

NASA/TM—1999-209272



# ACTS 118x Final Report High Speed TCP Interoperability Testing

David E. Brooks  
Sterling Software, Cleveland, Ohio

Craig Buffinton  
NetManage Inc., Cupertino, California

Dave R. Beering  
Infinite Global Infrastructures, LLC, Chicago, Illinois

Arun Welch  
Anzus Consulting, Orleans, Massachusetts

William D. Ivancic, Mike Zernic, and Douglas J. Hoder  
Glenn Research Center, Cleveland, Ohio

National Aeronautics and  
Space Administration

Glenn Research Center

---

July 1999

This report contains preliminary findings, subject to revision as analysis proceeds.

Trade names or manufacturers' names are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from

NASA Center for Aerospace Information  
7121 Standard Drive  
Hanover, MD 21076  
Price Code: A03

National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22100  
Price Code: A03

# Contents

<i>Contents</i>	i
<i>Executive Summary</i>	1
Background	1
Results	1
<i>Participants</i>	2
<i>Introduction</i>	4
Background	4
Goals	4
Conditions Which Affect TCP Efficiency	4
Why Should NASA Do This?	5
<i>Participants</i>	6
Government	6
Communication Industry	6
Computer Industry	6
Satellite Industry	7
Academia	7
<i>Connectivity Models</i>	7
<i>Network Configuration</i>	8
<i>Symmetric Interoperability Tests</i>	8
Configuration	8
Interoperability Matrix	9
Results	9
Solaris-to-Solaris	10
Solaris-to-OSF1	10
Solaris-to-NT4/5	10
Solaris-to-Irix6.4	10
OSF1-to-Solaris	10
OSF1-to-OSF1	10
NT4/5-to-NT4/5	10
Irix6.4-to-Solaris	11
Irix6.4-to-NT4/5	11
Irix6.4-to- Irix6.4	11
<i>Dual-Hop</i>	11

<b><i>Asymmetric Tests</i></b> _____	<b>11</b>
<b>Configuration</b> _____	<b>11</b>
<b>Results</b> _____	<b>12</b>
<b><i>FTP Software TCP Stack</i></b> _____	<b>12</b>
<b><i>TCP/IP applications</i></b> _____	<b>13</b>
<b><i>Where can things go wrong?</i></b> _____	<b>13</b>
<b>Alpha/Beta Systems</b> _____	<b>13</b>
<b>Link Problems</b> _____	<b>14</b>
<b><i>Testing Tools</i></b> _____	<b>14</b>
<b>TCP Analysis</b> _____	<b>14</b>
<b>Scripts</b> _____	<b>15</b>
<b><i>Where do we go from Here?</i></b> _____	<b>15</b>
<b><i>Conclusion</i></b> _____	<b>15</b>
<b><i>Appendix</i></b> _____	<b>17</b>
<b>Select Traces</b> _____	<b>17</b>
Solaris-to-Solaris (Sun-to-Sun) Asymmetric Tests _____	<b>17</b>
Solaris-to-OSF1 (SUN-to-Alpha) _____	<b>18</b>
NT5-to-Solaris (NT5-to-SUN) _____	<b>19</b>
IRIX-to-Solaris (SGI-to-SUN) Loopback _____	<b>20</b>
IRIX-to-Solaris (SGI-to-SUN) Loopback over ACTS _____	<b>21</b>
<b>Scripts and Machine Information</b> _____	<b>22</b>
Sun Solaris _____	<b>22</b>
Silicon Graphics IRIX _____	<b>25</b>
Compaq / Digital Equipment Alpha OSF1 _____	<b>28</b>
<b><i>References</i></b> _____	<b>33</b>

# **ACTS 118x Final Report High-Speed TCP Interoperability Testing**

David E. Brooks - Sterling Software  
Craig Buffinton - NetManage Inc.  
Dave R. Beering - Infinite Global Infrastructures, LLC  
Arun Welch - Anzus Consulting  
William D. Ivancic, Mike Zernic, and Douglas J. Hoder - NASA Glenn Research Center

## **Executive Summary**

### **Background**

The Space Program Office and the Communication Technology Division of the NASA Glenn Research Center have been working with the United States satellite communication industry over the past 15 years to develop advanced communication and networking technologies to improve commercial satellite communications. The Advanced Communication Technology Satellite (ACTS) and associated experiments are a primary example of this technology development. In general, these communication and networking technologies can be directly applied to NASA operations and NASA missions.

With the recent explosion of the Internet and the enormous business opportunities available to communication system providers, great interest has developed in improving the efficiency of data transfer using the Transmission Control Protocol (TCP) of the Internet Protocol (IP) suite. The satellite system providers are interested in solving TCP efficiency problems associated with long delays and error-prone links. Similarly, the terrestrial community is interested in solving TCP problems over high-bandwidth links. Whereas the wireless community is interested in improving TCP performance over bandwidth constrained, error-prone links.

NASA realized that solutions had already been proposed for most of the problems associated with efficient data transfer over large bandwidth-delay links (which include satellite links). The solutions are detailed in various Internet Engineering Task Force (IETF) Request for Comments (RFCs). Unfortunately, most of these solutions had not been tested at high-speed (155+ Mbps). Therefore, the NASA's ACTS experiments program initiated a series of TCP experiments to demonstrate scalability of TCP/IP and determine to what extent the protocol can be optimized over a 622 Mbps satellite link. These experiments were known as the 118i and 118j experiments. During the 118i and 118j experiments, NASA worked closely with SUN Microsystems and FORE Systems to improve the operating system, TCP stacks, and network interface cards and drivers. We were able to obtain instantaneous data throughput rates of greater than 520 Mbps using TCP over Asynchronous Transfer Mode (ATM) over a 622 Mbps Synchronous Optical Network (SONET) OC12 link. Following the success of these experiments and the successful government/industry collaboration, a new series of experiments, the 118x experiments, were developed. The objectives of the 118x experiments were:

- 1) to work in partnership with the government technology oriented labs, computer, telecommunication, and satellite industries to promote the development of interoperable, high-performance TCP/IP implementations across multiple computing / operating platforms;
- 2) to work with the satellite industry to answer outstanding questions regarding the use of standard protocols (TCP/IP and ATM) for the delivery of advanced data services, and for use in spacecraft architectures; and
- 3) to conduct a series of TCP/IP interoperability tests over OC12 ATM over a satellite network in a multi-vendor environment using ACTS.

### **Results**

We were able to achieve our goals of promoting and assisting in the development TCP implementations for high-speed, high-delay links while simultaneously answering the outstanding questions regarding the use of standard protocols for the delivery of advanced data services over satellites and for use in spacecraft architectures. However, given the time frame for the 118x experiments, April-September 1998, and given the complexity of these experiments, we were only able to complete a portion of the planned interoperability tests.

The maximum theoretical throughput for TCP over classical Asynchronous Transfer Mode (ATM) over a Synchronous Optical Network (SONET) is approximately 134 Mbps for a 155 Mbps link and 537 Mbps for a 622 Mbps link when taking ATM and SONET overhead into consideration.

We were able to test – to varying degrees – interoperability on four operating systems: SUN’s Solaris, Microsoft’s NT4 and NT5, Silicon Graphics IRIX, and Compaq’s OSF1. Tables 1 and 2 highlight the sustained average throughputs we were able to obtain for symmetric links. Table 1 shows result when operating in a local area network (LAN) environment with 10’s of milliseconds of delay. Table 2 shows results when operating over a satellite link with 570 milliseconds of delay. The variation of results can be due to TCP and operating system implementations, network interface cards, line drivers, and/or workstation processor speeds and resources. It should be noted that these results show the state-of-the-system as of November 1, 1998. One should expect that the systems will become more stable and eventually obtain near theoretical throughput within the next few years – particularly for OC3 links.

Operating System	Data Link Rate (Mbps)	Acknowledgment Link Rate (Mbps)	Average Data Throughput (Mbps)
OSF1	155	155	133
IRIX	622	155	500+
NT4/NT5	622	622	357
Solaris	622	622	400-500

**Table 1: Data Throughput for TCP over ATM over SONET in A LAN Environment**

Operating System	Data Link Rate (Mbps)	Acknowledgment Link Rate (Mbps)	Average Data Throughput (Mbps)
OSF1	155	155	120 <sup>1</sup>
IRIX	622	155	465 <sup>2</sup>
NT4/NT5	622	622	Unstable
Solaris	622	622	473

**Table 2: Data Throughput for TCP over ATM over SONET over a 570 msec Delay**

- 1 OSF1 transmitting data, Solaris Acknowledging
- 2 Solaris transmitting data, IRIX Acknowledging

We also ran some bulk data transfers over asymmetric links indicative of a relay satellite. Here, the acknowledgment return link bandwidth was constrained in order to determine the ratio of data transmission bandwidth to acknowledgment bandwidth required for bulk data transfers. These tests were only performed using the Solaris operating system. We were able to achieve data throughputs of 317 and 181 Mbps with acknowledgment links of 2.0 and 1.0 Mbps respectively. Thus showing a ratio of greater than 150:1 data throughput to acknowledgment for single user bulk data transfers.

## Participants

A consortium was established to perform the 118x experiments. Participants included government agencies, the communication, computer, and satellite industries and academia. Participation took place in a variety of forms including: engineering support, in-kind equipment loans, software support, communication links, and consulting.

The following is a brief list of participants:

### Government

- NASA Glenn Research Center; ACTS Project Office / Satellite Networks & Architectures Branch

- NASA Johnson Space Center; Space Operations Management Office (SOMO)
- NASA Jet Propulsion Laboratory; Space Communications Protocol Standard (SCPS) Group
- Lawrence Livermore National Laboratory National Transparent Optical Networking Consortium Lead
- Naval Research Laboratory

#### Communication Industry

- Sprint (Laboratory space, terrestrial network)
- Cisco Systems
- FORE Systems

#### Computer Industry

- Ampex Data Systems
- Sun Microsystems
- Compaq / Digital Equipment
- Microsoft
- Intel
- FTP Software (currently NetManage)

#### Satellite Industry

- Hughes Space & Communications
- Lockheed Martin Corporation
- Space Systems / LORAL
- Spectrum Astro

#### Academia

- Pittsburgh Supercomputing Center
- Ohio University

# Introduction

## Background

The Space Program Office and the Communication Technology Division of the NASA Glenn Research Center have been working with the United States satellite communication industry over the past 15 years to develop advanced communication and networking technologies to improve commercial satellite communications. The Advanced Communication Technology Satellite (ACTS) and associated experiments are a primary example of this technology development. In general, these communication and networking technologies can be directly applied to NASA operations and NASA missions.

With the recent explosion of the Internet and the enormous business opportunities available to communication system providers, great interest has developed in improving the efficiency of data transfer using the Transmission Control Protocol (TCP) of the Internet Protocol (IP) suite. The satellite system providers are interested in solving TCP efficiency problems associated with long delays and error-prone links. Similarly, the terrestrial community is interested in solving TCP problems over high-bandwidth links. Whereas the wireless community is interested in improving TCP performance over bandwidth constrained, error-prone links.

Even before the recent explosion of the Internet, NASA Glenn Research Center had been working with various users such as Boeing Aircraft [Ref. 1], Ohio Super Computer [Ref. 2] and the Aries Project [Ref. 3] to distribute large data sets over satellites. During this time, NASA heavily researched the current state of the TCP protocol and its limitations. As a result, NASA realized that solutions had already been proposed for most of the problems associated with efficient data transfer over large bandwidth-delay links (which include satellite links). The solutions are detailed in various Internet Engineering Task Force (IETF) Request for Comments (RFCs). Unfortunately, most of these solutions had not been tested at high-speed (155+ Mbps). Therefore, the NASA's ACTS experiments program initiated a series of TCP experiments to demonstrate scalability of TCP/IP and determine how far the protocol can be optimized over a 622 Mbps satellite link. These experiments were known as the 118i and 118j experiments. During the 118i and 118j experiments, NASA worked closely with SUN Microsystems and FORE Systems to improve the operating system, TCP stacks, and network interface cards and drivers. We were able to obtain instantaneous data throughput rates of greater than 520 Mbps using TCP over Asynchronous Transfer Mode (ATM) over a 622 Mbps Synchronous Optical Network (SONET) OC12 link. Following the success of these experiments and the successful government/industry collaboration, a new series of experiments, the 118x experiments, were developed. Participants included FORE, CISCO, SUN, Microsoft, Compaq, Lockheed/Martin, Hughes, NASA Glenn, Sprint, NetManage and AMPEX.

