

Internet Technology on Spacecraft

James Rash
Goddard Space Flight Center
Greenbelt, MD 20771 (301) 286-5246 James.Rash@gsfc.nasa.gov

Ron Parise, Keith Hogie, Ed Criscuolo, Jim Langston
Computer Sciences Corp (CSC)
7700 Hubble Dr. Lanham-Seabrook, MD 20706 (301) 286-3203
Ron.Parise@gsfc.nasa.gov, Keith.Hogie@gsfc.nasa.gov, Ed.Criscuolo@gsfc.nasa.gov, jlangsto@csc.com

Abstract. The Operating Missions as Nodes on the Internet (OMNI) project has shown that Internet technology works in space missions through a demonstration using the UoSAT-12 spacecraft. An Internet Protocol (IP) stack was installed on the orbiting UoSAT-12 spacecraft and tests were run to demonstrate Internet connectivity and measure performance. This also forms the basis for demonstrating subsequent scenarios. This approach provides capabilities heretofore either too expensive or simply not feasible such as reconfiguration on orbit. The OMNI project recognized the need to reduce the risk perceived by mission managers and did this with a multi-phase strategy. In the initial phase, the concepts were implemented in a prototype system that includes space similar components communicating over the TDRS (space network) and the terrestrial Internet. The demonstration system includes a simulated spacecraft with sample instruments. Over 25 demonstrations have been given to mission and project managers, National Aeronautics and Space Administration (NASA), Department of Defense (DoD), contractor technologists and other decisions makers. This initial phase reached a high point with an OMNI demonstration given from a booth at the Johnson Space Center (JSC) Inspection Day 99 exhibition. The proof to mission managers is provided during this second phase with year 2000 accomplishments: testing the use of Internet technologies onboard an actual spacecraft. This was done with a series of tests performed using the UoSAT-12 spacecraft. This spacecraft was reconfigured on orbit at very low cost. The total period between concept and the first tests was only 6 months! On board software was modified to add an IP stack to support basic IP communications. Also added was support for ping, traceroute and network timing protocol (NTP) tests. These tests show that basic Internet functionality can be used onboard spacecraft. The performance of data was measured to show no degradation from current approaches. The cost to implement is much less than current approaches due to the availability of highly reliable and standard Internet tools. Use of standard Internet applications onboard reduces the risk of obsolescence inherent in custom protocols due to extremely wide use across all domains. These basic building blocks provide the framework for building onboard software to support direct user communication with payloads including payload control. Other benefits are payload to payload communication from dissimilar spacecraft, constellations of spacecraft, and reconfigurability on orbit. This work is funded through contract with the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC).

Introduction

This paper will discuss the use of standard Internet applications and routing protocols to meet the technology challenge of providing dynamic communication among heterogeneous instruments, spacecraft, ground stations, and constellations of

spacecraft. The objective is to characterize typical mission functions and automated end-to-end transport of data in a dynamic multi-spacecraft environment using off-the-shelf, low-cost, commodity-level standards. These functions and capabilities will become increasingly significant in the years to come as both Earth and space science missions fly more and more sensors and the present labor-intensive, mission-specific techniques for processing and routing data become prohibitively expensive. This work is about defining an architecture that allows science missions to be deployed "faster, better, and cheaper" by using the technologies that have been extremely successful in today's Internet.

Copyright © 2000 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental Purposes. All other rights are reserved by the copyright owner.

Overview of Internet Protocols in Space

The goal of the OMNI project is to define and demonstrate an end-to-end communication architecture for future space missions. The authors have combined their knowledge and experience in Internet technologies and space communication systems in developing the following end-to-end data communication concept.

End-to-End Network Concept

The data communication requirements of many advanced space missions involve seamless, transparent connectivity between space-based instruments, investigators, ground-based instruments and other spacecraft. The key to an architecture that can satisfy these requirements is the use of the Internet Protocol¹ (IP). IP is the technology that drives the public Internet and therefore draws billions annually in research and development funds. Most private networks also utilize IP as their underlying protocol. IP provides a basic standardized mechanism for end-to-end communication between applications across a network. The protocol provides for automated routing of data through any number of intermediate network nodes without affecting the endpoints.

Spacecraft environments pose numerous challenges that have direct analogs in the ground-based mobile IP and wireless networking industries, such as:

- continually intermittent links
- multiple mobile nodes forming a dynamic network topology
- maintaining a single address for a spacecraft as it uses different ground stations
- highly asymmetric or unidirectional communication links

The increasing popularity of laptop computers, handheld digital assistants, and Internet cell phones has driven the development of protocols to handle mobile nodes, such as Mobile IP² (MIP) and mobile routing. They are also driving the development of new protocols such as Cellular IP³, Dynamic Source Routing⁴ (DSR), and other ad-hoc-networking protocols.

This paper will examine several standard Internet applications and protocols specified by the Internet Engineering Task Force (IETF), and map them to spacecraft applications. It will also describe how standard, commercially available communication

hardware and software was used to test some of these concepts with an orbiting spacecraft.

