IAC-08-B2.3.10
USE OF THE DELAY-TOLERANT NETWORKING BUNDLE PROTOCOL FROM SPACE

Lloyd Wood

Global Government Solutions Group, Cisco Systems, United Kingdom
lwood@cisco.com

Will Ivancic,[*] Wesley M. Eddy,[$*] Dave Stewart,[$*] James Northam,[+] Chris Jackson,[+] Alex da Silva Curiel[+]

[*] NASA Glenn Research Center, Cleveland, Ohio, United States
[$] Verizon Federal Network Systems, Cleveland, Ohio, United States
[+] Surrey Satellite Technology Ltd, University of Surrey Research Park, Guildford, United Kingdom
william.d.ivancic@nasa.gov, weddy@grc.nasa.gov, dstewart@grc.nasa.gov
j.northam@sstl.co.uk, c.jackson@sstl.co.uk, a.da-silva-curiel@sstl.co.uk

The Disaster Monitoring Constellation (DMC), constructed by Surrey Satellite Technology Ltd (SSTL), is a multi-satellite Earth-imaging low-Earth-orbit sensor network where captured image swaths are stored onboard each satellite and later downloaded from the satellite payloads to a ground station. Store-and-forward of images with capture and later download gives each satellite the characteristics of a node in a Delay/Disruption Tolerant Network (DTN). Originally developed for the 'Interplanetary Internet', DTNs are now under investigation in an Internet Research Task Force (IRTF) DTN research group (RG), which has developed a 'bundle' architecture and protocol. The DMC is currently unique in its adoption of the Internet Protocol (IP) for its imaging payloads and for satellite command and control, based around reuse of commercial networking and link protocols. These satellites' use of IP has enabled earlier experiments with the Cisco router in Low Earth Orbit (CLEO) onboard the constellation's UK-DMC satellite. Earth images are downloaded from the satellites using a custom IP-based high-speed transfer protocol developed by SSTL, *Saratoga*, which tolerates unusual link environments. *Saratoga* has been documented in the Internet Engineering Task Force (IETF) for wider adoption. We experiment with use of DTNRG bundle concepts onboard the UK-DMC satellite, by examining how *Saratoga* can be used as a DTN 'convergence layer' to carry the DTNRG Bundle Protocol, so that sensor images can be delivered to ground stations and beyond as bundles. This is the first successful use of the DTNRG Bundle Protocol in a space environment. We use our practical experience to examine the strengths and weaknesses of the Bundle Protocol for DTN use, paying attention to fragmentation, custody transfer, and reliability issues.

1. Introduction

Delay/Disruption Tolerant Networking (DTN) has been defined as an end-to-end store-and-forward architecture capable of providing communications in highly-stressed network environments considered 'unusual' from the perspective of the terrestrial Internet.

To provide the store-and-forward service, a "bundle" protocol sits at the application layer of some number of constituent internets, forming a store-and-forward overlay network.[1]

Key capabilities of the Bundle Protocol include:

- Custody transfer – the ability for a bundle node to take full responsibility for a bundle reaching its final destination.
- Ability for implementations to cope with intermittent connectivity if required.
- Ability for implementations to cope with long propagation delays if required.
- Ability to take advantage of scheduled, predicted, and opportunistic connectivity (in addition to continuous connectivity).
- Late binding of overlay network endpoint identifiers to constituent internet addresses.[2]

