

Link and Routing Issues for Internet Protocols in Space¹

Keith Hogie, Ed Criscuolo, Ron Parise
Computer Sciences Corp
7700 Hubble Dr.
Lanham-Seabrook MD 20706
(301) 794-2999 Keith.Hogie@gsfc.nasa.gov

Abstract—The first step in using Internet Protocols in space is to establish the basic Internet datagram delivery service over satellite RF links. This paper discusses the low-level data link and data routing issues related to using Internet protocols to support spacecraft communications. It covers issues related to layer 1 (physical), layer 2 (data link), and layer 3 (network). It does NOT cover layer 4 (transport) and above.

At the physical layer, the paper presents various applications of forward-error-correction (FEC) coding techniques, such as convolutional coding and Reed-Solomon. It describes approaches for using these techniques in ways that are independent of the protocols used at the data link layer and above.

At the data link layer, common, commercially available framing schemes are discussed along with how they can be easily deployed. A rationale is provided for the selection of HDLC/frame relay framing along with IETF multi-protocol encapsulation.

At the network layer, the Internet Protocol end-to-end addressability and routing is discussed in the context of space-based applications. Standard solutions for dealing with the intermittent and mobile links of satellites are also discussed. These include a discussion of Mobile IP and mobile routing protocols.

Finally, deployment of these protocols in both spacecraft and ground systems are discussed. Details of current implementations by the Operating Missions as Nodes on the Internet (OMNI) project at NASA/GSFC using operational space and ground systems such as UoSAT-12 and TDRSS are also provided.

TABLE OF CONTENTS

1. INTRODUCTION
2. OVERVIEW OF LOWER LAYER PROTOCOLS IN SPACE
3. PROPOSED ARCHITECTURE
4. PHYSICAL LAYER ISSUES
5. LINK LAYER ISSUES
6. NETWORK LAYER ISSUES
7. GROUND-BASED DEMONSTRATIONS
8. SPACE BASED DEMONSTRATION
9. CONCLUSIONS
10. ACKNOWLEDGEMENTS

1. INTRODUCTION

This paper discusses issues related to the use of standard Internet protocols and link layers for satellite communication. It focuses on the lower layer protocols that deliver data over the space link and how those protocols can operate with the Internet Protocol (IP)[1] to provide a

universal, end-to-end data communication architecture for space. This low-level datagram delivery approach using off-the-shelf, low-cost, commodity-level standards will become increasingly significant in the years to come. Both Earth and space science missions plan to fly more and more sensors and have them interact to form a "SensorWeb"[2]. The present labor-intensive, mission-specific techniques for processing and routing data do not scale well and will 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.

2. OVERVIEW OF LOWER LAYER PROTOCOLS IN SPACE

The goal of the Operating Missions as Nodes on the Internet (OMNI) project at NASA's Goddard Space Flight Center (GSFC) 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, command/control, and data processing 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 Protocol. IP is the technology that drives the public Internet and therefore draws billions of dollars annually in research and development funds. Most private networks, including NASA's operational communication network, 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.

Thirty years ago, spacecraft communication was a very special business with organizations such as NASA breaking new ground and designing new protocols for the "special" space environment. However, communication technology has undergone huge changes over the last thirty years. Communication applications such as cell phones, Internet telephone calls anywhere in the world, and worldwide network access from a handheld computer, that once would have seemed impossible, are now in common use. Spacecraft environments still pose numerous challenges but most of these have direct analogs and solutions in the ground-based mobile IP and wireless networking industries, such as:

¹ U.S. Government work not protected by U.S. copyright

- intermittent communication links
- highly asymmetric or unidirectional communication links
- bit error rates higher than most hardwired links
- multiple mobile nodes forming a dynamic network topology
- maintaining a single address for a spacecraft as it uses different ground stations

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 IP (MIP) [3] and mobile routing. They are also driving the development of new protocols such as Cellular IP[4], Dynamic Source Routing (DSR) [5], and other ad-hoc-networking protocols.

This paper will examine selected lower layer 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 were used to test some of these concepts with an orbiting spacecraft.

3. 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 commercial-off-the-shelf (COTS) hardware and protocols while avoiding creating any new "space-specific" solutions. A high-level view of this architecture appears in figure 1.

The major change from today's satellite communication systems is to change the format of the data on the space-to-ground link. Changing the spacecraft data format to match formats used by standard Internet devices allows the extensive use of COTS networking devices in ground stations. This greatly reduces equipment cost and allows much easier upgrades to accommodate future mission needs.

The ground station may also be either a passthrough site or a store-and-forward site. The passthrough operation is implemented using standard Internet routing. Store-and-forward operations can be built using standard computers and distributed computing technology.

The figure shows two different types of spacecraft. One spacecraft only has a single IP address and is very similar to current spacecraft. Instruments still interface with the command and data handling (C&DH) system and the change is in the data format from the C&DH to the radio frequency (RF) system. This is similar to a mobile laptop using standard Mobile IP solutions. The other type of spacecraft shown is one with an onboard local area network (LAN) and multiple IP addresses. The onboard LAN requires Mobile routing protocols and an onboard router to hide the mobility issues from each node on the LAN.

Of course, as spacecraft become more easily accessible network nodes, communication security becomes more of an issue. But as the Internet becomes more critical to everyday business operations, extensive security solutions such as Virtual Private Networking[6] (VPN) are being developed which can also be applied to satellite missions.

The key to this whole architecture is that it is built upon the protocol layering concepts of the ISO OSI network reference model.

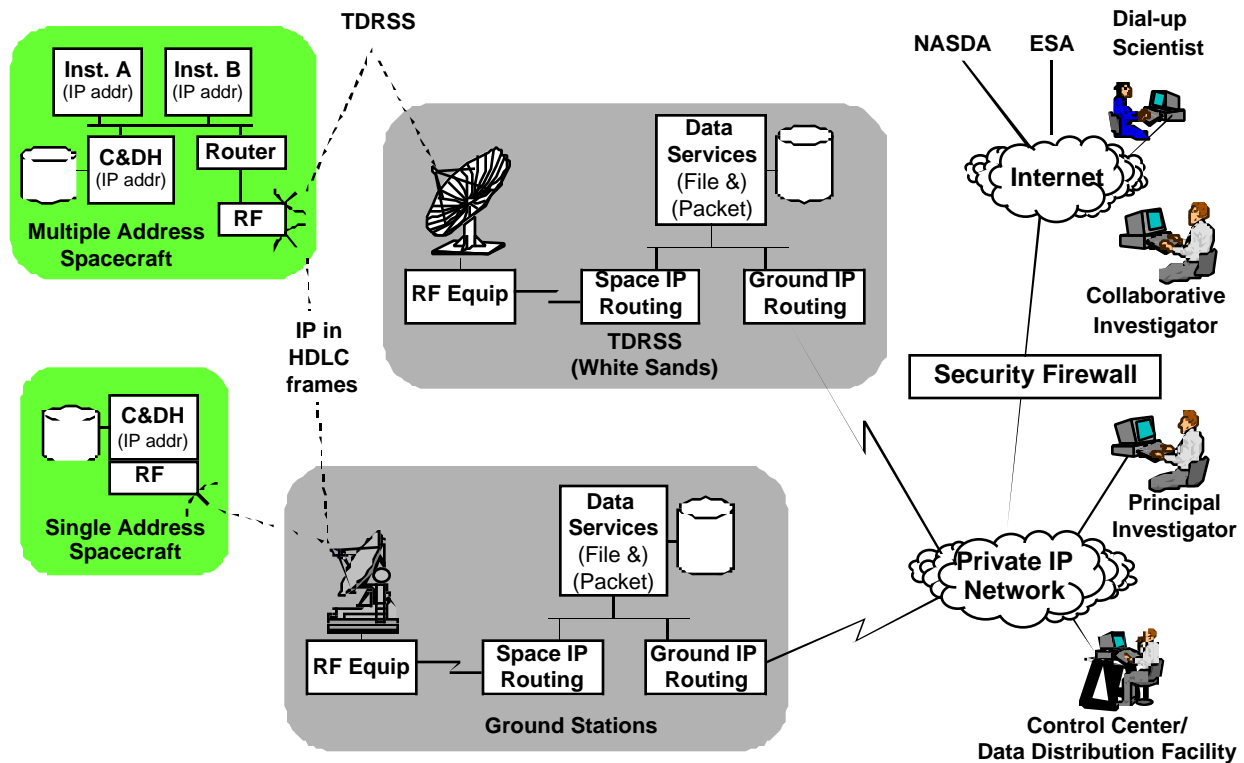


Figure 1 - End-to-End Space Internet

The ISO OSI model defines a protocol stack with seven layers for any network. Those seven layers and their associated areas of standardization are:

