of sending order among the three senders[ii]. These combinations are picked randomly using the output of a random number function. In addition, by using the same random function, each sender is also randomly started from one to eight seconds apart from the previous sender.

## *Defined Variables*

The following are a summary of the variables and options used in both the TCP and SCPS protocol single flow tests:

File sizes: 100 Kbytes, 1 Mbytes, 10 Mbytes, 100 Mbytes
Delays: 10 ms, 250 ms, 500 ms
BERs: zero (baseline), 1e-8, 1e-7, 1e-6, 1e-5, 1e-4[iii]
TCP option: SACK

---

[ii] We utilized three separate transmitting and receiving machines because our out-of-kernel SCPS-TP implementation would not allow a single sending host to inject three separate flows into three different ports.
[iii] Tests will be run only if the average throughput of the test at 1e-5 BER is not less than 1 Mbps.

SCPS options for SCPS tests:
1) Van Jacobson Congestion Control, and ACK Every Other Packet (SCPS-VJ).
2) Pure Rate Control with ACK Every Other Packet (SCPS-Pure Rate Control, Option F2).
3) Pure Rate Control with Strictly Delayed ACKs (SCPS-Pure Rate Control, Option F0).
4) Vegas Congestion Control with ACK Every Other Packet, and Assume Congestion (SCPS-Vegas Congestion)
5) Vegas Assume Congestion (SCPS-Vegas Congestion)

The details of each option are described in our SCPS/TCP report [5].

Due to time and funding limitations, the multiple flow testing for TCP-SACK, SCPS-VJ and SCPS-Vegas-Congestion was performed using only 50 Mbytes file sizes at a 500 ms delay with the BER set to zero, 1e-7, and 1e-5.

## THEORETICAL BOUNDS

Figure 2 shows the theoretical throughput for TCP and rate-based protocols with packet sizes of 1024 Bytes. Smaller packets will improve the performance of all protocols at higher error rates while reducing the performance slightly at lower error rates. This reduction is due to the packet overhead relative to the payload size and to slow start in TCP. All congestion friendly protocols that use the additive increase, multiplicative decrease congestion control algorithms will have a similar throughput characteristic. Notice, that such congestion friendly protocols perform poorly for long file transfers over error prone, long bandwidth-delay product links. Rate-based protocols are far more tolerant of errors and are relatively insensitive to delay – particularly for very large file transfers.

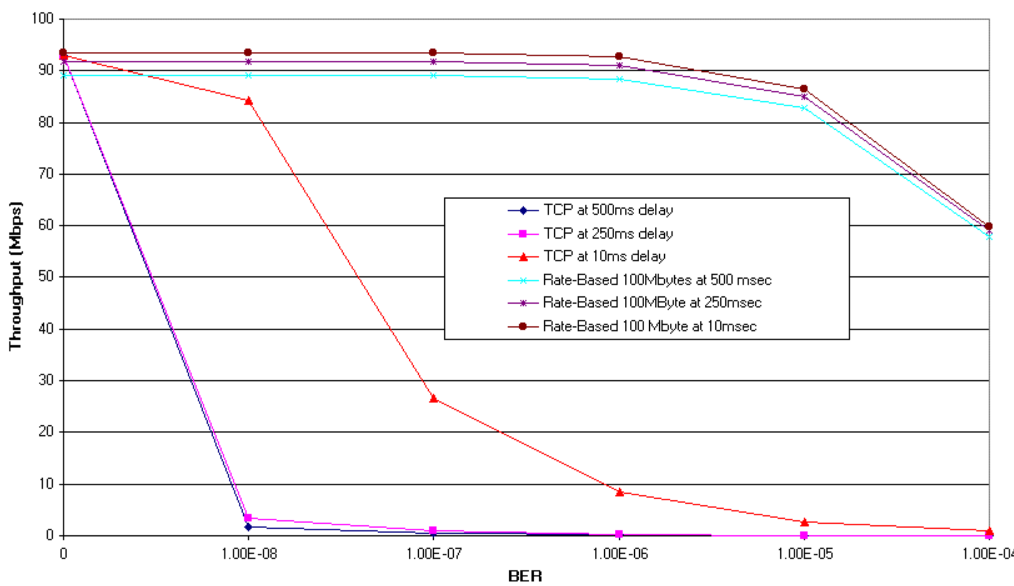


**Figure 2: Theoretical Throughput for TCP and Rate-Based Protocols (1024 Byte Packet)**

The maximum theoretical throughput for TCP and SCPS-VJ is given by in equation (1). Note that this formula if for steady state conditions assuming an infinite file transfer where slow start and the time to perform a three-way handshake is no longer relevant [10].

