Delay/Disruption Tolerant Networking for the International Space Station (ISS)

Adam Schlesinger
NASA JSC
Houston, TX 77058
281-483-6713
Adam.m.Schlesinger@nasa.gov

Brett M. Willman
NASA JSC
Houston, TX 77058
281-483-6713
Brett.m.willman@nasa.gov

Lee Pitts
NASA MSFC
Huntsville, AL 35802
256-544-0666
Robert.l.pitts@nasa.gov

Suzanne R. Davidson
The Boeing Company
Houston, TX 77059
832-224-7306
Suzanne.r.davidson@boeing.com

William A. Pohlchuck
The Boeing Company,
Houston, TX 77059
832-224-7302
William.a.pohlchuck@boeing.com

***Abstract*—Disruption Tolerant Networking (DTN) is an emerging data networking technology designed to abstract the hardware communication layer from the spacecraft/payload computing resources. DTN is specifically designed to operate in environments where link delays and disruptions are common (e.g., space-based networks). The National Aeronautics and Space Administration (NASA) has demonstrated DTN on several missions, such as the Deep Impact Networking (DINET) experiment, the Earth Observing Mission 1 (EO-1) and the Lunar Laser Communication Demonstration (LLCD). To further the maturation of DTN, NASA is implementing DTN protocols on the International Space Station (ISS). This paper explains the architecture of the ISS DTN network, the operational support for the system, the results from integrated ground testing, and the future work for DTN expansion.**

TABLE OF CONTENTS

1. INTRODUCTION 1
2. ISS DTN ARCHITECTURE....................................... 2
3. TEST RESULTS... 3
4. OPERATIONAL SUPPORT OF THE DTN SYSTEM.... 7
6. FORWARD PLAN 9
7. SUMMARY 10
ACKNOWLEDGEMENTS.. 10
REFERENCES 10
BIOGRAPHY 10

1. INTRODUCTION

In October 2010, the Interagency Operations Advisory Group (IOAG), consisting of NASA and major international partners, published an operations concept for a Solar System Internetwork (SSI), recommending a timely evolution toward a fully operational SSI [1]. An SSI is necessary, as future space exploration concepts will introduce much more complex topologies, with data transfer over multiple hops via relay spacecraft and other intermediate nodes. Like the terrestrial Internet, the SSI will offer users a well-defined, standardized platform upon which to build a wide variety of applications by accessing end-to-end network services. The SSI will utilize the Delay/Disruption Tolerant Networking (DTN) protocol suite, which can be used in any scenario, including those with longer light times or frequent link disruptions, where conventional Internet Protocols (IP) fail.

DTN is an architecture and protocol suite designed to abstract networks of distinct protocols that are connected via disruptive and latent links [2]. The ISS is no outlier to either issue. Orbiting at approximately 300 km and travelling at 17,000 mph, ISS's primary high bandwidth data downlink path is connected via the geosynchronous Tracking and Data Relay Satellite System (TDRSS) Ku-band system. Since each ISS orbit varies, TDRSS coverage and data latency varies; coverage is predictable, but not continuous and latency can range from approximately 0.6 seconds to 1 second.

Given the non-continuous and latent architecture of the data network, most common Commercial Off-The-Shelf (COTS) protocols do not operate as expected; thus, NASA has developed custom protocols to tackle the issues. For example, NASA has developed a custom file transfer software called Software to Ready Data Files to Send Hastily (SWRDFSH). It is software that is not too dissimilar from an automated File Transfer Protocol (FTP) service for the Flight Control Team (FCT), but it deals with uniqueness of the ISS data network environment. For other systems, the FCT meticulously schedules payload data downlinks to avoid conflicts with

978-5090-1613-6/17//$31.00 ©2017 IEEE

network outages due to TDRSS connectivity. This demands constant attention from not only the FCT, but also the Payload Investigators (PIs) must understand how and when their data is downlinked. Seeing into the future, NASA understands that custom protocol suites and manual scheduling are not long term solutions for coordinating data downlinks in this dynamic environment. DTN is one solution.

The ISS Program is implementing DTN as a service to the Payload and Operations community. This paper describes the implementation of the institutional DTN service on ISS, highlighting the architecture design and characterization of the DTN system from flight equivalent testing that has been performed. The ISS institutional DTN implementation greatly enhances the reliability of payload science data transmissions and reduces operational overhead and planning, as well as providing an architecture to support mission applications while advancing DTN technology support for the future SSI.

2. ISS DTN ARCHITECTURE

The ISS is a unique environment and presents an opportunity where the IP systems, traditional Consultative Committee for Space Data Systems (CCSDS), and DTN coexist and support payloads. DTN can be represented as a mesh network, as shown in Figure 1, utilizing CCSDS Bundle Protocol (BP) [3] and Licklider Transmission Protocol (LTP) [4] across ground and space links to the ISS. Most nodes should have alternate paths around failed adjacent nodes. Intermittent connections can be found between payload users and the Huntsville Operations Support Center (HOSC) and between the HOSC and ISS.

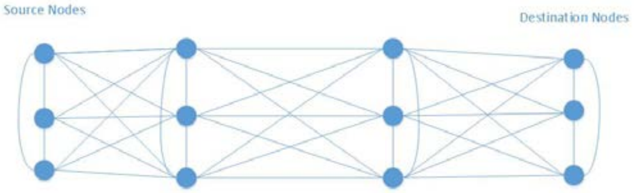


Figure 1. DTN Topology

Data flows are bidirectional. Due to connection restrictions, uplink and downlink are constrained and highly asymmetric. Payloads and payload developers do not have capacity or knowledge to support multiple paths and the HOSC to ISS link is a singleton over the TDRSS Ku-band. Therefore the ISS DTN system is as shown in Figure 2.

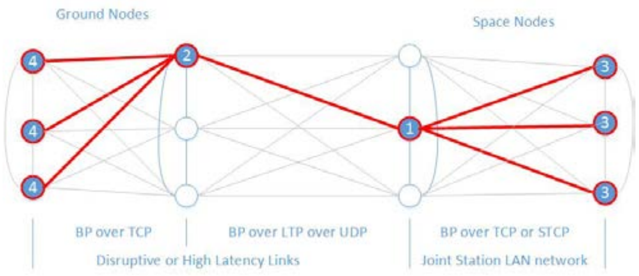


Figure 2. ISS DTN Topology

ISS DTN was developed with specific Convergent Layer Adapters (CLA) for each link; with BP running over Transmission Control Protocol (TCP), LTP, User Datagram Protocol (UDP), and Simple TCP (STCP). The goal was to be as compliant with the CCSDS specification as possible and use the most common and applicable service.