1. physical - bit stream details, signal voltage, cable specifications, modulation techniques
2. data link - supports transfer of data across the physical link in frames
3. network - hides data transmission details from upper layers, end-to-end addressing
4. transport - data stream multiplexing, reliable and unreliable transparent data transfer between end points
5. session - provides control structure to establish, manage and terminate connections
6. presentation - performs data services (ASCII-EBCDIC, encryption, security)
7. application - provides services to users of the OSI environment (file transfer, remote login, network management)

Each layer has defined interfaces with upper and lower layers and performs a single function in the overall communication process. However, today's Internet is best represented with a 5-layer model as shown in figure 2. This is similar to the OSI model in the bottom 4 layers with the top three OSI layers (application, presentation, and session) merged into the application layer.

A key component of the diagram is the single network layer protocol (IP) that provides a common denominator that ties together all the other protocols. It provides global addressing and datagram delivery while also isolating the upper and lower layers. This global addressability has allowed the Internet to grow to hundreds of millions of users while still supporting data delivery between all users. The layer isolation has been critical in allowing evolution in both the lower and upper layers with minimal change and reconfiguration of existing users.

This paper is focused on the lower layer protocol issues related to extending IP to the spacecraft environment. Some of the primary protocols identified for space related use are shown in the bottom three layers of figure 2. However, getting IP to the spacecraft can not be seen as the ultimate goal. We must look beyond just moving IP packets around on a RF link and continuing to operate spacecraft mission systems in the same old manner. Upper layer protocol issues are documented in another OMNI paper titled "Transport Protocols and Applications for Internet use in Space"[7]. The true power of IP to the spacecraft lies in its global addressing and datagram delivery and the use of higher level protocols to change the way that the ground and the spacecraft interact. This allows the use of off-the-shelf solutions from other disciplines, such as system and network management.

Figure 3 indicates a more specific selection of protocols in a spacecraft to end user protocol stack diagram. The key points of this diagram are that the upper layer network services, layer 3 and above, operate on an end-to-end basis

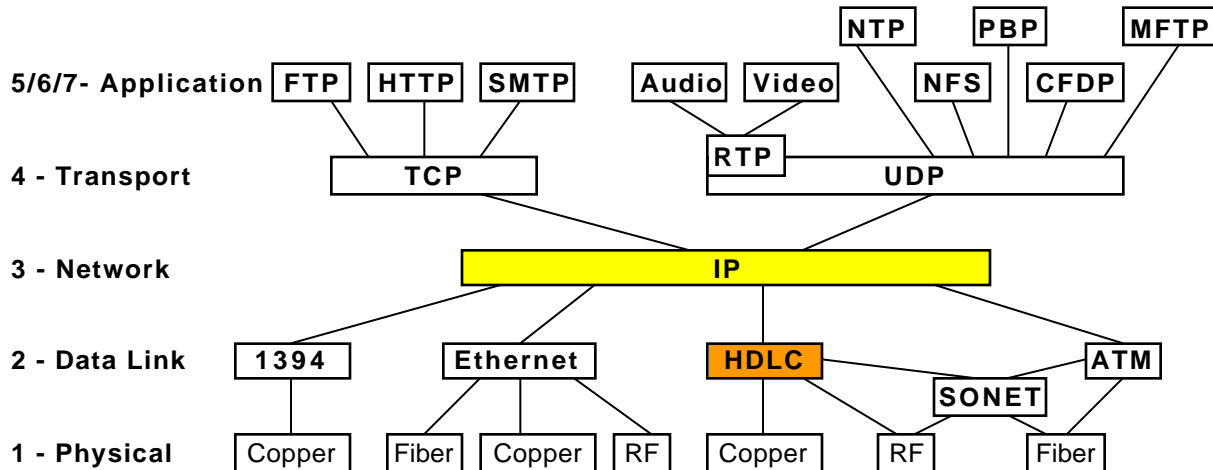


Figure 2 - Internet Layers and Protocols

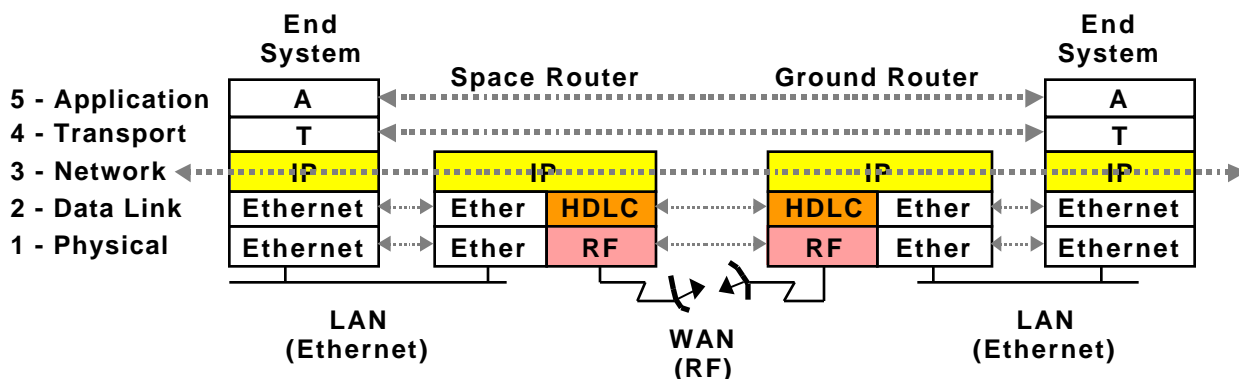


Figure 3 - Proposed space-to-ground protocol stacks

and are independent of the various physical media and data link protocols at layers 1 and 2. The end systems don't even know what sort of other data links, in this case a single RF link, are in the path between the end users. Network systems, such as routers or bridges, between the end systems operate in only the lower 2 or 3 layers of the protocol stack and are completely unaware of what upper layer protocols are passing over them. This concept of upper layer independence from the lower layers has been exploited extensively in the Internet. This layered and modular approach is what allows the Internet, and potentially space networks, to make changes and upgrades in one area without any impact on other areas. In the Internet the lower layer communication links are constantly being upgraded to higher rates and different link protocols but no changes are required by the hundreds of millions of Internet users.

In current space-based communications, the data link and physical layers are represented by the RF link from the ground station to the spacecraft and by framing mechanisms such as time division multiplex (TDM) major/minor frames and Consultative Committee for Space Data Systems (CCSDS) transfer frames. Current missions don't implement a standard network layer to provide network-wide addressability. Once data leaves the space-to-ground link other mechanisms such as dedicated circuits or address translation gateways are used to direct the data to its next destination. The application layer is implemented with concepts such as CCSDS packets. However, the intervening layers change with the medium that is being used for transferring the data. For the spacelink itself, the transport and network layers are non-existent and the session layer is defined by acquisition of signal (AOS) and loss of signal (LOS) and commands sent by the end-user to start and stop transmissions. In the ground link, the network and transport layers are satisfied with Nascom and, in many cases, IP protocols.

In contrast, a terrestrial network has a much more orderly progression of layers. The Internet is a strong example of how the layers actually interact. In the transfer of data across the network, the application, presentation, and session layers are defined by protocols such as network file system (NFS)[8], telnet[9], and file transfer protocol (FTP)[10]. The data formats are preserved throughout the end-to-end transfer of data. The transport layer is either the transmission control protocol (TCP) [11] or the user datagram protocol (UDP) [12] and the network layer is IP. Again, these formats are preserved throughout the network. At the data link and physical layers, the frame headers change as the physical media changes. As data moves from the local area network to a wider area network, the lower levels of the network model change, but the upper models are preserved. This is not the case in the current space mission networks where the change in the network protocol begins at the transport layer.

