COMPARISON OF FILE TRANSFER USING SCPS FP AND TCP/IP FTP OVER A SIMULATED SATELLITE CHANNEL

Stephen Horan and Ru-hai Wang
New Mexico State University

ABSTRACT

The CCSDS SCPS FP file transfer performance is compared with that of TCP/IP FTP in a simulated satellite channel environment. The comparison is made as a function of channel bit error rate and forward/return data rates. From these simulations, we see that both protocols work well when the channel error rate is low (below 10^{-6}) and the SCPS FP generally performs better when the error rate is higher. We also noticed a strong effect on the SCPS FP throughput as a function of forward transmission rate when running unbalanced channel tests.

KEY WORDS

SCPS FP, TCP/IP FTP, networking protocols, channel simulations

INTRODUCTION

In order to reduce the risk associated with new technology in space data systems, there needs to be a considerable amount of testing of that technology prior to flight system usage. This testing is also needed in software-based technology such as networking protocols for command and telemetry systems. As more satellite designers consider using techniques such as Internet-type protocols in their designs, the networking protocols themselves must be testing in channel environments similar to that found in space channels. This includes channel error performance, propagation delays, and data rate restrictions. This includes space channels where the command link (forward link) operates at a much slower rate than the data link (return link).

The Internet-type protocols such as the TCP/IP file transfer protocol (FTP) do not normally operate on links with large point-to-point delays. Recent advancements such as the Consultative Committee for Space Data Systems (CCSDS) Space Communications Protocol Standard (SCPS) suite of protocols are designed to overcome these space channel problems and provide reliable data transport. The SCPS suite of protocols includes four elements: a file transfer protocol (FP), a transport protocol (TP), a security protocol (SP), and a network protocol (NP) [1]. In the tests described here, we examine the relative performance of the SCPS FP protocol to that of the TCP/IP FTP protocol when transmitting various length data files. The files were tested using the Space-to-Ground Link Simulator (SGLS) described in a companion paper in these *Proceedings* [2]. The tests examined performance of the protocols at varying channel Bit Error Rates (BER) and with varying forward and return data rates. In this paper, we describe the test configuration and give sample results that illustrate the performance. The next section briefly describes our test environment along with the SCPS configuration, the SGLS simulator, and the test methodology. Following the test description, we present the test results and analysis for the two protocols under the conditions of balanced channel data rates (the same data rate on the forward and return links) and unbalanced channel data rates (differing data rates on the forward and return links). Finally, we present the conclusions drawn about the relative performance of the protocols from the simulation runs conducted to date.

TEST ENVIRONMENT

1. PROTOCOL CONFIGURATION

The SCPS protocol stack was designed to be modular and to be tailored by the user to meet specific mission needs. Because the configuration is flexible, the performance can also vary depending upon the options chosen. In our tests, the following default SCPS configuration was used

- SCPS-SP is not included in the protocol stack
- The networking protocol is IP and not SCPS-NP
- The encapsulation protocol is the User Data Protocol (UDP) rather than IP raw sockets
- SCPS-FP operates over SCPS-TP rather than over the system's TCP in kernel

We used the version 1.1.8 reference implementation of the SCPS protocol stack that was provided by MITRE [3]. The TCP/IP protocol stack used was provided with the host computer operating system distribution and was used without changes. Both the TCP/IP and SCPS protocols utilized the Point-to-Point Protocol (PPP) link-level protocol provided as part of the Linux operating system distribution.

2. SGLS SIMULATION

The networking simulation environment is built around the SGLS space channel simulator mentioned earlier. The full details of the environment and testing can be found in [4] and [5]. Two 133-MHz PCs having 16 MB of memory and running the Red Hat Linux 5.2 operating system are connected to the SGLS via RS-232 cables. These computers are used for hosting the networking protocols and as the source and destination nodes for data files transmitted during the tests.

