1569303096

Large File Transfers from Space using Multiple Ground Terminals and Delay-Tolerant Networking

William D. Ivancic, Phillip Paulsen, Dave Stewart, Wesley Eddy, James McKim
NASA Glenn Research Center
John Taylor, Scott Lynch, Jay Heberle
Universal Space Network
James Northam, Chris Jackson
Surrey Satellite Technology Limited
Lloyd Wood
Centre for Communication Systems Research, University of Surrey

***Abstract*—We use Delay-Tolerant Networking (DTN) to break control loops between space-ground communication links and ground-ground communication links to increase overall file delivery efficiency, as well as to enable large files to be proactively fragmented and received across multiple ground stations. DTN proactive fragmentation and reactive fragmentation were demonstrated from the UK-DMC satellite using two independent ground stations. The files were reassembled at a bundle agent, located at Glenn Research Center in Cleveland Ohio. The first space-based demonstration of this occurred on September 30 and October 1, 2009. This paper details those experiments.**

***Index Terms*—Communication, delay-tolerant networking, DTN, satellite, Internet, protocols, bundle, IP, TCP.**

I. Introduction

NASA's Earth Science Technology Office (ESTO) is interested in automating terrestrial and space-based sensor webs, as well as in developing technologies which allow sensor webs to interact autonomously and improve access to sensor data. NASA Glenn Research Center has performed research related to secure, autonomous, integrated space/ground sensor webs.

The overall goal of the secure autonomous integrated space/ground sensor web project was to demonstrate secure coordinated network-centric operations of space/ground assets owned and operated by multiple parties. In order to accomplish this, a network consisting of terrestrial sensors (seismic sensors), a Virtual Mission Operations Center (VMOC), multiple ground stations and a spacecraft were used. The concept of operation is illustrated in Figure 1. A seismic sensor update is received by the VMOC that indicates the location of some exceptional event of interest. The VMOC then decides what other sensors or sensor networks can be brought to bear in order to gain more information on that event. In this situation, the terrestrial sensor web is the Global Seismic Network, with trigger information obtained from the United States Geological Survey (USGS).

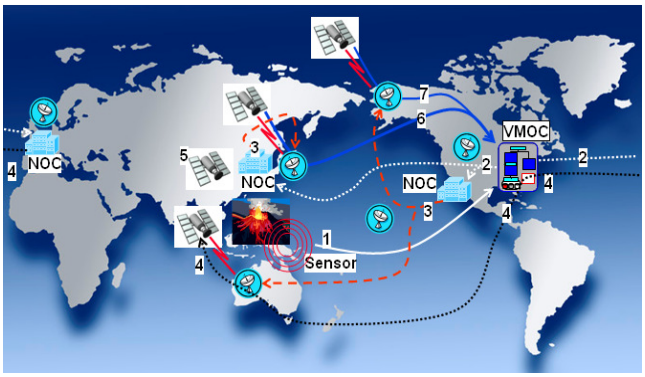

Figure 1 – Concept of Operations for a Secure, Autonomous, Integrated Space/Ground Sensor Web

Figure 1 shows the overall concept of operations (CONOPS) as a series of events, labeled 1 through 7.

1) The VMOC receives a trigger of a seismic event from the USGS notification system, setting the autonomous sensor web into motion.
2) The VMOC's job is to task other sensors and reserve and/or configure whatever infrastructure is necessary to obtain the requested data. In this instance, the other sensor is the United Kingdom Disaster Monitoring Constellation (UK-DMC) satellite, built and operated by Surrey Satellite Technology Limited (SSTL). The supporting infrastructure includes Internet-enabled ground stations.
3) Once the VMOC coordinates all facilities for availability and requests reservation of those assets, the individual entities cooperate to configure and operate their facilities and infrastructure to provide functionality and capabilities, rather than the VMOC assuming complete control.
4) Commands are sent to the spacecraft regarding when to capture sensor data and when to transmit that data to the ground. Those commands are currently passed to the satellite via SSTL's 'home' ground station; however, they could conceivably be passed to the satellite via a third-party ground station, such as one of the other international DMC ground stations, or the Australian station of Universal Space Network (USN), as shown in Figure 1.
5) The remote-sensing image is taken over the area of interest corresponding to the seismic event.
6) The image is then downloaded. Figure 1 shows this image being downloaded to a third party ground station in Japan.
7) If an image is too large to be transmitted in its entirety during a single pass, the remaining portion of the file can be transmitted via a second ground terminal. In Figure 1, the second ground station is at USN's site in Alaska.

