MTR 96W0000075V01

MITRE TECHNICAL REPORT

Final Report of the SCPS-TP Testing on the UK DRA STRV

September 1996

Robert C. Durst
Darrell E. Ernst
Gregory J. Miller
Eric J. Travis

Sponsor:	USSPACECOM/J4 HQ NASA/CODE OI	**Contract No.:**	F-19628-94-C-0001 DAAB0796-C-E601
Dept. No.:	D048 W159	**Project No.:**	6380 8S06

The views, opinions, and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

Approved for public release; distribution unlimited.

Washington C³ Center
McLean, Virginia

Project 6380 Approval:

MITRE Department Approval: original signed by
Gary M. Comparetto
Associate Department Head

MITRE Project Approval: original signed by
Gary M. Comparetto
Associate Project Leader

Project 8S06 Approval:

MITRE Department Approval: original signed by
Duncan Thomson
Technical Area Manager

MITRE Project Approval: original signed by
Randolph C. Mitchell
Project Leader

Abstract

This paper documents the results of the SCPS Transport Protocol (SCPS-TP) portion of the SCPS/STRV Flight Experiment (SSFE). This experiment involved hosting the SCPS-TP protocol on the STRV 1b spacecraft and testing the operation of the protocol between the space-based endpoint and the ground. The communication environment imposed round-trip delays of approximately 8 seconds, error rates from 0 to > 10^{-4}, and very low speed links (1000 bps from space to ground, and 125 bps from ground to space). The experiment examined the effects on throughput, link utilization, and bit-efficiency of the TCP Timestamps capability, a Selective Negative Acknowledgment (SNACK) capability, and an end-to-end Header Compression capability. The paper presents the results of a factorial experiment conducted in a laboratory environment configured to simulate the flight test environment, then presents the results of those configurations from the factorial experiment that were tested in the flight environment. The experiment results show that SNACK and Header Compression greatly improve throughput, while the TCP Timestamps capability reduces throughput.

KEYWORDS: SCPS STRV TCP

Executive Summary

Purpose of This Document

The Space Communications Protocol Standards - Transport Protocol (SCPS-TP) is being developed by the joint NASA/DOD Space Communications Protocol project. This report documents the findings of the SCPS/STRV Flight Experiment (SSFE) SCPS-TP test.

Background

In the fall of 1992, NASA and the DOD jointly established a technical team (the SCPS Technical Working Group, or "SCPS-TWG") to explore possibilities for developing common space data communications standards. By the end of 1993 the team concluded that wide segments of the U.S. civil and military space communities have common needs for protocols to support in-flight monitoring and control of civil and military spacecraft. In 1994, the U.K. Defence Research Agency joined the SCPS-TWG with specific interoperability interests for the U.K. Skynet series of military communications satellites.

The program of work to develop these protocols includes specification, simulation, implementation, and testing. The SCPS/STRV Flight Experiment is the latest in a series of tests, that has included simulation, laboratory testing, and a bent-pipe test over a satellite link. The SCPS/STRV Flight Experiment was the first test to actually host the prototype software on a spacecraft, and was intended to evaluate performance and functionality in the anticipated implementation and operational environments.

The protocols tested in the SSFE include the SCPS File Protocol, the SCPS Transport Protocol, and the SCPS Security Protocol. All of the SCPS File Protocol testing made use of the SCPS Transport Protocol, and the SCPS Security Protocol testing used the SCPS Transport Protocol as its data source. The tests of the SCPS File Protocol and SCPS Security Protocol are documented separately (reference [14], [15]).

The SSFE was conducted between 2 January 1996 and 30 April 1996 and between 16 July 1996 and 31 July 1996. The SCPS-TP tests were conducted by U.K. Defence Research Agency personnel stationed at Lasham, England and at Farnborough, England, and by MITRE and Gemini Industries personnel at Reston, Virginia. The tests were conducted at Lasham, England and Reston, Virginia.

SSFE SCPS-TP Test Objectives

The objectives of the transport protocol portion of the SSFE were as follows:

- to gain experience in hosting SCPS-TP on an actual spacecraft and

- to examine the performance of SCPS-TP when running over a real space/ground data link.

In examining the performance of SCPS-TP, we tested three specific capabilities. The following list cites the primary benefits expected from each of the capabilities:

- TCP Timestamps: this capability improves SCPS-TP's estimate of round trip time, which can become distorted in error-prone environments.
- SCPS-TP Header Compression: this capability reduces protocol overhead by reducing the size of SCPS-TP headers.
- Selective Negative Acknowledgment: this capability improves SCPS-TP's error response by providing detailed information about missing or corrupted data.

We wished to determine the extent to which each of these capabilities affected performance at various bit-error rates. We also wished to determine if there were any significant interactions between the options that would restrict the ability of a user or program to pick the options individually.

We met the objectives stated above. The process of hosting the SCPS-TP protocol onto the STRV was a difficult one, primarily due to the limited availability of C-language development tools for the MIL-STD-1750A processor. The generally poor quality of development tools delayed our discovery and correction of two implementation errors. This rendered invalid the results of the first set of tests that we conducted. We were able to conduct a limited amount of retesting, which was used to confirm the results we gathered in the laboratory.

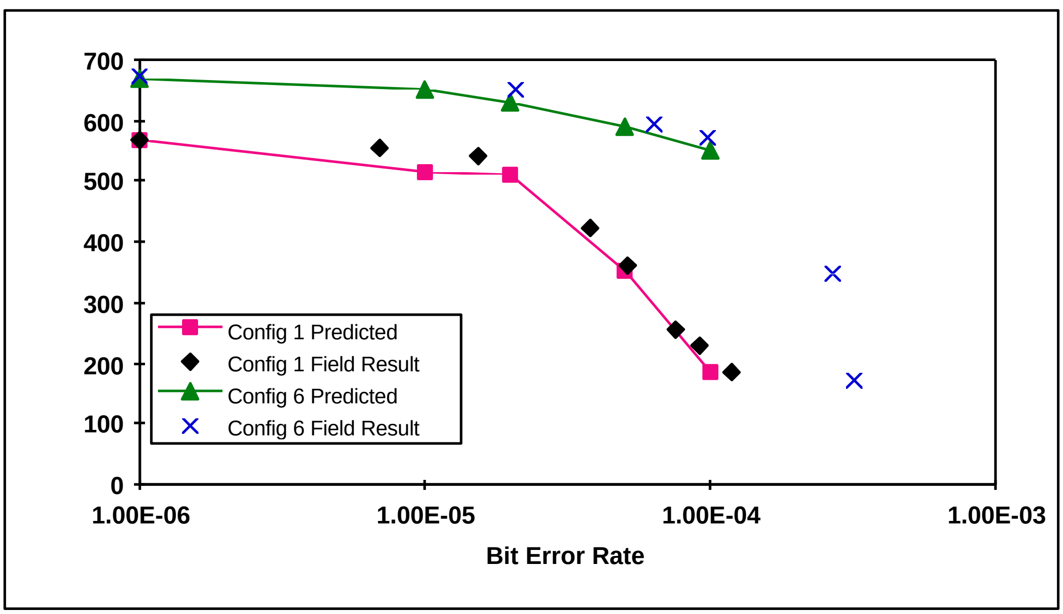
Summary of Results

The experiment was conducted in three phases: we performed initial testing in the field, then we tested the protocols extensively in the laboratory, then we performed final testing in the field. The field results presented in this report reflect the results of the final field testing.

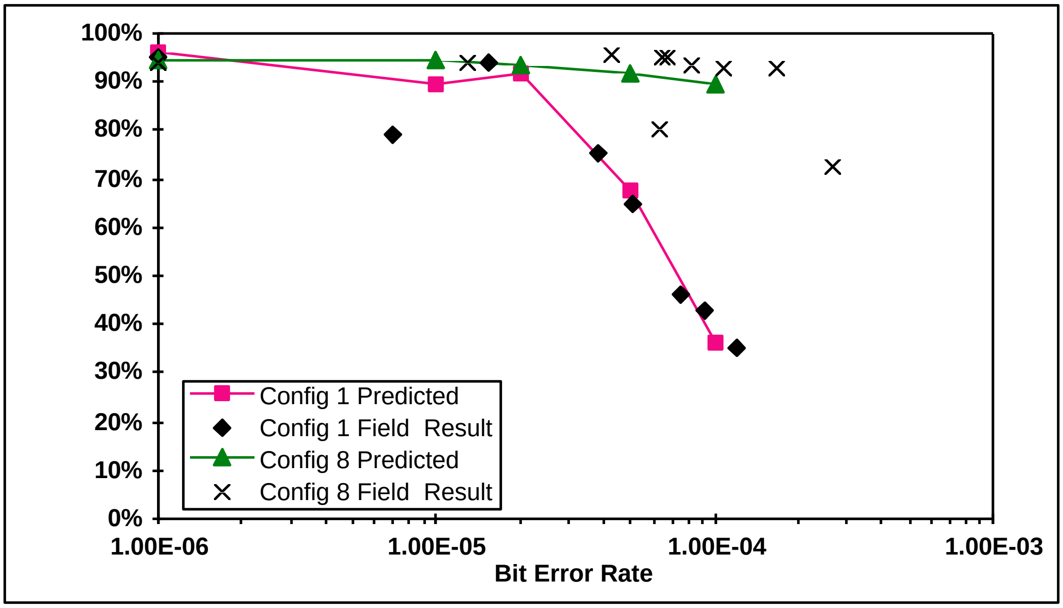
We made the following performance measurements in all tests: throughput, link utilization, and bit-efficiency. Throughput is a measure of the average rate at which the protocol can move user data, and is one of the most commonly used measurements of communication protocol performance. Link utilization is a measure of the ability of the protocol to "keep the pipe full." This ability is important in space communication, in which contact times may be limited. The protocol should not allow the link to be idle when data is waiting to be transmitted. Finally, bit-efficiency is a measure of the amount of protocol overhead required to transfer a user's data. The overhead includes protocol headers, acknowledgment traffic, and any retransmissions required to get the user data to its destination. Bit efficiency is important in spacecraft communications, because link capacity is generally a scarce resource.

Eight protocol configurations were tested in the laboratory, and of those eight, five were tested in the field. The following graphs briefly summarize the field test results, with predictions based on the laboratory results shown for reference. The three graphs correspond to the three types of performance measures made: throughput, link utilization, and bit-efficiency. Each graph presents the results of the protocol configurations that performed the best and the worst in the field. The laboratory predictions are based on the mean response resulting from 10 tests at each of the following bit-error rates: 10^{-6}, 10^{-5}, $2x10^{-5}$, $5x10^{-5}$, 10^{-4}.

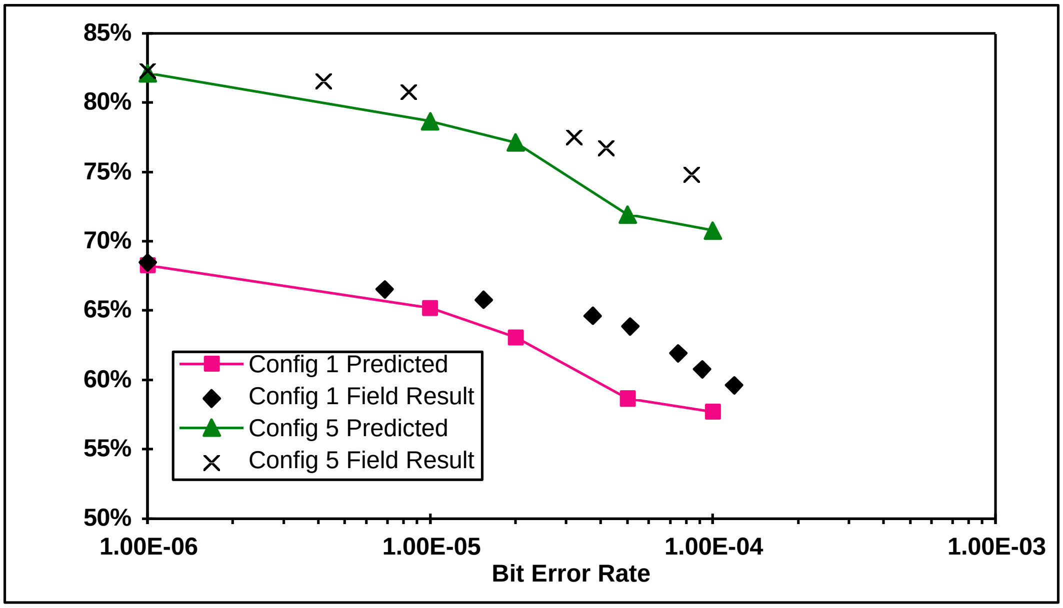
The first graph presents the throughput results. Readers should bear in mind that the maximum possible throughput of a SCPS-TP connection is 768 bps, not including SCPS-TP protocol overhead. The graph shows that the best throughput was obtained by the configuration (Configuration 6) that enabled the Selective Negative Acknowledgment and SCPS-TP Header Compression capabilities, described above. The poorest throughput in the field resulted from the configuration (Configuration 1) that had none of the SCPS-TP capabilities enabled. (Note that the laboratory results indicate that the configuration that enabled TCP Timestamps and none of the other capabilities would have shown lower throughput than Configuration 1, but this configuration was not tested in the field.)



The next graph presents link utilization results. The graph shows that the best link utilization was obtained by the configuration (Configuration 8) that enabled all of the capabilities under test. The poorest link utilization in the field resulted from the configuration (Configuration 1) that had none of the SCPS-TP capabilities enabled.



The final graph presents bit-efficiency results. The graph shows that the best bit-efficiency was obtained by the configuration (Configuration 5) that enabled SCPS-TP Header Compression. The poorest bit-efficiency in the field resulted from the configuration (Configuration 1) that had none of the SCPS-TP capabilities enabled. However, as with the throughput tests, the configuration with TCP Timestamps (only) enabled had worse bit-efficiency than Configuration 1 in the laboratory tests, but was not tested in the field. Note that the bit-efficiency results from the field tests tend to be higher than the results from the laboratory tests of corresponding configurations. This is due to an inherent difference between the field test environment and the laboratory test environment.



Conclusions

The SCPS-TP protocol appears to be well-suited to the long-delay, potentially high bit-error rate environment of the STRV. All configurations were able to sustain connections at bit-error rates of 10^{-4} and yield throughput in excess of 130 bps (17% of maximum possible). The Selective Negative Acknowledgment (SNACK) capability was principally responsible for the ability of the SCPS-TP to operate well in high bit-error rate environments (the configuration with the SNACK capability enabled showed only a 15% drop from maximum throughput at a bit-error rate of approximately 10^{-4}). The SCPS-TP Header Compression accounted for an 18% increase in throughput over the configurations that did not use Header Compression at zero bit error rate.

The following conclusions derive from the laboratory testing and are confirmed by the flight test results:

1. The SNACK capability significantly improves throughput at high bit-error rates, and has no negative effects on throughput at low bit-error rates.
2. The TCP Timestamps capability has a negative effect on throughput at low bit-error rates. It has a strongly negative effect on bit-efficiency. When used in combination with SNACK, throughput is lower than when using SNACK alone. (The magnitude of the negative effect of TCP Timestamps on throughput is exaggerated by the small packet size imposed by the STRV. With larger packet sizes, this effect is mitigated.)
3. The SCPS-TP Header Compression capability has a significant, positive effect on throughput at bit-error rates of $5x10^{-5}$ and below. Header Compression improves bit-efficiency at all bit-error rates. (The positive effect of Header Compression on

throughput is exaggerated by the small packet size imposed by the STRV in the same manner that the negative effect of the TCP Timestamps is, above. As with TCP Timestamps, the effect of Header Compression on throughput will diminish as the packet size increases.)

The following conclusions are supported by the laboratory testing, but were neither confirmed nor refuted by the flight test results:

1. The SNACK capability significantly improves link utilization at high bit-error rates, has no negative effects on link utilization at low bit-error rates, and has no impact on bit-efficiency.
2. The TCP Timestamps capability has a moderately positive effect on link utilization. When used in combination with SNACK, link utilization is improved slightly.
3. The SCPS-TP Header Compression capability has no effect on link utilization.

Recommendations

We document recommendations primarily directed at ourselves in Appendix C, Lessons Learned. The following recommendations are directed toward potential users of SCPS-TP and toward the sponsors of this effort.

1. Push ahead in the effort to standardize SCPS-TP and deploy it in environments that have similar delay and error characteristics to the STRV environment.
2. When using SCPS-TP in STRV-like environments, enable SNACK.

   SNACK has no negative effects when errors are not present, and is primarily responsible for the protocol's ability to sustain relatively high throughputs at high bit-error rates.
3. When using SCPS-TP in STRV-like environments, enable Header Compression.

   The Header Compression capability reduced the size of SCPS-TP headers, improving throughput and bit-efficiency. These effects were particularly dramatic because the maximum packet size of the STRV was small. As the packet size increases, the positive effect of Header Compression will diminish.
4. When using SCPS-TP in STRV-like environments, disable TCP Timestamps.

   The TCP Timestamps capability reduced throughput at low bit-error rates, and provided no significant improvement in throughput at high bit-error rates when SNACK was in use. As with Header Compression, the negative effects of TCP Timestamps are exaggerated by the small packet sizes on STRV.
5. Evolve the program of testing toward integrated tests.

Although there are still specific SCPS-TP capabilities to be tested, the focus of future tests should be integrated-stack testing. Tests of individual protocol capabilities can be conducted either as part of integrated-stack testing or as a small, focused portion of a larger test. The SCPS-NP, which has not as yet undergone flight testing, will probably benefit from more substantial, focused testing. However, this can still be conducted in the context of an overall test.

Foreword

This is Volume 1 of the Final Report of the SCPS-TP Testing on the UK DRA STRV. It contains the body of the report. The Appendixes appear in Volume 2, which is printed under separate cover.

Acknowledgments

The authors express their gratitude to the Defence Research Agency (DRA) team that helped make this test possible. The (extended) DRA team included Chris Caines, John Eves, Duncan Fortune, Paul Healey, Trevor Roberts, Nick Shave, LT CDR Kevin Shaw, David Stanton, Darren Vincent, John Weaver, and Nigel Wells.

The authors also express their gratitude to the project sponsors for their continuing support, assistance, and encouragement.

I (Robert Durst) thank my wife for enduring my long hours and special moods during this project.

Table of Contents

Section	Page
Introduction	**1**
Objectives	**3**
Design	**5**
Configuration	**7**
4.1 Laboratory Configuration	7
4.1.1 Equipment Configuration	7
4.1.2 Protocol Configuration	8
4.2 Field Configuration	10
4.2.1 Equipment Configuration	10
4.2.2 Protocol Configuration	13
Methods	**15**
5.1 Laboratory Testing	15
5.1.1 Test Execution	15
5.1.2 Data Collected	17
5.1.3 Data Reduction	17
5.2 Field Testing	17
5.2.1 Test Execution	18
5.2.2 Error Introduction and Measurement	19
5.2.3 Data Collected	20
5.2.4 Data Reduction	20
Results	**23**
6.1 Laboratory Experiment Results	23
6.1.1 Throughput Results	23
6.1.2 Link Utilization Results	32
6.1.3 Bit-Efficiency Results	36
6.1.4 Conclusions Based on Laboratory Testing	40
6.2 Field Experiment Results	40
6.2.1 Predicted Results Based on Laboratory Tests	41
6.2.2 Field Results	41
Conclusions and Recommendations	**65**

Section	Page
7.1 Conclusions	65
7.2 Recommendations	66
List of References	**69**
Glossary	**73**
Distribution List	**77**

List of Figures

Figure		Page
1	SSFE Laboratory Test Configuration	8
2	SSFE Onboard Software Configuration	11
3	SSFE Ground System Configuration	12
4	Throughput Performance of SCPS-TP Configurations 1 and 2	27
5	Throughput Performance of SCPS-TP Configurations 3 and 4	27
6	Throughput Performance of SCPS-TP Configurations 5 and 6	28
7	Throughput Performance of SCPS-TP Configurations 7 and 8	29
8	Throughput Performance of SCPS-TP Configurations 1 and 5	30
9	Effect of Timestamps on Throughput (Without Header Compression)	31
10	Effect of Timestamps on Throughput (With Header Compression)	31
11	Effect of the Interaction Between Timestamps and SNACK on Throughput	32
12	Link Utilization of SCPS-TP Configurations 1 and 2	33
13	Link Utilization of SCPS-TP Configurations 3 and 4	34
14	Link Utilization of SCPS-TP Configurations 5 and 6	35
15	Link Utilization of SCPS-TP Configurations 7 and 8	35
16	Effect of the Interaction Between SNACK and Timestamps on Link Utilization	36
17	Bit Efficiency of SCPS-TP Configurations 1 and 2	38
18	Bit Efficiency of SCPS-TP Configurations 3 and 4	38
19	Bit Efficiency of SCPS-TP Configurations 5 and 6	39
20	Bit Efficiency of SCPS-TP Configurations 7 and 8	39

Figure		Page
21	Predicted Throughput for Configuration 8	41
22	Predicted Versus Measured Throughput for Configuration 1	42
23	Predicted Versus Measured Throughput for Configuration 5	43
24	Predicted Versus Measured Throughput for Configuration 6	45
25	Predicted Versus Measured Throughput for Configuration 7	47
26	Predicted Versus Measured Throughput for Configuration 8	48
27	Comparison of Throughput Results for Configuration 5 and Configuration 6	50
28	Comparison of Throughput Results for Configuration 6 and Configuration 8	51
29	Predicted Versus Measured Link Utilization for Configuration 1	52
30	Predicted Versus Measured Link Utilization for Configuration 5	54
31	Predicted Versus Measured Link Utilization for Configuration 6	55
32	Predicted Versus Measured Link Utilization for Configuration 7	56
33	Predicted Versus Measured Link Utilization for Configuration 8	57
34	Link Utilization of Configuration 5 and Configuration 7	58
35	Link Utilization of Configuration 1 and Configuration 5	59
36	Predicted Versus Measured Bit Efficiency for Configuration 1	60
37	Predicted Versus Measured Bit Efficiency for Configuration 5	61
38	Predicted Versus Measured Bit Efficiency for Configuration 6	61
39	Predicted Versus Measured Bit Efficiency for Configuration 7	62
40	Predicted Versus Measured Bit Efficiency for Configuration 8	62

List of Tables

Table		Page
1	SCPS-TP Configurations for Laboratory Testing	9
2	SCPS-TP Parameter Settings for Laboratory Experiments	9
3	SCPS-TP Configurations for Flight Testing	13
4	Post-Processed Data Items Recorded for Each Laboratory Test	18
5	Post-Processed Data Items Recorded for Each Field Test	21
6	Allocation of Variation in Throughput Performance	25
7	Allocation of Variation in Link Utilization	33
8	Allocation of Variation in Bit Efficiency	37

Section 1

Introduction

In the fall of 1992, NASA and the DOD jointly established a technical team (the SCPS Technical Working Group, or “SCPS-TWG”) to explore possibilities for developing common space data communications standards, with a principal focus on the activities associated with in-flight monitoring and control of civil and military spacecraft. In practical terms, these activities involve a ground control center conducting a dialog with a remote spacecraft to transmit telecommands, to up-load and verify onboard software loads, and to confirm correct spacecraft performance via a flow of telemetry.

The team adopted a two-pronged approach in its study phase: part of the team conducted a top-down survey of representative civil and military space data communications requirements, while the remainder of the team conducted a bottom-up analysis of available standard data communications protocols. The team compared the results to see how capabilities matched requirements, and formulated recommendations for future work. In evaluating existing capabilities, first priority was given to commercially-supported “off the shelf” standards. However, recognizing unique requirements of the space mission environment (long propagation delays, noise-induced errors, and limited spacecraft data processing resources and communications capacity), the team also considered other options. By the end of 1993 the team concluded that wide segments of the U.S. civil and military space communities have common needs for:

- An efficient file handling protocol, capable of supporting file transfers initiated either from ground-based systems or space-based systems
- A data transport protocol that provides the user with selectable levels of reliability, based on operational need, between computers that are communicating over a network containing one or more space data transmission paths
- Optional data protection mechanisms to assure the end-to-end security and integrity of such message exchange
- An efficient protocol to support connectionless routing of messages through networks containing space data links.

Following the study phase, the SCPS-TWG began development of four specifications, one for each of the protocols, that address the above requirements: the SCPS File Protocol (SCPS-FP), the SCPS Transport Protocol (SCPS-TP), the SCPS Security Protocol (SCPS-SP), and the SCPS Network Protocol (SCPS-NP). These draft specifications have been submitted to the appropriate DOD authority for adoption as military standards, and to the appropriate international body for consideration of adoption as international standards. At the completion of these standards activities, resulting standards may then be adopted by any military, civil, or commercial organization for use in any space system. It is the intent of NASA and DOD that commercial vendors produce the SCPS protocols as widely-distributed commercial products, thus helping to reduce the cost of space systems while increasing their interoperability.

Part of the protocol development includes a rigorous test program. Previously, the SCPS-TP was tested in a "bent-pipe" environment, in which two ground-based workstations communicated via a satellite link [13]. One of the key elements of the test program is a "protoflight" test, in which the SCPS protocols are flown as experimental payloads on one or more satellites. The first component of this protoflight phase of testing involved testing the SCPS protocols onboard a Space Technology Research Vehicle (STRV) operated by the UK Defence Research Agency (DRA). This test, the SCPS-STRV Flight Experiment (SSFE), included tests of the SCPS File Protocol (SCPS-FP), the SCPS Transport Protocol (SCPS-TP), and the SCPS Security Protocol (SCPS-SP). This report documents the results of the initial protoflight testing of the SCPS Transport Protocol, which is an extension of the testing documented in [13].

This document is organized into seven sections and five appendixes. Section 1 is this Introduction. Section 2 puts forth the experiment objectives. Section 3 describes the experiment plan and Section 4 describes the experiment configuration. Section 5 documents the experiment methods that were used to conduct the experiments and gather the data. Section 6 presents the experiment results, and Section 7 presents conclusions and recommendations. Appendix A documents the equations that are unique to this paper that were used in the data analysis. Appendix B contains the data collected during the experiment. Appendix C documents some of the lessons learned during the experiment. Appendix D presents an overview of the SCPS-TP protocol, and Appendix E briefly describes the SCPS-TP implementation.

Section 2

Objectives

The objectives of the transport protocol portion of the SSFE were as follows:

- to gain experience in hosting the SCPS-TP on an actual spacecraft and
- to examine the performance of the SCPS-TP when running over a real space/ground data link.

The first objective was achieved - the experience of hosting the SCPS protocols on a resource-constrained platform led to modifications in several implementation approaches. The protocol *specifications* appear to require no significant modification as a result of the testing. The lessons learned from conducting the experiment and porting the SCPS-TP protocol to the STRV platform are documented in Appendix C of this document.

The second objective (to examine the performance of the SCPS protocols in the STRV flight environment) was also met with the following caveat: some performance results derive from testing over an actual space-ground link and some from testing over a simulated link. We collected laboratory data that thoroughly demonstrates specific aspects of the behavior of the SCPS transport protocol in a simulation of the STRV flight environment. During the course of the flight test we identified and corrected two coding errors in the software that significantly affected performance. We retested the protocols and found good correspondence between the laboratory and the field data, but did not have enough time during the retest to take as much data as originally planned. Nevertheless, experiments that we ran onboard the STRV confirm, to a large degree, the conclusions that were based on the data taken in the laboratory. For some conclusions, we were unable to confirm the laboratory conclusion, but saw no evidence that would refute those conclusions. In our discussion of the field experiment results, we discuss how the data from each test relates to the corresponding data taken in the laboratory.

The performance measures were throughput, link utilization, and bit-efficiency. Throughput measures the average rate at which the protocol can move user data, and is one of the most commonly used measurements of communication protocol performance. Link utilization is a measure of the ability of the protocol to “keep the pipe full” when there is data ready to be transmitted. This ability is important in space communication, in which contact times may be limited. The protocol should not allow the link to be idle for significant periods of time. Finally, bit-efficiency is a measure of the amount of protocol overhead required to move a user’s data. The overhead includes protocol headers, acknowledgment traffic, and any retransmissions required to get the user data to its destination. Bit efficiency is important in spacecraft communications, because link capacity is generally a scarce resource.

Section 3

Design

The SCPS-TP SSFE design consisted of five $2^k r$ factorial experiments, one at each of five bit-error rates. (The $2^k r$ terminology is explained in detail in reference [11]. In brief, under a $2^k r$ factorial experiment design, there are k factors that can each take on one of two discrete levels, yielding a total of 2^k experiments. Each of the 2^k experiments is replicated r times.) This type of experiment design allows us to examine the effect that each factor contributes to performance through a technique known as allocation of variation. Since we use multiple replications, we can isolate the effects of experimental error, as well.

Our factorial experiment, conducted in the laboratory, was configured with $k = 3$ and $r = 10$, meaning that we tested three different factors, and replicated each test ten times. The three factors of interest to us were three different SCPS-TP capabilities: TCP Timestamps, SCPS Header Compression, and SCPS SNACK.

The TCP Timestamps capability, defined in [10], provides two primary benefits: it improves TCP's estimate of round-trip time, which can become distorted in error-prone environments; and it serves as an extension to the sequence number space for very high-rate applications. (It is in the ability to improve the estimate of round trip time that we are interested.) The SCPS Header Compression capability replaces or omits invariant fields in the SCPS-TP headers, to reduce protocol overhead. The SCPS SNACK capability improves the protocol's response to errors. Refer to Appendix D for a more detailed description of each of these capabilities.

We wished to determine the extent to which each of these capabilities affected performance at various bit-error rates. We also wished to determine if there were any significant interactions between the options that would restrict the ability of a user or program to pick the options individually. Each test consisted of a bulk-data transfer from the spacecraft to the ground. The size of each transfer was approximately 50000 bytes of user data (the amount of data per transfer was rounded to a multiple of the amount of user data per packet).

Since we repeated the laboratory experiments ten times at five different bit-error rates, we conducted a total of 400 tests ($2^3 * 10 * 5 = 400$). Each test took between 10 and 45 minutes to execute. For each test, we measured throughput, link utilization, and bit-efficiency. (Appendix B contains the experimental data for the laboratory experiments and the field experiments. Appendix A contains the equations used to calculate any non-directly-measured results.)

We did not attempt to fully replicate the laboratory experiment in the field. The laboratory experiment design depends on our ability to control the bit-error rate of the test. This was possible in the laboratory environment, but not in the field. In addition to our inability to control the bit-error rate on the link, there was simply not time to collect the volume of data necessary to replicate the results. Rather, we decided to take data for selected configurations over a broad range of bit-error rates with the intent of confirming that the laboratory data was a valid representation of the protocol in the field. We calculated predicted responses for each of the measurements of interest (throughput, link utilization, and bit-efficiency) over the range of bit-error rates that we tested in the laboratory, and compared these predictions to the results obtained from the flight-tests.

Section 4

Configuration

We conducted the SSFE SCPS-TP experiments in the laboratory, using the SCPS test bed, and in the field, using equipment largely provided by the UK Defence Research Agency in Lasham, England. This section describes the experiments conducted in each of those two locations.

4.1 Laboratory Configuration

4.1.1 Equipment Configuration

We tested SCPS-TP in the laboratory, using a test configuration that simulated, to the extent practicable, the delays, data rates, and error rates of the field environment. For STRV development, DRA lent us a space-qualified MIL-STD-1750A onboard computer (OBC) equipped with a RS-232 interface configured to transmit and receive at 9600 bits per second (bps). The OBC has the identical flight read-only memories (ROMs) that the STRV 1b uses, but it did not have any of the CCSDS hardware to provide telemetry framing or uplink processing.[1]

In Figure 1, the OBC is the space-qualified processor mentioned above. The OBC Relay is a Sun workstation that attaches to the OBC via RS-232, and to the rest of the SCPS testbed via Ethernet.

The Spanner is also a Sun workstation that attaches to the Ethernet. In the laboratory test environment, the delays, data rates, and error rates of the satellite environment are emulated by the Spanner program. Using this configuration, we can route CCSDS Telemetry and Telecommand packets through Spanner to impose the delays, data rates, and error rates of the STRV communication environment. (Note that we have not attempted to accurately model the error *distributions* of the STRV space link environment with Spanner - it inserts bit-errors according to a Bernoulli process [3].)

---

1 This difference between the OBC and the flight system is significant - the flight system generates a stream of synchronous frames, while the OBC can generate asynchronously. We found in our analysis of the experiment data that this difference produced a discrepancy between the bit-efficiency of the tests conducted in the laboratory and of those conducted in the field, due to queuing in the laboratory's Spanner system, which simulates delays and errors. We will discuss this in our presentation of the field test bit-efficiency results.

The Ground System is the machine that acts as the ground-based SCPS-TP endpoint. It sends CCSDS packets bound for the OBC to the Spanner host, and receives packets from Spanner that originated on the OBC. Note that the outbound interface for the Ground System workstation is Ethernet. There is no handshaking between the Ground System workstation and the Spanner host.

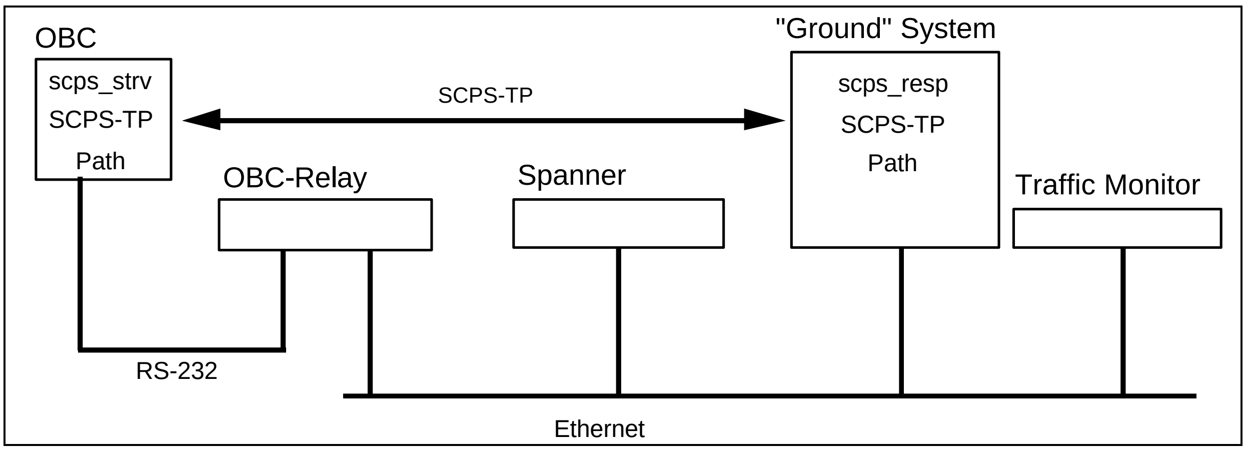


Figure 1. SSFE Laboratory Test Configuration

We used a fourth workstation to independently monitor the traffic on the Ethernet, logging all packets exchanged between the OBC Relay and Spanner, and between Spanner and the Ground System. The utility that monitors the Ethernet traffic is called *tcpdump*, and was developed by Lawrence Berkeley Labs. We have extended it to understand the SCPS extensions to TCP in general, and, for this test, CCSDS Telemetry and Telecommand packets.

With this configuration we were able to test SCPS-TP's reliable transfer mechanisms using a broad range of error rates and configuration options. The results of this testing are reported in Section 6.

4.1.2 Protocol Configuration

For the SSFE testing, we were particularly interested in the effects of three SCPS-TP capabilities on performance:

- the TCP Timestamps option;
- the SCPS-TP header compression capability;
- the SCPS-TP Selective Negative Acknowledgment (SNACK) option.

We conducted the same experiment at five different bit-error rates: 10^{-6}, 10^{-5}, $2x10^{-5}$, $5x10^{-5}$, and 10^{-4}. The experiment was of the form $2^k r$, where k=3 and r=10. The 2^3 design yielded eight configurations to test, as shown in Table 1:

Table 1. SCPS-TP Configurations for Laboratory Testing

Configuration	Header Compression	TCP Timestamps	SNACK
Config_1	Off	Off	Off
Config_2	Off	Off	On
Config_3	Off	On	Off
Config_4	Off	On	On
Config_5	On	Off	Off
Config_6	On	Off	On
Config_7	On	On	Off
Config_8	On	On	On

Note that while Config_1 has none of the SCPS-TP capabilities enabled that were under test, it is *not* representative of the performance of TCP. Config_1 is still using features that are not available in TCP: rate control, reduced acknowledgment frequency, and non-use of congestion control. Refer to [18] for a more detailed discussion of these and other SCPS-TP features. For a high-level overview of SCPS-TP features, refer to Appendix D of this document. For a comparison of SCPS-TP performance to TCP performance, refer to [13].

Table 2, below, presents the settings of parameters that were invariant throughout the laboratory tests. Note that downlink packet sizes were fixed at 90 bytes throughout the testing.

Table 2. SCPS-TP Parameter Settings for Laboratory Experiments

Configuration Parameter	Onboard Setting	Ground Setting
Buffer Size Send buffer Receive Buffer	 19712 bytes 19456 bytes	 71928 bytes 69880 bytes
Outbound SCPS-TP Packet Size	≤ 84 bytes	≤ 250 bytes
Rate control settings	1000 BPS	125 BPS
Congestion control	Off	Off
Ack Frequency	1 Ack/16 Seconds (approx. 1 Ack per	1 Ack/8 Seconds (approx. 1 Ack per

	2 round trip times)	round trip time)
Window scaling	On	On

The amount of user data per packet varied according to header size, and was allowed to fully fill the 90-byte packets. The total volume of data for each run was sized to result in an integral number of fully-filled data packets.

4.2 Field Configuration

This section describes the equipment and protocol configurations of the flight and ground segments.

4.2.1 Equipment Configuration

In the discussion of the flight and ground segments, we start with general information, then present information more specific to the SCPS elements of the segment.

4.2.1.1 Flight Segment

The STRV program includes two satellites, STRV 1a and STRV 1b. The two satellites are in Geostationary Transfer Orbit (GTO), and control operations are conducted from a DRA site at Lasham, England. The STRV spacecraft weigh approximately 50 kg each, are cuboid (approximately 0.45 m on each side), with body-mounted solar arrays on four sides. The spacecraft were released spinning at five revolutions per minute by an Ariane 4 launch vehicle, and the attitude control system maintains an approximate solar aspect angle of 90 degrees. Each spacecraft has two onboard computers that use the MIL-STD-1750A microprocessor, manufactured using a Silicon-on-Sapphire process for radiation tolerance. Each primary computer (and the secondary computer on STRV 1b) has 128 kB Random Access Memory (RAM) and 64 kB Read Only Memory (ROM), both addressable as 16-bit words. The spacecraft use S-band frequencies for communication directly to a 12 meter antenna at Lasham and via the NASA Deep Space Network (DSN) antennas to the control center at Lasham.

The STRV spacecraft are believed to be the first in Europe to implement (in full) the Consultative Committee for Space Data Systems (CCSDS)-compatible European Space Agency (ESA) Packet Telemetry and Telecommand standards.

We were allowed to use the secondary processor (OBC2) of the STRV 1b spacecraft for the SSFE. Figure 2 illustrates the SCPS software architecture as implemented in OBC2. OBC2 of the STRV 1b has a MIL-STD-1750A CPU and 64k 16-bit words of random access memory. The lower 32k words of the address space are shadowed by Read Only Memory (ROM) containing code to support the primary STRV 1b mission. A small portion of this code comprises an executive that provides access to basic OBC functionality: the ability to receive telecommands, the ability to send telemetry, the ability to read the system clock, etc.

All onboard code but that which provides the basic OBC functionality is overwritten when the SCPS software is uploaded to RAM.

The remainder of the software to support the SCPS-TP testing consists of two separately-compiled programs. The first is the SCPS Kernel. This is a utility program that provides basic control operations, but is not active when a test is running. The SCPS Kernel accepts CCSDS telecommands and generates CCSDS telemetry packets. It provides the ground-based user with the ability to examine and change onboard memory, to compute cyclic-redundancy codes (CRCs) over various ranges of memory, to upload and download areas of memory, and to pass control to other programs for execution (and to receive control back from them upon termination of the program).

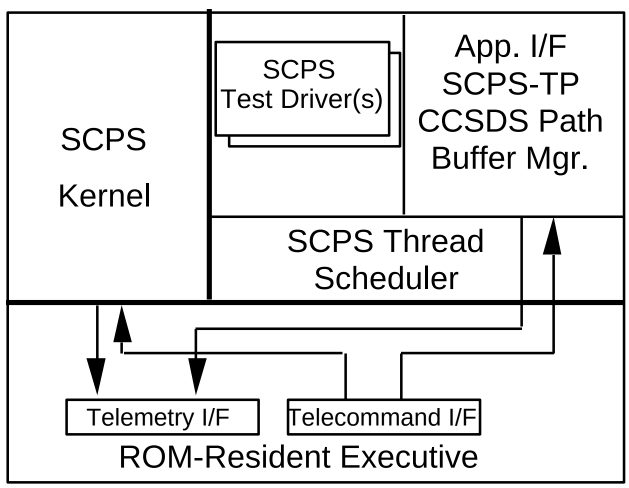


Figure 2. SSFE Onboard Software Configuration

The second program to support SCPS-TP testing is the SCPS-TP implementation and its test drivers. The SCPS-TP implementation consists of three basic parts: the test driver applications, the protocols, and a thread scheduler. The test driver applications provide the ability to source and/or sink data via SCPS-TP sockets. The protocols consist of SCPS-TP and the software to generate CCSDS Telemetry packets and parse CCSDS Telecommands, along with the SCPS socket interface software and the buffer management software. Finally, the thread scheduler provides a simple mechanism to support multiple concurrent threads of execution. In this manner, the test driver applications may be more loosely-coupled from the protocol implementations than if there were only a single thread of control.

4.2.1.2 Ground Segment

The STRV ground segment is situated at Lasham, England. It is based on a network of PCs using a combination of COTS and custom software. Figure 3 provides a high level overview of the system configuration. The SCPS Workstation, shown on the left-hand side of the figure, is a 486-based Personal Computer (PC) running the FreeBSD operating system. The PC has a 14.4 kbps internal modem, through which the PC is connected to the Internet via a commercial internet provider. The SCPS software runs as an application on this workstation

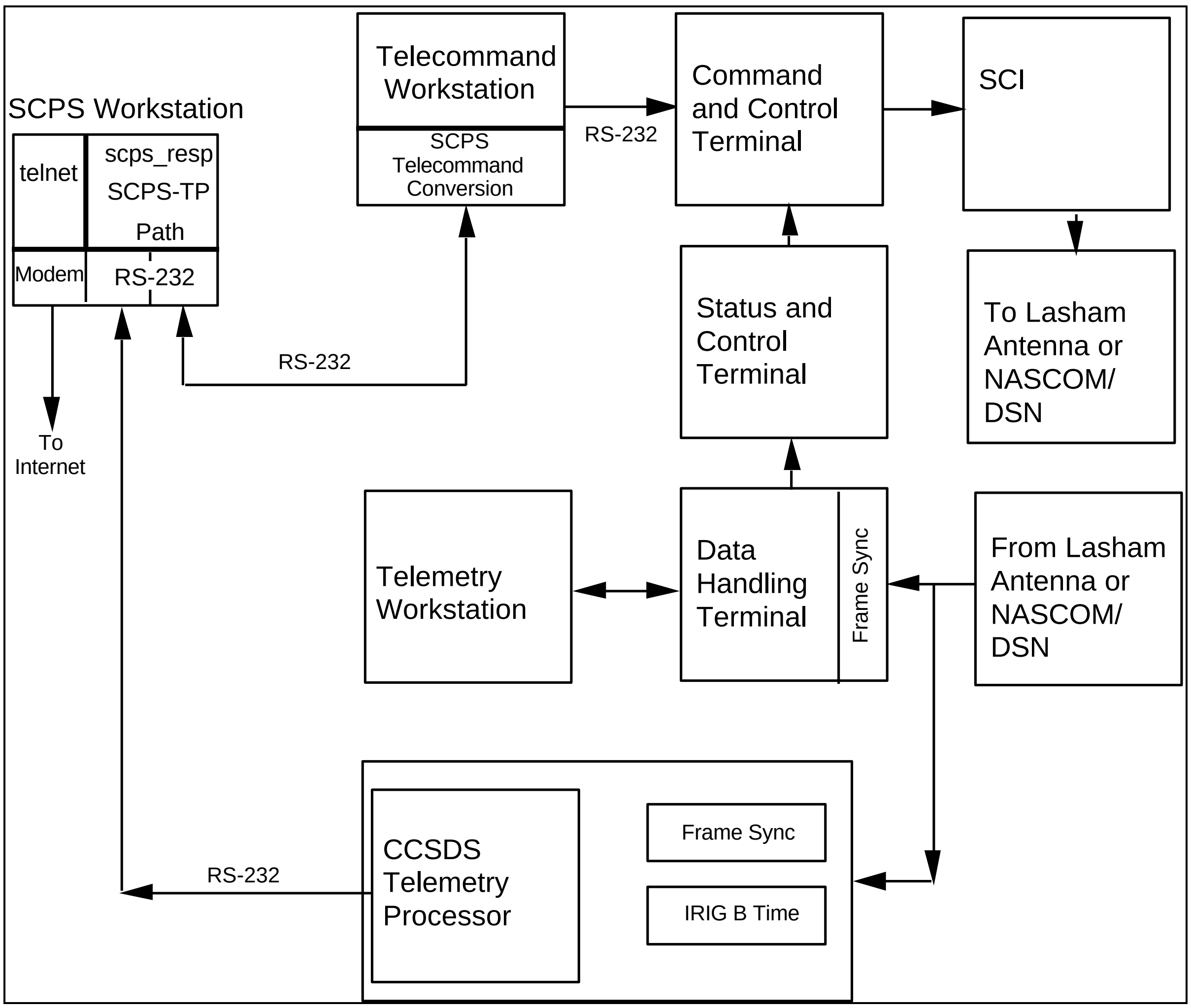


Figure 3. SSFE Ground System Configuration

and sends CCSDS Telecommand Packets (refer to [24], [25], [26], [27]) over an RS-232 line to the Telecommand Workstation. The Telecommand Workstation forwards the packets to the spacecraft by means of the Command and Control Terminal, the SCI, and ultimately via using a 12 meter antenna that is collocated with the Lasham ground station, via the NASA Deep Space Network, or via both. The Telecommand Workstation provides performance feedback to the SCPS Workstation over the same RS-232 line over which the SCPS Workstation sends commands. When the Telecommand Workstation receives an indication from the Command and Control Terminal that the telecommand has been enqueued for transmission, it passes a “Ready” signal back to the SCPS Workstation. The SCPS Workstation does not send another telecommand until it receives this “Ready” signal. This handshaking ensures that a queue does not build in the Telecommand Workstation.

The SCPS Workstation receives CCSDS Telemetry Packets [7] and frame status information from the CCSDS Telemetry Processor over another RS-232 line. The CCSDS Telemetry Processor works independently of the primary telemetry processing system, which is the “Data Handling Terminal.”

4.2.2 Protocol Configuration

The flight portion of the testing consists of a subset of the configurations examined in the laboratory testing. The configurations that we tested are shown in Table 3. As shown, the testing did not include the full range of tests with SCPS-TP header compression disabled. Rather, a single baseline configuration, Configuration 1, was tested with header compression disabled, and the remainder of the tests considered all combinations of the other two options, but with header compression enabled.

Table 3. SCPS-TP Configurations for Flight Testing

Configuration	Header Compression	TCP Timestamps	SNACK
Config_1	Off	Off	Off
Config_5	On	Off	Off
Config_6	On	Off	On
Config_7	On	On	Off
Config_8	On	On	On

The settings of parameters that were invariant throughout the flight tests are identical to those presented in Table 2. As with the laboratory testing, downlink packet sizes were fixed at 90 bytes throughout the tests. The amount of user data per packet varied according to header size, and was allowed to fully fill the 90-byte packets. Total volume of data for each run was sized to result in an integral number of fully-filled data packets.

Section 5

Methods

Both the laboratory tests and the field tests use the Expect scripting language [21] as the means of automating the test procedures. In our early tests, we attempted to conduct the tests “by hand.” The result was often successful, but too often we would make minor mistakes that rendered the test unusable (such as writing over a result file). Occasionally, we made more significant mistakes that required us to re-upload the software (such as transferring control to the wrong memory location to start the test). Clearly, we needed a means to provide repeatability in our processes. The Expect scripting language, and Tcl (i.e., the Tool Control Language) [29], on which Expect is built, provide such repeatability. Additionally, in the laboratory, entire test sessions could be scripted. With each laboratory test taking between 10 and 45 minutes to execute, and with 400 tests in the test suite, a full test run takes almost a week of uninterrupted operation. In the field, Expect gave us the ability to easily ensure that the configuration was correct for each test, that the test was started properly, and that the test results were collected and stored in a consistent manner. In addition, we had the Expect script invoke awk scripts to perform the first stages of data reduction “on the fly”, so that subsequent post-processing was less difficult. (Note: “awk” is a pattern-matching language that is useful for searching through files and extracting specific pieces of information.)

This section describes the methods used to conduct the tests, to gather the results, and to reduce the data into usable information. The processes are similar for the laboratory and the field tests, but sufficiently different that they are presented as separate subsections.

5.1 Laboratory Testing

The laboratory testing involved coordination of several UNIX-based workstations and the MIL-STD-1750A-based Onboard Computer. This section discusses how the tests were conducted and how the operation of the workstations and OBC were coordinated.