The SGLS channel error simulator is inserted between the source and destination computers to provide the simulated space-to-ground channel. The SGLS provides the capability to configure links with the following properties:

a. independent forward and return data transmission rates from 2400 baud through 115200 baud, based upon user selection;
b. independent link error rates on the forward and return data links, from 10^{-2} down to 10^{-6}, based upon user selection;
c. independent statistical error models for the forward and return data links, i.e. forward link may be AWGN while the return link is AWGN mixed with interference.

3. TEST METHOD

In the tests described here, the channel error statistics were drawn from a Gaussian white noise distribution and were run at BER levels of 10^{-4}, 10^{-5}, 10^{-6}, and 0 (no channel errors). File transfer times were used as the figure of merit for the link and were measured by transmitting 1 KB, 10 KB, 100 KB, and 1000-KB files. In each test, ten transfers were run for each file size at the BER and data rate under test. The results for the ten runs were then averaged and this average time was adopted as the final transmitting time for each file size used in the performance analysis. Both the FP and FTP protocols were tested under the conditions of balanced and unbalanced data links. Occasionally, a transfer was not completed properly due to the transfer timing out or taking an exceedingly long time to complete, and its run time was not included in the average but the result was noted and mentioned in the discussion of the test results.

BALANCED DATA RATE TESTS

The first set of tests conducted were with the same data rate on the forward and return links forming the set of balanced rate tests. In these tests, baud rates of 9600, 19200, 57600, and 115200 bits per second were used in testing both protocols. The tests were run with the TCP/IP FTP configuration first, followed by the SCPS FP configuration. The results of the FTP tests are summarized by baud rate in Figure 1 where the transmission times for the various file sizes are displayed as a function of bit error rate. Each plot shows the transmission times for the 1-KB, 10-KB, 100-KB, and 1000-KB files with the 1-KB files taking the shortest time and the 1000-KB files taking the longest time. On each plot, the diamond marker on the y-axis represents the average time to transmit the same file ten times using a direct, null-modem cable without the SGLS in the process. This is to give a reference indication of the best performance possible with the SGLS