Extending the ground network to the spacecraft requires some very simple concepts:

- the spacecraft is either a computer or a network of computers with some very specialized peripherals (kind of like a lab computer with measuring devices hanging off of it).
- the ground station antenna (and associated gear) is a RF transmitter/receiver and data router for layers 1, 2 & 3 in the same way that a bridge or router is a media

converter between an Ethernet LAN and a serial interface for a wide-area network (WAN).

Figure 3 indicates a potential stack diagram for an end-to-end communication system using standard network technologies for all ground communication and adding in the necessary modification to fit the space link into the overall networking model. The main purpose for a diagram of this type is to identify the data formats and protocols used on each link and to verify that identical physical and data link protocols are used on the ends of each link. The dotted line at layer 3 indicates an end-to-end network protocol such as IP which provides end-to-end addressing and hides the details of the lower layers from the upper layers. It should be noted that the only space specific part of this diagram is the RF link between the spacecraft and ground. All other parts use standard Internet technology.

Onboard Spacecraft IP LAN

One of the key features of this architecture is the incorporation of an IP stack in the onboard processor. It may also include the use of peer-to-peer IP networking via an onboard LAN. The use of IP provides end-to-end network addressing between any combination of onboard systems with each other, multiple ground sites, and potentially other spacecraft. An onboard LAN supports distributed processing and "smart" instruments, while IP in an onboard processor supports legacy processor-controlled "dumb" instruments. There is an IP address associated with the processor, each "smart" instrument, and the router, and they are directly reachable from any node on the network. The router takes care of delivering packets to the appropriate LAN 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 CANbus, Ethernet, IEEE-1355, and IEEE-1394 (Firewire).

Spacecraft Router Function

A router is a network device that has two or more network interfaces and forwards or "routes" IP packets among its interfaces based on their network destination address. At its basic functional level, the router performs simple conversion from one link-level interface to another. For example, the spacecraft router in figure 3 could be converting high-level data link control (HDLC) [13] framing on a serial link into Ethernet framing on a LAN. In small, simple spacecraft, there may be only a small number of "dumb" processor-controlled instruments. In this case, 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 and spacecraft redesign while still retaining the benefits of end-to-end IP networking.

4. PHYSICAL LAYER ISSUES

Before a spacecraft can transfer any data to the ground or another spacecraft, a communication link must first be established. At the lowest level this consists of activities like tuning transmitters and receivers and pointing antennas. This all assumes that technology for using radio frequencies or optical techniques have been implemented in the space and ground systems that can deliver the necessary bits across the required distance. Future missions are considering scenarios that require gigabit data rates from orbits beyond the moon and at planetary distances.

Developing the basic transmission technologies and implementing them within the power, size, and weight restrictions of spacecraft continues to be a major challenge. This paper does not address the lowest level RF modulation and transmission mechanisms but focuses on techniques to be used once bits have been delivered across the link. A major theme of this paper is that work on developing new transmission technology can and should be done independent of the protocols used over the space communication link. This is part of the layering concept where the interface between the physical bit delivery layer and the data link framing is at the bit level. This section describes some of the techniques used at the physical layer and they include some sort of framing. However, the purpose of that framing is only to operate on a bitstream and improve link quality. There is no addressing or protocol information of any sort at this layer.

Bit delivery

The basic function of the physical layer is to provide a mechanism to deliver bits across a point-to-point link or between two nodes on a multi-node local area network. Sending bits over a link requires the use of some type of modulation or coding technique to place bits on the physical media and to extract them on the other end. A simple physical modulation technique is to represent a 0 with a low voltage and a 1 with a high voltage. If the link is a relatively noise-free, a direct connection with a pair of lines with data on one line and a clock signal on the other can be used. The data is recovered by simply sampling the data line on each clock cycle. This type of signaling is used in common serial line protocols such as RS-449/422 and V.35.

However, if the link only has a single line the clock and data must be combined on the link in a form that can be recovered on the other end of the link. This is normally used on media such as Ethernet, optical, and RF links. Bit recovery normally consists of detecting transitions on the line and synchronizing a phase-locked loop to recover a clock signal and then extracting bits from the received signal. But, this can lead to data recovery problems if there is a long string of zeros or ones because the phase-locked loop can drift and random bits may be added or deleted.

Modulation and Coding

There are many bit level modulation and coding techniques available that can provide more reliable data recovery over these serial links with an embedded clock signal. The exact techniques vary widely depending on the physical transmission media being used. Some of the most common media are copper wires, fiber optic cable, and radio frequency (RF) wireless transmissions. Some of the commonly used physical modulation and coding schemes are:

- Manchester coding for 10 Mbps Ethernet
- 4B/5B for 100 Mbps Ethernet and Fiber Data Distributed Interface (FDDI)
- 8B/10B for Gigabit Ethernet and Synchronous Optical Network (SONET)
- Biphase shift keying (BPSK) and quadrature phase-shift keying (QPSK) for RF systems.

The details of these physical modulation techniques are not covered in this paper. However, one point to note is that the same modulation technique must be used on both ends

of a physical communication link. This can be seen in protocol stack diagrams by noting that the bottom layer protocol must always match between any two devices on the same link. The main issue is that the modulation and coding technique used is independent of the upper layer framing. This allows the use of any coding technique, including those optimized for space use, with standard data link layer framing and IP protocols.

Forward Error Correction Coding

Another approach to dealing with potential erroneous bit recovery on these links is to include additional bits that the receiver can use to detect and correct damaged bits. This type of coding is referred to a forward-error-correction (FEC) since the error correction information is passed forward with the data. Various FEC coding schemes have been devised over the years. Some of the most common FEC techniques are convolutional coding and Reed-Solomon (R-S) coding.

The major difference between these two coding techniques is that convolutional coding operates on a serial bitstream with no specific byte boundaries while Reed-Solomon coding operates on fixed size blocks of data. A convolutional encoder accepts individual bits, adds additional coding bits based on a predictable algorithm, and passes out the encoded bitstream. A convolutional decoder reverses this process by identifying the original pattern, removing the additional bits, and passing out the original bitstream. The additional bits provide sufficient information so that some errors can be detected and corrected by the decoder.

Reed-Solomon coding does not insert bits into the middle of the data but appends check symbols to a whole block of data. These symbols can later be used to detect and correct errors that may have been introduced in the data. Since RS coding operates on a block of data the receiver must locate the RS synchronization pattern at the beginning of the code block. The CCSDS Reed-Solomon coding specification[14] uses a 4-byte synchronization pattern (0x1acff1d) to delimit the code blocks and a (223,32) coding scheme. Using a 4-byte pattern and fixed length blocks provides a robust sync detection in more severe bit error environments. The long sync pattern is less likely to spuriously occur due to bit errors and the fixed length blocks allow the receiver to "flywheel" or assume where a sync pattern should be and continue processing data without dropping lock.

The Intelsat Technical Note TN309.5 specifies a Reed-Solomon code for carriers to use and it has a 4-byte sync pattern (0x5a0fbe66) and Reed-Solomon code parameters of (219,201,9). It also specifies an interleaving scheme to distribute burst errors over wider areas of data and increase the probability of error correction. A common use of these Intelsat communication links is to provide WAN connectivity between switches and routers transmitting HDLC frames. Another commercial application of Reed-Solomon coding is in Digital Video Broadcasting (DVB) which uses yet another Reed-Solomon coding algorithm. The main point is that many communication applications use forward error correction techniques today but it is used to simply provide better link quality and is independent of any data link framing implemented by higher level users.

This is different from many current spacecraft systems where the RS framing is also used as the data link framing. However, this then forces each data link frame to be fixed length to match the RS code block length. The main problem with this is that science and engineering data packets are normally not the same size as the RS frame.