5.1.1 Test Execution

Recall Figure 1, which shows the laboratory equipment configuration. There are four UNIX workstations: the OBC-Relay, the Spanner host, the “Ground” system, and the traffic monitor. We chose to host the Expect language on the OBC-proxy host, primarily because the job of relaying packets to and from the OBC was not taxing.

Prior to testing, we loaded the SCPS kernel and the SCPS-TP implementation (including the thread scheduler, buffer manager, CCSDS path code, SCPS-TP code, and test drivers) onto the OBC and invoked the SCPS kernel. The SCPS kernel provides basic functionality:

reading and writing memory locations, passing control to a user-specified address, etc. These basic capabilities are used in the execution of the tests.

We compiled and loaded onto the Ground system host the eight executable SCPS-TP responder programs that corresponded to Configurations 1 through 8, as previously defined in Table 1. Additionally, we ensured that the *spanner* program was resident on the Spanner host, and the *tcpdump* program was resident on the Traffic Monitor host.

The basic flow of an individual test is as follows:

1. On the OBC Relay host, terminate and restart the CCSDS Path packet relay programs *relay_up* and *relay_down*. These programs relay packets from the Ethernet to the OBC and from the OBC to the Ethernet, respectively.
2. On the Spanner host, terminate and restart the *spanner* program with the bit-error rate appropriate to this run, plus the constant parameters specifying 250 millisecond propagation delays, 1000 bps downlink data rate, and 125 bps uplink data rate.
3. On the Packet Monitor host, terminate and restart the *tcpdump* program to log traffic on the Ethernet. The *tcpdump* program is invoked with filters that restrict its logging to only those packets that are relevant to the ongoing test. Redirect the output of the tcpdump program to a log file.
4. On the Ground system host, terminate and start the SCPS-TP responder program corresponding to the protocol configuration under test. Specify the Spanner host as an intermediate destination for any data bound for the OBC. Redirect the output of the responder program to a file. (The responder program prints debug information, including a snapshot of every packet received, and post-run summary information.)
5. Command the OBC to return control to the SCPS Kernel and verify the command has been acted upon. Write to specified onboard memory locations to select the appropriate SCPS test driver, the number of packets to transfer, and the amount of user data per packet to transfer. Verify these memory locations by reading them back and monitoring the debug output of the SCPS-TP responder program. (The SCPS-TP responder program is waiting for the OBC to initiate a SCPS-TP connection. Any packets received that are not part of the SCPS-TP protocol are printed, but otherwise ignored by the SCPS-TP protocol.)
6. Command the OBC to start the test. Verify that the test has started by monitoring the output of SCPS-TP responder program on the Ground system host. Continue to monitor the test execution until the responder program indicates that the connection has closed, or a timeout value expires. If the connection was properly closed, set a test status indication to “Pass.” If the timeout counter expired, set the test status indication to “Failed.”
7. Collect the various log files from the Ground system host, the Spanner host, and the Traffic Monitor host. Perform the initial post-processing and write the results to a file

along with test identification information (e.g., configuration, repetition, bit-error rate, etc.).

The remaining two subsections describe the data that is collected for an individual test, and the initial post-processing that is performed to summarize the results of an individual test.

5.1.2 Data Collected

For each individual test, the following log files were collected:

- The *spanner* log file, which contains an indication of each packet dropped, by direction (i.e., uplink or downlink).
- The *tcpdump* log file, which contains a timestamped copy of each packet on the Ethernet. These detailed files are used primarily for debugging, and are not reported on in this document.
- The SCPS-TP responder log file, which contains a printout of every packet received by the SCPS-TP responder, plus debugging information and end-of-test summary information.

5.1.3 Data Reduction

The post-processing that is done as part of the test execution is performed primarily by means of the awk [1] programming language. Awk is a pattern-matching language that is well-suited to the data reduction tasks required to transform the various log files into something that is compact and representative of the test results.

Table 4 contains a list of the data items that are produced from the log files after each laboratory test completes. These data items, along with identifiers for the test, such as the configuration number, the requested bit-error rate, and the repetition number, are recorded in a summary file. The primary experimental results that we derive from this information are throughput, (down) link utilization, and bit-efficiency. The equations to calculate these results are given in Appendix A. The post-processed data items, throughput, link utilization, and bit-efficiency data for each laboratory test appear in Appendix B.

5.2 Field Testing

The field testing environment differed significantly from the laboratory testing environment. For the field testing, we had the ability to write scripts only for the SCPS workstation. In some ways, this simplified testing, since there was not the need to coordinate the actions of many different machines. However, there were necessary data items that could not be directly measured, such as the number of lost packets, that had to be calculated. In addition, we did not have a packet monitor active in the system, so we generated *tcpdump* log files at the SCPS Workstation (refer to Figure 3. SSFE Ground System Configuration).

Table 4. Post-Processed Data Items Recorded for Each Laboratory Test

Data Item	Description
Elapsed Time	The time from when the first SCPS-TP packet sent until the last SCPS-TP packet is received
Elapsed Time in Data Transfer Phase	The time from when the first data-carrying SCPS-TP packet is sent until the last data-carrying packet is acknowledged.
Packet Count Down	The number of packets sent from OBC to Ground. Separated into Compressed SCPS-TP packets and Uncompressed SCPS-TP packets.
Packet Count Up	The number of packets sent from Ground to OBC. Separated into Compressed SCPS-TP packets and Uncompressed SCPS-TP packets.
Byte Count Down	The total number of bytes of data sent by the SCPS-TP entity on the OBC to Ground, inclusive of SCPS-TP headers, but not including CCSDS Packet Headers or framing overhead.
Byte Count Up	The total number of bytes of data sent by the SCPS-TP entity on the Ground to the OBC, inclusive of SCPS-TP headers, but not including CCSDS Packet Headers or framing overhead.
Packet Drops	The total number of (downlink) packets dropped or corrupted during the SCPS-TP session. (Note: the number of uplink packets dropped or corrupted during the SCPS-TP session is also recorded, but since the flight system uses the CCSDS COP-1 retransmission protocol, this value is always zero.

5.2.1 Test Execution

We hosted the Expect scripting language on the SCPS Workstation, along with the SCPS-TP responder software. As with the laboratory testing, the SCPS protocol software was uploaded to the spacecraft prior to the execution of the test, and the SCPS Kernel was invoked. The test script performed the following actions for each test:

1. Terminate and start the SCPS-TP responder program corresponding to the protocol configuration under test. Redirect the output of the responder program to a file. Due to the absence of a Traffic Monitor host, the responder program also produced a *tcpdump*-compatible log of all packets sent and received.
2. Command the STRV 1b to return control to the SCPS Kernel and verify that the command has been acted upon. Write to specified onboard memory locations to select

the appropriate SCPS test driver, the number of packets to transfer, and the amount of user data per packet to transfer. Verify these memory locations by reading them back and monitoring the debug output of the SCPS-TP responder program.

3. Command the STRV 1b to start the test. Verify that the test has started by monitoring the output of SCPS-TP responder program on the SCPS Workstation. Continue to monitor the test execution until the responder program indicates that the connection has closed, or a timeout value expires. If the connection was properly closed, set a test status indication to “Pass.” If the timeout counter expired, set the test status indication to “Failed.”
4. Save the log files from the SCPS Workstation. Perform the initial post-processing and write the results to a file along with test identification information (e.g., configuration, repetition, etc.).

5.2.2 Error Introduction and Measurement

In the field, we did not have the precise control over the bit-error rate that we have in a laboratory environment. The signal quality for STRV-1b varied widely, from zero bit-errors over the course of a 10-minute test, to so many bit-errors that the spacecraft signal could not be distinguished from background noise.

For those runs in which we needed bit-errors during the time that the link was clean, we tried two primary methods of error introduction. The first involved cycling power on the frame synchronizer that was part of the CCSDS Telemetry Processor. This had three primary disadvantages: first, the processor required several frames to resynchronize, so the shortest link outage was several seconds long; second, the distribution of errors is different than if errors were “real”, especially at low error rates; and third, the fact that telemetry was missing was not recorded by the Telemetry Workstation, which received its telemetry before the frame synchronizer and was therefore not useful in independently verifying the bit-error rate. The second alternative proved to be much more satisfactory: we drove the antenna “off point” in both azimuth and elevation. By gradually adjusting the antenna pointing, we discovered that we could gain some reasonable control over bit-error rate. The adjustments were in the range of -250 seconds in elevation and -60 seconds in azimuth.

To measure bit-error rate, we used three independent techniques: we counted packet loss, we counted bad bits in the known portions of idle packets, and we counted bad bits in the known portions of SCPS-TP packets. The first, counting packet loss, is the technique used to plot the data presented in Section 6. We chose packet loss as the basis of our bit-error rate estimates for two reasons. First, it yields a conservative (lower) estimate of bit-error rate, since the technique for calculating bit-error rate assumes a maximum of one bit-error per packet (see Appendix A, Equation 9) while many bit-errors per packet are possible. Second, SCPS-TP responds to packet errors. The response of SCPS-TP to a packet with an error packet is the same whether that packet has one error or ten. (Note that SCPS-TP’s response *is* affected by whether several *consecutive* packets are in error, but that situation is accounted for with the packet-loss technique.) Of the three techniques cited above, the packet-error rate method always yielded a result that was better than or equal to the technique that counted known portions of SCPS-TP packets. There was relatively good consistency between the three techniques, with differences typically being approximately 2:1 between the techniques. There were two tests in which the differences between techniques were approximately 10:1, and in both of these tests the packet-error counting technique yielded the lower estimate of bit-error rate.

We *format* our results with an X-axis in units of bit-error rate rather than as in units of packet-error rate because bit-error rate is a more widely-understood concept. Readers should bear in mind that the results presented here were taken with packets that were no larger than 90 bytes. Longer packets would have fared more poorly at the high bit-error rates. (Note that bulk data transfers will typically use the largest packet size available, while command-response traffic may vary in size but tends to be in the tens-to-low-hundreds of bytes per packet range.)

5.2.3 Data Collected

The following files were collected for each test run:

- The *tcpdump* log file, generated by the SCPS-TP responder program, which contains a timestamped copy of each packet sent or received by the SCPS Workstation. (Note that downlink packets lost in transmission, rather than corrupted, are not captured. We used the CCSDS Telemetry Packet sequence number to identify the maximum number of packets that had been sent, which allowed us to calculate the number of packets that were not received by the SCPS Workstation.)
- The SCPS-TP responder log file, which contains a printout of every packet received by the SCPS-TP responder, plus debugging information and end-of-test summary information.

5.2.4 Data Reduction

Table 5 lists the data items that were generated by the data reduction activities conducted in the field. The data reduction activities for the field data were slightly different than for the laboratory data. The primary difference was the way in which the bit-error rate was calculated. Instead of having a direct measure of packet drops (as provided by the Spanner log file in the laboratory tests), we had calculate the number of lost packets. We did this by subtracting the number of packets received from the maximum CCSDS Telemetry packet sequence number on the link (this sequence number is reset to zero for each test). In addition, we logged the start and end times of each test, for correlation with other log files.

Table 5. Post-Processed Data Items Recorded for Each Field Test

Data Item	Description
Test Start Time	Time (in local time) of receipt of initial SCPS-TP packet
Test End Time	Time (in local time) of receipt of final SCPS-TP packet
Elapsed Time	Time from when the first SCPS-TP packet is received until the last SCPS-TP packet is received
Elapsed Time in Data Transfer Phase	Time from when the first data-carrying SCPS-TP packet is received until the last data-carrying packet is acknowledged.
Packet Count Down	Number of good packets received by the ground system. Separated into Compressed SCPS-TP and Uncompressed SCPS-TP packets.
Packet Count Up	Number of packets sent from ground to spacecraft. Separated into Compressed SCPS-TP and Uncompressed SCPS-TP packets.
Byte Count Down	Total number of bytes of SCPS-TP received without error by the ground, inclusive of SCPS-TP headers, but not including CCSDS Packet Headers or framing overhead.
Byte Count Up	Total number of bytes of data sent by the SCPS-TP entity on the ground to the spacecraft, inclusive of SCPS-TP headers, but not including CCSDS Packet Headers or framing overhead.
Max Sequence Number	Maximum CCSDS Telemetry packet sequence number appearing in the test. The number of packets that were lost or corrupted in a particular data run are calculated by subtracting from this value the total number of packets received.

As with the laboratory testing, the primary experimental results that we derive from this information are throughput, down link utilization, and bit-efficiency. The equations to calculate these results are given in Appendix A. Note that bit-error rate must be calculated for the field data. The equation to calculate bit-error rate from packet loss count also appears in Appendix A. The post-processed data items, throughput, link utilization, and bit-efficiency data for each field test appear in Appendix B.

Section 6

Results

This section presents and discusses the results of the laboratory and field tests. The laboratory results are presented first, and reflect an environment over which we had more control than in the field. The field tests were conducted to confirm the laboratory results.

6.1 Laboratory Experiment Results

We collected throughput, link utilization, and bit-efficiency data for each of the eight configurations at five different bit-error rates. The tables in Appendix B contain the detailed results. We conducted each test ten times, and computed an *allocation of variation* to help understand the results. An allocation of variation is a technique that allows us to isolate the contribution of each factor in the experiment, and to identify the interaction among factors. For each of the primary results (throughput, link utilization, and bit-efficiency), we present and discuss the allocation of variation as a summary of SCPS-TP performance. We follow that with graphs of the performance as appropriate.

6.1.1 Throughput Results

Throughput measures the average rate at which the protocol can move user data, and is one of the most commonly used measurements of communication protocol performance. Throughput is presented here in units of bits per second, and is calculated by dividing the number of bits of user data transmitted over the connection by the amount of time that the protocol spends in the data transfer phase of its connection. (We measure the data transfer phase of the connection as the elapsed time from the transmission of the first data packet to the transmission of the acknowledgment for the *last* data packet.) Note that throughput calculations are based on the amount of *user data* transferred. Retransmissions are not counted as user data.

Summary

Table 6 presents the results of the allocation of variation in throughput performance. This table summarizes the results of five separate $2^k r$ experiments, where k=3 and r=10. The five separate experiments correspond to the bit-error rates listed in the first column of the table. The second column of the table lists the range of variation in the mean throughput results. That is, each of the eight configurations was tested ten times and the average was computed for each configuration. The variation column represents the difference between the highest average and the lowest average. Since this table is addressing throughput performance, the units of the variation are bits per second. The link data rate was 1000 bps, and SCPS-TP headers and user data could occupy 252 bytes of each 328-byte frame, so the maximum possible throughput was 768 bps. The next eight columns of Table 6 attribute a portion of the variation to one of the three SCPS-TP capabilities under test (SNACK, Timestamps, and Header Compression), to interactions among the capabilities, or to experimental error. For a thorough explanation of the technique of allocation of variation, including examples, refer to [11], pages 293-313.

Consider the first line of Table 6. This line of the table addresses our results at a bit-error rate of 10^{-6}. The table informs us that the difference between the best and worst mean throughput at this bit-error rate was 213 bps, or almost 28% of the available data rate. (The table doesn't show us the absolute throughput values; however, the throughput performance graphs, presented later in this section, show the range of throughputs obtained.)

The third column of the table indicates that the Selective Negative Acknowledgment (SNACK) capability contributed nothing, either positively or negatively, to the variation in throughput. SNACK is used to recover from lost or corrupted packets, and the SNACK option is not transmitted except when errors are present. At a bit-error rate of 10^{-6}, errors are extremely rare, so it is not particularly surprising that SNACK had no measurable effect.

The fourth column of the table considers the TCP Timestamps capability, which contributes negatively to throughput, and accounts for approximately 34% of the variation listed in column 2. This, too, is not surprising. The TCP Timestamps capability improves SCPS-TP's ability to estimate round-trip time. Using this improved estimate of round-trip time, SCPS-TP can more accurately set its retransmission timeout value, which is used in error recovery (the retransmission timeout value is more important when the SNACK option is not in use, as we shall see later). Unlike the SNACK option, the TCP Timestamps capability is present even when it is not needed, and it is relatively large compared to the size of the SCPS-TP header. The size of the TCP Timestamps option contributes negatively to throughput by "crowding out" user data. Since the maximum size of the STRV packets is 90 bytes, we see from the table that this effect is significant: over one third of the variation in throughput at 10^{-6} is due to the negative effects of the TCP Timestamps option. (Note that this impact would decrease if the size of the packet were increased.)

The fifth column presents the effect of SCPS-TP Header Compression. SCPS-TP Header Compression makes protocol headers smaller by reducing invariant information, allowing more user data per packet. The effect of SCPS-TP header compression accounts for 64% of the variation in throughput reported in column 2, and its impact is to improve throughput.

We defer discussion the interaction components of the table, columns 6 through 9, to the next paragraph. The final column of the table indicates the portion of the variation that is not explained by the three factors under consideration: SNACK, Timestamps, and Header Compression. If we scan down the table, we see that the error column grows, then diminishes. This corresponds to the throughput performance of the protocol as it nears the "knee of the curve" - at bit-error rates between $2x10^{-5}$ and 10^{-4}, we see relatively large swings in throughput between subsequent iterations of the same protocol configuration. The error column captures this inconsistency in performance. We see that same inconsistency displayed graphically later in this section.

To understand the import of columns 6 through 9, the interaction components, consider the last row of Table 6, the one for the BER = 10^{-4}. Here we see that the variation in throughput is considerably larger than for any of the other bit-error rates. We also see that the SNACK capability represents a large, positive portion of this throughput. We see that TCP Timestamps contribute positively to this variation in throughput, but to a very small degree. We see also that the relative contribution of SCPS-TP Header Compression is still positive, and significant, but not nearly as great as the contribution of the SNACK capability. Now look at the interaction columns, particularly column 6, labeled "SN, TS", which considers the interaction between the SNACK capability and the TCP Timestamps capability. The value is -13%, and represents the effect of SNACK and TCP Timestamps operating in combination. The SNACK, Timestamps, and "SN, TS" columns tell us that if we use SNACK, we can expect a large, positive change in throughput at this bit-error rate. Also, if we use TCP Timestamps, we can expect a small, but positive effect on throughput. If we use *both* SNACK and TCP Timestamps, we can expect an improvement that is less than the sum of the improvements made by SNACK and TCP Timestamps individually. Quantitatively, we would expect a change of approximately (64% + 2% - 13% = 53%) of the variation. The percentages in the table are best used as *qualitative* indications of relative effect. We will examine methods of predicting future performance when we consider the Field Test results.

Table 6. Allocation of Variation in Throughput Performance

BER	Variation	SNACK	Timestamps	Compr.	SN,TS	SN,CM	TS,CM	All	Error
1E-6	213	0%	-34%	64%	0%	0%	1%	0%	1%
1E-5	215	1%	-25%	63%	-1%	0%	1%	0%	9%
2E-5	235	3%	-28%	49%	0%	0%	2%	0%	17%
5E-5	263	28%	-3%	42%	-3%	0%	1%	0%	21%
1E-4	362	64%	2%	13%	-13%	0%	2%	0%	6%

To summarize the information from Table 6, we see that SNACK makes a strong, positive contribution to throughput at bit-error rates of $5x10^{-5}$ and above. TCP Timestamps contribute negatively to throughput at low bit-error rates, but the contribution is less negative as bit-error rates increase. SCPS-TP Header Compression makes a strong, positive contribution to throughput at low bit-error rates, but its contribution is gradually overtaken by the effects of SNACK as bit-error rates increase. When SNACK is used in conjunction with TCP

Timestamps, the average throughput is better than with TCP Timestamps alone, but not as good as with SNACK alone.

Experimental Data

The following six graphs show the throughput of SCPS-TP as a function of bit-error rate. The first four of the graphs show each of the eight configurations, two to a graph. The last two highlight specific pairs of configurations. Each point on the graph represents the average of ten tests at the specified bit-error rate. The lines that extend above and below the data point correspond to the 90% confidence interval for that data point. The 90% confidence interval means that we can say, with 90% confidence, that the true mean performance of that configuration at that bit-error rate lies within the range delimited by the confidence interval. Note that for some data points, particularly those at 10^{-6} BER, the confidence interval is almost indistinguishable from the data point itself. This indicates that there was very little variation in the performance of the tests at that bit-error rate. A wider confidence interval means that there was more variation in the throughput results.

Figure 4 shows the throughput performance versus bit-error rate for two of the protocol configurations. The vertical axis of the graph extends to 768 bits per second, the maximum throughput possible after framing and CCSDS Telemetry packet overhead is taken into account. The line labeled Configuration 1, as indicated in Table 1, represents the throughput of the protocol without the SNACK, TCP Timestamps, or SCPS-TP Header Compression capabilities enabled. The line labeled Configuration 2 shows the throughput with SNACK enabled, but with Header Compression and TCP Timestamps disabled. Note that the performance of Configuration 1 and 2 are very similar until the bit-error rate exceeds $2x10^{-5}$, then the performance of Configuration 1 drops significantly. Note also that the confidence intervals surrounding the points for Configuration 2 are generally smaller than those for Configuration 1, indicating less variability in performance from test to test.

Figure 5 shows the throughput performance of Configurations 3 and 4 versus bit-error rate. Configuration 3 has the TCP Timestamps capability enabled, while Configuration 4 has both TCP Timestamps and SNACK enabled. Compare this figure with Figure 4. Note that the throughput at low bit-error rates is significantly lower than the throughput at equivalent bit-error rates in Figure 4. This is due to the overhead of the TCP Timestamps option, which appears on every packet. Note also that Configuration 4, the configuration with the SNACK capability enabled, significantly improves performance as the bit-error rate exceeds $2x10^{-5}$.

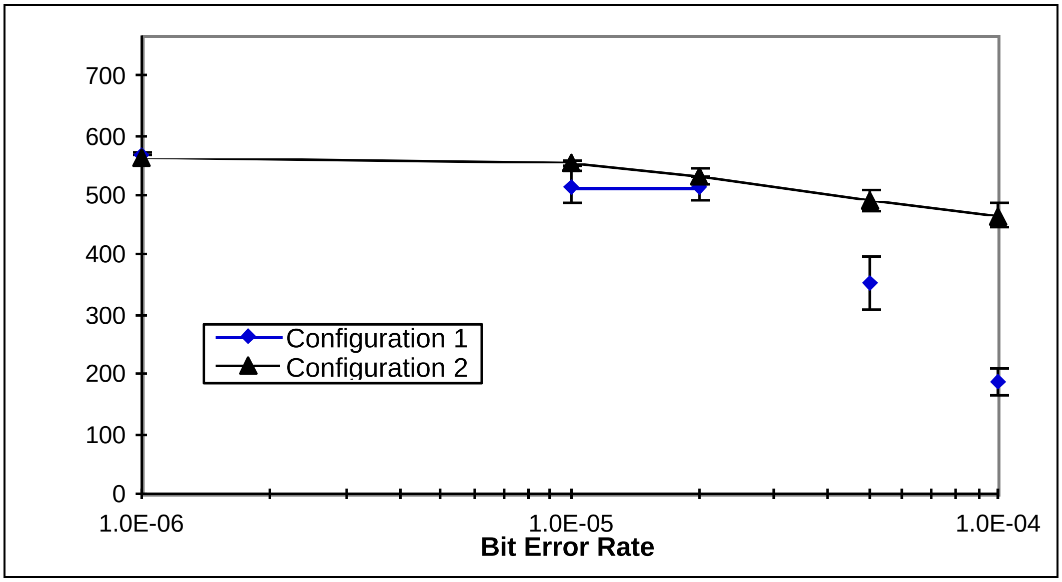


Figure 4. Throughput Performance of SCPS-TP Configurations 1 and 2

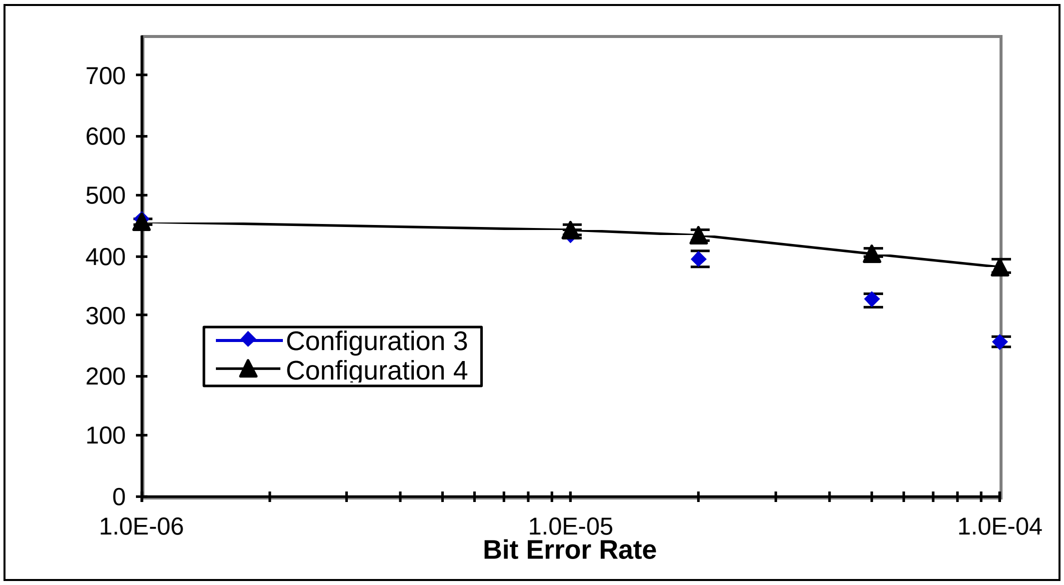


Figure 5. Throughput Performance of SCPS-TP Configurations 3 and 4

Figure 6 and Figure 7 present throughput performance for the other four configurations: those with the SCPS-TP Header Compression capability enabled. The effect of Header Compression is to reduce the size of the SCPS-TP headers, allowing more user data per packet. As a result, these four curves are shifted up on the y-axis when compared to the curves in Figure 4 and Figure 5. The shape of each curve is otherwise relatively unchanged.

Recall what the allocation of variation in Table 6 indicated: that the use of the SNACK capability has no effect at low bit-error rates, and a large, positive effect on throughput at high bit-error rates. We can see this effect in Figure 4 through Figure 7: each graph compares a configuration that has the SNACK capability enabled to an equivalent configuration without SNACK.

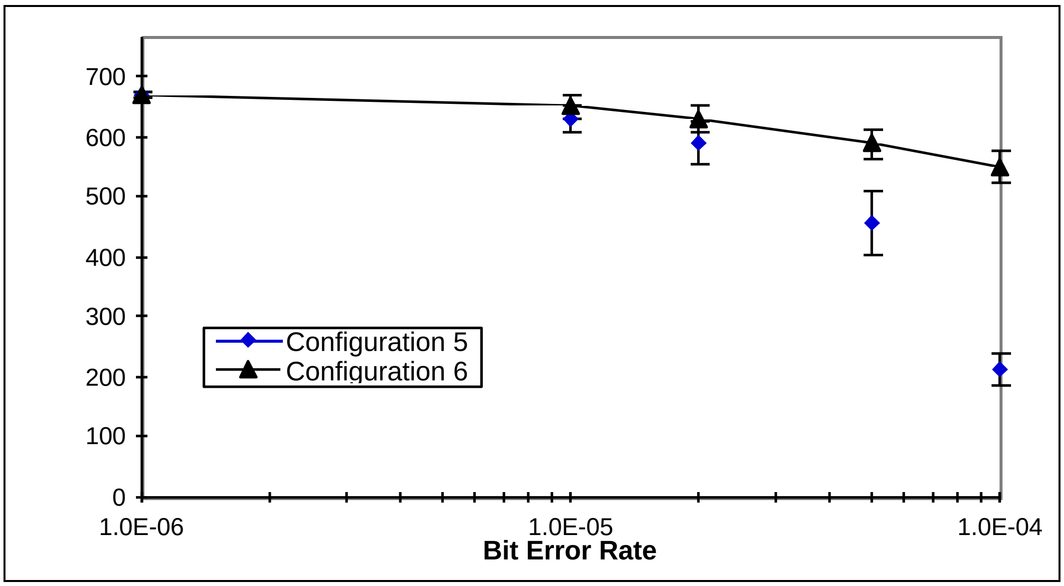


Figure 6. Throughput Performance of SCPS-TP Configurations 5 and 6

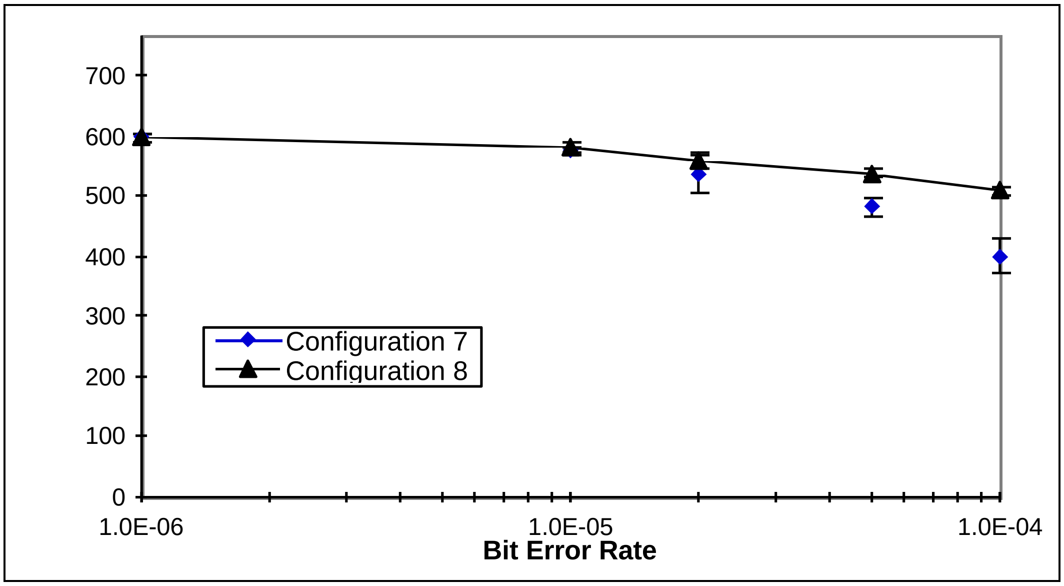


Figure 7. Throughput Performance of SCPS-TP Configurations 7 and 8

Table 6 also shows that the contribution of SCPS-TP Header Compression is significant, but diminishes as the bit-error rate increases. The graph in Figure 8 confirms this: we compare the throughput performance of Configuration 1 (the configuration with none of the capabilities under consideration enabled) with the throughput performance of Configuration 5 (the configuration with Header Compression enabled). We can see that the Header Compression capability significantly improves throughput, but that the improvement is not as great at a BER of 10^{-4} as it is at a BER of 10^{-6}. Why is this so? At low bit-error rates, the channel is essentially fully utilized. Throughput is dominated by the time that it takes to clock out the data. Header Compression helps here, because it reduces the amount of clocking overhead associated with a given amount of user data. At high bit-error rates, the channel is less fully utilized, because time is being spent waiting for retransmission timers to expire. Throughput is no longer dominated by the time to clock out the data. Rather, the retransmission timer plays an important part in throughput, as well. (This is why the use of TCP Timestamps helps more at higher bit-error rates, as subsequent graphs show.)

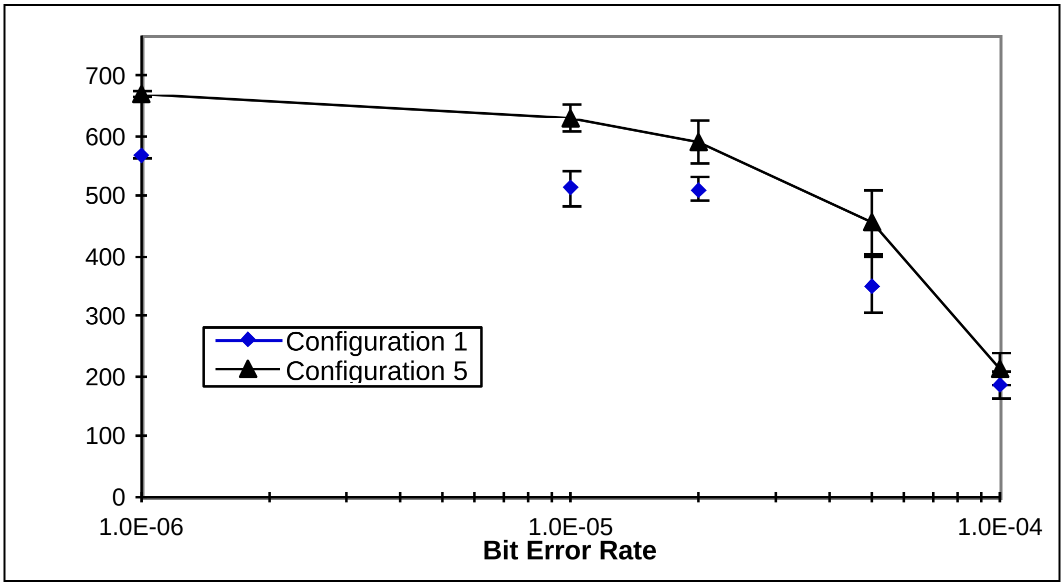


Figure 8. Throughput Performance of SCPS-TP Configurations 1 and 5

Now consider Figure 9 and Figure 10. The allocation of variation indicates that at low bit-error rates, the use of the TCP Timestamps capability has a negative effect on throughput, but that at high bit-error rates, the TCP Timestamps capability has a small, positive effect on throughput. We see this effect evidenced by the cross-over of the throughput lines for Configuration 1 (without Timestamps) and Configuration 3 (with Timestamps), and again by the cross-over of Configuration 5 (Header Compression but no Timestamps) and Configuration 7 (Header Compression and Timestamps).

Finally, we show in Figure 11 the effect of the interaction between SNACK and Timestamps. The throughput of three configurations versus bit-error rate are shown: one with SNACK only (Configuration 2), one with Timestamps only (Configuration 3), and one with both (Configuration 4). Table 6 asserts that at high bit-error rates, the throughput of a configuration that uses both SNACK and Timestamps will be better than Timestamps alone, but not as good as a configuration that only uses SNACK. We see this borne out in Figure 11.

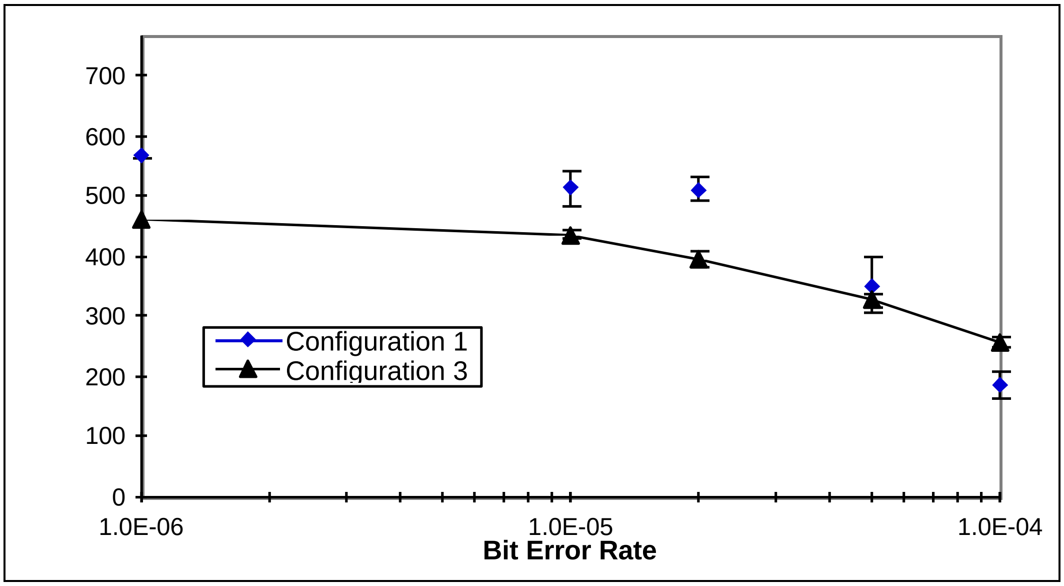


Figure 9. Effect of Timestamps on Throughput (Without Header Compression)

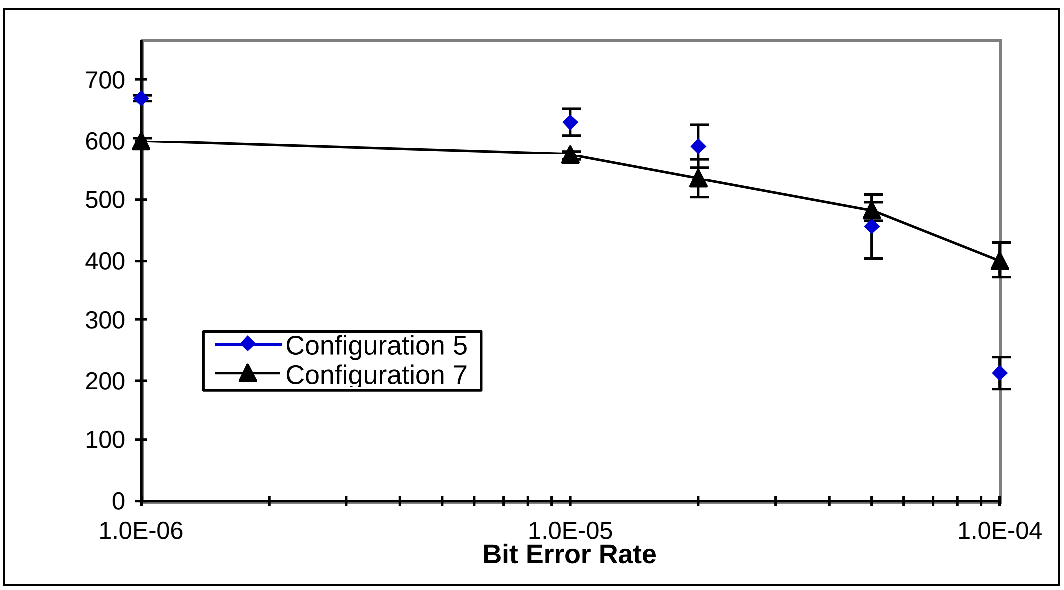


Figure 10. Effect of Timestamps on Throughput (With Header Compression)

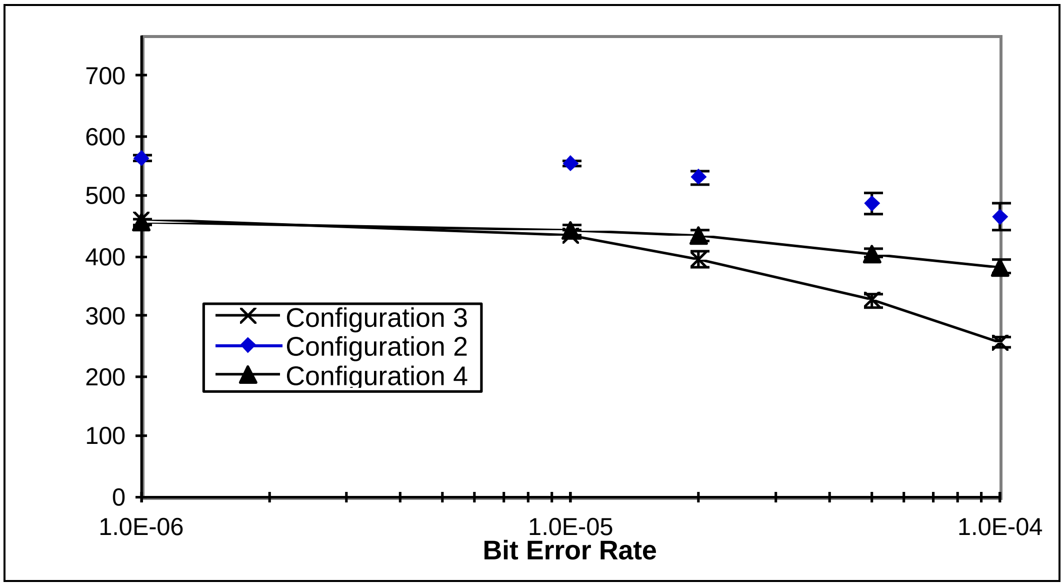


Figure 11. Effect of the Interaction Between Timestamps and SNACK on Throughput

6.1.2 Link Utilization Results

Link utilization is a measure of the ability of SCPS-TP to "keep the pipe full" (that is, to not allow the link to become idle when there is data ready to be transmitted). This is important in spacecraft communication when contact times with a satellite are limited and link capacities are constrained. Link utilization is calculated as a percent of maximum down link capacity, and is measured over the entire duration of the connection.

Summary

Table 7 shows the effect on link utilization of the three SCPS-TP capabilities being tested for five different bit-error rates. The structure of this table is identical to that of Table 6, which we examined in our discussion of throughput. Let us briefly examine what this table tells us.

First, at low bit-error rates (2×10^{-5} and below) there is very little variation in link utilization (6% or less). As the bit-error rate increases, we see that the variation in link utilization also increases (telling us that something is affecting link utilization). We see that both SNACK and Timestamps have a positive effect on link utilization at high bit-error rates, and that the effect of SNACK is more pronounced than that of Timestamps. Further, we see that there is an interaction between the effects of SNACK and Timestamps on link utilization: while SNACK and Timestamps both improve link utilization, when both are used, the total improvement is only slightly better than if only SNACK had been used. Finally, we see that Header Compression has little effect on link utilization over the entire range of bit-error rates.

Table 7. Allocation of Variation in Link Utilization

BER	Variation	SNACK	Timestamps	Compr.	SN,TS	SN,CM	TS,CM	All	Error
1E-6	2%	-2%	-2%	-4%	-3%	0%	0%	-1%	87%
1E-5	6%	5%	2%	0%	-7%	-3%	0%	3%	79%
2E-5	6%	13%	0%	0%	1%	0%	1%	-1%	83%
5E-5	24%	41%	7%	3%	-5%	-2%	0%	0%	42%
1E-4	55%	65%	16%	0%	-12%	0%	1%	0%	5%

Experimental Data

Figure 12 shows link utilization versus bit-error rate for Configuration 1 (none of the three SCPS-TP capabilities enabled) and for Configuration 2 (SNACK enabled). The graph shows the significant improvement that SNACK makes in link utilization at higher bit-error rates. Note that at low bit-error rates, the link utilization is over 95%, and that with SNACK enabled, it remains close to 90% even at a bit-error rate of 10^{-4}.

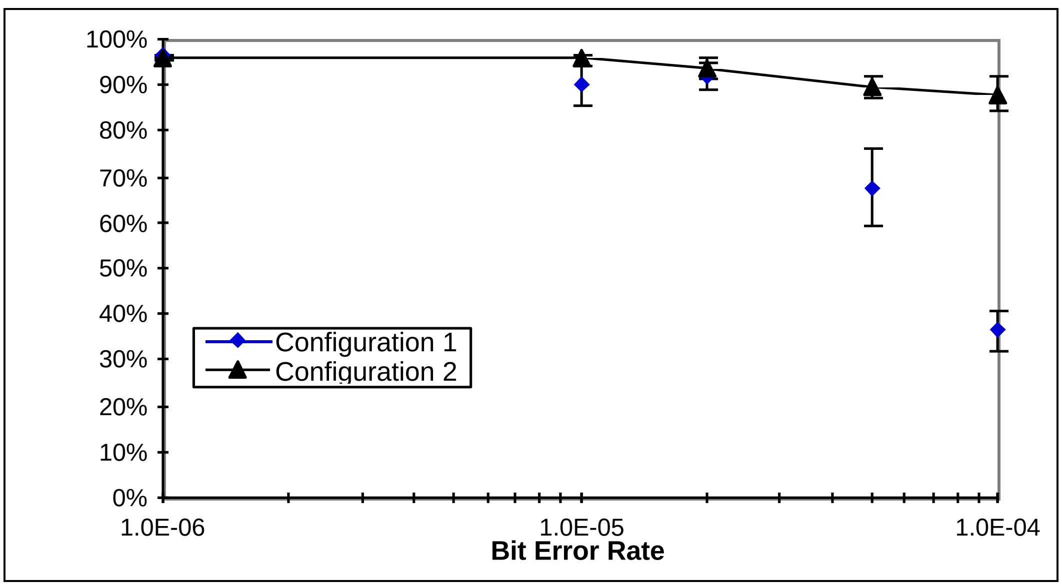


Figure 12. Link Utilization of SCPS-TP Configurations 1 and 2

Figure 13 shows the link utilization versus bit-error rate for Configuration 3 (Timestamps enabled) and Configuration 4 (both SNACK and Timestamps enabled). The results show that

Configuration 3 improves link utilization over Configuration 1 (shown in Figure 12), but does not improve link utilization as much as with SNACK enabled.

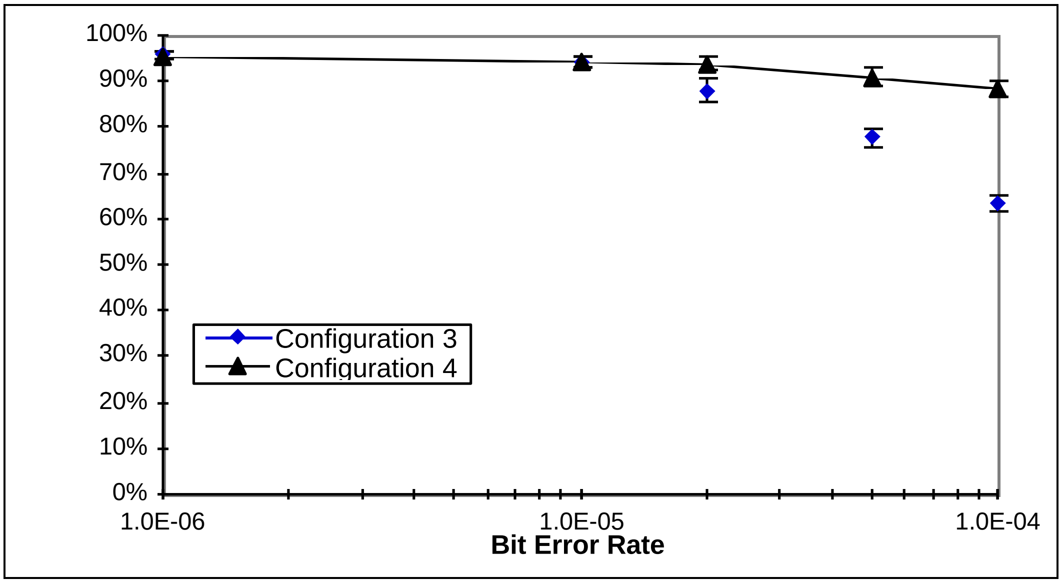


Figure 13. Link Utilization of SCPS-TP Configurations 3 and 4

Figure 14 and Figure 15 show link utilization versus bit-error rate for the configurations the have SCPS-TP Header Compression enabled. As Table 7 would lead us to believe, there is little difference between Configurations 5 through 8 and the corresponding configurations that do not have SCPS-TP Header Compression enabled (Configurations 1 through 4, respectively).

Finally, let us consider the combined effect of SNACK and Timestamps on link utilization. Table 7 indicates that at low bit-error rates, there is little difference in link utilization between the configurations with SNACK, with Timestamps, and with both. However, at high bit-error rates, we see that the use of SNACK results in higher link utilization than does the use of Timestamps, and that the combination of SNACK and Timestamps is only slightly better than SNACK by itself. Figure 16 confirms this graphically. Configurations 2 and 4 both have the SNACK capability enabled. Configurations 3 and 4 both have Timestamps enabled. Configurations 2 and 4 are almost indistinguishable from each other (Configuration 4 shows slightly higher link utilization than does Configuration 2 at bit-error rates greater than $2x10^{-5}$), and both of them show link utilization significantly above that of Configuration 3.