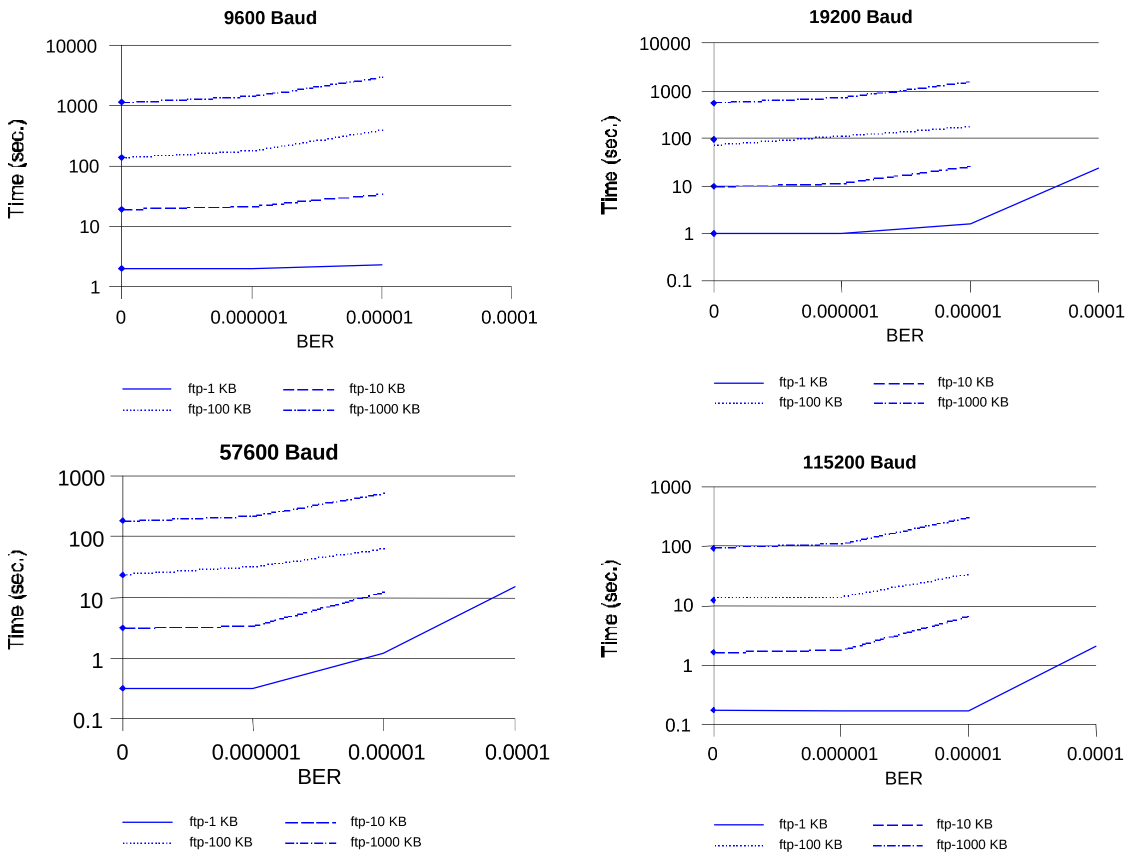

Figure 1 - File transmission time results using the FTP service as a function of BER and baud rate.

configuration at the indicated data transmission rate. Interesting items noted during these tests include:

a. The file transfer process at a BER of 10^{-4} was generally not possible. In these cases, after a relatively long period of no activity on the link, the file transfer was aborted, either automatically or manually, and restarted. The only file lengths that could be delivered at this BER were the 1-KB files. However, in each of the cases where delivery was possible, no test completed all ten experimental runs. The completion rates were

   i. At 9600 baud, 0 of 10 experiment runs were completed,
   ii. At 19200 baud, 8 of 10 experiment runs were completed,
   iii. At 57600 baud, 2 of 10 experiment runs were completed, and
   iv. At 115200 baud, 2 of 10 experiment runs were completed.

b. The file transfer throughput at a BER of 10^{-6} was nearly the same as the throughput at a BER of 0. However, as the BER was increased to 10^{-5}, the transmission times rapidly increased.

Figure 2 summarizes the results for the SCPS FP tests in the same manner as Figure 1. Each graph represents a different baud rate and shows the transmission time as a function of BER for the 1-KB, 10-KB, 100-KB, and 1000-KB files. Again on each plot, the diamond marker on the y-axis represents the average time to transmit the same file ten times using a direct, null-modem. The following interesting items were noted during these tests:

a. Successful completion of the file transfer process at a BER of 10^{-4} was most likely for the 1-KB file only. Again, for the longer files, the transmission was aborted after long periods of no link activity. As in the TCP/IP experiments, in each of the 1-KB transfer cases, no test completed all ten runs. However, the completion rates were higher than with TCP/IP FTP and were as follows:

   i. At 9600 baud, 0 of 10 experiment runs were completed, which was the same as FTP;
   ii. At 19200 baud, 8 of 10 experiment runs were completed, which was the same as FTP;
   iii. At 57600 baud, 6 of 10 experiment runs were completed, which was 40% higher than FTP, and At 115200 baud, 5 of 10 experiment runs were completed, which was 30% higher than FTP.

b. As with the TCP/IP FTP service, the file transfer throughput at a BER of 10^{-6} was nearly the same as the throughput at a BER of 0. However, as the BER was increased to 10^{-5}, the transmission times for SCPS FP did not show the same rapid increased as the TCP/IP FTP times did. Not all of the SCPS FP experiments were able to complete ten trials at a BER of 10^{-5}. This was a problem for the 100-KB and 1000-KB file lengths as follows:

   i. At 9600 baud, only 9 of the 10 experiments with the 100-KB files completed,
   ii. At 19200 baud, only 9 of 10 experiments completed with both the 100-KB and 1000-KB files, and
   iii. At 115200 baud, only 9 of 10 experiments completed with the 1000-KB files.

