

## PERFORMANCE OF SCPS IN COMPLEX NETWORKS

Edwards, E., Younghusband, C., Fodil-Lemelin, K.  
Xiphos Technologies Inc.  
Suite 800, 3981 boul. St-Laurent, Montreal, QC, Canada H2W 1Y5  
[ece@xiphos.ca](mailto:ece@xiphos.ca)

Bélisle, C., Peng, J., Phisel, M.  
Communications Research Centre  
3701 Carling Ave., P.O. Box 11490, Station H, Ottawa, ON, Canada K2H 8S2  
[claud.belisle@crc.ca](mailto:claud.belisle@crc.ca)

Hartman, L.  
Canadian Space Agency  
6767 route de l'Aéroport, Saint-Hubert, QC, Canada J3Y 8Y9  
[leo.hartman@space.gc.ca](mailto:leo.hartman@space.gc.ca)

### **Abstract**

In this paper, a new, high fidelity, independent implementation of the Transport Protocol of the Space Communication Protocol Specification (SCPS-TP) is presented and the measured performance over a satellite link is compared with emulated and simulated results. The simulation model is then employed to extrapolate a complex, multi-user, hybrid network scenario and estimate the performance potential. These results are discussed and the behaviour is compared against contemporary, state-of-the-art terrestrial TCP stacks.

### **Introduction**

Satellite communication is rapidly gaining in popularity as a mean to deliver Internet traffic. It is seen as a competitive high-speed solution alternative to DSL or Internet via cable, in particular to connect remote communities, provide access to mobile assets and directly connect geographically dispersed enterprises. Communication with spacecraft is also moving toward IP-type communications, to leverage the wealth of proven IP tools, software, and infrastructure. This is leading to the creation of seamless end-to-end networks, extending from ground segment operators to spacecraft onboard subsystems.

It is well known, however, that the performance of the popular Internet transport protocol TCP, which ensure reliable data delivery, rapidly diminishes when traffic is carried over long delay and high bit-error rate links<sup>1</sup>. The main reasons relate to the window size, slow start and congestion control algorithms<sup>2</sup>. A number of proposals have been submitted to mitigate the effect of TCP on satellite networks<sup>3,4,5,6</sup>. However, these enhancements have not all been implemented into the most common commercial TCP stacks. NASA JPL and

the Consultative Committee on Space Data Systems (CCSDS) addressed these performance problems and standardized a solution: the Space Communications Protocol Standard – Transport Protocol (SCPS-TP)<sup>7</sup>. This standard recommends a set of changes to TCP to address performance problems while fully interoperating with Internet Protocol (IP). SCPS-TP is also now a MIL and ISO standard<sup>8,9</sup>.

An implementation of SCPS-TP was by done Xiphos Technologies, and is being distributed commercially as XIPLink. The Communications Research Centre (CRC) has implemented a simulation in OPNET, allowing for large-scale simulation of SCPS-enhanced communications prior to deployment.

In this paper the functionalities of the SCPS Transport Protocol (SCPS-TP) that address the limitations of TCP are reviewed. The test environment used to evaluate the performance of the XIPLink implementation and CRC simulation model is then described. The performance results are discussed and compared against contemporary, state-of-the-art terrestrial TCP stacks. The simulation is then employed to extrapolate complex, multi-user, hybrid network scenarios and estimate the performance potential.

### **SCPS-TP Functionalities**

SCPS-Transport Protocol (SCPS-TP) addresses the added complexities of the space environment with a number of different extensions and modifications to TCP. For satellite communications, the most significant are:

- Window scaling (RFC 1323);
- Round Trip Time Measurement (RFC 1323);
- Protection Against Wrapped Sequence Numbers (RFC 1323);

- Selective Negative Acknowledgment (SNACK) (adapted from RFC 1106);
- Header compression (adapted from RFC 1144);
- Low-loss congestion control (e.g. Vegas) or optional non-use of congestion control;
- Corruption Experienced (Default Source of Data Loss, DSDL)
- TCP Timestamps.