Benefits

Increasingly, future space missions featuring multiple spacecraft are being proposed. This includes constellations (disjoint or formation-flying), sensor-webs⁵ of heterogeneous spacecraft, and collaborative science between spacecraft belonging to different missions or with ground-based sensors. As the number of spacecraft involved increases, the number of possible end-to-end routes for data goes up geometrically. The present methods utilizing manual data routing, non-interoperable protocol options, and custom data handling applications are not scalable and rapidly become unaffordable due to the large amount of manpower and custom development required.

Using standard Internet protocols will allow robust, dynamic, and automated end-to-end data delivery. Using standard Internet applications, such as FTP⁶, will allow exchange of data without designing custom data handling and translation software. Also, custom applications that require direct interaction between space-based and ground-based components are easily implemented through a familiar interface. These factors will reduce the risk and development time for space missions, increase the accessibility of science data, and enable cost-effective realization of new science capabilities such as:

- Data exchange directly between spacecraft
- Correlation of data in space
- Collaborative science among unrelated sensors within and across missions
- Easy reconfiguration and deployment of new types of collaborative science across multiple missions
- Direct access to science payloads and data by principal investigators or collaborators

Along with these new capabilities, significant benefits are envisioned across all phases of a mission life-cycle. Significant cost savings and risk reduction are achievable in all the following areas:

Mission Design

Mission designers are currently constrained by the narrow range of services provided by the current system. Any deviation from the basic telemetry and telecommand services immediately drives the cost up by requiring added software development to provide

functionality already present in the IP network model. Mission design will be faster and simpler using industry standard communication interfaces. This will allow designers to spend less time doing data communication system design and focus more on the specific science and mission details. Minimizing mission specific interface control documents (ICDs) will also expedite design and reduce costs as well as the risk of errors.

Software Development

Most software engineers are already experienced in designing applications that utilize Internet protocols for communications thereby reducing or eliminating the learning curve required for custom protocols. Software design and development will be able to take advantage of the large base of software and services already defined, documented, implemented, and tested in commercially available operating systems and applications. Custom on-board applications will be developed and tested more quickly using familiar interfaces.

Hardware Design

There are currently several on-going projects to develop flight quality versions of standard network components. These components will facilitate a building block approach to spacecraft system design. All subsystems can have a standard interface and can be prototyped and built independently without fear of incompatibilities later in the overall development.

Integration and Test

Current integration and test procedures require the use of specialized hardware and software to simulate the interfaces to the spacecraft. Using a common suite of standard communication interfaces for spacecraft subsystems will allow laboratory testing of the subsystems via the same interfaces that will be used in flight. The development of radiation hardened, standard local area network components will facilitate all aspects of the integration and test phases. Using Internet technology as a common communication mechanism from spacecraft subsystems to the end user will allow very early functional testing of subsystems to identify problems long before traditional integration and test. Catching problems earlier can provide significant cost savings and risk reduction.

Operations

All of the advantages from the earlier mission phases also carry over into the mission operation phase. Using the same communication interfaces from initial design and development through operation allows the same monitoring and control software and knowledge bases to be used across the entire mission life-cycle. This avoids developing and testing separate interfaces and software systems for initial development, integration testing, and operation.

Security

Any suggestion of the use of public networks for data distribution, or common data transmission protocols on the space-to-ground link, naturally generate questions regarding security. But this is also a serious concern for businesses that use public networks for financial transactions. These businesses safely transfer trillions of dollars per day over public networks using commercially developed security solutions. The requirements of business users such as these have spurred the development of a suite of IP based security protocols such as IPsec⁷, virtual private networks⁸ (VPN), and secure socket layer (SSL).

There are three basic tools available to provide a secure link on both the command and telemetry data path between the spacecraft and the user. These are:

Authentication

Authentication provides a means of ensuring that data packets received by the user are indeed originating from the desired address (i.e. spacecraft or instrument). This is commonly accomplished through virtual private network technology, which is widely available for all platforms.

Encryption

Encryption of data in the IP packets can protect the data from discovery by unauthorized persons. A wide variety of commercial encryption packages are available to provide this capability.

Private Networks

Private networks can provide additional security by eliminating outside access to the data stream. NASA currently uses a closed network between all of its ground stations and centers and can provide secure

connections to this network to outside investigators at universities and other institutions.

All of the above tools can be employed individually or in any combination to satisfy the mission specific security requirements. Authentication and encryption can be implemented either from the user to the groundstation, or from the user all the way to the spacecraft depending on the mission requirements and the availability of on-board computing resources. They can be used on the uplink, downlink, or both, and in conjunction with a private network provide an extremely secure system.

IP-Based Operations Scenarios

An IP-based communication architecture can support all existing operations concepts and makes some new, complex concepts realistic.

For example, real-time engineering and housekeeping data can be monitored during a pass using UDP/IP packets. This only requires a uni-directional downlink. On the other hand, if a return link is available, guaranteed reliable delivery of data packets can be achieved by using TCP/IP. In both cases, only a simple application layer function needs to be written for the flight software.