## Goals

The overall goals of the 118x experiments were:

- 1) to work in partnership with the government technology oriented labs, computer, telecommunication, and satellite industries to promote the development of interoperable, high-performance TCP/IP implementations across multiple computing / operating platforms;
- 2) to work with the satellite industry to answer outstanding questions regarding the use of standard protocols (TCP/IP and ATM) for the delivery of advanced data services, and for use in spacecraft architectures; and
- 3) to conduct a series of TCP/IP interoperability tests over OC12 ATM over a satellite network in a multi-vendor environment using ACTS.

## Conditions Which Affect TCP Efficiency

Three issues needed to be addressed when considering TCP performance: congestion, the bandwidth-delay product and bit errors.

Beginning around the fall of 1986, the Internet began showing signs of congestion collapse. To alleviate this problem, congestion control algorithms such as the slow start algorithm were adopted into the TCP standard implementations [Ref. 4 and 5]. These algorithms have been continually enhanced and provide an elegant solution to congestion control in an environment consisting of multi-faceted users operating on a variety of interconnected networks, the Internet. Many congestion control algorithms -- slow start in particular -- may result in inefficient



bandwidth utilization for end-to-end communications where a moderated amount of data is being transferred over a link exhibiting large bandwidth-delay characteristics.

Networks with bandwidth-delay products greater than 65535 bytes are referred to as long fat networks (LFNs). The 16 bit Window field in standard TCP results in this 65535 bytes Window limitation – approximately 60% percent of a T1, 1.544 Mbps, over a geosynchronous satellite link. Also, it is a possibility that packet sequence numbers could be used more than once in a LFN. Adding extensions to TCP for scaled windows and timestamps solves these problems. The specification that defines these extensions is found in RFC 1323, TCP Extensions for High Performance [Ref. 6].

Currently, any loss of TCP data is considered to be caused by congestion. As such, congestion control algorithms may be triggered for congestion when data experiences corruption. The TCP fast retransmit and fast recovery [Ref. 7], and the selective acknowledgment options [Ref. 8] improve TCP performance in many situations where congestion and/or corruption may occur.

### **Why Should NASA Do This?**

Throughout the 118 suite of experiments the following observations and questions continually surfaced:

The extensions to TCP are already standardized in RFC 1323 and RFC 2018. Also, many vendors are implementing or planning to implement these extensions in their operating systems. Why aren't the operating system and computer vendors performing interoperability testing?

It is true that the extensions to TCP have already been standardized and have been or will soon be implemented in most operating systems such as SUN's Solaris 7.0 and Microsoft's Windows '98 and NT5. In addition, these protocols have gone through interoperability evaluation at low and moderate rates and bandwidth-delays -- that typical of today's Internet and near term projections. It is in the best interest of the operating system and computer hardware vendors to ensure that the protocols interoperate

Why should the Government in general and specifically NASA expend resources to perform TCP interoperability testing?

There simply is not a large enough user or application base for vendors to justify expending additional resources to test interoperability at extremely high bandwidth-delay products such as those achievable with the ACTS OC12 link – 622 Mbps at 570 milliseconds of delay. Few users require or can afford to transmit large amounts of data at such rates over satellite links even if commercially available today. Two users that have these requirements and the resources to pay for the link capacity are NASA and the Department of Defense (DoD). Both require large database transfers to and/or from remote locations that have no terrestrial infrastructure. They must use satellites. There is no alternative. This is one reason that DoD assisted funding for the ACTS High-Data-Rate (HDR) Terminal. In addition, by understanding the TCP protocol and associated implementation requirements (hardware, software, network interface cards, and firmware) the government will know where it can best apply commercial TCP and when other commercial transport protocols or customized protocols will be necessary.

Why use ACTS? Can't TCP interoperability testing over long delay links (GEO) could be performed without a satellite?

Currently TCP interoperability can be performed at 155 Mbps using a channel simulator such as the Adtech SX14 or an impairment module such as is available from HP in their ATM Broadband Series Test System and may be available in other manufacturers broadband test systems. These channel simulators and impairment modules allow complete control over the link characteristics including: data-rate, error-rate and error-distribution. However, as best as could be determined, there are no commercially available delay/impairment modules or test systems that operate at 622 Mbps or greater. One primary goal of ACTS expectation is to investigate satellite terrestrial setup, standards, and protocols of which in this case offer space-based validation segment albeit as a 622 Mbps "delay module." ACTS is the only known 622 Mbps delay module available. The 622 interface is SONET and coded using a Reed-Solomon code. Therefore,

*the system tends to run error free or not at all.* This extremely important to note as this implies that only the portion of the TCP interoperability code tested during the 118x experiments is the large window portion. If the link is running properly, there should be no errors and no congestion. Therefore, fast-retransmit/fast-recovery and selective acknowledgment (SACK) should not be exercised for a perfectly tuned system (advertised receive window is set to the bandwidth-delay product of the link). Another primary goal of ACTS expectation is to assist the governments transition to commercial communications. Therefore any investigations, understandings, and contributions that can be validated using a space platform rather than lab simulations should accelerate the standardization of commercial protocols for space architecture.

## **Participants**

A consortium was established to perform the 118x experiments. Participants included government agencies, the communication, computer, and satellite industries and academia. Participation took place in a variety of forms including: engineering support, in-kind equipment loans, software support, communication links, and consulting.

The following is a brief list of the participants and some of their contributions:

### **Government**

The ACTS Project Office and the Satellite Networks and Architectures Branch at the NASA Glenn Research Center were active participants. The ACTS Project Office provided the satellite, ground terminals and engineering support while the Satellite Networks and Architectures Branch provided consulting regarding TCP implementations.

The Space Operations Management Office (SOMO) at the NASA Johnson Space Center participated as an observer as well as providing some direction. SOMO is interested in the utilizing commercial protocols for NASA operations and missions wherever practical in order to reduce overall networking costs.

The NASA Jet Propulsion Laboratory; Space Communications Protocol Standard (SCPS) Group provided the SCPS software that we were hoping to test over this high-speed, high-delay link. Unfortunately, time did not permit completion of these tests. We hope to carry this work on in the future.

The Lawrence Livermore National Laboratory functioning as the National Transparent Optical Networking (NTON) Consortium Lead provided a high speed Dense Wave Division Multiplex (DWDM) optical connection from the ACTS ground terminal located at Lawrence Livermore National Laboratory (LLNL) to the Sprint Advanced Technology Laboratory in Burlingame, California. In addition, LLNL helped establish and maintain the ground terminal at its site.

Naval Research Laboratory (NRL) participated as an observer in 118x. Using the knowledge obtained from the 118x experiments, NRL performed a duplex-link, ship-to-shore experiment in November 1998 (Ref. ACTS Experiment 149). This experiment demonstrated TCP and other communication protocols and applications operation over a bi-directional 45 Mbps satellite link and set a new record for ship-to-shore communications.

### **Communication Industry**

Sprint provided terrestrial network connections as well as use of space and equipment in their Advance Technology Laboratory in Burlingame, California. In addition, Sprint provided engineering support at their facility.

Both Cisco Systems and FORE Systems provided ATM Switches in addition to engineering support and hardware/driver upgrades.

### **Computer Industry**

Ampex Data Systems provided high-capacity, high-speed tape drives. We had planned to enhance the 360 Mbps aggregate tape-to-tape transfers originally demonstrated at SuperComputing '97 (ACTS experiment #118j).

Unfortunately, we were unable to complete this within the allotted time.

Sun Microsystems, Compaq / Digital Equipment and Intel provided in-kind equipment and engineering support. Both Compaq and Sun also provided operating system and driver upgrades.

Microsoft provided its NT4 and NT5 operating system running on the Intel Pentium II Development Systems as well as engineering support.

FTP Software (currently NetManage) tested their TCP software stack running over a 155 Mbps emulated link using IP over ATM (RFC 1577) running on SONET over ATM OC3. The tests were performed at Lockheed Martin facilities. We were unable to test over ACTS due to time constraints although all parties desire to continue this work.

## Satellite Industry

Hughes Space & Communications, Lockheed Martin Corporation, Space Systems / LORAL and Spectrum Astro generally participated as observers and provided insight into their desires and needs, as well as this industry as a whole. In addition, Lockheed worked closely with FTP Software.

## Academia

Pittsburgh Supercomputing Center provided support with the NetBSD operating system. Of particular interest was testing of Pittsburgh Supercomputing's TCP autotuning algorithm.

Ohio University provided support TCP trace analysis and TCP testing tool development.

## Connectivity Models

There are three types of connectivity models we planned to work with in I18x: the communication satellite model, the relay satellite model, and the direct broadcast satellite model [Figures 1, 2 and 3 respectively].

For the majority of our testing, we used the communications satellite model. The communications satellite was modeled with symmetrical, balanced links between ground terminals representing a fully meshed satellite network or a trunking system. A communication satellite network may also exhibit moderate asymmetry -- particularly for hub-spoke star networks.

The relay satellite model has a highly asymmetric link. The return channel bandwidth is only a small percentage of the data-path bandwidth. This model was put together at the request of NASA's Space Operations Management Office to investigate use of TCP for bulk data transfer over the Tracking and Data Relay Satellite System (TDRSS).

The direct broadcast satellite (DBS) model represents a hybrid satellite/terrestrial network where data is distributed using a high-bandwidth satellite channel and acknowledged using a low-bandwidth terrestrial link. This model closely represents a commercial system such as the Hughes Direct PC product. During I18x experiments, the DBS model was not exercised due to time limitations.