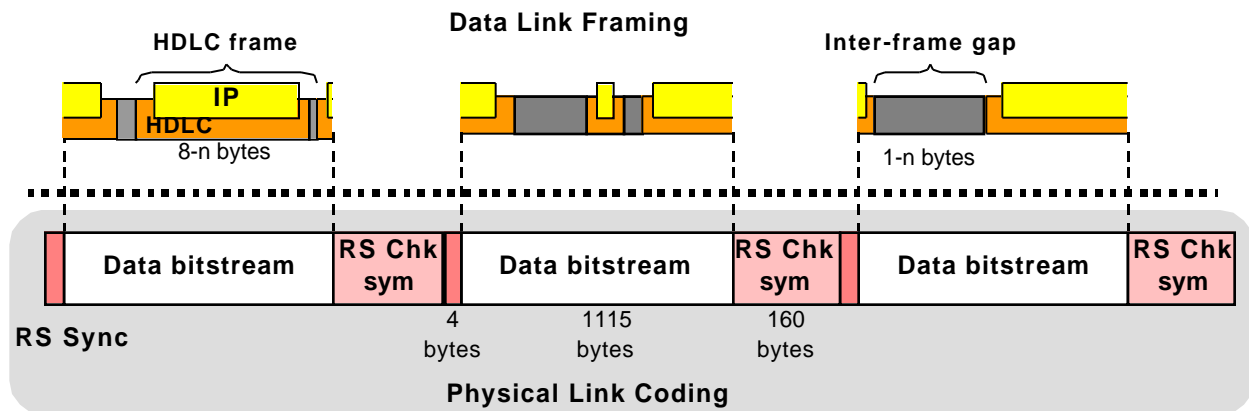


Figure 2 - Separation of HDLC Framing and RS Coding

Fitting various length packets into fixed length RS frames means that additional information must be included along with the packets. This information indicates where the first packet starts in a frame and how long each packet is. Since the various packet sizes do not fit evenly into RS frames, packets are also split between frames.

If there are too many bit errors in a frame the Reed/Solomon coding will not be able to correct the bits. In this case the frame is discarded along with the part of the packet from the previous and following frames.

Separation of Framing and Coding

One of the most important issues in this paper is to note that unlike current space communication systems, commercial network products perform forward error correction (FEC) coding, such as Reed-Solomon or convolutional, *independently* from the data link 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 data link frames as a bit-stream to be protected. This is a key difference between the current data formats used in many space missions and the OMNI architecture.

This separation, as illustrated in figure 4, is the standard way Internet connectivity is deployed across commercial satellite links. Commercially available satellite modems support many modulation and coding techniques to improve the bit error rate (BER) of bits passed through communication satellites. However, the inputs and outputs of these modems are simply a clock and data bitstream. This allows users to connect whatever network equipment they want and use any framing protocol desired. There is no relationship between the users data link framing and any framing that might be used over the RF link. This approach allows future spacecraft to use new and better coding schemes by only changing the FEC processor in their transmitters/receivers without any changes in the rest of the installed equipment onboard or in ground systems.

Reed-Solomon coding is also commonly used as a bit level FEC mechanism for many other applications such as cable modems, ADSL, cell phones, direct-broadcast TV, and CD-ROMs. These applications do not use the RS code block for data link framing but simply to provide better data quality to the bitstream being delivered.

Finally, separating the Reed/Solomon code block framing from the data link framing eliminates the current need for fill frames and fill packets. Since the space link uses synchronous clocking, conditions occur where there is no upper layer data to be sent but frames must still be output. Current protocols implement fill packets to be used to fill out frames to meet frame output timing requirements. This added complexity goes away when RS coding is separated from data link framing.

The Reed-Solomon coding simply operates on a bit level and is constantly accepting bits without any relationship to whether the upper layers are sending frames or not. This is the way Reed/Solomon coding is used in all other commercial applications. This is also the way that Reed/Solomon coding has been used on the WIND and POLAR spacecraft for the last 5 years.

5. LINK LAYER ISSUES

The link layer builds on the bit delivery capabilities of the physical layer and provides a mechanism for delimiting a group of bits into an identifiable frame of data. The link layer also adds addressing information, possibly control information, and some type of frame level error detection mechanism, normally a cyclic redundancy check (CRC).

A space mission communication system consists of many different data links to carry data from the science instrument on the spacecraft, down to the ground, and eventually to the scientist. The OMNI project sees great potential for implementing "faster, better, cheaper" satellite communication systems using the link layers that have been very successfully used to build the Internet. Based on the most common COTS technology, this would consist of Ethernet framing on LANs onboard the spacecraft and either HDLC or packet over SONET framing on the space-to-ground link. On the ground end, COTS network equipment is widely available to support data rates up to 45 Mbps using HDLC framing over clock and data serial interfaces, and rates of 622 Mbps and beyond using SONET.

Some other onboard LAN technologies that are currently being worked on are the IEEE-1355 (SpaceWire) and IEEE-1394 (FireWire). The mapping of the Internet Protocol into these media is not as well defined as Ethernet but work is underway in the IETF to define IP over IEEE-1394.

Onboard Ethernet Framing

Recent developments in industrial automation have shown great potential for using standard Ethernet technology for data communication in real-time environments like spacecraft. Major efforts are underway to use Ethernet in industrial environments that have always had requirements for real-time, deterministic, reliable, and secure operations. Many companies have come together to form the following groups:

- Industrial Ethernet Association - <http://www.industrialethernet.com/>
- Industrial Automation Open Networking Alliance - <http://www.iaona.com/>
- GE Cisco Industrial Networks - <http://www.gecisco.com>

The OMNI project sees great potential for building on the current industrial Ethernet work to develop a cost effective Ethernet solution for use on spacecraft.

This is based on the fact that Ethernet has taken over the majority of the data communication world with a huge number of Ethernet interfaces being deployed. Ethernet also supports a wide range of data rates from 10 to 100 to 1,000 Mbps with a 10,000 Mbps version currently under definition. There is also a tremendous amount of research and development going into standard upper layer protocols for use over Ethernet, ruggedized connectors, and new strategies for using Ethernet in process control environments.

Studies in the late 1980s showed that Ethernet response times can consistently be maintained under 2 milliseconds for a lightly loaded Ethernet network and under 30 milliseconds for a heavily loaded network. The key to successfully using Ethernet is the proper design of the network topology and traffic patterns and using devices such as switches to separate traffic and reduce collisions.

Work is also underway in the industrial Ethernet community to define upper level programming languages and an application programming interface (API) to standardize the software used on real-time Ethernet LANs. The API will address issues related to timing, low-level device control, and real-time response. This work should also be useful for future spacecraft designers.

RF Link HDLC Framing

Based on its near-universal use on the terrestrial Internet, the OMNI project chose HDLC framing for the link-level protocol on space-to-ground links. This allows simple, straightforward interfacing with existing commercial routers in the ground station. HDLC has been used in

communication equipment for over 30 years and provides basic framing for many serial line protocols such as IBM's synchronous data link control (SDLC), Frame Relay, X.25, and ADCCP.

As indicated in figure 5, at the physical link 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 (01111110). 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. On receipt, these extra zero bits are automatically removed from the data by the hardware.

Using data link framing that relies on only a single byte flag pattern to delimit frames is a concern for noisy environments like space. However, using HDLC on top of forward error correcting codes addresses that problem. Before the HDLC sync is even an issue the lower layer coding must be successfully processed. Convolutional and Reed-Solomon coding use much stronger synchronization mechanisms and once they have been processed, the HDLC will be more reliable. This is especially true when using Reed-Solomon coding since the result of the R-S processing is normally a perfect bit stream. In present space data processing systems, if the R-S block is so badly damaged that the coding cannot correct it, the block is discarded. Using HDLC on top of Reed-Solomon actually allows the damaged bits to be passed on to the HDLC frame processing to see if it can locate some good frames within the damaged R-S block. HDLC can use its 16-bit CRC to determine if it has extracted a good frame from the R-S code block. This could potentially result in successfully extracting more data from uncorrectable Reed-Solomon blocks than is possible today.