Bandwidth = 0.93*MSS/RTT*sqrt(p) (1)

Where
MSS = maximum segment size
RTT = Round Trip Time
p = packet error rate

A first order approximation of the theoretical throughput for rate-based protocol for each delay also was calculated using equation (2). Note that we have included the effects of the round trip time needed to confirm reception of the final data packet. The effect of overhead on the throughput is included as (payload_size)/(payload_size+packet_overhead).

$$\text{Throughput (Mbps)} = \frac{(1024)\ \text{File size} * 8}{(1024+58)*(\ (\text{File size}*8*p/R) + (\ \text{File size}*8\ /R) + RTT)} \qquad (2)$$

Where
R = setting rate (Mbps)
p = packet error rate
File size in Mbytes
RTT = round trip time (second)

**TESTING RESULTS**

For congestion friendly protocols such as TCP, SCPS-Van Jacobson, SCPS-Vegas-Assume-Congestion and SCPS-Vegas-Assume-Corruption, single flow tests were performed to baseline the performance. These tests were then run for multiple flows with the three receiver buffers set to the bandwidth-delay product associated with a 15 Mbps link and 500 msec delay. The multi-flow tests are the most meaningful for congestion friendly protocols, as one would be well advised to utilize a rate-based protocol if one knows the available bandwidth and could be assured that no congestion occurs in that link.

For rate-based protocols such as MFDP, MDP and SCPS-Pure-Rate-Control, only single flow testing is necessary as one would not be advised to utilized a rate based protocol in a system where congestion is present.

All test results for both the single-flow, multi-flow and rate-base protocols tests as well as the detailed protocol configuration parameters (buffer sizes, timer settings and protocol versions) are provided in the detailed test report [5]. A few of those tests, which provide insight to the overall performance of the protocols, are highlighted in the following sections.

***Single Flow Results***

Figure 3 shows the single flow tests for packet sizes of 1024 bytes and RTTs of 500 milliseconds. For moderate and high BER, SCPS-Vegas-Assume-Congestion performs slightly better then TCP-SACK; whereas TCP-SACK performs slightly better than SCPS-Van Jacobson at zero and moderate BER. Notice that larger files have a better throughput at zero BER than smaller files. This is because slow start has less affect as files get larger. Also, notice that smaller files have better throughput at higher BERs than larger files. For TCP-SACK and SCPS-VJ, this is due to the additive increase, multiplicative decrease congestion control algorithms. SCPS-Vegas-Assumed-

Corruption performs better than all other congestion friendly protocols at higher BER, but still does not perform as well as a pure rate-based protocol as there is some congestion control being performed [11].