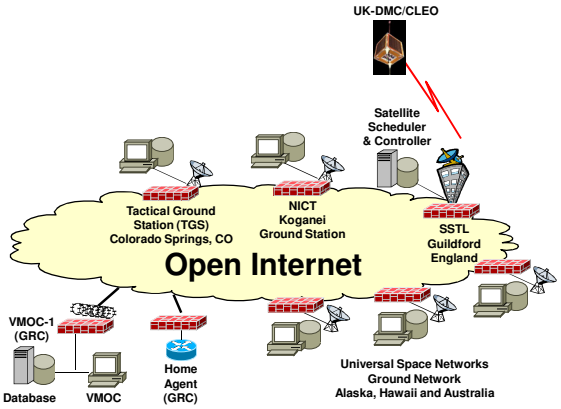

Figure 2 – Space/Ground Network

A limited demonstration of the overall concept occurred in July 2009 [1] [2]. This paper describes a later demonstration of large-file transfer over multiple ground stations, corresponding to steps 6 and 7, in September and October of 2009. These tests were successfully performed using the network shown in Figure 2.

II. Store-and-Forward Protocol

In order to perform large file transfers, using multiple independent ground stations, it was necessary to utilize store-and-forward technologies to break control loops across the end-to-end path into separate consecutive space-ground and ground-ground control loops. This increased download efficiency across each link.

These tests were performed using the UK-DMC satellite. The UK-DMC communication system uses a slow 9600 bits-per-second (bps) uplink for commanding and the much faster 8.134 Mbps dedicated downlink for transmitting high-resolution imagery. Any ground-to-ground communication is over the open, shared, congested Internet, and could have effective throughput of 10s of kbps to 10s of Mbps with no guarantees. It is imperative that the downlink be fully utilized so that as much data is transferred from the satellite as possible during a minutes-long pass over a ground station.

By implementing store-and-forward techniques, we can break the communication control loops from space-to-ground and ground-to-ground and choose optimal transport protocols for each link to increase delivery throughput. For these experiments, the experimental Delay-Tolerant Networking (DTN) "bundle protocol" was used as the store-and-forward protocol [3] [4]. The *Saratoga* transport protocol [5] [6] [7] was used to optimize data transfer of bundles from space to ground across the private space link, while the widespread Transmission Control Protocol (TCP) was used as the "bundle convergence layer" for ground-to-ground data transfer across the public, shared, Internet.

III. Large File Transfers via a Single Ground Terminal

On August 27 and 28 of 2008, large file transfer experiments were performed using DTN proactive fragmentation and two satellite passes over a single ground station at SSTL in Guildford, United Kingdom [4]. The two passes emulated use of multiple independent ground terminals. Following those tests, work progressed to establish infrastructure required to perform the same tests over multiple ground stations [Figure 2]. In July of 2009, that infrastructure was completed and tested. There now were sufficient ground stations and corresponding DTN bundle agent nodes available to be able to perform multi-terminal testing. The DTN bundling-capable ground stations were at Guildford, England, as before, and at stations in Alaska, Hawaii and Australia (operated by Universal Space Network). Work also progressed to include a ground station in Koganei, Japan, operated by the National Institute of Communication Technology (NICT) of Japan. The US Army's Tactical Ground Station, in Colorado Springs, was removed from this network, due to changes in their operational priorities.

IV. Large File Transfers via Multiple Independent Ground Terminals

On September 30 and October 1, 2009, successful demonstrations of DTN proactive fragmentation and, by accident, reactive fragmentation using two independent ground stations, occurred. Proactive fragmented bundles were received at ground stations in Alaska and Hawaii and reassembled at a bundle agent at NASA Glenn Research Center in Cleveland, Ohio [Figure 3].

A. *Test Details*

The UK-DMC satellite passed over USN ground stations in Alaska and Hawaii. The first pass was over Alaska and the second pass over Hawaii, about 73 minutes later.