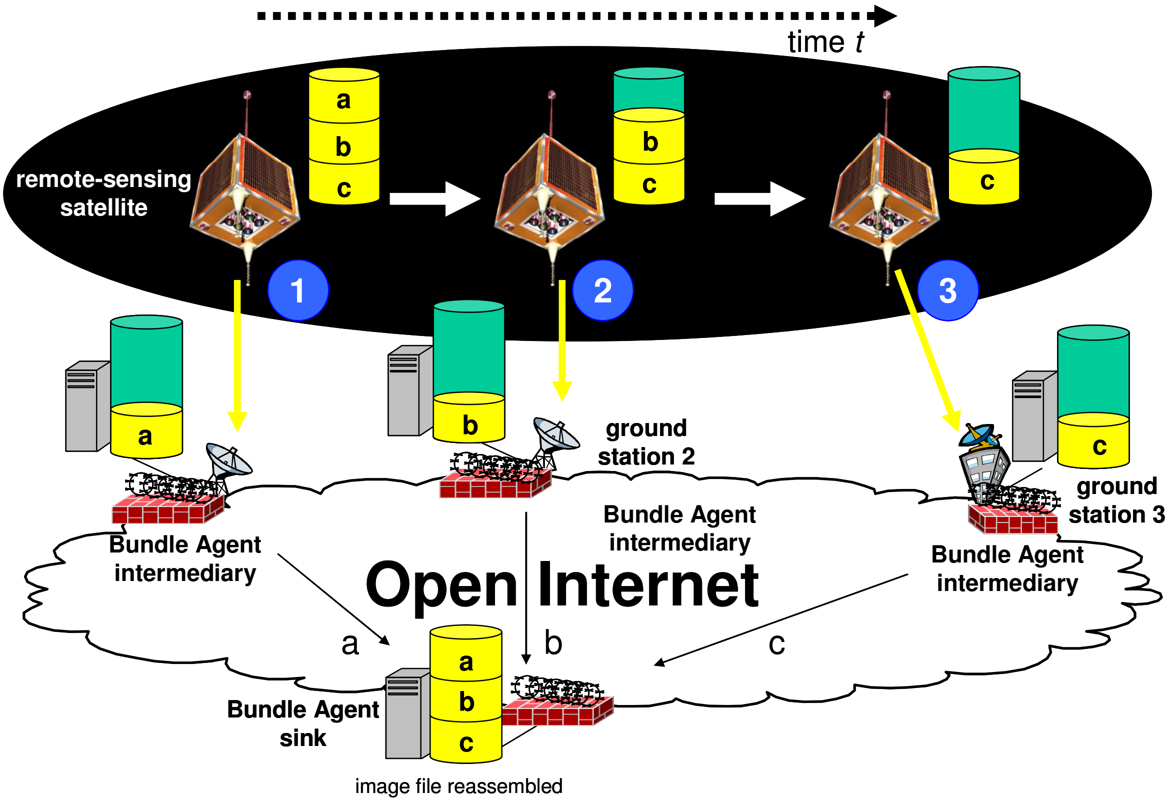


Figure 1: Use of bundling and fragmentation across multiple passes

The Bundle Protocol suite is intended to consist of a group of well-defined protocols that, when combined, enable a well-understood method of performing store-and-forward communications.

DTN networks can be thought of as operating across varying conditions across several different axes, depending on the design of the subnet being traversed:

- low or high propagation delay;
- dedicated or shared, congested links;
- links with intermittent disruption and outages, or scheduled planned connectivity.

In a low-propagation-delay environment, such as may occur in near-planetary or terrestrial environments, bundle agents can utilize chatty underlying Internet transport protocols, such as TCP, that negotiate connectivity and handshake connections in real-time.

In high-propagation-delay environments such as deep space, DTNRG bundle agents must use other methods, such as some form of scheduling, to set up connectivity between the two bundle agents, and can use less chatty transfer protocols over IP.

2. The DMC operating environment

Low Earth Orbit (LEO) is a low-propagation-delay environment of less than ten milliseconds delay to ground, with long periods of disconnection between scheduled passes over ground stations.

For the DMC satellites, contact times consist of 5 to 14 minutes per pass, depending on relative positioning of the ground station and satellite track, with one or two available ground station contact times per 100 minute orbit.

The ground stations are connected across the public terrestrial Internet, which has different operating conditions (shared, competing, congestion-sensitive, always on) from the private links between satellite and ground station (intermittent but scheduled, and dedicated to downloading.)

3. The rate mismatch problem

Figure 1 illustrates a LEO satellite ground network with a bundle agent sink located at a remote location. The final destination for the downloaded imagery could be a satellite control station and office or a laptop ‘in the field’ with wireless connectivity – it really doesn't matter. In this example, an image is to be transferred from the DTN source, the LEO satellite, to the DTN sink. In this example, the image file is too large to be transferred during one pass over a single ground station. Three passes are required to transfer the complete file to ground. These could all be via the same ground station, or could utilize three different ground stations, from left to right in the diagram.

The minimum time a complete image file could be transferred using a single ground station is a little over 300 minutes, assuming one pass per 100-minute orbit. However, using three different ground stations, the entire image could be downloaded in a fraction of an orbit, by downloading fragments of the image to each ground station and reassembling the complete image file on the ground.

If some type of rate-based file transfer is used between the sink and source, problems will arise if ground link capacity does not match or exceed the rate of the space-to-ground link; the transfer becomes limited by any bottleneck in the path. In order to increase the download rates across each link, the transfer can be split into multiple separate hops, where the download is stored and forwarded locally across each hop – note, this is the situation whether using a single ground station or multiple ground stations.

The requirement is to get the image off the spacecraft as efficiently as possible, as spacecraft pass time is the major constraint, and then transfer separately across the different environment of the terrestrial Internet afterwards.

The DTNRG's Bundle Protocol is one example of a way to provide such functionality to split the path into separate hops and control loops. It can thus compensate for rate mismatches between the private space-to-ground link and the shared path between ground station and remote destination for the image.

4. Characteristics of the UK-DMC satellite

The UK-DMC satellite is one of five similar imaging satellites currently launched into low Earth orbit in similar sun-synchronous planes. It was launched in September 2003, with a design lifetime of five years. This imaging constellation continues to grow, with at least four more satellites to be added in the next two years to maintain a continuous on-orbit imaging capability. While these satellites are government-owned, the UK-DMC satellite is also used to provide imagery for commercial resale when not otherwise tasked in imaging campaigns or supporting disaster relief. Anyone may buy a requested image.[3]

The UK-DMC is primarily an operational imaging satellite, and not an experimental satellite. However, SSTL has also run secondary experiments onboard the UK-DMC such as investigating GPS reflectometry[4] and networking experiments have taken advantage of an onboard Internet router.[5,6] SSTL continues to permit NASA to utilize the UK-DMC satellite for experimentation with new forms of networking.

The UK-DMC satellite's onboard payloads include:

- The Cisco router in Low Earth Orbit (CLEO). CLEO has been used for network testing and is its own experiment to simply show that a commercial-off-the-shelf router could survive and function in orbit. CLEO is not used for DTNRG Bundle Protocol testing.
- Three Solid-State Data Recorders (SSDRs)
  - one SSDR based around a StrongARM Processor, supporting the onboard GPS reflectometry experiment.
  - two SSDRs with Motorola MPC8260 PowerPC processors, supporting the imaging cameras. One of these SSDRs is used for DTN testing. These run the RTEMS operating system, which supports the POSIX API and BSD sockets. These have a constrained operating system firmware size limit of 1 MByte, and storage capacities of 1 GByte and 512MByte RAM respectively.
- An uplink of 9600 bits per second, and downlink of 8.134 Mbps – this is highly asymmetric. Both links use the proven IPv4/Frame Relay/HDLC commercial-standard protocol stack developed for space use by Keith Hogie.[7] IPv6 has been tested over these links, using the onboard CLEO router.[8,9] The IP-based transport protocol used for downloading images is SSTL's original implementation of *Saratoga*, retroactively called version 0, running over UDP.