Ground nodes communicate with each other over a reliable service: TCP. Payload control centers (whether they are a "guy in a garage" or a Telescience Center) communicate with the HOSC via bundle protocol over a TCP session. This is true whether they have a contact with their payload or not. Uplink uses encapsulation of UDP data within the Advanced Orbital Standard (AOS) protocol. Embedded in the encapsulation is LTP. It provides a TCP-like connectivity over the space link. Some additional services and capabilities from the CCSDS specification support payloads, specifically:

1. Compressed Bundle Header Extension (CBHE) supports more efficient bandwidth usage [5]
2. CCSDS File Delivery Protocol (CFDP) provides reliable file transfers [6]
3. Persistent data stores allow loss of compute capability on the ground without loss of bundles

Additional features from the CCSDS specification will be available at a later date to facilitate payload usage. Provided by the specification is Aggregate Custody Signals (ACS) for more efficient use of the forward link [3]. ACS compresses acknowledgement of bundles (data) and may reduce maximum DTN uplink traffic as a function of custody downlink by a factor of **8:1**. For the ISS shared link which is highly asymmetric on the order of **30:1,** this may substantially reduce required contact times. Another capability is Delay Tolerant Payload Conditioning (DTPC) which supports in-order bundle delivery [3]. DTPC is an application service, which in the DTN environment provides end-to-end TCP like connectivity. Therefore special software may not be required to reorder data for a COTS application.

The architecture of Figure 3 supports DTN users locally and remotely interfacing with the HOSC. Payloads may use the capability if they have compliant DTN nodes. As shown in the figure, Payload Investigators or HOSC Cadre members can utilize DTN services by requesting Ku-band IP services and connecting to the HOSC Payloads Gateway. Data flows can be initiated from the vehicle to the ground or in reverse.

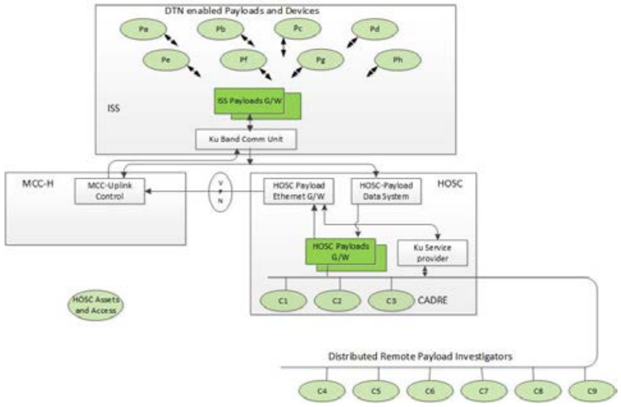


Figure 3. ISS DTN Architecture

More specifically, DTN nodes on the ground and in space are *commodity* items, readily available and easily procured. There are three specific requirements supporting the bundling protocol: access to a reliable storage service, forward advancing time, and reliable communication services. These services are provided by most modern systems.

NOTE: The circled numbers referenced in the paragraphs below are depicted in Figure 2.

Therefore, the onboard platform ① of choice for the DTN gateway is the ISS standard computing platform, an IBM (Lenovo) T61P laptop. The platform modifications are mostly cosmetic and onboard has 4 GBytes of RAM and a 150 GB hard drive on a Virtual LAN that is accessible to the Joint Station LAN (JSL) or payload VLAN. The Operating System (OS) of choice is a Debian Virtual Machine. A virtual machine was chosen to allow ease of delivery and recovery. ION 3.3.1 is the open-source version of the bundle protocol developed by the Jet Propulsion Lab (JPL) and was integrated by the ISS program for payload use.

The HOSC decided to integrate the ground gateway ② into the current command and control system to minimize management interaction with payloads, giving them near unobstructed access to DTN services. The gateway is a RedHat™ LINUX virtual machine using DTN2.9 software. It has been modified to be highly interoperable and supports the HOSC Cadre in uplink management. The virtual server is composed of 6 cores, 8 GBytes of RAM, and a persistent data store of 120 GB. This provides a bundle storage capacity of 4.4 hours at a 60 Mbps continuous downlink rate. DTN2.9 was selected because of its "open source" origins and its unique features above the basic bundle protocol.

For ground client ④ use, the ISS has incorporated ION 3.3.1, JPL's DTN software, into Telescience Resource Toolkit (TReK) and EPC (Enhanced HOSC System (EHS) PC). TReK is the HOSC remote operations solutions and is widely used by the payload community both nationally and internationally. TReK is available in Windows™ and LINUX releases. It can be used to monitor and control ISS payloads from anywhere in the world. It comprises a suite of software applications that provide generic data system capabilities to include CCSDS DTN and access to HOSC services. The EPC client is primarily used by the HOSC Cadre but is available for remote users. EPC is hosted on Windows™ and is complementary with a current release of EHS, the ISS Payload control center software. However, a user may have a DTN version to support a specific application. This is allowed as long as the basic bundle protocols are supported on the gateways and a compatible client is provided onboard.

An onboard DTN client ③ can be of various origins and hosted on numerous operating systems. There are numerous DTN implementations including the ISS baseline of DTN2 and ION, as well as IBR-DTN and JDTN to name a few. The ISS program is utilizing a TReK devised toolkit to support on-orbit clients using DTN and other options for managing payloads for Windows™ and LINUX based systems. When the TReK toolkit is incorporated on a payload, such features as CFDP, DTN, and web based management of payloads is possible with a pre-tested suite of tools:

- Send/Receive data using standard IP protocols (Unicast, TCP Listener, TCP Server, TCP Client).
- Create, populate, build, and decompose packets.
- Transfer files (send and receive) using CFDP.
- Configure and manage (start, stop, monitor) ION DTN node.
- Support Payload Health and Status.

3. Test Results

A. Certification Test Setup

Certification testing for the DTN system included a multi-site configuration including on-orbit software and equivalent flight hardware. The ISS components were simulated in the Johnson Space Center (JSC) Sonny Carter Training Facility (SCTF) in Houston, Texas. The ground components were simulated in both JSC SCTF and the HOSC in Huntsville, Alabama. Figure 4 shows the setup.

