Network Working Group
Internet-Draft
Intended status: Experimental
Expires: October 21, 2014

L. Wood
Surrey alumni
J. McKim
RSIS
W. Eddy
MTI Systems
W. Ivancic
NASA
C. Jackson
SSTL

April 19, 2014

Using Saratoga with a Bundle Agent as a Convergence Layer for Delay-Tolerant Networking

draft-wood-dtnrg-saratoga-14

Abstract

Saratoga is a simple, lightweight, UDP-based transfer protocol. This describes how to use Saratoga as a Delay-Tolerant Networking (DTN) "convergence layer" with the Bundle Protocol and its Bundle Agents, building on the Saratoga specification in draft-wood-tsvwg-saratoga.

Status of This Memo

Copyright Notice

Table of Contents

1. Background

The Saratoga protocol is specified in [I-D.wood-tsvwg-saratoga]. Saratoga can optionally be used for Delay/Disruption-Tolerant Networking (DTN) [RFC4838], as a "convergence layer" to exchange bundles between peer nodes. Saratoga was originally designed prior to work on the Bundle Protocol [RFC5050] by the DTN Research Group (DTNRG). It was later recognized that Saratoga could also be used to reliably exchange bundles between DTNRG Bundle Agents by using a logical mapping from DTNRG bundles to Saratoga files and back. The DTN concept encompasses networks where ad-hoc, intermittent connectivity is expected, connections may be infrequently established or short-lived, and end-to-end paths are not present. As Saratoga was designed for intermittent, disrupted, space communications, Saratoga's operating environment is a DTN network. This makes the Saratoga transfer protocol a natural fit as a convergence layer for the DTNRG's Bundle Protocol, although the Bundle Protocol is not necessary for operation of Saratoga.

This document contains notes on use of Saratoga for the bundle transfer procedure.

Bundle transfers over Saratoga from space and the UK-DMC satellite, with use of proactive fragmentation, have now been demonstrated. More details are provided in [IAC-2008] [SSTL-2008] [Ivancic10].

2. Applicability Statement

Why use Saratoga as a DTN convergence layer? The DTN architecture already has a number of choices of convergence layer. Convergence layers have been proposed for various link types, e.g. Ethernet or

Bluetooth. As IP already runs over many link types, a convergence layer that can run over many links using IP is likely to take advantage of TCP or UDP.

For traversing the terrestrial Internet while supporting congestion control, a simple TCP convergence layer has been implemented in the DTN software reference implementation [I-D.irtf-dtnrg-tcp-clayer]. A simple UDP convergence layer, able to be used over dedicated private links where congestion control is not required, is also present. However, that simple UDP convergence layer presumes that a bundle will always fit into a single UDP packet, does not support segmentation of bundles across multiple UDP packets, and does not guarantee reliable delivery with retransmissions. Its use is discouraged [I-D.irtf-dtnrg-udp-clayer].

Two protocols capable of supporting segmentation of large bundles across multiple UDP packets, with ARQ-based flexible delivery robust to packet loss, are Saratoga [I-D.wood-tsvwg-saratoga] and the Licklider Transmission Protocol (LTP) [RFC5325].

Both Saratoga and LTP were designed based on experience gained with using the CCSDS File Delivery Protocol (CFDP), which was developed for the Consultative Committee for Space Data Systems (CCSDS). The main design difference between LTP and Saratoga is that LTP transfers arbitrary un-named data blobs (binary large objects), requiring a higher layer (normally delay-tolerant-networking bundling) to handle naming, while Saratoga transfers named files including file metadata, and can be independent of higher layers. Both protocols can run over the User Datagram Protocol, UDP [RFC0768], though LTP also is intended to run at other layers in the stack (including directly over the link), while Saratoga is only intended to run above the UDP or UDP-Lite transport protocols. If errors in delivered content can be tolerated (perhaps because the data being transferred has its own integrity checks), Saratoga can also be used to transfer an entire file or stream without error checking, using UDP-Lite [RFC3828], which can protect only header content from errors.

Saratoga includes a file checksum mechanism to detect transfer errors and to provide an overall degree of reliability. Licklider has no similar reliability mechanism, although Licklider's optional security mechanism [RFC5327] can be implemented to give some error detection.