Saratoga version 0 is the existing operational SSTL file transport protocol, originally developed to replace and improve transfer performance rates over an implementation of CCSDS CFDP that was previously used by SSTL. *Saratoga* version 1 is a slightly improved specification, with enhancements to *Saratoga* version 0, which has now been documented publicly as a contribution to the IETF.[10]

How Saratoga can be used as a bundle convergence layer to carry DTN bundles has also been publicly documented.[11]

5. Experimental bundling implementation

5.1. *Onboard the UK-DMC satellite*

Figure 2 shows how DTN bundling is implemented onboard the UK-DMC and in the ground infrastructure.

Saratoga (at time of writing, the operational version 0) acts as a bundle transport 'convergence' layer on the space-ground link. Only the bundle forwarding portion of DTN was implemented onboard as a simple networking "shim" since available code space is constrained. A goal is to have the onboard DTN implementation be transparent to normal UK-DMC operations, living side-by-side with the existing operational code in a non-disruptive manner. This was considered acceptable for testing as the UK-DMC acts only as a source of DTN data, and does not need to receive and parse bundles from elsewhere.

Thus, the DTN-bundle-receiving intelligence only needed to be present in the ground station implementation of the *Saratoga* client and the DTN bundle agent. The *Saratoga* client in the ground station queries the UK-DMC satellite for a directory of files, and then requests any bundle metadata files with a ".dtn" extension and an associated satellite image file. The satellite image file and associated metadata files are transferred to the ground, where the *Saratoga* client reassembles the bundles and then presents them to the full DTN bundle agent – full DTN-2 bundle agent implementations were used both at the ground station and the final DTN destination.[12] Finally, to demonstrate proactive fragmentation, the DTN fragments are reassembled at the final DTN destination.

Implementing bundle functionality on the satellite required that it was first implemented and tested on the ground.

5.2 *Ground development and testing*

Figure 3 shows the DTN ground testbed, where bundling over *Saratoga* was prototyped, with a schematic diagram given in Figure 4.

This development testbed, which reused the CLEO ground-based testbed duplicating in-orbit UK-DMC hardware, contains:

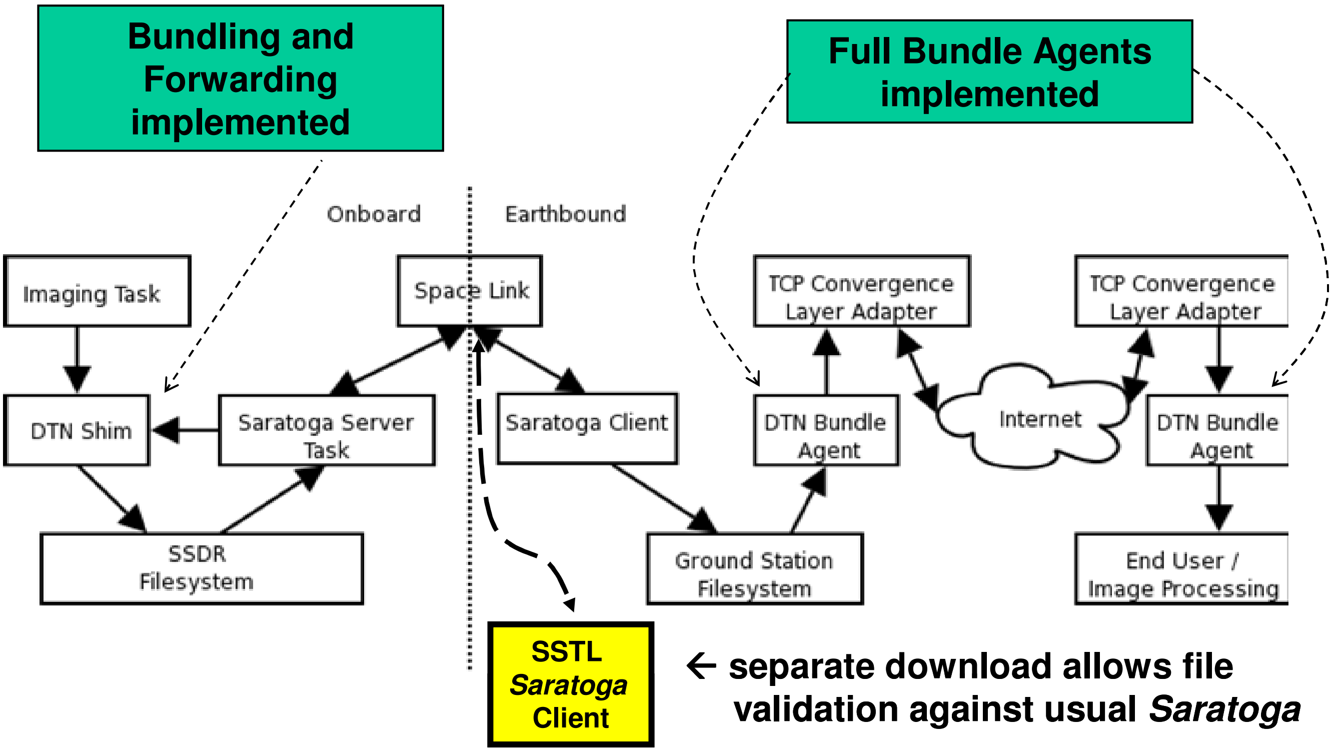
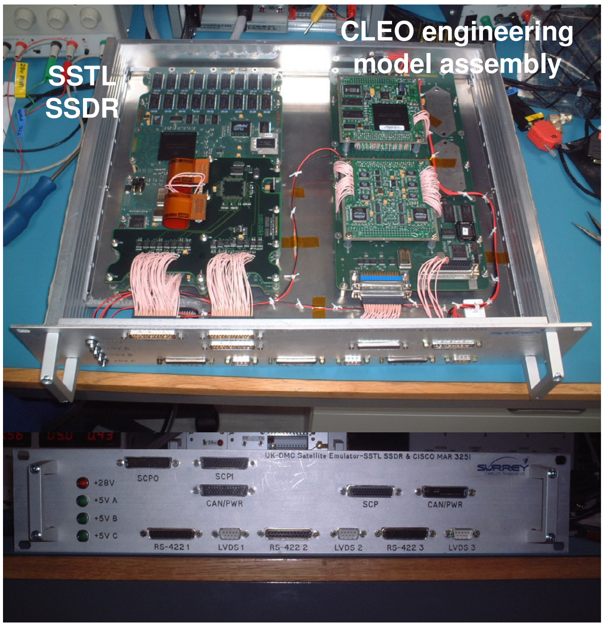

