

SCPS-TP: A Satellite-Enhanced TCP

Keith Scott

The MITRE Corporation /
NASA Jet Propulsion Laboratory



Agenda

- Purpose
- Background
- Stressed Communication Environments
- SCPS-TP Features
- SCPS-TP Performance
- Performance Enhancing Proxies (PEPs)
- Ongoing and Future SCPS-TP Work
- Conclusions



Purpose

- **To present the Space Communications Protocol Standard – Transport Protocol (SCPS-TP) and SCPS-TP-based PEPs and their applicability to satellite communications**
 - Background
 - SCPS-TP Capabilities
 - Performance
 - SCPS-TP Performance Enhancing Procies
 - Ongoing Work



Background

- **Motivation for Internet communication *with* spacecraft**
 - Hypothesis 1: Internet/Intranet interfaces to spacecraft will simplify spacecraft bus and payload development and test
 - Hypothesis 2: Internet/Intranet access to spacecraft will reduce operations costs
 - Hypothesis 3: Internet/Intranet access to spacecraft will improve the quantity and timeliness of science data

- **Motivation for Internet communication *via* spacecraft**
 - Satellite communications to remote regions
 - Exploitation of broadcast technology

- **NASA has been working to establish and optimize Internet communication *with* spacecraft and *via* spacecraft since 1992**

'Stressed' Communications Environments Can Limit TCP Performance

■ High bit error rates

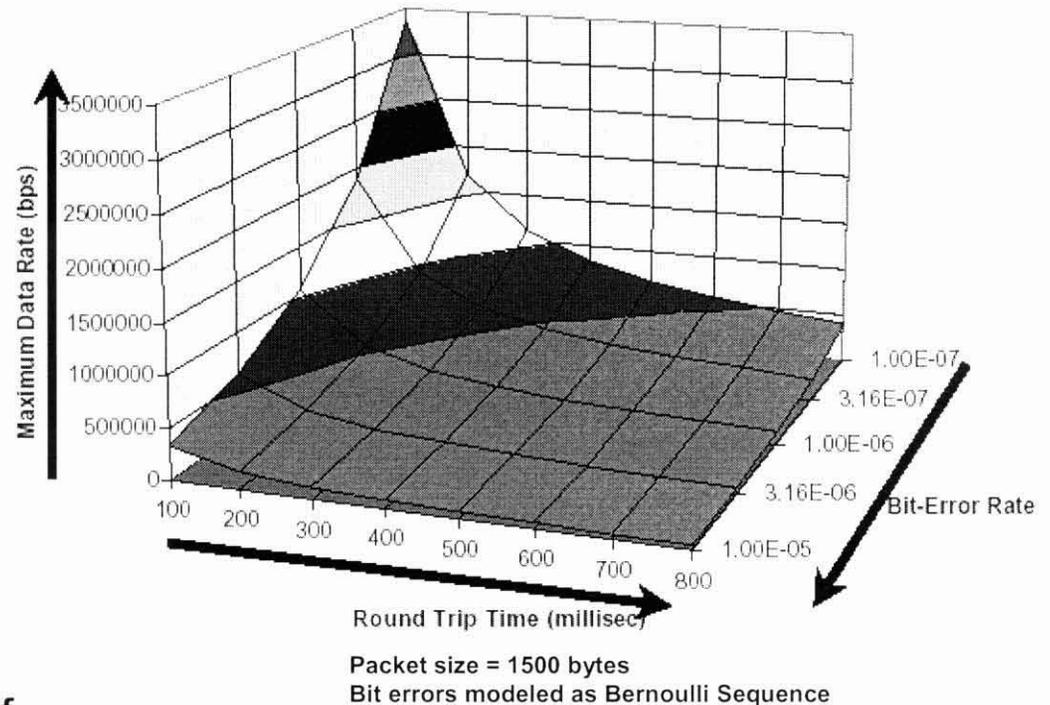
- TCP responds to all loss (including bit errors) by cutting its transmission rate in half and then rebuilding it by 1 packet per round trip time

■ Large delays

- TCP is 'self-clocked', with the round trip time (RTT) of the connection as the time constant
- Large RTT -> very slow recovery of transmission rate after congestion event

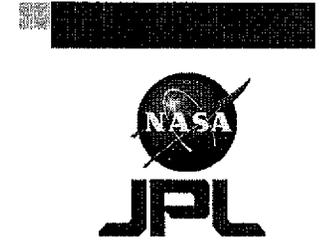
■ High asymmetry

- If the acknowledgement channel can't handle the ACK traffic, it will limit the sending rate



$$BW < \left(\frac{MSS}{RTT} \right) \frac{1}{\sqrt{p}}$$

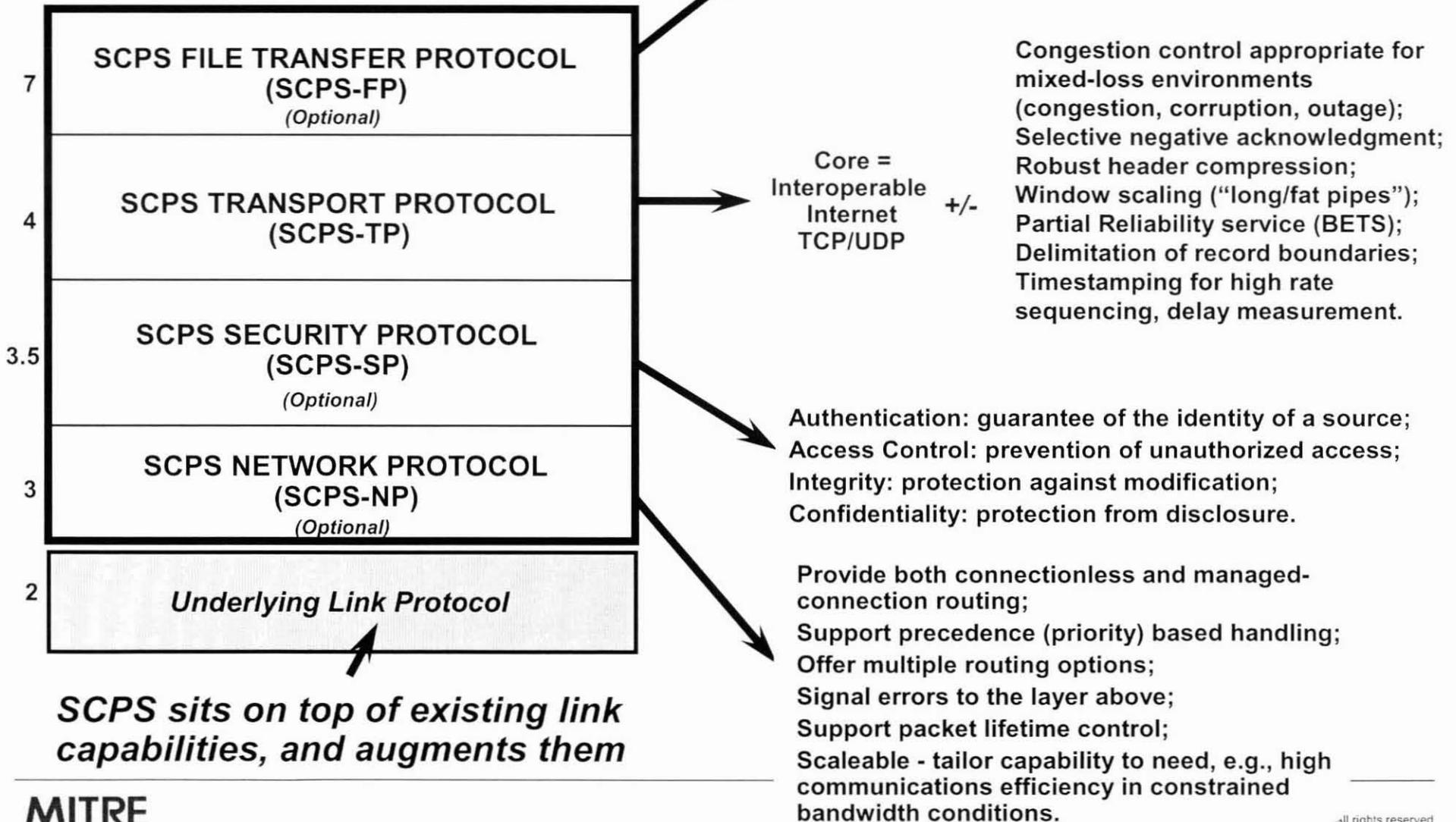
BW	=	bandwidth
MSS	=	Maximum TCP segment size
RTT	=	Round trip time
P	=	Packet loss probability