From these results, we can see that at the low BER ($0 \sim 10^{-6}$) configurations, both FTP and FP have essentially the same transmission times. As the BER increases, the SCPS FP throughput degrades slowly while the FTP throughput drops rapidly. This performance can be understood in terms of the congestion control methods of both protocols. In a space channel environment, the channel bit errors are the primary source of packet loss. It is known that TCP/IP considers packet losses to be the result of network congestion. When an error occurs, TCP invokes its congestion control rule and reduces its congestion

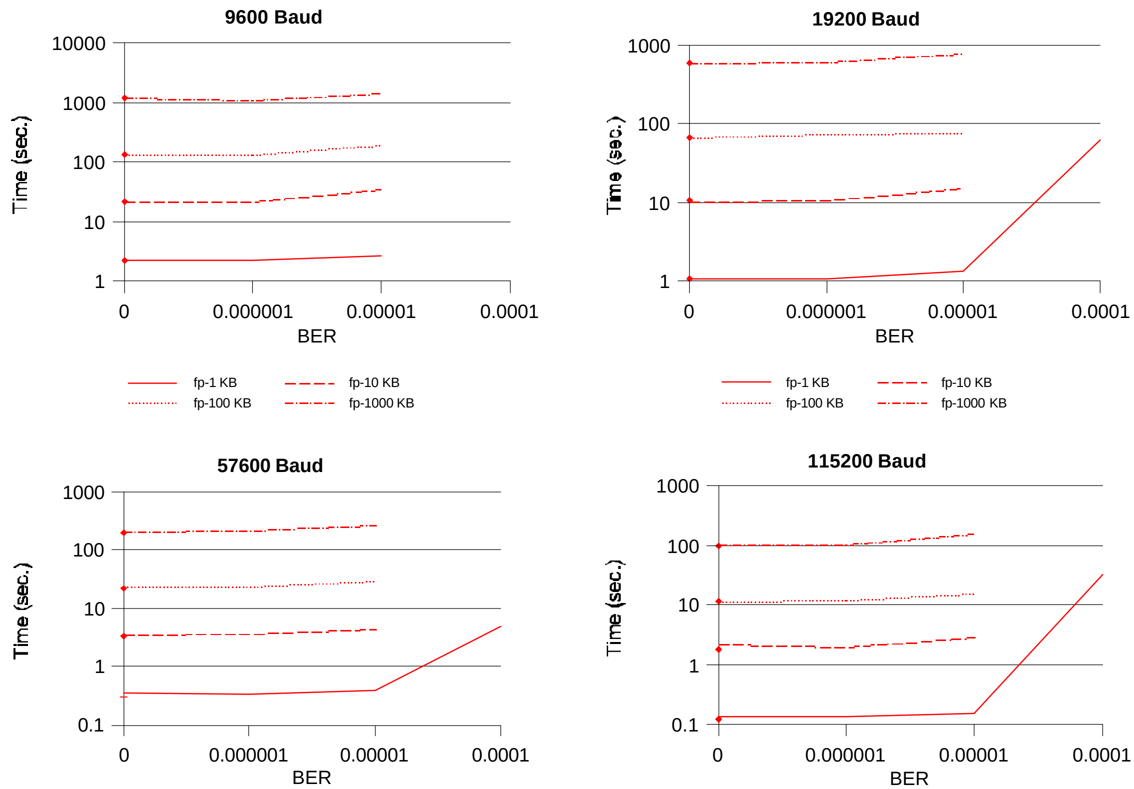

Figure 2 - File transmission time results using the FP service as a function of BER and baud rate.

window. When the BER is very low, TCP does not frequently invoke congestion control so the throughput remains high while a high BER causes TCP to reduce its congestion window and lower the throughput.