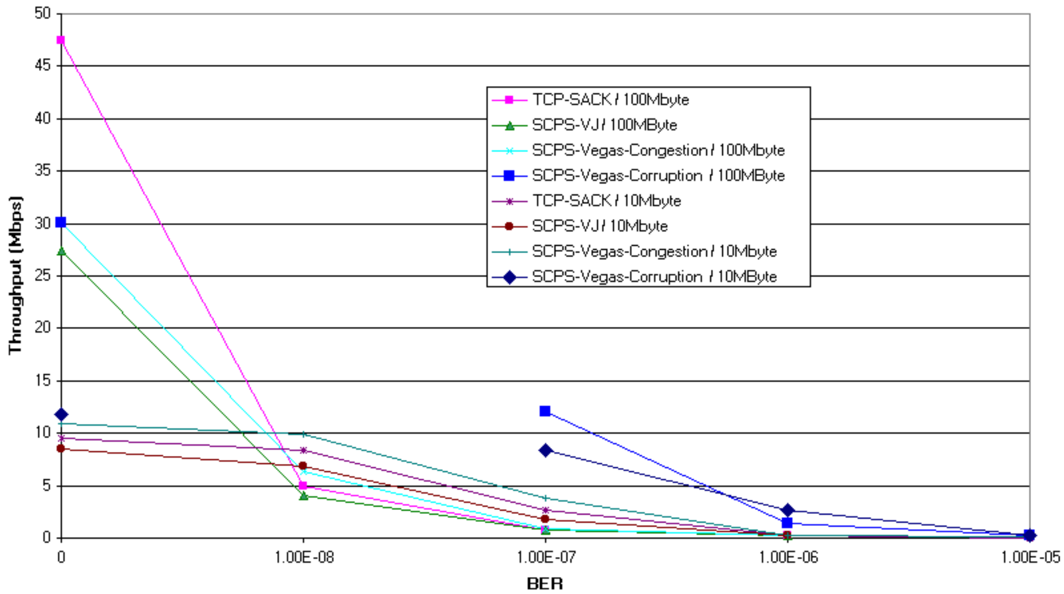


**Figure 3: Throughput vs. BER (1024 Byte Packet, 500 msec RTT)**

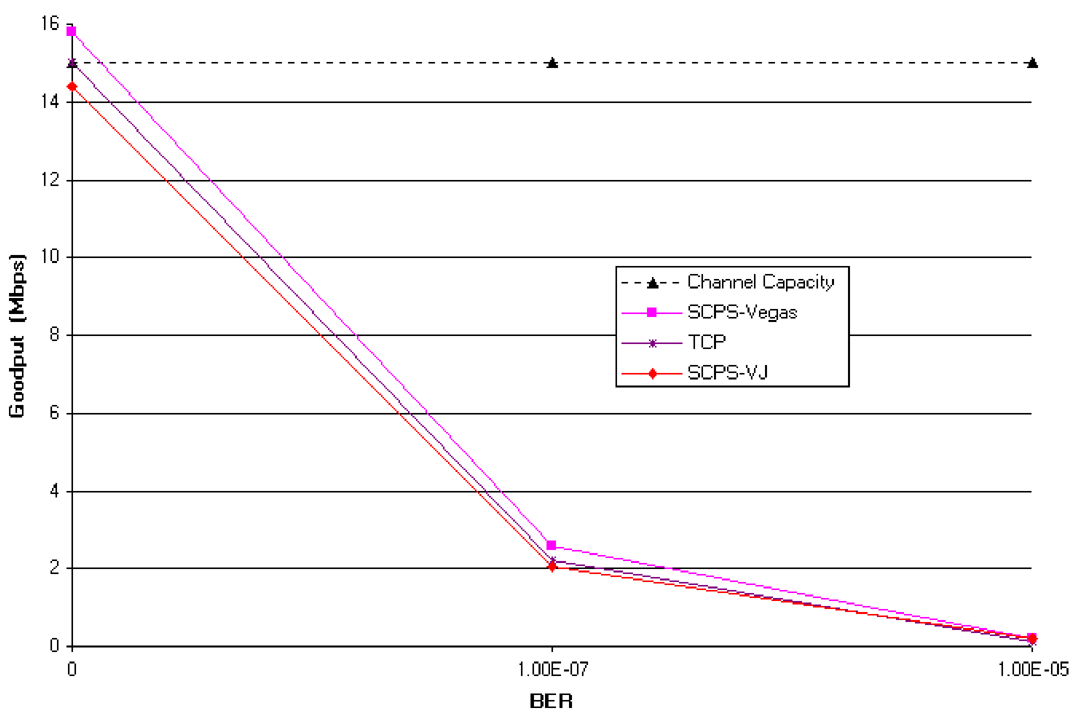


**Figure 4: Total Throughput for Congestion Friendly Protocols 3 (nearly) simultaneous flows over a 15 Mbps rate limited channel**

### *SCPS Multiple Flow Results*

Unlike the single flow tests, all three pairs of flows competed for the available bandwidth. For these tests, the bandwidth was set to 15 Mbps at the ATM interfaces of the routers. SCPS-Vegas-Assume-Corruption was not tested in this environment, as it would have been a miss-application of the protocol.

Similar to the single flow tests, SCPS-Vegas-Assume-Congestion performed slightly better than TCP-SACK, which performed slightly better than SCPS-VJ at zero and moderate BERs. At a BER of 1e-7 and 1e-5, the throughput of each pair in multiple flow tests had almost the same performance as in the single flow tests under these same error conditions. This was because the packet loss due to errors dictates the performance rather than actual congestion. Notice that the total average throughput can exceed the network capacity. This is due to the random offset start times for the three flows, where the flows are all started at random intervals, causing individual flows to transfer and complete during different usages of the available bandwidth. The total average throughput of any one set of tests can exceed the 15 Mbps, particularly if the first and last transfers do not overlap by much.