Options for Improving Performance over Stressed (e.g. Satellite) Links

- **Tune TCP Parameters of end systems**
 - Difficult
 - Brittle
- **Change the transport protocol itself**
 - Technical, legacy, political issues
- **Use performance enhancing proxies (PEPs)**
 - No modifications to end systems
 - Can specifically tune a few boxes (the PEPs)

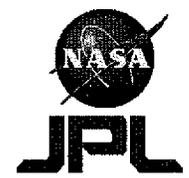
SCPS Capabilities





SCPS-TP Features

- **Range of congestion control mechanisms**
 - Appropriate reaction to congestion / corruption
 - Take full advantage of dedicated resources
 - ‘TCP-friendly’ when used over shared resources
 - Rate control absolutely prevents ‘self-congestion’
- **SNACKs**
 - Bandwidth-efficient retransmission scheme for quickly retransmitting (possibly large) holes
- **Robust Header Compression**
 - Don’t lose a whole window of data due to a single packet loss
- **Standard**
 - Multiple vendor support
- **TCP Compatibility**
 - Just in case you’ve only got one proxy

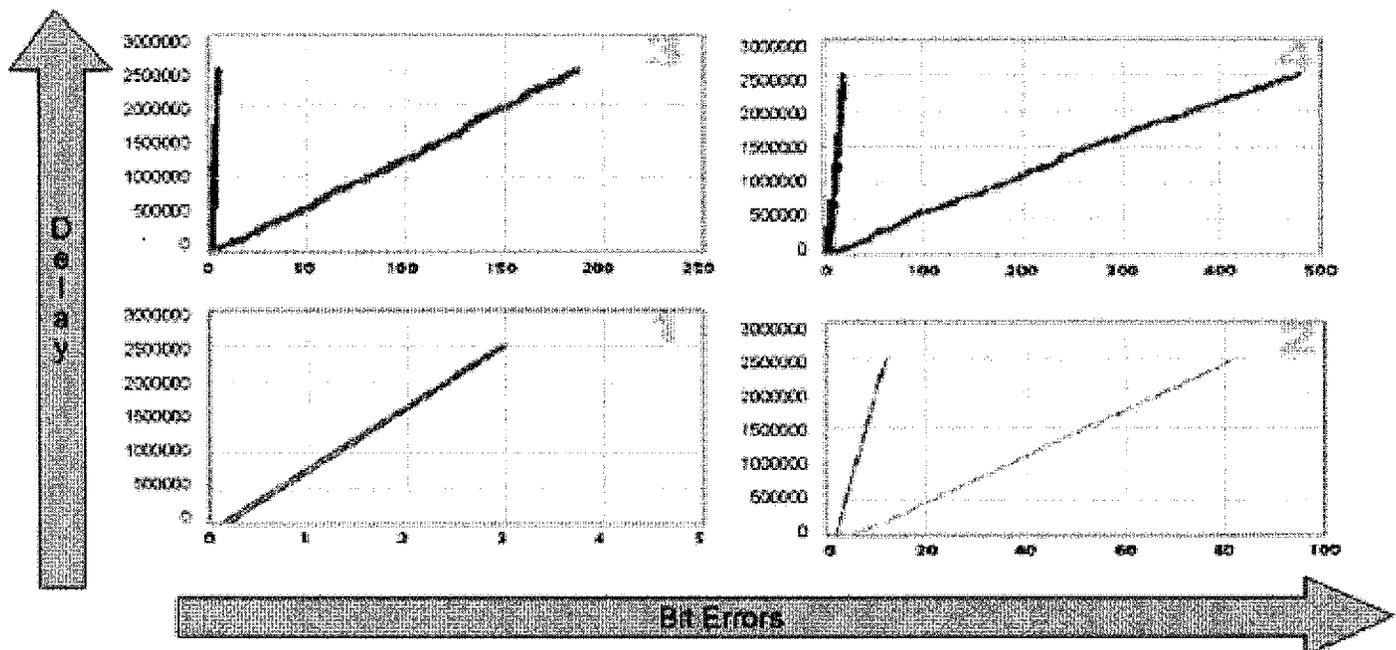


TCP Tranquility (SCPS-TP) Performance

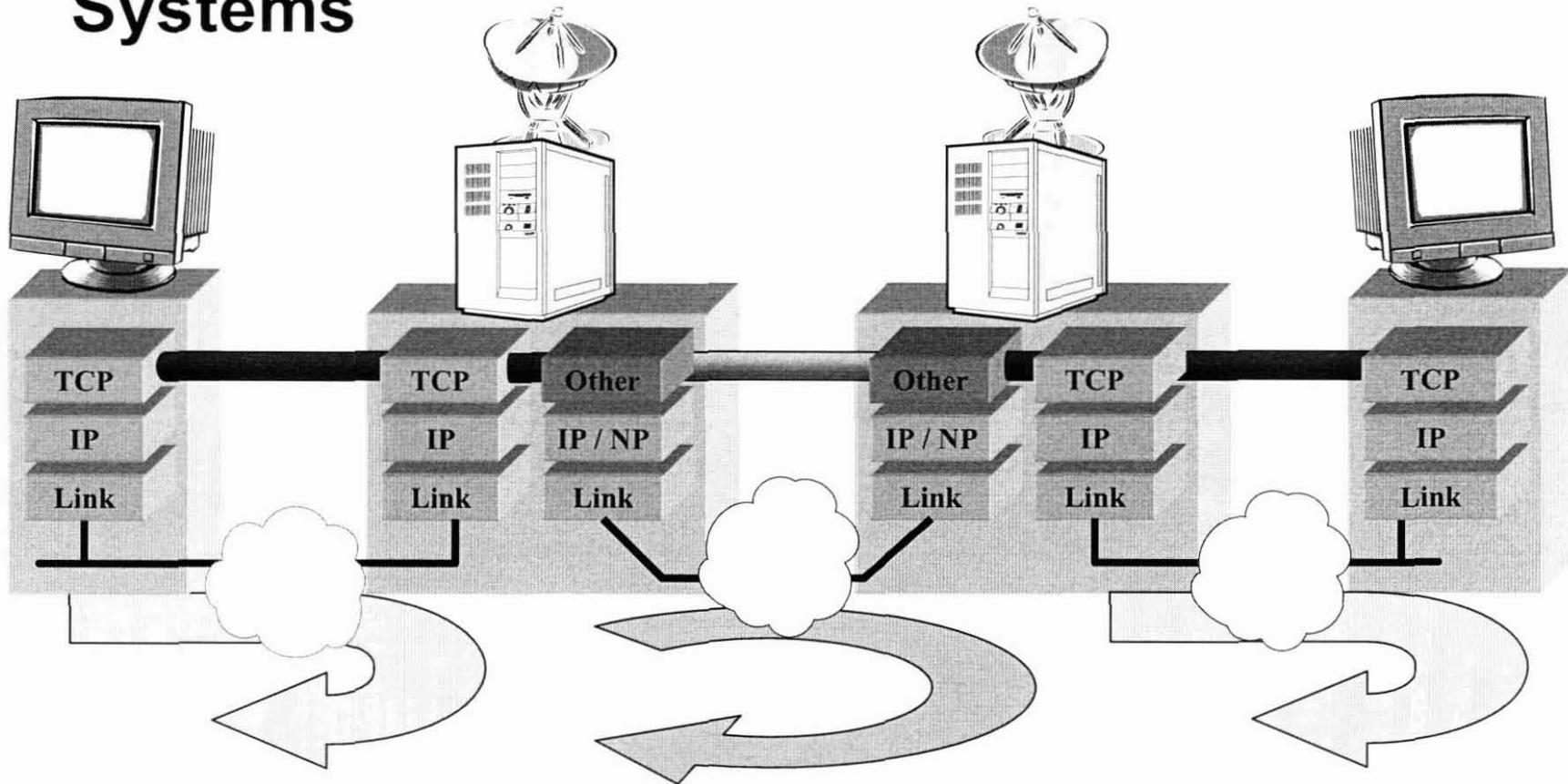
GlobalProtocols TCP Tranquility (SkipWare)

– http://www.globalprotocols.com/tech_tests.html