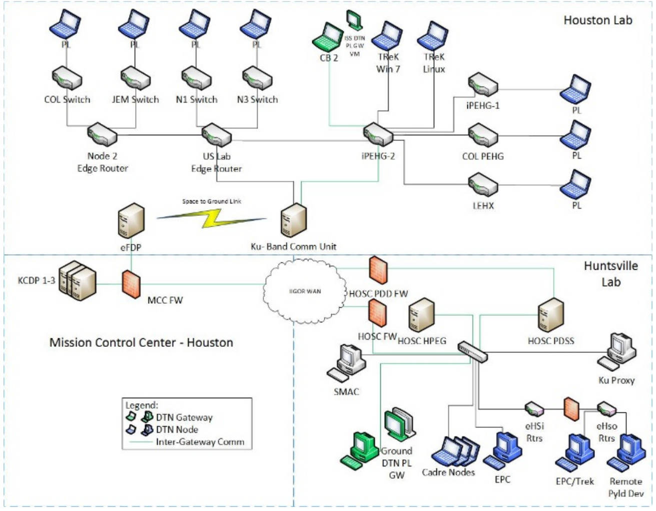


Figure 4. DTN Test Setup

Laptops represent DTN nodes/endpoints for on-orbit and ground. The ISS DTN PL Gateway represents the on-orbit gateway that communicates with various DTN nodes including Express Laptop Computers (ELCs) and TReK laptops. The on-orbit system communicates with the ground through an on board Ku-Band communications unit and ground distribution system. The HOSC and Mission Control Center (MCC) represent ground nodes. The HOSC includes a DTN gateway node that communicates with various ground DTN nodes including internal HOSC Cadre and TReK nodes. For the DTN testing, flight equivalent hardware was used to simulate the ISS system with a high degree of fidelity. Flight equivalent hardware was used for the Joint Station LAN, HOSC, and the Ku-Band communications system.

The ISS PL DTN Gateway used ION DTN on Debian Squeeze and Windows 7 operating systems. The HOSC ground gateway used a RedHat™ LINUX OS that runs DTN2.

Various tools were used to test the DTN system. Tools provided by the ION software include:

- bping/bpecho – used to verify basic connectivity similar to a "ping" in an Ethernet environment.
- bpdriver/bpcounter – used to generate and receive data bundles to verify data transmissions. The bpdriver/bpcounter tools allow sending a specified number of bundles of a specified size between two nodes. The tools are used to verify receipt of bundles and throughput.
- cfdptest – used to send/receive files between nodes using CFDP.

Other tools included the DTN2 status tool, used to verify bundle counts and dropped bundles on the ground gateway. The iperf tool was used to verify effective bandwidth between nodes. The TReK toolkit was used to simulate payload operations. The TReK toolkit was primarily used for sending files of various sizes using CFDP.

B. *Certification Test Descriptions*

The ISS DTN system testing included the following test categories: nominal payload DTN communication (Short Duration), nominal (uplink/downlink) DTN with a Loss of Communication, and CFDP file transfers. Bundle throughput and retransmission counts were examined and file transfers were executed.

Communication outages during the test were performed using two different techniques. One technique used in the dry run certification tests was by use of a Wide Area Network (WAN) emulator. Losses of 100% were introduced on both the uplink and downlink at the WAN emulator. The WAN emulator also provides a mechanism to introduce a fixed transmission delay into the system. Up to a 600 ms delay was introduced using the WAN emulator to simulate satellite and round-trip delay. The second technique for simulating a communication outage involved powering off a component of the on-orbit space to ground communications unit.

The tests to validate the DTN system utilized a configured space to ground downlink rate of 20 Mbps and an uplink rate of 4 Mbps. The maximum allowable bandwidth for the ISS system is currently 90 Mbps downlink for DTN-related data, although generally it is limited to about 30 Mbps. The maximum allowable uplink bandwidth is around 20 Mbps.

C. *Nominal Payload DTN Test*

The first part of the Nominal Payload DTN (Short Duration) test consisted of single node to node communication between a number of on-orbit and ground DTN nodes. The test utilized the ION provided bpdriver/bpcounter tools to test throughput. The bpdriver tool generates a specified number of bundle streams at a specified bundle size. Table 1 illustrates the results for a bundle size of 100000 bytes. 100 MB of data were sent in each case.

Table 1. Nominal Payload DTN (Short Duration) Single Node to Node Communication

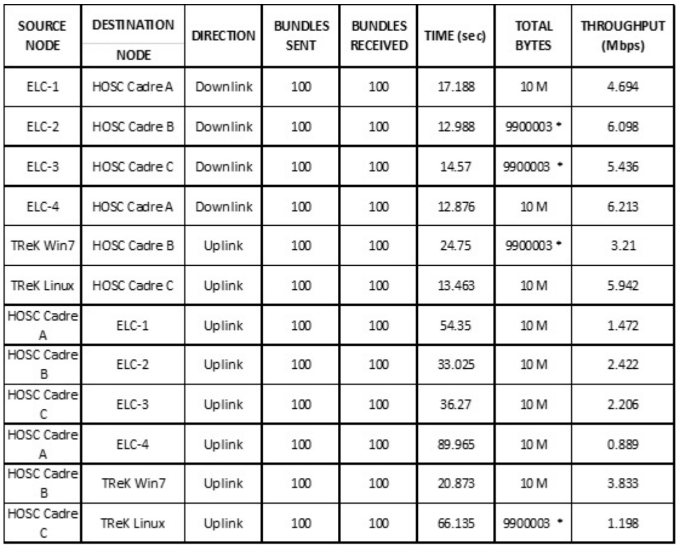
SRC	DEST	DIRECTION	BUNDLES SENT	BUNDLES RECEIVED	TIME (sec)	TOTAL BYTES	THROUGH-PUT (Mbps)	DATA SEGMENTS (Total/Rexmit)	REPORT SEGMENTS (Total/Rexmit)
TReK Windows	HOSC Cadre B	Downlink	500	500	23.987	50 MB	16.675	3501/0	501/0
TReK Linux	HOSC Cadre C	Downlink	500	500	24.165	50 MB	16.533	3501/0	501/0
HOSC Cadre B	TReK Win7	Uplink	500	500	100.792	50 MB	3.969	36002/0	502/0
HOSC Cadre C	TReK Linux	Uplink	500	500	100.811	50 MB	3.96	36003/0	501/0

The Table 1 test assets included three ground Cadre nodes, four on-orbit ELCs, and two laptops with the TReK toolkit installed (one Windows based and the other RedHat™ LINUX based). Overall results demonstrated an effective throughput of approximately 15 Mbps downlink and 4 Mbps uplink. Minimal retransmits occurred.