Bundling and Forwarding implemented
Full Bundle Agents implemented
Onboard
Earthbound
Imaging Task
Space Link
TCP Convergence Layer Adapter
TCP Convergence Layer Adapter
DTN Shim
Saratoga Server Task
Saratoga Client
DTN Bundle Agent
Internet
DTN Bundle Agent
SSDR Filesystem
Ground Station Filesystem
End User / Image Processing
SSTL Saratoga Client
← separate download allows file validation against usual Saratoga

Figure 2: How bundling was implemented for downloads from the UK-DMC satellite

- The PowerPC-based Solid-State Data Recorder (SSDR) that resides in the Cisco router in Low Earth Orbit (CLEO) engineering model, where the bundle file is generated.
- A channel emulator that emulates the 9600 bps uplink and the 8.134 Mbps downlink. This uses a Spirent SX-14 data link simulator to provide channel delay and bit-error-rate emulation independently on both the uplink and downlink.
- A DTN bundle agent acting as the ground station, which queries the DTN source onboard the SSDR for files and bundles sent using the SSTL *Saratoga* version 0 file transport protocol.
- A remote sink for DTN bundles – another bundle agent.

All network layer communications used IPv4, with the simulated space/ground data link implemented using Frame Relay/HDLC to match the real link as closely as possible.



a. top view, before adding fans and heatsinks.
b. front view of ports

Figure 3 – CLEO ground-based testbed

5.2 *Overall goals of these bundle experiments*

The goals of the experiments were to:

- demonstrate that NASA Glenn's code

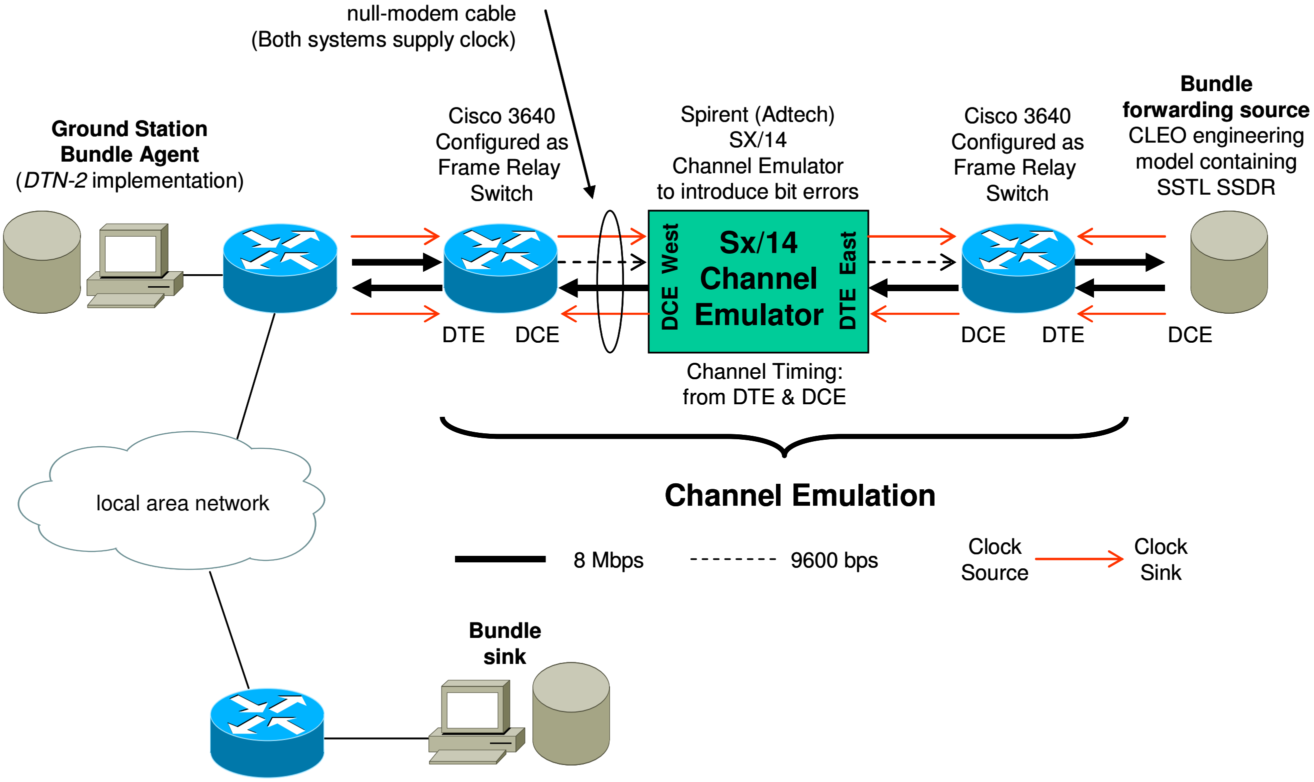


Figure 4: DTN Testbed

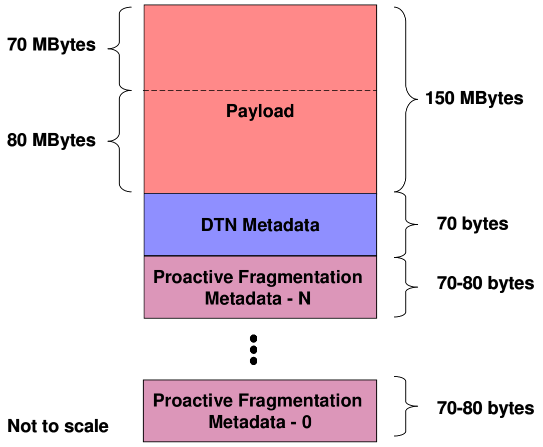


Figure 5: Bundles on the UK-DMC

additions can coexist with SSTL's code without affecting normal SSTL spacecraft or ground station operations;

- demonstrate bundle transfers from the UK-DMC satellite to SSTL and NASA Glenn; and,
- demonstrate proactive fragmentation of bundles to allow downloads across multiple passes.

The ability to run DTN bundling without affecting normal SSTL operations can allow the DTN bundling code to remain loaded as part of the operational system. NASA will not need to take the UK-DMC out of normal operations for dedicated experimental use. This lack of impact on normal imaging operations and decreased opportunity cost will result in significant cost savings for future tests and demonstrations.

Demonstrating normal DTN bundle transfers verifies DTN operation and shows that *Saratoga* can also be used as a bundle convergence layer. Proactive fragmentation allows the download to tolerate disruption between satellite passes, and is required to perform large file transfers over multiple passes and multiple ground stations.

6. Bundling tests from orbit

In order to efficiently run as many bundling tests as possible during a single satellite contact time, an analysis was performed to determine the optimal satellite image size to take.

Calculations showed that, in the pass time available, an image size of approximately 160 Mbytes would allow us to run a full 160-Mbyte file transfer, a 160-Mbyte DTN bundle transfer, and two 80-Mbyte DTN bundle fragment transfers during a satellite pass (single continuous contact).

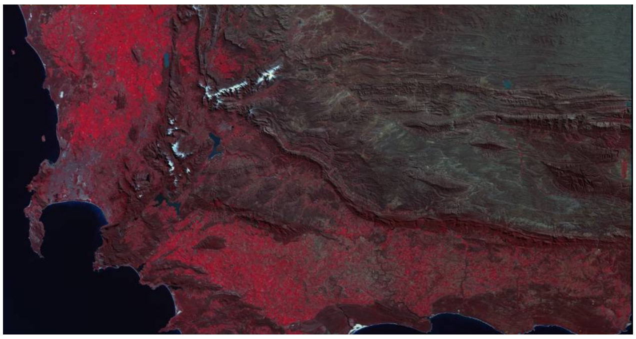


Figure 6: Image delivered via bundles

Figure 5 shows how bundles were created onboard the UK-DMC satellite. When the image was acquired, the large 150-Mbyte image was stored in the SSDR and automatically named by the operating system.

Partially-successful tests of bundling image files over Saratoga were carried out in January 2008. We have previously described those tests and the initial problems that we then encountered in detail.[13]

An unsuccessful image download was carried out during two passes on 26 August, using an older code version that led to corrupted fragments. Replacement code with a bugfix giving correct fragmentation offets was then uploaded to the UK-DMC's SSDR. A remote sensing image swath over South Africa was taken on 08:27 UTC on 27 August 2008. Successful download tests, with reassembly of that proactively fragmented image file downloaded over two passes, were carried out that morning. In these successful tests, the image taken by the UK-DMC satellite's cameras was stored as a single bundle as well as proactively fragmented into two bundles onboard the UK-DMC's SSDR, as shown in Figure 5. These bundle fragments were then downloaded during two passes over SSTL's ground station, to a bundle agent living on a computer donated by NASA Glenn. That bundle agent then forwarded the bundle fragments over TCP to NASA Glenn Research Center, in Cleveland, Ohio, where the fragments were reassembled into a 150-Mbyte file containing the raw sensor data.

That file was then returned to SSTL for post-processing to generate the final image. Figure 6 shows the resulting image of Southern Africa. The Cape of Good Hope and False Bay are to the west. This is a false-colour image; vegetation is red, while the Karoo desert, inland on the plateau, is grey.