1. An error free 2 Mbps geosynchronous satellite link with a 540 millisecond round trip time
2. A 2 Mbps geosynchronous satellite link with a 5×10^6 BER
3. An error-free 11 Mbps 802.11 Wireless LAN connection
4. An 11 Mbps Wireless LAN experiencing a 5×10^6 BER



Transport-Layer Gateways: Enhancements for Unmodified End Systems



Gateways break the end-to-end connection into three separate transport-layer control loops. The middle loop can use a different transport protocol that's specifically tuned to the satellite channel.



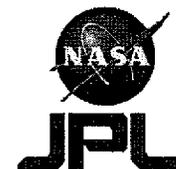
Proxies and The End-to-End Argument

- The basic argument is that, as a first principle, certain required end-to-end functions can only be correctly performed by the end systems themselves

- Often interpreted as: “If you do anything to a TCP connection in the middle of the network, you’re altering the end-to-end semantics of the connection, and that’s verboten”

- The reality: there is state in the network now that’s altering the end-to-end semantics
 - NAT boxes, Proxies, Firewalls, Encryption devices

- Pragmatists may care more about performance



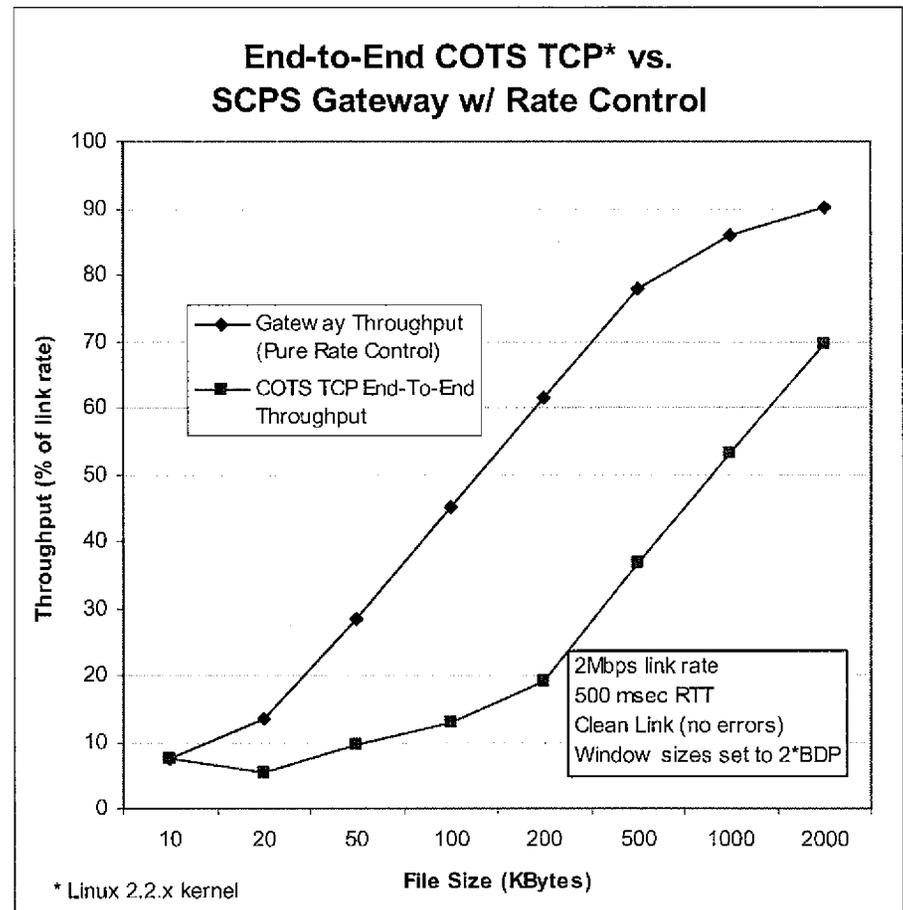
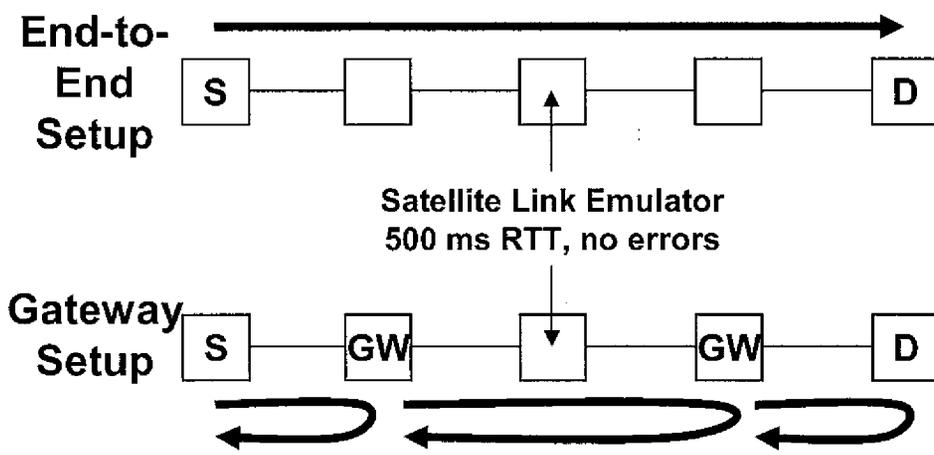
Motivation for a *SCPS*-based PEP

