Saratoga: a Delay-Tolerant Networking convergence layer with efficient link utilization

Lloyd Wood, Wesley M. Eddy, Will Ivancic, Jim McKim and Chris Jackson

Abstract—**Saratoga is a lightweight transport protocol based on the User Datagram Protocol (UDP/IP). Saratoga was developed by Surrey Satellite Technology Ltd (SSTL) for file transfers of imaging data recorded onboard the Internet-Protocol-based Disaster Monitoring Constellation (DMC) satellites, and has been in operational use from low Earth orbit since 2004. Saratoga focuses only on efficient communication to the next hop when link connectivity is available, by filling the link with packets sent at line rate. This ensures that as much data as possible is transferred to the peering node during a twelve-minutes-or-less pass over a satellite ground station. Saratoga uses a minimal bandwidth-efficient negative acknowledgement mechanism to ensure reliable data transfer. We examine how Saratoga can be adapted to serve as an efficient convergence layer for Delay-Tolerant Networking (DTN), by transferring DTN bundles as well as files. This will allow DTN networks to increase efficiency of communication across briefly-available disrupted links – for long-distance deep space links as well as for short-distance terrestrial mobile ad-hoc networks.**

Index Terms—**UDP, Saratoga, Delay Tolerant Networking, DTN**

I. INTRODUCTION

Delay-Tolerant Networking was originally envisaged as a necessary part of the “Interplanetary Internet”: a store-and forward architecture able to cope with extremely long delays and intermittent connectivity [1]. As DTN is store-and forward in nature, it is often compared to Internet email. However, email depends upon the Domain Name System (DNS) and the Simple Mail Transfer Protocol (SMTP) – which both depend upon the Transmission Control Protocol (TCP). TCP is useful for end-to-end communications across a shared terrestrial Internet where fairness between simultaneous flows competing for link capacity and an awareness of congestion is the norm, and not for store-and-forward communications across a single unshared link, which a lone TCP flow is unable to use effectively [2]. As a result, the DTN must be independent of TCP in the store-and-forward network for which DTN is intended. For long-distance interplanetary communications and for the mobile ad-hoc and sensor networks for which the DTN architecture was later found suitable under the ‘disruption’ name [3], a transport protocol other than TCP is needed. However, the DTN can use TCP or another congestion-aware protocol, such as DCCP or SCTP, to safely and fairly traverse the shared terrestrial Internet between DTN nodes that may also be intermittently connected to this other, different, ad-hoc or Interplanetary Internet.

DTN calls the transport protocols that it uses to move data across different networks *convergence layers*. A convergence layer should make best use of the intermittent and temporary links in ad-hoc networks, by transferring as much data as possible while the link between peering DTN nodes is up and available. Such a convergence layer will also benefit from running over the Internet Protocol (IP), as IP is almost universal, and many existing links support IP. Running over IP avoids reinventing the wheel in re-implementing support for DTN over many different media. TCP is unsuitable for and inefficient in these networks due to its assumptions about congestion. However, making the convergence layer IP datagram-based enables widespread compatibility and use. UDP is an obvious starting point for an IP-based transport that does not need to be aware of traditional Internet congestion. Designed with those constraints in mind, the *Saratoga* protocol outlined in this paper can be that DTN convergence layer.

II. OVERVIEWING SARATOGA

Saratoga is intended to provide a single flow between two peers, multiplexing multiple separate sessions, across a dedicated link outside the “fair competition” remit of the traditional shared Internet. Saratoga is designed for performance, to fill the link with traffic as efficiently as possible. Link utilization is an explicit design goal; handling Internet congestion from competing flows is not.

Saratoga is a rate-based UDP file transfer protocol capable of transferring large files. Saratoga has been in operational use since 2004 to move mission imaging data from the Disaster Monitoring Constellation (DMC) remote-sensing satellites to ground stations. These satellites, built by Surrey Satellite Technology Ltd (SSTL), all use IP for payload communications. Five satellites have been launched so far. Four are currently operational in low Earth orbit and

---

Manuscript prepared for camera proof 21 June 2007.

Work on this document at NASA's Glenn Research Center was funded by NASA's Earth Science Technology Office (ESTO).

L. Wood is with the Global Government Solutions Group, Cisco Systems, Bedfont Lakes, Feltham, Middlesex, England (email lwood@cisco.com).