The onboard Solid State Data Recorder (SSDR) computer, storing the remote sensing image taken by the onboard cameras in its RAM, had to remain powered on during eclipse (dark night), but all transmissions were made during daylight to put as little strain on the satellite power system and onboard battery as possible.

Images of 150 Mbytes in size were captured and proactively fragmented into 80-Mbyte and 70-Mbyte bundles.

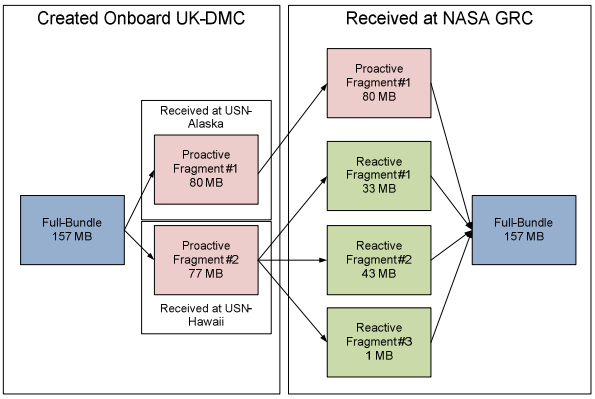

Figure 3 – Large File Transfers using Proactive and Reactive DTN

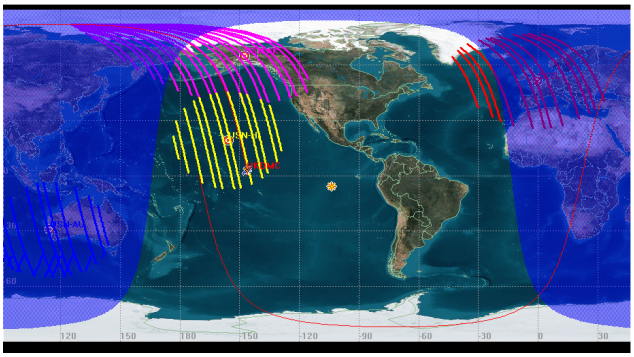

Figure 4 – Plot of Satellite Passes over Ground Stations

Creating a large 150-Mbyte file enabled demonstration of proactive fragmentation, but also provided sufficient time during a pass to also download the entire file using non-DTN techniques for comparison. It also allowed some time for recovery from human operating error.

Note, SSTL image files are often up to nearly a Gbyte on the UK-DMC satellite, and are much larger on SSTL's newer satellites with higher-resolution imagers [4].

B. Choosing a Satellite Pass

For these experiments, modeling in the Satellite Took Kit (STK) was used to select potential passes for the UK-DMC satellite over Alaska and Hawaii [Figure 4]. In order to conserve the satellite batteries, all passes are in daylight. Conveniently, a high elevation pass over Alaska will result in a high elevation pass over Hawaii approximately 1 orbit later. The colored lines in figure 4 show the line-of-sight contact time over the ground stations of interest when in view of the satellite and when in daylight.

Once a potential pass is identified, the USN ground station schedule is evaluated for availability. If USN can meet the pass times, a request is put in to reserve the ground station at the appropriate UTC times [Table 1]. If that request is accepted, a request for services is sent to SSTL. SSTL then evaluates the request against its commitments, and either validates the requests, or indicates that such a request cannot be met due to previous commitments or insufficient satellite resources (*e.g.* low battery levels, unavailable on board storage due to other imaging commitments). If SSTL cannot meet the request, the process is then started from the beginning, with a request for available ground station time preceding a request for available satellite time. This is done in that order as the confirmation for ground station time is currently much quicker, being performed via email, with updated operations schedules also automatically being sent via email.

Table 1 – USN Request for Service

UKDMC,USAK01,Add,9/30/2009,273,17:11:00,9/30/2009,273,17:24:00
,0:13:00,High Rate Pass access SSDR
UKDMC,USHI01,Add,9/30/2009,273,18:37:30,9/30/2009,273,18:50:30,
0:13:00,High Rate Pass access SSDR
UKDMC,USAK01,Add,10/1/2009,274,17:49:00,10/1/2009,274,18:02:00
,0:13:00,High Rate Pass access SSDR
UKDMC,USHI01,Add,10/1/2009,274,19:14:00,10/1/2009,274,19:27:00,
0:13:00,High Rate Pass access SSDR

USN asset reservations can also be quickly cancelled via email. Current scheduling of SSTL's assets is performed manually. Thus, there is a mix of manual processes and automation for scheduling assets. Eventually, this scheduling will be fully automated [1].