- **Highly tunable congestion control mechanisms**
 - Rate control for reserved-bandwidth paths
 - TCP Vegas (and variants) for potentially shared paths
- **Standard protocol (TCP)**
 - SCPS-TP is a standardized set of TCP extensions
 - Backward compatible with other TCP implementations



TCP Tranquility Gateway (PEP) Performance

- End-to-end TCP (no PEP) vs. SCPS-TP PEP
 - Gateway Settings
 - Rate control set to the satellite channel rate
 - Congestion control turned off





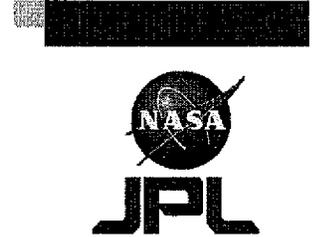
SCPS-TP-Based Implementations and Products

■ SCPS Reference Implementation

- End-system configuration
- TCP Performance Enhancing Proxy (PEP) configuration

■ Commercial products

•Vendor	•Product	•URL
•Global Protocols	•Skipware	• http://www.skipware.com
Inc. •Xiphos	•XipLink	• http://www.xiphos.ca/xiplink/index.html
•Hot lens	•Bandwidth Turbo	• http://www.hotlens.com/products/index.html
•Comtech EFData	Booster •Turbo IP	• http://206.223.8.10/linksite/manuals/datasheets/ds-turboip.pdf
•Cryptek	•Diamond Sat	• http://www.cryptek.com/Downloads/DIAMONDSAT.pdf
•Growell	•TurboBooster	• http://www.growellusa.com/p_booster01_01.html
•Expand Networks	•Expand Networks Accelerator:	• http://www.expand.com/product/Features.html



Ongoing SCPS-TP Work

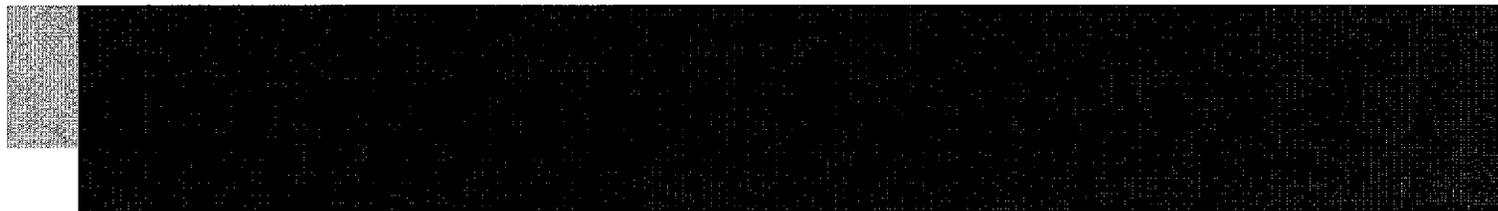
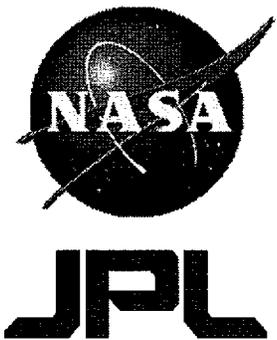
- **CCSDS Cislunar Networking Working Group**
 - Updates to SCPS-TP Spec
 - Selective ACKnowledgement (SACK)
 - Explicit Congestion Notification (ECN)
 - Other vendor enhancements that need standardization

- **Other implemented features / research topics**
 - Integrated application-layer (e.g. http) cacheing
 - Data compression
 - Guaranteed minimum bandwidth + congestion-control
 - React appropriately to bandwidth changes on outbound path
 - Making the best use of the combination of (SNACK, SACK, ECN, Vegas)



Conclusions

- SCPS-TP has a number of attractive features for communicating over satellite and other stressed links
- Transport layer proxies permit easy enhancement of end-to-end traffic
- Proxies using the SCPS-TP protocol present a number of attractive features



BACKUP



Activities: Protocol Development and Tuning (Cont.)

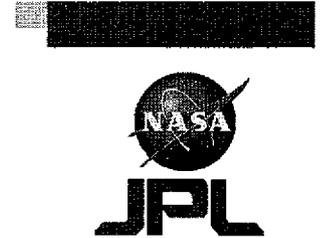
■ Mobile IP Extensions

- Registration acceleration (Ground station registers on behalf of all spacecraft endpoints)
- Mobile router -- Mobile IP registration of a subnet router

■ Quality of Service

- Integrated Services to Spacecraft (RSVP extensions for TCP proxy gateways)

Activities: Testing and Demonstrations (Cont.)



