Conceptual Design of an IP-based Satellite Bus Using Internet Technologies

Richard A. Slywczak
NASA/Glenn Research Center (GRC)
21000 Brookpark Road, MS 54-5
Cleveland, Ohio 44135
216.433.3493
Richard.A.Slywczak@nasa.gov

Abstract. The goal of this paper is to develop a generic, reconfigurable spacecraft bus architecture that implements IP-based protocols and networking hardware that is common to terrestrial networks.

First, a description of the communications architecture for an operational Earth Science mission is presented. The Tropical Rainfall Measuring Mission (TRMM) was selected as an example that shows a typical Earth science mission with a nice complement of varying data rate instruments. We will be able to show through the satellite architecture where IP-based protocols will benefit a new design.

Secondly, we develop an IP-based satellite bus design with an Ethernet backbone using standard terrestrial networking components and protocols. The design will be highly configurable to meet many different mission requirements. Adapting the design to the TRMM communications architecture will test the feasibility. We will indicate the subsystems that are part of the design and show examples of how TCP/IP will operate on board the satellite bus.

Finally, we present the type of research needed to make IP-based missions a reality. This roadmap will provide NASA the guidance to design complex architectures that will become part of their mission portfolio in the next decade.

Introduction

Ongoing NASA research is developing complex satellite missions ranging from single satellites and constellations to space networks and sensor webs. These missions will require more interoperability, autonomy, and coordination than previous missions. To meet these goals, research at NASA has concentrated on extending the TCP/IP protocol suite for space-based applications. Extending TCP/IP promises many benefits for NASA by providing seamless communications between space and ground systems. Over the past couple of years, NASA has been testing the TCP/IP protocol suite with test satellites to demonstrate web communications and FTP transfers. These tests show that satellites in LEO orbit can successfully use the TCP/IP suite of protocols to communicate effectively with the current technology[1].

Even though NASA has been working on developing and extending protocols, the agency must also concentrate on designing architectures for new types of missions. Challenges will range from loosely formed constellations that will have limited communications in space and post process all data terrestrially to tightly formed constellations where each of the satellites in the constellations must inform the others of its position and other command information.   We can extend the problem space to satellites that are not part of a constellation but might want to communicate with other satellites so that each can take measurements over the same region of the Earth. Therefore, it is obvious that satellite intercommunication is essential and the end result is that NASA must design standards based architectures that simplify these issues.

However, the first step in designing these complex missions is to start with a well-defined problem and that is the focus of this work. This paper will outline a generic IP-based, Ethernet backbone for a LEO based satellite; a mission of this type can be divided into three communication segments, as shown in Figure 1.

Terrestrial Communications – The terrestrial communications segment, represented by the bottom third of Figure 1, is the simplest of the segments. The protocols that we are interested in leveraging for space have been developed and extensively used and tested in the terrestrial environment. Even today, once the data are downlinked from satellites, we are able to transfer it to the end users using the TCP/IP protocol suite and

networking technologies, such as routers, hubs, etc. For the ground segment, the TCP/IP protocol suite is the standard and dominant protocol.

Space to Ground Communications – Space to ground communications, represented by the middle segment of