On both September 30th and October 1st, 2009, NASA was able to successfully schedule all needed ground and space assets for testing and request image capture. Since the actual image content was not of importance for these large-file tests, just the size, the image was taken just prior to the 1st pass.

C. Multi-Terminal Testing – September 30, 2009

On September 30, 2009, the following commands were issued to the UK-DMC satellite:

- 150 Mbyte image capture at 17:00:21 UTC
- MD5 file checksum hash command at 17:02:02 UTC
- Downlink 1 - 17:11:00 to 17:24:00 (Full downlink duration scheduled, eclipse starts at 17:30 UTC)
- Downlink 2 - 18:37:30 to 18:50:30

NASA had previously added functionality to SSTL's operational code, including creation of bundles for the first Interplanetary Internet tests in space [4], and a command to check file integrity. SSTL does not use checks of entire file integrity in their normal operations, as these take some time to run, and processing of image data (orthorectification, calibration) will expose errors. Instead, a strong HDLC frame CRC across each IP packet, coupled with SSTL's *Saratoga* transport protocol to resend data, provides sufficient reliability for SSTL's needs. NASA added an optional MD5 hash command to enable checking of the reconstructed downloaded file, generated from bundle fragments, against the original onboard file.

The test plan and procedures for 30 Sept 2009 were performed in the following order:

Order of tests for Pass 1 over Alaska:

1) Download the System Log File, Syslog, and check the MD5 checksum
2) Download Proactive Fragment #1 (DTN proactive Fragmentation)
3) Download File using GRC *Saratoga*
4) Download Syslog again
5) Check that bundle fragment #1 is transferred to Bundle Master (the NASA Glenn Bundle Agent that is the destination and reassembly point)
6) Check that fragment #1 was received by Bundle Master.

As there are approximately 70 minutes between the Alaska pass and the Hawaii pass, there is sufficient time to validate reception of bundle fragment #1.

Order of tests for Pass 2 over Hawaii:

1) Download Syslog
2) Download Fragment #2 (DTN proactive Fragmentation)
3) Download File using GRC *Saratoga*
4) Download Syslog again

5) Check that bundle fragment #2 was transferred to Bundle Master and confirm that fragment #2 was received by Bundle Master.
6) Check the MD5 calculation of the recombined file to confirm accurate successful reception.

During the pass over Alaska, the results showed a successful download of the Syslog, of the 80 Mbyte fragment (fragment #1) and the entire non-bundle 150 Mbyte file. There was also a successful download of the Syslog file, the 70 Mbyte fragment (fragment #2) and the entire 150 Mbyte non-bundle file over Hawaii.

Oddly, no bundles from Alaska or Hawaii were received at the NASA Glenn bundle agent destination, designated the Bundle Master. During the earlier single terminal test of August 2008 this problem had not been observed [4]. Upon investigation, a DTN routing problem was discovered. In the 2008 single terminal tests, there was a default route in the DTN2 configuration file at the ground station in Guildford, England (i.e. *dtn://**). That default route had been removed from both the Alaska and Hawaii bundle agents. Thus, the bundles were sitting at Hawaii and Alaska awaiting a route. Due to a typographic error, a proper route was not available.

The route in the bundle Alaska and Hawaii bundle agents was:

link add link_grc1 bundling1:4556 ONDEMAND tcp
route add dtn://bundling-grc1/ link_grc1*

However, the Endpoint Identifier (EID) of the Bundle created onboard the UK-DMC satellite was *dtn:bundling-grc1*. Note, there are no forward slashes in the onboard EID, but there were forward slashes in the ground station bundle forwarding nodes (*i.e.* dtn://bundling-grc1). To clarify: *dtn://bundling-grc1/** was the initial route configuration on 30 Sept 2009 and this didn't work as expected. The bundles that were not being forwarded were addressed to the different *dtn:bundling-grc1*. In 2008, with the default route in the SSTL ground station bundle forwarding agent, the bundle was forwarded properly. With no default routes, the bundle fragments at Alaska and Hawaii were simply being stored until a valid route would become available or the bundle would expire.