The second part of the Nominal Payload DTN (Short Duration) test consisted of simultaneous node to node communication. Six simultaneous downlinks and six uplinks were performed.

Table 2 illustrates the results.

Table 2. Nominal Payload DTN (Short Duration) Simultaneous Node to Node



SOURCE NODE	DESTINATION NODE	DIRECTION	BUNDLES SENT	BUNDLES RECEIVED	TIME (sec)	TOTAL BYTES	THROUGHPUT (Mbps)
ELC-1	HOSC Cadre A	Downlink	100	100	17.188	10 M	4.694
ELC-2	HOSC Cadre B	Downlink	100	100	12.988	9900003 *	6.098
ELC-3	HOSC Cadre C	Downlink	100	100	14.57	9900003 *	5.436
ELC-4	HOSC Cadre A	Downlink	100	100	12.876	10 M	6.213
TReK Win7	HOSC Cadre B	Uplink	100	100	24.75	9900003 *	3.21
TReK Linux	HOSC Cadre C	Uplink	100	100	13.463	10 M	5.942
HOSC Cadre A	ELC-1	Uplink	100	100	54.35	10 M	1.472
HOSC Cadre B	ELC-2	Uplink	100	100	33.025	10 M	2.422
HOSC Cadre C	ELC-3	Uplink	100	100	36.27	10 M	2.206
HOSC Cadre A	ELC-4	Uplink	100	100	89.965	10 M	0.889
HOSC Cadre B	TReK Win7	Uplink	100	100	20.873	10 M	3.833
HOSC Cadre C	TReK Linux	Uplink	100	100	66.135	9900003 *	1.198

* A Total Byte value of 9900003 resulted in some instances due to an out of order bundle being received. The starter bundle is 3 bytes long. Since a data bundle was received before the starter bundle, the first bundle was interpreted as the starter, and the 3 byte starter bundle was interpreted as part of the data. DTN does not guarantee delivery in sequence. Due to a receiving application limitation (i.e., bpcounter), the bundles were misinterpreted. This behavior does not indicate an issue with DTN.

Results in Table 2 show that all bundles were received at the destination nodes. The effective throughput of each transfer was substantially reduced from the single node to node values due to the simultaneous transfers utilizing a limited resource. Throughput is determined from the time the first packet is received at the destination until the last packet is received. DTN transfers do not necessarily occur round robin. The transfer which is started first may send more bundles initially even after the other transfers have started. Other transfers will remain idle as one transfer appears to "take over" for a time. Because of this behavior, a transfer can start with a single frame and then be idle for an extended period of time. The throughput rate in this case would be particularly low since the idle time would be included in the calculation. Total retransmits were 262 and 243 for uplink and downlink, respectively.

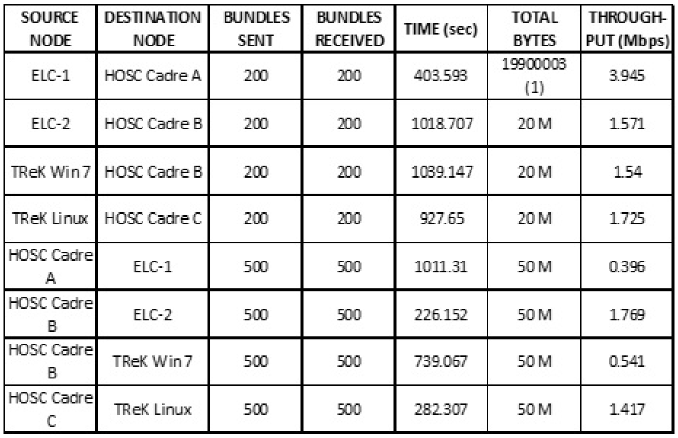
The third part of the Nominal Payload DTN (Short Duration) test consisted of simultaneous node to node communication with a decreased space to ground rate. The downlink rate was reduced from 20 Mbps to 10 Mbps. All bundles were transferred. Throughput rates ranged from 0.5 to 4 Mbps downlink and 0.75 to 2.6 Mbps uplink. As expected, downlink rates are near half the rate than the test with an LTP configured bandwidth of 20 Mbps, and uplink results are similar.

The fourth part of the Nominal Payload DTN (Short Duration) test consisted of a simultaneous node to node communication with an increased space to ground rate. The configured LTP downlink rate is 20 Mbps (increased from 10 Mbps in the third part of the test). All bundles were transferred. Throughput rates ranged from 3 to 7 Mbps downlink and 0.75 to 3.9 Mbps uplink. As expected, downlink rates were near double the rate than the test with an LTP configured bandwidth of 10 Mbps, and uplink results are similar.

D. Nominal DTN Loss of Communication Test

The second category of testing was the Nominal (Uplink/Downlink) DTN Loss of Communication tests. The test contained eight simultaneous transfers (4 downlink and 4 uplink). The Ku-Band outage duration was 10 minutes. Table 3 illustrates the results.

Table 3. Nominal Payload DTN Loss of Communication



SOURCE NODE	DESTINATION NODE	BUNDLES SENT	BUNDLES RECEIVED	TIME (sec)	TOTAL BYTES	THROUGH-PUT (Mbps)
ELC-1	HOSC Cadre A	200	200	403.593	19900003 (1)	3.945
ELC-2	HOSC Cadre B	200	200	1018.707	20 M	1.571
TReK Win 7	HOSC Cadre B	200	200	1039.147	20 M	1.54
TReK Linux	HOSC Cadre C	200	200	927.65	20 M	1.725
HOSC Cadre A	ELC-1	500	500	1011.31	50 M	0.396
HOSC Cadre B	ELC-2	500	500	226.152	50 M	1.769
HOSC Cadre B	TReK Win 7	500	500	739.067	50 M	0.541
HOSC Cadre C	TReK Linux	500	500	282.307	50 M	1.417

(1) A Total Byte value of 9900003 resulted due to an out of order bundle being received.