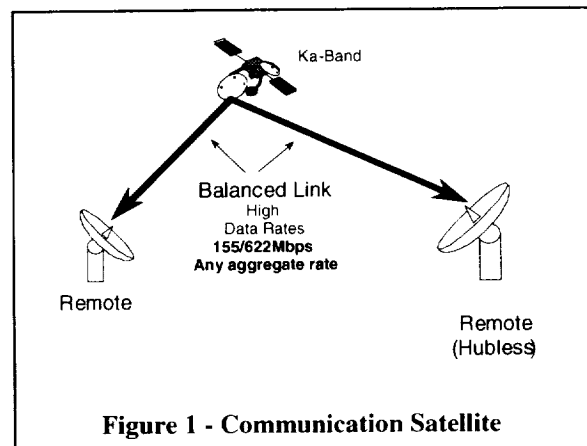


Figure 1 - Communication Satellite

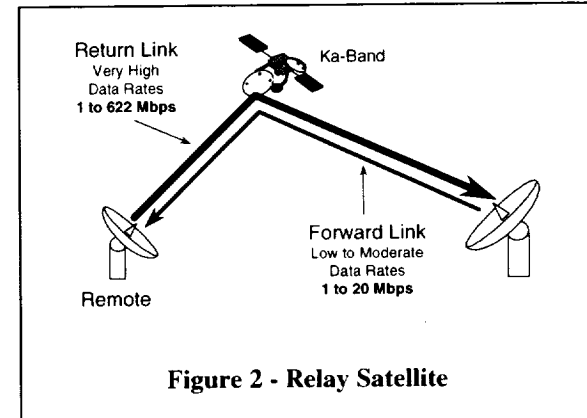


Figure 2 - Relay Satellite

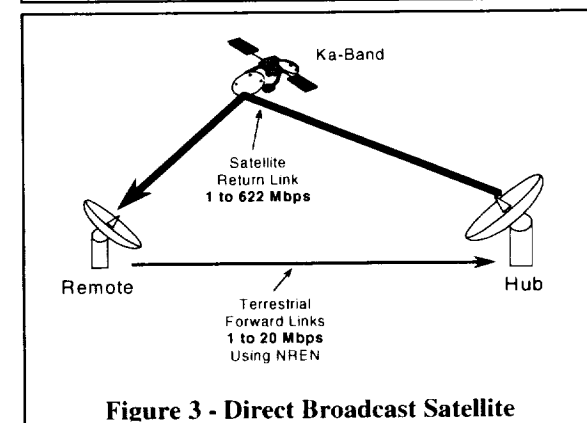
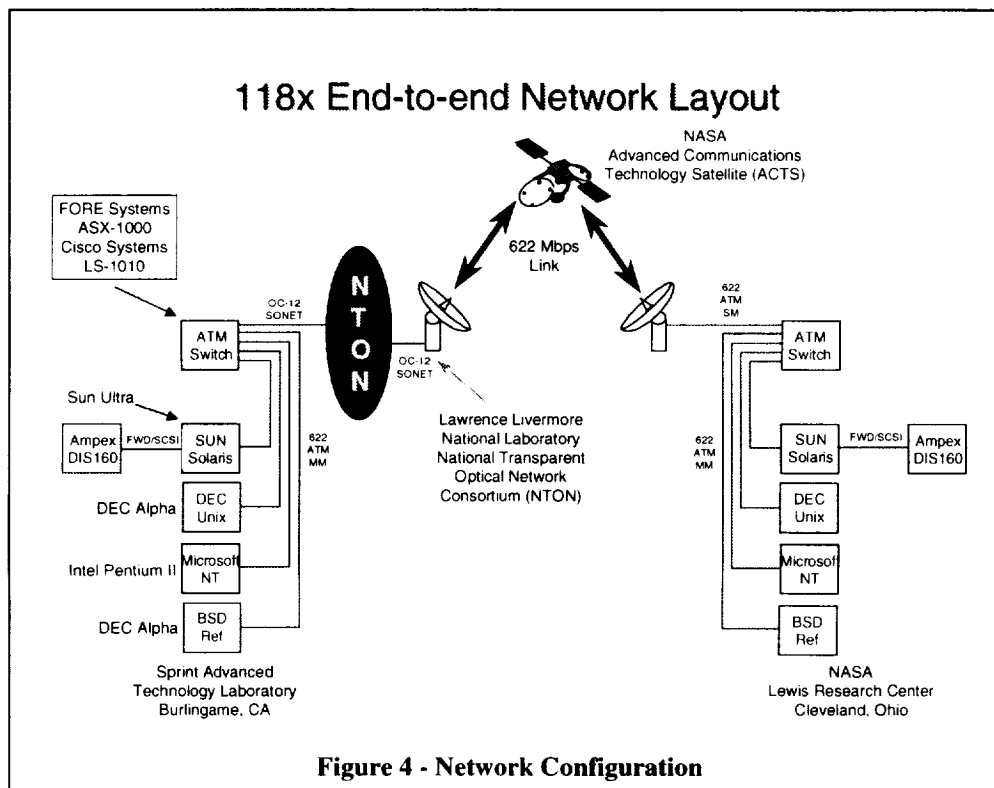


Figure 3 - Direct Broadcast Satellite

## Network Configuration

Figure 4 shows the overall network configuration for the 118x experiments. Two High-Data-Rate terminals (HDR) and the Advance Communication Technology Satellite (ACTS) provided the satellite link with an effective bi-directional data throughput of 622 Mbps. The interfaces to the ACTS ground terminals is 622 Mbps using the SONET physical link protocol. The HDR located at NASA's Glenn Research Center (GRC) in Cleveland, Ohio was connected directly to a FORE ASX-1000 ATM switch. The workstations and personal computers were also connected directly to this switch. No routers were used for these experiments. The second HDR was located at Lawrence Livermore National Laboratory. The HDR was connected to an ATM switch at Sprint's Advanced Technology Laboratory in Burlingame, California through the National Transparent Optical Network Consortium (NTONC) using Dense Wave Division Multiplexing (DWDM) technologies. At Sprint's Advance Technology Laboratory, there was a mirror image of the GRC site. Again, no routers were used at the Sprint site.

Workstations, ground stations, and switches in this network were also connected via the Internet in order to configure, control, and monitor all workstations remotely. This was critical in order to operate and maintain the network. The ACTS satellite has many experiments scheduled, therefore, the satellite link was only available during scheduled experiment times.



## Symmetric Interoperability Tests

### Configuration

For the symmetric tests, the network configuration represented in Figure 4 was setup to represent a duplex trunking satellite network as shown in Figure 1. These tests were designed to determine TCP interoperability. It is important to note that only the large-window, time-stamp, and protect against sequence number wrap-around portions of the TCP stack should be exercised during these tests as the system should have been operating error-free and congestion-free.

## Interoperability Matrix

Our goals for these interoperability tests was to complete the interoperability matrix shown in Table 3 while simultaneously improving upon the performance numbers as problems were discovered and corrected. Tests were performed for both a local-area network (LAN) and a geostationary orbiting satellite (GEO) with a 540 millisecond delay. The tests that were fully completed and documented are indicated in the matrix. The circle indicates completed LAN tests whereas the square indicates completed satellite tests.

*Note! This matrix is dynamic as there will be continued improvement in the performance numbers over time as problems are uncovered and corrected.*

		Receive						
		Solaris	OSF1	NT4/5	Irix6.4	Win98	NetBSD	Linux
Transmit	Solaris	Ⓚ	Ⓛ	II	Ⓚ	II	II	II
	OSF1	Ⓛ	Ⓚ	II	II	II	II	II
	NT4/5	II	II	Ⓚ	II	II	II	II
	Irix6.4	Ⓛ	II	Ⓛ	I	II	II	II
	Win98	II	II	II	II	I	II	II
	NetBSD	II	II	II	II	II	I	II
	Linux	II	II	II	II	II	II	I

LAN
  GEO (540 msec Delay)

**Table 3 Interoperability Matrix**

Many factors work together to produce the performance results of a given TCP transmit/receiver pair. These factors include software, TCP stacks, network interface cards (NIC), NIC drivers, workstation memory, processor speeds, and proper operating system configuration settings. Much of the code and hardware used in these tests was Alpha and Beta version. Therefore, much of the code was not fully documented and often contained bugs that needed to be corrected.

## Results

We were able to achieve our goals of promoting and assisting in the development TCP implementations for high-speed, high-delay links while simultaneously answering the outstanding questions regarding the use of standard protocols for the delivery of advanced data services over satellites and for use in spacecraft architectures. However, given the time-frame for the 118x experiments, April-September 1998, and given the complexity of these experiments, we were only able to complete a portion of the planned interoperability tests.

The maximum theoretical throughput for TCP over classical Asynchronous Transfer Mode (ATM) over a Synchronous Optical Network (SONET) is approximately 134 Mbps for a 155 Mbps link and 537 Mbps for a 622 Mbps link when taking ATM and SONET overhead into consideration.

The following are the results we were able to obtain in the time allocated to the 118x experiments for interoperability. The details and all raw data is available from the 118x Web Site, <http://mrpink.lerc.nasa.gov/118x><sup>1</sup>. All tests were run with optimal buffer size allocations and used classical IP over ATM, the IP packet set to 9180 bytes.

The results are heavily dependent on the hardware, software and operating system used. Those details are available in the raw data available at the Web site. Example of the details captured during each experiment run for given operating systems, switches and ground terminal are provided in the Appendix.

The following notation used to present the results is Transmitter-to-Receiver where Transmitter is the data source (server) and Receiver is the data sink (client).

<sup>1</sup> As of this printing, the site is password protected as the work was preliminary at the time of the experiments. The password protection may be removed by the time this report is released. If not, passwords can be obtained by contacting the ACTS Experiments Program Office.

## **Solaris-to-Solaris**

The Sun machines used in these tests were Sun Ultra 2 dual processor machines with 512 Mbytes of main memory and over 400 Mbytes of virtual memory. The operating system used was Solaris 5.7 with software patches to allow for SACK operation. We were able to obtain an average throughput of 482 Mbps for a LAN connection with 1 Mbits delay-bandwidth buffer and 473 Mbps for a 540 msec delay (GEO, geostationary satellite orbit) using a DELAY-BANDWIDTH BUFFER of 32,768,000 bits. To obtain the GEO rates, one of the two server processors was operating at 97 percent capacity. We were also able to reach instantaneous throughput rates of up to 520 Mbps. However, the workstations we had used the Sun SBUS ATM cards. These cards could not sustain the 520 Mbps rate. We expect that if we replace the SBUS cards with cards that used the PCI architecture that we would have obtained near line rates.

## **Solaris-to-OSF1**

The machines used in these test were the Sun Ultra 2 and the Digital/Compaq Alpha 600au. The Alpha used the OSF1 V4.0 version 878 operating system. We were able to obtain an average throughput of 133 Mbps for a LAN connection with 1 Mbits delay-bandwidth buffer in both the server and client machines and 104 Mbps for GEO delays using a DELAY-BANDWIDTH BUFFER of 8,192,000 bits. To obtain the GEO rates, one of the two server processors was operating at 15 percent capacity. We only had OC3 network interface cards available on the Alpha; therefore all testing was performed at 155 Mbps.

## **Solaris-to-NT4/5**

The machines used in these tests were the Sun Ultra 2 and the INTEL development platform and Fore Systems 1xPCA200E and 1xPCA622HE ATM cards. The INTEL development platform used quad 400 MHz processors and had 2 Gbytes of system ram. The drivers for the ATM cards were alpha/beta status and the TCP/IP stack was beta. The operating system was a custom version combining elements of Microsoft NT version 4 and 5. We were able to obtain an average throughput of 263 Mbps for a LAN connection with 1 Mbits delay-bandwidth buffer in the server and 9Kbits of delay-bandwidth buffer in the client. Instabilities in the TCP client stack when operating over long delays were not resolved by the time 118x completed.

## **Solaris-to-Irix6.4**

The machines used in these test were the Sun Ultra 2 and the Silicon Graphics ONYX2 with dual 180 MHz IP27 Processors, 256 Mbytes of main memory and 32 Kbytes of instruction cache. The operating system was IRIX64 tng 6.4 02121744 IP27. We were able to obtain an average throughput of 481 Mbps for a LAN connection with 1 Mbits delay-bandwidth buffer in both the server and client machines and 465 Mbps for GEO delays using a delay-bandwidth buffer of 32768000 bits. To obtain the GEO rates, one of the two server processors was operating at 82 percent capacity.

## **OSF1-to-Solaris**

The machines used in these tests were the Digital/Compaq Alpha 600au and Sun Ultra 2. We were able to obtain an average throughput of 133 Mbps for a LAN connection with 8 Mbits delay-bandwidth buffer in both the server and client machines and 120 Mbps for GEO delays using a delay-bandwidth buffer of 8,192,000 bits. To obtain the GEO rates, Alpha (server) processor was operating at 54 percent capacity. We only had OC3 network interface cards available on the Alpha; therefore all testing was performed at 155 Mbps.

## **OSF1-to-OSF1**

The machines used in these tests were the Digital/Compaq Alpha. We were able to obtain an average throughput of 133 Mbps for a LAN connection with 1 Mbits delay-bandwidth buffer in both the server and client machines. Due to time 118x time limitations and minimal availability of ACTS, GEO tests were not completed.

## **NT4/5-to-NT4/5**

The machines used in these tests were the INTEL development platforms. The operating system was a custom version combining elements of Microsoft NT version 4 and 5. LAN testing was performed at Microsoft. Our understanding is TCP rates at near line-rate (537 Mbps) were achieved in the LAN environment. However, we do

not have tcpdump files to verify this. The stack was extremely unstable (machine crashes and unusual protocol behaviors) at 622 Mbps rates over ACTS. For various runs we obtained performance ranging in the low Mbps up to approximately 200 Mbps.

### Irix6.4-to-Solaris

The machines used in these test were the Silicon Graphics ONYX2 and the Sun Ultra 2. We were able to obtain an average throughput of 437 Mbps for a LAN and 257 Mbps for GEO delays using a delay-bandwidth buffer of 32768000 bits. To obtain the GEO rates, the ONYX2 server processor was operating at 81 percent capacity.

### Irix6.4-to-NT4/5

The machines used in these tests were the Silicon Graphics ONYX2 and the INTEL development platform. LAN baselines were not run due to time limitations. We were able to obtain an near theoretical line throughput for GEO delays using a delay-bandwidth buffer of 32768000 bits. However, these rates were not sustainable and caused the NT4/5 system to crash.

### Irix6.4-to- Irix6.4

We where unable to perform these test as we currently only have one SGI Onyx with one OC12 card. We need to get a second machine and additional OC12 cards to complete this testing.

## Dual-Hop

During the early stages of the 118x experiments, we experience fiber outages in the NTON. While these were being corrected the network was put into a loopback configuration as shown in Figure 5. This created a dual hop situation effectively doubling end-to-end delay and thereby doubling the bandwidth-delay product. Thus we were able to demonstrate high-speed TCP operation in a dual-hop environment. As far as the TCP stack is concerned, demonstrating TCP performance over this bandwidth-delay product (622 Mbps x 1080 msec) effectively demonstrates a Gbps transmission over a single hop geostationary satellite relative to the protocol.