SCPS was designed with a different approach to avoid this problem and only invokes congestion control when it has evidence of network congestion actually occurring. SCPS also uses the round trip timer (RTT), Window Scaling and Selective Negative Acknowledgement (SNACK) techniques to increase the amount of data that can be transmitted [6]. In low-BER conditions, these techniques do not greatly enhance throughput and SCPS FP performs similar to FTP. But with high BER channels, the two protocols work differently. SCPS does not interpret bit errors as congestion. Instead, SCPS uses an open-loop token bucket rate control mechanism [7] that meters out transmissions at a specified rate. Additionally, the RTT and large Window Size capabilities make the sender continually transmit new data while recovering from bit-error-induced packet loss. The usage of the SNACK can also help improve the throughput under high-BER conditions. This is why SCPS FP does a better job of maintaining a transmission time similar to the no-error case at a large BER while the FTP throughput suffers as the error rate increases.

The plots for a BER of 10^{-4} in Figures 1 and 2 are based on only a few completed transfers, since both protocols had great difficulty in maintaining a connection at this BER, so their reliability as a measure of true performance is very low.

UNBALANCED DATA RATE TESTS

The second set of tests conducted were with different data rates on the forward and return links forming the set of unbalanced rate tests. In these tests, the return link was fixed at a 115200-bps transmission rate while the forward link transmission rate was either 2400 bps or 19200 bps. These forward link rates will allow us to see how the acknowledgement process affects the channel throughput. As before, the tests were run with the TCP/IP FTP configuration first followed by the SCPS FP configuration. The results of the FTP tests with two different baud rates are summarized in Figure 3 where the transmission times for the various file sizes are displayed as a function of BER. Interesting items noted during these tests include:

a. The file transfer process has the same sensitivity to BER as was seen in the balanced-transmission-rate TCP/IP FTP tests. BERs of 10^{-5} and 10^{-4} had a significant delaying effect on the transfer time.
b. The throughput is not as great as that found in the balanced transmission tests for the same return channel baud rate. For a forward/return link combination of 2400/115200 baud, the throughput for all four different file sizes is generally close to that for a balanced 57600-baud link. This indicates that a slow acknowledgment over the 2400-baud forward link reduces the link transmission rate to half of that found in a balanced configuration.
c. When the forward link rate is changed from 2400 baud to 19200 baud, the throughput does not change significantly which indicates that FTP is not highly sensitive to the acknowledgement rate.

As in the balanced transmission tests, only the 1-KB files could be transmitted at a BER of 10^{-4} for the 2400/115200-baud link combination. In this case, only 8 of the 10 transmission tests were completed. No attempt was made to try the 19200/115200 case at a BER of 10^{-4} since it was expected that only the 1-KB file case would be successful as in the previous tests.

The results of the SCPS FP experiments are displayed in Figure 4 in the same format as in Figure 3. Interesting items noted during these tests were as follows:

a. Unlike the TCP/IP FTP service, the SCPS file transfer process at a BER of 10^{-5} had nearly the same throughput as did the transfer process at a BER of 0.