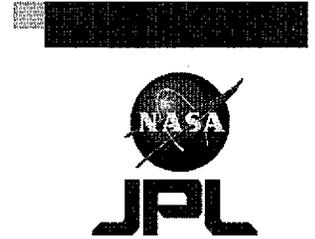
■ STRV 1b

- FTP/TCP/SCPS-TP testing on UK Defence Research Agency Satellite

■ Mobile Router

- Field trial of mobile router using US Coast Guard ship, 802.11 and commercial satellite service

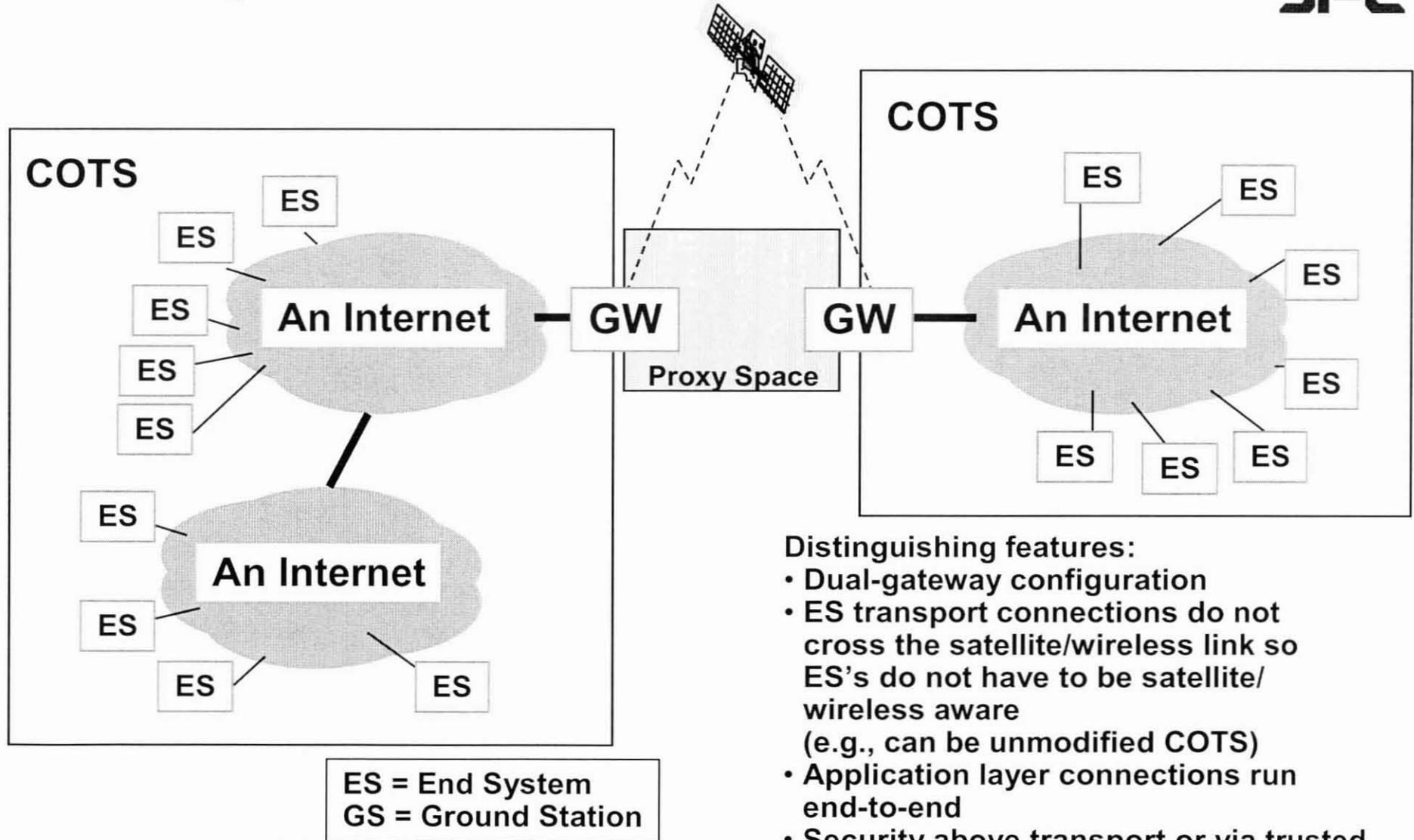
Results to date (Cont.)



■ Products

- Refinements to commercial Internet implementations for satellite paths resulting from ACTS testing
- Open standard SCPS protocols and proxies
- SkipWare and XipLink

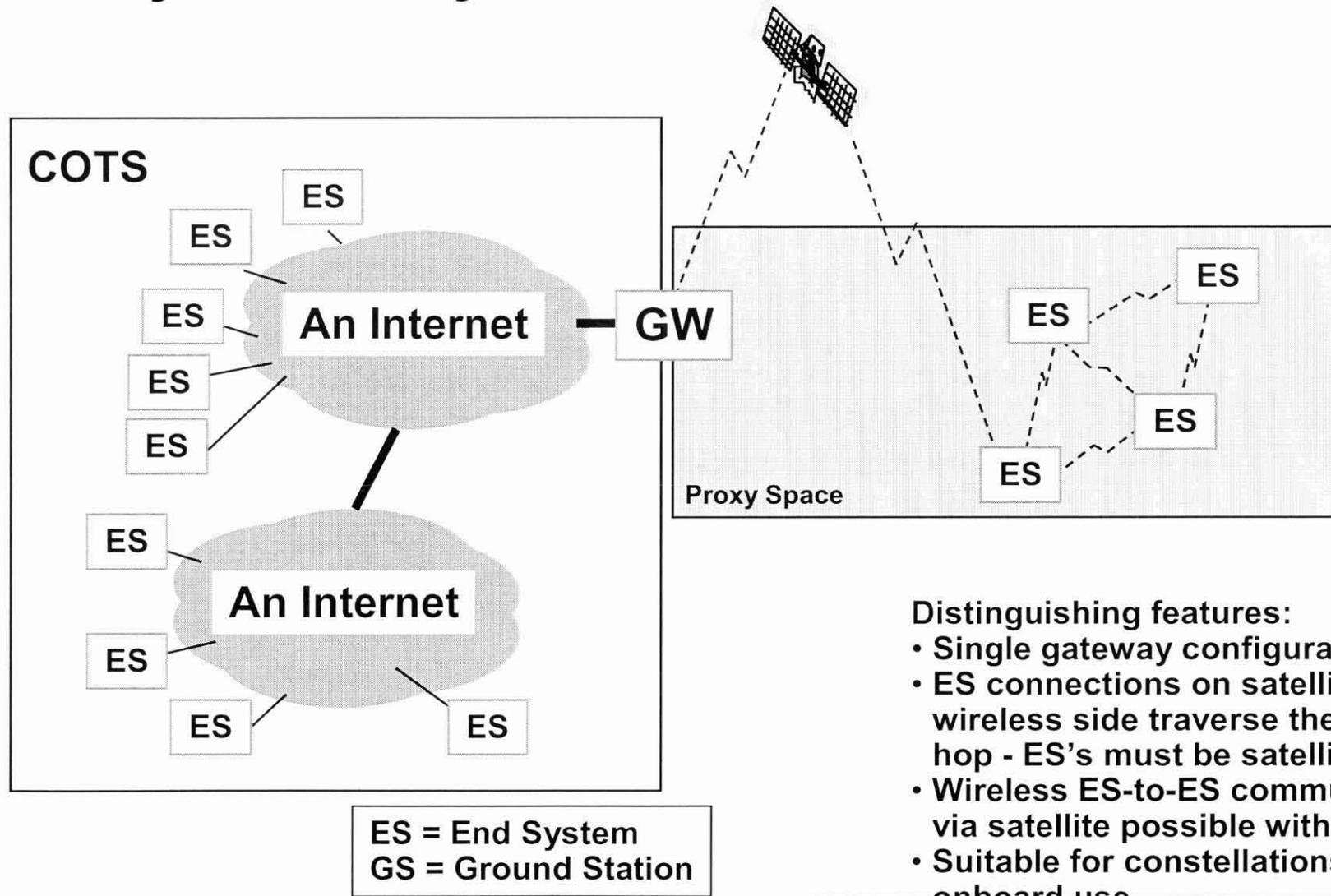
Architectural Alternative: Proxy-Based Architecture