One other aspect of the SCPS-TP specification is equally important, but does not have direct bearing on performance: a SCPS-TP-enabled stack will default to conventional TCP when it connects to a non-SCPS host. As with IP compatibility, this gives a powerful advantage to SCPS over custom protocol solutions, allowing SCPS to simply replace existing TCP stacks. A SCPS host connecting to another SCPS host through a hostile environment can do so with a comparable level of performance as its connection to a wired, conventional Internet host, and do so transparently.

The SCPS-specific functions most significant to satellite communications are detailed below.

#### Header Compression

SCPS-TP incorporates TCP header compression that reduces the size of the TCP header by 50%. It works by summarizing those elements that remain static in a session and eliminating information that is not relevant to the respective segment being transmitted. This feature is useful when there is a limited amount of bandwidth available. This compression occurs at the transport layer rather than the data link level, to isolate header compression/expansion requirements to the source and destination endpoints.

#### Selective Negative Acknowledgement (SNACK)

SNACK is a negative acknowledgment and a request for immediate retransmission of specific data. While standard TCP has to wait for three duplicate ACKs to be received before initiating retransmission, the SNACK option causes the sender to immediately retransmit all segments necessary to fill the signaled holes. This is important since sufficient duplicate ACKs may never be received if multiple losses occur within a transmission window, or if the acknowledgment frequency is low over asymmetric channels. By using SNACK, the sender can retransmit lost segments more quickly, allowing more efficient use of the link.

#### Congestion Control

SCPS provides an option to enable or disable congestion control mechanisms. When congestion control is enabled, SCPS-TP has the choice of using either the standard TCP congestion control or TCP Vegas congestion control.

TCP Vegas estimates the available bandwidth without repeatedly incurring loss as standard TCP does. TCP Vegas does not continually increase the congestion window; instead, it tries to detect incipient congestion by comparing the measured throughput to the expected throughput. A large difference between the measured throughput and the expected throughput means network congestion is going to happen if proper action is not taken. If the difference is larger than a certain value, it decreases the transmission rate. If the difference is acceptable, TCP Vegas increases the transmission rate as usual. The net result is that the transmission rate of TCP Vegas varies around the optimum rate, rather than undergoing the dramatic increase and decreases of TCP.

#### Corruption Response

SCPS-TP can distinguish corruption losses from congestion losses. Evidence of corruption may be obtained from inter-layer signaling or from a management information query. When a connection receives an indication that corruption has been experienced, it sends the Corruption Experienced Option to its peer. Upon receiving Corruption Experienced Option, a connection then assumes that packet loss is due to corruption rather than congestion for two round trip times (RTT) or until it receives additional link state information. If a loss is caused by corruption, the connection merely retransmits the lost segment without reducing its transmission rate.

The SCPS-TP extensions and the environmental constraints they address are summarized very well in CCSDS introductory documentation.<sup>10</sup>

#### Test Environment

To validate the XIPLink SCPS-TP implementation, the CRC simulation model, and the veracity of laboratory emulation measurements, live satellite tests were conducted over ANIK E. In this section, the validation test environment is described.

#### Live Satellite

The XIPLink implementation was assessed via ANIK E satellite at the Communications Research Centre SATCOM Testbed in cooperation with Telesat Canada. The link rate was T1 on the forward channel and 256 kbps on the reverse channel. The round trip time was 525.0 ms, and the BER was varied. In order to reach BER as low as  $10^{-5}$  it was necessary to disable the Reed-Solomon error correction. Initial tests were conducted based on the BER reported by satellite modems, but early analysis suggested that the link quality was in fact much better. Links were then tested by Firebird BERT (Bit Error Rate Test) and

reconfigured according to the link quality reported by the BERT.