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 non-return-to-zero (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

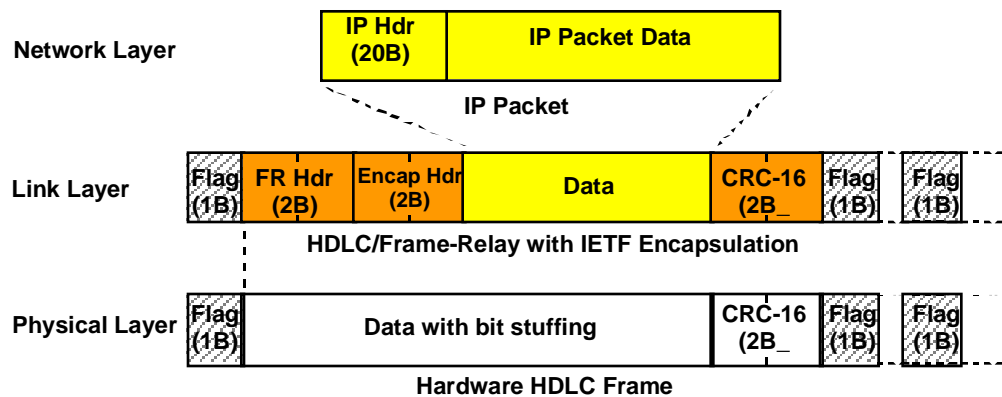


Figure 3 - HDLC/Frame Relay/IP formats

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. None of the protocols layers need to worry about generating “fill packets” or “fill frames”.

The OMNI project considered various commercially available encapsulation mechanisms for use over HDLC. There were two major criteria for selecting the encapsulation method to use:

- the encapsulation could not require full-duplex links since full-duplex links might not be available during a spacecraft emergency
- the encapsulation must be interoperable between many vendors routers since no group can ensure that all routers at all ground stations will come from the same vendor

The first criteria ruled out protocols like Serial Line IP (SLIP) and Point-to-Point Protocol (PPP)[15] because they need full-duplex links for parameter negotiation at startup. The second criteria ruled out protocols such as Cisco’s default HDLC encapsulation which uses a Cisco specific HDLC header.

This led to the choice of the IETF encapsulation for multi-protocol over frame-relay/HDLC specified in RFC 2427[16]. In the OMNI tests with UoSAT-12 the actual header format consisted of simply inserting 4 bytes of fixed information at the start of each HDLC frame. The first 2 bytes are a standard Frame Relay header with a few status bits and a virtual channel number or Data Link Connection Identifier (DLCI). Also, since this is a standard Frame-Relay header, a spacecraft could actually use the DLCI to provide additional channelization and routing in addition to the IP capabilities. This could be used along with standard Frame-Relay equipment at the ground station. The next 2 bytes in the header simply indicate that the contents of this frame are an IP packet. There are also standard IETF definitions that allow the transport of other protocols in the data area of the frame.

This data link framing provides capabilities identical to those used by current spacecraft. An application level science or telemetry packet inside of a User Datagram Protocol (UDP) packet with an IP header and HDLC is delivered through space exactly like current data. The main difference is that by using IP and HDLC headers the data leaving the spacecraft is in a format that can be directly ingested by COTS Internet equipment on the ground.

Packet Over SONET

Supporting data rates over 45 Mbps using commercial routers requires using a framing technique other than just HDLC. Commercial routers have interfaces that support data rates up to 45 Mbps using HDLC framing over High-Speed Serial Interfaces (HSSI) but shift to Synchronous Optical Network (SONET) interfaces for data rates of 155 Mbps, 622 Mbps and 2.4 Gbps. These interfaces have traditionally used Asynchronous Transfer Mode (ATM) cells to frame IP packets over SONET.

One objection to using ATM for science satellite communication is the 10% overhead imposed by the ATM cell format. ATM cells contain 48 bytes of data with an additional 5 bytes of cell header. IP packets must be broken into 48 byte pieces with some additional information added

to help the receiver reassemble the packet. This process of splitting the IP packet adds complexity and results in additional error cases where the loss of a single ATM cell results in the loss of the entire IP packet. In an environment like ground fiber links with large amounts of bandwidth these issues have traditionally been accepted. However, as the Internet grows and users want more and more bandwidth, alternatives to ATM cells have arisen.

One of the more popular alternatives to ATM cells for high-speed IP support is to bypass the overhead of ATM and put IP packets into SONET. This format is called Packet over SONET (POS) [17]. There is still some framing needed but the framing has gone back to the traditional mode of using HDLC framing to put one IP packet in one HDLC frame and carry that over SONET. A PPP header is also added and the end result is very similar to the multi-protocol over Frame Relay format described above.

One concern the authors have with this format is that PPP requires a full-duplex link so it can negotiate some parameters. This presents a problem for spacecraft use because there must be a way to send blind commands to a spacecraft without any two-way communication. This is necessary for spacecraft emergency situations when normal two-way communication with the spacecraft is not available.

However, spacecraft with this type of high-rate downlink normally have multiple transmitters operating at both low and high rates. They also would not normally be attempting any high-rate downlink if the spacecraft was in trouble. A choice of link protocols for data rates above 45 Mbps needs further work to determine the applicability of Packet over SONET for spacecraft.

Framing Overhead

A major concern for satellite system engineers is both the processing overhead and byte overhead associated with protocols. This is not a major issue for onboard LAN protocols where bandwidth is not as severely limited. Overhead is an issue on the space-to-ground link where bandwidth is often limited due to standard RF link budgets affected by power, error rate, signal quality, and distance.

The overhead of HDLC is very minimal with only the following fields

- 1-byte flag or sync byte
- 4-byte Frame Relay and IP encapsulation header
- 2-byte CRC for error detection

This framing overhead is as small as other space framing formats used today.

Another aspect of the HDLC framing is its bit-stuffing. This ensures that on the transmission media there are never more the 5 one bits in a row. Breaking up strings of ones is necessary to avoid patterns that would look like a flag byte and signal the beginning or end of a frame. Inserting ones into the data stream results in added bit overhead for HDLC. The extreme case would be an overhead of 20%, which would result from a frame containing all one bits and a zero bit would be inserted after every fifth bit. However this scenario is very unlikely since sending frames of all ones would be a waste of bandwidth anyway. That sort of data can be easily compressed with major reductions in data volume. Another option is to apply data randomization before passing the data down to the HDLC layer. This further reduces the likelihood of long strings of ones. Some examination of data files from the WIND, POLAR, and

SOHO spacecraft indicate a realistic HDLC bit-stuffing overhead is in the 1-3% range.

6. NETWORK LAYER ISSUES

The network layer is the key to the global connectivity provided by any network and especially the Internet. Frames at the link layer normally contain source and destination addresses but those addresses are only valid at the link layer. Those addresses are only used to deliver frames to the proper device on a single physical link. Once a device receives a frame, the data link headers are discarded and only the network layer information remains. The network addresses are globally unique and remain with the upper layer data to provide the information needed for a network of routing devices to forward data to its final destination.

Internet Protocol

Figure 6 shows the basic format of the IP header. The primary fields to note are the 32-bit source and destination addresses. These are used to deliver the datagram to its destination. The source address also acts like the return address on a letter and tells the recipient where this data came from and how to communicate back to the source. This feature of source and destination addresses on each datagram is critical to supporting future cooperative science and constellation space missions. Supporting communication among large numbers of spacecraft requires either a central communication hub that knows the addressing details of each spacecraft or an automated mechanism like Internet addressing and routing.