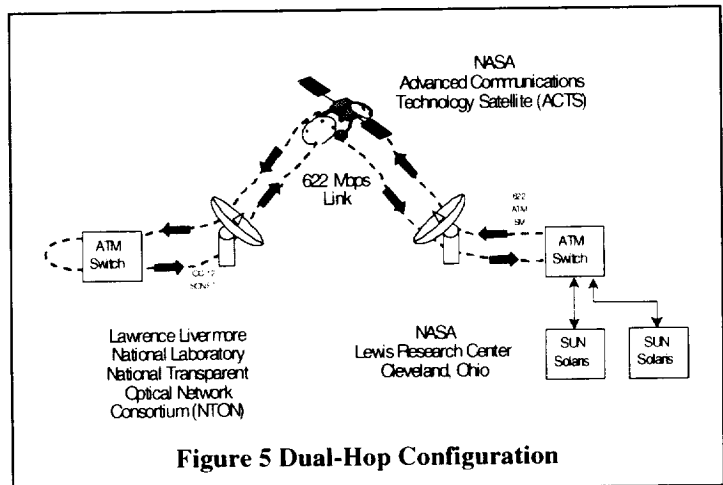


Figure 5 Dual-Hop Configuration

The link in each direction supports approximately 539 Mbps of user data throughput, after overhead is subtracted (ATM, SONET, TCP/IP). Given that the acknowledgements occupy approximately 1/30 as much bandwidth as the forward data stream, we need to size our TCP/IP windows such that the data transfer bandwidth is less than about 97% of the available bandwidth. For our 539 Mbps forward link, the BW-delay product is 72,293,375 Bytes, given a delay of 1.073 seconds. We allowed four percent of the link to be allocated for acknowledgements. Therefore, we set our TCP/IP socket buffer to 69,401,640 bits. With these setting we were able to achieve an average TCP throughput of approximately 344.16 Mbps with server CPU usage at 83 percent.

## Asymmetric Tests

### Configuration

For the asymmetric tests, the network configuration represented in Figure 4 was setup to represent NASA's Tracking and Data Relay Satellite System (TDRSS) or any other relay satellite network [Fig. 2]. These tests were performed to empirically determine "For a given forward satellite channel capacity, how much return capacity can be support using TCP/IP?" In other words, what is the ratio of data path bandwidth to acknowledgment path bandwidth for bulk data transfers. For these experiments, the Sun workstations were used. The operating system was OS version 5.7 with Kernel version SunOS Release 5.7 version kepoon.

For NASA's TDRSS, White Sands, New Mexico is the center of reference. A remote spacecraft sending data to White Sands is "returning" data, and the link is called a Return Link. Conversely, if a link is mapped from White out to the remote spacecraft, they call it a Forward Link. For the 118x tests, we established a Forward Link on the 118x test network operating at 1 Megabit per second. That link speed was enforced at the Sun workstation's ATM interface by a setting the rate queue, and also in the FORE Systems ATM switch by implementing a Constant Bit Rate traffic contract. The Return Link was not constrained, so theoretically it could go up as high as 622 Megabits per second. The Sun workstations set up to support TCP congestion windows as large as 32 Megabytes. When we ran the tests between the two workstations without the traffic contracts enforced, the workstations were able to support sustained data rates of approximately 500 Megabits per second. We then ran the tests with the traffic contract enforced for the forward channel at one Megabit per second, and then again at two Megabits per second.

## Results

The tests were run with a 1 Megabit traffic contract, a 2 Megabit traffic contract, and a 3 Megabit traffic contract. For the 1 and 2 Mbps Forward Link Cases we were able to obtain a sustained return link data rates of 177.14 Mbps and 309.45 Mbps as measured by  $ttcp^2$ . The  $ttcp$  measurements are average values and included the slow-start throughput in the calculation whereas the PDUs per second measurement is a windowed value. Using the PDUs per second one can calculating the throughput as:

PDUs per second x 9180 Bytes per PDU x 8 bits per Byte.

UPC Mbps	Peak PDU	Sustained PDU (PDUs/sec)@1 sec poll	Avg.TPut Mbps	Traced?	atmspeed Reported
1	3476	2885	177.14	Y	1 Mbps
1.5	4343 (?)	2885	182.53	Y	1 Mbps
2.0	5911	5776	309.45	Y	2 Mbps
2.5	N/A	N/A		N/A	N/A
3.0	6987	6800	357.60	Y	3 Mbps

**Table 4 Asymmetric Link Throughput Results**

From the PDU data, for the 1 and 2 Mbps Forward Link Cases we were able to obtain a sustained return link data rates of 211 Mbps and 422 Mbps. Thus the ratio of return-to-forward link is approximately 200:1 using the PDU data. The throughput for the 3 Mbps Forward Link Case is no longer linear as the maximum return link throughput had been reached. We were unable to obtain good data for Constant Bit Rate traffic contracts of 1.5 Mbps and 2.5 Mbps because the FORE ATM switches setup the traffic contacts in integer steps. Setting up a contract for 1.5 Mbps resulted in a contract for 1 Mbps.

## FTP Software TCP Stack

FTP Software (currently NetManage) wrote a third party TCP stack to run on general workstations. Time did not allow for this stack to be tested over ACTS. However it was tested in a laboratory environment at Lockheed Martin using an Adtech SX/14 channel simulator to generate a 540 millisecond delay. FTP OnNet Kernel 4.0 software was running on a Windows 95 OSR2 using a custom built Empac P2 with 400Mhz CPU; 128 MB RAM; and a PCA-200E Fore ATM card with version 5.0.0.25096 drivers. The second machine was a Sun Ultra 60 UPA/PCI (Ultra SPARC-II 296 MHz) running Solaris 2.7 beta and using a Fore ATM card. TTCP, a standard TCP test tool, was used for memory to memory tests. Ftp was used for file transfer tests. The following application buffers were used: TTCP 256KB; NM ftp server 256KB; NM ftp client 16KB; and MS ftp client 8KB.

Samples of some of the results are shown in Table 5. CPU utilization was near full capacity for most of these tests. For memory-to-memory tests, the results are quite very good - particularly for low delay. Additional work needs to be done regarding FTP transfers. Many factors working together result in less than desirable throughput for FTP.

<sup>2</sup> Ttcp measurements often vary by 2.4% from the actual throughput as many  $ttcp$  coding implementations considers a kilobit to be 1024 bits rather than 1000 bits.



For file transfers from the PC to the Sun, the Sun FTP server and client permitted a maximum receive window size of 65535, thus slowing throughput with delay dramatically. For file transfers with delay from the Sun to the PC retransmissions slowed the throughput dramatically. Often, the application buffers could not be increased to 256K for FTP resulting in increased workload for the CPU. In addition, the DISK I/O limited performance. We also noticed some problems with the Fore ATM drivers when transmission was over a long delay.

Description	Delay (msec)	Receive Window (Bytes)	Measured Throughput (kBytes/sec)	Measured Throughput (Mbits/sec)	Theoretical Throughput (Mbits/sec)	Throughput % of Theory
<b>Memory-to-Memory Transfers</b>						
PC-to-Sun memory-to-memory	540	8,384,768	11,419	93.544	124.219	75.31%
PC-to-Sun memory-to-memory	0	8,384,768	14,936	122.356	134.500	90.97%
Sun-to-PC memory-to-memory	540	8,422,686	11,191	91.677	124.781	73.47%
Sun-to-PC memory-to-memory	0	8,422,686	16,122	132.071	134.500	98.19%
<b>PC-to-Sun File Transfers</b>						
NM server / Sun Client	540	65,535	113	0.926	0.971	95.35%
NM server / Sun Client	0	65,535	8,094	66.306	134.500	49.30%
NM client / Sun server	0	65,535	8,396	68.780	134.500	51.14%
MS client / Sun server	0	65,535	6,393	52.371	134.500	38.94%
<b>Sun-to-PC File Transfers</b>						
NM server / Sun Client	540	8,422,686	1,259	10.314	124.781	8.27%
Sun server / NM Client	540	8,422,686	1,245	10.199	124.781	8.17%
NM server / Sun Client	0	8,422,686	7,559	61.923	134.500	46.04%
Sun server / NM Client	0	8,422,686	4,797	39.297	134.500	29.22%
Sun server / MS Client	0	8,422,686	4,163	34.103	134.500	25.36%
Note! kBytes = 1024 bytes; Mbits = 1,000,000 bits						
<b>Table 5 FTP Software File Transfer Results</b>						

In delay tests the Fore driver occasionally returned pending status on a series of sends. This was observed for ranges of 8 to 22 consecutive sends in internal stack traces. Receive traces showed these pending status packets received out of order. The resulting duplicate acks for the last in order segment received caused the sender to enter the fast retransmit / fast recovery state. This slowed transfers dramatically when in fact no packets were actually lost, they were only misordered. It is valid for the Fore driver to return pending status and OnNet Kernel 4.0 NDIS interface is prepared to handle this. NDIS specs do not indicate that the driver is required to send pending status packets in order. In order to force ordering of packets, the FTP software was modified to wait for a pending send to complete before sending more output. This work around was fine for these particular tests however the same code ran dramatically slower when tested with a 100 Mb ethernet card that consistently returns pending status.

## TCP/IP applications

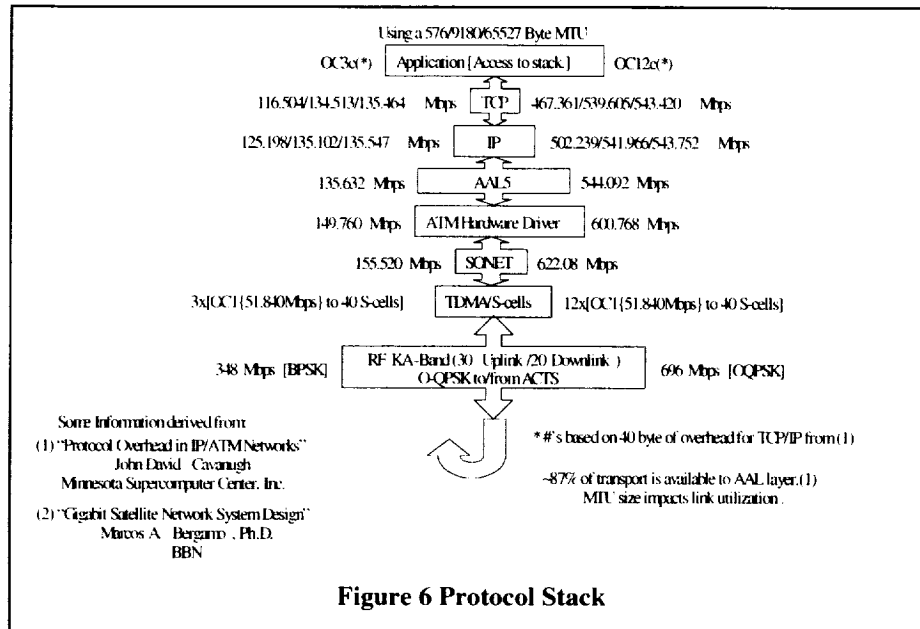
We planned to execute tape-to-tape transfers during 118x using the Ampex DIS160 high-capacity tape drives. However, time did not permit us to complete these tests.

## Where can things go wrong?

### Alpha/Beta Systems

118x was a research project. The goal was to help get new systems running and interoperating. As such, we were working with "alpha" and "beta" systems: TCP code, hardware, operating systems and line drivers. Figure 6 shows the protocol stack and the interaction of various hardware and software components. The TCP/IP software is part of the operating system kernel. Large bandwidth-delay products result in TCP utilizing large portions of system

memory for buffering. Thus, good management of memory is crucial for proper operation of TCP in this environment. The TCP packets are encapsulated into an ATM Common Part Convergence Sub-layer (CPCS) Payload Data Unit (PDU) for ATM Application Layer 5 (AAL5). The AAL5 PDU is then separated into 53 byte ATM cells. The ATM cells are mapped into a SONET OC3 or OC12 frame and sent on to the ground terminal. The line driver performs the TCP-to-ATM mapping. This is where the network interface card (NIC) interacts with the workstation operating system. The ATM-to-SONET is generally performed in dedicated hardware on the NIC.



When the system did not operate as expected and the satellite link was operating properly, the problem could be in anywhere in the system. Rarely could the problem be immediately attributed to one portion of the system because of the interactions of all the pieces.

## Link Problems

ACTS is an experimental satellite. In addition, the HDR terminals were design for a two-year operational life. Thus, we had a number of problems with the ACTS/HDR reliability. We also were a non-paying guest on NTON. NTON is also an experimental system. Thus, we had a number of problems with the physical links that caused delays in performing these tests.