Recorded science and engineering data can be stored onboard in files in a standard file system. These files can later be transferred to the ground with guaranteed complete, time-ordered records using an off-the-shelf application such as FTP. If the round-trip delay times are too great to use a TCP-based protocol such as FTP, a UDP-based protocol, such as Starburst/MFTP, could be used instead.

Onboard clock synchronization, typically a thorny problem, can be handled by using the Network Time Protocol⁹ (NTP). NTP can automatically calculate propagation delay times, time variance, and drift rates, relative to one or more reference timeservers. It can set the spacecraft's clock and even perform periodic drift mitigation. The NTP protocol is capable of precision on the order of 240 picoseconds *if* sufficient bandwidth and CPU speed are available.

Store-and-forward commanding, and data delivery, can be achieved by using Simple Mail Transport Protocol¹⁰

(SMTP) to deliver the files as attachments. For example, in the commanding case, the control center emails the command load to the spacecraft as an attachment. A mail server at the groundstation stores the file until the next contact and then automatically transfers it to the spacecraft for processing. Similarly, in the data delivery case, the C&DH processor, or even a "smart" instrument, emails the data file as an attachment to a message sent to the principal investigator. The onboard mail server stores the file until the next contact and then transfers it to the ground for automatic delivery to the owner of the data.

The IP suite supports various commanding scenarios. When the downlink is not available for acknowledgement, "blind" real-time commands can be sent to the spacecraft using UDP. This is required for emergency situations such as rescuing a spacecraft from tumbling. For nominal operations, reliable commanding can be achieved by using TCP, which automatically takes care of performing the handshaking and any necessary retransmissions.

These basic capabilities, and the end-to-end network addressing capability of IP, can be used to support new, complex scenarios. These include user interaction, spacecraft cross-support, and ad-hoc collaborations.

These scenarios also highlight the capabilities needed for constellations of spacecraft. Formation flyers could send messages back and forth to keep their group navigation within specifications. Constellations of nanosats could message their data to larger members with the power for delivery to the ground.

Proposed Architecture

Recognizing the clear benefits of IP as an end-to-end networking protocol, the OMNI project developed a reference system architecture for the space and ground segments of future IP missions. The goals were to maximize the use of COTS hardware and protocols while avoiding creating any new "space-specific" solutions. A high-level view of this architecture appears in figure 1. Several notable features are present, and are discussed in the following sections.

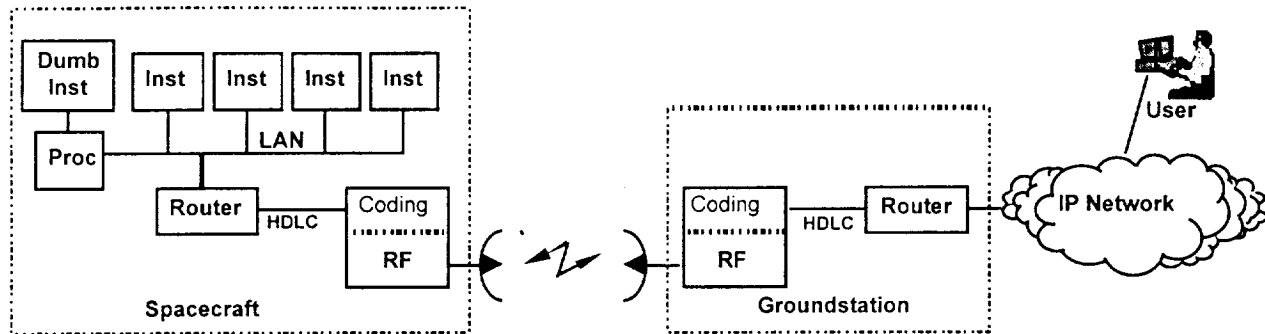


Figure 1 – System Architecture for an IP Mission

Onboard Spacecraft IP LAN

One of the key features of this architecture is the incorporation of an IP stack in the onboard processor, and the use of peer-to-peer IP networking via an onboard LAN. This allows end-to-end network addressing, distributed processing and “smart” instruments, while providing for legacy processor-controlled “dumb” instruments. The processor, each “smart” instrument, and the router each have their own IP address on the LAN, and are directly reachable from any node on the network. The router takes care of delivering packets to the appropriate address without processor supervision. This is in contrast to the conventional master-slave architecture of a typical 1553 bus spacecraft, where the processor must be responsible for all bus traffic by managing the bus time-slicing in real time. Candidate LANs include CAN, Ethernet, IEEE-1355, and IEEE-1394 (Firewire).

Spacecraft Router Function

At its basic functional level, the router performs simple conversion from one link-level interface to another. For example, the spacecraft router in figure 1 could be converting HDLC¹¹ framing on a serial link into Ethernet framing on a LAN. In small, simple spacecraft, there may be only a single “dumb” processor-controlled instrument. In these cases, a LAN would not be needed, and the spacecraft would have a single IP address for the processor. The remaining router functions could then be performed in software on the processor. This configuration minimizes costs while still retaining the benefits of end-to-end IP networking.