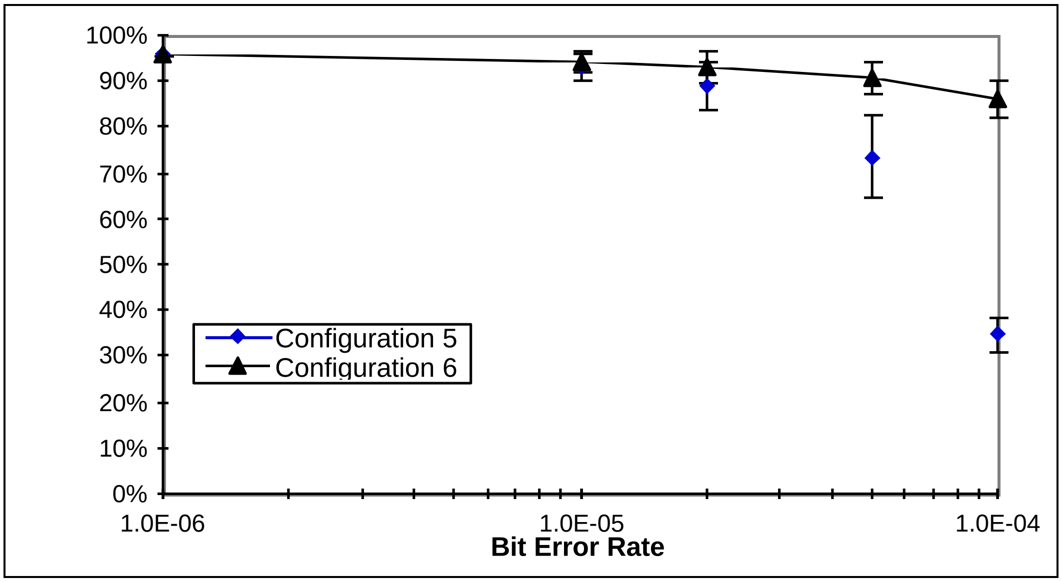


Figure 14. Link Utilization of SCPS-TP Configurations 5 and 6

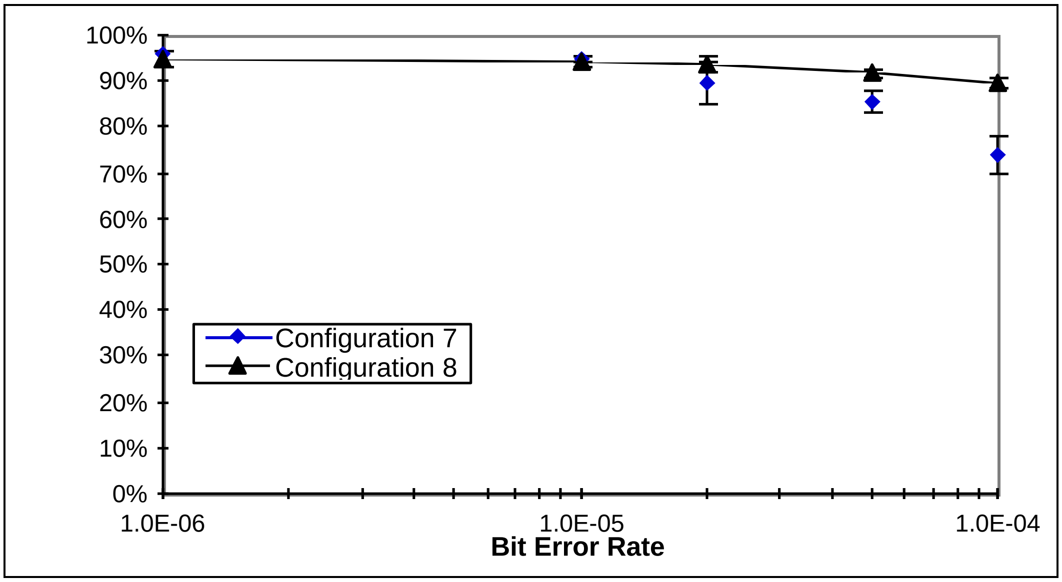


Figure 15. Link Utilization of SCPS-TP Configurations 7 and 8

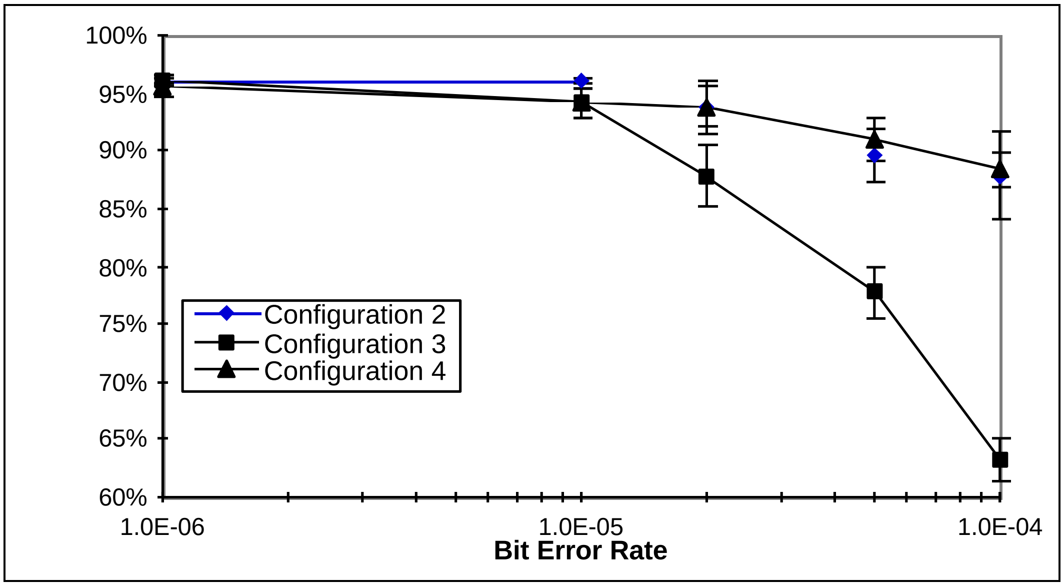


Figure 16. Effect of the Interaction Between SNACK and Timestamps on Link Utilization

6.1.3 Bit-Efficiency Results

Bit efficiency measures the transmission overhead imposed by a protocol or a set of protocols. Here, we measure the combined bit-efficiency of SCPS-TP and the CCSDS Telemetry and Telecommand protocols. Bit efficiency is measured as the amount (in bytes) of user data transferred divided by the total number of bytes transmitted by both sides in order to accomplish that transfer. It is expressed as a percentage, with 100% bit-efficiency indicating that there was no (protocol header) overhead involved in moving the user's data. Retransmission data count as protocol overhead, so we expect that the bit-efficiency of a retransmission protocol will drop as the bit-error rate increases. Since bit-efficiency measures protocol overhead, it can be affected by several factors. One of these factors is packet size. The STRV maximum packet size is 90 bytes. This is a relatively small packet, and the relative size of the SCPS-TP and CCSDS headers is large. Another factor that affects bit-efficiency is the rate at which acknowledgments are generated. Since all protocol overhead (acknowledgments as well as data packet headers and retransmission packets) is considered in the bit-efficiency calculation, the rate at which we generate acknowledgments affects bit-efficiency. Our acknowledgment rate was approximately one acknowledgment per round trip.

Summary

We see from Table 8 that there is remarkable consistency in the bit-efficiency results across all bit-error rates. The variation in bit-efficiency is between 28% and 30%, and the effect of SNACK on bit-efficiency is negligible. The Timestamps capability affects bit-efficiency negatively because the Timestamps option is large. The Header Compression capability contributes positively to the variation in bit-efficiency, because it makes the SCPS-TP headers smaller. There is essentially no interaction between any of the capabilities, and almost no experimental error.

Table 8. Allocation of Variation in Bit Efficiency

BER	Variation	SNACK	Timestamps	Compr.	SN,TS	SN,CM	TS,CM	All	Error
1E-6	28%	0%	-33%	65%	0%	0%	1%	0%	0%
1E-5	29%	0%	-32%	65%	0%	0%	1%	0%	1%
2E-5	29%	0%	-32%	65%	0%	0%	1%	0%	2%
5E-5	29%	2%	-29%	66%	0%	0%	1%	0%	2%
1E-4	30%	2%	-34%	61%	0%	0%	1%	0%	2%

Experimental Data

Figure 17 through Figure 20 show graphs of bit-efficiency versus bit-error rate for each of the eight configurations. All are plotted with the same y-axis for ease of comparison among graphs. All configurations show a decrease in bit-efficiency as the bit-error rate increases. The figures show that SNACK has little effect on bit-efficiency, that Timestamps negatively affect bit-efficiency, and that Header Compression positively affects bit-efficiency. (Recall that even-numbered configurations have the SNACK capability enabled; that Configurations 3, 4, 7, and 8 have Timestamps enabled, and that Configurations 5 through 8 have Header Compression enabled.)

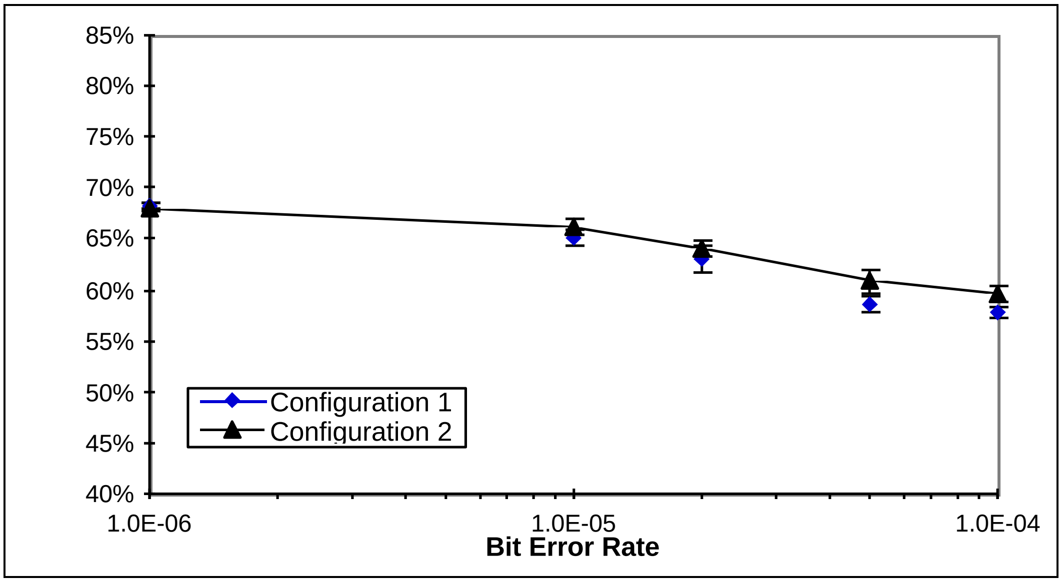


Figure 17. Bit Efficiency of SCPS-TP Configurations 1 and 2

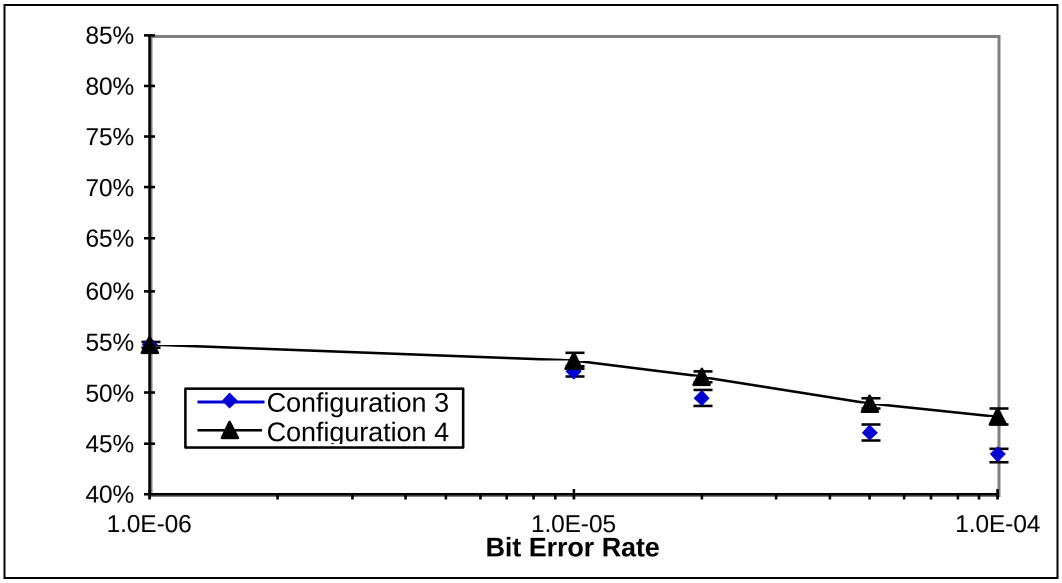


Figure 18. Bit Efficiency of SCPS-TP Configurations 3 and 4

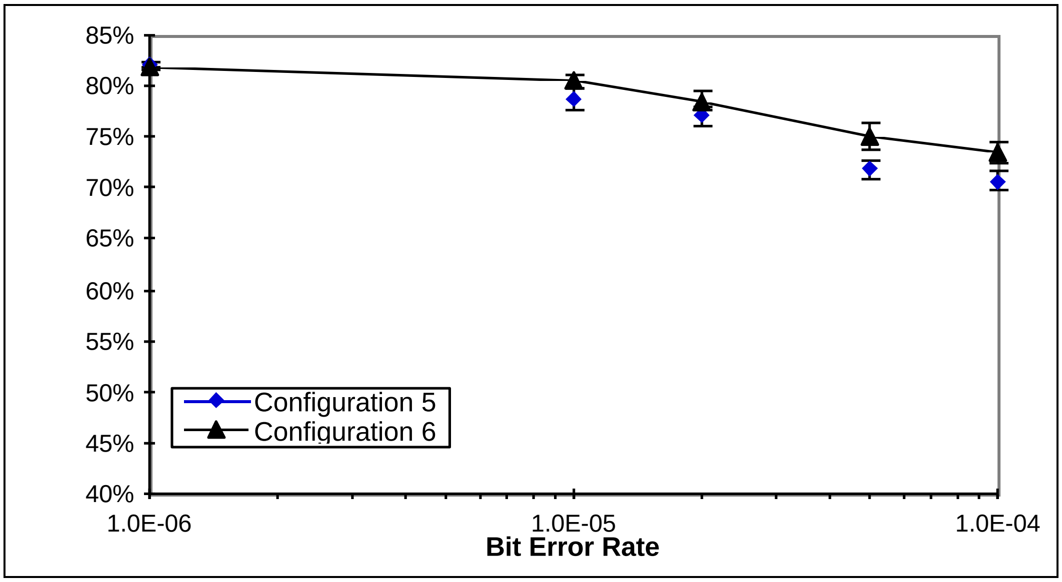


Figure 19. Bit Efficiency of SCPS-TP Configurations 5 and 6

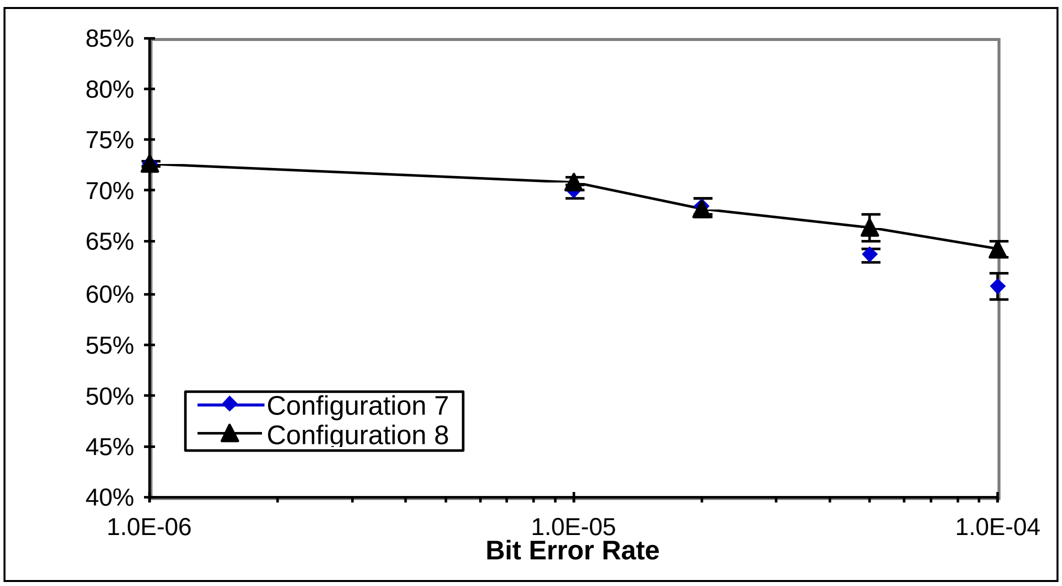


Figure 20. Bit Efficiency of SCPS-TP Configurations 7 and 8

6.1.4 Conclusions Based on Laboratory Testing

The results from the laboratory testing support the following conclusions:

1. The SNACK capability significantly improves throughput and link utilization at high bit-error rates, has no negative effects on throughput or link utilization at low bit-error rates, and has no impact on bit-efficiency.
2. The TCP Timestamps capability has a negative effect on throughput at low bit-error rates. It has a strongly negative effect on bit-efficiency, and a moderately positive effect on link utilization. When used in combination with SNACK, throughput is lower than when using SNACK alone, but link utilization is improved slightly. (Note that the magnitude of the negative effect of TCP Timestamps on throughput is exaggerated by the small packet size imposed by the STRV. With larger packet sizes, this effect is mitigated.)
3. The SCPS-TP Header Compression capability has a significant, positive effect on throughput at bit-error rates of 5×10^{-5} and below. Header Compression improves bit-efficiency at all bit-error rates, and has no effect on link utilization. (The positive effect of Header Compression on throughput is exaggerated by the small packet size imposed by the STRV in the same manner that the negative effect of the TCP Timestamps is, above. As with TCP Timestamps, the effect of Header Compression on throughput will diminish as the packet size increases.)

Recommendations Based on Laboratory Testing

An STRV-like communication environment can be characterized by low data rates, small packet sizes, and potentially high error rates, with bursty errors. These environments are best served by the use of the SCPS-TP Header Compression and SNACK capabilities, and *not* by the use of TCP Timestamps.

As packet sizes and (bi-directional) link data rates increase, the need for SCPS-TP Header Compression decreases.

6.2 Field Experiment Results

The data gathered in the laboratory testing allows us to make predictions about the performance of SCPS-TP when operated in the field. We present an overview of the prediction method, followed by the throughput, link utilization, and bit-efficiency results obtained from the field testing compared to the predicted results. For each type of result, we examine the conclusions drawn above and determine whether the field test data confirm those conclusions.

6.2.1 Predicted Results Based on Laboratory Tests

We can predict the mean response and confidence intervals for any combination of the three capabilities tested in the laboratory, using the laboratory results as a basis. As a part of calculating the allocation of variation in Table 6, Table 7, and Table 8, we built a regression model of the response. We can use this model, presented in Appendix A, to predict performance in the field. We present the predicted throughput performance for Configuration 8 in Figure 21, below. Predictions such as these will serve as the point of reference for plotting the Field Test results. Using these predictions as a guide, we can easily see whether the test results deviate from our expectations. We also present the Field Test results for link utilization and bit-efficiency in the context of the predicted responses for each.

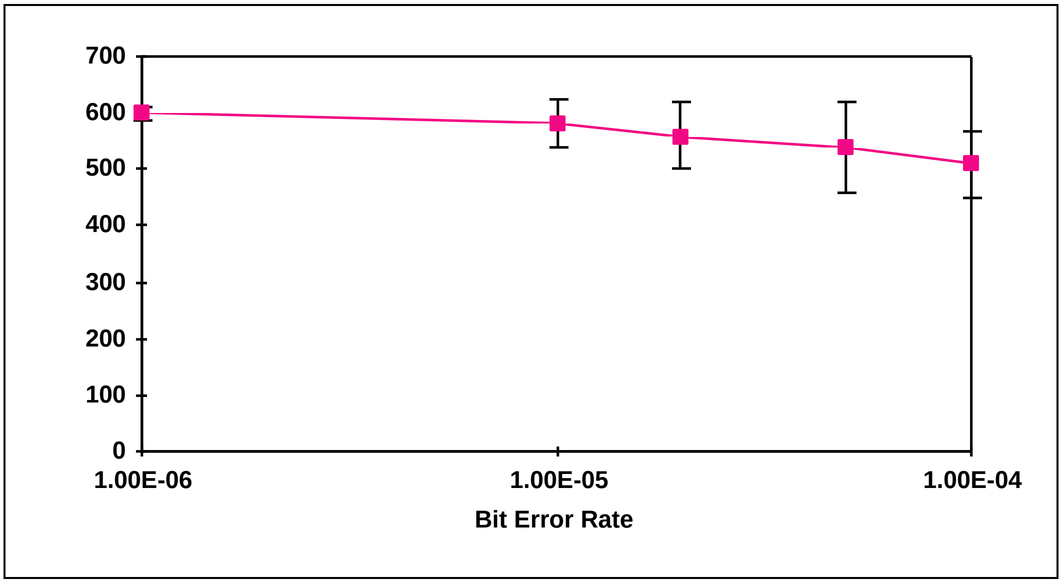


Figure 21. Predicted Throughput for Configuration 8

6.2.2 Field Results

This section presents the results of the field tests. Throughput, link utilization, and bit-efficiency results are each presented in separate sub-sections. Each subsection presents the results for the five configurations that were tested onboard the STRV. Results are shown as individual data points on graphs that also present the predicted responses. For each subsection, we summarize the results, present a discussion of the results where appropriate, and then draw conclusions.

6.2.2.1 Throughput Results

This section presents the user data throughput results for each of the SCPS-TP configurations tested. The throughput performance for each configuration is plotted on a separate graph, with the predictions based on the laboratory data plotted for reference.

Figure 22 shows the throughput of SCPS-TP Configuration 1 versus bit-error rate for the Field Test data (shown as individual data points) and the predicted results, which were based on the laboratory data. The “Predicted” line represents the mean of 10 runs at each of the following five bit-error rates: 10^{-6}, 10^{-5}, $2x10^{-5}$, $5x10^{-5}$, and 10^{-4}. The lines extending above and below each of the predicted values indicate a 90% confidence interval for the prediction. We see that throughput is down 9% from its maximum when the bit-error rate reaches 10^{-5}, and is down by 10% from the maximum at $2x10^{-5}$. From that point, the throughput falls steeply, down 38% from its maximum value when the bit-error rate reaches $5x10^{-5}$, and down by 67% at 10^{-4}.

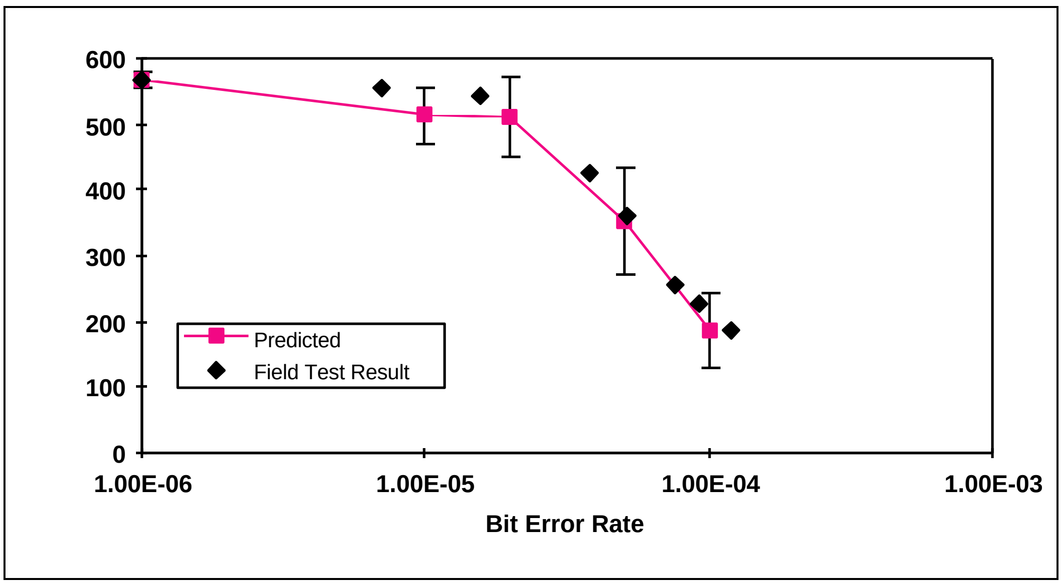


Figure 22. Predicted Versus Measured Throughput for Configuration 1

Q: Why do the confidence intervals widen between $2x10^{-5}$ and $5x10^{-5}$?

A: This widening of the confidence intervals reflects an increase in the range of results seen in the laboratory testing. The predicted confidence intervals are based on the laboratory results, and when there is a greater degree of variation in the laboratory throughput (at a given bit-error rate), the confidence interval of the predicted result reflects that increased range. For example, at 10^{-6}, the difference between the minimum and maximum throughput was approximately 12 bps, while at $2x10^{-5}$, the range was 105 bps. That degree of variability in the laboratory result means that we should expect some variability in the field result. However, we expect that an *average* of several tests at the same bit-error rate would approach the mean laboratory result.

Q: Is the fact that two points deviate from the prediction in the $5x10^{-6}$ to $2x10^{-5}$ range significant?

A: No, because, in fact, they *do not* deviate from the prediction. They are not exactly equal to the mean response, but they are within the predicted range of responses, as indicated by the nearby confidence intervals. (Actually, those confidence intervals only apply to the specific bit-error rate for which they were calculated. While it is legitimate to interpolate between the *mean* responses, it is not necessarily valid to do so between the confidence intervals.)

Conclusions:

There is good correspondence between the predicted results and the field test results for the throughput tests of Configuration 1. All data points are either very close to the interpolated prediction of mean response or are within the limits of nearby confidence intervals.

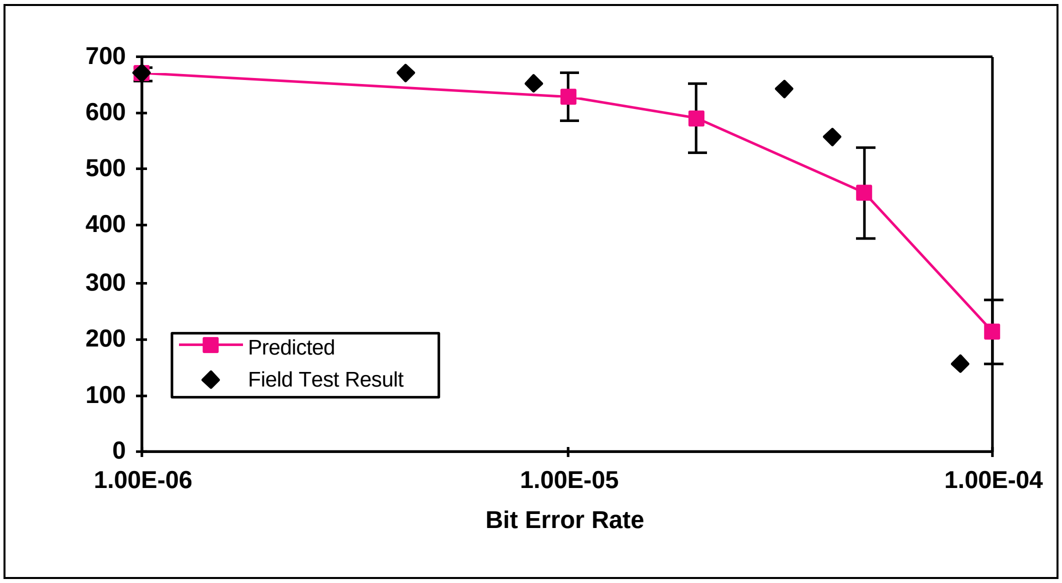


Figure 23. Predicted Versus Measured Throughput for Configuration 5

Figure 23 shows the throughput versus bit-error rate curve for Configuration 5. Configuration 5 has the Header Compression capability enabled, but not SNACK or TCP

Timestamps. The throughput is 3% below the maximum at a BER of $8x10^{-6}$, 5% below maximum at $3x10^{-5}$, 17% below maximum at $4x10^{-5}$, and 77% below maximum at $8.5x10^{-5}$.

Q: Why is the data point just to the right of $2x10^{-5}$ so high?

A: As mentioned previously, while one can reasonably interpolate between the mean response data points, interpolating between adjacent confidence intervals is probably invalid. The “knee” of the throughput curve appears to occur between $2x10^{-5}$ and $5x10^{-5}$, and our experience leads us to expect a significant degree of variability in throughput performance as the operating point approaches the knee of the throughput curve.

We examined the log files for the data point in question, and saw that everything went smoothly during the test. There were 16 packets that were corrupted during the run, and the “fast retransmit” capability allowed all of them to be retransmitted without waiting for the retransmission timer to expire. The sender interrupted its transmission of data only to interleave a retransmission; it never had to stop and wait. As a result, the throughput did not suffer significantly during the test. Had the errors occurred at other points during the run, it is possible that the sender’s buffers might have filled, causing the sender to have to wait until a retransmission timer expired (since this configuration does not use SNACK). However, this was not the case, and the run continued unhindered by the losses.

Q: Why is the data point just to the left of the 10^{-4} prediction so low?

A: In examining the log files for that run, we saw that several data packets were corrupted in close proximity to each other, including all three packets in one frame. The laboratory testing uses a Bernoulli process to simulate errors, rather than using a burst error model. The fast retransmit capability handles errors that follow a Bernoulli process fairly well, but is not particularly well suited to burst errors (the SNACK capability is, but is not enabled in Configuration 5). As a result, the protocol had to stop and wait for the retransmission timer to expire, which caused the throughput to be low.

Q: Why are there not more field test data points?

A: We conducted initial testing of SCPS-TP on the STRV between February and April of 1996. The performance data collected at that time was adversely affected by two implementation errors that we subsequently found and corrected. We collected a full suite of data for the corrected implementation in the laboratory, and were permitted to perform a limited amount of retesting on the satellite during July of 1996. The result of that retesting is presented here, since it is more indicative of the proper operation of the protocol than the previous results.

Conclusions:

There is generally good correspondence between the laboratory prediction and field test measurements of Configuration 5 throughput results. One test experienced better-than-expected throughput, as a result of "fortunate" spacing of the data losses. Another test suffered poorer-than-expected throughput, due to the burstiness of the errors during that particular test.

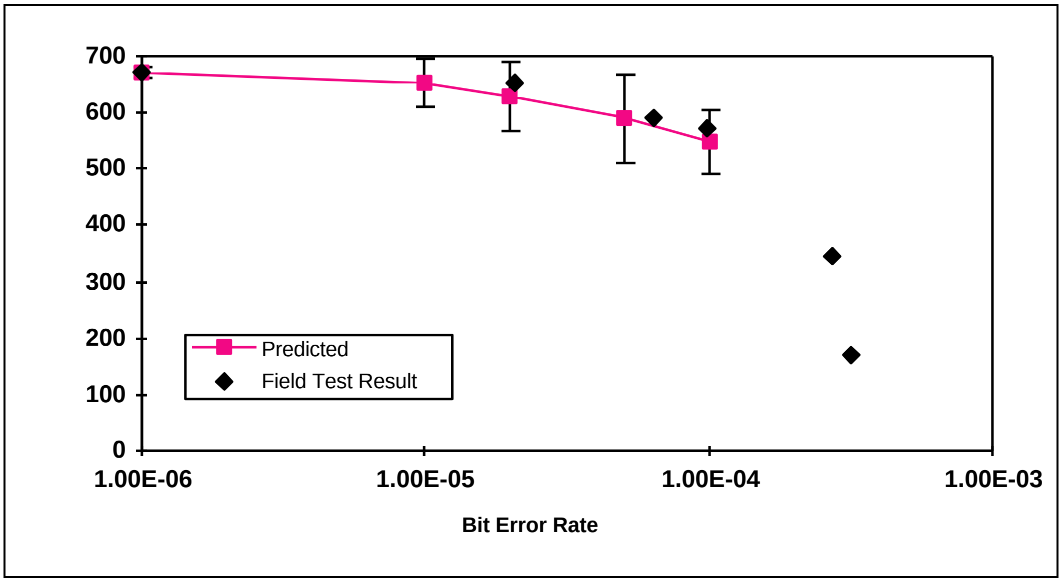


Figure 24. Predicted Versus Measured Throughput for Configuration 6

Figure 24 shows the throughput performance versus bit-error rate of SCPS-TP Configuration 6. Configuration 6 enables SCPS-TP Header Compression and SNACK, but not TCP Timestamps. The throughput is 3% below its maximum at the data point with a bit-error rate of 2.1×10^{-5}, 12% below maximum at 6.4×10^{-5},15% below maximum at 9.8×10^{-5}, 48% below maximum at 2.7×10^{-4}, and 74% below maximum at 3.2×10^{-4}.

Q: The data points between 10^{-4} and 10^{-3} seem to show a sharp fall-off in throughput. Is such a sharp drop reasonable?

A: A fall in throughput is reasonable, although we noticed from an analysis of those test runs that SCPS-TP was retransmitting more data than necessary. Additional retransmission traffic will reduce throughput by "crowding out" new user data. The additional retransmissions were the result of the "fast retransmit" policy being set more aggressive than necessary. We are investigating more appropriate settings for fast retransmit in long-delay environments.

Q: How high a bit-error rate could the protocol sustain?

A: There are at least two ways to answer that question. The first is practical - if we simply extrapolate the throughput line to the point on the x-axis where we reach zero throughput, we would expect that the maximum bit-error rate to be approximately $4x10^{-4}$.

However, if we wish to determine theoretically the point at which the protocol will actually *fail*, we can do so. (We might wish to know this if we have a particularly noisy channel and information that is more important to communicate reliably than immediately.) The point at which the protocol gives up is determined by the user's setting for the *maximum retransmissions* parameter. This parameter determines how many times a single packet will be retransmitted before declaring that the connection has failed. (It for example, the maximum retransmissions parameter is set to 10 retransmissions, and *every packet* on the connection is retransmitted 9 times, the connection will not fail, and progress will continue, albeit very slowly. On the other hand, if no packets have been retransmitted, but then one packet requires more than 10 retransmissions, the connection will be aborted and an error message reported to the user.)

Appendix A contains the derivation and equations to support this calculation. However, for a test with 622 packets, each 90 bytes long, as we ran here, and a maximum number of transmissions equal to 10, there is a 99.9% probability that the configuration could sustain a bit-error rate of $4.25x10^{-4}$. Note that if the run is longer (more unique packets), the maximum bit-error rate is lower. For example, if we increase the length of the run to 1000 packets, the maximum BER is $4.02x10^{-4}$. If we revert to the original 622-packet run, but increase the maximum retransmissions to 50, the maximum sustainable bit-error rate increases to $2.01x10^{-3}$. However, if very many packets must be retransmitted more than a few times each, throughput will be *very* low.

Conclusions:

There is very good correspondence between the predicted throughput performance and the field test measurements of throughput for Configuration 6. All data points fall close to the interpolated mean response prediction, and all are well within nearby confidence intervals.

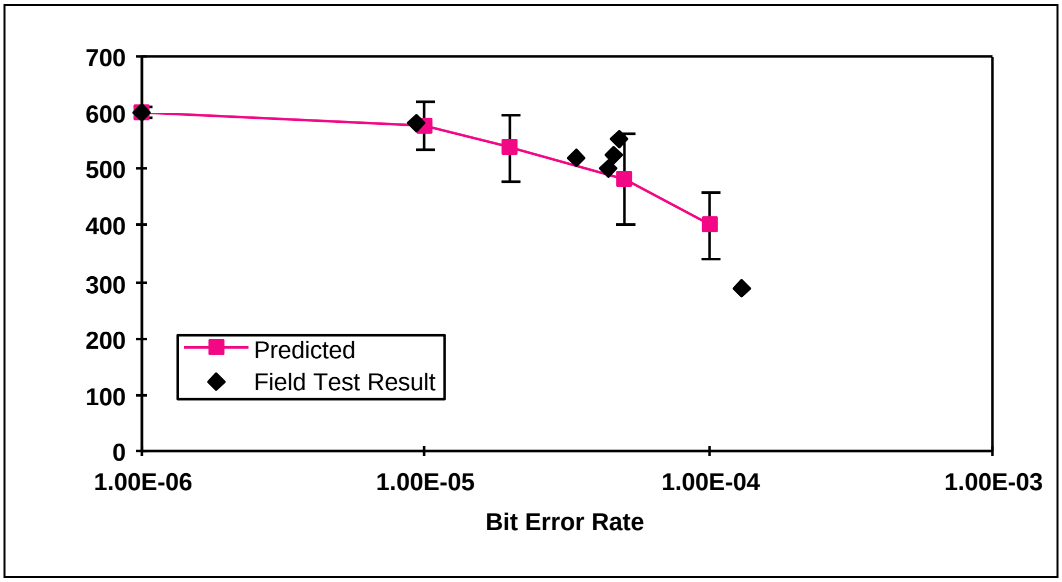


Figure 25. Predicted Versus Measured Throughput for Configuration 7

Figure 25 shows the predicted and measured throughput versus bit-error rate for Configuration 7, which has Header Compression and TCP Timestamps enabled, but not SNACK. We see very close correspondence to the predicted mean response at 10^{-6}, 10^{-5}, $3x10^{-5}$, and $4x10^{-5}$. Two data points between $4x10^{-5}$ and $5x10^{-5}$ are higher than the mean response, but still within the confidence interval for $5x10^{-5}$.

Conclusions:

The throughput performance for the field test of Configuration 7 corresponds well with the laboratory predictions. All data points (at bit-error rates less than 10^{-4}) either fall very close to the interpolated mean predicted response or are within a nearby confidence interval.

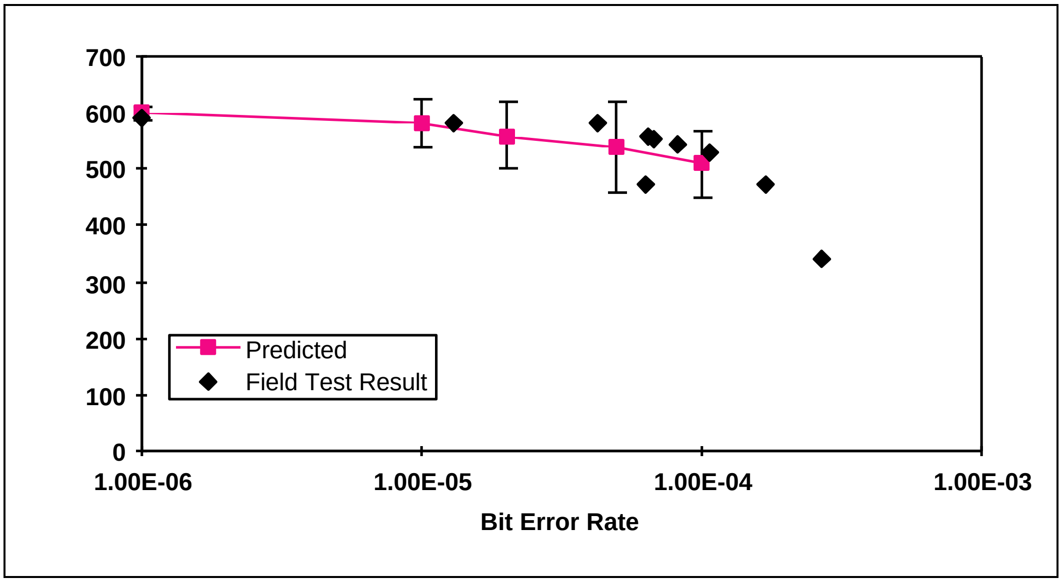


Figure 26. Predicted Versus Measured Throughput for Configuration 8

Figure 26 shows the predicted and measured throughput versus bit-error rate for Configuration 8, which has Header Compression, TCP Timestamps, and SNACK enabled. The data points between 10^{-5} and $5x10^{-5}$ show a 2% drop in throughput compared to the maximum; the data points between $5x10^{-5}$ and 10^{-4} show, in order of increasing bit-error rate, 20%, 6%, 7%, and 8% throughput reduction from the maximum throughput measured in the field test. The data points beyond 10^{-4} show a 10%, 20%, and 43% decline in throughput, in order of increasing bit-error rate.

Q: Why is the throughput for the data point between $5x10^{-5}$ and 10^{-4} so low?

A: In examining the packet log for this run, we see that two consecutive frames were lost at the end of the run. These frames contained the final five data packets and the FIN (end of connection) packet. Since no other data was forthcoming from the sender, the receiver did not know that data had been lost, and therefore could not send a SNACK. The six missing packets each had to time out and be retransmitted individually. The increased time due to timeout and retransmission caused the drop in throughput.

Q: The throughput of the field test data appears to be (relatively) consistently above predicted mean response for bit-error rates between $5x10^{-5}$ and 10^{-4}. Why is this?

A: The laboratory environment does not exactly match the field environment. One difference is that the STRV 1b spacecraft has a synchronous, frame-oriented downlink while the OBC in the laboratory does not. (The OBC supports a CCSDS packet interface, but none of the framing or clocking - the output is an asynchronous RS-232 port that operates at 9600 bps.) The lack of synchrony in the laboratory means that the Spanner program can build a persistent queue of downlink packets, since we drive the downlink at its maximum data rate. This queue means that some packets take longer to reach the "ground" in the laboratory than they do in the field. When a packet is lost, SCPS-TP sends duplicate acknowledgments plus, possibly, a SNACK, to indicate the loss. When the sending SCPS-TP receives a number of duplicate acknowledgments, it assumes that the packet has been lost and retransmits it (this technique is called "fast retransmit," since it does not depend on a timer expiration for retransmission). Consider the effect of a long queue of packets in the downlink: the retransmission must wait behind several other downlink packets in queue. During its wait, more duplicate acknowledgments are sent, which may result in additional retransmissions of the packet. These additional retransmissions that were caused by queuing in Spanner on the downlink caused the mean throughput in the laboratory to be slightly lower than in the field.

Conclusions:

The field test results for Configuration 8 correspond well with the laboratory tests, with all data points falling within the bounds of nearby confidence intervals. The field test results appear to have slightly higher throughput performance than the predicted mean performance for bit-error rates between $5x10^{-5}$ and 10^{-4}. We believe that this is due to differences in the queuing behavior between the actual field test environment and the laboratory environment.

This concludes the presentation of the throughput results from the field test. At this point, let us examine the throughput-related conclusions put forth at the end of the laboratory testing section and identify which are and are not supported by the field test results:

1. *The SNACK capability significantly improves throughput* and link utilization *at high bit-error rates, has no negative effects on throughput* or link utilization *at low bit-error rates,* and has no impact on bit-efficiency.

   Configuration 5 and Configuration 6 differ only in the fact that Configuration 6 has the SNACK option. The field experiment throughput performance is identical for these two configurations in the absence of bit-errors, and essentially similar up to bit-error rates of approximately $3x10^{-5}$ (see Figure 27). Beyond $3x10^{-5}$, however, the throughput for Configuration 6 exceeds the throughput for Configuration 5. Further, the Configuration 6 data points at $2.7x10^{-4}$ and $3.2x10^{-4}$ both showed throughput results that were higher than that of the Configuration 5 test at $8.4x10^{-5}$. On the basis of these results, we consider conclusion 1 to be confirmed with respect to throughput.

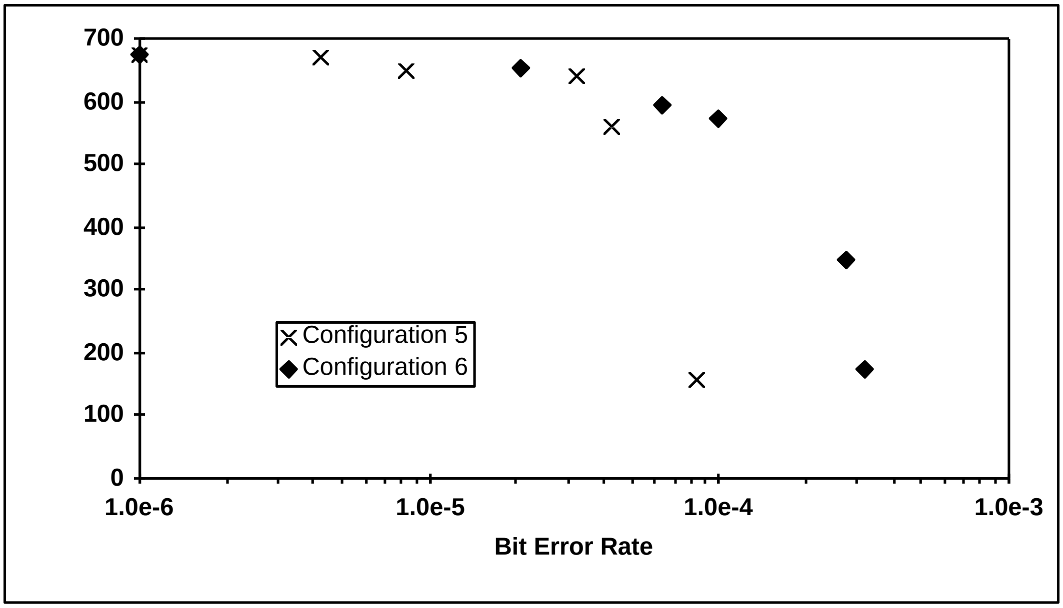


Figure 27. Comparison of Throughput Results for Configuration 5 and Configuration 6

2. *The TCP Timestamps capability has a negative effect on throughput at low bit-error rates.* It has a strongly negative effect on bit-efficiency, and a moderately positive effect on link utilization. *When used in combination with SNACK, throughput is lower than when using SNACK alone*, but link utilization is improved slightly.

   The best throughput obtained from Configuration 7, which has TCP Timestamps enabled, was 603 bps. Configuration 5, which differs only from Configuration 7 in that it does not have TCP Timestamps enabled, had a maximum throughput of 674 bps. The conclusion regarding the negative effect of TCP Timestamps on throughput at low bit-error rates is confirmed.

   Refer to Figure 28. For the range of bit-error rates over which the laboratory data was taken (10^{-6} through 10^{-4}), every field data point for Configuration 6 (SNACK but no Timestamps) exhibits higher throughput than the field data for Configuration 8 (SNACK and Timestamps). For this range of bit-error rates, the conclusion is confirmed with respect to throughput. (Beyond 10^{-4}, the throughput results for both configurations are similar, indicating the possibility of eventual convergence of throughput results. However, the throughput curve at these bit-error rates appears to have passed its “knee”, and the point of convergence may be zero.)

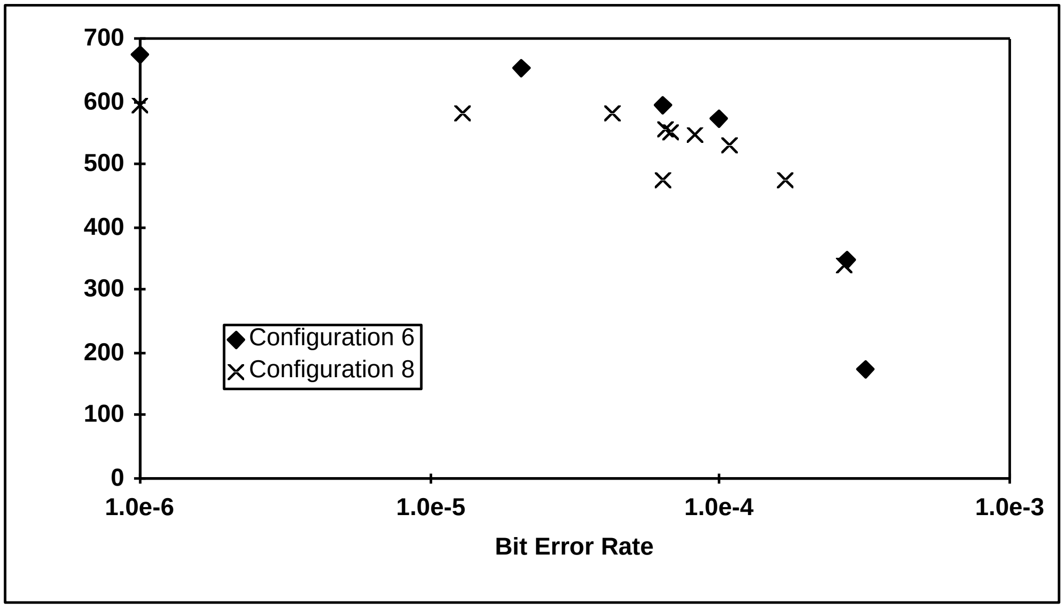


Figure 28. Comparison of Throughput Results for Configuration 6 and Configuration 8

3. *The SCPS-TP Header Compression capability has a significant, positive effect on throughput at bit-error rates of $5x10^{-5}$ and below.* Header Compression improves bit-efficiency at all bit-error rates, and has no effect on link utilization.

   Between the bit-error rates of 10^{-6} and $5x10^{-5}$, every field data point of Configuration 5 (Header Compression) exhibits higher throughput than the field data for Configuration 1 (no Header Compression). Typical differences between similar bit-error rates exceed 100 bps. We consider this conclusion to be confirmed with respect to throughput.

6.2.2.2 Link Utilization Results

. Link utilization is a measure of the ability of the protocol to "keep the pipe full" when there is data ready to be transmitted. This ability is important in space communication, in which contact times may be limited. The protocol should not allow the link to be idle for significant periods of time. This section presents the link utilization results for the data channel for the SCPS-TP configurations tested in the flight test. Each configuration is presented on a separate graph, with the laboratory prediction plotted for reference. We summarize the results, present a discussion of the results when appropriate, then draw conclusions.

Figure 29 presents the link utilization results of Configuration 1 versus bit-error rate, for both the field test data and the predictions based on the laboratory tests. Note that the confidence interval for the prediction at 2×10^{-5} exceeds 100% link utilization. Clearly, link utilization in excess of 100% cannot occur. The prediction results from the variability of the laboratory data, and the fact that the statistical technique used to generate the prediction does not consider the fact that the maximum possible value is 100%.