- During the early portions of the testing, we lost NTON due to a back-hoe cut. Latter on in the experiment for approximately 2 weeks when we lost the physical connection through NTON. Connection could be made from LeRC to LLNL but not from LLNL to Burlingame. We performed the dual-hop tests during this time.
- We had poor ground terminal operation at LLNL and attributed it initially to a modem failure. The modem was sent out for repair and testing resulting in some down time.
- We had numerous SONET pointer slips causing SONET errors at the physical layer (bit interleaved parity - BIPS and far end block errors - FEBEs). The problem was corrected by replacing a SONET board at LeRC.
- Many times we had trouble getting the OC12 configuration operating properly over ACTS.

## Testing Tools

### TCP Analysis

During the 118x experiments, it became evident that better tools were needed to debug and analyze the TCP stack -- particularly with multiple vendors involved. Many tools are publicly available to perform TCP capture and analysis. Those were chose to use where: *ttcp*, *tcpdump*, *tcptrace*, *atmsnoop* and *xplot*. We had to develop special testing

techniques and modify some of these to operated in the high-speed environment we were operating in. The details of this work are available in a paper entitled, "High-Speed TCP Testing" [Ref. 9].

## Scripts

In order to obtain all of the pertinent information from each TCP run a number of script files were generated. Information that we considered pertinent included: general workstation statistics, SONET ATM layer statistics from the switches at both the HDRs' and the workstations' ATM ports, TCP/IP statistics, TCP/IP settings, workstation driver information, type of ATM interface, and special workstation settings. The script files were combined into a single script that starts a bulk transfer between two workstations with the entire run recorded under a common directory on the initiating workstation. After the test, the receiver information is automatically copied to the transmitter side. Examples of the script output are presented in the Appendix.

## Where do we go from Here?

### A) Investigate Other TCP Options

The ACTS HDRs use Reed-Solomon forward-error-correction (FEC) coding. This results in an error-free link. In addition, we only had a single TCP connection running over the link at any time -- no congestion. Therefore, only the large-windows portion of the TCP stacks was exercised. The fast-retransmit, fast-recovery and selective acknowledgement (SACK) options should not have been. If these option were exercised, either something was wrong with the link or the TCP stack (including the hardware and drivers). The fast-retransmit, fast-recovery and SACK options need to be exercised.

### B) Perform LAN Benchmarks

The 118x TCP interoperability experiments were run over ACTS with one set of equipment located in Cleveland, Ohio and a second set in Burlingame, California. Our desire was to originally have all the equipment shipped to GRC first for stringent baselining and LAN testing. This did not happen. Therefore, we did not have baseline information of the various stacks in a LAN environment. The TCP stacks need to be baselined and include baselining for the fast-retransmit, fast-recovery and SACK options as well as for large windows. This can be accomplished using at OC3 rates using an HP BSTS and/or an Adtech SX/14 to impair the channel. This cannot be accomplished at OC 12 rates as we are not aware of any channel impairment equipment that operates at OC12 rates.

### C) Build Upon these Findings and Experiences

A series of experiments is needed to evaluate various commercial and research TCP implementations in a controlled environment and determine the following:

- 1) Does the TCP implementation function properly relative to the specifications found in RFCs 1323 and 2018 (TCP Enhancements Compliance)?
- 2) If so, at what bandwidth-delay can the stack operate before breaking down and what is the cause of the breakdown (TCP Breakdown)?
- 3) How well does this TCP stack interoperate with other TCP stacks (TCP Interoperability)?

TCP Enhancements compliance testing is necessary to ensure we are using fully operational TCP implementations when evaluating TCP interoperability. Determining TCP breakdown will allow us to know to what bandwidth-delay we can properly evaluated TCP interoperability for various vendor's TCP implementations. In addition, we will be able to provide this information to the commercial vendors thereby enabling them to improve their products.

## Conclusion

We believe the overall goals of the 118x experiments were met. We were able to establish a partnership with the computer, telecommunication, and satellite industries to promote the development of interoperable, high-performance TCP/IP implementations across multiple computing / operating platforms. We also were able to answer many outstanding questions regarding the use of standard protocols (TCP/IP and ATM) for the delivery of advanced data services, and for use in spacecraft architectures. Of particular importance was the validation of TCP implementation, allowing hundreds of Mbps throughput in a symmetric scenario as well as establishing a return-to-forward link ratio of 200:1 in a asymmetric scenario. However, there is still much work that was not completed. In particular, the TCP stacks need to be baselined and tested for interoperability with all aspects of TCP exercised

including: large-windows, fast-retransmit, fast-recovery and SACK. Follow-on experiment activities are being formulated to address these areas.

# Appendix

## Select Traces

### Solaris-to-Solaris (Sun-to-Sun) Asymmetric Tests

Figure 7 shows the trace for the asymmetric configuration for that of a relay satellite such as TDRSS [Fig. 2]. Notice the exponential slow start up to the congestion window, "cwnd". Here the transmitter has completely filled the receivers buffer. At that point the receiver can only acknowledge data as fast as the inbound (ack) link will allow. For this particular test the inbound acknowledgements were restricted to a 1 Mbps link.

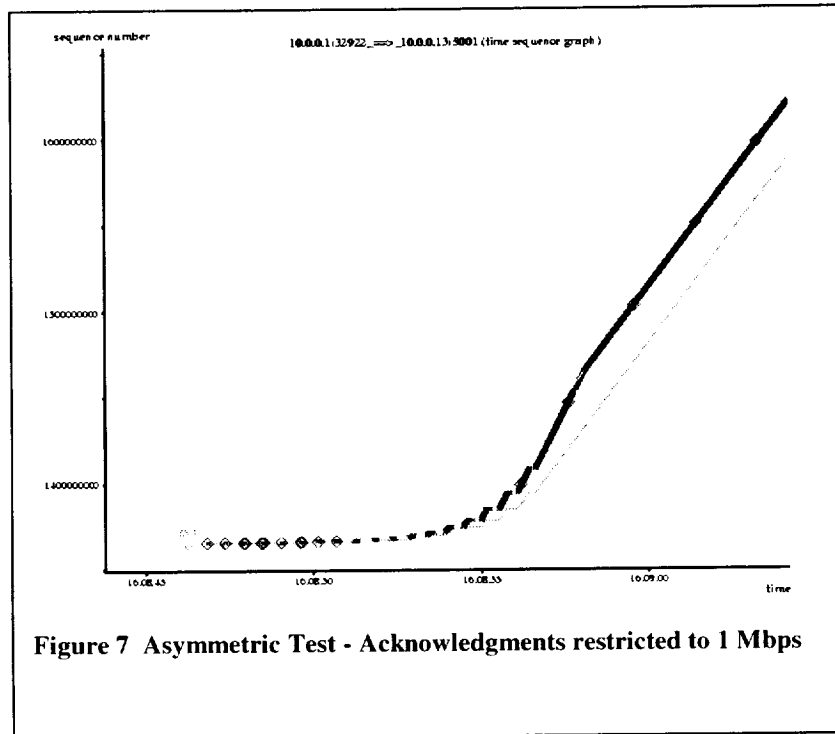
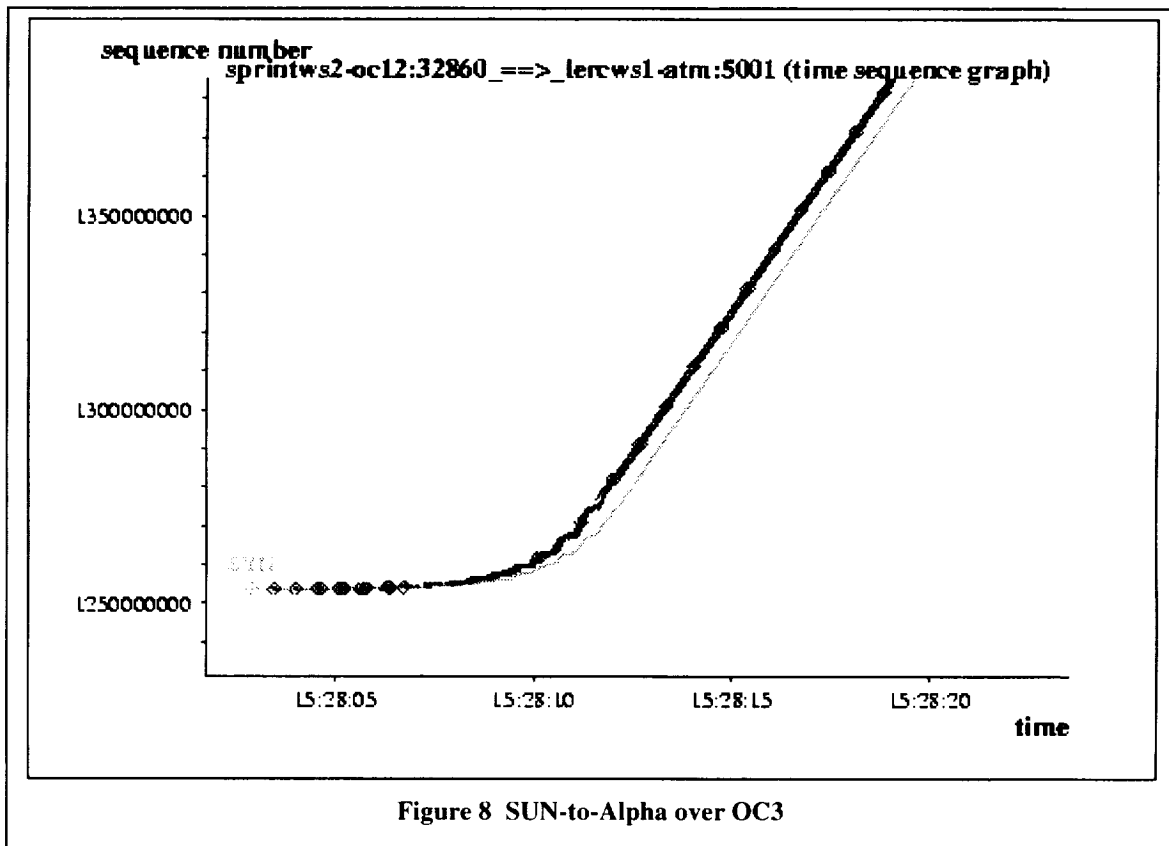


Figure 7 Asymmetric Test - Acknowledgments restricted to 1 Mbps

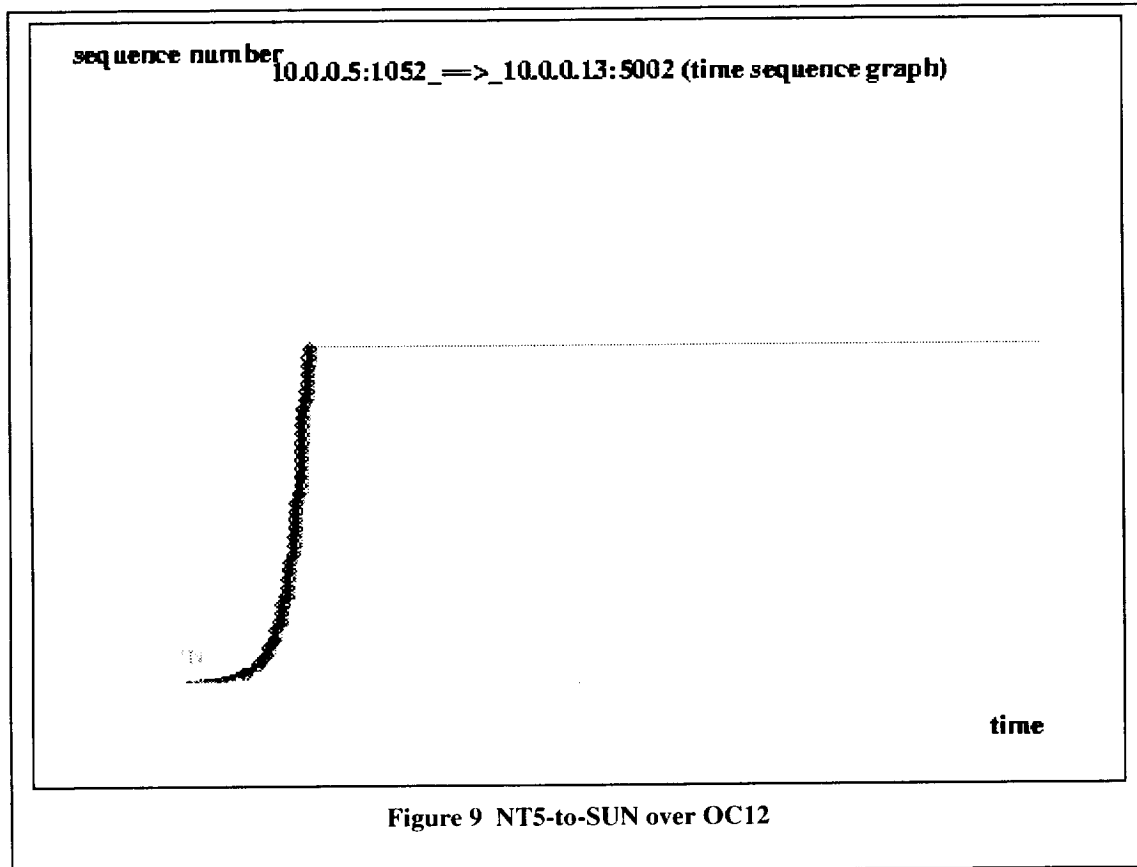
### Solaris-to-OSF1 (SUN-to-Alpha)

Figure 8 shows the trace for data transfer from a Sun workstation located at Sprint to an Alpha workstation at Glenn as depicted in the configuration in Figure 4. The Alpha workstation only had OC3 interface cards. Therefore, the maximum theoretical throughput is approximately 135 Mbps. This run obtained a throughput of approximately 101 Mbps. From the large trace everything appears to be functioning properly. Thus we should be able to obtain approximately 130 Mbps. Further investigation is necessary to determine why this was not the case as it does not appear to be due to the protocol implementation.