HDLC Framing

Based on its near-universal use on the terrestrial Internet, HDLC framing was chosen for the link-level protocol on space-to-ground links. This allows simple, straightforward interfacing with existing commercial routers in the groundstation. Interoperability between multiple vendors is assured by choosing the IETF encapsulation for multi-protocol over frame-relay/HDLC.

At the physical layer, HDLC framing is extremely simple, consisting of only a 1-byte flag pattern, a variable number of data bytes, and a 2-byte CRC. During any idle time, successive flag bytes are output until the next frame begins. Flag bytes consist of a zero bit, 6 one bits, and a zero bit. In order to prevent this pattern from occurring in the data, the HDLC hardware performs “bit stuffing” when sending data. Any sequence of 5 one bits in the data automatically has a zero bit inserted after it, thus insuring that any sequence of 6 consecutive one bits *must* be a flag byte. When receiving, these extra zero bits are automatically removed from the data.

While the primary purpose of “bit stuffing” is to ensure the uniqueness of the flag byte, it also has an additional benefit. It ensures that a long unbroken sequence of one bits in the data does not produce a signal to the transmitter that does not have periodic transitions. These periodic transitions are important at the receiver, where a bit-synchronizer depends on them to extract the clock and data bitstreams from the raw signal. Along the same lines, the use of standard NRZI coding for the HDLC output will insure that an unbroken sequence of zero bits in the data stream becomes transformed into an alternating sequence of

ones and zeros. Thus, the use of "bit stuffing", idle flag bytes, and NRZI coding insures that the transmitter will never send an unmodulated carrier, and the receiver will see a transition *at least* once every 6 bit times. It is important to note that these "space specific" requirements can be met by standard COTS hardware and protocols without inventing any "space specific" solutions. It should be further noted that these solutions are isolated to the lowest layer and are transparent to the upper layers. The application layer does not have to concern itself with generating "fill packets" or "fill frames".

Separation of Framing and Coding

It is important to note that any forward error correction (FEC) coding, such as Reed-Solomon or convolutional, is performed *independently* from the framing. This is in accordance with the OSI layered model of networking, where framing is carried on at the data link layer and coding is down at the physical layer. The coding simply treats the HDLC data as a bit-stream to be protected.

Ground-Based Demonstrations

In late 1998, the OMNI project began constructing a "proof of concept" ground-based prototype of an IP spacecraft. An instance of the reference architecture was constructed using a PowerPC processor and the VxWorks operating system. These are both commonly used in actual flight missions. VxWorks comes with a standard IP stack available. An Ethernet-based camera server was used as a "smart" instrument, and a weather station and a GPS receiver were used as serial-port based "dumb" instruments. A small Cisco 1601 router initially handled the interface between the Ethernet LAN and the HDLC serial link to the RF gear. Later, it was replaced by a 1 1/2" x 1/2" x 2" TinyRouter. In the lab, the RF gear was replaced with an AdTech Data Channel Simulator to provide settable delays and error rates. Later, the RF link was provided by an ECOMM transceiver, going through the NASA Tracking and Data Relay Satellite System (TDRSS) in geosynchronous orbit. At the TDRSS groundstation in White Sands, NM, the serial data lines to and from the transmitter and receiver were connected to a serial port on a router already connected to the Internet by the South Pole TDRSS Relay (SPTR) project.

In February of 1999, we mounted the entire IP spacecraft prototype in a Plymouth Voyager minivan, and began conducting mobile demos, flowing real-time

telemetry and image data in UDP/IP from our "Voyager spacecraft", via TDRSS, to a "mission operations center" at Goddard Space Flight Center. Figure 2 shows our "spacecraft" in a very low "parking" orbit. Note the autopointing S-band antenna on the roof and the weather station on the rear. Subsequent tests over the next few months demonstrated the utility of a wide variety of protocols and applications, such as FTP, NTP, NFS¹², and http, in spacecraft operations. Scenarios involving intermittent contact, station handover, and one-way links were all successfully demonstrated.



Figure 2 – The OMNI "Spacecraft"

In August of 1999, we constructed a duplicate system and used it to return instrument data and images from the deck of a ship in the Black Sea that was observing the last total solar eclipse of the millennium. The data was webcast in real time and viewed by thousands. In addition, the system provided SMTP (e-mail) and Voice-Over-IP capabilities.

In November of 1999, the van-mounted system provided support for the Johnson Space Center (JSC) Inspection Day '99 Exhibition, providing real-time images and two-way voice from the van to visitors at the display booth at JSC.