It is important to note that the bundles were created with a lifetime of 3 days. Thus, the received bundle fragments were not in danger of expiring during these experiments.

Examination of the DTN2 status showed:

In Alaska:

Currently Pending Bundles (1):

11503: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 80000000

In Hawaii:

Currently Pending Bundles (1):

26558: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 77260545

The following route was added in both the Alaska and Hawaii bundle agents:

route add dtn:bundling-grc1/ link_grc1*

Technically, this should not have matched either route, or, if the slashes do not matter, then it should have matched both. The fact that the route matched one, but not the other, is logically inconsistent. Thus, there appears to be a parsing bug in the implementation of the DTN2 bundle software used (version 2.3.0).

The route in Alaska was added first, and forwarding of proactive bundle fragment #1 began. We receive bundle fragment #1 from Alaska at GRC "bundle master":

bundling-grc dtn% bundle list
Currently Pending Bundles ... :
154400: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 80000000

Examining the information at the NASA DTN2 destination bundle agent from source *uk-dmc* showed that a full proactive bundle fragment was received.

dtn% bundle info 154402
bundle id 154402:
source: dtn:uk-dmc/i
dest: dtn:bundling-grc1/i
custodian: dtn:none
replyto: dtn:none
prevhop:
payload_length: 33128366
priority: 0
custody_requested: false
local_custody: false
singleton_dest: false
receive_rcpt: false
custody_rcpt: false
forward_rcpt: false
delivery_rcpt: false
deletion_rcpt: false
app_acked_rcpt: false
creation_ts: 307602048.0
expiration: 604800
is_fragment: true
is_admin: false
do_not_fragment: true
orig_length: 157260545
frag_offset: 80000000
transmission_count: 0

Examining the bundles at the Hawaii Bundle agent showed a proactive bundle fragment of 77260545 bytes pending, due to no current route being available.

localhost dtn% route dump bundle list
Currently Pending Bundles (1):
26558: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 77260545

A route was added to the Hawaii bundling agent:

localhost dtn% route add dtn:bundling-grc1/ link_grc1*

and forwarding is established. During forwarding, an interesting turn of events occurred. While monitoring incoming packets at the NASA Glenn bundle master (*bundling-grc1*), it became evident that the TCP connection

slowed, stopped and restarted. This happened three times, resulting in three *REACTIVE* fragments of our 77 Mbyte proactive fragment. This reactive fragmentation was documented in the Hawaii log file captured during our "putty" sessions [8]. The following shows for the second reactive fragment:

localhost dtn% bundle list
Currently Pending Bundles (1):
26558: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 77260545
localhost dtn% [1254339449.785320 /dtn/bundle/daemon warning] event BUNDLE_RECEIVED took 2338 ms to process
localhost dtn% bundle list
Currently Pending Bundles (1):
26559: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 44136275
localhost dtn% bundle info
wrong number of arguments to 'bundle info' expected 3, got 2 while evaluating {bundle info}
localhost dtn% bundle info list
Currently Pending Bundles (1):
26559: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 44136275
localhost dtn% bundle list info 26559
bundle id 26559:
source: dtn:uk-dmc/i
dest: dtn:bundling-grc1/i
custodian: dtn:none
replyto: dtn:none
prevhop:
payload_length: 44136275
priority: 0
custody_requested: false
.... all false ....
do_not_fragment: false
orig_length: 157260545
frag_offset: 113124270
transmission_count: 0

The following bundle fragments were received at the bundle master destination (*bundling-grc1*) and awaited processing:

Currently Pending Bundles ... :
154400: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 80000000
154402: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 33128366
154403: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 42758078
154404: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 1382309

At the bundle recombining destination, there was an 80-Mbyte proactive bundle fragment #1 from Alaska and a 70-Mbyte proactive fragment #2 from Hawaii, but the 70-Mbyte proactive fragment #2 was received in three reactive fragments of 33.1 Mbytes, 42.8 Mbytes and 1.3 Mbytes, as shown in Figure 3.