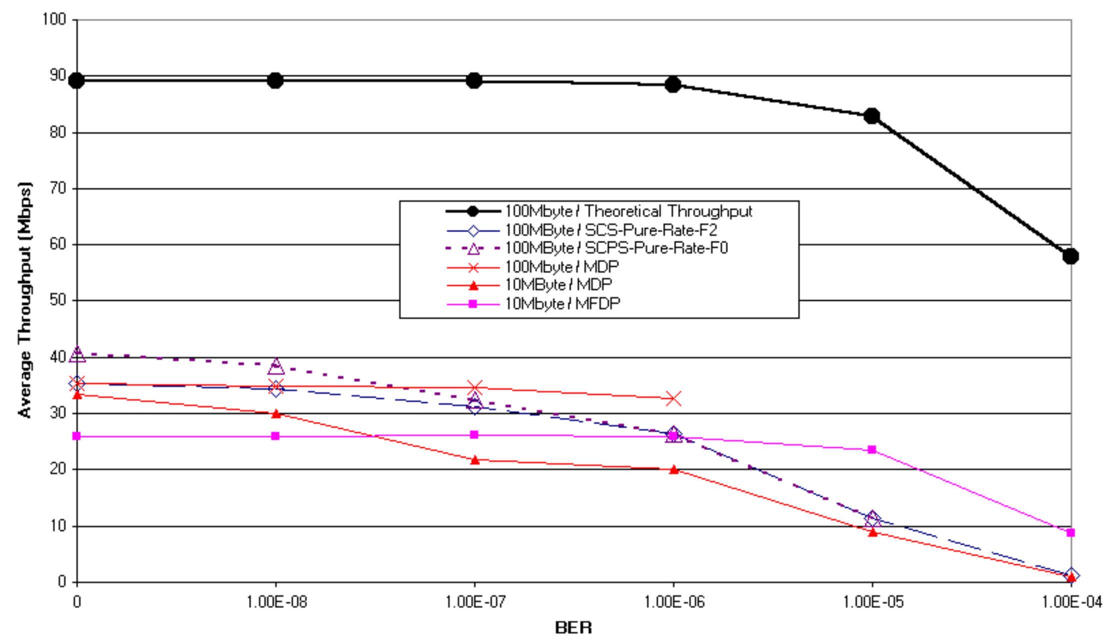


**Figure 5: Rate-Based Protocol (1024 Byte Packet, 500 msec RTT)**

### *Rate-Based Protocol Results*

Figure 5 shows the test results for pure rate-based protocols with packet sizes of 1024 bytes. None of the rate-based protocols tested came close to meeting the theoretical throughput when we transmitted large files. This is probably due to the capabilities of our machines along with implementation issues. MFDP was the most consistent performer at high BER. MDP performed well up to approximate 20 Mbps. After that, the receiver machine of MDP had difficulty performing all the necessary processing to match the sending machine. In face, for 100 Mbyte files, MDP would not complete the transfer as the receiver would choke. SCPS rate-bases protocol also fell off faster than predicted at the high BERs and did not perform anywhere near theoretical even at low BERs. An in-kernel implementation of SCPS may improve performance.

**CONCLUSIONS**

We have studied the effect of delay and BER on the performance of congestion friendly and rate based protocols in uncongested and limited congested emulated space links. The results correlate well with other testing of SCPS-TP and TCP.

- The single stream and multi-stream test results clearly illustrate that the SCPS-Vegas enhancements to TCP provide measurable performance improvements over the TCP SACK implementation tested. The value of these performance increases is subjective and would need to be judged on a mission by mission basis.
- Very small transactions such as command and control should see little difference in performance for TCP or any variant of SCPS-TP or a rate-based protocol.
- In extremely errored environments with high RTT delays, a rate-based protocol is advisable if you properly engineer the network. However, one must beware of using rate-based protocols on shared networks unless you can reserve bandwidth. In addition, rate-based protocols may be applicable for any environment where bandwidth reservation is practical and available.
- Even with equal performance, the deployment of an in-kernel rate-based protocol such as SCPS rate-based protocols may be more desirable than the deployment of MDP or other application level protocols when unicast data delivery is the goal. The SCPS rate-based protocol is a sending-side only modification; thus, all "standard" TCP applications can be deployed without modification.
- The existing standard transport protocols[iv] and capabilities (drawn from a variety of communities) appear to satisfy all known mission needs; however, the space community should maintain as awareness of current and future TCP research. New TCP research may dramatically improve TCP operation for near planetary environments. Some pertinent areas include Stream Control Transmission Protocol (SCTP), TCP Pacing with Packet Pair Probing, TCP Westwood, and TCP Explicit Transport Error Notification (ETEN).