Space-Based Demonstration

The OMNI project had been looking for opportunities to demonstrate IP in space for the last year. However, many of the spacecraft candidates were deemed unsuitable due primarily to their onboard communication hardware. The key issue was to find a spacecraft that could support HDLC framing in hardware to allow simple, straightforward interfacing

with existing commercial routers. These requirements made UoSAT-12 (figure 3) an ideal test platform, as it already used HDLC framing to carry its AX.25 protocol. Since HDLC I/O hardware was already present on-board, only flight software changes would be required to adapt UoSAT-12 to use IP. Changes to the ground station would also be minimal, requiring only the addition of a standard commercial router and a programmable switch. See figure 4.

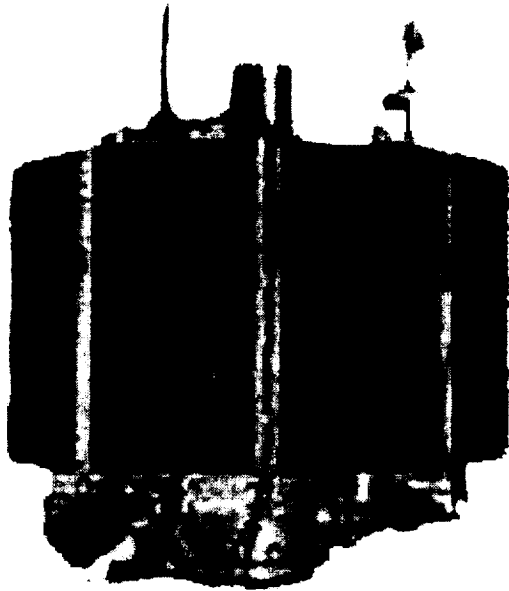


Figure 3 – UoSAT-12 Spacecraft

In December 1999, the OMNI project contractor, Computer Sciences Corp. (CSC), began negotiations to have an IP stack ported to one of the spacecraft's onboard processors, using HDLC/Frame-Relay for the link-layer protocol over the RF communication system. This would allow the ground station to directly connect the receiver to a standard low-cost router, providing end-to-end IP connectivity between an IP address on the spacecraft and any node on the Internet. Further discussions covered porting File Transfer Protocol (FTP) and Network Time Protocol (NTP) to the spacecraft's operating system.

In February 2000, CSC let an initial contract to VyTek Wireless of Pittsburgh, PA, formerly VyTek LLC, to supply the software porting, spacecraft time, and ground-station time for a series of flight tests. Initial tests of IP connectivity were scheduled for early April 2000, with tests of spacecraft clock synchronization, reliable file transfer, and blind commanding to follow in May/June 2000.

Follow-on work is planned to demonstrate http file delivery, mobile IP, security, and store-and-forward commanding and data delivery using Simple Mail Transfer Protocol (SMTP). These tests are expected to be performed in 3Q/4Q 2000.

Spacecraft Implementation

Since the UoSAT-12 spacecraft was already in orbit, only software modifications were possible. Because the spacecraft was built to be a flexible prototype test environment, it was straightforward to upload and test new software to support IP over HDLC.

Ground Station Implementation

Since the SSTL ground station already supported HDLC framing, a standard Internet router was the only addition needed. Figure 4 indicates the basic components of the ground station and where the router was added in parallel with the existing AX.25 communication front-end.

The only station reconfiguration required is to select which system is connected to the transmitter. This is done with a controllable switch, which supports fully automated passes for either the IP or AX.25 mode.

The SSTL ground station is built on an Ethernet LAN with firewalls and router connectivity to the Internet. Two addresses were used on the ground station LAN to support these tests. One address was used for the Ethernet interface on the router and the other address was assigned to the spacecraft.

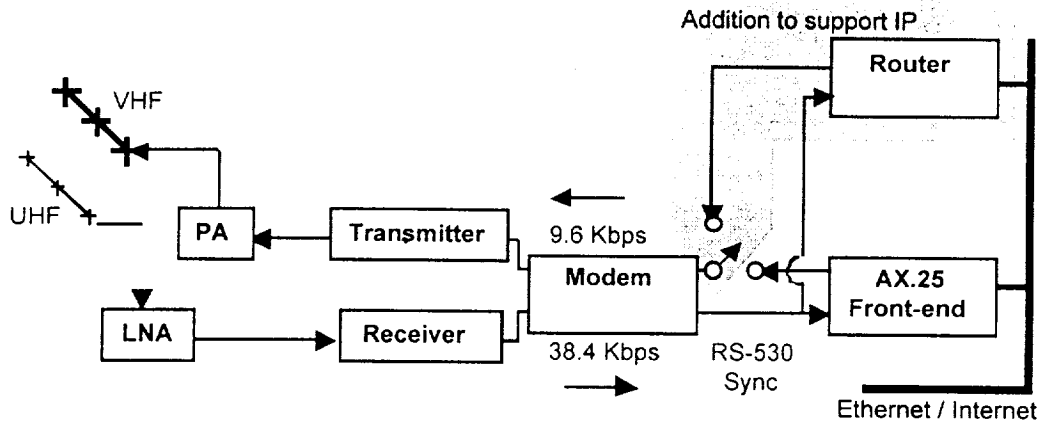


Figure 4 - SSTL ground station configuration

Flight Tests

The initial IP tests with UoSAT-12 were designed to verify the operation of standard Internet packet routing from the spacecraft's operating system to anywhere on the Internet. The "ping" utility was used to accomplish this and to characterize the basic link propagation delay. These tests were successfully completed on April 11, 2000, and are covered in more detail in a paper presented at the Small Satellite 2000 conference¹³. We believe this to be the first instance of a spacecraft as a fully RFC-compliant node on the Internet.