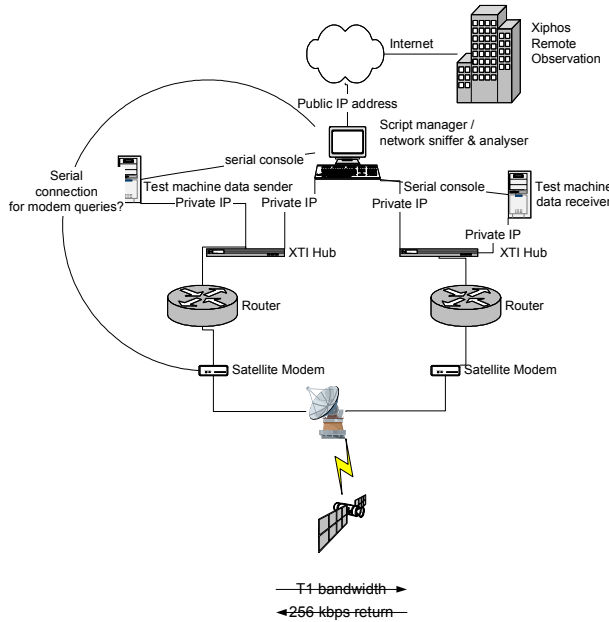


Figure 1: Satellite Test Configuration

Emulation

Performance tests on the XIPLink implementation were conducted at Xiphos Technologies, using the open source software application ‘Dummysnet’ running on FreeBSD. Dummysnet provided the capability to emulate properties of satellite links, such as a variable delay between end nodes and the ability to randomly drop data at a given BER. Dummysnet was also modified to simulate intermediate routers with buffering of 25KB. Further changes were made such that it could optionally corrupt data passing through it, in addition to capability of dropping bits at a given BER. This change allowed Corruption Experienced Option to be tested and the performance evaluated.

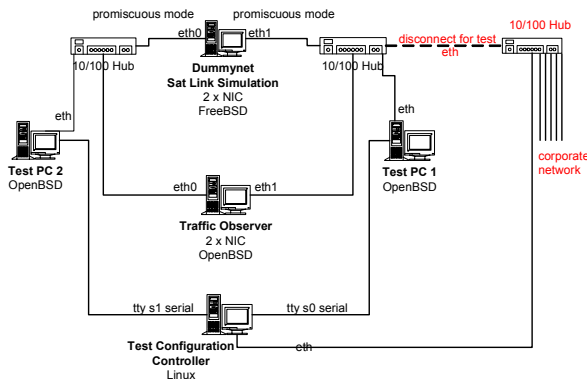


Figure 2: Emulation Configuration

Simulation

An OPNET environment was used to simulate the CRC model. The model, validated against the Live Satellite and Emulation trials, was used to simulate larger and more complex networks. Only some of the SCPS-TP mechanisms have been implemented in the OPNET model: Header Compression, and both short and long forms of SNACK. Therefore, only baseline TCP, SCPS-TP with Header Compression and SCPS-TP with Short SNACK and Header Compression were compared. Other SCPS-TP features such as Vegas Congestion Control will be implemented in the future.

Validation Test Cases

The test cases were selected to validate the functional behaviour of the code and simulation. No attempt was made to optimize the configuration of SCPS or tune for specific cases.

Emulation, simulation and satellite tests were conducted using a single socket connection sending a 3 MB of a data file, with random contents generated prior to the tests. Tests were run using scripts for automatic processing.

SCPS-TP was compared with a baseline that is representative of the default install of Windows 2000; this is consistent with the capabilities found in modern IP stacks. The following table gives some of the main characteristics affecting the network performance.

Table 1: Configuration parameters for simulation, emulation, and live satellite tests

TCP Parameters	Case 1A: Baseline (Windows 2000)	Case 1B: SCPS-TP Header Compression	Case 1C: Short SNACK and Header Compression
FR / FR	Enabled	Enabled	Enabled
SACK	Enabled	Enabled	Superseded
SNACK	N/A	No	Enabled
Header Compression	N/A	Yes	Yes
Maximum Window Size	64 kB	64 kB	241 kB
Initial Window Size	2 MSS (2920 B)	2 MSS	2 MSS
Window Scaling	Enabled	Enabled	Enabled
Initial RTO	3 sec	3 sec	3 sec

The baseline was used to validate the comparison process, prior to SCPS-TP case testing. It was necessary to ensure that tests were measured and compared in the same way; for example that the starting time of a data transfer was the instant SYN segments were sent, as opposed to full connection establishment.