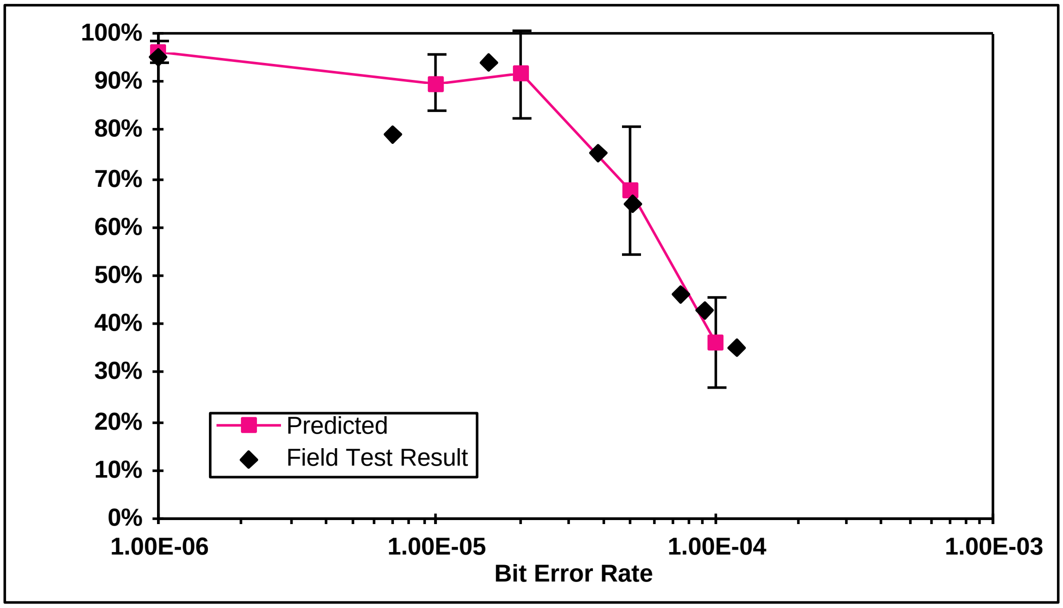


Figure 29. Predicted Versus Measured Link Utilization for Configuration 1

Q: For the data point just to the left of the 10^{-5} prediction, why is the link utilization so low?

The STRV satellites are in a geostationary transfer orbit, with potentially very long visibility periods. However, signal quality is often too poor to productively use, a result of antenna modifications that were required just before launch. The STRV operators have devised a system in which they send pairs of time-tagged commands to turn on the transmitter and then turn it off ten minutes later. These pairs are typically spaced once per hour during visibility periods. At the time that the transmitter is scheduled to come on, the operators scan for the spacecraft. If the link quality is adequate, the operators can load a command to turn the transmitter back on immediately after it automatically turns itself off. Generally, there is an imperceptibly small delay between the time-tagged off command and the subsequent command to turn the transmitter back on. In some cases, however, there were several seconds that elapsed before the transmitter came back on. The data point in question was affected by such a delay. The transmitter went off just after the very last data packet of the run was received, but before the connection was closed. As a result, the throughput (which is measured only over the data transfer portion of the connection) was unaffected, but link utilization (which is measured over the entire connection) suffered from the several-seconds delay in turning the transmitter back on and the subsequent need to retransmit the connection close request.

Q: Why does the laboratory prediction of link utilization appear to increase between 10^{-5} and $2x10^{-5}$?

While seven of the ten laboratory test runs at 10^{-5} had link utilization values in the mid-90% range, three of the runs had values significantly lower, reducing the mean link utilization to 90%. At $2x10^{-5}$, the mean utilization was 92%. However, an examination of the confidence intervals for the two ranges indicates that there is not a statistically significant difference between the two values.

Conclusions:

With the exception of one test that was affected by a problem with turning the onboard transmitter back on, all data points correspond very well with the laboratory predictions.

Figure 30 presents the link utilization results versus bit-error rate for Configuration 5. Note the similarity between this graph and the corresponding throughput results, shown in Figure 23. For a configuration that does not have the SNACK capability, we expect this correspondence to hold, as it does here. For configurations with the SNACK capability, we generally expect that link utilization will not fall off as much at high bit-error rates, due to the ability of the SNACK option to help keep the data channel loaded. Past a certain bit-error rate, however, even SNACK-equipped configurations will experience retransmission time outs, causing a drop in link utilization.

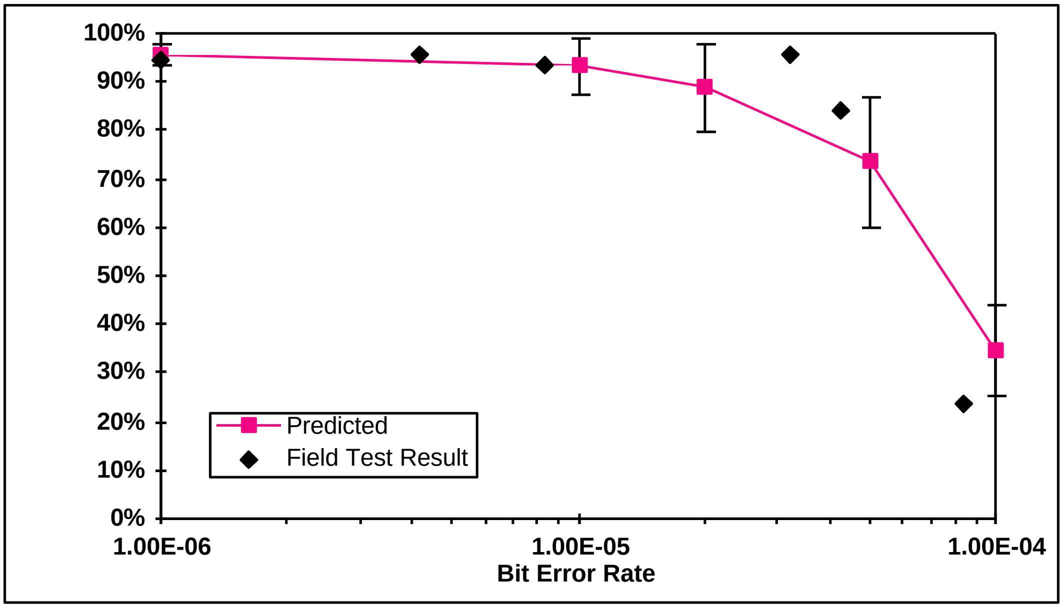


Figure 30. Predicted Versus Measured Link Utilization for Configuration 5

Q: Why is the link utilization for the data point to the right of the $2x10^{-5}$ prediction so high?

A: In our discussion of Figure 23, we noted that the test in question experienced all of its losses during situations in which the fast retransmit capability could cause a retransmission of the packet without the sender having to stop and wait for a retransmission time out. The same good fortune that helped with throughput has a correspondingly beneficial effect on link utilization.

Q: Why is the link utilization for the right-most field test data point so low?

A: Again, the correspondence with the throughput result from Figure 23 holds. The error distribution was such that retransmission time outs occurred, reducing both throughput and link utilization. The laboratory tests did not use a burst model for errors, hence the discrepancy.

Conclusions:

There is generally good correspondence between the predicted link utilization results and the field test results. The same two data points that did not correspond well in the discussion of the associated throughput results deviate from the predictions here, as well. One test received the benefit of fortunately-spaced errors. The other suffered from an error distribution that was different than that in the laboratory environment.

Figure 31 presents predicted and measured link utilization versus bit-error rate for Configuration 6. Note that at 10^{-4}, the predicted link utilization is still above 80%, while in both previous configurations it had fallen to below 40% at 10^{-4}. The reason for this difference is the presence of the SNACK option. The SNACK option allows the receiver to identify and request immediate retransmission for missing data. The use of SNACK reduces the occurrence of retransmission time outs, improving both link utilization and throughput.

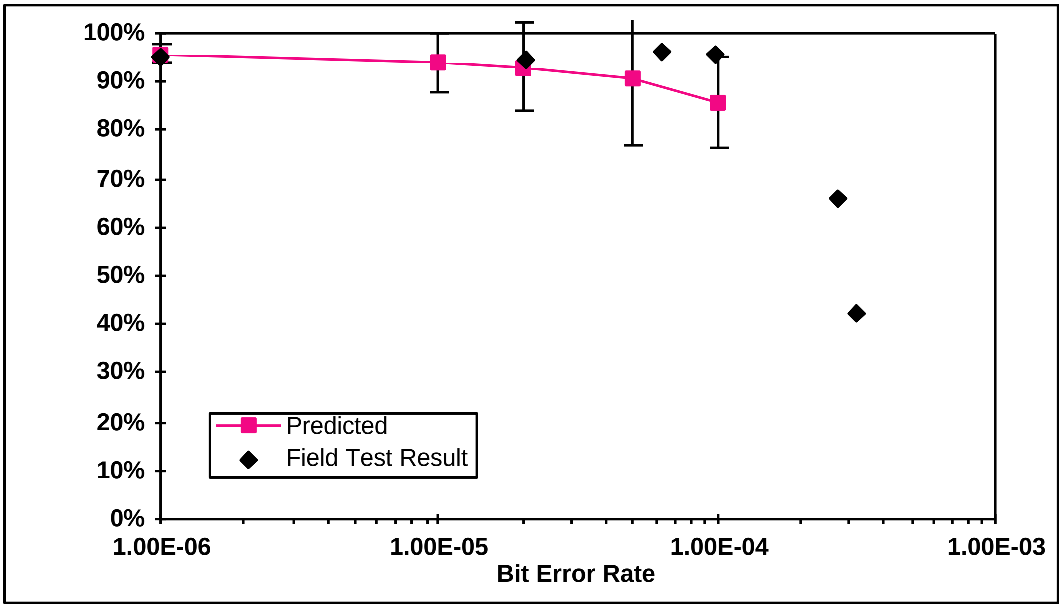


Figure 31. Predicted Versus Measured Link Utilization for Configuration 6

Q: The link utilization for the two data points between 5×10^{-5} and 10^{-4} seems higher than predicted. Is there any reason for this other than normal variability between runs?

A: Yes. In examining the log files from these runs, we noticed that a configuration parameter (the maximum segment size) was incorrectly configured. The configuration error, present in all Configuration 6 field tests, caused more data to be retransmitted than necessary at high bit-error rates. This configuration error was not present in the laboratory tests, and the means by which the field configuration changed from the laboratory configuration is still under investigation. However, the additional retransmission data caused the link utilization to be higher than expected, without a significant effect on throughput.

Conclusions:

The field data points at 2×10^{-5} and below show very good correspondence with the laboratory predictions. At bit-error rates above 5×10^{-5}, a configuration error in the field configuration resulted in unnecessary retransmissions, which made link utilization higher than expected. The configuration error has been identified, but the cause is still under investigation.

Figure 32 presents predicted and measured link utilization versus bit-error rate for Configuration 7 (Timestamps enabled). All field test data at bit-error rates below 10^{-4} either match the predicted values very closely, or are well within nearby confidence intervals. Note that the link utilization of the laboratory prediction is above 70% at 10^{-4}. This value is not as high as Configuration 6 (with SNACK), but is much higher than those configurations that do not have Timestamps or SNACK.

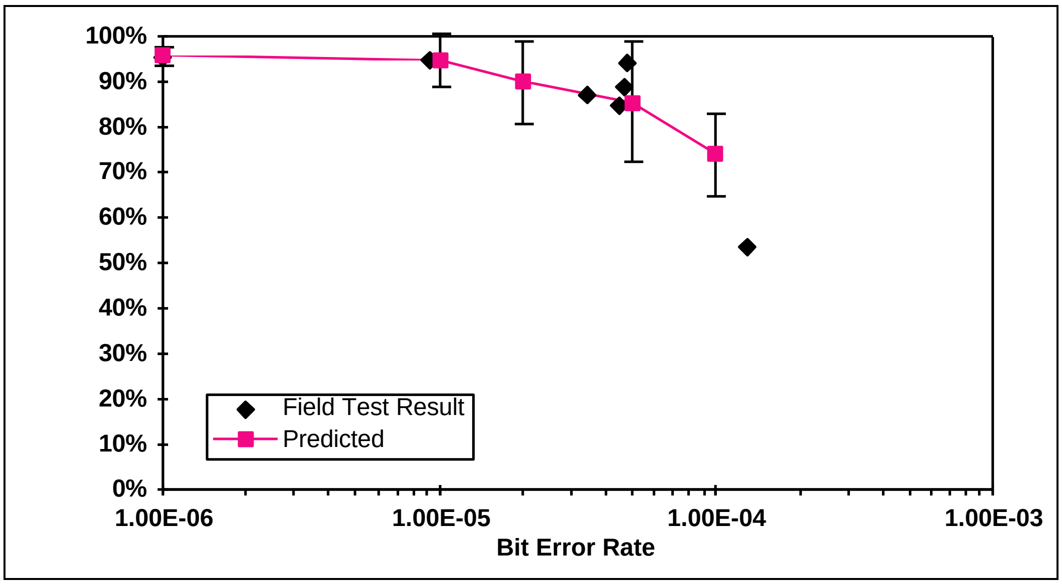


Figure 32. Predicted Versus Measured Link Utilization for Configuration 7

Conclusions:

The link utilization performance for the field test of Configuration 7 corresponds well with the laboratory predictions. All data points (at bit-error rates less than 10^{-4}) either fall very close to the interpolated mean predicted response or are well within a nearby confidence interval.

Figure 33 shows the predicted and measured link utilization versus bit-error rate for Configuration 8. Note that the predicted value of link utilization is above 90% at 10^{-4}, better than any other configuration tested in the field. This is consistent with the prediction based on the Allocation of Variation presented in Table 7, which indicated that the combination of SNACK and Timestamps would improve link utilization more than either alone. We see that the field test data bears this out: All data points are within nearby confidence intervals. The data point that is below the prediction (between $5x10^{-5}$ and 10^{-4}) was discussed in the presentation of Figure 26: the final two frames of the connection were lost, requiring the connection to wait for six expirations of the retransmission timer. The resulting delay reduced both throughput and link utilization.

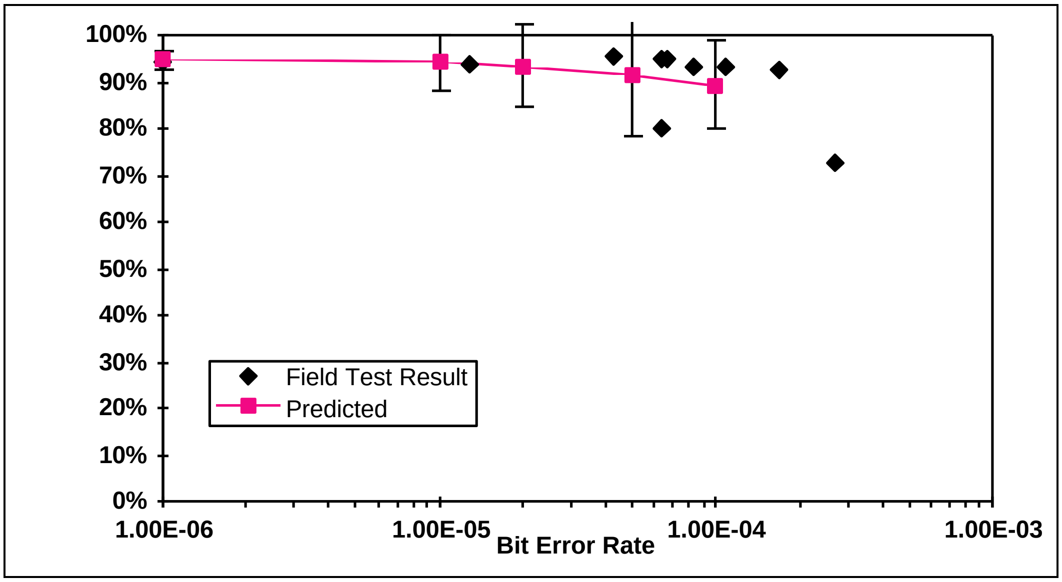


Figure 33. Predicted Versus Measured Link Utilization for Configuration 8

Conclusions:

There is good correspondence between the predicted and measured values of link utilization for Configuration 8. All data points are within nearby confidence intervals.

This concludes the presentation of the link utilization results. At the end of the laboratory results, the following conclusions were drawn regarding link utilization:

1. *The SNACK capability significantly improves* throughput and *link utilization at high bit-error rates, has no negative effects on* throughput or *link utilization at low bit-error rates*, and has no impact on bit-efficiency.

   The configuration error mentioned in the discussion of Figure 31 affected the link utilization results of Configuration 6 (the configuration with SNACK enabled) at bit-error rates between $2x10^{-5}$ and $5x10^{-5}$. This error prevents us from confirming this conclusion with respect to link utilization based on the field testing.

2. *The TCP Timestamps capability has* a negative effect on throughput at low bit-error rates. It has a strongly negative effect on bit-efficiency, and *a moderately positive effect on link utilization. When used in combination with SNACK*, throughput is lower than when using SNACK alone, but *link utilization is improved slightly*.

   Figure 34 shows the link utilization results of Configuration 5 (no Timestamps) and Configuration 7 (Timestamps enabled). The link utilization performance of the two configurations is consistent until the bit-error rate exceeds $3x10^{-5}$, at which point the link utilization for Configuration 7 is consistently better than that of Configuration 5. However, there is not a strong enough trend for either configuration to be able to confirm the first part of the conclusion. We cannot confirm the second part of the conclusion due to the configuration error that affected Configuration 6, mentioned previously.

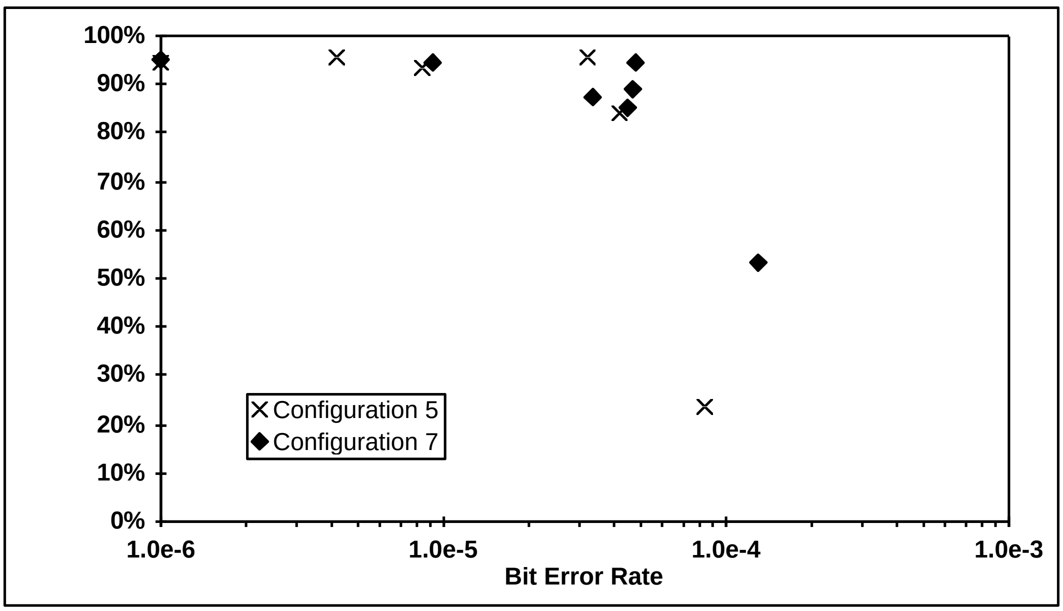


Figure 34. Link Utilization of Configuration 5 and Configuration 7

3. *The SCPS-TP Header Compression capability* has a significant, positive effect on throughput at bit-error rates of $5x10^{-5}$ and below. Header Compression improves bit-efficiency at all bit-error rates, and *has no effect on link utilization.*

   Configuration 1 has Header Compression disabled, while Configuration 5 has it enabled. Figure 35 shows the link utilization results of the two configurations. (Recall that the Configuration 1 data point at approximately $8x10^{-6}$ has a low link utilization resulting from an onboard transmitter problem - it should not be considered in the comparison.) There is enough uncertainty about the point at which the link utilization for each configuration begins to decline that we can neither confirm nor refute the conclusion.

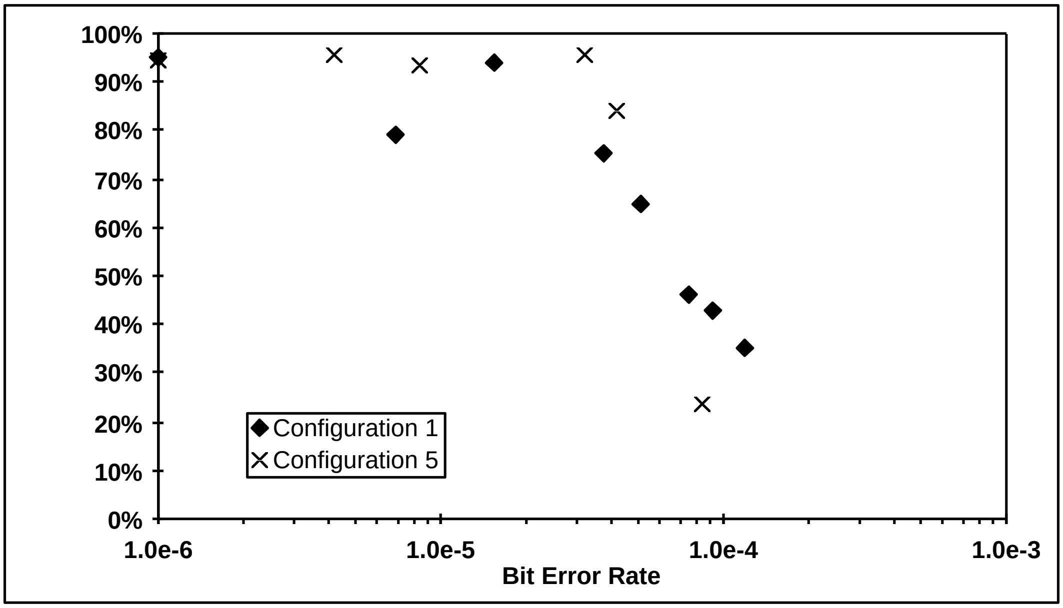


Figure 35. Link Utilization of Configuration 1 and Configuration 5

6.2.2.3 Bit-Efficiency Results

An examination of bit-efficiency is important if SCPS-TP is to be operated over restricted capacity links, which may be sensitive to the amount of protocol overhead present. This section presents the bit-efficiency results for the SCPS-TP configurations tested in the flight test. Each configuration is presented on a separate graph, with the laboratory prediction plotted for reference.

In our discussion of Figure 26, we noted a difference in the laboratory and field environments that had to do with the synchronous nature of the STRV 1b downlink versus the asynchronous nature of the OBC in the laboratory. This difference caused the Spanner program to build queues of downlink packets in the laboratory tests. The effect of this difference on throughput and link utilization was relatively minor - the field results tended to be slightly higher than the laboratory results, but still well within the confidence intervals resulting from the predictions. However, the effect on bit-efficiency is more pronounced, especially at mid-range bit-error rates (between $2x10^{-5}$ and 10^{-4}). Recall that the “fast retransmit” behavior is triggered by the data sender receiving a certain number of duplicate acknowledgments (i.e., acknowledgments that are not advancing the acknowledgment number). This is the primary means of triggering a retransmission when the SNACK capability is not in use, as it results in retransmissions sooner than if the sender waited for the retransmission timer to expire. However, consider the case in which a long queue of packets has built within Spanner: the retransmission, when triggered, is queued behind several other packets. The receiver has not yet received the retransmission, so it continues to send duplicate acknowledgments, which result in additional (unnecessary) retransmissions of the packet that is in queue. The net effect is that the bit-efficiency of the laboratory tests is lowered, due to the additional packets (both acknowledgments and retransmissions) being sent. This situation did not occur in the field, but in other field configurations it could. To address this, the SCPS-TP may either be operated with the congestion control capability enabled (the purpose of the congestion control capability is to prevent the formation of such queues), or the rate control settings may be adjusted to something slightly below the maximum capacity of the link. Note that at very high bit error rates (approaching and above 10^{-4}), the discrepancy between laboratory and field results diminishes. This is because the configurations that depend on fast retransmission begin to experience retransmission time outs at high bit-error rates. These time outs cause the data channel to occasionally become idle, which has the side-effect of draining off any queue that has built within the Spanner program.

Figure 36 through Figure 40 present the bit-efficiency results for the SCPS-TP configurations tested in the field. Since the results are affected by the intrinsic difference between the laboratory and field environments discussed above, the commentary on each configuration is limited.

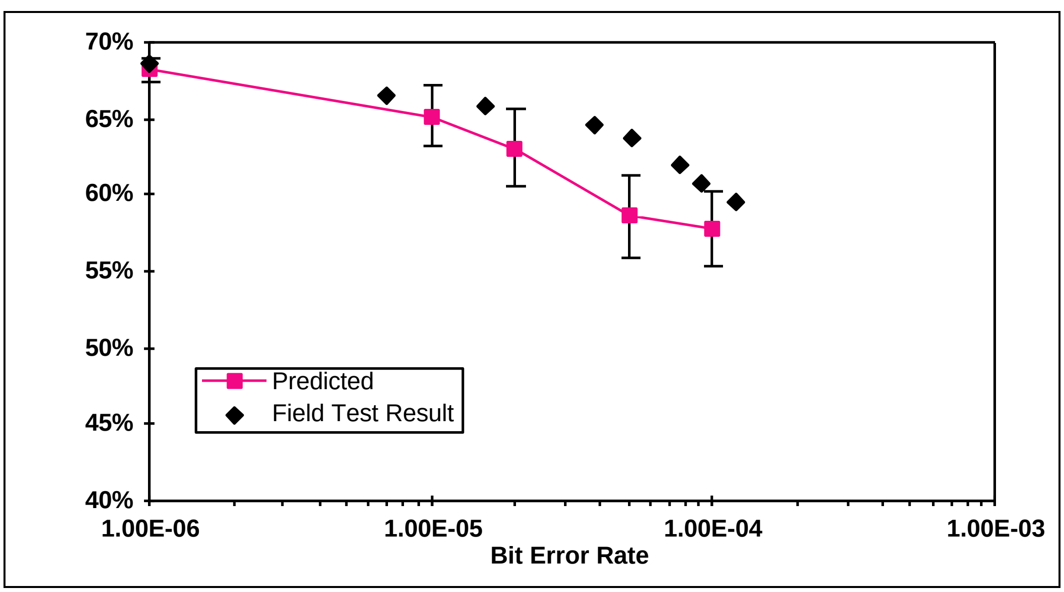


Figure 36. Predicted Versus Measured Bit Efficiency for Configuration 1

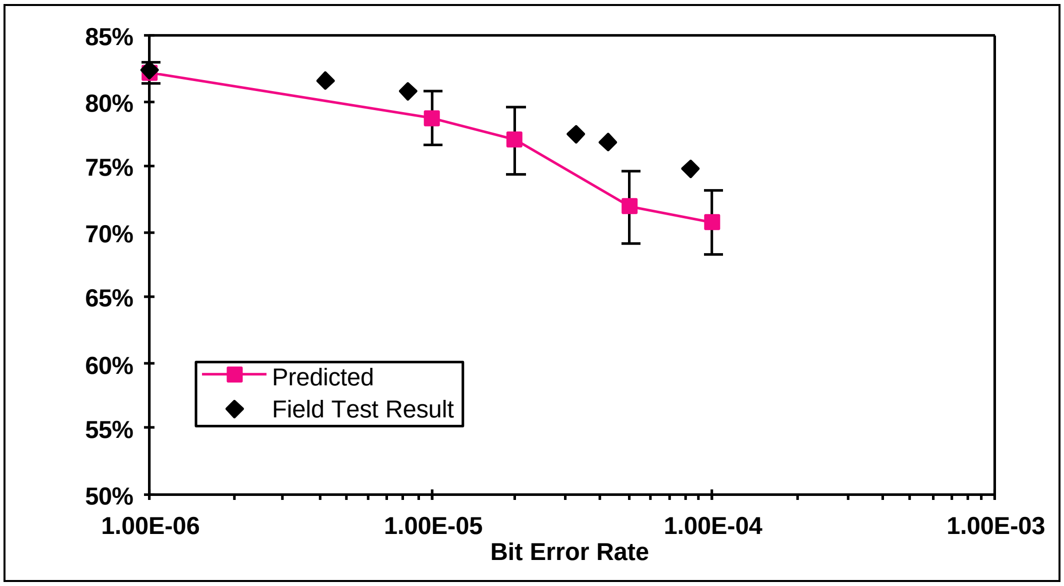


Figure 37. Predicted Versus Measured Bit Efficiency for Configuration 5

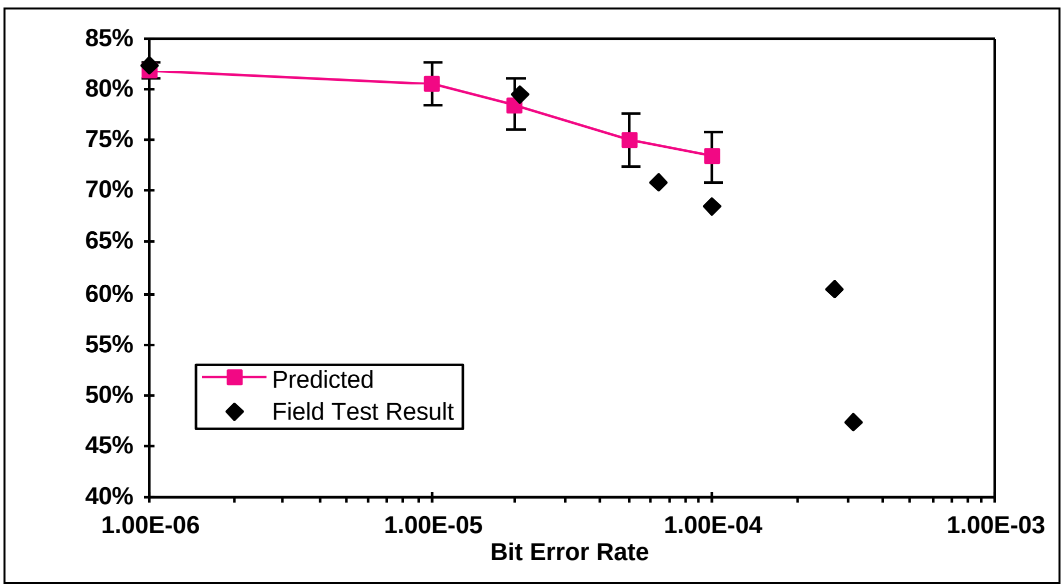


Figure 38. Predicted Versus Measured Bit Efficiency for Configuration 6

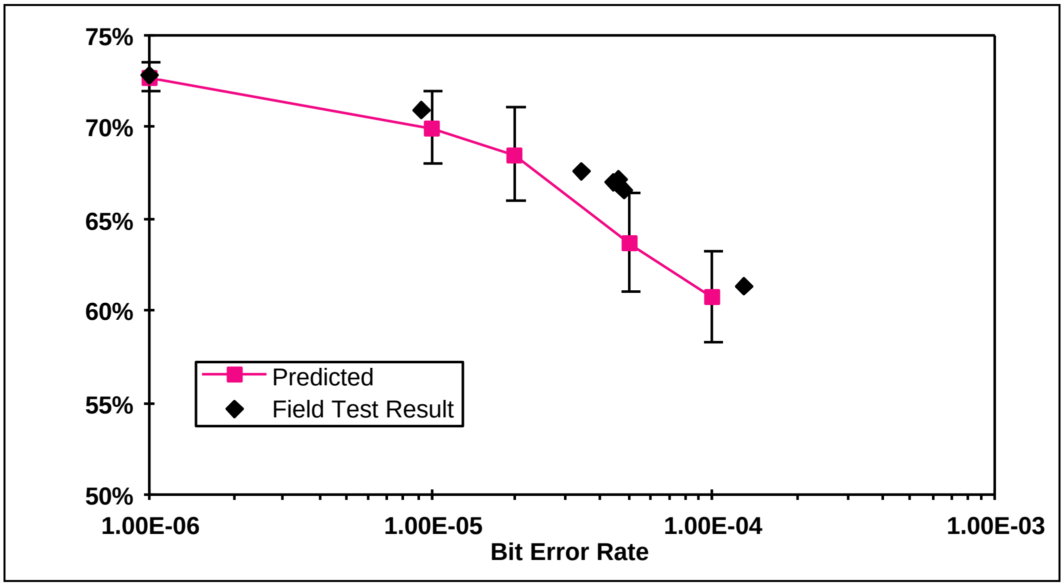


Figure 39. Predicted Versus Measured Bit Efficiency for Configuration 7

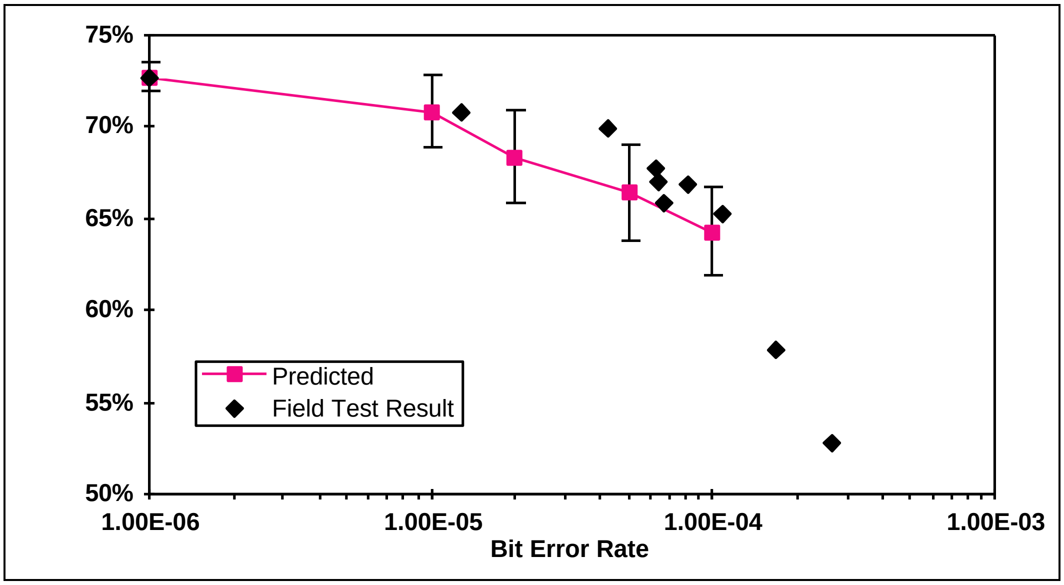


Figure 40. Predicted Versus Measured Bit Efficiency for Configuration 8

Q: In general, the field bit-efficiency results have been higher than expected. Why, in Figure 38, are the two data points between $5x10^{-5}$ and 10^{-4} *lower* than the prediction?

A: Recall the discussion of Figure 31, in which these same two data points are considered. As a result of a configuration error, some invocations of the SNACK capability caused data to be retransmitted unnecessarily. These unnecessary retransmissions reduced the bit-efficiency for the data points in question.

Conclusions:

An intrinsic difference between the field and laboratory configurations resulted in the bit-efficiency of the field data to generally be higher than predicted in the laboratory. This affects our ability to use correspondence with the laboratory results as the basis for confirmation of the conclusions.

At the end of the laboratory results, the following conclusions were drawn regarding bit-efficiency:

1. *The SNACK capability* significantly improves throughput and link utilization at high bit-error rates, has no negative effects on throughput or link utilization at low bit-error rates, and *has no impact on bit-efficiency*.

   As a result of the incorrect configuration, we can neither confirm nor refute this conclusion based on the field test results.

2. *The TCP Timestamps capability* has a negative effect on throughput at low bit-error rates. It *has a strongly negative effect on bit-efficiency*, and a moderately positive effect on link utilization. When used in combination with SNACK, throughput is lower than when using SNACK alone, but link utilization is improved slightly.

   The results of the field testing of bit-efficiency of Configuration 5 (no TCP Timestamps) were shown in Figure 37 to vary between approximately 82% (at 10^{-6}) and 75% (at $8x10^{-5}$). The bit-efficiency of Configuration 7 (TCP Timestamps enabled) varies between 73% and 61% over roughly the same range of bit-error rates (refer to Figure 39). This confirms conclusion 2.

3. The SCPS-TP Header Compression capability has a significant, positive effect on throughput at bit-error rates of $5x10^{-5}$ and below. *Header Compression improves bit-efficiency at all bit-error rates*, and has no effect on link utilization.

   The field test results of bit-efficiency for Configuration 5 (Header Compression enabled) varies between 82% and 75% over the bit-error rate range of 10^{-6} to $8x10^{-5}$. The field test results for Configuration 1 (no Header Compression) vary between 69%

and 62% over approximately the same range of bit-error rates. This confirms conclusion 3.

Section 7

Conclusions and Recommendations

The SCPS Transport Protocol testing portion of the SSFE was a valuable step in the overall test program for SCPS. The test provided us with experience and an understanding of what to expect when integrating SCPS-TP into operational satellites and control centers. Further, it provided us insight into the performance of SCPS-TP over links in which burst errors are not unusual, and the bit-error rates high. Finally, we gained insight into the operation of the protocol in resource-constrained environments.

The objectives of the transport protocol portion of the SSFE were as follows:

- to gain experience in hosting SCPS-TP on an actual spacecraft and
- to examine the performance of SCPS-TP when running over a real space/ground data link.

We met those objectives. The process of hosting the SCPS-TP protocol onto the STRV was a difficult one, primarily due to the limited availability of C-language development tools for the MIL-STD-1750A processor. The generally poor quality of development tools led to our failure to discover and correct two implementation errors, which rendered invalid the results of the first set of test data that we took. We were able to conduct a limited amount of retesting, which was used to confirm the results we gathered in the laboratory.

7.1 Conclusions

The SCPS-TP protocol appears to be well-suited to the long-delay, potentially high bit-error rate environment of the STRV. All configurations were able to sustain connections at bit-error rates of 10^{-4} and yield throughput in excess of 130 bps (17% of maximum possible). The Selective Negative Acknowledgment (SNACK) capability was principally responsible for the ability of the SCPS-TP to operate well in high bit-error rate environments (the configuration with the SNACK capability enabled showed only a 15% drop from maximum throughput at a bit-error rate of approximately 10^{-4}). The SCPS-TP Header Compression accounted for an 18% increase in throughput over the configurations that did not use Header Compression at zero bit error rate.

The following conclusions follow from the laboratory testing and confirmed by the flight test results:

1. The SNACK capability significantly improves throughput at high bit-error rates, and has no negative effects on throughput at low bit-error rates.

2. The TCP Timestamps capability has a negative effect on throughput at low bit-error rates. It has a strongly negative effect on bit-efficiency. When used in combination with SNACK, throughput is lower than when using SNACK alone. (Note that the magnitude of the negative effect of TCP Timestamps on throughput and bit-efficiency is exaggerated by the small packet size imposed by the STRV. With larger packet sizes, this effect is mitigated.)
3. The SCPS-TP Header Compression capability has a significant, positive effect on throughput at bit-error rates of $5x10^{-5}$ and below. Header Compression improves bit-efficiency at all bit-error rates. (The positive effect of Header Compression on throughput is exaggerated by the small packet size imposed by the STRV in the same manner that the negative effect of the TCP Timestamps is, above. As with TCP Timestamps, the effect of Header Compression on throughput will diminish as the packet size increases.)

These conclusions were formed as a result of the laboratory testing, but could neither be confirmed nor refuted by the flight test results:

1. The SNACK capability significantly improves link utilization at high bit-error rates, and has no negative effects on link utilization at low bit-error rates, and has no impact on bit-efficiency.
2. The TCP Timestamps capability has a moderately positive effect on link utilization. When used in combination with SNACK, link utilization is improved slightly.
3. The SCPS-TP Header Compression capability has no effect on link utilization.

7.2 Recommendations

We document recommendations primarily directed at ourselves in Appendix C, Lessons Learned. The following recommendations are directed toward potential users of SCPS-TP and toward the sponsors of this effort.

1. Push ahead in the effort to standardize SCPS-TP and deploy it in environments that have similar delay and error characteristics to the STRV environment.
2. When using SCPS-TP in STRV-like environments, enable SNACK.

   SNACK has no negative effects when errors are not present, and is primarily responsible for the protocol's ability to sustain relatively high throughputs at high bit-error rates.
3. When using SCPS-TP in STRV-like environments, enable Header Compression.

   The Header Compression capability reduced the size of SCPS-TP headers, improving throughput and bit-efficiency. These effects were particularly dramatic because the maximum packet size of the STRV was small. As the packet size increases, the positive effect of Header Compression will diminish.

4. When using SCPS-TP in STRV-like environments, disable TCP Timestamps.

   The TCP Timestamps capability reduced throughput at low bit-error rates, and provided no significant improvement in throughput at high bit-error rates when SNACK was in use. As with Header Compression, the negative effects of TCP Timestamps are exaggerated by the small packet sizes on STRV.

5. Evolve the program of testing toward integrated tests.

   Although there are still specific SCPS-TP capabilities to be tested, the focus of future tests should be integrated-stack testing. Tests of individual protocol capabilities can be conducted either as part of integrated-stack testing or as a small, focused portion of a larger test. The SCPS-NP, which has not as yet undergone flight testing, will probably benefit from more substantial, focused testing. However, this can still be conducted in the context of an overall test.

List of References

1. Aho, Alfred V., Brian W. Kernighan, and Peter J. Weinberger, 1988, *The AWK Programming Language*, Addison-Wesley Publishing Company, Reading, Massachusetts.
2. L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New Techniques for Congestion Avoidance," *Proceedings of SIGCOMM 1994,* pp. 24-35, London, U. K., October 1994.
3. Çinlar, Erhan, 1975, *Introduction to Stochastic Processes*, Prentice-Hall, Inc., Englewood Cliffs, NJ.
4. Connolly, T., P. Amer, P. Conrad, November 1994, "An Extension to TCP: Partial Order Service," RFC 1693.
5. Cooper, G., March 1986, TCP implementation, IMAGEN Corporation.
6. Fortune, D., January 1996, "SSFE Ground Segment Interface Control Document," ESYS-95107-RPT-01, European Systems Limited, Guildford, Surrey, UK.
7. Fox, R., June 1989, "TCP Big Window and Nak Options," Internet Engineering Task Force document RFC 1106.
8. IETF, 1989, "Requirements for Internet Hosts - Communications Layers," Internet Engineering Task Force document RFC 1122.
9. Jacobson, V. and R. Braden, October 1988, "TCP Extensions for Long-Delay Paths," Internet Engineering Task Force document RFC 1072.
10. V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for Long-Delay Paths," Request for Comments 1323, IETF, May 1992.
11. Jain, Raj, 1991, *The Art of Computer Systems Performance Analysis-Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, New York, NY
12. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), January 1996, Report Concerning Space Communications Protocol Standards, Advanced Orbiting Systems Upper Layer Protocols: Rationale, Requirements and Application Notes , SCPS 710.0-G-0 (Draft), DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA
13. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), May 1996, "Space Communications Protocol Standards (SCPS) Bent-Pipe Experiment Report", SCPS-D71.51-Y-1, Colorado Springs, CO.

14. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), May 1996, “SCPS File Protocol Report on the SCPS/STRV-1b Flight Experiment”, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

15. “Space Communications Protocol Standards Security Protocol (SCPS-SP) FY96 Report”, September 1996, Sparta, Inc., Columbia, MD.

16. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), April 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS Network Protocol (SCPS-NP) Specification, SCPS 713.0, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

17. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), April 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS Security Protocol (SCPS-SP) Specification, SCPS 7135.0-C-1, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

18. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), March 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS Transport Protocol (SCPS-TP) Specification, SCPS 714.0, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

19. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), June 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS File Transfer Protocol (SCPS-FP) Specification, SCPS 717.0, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

20. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), September 1993, Recommended Development of Interoperable Data Communications Standards for Dual-Use by US Civil and Military Space Projects, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

21. Libes, Don, 1995, *Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs*, O’Reilly & Associates, Inc., Sebastopol, CA

22. Packet Telemetry, Blue Book, CCSDS 102.0-B-3, Issue 3 November 1992, or later issue.

23. Stevens, W. R., 1994, *TCP/IP Illustrated, Volume 1: The Protocols*, Reading, Massachusetts: Addison-Wesley

24. Telecommand—Part 1 Channel Service, Blue Book, CCSDS 201.0-B-1, January 1987, or later issue.

25. Telecommand—Part 2 Data Routing Service, Blue Book, CCSDS 202.0-B-2, January 1987, or later issue.

26. Telecommand—Part 2.1 Command Operation Procedures, Blue Book, CCSDS 202.1-B-1, October 1991, or later issue.

27. Telecommand—Part 3 Data Management Service, Blue Book, CCSDS 203.0-B-1, January 1987, or later issue.

28. Turner, J., October 1986, "New Directions in Communications (Or Which Way to the Information Age?)," *IEEE Communications* , Vol. 24, No. 10, 8-15.

29. Welch, Brent B., 1995, *Practical Programming in Tcl and Tk*, Prentice Hall PTR, Upper Saddle River, NJ.

30. Wells, N., August 1994, “The Space Technology Research Vehicles STRV-1a and -1b: First In-Orbit Results,” Paper presented at the 8th Annual AIAA/Utah State University Conference on Small Satellites, Utah State University, Logan, UT.

Glossary

ABR	Available Bit Rate
Ack	Acknowledgment
AFSCN	Air Force Satellite Control Network
API	Application Programming Interface
APID	Application Process Identifier
ATM	Asynchronous Transfer Mode
awk	Aho, Weinberger, Kernighan
BER	Bit-Error Rate
BMDO	Ballistic Missile Defense Organization
bps	bits per second
BSD	Berkeley Software Distribution
CCSDS	Consultative Committee for Space Data Systems
CLCW	Command Link Control Word
CLTU	Command Link Transmission Unit
COTS	Commercial Off The Shelf
CRCs	Cyclic-Redundancy Codes
DAU	Data Acquisition Unit
DLU	Down Link Unit
DOD	Department of Defense
DRA	Defence Research Agency
DSN	Deep Space Network
ESA	European Space Agency
GMT	Greenwich Mean Time
GTO	Geostationary Transfer Orbit
HQ	Headquarters

IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
JPL	Jet Propulsion Laboratory
kB	kilo byte (1024 bytes)
kbps	kilobit per second (1000 bits per second)
kg	kilogram
LNA	Linear Amplifier
MIL STD	Military Standard
MSS	Maximum Segment Size
MTR	MITRE Technical Report
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communication Network
OBC	Onboard Computer
OBDH	Onboard Data Handling System
PAWS	Protect Against Wrapped Sequence Numbers
PC	Personal Computer
RAM	Random Access Memory
RF	Radio Frequency
RFC	Request for Comments
ROMs	Read-Only Memories
RSL	Received Signal Level
RTS	Remote Tracking Station
RTT	Round Trip Time
Rx	Receiver
SCPS	Space Communications Protocol Standards
SCPS-FP	Space Communications Protocol Standards - File Protocol

SCPS-NP	Space Communications Protocol Standards - Network Protocol
SCPS-SP	Space Communications Protocol Standards - Security Protocol
SCPS-TP	Space Communications Protocol Standards - Transport Protocol
SCPS-TWG	Space Communications Protocol Standards - Technical Working Group
SIGCOMM	(Association for Computing Machinery) Special Interest Group on Communications
SACK	Selective Acknowledgment
SDIB	STRV Data Interchange Bus
SNACK	Selective Negative Acknowledgment
SSFE	SCPS-STRV Flight Experiment
STRV	Space Technical Research Vehicle
SWS	Silly Window Syndrome
SYN	Synchronize
Tcl	Tool Control Language
TCP	Transmission Control Protocol
Tk	Tcl X Windows Toolkit
Tx	Transmitter
UK	United Kingdom
ULU	Up Link Unit
USSPACECOM	United States Space Command

Distribution List

Internal

W159
R. C. Durst (10)
D. E. Ernst
G. J. Miller
R. C. Mitchell
A. S. Reitzel
N. L. Schult
D. Thomson
M. J. Zukoski

D048
G. M. Comparetto
B. A. Feldmeyer
J. G. McIlnay
J. Muhonen
R. Ramirez
E. J. Travis (Gemini Industries),
c/o MITRE Reston M/S W650

Project

USSPACECOM, J4
Attn: Col. Flynn
250 S. Peterson Blvd.
Peterson Air Force Base
Colorado Springs, CO 80914-3310

USSPACECOM, J4P
Attn: Kathleen Dismukes
250 S. Peterson Blvd.
Peterson Air Force Base
Colorado Springs, CO 80914-3310

SMC/ADC
Attn: Capt. J. Gadway
SMC/ADC
Los Angeles AFB, CA 90009-2960
(5 copies + disk in DOS format)

SMC/AX
Attn: Maj. J. Cole
SMC/AXES
Los Angeles AFB, CA 90009-2960
(original + 4 copies + disk in DOS format)

Mr. John Rush
HQ NASA Code OI
300 E. Street S.W.
Washington, DC 20546
(5 copies + disk)

USA CECOM
ATTN: AMSEL-PE-FC
Ft. Monmouth, NJ 07703-5027
(Letter of Transmittal only)

JPL M/S 301-235
Attn: Mr. Adrian Hooke
4800 Oak Grove Drive
Pasadena, CA 91109-8099
(5 copies + disk)

External

DISA/JITC
Attn: Mr. Leo Hansen
DISA/JITC
Ft. Huachuca, AZ 85613-7020

Mr. Nick Shave
Space Systems
Defence Research Agency
R16 Bldg.

Farnborough, Hampshire
GU146TD, U.K.
(5 copies)

LT CDR Kevin Shaw
HMS Northumberland
BFPO 345
United Kingdom

Mr. Steven R. Sides
SAIC
The Stonybrook School
Route 25a and Cedar Street
Stonybrook, NY 11790

Mr. Howard Weiss
SPARTA, Inc.
9861 Broken Land Parkway
Suite 300, Columbia, MD 21046

MTR 96W0000075V02

MITRE TECHNICAL REPORT

Final Report of the SCPS-TP Testing on the UK DRA STRV

Volume 2: Appendixes

September 1996

Robert C. Durst
Darrell E. Ernst
Gregory J. Miller
Eric J. Travis

Sponsor:	USSPACECOM/J4 HQ NASA/CODE OI	**Contract No.:**	F-19628-94-C-0001 DAAB0796-C-E601
Dept. No.:	D048 W159	**Project No.:**	6380 8S06

The views, opinions, and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

This document was prepared for authorized distribution only. It has not been approved for public release.