### NT5-to-Solaris (NT5-to-SUN)

Figure 9 show the trace for data transferred from an Intel development platform running Microsoft NT/5 to a Sun workstation running Solaris. The test is configured as shown in Figure 4 and running at OC12 rates. Notice that the system appears to be performing properly but suddenly stops. At this point, the NT5 system crashes and has to be rebooted. One area we suspect may be causing this is memory management or resource limitation problems. Obviously, further work is required to understand and correct this problem.



## IRIX-to-Solaris (SGI-to-SUN) Loopback

This IRIX-to-Solaris loopback test is considered invalid in that we do not consider the data to reflect to capabilities of either the IRIX or Solaris TCP stacks. Rather, it is included here as an item of interest to show how important it is to properly configure the network and to analyze the data as you go.

During the I18x experiments, there were times when the end-to-end link could not be established from Burlingame to Glenn. During one of these times we established a loopback configuration shown in Figure 10. The data was transmitted over the OC12 link through ACTS and the acknowledgements were transferred back through a channel emulator that was configured to provide an error-free channel with 250 milliseconds of delay. The trace of this test is shown in Figures 11a, 11b, and 11c. From the large trace in Figure 11a and the blowup in Figure 11b, one can see a series of retransmission that cause the congestion window to close. The system then enters linear growth as shown by the slow, but steady increase in the transmission of packets. The retransmission timers appear to be triggered even though the original data is being acknowledged rather than the retransmitted packets. We are not sure what caused the problem but believe it may be some subtlety in the test configuration – particularly since the SGI-to-SUN tests over ACTS perform much better [Fig. 12]. Further investigation is necessary.

A close examination of Figure 11c shows another oddity that resulted from the configuration setup. Note that the acknowledgements appear to be received by the server approximately 250 milliseconds after the data is transmitted. However, the round trip time is set for over 500 milliseconds and is split equally between the ACTS link and the channel emulator. Unfortunately, we captured the acknowledgements prior to passing through the channel emulator as shown in Figure 10. We should have captured the acknowledgements after the channel emulator. This did not affect the validity of the data, only the time relationship between the data and the acknowledgments.

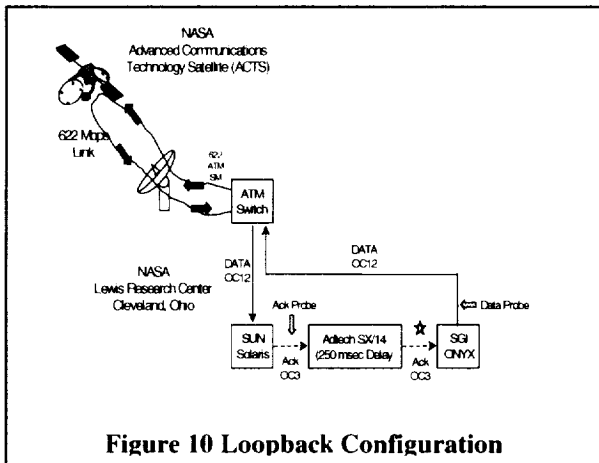


Figure 10 Loopback Configuration

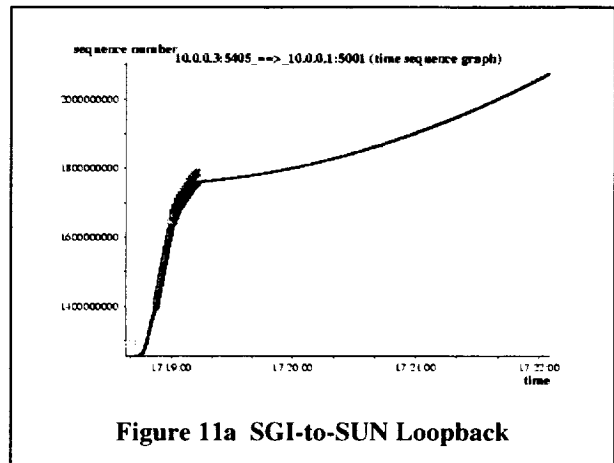


Figure 11a SGI-to-SUN Loopback

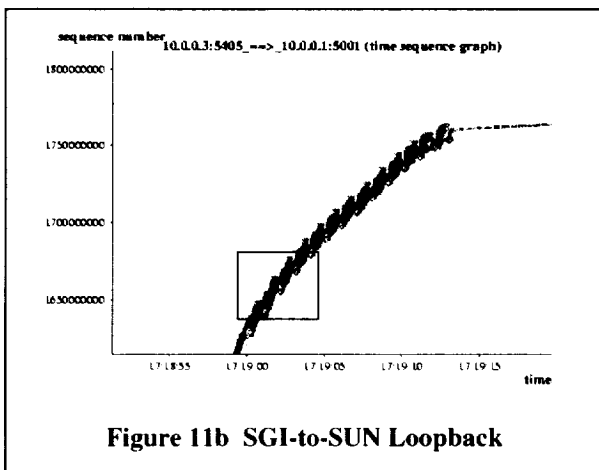


Figure 11b SGI-to-SUN Loopback

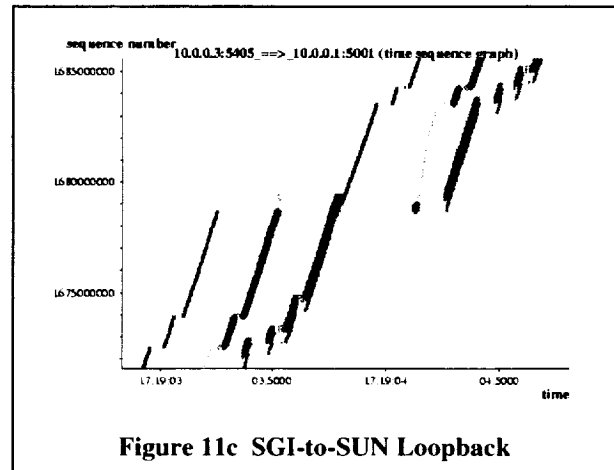
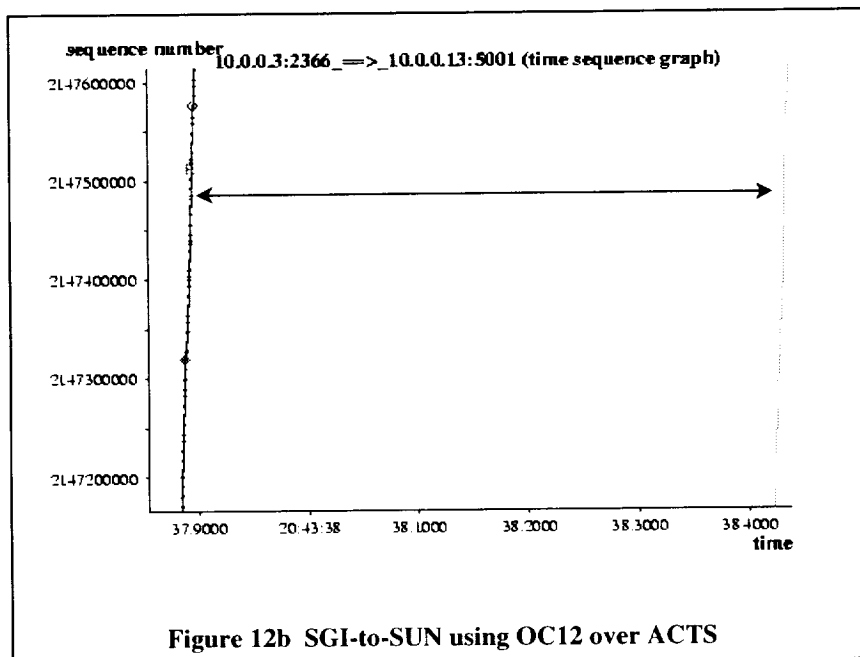
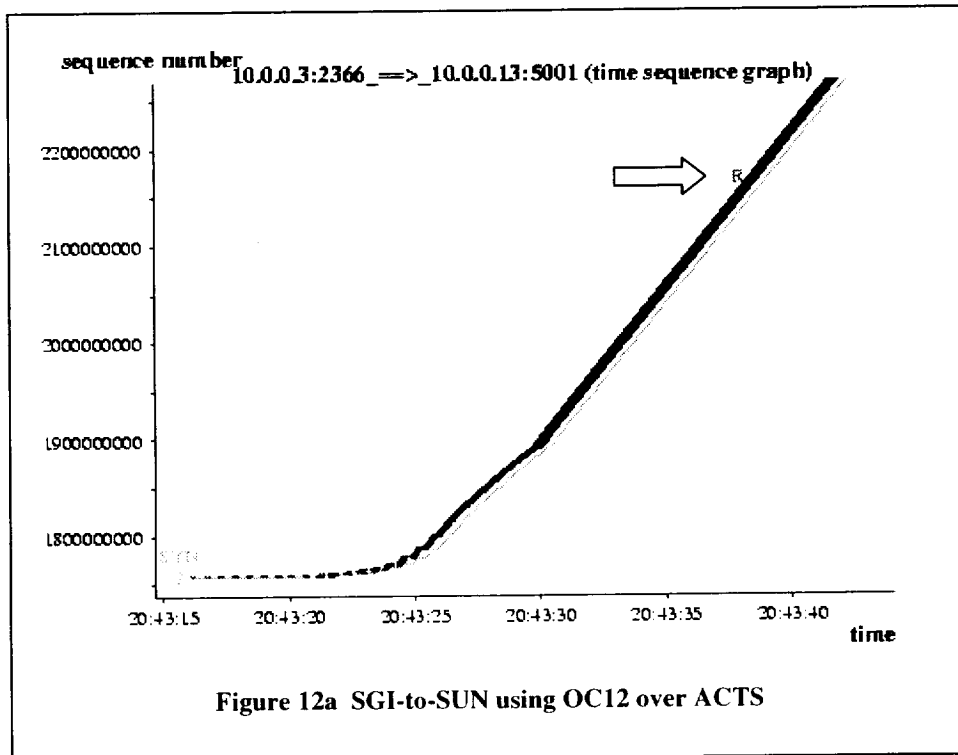


Figure 11c SGI-to-SUN Loopback



## IRIX-to-Solaris (SGI-to-SUN) Loopback over ACTS

Figure 12a shows the trace of an IRIX-to-Solaris transmission over the ACTS satellite using OC12 in both directions. From the large trace everything appears to be functioning properly. However, a blowup of the linear area shows that occasionally a packet is being marked as a retransmitted packet even through it appears to be the original transmission of the packet [Fig 12b]. Thus, there appears to be a minor "bug" somewhere. Further investigation of the raw data is necessary. Also, there is a secondary knee in the curve indicating a slight change in the transmission rate. We currently are not able to explain this anomaly.



## Scripts and Machine Information

The following are examples of the machine and ATM switch information that was obtained for each run. For some vendor's machines we were able to record TCP setting and statistics. However, this useful information was not available on all machines. The Intel Development System using NT4/5 did not provide access to these parameters. SONET information was collected at the ATM switch ports at the HDR and the workstation. Workstation ATM interface level statistics were collected at the ATM network interface card on the workstation. Notice that for different workstations, the interface cards are different as is the information available about that interface. Three systems are represented: Solaris, OSF1, and IRIX.