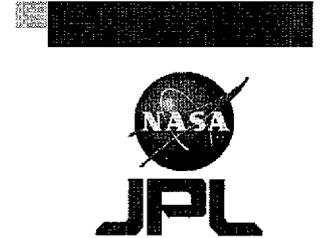
Distinguishing features:

- Dual-gateway configuration
- ES transport connections do not cross the satellite/wireless link so ES's do not have to be satellite/wireless aware (e.g., can be unmodified COTS)
- Application layer connections run end-to-end
- Security above transport or via trusted gateways

Architectural Alternative: Hybrid Proxy Architecture

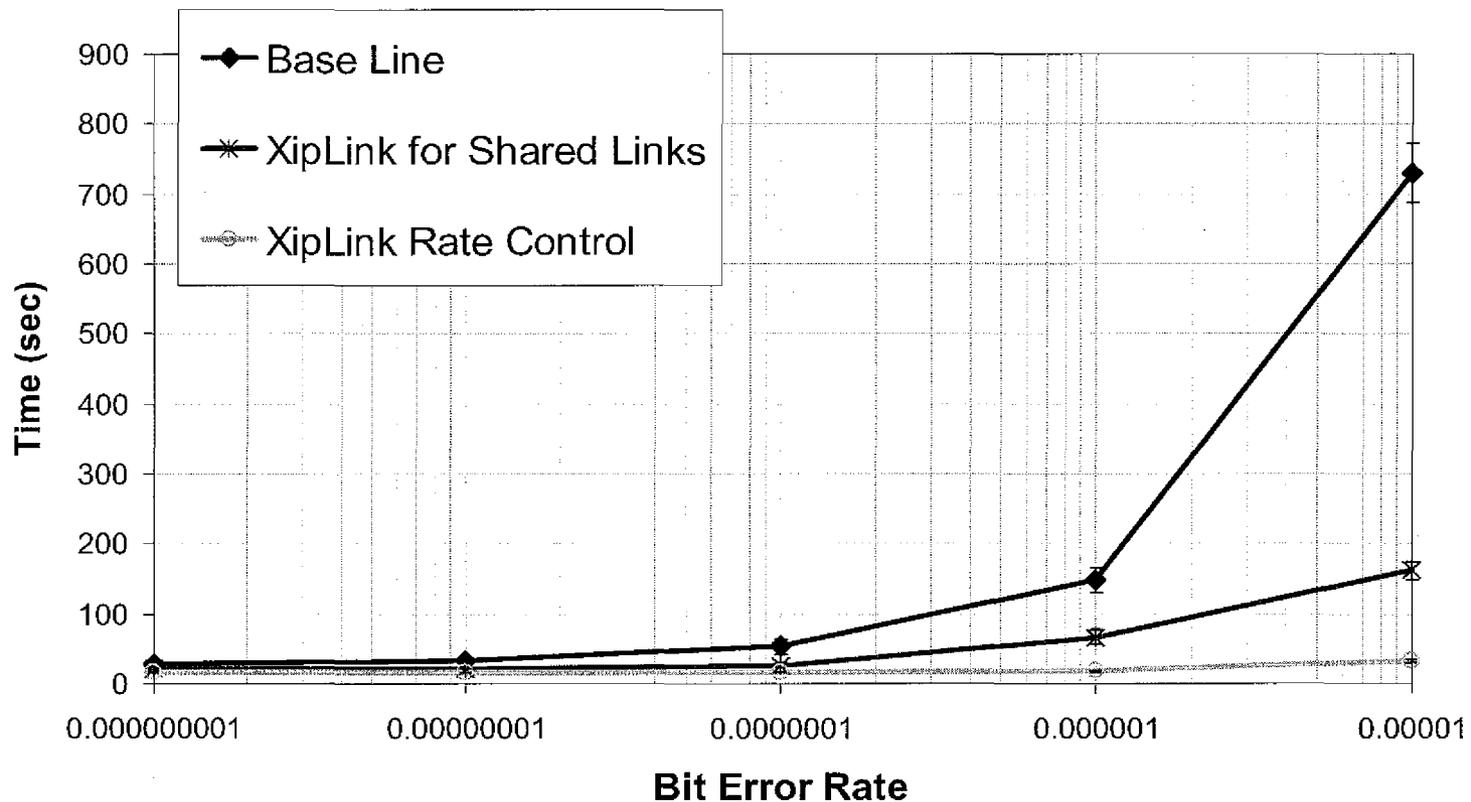


- Distinguishing features:
- Single gateway configuration
 - ES connections on satellite/ wireless side traverse the satellite hop - ES's must be satellite aware
 - Wireless ES-to-ES communication via satellite possible without GW's
 - Suitable for constellations or onboard use