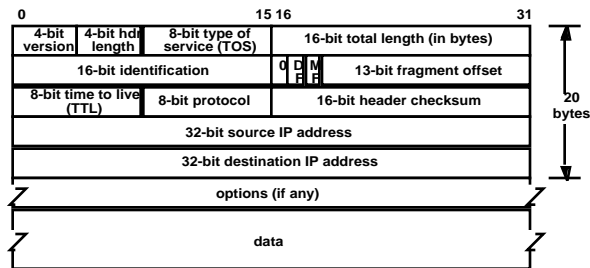


Figure 4 - Internet Protocol Header Format

Many of the other fields (e.g. total length, identification, don't fragment, more fragments, and fragment offset) are involved in fragmenting and reassembling large IP datagrams for transport over data links that cannot carry the whole datagram in a single frame. The "time to live" (TTL) field sounds like a field that might be affected by long delay space links like L1/L2 or Mars and beyond. However, this field is not really a time but a count that decrements each time the datagram passes through a router. Its function is to protect against misconfigured routers with a routing loop where a packet could be passed around the network forever. The TTL causes the packet to age and be discarded after a short time. The "type of service" field provides a mechanism for prioritizing datagrams if needed. Finally, the "protocol" field indicates the next level protocol (e.g. UDP, TCP) contained in the data portion of this datagram.

The IP header on each Internet datagram consists of these 20 bytes in a fixed format. There are also standard options that can be added to carry additional control or status information. Many of the options are used to either specify

a particular path the datagram should take from router to router, or to record the route the datagram actually took.

Another option is the timestamp option which requests each node that handles the datagram to add their local timestamp to the header. This option was used in tests with the UoSAT-12 spacecraft as a simple way to read the spacecraft clock. Internet Control Message Protocol (ICMP) or PING packets were sent from a ground router with the timestamp option set. The ground router entered its local time, the spacecraft entered its time when it handled the PING packet, and then the ground router added a final timestamp when it received the PING response.

Datagram routing

The basic construct the Internet is built on is a capability for network devices to simply forward datagrams toward a destination address. There is no guaranteed delivery in the network layer. Its basic function is to provide addressing and associated datagram delivery. However, datagram delivery becomes a complex problem as a network grows to the size of the Internet. When a network is small, the lookup tables that tell each router how to forward datagrams toward their destination are small, simple and relatively static. The maintenance of these routing tables can be performed in a manual or simple automation fashion. This is similar to the manual management of current spacecraft data deliver systems.

When a network reaches the size and continual growth of the Internet, the updating of routing tables must be implemented automatically between routers. There has been a tremendous amount of work done by groups such as the Internet Engineering Task Force (IETF) to develop and deploy robust and automated routing technologies. Many routing protocols such as Routing Information Protocol (RIP) [18], Open Shortest Path First (OSPF) [19], and Border Gateway Protocol (BGP) [20] have been developed to support the Internet. These protocols use the relatively stationary topology of the Internet to determine paths to subnets and inform each router where to forward packets to reach each subnet. Routers also have default routes that are used as a path of last resort when there is no better route information available.

Routing tables are reasonably static but change whenever a link fails and the routers adjust to define new paths for datagram forwarding. The Internet addressing scheme also assumes that a device with an IP address remains attached to its subnet. However, when we start using IP addresses on spacecraft or other mobile devices some new routing issues arise.

Mobile IP

In today's spacecraft communication, control centers normally send commands to the ground station the spacecraft is passing over and the command is uplinked to the spacecraft. The major issue is that the control center must know where to send the commands and address them accordingly. However, as large constellations of spacecraft are deployed, advance planning and scheduling of contacts becomes more complex and expensive and an automated solution for delivering commands to spacecraft is desirable.

As indicated in figure 7, when a spacecraft has an Internet address (e.g. 150.15.15.18), that address will be part of a ground based subnet (e.g. 150.15.15.x). Any IP datagrams addressed to the spacecraft address from anywhere on the Internet will be routed using standard Internet routing and will be delivered to the associated ground subnet. A router on that subnet will recognize that the datagram destination address resides in space and forward the datagram up the RF link. This assumes that the spacecraft is passing over the ground station with the spacecraft's home subnet.

One other option is to simply give the spacecraft an IP address in the subnet of each of its ground stations. However, this requires both the spacecraft and ground controllers to keep track of which station is currently in use and the corresponding addresses. This solution also gets more complex as more ground stations are added and does not support the use of additional ground stations without spacecraft reconfiguration.

Another option is to use the Dynamic Host Configuration Protocol (DHCP) to let the spacecraft learn an IP address for

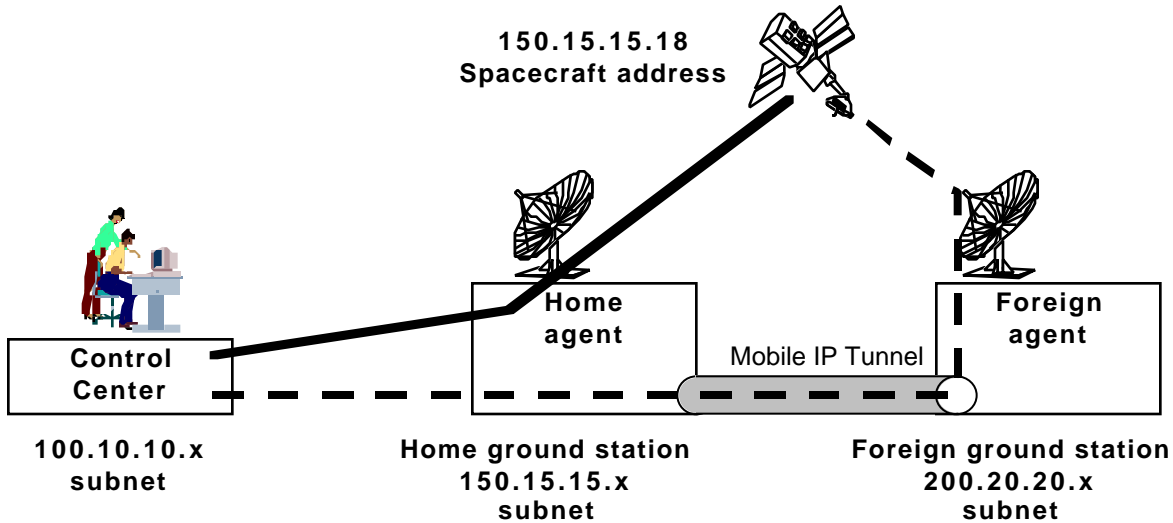


Figure 5 - Mobile IP for Spacecraft

However, if the spacecraft is over a foreign ground station (e.g. 200.20.20.x), the commands from the control center addressed to the spacecraft (e.g. 150.15.15.18) will end up at the home subnet (e.g. 150.15.15.x). This will not get the commands to the spacecraft. However, this is exactly the same problem encountered by many wireless devices such as mobile laptop computers. The IETF has developed standards called Mobile IP (RFC 2002) to deal with this problem. These protocols use an initial protocol exchange to allow the mobile device or spacecraft to determine if it is in contact with its home subnet and associated home agent software or a foreign subnet and its foreign agent software. If the mobile device is in contact with a foreign subnet, the foreign agent establishes a tunnel back to the home agent at the home subnet. Then when the control center sends a datagram to the spacecraft address, the packet goes to the home router where the home agent notices that there is a tunnel to the spacecraft via a foreign router. The packet is then sent through the tunnel to the foreign agent which passes it out its serial interface and up to the spacecraft.

This sort of Mobile IP scenario is mainly only an issue for sending data to the spacecraft. When any packets are sent from the spacecraft to any ground station, the ground station simply uses the destination address to forward the packets using standard Internet routing rules. One possible exception is if the foreign ground station has additional routing rules, for security reasons, which prevent it from forwarding packets whose source address is not within the foreign subnet. Then the tunneling features of Mobile IP would be needed to encapsulate the spacecraft packets for delivery to their home subnet.

There are other possible options for solving these mobility issues but Mobile IP currently provides the best solution for automation and scalability.

use during that contact. This allows the spacecraft to downlink data but still has problems with getting commands to the spacecraft. It requires some mechanism for telling the control center the current IP address of the spacecraft.

These cases have only addressed a spacecraft or mobile host with a single IP address. If the mobile device or spacecraft has a LAN with multiple IP addresses then the problem gets more complex. One solution is for each node on the spacecraft with an IP address to perform Mobile IP registration and set up tunnels for each. However, this does not scale well and causes additional traffic for all of the registrations and additional software for each node. The solution currently being worked on in the IETF is called Mobile Routing. It involves a router that performs all of the Mobile IP operations and none of the nodes on the LAN even realize they are mobile. They simply operate just like they do on a fixed LAN. The research and development in this area is being driven by concepts in which all future automobiles will have onboard LANs with Internet addresses and full mobile Internet connectivity. The large size of the automobile market, a potential market for mobile routers, is huge and the commercial research and development investments are substantial.