The following additional comparisons were made between the emulation and satellite link environments, as they include SCPS-TP features not presently available in the OPNET simulation.

**Table 2: Configuration parameters for emulation and satellite (only) tests**

TCP Parameters	Case 2A: Vegas, Short SNACK and Header Compression	Case 2B: SNACK and Non-Use of Congestion Control
Fast Retransmit – Fast Recovery	No, Vegas Congestion Control	Disabled
SACK	Superseded	Superseded
SNACK	Yes	Yes
Header Compression	Yes	Yes
Maximum Window Size	241 kB	241 kB
Initial Window Size	8 MSS	80kB
Window Scaling	Enabled	Enabled
Initial RTO	3 sec	3 sec

### Complex Network Simulation

A larger scale environment was also modeled and simulated using OPNET. Thirty-five workstations, each generating a mixture of HTTP, FTP and e-mail were modeled. Each application had specific traffic models. In this paper, the results of the FTP application are reported. On average, each workstation was requesting files of 400 kB in size every 3 minutes. Using exponential distributions, 99% of the FTP file sizes ranged from 2 kB to 2 MB, with 63% less than the 400 KB and 40% within 200 KB to 600 KB range.

To ensure that results would be representative of satellite traffic and would not be biased by local cache transactions, the workstations were forced to request data from a server attached to another ground terminal. The satellite link attributes were varied to emulate different link qualities. The link quality is defined in terms of information bit error rate (BER) and ranges from  $10^{-5}$  to  $10^{-9}$ .

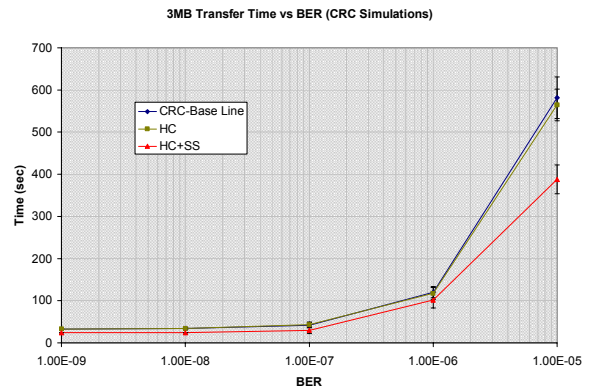
### Test Results

For each test case, two graphs, showing file download time versus link bit error rates, are provided. The first graph shows the full test range of BER, from  $10^{-9}$  up to  $10^{-5}$ . For clarity, the second curve shows only the range from  $10^{-9}$  up to  $10^{-7}$ . The error bars on each curve denote the standard deviation of the measured results. Table 3 gives a description of the graphs' legends.

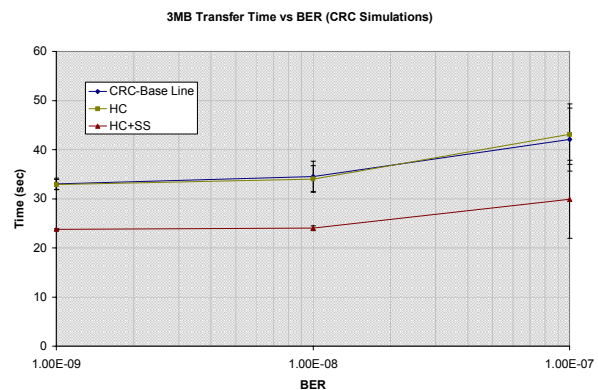
**Table 3: Description of Graph Legends**