Washington C^3 Center
McLean, Virginia

Project 6380 Approval:

MITRE Department Approval: original signed by
Gary M. Comparetto
Associate Department Head

MITRE Project Approval: original signed by
Gary M. Comparetto
Associate Project Leader

Project 8S06 Approval:

MITRE Department Approval: original signed by
Duncan Thomson
Technical Area Manager

MITRE Project Approval: original signed by
Randolph C. Mitchell
Project Leader

Abstract

This paper documents the results of the SCPS Transport Protocol (SCPS-TP) portion of the SCPS/STRV Flight Experiment (SSFE). This experiment involved hosting the SCPS-TP protocol on the STRV 1b spacecraft and testing the operation of the protocol between the space-based endpoint and the ground. The communication environment imposed round-trip delays of approximately 8 seconds, error rates from 0 to > 10^{-4}, and very low speed links (1000 bps from space to ground, and 125 bps from ground to space). The experiment examined the effects on throughput, link utilization, and bit-efficiency of the TCP Timestamps capability, a Selective Negative Acknowledgment (SNACK) capability, and an end-to-end Header Compression capability. The paper presents the results of a factorial experiment conducted in a laboratory environment configured to simulate the flight test environment, then presents the results of those configurations from the factorial experiment that were tested in the flight environment. The experiment results show that SNACK and Header Compression greatly improve throughput, while the TCP Timestamps capability reduces throughput.

KEYWORDS: SCPS STRV TCP

Executive Summary

Purpose of This Document

The Space Communications Protocol Standards - Transport Protocol (SCPS-TP) is being developed by the joint NASA/DOD Space Communications Protocol project. This report documents the findings of the SCPS/STRV Flight Experiment (SSFE) SCPS-TP test.

Background

In the fall of 1992, NASA and the DOD jointly established a technical team (the SCPS Technical Working Group, or "SCPS-TWG") to explore possibilities for developing common space data communications standards. By the end of 1993 the team concluded that wide segments of the U.S. civil and military space communities have common needs for protocols to support in-flight monitoring and control of civil and military spacecraft. In 1994, the U.K. Defence Research Agency joined the SCPS-TWG with specific interoperability interests for the U.K. Skynet series of military communications satellites.

The program of work to develop these protocols includes specification, simulation, implementation, and testing. The SCPS/STRV Flight Experiment is the latest in a series of tests, that has included simulation, laboratory testing, and a bent-pipe test over a satellite link. The SCPS/STRV Flight Experiment was the first test to actually host the prototype software on a spacecraft, and was intended to evaluate performance and functionality in the anticipated implementation and operational environments.

The protocols tested in the SSFE include the SCPS File Protocol, the SCPS Transport Protocol, and the SCPS Security Protocol. All of the SCPS File Protocol testing made use of the SCPS Transport Protocol, and the SCPS Security Protocol testing used the SCPS Transport Protocol as its data source. The tests of the SCPS File Protocol and SCPS Security Protocol are documented separately (reference [14], [15]).

The SSFE was conducted between 2 January 1996 and 30 April 1996 and between 16 July 1996 and 31 July 1996. The SCPS-TP tests were conducted by U.K. Defence Research Agency personnel stationed at Lasham, England and at Farnborough, England, and by MITRE and Gemini Industries personnel at Reston, Virginia. The tests were conducted at Lasham, England and Reston, Virginia.

SSFE SCPS-TP Test Objectives

The objectives of the transport protocol portion of the SSFE were as follows:

- to gain experience in hosting SCPS-TP on an actual spacecraft and

- to examine the performance of SCPS-TP when running over a real space/ground data link.

In examining the performance of SCPS-TP, we tested three specific capabilities. The following list cites the primary benefits expected from each of the capabilities:

- TCP Timestamps: this capability improves SCPS-TP's estimate of round trip time, which can become distorted in error-prone environments.
- SCPS-TP Header Compression: this capability reduces protocol overhead by reducing the size of SCPS-TP headers.
- Selective Negative Acknowledgment: this capability improves SCPS-TP's error response by providing detailed information about missing or corrupted data.

We wished to determine the extent to which each of these capabilities affected performance at various bit-error rates. We also wished to determine if there were any significant interactions between the options that would restrict the ability of a user or program to pick the options individually.

We met the objectives stated above. The process of hosting the SCPS-TP protocol onto the STRV was a difficult one, primarily due to the limited availability of C-language development tools for the MIL-STD-1750A processor. The generally poor quality of development tools delayed our discovery and correction of two implementation errors. This rendered invalid the results of the first set of tests that we conducted. We were able to conduct a limited amount of retesting, which was used to confirm the results we gathered in the laboratory.

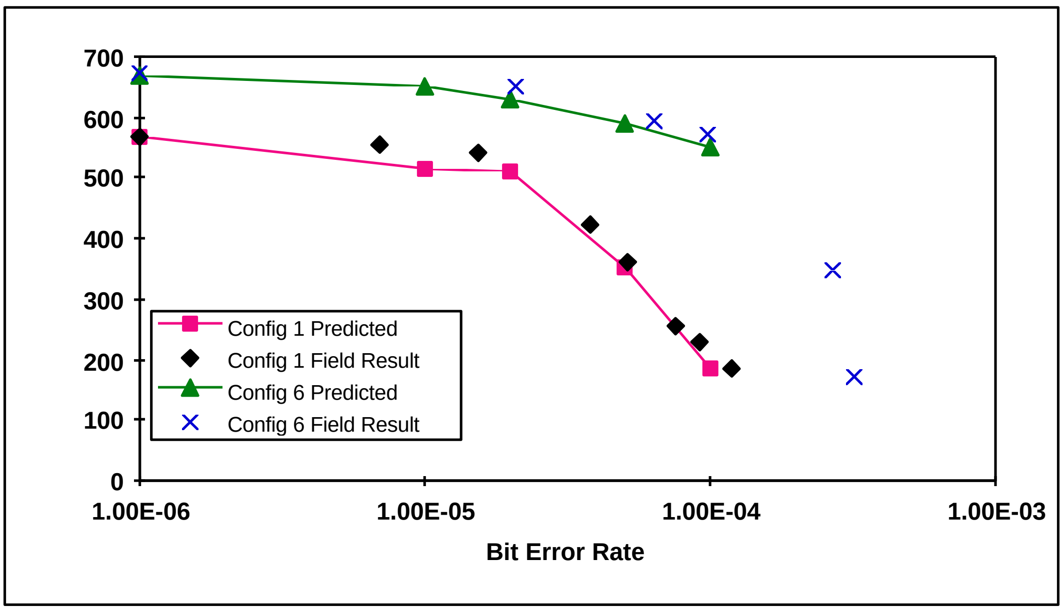
Summary of Results

The experiment was conducted in three phases: we performed initial testing in the field, then we tested the protocols extensively in the laboratory, then we performed final testing in the field. The field results presented in this report reflect the results of the final field testing.

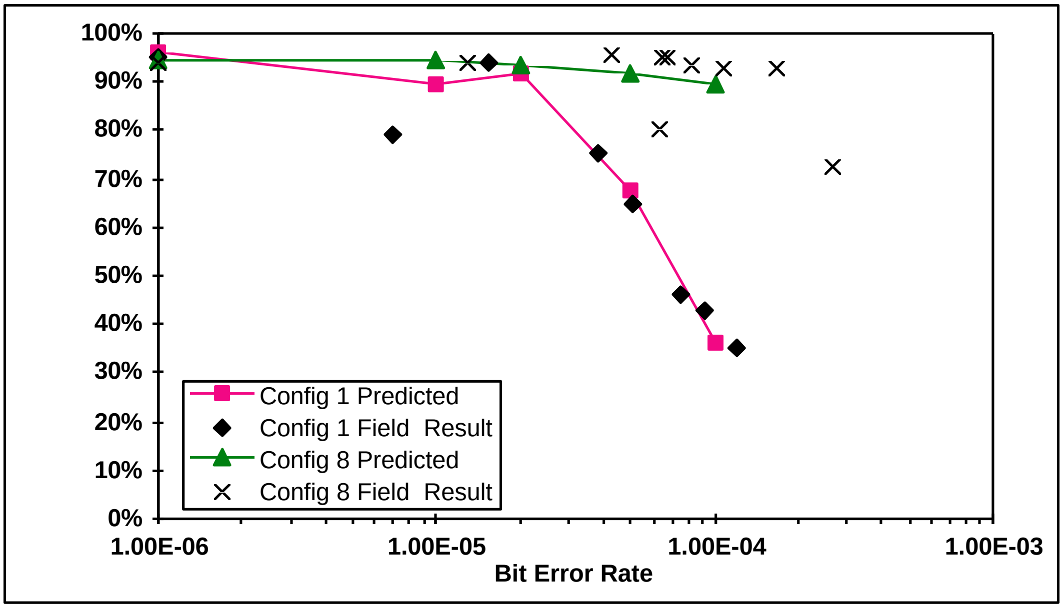
We made the following performance measurements in all tests: throughput, link utilization, and bit-efficiency. Throughput is a measure of the average rate at which the protocol can move user data, and is one of the most commonly used measurements of communication protocol performance. Link utilization is a measure of the ability of the protocol to "keep the pipe full." This ability is important in space communication, in which contact times may be limited. The protocol should not allow the link to be idle when data is waiting to be transmitted. Finally, bit-efficiency is a measure of the amount of protocol overhead required to transfer a user's data. The overhead includes protocol headers, acknowledgment traffic, and any retransmissions required to get the user data to its destination. Bit efficiency is important in spacecraft communications, because link capacity is generally a scarce resource.

Eight protocol configurations were tested in the laboratory, and of those eight, five were tested in the field. The following graphs briefly summarize the field test results, with predictions based on the laboratory results shown for reference. The three graphs correspond to the three types of performance measures made: throughput, link utilization, and bit-efficiency. Each graph presents the results of the protocol configurations that performed the best and the worst in the field. The laboratory predictions are based on the mean response resulting from 10 tests at each of the following bit-error rates: 10^{-6}, 10^{-5}, $2x10^{-5}$, $5x10^{-5}$, 10^{-4}.

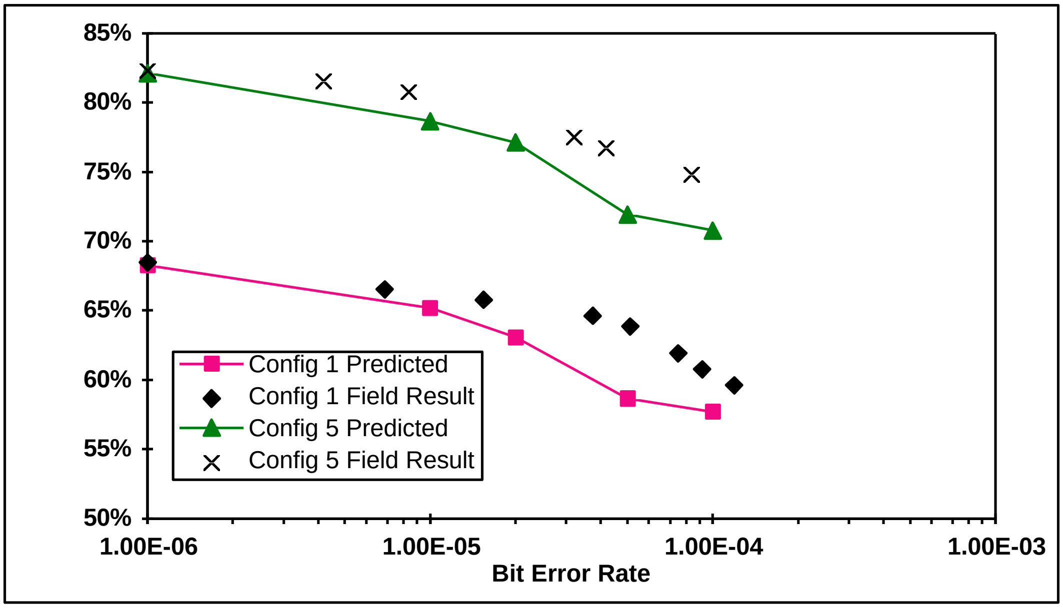
The first graph presents the throughput results. Readers should bear in mind that the maximum possible throughput of a SCPS-TP connection is 768 bps, not including SCPS-TP protocol overhead. The graph shows that the best throughput was obtained by the configuration (Configuration 6) that enabled the Selective Negative Acknowledgment and SCPS-TP Header Compression capabilities, described above. The poorest throughput in the field resulted from the configuration (Configuration 1) that had none of the SCPS-TP capabilities enabled. (Note that the laboratory results indicate that the configuration that enabled TCP Timestamps and none of the other capabilities would have shown lower throughput than Configuration 1, but this configuration was not tested in the field.)



The next graph presents link utilization results. The graph shows that the best link utilization was obtained by the configuration (Configuration 8) that enabled all of the capabilities under test. The poorest link utilization in the field resulted from the configuration (Configuration 1) that had none of the SCPS-TP capabilities enabled.



The final graph presents bit-efficiency results. The graph shows that the best bit-efficiency was obtained by the configuration (Configuration 5) that enabled SCPS-TP Header Compression. The poorest bit-efficiency in the field resulted from the configuration (Configuration 1) that had none of the SCPS-TP capabilities enabled. However, as with the throughput tests, the configuration with TCP Timestamps (only) enabled had worse bit-efficiency than Configuration 1 in the laboratory tests, but was not tested in the field. Note that the bit-efficiency results from the field tests tend to be higher than the results from the laboratory tests of corresponding configurations. This is due to an inherent difference between the field test environment and the laboratory test environment.



Conclusions

The SCPS-TP protocol appears to be well-suited to the long-delay, potentially high bit-error rate environment of the STRV. All configurations were able to sustain connections at bit-error rates of 10^{-4} and yield throughput in excess of 130 bps (17% of maximum possible). The Selective Negative Acknowledgment (SNACK) capability was principally responsible for the ability of the SCPS-TP to operate well in high bit-error rate environments (the configuration with the SNACK capability enabled showed only a 15% drop from maximum throughput at a bit-error rate of approximately 10^{-4}). The SCPS-TP Header Compression accounted for an 18% increase in throughput over the configurations that did not use Header Compression at zero bit error rate.

The following conclusions derive from the laboratory testing and are confirmed by the flight test results:

1. The SNACK capability significantly improves throughput at high bit-error rates, and has no negative effects on throughput at low bit-error rates.
2. The TCP Timestamps capability has a negative effect on throughput at low bit-error rates. It has a strongly negative effect on bit-efficiency. When used in combination with SNACK, throughput is lower than when using SNACK alone. (The magnitude of the negative effect of TCP Timestamps on throughput is exaggerated by the small packet size imposed by the STRV. With larger packet sizes, this effect is mitigated.)
3. The SCPS-TP Header Compression capability has a significant, positive effect on throughput at bit-error rates of 5×10^{-5} and below. Header Compression improves bit-efficiency at all bit-error rates. (The positive effect of Header Compression on

throughput is exaggerated by the small packet size imposed by the STRV in the same manner that the negative effect of the TCP Timestamps is, above. As with TCP Timestamps, the effect of Header Compression on throughput will diminish as the packet size increases.)

The following conclusions are supported by the laboratory testing, but were neither confirmed nor refuted by the flight test results:

1. The SNACK capability significantly improves link utilization at high bit-error rates, has no negative effects on link utilization at low bit-error rates, and has no impact on bit-efficiency.
2. The TCP Timestamps capability has a moderately positive effect on link utilization. When used in combination with SNACK, link utilization is improved slightly.
3. The SCPS-TP Header Compression capability has no effect on link utilization.

Recommendations

We document recommendations primarily directed at ourselves in Appendix C, Lessons Learned. The following recommendations are directed toward potential users of SCPS-TP and toward the sponsors of this effort.

1. Push ahead in the effort to standardize SCPS-TP and deploy it in environments that have similar delay and error characteristics to the STRV environment.
2. When using SCPS-TP in STRV-like environments, enable SNACK.

   SNACK has no negative effects when errors are not present, and is primarily responsible for the protocol's ability to sustain relatively high throughputs at high bit-error rates.
3. When using SCPS-TP in STRV-like environments, enable Header Compression.

   The Header Compression capability reduced the size of SCPS-TP headers, improving throughput and bit-efficiency. These effects were particularly dramatic because the maximum packet size of the STRV was small. As the packet size increases, the positive effect of Header Compression will diminish.
4. When using SCPS-TP in STRV-like environments, disable TCP Timestamps.

   The TCP Timestamps capability reduced throughput at low bit-error rates, and provided no significant improvement in throughput at high bit-error rates when SNACK was in use. As with Header Compression, the negative effects of TCP Timestamps are exaggerated by the small packet sizes on STRV.
5. Evolve the program of testing toward integrated tests.

Although there are still specific SCPS-TP capabilities to be tested, the focus of future tests should be integrated-stack testing. Tests of individual protocol capabilities can be conducted either as part of integrated-stack testing or as a small, focused portion of a larger test. The SCPS-NP, which has not as yet undergone flight testing, will probably benefit from more substantial, focused testing. However, this can still be conducted in the context of an overall test.

Foreword

This is Volume 2 of the Final Report on the SCPS-TP Testing on the UK DRA STRV. This volume contains the Appendixes to the report, including the equations used in calculating derived results, the experiment data, our lessons learned, and four appendixes of explanatory material. The first of these explanatory appendixes provides an overview of the SCPS-TP protocol. The second briefly describes the SCPS-TP implementation that was ported to the STRV satellite. The third describes the DRA Flight and Ground Segments. The final appendix describes the SSFE communication environment.

Table of Contents

Section		Page
List of References		**1**
Appendix A	**Equations**	**5**
Appendix B	**Experiment Data**	**13**
Appendix C	**Lessons Learned**	**53**
Appendix D	**SCPS Transport Protocol Overview**	**57**
Appendix E	**Implementation Background**	**67**
Appendix F	**STRV Spacecraft and Control Center Descriptions**	**71**
Appendix G	**The SSFE Communications Environment**	**81**
Glossary		**95**
Distribution List		**99**

List of Figures

Figure		Page
A-1	Predicted Throughput for Configuration 8	10
F-1	The STRV-1 Orbit	72
F-2	The STRV Spacecraft	73
F-3	STRV Spacecraft Architecture	75
F-4	STRV Control Center Architecture for SSFE	77
G-1	STRV Communications Model via Lasham Antenna	81
G-2	STRV Communications via DSN	82
G-3	STRV Signal via Lasham Antenna	85
G-4	Fast RSL Sample With Frame Errors Superimposed	85
G-5	Typical STRV Signal at a DSN Antenna	86
G-6	STRV Range and Propagation Delay For Lasham	87
G-7	Telemetry Log File Structure	88
G-8	Errored Telemetry Frame	93

List of Tables

Table		Page
A-1	Parameters for Estimating Throughput Response	9
A-2	Predicted Throughput Response for Configuration 8	9
B-1	Laboratory Data - Configuration 1 (1 of 4)	14
B-2	Laboratory Data - Configuration 1 (2 of 4)	15
B-3	Laboratory Data - Configuration 1 (3 of 4)	16
B-4	Laboratory Data - Configuration 1 (4 of 4)	17
B-5	Laboratory Data - Configuration 2 (1 of 4)	18
B-6	Laboratory Data - Configuration 2 (2 of 4)	19
B-7	Laboratory Data - Configuration 2 (3 of 4)	20
B-8	Laboratory Data - Configuration 2 (4 of 4)	21
B-9	Laboratory Data - Configuration 3 (1 of 4)	22
B-10	Laboratory Data - Configuration 3 (2 of 4)	23
B-11	Laboratory Data - Configuration 3 (3 of 4)	24
B-12	Laboratory Data - Configuration 3 (4 of 4)	25
B-13	Laboratory Data - Configuration 4 (1 of 4)	26
B-14	Laboratory Data - Configuration 4 (2 of 4)	27
B-15	Laboratory Data - Configuration 4 (3 of 4)	28
B-16	Laboratory Data - Configuration 4 (4 of 4)	29
B-17	Laboratory Data - Configuration 5 (1 of 4)	30
B-18	Laboratory Data - Configuration 5 (2 of 4)	31

Table		Page

B-19	Laboratory Data - Configuration 5 (3 of 4)	32
B-20	Laboratory Data - Configuration 5 (4 of 4)	33
B-21	Laboratory Data - Configuration 6 (1 of 4)	34
B-22	Laboratory Data - Configuration 6 (2 of 4)	35
B-23	Laboratory Data - Configuration 6 (3 of 4)	36
B-24	Laboratory Data - Configuration 6 (4 of 4)	37
B-25	Laboratory Data - Configuration 7 (1 of 4)	38
B-26	Laboratory Data - Configuration 7 (2 of 4)	39
B-27	Laboratory Data - Configuration 7 (3 of 4)	40
B-28	Laboratory Data - Configuration 7 (4 of 4)	41
B-29	Laboratory Data - Configuration 8 (1 of 4)	42
B-30	Laboratory Data - Configuration 8 (2 of 4)	43
B-31	Laboratory Data - Configuration 8 (3 of 4)	44
B-32	Laboratory Data - Configuration 8 (4 of 4)	45
B-33	Field Test Data (1 of 4)	48
B-34	Field Test Data (2 of 4)	49
B-35	Field Test Data (3 of 4)	50
B-36	Field Test Data (4 of 4)	51
F-1	STRV Spacecraft Design Summary	74
G-1	STRV Link Specifications	83
G-2	Telemetry Frame Structure	89

Table		Page
G-3	Telemetry Frame Trailer Added by Frame Synchronizer	89
G-4	Typical Telemetry Frame	91

List of References

1. Aho, Alfred V., Brian W. Kernighan, and Peter J. Weinberger, 1988, *The AWK Programming Language*, Addison-Wesley Publishing Company, Reading, Massachusetts.
2. L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New Techniques for Congestion Avoidance," *Proceedings of SIGCOMM 1994,* pp. 24-35, London, U. K., October 1994.
3. Çinlar, Erhan, 1975, *Introduction to Stochastic Processes*, Prentice-Hall, Inc., Englewood Cliffs, NJ.
4. Connolly, T., P. Amer, P. Conrad, November 1994, "An Extension to TCP: Partial Order Service," RFC 1693.
5. Cooper, G., March 1986, TCP implementation, IMAGEN Corporation.
6. Fortune, D., January 1996, "SSFE Ground Segment Interface Control Document," ESYS-95107-RPT-01, European Systems Limited, Guildford, Surrey, UK.
7. Fox, R., June 1989, "TCP Big Window and Nak Options," Internet Engineering Task Force document RFC 1106.
8. IETF, 1989, "Requirements for Internet Hosts - Communications Layers," Internet Engineering Task Force document RFC 1122.
9. Jacobson, V. and R. Braden, October 1988, "TCP Extensions for Long-Delay Paths," Internet Engineering Task Force document RFC 1072.
10. V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for Long-Delay Paths," Request for Comments 1323, IETF, May 1992.
11. Jain, Raj, 1991, *The Art of Computer Systems Performance Analysis-Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, New York, NY
12. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), January 1996, Report Concerning Space Communications Protocol Standards, Advanced Orbiting Systems Upper Layer Protocols: Rationale, Requirements and Application Notes , SCPS 710.0-G-0 (Draft), DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA
13. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), April 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS Network Protocol (SCPS-NP)

Specification, SCPS 713.0, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

14. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), April 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS Security Protocol (SCPS-SP) Specification, SCPS 7135.0-C-1, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

15. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), March 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS Transport Protocol (SCPS-TP) Specification, SCPS 714.0, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

16. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), June 1995, Concept Paper: ADVANCED ORBITING SYSTEMS, UPPER LAYER PROTOCOLS, SCPS File Transfer Protocol (SCPS-FP) Specification, SCPS 717.0, DRAFT, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

17. The Joint NASA/DOD Space Communications Protocol Standards Technical Working Group (SCPS-TWG), September 1993, Recommended Development of Interoperable Data Communications Standards for Dual-Use by US Civil and Military Space Projects, DOD and NASA, Jet Propulsion Laboratory, Pasadena, CA, USA

18. Libes, Don, 1995, *Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs*, O'Reilly & Associates, Inc., Sebastopol, CA

19. Packet Telemetry, Blue Book, CCSDS 102.0-B-3, Issue 3 November 1992, or later issue.

20. Stevens, W. R., 1994, *TCP/IP Illustrated, Volume 1: The Protocols*, Reading, Massachusetts: Addison-Wesley

21. Telecommand—Part 1 Channel Service, Blue Book, CCSDS 201.0-B-1, January 1987, or later issue.

22. Telecommand—Part 2 Data Routing Service, Blue Book, CCSDS 202.0-B-2, January 1987, or later issue.

23. Telecommand—Part 2.1 Command Operation Procedures, Blue Book, CCSDS 202.1-B-1, October 1991, or later issue.

24. Telecommand—Part 3 Data Management Service, Blue Book, CCSDS 203.0-B-1, January 1987, or later issue.

25. Turner, J., October 1986, "New Directions in Communications (Or Which Way to the

Information Age?)," *IEEE Communications* , Vol. 24, No. 10, 8-15.

26. Welch, Brent B., 1995, *Practical Programming in Tcl and Tk*, Prentice Hall PTR, Upper Saddle River, NJ.

27. Wells, N., August 1994, “The Space Technology Research Vehicles STRV-1a and -1b: First In-Orbit Results,” Paper presented at the 8th Annual AIAA/Utah State University Conference on Small Satellites, Utah State University, Logan, UT.

Appendix A

Equations

The laboratory results are calculated from the post-processed data using Equation 1 through Equation 11. Subsequent equations support predictions of performance (Equation 12 through Equation 14), and theoretical bit-error rate limits of a retransmission protocol (Equation 15 through Equation 17).

A.1 Equations To Operate on Post-Processed Data

Equation 1: Throughput Calculation

$$throughput = \frac{total\_user\_data \text{ (in bytes)} \, x \, 8 \text{ (bits / byte)}}{time\_in\_data\_transfer\_phase \text{ (in seconds)}}$$

Equation 2: Total User Data

The amount of user data per test was set to yield an integral number of fully-filled packets. As a result, this amount is a variable in the above equations. Equation 2 provides the proper values for *total_user_data*.

$$total\_user\_data \text{ (in bytes)} = \begin{cases} Configuration\_1: 50048 \\ Configuration\_2: 50048 \\ Configuration\_3: 50024 \\ Configuration\_4: 50024 \\ Configuration\_5: 50008 \\ Configuration\_6: 50008 \\ Configuration\_7: 50048 \\ Configuration\_8: 50048 \end{cases}$$

Equation 3: Link Utilization

$$link\_utilization = \frac{total\_downlink\_data}{elapsed\_time\_of\_test} \, x \, \frac{1}{downlink\_capacity}$$

Equation 4: Total Downlink Data

where $$total\_downlink\_data = SCPS\_TP\_bytes\_down + (CCSDS\_Header\_Size x total\_downlink\_packets)$$

Note that *total_downlink_data* does not include any data for packets that were missing or in error.

Equation 5: CCSDS Header Size

and $CCSDS\_Header\_Size = 6 \text{ (bytes)}$

Equation 6: Downlink Capacity

and $$downlink\_capacity = \frac{SCPS\_Packet\_Capacity\_per\_Frame}{Frame\_Size} = \frac{3x90}{328}.$$

Equation 7: Bit Efficiency

$$bit\_efficiency = \frac{total\_user\_data}{total\_downlink\_data + total\_uplink\_data}$$

Equation 8: Total Uplink Data

where $$total\_uplink\_data = SCPS\_TP\_bytes\_up + (CCSDS\_Header\_Size x total\_uplink\_packets)$$

Equation 9: Bit-Error Rate (Based on Packet Success Rate)

Note: this equation assumes that there is a single bit-error per lost or corrupted packet, and that errors follow a Bernoulli process [3].

$$BER = 1 - q^{\frac{1}{N}}$$

where

Equation 10: Packet Success Rate

$$q = \frac{Total\_downlink\_packets \text{ (received)}}{Maximum\_CCSDS\_Sequence\_Number + 1}$$

(Note that *Total_downlink_packets* is a count of the packets actually *received* by the SCPS-TP process, and does not include any that are in error or missing. Also note that the *Maximum_CCSDS_Sequence_Number* is zero-based, hence its value is incremented by one to get the actual number of packets sent by the onboard SCPS-TP entity.)

and

Equation 11: Average Downlink Packet Size

$$N = \frac{Total\_downlink\_data}{Total\_downlink\_packets}$$

A.2 Predicted Results Based on Laboratory Tests

We can predict the mean response and confidence intervals for any combination of the three capabilities tested in the laboratory, using the laboratory results as a basis. As a part of calculating the allocation of variation in Section 6, Table 6, Table 7, and Table 8, we built a regression model of the response. The general equation for the mean response, $\hat{y}$, of a 2^3 experiment design is as follows:

Equation 12: Mean Response of a 2^3 Experiment

$$\hat{y} = q_0 + q_A x_A + q_B x_B + q_C x_C + q_{AB} x_A x_B + q_{BC} x_B x_C + q_{AC} x_A x_C + q_{ABC} x_A x_B x_C$$

For the throughput response, the model is parameterized based on the following x values and the q values in :

$$x_A = \begin{cases} \text{SNACK disabled}: -1 \\ \text{SNACK enabled}: +1 \end{cases}$$

$$x_B = \begin{cases} \text{Timestamps disabled}: -1 \\ \text{Timestamps enabled}: +1 \end{cases}$$

$$x_C = \begin{cases} \text{Header Compression disabled}: -1 \\ \text{Header Compression enabled}: +1 \end{cases}$$

To calculate the 90% confidence intervals for one run of a confirmation experiment in the future, we use the following equations, and use the *q* values and the standard deviation of the error term (s_e) from Table A-1. The parameters in Table A-1 are calculated as part of the allocation of variation, described earlier. The data supporting the allocation of variation appears in Appendix B. Refer to [11], pages 295-301 for an in-depth discussion of developing confidence intervals for predicted responses.

Equation 13: Effective Degrees of Freedom for a $2^3$10 Experiment

$$n_{eff} = \frac{2^3 x 10}{9} = 8.89 \text{ for a } 2^3 10 \text{ experiment}$$

Equation 14: Standard Deviation of the Mean Response for One Test in the Future

$$s_{\hat{y}_1} = s_e \left(\frac{1}{n_{eff}} + 1 \right)^{1/2}$$

Therefore, the 90% confidence interval for the predicted mean response is given by:

$$\hat{y} \mp t_{[0.95;2^3(r-1)]} s_{\hat{y}_1} \quad \text{where r} = 10 \text{ and therefore } t_{[0.95,72]} \cong 1.67$$

To build the model for a specific configuration, we select the capabilities that are enabled for that configuration, and pick the appropriate *x* values accordingly. For example, consider Configuration 8. In this configuration, all capabilities (SNACK, Timestamps, and Header Compression) are enabled. Therefore, $x_A = x_B = x_C = 1$. We then solve the mean response equation to produce $\hat{y}$ for each bit-error rate. To produce the 90% confidence interval for one future test, we solve for $s_{\hat{y}_1}$ to produce the standard deviation of the estimated mean response. We multiply this value by the value looked up in the table of the *t*-distribution and subtract the result from $\hat{y}$ to produce the lower bound of the 90% confidence interval, and add it to $\hat{y}$ to produce the upper bound. The result of these calculations for the throughput response of Configuration 8 appear in Table A-2.

Table A-1. Parameters for Estimating Throughput Response

BER:	10^{-6}	10^{-5}	$2x10^{-5}$	$5x10^{-5}$	10^{-4}
q_0	573.97	548.50	523.77	455.29	370.47
q_A	-1.06	9.24	14.71	50.34	106.38
q_B	-44.33	-38.72	-42.20	-17.34	16.76
q_C	61.18	61.50	55.96	61.55	47.03
q_{AB}	-0.48	-6.15	0.32	-16.61	-47.62
q_{AC}	0.64	-2.77	0.59	-3.65	4.87
q_{BC}	8.94	8.12	11.33	10.28	19.94
q_{ABC}	-0.16	2.12	-4.52	-2.07	-9.26
s_e	6.28	24.74	34.87	46.01	33.21

Table A-2. Predicted Throughput Response for Configuration 8

BER:	$\hat{y}$	90% Confidence Low	90% Confidence High
10^{-6}	599	588	610
10^{-5}	582	539	625
$2x10^{-5}$	560	499	620
$5x10^{-5}$	538	458	618
10^{-4}	509	451	566

The graph of the predicted response is shown in Figure A-1. Predicted Throughput for Configuration 8, below. These predictions will serve as the point of reference for plotting the Field Test results. Using these predictions as a guide, we can easily see whether the test results deviate from our expectations. We also present the Field Test results for link utilization and bit-efficiency in the context of the predicted responses for each.

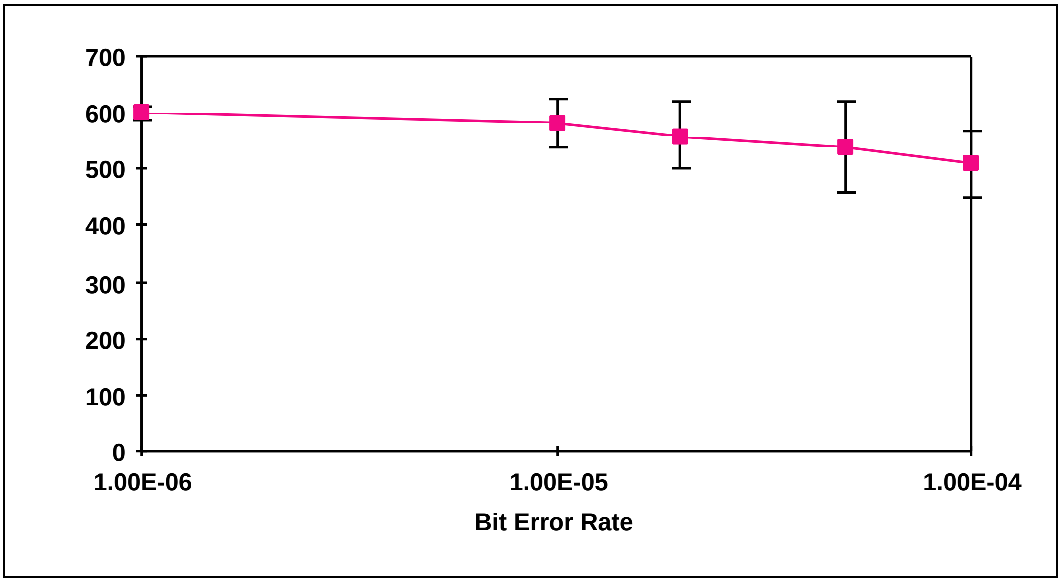


Figure A-1. Predicted Throughput for Configuration 8

A.3 Theoretical Bit-Error Rate Limit of a Retransmission Protocol

To calculate the BER at which the protocol fails, we can construct a simple analytical model. The model depends on several things: the maximum retransmissions setting, the error distribution on the link, the packet size, the length of the run, and the level of assurance we desire that the run will succeed. We model errors as random occurrences, so the answer is given as a probability. Let us assume that the error distribution follows a Bernoulli process. Assume that we wish to be 99.9% sure that the run will finish, and that the user data fits into 622 unique packets (which happens to be the value for Configuration 6). Therefore, we wish to be 99.9% sure that 622 unique packets arrive at the destination. We can calculate the probability of arrival for each of those packets in order to have the probability that they *all* arrive equal to 99.9%:

Equation 15: Probability of Success of the Entire Connection

$$P(success\_of\_run) = P(success\_per\_unique\_packet)^{number\_of\_unique\_packets}, \text{ or,}$$

$$0.999 = P(success\_per\_unique\_packet)^{622}$$

$$P(success\_per\_unique\_packet) = 0.99999839$$

Given the required probability of success for each unique packet (which may be retransmitted some number of times), we can calculate the necessary probability of success for each successive transmission of that packet, based on the maximum number of retransmissions. A reasonably straightforward way to calculate this is to calculate the probability that *all* transmissions will fail, and then subtract that value from one to get the probability that at least *one* will succeed. Let us assume that *max_rexmits* is set to 9, so we have a total of 10 transmissions (the initial one plus 9 retransmissions) before we reach failure.

Equation 16: Probability of Failure for All Retransmissions of a Packet

$$P(all\_transmissions\_fail) = (1 - P(success\_per\_unique\_packet))^{1/\max\_rexmits}$$
$$P(all\_transmissions\_fail) = (1 - 0.99999839)^{1/10} = 0.2634163$$

Translating that value to a probability of success, we have:

Equation 17: Probability of Success for At Least One Transmission of a Packet

$$P(all\_transmissions\_succeed) = 1 - P(all\_transmissions\_fail) = 1 - 0.2634163 = 0.7365837$$

If we assume that errors follow a Bernoulli process, we can calculate the maximum bit-error rate for a given packet size, based on Equation 9. Let us assume that the packets are 90 bytes long:

$$BER = 1 - P(any\_transmissions\_succeed)^{1/bits\_per\_packet} = 1 - 0.7365837^{1/90x8} = 4.25E\text{-}04$$

Note that if the run is longer (more unique packets), the maximum bit-error rate is lower. For example, if we increase the length of the run to 1000 packets, the maximum BER is $4.02x10^{-4}$. If we revert to the original 622-packet run, but increase the maximum retransmissions to 50, the maximum sustainable bit-error rate increases to $2.01x10^{-3}$.

Appendix B

Experiment Data

The column headings for the laboratory data tables are as follows:

Result	An indication of whether the test passed or failed. Failed tests have been sorted out of the data set, and do not appear in these tables
Config	The protocol configuration (1 through 8) under test
BER	The bit-error rate requested of Spanner
Rep	The repetition count of the test
Elapsed Time (SCPS-TP)	The amount of time reported by the SCPS-TP responder between the beginning and end of the connection
Bytes Up	The number of bytes of data in SCPS-TP packets (including SCPS-TP overhead, but not CCSDS path packet overhead) transmitted by the Ground System to the OBC
Bytes Down	The number of bytes of data in SCPS-TP packets transmitted by the OBC to the Ground System
Pkts Up	The number of packets transmitted by the Ground System. The "U" designation means uncompressed, "C" means compressed.
Pkts Down	The number of packets transmitted by the OBC.
Data Xfer Time	The time from when the first data packet was sent until the last data packet was acknowledged.
Elapsed Time (tcpdump)	The amount of time between the beginning and end of the connection, determined by post processing tcpdump output.
Pkt Drops	The number of packets dropped by the Spanner program to simulate bit-errors. "Up" means that a packet destined for the OBC was dropped, "Down" means that a packet destined for the Ground System was dropped.
Throughput	The throughput of the connection per Equation 1, in bits per second.
Link Utilization	The down link utilization per Equation 3, expressed as a percentage.
Bit Efficiency	The bit-efficiency of the connection per Equation 7, expressed as a percentage.

Table B-1. Laboratory Data - Configuration 1 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	1	1E-6	5	710.4	1988	65796	90	0	786	0	703.4	711.3
Pass	1	1E-6	10	710.6	1988	65796	90	0	786	0	703.6	711.5
Pass	1	1E-6	2	710.7	1988	65796	90	0	786	0	703.7	711.6
Pass	1	1E-6	8	711.9	1988	65796	90	0	786	0	701.9	712.9
Pass	1	1E-6	9	712.1	2032	65964	92	0	788	0	705.2	713.1
Pass	1	1E-6	6	712.8	1988	65796	90	0	786	0	702.7	713.8
Pass	1	1E-6	4	713.0	2054	66048	93	0	789	0	706.0	713.9
Pass	1	1E-6	7	714.1	2010	66468	91	0	794	0	704.0	715.0
Pass	1	1E-6	3	719.0	2142	66300	97	0	792	0	708.9	719.9
Pass	1	1E-6	1	723.8	2230	66636	101	0	796	0	716.8	724.7
Pass	1	1E-5	1	719.3	2274	66804	103	0	798	0	712.3	720.2
Pass	1	1E-5	8	747.3	2560	67664	116	0	809	0	740.3	748.2
Pass	1	1E-5	6	757.6	2604	67936	118	0	813	0	744.3	758.5
Pass	1	1E-5	5	757.8	2582	67832	117	0	811	0	735.1	765.3
Pass	1	1E-5	10	761.7	2802	68460	127	0	820	0	745.3	766.1
Pass	1	1E-5	3	761.7	2736	68420	124	0	818	0	751.6	762.6
Pass	1	1E-5	9	772.6	3220	69932	146	0	836	0	762.5	773.5
Pass	1	1E-5	4	846.3	2736	68252	124	0	816	0	839.4	847.3
Pass	1	1E-5	2	898.5	2802	68504	127	0	819	0	891.5	899.4
Pass	1	1E-5	7	940.4	3462	70415	157	0	844	0	933.3	941.2
Pass	1	2E-5	2	737.0	2582	67896	117	0	811	0	726.9	737.9
Pass	1	2E-5	8	745.7	2780	68568	126	0	819	0	735.6	746.6
Pass	1	2E-5	5	756.6	2758	68504	125	0	819	0	740.2	760.9
Pass	1	2E-5	1	760.8	2890	69092	131	0	826	0	750.6	761.7
Pass	1	2E-5	3	782.0	3352	70920	152	0	847	0	771.9	782.9
Pass	1	2E-5	9	796.1	3396	70688	154	0	845	0	786.0	797.1
Pass	1	2E-5	4	797.3	3550	71064	161	0	851	0	780.9	801.7
Pass	1	2E-5	7	823.3	3506	70940	159	0	848	0	816.3	824.2
Pass	1	2E-5	10	869.8	4430	74316	201	0	892	0	845.1	878.4
Pass	1	2E-5	6	904.5	3330	70268	151	0	840	0	897.6	905.4
Pass	1	5E-5	10	895.3	4584	74636	208	0	892	0	878.9	899.7
Pass	1	5E-5	3	915.7	4188	73544	190	0	879	0	908.8	916.6

Table B-2. Laboratory Data - Configuration 1 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	569.2	96%	69%
0	0	569.0	96%	69%
0	0	569.0	96%	69%
0	0	570.4	96%	69%
1	0	567.8	96%	68%
0	0	569.8	96%	69%
1	0	567.1	96%	68%
1	0	568.7	97%	68%
1	0	564.8	96%	68%
1	0	558.6	96%	67%
3	0	562.1	97%	67%
5	0	540.9	94%	66%
6	0	537.9	93%	66%
4	0	544.6	92%	66%
6	0	537.2	93%	65%
12	0	532.7	93%	65%
12	0	525.1	94%	63%
6	0	477.0	84%	65%
8	0	449.1	79%	65%
15	0	429.0	78%	63%
6	0	550.8	96%	66%
8	0	544.3	96%	65%
9	0	540.9	94%	65%
12	0	533.4	94%	64%
17	0	518.7	94%	62%
17	0	509.4	92%	62%
16	0	512.7	92%	62%
15	0	490.5	90%	62%
25	0	473.8	88%	59%
15	0	446.1	81%	63%
38	0	455.5	86%	58%
34	0	440.6	84%	59%

Table B-3. Laboratory Data - Configuration 1 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	1	5E-5	7	999.2	4430	74132	201	0	886	0	992.2	1000.1
Pass	1	5E-5	1	1074.1	4276	73880	194	0	883	0	1067.2	1075.0
Pass	1	5E-5	8	1083.2	4430	74048	201	0	885	0	1076.3	1084.1
Pass	1	5E-5	2	1088.3	4232	73460	192	0	878	0	1081.3	1089.2
Pass	1	5E-5	4	1308.6	4650	74972	211	0	896	0	1301.6	1309.6
Pass	1	5E-5	9	1320.5	4144	73460	188	0	878	0	1313.5	1321.4
Pass	1	5E-5	6	1417.1	5442	77996	247	0	932	0	1410.1	1418.0
Pass	1	5E-5	5	1872.0	4298	73760	195	0	882	0	1865.1	1872.9
Pass	1	1E-4	7	1605.9	4782	75056	217	0	897	0	1598.9	1606.8
Pass	1	1E-4	4	1853.3	4914	75560	223	0	903	0	1848.5	1856.4
Pass	1	1E-4	10	1898.1	4804	74972	218	0	896	0	1892.2	1900.1
Pass	1	1E-4	8	1932.0	4694	74992	213	0	897	0	1925.0	1932.9
Pass	1	1E-4	3	2117.2	4672	74264	212	0	888	0	2110.2	2118.1
Pass	1	1E-4	6	2120.6	4430	74048	201	0	885	0	2113.6	2121.5
Pass	1	1E-4	2	2482.7	4496	74888	204	0	895	0	2475.7	2483.6
Pass	1	1E-4	5	2513.5	4914	75728	223	0	905	0	2506.5	2514.4
Pass	1	1E-4	1	2798.8	5112	76448	232	0	914	0	2791.8	2799.7
Pass	1	1E-4	9	2940.1	5134	76000	233	0	909	0	2920.3	2941.1

Table B-4. Laboratory Data - Configuration 1 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughpu t (bps)	Link Util	Bit Eff
38	0	403.5	77%	59%
42	0	375.2	72%	59%
34	0	372.0	71%	59%
36	0	370.3	70%	60%
49	0	307.6	60%	58%
42	0	304.8	58%	60%
48	0	283.9	57%	55%
55	0	214.7	41%	59%
69	0	250.4	49%	58%
69	0	216.6	42%	57%
74	0	211.6	41%	58%
78	0	208.0	40%	58%
80	0	189.7	37%	59%
80	0	189.4	36%	59%
75	0	161.7	31%	58%
83	0	159.7	31%	57%
89	0	143.4	28%	57%
88	0	137.1	27%	57%

Table B-5. Laboratory Data - Configuration 2 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	2	1E-6	9	710.0	1992	65796	90	0	786	0	703.1	711.0
Pass	2	1E-6	5	710.5	1992	65796	90	0	786	0	703.5	711.4
Pass	2	1E-6	6	710.5	1992	65796	90	0	786	0	703.5	711.4
Pass	2	1E-6	3	711.6	2022	65880	91	0	787	0	704.7	712.6
Pass	2	1E-6	4	712.4	1992	65796	90	0	786	0	702.3	713.3
Pass	2	1E-6	2	713.8	2088	66132	94	0	790	0	706.9	714.7
Pass	2	1E-6	1	718.1	2206	66468	99	0	794	0	711.2	719.1
Pass	2	1E-6	8	721.9	2324	66888	104	0	799	0	715.0	722.8
Pass	2	1E-6	10	724.5	2324	66888	104	0	799	0	717.6	725.4
Pass	2	1E-6	7	730.8	2126	66152	95	0	791	0	723.4	731.8
Pass	2	1E-5	6	718.4	2110	66132	95	0	790	0	708.3	719.4
Pass	2	1E-5	7	720.3	2310	66804	103	0	798	0	713.4	721.3
Pass	2	1E-5	8	720.5	2304	66804	102	0	798	0	713.6	721.5
Pass	2	1E-5	1	720.7	2318	66804	103	0	798	0	713.7	721.6
Pass	2	1E-5	10	734.5	2508	67476	112	0	806	0	721.2	735.4
Pass	2	1E-5	4	735.4	2538	67560	113	0	807	0	722.1	736.3
Pass	2	1E-5	3	740.9	2834	68484	125	0	818	0	733.9	741.8
Pass	2	1E-5	5	741.2	2864	68568	126	0	819	0	734.3	742.2
Pass	2	1E-5	2	744.9	2812	68400	124	0	817	0	731.6	745.8
Pass	2	1E-5	9	744.9	2916	68652	128	0	820	0	734.8	745.8
Pass	2	2E-5	8	729.3	2592	67812	114	0	810	0	722.3	730.2
Pass	2	2E-5	7	743.8	2960	68820	130	0	822	0	736.9	744.7
Pass	2	2E-5	1	748.1	3106	69408	137	0	829	0	741.2	749.0
Pass	2	2E-5	2	755.9	3078	69912	135	0	835	0	742.6	756.8
Pass	2	2E-5	5	764.5	2866	68500	125	0	822	0	744.4	772.0
Pass	2	2E-5	3	766.5	2954	68944	129	0	825	0	755.9	767.4
Pass	2	2E-5	6	774.3	3064	69280	134	0	829	0	760.6	775.3
Pass	2	2E-5	4	775.3	3490	71612	153	0	856	0	755.8	779.7
Pass	2	2E-5	9	788.7	3466	70348	149	0	844	0	765.5	793.1
Pass	2	2E-5	10	837.4	2810	68168	121	0	815	0	829.9	838.4
Pass	2	5E-5	1	749.5	2944	69120	126	0	826	0	736.3	750.4
Pass	2	5E-5	10	800.2	3610	70852	153	0	850	0	777.1	804.6