Data Prioritization

Current spacecraft protocols do not really provide any special support for indicating different priorities for types of data. The virtual channels in CCSDS protocols are sometimes used for this but they only provide a few levels of priority and their meaning is different for each mission.

There are many options available in Internet protocols for supporting a wide range of priority mechanisms. These options include the following:

- DLCIs in the HDLC/Frame Relay header that can be used similar to the CCSDS virtual channels.
- Type of Service bits in the IP header that have traditionally been ignored but are now beginning to see use to support prioritization of Internet services
- Priority queuing in routers allows sorting individual packets into multiple priority queues based on fields such as the transport protocol (e.g. UDP, TCP) and transport protocol port numbers (e.g. 1-65535).

These options can be used to prioritize urgent information such as gamma ray burst notifications at the highest priority. Other timely data such as housekeeping data could be at a lower priority telemetry and playback data at even lower priorities. There are also other protocols being developed for Internet use to deal with specific prioritization and quality of service issues. This work is being driven by applications such as voice over IP and streaming video.

Network Protocol Overhead

Today's satellite protocols focus on using minimal overhead formats for delivering data from spacecraft to the ground. The HDLC framing described earlier provides minimal overhead at the link layer. The major overhead from using Internet protocols comes in the Network and Transport layers.

The IP header consists of a fixed 20-byte header with optional field up to a maximum of 64 bytes. The majority of packets on the Internet use the base 20-byte header. Packets will also have an additional transport header of 8 bytes for UDP packets and 20 bytes for TCP packets. This results in the following overheads for some selected packet sizes shown in table 1.

Table 1 - Network/Transport Protocol Overhead

	User Data Sizes (bytes)			
	100	500	1000	1400
IP (20)	16.6%	3.8%	1.9%	1.4%
UDP/IP (28)	21.8%	5.3%	2.7%	1.9%
TCP/IP (40)	28.5%	7.4%	3.8%	2.7%

The table indicates high overhead for small packets of user data but that overhead drops to much more reasonable values for larger packets. This would seem to indicate some work needs to be done to try to reduce this overhead if these protocols are to be used for spacecraft. Once again the Internet, and its continually growing range of applications, has already encountered this problem and is working on solutions.

A major driving force for reducing the overhead of Internet protocols is the rapid growth in voice over IP deployment. Voice over IP sends lots of small, digitized voice samples over IP using the UDP transport layer. The 28 bytes of overhead represent a significant portion of the voice packet. There is significant interest in reducing this overhead to allow voice over IP to grow to support millions of users without having most of the bandwidth used for protocol header.

Some new header compression algorithms have already been developed in the IETF and are available in some vendor's products. The current standards are RFC 2507[21] and 2508[22]. They utilize the fact that most of the fields in the headers for a particular session (e.g. source/destination address and port numbers) remain the same for each packet. These algorithms send some packets with full headers and most of the packets with the UDP/IP or TCP/IP header compressed to contain only the information that has changed from the last packet. This results in these headers shrinking to 6-7 bytes. This results in Internet packets that are equal to or in some cases less than the overhead of current space protocols. There is currently an IETF Working Group called Robust Header Compression (ROHC) (<http://www.ietf.org/html.charters/rohc-charter.html>) that is working on additional compression options for even smaller headers.

7. GROUND-BASED DEMONSTRATIONS

In late 1998, the OMNI project began constructing a "proof of concept" ground-based prototype of an IP spacecraft. Initial demonstrations were performed with instruments in a van sending data back to a prototype control center at GSFC via NASA's Tracking and Data Relay Satellite System (TDRSS) links. These demonstrations used TDRSS Internet connectivity that had been installed and used to provide a communication link to scientists at the South Pole.

Many of the tests consisted of simply sending a one-way data stream in UDP packets from the van, through TDRSS to White Sands, and having them routed back to GSFC with standard Internet addressing. There was no forward link or uplink to provide two-way communication. These demonstrations were both easy to schedule, since they only required minimal TDRSS support for Multiple Access (MA) Return-only service, and they tolerated the intermittent TDRSS connectivity as the van drove around GSFC between buildings and under trees. This also demonstrated the use of Internet protocols for simple data flows very similar to those used on today's spacecraft with TDM and CCSDS data structures.

When two-way communication service was available, these tests also included other protocols such as file transfers using both FTP and NFS and audio and video streaming.

8. SPACE-BASED DEMONSTRATION

In late 1999 the OMNI project had been looking for opportunities to test these "Internet in Space" concepts on an orbiting spacecraft. 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, a spacecraft launched in May 1999 by Surrey Satellite Technology Ltd. (SSTL), an ideal test platform, as it already used HDLC framing to carry its AX.25 protocol. The AX.25 protocol and HDLC framing have been used on over 20 spacecraft over the last 10 years. Since HDLC interface 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.

Ground Station Implementation

Since the SSTL ground station already supported HDLC framing, a standard Internet router was the only addition needed. Figure 8 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 was 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.

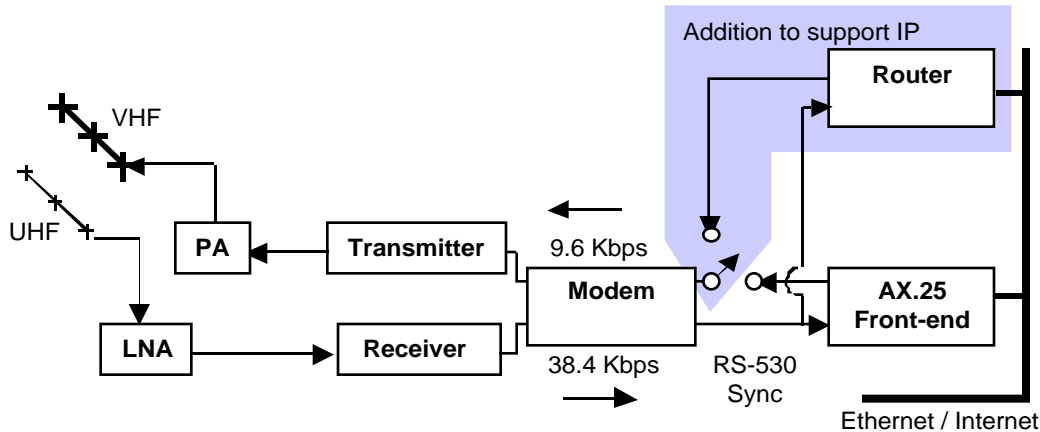


Figure 6 - SSTL Ground Station Configuration

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.

Flight Tests

In February 2000 work was initiated to port a standard IP stack to the SpaceCraft Operating System (SCOS) used on the UoSAT-12 spacecraft. In April 2000 the first basic connectivity tests using IP to a spacecraft were performed. Standard ICMP echo request (PING) packets were sent from both GSFC and the Surrey ground station to the spacecraft.

The packets passed through a standard router at the Surrey ground station and were transmitted to the UoSAT-12 spacecraft. The standard IP stack onboard UoSAT-12 returned echo response packets addressed to the separate sources. Those packets then passed through the ground station router and were delivered to their respective destinations using standard Internet routing. These tests verified proper operation of both the end-to-end IP routing and the HDLC framing on the space-to-ground link.

The results from a PING test to verify basic HDLC and IP operation are shown in figure 9. PINGs were sent to UoSAT-12 continuously from the router at SSTL while a PING was sent once every 10 seconds from NASA/GSFC. The figure shows the successful replies from the spacecraft from AOS to LOS. The variation in the propagation time to UoSAT-12 on the horizon at AOS (approximately 3,000 miles) to the highest elevation overhead (approximately 400 miles) is shown in the slight curvature in the response time

plot. The bottom line in the plot shows the antenna elevation. The curve above it show the theoretical round trip time computed based on data rates and distance. The large number of data points and the curve fit inside of them are the actual round-trip times for each PING response received. The line through them is a curve fit that corresponds very well with the theoretical line with the difference being the processing time on the spacecraft and ground equipment. The top data points are the PING responses recorded at GSFC.