W. M. Eddy is with Verizon Federal Network Systems, contracted to NASA Glenn Research Center (email weddy@grc.nasa.gov).

W. Ivancic is with NASA Glenn Research Center, Cleveland, Ohio (email William.D.Ivancic@grc.nasa.gov).

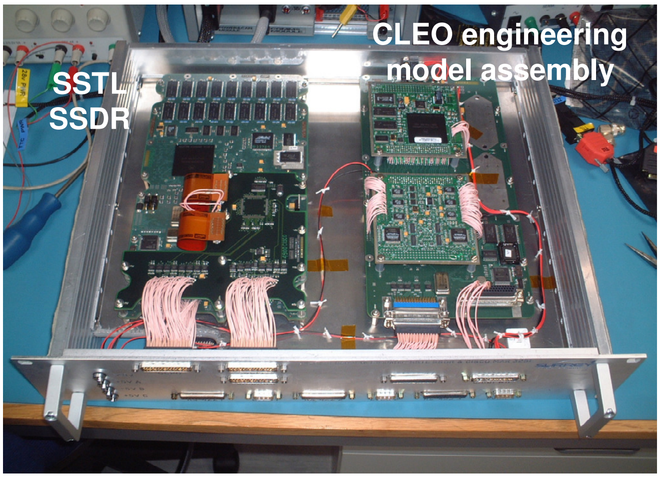
J. McKim is with RS Information Systems, contracted to NASA Glenn Research Center (email James.H.McKim@grc.nasa.gov).

C. Jackson is with Surrey Satellite Technology Ltd, University of Surrey Research Park, Guildford, England (email C.Jackson@sstl.co.uk).

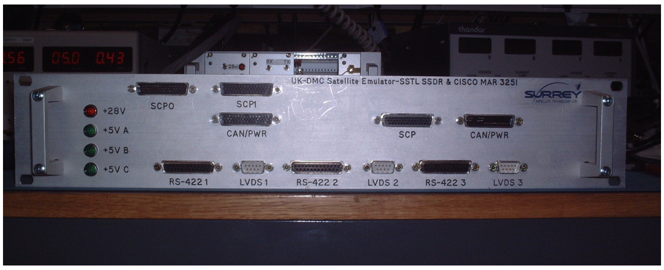
transferring imagery files, with ground sample distance (GSD) resolutions of 32 metres or better [Fig. 1], to ground [4].

False-colour imagery. Green is vegetation; red is flooded. Full image taken is shown in corner thumbnail. Supplied to the US Geological Survey by DMC International Imaging.

Fig. 1. Example DMC imagery - New Orleans, aftermath of Hurricane Katrina, NigeriaSat-1, 2 September 2005.



a. top view before addition of heatsinks/fans



b. front view showing interface ports and flexible patchbay.

Fig 2. CLEO ground-based testbed used for development

Saratoga was originally designed to transfer files of only up to 4GB in size, but has evolved to keep pace with the needs of higher-resolution, ever-larger, imaging files. Three more DMC satellites, with higher-resolution imaging capabilities, are now under construction. How the DMC satellites use IP to communicate with the ground is described elsewhere [5,6].

Saratoga was originally developed to move large files from these remote sensing satellites to ground stations during a pass of twelve minutes' duration or less. This limited contact time drove a desire to transfer as much error-free data as possible. Saratoga relies on the UDP checksum for basic error checking, and can implement an end-to-end checksum..

Saratoga displaced an implementation of the CCSDS File Delivery Protocol, CFDP [7], that was used previously [8].

Although Saratoga was originally developed for file transfers, it was recognized that this protocol was also useful for DTN bundle exchange [9], by a simple mapping between bundles and files.

III. OTHER CONVERGENCE LAYERS

Another UDP/IP protocol intended as a DTN convergence layer and under development is the Licklider Transmission protocol (LTP) [10].

Saratoga and LTP share many of the same design motivations [11]. The main design difference between LTP and Saratoga is that LTP transfers arbitrary un-named blobs of data, while Saratoga transfers named files, including some degree of file metadata and directory listings.

CCSDS CFDP, which also includes file-based transfers, may be used above LTP (or further above bundling) to provide a similar file-handling service. CFDP, first created by LTP's primary designer, included a file system to encourage bitstream-oriented spacecraft payload designers to move to adopting the decades-old computing filesystem paradigm; spacecraft and recorder bitstreams predate onboard computing. Like bundling and LTP, CFDP is quite diverse in its operational options. The Messenger mission to Mercury is only flying pieces of CFDP deemed immediately useful in a "CFDP-Lite" [12], while other parts of that protocol were rejected for use onboard Messenger.