Sample SCPS PEP Performance

Time to Transfer a 3MB over a 525 ms RTT T1 Link versus Bit Error Rate

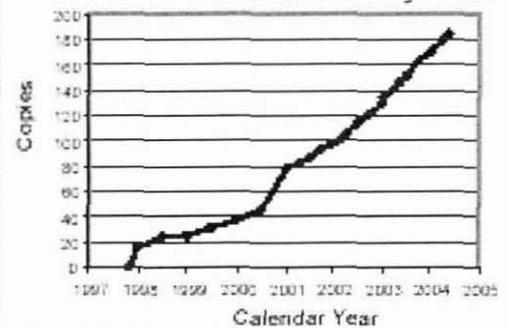


MILSATCOM

NPOM

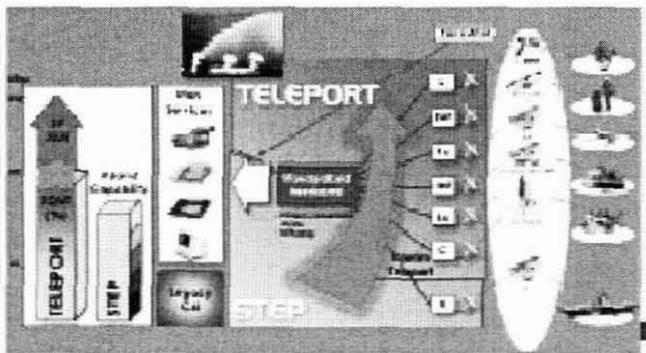
Network Protocols over MILSATCOM

SCPS RI Distribution by Date



Space Communications Protocol Standards (SCPS) Approval

The Theater Joint Tactical Networks Configuration Control Board (TJTN-CCB) and Joint Staff have approved the Space Communications Protocol Standards—Transport Protocol (SCPS-TP) for Standardized Tactical Entry Point (STEP) Teleport sites. SCPS-TP is one of several solutions to the Transmission Control Protocol (TCP) over satellite problem. In dealing with this problem, the services investigated and implemented several solutions. These solutions, however, were largely proprietary, so a need developed to investigate and determine the best solution. SCPS-TP was shown to be the most effective and interoperable solution for TCP enhancements, according to extensive Joint Terminal Engineering Office (JTECO) testing performed for the 2002 Network Protocols over MILSATCOM (NPOM) working group. The SCPS-TP protocol was tested at both the Joint User Interoperability Communications Exercise (JUICE) and the Department of Defense Interoperability Communications Exercise (DICE). During both exercises, the SCPS-TP protocol performed well and demonstrated both TCP traffic enhancement capability and interoperability with other TCP devices. As a result of these efforts, the TJTN-CCB voted to standardize on SCPS-TP-enabled devices for the STEP/Teleport Program; therefore SCPS-TP-enabled TCP accelerators will be installed at each STEP site beginning in Fall 2003.

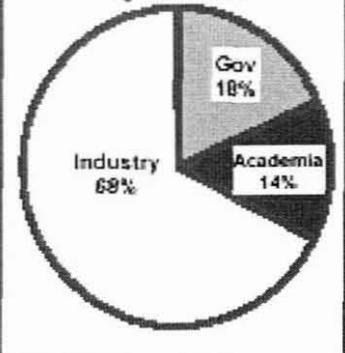


In 1991, a joint effort began among the National Aeronautics and Space Administration (NASA), the United States Strategic Command (USSTRATCOM) and the Jet Propulsion Labs (JPL) to develop an interoperable suite of end-to-end data protocols for satellite networks. Recognizing that there was a problem using TCP over high delay, high bit error rate links, a team of protocol engineers, drawn from the satellite community, started developing internet over-satellite standards. This work resulted in the SCPS protocol suite, which not only is based on Internet protocols, but is also fully interoperable with Internet Engineering Task Force (IETF) standards.

Joint
NASA-DOD
SCPS Project
1991-1999

The SCPS consortium looked to establish the SCPS protocol suite as an international standard for the space networking community. The International Standards Organization (ISO), the Consultative Committee for Space Data Systems (CCSDS), and the U.S. Department of Defense (DoD) have adopted the SCPS protocol suite as a standard. The different organizations and their corresponding SCPS standards numbers are listed in Table 1.

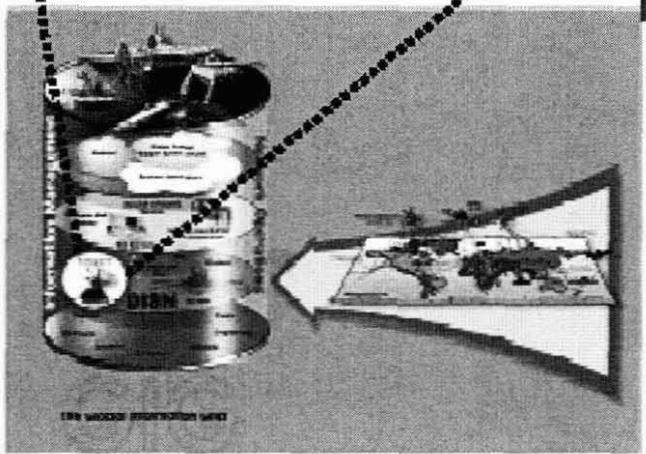
SCPS RI Distribution, by Sector



The SCPS Standards			
SCPS Protocol	MILSATCOM	CCSDS	ISO
SCPS File Protocol	MIL-STD-2045-47000	717.0-B	15894
SCPS Transport Protocol	MIL-STD-2045-44000	714.0-B	15893
SCPS Security Protocol	MIL-STD-2045-43001	713.5-B-1	15892
SCPS Network Protocol	MIL-STD-2045-43000	713.0-B	15891

Table 1. SCPS Standards

Some vendors that have already implemented SCPS are: Comtech TurboIP, Linux VPN Router w/ Skyware and the Nphos NPLink Mini Gateway, with more on the way.



CCSDS: The Fleet

Space Domain

Spacecraft Platforms
On-Board Systems
Space Qualified ASICs

CCSDS
Consultative Committee for Space Data Systems

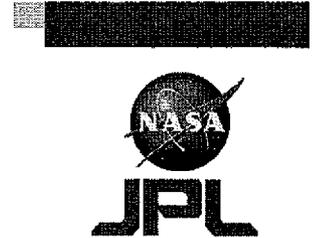
**300 Missions now using
CCSDS Space Link Protocols**
<http://www.ccsds.org/CCSDS/missions.jsp>

Ground Domain

Commercial Ground Networks
Command & Telemetry Data Processing

Shuttle CS Gateway

ULDS



Options to Improve TCP Performance in 'Stressed' Environments

- **Change the TCP stacks on the end systems**
 - Tune 'stock' TCP parameters
 - http://www.psc.edu/networking/perf_tune.html
 - TCP Modifications
 - TCP Westwood
 - TCP Peach
 - TCP Tranquility (SCPS-TP)
 - Issues
 - Fairness – If the modified TCP will share resources (network) with existing 'stock' TCPs, will it starve out the stock TCPs?
 - Tuning – TCP tuned for satellite channel is likely de-tuned for LAN
 - Maintenance – have to touch every end system (both ends) Very difficult to modify the TCP stack in, say, Windows OS
- **Put something in the middle of the network to improve TCP performance (PEPs – Performance Enhancing Proxies)**
 - Mentat Sky-X gateways
 - TCP Tranquility (SCPS-TP) gateways