HC,H	Header Compression
SS,S	Short SNACK
D	Default source of data loss (Corruption Experienced Option)
V	Vegas congestion control
N	Non-use of congestion control

### Simulation

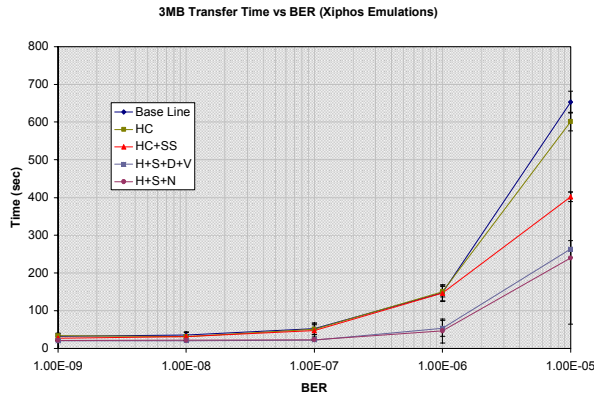


**Figure 3: Simulation results, full range of BER**

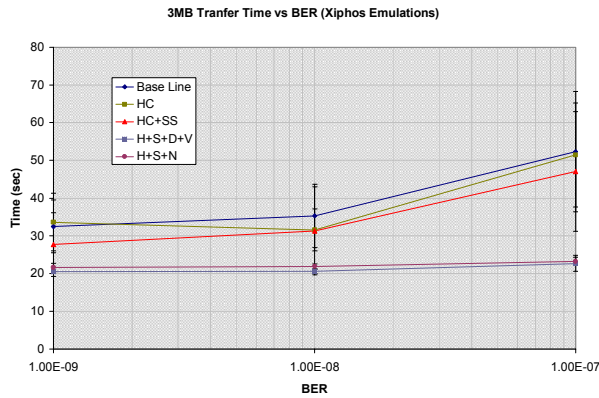


**Figure 4: Simulation results,  $10^{-9}$  to  $10^{-7}$  BER**

Emulation



**Figure 5: Emulation results, full range of BER**



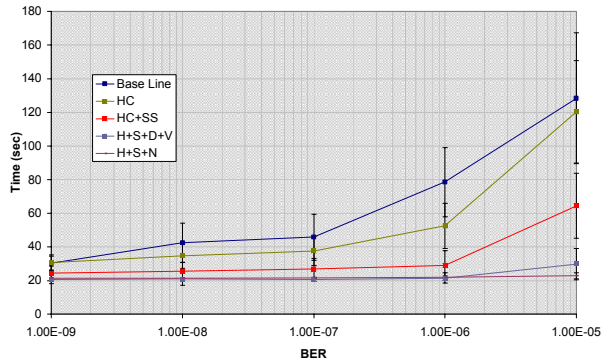
**Figure 6: Emulation results,  $10^{-9}$  to  $10^{-7}$  BER**

Live Satellite

For the satellite tests, it was necessary to calibrate the measured BERs against the effective BERs. This unfortunate complication came about as a result of the satellite modems, which employed Viterbi encoding on the data. The actual BER (as seen by the SCPS hosts) was measured directly, as previously mentioned, and thus accounted for any modifications to the data throughput performance caused by the modems. However, because the modem encoding rejected packets in blocks of 7 bits (in the event that they could not be corrected), the BERT was in fact seeing a block error rate that was one-seventh that of the bit error rate.

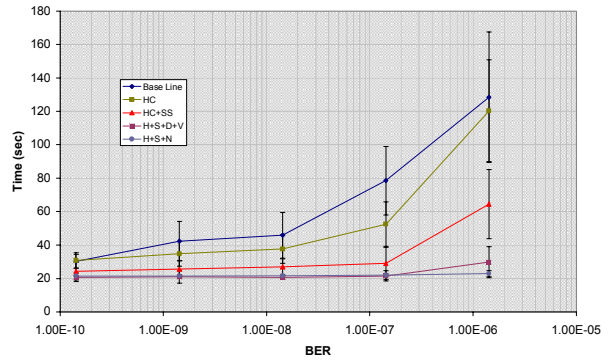
Because the blocks were contiguous and very small compared to the 1500-byte SCPS (or TCP) packets, they had essentially the same effect on the protocol stack as would a single lost bit. Thus, it was necessary to divide the measured BER by a factor of 7 to reflect its true effect on the transport layers under test.