There is clearly an opportunity for a UDP-based transfer protocol to act as a DTN convergence layer for the ad-hoc or store-and-forward new Internet, including the Interplanetary Internet. We believe that simplicity of specification and implementation, and efficiency for low-end onboard processors, along with ease of use, will drive adoption.

Other convergence layers have been suggested to support interfacing the DTN bundle layer directly to a link layer, e.g. directly over Bluetooth, or directly over 802.11. While such an approach may be able to include link-specific engineering optimizations, we believe that the per-link-layer engineering required makes this approach prohibitive, and that the cost of engineering and maintaining the resulting code is not scalable given the vast number of diverse link layers in existence.

Many of these link layers already support IP, so an IP-based convergence layer requires little extra work for a new link layer that already supports IP. The contrast between an IP-based shim approach and a direct approach to the DTN network stack is shown [Fig. 3].

IV. DEVELOPMENT HISTORY

The Saratoga protocol was originally designed and implemented by SSTL for its dedicated 200MHz Power-PC-processor computers flown as Solid State Data Recorders (SSDRs). These run the RTEMS real-time embedded operating system. SSDRs control imaging cameras or onboard experiments.

Saratoga replaced an RTEMS implementation of CFDP also written at SSTL, and previously used on orbit. In this operational experience, the implementation of Saratoga was found to be more lightweight in both code footprint and processor overhead, and, more importantly, to significantly outperform CFDP in transfer speed and the amount of data that could be copied to ground within a single satellite pass.

With an S-band downlink speed of 8.1Mbps, roughly 650MB, or a CD-ROM's worth of data, can be transferred during a twelve-minute pass – but only if packets are output continuously at line rate on the downlink. Saratoga can achieve this in the given limited hardware environment; the implementation of CFDP it replaced could not. The UK-DMC satellite's onboard SSDRs have 1 and 1.5 GB of RAM respectively for storage, meaning that the initial 4GB file limit of Saratoga's early design is not reached.

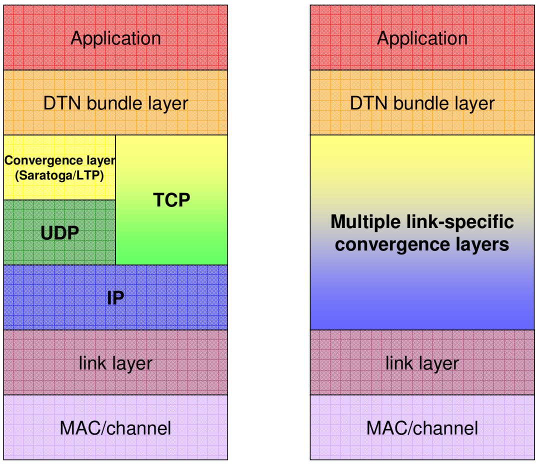
Implementations of Saratoga have since been used at 20 and 40 Mbps on the X-band downlink supporting the high-resolution panchromatic four-metre GSD imager on Beijing-1, the Chinese satellite in the DMC, again filling the downlink at line rate.

While Saratoga was being implemented in 2004, preparations were being made to test CLEO, the Cisco router in Low Earth Orbit, onboard the UK-DMC satellite [5].

Part of these preparations involved construction of a ground-based testbed, containing a sister to the CLEO router assembly in orbit, alongside an SSTL SSDR [Fig. 2]. This testbed, which interfaces to a PC acting as the computer running the satellite platform, was given to NASA Glenn Research Center, in order to allow them to become familiar with the operational configuration of SSTL's computers. The sister router was set up to work with that configuration, via experience with the testbed, making the most of limited access to CLEO. NASA has a long history of testing IP-based technologies with SSTL, dating back to UoSAT-12 [6].

That testbed served CLEO successfully in a configuration role, and allowed rapid setup on orbit. After the successful testing of CLEO, we realized that the testbed could be used in a new development role to test embedded software for the SSTL SSDRs. The compiler tools and build chain were also installed at NASA Glenn. This was seen as an opportunity to work towards testing DTN bundle code operationally in orbit.

Within this work, a consideration of the available methods for bundle convergence layers revealed that no good candidates for long-distance space links were available. All available DTN code was tailored for terrestrial use and testing, and lacked any rate-based transfer options for large bundles.



a. using IP as universal shim b. direct on link.

Fig. 3. Stack diagrams of convergence layer approaches

Some of the available convergence methods involved using a TCP connection with the associated startup and loss issues, or sending a bundle within a single UDP datagram (limited to a maximum of 64KB). We believe that useful bundles will be far larger than 64KB, and often larger than tens of gigabytes.