Subsequent tests were designed to demonstrate implementing clock synchronization (a standard operational spacecraft function), using standard IP-based applications. For the clock synchronization tests, a standard NTP client was ported to the UoSAT-12 spacecraft. It was used to automatically synchronize the onboard clock to UTC. On the ground, the US Naval Observatory's timeserver (tick.usno.navy.mil) was used as the reference timeserver. This configuration is shown in figure 5. This represents somewhat of a worst-case test, as the USNO is a quarter of the way around the world, over 20 router hops, from the UoSAT-12 ground station in Surrey, UK. In a real operations environment, a timeserver of the required accuracy would be located at the groundstation to minimize the network latency and variation that NTP has to factor out. However, NTP is designed to deal with these factors, and the resulting levels of accuracy might be quite adequate for many space missions even under worst case conditions.

Two tests were performed, both following the same scenario. The test starts out with the onboard NTP

server running, but disabled from actually changing the spacecraft's clock. The onboard server periodically negotiates with the USNO timeserver to factor out network delay. If it is successful, the onboard server calculates the offset it thinks it has to apply to the spacecraft's clock. This value is sent to the ground in a UDP telemetry stream, where it is logged for later analysis. For purposes of this testing, the NTP negotiation period is set artificially low to 30 seconds so that a reasonable number of data points can be collected during the 14 minute pass. A short time into the test, a command is sent to the spacecraft to enable NTP to actually change the onboard clock. NTP will require two successful offset calculations before it will adjust the clock. Later in the test, a command is sent to the spacecraft to manually set the onboard clock in error by a large amount (2-3 seconds). After two successful offset calculations, NTP should again reset the clock. If the time is off by more than one second, the spacecraft NTP client adjusts to the proper second during one adjustment period, and adjusts for fractional seconds on the next adjustment period.

The results for the test run on April 14, 2000 are shown in figure 8. The pass began with a spacecraft clock offset from UTC of approximately +300 ms. Two calculations after NTP time-changing was enabled, the calculated offset dropped to less than two clock ticks (20 ms) and stayed there until it was manually set in error from the ground. A ground command was used to set the clock ahead by approximately 3.25 seconds. Two offset calculations after that, NTP had reset the clock to within six clock ticks of UTC, taking an additional two offset calculations to settle within two clock ticks of UTC.