The Loss of Signal (LOS) was simulated by disabling the virtual channel associated with the payload data on the ground equipment. The large number of retransmits are expected since bundles are still sent during the LOS. After restoration of the communication link, buffered data was interleaved with the live data, and all bundles and bytes were transferred.

E. CFDP Test

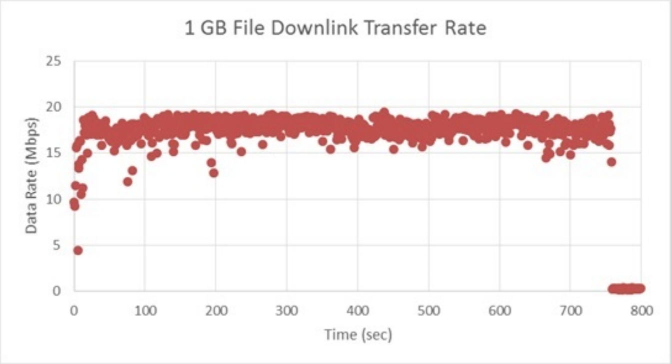
The third category of testing consisted of CFDP transfers. CFDP transfers were initiated by two different mechanisms: the ION provided cfdptest tool and the TReK toolkit provided CFDP GUI.

The first part of the CFDP tests was single transfers. File sizes of 1 MB, 50 MB, and 175 MB were sent between various ground and on-orbit nodes. Both downlinks and uplinks were performed. Table 4 illustrates the results. Figure 5 shows the space to ground bandwidth utilization during the 1 GB downlink transfer.

Table 4. CFDP Single Transfers

SRC	DEST	Data Segment (Total/# Rexmit)	Report Segment (Total/# Rexmit)	Canceled by Sender (Total/# Rexmit)	Canceled By Rcvr (Total/# Rexmit)	Filesize
Cadre C	TReK Win7	757/4	Oct-00	0/0	0/0	1 MB
TReK Win7	Cadre B	73/0	Sep-00	0/0	0/0	1 MB
Cadre C	TReK Win7	37582/0	405/0	0/0	0/0	50 MB
TReK Win7	Cadre B	3642/0	410/0	0/0	0/0	50 MB
Cadre C	TReK Win7	131128/2	1413/0	0/0	0/0	175 MB
TReK Win7	Cadre B	12707/0	1428/0	0/0	0/0	175 MB
Cadre C	TReK Linux	37595/13	407/0	0/0	0/0	50 MB
TReK Linux	Cadre A	3648/0	416/0	0/0	0/0	50 MB
Cadre C	TReK Linux	131128/2	1413/0	0/0	0/0	175MB
TReK Linux	Cadre A	12716/0	1437/0	0/0	0/0	175 MB
TReK Win7	Cadre B	69336/28	7785/6	0/0	0/0	1 GB

Figure 5. 1 GB File Downlink Transfer Rate



Files were successfully transferred as shown by verifying the checksum of the destination file against the source file.

The second part of the CFDP tests were nominal simultaneous file transfers with and without an LOS. Tables 5 and 6 illustrate the results without an LOS and with a 10 minute LOS, respectively.

Table 5. CFDP Simultaneous Transfers (No LOS)

SRC/DEST	Data Segment (Total/# Rexmit)	Report Segment (Total/# Rexmit)	Canceled by Sender	Canceled By Rcvr	Filesize
Cadre C-> TReK Linux Cadre B-> TReK Win7	79144/3397	1101/0	0/0	134/0	50 MB 50 MB
TReK Linux -> Cadre C TReK Win7 -> Cadre A	25407/30	2895/47	0/0	Jan-00	175 MB 175 MB

Table 6. CFDP Simultaneous Transfers (with 10 min LOS)

SRC/DEST	Data Segment (Total/# Rexmit)	Report Segment (Total/# Rexmit)	Canceled by Sender	Canceled By Rcvr	Filesize
Statistics During LOS					
Cadre C -> TReK Linux Cadre B -> TReK Win 7	12918/999	196/0	0/0	0/0	50 MB 50 MB
TReK Linux -> Cadre C TReK Win7 -> Cadre A	3880/4	453/10	0/0	0/0	175 MB 175 MB
Final Statistics					
Cadre C -> TReK Linux Cadre B -> TReK Win7	77093/1249	929/0	0/0	18/0	50 MB 50 MB
TReK Linux -> Cadre C TReK Win7 -> Cadre A	26193/495	3273/208	16/0	66/198	175 MB 175 MB

All files were transferred. Checksums were correct in all cases.

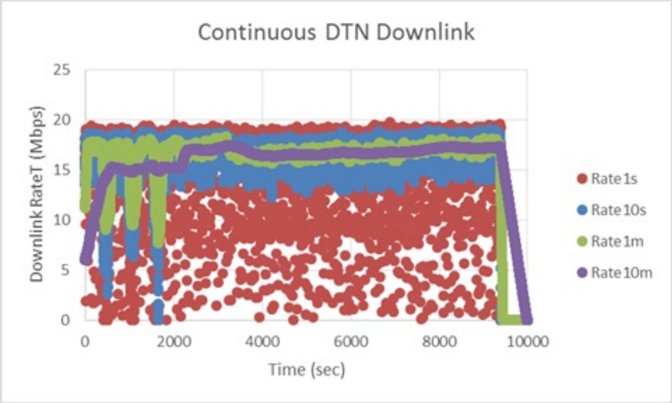
F. Additional Characterization Tests

Other characterization tests were performed in addition to those described for the DTN certification tests. These tests included rebooting the on board gateway while a CFDP transfer was in progress, long duration tests, and an all Linux based four node ION test.

The first additional characterization test of rebooting the on board gateway during a CFDP transfer resulted in successful file transfers. Bundles were queued on the ground gateway during the reboot period. Those bundles were successfully sent and received by the destination node after the reboot had completed. Conversely, the bundles that were queued on the on board gateway were successfully transmitted and received by the ground node after the completion of the reboot. The source nodes for both directions of transfer did not queue data since the gateways were able to queue the data during the reboot period. No data were lost.