The image data was also downloaded using SSTL's standard operational method, using *Saratoga* only, for comparison with the bundle delivery method and validation of the bundle delivery.

We noticed some minor differences in operation and performance between the NASA Glenn and SSTL implementations of *Saratoga*.

The NASA Glenn *Saratoga* implementation can currently time out and reset to requesting the start of the file, rather than the left edge of its window, which needs to be fixed.

The more mature SSTL implementation performs slightly more efficiently by combining selective negative acknowledgements for nearby blocks, even though some unnecessary data resend results. This technique avoids congestion of the bottleneck 9600 bps uplink, leading to better download performance when the bit error rate is high, at the start and end of passes when the satellite is at a low elevation.

7. Known problems and issues with bundling

7.1 Reliability, error detection & checksums

The current Bundle Protocol specification does not address reliability, in that it has no checksum support for error detection and rejection of corrupted bundles. That means that one cannot determine if the bundle information received at each node was received error-free or not.

Error detection is a very basic networking concept that was overlooked in the Bundle Protocol design. The design of the bundle architecture completely ignores the well-known *end-to-end principle*.[14]

Without useful error detection, the Bundle Protocol's custody transfer mechanism cannot guarantee that a node taking responsibility for final delivery of a bundle has actually received an uncorrupted copy of that bundle to send on.

Leaving error recovery up to the applications is only possible when the applications are tightly coupled across the network, with a tight control loop for resends of errored data. DTN networks, by their ad-hoc nature, are loosely coupled, and there may not be any direct communication or control loop between applications at end nodes, requiring increased assistance from the network to improve performance – in line with the end-to-end principle.

We have proposed a workaround to add reliability into the existing protocol infrastructure. This is to use the bundle security specification and to wrap the bundle using a reliability-only cipher rather than a security cipher that provides a reliability check as a side-effect of security.[15] However, the bundle security specification was not implemented onboard the UK-DMC satellite. We have previously described problems encountered due to the lack of error checking in the Bundle Protocol.[13]

Using the bundle security protocol to implement reliability has some drawbacks, in that checking the reliability of secured payloads is not possible. It would be necessary to nest a secured payload within an outer reliability check, much as an IPSec packet can nestle in an Ethernet frame with a strong Cyclic Redundancy Check (CRC) across the entire packet and frame, so that third-party nodes lacking keys to content can check that they have reliably received and are reliably relaying unknown content.

To provide a measure of reliability checking, we have now implemented an optional MD5 checksum for the *Saratoga* protocol, which can be used to compare hash values of files before and after downloading. The MD5 computation can take several minutes to run over a large file, so is likely to be used sparingly onboard. Given that image data is often downloaded in 'one shot' before being deleted to make room for new images, and post-processed heavily with human inspection, the need to resend image files with slight corruption is minor, although knowing where that corruption may lie in the image data would be useful.

However, overall reliability checking becomes very important when e.g. uploading code to be executed. We await further tests to experiment with MD5 checksums onboard.

7.2 Time synchronization problems

During our initial ground testing it became clear that network time synchronization is critical for the Bundle Protocol, which assumes that all communicating bundle nodes share an understanding of local UTC time. This is probably not a reasonable requirement for many DTN networks, as most DTN networks will be nondeterministic. Furthermore, the Bundle Protocol is a network overlay at the application layer that may be running on top of ad-hoc networks in highly stressed environments. The requirement that all DTN networks running the Bundle Protocol must be synchronized to enable interoperation is not necessarily one that is either practical or deployable.

With scheduled LEO passes over a ground station, it is necessary to know what the time is to support the pass opportunity. However, in our initial

CLEO/VMOC testing, nodes in the field at Vandenberg were still able to operate with clocks set several minutes adrift; the loosely-coupled architecture tolerated this.[6]

The clock synchronization problem was experienced during initial ground testing. All DTN bundle agents were originally configured and tested at NASA GRC in Cleveland, Ohio. One bundle agent was sent to Guildford, England. A second was sent to Universal Space Networks (USN) in Alaska. When performing initial bundle transfers from SSTL to GRC to USN, it was noticed that the machine clocks had drifted sufficiently enough to result in the bundle time stamps being out of synchronization. The bundles were therefore rejected due to time-stamp mismatches leading to unexpected expiry of the bundles. Once the machines were resynchronized, bundle transfers operated correctly.

Expecting DTN nodes with loosely-coupled ad-hoc connectivity to be rightly coupled with respect to their understanding of clock time has interesting ramifications. A side effect of requiring shared use of UTC time is that it would not be possible for a node to learn the correct time using the Bundle Protocol, as its bundles sent asking for the time are likely to be judged expired or invalid and be discarded. Another protocol would be required to do clock 'housekeeping'. Another is that for nodes 'in the field' for a long time (decades), some way of communicating newly-decided leap seconds is required to prevent clock drift.

Problems with a shared universal clock were articulated at the 71st Internet Engineering Task Force meeting in March 2008. Others have noted similar problems.[16]

7.3 Agreement on the TCP convergence layer

Multiple different incompatible TCP convergence layers are in already in use for carrying bundles across the terrestrial Internet; not all methods are documented. An agreed way of carrying bundles over TCP needs to be described. There was informal discussion of this at the 72nd Internet Engineering Task Force meeting in July 2008.