The bundle fragments were recombined using a NASA file recombining script, *dtnrecv*.

dtnrecv -n 1 -O $1 dtn:bundling-grc1/i

An MD5 hash was then calculated for the recombined received file, and validated against the original MD5 calculation performed onboard the UK-DMC satellite to ensure that the fragments were reassembled and the bundle was received correctly. The results show a perfect match.

From the Spacecraft Syslog file:

<14>syslog[17:02:02+052 30/09/2009]: looking for files in /home to run MD5 on
<14>syslog[17:02:02+054 30/09/2009]: syslog.txt
<14>syslog[17:02:02+055 30/09/2009]: tmp
<14>syslog[17:02:02+056 30/09/2009]: DU000999pm
<14>syslog[17:02:02+058 30/09/2009]: running MD5 on /home/DU000999pm
<14>syslog[17:07:27+028 30/09/2009]: MD5 of DU000999pm : 0x6964a515 , 0xf7672d18, 0x7a89ee21, 0xce7aeab7

At GRC BundleMaster:

[weddy@Bundle-Master Sep302009_multiterminal_AK-HI_pass]$ md5sum DU000999pm_bak
6964a515f7672d187a89ee21ce7aeab7 DU000999pm_bak

D. Multi-Terminal Testing – October 1, 2009

On Thursday, October 1, 2009, a nearly-identical test to that of September 30th was executed. However, since the September 30th test was fully successful, a slight modification was performed. For the Alaska pass, fragment #2, the 70-Mbyte proactive fragment, was downloaded. During the second pass over Hawaii, the first, 80-Mbyte, bundle fragment was downloaded. Since the proper routes were now in place, these bundle fragments were forwarded as soon as they were received at the ground stations, without being delayed due to not having a known route to destination.

On October 1, 2009, the following commands were issued to the UK-DMC satellite:

- 150 Mbyte Image capture at 17:30:21 UTC
- MD5 Command at 17:32:02 UTC
- Downlink 1 - 17:49:00 to 18:02:00 (Full downlink duration scheduled, eclipse starts at 18:08 UTC)
- Downlink 2 - 19:14:00 to 19:27:00

Order of tests for Pass 1 over Alaska:

1) Download the System Log File, Syslog, and check the MD5 checksum
2) Download proactive fragment #2 (DTN proactive fragmentation)
3) Download file using NASA Glenn *Saratoga*
4) Download Syslog again

Order of tests for Pass 2 over Hawaii:

1) Download Syslog
2) Download fragment #1 (DTN proactive Fragmentation)
3) Download file using NASA Glenn *Saratoga*
4) Download Syslog again

As with the 30 September tests, successful download occurred at both ground stations. The Syslog file was downloaded multiple times at each ground station, and the proactive fragments were downloaded once, as was the entire file without bundling for a comparison check.

The TCP connection between Hawaii and GRC again timed out. This time-out only occurred once, resulting in two reactive fragments. The proactive and reactive fragments were reassembled, and the file MD5 calculation matched that onboard the spacecraft.

From the perspective of the bundle destination, NASA Glenn BundleMaster (*bundling-grc1*), the following bundles fragments were received and awaited processing:

bundling-grc dtn% bundle list
Currently Pending Bundles (3):
156002: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 77260545
156003: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 12320689
156004: dtn:uk-dmc/i -> dtn:bundling-grc1/i length 67683407

From the Spacecraft Syslog File:

<14>syslog[17:32:02+075 01/10/2009]: DU000998pm
<14>syslog[17:32:02+076 01/10/2009]: running MD5 on /home/DU000998pm
<14>syslog[17:37:27+054 01/10/2009]: MD5 of DU000998pm : 0x55e1a3d4 , 0xb0906b00, 0xd95084bf, 0x1353ec50

At GRC BundleMaster:

[weddy@Bundle-Master Oct012009_multiterminal_AK-HI_pass]$ md5sum Oct012009_multiterminal_ak-hi_img
55e1a3d4b0906b00d95084bf1353ec50

E. Reactive Fragmentation

We did not determine why the TCP connection from USN Hawaii to NASA Glenn in Cleveland was timing out. However, these time-outs led to a demonstration of reactive fragmentation – albeit unintentionally and by accident. Without support for reactive fragmentation and reassembly, these tests would not have been successful. Being able to handle reactive fragmentation is, at least in this case, highly desirable.