Table B-6. Laboratory Data - Configuration 2 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	569.5	96%	69%
0	0	569.1	96%	69%
0	0	569.1	96%	69%
1	0	568.2	96%	68%
0	0	570.1	96%	69%
1	0	566.4	96%	68%
2	0	563.0	96%	68%
3	0	560.0	96%	67%
3	0	558.0	96%	67%
1	0	553.5	94%	68%
1	0	565.2	96%	68%
5	0	561.3	96%	67%
7	0	561.1	96%	67%
5	0	561.0	96%	67%
4	0	555.2	96%	66%
6	0	554.4	96%	66%
9	0	545.6	96%	65%
10	0	545.3	96%	65%
9	0	547.2	96%	65%
11	0	544.9	96%	65%
9	0	554.3	97%	66%
11	0	543.4	96%	65%
10	0	540.2	97%	64%
14	0	539.1	96%	64%
12	0	537.8	92%	65%
13	0	529.7	94%	64%
11	0	526.4	93%	64%
18	0	529.8	96%	62%
21	0	523.0	92%	63%
16	0	482.4	85%	65%
22	0	543.8	96%	64%
31	0	515.3	92%	62%

Table B-7. Laboratory Data - Configuration 2 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	2	5E-5	3	802.1	3688	71444	158	0	854	0	788.3	803.0
Pass	2	5E-5	5	818.4	3810	71860	165	0	862	0	795.8	822.8
Pass	2	5E-5	8	825.8	4334	73352	183	0	879	0	805.8	833.3
Pass	2	5E-5	9	853.7	3934	72116	167	0	862	0	846.2	854.6
Pass	2	5E-5	2	854.6	4494	74128	191	0	889	0	835.0	859.0
Pass	2	5E-5	6	883.4	4902	75176	211	0	903	0	856.2	893.2
Pass	2	5E-5	7	883.7	3646	71212	155	0	852	0	863.9	884.6
Pass	2	5E-5	4	901.0	4118	72452	175	0	866	0	893.5	901.9
Pass	2	1E-4	10	807.7	4098	72284	165	0	864	0	800.2	808.6
Pass	2	1E-4	4	817.6	4526	74468	191	0	890	0	801.3	822.0
Pass	2	1E-4	2	821.2	3846	71884	159	0	860	0	807.5	822.2
Pass	2	1E-4	7	826.8	4358	73648	179	0	881	0	819.3	827.8
Pass	2	1E-4	9	856.5	4784	75560	198	0	903	0	849.0	857.4
Pass	2	1E-4	6	859.4	3614	71967	143	0	864	0	848.7	860.3
Pass	2	1E-4	1	872.0	4614	74002	187	0	889	0	852.0	879.5
Pass	2	1E-4	8	926.8	3472	71843	138	0	861	0	919.3	927.7
Pass	2	1E-4	3	930.8	4758	75140	195	0	898	0	923.3	931.7
Pass	2	1E-4	5	1013.8	4124	73816	168	0	883	0	1006.3	1014.7

Table B-8. Laboratory Data - Configuration 2 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
30	0	507.9	93%	62%
26	0	503.2	91%	61%
39	0	496.9	92%	60%
34	0	473.1	88%	61%
31	0	479.5	90%	59%
30	0	467.6	88%	58%
35	0	463.5	84%	62%
31	0	448.1	84%	60%
60	0	500.4	93%	61%
47	0	499.7	94%	59%
47	0	495.8	91%	61%
55	0	488.7	93%	59%
66	0	471.6	92%	58%
69	0	471.7	87%	61%
68	0	469.9	88%	59%
67	0	435.5	81%	62%
63	0	433.7	84%	58%
66	0	397.9	76%	59%

Table B-9. Laboratory Data - Configuration 3 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	3	1E-6	8	873.4	3748	80952	110	0	966	0	864.9	874.3
Pass	3	1E-6	4	874.3	3816	81120	112	0	968	0	865.7	875.2
Pass	3	1E-6	3	875.7	3748	80952	110	0	966	0	863.2	876.6
Pass	3	1E-6	10	875.8	3748	80952	110	0	966	0	863.4	876.7
Pass	3	1E-6	6	876.0	3748	80952	110	0	966	0	863.5	876.9
Pass	3	1E-6	1	876.1	3748	80952	110	0	966	0	863.7	877.1
Pass	3	1E-6	7	876.1	3748	80952	110	0	966	0	863.7	877.1
Pass	3	1E-6	2	882.0	3986	81456	117	0	972	0	869.6	882.9
Pass	3	1E-6	9	892.7	4088	81792	120	0	976	0	872.5	893.7
Pass	3	1E-6	5	893.2	4428	82464	130	0	984	0	880.8	894.1
Pass	3	1E-5	7	901.5	4836	83388	142	0	995	0	893.0	902.4
Pass	3	1E-5	1	902.7	4802	83304	141	0	994	0	890.3	903.6
Pass	3	1E-5	4	919.1	5448	84816	160	0	1012	0	910.6	920.0
Pass	3	1E-5	8	920.6	4870	83420	143	0	996	0	912.1	921.5
Pass	3	1E-5	2	921.9	5312	84648	156	0	1010	0	905.6	922.8
Pass	3	1E-5	5	927.0	5312	84564	156	0	1009	0	902.9	927.9
Pass	3	1E-5	10	927.7	5176	84144	152	0	1004	0	915.3	928.7
Pass	3	1E-5	6	940.5	5142	84272	151	0	1008	0	916.4	941.4
Pass	3	1E-5	3	966.5	5074	83687	149	0	1001	0	958.0	967.4
Pass	3	1E-5	9	985.1	5958	85888	175	0	1026	0	972.7	986.0
Pass	3	2E-5	6	970.4	6094	86612	179	0	1034	0	938.4	979.0
Pass	3	2E-5	10	992.3	6842	87756	201	0	1047	0	976.0	993.3
Pass	3	2E-5	5	1000.7	6264	86656	184	0	1037	0	972.7	1005.8
Pass	3	2E-5	3	1001.9	5958	85856	175	0	1025	0	993.3	1002.8
Pass	3	2E-5	8	1012.5	6876	87872	202	0	1049	0	1004.0	1013.4
Pass	3	2E-5	1	1016.0	6298	86444	185	0	1032	0	1007.5	1016.9
Pass	3	2E-5	7	1030.4	6264	86608	184	0	1035	0	1014.1	1031.3
Pass	3	2E-5	4	1089.4	7556	89380	222	0	1070	0	1073.0	1090.3
Pass	3	2E-5	9	1091.2	7862	90276	231	0	1077	0	1047.5	1092.1
Pass	3	2E-5	2	1133.3	5754	84648	169	0	1010	0	1120.9	1134.2
Pass	3	5E-5	3	1084.4	7420	89552	218	0	1069	0	1075.9	1085.3
Pass	3	5E-5	7	1177.8	8236	91180	242	0	1089	0	1165.3	1178.7

Table B-10. Laboratory Data - Configuration 3 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	462.7	96%	55%
1	0	462.3	97%	55%
0	0	463.6	96%	55%
0	0	463.5	96%	55%
0	0	463.4	96%	55%
0	0	463.3	96%	55%
0	0	463.3	96%	55%
1	0	460.2	96%	54%
1	0	458.7	95%	54%
2	0	454.4	96%	53%
6	0	448.2	96%	53%
5	0	449.5	96%	53%
7	0	439.5	96%	51%
7	0	438.8	94%	53%
8	0	441.9	96%	52%
10	0	443.2	95%	52%
9	0	437.2	94%	52%
9	0	436.7	93%	52%
9	0	417.7	90%	52%
12	0	411.4	91%	51%
13	0	426.4	92%	50%
16	0	410.0	92%	49%
16	0	411.4	90%	50%
15	0	402.9	89%	51%
16	0	398.6	90%	49%
17	0	397.2	89%	50%
14	0	394.6	87%	50%
16	0	373.0	85%	48%
22	0	382.1	86%	47%
8	0	357.0	78%	51%
33	0	372.0	86%	48%
37	0	343.4	81%	47%

Table B-11. Laboratory Data - Configuration 3 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	3	5E-5	8	1191.6	7692	89636	226	0	1070	0	1183.1	1192.6
Pass	3	5E-5	5	1238.7	9052	92996	266	0	1110	0	1230.2	1239.6
Pass	3	5E-5	2	1240.2	8950	93248	263	0	1113	0	1231.7	1241.2
Pass	3	5E-5	1	1241.1	8304	91180	244	0	1089	0	1232.6	1242.1
Pass	3	5E-5	6	1274.3	8814	92441	259	0	1104	0	1265.8	1275.2
Pass	3	5E-5	10	1286.5	9528	94088	280	0	1123	0	1279.0	1288.5
Pass	3	5E-5	9	1305.9	8814	92240	259	0	1101	0	1297.4	1306.8
Pass	3	5E-5	4	1340.3	10242	95600	301	0	1141	0	1331.8	1341.2
Pass	3	1E-4	8	1459.6	9596	93668	282	0	1118	0	1451.1	1460.5
Pass	3	1E-4	7	1496.0	10004	94960	294	0	1134	0	1487.5	1496.9
Pass	3	1E-4	9	1497.2	9290	92996	273	0	1110	0	1488.7	1498.1
Pass	3	1E-4	10	1514.1	10378	94928	305	0	1133	0	1505.6	1515.0
Pass	3	1E-4	2	1551.4	9324	93280	274	0	1114	0	1542.9	1552.3
Pass	3	1E-4	5	1573.3	11500	97532	338	0	1164	0	1563.3	1573.3
Pass	3	1E-4	6	1608.1	11262	97280	331	0	1161	0	1599.6	1609.0
Pass	3	1E-4	4	1609.3	10718	95180	315	0	1136	0	1600.8	1610.3
Pass	3	1E-4	3	1627.4	10514	95180	309	0	1136	0	1619.9	1629.4
Pass	3	1E-4	1	1755.8	10820	96440	318	0	1151	0	1747.3	1756.8

Table B-12. Laboratory Data - Configuration 3 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
42	0	338.2	78%	48%
33	0	325.3	78%	45%
39	0	324.9	78%	45%
34	0	324.7	76%	47%
46	0	316.2	75%	46%
39	0	312.9	76%	45%
39	0	308.5	74%	46%
39	0	300.5	74%	44%
76	0	275.8	67%	45%
67	0	269.0	66%	44%
69	0	268.8	65%	45%
75	0	265.8	65%	44%
74	0	259.4	63%	45%
73	0	256.0	65%	42%
78	0	250.2	63%	43%
76	0	250.0	62%	44%
81	0	247.0	61%	44%
93	0	229.0	57%	43%

Table B-13. Laboratory Data - Configuration 4 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	4	1E-6	5	872.8	3752	80952	110	0	966	0	864.3	873.8
Pass	4	1E-6	3	874.4	3752	80952	110	0	966	0	862.0	875.3
Pass	4	1E-6	2	874.9	3862	81204	113	0	969	0	866.4	875.8
Pass	4	1E-6	10	875.0	3930	82800	115	0	988	0	909.6	919.1
Pass	4	1E-6	9	875.7	3896	81288	114	0	970	0	867.1	876.6
Pass	4	1E-6	4	876.0	3752	80952	110	0	966	0	863.5	876.9
Pass	4	1E-6	8	876.7	3938	81372	115	0	971	0	868.2	877.6
Pass	4	1E-6	7	881.5	3930	81288	115	0	970	0	869.1	882.4
Pass	4	1E-6	1	883.5	3998	81456	117	0	972	0	871.1	884.5
Pass	4	1E-6	6	918.4	3878	81152	113	0	969	0	909.3	919.3
Pass	4	1E-5	10	882.1	3972	81708	116	0	975	0	869.7	883.0
Pass	4	1E-5	5	884.7	4124	82044	120	0	979	0	872.2	885.6
Pass	4	1E-5	9	887.6	4234	82380	123	0	983	0	875.2	888.5
Pass	4	1E-5	2	902.9	4922	83640	143	0	998	0	890.5	903.8
Pass	4	1E-5	6	907.6	4302	82076	125	0	980	0	898.5	908.5
Pass	4	1E-5	4	914.4	4504	82580	130	0	986	0	905.4	915.4
Pass	4	1E-5	1	923.3	4276	82076	124	0	980	0	906.4	924.2
Pass	4	1E-5	3	931.0	4436	82360	128	0	984	0	918.1	932.0
Pass	4	1E-5	8	957.4	5200	84272	150	0	1008	0	932.7	958.2
Pass	4	1E-5	7	966.8	5488	84796	158	0	1013	0	953.9	967.7
Pass	4	2E-5	2	895.8	4716	82968	136	0	990	0	883.3	896.7
Pass	4	2E-5	9	906.4	5082	83976	147	0	1002	0	897.8	907.3
Pass	4	2E-5	1	918.6	5548	85152	160	0	1016	0	906.1	919.5
Pass	4	2E-5	6	923.9	5632	85404	162	0	1019	0	907.6	924.9
Pass	4	2E-5	4	925.5	5512	85320	158	0	1018	0	905.2	926.4
Pass	4	2E-5	5	949.1	5478	84628	157	0	1011	0	932.2	950.0
Pass	4	2E-5	8	952.0	5690	85016	163	0	1015	0	942.9	952.9
Pass	4	2E-5	10	956.4	5250	84240	151	0	1007	0	935.6	957.3
Pass	4	2E-5	7	964.2	5454	84512	157	0	1009	0	956.2	964.2
Pass	4	2E-5	3	1003.1	5876	85532	168	0	1023	0	978.4	1004.0
Pass	4	5E-5	9	942.4	6468	87504	184	0	1044	0	931.1	944.4
Pass	4	5E-5	5	966.9	6614	87924	189	0	1049	0	950.6	967.8

Table B-14. Laboratory Data - Configuration 4 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	463.0	96%	55%
0	0	464.3	96%	55%
1	0	461.9	97%	55%
1	0	440.0	94%	54%
1	0	461.5	97%	55%
0	0	463.4	96%	55%
2	0	461.0	97%	54%
1	0	460.5	96%	55%
1	0	459.4	96%	54%
1	0	440.1	92%	55%
2	0	460.2	96%	54%
4	0	458.8	96%	54%
5	0	457.3	97%	54%
6	0	449.4	96%	52%
3	0	445.4	94%	54%
7	0	442.0	94%	53%
5	0	441.5	92%	54%
5	0	435.9	92%	54%
9	0	429.1	92%	52%
11	0	419.5	91%	51%
10	0	453.0	96%	53%
11	0	445.7	96%	52%
13	0	441.6	96%	51%
16	0	440.9	96%	51%
19	0	442.1	96%	51%
14	0	429.3	93%	52%
15	0	424.4	93%	51%
10	0	427.7	92%	52%
14	0	418.5	91%	52%
16	0	409.0	89%	51%
31	0	429.8	96%	49%
29	0	421.0	95%	49%

Table B-15. Laboratory Data - Configuration 4 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	4	5E-5	10	974.2	6378	86528	179	0	1033	0	965.2	975.1
Pass	4	5E-5	1	1001.7	6776	87484	190	0	1045	0	985.3	1002.6
Pass	4	5E-5	2	1006.8	6670	87820	189	0	1049	0	991.0	1008.8
Pass	4	5E-5	3	1011.2	6600	87284	186	0	1042	0	1003.2	1013.2
Pass	4	5E-5	7	1013.5	7450	89132	211	0	1064	0	1004.4	1014.4
Pass	4	5E-5	4	1020.7	6568	87348	186	0	1044	0	992.2	1021.7
Pass	4	5E-5	8	1058.1	7446	89460	209	0	1071	0	1017.9	1059.1
Pass	4	5E-5	6	1079.8	7594	89544	215	0	1072	0	1035.6	1080.7
Pass	4	1E-4	6	998.6	7242	88656	199	0	1060	0	978.4	999.5
Pass	4	1E-4	4	1025.0	6838	88712	185	0	1059	0	1015.9	1025.9
Pass	4	1E-4	10	1027.1	7020	89500	192	0	1069	0	1014.1	1028.0
Pass	4	1E-4	9	1043.8	8042	90612	223	0	1081	0	996.2	1044.7
Pass	4	1E-4	2	1058.6	7458	89784	203	0	1073	0	1050.1	1059.5
Pass	4	1E-4	1	1063.1	7270	89416	197	0	1068	0	1046.3	1064.1
Pass	4	1E-4	7	1067.1	7084	88628	192	0	1058	0	1058.1	1068.1
Pass	4	1E-4	8	1070.3	7630	88760	209	0	1060	0	1061.3	1071.2
Pass	4	1E-4	3	1116.4	8934	93584	245	0	1117	0	1099.6	1117.3
Pass	4	1E-4	5	1146.2	9102	93212	253	0	1113	0	1137.2	1147.1

Table B-16. Laboratory Data - Configuration 4 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
36	0	414.6	92%	50%
39	0	406.1	91%	49%
37	0	403.8	91%	49%
38	0	398.9	90%	49%
39	0	398.4	92%	48%
32	0	403.3	89%	49%
45	0	393.2	88%	48%
35	0	386.4	86%	48%
63	0	409.0	92%	48%
77	0	393.9	90%	49%
70	0	394.6	91%	48%
68	0	401.7	90%	47%
78	0	381.1	88%	48%
75	0	382.5	88%	48%
69	0	378.2	86%	48%
65	0	377.1	86%	48%
90	0	364.0	87%	45%
71	0	351.9	85%	45%

Table B-17. Laboratory Data - Configuration 5 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	5	1E-6	10	598.3	944	55364	1	75	1	661	592.2	599.2
Pass	5	1E-6	9	599.3	944	55364	1	75	1	661	593.2	600.2
Pass	5	1E-6	4	599.6	944	55364	1	75	1	661	593.5	600.5
Pass	5	1E-6	6	599.9	944	55364	1	75	1	661	593.9	600.9
Pass	5	1E-6	1	600.2	944	55364	1	75	1	661	594.1	601.1
Pass	5	1E-6	2	600.5	944	55364	1	75	1	661	594.5	601.4
Pass	5	1E-6	3	600.5	944	55364	1	75	1	661	594.5	601.5
Pass	5	1E-6	5	604.2	956	55364	1	76	1	661	595.6	605.1
Pass	5	1E-6	7	614.9	1028	55800	1	82	1	667	606.4	615.8
Pass	5	1E-6	8	617.1	1088	56204	1	87	1	671	611.0	618.0
Pass	5	1E-5	10	612.7	1088	56204	1	87	1	671	606.6	613.6
Pass	5	1E-5	8	617.1	1112	56456	1	89	1	674	611.1	618.0
Pass	5	1E-5	1	626.0	1304	57632	1	105	1	688	620.0	627.0
Pass	5	1E-5	7	628.1	1304	57632	1	105	1	688	622.0	629.0
Pass	5	1E-5	5	631.0	1100	56136	1	88	1	671	625.0	631.9
Pass	5	1E-5	4	636.1	1256	57324	1	101	1	686	622.6	637.0
Pass	5	1E-5	2	641.3	1364	57968	1	110	1	692	627.7	642.2
Pass	5	1E-5	9	645.8	1496	58808	1	121	1	702	639.7	646.7
Pass	5	1E-5	3	657.1	1544	59228	1	125	1	707	651.1	658.0
Pass	5	1E-5	6	749.2	1436	58248	1	116	1	697	743.1	750.1
Pass	5	2E-5	2	632.9	1268	57312	1	102	1	685	626.9	633.9
Pass	5	2E-5	5	639.5	1400	58304	1	113	1	696	633.4	640.4
Pass	5	2E-5	3	645.7	1316	57648	1	106	1	689	637.1	646.6
Pass	5	2E-5	4	657.7	1556	59228	1	126	1	707	649.1	658.6
Pass	5	2E-5	1	662.6	1352	57900	1	109	1	692	656.5	663.5
Pass	5	2E-5	10	666.5	1574	59290	1	127	1	712	650.5	671.0
Pass	5	2E-5	7	688.9	1640	59732	1	133	1	713	680.3	689.8
Pass	5	2E-5	9	823.2	1340	57816	1	108	1	691	817.1	824.1
Pass	5	2E-5	8	851.3	1736	60336	1	141	1	721	845.2	852.2
Pass	5	2E-5	6	649.8	1412	58152	1	114	1	695	643.7	650.7
Pass	5	5E-5	3	716.0	2360	64436	1	193	1	769	705.0	717.0
Pass	5	5E-5	6	729.4	2144	62856	1	175	1	751	720.8	730.3

Table B-18. Laboratory Data - Configuration 5 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	675.5	96%	82%
0	0	674.4	96%	82%
0	0	674.0	96%	82%
0	0	673.7	96%	82%
0	0	673.4	96%	82%
0	0	673.0	96%	82%
0	0	673.0	96%	82%
0	0	671.7	95%	82%
2	0	659.8	94%	82%
2	0	654.7	95%	81%
3	0	659.5	95%	81%
4	0	654.7	95%	80%
5	0	645.3	96%	78%
6	0	643.2	95%	78%
2	0	640.1	93%	81%
5	0	642.6	94%	79%
7	0	637.3	94%	78%
10	0	625.4	95%	77%
13	0	614.5	94%	76%
11	0	538.4	81%	77%
7	0	638.2	94%	79%
10	0	631.6	95%	77%
6	0	627.9	93%	78%
13	0	616.3	94%	76%
14	0	609.4	91%	78%
13	0	615.0	92%	76%
19	0	588.1	90%	75%
11	0	489.6	73%	78%
13	0	473.3	74%	74%
9	0	621.5	93%	78%
26	0	567.5	94%	69%
30	0	555.0	90%	71%

Table B-19. Laboratory Data - Configuration 5 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	5	5E-5	9	731.0	1748	60336	1	142	1	721	725.0	732.0
Pass	5	5E-5	5	809.7	2084	62604	1	170	1	748	803.7	810.7
Pass	5	5E-5	7	886.7	1976	61812	1	161	2	738	880.6	887.6
Pass	5	5E-5	2	898.4	2024	62520	1	165	1	747	892.4	899.3
Pass	5	5E-5	8	917.7	1868	60713	1	152	1	729	911.5	918.5
Pass	5	5E-5	1	997.3	2036	62100	1	166	1	742	991.2	998.2
Pass	5	5E-5	4	1086.4	1988	62268	1	162	1	744	1080.3	1087.3
Pass	5	5E-5	10	1349.7	1868	61680	1	152	1	737	1343.7	1350.7
Pass	5	1E-4	4	1432.2	1952	62100	1	159	1	742	1426.2	1433.2
Pass	5	1E-4	5	1556.1	2240	62604	1	183	1	748	1550.0	1557.0
Pass	5	1E-4	7	1671.9	1856	60756	1	151	1	726	1665.8	1672.8
Pass	5	1E-4	3	1961.9	2228	62688	1	182	1	749	1955.8	1962.8
Pass	5	1E-4	6	1972.8	2312	63108	1	189	1	754	1966.8	1973.7
Pass	5	1E-4	2	2004.7	2312	63612	1	189	1	760	1998.6	2005.6
Pass	5	1E-4	1	1688.3	2264	63276	1	185	1	756	1682.2	1689.2
Pass	5	1E-4	9	2242.6	2360	64116	1	193	1	766	2236.6	2243.5
Pass	5	1E-4	10	2476.7	2408	64032	1	197	1	765	2470.7	2477.6
Pass	5	1E-4	8	2489.2	2144	62940	1	175	1	752	2483.2	2490.1

Table B-20. Laboratory Data - Configuration 5 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughpu t (bps)	Link Util	Bit Eff
35	0	551.8	86%	74%
34	0	497.8	80%	71%
27	0	454.3	73%	72%
36	0	448.3	72%	71%
27	0	438.9	69%	74%
30	0	403.6	65%	72%
35	0	370.3	60%	72%
40	0	297.7	48%	73%
52	0	280.5	45%	72%
74	0	258.1	42%	71%
49	0	240.2	38%	74%
68	0	204.6	33%	71%
68	0	203.4	33%	70%
65	0	200.2	33%	70%
60	0	237.8	39%	70%
71	0	178.9	30%	69%
78	0	161.9	27%	69%
71	0	161.1	26%	71%

Table B-21. Laboratory Data - Configuration 6 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	6	1E-6	3	600.0	948	55364	1	75	1	661	593.9	600.9
Pass	6	1E-6	8	600.1	948	55364	1	75	1	661	594.1	601.0
Pass	6	1E-6	5	600.6	948	55364	1	75	1	661	594.5	601.5
Pass	6	1E-6	7	600.7	948	55364	1	75	1	661	594.6	601.6
Pass	6	1E-6	2	600.9	948	55364	1	75	1	661	594.9	601.9
Pass	6	1E-6	4	601.1	988	55868	1	77	1	667	595.1	602.1
Pass	6	1E-6	1	603.9	960	55364	1	76	1	661	595.3	604.8
Pass	6	1E-6	10	604.2	1016	55700	1	80	1	665	598.1	605.1
Pass	6	1E-6	6	604.6	1016	55700	1	80	1	665	598.5	605.5
Pass	6	1E-6	9	613.4	1128	56288	1	88	1	672	605.4	613.4
Pass	6	1E-5	7	605.8	980	55868	1	77	1	667	597.2	606.7
Pass	6	1E-5	3	606.3	1036	55784	1	81	1	666	597.7	607.2
Pass	6	1E-5	9	606.6	1060	55952	1	83	1	668	600.5	607.5
Pass	6	1E-5	4	607.3	1060	56036	1	83	1	669	601.2	608.2
Pass	6	1E-5	2	609.5	1100	56120	1	85	1	670	603.5	610.5
Pass	6	1E-5	5	613.0	1128	56288	1	88	1	672	604.4	613.9
Pass	6	1E-5	6	614.3	1128	56288	1	88	1	672	605.7	615.2
Pass	6	1E-5	10	626.9	1300	57716	1	101	1	689	615.8	627.8
Pass	6	1E-5	8	638.8	1276	56988	1	97	1	682	624.7	639.7
Pass	6	1E-5	1	714.8	1260	56976	1	97	1	681	708.2	715.7
Pass	6	2E-5	9	612.0	1184	56456	1	90	1	674	606.0	613.0
Pass	6	2E-5	8	612.4	1132	56204	1	87	1	671	603.8	613.3
Pass	6	2E-5	6	621.6	1300	57008	1	97	2	680	613.0	622.5
Pass	6	2E-5	10	622.1	1328	57296	1	100	1	684	613.5	623.0
Pass	6	2E-5	1	626.0	1368	58052	1	104	1	693	619.9	626.9
Pass	6	2E-5	7	635.1	1536	58724	1	116	1	701	629.0	636.0
Pass	6	2E-5	3	640.2	1628	59228	1	121	1	707	631.7	641.1
Pass	6	2E-5	4	656.3	1534	58174	1	114	1	697	636.7	657.2
Pass	6	2E-5	5	713.1	1380	57564	1	105	1	688	706.6	714.1
Pass	6	2E-5	2	715.6	1232	56724	1	94	1	678	709.1	716.5
Pass	6	5E-5	10	626.3	1400	57548	1	104	1	687	620.2	627.2
Pass	6	5E-5	2	641.2	1668	59144	1	121	1	706	632.6	642.1

Table B-22. Laboratory Data - Configuration 6 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	673.6	96%	82%
0	0	673.4	96%	82%
0	0	673.0	96%	82%
0	0	672.8	96%	82%
0	0	672.5	96%	82%
2	0	672.3	97%	82%
0	0	672.0	95%	82%
1	0	668.9	96%	82%
1	0	668.4	96%	82%
3	0	660.8	96%	81%
1	0	669.8	96%	82%
2	0	669.3	96%	82%
2	0	666.2	96%	81%
3	0	665.4	96%	81%
4	0	662.9	96%	81%
3	0	661.9	96%	81%
3	0	660.5	95%	81%
6	0	649.6	96%	78%
6	0	640.4	93%	79%
4	0	564.9	83%	79%
8	0	660.2	96%	80%
6	0	662.6	95%	81%
12	0	652.6	95%	79%
12	0	652.1	96%	79%
10	0	645.4	96%	78%
13	0	636.0	96%	77%
16	0	633.4	96%	76%
13	0	628.4	92%	77%
8	0	566.2	84%	78%
6	0	564.2	82%	80%
13	0	645.1	96%	78%
23	0	632.4	96%	76%

Table B-23. Laboratory Data - Configuration 6 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	6	5E-5	7	644.9	1696	59984	1	126	1	716	636.4	645.8
Pass	6	5E-5	6	670.7	1796	59328	1	127	1	709	661.6	671.6
Pass	6	5E-5	9	685.9	1762	59562	1	127	1	717	666.9	690.3
Pass	6	5E-5	4	695.0	2040	61008	1	146	1	729	689.0	696.0
Pass	6	5E-5	1	700.0	2260	63036	1	167	1	754	688.4	700.9
Pass	6	5E-5	5	735.2	1864	60000	1	136	1	717	728.6	736.1
Pass	6	5E-5	8	737.4	1664	58908	1	122	1	704	730.9	738.4
Pass	6	5E-5	3	770.6	1848	59930	1	132	1	717	764.0	771.5
Pass	6	1E-4	3	652.2	1792	59083	1	120	1	707	646.1	653.1
Pass	6	1E-4	9	686.8	2016	60945	1	134	1	731	673.3	687.8
Pass	6	1E-4	7	691.3	2284	62196	1	157	1	744	679.7	692.2
Pass	6	1E-4	8	707.4	2308	62880	1	161	1	753	693.4	708.4
Pass	6	1E-4	2	756.4	1832	60000	1	124	1	717	749.9	757.4
Pass	6	1E-4	6	759.8	1920	59580	1	126	1	712	753.2	760.7
Pass	6	1E-4	5	768.3	1976	60420	1	134	1	722	761.7	769.2
Pass	6	1E-4	4	773.7	1980	60341	1	131	2	724	767.2	774.7
Pass	6	1E-4	1	778.0	2072	61596	1	144	1	736	771.4	778.9
Pass	6	1E-4	10	839.4	2200	62028	1	154	1	742	832.8	840.3

Table B-24. Laboratory Data - Configuration 6 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
21	0	628.7	97%	75%
23	0	604.7	92%	76%
26	0	599.9	90%	75%
27	0	580.7	91%	73%
31	0	581.2	94%	71%
21	0	549.1	85%	75%
18	0	547.4	83%	76%
27	0	523.6	81%	75%
44	0	619.2	94%	76%
57	0	594.2	92%	73%
48	0	588.6	94%	72%
51	0	577.0	92%	71%
43	0	533.5	83%	75%
46	0	531.1	82%	75%
43	0	525.2	82%	74%
53	0	521.5	81%	74%
45	0	518.6	82%	73%
45	0	480.4	77%	72%

Table B-25. Laboratory Data - Configuration 7 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	7	1E-6	9	670.8	1736	61940	1	84	1	739	663.7	671.7
Pass	7	1E-6	3	671.0	1736	61940	1	84	1	739	663.9	671.9
Pass	7	1E-6	4	671.1	1736	61940	1	84	1	739	664.0	672.1
Pass	7	1E-6	6	671.6	1736	61940	1	84	1	739	664.5	672.5
Pass	7	1E-6	8	671.8	1736	61940	1	84	1	739	664.7	672.7
Pass	7	1E-6	5	672.5	1736	61940	1	84	1	739	665.4	673.4
Pass	7	1E-6	2	674.3	1776	62108	1	86	1	741	667.2	675.2
Pass	7	1E-6	10	674.4	1796	62192	1	87	1	742	667.3	675.4
Pass	7	1E-6	1	678.7	1796	62108	1	87	1	741	668.5	679.6
Pass	7	1E-6	7	682.3	1936	62696	1	94	1	748	675.1	683.2
Pass	7	1E-5	3	680.4	1916	62612	1	93	1	747	673.3	681.3
Pass	7	1E-5	10	686.5	2016	62948	1	98	1	751	679.4	687.4
Pass	7	1E-5	1	688.9	2116	63368	1	103	1	756	681.8	689.8
Pass	7	1E-5	7	703.4	2236	63788	1	109	1	761	696.3	704.3
Pass	7	1E-5	6	705.3	2376	64292	1	116	1	767	695.2	706.2
Pass	7	1E-5	5	707.2	2396	64376	1	117	1	768	697.1	708.1
Pass	7	1E-5	2	709.3	2396	64712	1	117	1	772	693.2	710.3
Pass	7	1E-5	8	714.6	2496	64712	1	122	1	772	704.5	715.6
Pass	7	1E-5	4	718.6	2476	64628	1	121	1	771	708.5	719.5
Pass	7	1E-5	9	722.3	2436	64400	1	119	1	769	712.2	723.3
Pass	7	2E-5	4	706.9	2516	65012	1	123	2	775	699.8	707.9
Pass	7	2E-5	7	711.8	2456	64712	1	120	1	772	701.7	712.7
Pass	7	2E-5	8	721.5	2636	65552	1	129	1	782	705.4	722.5
Pass	7	2E-5	6	725.7	2196	63872	1	107	1	762	715.6	726.6
Pass	7	2E-5	5	738.3	2576	65072	1	126	1	777	731.2	739.3
Pass	7	2E-5	1	739.6	2656	65408	1	130	1	781	729.5	740.6
Pass	7	2E-5	10	749.8	2196	63644	1	107	1	760	742.7	750.7
Pass	7	2E-5	3	750.1	2536	64820	1	124	1	774	743.0	751.0
Pass	7	2E-5	9	770.7	2916	66316	1	143	2	792	760.6	771.6
Pass	7	2E-5	2	990.9	2996	66416	1	147	1	793	983.8	991.8
Pass	7	5E-5	4	764.1	3276	67808	1	161	2	809	757.0	765.1
Pass	7	5E-5	6	809.6	3576	68744	1	176	1	820	796.5	810.5

Table B-26. Laboratory Data - Configuration 7 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	603.3	96%	73%
0	0	603.1	96%	73%
0	0	603.0	96%	73%
0	0	602.6	96%	73%
0	0	602.4	96%	73%
0	0	601.7	96%	73%
1	0	600.1	96%	73%
1	0	600.0	96%	73%
1	0	598.9	95%	73%
1	0	593.0	96%	72%
2	0	594.7	96%	72%
2	0	589.3	95%	71%
4	0	587.2	96%	71%
8	0	575.0	94%	70%
6	0	575.9	95%	70%
7	0	574.4	95%	69%
7	0	577.6	95%	69%
9	0	568.3	94%	69%
8	0	565.1	94%	69%
6	0	562.2	93%	69%
9	0	572.1	96%	69%
12	0	570.6	95%	69%
10	0	567.6	95%	68%
10	0	559.5	92%	70%
12	0	547.6	92%	68%
10	0	548.8	92%	68%
6	0	539.1	88%	70%
8	0	538.9	90%	69%
16	0	526.4	90%	67%
14	0	407.0	70%	67%
20	0	528.9	92%	65%
31	0	502.7	88%	64%

Table B-27. Laboratory Data - Configuration 7 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	7	5E-5	1	816.9	3436	68348	1	169	1	816	809.8	817.8
Pass	7	5E-5	10	826.9	3536	68976	1	174	1	825	813.8	827.8
Pass	7	5E-5	7	844.8	4076	70952	1	201	1	847	837.7	845.7
Pass	7	5E-5	8	850.9	3876	69944	1	191	1	835	843.8	851.9
Pass	7	5E-5	2	856.8	3456	68516	1	170	1	818	849.6	857.7
Pass	7	5E-5	9	858.0	3416	68684	1	168	1	820	850.8	858.9
Pass	7	5E-5	3	872.7	3616	69692	1	178	1	832	865.6	873.6
Pass	7	5E-5	5	911.1	3396	68264	1	167	1	815	904.0	912.1
Pass	7	1E-4	3	828.1	3476	68912	1	171	1	822	821.0	829.0
Pass	7	1E-4	4	864.6	3556	68768	1	175	1	821	857.5	865.6
Pass	7	1E-4	9	950.2	4276	71540	1	211	1	854	943.1	951.1
Pass	7	1E-4	10	991.3	4316	72044	1	213	1	860	984.2	992.3
Pass	7	1E-4	2	1031.6	4716	73640	1	233	1	879	1024.5	1032.5
Pass	7	1E-4	1	1064.6	4516	72128	1	223	1	861	1057.5	1065.5
Pass	7	1E-4	6	1081.2	3876	70280	1	191	1	839	1074.1	1082.1
Pass	7	1E-4	8	1111.5	4676	73388	1	231	1	876	1104.3	1112.4
Pass	7	1E-4	5	1116.7	4556	73052	1	225	1	872	1109.6	1117.7
Pass	7	1E-4	7	1161.9	5036	74060	1	249	1	884	1154.8	1162.8

Table B-28. Laboratory Data - Configuration 7 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
21	0	494.4	87%	64%
29	0	492.0	87%	64%
30	0	478.0	87%	62%
34	0	474.5	86%	63%
30	0	471.3	83%	64%
33	0	470.6	83%	64%
38	0	462.6	83%	63%
24	0	442.9	78%	65%
45	0	487.7	87%	64%
37	0	466.9	83%	64%
46	0	424.6	78%	61%
59	0	406.8	76%	60%
62	0	390.8	74%	59%
52	0	378.6	71%	60%
65	0	372.8	68%	62%
72	0	362.6	69%	59%
67	0	360.8	68%	59%
66	0	346.7	66%	58%

Table B-29. Laboratory Data - Configuration 8 (1 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	8	1E-6	6	670.4	1740	61940	1	84	1	739	663.3	671.3
Pass	8	1E-6	9	670.5	1740	61940	1	84	1	739	663.4	671.5
Pass	8	1E-6	2	671.0	1740	61940	1	84	1	739	663.9	672.0
Pass	8	1E-6	10	671.1	1740	61940	1	84	1	739	664.0	672.0
Pass	8	1E-6	7	671.6	1768	62444	1	85	1	745	664.5	672.5
Pass	8	1E-6	3	673.6	1768	62024	1	85	1	740	666.5	674.5
Pass	8	1E-6	8	674.6	1808	62192	1	87	1	742	667.5	675.5
Pass	8	1E-6	5	677.6	1816	62108	1	87	1	741	667.5	678.6
Pass	8	1E-6	1	709.1	1924	62384	1	92	1	745	701.4	710.0
Pass	8	1E-6	4	734.4	1794	62042	1	86	1	741	667.2	735.3
Pass	8	1E-5	7	674.2	1796	62108	1	86	1	741	667.1	675.1
Pass	8	1E-5	6	680.2	1904	62864	1	91	1	750	670.1	681.2
Pass	8	1E-5	1	689.2	2188	63452	1	104	1	757	682.1	690.1
Pass	8	1E-5	2	692.5	2192	63536	1	105	1	758	682.4	693.4
Pass	8	1E-5	10	697.3	2268	63704	1	108	1	760	684.2	698.2
Pass	8	1E-5	8	701.7	2136	63244	1	101	1	756	691.1	702.7
Pass	8	1E-5	3	706.8	2436	64376	1	116	1	768	690.7	707.7
Pass	8	1E-5	9	715.2	1960	62552	1	93	1	747	707.6	716.1
Pass	8	1E-5	5	716.5	2264	63644	1	107	1	760	708.9	717.4
Pass	8	1E-5	4	719.6	2360	64040	1	111	1	767	699.9	720.5
Pass	8	2E-5	6	699.1	2416	64544	1	115	1	770	691.9	700.0
Pass	8	2E-5	2	701.1	2452	64964	1	116	1	775	690.9	702.0
Pass	8	2E-5	7	706.0	2560	64796	1	121	1	773	695.9	706.9
Pass	8	2E-5	1	708.2	2232	63476	1	105	1	758	700.6	709.1
Pass	8	2E-5	8	710.3	2588	65132	1	122	1	777	694.2	711.2
Pass	8	2E-5	10	719.9	2548	64904	1	120	1	775	712.3	720.8
Pass	8	2E-5	4	727.4	2540	64652	1	118	1	772	719.7	728.3
Pass	8	2E-5	9	750.7	2976	66328	1	141	1	795	725.0	751.6
Pass	8	2E-5	5	766.9	3404	67928	1	160	1	811	759.3	767.8
Pass	8	2E-5	3	777.6	2632	64988	1	123	1	776	770.0	778.5
Pass	8	5E-5	4	730.9	2864	65868	1	131	1	788	722.9	730.9
Pass	8	5E-5	10	733.1	2876	65912	1	134	1	787	725.9	734.0

Table B-30. Laboratory Data - Configuration 8 (2 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
0	0	603.6	96%	73%
0	0	603.5	96%	73%
0	0	603.0	96%	73%
0	0	603.0	96%	73%
1	0	602.6	97%	72%
1	0	600.7	96%	73%
1	0	599.8	96%	73%
2	0	599.8	95%	73%
2	0	570.8	92%	72%
2	0	600.1	88%	73%
3	0	600.2	96%	73%
3	0	597.5	96%	72%
6	0	587.0	96%	71%
4	0	586.7	95%	71%
6	0	585.2	95%	70%
6	0	579.4	94%	71%
7	0	579.7	95%	69%
3	0	565.9	91%	72%
6	0	564.8	92%	70%
8	0	572.0	93%	70%
8	0	578.6	96%	69%
10	0	579.5	96%	69%
11	0	575.4	95%	69%
9	0	571.5	93%	71%
11	0	576.8	95%	68%
12	0	562.1	94%	69%
15	0	556.3	92%	69%
12	0	552.2	92%	67%
19	0	527.3	92%	65%
14	0	520.0	87%	69%
25	0	553.9	94%	67%
20	0	551.5	94%	67%

Table B-31. Laboratory Data - Configuration 8 (3 of 4)

Result	Config	BER	Rep	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time	Elapsed Time (tcpdump)
Pass	8	5E-5	2	738.8	2960	66184	1	137	1	791	728.2	739.8
Pass	8	5E-5	1	749.1	2500	64400	1	114	1	769	741.5	750.0
Pass	8	5E-5	6	755.0	3020	66332	1	136	1	792	747.3	755.9
Pass	8	5E-5	7	755.2	2792	65408	1	127	1	781	747.6	756.2
Pass	8	5E-5	9	762.9	3400	68200	1	153	1	815	749.2	763.8
Pass	8	5E-5	8	764.9	2924	66164	1	130	1	790	757.3	765.8
Pass	8	5E-5	3	781.8	3588	68240	1	164	1	817	759.1	782.7
Pass	8	5E-5	5	794.4	3972	70004	1	186	1	838	768.7	795.2
Pass	8	1E-4	2	776.1	3268	67044	1	144	1	802	762.4	777.0
Pass	8	1E-4	8	778.5	3096	66752	1	131	1	797	770.8	779.4
Pass	8	1E-4	4	779.9	3888	69416	1	175	1	831	763.2	780.8
Pass	8	1E-4	3	784.4	3788	68836	1	166	2	822	773.7	785.3
Pass	8	1E-4	1	795.7	3216	68012	1	137	1	812	788.1	796.6
Pass	8	1E-4	6	801.6	3644	69188	1	156	1	826	794.0	802.5
Pass	8	1E-4	5	804.7	3364	67508	1	150	1	806	797.1	805.6
Pass	8	1E-4	9	810.4	3636	68684	1	160	1	820	803.3	811.3
Pass	8	1E-4	10	810.8	4172	70784	1	186	1	845	803.2	811.7
Pass	8	1E-4	7	828.9	3452	68264	1	150	1	815	821.3	829.9

Table B-32. Laboratory Data - Configuration 8 (4 of 4)

Pkt Drops (Dn)	Pkt Drops (Up)	Throughput (bps)	Link Util	Bit Eff
20	0	549.8	93%	67%
21	0	540.0	89%	69%
32	0	535.7	91%	67%
25	0	535.5	90%	68%
41	0	534.4	93%	65%
37	0	528.7	90%	67%
29	0	527.5	91%	64%
25	0	520.9	92%	62%
42	0	525.2	90%	66%
55	0	519.4	89%	66%
46	0	524.6	93%	63%
52	0	517.5	91%	64%
63	0	508.1	89%	65%
67	0	504.3	90%	64%
43	0	502.3	87%	65%
58	0	498.4	88%	64%
55	0	498.5	91%	62%
59	0	487.5	86%	65%

The column headings for the field test data tables are as follows:

Result	An indication of whether the test passed or failed. Failed tests have been sorted out of the data set, and do not appear in these tables
Config	The protocol configuration (1 through 8) under test
Date	The date of the test
Start Time	The time *in British local time, not GMT*, that the test began.
End Time	The time that the test ended.
Elapsed Time (SCPS-TP)	The amount of time reported by the SCPS-TP responder between the beginning and end of the connection
Bytes Up	The number of bytes of data in SCPS-TP packets (including SCPS-TP overhead, but not CCSDS path packet overhead) transmitted by the SCPS Workstation to STRV 1b
Bytes Down	The number of bytes of data in valid SCPS-TP packets *received by the SCPS Workstation* (does NOT account for lost or corrupted data)
Pkts Up	The number of packets transmitted by the Ground System. The "U" designation means uncompressed, "C" means compressed.
Pkts Down	The number of valid SCPS-TP packets received by the SCPS workstation
Data Xfer Time	The time from when the first data packet was received until the last data packet was acknowledged.
Elapsed Time (tcpdump)	The amount of time between the beginning and end of the connection, determined by post processing tcpdump output.
Pkt Drops	The number of packets lost or corrupted in transmission. "Up" means that a packet destined for the STRV 1b was lost or corrupted, "Down" means that a packet destined for the SCPS Workstation was lost or corrupted.
Max Path Seq No.	The maximum CCSDS Telemetry packet sequence number appearing on the SCPS-TP connection.
q-down	The quotient of the number of downlink packets received divided by the number of packets sent. (Probability of success of a downlink transmission.)
Average Pkt Size	The size in bytes resulting from dividing the bytes down entry by the pkts down entry (used in calculating bit-error rate).

BER	The bit-error rate estimated from packet loss. Per Equation 9, assumes errors follow a Bernoulli process.
Notes	Notes pertaining to the run. Specifically, “N” means that bit-errors were naturally-occurring; “A” means that bit-errors were created by steering the antenna off track; “I” means that bit-errors were induced by cycling power on the frame synchronizer. The “I” is typically followed by a string such as “1/3 min”, which indicates that the power was cycled once every three minutes. Other notes: *1 - the Telecommand Workstation had to be restarted during the run; *2 - the Telecommand Workstation failed during the run; *3 - the protocol endured a sustained loss at the very end of the test.
Throughput	The throughput of the connection per Equation 1, in bits per second.
Link Utilization	The down link utilization per Equation 3, expressed as a percentage.
Bit Efficiency	The bit-efficiency of the connection per Equation 7, expressed as a percentage.
Frame Analysis BER	An independent check on the bit-error rate performed by analyzing the telemetry workstation logs on a frame-by-frame basis.