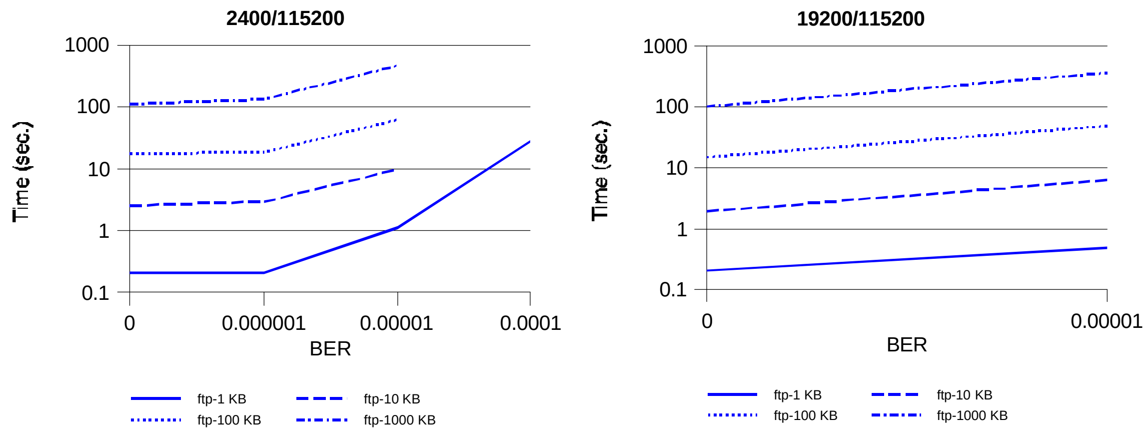

Figure 3 - Unbalanced transmission test time for the FTP service as a function of file size and BER

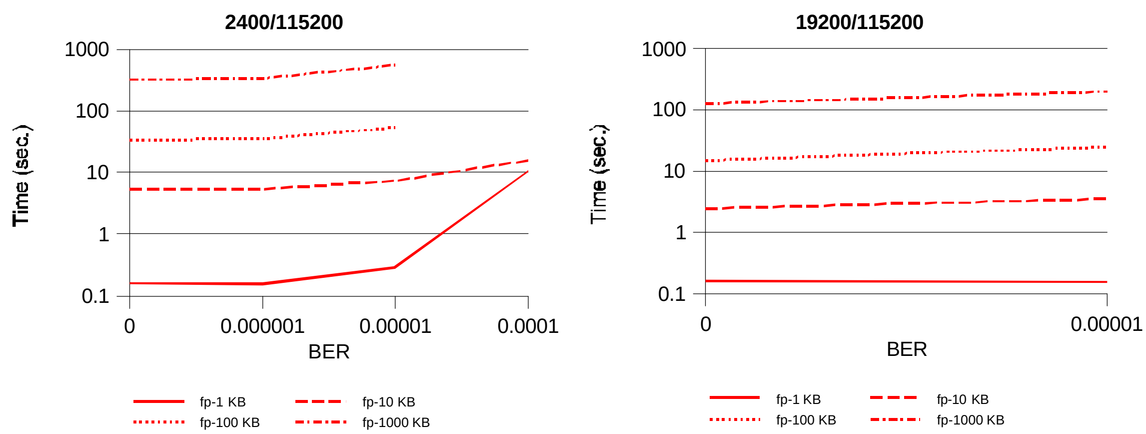

Figure 4 - Unbalanced transmission test for the FP service as a function of file size and BER

b. As in the FTP tests, the throughput is not as great as that found in the balanced transmission tests for the same data rate on the return link. For the 2400/115200 bps link combination, the transmission time for the 1-KB files is similar to that expected for a balanced 115200-baud link with a low BER. However, for longer files, the throughput is lower than that for a 57600-baud link. The slow acknowledgment rate through the 2400-baud forward link appears slows the return link throughput to one half that found on a balanced link. When the forward link rate is increased to 19200 baud, the throughput increases over the return link.

c. The difference in throughput between the cases of 2400/115200 baud and 19200/115200 baud is much greater than that found in the TCP/IP tests as can be seen by the decrease in transfer times between the two cases. Increasing the forward link rate to 19200 baud gave a throughput increase of approximately a factor of 2 for SCPS-FP while the FTP results were nearly the same.