Once the end-to-end connectivity was operational, additional

tests were performed to have the spacecraft automatically set its clock using the Network Time Protocol (NTP) [23] by referencing a time server (tick.usno.navy.mil) at the US Naval Observatory (USNO). This represents somewhat of a worst-case test, as the USNO is across the ocean, 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. Tests were also performed using the standard File Transfer Protocol (FTP) to retrieve files from UoSAT-12 and to send files to it.

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 4Q 2000.

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 also be configured and tested along with the mobile IP environment.

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

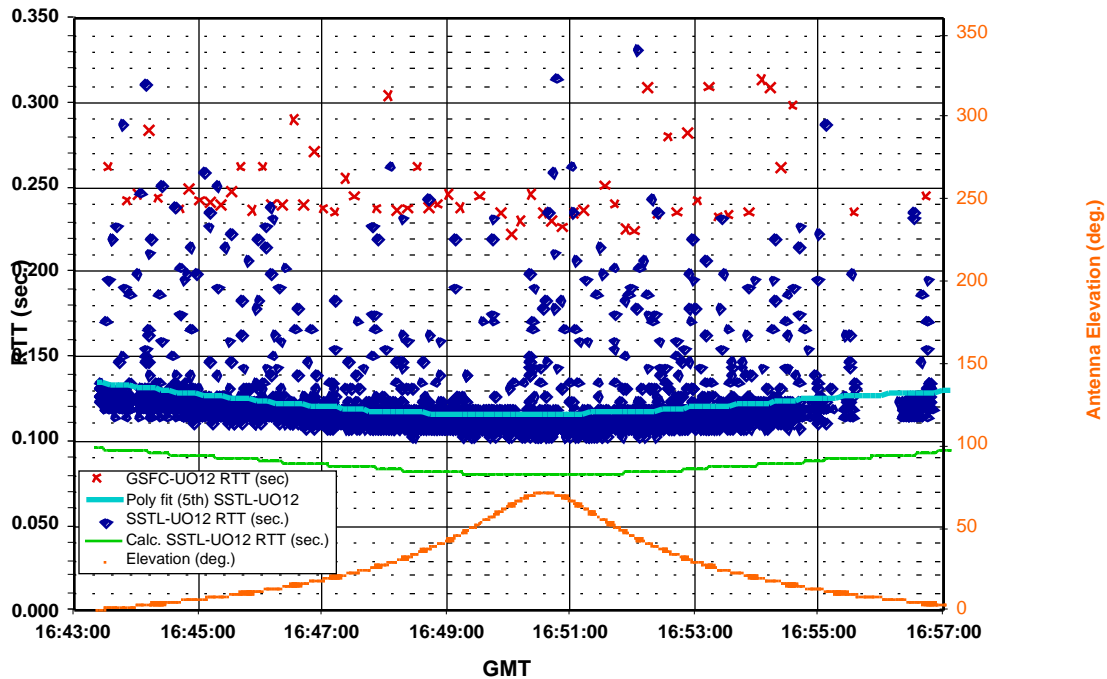


Figure 9 - UoSAT-12 PING test results

9. CONCLUSIONS

The OMNI project at NASA/GSFC has defined a flexible end-to-end spacecraft data communication architecture using Internet protocols. The last 22 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 over frame relay encapsulation 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.

Introducing a network protocol like IP in the space 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 in COTS products.

Full deployment of Internet protocols for spacecraft will require ground station upgrades and more system engineering to deploy Mobile IP and security solutions. However, these can all be addressed with commercially available products and solutions.

The major missing pieces are components for the spacecraft. Technologies like Ethernet and HDLC are currently in use on some low-earth orbit spacecraft where radiation is not a major issue. More work is needed to develop fully space-qualified components for onboard serial interfaces to the RF equipment and for onboard LANs.

10. 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 with UoSAT-12, and Cisco Systems for the loan of a Cisco 1601 router for use in the SSSL ground station.

Keith Hogie - Computer Sciences Corporation - Mr. Hogie has an extensive background in designing and building satellite data processing systems, control centers, and networks at GSFC. He has developed ground data processing systems and control centers for over 14 spacecraft over the last 25 years at NASA/GSFC, and led the development of the NASA Internetworking Laboratory Environment in 1990. He is the technical leader of the OMNI project at GSFC where he is applying his networking and satellite background to develop and demonstrate new communication technologies for future space missions.



Edward Criscuolo Jr. joined Computer Sciences Corp. in 1991 as a Senior Computer Scientist working for the Goddard Space Flight Center. In that time, he has been the task lead for a number of spacecraft ground system projects that span many aspects of Goddard space missions, including Planning & Scheduling systems, spacecraft command management, and level-0 processing of telemetry and science data. In 1999, Mr. Criscuolo joined the OMNI project as a senior project member, where his duties include systems analysis, systems engineering, top-level design, prototype development, and backup technical lead.



Dr. Ron Parise - Computer Sciences Corporation - In 1984, Dr. Parise was selected as a payload specialist astronaut and was involved in mission planning, simulator development, integration and test activities, flight procedure development, and scientific data analysis. He has logged 615 hours in space as a member of the STS-35 and STS-67 crews. In 1996 Dr. Parise assumed a communications engineering support role for Mir, International Space Station (ISS), and the X-38 project. In 1997 Dr. Parise also began working with the OMNI project as a scientific liaison and systems architect.



REFERENCES

- [1] "Internet Protocol, DARPA Internet Program Protocol Specification", Internet Engineering Task Force RFC-791, September 1981
- [2] "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

- [3] "IP Mobility Support", Internet Engineering Task Force RFC-2002, October 1996
- [4] "Cellular IP - A New Approach to Internet Host Mobility," ACM Computer Communication Review, January 1999
- [5] "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet Engineering Task Force, Internet-Draft Mannot-DSR-03 , 22 October 1999
- [6] "Security Architecture for the Internet Protocol", Internet Engineering Task Force RFC-2401, September 1998
- [7] "Transport Protocols and Applications for use in Space", E. Criscuolo, K. Hogie, R. Parise, IEEE Aerospace Conference 2001, March 2001
- [8] "NFS: Network File System Protocol Specification", Internet Engineering Task Force RFC-1094, March 1989
- [9] "Telnet Protocol Specification", Internet Engineering Task Force RFC-854, May 1983
- [10] "File Transfer Protocol", Internet Engineering Task Force RFC-959, October 1985
- [11] "Transmission Control Protocol", Internet Engineering Task Force RFC-793, September 1981
- [12] "User Datagram Protocol", Internet Engineering Task Force RFC-768, August 1980
- [13] "High-level Data Link Control (HDLC) - Frame Structure", ISO-3309
- [14] "Telemetry Channel Coding", Consultative Committee for Space Data Systems, CCSDS 101.0-B-4, May 1999
- [15] "PPP in Frame Relay", Internet Engineering Task Force RFC-1973, June 1996
- [16] "Multiprotocol Interconnect over Frame Relay", Internet Engineering Task Force RFC-2427, September 1998
- [17] "PPP over SONET/SDH", Internet Engineering Task Force RFC-2615, June 1999
- [18] "Router Information Protocol (RIP) Version 2", Internet Engineering Task Force RFC-2453, November 1998
- [19] "Open Shortest Path First (OSPF) Version 2", Internet Engineering Task Force RFC-2328, April 1998
- [20] "BGP4/IDRP for IP---OSPF Interaction", Internet Engineering Task Force RFC-1745, December 1994
- [21] "IP Header Compression", Internet Engineering Task Force RFC-2507, February 1999
- [22] "Compressing IP/UDP/RTP Headers for Low-speed Serial Links", Internet Engineering Task Force RFC-2508, February 1999
- [23] "Network Time Protocol (Version 3) Specification, Implementation and Analysis", Internet Engineering Task Force RFC-1305, March 1992