### Sun Solaris

tcp run on Fri Sep 4 12:51:08 PDT 1998:

#### General Information

```
Host Name is      sun-sprint
Host Aliases is   sun-sprint loghost
Host Address(es) is xxx.x.xx.xxx
Host ID is        808ca7bd
Serial Number is  2156701629
Manufacturer is   Sun (Sun Microsystems)
System Model is   Ultra 2 Model 2170
Main Memory is    512 MB
Virtual Memory is 662 MB
ROM Version is    OBP 3.7.0 1997/01/09 13:06
Number of CPUs is 2
CPU Type is       sparc
App Architecture is sparc
Kernel Architecture is sun4u
OS Name is        SunOS
OS Version is     5.7
Kernel Version is SunOS Release 5.7 Version kcpoon_[on998-1]_06/22/98 [UNIX(R) System V Release 4.0]
Boot Time is      Fri Sep 4 11:28:09 1998
```

=> Active /etc/system parameters:

```
-----
set ba:atm_strhead_buf=32768000
set ba:atm_qfullnoflowctl=1
set sq_max_size=0
```

=> SunOS\_5.7 TCP system parameters using /usr/sbin/ndd /dev/tcp:

```
tcp_cwnd_max = 37000000
tcp_maxpsz_multiplier = 2
tcp_mss_def = 536
tcp_mss_max = 65495
tcp_mss_min = 1
tcp_naglim_def = 4095
tcp_rexmit_interval_initial = 3000
tcp_rexmit_interval_max = 60000
tcp_rexmit_interval_min = 200
tcp_wroff_xtra = 32
tcp_deferred_ack_interval = 50
tcp_snd_lowat_fraction = 0
tcp_sth_rcv_hiwat = 0
tcp_sth_rcv_lowat = 0
tcp_dupack_fast_retransmit = 10
```

```

tcp_ignore_path_mtu = 0
tcp_xmit_hiwat = 8192
tcp_xmit_lowat = 2048
tcp_recv_hiwat = 8192
tcp_recv_hiwat_minmss = 4
tcp_fin_wait_2_flush_interval = 675000
tcp_co_min = 64
tcp_max_buf = 37000000
tcp_zero_win_probesize = 1
tcp_strong_iss = 1
tcp_rtt_updates = 10
tcp_wscale_always = 1
tcp_tstamp_always = 0
tcp_tstamp_if_wscale = 1
tcp_rexmit_interval_extra = 0
tcp_deferred_acks_max = 8
tcp_slow_start_after_idle = 2
tcp_slow_start_initial = 2
tcp_co_timer_interval = 20
tcp_sack_permitted = 1

```

=> TCP and IP netstat information

```

TCP   tcpRtoAlgorithm = 4 tcpRtoMin = 200
      tcpRtoMax = 60000 tcpMaxConn = -1
      tcpActiveOpens = 29 tcpPassiveOpens = 30
      tcpAttemptFails = 1 tcpEstabResets = 1
      tcpCurrEstab = 6 tcpOutSegs = 1183202
      tcpOutDataSegs = 4658 tcpOutDataBytes = 172502
      tcpRetransSegs = 109 tcpRetransBytes = 414581
      tcpOutAck = 1178540 tcpOutAckDelayed = 2413
      tcpOutUrg = 2 tcpOutWinUpdate = 0
      tcpOutWinProbe = 0 tcpOutControl = 171
      tcpOutRsts = 26 tcpOutFastRetrans = 0
      tcpInSegs = 2439283
      tcpInAckSegs = 4421 tcpInAckBytes = 137658
      tcpInDupAck = 3827 tcpInAckUnsent = 0
      tcpInInorderSegs = 2433143 tcpInInorderBytes = 47030744
      tcpInUnorderSegs = 3728 tcpInUnorderBytes = 32897716
      tcpInDupSegs = 47 tcpInDupBytes = 73588
      tcpInPartDupSegs = 0 tcpInPartDupBytes = 0
      tcpInPastWinSegs = 0 tcpInPastWinBytes = 0
      tcpInWinProbe = 0 tcpInWinUpdate = 0
      tcpInClosed = 3 tcpRttNoUpdate = 17
      tcpRttUpdate = 4368 tcpTimRetrans = 155
      tcpTimRetransDrop = 3 tcpTimKeepalive = 0
      tcpTimKeepaliveProbe = 0 tcpTimKeepaliveDrop = 0
      tcpListenDrop = 0 tcpListenDropQ0 = 0
      tcpHalfOpenDrop = 0 tcpOutSackRetrans = 0

IP    ipForwarding = 2 ipDefaultTTL = 255
      ipInReceives = 2442936 ipInHdrErrors = 0
      ipInAddrErrors = 0 ipInCksumErrs = 0
      ipForwDatagrams = 0 ipForwProhibits = 0

```

```

ipInUnknownProtos = 62      ipInDiscards = 0
ipInDelivers      =2439790  ipOutRequests  =1185314
ipOutDiscards    = 0 ipOutNoRoutes = 0
ipReasmTimeout   = 60      ipReasmReqds  = 0
ipReasmOKs       = 0 ipReasmFails  = 0
ipReasmDuplicates = 0      ipReasmPartDups = 0
ipFragOKs        = 0 ipFragFails   = 0
ipFragCreates    = 0 ipRoutingDiscards = 0
tcpInErrs        = 0      udpNoPorts    = 1725
udpInCksumErrs   = 0      udpInOverflows = 0
rawipInOverflows = 0

```

=> Switch ATM interface level stats for HDR port

-----  
hdrsprint is on switch sprintsw1 using port 1A1.

```

sonetLineBIPs = 47
sonetLineFEBEs = 0
sonetLineAISs = 0
sonetLineRDIs = 0
sonetSectionBIPs = 3
sonetSectionLOSs = 0
sonetSectionLOFs = 0
sonetPathBIPs = 1034456642
sonetPathFEBEs = 2598436405
sonetPathLOPs = 427
sonetPathAISs = 560430
sonetPathRDIs = 67016
sonetPathUNEQs = 4308
sonetPathPLMs = 46768

```

=> Switch ATM interface level stats for workstation

-----  
sun-sprint is on switch sprintsw1 using port 1B1.

```

sonetLineBIPs = 3781
sonetLineFEBEs = 1839
sonetLineAISs = 1
sonetLineRDIs = 7
sonetSectionBIPs = 984
sonetSectionLOSs = 3
sonetSectionLOFs = 0
sonetPathBIPs = 3671
sonetPathFEBEs = 8702
sonetPathLOPs = 0
sonetPathAISs = 0
sonetPathRDIs = 7
sonetPathUNEQs = 0
sonetPathPLMs = 0

```

=> Workstation ATM interface level stats:

-----  
intrs 2435908 inits 7  
ipackets 2436710 opackets 1180661

```

ierrors          1  oerrors          0
out of rbufs     0  out of tbufs     0
canput fails     0  flow ctls       0
copy receives    20871 allocb fails  0
too many bytes   0  rx overflows    1
out of txds      1977 bad cres     0
no receivers     0  err encaps      0
err acks        0  txc overflows   0
rx memnotav     0  rx statenotav   0
rx badcells     3  rx flush count  17
rx dirty count   0  rx targ kicks   0
sbufnum         192 hbufnum          0
IP disabled VCs  0  rx bogus len    0
rx PFIFO full    0

```

atmspeed = 1 Mbps

Driver Type using pkginfo -l SUNWatm:

```

PKGINST: SUNWatm
NAME: SunATM Device Drivers
VERSION: 3.0
PSTAMP: mythos980506190734

```

## Silicon Graphics IRIX

ttcp run on Tue Sep 22 14:16:52 EDT 1998:

IRIX64

IRIX64 tng 6.4 02121744 IP27

FPU: MIPS R10010 Floating Point Chip Revision: 0.0

CPU: MIPS R10000 Processor Chip Revision: 2.6

2 180 MHz IP27 Processors

Main memory size: 256 Mbytes

Instruction cache size: 32 Kbytes

Data cache size: 32 Kbytes

Secondary unified instruction/data cache size: 1 Mbyte

Integral SCSI controller 0: Version QL1040B (rev. 2)

Disk drive: unit 1 on SCSI controller 0

CDROM: unit 6 on SCSI controller 0

Integral SCSI controller 1: Version QL1040B (rev. 2)

Disk drive: unit 2 on SCSI controller 1

Disk drive: unit 3 on SCSI controller 1

Disk drive: unit 4 on SCSI controller 1

IOC3 serial port: tty1

IOC3 serial port: tty2

IOC3 serial port: tty3

IOC3 serial port: tty4

IOC3 parallel port: plp1

Graphics board: Reality

Integral Fast Ethernet: ef0, version 1

Iris Audio Processor: version RAD revision 7.0, number 1

IOC3 external interrupts: 1

IRIX64 No support yet, for grabbing the TCP settings....

=> TCP and IP netstat information

tcp:

- 5586396 packets sent
  - 5390797 data packets (3181974920 bytes)
  - 5343 data packets (44921894 bytes) retransmitted
  - 45318 ack-only packets (33176 delayed)
  - 0 URG only packets
  - 0 window probe packets
  - 138661 window update packets
  - 6277 control packets
- 3639326 packets received
  - 462544 pcb cache misses
  - 2662259 acks (for 3172696118 bytes)
  - 402296 ack predictions ok
  - 8606 duplicate acks
  - 0 acks for unsent data
  - 1094335 packets (2903221797 bytes) received in-sequence
  - 951523 in-sequence predictions ok
  - 273 completely duplicate packets (2370775 bytes)
  - 0 packets with some dup. data (0 bytes duped)
  - 5494 out-of-order packets (34956286 bytes)
  - 100 packets (885034 bytes) of data after window
  - 0 window probes
  - 4963 window update packets
  - 5 packets received after close
  - 0 discarded for bad checksums
  - 0 discarded for bad header offset fields
  - 0 discarded because packet too short
  - 0 discarded because of old timestamp
- 2097 connection requests
- 2099 connection accepts
  - 0 listen queue overflows
  - 0 bad connection attempts
  - 0 drops from listen queue
- 4192 connections established (including accepts)
- 5766 connections closed (including 1007 drops)
- 3 embryonic connections dropped
- 2510505 segments updated rtt (of 372969 attempts)
- 197 retransmit timeouts
  - 1 connection dropped by rexmit timeout
- 1 persist timeout
  - 0 connections dropped by persist timeout
- 800 keepalive timeouts
  - 1 keepalive probe sent
  - 0 connections dropped by keepalive

ip:

- 3783250 total packets received
- 0 bad header checksums
- 0 with size smaller than minimum
- 0 with data size < data length
- 0 with header length < data size
- 0 with data length < header length
- 0 with bad options
- 0 fragments received

```

0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
3783220 packets for this host
123 packets recvd for unknown/unsupported protocol
0 packets forwarded (forwarding enabled)
30 packets not forwardable
0 redirects sent
5603814 packets sent from this host
0 output packets dropped due to no bufs, etc.
0 output packets discarded due to no route
0 datagrams fragmented
0 fragments created
0 datagrams that can't be fragmented
1195/1250 mbufs in use:
  913 mbufs allocated to data
  97 mbufs allocated to socket structures
  136 mbufs allocated to protocol control blocks
  37 mbufs allocated to socket names and addresses
  12 mbufs allocated to interface addresses
35/36 mapped pages in use
576 Kbytes allocated to network (97% in use)
0 requests for memory denied
0 requests for memory delayed
0 calls to protocol drain routines

```

Resource	Failures	Avail	In Use	Max Used	Total Used
streams	0	0	45	53	2272127
events	0	0	2	2	2
queues	0	0	198	238	9091872
link blks	0	0	4	4	24
mdb blks	0	13	6	310	10214377
msg blks	0	7	0	186	848037

=> Switch ATM interface level stats for HDR port

-----  
hdlrlrc is on switch leresw2 using port 2D1.

```

sonetLineBIPs = 6725
sonetLineFEBEs = 2464
sonetLineAISs = 0
sonetLineRDIs = 8
sonetSectionBIPs = 1292369103
sonetSectionLOSs = 309279
sonetSectionLOFs = 0
sonetPathBIPs = 2340183131
sonetPathFEBEs = 3361813675
sonetPathLOPs = 237590
sonetPathAISs = 0
sonetPathRDIs = 623
sonetPathUNEQs = 282
sonetPathPLMs = 0