LTP code then available was written in Java and unsuitable for integration with the RTEMS environment.

It was soon recognized that the Saratoga protocol developed at SSTL had something new to offer as an efficient convergence layer, and work began to focus on mapping DTN bundles onto Saratoga and its filesystem support.

V. DESIGN CONSTRAINTS

As the UK-DMC satellite is effectively a stub network interfacing to few ground stations, and primarily delivering data rather than receiving it and routing it elsewhere, dynamic DTN addressing and routing are outside the scope of our area of focus. Large data files generated by the onboard imaging task can be encapsulated in bundles for delivery to a static ground address. The bundle forwarding operation consists entirely of using Saratoga to pass the bundles down to any in-contact ground station.

Data is delivered to the ground station via the satellite's 8.1Mbps downlink, while commands are received over a limited 9600bps uplink. The highly asymmetric uplink/downlink environment mandates efficiency, and Saratoga has a terse negative-acknowledgement and hole-filling strategy that is suitable for this 170:1 link asymmetry. (TCP is widely recognized to be unsuitable for links where asymmetry exceeds 50:1, due to congestion of the reverse path by acknowledgements [13]. This is separate from, and in addition to, other congestion-aware design choices, such as slow start, that prevent effective local link utilization by TCP.)

Some means of recovery of packets lost due to channel errors is needed, but these packet losses do not cause the sending peer to decrease its sending rate.

High link utilization while connected is important to get the most utility out of limited and sporadically-available periods of connectivity.

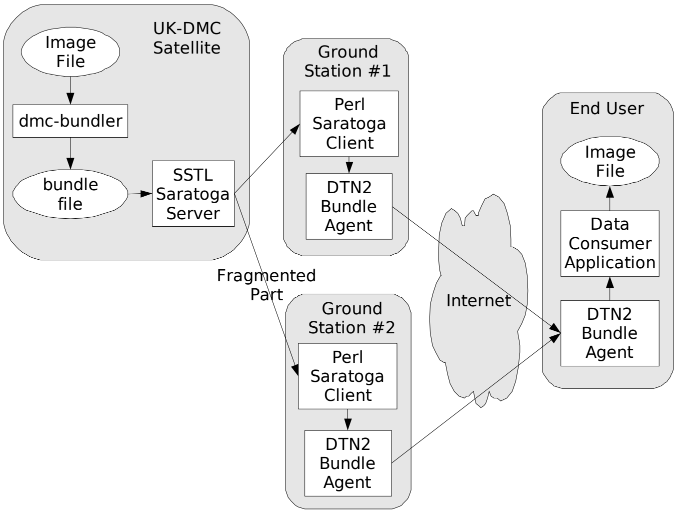


Fig. 4. Envisaged demonstration use of Saratoga.

Congestion control to arbitrate fairly between competing traffic is not important, as nodes peer across dedicated links without competition from other nodes, or even from themselves thanks to multiplexing all sessions within a single flow. Any sharing of link capacity uses an explicit coarse-grained scheduling model, and not simultaneous competition.

In this model, individual payloads take turns to get dedicated use of the link and deliver the information that they hold, before shutting up and shutting down to conserve power [5]. Only low-rate broadcast telemetry is multiplexed in as well while a payload delivers data. Thus, loss is guaranteed to never be due to congestion. In a shared wireless ad-hoc environment, allocation of channel resources to nodes becomes a MAC-layer function, rather than a transport function, and so a similar consideration may apply.

This model has other ramifications: as there is only one flow containing all sessions traveling across a single link, Quality of Service (QoS) can be most effectively carried out by the sending bundle application interleaving and prioritizing an urgent session by sending that session's packets out ahead of others. The differentiated services architecture becomes irrelevant for a single hop; DiffServ's strength is in preserving priority in delivery between competing flows across multiple hops, and there is little useful work for it to do across a single hop between peers with no competition,

Output and input queuing also become less important; the Saratatoga-level retransmission mechanism can work directly from a file stored in main memory, and thus allow for shallower link queues and faster recovery than when link-layer ARQ and its associate queues are heavily relied upon.