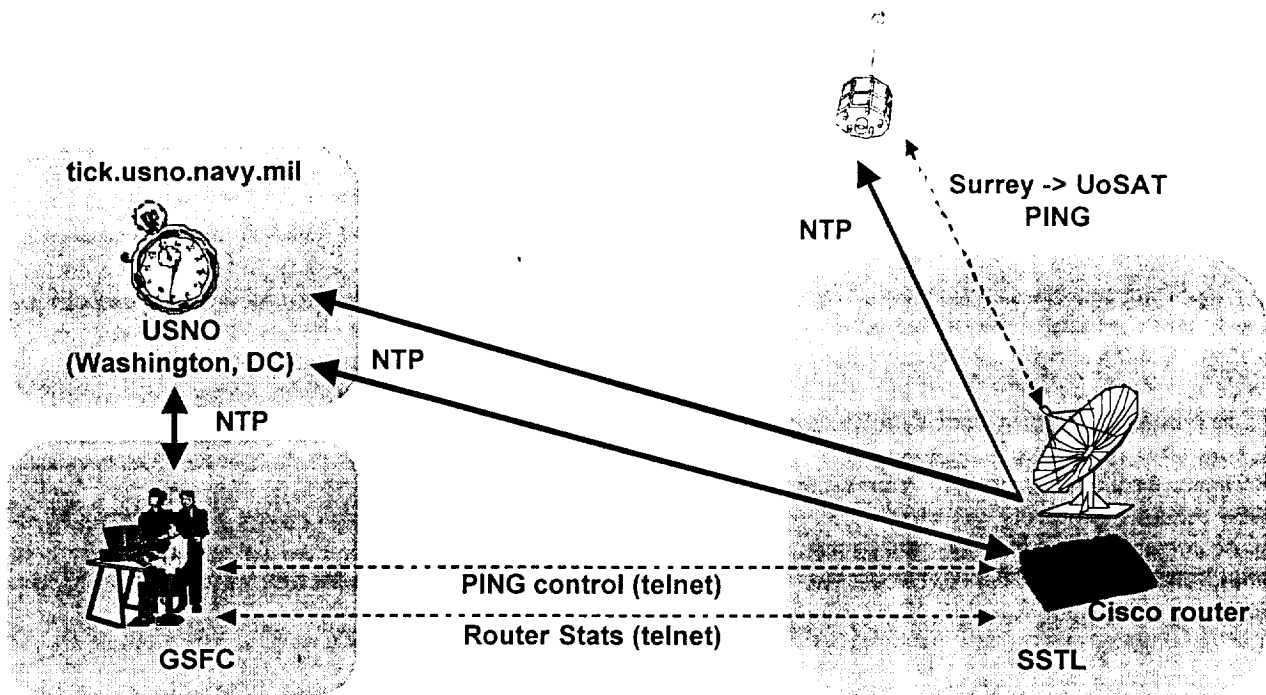


Figure 5 - NTP test configuration

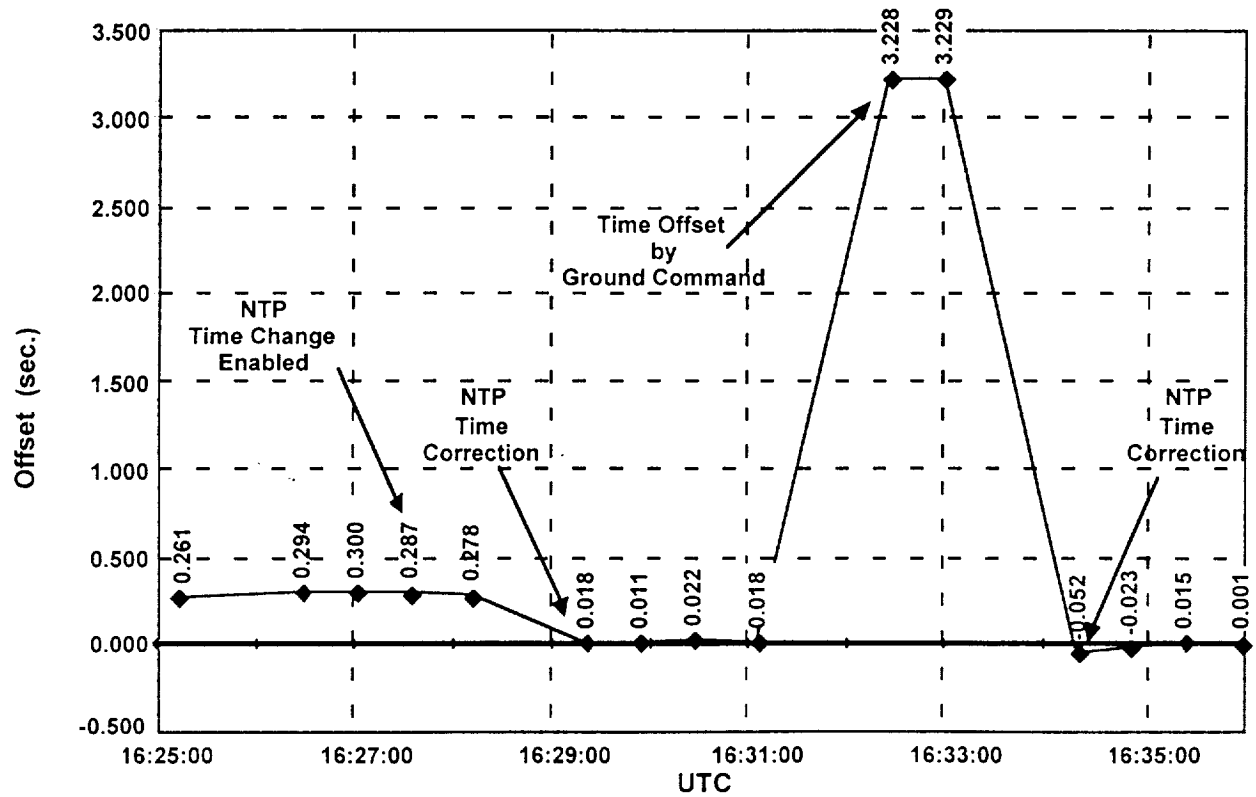


Figure 6 - NTP test result

Future Work

Testing of the File Transfer Protocol (FTP operating over the Transmission Control Protocol (TCP)) was initiated after the NTP tests and is currently in progress. Files were successfully transferred and work is currently underway to adjust some of the standard TCP tuning parameters to optimize performance. The results are nearing the theoretical maximum throughput, and will be the subject of a future paper.

The activities so far have been done in fairly simple and controlled configurations. More work is needed to investigate additional protocols required to properly deploy Internet protocols in worldwide, operational space communication networks. Implementing HDLC framing and IP packets on spacecraft and installing routers at ground stations are not major problems. The main issues are to deal with network security and the highly mobile aspects of spacecraft.

The OMNI project is in the process of expanding its test environment to include multiple spacecraft simulators and ground nodes for testing mobile IP and mobile routing protocols. These investigations plan to use the Linux and VxWorks operating systems on the spacecraft simulators and Cisco IOS 12.1 or newer software on the ground routers.

Security solutions based on Internet security protocols (IPsec) and virtual private networks (VPNs) will be configured and tested along with the mobile IP environment.