As these figures illustrate, with the lower forward transmission rate of 2400 baud, the SCPS FP generally runs slower than TCP/IP. A clear exception are the 2400/115200 link tests where the throughput for the 1-KB files is generally higher with FP than FTP while all other cases run slower than FTP. This is because the small segment of a 1-KB file is less susceptible to bit errors and the transfer can be completed in one window cycle. This performance trend can be analyzed by considering the implications of the protocol timestamp to the performance. By default, the SCPS FP has its timestamp option turned on while TCP has its timestamps turned off be default. Enabling the timestamp option adds an additional 12 bytes to the transport segment header of 20 bytes. The resulting SCPS acknowledgement packet is 52 bytes long while the FTP acknowledgement packet remains at 40 bytes. When the acknowledgement link data rate is low, for example 2400 baud, the SCPS-TP sends fewer acknowledgements per unit time thereby decreasing the amount of data that can be transmitted [3]. But SCPS can improve its throughput relative to TCP by using the rate control option. This option is designed to control the maximum rate at which SCPS can send data. Specifically, it controls the frequency with which acknowledgements will be sent to avoid congestion on the acknowledgement link. These capabilities contribute significantly to the throughput in a high-acknowledgement-rate environment, especially when the link has a high BER. This is verified in the 19200/115200 bps tests where SCPS FP and TCP/IP FTP take about the same transmission time when the channel error rate is close to 0. When the channel BER increases to 10^{-5}, SCPS FP maintains its nearly flat transmission time while the TCP/IP transmission time starts increasing rapidly. At this BER, the SCPS FP throughput becomes higher than that of FTP. It is expected that when the acknowledgement link data rate becomes higher, for example 38400 or 57600 bps, the FP throughput will be much higher than FTP's.

CONCLUSIONS

As mentioned earlier, no attempts were made to optimize the protocol configurations in these tests but to use them in "out of the box" mode. From the testing completed to date, we conclude that channels, with bit error rates lower than 10^{-5}, need to be used if the data transmission is run without some form of forward error correcting coding. As the BER falls below this, TCP/IP FTP performance may become unacceptable for most space applications. While the SCPS FP performs better at higher BER channels than does FTP, the completion rate for transmissions may be a problem. As in the terrestrial Internet, incomplete data transmissions or timeouts may still be a problem with space channels even if the SCPS protocol is used. For this reason, users may need to limit file transfer size to better ensure data set completion in limited pass time. When running unbalanced forward and return link data rates, the data throughput is limited by the forward link rate that is assumed to be the slower link here. The slower forward link slows down the acknowledgement time, which lowers the overall performance of the link. In general, we conclude that the SCPS performs better than TCP/IP for the types of channels investigated here. However, even having a protocol stack optimized for space applications does not remove all of the problems common in the terrestrial Internet nor does it relieve the user from determining the optimal file size, data rate, and pass timing for reliable transmission over these networks.

ACKNOWLEDGEMENTS

The development of this simulator and the testing of the networking protocols were funded under NASA grant NAG5-7520 to New Mexico State University. We acknowledge the assistance of MITRE in providing the SCPS reference implementation software and in discussions related to making the software operational.

REFERENCES

[1] Consultative Committee for Space Data Systems, "Space Communications Protocol Specification (SCPS) - File Protocol (SCPS-FP)," CCSDS 717.0-R-3, September 1997.

[2] Horan, S., " A LabVIEW-Based Channel Error Simulator," Proceedings of the International Telemetering Conference, Las Vegas, NV, October 1999.

[3] Feighery, P., private communications, 1998, 1999.

[4] Horan, S. and Wang, R., "Design Of A Channel Error Simulator Using Virtual Instrument Techniques For The Initial Testing Of TCP/IP And SCPS Protocols," NMSU-ECE-99-002, New Mexico State University, Las Cruces, NM, April, 1999.

[5] Horan, S. and Wang, R., "Enhancement Of The NMSU Channel Error Simulator To Provide Unbalanced Forward And Return Transmission Rates," NMSU-ECE-99-003, New Mexico State University, Las Cruces, NM, April 1999.

[6] Durst, R., Feighery, P., Trowrs, E., "User's Manual for the SCPS Transport Protocol," MITRE, WN97W0000018, March 1997.

[7] Partirdge, C., Gigabit Networking, Addison Wesley, 1994.