**RECOMMENDATIONS**

Whereas the SCPS rate-based protocols utilize the TCP header and appear as sender only modifications, the protocol is advertising TCP as the protocol number. However, SCPS-TP pure rate-base is not performing congestion control. The authors suggest that for such operation SCPS-TP should advertise a different protocol number to ease quality-of-service provisioning. Failure to do so may result in SCPS rate-base flows dominated shared links or being identified as rogue sources.

All rate-based protocols need further testing on faster machines and for different operating systems to determine if that will improve performance or if these protocols need further

---

[iv] SCPS-Network Protocol was not evaluated in this study due to lack of hardware and software implementations. All routing was performed over IPv4, which is deployed throughout the Internet. Note that SCPS-NP will not accommodate the National Security Agency's Internet. High Assurance Internet Protocol Interoperability Specification (HAIPIS); thus, use of SCPS-NP for secure government applications may be problematic.

development. For the commercial rate-based protocols tested, this poor performance may be due to the algorithms and coding being optimized for multicast operation.

An investigation of SCPS performance under different operating systems and using faster machines is recommended. In addition, using an in-kernel SCPS-TP implementation may result in better performance of the SCPS protocols.

For specific environments, SCPS-Vegas should be investigated as the Vegas algorithm has some know problems [12,13].

- Vegas uses an estimate of the propagation delay, *baseRTT*, to adjust its window size. Thus, it is very important for a TCP Vegas connection to have an accurate estimate of this quantity. Rerouting a path may change the propagation delay of the connection, and this could result in a substantial decrease in throughput. Therefore, Vegas may not perform well in mobile environments or over intermittent links.
- Each TCP Vegas connection attempts to keep a few packets in the network. When the estimation of the propagation delay is off, this could lead the connections to inadvertently keep many more packets in the network, causing a persistent congestion.
- The Vegas congestion avoidance algorithm intentionally lowers its transmission rate under heavy congestion. Thus, in head-to-head transfers, TCP-Reno steals bandwidth from Vegas. This is one possible reason why Vegas has not seen wide deployment in the Internet.

---

[1] Darrell E. Ernst, R. Durst: Space Communications Protocol Standards (SCPS) Bent-Pipe Experiment Report SCPS D71.51-Y-1, May 1966

[2] S. Horan and R. Wang, "Comparison of File Transfer Using SCPS FP and TCP/IP FTP over a Simulated Satellite Channel," Proc. International Telemetering Conference, 34, p. 87-95, October 1999.

[3] J. Muhonen, R. Durst: Space Communications Protocol Standards (SCPS) FY97 DOD Test Report, February 1998

[4] E. Edwards, C. Younghusband, C. Belise J. Peng, M. Phisel, and L. Hartman: An Independent Implementation of SCPS-TP with Simulation Extrapolation for Complex Networks, AIAA International Communications and Satellite Systems Conference and Exhibit 12-15 May, 2002 Montreal, Quebec, CA paper1912

[5] D. Tran, F. Lawas-Grodeck, R. Dimond, W. Ivancic: SCPS-TP, TCP and Rate-Based Protocol Evaluation, NASA TM-xxxxx, Anticipated publication date approximately August 2002

[6] http://www.xiphos.ca, July 2002

[7] http://www.skipware.com, July 2002

[8] ftp://ftp.ee.lbl.gov//tcpdump.tar.Z

[9] http://jarok.cs.ohiou.edu/software/

[10] M. Mathis, " The Macroscopic Behavior of the Congestion Avoidance Algorithm", Computer Communications Review, Volume 27, number 3, July 1997.

[11] Space Communications Protocol Specification (SCPS) – Transport Protocol (SCPS-TP), CCSDS 714.0-B-1, Blue Book, May 1999

[12] J. Mo, R. J. La, V. Anantharam, and J.Walrand: Analysis and Comparison of TCP Reno and Vegas, Infocom 1999

[13] J. Suk Ahn, P. B. Danzig, Z. Liu, L. Yan: Evaluation of TCP Vegas: Emulation and Experiment, SIGCOMM 1995