There are already a number of documented ways to carry bundles over UDP, including *Saratoga*, which only uses UDP, and the Licklider Transfer Protocol (LTP) whose primary use is over CCSDS protocols. Multiple incompatible ways of carrying bundles directly in UDP are also in use, and need agreement.

It is interesting that, although bundling is intended to work over a wide range of networks and protocols via convergence layers, most of its use and its development has been over IP. IP provides a universal convergence layer that is popular and well-understood.

7.4 Agreement on naming schemes

Different Bundle Protocol implementations are currently supporting multiple different naming schemes for Bundle Protocol Endpoint Identifiers (EIDs), with different rules for forming and interpreting EIDs.

The Bundle Protocol has some degree of built-in naming flexibility by using a generic Uniform Resource Identifier (URI) format for its EIDs, with the URI scheme indicating how the remainder of the EID string should be parsed. However, the DTNRG has not yet rigorously specified or adopted any common EID schemes.

A basic scheme that facilitates initial testing and implementation would be helpful, and would provide a common base for which multiple implementations could be expected to interoperate regardless of their support for other EID schemes. As routing to destinations is meant to be based on EIDs, a common EID format becomes a prerequisite for routing between different DTN networks

7.5 Standardisation of routing methods

The need for common routing protocols is related to the issue of common EID schemes for naming of destinations. Forwarding without any routing protocol is possible through several means:

- if static routes are configured at each node, which is the antithesis of the ad-hoc DTN networks that bundling is intended for.
- if source routing is used, perhaps as a new bundle option.
- if the EID scheme implies forwarding rules somehow through clear use of hierarchy, which can be thought of as a form of source routing.

Automated routing protocols increase scalability, reduce operations and management overhead, and enable operations in completely ad-hoc settings.

It is likely that a number of subnet-specific routing protocols will be needed in order to enable the Bundle Protocol to perform well across the highly diverse range of environments that it is envisioned for. (The Bundle Protocol is already relying upon IP routing protocols to run across the terrestrial Internet.)

Interconnecting different DTN networks poses problems with gateways and sharing of routing information, leading to the separate internal and external routing models used by the Internet – which is complicated by the late binding to addresses of EIDs. With late binding, mapping EIDs to individual subnetworks can be problematic.

Agreement on a very basic routing protocol that simply aids in testing and debugging and may not perform optimally (similar to RIP for IP), would be useful in these early phases of DTN test and development.

Methods for auto-discovery of bundle agents have been proposed and tested, but not yet fully adopted in the DTNRG. Building on auto-discovery, methods of distributing advertisements of routes and predicted contacts would greatly increase the capabilities of the Bundle Protocol and bring it closer to the state needed for operational benefit.

7.6 <u>Network management</u>

DTN nodes currently have no support for remote management as is common in IP networks.

For an operational DTN, it would be very useful to have some type of network management capability, similar to the features of the Simple Network Management Protocol (SNMP) in IP networks. This capability could be used to report on node health, storage issues, undeliverable bundles, performance data, and so on. It could be used to remotely (re-)configure a bundle agent through sending network management bundles to conditionally fetch and set configuration parameters.

A powerful network management protocol might even be able to share functionalities with a DTN routing protocol, as it could be used to add/remove and enable/disable routes on the bundle agents under control.

No work has yet been done on DTN network management, though it seems to be essential in some proposed scenarios where DTN bundle agents are to be operated as long-term infrastructure elements.

However, the long delays and disruption that increase or break end-to-end control loops in certain DTN networks also make network management difficult. It is possible that network management would be subnet-specific, and would use a subnet-specific protocol, e.g. SNMP over IP, rather than the Bundle Protocol itself.

7.7 <u>Complexity</u>

The complexity of the Bundle Protocol's design, with a variety of optional fields, structures, novel binary formats[17] and concepts such as the mutable canonicalisation rules used by security (and thus inherited by reliability), can be considered as a hurdle for implementation, interoperability, and adoption – especially for those pieces of the design that have not yet been fleshed out and agreed.

However, it would be difficult to be as ambitious and all-encompassing as the Bundle Protocol and *not* be complicated.

7.8 <u>Content identification</u>

The Bundle Protocol does not identify the content it carries to select an application to hand the content off to. There is no notion of something similar to an IP port number or protocol ID, or type field, that can be used to pass bundles to higher-layer protocols or applications. This can lead to each EID scheme also supporting some way of indicating applications through the EID, with every application appearing as its own bundle node in the EID space – a problem reminiscent of creation of all the vanity domain names for webservers in the Internet's Domain Name System (DNS).

It can be argued that the web and email have become successful at delivering content partly because it is easy to determine what application should be invoked to receive a delivered file, due to their universal adoption of MIME.[18]

8. <u>Other approaches to DTN networking</u>

The Bundle Protocol is one approach to delay-tolerant networking. Other approaches do not require the Bundle Protocol.

One simple approach, leveraging existing standards, is to use the Hypertext Transfer Protocol (HTTP) as a transport-layer-independent 'session layer' between each two communicating DTN nodes, hop-by hop.[19]

New *Content-Source:* and *Content-Destination:* headers are added, which provide routing information end-to-end. Content-* headers are treated specially: HTTP servers must reject transfers with unknown Content-* headers. Adding these two new headers creates a separate DTN network that will not affect existing traditional web use of HTTP. Reuse and implementation of HTTP in this way to create HTTP-*DTN* appears straightforward.