If the bundle security protocol (BSP) bundle authentication block (BAB) [9] or the payload integrity block (PIB) [10] had been used, these separate fragments would have been discarded at the receiving bundle master due to failure of the authentication or integrity checks. It is worth re-examining how authentication and reliability are performed – particularly with regard to implementation policy. It may be possible to implement either in a manner whereby one can reconstruct the fragments hop-by-hop so long as the fragments follow the same path.

V. Conclusion

DTN bundle protocols were used to break control loops between space-ground communication links and ground-ground communication links to increase efficiency of file delivery, as well as to enable large files to be proactively fragmented and received at two independent ground stations. Without reactive fragmentation, these tests would not have been successful. Reactive fragmentation is unable to operate successfully using current implementations of bundle authentication and reliability. Application of authentication and implementation of authentication and reliability design and policy should be reconsidered to enable use with reactive fragmentation.

The DTN2 implementation holds onto bundles until valid routes or a default route are available, or until the bundle expires. If bundles were removed due to no available route, even though the lifetime had not expired, the tests would have failed. It is highly recommended that bundles only be removed once they expire, as valid routes may become available during the lifetime of the bundle, even if those routes do not initially exist.

During testing, the extensive logging and reporting capabilities of the DTN2 bundling implementation proved invaluable. Such logging and reporting capabilities should be encouraged for other DTN implementations.

References

[1] W. Ivancic, P. Paulsen, D. Stewart, J. Walke, L. Dikeman, S. Sage, E. Miller, J. Northam, C. Jackson, L. Wood, J. Taylor, S. Lynch and J. Heberle. “Virtual Mission Operations of Remote Sensors with Rapid Access to/from Space,” AIAA-2010-2305, AIAA SpaceOps 2010, Huntsville, Alabama, April 2010.

[2] J. Walke, L. Dikeman, L., S. Sage and E. Miller, “Secure Autonomous Automated Scheduling (SAAS)-Phase 1 Final Report,” NASA/CR-2010-216097, October 2009.

[3] K. Scott and S. Burleigh, “Bundle Protocol Specification,” IETF RFC 5050, experimental, November 2007.

[4] L. Wood, W. Ivancic, W. Eddy, D. Stewart, J. Northam, C. Jackson and A. da Silva Curiel, “Use of the Delay-Tolerant Networking Bundle Protocol from Space,” IAC-08-B2.3.10, 59th International Astronautical Congress, Glasgow, September 2008.

[5] L. Wood, W. Eddy, W. Ivancic, J. McKim and C. Jackson, “*Saratoga*: a Delay-Tolerant Networking Convergence Layer with Efficient Link Utilization,” International Workshop on Space and Satellite Communications (IWSSC '07), Salzburg, Austria, 13-14 September 2007.

[6] L. Wood, J. McKim, W. Eddy, W. Ivancic and C. Jackson, “Saratoga: A Scalable File Transfer Protocol,” work in progress as an internet-draft, draft-wood-tsvwg-saratoga-05, May 2010.

[7] L. Wood, J. McKim, W. Eddy, W. Ivancic and C. Jackson, “Using Saratoga with a Bundle Agent as a Convergence Layer for Delay-Tolerant Networking,” work in progress as an internet-draft, draft-wood-dtnrg-saratoga-07, May 2010.

[8] S. Tatham, “PuTTY: A Free Telnet/SSH Client”, March 2010. http://www.chiark.greenend.org.uk/~sgtatham/putty/

[9] S. Symington, S. Farrell, H. Weiss and P. Lovell, “Bundle Security Protocol Specification,” work in progress as an internet-draft, draft-irtf-dtnrg-bundle-security-16, July 2010.

[10] W. Eddy, L. Wood and W. Ivancic, “Reliability-only Ciphersuites for the Bundle Protocol,” work in progress as an internet-draft, draft-irtf-dtnrg-bundle-checksum-07, May 2010.