Tuning 'Stock' TCP Implementations

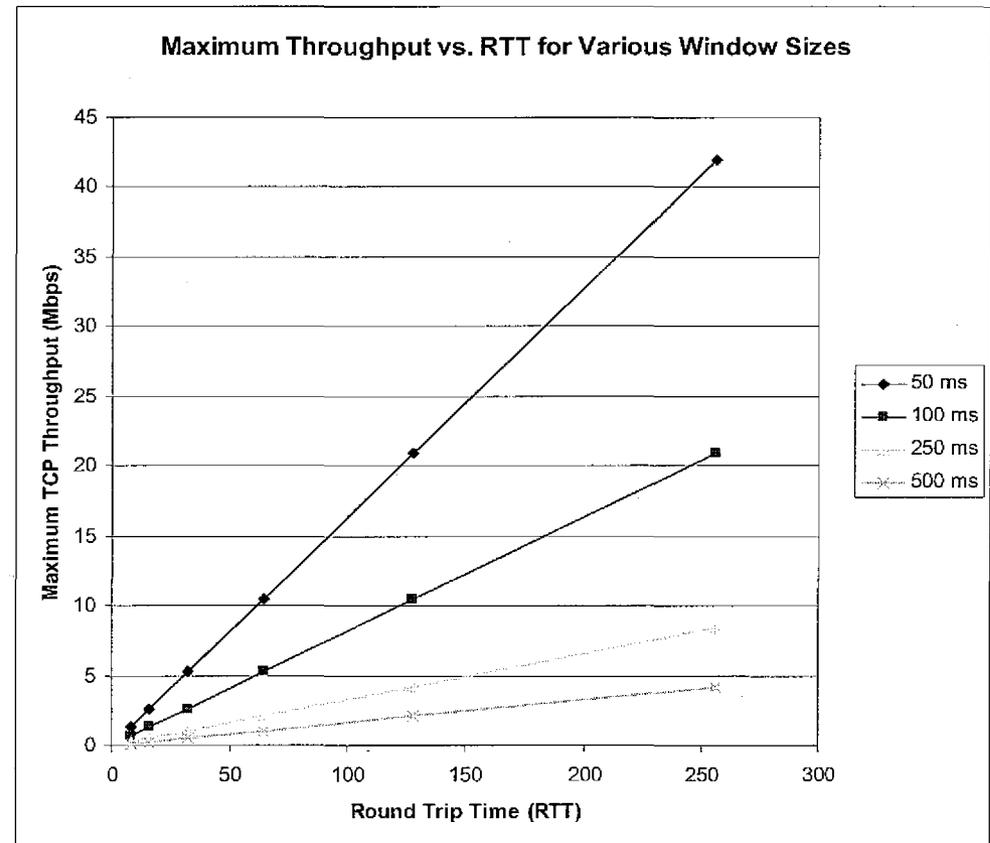
http://www.psc.edu/networking/perf_tune.html

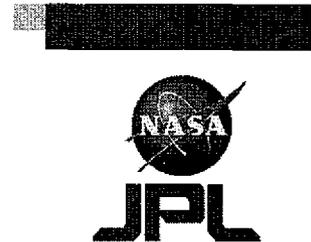
- Increase the amount of data that can be outstanding
 - Increase the receiver's buffer size
 - RFC 1323: allow TCP window size to be scaled past 64kB limit
 - Without this, the size of the receiver's buffer will impose an artificial limit on the transmission rate
- RFC2018 – selective acknowledgements (SACK)
 - Allows receiver to tell the sender exactly what has been received
 - Better recovery when multiple segments are lost in a window
- Header compression
 - Reduces the header overhead from TCP+IP from ~40 bytes to ~3-5 bytes for most cases
 - Requires state at both ends of the link – the loss of any packet will cause all future packets to become un-decodable
 - The sending TCP will have to time out and retransmit the lost packet. This is a very expensive operation



Performance of 'Stock' TCP vs. TCP Tuned for the Connection

- Increasing TCP's window size to 'fill the pipe' can dramatically increase performance
 - It's best if the window size can be set to the *bandwidth*delay product of the channel*
 - This is difficult in practice because
 - There's rarely ONE RTT value (consider communicating either over a satellite or over a LAN with a machine next to you)
 - To achieve the best performance, needs to be changed depending on the number of connections sharing the network





'Alternate' TCPs In End Systems

■ TCP Westwood

- Uses the spacing of acknowledgements to infer the amount of bandwidth available to the connection
- Implementation for FreeBSD 4.4 available

■ TCP Peach

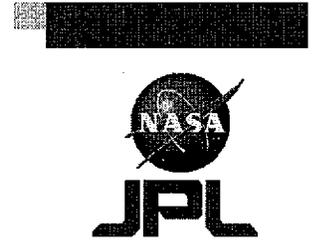
- Attempts to differentiate between losses caused by congestion and those caused by bit errors
- Approach: Send 'dummy' packets at a lower priority
 - If loss is due to congestion, lots of these will be dropped in addition to the 'real' data
 - If loss is due to corruption, few dummy packets will be lost – responds by NOT cutting transmission rate
 - Requires routers in the network to support multiple IP priorities
- Implementation for Linux 2.2 kernel available

'Alternate' TCPs in End Systems (cont'd)



■ TCP Tranquility (SCPS-TP)

- Standard TCP Options (IANA assigned TCP option #s 20—23)
 - SCPS Capabilities – used to negotiate willingness to use SCPS capabilities
 - Selective Negative Acknowledgements (SNACKs) – reduced ack traffic after loss (good for asymmetric channels)
 - End-to-End Header compression that is robust against packet loss -- packets after a lost packet can still be received
 - Record boundary indications – allows applications to delimit record boundaries within the TCP stream
 - Corruption Experienced Option – Assists in signaling network corruption / outage events
- Sender-side configuration options
 - Can use different congestion control mechanisms that are less sensitive to loss but still protect the network
 - Includes VJ congestion control, variant of TCP Vegas congestion control, and no congestion control options
 - Vegas option can use only queueing as congestion indication – *insensitive to loss*
 - Sender can limit sending rate independently of congestion control – prevents sender from constantly increasing transmission rate until loss
- Reference Implementation (including PEP) freely available



How TCP Works (1 slide version)

- **TCP is a reliable protocol**
 - Receiver acknowledges data received; sender retransmits anything that is lost
- **Window-based flow control**
 - Receiver specifies how much data can be in flight
 - Sender can send this much and then has to stop until data is acknowledged
- **TCP congestion control (Van Jacobson)**
 - Added to combat 'congestion collapses' during the late 1980s
 - TCP sender detects that packets have been lost and responds by cutting its transmission rate in half
 - Assumption is that all loss is an indication of network congestion
 - After cutting transmission rate, TCP will increase its transmission rate by 1 segment per round trip
 - TCP will keep increasing its rate until it *causes* another loss
- **TCP 'time unit' is round trip time**
 - Large RTT -> slow recovery after loss