The second additional characterization test was a longer duration test with a continuous data transfer. A 1 GB file was downlinked 20 times back to back. Figure 6 shows average data rates for space to ground bandwidth utilization over fixed periods of 1 second, 10 seconds, 1 minute, and 10 minutes. The 1 second rate illustrates the data rate can vary widely over a short duration; however, the data rates smooth as the period over which the average is calculated increases. The initial reduced average throughput (shown more pronounced in the 10 second and 1 minute rate intervals) is due to the fact that the source node is simultaneously reading the input files and transferring data to the on-orbit gateway. When the read of input files is completed at approximately the 2000 second mark, the average transfer rate begins to increase and stabilize around 18 Mbps for a 20 Mbps configured pipe. The time to complete the 20 file transfers is approximately 3 hours.

Figure 6. Continuous Downlink Transfer Rate



The third additional characterization test utilized a four node system using all Linux nodes and a simplified network. ISS flight equivalent laptops were used as the two nodes and two gateways. The on-orbit gateway matched the flight configuration. The purpose of the test was to determine the maximum performance that could be achieved with the ISS based laptops. Linux was chosen since the primary developing and testing of the ION product was with Linux. The flight bandwidth restrictions were also not enforced for this test. Approximately 50 Mbps uplink and downlink was achieved in this scenario.

4. Operational Support of the DTN System

Operating an Ethernet network orbiting the earth at 4.76 miles/s 249 miles above the earth is a feat unto itself. Managing the ISS Ethernet network requires creative methods to monitor, configure and operate the network communications on ISS, plus space-to-ground communications. The frequent Ku-band Satellite outages and long round trip delays add to the complexity of managing, monitoring and use of common application protocols to retrieve and monitor science experiments. DTN provides a service for payload developers that benefits not only the developer's operational overhead, but decreases the complexity of planning around Ku-band outages and bandwidth allocations for the flight operation teams.

A. Flight Operations

The ISS DTN gateway was originally deployed by the Flight Controllers at JSC and Marshall Space Flight Center (MSFC) in April 2016. The entire DTN architecture requires the cooperation of teams across multiple ground control centers and development/test facilities to ensure successful end-to-end operations. These teams include the Houston Mission Control Center Flight Operations Directorate and Huntsville Payload Integration Operations flight control teams as well as NASA Avionics and Software development/test teams. The on-orbit ISS Payload DTN gateway is primarily managed by the Payload Operations Integration flight control team out of the HOSC. This flight control team allocates not only Ku-band bandwidth, but also ensures the availability and smooth operation of the ISS Payload DTN gateway from the laptop hardware to the DTN ION application. They are responsible for any image or configuration updates. In addition, they are the front line for handling any on-orbit issues. These issues are worked in conjunction with HOSC, Houston MCC and other ground test teams to find quick answers and resolutions to avoid unnecessary downtime or gateway outages. Test labs are equipped with flight-equivalent hardware and software in preparation to simulate on-orbit operations when required.

Prior to first payload use of the system, the TReK demonstration laptop was used to perform an ISS-to-Ground gateway checkout using the bping tool and CFDP to exercise the system. This checkout was performed successfully in May 2016. Additional throughput tests will be performed prior to the first payload use of the system which is expected in spring 2017.

The ISS DTN Payload Gateway will be updated with addressing or service information for any new payload requesting access to DTN resources. In addition to the DTN addressing, the Station Support Computer (SSC) Servers also require updates to the Domain Name System (DNS) for new Payload information. These configuration changes are uplinked and deployed to the SSC file server and Payload DTN gateway by the Program Integration Office (PIO) and Flight Operations Directorate (FOD) flight control teams.

The ION and DTN2 protocol suites are in constant evolution to handle more services, protocols, larger networks, and higher bandwidths. This requires NASA to be vigilant when upgrading the primary DTN gateways to avoid impacting existing payloads already on ISS. NASA endeavors to avoid making changes that would incur additional costs and development by the developers. Hence all updates are assessed for impacts to payloads to ensure that any updates remain backwards compatible. The TreK tool also provides the flexibility to easily update payload end user nodes for addressing and convergence layers, as well as providing the latest stable platform for ION users.

New payloads are encouraged to test their systems at HOSC and/or SCTF. Both of these facilities include resources that mimic the on-board and end to end systems. If testing at these facilities is not feasible, the payload developer may also choose to use simulated resources at their development facility or tie into these test facilities from their remote facilities.

B. DTN Node Management

Initial development of the ISS DTN architecture identified the need for DTN Node management requirements. To handle management of the ISS Payload DTN Node, the DTN Node Management Daemon was developed. It is currently designed to run on ION DTN nodes, but can easily be extended to run on other DTN implementations. Providing a DTN management daemon avoided the reliance on applications such as SSH to send commands/status. It provides a single Application Programming Interface (API) for commands, command status, and statistics which is usable by both the client (user) and the server (DTN node). It is run as a service on the node similar to the ION bpecho tool. BP is used, not IP, for sending command, command status, and statistics information (messages). The software daemon is also responsible for making changes persistent across restarts or for rate controls between remote active nodes.

The DTN Node Management Daemon includes the ability to control output rates between the on-orbit DTN gateway node and the ground DTN gateway node. This capability supports the increase or decrease of output rates and ensures that the rate changes are persistent through a DTN gateway node restart. The rate changes were also made to be persistent not just when restarting, but also when switching from one remote active gateway to another. It also controls the flow of the file input/output (I/O) between the on-orbit DTN gateway node and the ground DTN gateway nodes; i.e., suspending, resuming, and cancelling flow of bundles related to designated file I/O "jobs" (CFDP).

Additional requirements met by the DTN Management Daemon supported route selection to newly designated active remote DTN gateway nodes via DNS reconfiguration and command, both of which can be user commanded. The DTN Management Daemon also made it possible to support enabling/disabling I/O with neighboring nodes, especially output control. Finally, the DTN Management daemon supports the gathering and reporting of basic level DTN node statistics. This feature provides the configurability to provide the statistics receiver identity, reporting interval, and enable/disable statistics reporting state.

The DTN daemon uses a C language programming interface, is based on operating as a DTN service, is BP-based, and uses a "flat" C structure for command, status and statistics messages. Figures 7 and 8, provide a high level overview of DTN Daemon Management Node Design and Operations.