Forward error coding (FEC) to get the most reliable transmission through a channel is best left near the physical layer in the modem, and complements the simple packet-ARQ-based negative acknowledgement strategy used by Saratoga to deal with the errors not handled by coding. When in use with a non-adaptive modem using a fixed coding scheme, Saratoga sees and recovers from most errors at the start and end of a pass over a ground station, when the satellite is at its lowest elevations and the atmospheric attenuation is highest. This hybrid strategy, with dual complementary reliability mechanisms, is considered the best tradeoff between choice of the constant added capacity overheads of heavy coding, or the added delays and windowing overheads of heavy acknowledgement use. This also prevents well-known potentially-problematic interactions between retransmission mechanisms present in the link layer and also in higher layers.

Being in LEO at 686km altitude, with a five millisecond round-trip time from satellite to ground station, allows use of both FEC and ACK mechanisms. The longer delays of deep-space links encourage stronger FEC to minimise use of closed-loop control and the delay overhead that ARQ would impose.

VI. WORK ON SARATOGA

Saratoga is being reworked for a public specification post the proof-of-concept implementations currently in use in orbit (all 'version zero' [14]). This new 'version one' of Saratoga is being prepared for demonstration in the space environment onboard the UK-DMC satellite [Fig. 5]. This demonstration aims to:

A. show Saratoga as the first efficient DTN convergence layer adapter for the space environment.
B. show large file and bundle transfers involving fragmentation and reassembly.
C. show use of multiple independent ground stations for a single file or bundle transfer.
D. demonstrate a lightweight DTN implementation technique for "constrained" nodes, typical of sensor networks, and space networks, that is upwardly compatible with more full-featured code such as DTN2.

This later version 1 is capable of running over either IPv4 or IPv6. The CLEO router in orbit was configured to run IPv6 as well as IPv4 [15]. This provides something onboard in orbit to test IPv6 compatibility against. Saratoga transfers could be carried out onboard between an SSDR running an onboard GPS reflectometry experiment and an imaging SSDR [5].

Another significant addition to version 1 of Saratoga is the ability to coordinate Saratoga operations with a local DTN bundle agent. A flag in the Saratoga file metadata is used to indicate that a transferred file should be treated as a bundle by the receiver and passed to the bundle agent, rather than simply placed as a file in storage. In order to facilitate this, we wrote a "BundleFile" convergence layer adapter for the commonly-used DTN2 bundle agent software [16] that takes the path of a file on disk as input over a POSIX named-pipe, reads the file data as a bundle, and unlinks the file from the filesystem, freeing up this space for Saratoga to use for other incoming bundle files.

As shown in [Fig. 4], there are currently two different Saratoga implementations of this version that interoperate. The SSTL Saratoga server onboard the SSDR (in the CLEO testbed, and, eventually, onboard the UK-DMC satellite) is a slight adaptation of SSTL's flight code for Saratoga version 0, while the Perl Saratoga client, for use on the ground, was written from scratch and used to help find the points that needed to be clarified in the specification. A Python Saratoga version 1 client has also been written.

Aside from Saratoga, another interesting facet of the demonstration shown in [Fig. 4] is the "dmc-bundler" code that is used to translate an image file into a bundle file that Saratoga can use. Normally, a full-fledged bundle agent such as DTN2 or JPL's Interplanetary Overlay Network (ION) handles this wrapping of application data units into DTN bundles. These bundle agents provide many services in addition to the basic bundling of data through, including making forwarding decisions, and managing multiple convergence layers and routing information sources. In the more limited case of a UK-DMC SSDR, there is only one suitable convergence layer available (Saratoga) and the routing information is essentially a binary link-up / link-down signal that is provided to Saratoga rather than to the bundle layer. This results in dmc-bundler code that is very simple and feature-free in comparison to other bundling implementations, with a very small code footprint.

The dmc-bundler code operates as a function within the shared memory model of RTEMS, while DTN2 and ION use separate processes and interprocess communication mechanisms. Given the small number of pre-existing tasks on an SSDR and the small amount of lines of code in the onboard software, adding one of the available full-fledged bundle agents would have nearly doubled the number of running tasks and more than quadrupled the number of lines of code in the system that have to be checked and maintained. Obviously, for this limited scenario, the dmc-bundler approach is desirable, and we speculate that this could be used for lightweight support of DTN bundle producers in other types of sensor networking, where computation capabilities and space for code are limited. For situations that actually involve adaptive routing and multiple dynamic convergence layers, this is clearly a too-simplistic approach, and a full-fledged DTN bundle agent platform must be used.

While this paper has discussed near-Earth disruption-biased DTN, a "put" mechanism in Saratoga allows it to efficiently use deep-space links where delay-biased DTN is prevalent, while optional use of UDP-Lite can allow some robustness to errors received when sending over unidirectional links. Comparison and contrast of Saratoga and LTP performance for these scenarios would be interesting future work.