Table B-33. Field Test Data (1 of 4)

Result	Config	Date	Start Time	End Time	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time
Pass	resp_1	24-Jul-96	18:33:27	18:45:34	718.9	1988	65796	90	0	786	0	704
Pass	resp_1	24-Jul-96	15:59:05	16:13:58	882.5	2560	66784	116	0	797	0	722
Pass	resp_1	24-Jul-96	14:28:18	14:40:59	753.0	2538	67160	115	0	803	0	738
Pass	resp_1	24-Jul-96	19:21:49	19:51:20	1763.2	3726	67832	169	0	811	0	1751
Pass	resp_1	31-Jul-96	11:15:10	11:51:28	2169.9	4250	67308	190	0	804	0	2157
Pass	resp_1	31-Jul-96	12:45:47	13:12:28	1592.7	3550	67328	161	0	805	0	1578
Pass	resp_1	31-Jul-96	13:19:22	13:38:10	1120.4	3088	67056	140	0	801	0	1106
Pass	resp_1	31-Jul-96	13:41:53	13:57:56	955.1	2824	67076	128	0	802	0	943
Pass	resp_1	19-Jul-96	13:15:16	13:47:11	1902.3	2956	67434	134	0	807	0	1890
Pass	resp_5	24-Jul-96	16:31:35	16:41:51	608.7	956	55364	1	76	1	661	594
Pass	resp_5	24-Jul-96	14:47:30	14:58:13	634.9	1388	56960	1	112	1	680	623
Pass	resp_5	25-Jul-96	11:23:54	11:36:09	726.8	1460	56976	1	118	1	681	714
Pass	resp_5	25-Jul-96	11:43:56	11:54:12	608.8	1016	55616	1	81	1	664	596
Pass	resp_5	25-Jul-96	12:29:38	13:13:45	2639.6	1736	56464	1	134	1	675	2562
Pass	resp_5	25-Jul-96	13:57:24	14:07:59	627.1	1076	55896	1	86	1	669	615
Pass	resp_5	22-Jul-96	13:25:13	13:47:55	1353.9	1298	55998	1	104	1	673	1326
Pass	resp_5	19-Jul-96	12:36:20	13:06:36	1807.5	1736	57294	1	141	1	698	1795
Pass	resp_6	24-Jul-96	17:46:58	17:57:12	603.5	948	55364	1	75	1	661	594
Pass	resp_6	24-Jul-96	13:46:27	13:57:02	624.5	1248	56204	1	94	1	671	613
Pass	resp_6	22-Jul-96	11:48:08	11:59:44	687.5	1836	60672	1	129	1	725	675
Pass	resp_6	19-Jul-96	11:15:00	11:26:56	708.5	2296	60572	1	158	1	723	697
Pass	resp_6	24-Jul-96	13:01:38	13:21:11	1162.4	2452	60992	1	155	1	728	1152
Pass	resp_6	19-Jul-96	15:29:47	16:08:42	2324.7	3192	75501	1	194	4	905	2315
Pass	resp_6	19-Jul-96	13:53:39	14:04:30	640.3	1264	57296	1	98	1	684	630
Pass	resp_7	24-Jul-96	17:32:09	17:43:34	674.4	1736	61940	1	84	1	739	664
Pass	resp_7	24-Jul-96	15:42:57	15:54:43	695.4	2056	62948	1	100	1	751	685
Pass	resp_7	24-Jul-96	13:28:39	13:41:52	781.9	2636	64232	1	129	1	767	772

Table B-34. Field Test Data (2 of 4)

Elapsed Time tcpdump	Pkt Drops (Dn)	Pkt Drops (Up)	Max Path Seq No.	q-down	Avg Pkt Size (dn)	BER Down	Notes	Through put (bps)	Link Utilization	Bit Efficiency	Frame Analysis BER
716.3	0	0	786	1.000	90	1.0e-6	N	569	95%	69%	1.00E-06
882.5	5	0	801	0.995	90	7.0e-6	N	555	79%	67%	7.13E-05
750.4	10	0	812	0.989	90	1.6e-5	N	543	94%	66%	1.93E-05
1763.0	55	0	866	0.936	90	9.1e-5	N	229	43%	61%	6.39E-04
2167.3	72	0	876	0.918	90	1.2e-4	N	186	35%	60%	1.09E-04
1590.1	45	0	850	0.947	90	7.6e-5	A	254	46%	62%	3.02E-04
1117.8	30	0	831	0.964	90	5.1e-5	A	362	65%	64%	7.93E-05
952.4	24	0	824	0.973	90	3.8e-5	A	425	75%	65%	8.60E-05
1902.3	33	0	835	0.966	90	4.8e-5	I 1/3min				3.56E-06
606.1	0	0	662	1.000	90	1.0e-6	N	674	95%	82%	1.00E-06
632.3	16	0	697	0.977	90	3.2e-5	N	642	96%	77%	2.79E-05
724.2	22	0	703	0.970	90	4.2e-5	N	560	84%	77%	3.65E-05
606.1	2	0	667	0.997	90	4.2e-6	N	671	95%	82%	3.22E-06
2637.0	42	0	718	0.942	90	8.4e-5	A	156	24%	75%	8.42E-05
624.5	4	0	674	0.994	89	8.3e-6	A	651	93%	81%	6.27E-06
1351.3	15	0	689	0.978	89	3.1e-5	I 1/3 min				
1805.2	31	0	730	0.958	88	6.2e-5	I 1/3min				
603.3	0	0	662	1.000	90	1.0e-6	N	674	95%	82%	1.00E-06
624.3	10	0	682	0.985	90	2.1e-5	N	653	95%	79%	2.20E-05
684.9	34	0	760	0.955	90	6.4e-5	N *1	593	96%	71%	5.58E-05
705.8	53	0	777	0.932	90	9.8e-5	N	574	96%	69%	8.79E-05
1162.2	158	0	886	0.823	90	2.7e-4	N	347	66%	60%	2.27E-04
2324.6	231	0	1140	0.797	89	3.2e-4	N	172	42%	47%	3.48E-03
640.1	18	0	698	0.981	90	2.6e-5	I 1/3min				
674.2	0	0	740	1.000	90	1.0e-6	N	603	95%	73%	1.00E-06
695.2	5	0	757	0.993	90	9.2e-6	N	584	95%	71%	7.06E-06
781.8	19	0	787	0.976	90	3.4e-5	N	518	87%	68%	2.88E-05

Table B-35. Field Test Data (3 of 4)

Result	Config	Date	Start Time	End Time	Elapsed Time (SCPS-TP)	Bytes Up	Bytes Down	Pkts Up - U	Pkts Up - C	Pkts Dn - U	Pkts Dn - C	Data Xfer Time
Pass	resp_7	19-Jul-96	14:55:40	15:08:45	774.0	2676	64148	1	131	1	766	764
Pass	resp_7	19-Jul-96	11:47:51	12:00:16	734.7	2816	64460	1	138	1	769	722
Pass	resp_7	31-Jul-96	14:57:43	15:21:02	1388.0	3836	64988	1	189	1	776	1378
Pass	resp_7	31-Jul-96	15:40:45	15:54:26	810.8	2716	64316	1	133	1	768	801
Pass	resp_7	18-Jul-96	18:15:41	18:32:49	1018.0	3096	64544	1	152	1	770	1008
Pass	resp_7	16-Jul-96	14:06:53	14:35:39	1716.0	3600	64820	1	174	1	774	1706
Pass	resp_8	24-Jul-96	17:17:14	17:28:49	684.8	1760	62117	1	85	1	750	675
Pass	resp_8	24-Jul-96	18:51:57	19:03:48	700.5	2116	62864	1	100	1	750	689
Pass	resp_8	18-Jul-96	17:20:40	17:32:31	700.6	2000	62444	1	95	1	745	691
Fail	resp_8	19-Jul-96	11:30:29	11:44:23	823.7	3148	64924	1	142	1	776	811
Pass	resp_8	18-Jul-96	12:17:50	12:32:16	855.4	2384	63200	1	113	1	754	845
Pass	resp_8	19-Jul-96	14:39:40	14:52:00	729.6	2424	63788	1	113	1	761	718
Pass	resp_8	16-Jul-96	13:25:54	13:38:30	745.2	2328	63224	1	109	1	755	733
Pass	resp_8	16-Jul-96	13:50:10	14:03:07	766.1	2452	63460	1	114	2	758	754
Pass	resp_8	25-Jul-96	10:17:28	10:37:33	1193.8	4392	68852	1	179	1	822	1181
Pass	resp_8	25-Jul-96	10:45:07	10:59:33	855.4	3832	67632	1	161	1	809	845
Pass	resp_8	31-Jul-96	15:24:37	15:37:05	737.3	2932	64292	1	132	1	767	725

Table B-36. Field Test Data (4 of 4)

Elapsed Time tcpdump	Pkt Drops (Dn)	Pkt Drops (Up)	Max Path Seq No.	q-down	Avg Pkt Size (dn)	BER Down	Notes	Through put (bps)	Link Utilization	Bit Efficiency	Frame Analysis BER
773.8	28	0	793	0.967	90	4.6e-5	N	524	89%	67%	5.52E-05
734.5	29	0	797	0.966	90	4.8e-5	N	554	94%	67%	4.53E-05
1387.8	77	0	853	0.911	90	1.3e-4	A	290	53%	61%	2.74E-04
810.6	26	0	794	0.969	90	4.5e-5	A	499	85%	67%	6.66E-05
1017.9	42	0	813	0.948	90	7.4e-5	I 1/3min				
1715.8	72	0	841	0.922	90	1.1e-4	I 1/min				
684.6	0	0	751	1.000	89	1.0e-6	N	593	94%	73%	1.00E-06
700.4	8	0	758	0.991	90	1.3e-5	N	581	94%	71%	1.12E-05
700.4	23	0	769	0.970	90	4.2e-5	I 1/3min	579	95%	70%	
823.5	32	0	809	0.960	90	5.6e-5	N *2				
855.2	35	0	790	0.956	90	6.3e-5	I *3	474	80%	68%	
729.4	37	0	798	0.955	90	6.4e-5	I 1/3min	558	95%	67%	
745.0	49	0	802	0.943	90	8.2e-5	I 1/min	546	93%	67%	
765.9	67	0	821	0.926	90	1.1e-4	I 1/min	531	93%	65%	
1193.6	174	0	997	0.825	90	2.7e-4	A	339	73%	53%	4.17E-04
855.2	106	0	914	0.886	89	1.7e-4	A	474	93%	58%	1.77E-04
737.1	38	0	806	0.953	90	6.7e-5	A	552	95%	66%	5.31E-05

Appendix C

Lessons Learned

This Appendix presents a list of the lessons resulting from the conduct of the SCPS-TP testing in the SSFE.

1. Ensure that plans and schedules include high fidelity laboratory testing prior to field testing.

   The SCPS test bed was vital in identifying and correcting the implementation errors that had proved impossible to find and correct in the field. We did not perform high fidelity laboratory testing for the SSFE before the initial code uploads, partially due to schedule considerations and partially due to the logistics involved with establishing the test configuration in England. As a result of this, we were forced to retest the SCPS-TP after correcting the implementation errors we eventually discovered in the laboratory. In the future, we should ensure that protocol configurations and test procedures are "wrung out" in the laboratory environment, with equivalent link data rates, delays, and error rates to the anticipated field environment. If, during the course of future tests, we see environments that are not as expected and we see results that deviate significantly from predictions, we should model the actual environments in the laboratory and generate new predictions before proceeding with those tests. (If at all possible, the testing schedule should be restructured to work around the unanticipated development, so that the laboratory testing can proceed in parallel with other, unaffected field tests.)

2. Ensure that the test environment includes proper instrumentation that does not interfere with the protocol testing.

   We did not have a way to measure bit-error rate in any way other than by post-processing log files after a test. We logged an extensive amount of data during the STRV testing, and used this data as the basis for determining packet error rates and bit-error rates. This was possible because the STRV data rates were very low, and the computers involved were relatively lightly loaded. In tests that require higher data rates, we will need to change our approach in order to avoid affecting the test results. We will need to put in place instrumentation that allows us to collect similar data and correlate it with the data collected elsewhere during the test.

3. Consider the availability of development and test tools when selecting subsequent test platforms and environments.

The MIL-STD 1750A processor on the STRV spacecraft is not well-supported by development tools for the C programming language. We devoted a considerable amount of time evaluating and debugging a free compiler for the 1750, and eventually abandoned the effort. We purchased a very expensive compiler of poor quality (but better than the free one), which was the only other choice apparent. We used an assembly-language simulator; some brief, borrowed time on a DRA in-circuit emulator; and the remaining debugging tools we wrote ourselves.

4. Build predictions for *all* measured responses, and check on a test-session by test session basis.

   We were surprised by the discrepancy in the bit-efficiency results. Since the throughput and link utilization results generally matched the laboratory values, we did not compute bit-efficiency on a per-test basis. It is doubtful that we could have devised a work-around for the cause of the discrepancy without detrimentally affecting the flight test, however, we should have identified the problem earlier.

5. Even though a value is within the predicted values, there may be still be problems.

   This is related to the previous lesson. The throughput and link utilization results were within expectations for the flight configuration that operated with SNACK enabled. However, the configuration problem that caused the unnecessary retransmissions could have been identified and corrected had we predicted and verified bit-efficiency.

6. The instrumentation we added to the protocol (specifically, the tcpdump-compatible packet logging) was very valuable.

   It allowed us to develop reasonably sophisticated diagnostic capabilities for “black box” testing. (The software development tools available for C-language development on the MIL STD 1750 are poor. The tools that we are describing are protocol analysis tools, and assume that the protocol is at least up and running.)

7. Automate the testing to simplify and to avoid errors

   One of the best things that we did during the STRV retesting was to develop *Expect* language scripts to automate the testing. It eliminated a wide range of human errors, reduced the stress of flight testing (somewhat), and allowed us to test in the laboratory 24 hours per day, 7 days per week.

8. We identified many implementation-specific lessons, related primarily to how our code handled very slow systems. We had no mechanism to enqueue multiple SNACKs, because before the STRV testing, we were always able to perform all of the SNACK-related retransmission before continuing to dequeue packets. That was not the case with STRV. We identified other problems resulting from an attempt to send a SNACK failed due to the slow link rates.

9. We learned how difficult it is to port code that has not been written for portability. The SCPS-TP prototype from which we started was developed as a platform to test functionality. It was not designed for portability. The STRV experiment presented two major porting challenges: port the code from a big-endian machine to a little-endian machine; and port the code from a 32-bit UNIX environment to a 16-bit, non-byte-addressible, resource starved environment with no operating system. Each of these was a significant undertaking in its own right, and to have both at the same time (with no development tools for the STRV) was truly challenging.

Appendix D

SCPS Transport Protocol Overview

The SCPS Transport protocol is based on TCP. In fact, SCPS-TP is essentially standard TCP augmented by a set of extensions and enhancements that consist of both implementation and specification changes. These modifications each respond to requirements derived from characteristics of the space environment, the mission communication scenarios, and other driving factors. Some of the constraints imposed by the space environment that led to TCP modifications include:

- Space link delays ranging from milliseconds to hours.
- Potentially noisy space links.
- Limited space link bandwidth.
- Limited periods of connectivity.
- A mismatch between the up-link and down-link channel capacities.
- Limited onboard processing power and memory (for programs and data buffering).
- Link interruptions caused by bursts of noise and antenna obscurations.

A number of TCP extensions have been defined to address these and other requirements. Some of these modifications were proposed by members of the research community and were adopted by the SCPS developers, while other enhancements were designed by the SCPS team, in some cases by drawing from the work of others. The following subsections describe some of the major SCPS extensions to TCP. The discussion of each feature contains a brief motivation or description of the constraint being addressed, a comparison to other similar enhancements that have been proposed if applicable, and a synopsis of the SCPS extension itself. The interested reader can refer to the SCPS-TP specification [15] for details on the SCPS extensions beyond those presented here.

D.1 Identifying the Source of Packet Loss

RFC 1106 [7] raises the important issue of differentiating between packet loss due to congestion and loss due to corruption. Because the appropriate response to congestion is quite different from the proper response to corruption, distinction between the two is essential when operating in an environment where both events are possible. However, the problem of identifying the source of packet loss can be generalized beyond simply differentiating between congestion and corruption. Different communications environments may be subject to loss caused by a variety of factors for which the appropriate responses differ, such as link outage or mobility, as well as network congestion and noise. A brief list of possible sources of packet loss includes:

- intermittent connectivity due to the ability to communicate with a satellite only during the portion of its orbit when it is in view.
- atmospheric interference that results in corruption.
- network congestion.
- losses due to hand-offs in a cellular communication network.
- antenna obscurations due to terrain features in line-of-site wireless communication.

The appropriate responses to these sources of packet loss differ. In the case of congestion, the correct response is to reduce the transmission rate. In the case of corruption, the appropriate response is to continue transmitting in the hope that some packets will reach the destination. In the case of a temporary link outage, the best response is to suspend transmission and wait for the link to be restored. As mentioned earlier, TCP fails to make the distinction between congestion and any other source of loss. Instead, TCP assumes that all packet loss is a result of network congestion, and reacts by invoking its congestion control algorithms. These algorithms immediately reduce the transmission rate drastically, and then gradually increase the rate again as long as no further loss occurs. This action allows the pipe to drain somewhat so that the congestion can subside. This approach is effective at controlling congestion, and at worst, results in oscillating behavior between congestion periods and nearly idle periods. However, this mechanism, which is triggered by any packet loss, reduces throughput and provides absolutely no benefit when the loss experienced is a result of noise, link outage, or changing connectivity.

The SCPS extensions provide a mechanism to change TCP's default assumption as to the source of segment loss from congestion to corruption. TCP relies on this default assumption in the absence of any other information about the state of the network. A network manager has the ability to set the default packet loss assumption appropriately for a particular network based on the most likely event. In addition, to decrease reliance on the default response, three signals are defined that provide explicit notification to TCP about the source of loss. These signals are the link outage signal, which indicates that the link is temporarily unavailable; the source quench, which signals the presence of congestion; and the corruption experienced signal, which notifies TCP of loss due to noise. Each of these signals evokes the appropriate response from TCP regardless of the default setting.

The congestion versus corruption problem also has ramifications for the proposed TCP extensions for use on networks in which the product of bandwidth and round-trip delay is high (> 64k bytes). Specifically, the utility of Selective Acknowledgment (SACK) and Window Scaling is limited when using congestion control since the goals of these mechanisms are in direct conflict, as noted by RFC 1106. The SACK and Window Scaling options are designed to keep the pipe full of data, while the intention of congestion control is to allow the pipe to drain. However, while SACK and Window Scaling offer little benefit when congestion control is in effect, their use is not detrimental to performance in such a case, with the exception of the bit overhead of SACK.

D.2 Congestion Control

In addition to the ability to enable or disable congestion control dynamically based on the source of packet loss, the SCPS extensions include modifications to the standard congestion control and avoidance algorithms. These changes consist of the implementation techniques proposed by TCP Vegas, and they do not involve any alteration to the TCP specification.

The SCPS TCP extensions also provide a rate control mechanism. This open-loop control system is similar to the token-bucket algorithm used for flow control of Available Bit-Rate (ABR) traffic in ATM [25]. It enforces an upper bound on the rate at which TCP can transmit to keep it from over-running the capacity of the link or the interface. A rate bound can be set by a network manager for each interface independently. This mechanism is especially important as a throttle when congestion control is disabled.

D.3 Retransmission and Acknowledgment Strategy

In the absence of congestion, the best throughput performance can be obtained only when link idle time is minimized. Selective acknowledgments provide more information to the data-sender than TCP's cumulative acknowledgments and can aid in keeping the channel occupied. This is especially true in a unidirectional data transfer, as opposed to an interactive dialog in which the next data to be transmitted depends on the reply to the last data sent. In an unidirectional transfer of data, the optimal throughput is realized when the data-sender never needs to halt transmission waiting for an acknowledgment. As discussed previously, operating with a large enough window to satisfy the bandwidth-delay product of the network is one requirement for keeping the sender active. However, large windows alone are not sufficient to keep the pipe full. A prudent retransmission and acknowledgment strategy is also required to efficiently utilize the bandwidth when packet loss due to sources other than congestion is experienced.

When packet losses are a result of corruption, and congestion control is not activated by these losses, a selective acknowledgment strategy will improve TCP throughput performance. As discussed above, the SCPS TCP extensions provide a means for distinguishing congestion from noise, and as a consequence TCP is able to invoke its congestion control algorithms appropriately. The retransmission and acknowledgment strategy described in this section is most efficient when losses are not caused by congestion, and congestion control is not invoked; however, it operates independently of the congestion control mechanisms. This strategy is optimized to keep the pipe full and utilize the bandwidth as efficiently as possible within the constraints imposed by the effective window. As a result, the best throughput performance is obtained when the sender is not congestion-window limited in transmitting. However, the congestion control logic may become active or inactive during the course of the connection without affecting the correct operation of the retransmission and acknowledgment strategy. In fact, if the network becomes congested, better performance will be obtained if the congestion control algorithms are invoked and the pipe is permitted to drain. The performance gains alluded to below will be realized in the case where losses are due to corruption, and this fact is recognized by TCP. When the losses are a result of congestion, the standard TCP congestion control and avoidance algorithms will govern the sender's throughput behavior, regardless of the retransmission and acknowledgment scheme employed.

TCP's cumulative acknowledgment mechanism provides limited information to the data-sender regarding which segments have successfully reached the destination. In the absence of complete information, the data-sending TCP must make assumptions about the state of the receiver's resequencing queue when retransmitting segments. The cumulative acknowledgment authoritatively tells the data-sender the highest sequence number that has successfully been received in-sequence. However, it provides no information about any segments that may have correctly been received out-of-sequence beyond the segment being ACKed. There are two opposing philosophies for handling this lack of information that lead to two distinct approaches for retransmission in reaction to a time-out or reception of multiple duplicate acknowledgments. These approaches are implementation details and are not governed by the TCP specification, and in practice, approaches that lie on the spectrum somewhere between these two extremes may be adopted. Note that Berkeley Standard Distribution (BSD) Unix Networking Software Release 2.0 follows the conservative approach described below [20].

1) The conservative approach dictates that when a retransmission is deemed appropriate, the sender retransmits *only* the oldest unacknowledged segment and no others. Once it retransmits this single segment, the sender may continue transmitting segments at the point where it left off before the retransmission, within the constraints of the effective window. The implicit assumption in this case is that only a single segment has been lost and the receiver has correctly received and queued all segments that were transmitted after the missing segment.

2) In contrast, the aggressive approach maintains that when a retransmission is necessary, the sender retransmits the oldest unacknowledged segment as well as every segment that it had sent after that segment. This "Go-Back-n" scheme operates under the assumption that the receiver has lost a succession of segments.

Both of these approaches have merit; however, each is optimized for a different operating environment. The conservative approach performs well when individual packets are sporadically lost. Here, unnecessary retransmissions of correctly received segments are avoided. In contrast, the aggressive scheme works well when sequences of consecutive packets are dropped. In the aggressive retransmission case, all of the lost segments are retransmitted without waiting for further information; although, it is imperative that such retransmissions be metered out slowly to avoid causing congestion. On the other hand, each retransmission approach performs poorly when faced with a loss profile that violates its assumptions. In the case of the conservative approach, when a string of consecutive segments is lost, only a single segment is retransmitted per round-trip time. Each time a segment is retransmitted, it is acknowledged; however the data-sender must wait for further notification (in the form of duplicate ACKs or a time-out) to discover the next lost segment. This strategy recovers very slowly from a sequence of losses. Conversely, the aggressive approach performs poorly when only a single segment is lost. In this case, bandwidth is wasted as a window's worth of segments are retransmitted when only a single retransmission was actually necessary.

By telling the sender exactly which segments have been successfully received (or which segments are missing), a selective acknowledgment strategy can provide better performance than the simple cumulative acknowledgment scheme of TCP, regardless of which retransmission approach is employed. (Again, we emphasize that the performance gain will be realized only when losses are not due to congestion, and TCP does not invoke a congestion control response when loss occurs.) This performance improvement, in terms of a reduced number of retransmissions, fewer retransmission time-outs, and higher throughput, can be more substantial in a noisy environment with a high bandwidth-delay product. The SCPS developers analyzed the RFC 1072 [9] SACK option and the RFC 1106 NAK option and borrowed ideas from them in designing the SCPS Selective Negative Acknowledgment (SNACK) option. Refer to RFC 1072 and RFC for a detailed discussion of these SACK and NAK, respectively; an analysis of these options with respect to the SCPS requirements follows.

The ability to acknowledge multiple discontiguous blocks is desirable, and is provided by the Selective Acknowledgment mechanism proposed in RFC 1072; however, three problems exist with the scheme:

1) The SACK option has limited coverage. Because an upper bound exists on the size of the TCP header (60 bytes, of which 20 bytes are consumed by the standard TCP header), and because a SACK option that specifies n blocks has a length of $4n + 2$ bytes, a single SACK option is capable of specifying at most 9 blocks. Moreover, if other TCP options are in use, which compete for space in the TCP header, a given SACK option may not be

able to specify even 9 blocks. This limitation reduces the advantage gained by using Selective Acknowledgments rather than cumulative acknowledgments, especially when operating with large windows in a noisy environment.

2) The SACK option is imprecise. When RFC 1323 [10] Window Scaling is enabled, the window size can effectively be as large as a 30-bit quantity. The SACK option uses a 16-bit field to identify the location, or Relative Origin, of a block of data within this window space. Likewise, the Block Size field, which specifies the size in bytes of a contiguous block of data is a 16-bit quantity. Clearly, these fields cannot address the entire window space when scaling is enabled. The two solutions to this problem proposed in RFC 1072 are both inadequate. The first is to expand the SACK Relative Origin and Block Size fields to 24 or 32 bits each. From the perspective of bit-efficiency, this solution is unacceptable. In addition, this approach drastically reduces the number of blocks that can be specified by a SACK option. The second proposal is to scale the SACK fields by the same value as the window. This solution introduces imprecision into the acknowledgment, since the SACK option must report block offsets and lengths in multiples of the window scale factor, which may not be a multiple of the segment size. To reconcile this imprecision, it is necessary to adopt a conservative approach and unnecessarily retransmit some data when there is doubt as to which segments are being acknowledged. This approach also makes inefficient use of the channel.

3) The SACK option is incompletely specified. RFC 1072 describes the format of the SACK option, but it does not cover other essential issues, such as when to send a SACK as the receiver of data, or how to process one as the data sender. RFC 1072 also fails to mention the interaction between the SACK option and the congestion control algorithms, which governs the system throughput in the event of packet loss in standard TCP.

While the RFC 1106 NAK option is reasonably bit-efficient, it has the ability to signal only a single “hole” that exists in the sequence space in the receiver's buffer. A more powerful mechanism, capable of specifying multiple holes is desirable in the noisy, long-delay environment to provide the sender with more complete information about the state of the receiver's potentially large out-of-sequence queue. (Note that large send and receive buffers are required when operating with large windows.)

The SCPS Selective Negative Acknowledgment (SNACK) is a variable-length TCP option that has the ability to signal the presence of multiple holes in the receiver's resequencing queue in a bit-efficient manner. The SNACK option consists of 5 fields: the kind and length fields required of all TCP options, followed by the “Offset” and "Hole1" fields (each 16 bits long) and an optional variable-length bit-vector. Offset specifies the displacement from the ACK number carried in the regular TCP header to the starting location of the first hole that is being signaled by this particular SNACK option. The Hole1 field specifies the size of this hole. Note that this hole is *not* necessarily the first hole in the receiver's overall out-of-sequence queue. Both the Offset and Hole1 values are expressed in Maximum Segment Size (MSS) units. The bit-vector then signals zero or more additional holes, also expressed in terms of MSS-sized blocks. The bit-vector maps the sequence space of the receiver's buffer beginning one byte beyond the end of the block specified by Hole1. Each “0” in the bit-vector signifies that one or more bytes are missing in the corresponding MSS-size block of the receiver's resequencing queue. (Note that any zeroes to the right of the rightmost “1” in the bit-vector are *not* interpreted as a hole.) The length of the bit-vector, which may vary at the SNACK-sender's discretion, is determined from the option length. Use of the SNACK option is enabled by another option, SNACK_OK, which may be sent only on the SYN segment of the connection.

While an out-of-sequence queue exists, the data-receiver scans its receive buffer, forming SNACK options and sending them on outgoing ACK segments. By setting the Offset field to zero, the data-receiver can NAK a block beginning at the ACK number in the TCP header. Alternatively, by specifying a non-zero Offset value, the SNACK option can begin by addressing any arbitrary portion of the sequence space. The latter capability is especially useful when the out-of-sequence queue is large, even in terms of MSS units, and a single SNACK option is unable to reference the entire sequence space because of the limit on the TCP header size. In such a case, multiple SNACK options can be sent in which each continues specifying holes in the receive buffer where the last left off.

At the data-receiver, there are considerations regarding whether transmission of the SNACK option should be delayed, and by how much, in anticipation of the missing segment(s) arriving out of order. This decision is implementation specific and should take into account the probability of segment misordering by the underlying network(s). Unless segment misordering is highly unlikely, delaying transmission of the SNACK option may offer some benefit. This assertion holds because the SNACK option forces immediate retransmission. For this reason, the SNACK-sender should be as certain as possible that retransmission in required. In the SCPS environment, acknowledgments are typically delayed substantially anyway because the uplink bandwidth (the acknowledgment channel) is severely limited in comparison to the downlink. Thus, ACKs are aggregated as much as possible, and in many cases, an ACK is sent just once per round-trip time.

As mentioned above, upon receipt of a SNACK option, the data-sender retransmits all segments necessary to fill the signaled holes. This behavior is similar to that dictated by the RFC 1106 NAK option, in contrast to the RFC 1072 SACK recommendation. Receipt of an RFC 1072 SACK requires the data-sender to simply mark the selectively ACKed segments as acknowledged, but does not cause retransmission of the segments that were previously sent but not acknowledged by the SACK. The more aggressive retransmission scheme adopted here is appropriate when the goal is to prevent retransmission time-outs, which cost more in terms of link idle time than unnecessary retransmissions cost in terms of wasted bandwidth. The data-sender may process the SNACK option by first retransmitting the segments that fill the gap corresponding to Hole1. If a bit-vector is present, the data-sender may easily process it by left-shifting the bit-vector until the last "1" is shifted out while retransmitting the segment corresponding to each "0" encountered in the process.

Finally, note that large buffers are required when trying to keep the pipe full using a selective acknowledgment scheme. To prevent the pipe from emptying while responding to SACKs, the retransmission buffer must be large enough to store the amount of data that can be transmitted in more than 2 RTTs. This interval is determined by the time between the initial transmission of a segment that is subsequently lost, and its eventual acknowledgment over 2 round-trip times later, with an intervening (possibly delayed) SNACK and a retransmission 1 RTT after the initial transmission.

D.4 Window Scaling

The SCPS extensions include the TCP Window Scaling option exactly as specified in RFC 1323.

D.5 Round-Trip Time Measurement

The SCPS extensions include both the RFC 1323 Timestamps option and the TCP Vegas mechanism for accurately measuring the round-trip time. One of these two mechanisms, but not both, is chosen for use on a particular connection.

D.6 Protection Against Wrapped Sequence Numbers

The SCPS extensions include the Protection Against Wrapped Sequence Numbers (PAWS) mechanism as specified in RFC 1323.

D.7 Best-Effort Transport Service (BETS)

TCP offers only a single, fully-reliable service while UDP offers only an unreliable service. Some applications that are used in the SCPS environment require an intermediate level of reliability. The Best Effort Transport Service (BETS) option provides such a data transfer service that guarantees uncorrupted and in-sequence data delivery, but possibly with gaps. In the terminology introduced in RFC 1693 [4], BETS provides a partially reliable, ordered service.

When BETS is enabled, the data-sending application has the ability to set thresholds of buffer utilization and retransmission count. Once one of these thresholds is exceeded, unacknowledged data is ignored and transmission of new data continues. The effect of BETS on the data-transmitter is that the application is never “blocked” indefinitely waiting for acknowledgments.

At the receiver, the application can specify maximum blocking thresholds in terms of buffer capacity and time. When one of these thresholds is exceeded due to an unfilled hole in the received data, the application is informed of the size of the gap (in octets) and is then delivered the data that is available. The effect of BETS is that the data-receiver is not blocked indefinitely waiting to fill “gaps” in the received data.

BETS supports three major types of operation:

1) High-speed data transfers of repeated data (such as images), for which the loss of a portion of the data (i.e., a portion of a scan line) is important only in that it might cause synchronization to be lost.

2) Reliable transfer of data in a buffer-limited environment, for which availability of new data is more important than retransmissions of old data, if buffers are exhausted.

3) Highly-reliable operation when no acknowledgment channel is available.

Missions with some or all of these types of communication scenarios should benefit from the Best Effort Transport Service.

D.8 SCPS TP Header Compression

SCPS TP Header Compression is to be used on connections that require high bit-efficiency. The requirement for high bit-efficiency may result from the presence of low-data rate links in one or both directions of the communication path, or from high utilization of those links. For most space missions, link bandwidth is considered a scarce resource, and the overhead of TCP segment headers is considered too high.

A significant amount of work has been done outside the SCPS activity to reduce the overhead of TCP/IP headers. This work is documented in RFC 1144. The compression specified in RFC 1144 is intended for use on low-speed serial links, and addresses the problems of maintaining interactive response for programs such as telnet. RFC 1144 header compression operates at the link layer. The link layer maintains connection state tables for inbound and outbound TCP/IP connections. The first header for each connection seen by the link layer is not compressed. Subsequent headers are encoded by sending only the fields that changed from the previous header. Additionally, the sequence number, acknowledgment number, urgent pointer, and window values are encoded as *changes* to the previous value. At the receiving side, an uncompressed TCP/IP header is created by applying the changes to the saved header to create a new TCP/IP header. This new header is saved in the connection state table and is forwarded to the destination along with the data. Since the information in the compression state tables will be corrupted if a segment is lost or damaged (misordering is typically not a problem at the link layer), the (unmodified) TCP checksum is included in all TCP segments. If the decompression state is corrupted, the TCP checksum will fail at the receiving TCP endpoint and the corrupted segment will be discarded. Retransmissions are forwarded uncompressed by the RFC 1144 compressor, and are used to resynchronize the decompressor's state. (Note that if a segment is lost or corrupted, *all segments following it will be decompressed improperly, causing them to be discarded by the receiving TCP endpoint.* This behavior continues until the sending TCP entity retransmits the lost segment(s), resynchronizing the compressor and decompressor. This is typically not a problem for RFC 1144 operation, since it is designed primarily for interactive operation in which there are typically only a few octets of data outstanding at one time.)

While RFC 1144's method of header compression works well in low bandwidth-delay networks that have stable connectivity and low bit-error rates, it is poorly suited for the SCPS environment. The high bandwidth-delay products mean that any failure of the decompression will result in significant data loss. Further, the possibilities of high bit-error rates or changing topologies or both increase the probability that decompression failure would happen more often in the SCPS environment than in the terrestrial environment.

SCPS TP header compression draws on ideas developed in RFC 1144 and previous header compression schemes. Unlike RFC 1144, SCPS TP's compression operates at the transport layer, in an end-to-end manner. Therefore, it is unaffected by changing network connectivity. RFC 1144's decision to send deltas from the previous value when the value changes makes it particularly susceptible to decompression failure if loss or corruption is experienced. Rather than sending deltas, SCPS TP sends the full field value. This results in slightly less per-packet efficiency, but allows resequencing to be successfully accomplished, eliminating the go-back-n retransmission behavior that RFC 1144 imposes. Further, SCPS TP sends sequence number fields whenever data is present (whether the field changed from the previous value or not), and always sends the window field when an acknowledgment field is sent. These modifications improve the robustness of SCPS TP compression, at the expense of some efficiency. SCPS TP allows certain options to be compressed - specifically, the Selective Negative Acknowledgment option (SNACK), and the TCP Timestamps option. These may occur frequently, and are parsed and compressed by the header compression software. Other TCP options are included uncompressed. Finally, since the SCPS TP compressed packets are essentially “stand-alone”, the checksum covers the compressed packet header and the user data, not the uncompressed header and the user data.

D.9 Other SCPS Extensions

The SCPS extensions include additional features that are not of significant relevance to this experiment, such as a record boundary option and a priority mechanism. For further discussion of these extensions, see [15].

Appendix E

Implementation Background

The TCP/IP protocols are typically implemented in the kernel of the Unix operating system and are accessed by applications through system calls. The kernel is a logical place for networking protocols to reside from both the conceptual and practical points of view. The protocols simultaneously provide services to multiple processes that reside outside the kernel, and they can operate more efficiently as kernel routines without the overhead and scheduling constraints associated with being a Unix process. However, there are disadvantages to building a prototype that resides within the Unix kernel. Development, debugging, and testing of kernel implementations is significantly more difficult and time-consuming than that of application programs. Also, kernel implementations are closely bound to the specific version of the operating system for which they are designed, as well as to the platform on which that version of the operating system runs. A final, practical issue is that to produce a kernel implementation of a protocol, a developer requires access to the kernel source code, which is prohibitively expensive for some of the popular commercial operating systems (e.g., SunOS).

In designing the TCP prototype for the SCPS project, the advantages of working outside the Unix kernel prevailed. A primary concern for the SCPS project was portability of the prototype code, which dictated a user-space implementation. In fact, the SCPS protocols are expected to be hosted on satellites that do not have a multi-purpose operating system into which the protocols could be integrated, so developing the prototype as a stand-alone program is beneficial. Another major concern of the SCPS project was the cost of the prototype in terms of development time. Here again, a user process implementation was the obvious choice. Consequently, the approach chosen by the SCPS development team was to implement an enhanced TCP prototype as a Unix application program.

One possible development choice was to extract the TCP/IP implementation from the Berkeley Unix source code and modify it to run as a user process. However, this option was not selected because the networking code is so intimately tied to the rest of the Unix operating system (e.g., the file and memory management systems) and is so difficult to dislodge from the kernel. Instead, a minimal implementation of TCP, called TinyTCP [5], that already was structured to run as a stand-alone program was chosen as the starting point for the prototype implementation. TinyTCP lacks many of the basic features required of an RFC 1122 [8] compliant TCP implementation; however, the philosophy was adopted that it is easier to add features to a minimal implementation than remove them from a full-featured implementation. Among the features missing from the original version of TinyTCP are the following:

- Window-based flow control
- Retransmission buffers

- Resequencing buffers
- An Application Programming Interface (API)
- Congestion control and avoidance mechanisms
- Round-trip time estimation
- Delayed acknowledgments
- Nagle's algorithm
- Karn's algorithm
- Sender and receiver Silly Window Syndrome (SWS) avoidance algorithms

The SCPS team added the above features, as well as the SCPS extensions, to the TinyTCP implementation in transforming it into the SCPS-TP prototype. We also made other fundamental changes to TinyTCP. First, we ported TinyTCP from its native environment to run as user-level Unix process over raw IP sockets on any Berkeley-derived Unix host. Next, we restructured the control flow of the program. When TinyTCP was originally ported to the Unix environment, the transport protocol and the application using its services, either the data-sender or data-receiver, were compiled and linked together into a single program and run as a Unix process. The transport protocol served as the scheduler for its application "process," by calling an application-supplied function when necessary. For example, when TinyTCP was ready to pass received data to the application, it invoked the application's data-consumer function. In building the SCPS-TP prototype, we replaced this control structure in favor of the more generic and flexible approach of lightweight threads and a thread scheduler.

Threads are low-overhead "processes" that execute in user space. Threads each maintain their own private variables and state information, but as parts of a single Unix process, they also share globally-accessible process state, including memory. The thread scheduler acts just like the process scheduler in a multi-tasking operating system that provides context-switching capability. It removes the currently running thread from the execution state, places it in the run queue, and gives control of the CPU to the next runnable thread in the queue. The SCPS thread-scheduler is non-preemptive; it does not decide when a context switch should occur and interrupt the currently executing thread asynchronously. Instead, each thread has the responsibility to actively relinquish control of the processor regularly by explicitly invoking the thread scheduler through a function call. Under this architecture, the transport protocol and the application are each implemented as separate threads, which are executed in a round-robin fashion by the thread scheduler. The thread scheduler, the application, and the transport protocol are all compiled and linked together into a single executable program.

The SCPS team has developed two prototype applications for use with the SCPS-TP protocol prototype, a data-sender and a data-consumer. Note that typical Unix TCP applications would require modification to run over the SCPS-TP prototype. While a socket-like API has been added to the SCPS-TP prototype, the semantics (as well as the syntax) of the service calls differ from those of Berkeley sockets. More significantly, every application that uses the SCPS-TP prototype must be structured as a thread that is linked in with the prototype. As such, an application must contain calls to the thread scheduler at periodic intervals. Consequently, popular TCP applications, like FTP, are not readily usable with the SCPS-TP prototype. Instead, a simple data-transmitting client (scps_init) and a data-receiving server (scps_resp) have been developed for use in debugging, testing, and evaluating the prototype.

The remainder of this Appendix addresses the major modifications that we made to the SCPS-TP prototype in order to support its operation onboard the STRV. We have not yet documented some of the more intricate code restructuring that we performed to address specific issues in that operational environment.

In porting the SCPS-TP prototype from the Unix user environment to the STRV environment, we made as few changes as possible. We had to restructure the code such that retransmission buffers and *mbuffs* existed as finite arrays, rather than as dynamically allocated structures. We also had to write the assembly-language portions of the thread scheduler for the MIL-STD-1750A processor. We revised the initiator and responder programs for onboard operation, such that they acquired their configuration parameters from well-known memory locations outside of the program. In this way, we could reconfigure the protocol, or even the protocol stack, by changing a few memory locations. Since the time to upload and check out code onboard the STRV was prohibitively long, we made no attempt to optimize the size of the implementation. Rather, we uploaded software for all of the tests that we intended to operate. Further, we did not scale back any of the generality of the protocol stack in an attempt to create a small implementation. This is an appropriate activity when one is preparing mission-oriented code, but was not a high priority for this test.

Another significant modification to the SCPS-TP prototype that resulted from the STRV port was the development of a general purpose CCSDS Telemetry and Telecommand network layer. This layer provides the same interface as the SCPS-NP, and deals in Internet Protocol (IP) addresses as its external interface. The protocol uses a look-up table for translation between APIDs and IP addresses, using the lower eight bits of the APID for a network destination address, and the upper three bits for a protocol identifier.

The SCPS-TP prototype makes use of relatively few system services: the system clock, a packet send service, a packet receive service, etc. We had to replace the invocations of the corresponding UNIX-based services with calls to the ROM-based utilities resident on the STRV. Typically, these calls involved calling assembly language routines from C, and in some cases developing the assembly language routines.

In order to provide a means to control the testing, to read and write onboard memory, to upload software, and to transfer control to a protocol under test (and receive control back from it when it completed), we needed an onboard executive. We developed such an executive, the SCPS Kernel, to provide the minimal functionality that was necessary in order to perform the testing. This Kernel, along with modifications to the SCPS-TP software to handle these transfers of control, gave us the means to control the onboard software with minimal overhead. In order to take advantage of ground system software made available by DRA, we used the a subset of command set used by the STRV Ada software (which was overwritten in RAM by the SCPS code). This had the added advantage of allowing the operators at Lasham to issue commands to the SCPS Kernel, specifically for the purpose of uploading software, but we also took advantage of this capability at other times during the testing.

Appendix F

STRV Spacecraft and Control Center Descriptions

F.1 The STRV Spacecraft

The STRV 1a and 1b missions were conceived with the principal aim of providing the space technology community with affordable access to earth orbit to allow in-orbit evaluation of new technologies. The spacecraft were designed, built and tested at DRA Farnborough and are operated from the DRA Lasham ground station in Southern England.[1] The short duration time scale of the project (from the design phase to operations in three years) has guaranteed the return of experimental data in a meaningful time frame.

Despite a maximum mass of only 55 kg each, a total of fourteen different experiments are incorporated in the design of the two vehicles. The majority of the technologies flown are associated with ongoing internal research programs within the Space Department of the DRA and in conjunction with UK industries and universities. In addition, there is a major international collaborative aspect to the project. The Ballistic Missile Defense Organization (BMDO) Materials and Structures Program has sponsored four experiments that were built at the Jet Propulsion Laboratory (JPL) and are flown aboard STRV 1b. The BMDO also negotiated access to the NASA Deep Space Network (DSN) antennas to supplement the DRA ground station. Connection to the DSN is achieved through the NASCOM system, the NASA institutional space communications network. The European Space Agency (ESA) also submitted experiments and solar panels and provided the program with design effort and radiation facility time. The Phillips Laboratory, Albuquerque, NM, provided solar panels and experimental solar cells as part of an experiment. The Geostationary Transfer Orbit (GTO) was used for the mission for radiation environment research and to achieve accelerated component testing. Figure F-1. The STRV-1 Orbit, illustrates the orbit and indicates phenomena that are of particular interest to the mission.

---

1 As of this writing, control of the two STRV spacecraft has been transferred to the University of Colorado at Boulder in a demonstration of international interoperability between diverse spacecraft and control centers.

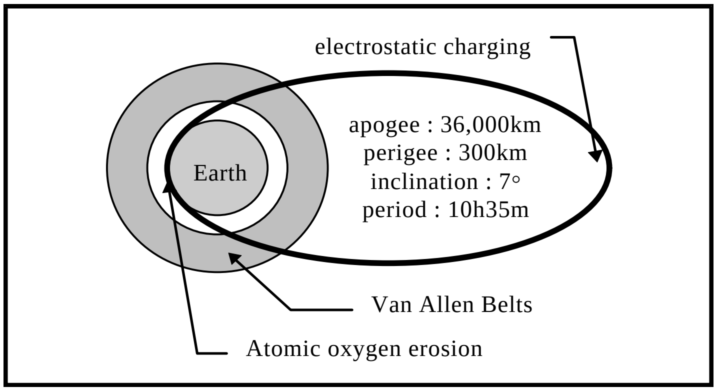


Figure F-1. The STRV-1 Orbit

The STRV 1a/b mission objectives were :

- To provide the technology community with cost effective access to a harsh earth orbit environment.
- To implement the mission with a short development timescale and return data quickly to the experimenters.
- To enhance DRA capabilities in all aspects of spacecraft design, evaluation and operations.

Both spacecraft were launched on 17 June 1994 by Ariane 4 from Kourou, French Guiana. The one year primary mission was highly successful and a large amount of scientific data was collected from all the on-board experiments. Due to the desire to continue operations for a number of experiments, the mission was extended and both spacecraft remain operational as of this writing. This mission extension provided the opportunity for utilization of one of the STRV spacecraft within the SCPS protoflight test and demonstration program.

The STRV spacecraft is shown in Figure F-2. Both spacecraft are nearly identical, differing mainly in their complement of experiment packages.

Note the spacecraft communications antenna protruding from the top of the spacecraft, and the structures on the top of the spacecraft deployed around the edges. The antenna and these structures play a key role in the communications difficulties that are discussed in Appendix G.

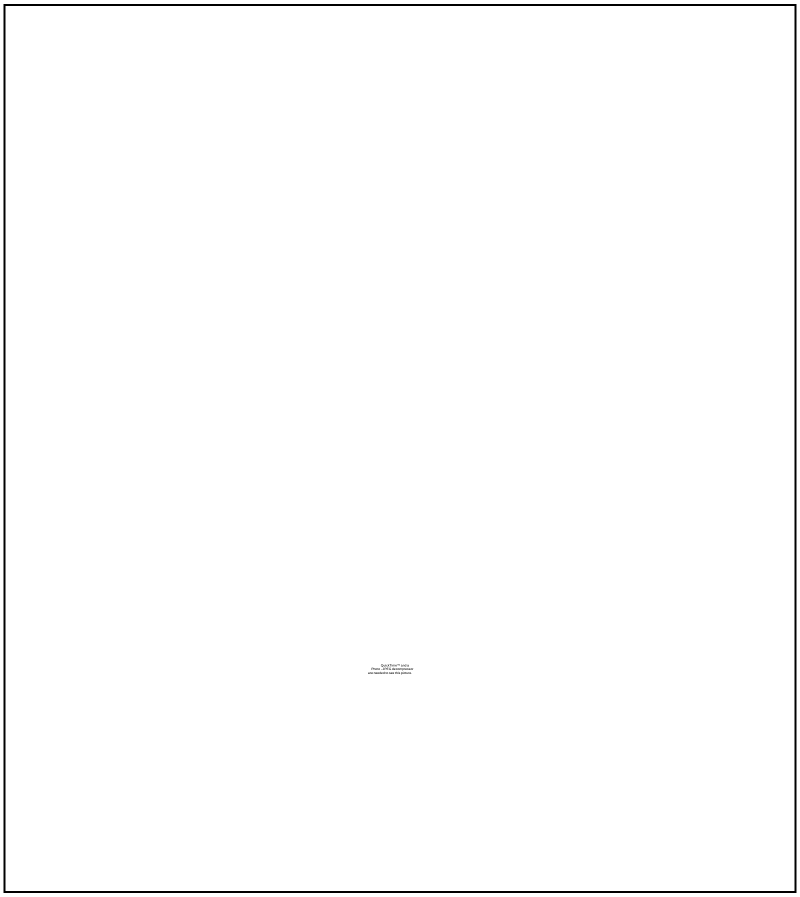


Figure F-2. The STRV Spacecraft

A summary of the spacecraft design is shown in Table F-1. [27]

Figure F-3. STRV Spacecraft Architecture, illustrates the salient features of the spacecraft architecture, with an emphasis on those elements important to the SSFE. There are six major subsystems comprising the spacecraft.

The power system provides regulated power to the other subsystems. It uses nickel cadmium batteries charged by the solar cells on the sides of the spacecraft.

The attitude control system control system uses ground-commanded magnetourquing at perigee to maintain the necessary aspect angle between the solar panels and the sun.

Although the experiments are shown as a single subsystem, they are in fact independent of each other. They bear no relationship to the SSFE.

Table F-1. STRV Spacecraft Design Summary

Design Feature	Description
Mass	50-53 Kg
Dimensions	450 x 450 x 450 mm
Structure	Carbon/PEEK thermoplastic skinned aluminum honeycombed panels
Power	GaAs body-mounted solar arrays, 31W to 33W (BoL)
Power storage	46 WHr (16 cell NiCd)
Attitude Control	Spin 5 rpm, magnetorquer control
Primary Computer	GEC Plessy MIL-STD-1750A Silicon-on Sapphire (SOS)
Primary RAM	128 kBytes SOS RAM
Primary ROM	64 kBytes SOS ROM and 4 kBytes SOS boot PROM
Communications	ESA TM/TC CCSDS standard on S-band, packet TM at 1 kbps, TC at 125 bps

The STRV Data Interchange Bus (SDIB) is an 8-bit parallel bus used to provide communications between the various subsystems of the spacecraft.

The radio frequency (RF) subsystem provides the means for communicating with the ground. It contains dual receivers (Rx) and transmitters (Tx). A single antenna (the second antenna was eliminated during development) serves both the receivers and the transmitters via a diplexer. Dual linear amplifiers (LNA) boost the received signal for the receivers.