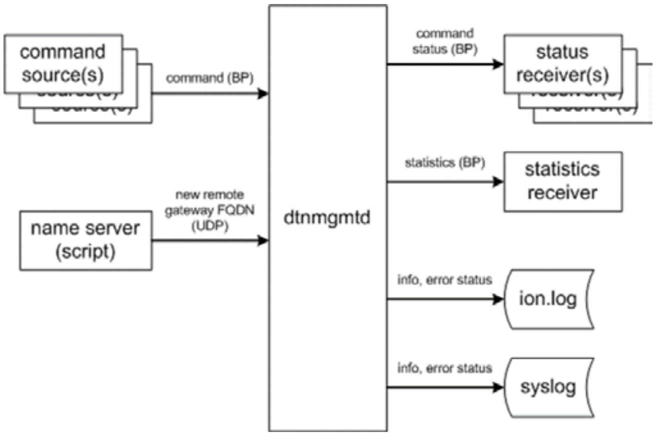


Figure 7. DTN Daemon Mgmt Node Design

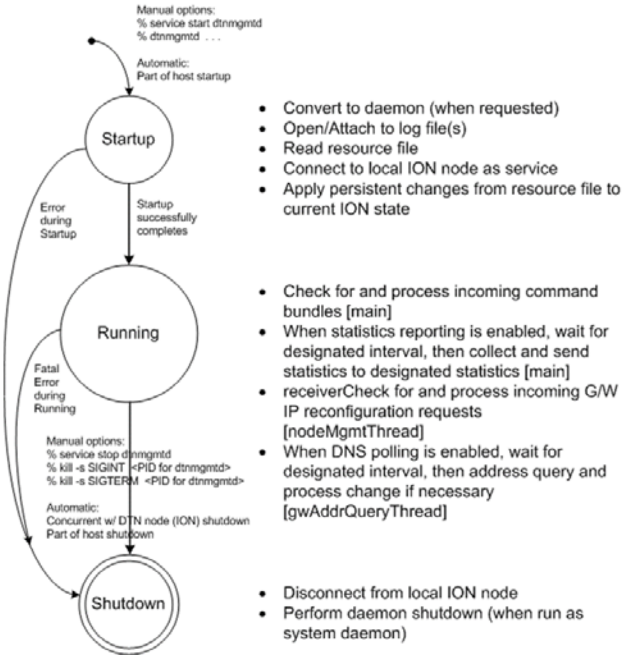

Manual options:
% service start dtnmgmtd
% dtnmgmtd . . .
Automatic:
Part of host startup
Startup
Error during Startup
Startup successfully completes
Running
Fatal Error during Running
Manual options:
% service stop dtnmgmtd
% kill -s SIGINT <PID for dtnmgmtd>
% kill -s SIGTERM <PID for dtnmgmtd>
Automatic:
Concurrent w/ DTN node (ION) shutdown
Part of host shutdown
Shutdown
Convert to daemon (when requested)
Open/Attach to log file(s)
Read resource file
Connect to local ION node as service
Apply persistent changes from resource file to current ION state
Check for and process incoming command bundles [main]
When statistics reporting is enabled, wait for designated interval, then collect and send statistics to designated statistics [main]
receiverCheck for and process incoming G/W IP reconfiguration requests [nodeMgmtThread]
When DNS polling is enabled, wait for designated interval, then address query and process change if necessary [gwAddrQueryThread]
Disconnect from local ION node
Perform daemon shutdown (when run as system daemon)

Figure 8. DTN Daemon Service Operations

5. Lessons Learned

Although the deployment of DTN on board the ISS seemed to be relatively straightforward, it took many iterations to finally achieve a system ready for deployment. A number of integration issues were discovered, both DTN and non-DTN related. Utilizing multiple OS platforms also complicated the effort as most of the ION development and testing was performed on Linux-based systems. However, the payload developer and node platforms used on the ISS are primarily Windows-based. In addition, issues were found between the integration of ISS ION and ground DTN2 gateway platforms.

A. TCP vs STCP

Initially, STCP was developed as a simplified convergence layer over TCP. Better performance was achieved using ION's STCP Convergence Layer (CL) vs. ION's TCP CL, particularly in the presence of disconnections common for the Station Support Computers (SSCs) on ISS (due to power cycles, wired/wireless transitions, etc.). ION STCP has since been modified to implement additional flow control mechanisms (blocking) to support the ISS file transfer software. This prevents the gateway nodes from overfilling during communication outages. The sender/receiver complexity is simpler using STCP CL. TCP CL increases the sender/receiver complexity since both must send/receive simultaneously.

B. ION Enhancements

Modifications were made to ION in order to support the ISS DTN Gateway and DTN Windows Nodes. ION-based applications can now be developed in Visual Studio. Support was also added for Metadata Extension Block per Request for Comments (RFC) 6258 which allows file transfer jobs to be linked to specific bundles. In addition, ION's LTP implementation was updated to compute limits on "report" segment generation dynamically, based on block size, segment size, and anticipated error rate. This enables LTP to try harder to complete block transmission when data loss is high. Error handling (reconnection attempts) was added for transient communication errors when using STCP which resulted in attempts to reconnect instead of closing the socket. Zero-copy object space allocation has been enhanced to enable bidirectional flow control on convergence-layer protocols that back-propagate blocking on buffer space requests. Improvements were also made to the TCPCLI bundle receiving functionality. All of the ION modifications have been incorporated into the open-source distribution.

C. DTN2 Enhancements

Updates for DTN2, as a result of ISS development include the following: Convergence Layer Adapters (LTP, UDP, and STCP) were added and updated. LTP and UDP CLAs were modified to give priority to sending non-data segments, bundle extraction was moved to a separate thread, and statistics were added. DTN2 features were extended to support inclusion into a modern NASA ground system. Features were included to support space missions and interoperate with ION, e.g. CBHE, ACS, and Extended Class of Service (ECOS). The code base was hardened and an external router was added for continuous operations. Deletion of bundles that were not delivered due to duplicate suppression was incorporated.