Transfer Time vs BER (3MB) Uncorrected



**Figure 7: Live satellite results, uncorrected**

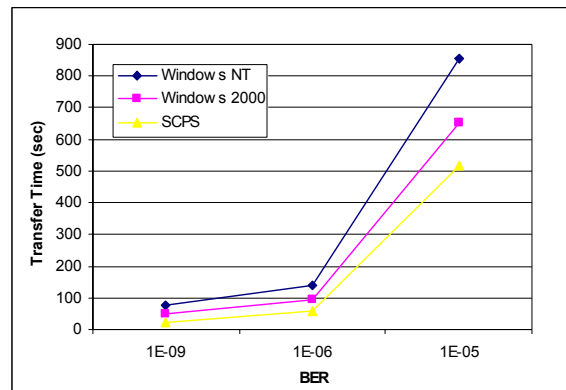
Transfer Time vs BER (3MB) Corrected



**Figure 8: Live satellite results, corrected**

Complex Network Simulation Results

Having confirmed the validity of the simulation, it was possible to generate results such as the following figure, which shows the average time required to download an FTP page for each of three TCP stack implementations over a 1.4 sec RTT, 680 kbps user data rate link for different information bit error rates.



**Figure 9: Average file transfer times for FTP**

## Discussion

These tests allow the direct comparison of three different methods for performance evaluation. They also permit general observations about the relative performance gains of the different SCPS-TP functionalities.

### Performance of SCPS-TP Features

In all cases, the relative merits of the SCPS-TP functionalities was as expected.

The smallest performance gain came from header compression. This is to be expected when the packet size is 1500 bytes, and the typical savings due to header compression is 12 bytes. The percentage gain here is small; the real strength of header compression is found when the MSS is reduced to achieve an optimal cross-section to bit errors. This optimization was not undertaken in these tests.

SNACK is a more powerful tool, and even at low BERs, provides a significant advantage in the satellite environment. This is due to its greater effectiveness in signaling for retransmissions in a delayed, high-BER environment.

Vegas provides considerable improvement, and should be considered seriously by any satellite network user.

NUCC was not tuned and it is known that significantly better performance can be reached (particularly at the higher BERs) by applying the algorithm more aggressively.

Since the modems dropped corrupted data, it was not possible to test the 'Corruption Experienced' feature of SCPS-TP.

### Correlation of Performance

In general, the fit between all three data sets is very good, and the cases match within one standard deviation of the data. This supports the validity of the simulation and emulation techniques for the reference cases under test.

### Capability for Extrapolation

Extrapolating the simulation, from the reference cases to complex environments, is the next natural step in this capability, made possible through the validation process.

In the example case, it can be clearly seen that a SCPS-TP stack is superior to the Windows 2000 TCP stack, which is itself better than the Windows NT stack. The SCPS sack is only using the SNACK, Header

Compression, and RFC 1323 options. The Windows 2000 TCP stack is showing a 30% improvement over the Windows NT implementation; the SCPS stack is demonstrating an improvement in download time of 60% over Windows NT and 40% over Windows 2000.

### Sources of Variation in the Data

Transport layer stacks do not tend to produce repeatable results in the presence of random errors, and require a number of repetitions to achieve statistically significant results. As seen by the error bars, the variation in the data increases with BER.

#### *Variation due to Heterogeneity*

Flow control schemes are impacted due to the heterogeneous nature of a file transport history: it begins with handshake, moves into slow start, transitions to a steady-state congestion control method, and then closes the connection. Depending on the timing of a bit error, the effect varies dramatically. For example, the loss of a bit during slow start can cause a transition into congestion control at a very low throughput, which will then leave a long "uphill" climb to achieve an equilibrium throughput. This will impact both Vegas and traditional congestion control. Data loss during the initial handshake or closing of the connection can cause a coarse-grained timeout, resulting in a lengthy pause.