The mobile IP and security work will focus on issues for deploying IP in operational space communication networks. Additional work is also planned to identify spacecraft control and data delivery applications to use over a space IP network. One of the main application areas to be investigated will be reliable file transfer in space environments. This will focus on file transfer applications that operate over UDP and that can then operate in communication environments with extremely long round-trip times and link bandwidth asymmetry.

The GSFC science user community is organizing a workshop to discuss Internet protocols in space, to be held at GSFC in November 2000. This will be an opportunity for anyone interested in space Internet concepts to discuss issues and propose solutions. For current information, see <http://siv.gsfc.nasa.gov/>.

Current information on test results and future activities will be posted on the OMNI project web site at <http://ipinspace.gsfc.nasa.gov/>.

Conclusions

The current activities have demonstrated that standard Internet protocols will function in a space environment and are useful and effective in typical spacecraft operations.

The last 18 months of tests and demonstrations have shown that HDLC framing and IP packets provide a very simple and flexible communication mechanism for space communication. HDLC framing is well supported in a wide range of COTS products and has been used on spacecraft for over 10 years. Using the Internet Protocol as a network layer allowed easy integration and testing of our end-to-end scenarios. Also, both HDLC and IP required no modifications to operate in intermittent space link conditions.

HDLC framing provides a minimal byte overhead along with a link level error check. The variable length of HDLC framing also results in very simple data packing and unpacking since one IP packet normally ends up in one HDLC frame. A large UDP packet can be sent, causing IP fragmentation, but this is under the application programmer's control and can be completely avoided if desired. The biggest benefit of using HDLC is that it is supported on virtually any communication hardware that has serial interfaces.

Using the IETF multiprotocol encapsulation over frame relay has proven to be very robust and supported on every piece of communication equipment we have worked with. We have mixed equipment from different vendors on serial links, and there have been no compatibility problems. Frame relay equipment can also be used to provide basic forwarding of frames without any IP processing involved. This provides additional flexibility in deploying communication systems.

While many of the Internet protocols (i.e. TCP, FTP, NTP) work in full-duplex communication scenarios, we have also successfully used others (i.e. UDP) in either receive-only or transmit only scenarios. During the tests described in this paper, a one-way UDP based telemetry stream was used for diagnostics and statistics data. These one-way data transmission modes must be supported in order to deal with spacecraft contingencies when a full-duplex link is not available. This is just

one more case of the Internet protocols being flexible enough to support a wide range of requirements.

Finally, introducing a network protocol like IP in the communication architecture has allowed us to easily support a wide range of communication scenarios and mission scenarios. Using IP has allowed us to communicate around the world and introduce new applications very quickly and easily. Most of the traditional interface control documents (ICDs) are eliminated since the Internet standards are already well specified, highly interoperable, and widely available.

Acknowledgments

The research described in this paper was carried out by personnel from Computer Sciences Corporation working for NASA's Goddard Space Flight Center under contract GS-35F-4381G S-36130-G, with additional efforts and support contributed by individuals from various GSFC organizations. The work was funded by NASA's Space Operations Management Office (SOMO) Communication Technology Project headed by Tom Costello. The authors would like to thank Dave Israel and GSFC code 450 for their pioneering IP work on the SPTR project, Chris Jackson of Surrey Space Technologies Ltd. and Harold Price of VyTek Wireless for their support on UoSAT-12, and Cisco Systems for the loan of a Cisco 1601 router for use in the SSTL ground station.

References

1. "Internet Protocol, DARPA Internet Program Protocol Specification", Internet Engineering Task Force RFC-791, September 1981
2. "IP Mobility Support", Internet Engineering Task Force RFC-2002, October 1996
3. "Cellular IP - A New Approach to Internet Host Mobility," ACM Computer Communication Review, January 1999
4. "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet Engineering Task Force, Internet-Draft Mannet-DSR-03 , 22 October 1999
5. "Real-Time Information Technology Challenges for NASA's Earth Science Enterprise", G. Prescott S. Smith K. Moe, The 20th IEEE Real-Time Systems Symposium, Dec. 1-3, 1999 Phoenix, Arizona, USA
6. "File Transfer Protocol", Internet Engineering Task Force RFC-959, October 1985
7. "Security Architecture for the Internet Protocol", Internet Engineering Task Force RFC-2401, November 1998
8. "Virtual Private Networks Identifier", Internet Engineering Task Force RFC-2685, September 1999
9. "Network Time Protocol (Version 3) Specification, Implementation and Analysis", Internet Engineering Task Force RFC-1305, March 1992
10. "Simple Mail Transfer Protocol", Internet Engineering Task Force RFC-821, August 1982
11. "High-level Data Link Control (HDLC) - Frame Structure", ISO-3309
12. "NFS: Network File System Protocol Specification", Internet Engineering Task Force RFC-1094, March 1989
13. "Internet Access to Space," Proceedings of the 14th Annual/USU Conference on Small Satellites, August 2000.