6. Forward Plan

DTN will continue to evolve for ISS as more payloads continue to choose to use DTN and demand higher availability and more bandwidth along with new services. Currently, there are 12 payloads in work that have indicated a desire to use the ISS DTN system. The existing ISS DTN platform is constrained by old laptop technology. The ISS DTN gateway is planned to be upgraded to a HP Zbook by the end of 2017. The improved laptop platform will have 32 GB of RAM and 2 TB storage. Not only does the laptop offer improved processing and greater storage capabilities, but it will also support Gigabit Ethernet. Characterization testing of the new platform is planned for October/November of 2016. Our challenge is to improve DTN processing capability to support higher gigabit rates. The challenge in this area will be ensuring sufficient storage/buffering capability during long communication outages, as well as maintaining backwards compatibility.

The initial deployment of ISS DTN supports DTN-aware nodes only, requiring each payload to be a DTN endpoint and support transmission of bundle protocol over TCP, STCP, or UDP convergence layers. Future plans for ISS DTN include the capability to support any payload whether DTN-aware or not. Our first use case for this capability will be the ECOSTRESS external wireless payload whose platform does not support the incorporation of DTN application processing. Therefore, the ISS DTN gateway is being modified with an additional virtual machine to receive standard TCP packets, bundle these packets and forward to the primary ISS DTN on-orbit gateway for downlink to the ground control center DTN gateway. The ground DTN nodes will in turn forward the bundled packets to the DTN endpoint at the remote developer facility. This capability can easily be expanded to accept UDP packets as well.

In addition, ION is in the midst of being updated to stabilize use of the TCP CL on ISS. ION 3.5.0 which incorporates this improvement was released in September 2016 and is under test.

As the ISS end-to-end DTN architecture expands for new payloads and ground nodes, as well as inclusion of ISS International Partners, the use of additional services will be required. Plans are in discussion to include ACS, DTPC, and Bundle Protocol Security (BPSec).

With the focus on ISS network security, potential DTN security services may also be implemented, such as the BPSec and associated key management exchange/distribution capabilities. These security services would need to be integrated with the HOSC DTN2 Gateway, and would require key management to be implemented at HOSC and Houston MCC.

Additional work is also being planned for the DTN Node Management Daemon which includes a focus on determining if better handling of Ku-Band LOS conditions are needed. The development of a method to support addition and removal of

DTN-enabled payloads is another desired capability.

7. SUMMARY

As of June 2016, ISS has established a permanent DTN infrastructure on ISS for payload operations. Due to the increased data rates and reliability over standard TCP-based file transfer protocols, payload usage is exceeding initial expectations. Challenges remain to expand the ISS DTN architecture to accommodate more payloads, higher data rate increases and expanded storage requirements. Efforts are in progress to identify new DTN services to support and protect payload science. NASA will continue to push the limits of DTN on ISS maturing DTN protocols for a Solar System Internetwork.

ACKNOWLEDGEMENTS

DTN resulted from over a decade of work led by NASA Advanced Exploration Systems, International Space Station Program and NASA Space Communications and Navigation, with work performed by Jet Propulsion Laboratory, NASA Johnson Space Center, NASA Marshall Space Flight Center, NASA Goddard Space Flight Center, NASA Glenn Research Center, Johns Hopkins University Applied Physics Laboratory, University of Colorado Boulder, and University of California, Berkeley. International involvement included: The Japanese Aerospace Exploration Agency, The European Space Agency, The German Space Agency, Democritus University of Thrace, and Università di Bologna among others.

REFERENCES

[1] Space Internetworking Strategy Group Operations Concept for Solar System Internetwork (SSI) – Final Version, Interagency Operations Advisory Group, October, 2010.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, "Delay-Tolerant Networking Architecture", RFC4838, April 2007.

[3] CCSDS Bundle Protocol Specification. Recommendation for Space Data System Standards, CCSDS 734.2-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2015.

[4] Licklider Transmission Protocol (LTP) for CCSDS. Recommendation for Space Data System Standards, CCSDS 734.1-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 2015.

[5] S. Burleigh, "Compressed Bundle Header Encoding (CBHE)", RFC6260, May 2011.

[6] CCSDS File Delivery Protocol (CFDP). Recommendation for Space Data System Standards, CCSDS 727.0-B-4. Blue Book. Issue 4. Washington, D.C.: CCSDS, January 2007.

BIOGRAPHY

Suzanne R. Davidson *is the Lead Engineer for the Joint Station LAN (JSL) of the International Space Station (ISS) under contract to the National Aeronautics and Space Administration (NASA) Lyndon B. Johnson Space Center. As an Associate Technical Fellow for the Boeing Company, her responsibilities include development of the ISS Ethernet Architecture, infrastructure design, hardware certification, hardware/software configuration, system level testing and sustaining engineering.*

Brett M. Willman *is the Deputy Manager of Computer Resources for the International Space Station (ISS) at the National Aeronautics and Space Administration (NASA) Lyndon B. Johnson Space Center. In this role, he is responsible for the lifecycle of computer systems on ISS, including infrastructure certification, configuration, maintenance, and decommission. Prior to his current position, Mr. Willman served as a flight controller at NASA's Mission Control Center for computer systems on ISS. He was technical lead for Expedition 13 and STS-119. Mr. Willman holds a Bachelor of Science in Computer Engineering and a Master of Computer Science with an emphasis in Network Engineering and Information Security from Texas A&M University.*

William A. Pohlchuck *is a software engineer for the Joint Station LAN (JSL) team. He is responsible for development of the on-orbit DTN gateways and integration of the DTN system on ISS. Mr. Pohlchuck holds a Bachelor of Science in Electrical Engineering from Ohio Northern University and a Master of Science in Electrical Engineering from the University of Utah.*

Adam Schlesinger *is the technical lead for the Advanced Exploration Systems (AES) Delay/Disruption Tolerant Networking (DTN) project and the Deputy Manager for the AES Avionics and Software project at NASA Johnson Space Center (JSC). Mr. Schlesinger has led efforts to infuse DTN technology on the International Space Station and with other AES projects. Mr. Schlesinger also supports the Human Exploration and Operations Mission Directorate (HEOMD), in the areas of avionics, communications, networks and security, to develop architectures and technologies for future space exploration. Mr. Schlesinger holds a Bachelor's degree in Electrical Engineering from the University of Michigan and a Master's degree in Electrical and Computer Engineering from the Georgia Institute of Technology.*

R. Lee Pitts *is the Lead System Engineer for the Huntsville Operations Support Center (HOSC). He is employed by COLSA 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 Launch System, Space Station, and SmallSat programs.*