Saratoga can also be used for delivery over unidirectional broadcast links. Another UDP-based convergence layer proposed for unidirectional links is Uni-DTN [I-D.kutscher-dtnrg-uni-clayer]. Uni-DTN is based on FLUTE forward layered coding for multicast delivery. Saratoga presumes that the forward error coding needed to

prevent errors in transmission is present at another layer in the stack, usually near the physical layer.

3. Using Saratoga with a DTN Bundle Agent

While Saratoga was first developed for efficient file transfer, the similarity between bundle payloads and files, in that both are arbitrary blobs of some number of octets, allows Saratoga to be used as a convergence layer for exchanging bundles between DTN bundle agents. This section explains the basic concepts involved in mapping bundle exchange onto the file transfer mechanism.

Routing of bundles is outside the scope of Saratoga and of this document. Once a complete bundle file has been transferred between peers using Saratoga, that bundle can be forwarded onwards along a next available hop in any way. Saratoga provides a mechanism for forwarding, but does not provide input to routing or forwarding decisions.

A DTN bundle agent can work alongside a Saratoga peer to move bundles. One simple method of communicating bundles between the bundle agent and the Saratoga peer is to have a shared directory that is accessible to both the bundle and Saratoga processes. To send a bundle, the bundle agent can place the complete bundle (the concatenated set of Bundle Protocol blocks) into a file in this shared directory. The local Saratoga instance is then able to _put_ this bundle to peers or allow them to _get_ it. A flag bit in the Saratoga METADATA and DATA packets indicates whether a particular file is a bundle or not. This enables the receiving Saratoga peer to know whether to handle the file itself, or to pass it to the local bundle agent.

When using Saratoga as a convergence layer to transfer bundles, the local bundle agent will either place bundles as files for Saratoga to transfer from its directory, or otherwise use interprocess communication to notify Saratoga of and provide a bundle to be transferred.

Key to the use of Saratoga for bundle transfer are:

- indicating the capability to interoperate with a local bundle agent. This involves advertising the capability to handle bundles via setting Flag bit 10 in Saratoga BEACON packets, and indicating when a bundle is being transferred by setting Flag bits 10 and 11 in the METADATA and DATA packets.

- identifying the Bundle Agent in use, by providing an Endpoint Identifier (EID) in the Saratoga BEACON packet.

Note that the name of a file holding a bundle is actually unimportant, as long as it can be determined that it does hold a bundle. The filename becomes temporary, and local only to the transfer. One implementation strategy is to name each bundle file with a file name constructed from two fields of the Primary Bundle Header: the DTN Endpoint Identifier (EID) of the destination node and the bundle's creation time field. In the rare case of filename collisions in using this scheme, additional octets can be appended to the filename following some arbitrary local scheme. Bundle files might be placed in different directories with different Saratoga-peer access controls depending on the intended next-hop, if this information is known ahead of time. In any case, Saratoga only provides the transfer mechanism, and any forwarding decisions based on routing intelligence would be made within the DTN bundle agents. All of this detail is considered a matter of implementation for the bundle agent, and is not specified here.

The identity field in the Saratoga BEACON packet allows a local DTN bundle agent to advertise its administrative EID via Saratoga. Other Saratoga peers that hear that BEACON can then notify their local DTN bundle agents of the contact. These notifications might be used to integrate contact information into a routing information base, as they are similar to the "hello" packets used in several routing protocols. However, this is outside the scope of this document.

The "epoch" format used in Saratoga timestamps in file object records is the number of seconds since January 1, 2000 in UTC, which is the same epoch used in the DTN Bundle Protocol for timestamps. This must include all leapseconds.

Saratoga instances can work in conjunction with DTN bundle agents to fill the role of a convergence-layer adapter between bundle agents connected via point-to-point links. Saratoga implementations designed to work this way should have a way of notifying bundle agents when they receive BEACONs from other nodes, and when transfers have completed. In order for custody transfer to function properly, notifications between the Saratoga instances and bundle agents on both sides of a fully-successful bundle file transfer are required.