The On-Board Data Handling (OBDH) system is the heart of the spacecraft. It contains two On-Board Computers (OBC-1 and OBC-2), a Command Distribution Unit (CDU), a Data Acquisition Unit (DAU), an Up Link Unit (ULU), and a Down Link Unit (DLU). The ULU receives CCSDS Command Link Transmission Units (CLTUs)[22] from the active receiver and reassembles the telecommands which are then passed on to the appropriate OBC for

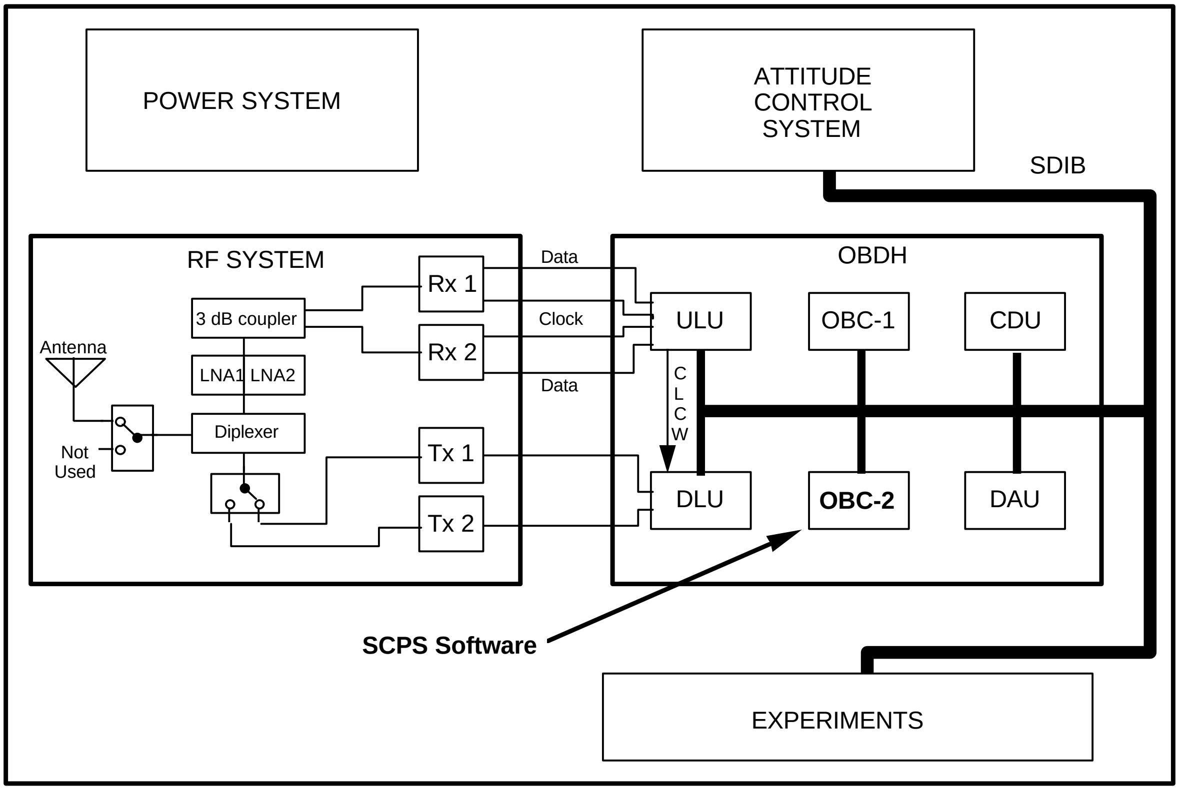


Figure F-3. STRV Spacecraft Architecture

processing[2]. The ULU also passes the CCSDS Command Link Code Word (CLCW), which is part of the CCSDS COP-1 retransmission protocol [23], directly to the DLU for insertion into the telemetry stream.

The OBCs, built by DRA around MIL-STD-1750 radiation hardened microprocessors, control all aspects of the spacecraft subsystems. Although they are completely redundant and interchangeable, in normal operations OBC-1 is used for spacecraft control and OBC-2 is used for the experiments. The computers are also used for storing data when the spacecraft is not in contact with the ground.

---

2 In the event of the failure of both computers or the SDIB, it is possible to route ULU output directly to the CDU. This connection is not shown in the figure.

The DAU is used for collecting status and state of health of the spacecraft platform and experiments. DAU data is sent either directly to the DLU for immediate transmission, or to an OBC if there is no communication with the ground.

The CDU is used to control the power to the various spacecraft components. It receives commands from the OBCs (or directly from the ground in an emergency) and implements the command by turning the power on or off to the applicable spacecraft component or experiment.

OBC-2 was allocated in its entirety to the SSFE. The SCPS software loads were prepared on the ground and uplinked to the spacecraft for writing into the OBC-2 RAM. Procedures were established for ensuring that OBC-1 remained in control at all times, and that the Lasham control center could control the spacecraft at all times. In the event of problems (and there were several), a reset command could be sent to the computer and it could reload its original basic software from PROM. Although this erased the SCPS software, it provided the means of maintaining positive control for spacecraft safety, and the only thing lost was the time needed to reload the SCPS software.

The Lasham Control Center

DRA Lasham is a small but major satellite tracking station located in the Hampshire countryside midway between London and Southampton, and near the Royal Oak Conference Centre. It is within a few miles of the US Air Force Satellite Control Network (AFSCN) Remote Tracking Station (RTS) Lion, located at Oakhanger. DRA Lasham is the principal point of entry for meteorological satellite data for western Europe. The STRV control center is located within the main building at DRA Lasham.

The control center architecture is illustrated in Figure F-4. STRV Control Center Architecture for SSFE. For a contact directly between the spacecraft and the Lasham control center, the 12 meter antenna located at Lasham is pointed towards the spacecraft at the time the

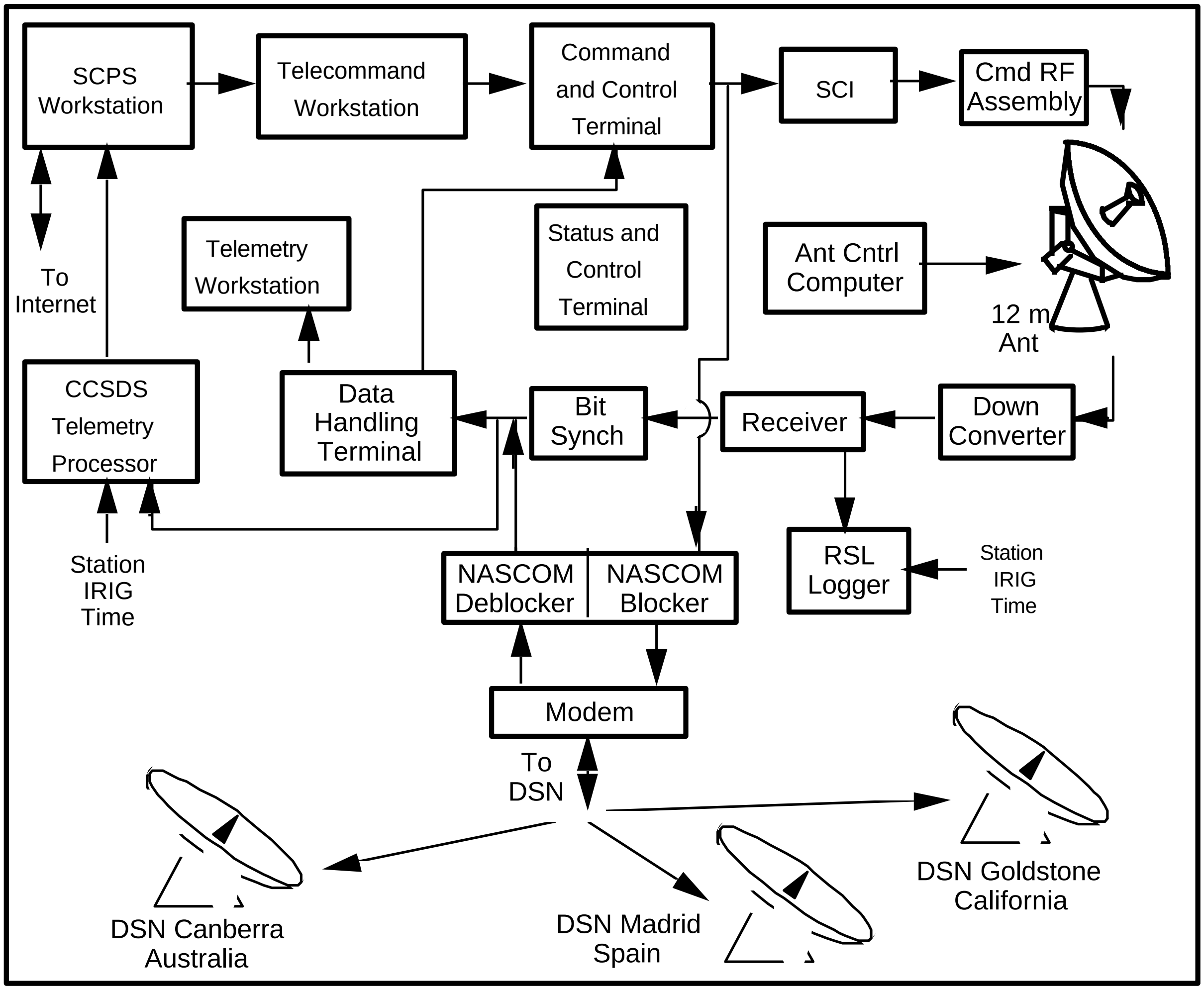


Figure F-4. STRV Control Center Architecture for SSFE

spacecraft was told during the previous contact to turn on its transmitters[3]. After down-conversion and approximately 45 dB of amplification, the signal is passed to the receiver. The receiver AGC is sent to a PC containing an analog-to-digital converter for logging the received signal level (RSL). The RSL logger workstation also contains an IRIG time code

---

3 The transmitters can also be commanded to turn on from the ground, and if the spacecraft has not heard from the ground after a certain period of time, it automatically starts transmitting.

decoder card[4] for providing accurate time for time tagging the RSL samples. The telemetry bit stream out of the receiver is passed to a bit synchronizer and then sent to the Data Handling Terminal (DHT). The DHT contains the frame synchronizer that formats the telemetry into the CCSDS telemetry frames, and the commercial software that archives and displays the telemetry. The same software runs on the Telemetry Workstation to provide additional display. The DHT also passes the CLCW to the Command and Control Terminal (CCT) as part of the COP-1 loop described earlier.

Commands for the spacecraft originate in the Telecommand Workstation (TCWS), either by direct entry or as a batch file received from an off-line workstation. The commands are passed to the Command and Control Terminal (CCT) which puts them into the CCSDS telecommand format known as Command Link Transmission Units (CLTUs). The CLTUs are sent to the Status and Control Interface (SCI) unit in the radome where they are then passed to the Command RF Assembly and transmitted to the spacecraft. If no commands are being sent, the SCI generates an idle sequence for the uplink.

The telemetry out of the bit synchronizer is also routed to the CCSDS Telemetry Processor, a modification of a unit in development by the DRA. This processor also contains a frame synchronizer, and software for decomposing CCSDS telemetry frames into their component data packets and presenting them to the SCPS workstation. This independent telemetry processor was necessary because the DHT does not process CCSDS telemetry frames that fail the cyclic redundancy check (CRC) performed by the frame synchronizer in the DHT. The SCPS team requested that frames that failed the CRC check still be passed on for upper layer processing because a) the SCPS-TP has its own methods of error detection, and b) the loss of a frame typically means the loss of three SCPS-TP packets, when conceivably none have been corrupted. With the scheme implemented by DRA, the SCPS workstation receives advisory information with each packet that indicates whether the telemetry frame in which the packet was contained passed or failed its CRC check.[6].

The SCPS workstation contained the ground complement of the SCPS protocol software. It received telemetry packets from the CCSDS Telemetry Processor for use by the protocols, and sent command packets to the TCWS for insertion into the uplink telecommand stream. The SCPS workstation was controlled primarily by telnet from the SCPS laboratory located at the MITRE facility in Reston, Virginia.

If the communications between the control center and the spacecraft was through the NASA Deep Space Network (DSN), the path was somewhat different. CCSDS CLTUs from

---

4 Inter-Range Instrumentation Group (IRIG) time codes are standard time codes used in many instrumentation and telemetry systems. The code used at Lasham is the 100 PPS modulated code. A station master clock synchronized to Global Positioning System (GPS) time distributes the code throughout the station for use where needed.

the CCT were routed to the NASCOM[5] blocker for formatting into the standard NASA data communications structure. The input data stream is buffered until there is a sufficient amount to release it in 4800-bit blocks. These blocks are sent over a modem link to one of the three DSN stations via the NASCOM communications network. The STRV used the DSN as the primary communications path for reasons described in Appendix G. Since the DSN stations are located in Spain, Australia, and California, the major constraint on communications was primarily antenna availability, not spacecraft visibility.

Telemetry data was received via the modem link in the reverse direction. The synchronized bit stream out of the NASCOM deblocker was routed directly to the DHT where it was processed as described earlier.

The dual communications paths available to the STRV control center allowed a large degree of flexibility in the SSFE. Because Lasham and Madrid are within a few degrees of latitude and longitude of each other, the antennas at these locations have nearly identical spacecraft visibility periods. This allows telemetry to be received by one of the antennas and telecommands to be sent by the other, thus permitting tests of the effects of different delays. It also provided a means of troubleshooting various configuration problems.

---

5 NASA Communications network.

Appendix G

The SSFE Communications Environment

The SSFE communications environment is best viewed from the standpoint of the OSI layered model. There are two models for STRV, depending on whether the link was via the Lasham 12 meter antenna or via one of the 26 meter DSN antennas. Figure G-1 shows the model for communications via the Lasham antenna. The three SCPS protocols are shown at the top of the stack. Since SCPS-SP is designed to work on top of SCPS-NP, which was not part of the SSFE, a small set of interface routines were developed to present a SCPS-NP-like interface to the CCSDS lower layers. This interface layer is not shown in the diagram, since it does not generate any information that is carried across the network. The SCPS stack rode on top of the CCSDS data link protocol which interfaced directly with the RF system. Each of the these components will be discussed in detail below.

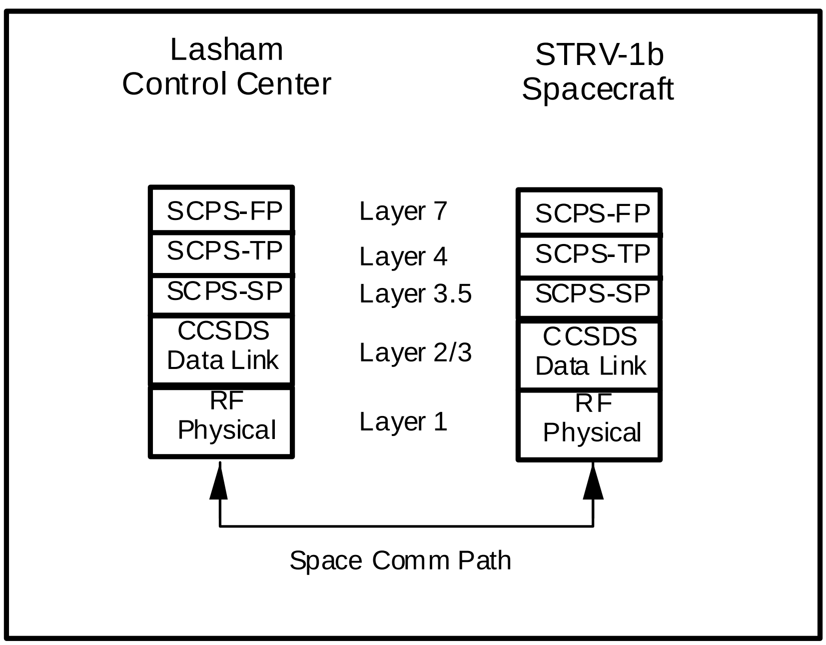


Figure G-1. STRV Communications Model via Lasham Antenna

Figure G-2 shows the model for communications via the DSN. The CLTU containing the SCPS packets are passed to the NASCOM blocker (labeled here as layer 2' for convenience) which is then transmitted over terrestrial communications links (which may use communications satellite hops for part of the path) using a 9.6 kbps modem. At the DSN station the data is deblocked and passed to the station RF system for uplink to the spacecraft. The telemetry follows the same path in the opposite direction. Under certain circumstances, it is possible that both models are in use simultaneously, with the uplink following one model and the downlink following the other.

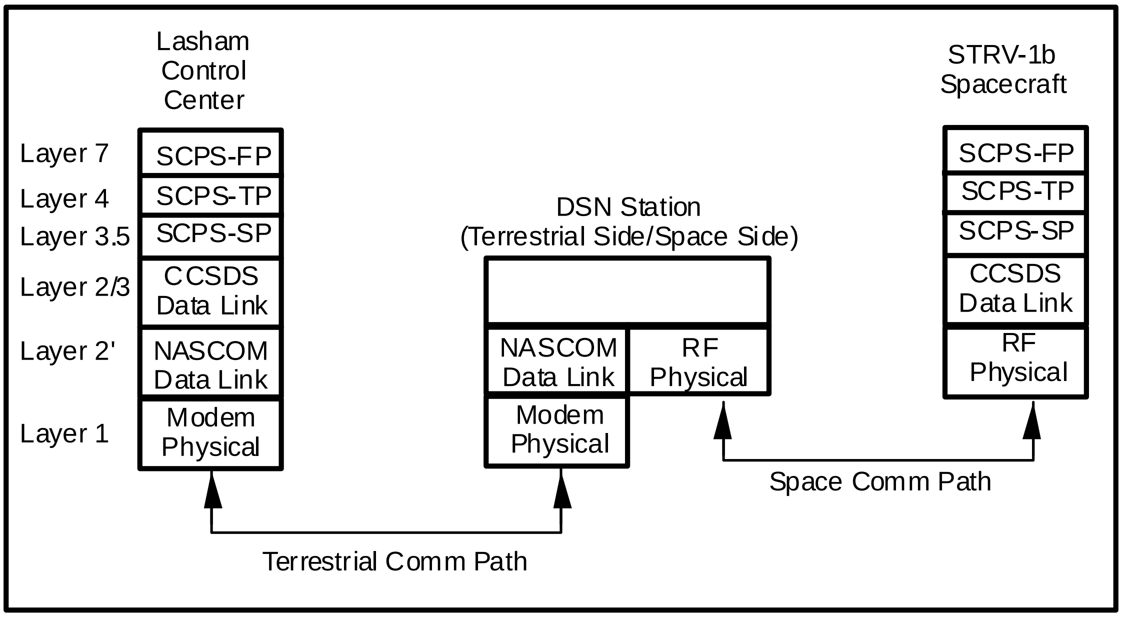


Figure G-2. STRV Communications via DSN

Table G-1 list the specifications for the STRV communications link. The RF parameters are for the S-band link between the spacecraft and the ground antenna. Data parameters are for the bit streams as they appear at the CCSDS data link layer. The specifications for the NASCOM blocker/deblocker (labeled layer 2' in the model) are considered immaterial to the SSFE; however, fact that the blocker held the data in limbo until a frame was filled contributed to transmission delays, but this behavior is not listed in the specifications.

Note that the telecommand frame length is variable. The frame size depends on the number of telecommand packets in the queue. If telecommand system does not wait to fill a frame; if there are no packets in the queue after as frame is formed, the frame is sent. If there are no packets submitted, the SCI generates idle frames - the uplink is always active.

Although there were two telemetry rates available, the SSFE used only the 1 kbps rate. Similar to the uplink, the telemetry downlink contains fill data (idle packets) when there is no data being submitted for downlinking.

Table G-1. STRV Link Specifications

Parameter	Value
Uplink frequency	2.09 GHz
Downlink frequency	2.27 GHz
Modulation Method	Split-Phase Binary/PSK/PM
Modulation Index	1.1 rad
Telemetry data rates (downlink)	1.0 kbps, 500 bps, selectable
Telecommand data rate (uplink)	125 bps
Coding	PCM split-phase binary (Biϕ-L)
Telemetry format	ESA Packet Telemetry (CCSDS)
Telemetry frame length	2624 bits
Telecommand format	CCSDS Telecommand
Telecommand frame length	Variable 64 to 2048 bits

From the standpoint of the SSFE, the most significant attributes of the STRV communications environment are the path delay and noise. The contributors to path delay include traffic congestion, processing delays of intervening equipment, clocking delay, and propagation time. From the viewpoint of the SCPS protocols, all delay appears the same - the delay affects the entire data set, and so, for a given session, the delay is relatively manageable. Noise is another matter. Noise corrupts portions or even individual units of the data set, and, as was experienced during the SSFE, the characteristics of the noise, and thus its affect on the data, can vary widely during a session.

The STRV RF link presented a challenge for both the SSFE and for the DRA mission personnel. The original mission plan had been for the Lasham 12 meter antenna to be the primary mission antenna. Just one week before launch, it was discovered that the STRV antennas were too long when the spacecraft were placed on the Ariane launcher immediately beneath the primary payload, an Intelsat communications satellite. After a very hurried test at Farnborough, the spacecraft antennas were cut from 19 cm to 12 cm. The affect on the spacecraft antenna pattern was almost catastrophic for the Lasham antenna system, as will be shown below. The DRA immediately requested and received support from the DSN as the primary means of communications with the spacecraft. The larger antennas of the DSN - 26 meters versus 12 meters - permitted almost 100 percent error-free communications. The cost was a substantial decrease in the amount of available communications time. Even though there were long periods of time every day in which the spacecraft were visible from at least one DSN station, the demand for DSN antenna time is large, and so users are apportioned support time accordingly.

The cutting of the antennas was a severe blow to the mission, but it was discovered that some communications between the spacecraft and the Lasham antenna were possible. This was a benefit to the SSFE because it allowed the SCPS protocols to be exercised under very stressful conditions. In addition, the ability to communicate with the spacecraft over long periods without worrying about a support schedule was very helpful. Since the Lasham uplink was very reliable, a SCPS software upload could be transmitted to the on-board computer and verified by commanding the checksum to be sent down. Since this only required a small amount of data to be returned, the command could be repeated until the telemetry frame containing the data could make it through the noise.

An attempt was made to calibrate the Lasham RF link for the SSFE so that reliable bit error rate estimates could be made to support the performance assessment of the transport protocol. This turned out to be difficult, but the effort revealed much about the signal environment the transport protocol had to overcome.

Figure G-3 shows a typical spacecraft signal as received by the antenna at Lasham. The vertical scale is the level of the signal at the input to the receiver which is about 45 dB above the level received by the antenna. The plot covers three SCPS-TP data runs on 31 July. The signal was sampled at a rate of once per second. The white line through the center of the data indicates the general trend of the mean signal level; however, as is readily evident, the signal has a wide range of variation over short periods of time. The dashed line running across the plot at about -84 dBm is the approximate threshold at which bit errors start occurring in the telemetry. Note that there appears to be a periodic nature to the variations. As it turns out, there are a number of cyclical features to the signal. The period of the cycle apparent in this plot is approximately 3.8 minutes. This is not the period of rotation of the spacecraft. One hypothesis is that this particular variation is caused by the precession of the spacecraft about its nutation axis[6]. There appears to be a longer period variation on the signal, possibly due to changes in the nutation angle. There is also a shorter period variation caused by spacecraft rotation that will be discussed below. The signal dropout shown in the plot was a common though not frequent phenomena during the SSFE.

---

6 The antenna is mounted almost exactly coincident with the +Z axis of the spacecraft which is aligned approximately with the North Pole of the earth. The spacecraft spins about the Z axis at a rate of about 5 rpm. The nutation angle was about 5 degrees in September 1994.

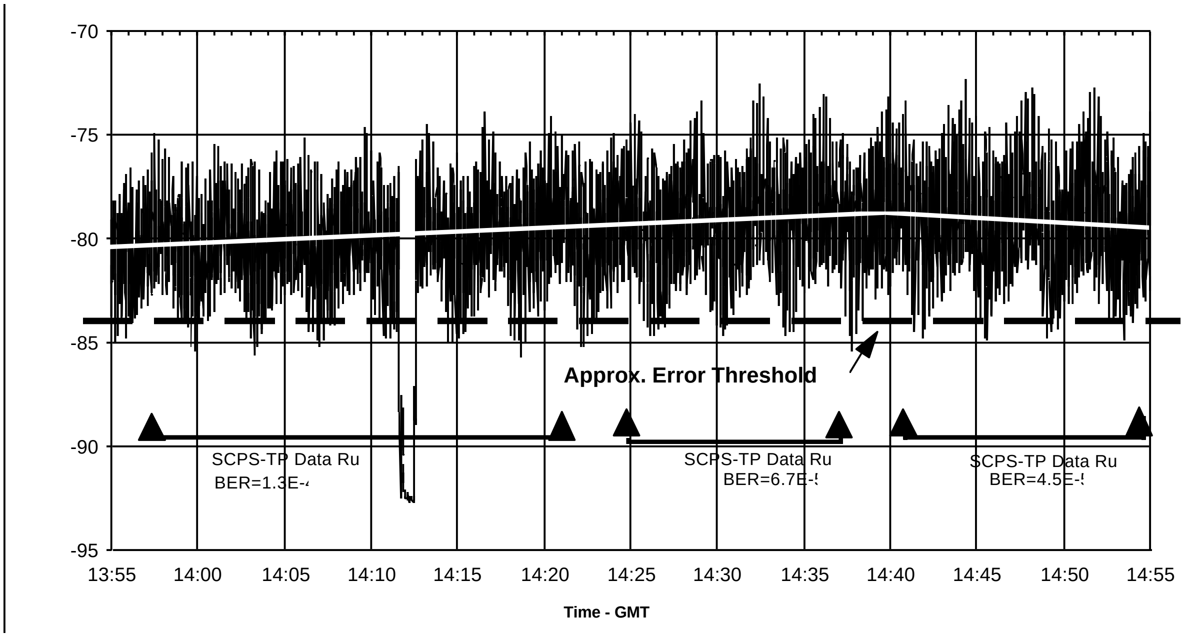


Figure G-3. STRV Signal via Lasham Antenna

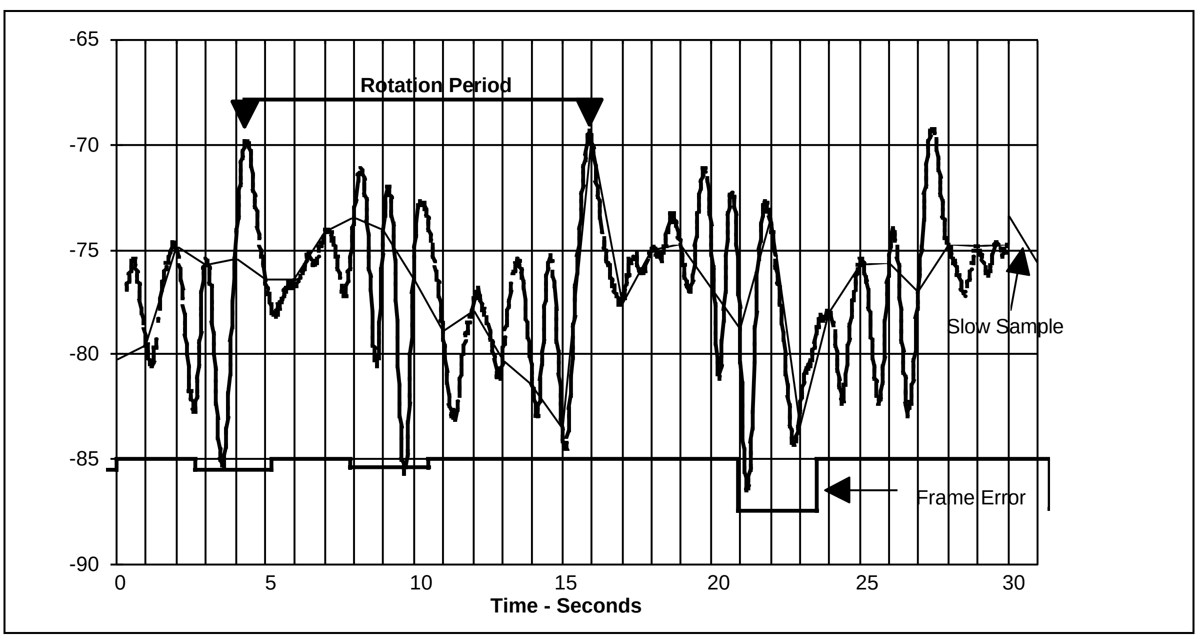


Figure G-4. Fast RSL Sample With Frame Errors Superimposed

Figure G-4 shows the fine structure of the spacecraft signal. Periodically during the data runs the RSL Logger was commanded to take a fast sample. These were made at a rate of about 100 per second for 30 seconds. (These were made infrequently because of the large data files generated by this sample rate.) The graph shows the slow speed sample lain over the fast samples to illustrate the smoothing of the data imposed by the slow sampling. The fast sample shows the cyclic nature of the signal caused by the rotation of the spacecraft. The period of rotation shown here is about 11.7 seconds, or approximately 5 rpm.

Superimposed on the RSL data is a representation of the frame errors that occurred during this session. The width of the error “pulse” is equivalent to the duration of a telemetry frame which is 2.624 seconds (2624 bits at 1 kbps). The Depth of the “pulse” is proportional to the number of bit errors detected in the frame: The two pulses at the right represent one error each while the pulse at the right represents five errors. Note that the errors coincide with signal levels below about -84.5 dBm. Note also the pulse-like nature of the signal at these low levels.

Figure G-5 illustrates a typical STRV signal as recorded at a DSN tracking station. The point in the antenna/receiver system at which the measurement is made is unknown. The apparent dropouts in the signal are intentional - the STRV spacecraft transmitters are turned off for 15 minutes every so often to allow them to cool. The point to be made is that the telemetry data received from the DSN stations was almost entirely error-free.

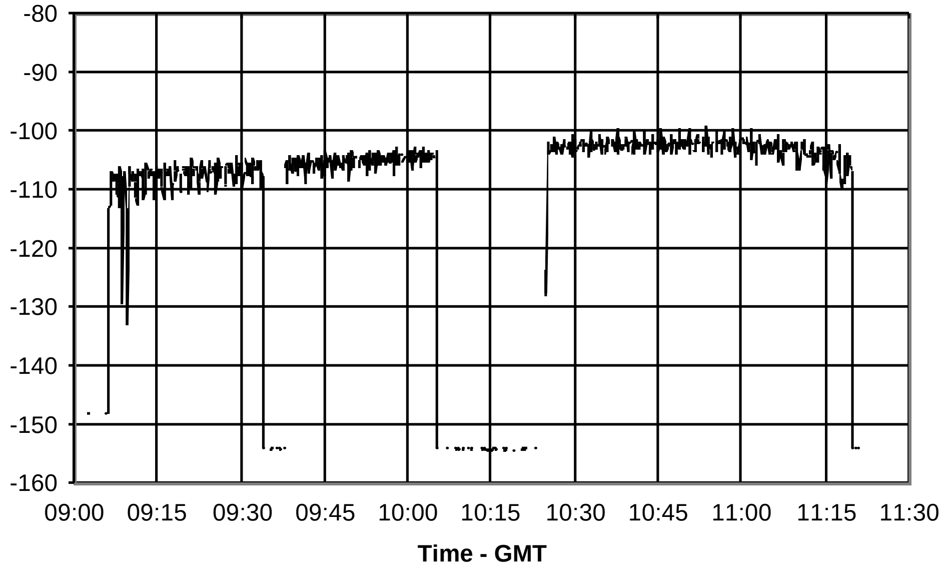


Figure G-5. Typical STRV Signal at a DSN Antenna

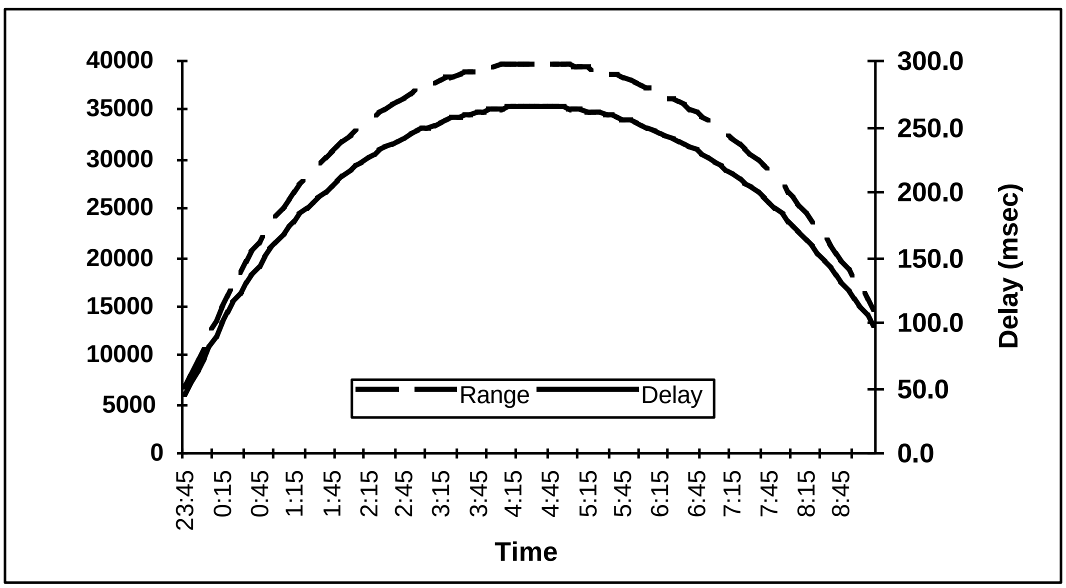


Figure G-6. STRV Range and Propagation Delay For Lasham

The period of time covered by the DSN signal plot is approximately two hours. Over that time period the signal is seen to increase from about -108 dBm to about -102 dBm and the start to decrease again. This illustrates the change in average signal level due to the range of the spacecraft from the tracking antenna. The change in range also changes the propagation delay. Figure G-6 shows the range and two way propagation time for a typical Lasham view. Recall that the spacecraft is in a geosynchronous transfer orbit with an apogee of about 36,000 km and perigee of 300 km. This gives it a period of 10 Hr. 35 min., with most of that time spent at the higher altitudes. Since the spacecraft is at an inclination of 7 degrees and Lasham is at 51 degrees North latitude, the spacecraft is never visible to Lasham below about 5000 km slant range. However, other factors reduce the usable view time, the most important factor being the aspect angle between the spacecraft and the ground antenna. Since the spacecraft antenna is pointing North, the Lasham antenna looks into the spacecraft antenna from a less than ideal angle even at the best of times.

The preceding discussion of the nature and behavior of the STRV RF signal provides the basis for understanding the corruption of the telemetry data, which was one of the principal areas of investigation for the SCPS-TP tests. The telemetry data is modulated directly onto the downlink signal. The largest data structure in the telemetry is the CCSDS frame. The frame contains successively smaller units of information, as defined by each of the successively higher protocols involved in the communication.

In order to be able to use all of the information contained in the telemetry so that corruption can be detected, it must go through some preliminary processing on the ground. The raw STRV telemetry bit stream from the bit synchronizer (Lasham contacts) or from the NASCOM blocker/deblocker (DSN contacts) is connected to a frame synchronizer card contained in the data handling terminal (DHT). The frame synchronizer identifies the frame, performs a CRC check on the data and passes it on to both the real-time processing software and to a telemetry log file.

The STRV telemetry frame transmitted by the spacecraft is 328 bytes long (2624 bits). On the ground, the frame synchronizer adds an additional 2 byte frame sequence header and an 8 byte trailer (total log file data frame = 338 bytes = 2704 bits) that contains a time tag and the CRC check results (pass/fail). When the DHT is initialized for a run, the logging software opens a log file and writes a 12 byte file header. The telemetry frames, along with their associated headers and trailers, are written sequentially to the log file. The frame sequence number begins at some arbitrary value between 4100 and 41FF hex and increments sequentially, always resetting to 4100. Note that since only the last byte sequences, frame sequence numbers repeat every 256 frames. The frame sequence number also restarts at some arbitrary value following an interruption of the telemetry data stream. Figure G-7 illustrates the telemetry log file structure.

	Data Frame #1			Data Frames 2 thru n
File Hdr	Frame Seq #	Telemetry Frame	Trailer	
12 byte	2 byte	328 byte	8 byte	338 byte each

Figure G-7. Telemetry Log File Structure

The file header contains identification information, the most important of which is the identification of the year. Bytes 4 and 5 contain the year (07CC hex = 1996). Following the file header is the first frame sequence number added to the received telemetry frame by the data logging software. This 2 byte number starts at a random value, as stated previously. The first telemetry frame starts immediately after the first frame sequence number. This is followed by the first frame trailer and then the next frame sequence number and so on.

The telemetry frame structure is shown in Table G-2. (Note: the first byte is numbered 0 using the *telemetry* frame numbering system. To obtain the position in the logger's *data* frame numbering system, add 2 to the values in the table. In the data frame numbering system, bytes 0 and 1 are occupied by the frame sequence number.).

Table G-2. Telemetry Frame Structure

Start Byte	End Byte	Field	Total Bytes	Contents
0	3	Synch Marker	4	1ACF FC1D
4	9	Transfer Frame Primary Header	6	0451 xxxx 0000
10	99	Telemetry Packet 1	90	User data (OBC or DAU)
100	189	Telemetry Packet 2	90	User data (OBC or DAU)
190	279	Telemetry Packet 3	90	User data (OBC or DAU)
280	321	Telemetry Packet 4 (DAU)	42	DAU data
322	325	Operational Control Field	4	01xx xxxx
326	327	Frame Error Control Word	2	0 to 65535 (CRC)

The 8 bytes appended by the frame synchronizer on the ground are shown in . The bytes are numbered as a continuation of the positions in Table G-3.

Table G-3. Telemetry Frame Trailer Added by Frame Synchronizer

Start Byte	End Byte	Field	Total Bytes	Contents
328	329	Time of year part 3	2	Least significant bits [7]
330	331	Time of year part 2	2	mid-significant bits
332	333	Time of year part 1	2	most significant bits
334	335	CRC Pass/Fail	2	0000 = fail 0001 = pass

A typical frame containing SCPS data is shown in Table G-4 below.

[7] The time of year is recorded in units of milliseconds since the beginning of the year. Thus, since there are 86,400,000 milliseconds in a day, a value of 86,399,000 (1 second less than a day) is 23:59:59 on January 1, and 86,401,000 (1 second more than a day) is 00:00:01 on 2 January. Note that the sequence of the bytes must be read in the following byte order: 332, 333, 330, 331, 328, 329 where byte 332 contains the most significant bits.

In order to be able to derive a bit error count from a data stream it is necessary to know either the original value of the individual bits (i.e., what the values were before the bits were transmitted) or to know the original value of a group of bits. Although it would be desirable to know 100 percent of the information, much knowledge can be gained when only portions of the information are known. The STRV telemetry stream contains fairly large quantities of "knowable" information; i.e., it is possible to determine what the value of some of the information was before it was transmitted.

The chain of knowledge begins with the first frame of data. When the frame synchronizer identifies the first frame of telemetry at the beginning of the session, the logging software adds an arbitrary starting sequence number to the beginning of the frame. From then on, the frames are numbered sequentially until it reaches 41FF and resets to 4100, or until there is a break in the telemetry stream of sufficient duration to stop the frame synchronizer. When the telemetry resumes and the frame synchronizer reacquires frame synch, the added frame sequence number is not the next number after the frame before the interruption. Therefore, significant data dropouts can be identified by checking the sequence numbers for breaks in the sequence. The amount of lost data can be computed directly by computing the difference between the time stamps appended to the ends of the frames of the frame just prior to the interruption and the first frame after the interruption. Since there are 2624 bits in a frame and a frame is 2.624 seconds long, dividing the time difference by 2.624 yields the exact value of the data loss.

Table G-4. Typical Telemetry Frame

Byte Nmbr	Data								Field
	Byte Count								
	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15	
	0000	0000	07CC	00A4	0001	0001			File Header
	41D2								Frame Sequence #
0-3	1ACF	FC1D							Frame Synch
4-9	0451	1B1B	0000						Frame Header
10-	87FF	C000	0053	5555	5555	5555	5555	5555	Tlm Packet (idle)
	5555	5555	5555	5555	5555	5555	5555	5555	
	5555	5555	5555	5555	5555	5555	5555	5555	
	5555	5555	5555	5555	5555	5555	5555	5555	
	5555	5555	5555	5555	5555	5555	5555	5555	
-99	5555	5555	5555	5555	5555				
100-	0207	C352	0001	0002	9A61	958D	0090	01C6	Tlm Packet (SCPS)
	8AE0	3690	9AD4	0001	0203	0405	0607	0809	
	0A0B	0C0D	0E0F	1011	1213	1615	1617	1A19	
	1A1B	1E9B	9ADE	2121	2625	F625	66A7	7AA9	
	2E2B	8CE4	6F2E	3521	3233	7435	3637	3839	
-189	3A3B	3C3F	3E3F	4041	4263				
190-	87FF	C000	0053	D555	5555	5555	5555	5555	Tlm Packet (idle)
	5555	5555	5555	5555	5555	5555	5555	5555	
	5555	5555	5555	5555	5555	5555	5555	5555	
	5555	5555	5555	5555	5555	5555	5555	5555	
	5555	5555	5555	5555	5555	5555	5555	5555	
-279	5555	5555	5555	5555	5555				
280-	8001	D81B	0023	02A4	7020	8040	75E1	0000	DAU Packet
	0000	0000	0000	0000	C453	3820	4000	5FCE	
-321	4000	0000	0000	0001	0000				
322-327	0108	0453	585F						Control & CRC
328-335	2596	4AF6	0004	0000					Trailer (added)

The next link in the chain of knowledge is obtained from the frame error field. The spacecraft computer computes a cyclic redundancy check (CRC) value for the telemetry frame and appends it to the frame just before it is transmitted to the ground. On the ground the frame synchronizer makes the same computation on the received frame and compares it to the value appended by the spacecraft. If the values match, the value of the CRC Pass/Fail field appended to the frame by the DHT is set to zero; if it doesn't match, it is set to one; thus, if the value of the field is one, then at least one bit is in error. Therefore, by counting the number of errored frames contained in a set of frames comprised by a data run, a lower limit for the bit error rate over the period of the run is obtained. For example, if the total number of frames is 1000, and there are 100 frames in error, the lower limit for the bit error rate during the run is:

$$100/(1000*2624)=3.81*10^{-5} \text{ BER}$$

The frames contain fixed fields which can be checked. For example, the frame synch marker should always have the hex value 1ACF FC1D. This provides a minimum of 32 bits that can be checked in every frame.

Finally, one can take advantage of the idle patterns placed in the telemetry when there is no user data being passed. Any or all of the 90 byte telemetry packets can be set to the idle pattern, which is a header of 87FF C000 followed by all fives. This gives up to an additional 2160 bits that can be checked for errors.

Figure G-8 shows a telemetry frame with errors. The first telemetry packet is idle, so the data following the packet header 87FF C000 0053 should be all fives, but one value is a 7. The second identifiable error is in a SCPS packet, and would not be detected using this method. The following paragraph describes a technique to count the errors in SCPS packets.

Another technique was developed, in which the known content of SCPS-TP data packets is analyzed. The SCPS-TP test drivers used a fixed test pattern for the user data of the packet. The location of this test pattern depends on the size of the SCPS-TP header, which changes as a function of the protocol configuration under test. However, if the protocol configuration is known, the start of the user data can be identified and the contents analyzed. This technique yielded results that were relatively close to the results derived by counting errors within the frames: the average difference was a factor of two, and the maximum difference was less than a factor of 5.

The preceding description of the RF link and the data stream provides a basis for understanding the corruption environment in which the SSFE was conducted. It also illustrates the difficulties associated with defining the error environment. The cyclic impulse-like noise on the RF link caused errors to be grouped within the data stream. The SCPS data did not occupy the entire data stream, but was confined to well-defined locations within it. Thus, it was possible, and even likely, that a SCPS packet would contain no errors even though the frame in which it was contained had a large number of corrupted bits. Which, then, is the true error environment in which SCPS operated? After much analysis, neither of these techniques was selected. Retransmission protocols respond to receipt or loss of an entire packet. The *degree* to which a packet is corrupted is irrelevant if *any* of the packet is corrupted. The technique described in Section 5 of this document is based on packet loss. It estimates the bit-error rate based on an assumption that *only one bit per corrupted packet is bad*. This yields a conservative estimate of bit-error rate, that should hold even if the errors are not bursty. As a result, it was decided to use the packet-error based technique in Section 5 of the document, and to describe the total environment in this appendix.

```
Error 1

1ACF FC1D 0451 B1B1 0000                                                    Frame Header

87FF C000 0053 5555 5555 5555 5555 5575 5555 5555 5555 5555 5555 5555 5555 5555
5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555
5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555 5555
Telemetry Packet 1 - Idle

0607 C2ED 0053 00A5 0037 0000 AB01 0000 0002 5018 4D01 150D 0000 0001 0203 0405
0607 0809 0A0B 0C0D 0E0F 1011 1213 1415 1617 1819 1A1B 1C1D 1E1F 2021 2223 2425
2627 2829 2A2B 2C2D 2E2F 3031 3233 3435 3637 3839 3A3B 3C3D 3E3F
Telemetry Packet 2 - SCPS

0607 C2EE 0053 00A5 0037 0000 AB41 0000 0002 5018 4D01 14CD 0000 0001 0203 0405
0607 0809 0A0B 1C0D 0E0F 1011 1213 1415 1617 1819 1A1B 1C1D 1E1F 2021 2223 2425
2627 2829 2A2B 2C2D 2E2F 3031 3233 3435 3637 3839 3A3B 3C3D 3E3F
Telemetry Packet 3 - SCPS

8001 44B1 0023 806D 7030 8840 6683 0000 0000 0000 0000 0000 0453 3830 4800 5FCE
0000 0000 0000 0000 0000 0108                                       Telemetry Packet 4
- DAU

0453 532B 9CF8 28E3 0004 0000                                            DHT Frame Trailer
Error 2
```

Figure G-8. Errored Telemetry Frame

Glossary

ABR	Available Bit Rate
Ack	Acknowledgment
AFSCN	Air Force Satellite Control Network
API	Application Programming Interface
APID	Application Process Identifier
ATM	Asynchronous Transfer Mode
awk	Aho, Weinberger, Kernighan
BER	Bit-Error Rate
BMDO	Ballistic Missile Defense Organization
bps	bits per second
BSD	Berkeley Software Distribution
CCSDS	Consultative Committee for Space Data Systems
CLCW	Command Link Control Word
CLTU	Command Link Transmission Unit
COTS	Commercial Off The Shelf
CRCs	Cyclic-Redundancy Codes
DAU	Data Acquisition Unit
DLU	Down Link Unit
DOD	Department of Defense
DRA	Defence Research Agency
DSN	Deep Space Network
ESA	European Space Agency
GMT	Greenwich Mean Time
GTO	Geostationary Transfer Orbit
HQ	Headquarters

IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
JPL	Jet Propulsion Laboratory
kB	kilo byte (1024 bytes)
kbps	kilobit per second (1000 bits per second)
kg	kilogram
LNA	Linear Amplifier
MIL STD	Military Standard
MSS	Maximum Segment Size
MTR	MITRE Technical Report
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communication Network
OBC	Onboard Computer
OBDH	Onboard Data Handling System
PAWS	Protect Against Wrapped Sequence Numbers
PC	Personal Computer
RAM	Random Access Memory
RF	Radio Frequency
RFC	Request for Comments
ROMs	Read-Only Memories
RSL	Received Signal Level
RTS	Remote Tracking Station
RTT	Round Trip Time
Rx	Receiver
SCPS	Space Communications Protocol Standards
SCPS-FP	Space Communications Protocol Standards - File Protocol

SCPS-NP	Space Communications Protocol Standards - Network Protocol
SCPS-SP	Space Communications Protocol Standards - Security Protocol
SCPS-TP	Space Communications Protocol Standards - Transport Protocol
SCPS-TWG	Space Communications Protocol Standards - Technical Working Group
SIGCOMM	(Association for Computing Machinery) Special Interest Group on Communications
SACK	Selective Acknowledgment
SDIB	STRV Data Interchange Bus
SNACK	Selective Negative Acknowledgment
SSFE	SCPS-STRV Flight Experiment
STRV	Space Technical Research Vehicle
SWS	Silly Window Syndrome
SYN	Synchronize
Tcl	Tool Control Language
TCP	Transmission Control Protocol
Tk	Tcl X Windows Toolkit
Tx	Transmitter
UK	United Kingdom
ULU	Up Link Unit
USSPACECOM	United States Space Command

Distribution List

Internal

W159
R. C. Durst (10)
D. E. Ernst
G. J. Miller
R. C. Mitchell
A. S. Reitzel
N. L. Schult
D. Thomson
M. J. Zukoski

D048
G. M. Comparetto
B. A. Feldmeyer
J. G. McIlnay
J. Muhonen
R. Ramirez
E. J. Travis (Gemini Industries),
c/o MITRE Reston M/S W650

Project

USSPACECOM, J4
Attn: Col. Flynn
250 S. Peterson Blvd.
Peterson Air Force Base
Colorado Springs, CO 80914-3310

USSPACECOM, J4P
Attn: Kathleen Dismukes
250 S. Peterson Blvd.
Peterson Air Force Base
Colorado Springs, CO 80914-3310

SMC/ADC
Attn: Capt. J. Gadway
SMC/ADC
Los Angeles AFB, CA 90009-2960
(5 copies + disk in DOS format)

SMC/AX
Attn: Maj. J. Cole
SMC/AXES
Los Angeles AFB, CA 90009-2960
(original + 4 copies + disk in DOS format)

Mr. John Rush
HQ NASA Code OI
300 E. Street S.W.
Washington, DC 20546
(5 copies + disk)

USA CECOM
ATTN: AMSEL-PE-FC
Ft. Monmouth, NJ 07703-5027
(Letter of Transmittal only)

JPL M/S 301-235
Attn: Mr. Adrian Hooke
4800 Oak Grove Drive
Pasadena, CA 91109-8099
(5 copies + disk)

External

DISA/JITC
Attn: Mr. Leo Hansen
DISA/JITC
Ft. Huachuca, AZ 85613-7020

Mr. Nick Shave
Space Systems
Defence Research Agency
R16 Bldg.

Farnborough, Hampshire
GU146TD, U.K.
(5 copies)

LT CDR Kevin Shaw
HMS Northumberland
BFPO 345
United Kingdom

Mr. Steven Sides
SAIC
The Stonybrook School
Route 25a and Cedar Street
Stonybrook, NY 11790

Mr. Howard Weiss
SPARTA, Inc.
9861 Broken Land Pkwy
Suite 300, Columbia, MD 21046