Fitting HTTP to *Saratoga* for long-delay or private networks is possible. HTTP is already widely used over TCP across the shared, congested, Internet. The two bundle hops used in this scenario – transport of the bundle over *Saratoga* from the UK-DMC's SSDR computer to the bundle agent in a computer in the ground station, then transport of the bundle over TCP across the Internet to NASA Glenn – would be replaced by two HTTP-*DTN* hops: HTTP-*DTN* transfer of the image file over *Saratoga* between satellite and ground station, then an HTTP-*DTN* transfer over TCP between ground station and NASA Glenn's machine. The *Content-Destination:* header would be set to indicate the DNS name of that machine, and resolved with late binding on the last HTTP-*DTN* hop, which is across a subnet that understands that name using DNS. (A static route is used on the wireless first hop for traffic; everything goes down the downlink.)

HTTP provides the ability to easily transfer content identified by MIME, providing the necessary content identification that we have identified as missing from the Bundle Protocol. Here, the MIMEtype used would identify that image data was being sent, to be handled by an image-handling application that handled files of that MIMEtype.

HTTP has a number of existing security protocols that could also be evaluated for suitability for reuse in unusual DTN conditions, on a case-by-case basis.

Conclusions

Delay-tolerant networking Bundle Protocol transfers have now been successfully demonstrated from orbit with the download of sensor data in proactively-fragmented bundles.

This has demonstrated the ability to download data across multiple satellite passes, despite the disruption and link loss experienced between those passes.

The DTN bundling shim onboard the UK-DMC and the ground station *Saratoga* client and bundle reconstitution mechanisms should continue to operate without affecting normal UK-DMC operations, giving NASA access to an operational DTN testbed in orbit when the UK-DMC's busy operational schedule permits.

Our practical experience gained with implementing and operating the Bundle Protocol from orbit enables us to consider aspects of the Bundle Protocol's design.

The lack of integrity checksums for reliability checks in the Bundle Protocol and the need for network time synchronization were shown to be real deployment issues during our first tests, and we are investigating new checksum mechanisms for the Bundle Protocol.

We hope that the problems that we have identified will be addressed in later versions of the DTN architecture and bundling specifications.

The DMC satellites and their use of the Internet Protocol for imaging transfers provide working operational examples of effective use of IP for sensor networks. This allows easy integration with the terrestrial Internet for data delivery. This mission-critical use of the *Saratoga* protocol and IP to carry sensor data performs well on a daily basis, without requiring the Bundle Protocol.

References

1. K. Scott and S. Burleigh, "Bundle Protocol Specification," IETF RFC5050, experimental, November 2007.
2. V. Cerf *et al.*, "Delay-Tolerant Network Architecture," IETF RFC 4838, informational, April 2007.
3. DMC Imaging International: http://www.dmcii.com/
4. S. Gleason *et al.*, "Processing of bistatically reflected GPS signals from low Earth orbit for the purpose of ocean remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 6, pp. 1229-41, June 2005.
5. W. Ivancic *et al.*, "Secure, Network-Centric Operations of a Space-Based Asset: Cisco Router in Low-Earth Orbit (CLEO) and Virtual Mission Operations Center (VMOC)," NASA Technical Memorandum 2005-213556, May 2005.
6. L. Wood *et al.,* "Using Internet nodes and routers onboard satellites," *International Journal of Satellite Communications and Networking*, vol. 25 issue 2, pp. 195-216, March/April 2007.

7. K. Hogie *et al*., "Using standard Internet Protocols and applications in space," *Computer Networks*, special issue on Interplanetary Internet, vol. 47 no. 5, pp. 603-650, April 2005.
8. W. Ivancic *et al.,* "IPv6 and IPSec Tests of a Space-Based Asset, the Cisco router in Low Earth Orbit (CLEO)," NASA Technical Memorandum 2008-215203, May 2008.
9. L. Wood *et al.,* "IPv6 and IPsec on a satellite in space," conference paper B2.6.06, 58th International Astronautical Congress, Hyderabad, India, September 2007.
10. L. Wood, J.McKim, W. M. Eddy, W. Ivancic and C. Jackson, "Saratoga: A Scalable File Transfer Protocol," work in progress as an internet-draft, draft-wood-tsvwg-saratoga-01, February 2008.
11. ——, "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-03, February 2008.
12. DTN reference implementation, October 2007 release, available from http://www.dtnrg.org/wiki/Code
13. W. Ivancic, W. M. Eddy, L. Wood, D. Stewart, C. Jackson, J. Northam and A. da Silva Curiel, Delay/Disruption-Tolerant Network Testing Using a LEO Satellite, NASA Earth Science Technology Conference (ESTC 2008), University of Maryland, June 2008.
14. J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277-288, November 1984.
15. W. M. Eddy, L. Wood and W. Ivancic, "Checksum Ciphersuites for the Bundle Protocol," work in progress as an internet-draft, draft-irtf-dtnrg-bundle-checksum-02, March 2008.
16. W. M. Eddy, "DTN Time Sync Issues," email to the IRTF dtn-interest mailing list, 1 April 2008, and subsequent discussion.
17. W. M. Eddy, "Using Self-Delimiting Numeric Values in Protocols," work in progress as an internet-draft, draft-eddy-dtn-sdnv-03, July 2007.
18. N. Freed and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," IETF RFC 2045, November 1996.
19. L. Wood and P. Holliday, "Using HTTP for delivery in Delay/Disruption-Tolerant Networks," work in progress as an internet-draft, draft-wood-dtnrg-http-dtn-delivery, February 2008.