second part of the challenge is to design an architecture for the on-board network. We propose making a technological leap in that the architecture will implement standard Ethernet and IP-based protocols; legacy architectures will not be considered in this design.
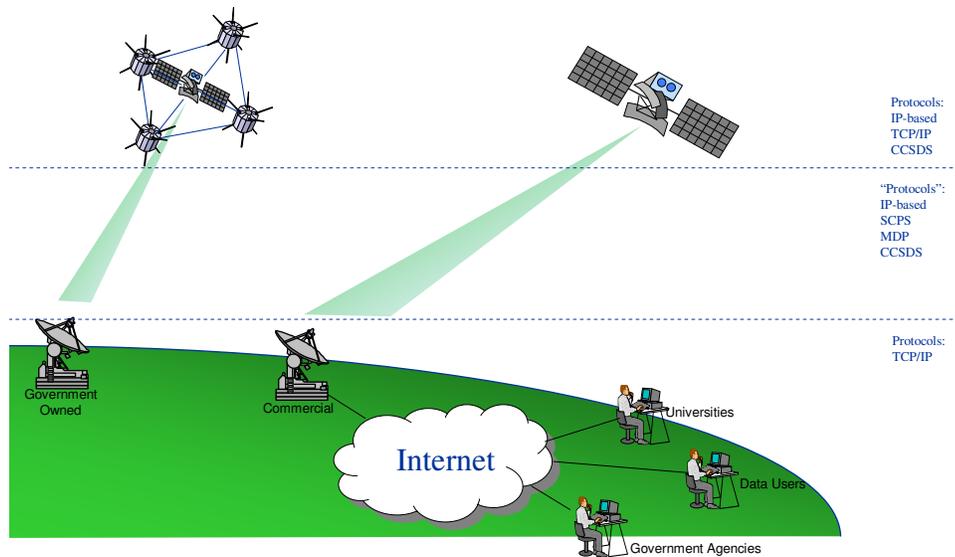
Figure 1 – Space Communications

Figure 1, has been studied significantly. There have been numerous proposals for changes to the IP-based protocol suite and development of new protocols. Some of the protocols that can be used in space to ground communications are the following: TCP/IP and UDP [2] which work well for LEO based orbits; Space Communications Protocol Standard (SCPS) [3] which are options that can be added to standard TCP/IP and UDP and is designed for Deep Space Missions; and Multicast Dissemination Protocol (MDP) [4] which adds reliability to UDP. Even the current specification of the Consultative Committee for Space Data Standards (CCSDS), which is the current standard for spacecraft communications, provides provisions for IP-based structures.

Spacecraft Segment – The spacecraft segment, represented by the top third of Figure 1, presents both protocol and architectural challenges. The on-board network has one simplifying assumption; it is a wired LAN, similar to terrestrial networks, and the TCP/IP protocol suite should provide performance similar to Earth based networks. We don't have to worry about problems like bit errors rates, latency, delays, etc. The

Goals for an IP-Based Satellite Bus

For a well defined and designed architecture, we need to develop a set of well-formed and clear objectives. These objectives will be evaluated against the new design to determine if it has significant advantages over existing architectures. This list of objectives is similar to those that we strive for in terrestrial networking and computing. Since terrestrial networking has been well tested, we will leverage as many concepts as possible for our on-board networks. The objectives for this design are listed as follows:

1. ***IP-Based*** – We would like to design an on-board network that will utilize some form of an IP-based protocol with an Ethernet backbone. One advantage is that the bus can be treated as a wired local area network and should be able to run TCP/IP with acceptable performance.
2. ***Plug-and-Play Design*** – Missions will be able to design and develop satellite busses by using components that will meet their requirements and

plug into the backbone of the spacecraft via standard connections. The components will be able to determine their configuration parameters and initialize themselves without operator intervention.

3. *Modular* – Each component will responsible for accomplishing one well defined function. Modularity must be considered for both the network design and instrument design. For instance, each science instrument will collect, maintain and transmit the data independently. If required, the instruments will be able to connect to a centralized recorder which will maintain the data until it is transmitted to a ground station.

4. ***Reconfigurable/Extensible*** – Since the requirements of each mission are different, we need an on-board networking architecture that has a high degree of reconfigurability. The generic architecture must provide the ability to not only expand but also remove elements that are deemed not necessary to the mission objectives.

5. ***Security*** – Security will be one of the most important issues in designing the communication infrastructure. Leveraging elements of terrestrial computing, a combination of techniques can be applied ranging from firewalls and routers to IPSec [5] and VPNs [6]. Using these technologies will help to protect the satellite from unauthorized users.

6. ***Data Integrity*** – The design must ensure that data can be collected and transferred with confidence that the data has not been corrupted. The protocol must be able to detect and rectify any corruption in the data and have it retransmitted from the instrument if corruption is detected.

7. ***Distributed Architecture*** – To have a truly distributed architecture, each component must be able to allocate resources, collect data and maintain the data until it can be written to mass storage or down linked to the ground. This system requires the creation of "smart" components as each will contain a processing device, short term memory, storage, and network capabilities.

8. ***Networked Environment*** – The goal is to leverage the knowledge of terrestrial networking and apply this to the on-board network environment. We have the option of incorporating both hardware and software elements (e.g., routers, hubs, firewalls, etc.) into our on-board design.

After showing the IP-based design, we will re-evaluate each of these considerations and show how the new design incorporates these elements.

Generic IP-Based Architecture

Based on the design objectives, the following will present a possible design for an IP-compatible communication infrastructure. The on-board architecture includes command and control, housekeeping, and science instruments to record measurements and data recorders to store the data until download; the bus also provides a standard interface to connect each of these components.

The redesigned network diagram of the bus is shown in Figure 2. The network is discussed in two parts: first the networks with their associated subnets and, secondly, the router/firewall combination along with connecting to the satellite via a secure mechanism.

On-Board Networks

The network on the satellite will be divided into at least three individual networks (or subnets), which are the Satellite Status & Maintenance Subnet, the Instrument Subnet, and the Recorder Subnet. These are defined as follows:

- *Satellite Status & Maintenance Subnet* will transport data from monitoring the health of the satellite, and, in addition, satellite commands from the ground. For example, Figure 2 shows the following main modules connected to this subnet: the ACS (Attitude Control System) Module and the HK (Housekeeping) Module and one or more additional subsystems. The additional subsystem(s) shows that other modules can be easily connected to this network, if required by the mission.

- *Instrument Subnet* is where the science instruments are located. The number of instruments that can be connected to the bus is restricted by either requirements of the mission or physical limitation of the bus. In addition, multiple instrument subnets are possible to accommodate high data rate instruments.

- *Recorder Subnet* contains the data recorder that will be responsible for data storage and management of the command and science instruments. The data recorder is placed on its own subnet to simplify the connections, since each of the other subnets will need to send data to it.

- *Additional Subnet* represents one or more additional subnet(s) that are needed based on specific mission requirements. Examples of additional subnets could be the logical division of instruments (command and/or science),

security purposes, or to keep the data rates on the subnet at an acceptable level. The number and purpose of the additional subnets will be based on the mission requirements. The limitations on the number of subnets should be based on either the number of networks that can be supported by the router or the power requirements of the satellite.

All subnets have access to a data recorder that will store data from the science and control modules. The data recorder is a passive device that is connected to the router via another subnet (i.e., the recorder subnet). Once the instruments collect data, it will be sent to the data recorder via the router. The data recorder is an IP addressable device that will store the data until it is requested for transmission to a ground terminal.
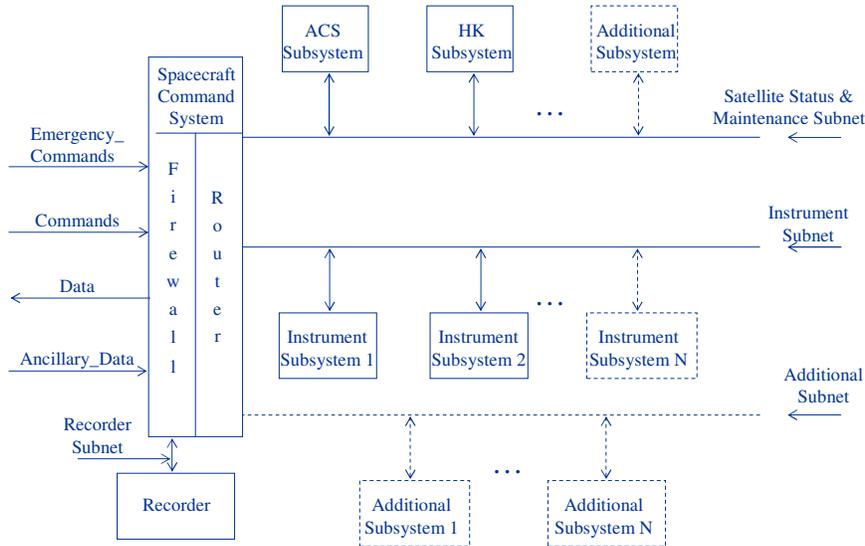

Figure 2: Generic IP-Based Satellite Bus Architecture

As with terrestrial networking, the satellite bus could have been designed in a number of different ways, but the rationale for using multiple subnets are as follows:

- *Reduction of Data Traffic.* The subnets will separate the traffic for the command and control functions and the instrument collection duties of the satellite. This limits collisions between the data, since they are on different physical networks.
- *Promote Security.* Using multiple subnets on the spacecraft will help to promote security by keeping the command/control and the instrument traffic on separate networks. For example, if an instrument scientist uploads commands to the spacecraft, the commands will not traverse the Satellite Status & Maintenance subnet and, therefore, will not be able to send damaging commands to the satellite or accidentally Distributed Denial of Service (DDOS) of the command and control bus.

In this design all instruments or modules are IP compatible and will be able to directly plug into the bus using space-qualified connectors to an Ethernet interface. The protocol for the satellite bus will be TCP/IP. The reasons for using TCP/IP on the satellite bus are as follows:

- *The Bus is a Wired LAN.* Once we get data to the spacecraft bus, the on-board architecture is a wired network similar to terrestrial networks. This environment eliminates traditional space communications problems, such as latency, bit errors, etc. Therefore, TCP/IP becomes an acceptable protocol for the on-board networking.
- *Reliable Data Communications Is Required.* The data is collected by the instruments and sent to the data storage device. The modules have no or limited methods of recreating or caching the data. Therefore, there has to be a reliable way of getting the data from the instruments to the recorder; TCP/IP will

provide the reliable data transfer with both data reliability and congestion schemes.

Communication with On-board Networking

The second part of the bus is the firewall and router, which serves two distinct functions. First, the firewall provides a security mechanism by scrutinizing packets based on rules implemented by the missions (e.g., IP addresses or port numbers). The second component, the router, is responsible for routing data packets to their destination subnet. Together, these components function as the interface between the local on-board network and the ground. All communications with the satellite will be required to pass through this interface before reaching any module on the satellite.

The firewall will provide a layer of security by filtering the traffic being sent to the satellite. It will validate the network packets based on a series of conditions or rules and can either accept a packet for processing or deny and drop the packet. If the packet passes the firewall rules, it will be sent to the router. Essentially, the router will provide the same basic functions as a terrestrial router. The main function will be to place the received packet onto the correct subnet so that the appropriate module can process it. A router will only be needed when there are multiple subnets. If the mission decides on one network (i.e., a single network containing both the science and command and control instruments), then the packet would be validated by the firewall and placed on the network.

Connecting to the Satellite

While the firewall will provide a mechanism for data security, it does not provide a complete solution, since data packets can be spoofed with a valid IP address but contain damaging data. Therefore, another security concept, a Virtual Private Network (VPN), should be utilized for connecting to the satellite to provide an additional measure of security. A VPN should be implemented to transmit critical or sensitive data safely from a source (e.g., Instrument Scientist, Mission Operations Center) to the satellite. It provides safe, secure, and private networking built on top of publicly accessible networks (e.g., the Internet). Using a VPN will permit commands and data to be securely sent to the satellite. VPNs provide the following characteristics:

- *Authentication:* It ensures that the data originated at a known valid source.
- *Access Control:* The VPN will prevent unauthorized users from accessing the network.

- *Confidentiality:* The VPN will prevent unauthorized users from reading data as it passes across the network.
- *Data Integrity:* The VPN will prevent anyone from tampering with the data as it is transmitted across the network.

With this design, missions are able to use either a VPN or firewall, or both. They can use the mechanism that satisfies their needs based on whether they are running over the open or closed Internet, level of complexity they would like to implement or the level of perceived threat to their mission.

Emergency Commanding

During the mission, the satellite can become unstable for a number of reasons, such as:

- The satellite could start tumbling in its orbit making the main antenna ineffective for receiving commands from the ground.
- The Network Interface Card (NIC) could become inoperable and not able to process the network packets.

During these instances, the ground will not be able to communicate with the satellite and it can become uncommandable. Without some type of backup, the entire mission could be in jeopardy of being lost. As a solution to this problem, the design (see Figure 2) allows for emergency commanding by connecting a low rate modem and omni-directional antenna to the bus. This low bit-rate connection that would be used for simple commands to query and/or stabilize the satellite. Once the satellite is stabilized, then commanding can resume using the normal interface.

The commands will still pass through the firewall so that they can be validated before being processed by the commanding system. With emergency commanding, only the vital components can be manipulated, such as the command and control systems, housekeeping systems, etc. The rest of the components of the satellite, such as the science instruments, will likely be in standby mode through anomaly correction schemes.

Validating the Architecture

The design, presented in Figure 2, will be validated by taking an existing science mission and determining if the existing architecture can be converted to the IP-based architecture. This test will help us determine whether the new architecture has the flexibility to meet the objectives set forth at the beginning of this paper. Eventually, we need to determine whether the new

architecture will be able to produce acceptable performance characteristics through emulation. The chosen satellite mission will have the following characteristics:

- LEO-based spacecraft – NASA has and will continue to launch a number of satellites to study our home planet. These missions are considered Low Earth Orbit (LEO) mission and are advantageous for this example, since we know that IP-based protocols will work well at LEO altitudes.
- Multi-Instruments – To meet specific mission requirements, satellite busses will host one or more science instruments. For this example, we would like to have a number of instruments with varying data rates which will increase the complexity of the design.
- Single Spacecraft – Finally, we will look at missions that contain a single spacecraft which will eliminate constellations missions.

Based on these criteria, we choose the Tropical Rainfall Measuring Mission (TRMM) which is a joint US and Japanese mission that was launched in 1997. The TRMM orbit is at 350 km with 35° inclination. The satellite orbits between $\pm 35^{\circ}$ latitude/longitude. TRMM will also fit nicely in our above objectives with the following characteristics:

- With an altitude of 350 km, it fits nicely into the LEO based category. This eliminates any of the problems associated with deep space missions, such as high latency, high bit error rates, etc.

The controller is the interface between the transponder, which is responsible for communication with the ground, and the data busses, which contain the satellite instruments. The controller is divided into an "A" and "B" side to provide redundancy; in case of a catastrophic problem, the "A" side can be deactivated and the "B" side can be activated to keep the mission going. In addition, a number of the important instruments are "cross strapped" to provide access, if only that instrument fails. The controller hosts the following instruments:
- **ACS Processor:** The bus controller for the attitude and control bus. It will determine which instrument has control of the data bus at any one time and is capable of transmitting data.
- **Clock:** Provides a centralized clock for the satellite so that each data sample can acquire a timestamp.

- TRMM has a nice complement of five (5) instruments with varying data rates. The instruments are the Precipitation Radar (PR), TRMM Microwave Imager (TMI), Visible and Infrared Scanner (VIRS), Lighting Imaging Sensor (LIS) and Clouds and Earth's Radiant Energy System (CERES).
- TRMM is a single spacecraft that will transmit data to the ground terminal via the Tracking and Data Relay Satellite System (TDRSS). In addition, the current bus architecture incorporates many of the standard design elements, such MIL-STD 1773 [7] busses, communication using CCSDS and centralized architectures.

TRMM Architecture

Next, we discuss the features of the current TRMM on-board communication architecture. This architecture will highlight a number of features that are common among the Earth Science spacecraft to demonstrate the similarities between the satellites; the concepts of a centralized clock, "A" and "B" sides, instruments, etc. can be found on many satellites. When we convert to an IP-based architecture, a number of these concepts will change, but will still provide the same service in a more standardized manner. The current TRMM architecture, as shown in Figure 3, will be discussed in two distinct parts: the controller and the data busses.

- **Uplink Interface:** Interface connected to the transponder that will receive data and commands from the ground.
- **S/C Processor:** The bus controller for the Spacecraft and Instrument busses. Its function is similar to the ACS Processor.
- **Memory:** A centralized memory bank for temporary storage of the data before it is written to the recorder.
- **Downlink Interface:** Interface connected to the transponder for downloading data to the ground terminal.

The second part of the architecture is the data busses which are in compliance with MIL-STD 1773. The 1773 standard is a duplication of the 1553 with the addition of fiber optic cabling as the medium for data transmission. The 1773 bus is centrally controlled by the bus controller that provides a standard interface for all equipment to connect to the bus. The system implements a command-response format. The data is

transmitted in a message which may be command, data, or status. The architecture contains the following three data busses:

- **Attitude Control 1773 Bus:** The Attitude Control Bus hosts the instruments that are

Precipitation Radar (PR), TRMM Microwave Imager (TMI), Lighting Imaging Sensor (LIS) and the Clouds and Earth's Radiant Energy System (CERES). In addition to the science instruments, the bus contains the Instrument
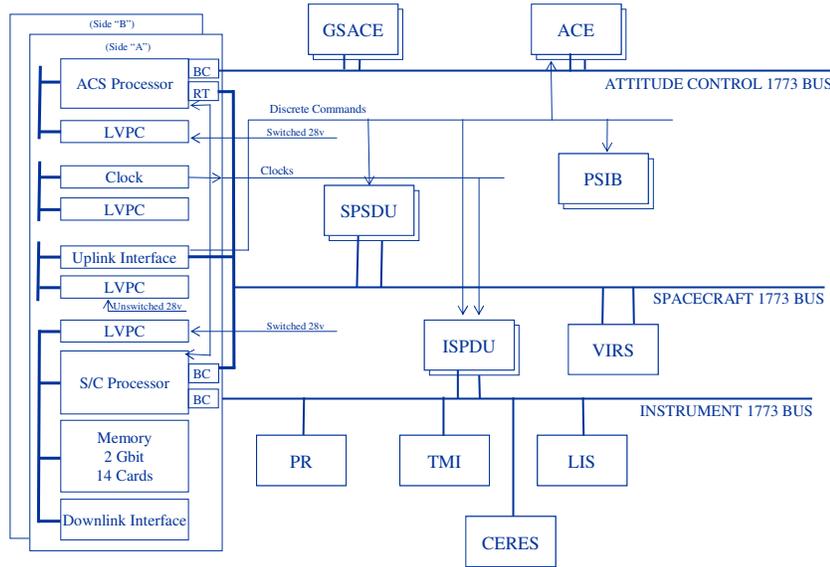

Figure 3: Current TRMM Architecture

responsible for controlling the satellite as well as recording information about its health. The instruments contained on this bus are the Attitude Control Electronics (ACE) and the Gimbal and Solar Array Control Electronics (GSACE).

- **Spacecraft 1773 Bus:** The Spacecraft Bus has a two-fold purpose. First, it regulates and controls power to the spacecraft through the Spacecraft Power Switching and Distribution Unit (SPSDU). This bus also contains the Visible and Infrared Scanner (VIRS) which is one of the main science instruments. VIRS is one of the higher data rate instruments on the satellite and warrants being separated out onto a separate bus from the other instruments.
- **Instrument 1773 Bus:** The Instrument Bus hosts a complement of four science instruments for the mission. They are the

Power Switching and Distribution Unit (IPSDU) which controls the power to the instruments.

The TRMM IP-Based Architecture

Figure 4 shows the combination of the generic IP-based architecture and the current TRMM architecture. Together, they create a possible IP-based architecture for the TRMM satellite. Our approach to creating the new architecture was a direct mapping from the current bus architecture, shown in Figure 3, into the generic architecture, shown in Figure 2. This direct mapping approach will allow us to compare and contrast the two architectures. To show the differences between the two architectures, we will revisit the objectives that were discussed in the beginning of the paper.
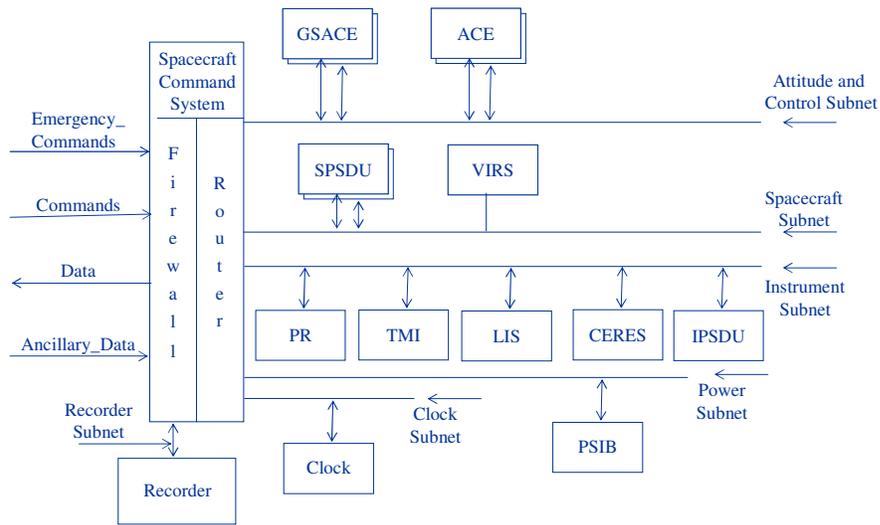
Figure 4: TRMM IP-Based Architecture

1. ***IP-Based*** – The new architecture is completely IP-based where each component has the ability to transmit IP-based packets via the Ethernet backbone through a standard connection. The underlying architecture will be based on Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [8] rather than a centralized architecture using bus controllers. The instruments will now be responsible for initiating data transfers rather than waiting on the bus controllers for permission.

2. ***Plug-and-Play Design*** – The spacecraft components will be able to dynamically configure themselves upon initialization. They can receive their configuration parameters via a central repository (e.g., DHCP) in the Spacecraft Command System. This objective is achievable since the bus architecture contains "smart" instruments which have the ability to make a connection to the central repository and process the configuration information. In addition, each component will be able to plug directly into the spacecraft bus through a standard interface.

3. ***Modular*** – There are two examples of modularity on the new spacecraft bus. First, network design is modular in the sense that each subnet is dedicated to one function. For example, the Instrument Subnet is dedicated only to the science instrument data. Secondly, the instruments implement a modular design, since each is responsible for collecting, storing and transmitting its own data.

4. ***Reconfigurable/Extensible*** – Since each project has its own requirements, reconfiguration of the new architecture is probably one of its most important aspects. In the new architecture we were able to add a new subnet to accommodate the clock which provides a source for each instrument to receive a timestamp. Also, just as adding to the architecture is important, a project also has the option of removing components. For example, if a VPN is sufficient for validating the connection to the satellite, the firewall can be removed to reduce the complexity of the satellite.

5. ***Security*** – The goal in providing security for the satellite is to leverage the lessons learned from terrestrial computer security. We have adopted both firewalls and VPNs. The firewall will inspect each packet that is transmitted to spacecraft against a set of rules that are developed by the mission. The rules might determine if a packet is from an authorized source or for an authorized destination. A firewall provides a good first line of security, but because of concerns about packet spoofing, users have the option of connecting to the spacecraft via a VPN. A VPN will be able to authenticate the user before sending packets to the spacecraft and provide the option of encryption, if desired.

6. ***Data Integrity*** – Data integrity will be provided through the TCP/IP protocol itself using the internal set of CRC and checksums. If any of these

checksums do not validate, internal TCP/IP mechanisms will be able to retransmit the packet until the validation is successful. While this is a reliable mechanism, research shows that the possibility exists for TCP/IP to validate a corrupt packet [9]. If the data is critical, then a stronger application level checksum might be advisable.

7. *Distributed Architecture* – The previous architecture was centrally controlled through the bus controller. It would determine which instrument would be able to transmit data, when and for how long. In the new scheme, the bus controllers have been eliminated, since the instruments will determine when they will transmit data to either the recorder or ground. In addition, the centralized memory bank has been distributed to each of the components so that they may store their data and instructions locally.

8. *Networked Environment* – The new design incorporates the following networking equipment: routers transmit the data between the different subnets, VPNs and firewalls add security, and TCP/IP suite provides the standard protocols for transmitting the data. In the old design, once the data was collected, it would be written off to a centralized recorder when permitted by the bus controller.

Further Research

The focus of this paper was to develop a generic IP-based satellite and show that this architecture was both flexible and feasible by taking an existing science mission and showing that we can change the communication infrastructure to one that is IP-based. While these goals were successful, a significant amount of work is still left to be accomplished. Areas of further research can be summarized as follows:

- *Detailed Design:* We have presented a high-level design for a generic IP-based satellite bus architecture. This design must be decomposed into a detailed design to look at each of the components to determine how they will fit into the structure and whether the components exist or need to be developed. In addition, this is a multi-disciplined study and we need to have this design validated by a number of engineering disciplines (e.g., spacecraft bus designers, antenna engineers, RF engineers, network engineers, security, etc.).

- *Extend to Complex Missions:* Even though we selected a simple, single, LEO based mission to serve as the sample for our design, NASA is in the process of developing more complex missions and we need to be ready for these missions having architectures developed for these missions. We need to start extending these architectures to constellations and space networks and determine optimal configurations. For example, which satellite will assume the role as the mothership in a tightly formed constellation or whether the satellites will act independently in a loosely formed constellation? How will the balance of power change if the mothership becomes damaged and who will be the successor?

- *Architecture Emulation:* The next logical step is to determine which architectures are feasible or which components will create a feasible configuration. How the instruments should be distributed on the spacecraft based on data rates and other characteristics? How the satellite will operate under normal and anomalous events? To answer these questions, we need to emulate the architectures to determine these configurations. These test scenarios must be studied from normal operations and anomalous perspectives. Using these emulations, we will be able to document the results and suggestions for missions.

- *Document the Components of the Architectures:* The final step is a two-fold approach to documenting the architectures. First, we need to document the basic architectures that are in an optimal configuration, but we must realize that we cannot develop architectures for every mission. In addition, we must document the components that can be used for satellite bus architectures along with their strengths and weaknesses. This technique will create a cookbook approach so that missions can design the infrastructure based on their unique requirements. They can determine which attributes are important and utilize those components. For example, is on-board processing more important than throughput?

Conclusions

The goal of this paper was to develop a generic IP-based architecture for a simple, single, LEO-based satellite mission. Even though the current centralized based MIL-STD 1773 architectures have performed well, there is an interest in merging the communications infrastructure of the satellite with the ground communications. This is achievable by leveraging lessons learned from terrestrial networking and applying those to space communications and determine the strengths and weaknesses. We have accomplished this in the ground networks and have performed a significant amount of research in the satellite to ground communications. However, the on-board network architectures are still relatively undefined.

As NASA develops more complex missions, the need for the communications infrastructure will in turn become more complex. In the near future, there will be constellation missions where the satellites must send command information amongst themselves in the constellation. This problem space can be expanded to include space networks and sensor web. NASA must define the on-board architectures and determine the components that are required to develop true end-to-end IP-based missions.

Acknowledgments

References

[1] Operating Missions as Nodes on the Internet (OMNI), "Summary of Recent UoSAT-12 IP Testing", 154 Experiment Team Meeting, July 26, 2000.

[2] Stevens W., "TCP/IP Illustrated Volume 1", Addision Wesley Publishing Company, 1995

[3] CCSDS, "Space Communications Protocol Standards (SCPS) Rationale, Requirements, and Application Notes – Draft Green Book", August 1998.

[4] Macker J. and Dang W., "The Multicast Dissemination Protocol (MDP) Version 1.6", Naval Research Lab Technical White Paper, April 1996.

[5] Kent S. and Atkinson R., "Security Architecture for the Internet Protocol", RFC 2401, IETF, November 1998

[6] Gleeson B., Lin A., Heinanen J., Armitage G., Malis A., "A Framework for IP Based Virtual Private Networks", RFC 2764, IETF, February 2000

[7] "MIL-STD-1553 & MIL-STD-1773", https:// ewhdbks.mugu.navy.mil/1553-bus.htm.

[8] Stallings W., "Data and Computer Communications", Fifth Edition, Prentice Hall Publishing Company 1997

[9] Stone J., Partridge C., "When the CRC and TCP Checksum Disagree", SIGCOMM 2000