#### *Variation due to Non-Optimization*

In the case of NUCC, the error bars become very large at high BER. This was due to a limitation in the code that prevented the algorithm from maintaining sending speed in the presence of BERs sufficiently large to cause the loss of the same packet on 2 subsequent retransmits. This only becomes a statistical possibility at high BER, and has since been optimized from the code.

### Sources of Inaccuracy in the Data

The greatest source of inaccuracy in the data is due to the limited control over BER during the satellite tests: while the simulator and emulators have statistically near-perfect random loss patterns, the satellite link would vary above and below the desired bit error rates, although every effort was made to keep the BER as desired for a given test. High BERs were difficult to stabilize and would drift considerably; the power levels required to create a BER of  $10^{-5}$  were so low that minor weather changes would have a measurable effect on the performance. The steep slope of the performance curve at high BER exacerbated this sensitivity.

### Experimental Considerations

The execution of live satellite tests always adds real-world complexity; in these tests, the measurement and control of BER proved to be critical.

## **Conclusions**

The XIPLink implementation of SCPS-TP is a very powerful performance enhancer in environments where the link may be subject to high bit error rates. The satellite tests have demonstrated that the emulator is an excellent representation of a real satellite link. Given the cost and complexity of live satellite testing, such a tool has considerable value for testing changes and tuning the performance of code.

Likewise, the tests have confirmed that the CRC simulation is a valid and powerful tool for network design, protocol development and tuning, and a wide spectrum of other analytical tasks. Extrapolation of this tool to complex environments provides a method of dealing with scenarios that are beyond the reach of practical satellite tests or even hardware emulations. The satellite tests have also demonstrated that extension of this simulation to include other SCPS-TP functions will be of value in identifying optimal configurations for satellite operations.

## **Acknowledgements**

The authors would like to thank the Canadian Space Agency, the CRC Satcom Testbed team, and Telesat Canada for their support in this investigation.

## **References**

- <sup>1</sup> C. Partridge, T.J. Shepard, "TCP/IP Performance over Satellite Links", IEEE Network, Sept.-Oct. 1997, pp.44-49.
- <sup>2</sup> P. Andreadis, J. Peng, "TCP over Satellite – A review of TCP over Satellite Links: Problems and Suggested Solutions", CRC Technical Note no. CRC-TN-2001-01, April 2001.
- <sup>3</sup> M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", RFC 2488, January 1999.
- <sup>4</sup> M. Allman, et al., Ongoing TCP Research Related to Satellites, RFC 2760, February 2000.
- <sup>5</sup> Allman, M., Paxson, V., Stevens, W., TCP Congestion Control, RFC 2581, January 1999.
- <sup>6</sup> J. Peng, P. Andreadis, C. Bélisle, and M. Barbeau, "Improving TCP Performance over Long Delay Satellite Links", OPNETWORK 2001, August 27<sup>th</sup> – August 31<sup>st</sup>, Washington D.C.
- <sup>7</sup> Consultative Committee for Space Data Systems, Space Communications Protocol Specification (SCPS): Transport Protocol, CCSDS-714.0-B-1. ([www.ccsds.org](http://www.ccsds.org))

---

<sup>8</sup> Department of Defense Interface Standard, Transport Protocol for High-Stress, Resource-Constrained Environments, MIL-STD-2045-44000.

<sup>9</sup> ISO 15893:2000 Space data and information transfer systems -- Protocol specification for space communications -- Transport Protocol

<sup>10</sup> Consultative Committee for Space Data Systems, Space Communications Protocol Specification (SCPS): Rationale, Requirements, and Application Notes, CCSDS draft Green Book (CCSDS 710.0-G-0.4), Table 6-3.