```

=> Switch ATM interface level stats for workstation

tng is on switch lercsw2 using port 2A1.

sonetLineBIPs = 17695  
sonetLineFEBEs = 20940  
sonetLineAISs = 0  
sonetLineRDIs = 1  
sonetSectionBIPs = 2897  
sonetSectionLOSs = 0  
sonetSectionLOFs = 0  
sonetPathBIPs = 5399  
sonetPathFEBEs = 2860  
sonetPathLOPs = 0  
sonetPathAISs = 0  
sonetPathRDIs = 1  
sonetPathUNEQs = 0  
sonetPathPLMs = 0

=> Workstation ATM interface level stats:

-----  
Physical statistics:

Output	Input	Errors
Cells	Cells	Framing Hdr-CRC
779701076	151288655	0 0

PMD statistics:

Section	Line	Line	Path	Path	Corr	Uncorr
BIP	BIP	FEBE	BIP	FEBE	HCS	HCS
0	0	0	0	0	0	0

ATM statistics:

Output	Input	Errors
Cells	Cells	VPI-OOR VPI-NoC VCI-OOR VCI-NoC
779701076	151288569	0 0 0 86

AAL5 statistics:

Output	Input	Errors
Cells	CS-PDUs	Cells CS-PDUs CProto Pay-CRC Congestn Drops
779701076	4900237	151288569 3254031 0 0 0 0

Device statistics:

Buffer Allocation Failures						
Type 1		Type 2		Receive Queue Full Carrier		
Small	Large	Small	Large	Receive	Queue	Full
417	0	0	0	0	ON	

FORE Systems Release: ForeThought\_5.0 Beta\_1.0 (25260)

fatm2: HE622 Media=OC12-MM-SC HW=0.0.0 FW=0.0.0 Serial=2949209  
Module\_ID=1 XIO\_Slot=2 PCI\_Slot=2  
MAC=00:20:48:2D:00:59

### Compaq / Digital Equipment Alpha OSF1

ttcp run on Tue Sep 8 15:22:45 EDT 1998:

Hostname: lercws1  
Operating System: OSF1  
Release: V4.0  
Version: 878  
Architecture: alpha  
Hardware Manufacturer: Digital



Hardware Serial Number: 0

=> OSF1\_V4.0 TCP system parameters using /sbin/sysconfig -q inet:

inet:

inifaddr\_hsize = 32  
ipdefttl = 64  
ipdirected\_broadcast = 0  
ipforwarding = 0  
ipfragttl = 60  
ipgateway = 0  
ippport\_userreserved = 5000  
ipsendredirects = 1  
ipsrroute = 1  
pmtu\_decrease\_intvl = 1200  
pmtu\_enabled = 1  
pmtu\_increase\_intvl = 240  
pmtu\_rt\_check\_intvl = 20  
subnetsarelocal = 1  
tcbbhashsize = 32  
tcbbquicklisten = 0  
tcp\_compat\_42 = 1  
tcp\_cwnd\_segments = 2  
tcp\_dont\_winscale = 0  
tcp\_keepalive\_default = 0  
tcp\_keepcnt = 8  
tcp\_keeppidle = 14400  
tcp\_keepinit = 150  
tcp\_keepintvl = 150  
tcp\_msl = 60  
tcp\_mssdflt = 536  
tcp\_nodelack = 0  
tcp\_recvspace = 32768  
tcp\_rexmit\_interval\_min = 2  
tcp\_rexmitthresh = 3  
tcp\_rtt\_dflt = 3  
tcp\_sendspace = 32768  
tcp\_ttl = 60  
tcptwreorder = 1  
tcp\_urgent\_42 = 1  
udpcksum = 1  
udp\_recvspace = 41600  
udp\_sendspace = 9216  
udp\_ttl = 30

=> OSF1\_V4.0 SOCKET system parameters using /sbin/sysconfig -q socket:

socket:

sbcompress\_threshold = 0  
sb\_max = 8192000  
sobacklog\_drops = 0  
sobacklog\_hiwat = 2  
somainconn = 1024  
somainconn\_drops = 0  
somainconn = 0  
tss\_disable = 0  
tssmap\_min\_len = 1024

tssmap\_max\_pages = 16384  
tssmap\_wireubc = 0  
tssmap\_quicksum = 0

=> TCP and IP netstat information

tcp:

99755 packets sent  
  96066 data packets (854932655 bytes)  
  12 data packets (73360 bytes) retransmitted  
  1898 ack-only packets (641 delayed)  
  0 URG only packets  
  2 window probe packets  
  1560 window update packets  
  221 control packets  
54899 packets received  
  49213 acks (for 854932812 bytes)  
  872 duplicate acks  
  0 acks for unsent data  
  5221 packets (29014445 bytes) received in-sequence  
  2 completely duplicate packets (172 bytes)  
  0 packets with some dup. data (0 bytes duped)  
  1024 out-of-order packets (8151720 bytes)  
  0 packets (0 bytes) of data after window  
  0 window probes  
  0 window update packets  
  4 packets received after close  
  0 discarded for bad checksums  
  0 discarded for bad header offset fields  
  0 discarded because packet too short  
95 connection requests  
82 connection accepts  
177 connections established (including accepts)  
160 connections closed (including 3 drops)  
0 embryonic connections dropped  
2484 segments updated rtt (of 2497 attempts)  
3 retransmit timeouts  
  0 connections dropped by rexmit timeout  
0 persist timeouts  
0 keepalive timeouts  
  0 keepalive probes sent  
  0 connections dropped by keepalive

ip:

57576 total packets received  
0 bad header checksums  
0 with size smaller than minimum  
0 with data size < data length  
0 with header length < data size  
0 with data length < header length  
0 fragments received  
0 fragments dropped (dup or out of space)  
0 fragments dropped after timeout  
0 packets forwarded  
0 packets not forwardable  
0 packets denied access  
0 redirects sent

0 packets with unknown or unsupported protocol  
57576 packets consumed here  
102043 total packets generated here  
10 lost packets due to resource problems  
0 total packets reassembled ok  
0 output packets fragmented ok  
0 output fragments created  
0 packets with special flags set  
29 Kbytes for small data mbufs (peak usage 460 Kbytes)  
800 Kbytes for mbuf clusters (peak usage 14584 Kbytes)  
51 Kbytes for sockets (peak usage 58 Kbytes)  
68 Kbytes for protocol control blocks (peak usage 78 Kbytes)  
3 Kbytes for routing table (peak usage 3 Kbytes)  
2 Kbytes for interface addresses (peak usage 2 Kbytes)  
10 Kbytes for socket names (peak usage 10 Kbytes)  
< 1 Kbyte for ip multicast addresses (peak usage < 1 Kbyte)  
< 1 Kbyte for interface multicast addresses (peak usage < 1 Kbyte)  
< 1 Kbyte for packet headers (peak usage 52 Kbytes)  
0 requests for mbufs denied  
0 calls to protocol drain routines  
963 Kbytes allocated to network

Network threads: 1 netisrthreads configured (peak active 1)

=> Switch ATM interface level stats for HDR port

-----  
hdlrlerc is on switch lercsw2 using port 2D1.

sonetLineBIPs = 34  
sonetLineFEBEs = 0  
sonetLineAISs = 0  
sonetLineRDIs = 6  
sonetSectionBIPs = 0  
sonetSectionLOSs = 0  
sonetSectionLOFs = 0  
sonetPathBIPs = 13  
sonetPathFEBEs = 176  
sonetPathLOPs = 0  
sonetPathAISs = 0  
sonetPathRDIs = 0  
sonetPathUNEQs = 0  
sonetPathPLMs = 0

=> Switch ATM interface level stats for workstation

-----  
lercws1 is on switch lercsw1 using port 1C1.

sonetLineBIPs = 575  
sonetLineFEBEs = 403  
sonetLineAISs = 0  
sonetLineRDIs = 0  
sonetSectionBIPs = 81190888  
sonetSectionLOSs = 1697  
sonetSectionLOFs = 0

sonetPathBIPs = 224  
sonetPathFEBEs = 182  
sonetPathLOPs = 0  
sonetPathAISs = 0  
sonetPathRDIs = 0  
sonetPathUNEQs = 0  
sonetPathPLMs = 0

=> Workstation ATM interface level stats:

-----  
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll Drop  
lis0 9180 <Link> 52493 0 97023 4 375 0  
lis0 9180 10 lercwsl-oc12 52493 0 97023 4 375 0

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<b>1. AGENCY USE ONLY</b> (Leave blank)	<b>2. REPORT DATE</b> July 1999	<b>3. REPORT TYPE AND DATES COVERED</b> Technical Memorandum	
<b>4. TITLE AND SUBTITLE</b> ACTS 118x Final Report High Speed TCP Interoperability Testing		<b>5. FUNDING NUMBERS</b>  WU-632-50-5A-00	
<b>6. AUTHOR(S)</b> David E. Brooks, Craig Buffinton, Dave R. Beering, Arun Welch, William D. Ivancic, Mike Zernic, and Douglas J. Hoder		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  E-11724	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191		<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>  NASA TM-1999-209272	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> National Aeronautics and Space Administration Washington, DC 20546-0001		<b>11. SUPPLEMENTARY NOTES</b> David E. Brooks, Sterling Software, Cleveland, Ohio; Craig Buffinton, NetManage Inc., Cupertino, California; Dave R. Beering, Global Infrastructures, LLC, Chicago, Illinois; Arun Welch, Anzus Consulting, Orleans, Massachusetts; and William D. Ivancic, Mike Zernic, and Douglas J. Hoder, NASA Glenn Research Center. Responsible person, William D. Ivancic, organization code 5610, (216) 433-3494.	
<b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b>  Unclassified - Unlimited Subject Category: 17  This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.		<b>12b. DISTRIBUTION CODE</b>  Distribution: Standard	
<b>13. ABSTRACT (Maximum 200 words)</b>  With the recent explosion of the Internet and the enormous business opportunities available to communication system providers, great interest has developed in improving the efficiency of data transfer over satellite links using the Transmission Control Protocol (TCP) of the Internet Protocol (IP) suite. The NASA's ACTS experiments program initiated a series of TCP experiments to demonstrate scalability of TCP/IP and determine to what extent the protocol can be optimized over a 622 Mbps satellite link. Through partnerships with the government technology oriented labs, computer, telecommunication, and satellite industries NASA Glenn was able to (1) promote the development of interoperable, high-performance TCP/IP implementations across multiple computing / operating platforms; (2) work with the satellite industry to answer outstanding questions regarding the use of standard protocols (TCP/IP and ATM) for the delivery of advanced data services, and for use in spacecraft architectures; and (3) conduct a series of TCP/IP interoperability tests over OC12 ATM over a satellite network in a multi-vendor environment using ACTS. The experiments' various network configurations and the results are presented			
<b>14. SUBJECT TERMS</b>  Satellite communications; Protocols; ATM; TCP		<b>15. NUMBER OF PAGES</b> 41	
		<b>16. PRICE CODE</b> A03	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b>

## References

---

- <sup>1</sup> C.M. Carlin, I. Lopez, L.S. Hedges: Remote Propulsion Simulation, ACTS Results Conference, September 11-13, 1995.
- <sup>2</sup> Carlson, W. E., Spencer, S., Geroch, M., Glenn, M., Bedford, K., Welsh, D., Kelley, J., Welch, A.: Visualization Of Results From A Distributed, Coupled, Supercomputer-Based Mesoscale Atmospheric And Lake Models Using *The NASA ACTS*, ACTS Results Conference, September 11-13, 1995.
- <sup>3</sup> Beering, D. R.: Satellite/Terrestrial Networks For Oil Exploration, ACTS Results Conference, September 11-13, 1995.
- <sup>4</sup> Van Jacobson, Michael Karels: Congestion Avoidance and Control, ACM SIGCOMM, Aug. 1988.  
<ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>.
- <sup>5</sup> Van Jacobson: Note to the IEFT end2end-interest working group, April 1990.
- <sup>6</sup> V. Jacobson, R. Braden, D. Borman TCP Extensions for High Performance. RFC 1323 May 1992. (Obsoletes RFC1072, RFC1185).
- <sup>7</sup> W. Stevens: Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, RFC 2001 January 1997.
- <sup>8</sup> Matt Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow. TCP Selective Acknowledgment Options, October 1996. RFC 2018.
- <sup>9</sup> D.E. Brooks, H. Gassman, D.R. Beering, A. Welch, D. J. Hoder, W.D. Ivancic: High-Speed TCP Testing November 1998. <http://ctd.lerc.nasa.gov/5610/inetprotocols.html>.