When Saratoga is used as a convergence-layer adapter, it is desirable to turn on Saratoga's end-to-end checksum facilty to provide an indication of correct bundle transfer. This is necessary due to the bundle protocol design not including its own reliability checks for internal robustness. See discussion of this problem with the bundle protocol in [I-D.irtf-dtnrg-bundle-checksum].

4. Proactive and Reactive Fragmentation

Proactive fragmentation - splitting a bundle, before transfer, into multiple separate Saratoga transfers that are reassembled after delivery at the receiving peer - is relatively straightforward. Reactive fragmentation - recovery of disrupted partial transfers - is less so.

For bundle file transfers, the local bundle agent could interact with Saratoga in order to perform a reactive fragmentation of the bundle whose transfer was interrupted by expiration of the inactivity timer. For DTN custody transfer, we expect complications to be encountered in making this reactive fragmentation work properly, and the details required to implement this functionality are left out of this specification until more experience has been obtained with reactive fragmentation in general.

This document does not specify the functionality required for reactive fragmentation of bundles as described in [RFC4838], other than what is needed to support disrupted delivery and hop-by-hop custody transfer of complete bundles. Reactive fragmentation of bundles lies outside the scope of custody transfer of complete bundles, and of this document.

However, the status of a transfer that Saratoga provides to a bundle agent could be used to trigger the reactive fragmentation of bundles if a bundle file transfer is interrupted part-way through (assuming at least the bundle protocol headers and some portion of the data was successfully transferred first). This would allow for efficient recovery when unplanned interruptions occur. This requires some coordination between the Saratoga node and the local bundle agent at each end. The local API or coupling between the Saratoga peer and its bundle agent does not affect the interoperability between either the Saratoga peers or the DTN bundle agents, assuming that both sides agree that fragmentation will occur at the lowest un-acknowledged octet of the bundle file after the disruption. Reactive fragmentation and any forwarding of the fragments onwards for reassembly at some downstream node is solely a bundle-agent problem.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

When Saratoga is also used with a bundle agent, the security and reliability considerations that have been outlined in detail in [I-D.irtf-dtnrg-bundle-checksum] should be borne in mind.

Security in DTNs is in general considered an open issue. If a framework of techniques for handling security in DTN scenarios emerges, Saratoga might be adapted to conform to this.

7. A Note on Naming

Saratoga is named for the USS Saratoga (CV-3), the aircraft carrier sunk at Bikini Atoll and now a popular diving site.

The philosophy behind the protocol and its use described here can be summarized as Saratoga Carries Upper Bundles Adequately, or SCUBA.

8. Informative References

[I-D.irtf-dtnrg-bundle-checksum]
Eddy, W., Wood, L., and W. Ivancic, "Reliability-only Ciphersuites for the Bundle Protocol", draft-irtf-dtnrg-bundle-checksum-09 (work in progress), May 2011.

[I-D.irtf-dtnrg-tcp-clayer]
Demmer, M., Ott, J., and S. Perreault, "Delay Tolerant Networking TCP Convergence Layer Protocol", draft-irtf-dtnrg-tcp-clayer-09 (work in progress), March 2014.

[I-D.irtf-dtnrg-udp-clayer]
Kruse, H. and S. Ostermann, "UDP Convergence Layers for the DTN Bundle and LTP Protocols", draft-irtf-dtnrg-udp-clayer-00 (work in progress), November 2008.

[I-D.kutscher-dtnrg-uni-clayer]
Kutscher, D., "Uni-DTN: A DTN Convergence Layer Protocol for Unidirectional Transport", draft-kutscher-dtnrg-uni-clayer-00 (work in progress), April 2007.

[I-D.wood-tsvwg-saratoga]
Wood, L., McKim, J., Eddy, W., Ivancic, W., and C. Jackson, "Saratoga: A Scalable File Transfer Protocol", draft-wood-tsvwg-saratoga-15 (work in progress) , April 2014.

[IAC-2008]
Wood, L., Ivancic, W., Eddy, W., Stewart, D., Northam, J., Jackson, C., and A. da Silva Curiel, "Use of the Delay-Tolerant Networking Bundle Protocol from Space", conference paper IAC-08-B.2.3.10, 59th International Astronautical Congress, Glasgow , September 2008.

[Ivancic10]
Ivancic, W., Eddy, W., Stewart, D., Wood, L., