Delay/Disruption-Tolerant Networking: Flight Test Results from the International Space Station

Andrew Jenkins, Sebastian Kuzminsky, Kevin K. Gifford
BioServe Space Technologies
University of Colorado
Boulder, CO 80309
303.492.0299
andrew.jenkins@colorado.edu, seb@highlab.com,
gifford@rintintin.colorado.edu

Robert L. Pitts, Kelvin Nichols
NASA Marshall Space Flight Center
MSFC
Huntsville, AL 35812
256.544.0666
lee.pitts@nasa.gov,
kelvin.nichols@nasa.gov

Abstract—The University of Colorado is working with NASA to extend Earth's internet into outer space and across the solar system. The new networking technology is called Disruption Tolerant Networking (DTN), and is being tested on the International Space Station. DTN will enable NASA and other space agencies around the world to better communicate with international fleets of spacecraft that will be used to explore the moon and Mars. This technology is evolving into an Interplanetary Internet.

In this paper we describe the design and features of the DTN-on-ISS implementation as well as reporting initial results from the experimental deployment. [1,2]

TABLE OF CONTENTS

1. INTRODUCTION

Communications in space are characterized by their disrupted, wireless nature. Whether due to occultation, scheduling, cost or solar flares, spacecraft must handle interruptions in connectivity and data losses [1]. Disruption Tolerant Networking (DTN) [2,3] can maximize the efficient use of links to and among spacecraft. As part of NASA's efforts to research DTN in space, the authors have deployed an implementation of the DTN Bundle Protocol (BP) [4] to a payload on the International Space Station (ISS).

This paper reviews networked space communication and provides the context of the DTN-on-ISS deployment. Next, it describes the requirements and features of the DTN-on-ISS implementation. Early results from DTN operations are presented and discussed. Constraints of the current system implementation are identified and enhancements to resolve these limitations are presented. Finally, a look forward to the future of DTN on the International Space Station and affiliated ground systems is presented.

2. RELATED WORK

Apollo astronauts on the lunar surface relayed voice communications through the Lunar Module [5]. The Command/Service Module, Lunar Module, and the Earth-based Manned Space Flight Network formed a communication network robust to failures and line-of-sight interruption. The Tracking and Data Relay Satellite System (TDRSS) [6] forms bent-pipe relays to Shuttles and the ISS. However, neither of these systems provides an automated *store-and-forward* capability.

The Mars neighborhood includes an increasingly dense mesh of orbiting and surface spacecraft that support secondary relaying activities simultaneous with primary science experiments [7]. The crucial store-and-forward buffer capacity is a constrained resource. As well, the human effort required to schedule communications among Mars assets and the DSN is significant.

The Interplanetary Internet Special Interest Group designed a space internet [8], and the Consultative Committee for Space Data Systems engineered interoperable protocols [9], underpinning the cross-layer, cross-mission, cross-agency support required for networked space exploration and science communication. These space-oriented groups collaborate with the Delay-Tolerant Networking Research Group [10], which includes terrestrial applications of DTN. The central protocol of the DTN architecture is the Bundle Protocol, BP, [4], which describes a mechanism for bundling data for store-and-forward delivery over a network that may face challenges in delay, asymmetry, disruption and power.

---

[1] 978-1-4244-3888-4/10/$25.00 ©2010 IEEE.
[2] IEEEAC paper #1279, Version 35, Updated 2009:12:01

Figure 1 – CGBA5 onboard ISS being attended to by Astronaut Terrence W. Wilcutt (from [15]).

The first experiments with the Bundle Protocol in space utilized the Disaster Monitoring Constellation [11,12] in January 2008. The Deep Impact Network Experiment (DINET) used the EPOXI (formerly DI) spacecraft and Earth nodes to simulate an Earth-Mars network in October 2008 [13,14].

These previous experiments demonstrated the feasibility of the bundle protocol and related technologies to form an automated store-and-forward overlay network among spacecraft and Earth assets. Similar to DMC and DINET, our deployment of DTN involved modifying software that is currently in-flight and has a primary purpose that is not DTN. The previous deployments were experimental and short-lived. In contrast, this paper describes an architecture to transition an ISS life science payload to use DTN for day-to-day operations for the remainder of its lifetime.

3. DTN-ON-ISS IMPLEMENTATION

The deployment of DTN and the Bundle Protocol to the International Space Station begins with developing and installing a bundle protocol agent (BPA, a bundle router) to the Commercial-Grade Bioprocessing Apparatus 5 (CGBA5, Figure 1). CGBA5 is primarily an environmental control chamber for life science experiments, but provides an embedded computational/communications platform with these characteristic features:

- 1 GHz Intel Celeron processor (32-bit).
- 1 GB RAM; 4 GB solid-state disk.
- Debian Etch operating system on Linux 2.6.21.

CGBA5 is remotely monitored from Boulder, CO in the Payload Operations Control Center (POCC). It may be commanded and controlled either by a crewmember on station, or remotely from the POCC, via the Huntsville Operations Support Center (HOSC).

Payload users like the CGBA5 team interface with ISS-to-Ground communications interface systems located in the HOSC. The HOSC provides two systems of interest: 1) The Payload Data Services System, PDSS, is used to transmit payload telemetry to remote control sites such as the Boulder POCC. Each payload makes use of an application identifier (APID) that maps the per-payload telemetry to a destination IP address and UDP port; and 2) The Enhanced HOSC Support, EHS, Remote Interface System (ERIS) is used for issuing commands and message acknowledgements from the ground to the on-orbit payload. Figure 2 also shows the payload Rack Interface Computer, RIC, onboard the ISS, which serves as an IP gateway to the ISS payload LAN.

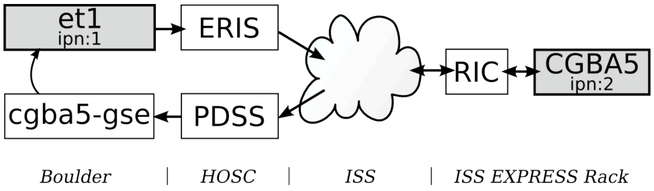

Figure 2 – The Ground-Space Disruption Tolerant Network between the CU-Boulder POCC and the ISS.

Non-DTN Operations

Before the deployment of DTN, remote operations involved scheduling sessions for issuing commands to CGBA5 days or weeks in advance. Commands were limited to 88-byte packets, and a typical session might include 5 or fewer commands (440 bytes a week). *Commands* could not be used to provide communications feedback, so *telemetry* from CGBA5 was delivered via a "transmit-in-the-blind" mechanism:

- Always transmit per-second *health and status*.
- Service longer-term science and status through a priority-based repeat system named *playback*.

The playback system downlinks the newest telemetry files first and uses the remaining bandwidth to repeat older files. The downlink path is interrupted by losses in connectivity between the ISS and Earth due to normal TDRSS handovers, as well as losses experienced as UDP packets traverse a congested Internet from Alabama to Colorado. To compensate for losses without feedback, the telemetry files may be replayed hundreds or thousands of times. For files in which the very first attempt was successful, this represents a large overhead of useless retransmissions as depicted in Figure 3.

A custom framing and multiplexing protocol named *channel* is used to support multiple applications on the payload. Each CGBA5 application (such as playback) utilizes its own channel, just like a UDP port. In addition, CGBA5 applications can submit data units up to 2048 bytes regardless of the underlying RIC frame size, as shown in Table 1.

Table 1 – Channel Characteristics for CGBA5

	Packet Size (bytes)		Bandwidth
	RIC	Channel	(bits/s)
Uplink	96	2048	150
Downlink	1248	2048	400,000

Disruption-Tolerant Operations

The first step in Disruption-Tolerant operations is adding a feedback link to increase the downlink efficiency of the payload. This enables a simple form Automatic Repeat request (ARQ) that is delay and disruption tolerant. This system preserves reliability with much less overhead, thus allowing for higher fidelity science operations because the downlink is used much more efficiently.

The HOSC extended their systems to support a new class of commands that may only carry feedback acknowledgments. These are administrative bundles (custody signals and status reports) in the language of the Bundle Protocol. The ISS Payload Rack Officers at the HOSC can readily disable or enable these DTN acknowledgements without advance notice to the payload team in response to higher priority commanding requirements.

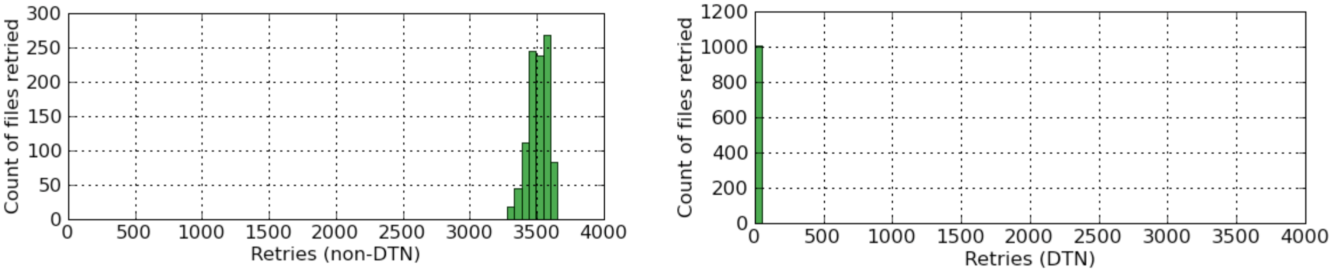

Figure 3 – Histograms of redundant space-to-ground transmissions for three days (N=1008) of typical non-DTN (left) and DTN (right) operations. In the non-DTN case, files are received on the ground between 3276 and 3651 times. In the DTN case, no file is received on the ground more than four times, 96% of files are received only once.

With this link, enhancements to payload software allow monitoring of the DTN software and new telemetry software adapted to the DTN architecture. Modifications to operations procedures and software accompany the new DTN software. The DTN software communicates via its own channels in the multiplexing system, so legacy commanding and telemetry systems are still supported.

The payload and POCC Ground Support Equipment (GSE) software's DTN capabilities are built around the Interplanetary Overlay Network (ION) software originally developed at the Jet Propulsion Laboratory. Some versions of ION are publicly available through Ohio University,[3] including the version utilized by CGBA5.

In addition to the downloadable releases, some specialization has been performed by the authors. Specialization that consists of bug-fixes and the addition of new capabilities described in this paper that are not present in ION are publicly available[4], either under the ION license or as free software. Specialization that adapts ION to CGBA5 is not available, in compliance with ITAR and EAR restrictions. The ION software suite is desirable for many embedded applications because it is under active development, and is lightweight (incorporating features deemed most relevant to space applications). The total uplink size of the packaged version of ION used on CGBA5 (including extra applications) is 524kB. Table 2 outlines the sizes of the major components of ION for a recent uplink (build82). While some ION components are unused on CGBA5, they are uplinked anyway since they are a part of the normal ION distribution, and may be used at a later date.

4. OPERATIONS DATA

Deployment to the International Space Station began in June 2009. After a checkout, the first experiments occurred on July 10, 2009 [16,17,18] and involved downlinking images of a previous CGBA5 experiment where a metal salt is added to a silicate solution and insoluble silicates form. A frame of the experiment was sliced into small pieces, and these pieces were downlinked over a disrupted space-to-

Table 2 – The size of ION components on CGBA5

	Size	%
Uncompressed	1638400	100%
Compressed Debian package	535680	33%
bping, bpchat, bpstats2	30936	2%
Core executables	720060	44%
Library code	370176	23%
Packaging support	19328	1%
AMS (unused)	271436	17%
LTP, UDP, TCP CLAs (unused)	226464	14%

---

[3] https://ion.ocp.ohiou.edu

[4] Utilities are at http://bioserve.colorado.edu/www/2009/06/test-apps-for-ion-available . Bug-fixes are discussed on the ion-users mailing lists.

ground link. Figure 4 shows an example frame from the video, and the full video is available online[5].

This initial deployment demonstrated the success of the bundle protocol in handling disruptions. The payload (data sender) had no feedback regarding the state of the space-to-ground link, and the experiment was chosen to occur over a planned TDRSS handover. During TDRSS handovers, the space-to-ground and ground-to-space links experience disruptions on the order of several minutes. The payload responded to this disruption as designed, by custodial retransmission after a configurable timeout.

The next evolution of the DTN-on-ISS network involved using the DTN for unattended operations. The payload would downlink its status telemetry files via the non-DTN transmit-in-the-blind configuration as well as via a DTN configuration. Figure 3 shows the result for a 3-day period in which 14 files an hour were generated. In this period, the non-DTN scheme resulted in an average of 3504 redundant receptions per file. The DTN scheme performed much better at an average of 0.06 redundant receptions per file. While many automatic repeat request (ARQ) systems would provide similar benefits over an inefficient transmit-in-the-blind scheme, they do not meet the same interplanetary networking goals as DTN.

Operations Lessons Learned

Throughout this experiment, the operations personnel at the HOSC supported the investigators at the University of Colorado, and communication was key. From the perspective of HOSC and MCC flight controllers, DTN custody signals are payload commands. The concept of autonomously generated payload commands requires close scrutiny from those who are charged with protecting the lives of the crew and the success of the ISS mission as a whole.

5. CUSTODY SIGNAL COMPRESSION

The need for custody signal compression

To downlink reliably in spite of disruptions of many minutes, the CGBA5 bundle agent transmits bundles with custody transfer. In this mode, a bundle agent who has *custody* of a bundle will retain custody (become the *custodian)* until it receives a signal from the next custodian. While custody is retained, the bundle agent will retain a copy of the bundle. If it believes custody transfer has failed (due to an external event like link-layer notification or internal *custody transfer countdown timer* timeout), it may attempt to re-forward the bundle, possibly selecting a different route or otherwise using more robust communication.

A bundle agent is notified that another agent has taken custody of a bundle via a custody signal. Custody signals themselves are bundles with a special payload layout. Unlike link-layer acknowledgements, the next custodian is not necessarily the next bundle hop; if a bundle agent is unwilling to take custody (perhaps due to storage constraints or low power) it can forward the bundle along in the hopes that a bundle agent further along the route will accept custody.

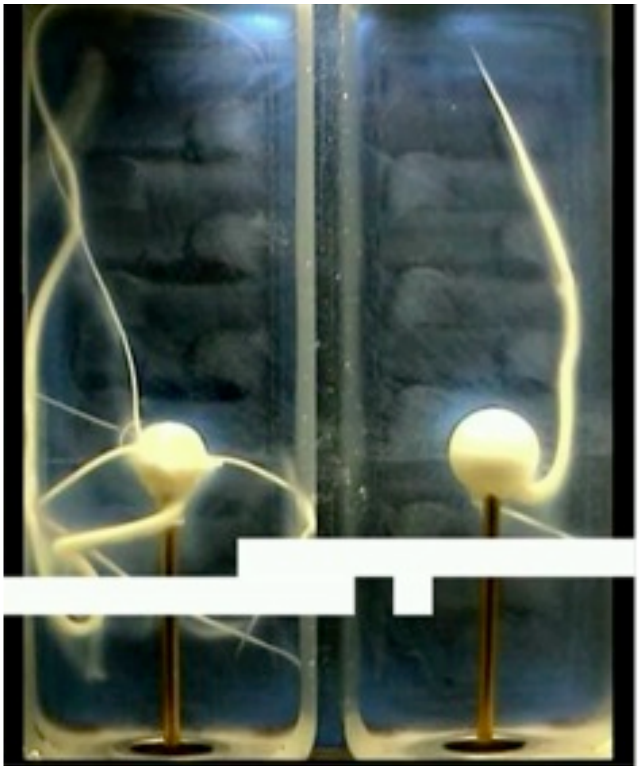

Figure 4 – A snapshot from a video of DTN downlinking images of insoluble silicates in spite of disruptions.

The downlink from CGBA5 can send any number of bundles, up to 400 Kbits/second (bundle headers and payload). To share bandwidth with other ISS payloads, we limit the uplink to CGBA5 to one bundle every 5 seconds, with 90 bytes available for the bundle header and any payload (in comparison with Table 2.4, we are here assuming that 6 bytes of RIC payload are used for channel header; 96-6=90). The downlink has about 2800 times more bandwidth than the uplink.

This link asymmetry is justified by design for non-networked spacecraft. Before DTN, ISS payload users were not permitted to send automated network acknowledgments to their payloads. If the Interplanetary Internet is to be successful, this kind of asymmetry (especially but not exclusively present in legacy systems) must be accommodated. In order to efficiently utilize the CGBA-5 downlink channel, any acknowledgment (custody signal or otherwise) uplinked must be smaller than the data it is acknowledging by at least a factor of 2800. When considering custody transfer as an acknowledgment mechanism, calculations indicate that the minimum efficient

---

[5] http://bioserve.colorado.edu/~andrew/dtncanvas

downlink bundle size for the CGBA5 communications link is given below:

Equation 1:

$$\overbrace{\underset{\text{CBHE BPB}}{(24} + \underset{\text{Admin Block Hdr}}{3} + \underset{\text{Custody Sig}}{22)}}^{\text{Minimum CT Signal Bundle Size (49 Bytes)}} \cdot 2800 = 140\text{KB}$$

One of the first applications on CGBA5 that we are attempting to switch from using our “channel” transport protocol to using BP is called playback. This application transmits files containing telemetry data from CGBA5 to the GSE computer. The data files are stored, compressed and transmitted. The average file size is approximately one kilobyte. Our “send files over BP” application puts each file into its own bundle, and sends it with custody transfer enabled. For a 400 kbps downlink telemetry rate we have:

Equations 2 & 3:

$$1\frac{\text{file}}{\text{bundle}} * 1\frac{\text{KiloByte}}{\text{file}} * 8\frac{\text{bits}}{\text{Byte}} = 8\frac{\text{kilobits}}{\text{bundle}}$$

$$400\frac{\text{kilobits}}{\text{sec}} / 8\frac{\text{kilobits}}{\text{bundle}} = 50\frac{\text{bundles}}{\text{sec}}$$

Per section 5.10.1 of RFC 5050, a Bundle Agent that accepts custody for a bundle must generate a “Succeeded” custody signal for the bundle and send it to the bundle’s current custodian. Thus the GSE must send 50 custody signals per second back to CGBA5 on ISS:

Equation 4:

$$50\frac{\text{CT Signals}}{\text{sec}} \cdot 49\frac{\text{bytes}}{\text{CT Signal}} = \overbrace{2500\frac{\text{Bytes}}{\text{sec}}}^{\text{Uplink Req}} \gg \overbrace{18\frac{\text{Bytes}}{\text{sec}}}^{\text{Uplink Avail}}$$

As cited in Eq (1), the minimum size of a CBHE-encoded primary bundle block (for a bundle larger than 16KB) is 24 bytes. The minimum size of the Administrative Block Header is 3 bytes. The minimum size of a bundle payload block containing a custody signal for a bundle with a small CBHE source is 22 bytes.

A mechanism for Custody Signal Compression

We propose extending the bundle protocol to support a new kind of custody signal. This new custody signal will accept custody for multiple bundles, using a compact encoding. An example compression based on ideas from TCP Selective Acknowledgment [19] is shown in Figure 5.

CGBA5 could downlink 50 bundles per second if not limited by GSE uplink custody signal bandwidth. Using the example compression, the GSE could uplink one custody signal bundle that signaled reception of 50 out of 51 bundles each second, spanning 5 seconds, with an additional 23 bytes above the uncompressed custody signal. Thus, a new nominal custody signal bundle size (for 250 bundles) would be (compare to Equation (1)):

Equations 5 & 6:

$$\overbrace{\underset{\text{CBHE BPB}}{(24} + \underset{\text{Admin Block Hdr}}{3} + \underset{\text{Comp Custody Sig}}{45)}}^{\text{CT Signal Bundle Size (250 signals, 72 Bytes)}} \cdot$$

$$2800 \cdot \frac{1 \text{ CT Sig Bundle}}{250 \text{ Bundles}} = 0.80\text{KB}$$

Thus, the optimal bundle must be no smaller than 800 bytes. This allows the downlink channel to be fully utilized by normal CGBA bundle sizes (1 KB > 800 B) without limitation by the uplink channel.

The bundle protocol does not suggest that bundles forwarded to a particular bundle agent are forwarded in sequential order, which would provide the maximum compression. However, we note that implementations of bundle agents are free to perform this optimization if they wish, and even if they do not, the overhead of bundle headers and endpoint identifiers is much greater than that of sequence numbers.

A simple mechanism for interoperating with bundle agents that may not support compression is needed. A compression-enabled agent should send a compressed custody signal if it doesn’t know that the custodian supports compression. If the custodian later re-forwards any bundles that were mentioned in the compressed signal, the compression agent should assume that the custodian doesn’t support compression and send uncompressed custody signals. This technique is vulnerable to false-negatives if custody signals are lost in the network, and more advanced heuristics are possible.

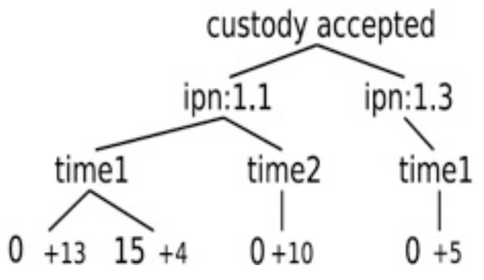

Figure 5 – Experimental custody transfer compression encoding for 36 bundles in one CT signal. The bundle from ipn:1.1 at time1 with sequence number 14 is not acknowledged.

6. CUSTODY TRANSFER RETRIES

Custody transfer is a mechanism to promote reliability that is described in RFC5050 and is described as an open research area [20]. It designates custodians in the network that have an elevated responsibility to deliver bundles. An example scenario that benefits from custody transfer is presented in Figure 6. Here, node 1 is sending a bundle to node 3, and node 2 is a custodian (this would be reasonable in the case where link A is expensive so end-to-end retries should be avoided if possible). Link A is unidirectional, so node 2 can only send acknowledgments (like custody signals) through link B, and then through a disrupted network. In this case, bidirectional acknowledgments (as used in TCP and LTP) are not possible. End-to-end retries are undesirable due to the cost of link 1. Custody transfer, namely retries from 2 to 3 without involving 1, is a solution.

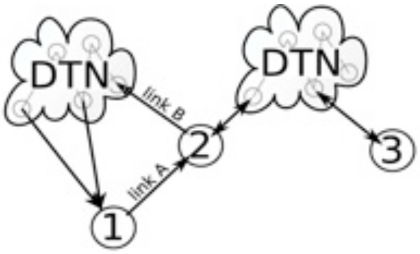

Figure 6 – A network of bundle agents for which convergence layers cannot provide reliability. Node 1 has an expensive unidirectional link A to a custodian, node 2. Node 2 can only provide custody signals back to node 1 through link B and a disrupted network.

Future disruption-tolerant networks may retry custody transfer at intervals that are informed by routing, contact schedules, network management events, CLA events, or other sources. While the details of exchanging bundles for custody signals are described in RFC5050, there are many aspects not described and not yet understood for DTNs like the IPN: when should custodial retransmission be triggered, who should be a custodian, and whether the benefit of custody transfer is worth the extra effort.

As a first step, RFC5050 describes an optional mechanism, which triggers retransmissions based on the expiration of a Custody Transfer Countdown Timer, or CTCT. BioServe has added a simple timeout-based CTCT to the BioServe branch of ION. This CTCT leverages custody transfer retry support developed by JPL. Research into more intelligent CTCTs is ongoing.

7. CONCLUSIONS

In this paper we have described a significant NASA flight experiment with the goal of maturing the DTN technology to enable *networked* space-based communications. The project, while still in its infancy, has wide-spread implications as the basis for an Interplanetary Internet.

We have identified two key areas to address for DTN protocol improvement utilizing selective acknowledgements and improving custody transfer algorithms. From a practical network management perspective, we advocate for in-depth instrumentation to aid in the assessment of the DTN protocol suite performance.

8. FUTURE WORK

Next, the DTN-on-ISS network will be expanded to include another CGBA payload called CGBA4; this will expand the network to 2 space nodes and 2 ground nodes and enable experimentation with cross-node routing and one-way custody transfer.

For the results described in this paper, the HOSC supports DTN by augmenting its command infrastructure for autonomous, delayed and disrupted commanding but does not perform the higher-layer responsibilities of a bundle protocol agent such as bundle routing, custody transfer, bundle quality of service, or the bundle security protocol. The HOSC is extending its DTN capabilities to include a bundle protocol agent located at the HOSC. In addition, the HOSC bundle protocol agent will not be ION-based so it will provide important long-term interoperability testing between bundle protocol implementations. In the first stage of this project the CGBA5 payload will function as the DTN Gateway onboard the ISS and will proxy communications from other ISS payloads to the HOSC.

Compressed custody signals and custody transfer countdown timers remain active areas of research for the authors.

The Japanese Aerospace Agency, JAXA, has their own experiment module ("Kibo") at the ISS. NASA, JAXA, and CU-Boulder have developed an initial plan to deploy a DTN node at the JAXA mission control center in Tsukuba (near Tokyo) that can communicate via DTN to the CGBA5 payload. Interestingly, this experiment enables DTN communications over both the NASA TDRSS link to the ISS and the JAXA DRTS communications link to ISS.

The METERON (Mars End-To-End Robotic Operations Network) project is interested in utilizing DTN as the base network technology for rover exploration of Mars. Additionally, ESA has its own experiment module onboard ISS ("Columbus") for possible execution of DTN experiments similar to the NASA-JAXA DTN experiment concept.

9. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of Adrian Hooke and the NASA DTN Readiness Group, Dr. Vint Cerf, and the helpful support of Scott Burleigh, David Young, Keith Scott and others in the NASA ION Working Group. This work was performed under NASA contract **NNJ05HE10G.**

REFERENCES

[1] Robert C. Durst, Patrick D. Feighery and Keith L. Scott, "Why not use the Standard Internet Suite for the Interplanetary Internet?" Available from: http://www.ipnsig.org/techinfo.htm.

[2] Vinton G. Cerf, Scott Burleigh, Adrian Hooke et al., "Delay-Tolerant Networking Architecture," Internet Research Task Force RFC 4838, April 2007.

[3] Forrest Warthman, "Delay-Tolerant Networks (DTNs): A Tutorial v1.1, " Available from: http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf, March 2003.

[4] Keith Scott and Scott Burleigh, "Bundle Protocol Specification," Internet Research Task Force RFC 5050, April 2007.

[5] John H. Dabbs and Oron L. Schmidt, "Apollo Experience Report – Voice Communications Techniques and Performance," NASA Technical Note D-6739, March 1972.

[6] J. Teles, M.V. Samii and C.E. Doll, "Overview of TDRSS," *Adv. Space Res.* 16, (12)67-(12)76, 1995.

[7] Roy Gladden et al., "Mars Relay Coordination Lessons Learned," IEEE Aerospace Conference Proceedings, March 5-12 2005.

[8] The Interplanetary Internet Special Interest Group, http://www.ipnsig.org .

[9] Consultative Committee on Space Data Systems, http://www.ccsds.org .

[10] Delay Tolerant Networking Research Group, http://www.dtnrg.org .

[11] Lloyd Wood et al., "Use of the Delay-Tolerant Networking Bundle Protocol from Space," 59th International Astronautical Congress, September 2008.

[12] Will Ivancic et al., "Delay/Disruption-Tolerant Network Testing Using a LEO Satellite," 8th NASA Earth Science Technology Conference, June 2008.

[13] Ross M. Jones, "Disruption Tolerant Network Technology Flight Validation Report," JPL Publication 9-2, February 2009.

[14] Jay Wyatt, Scott Burleigh, Ross Jones, Leigh Torgerson and Steve Wissler, "Disruption Tolerant Networking Flight Validation Experiment on NASA's EPOXI Mission," 1st Intl. Conference on Advances in Satellite and Space Communications, 2009.

[15] Experiment/Payload Overview: Delay-Tolerant Networking. ISS Program Scientist's Office. Available from: http://www.nasa.gov/mission_pages/station/science/experiments/DTN.html .

[16] Lester Haines, "NASA fires up the 'interplanetary internet'," The Register, 7th July 2009, Situation Publishing, London UK. Available from: http://www.theregister.co.uk/2009/07/07/dtn_node/ .

[17] Brian Handwerk, "'Space Internet' to Link Worlds by 2011?" National Geographic News, 9th July 2009, National Geographic Society, Washington, D.C. Available from: http://news.nationalgeographic.com/news/2009/07/090709-space-internet.html .

[18] Terry Jessup, "CU Researchers Working On Outer Space Internet," CBS 4 Denver, 7th July 2009, Denver, Colorado. Available from: http://cbs4denver.com/local/CU.researchers.outer.2.1075798.html .

[19] Matt Mathis, Jamshid Mahdavi, Sally Floyd and Allyn Romanow, "TCP Selective Acknowledgement Options," Internet Engineering Task Force RFC 2018, October 1996.

[20] Kevin Fall and Stephen Farrell, "DTN: An Architectural Retrospective," IEEE Journal on Selected Areas in Communication, Vol. 26 No. 5, June 2008.

BIOGRAPHY

Andrew Jenkins *is a software developer with the BioServe Space Technologies Automation Group. He has developed software and hardware for scientific payloads for deep-space, Earth- and Sun-observing, and manned missions. He has B.S. and M.S. degrees in Electrical and Computer Engineering from the University of Colorado at Boulder.*

Sebastian Kuzminsky *is a software developer with the BioServe Space Technologies Automation Group. He has developed communications, command, and control software for spaceflight for the past 15 years and is the lead software architect for the BioNet-DTN command, control and communications system.*

***Kevin Gifford** is a researcher with BioServe Space Technologies at the University of Colorado in Boulder Aerospace Engineering Department. He has developed communications, command, and control software for spaceflight for the past 15 years. Kevin is the Principal Investigator for the BioServe DTN-on-ISS project and for the open-standard BioNet distributed command, control and communications architecture.*

***Robert L. Pitts** is the Lead System Engineer for the Huntsville Operations Support Center (HOSC). He is employed by the Computer Science Corporation and responsible for the initial and ongoing design, development, maintenance and support of HOSC systems and the continued operational support that it provides to NASA's Space Shuttle, Space Station, and Constellation programs.*

***Kelvin Nichols** is an engineer with NASA at the Marshall Space Flight Center located in Huntsville, Alabama. He has worked with the Payload Operations and Support Center (POIC) in support of the International Space Station's (ISS) voice, command, and telemetry ground distribution systems for the past 9 years.*