Saratoga has been specified in an Internet draft and proposed to the Internet Engineering Task Force (IETF) as an experimental protocol [17]. Further work with Saratoga would instrument the flight code and measure its performance in use transferring bundled imagery data from the UK-DMC satellite, paying particular attention to robustness in the face of link errors at the start and end of passes.

VII. A NOTE ON NAMING

Saratoga is named for the USS Saratoga (CV-3), the aircraft carrier sunk at Bikini Atoll that is now a popular diving site. The philosophy behind the Saratoga protocol's use for DTN can be succinctly stated as Saratoga Carries Upper Bundles Adequately, or SCUBA.

VIII. CONCLUSIONS

We have presented and described Saratoga, a new convergence layer for Delay-Tolerant Networking.

Saratoga has the advantage of already being implemented, tested and run operationally on mission-critical data in a demanding environment – space. This experience with Saratoga as a mature, in-use protocol bodes well for taking it to the IETF with a public specification [17].

We envisage a world where Saratoga can be used over Internet protocols for DTN in ad-hoc, sensor and store-and forward networks. At the same time, DTN bundles will traverse the terrestrial fixed Internet as congestion-controlled flows, using TCP, DCCP, or similar. The convergence layer adapts DTN not only for the link, but for the operational assumptions within which the link is used.

Multiple DTN convergence layers can coexist over IP within the different environments that the different convergence layers have been designed for.

REFERENCES

[1] V. Cerf, S. Burleigh *et al.*, "Delay Tolerant Network Architecture," IETF RFC 4838, April 2007.

[2] L. Wood, C. Peoples *et al.*, "TCP's protocol radius: the distance where timers prevent communication," Third International Workshop on Satellite and Space Communications (IWSSC '07), September 2007.

[3] K. Fall, "A Delay Tolerant Networking Architecture for Challenged Internets," ACM SIGCOMM 2003, August 2003.

[4] C. Underwood, S. Machin, P. Stephens, D. Hodgson, A. da Silva Curiel and M. Sweeting, "Evaluation of the Utility of the Disaster Monitoring Constellation in Support of Earth Observation Applications," 5th IAA Symposium on Small Satellites for Earth Observation, Berlin, Germany, 4-8 April 2005.

[5] L. Wood, W. Ivancic, C. Jackson *et al.*, "Using Internet nodes and routers onboard satellites", *International Journal of Satellite Communications and Networking*, volume 25 issue 2, pp. 195-216, March/April 2007.

[6] K. Hogie, E. Criscuolo and R. Parise, "Using standard Internet Protocols and applications in space," *Computer Networks*, vol. 47 no. 5, pp. 603-650, April 2005.

[7] *CCSDS File Delivery Protocol (CFDP)*, CCSDS Blue Book, issue 4, January 2007.

[8] AlSAT-1 DMC satellite working well in orbit with first use of IP, SSTL press release, 6 December 2002.

[9] K. Scott and S. Burleigh, "Bundle Protocol Specification," work in progress as an IETF internet draft, April 2007.

[10] M. Ramadas *et al.*, "Licklider Transmission Protocol – Specification," work in progress as an IETF internet draft, April 2007.

[11] S. Burleigh *et al.*, "Licklider Transmission Protocol – Motivation", work in progress as an IETF internet draft, April 2007.

[12] Special Delivery: NASA's MESSENGER Sends Flyby Data to Earth Using CCSDS Delivery Protocol Developed for Deep Space by International Team, CCSDS press release, 10 August 2005.

[13] H. Balakrishnan, R.H. Katz, and V.N. Padmanbhan, "The Effects of Asymmetry on TCP Performance", ACM Mobile Networks and Applications, vol. 4, no. 3, pp. 219-241, October 1999.

[14] C. Jackson, "Saratoga File Transfer Protocol," internal SSTL technical document no. DMNG-61536, 2004.

[15] L. Wood, W. Ivancic, C. Jackson *et al.*, "IPv6 and IPsec on a satellite in space," 58th International Astronautical Congress, Hyderabad, India, September 2007.

[16] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra, "Implementing Delay Tolerant Networking", Intel Research Berkeley Technical Report IRB-TR-04-020, December 2004.

[17] L. Wood, J. McKim, W. Eddy, W. Ivancic and C. Jackson, "Saratoga: A Convergence Layer for Delay Tolerant Networking," work in progress for an -01 internet-draft, submitted to the IETF for the July 2007 meeting.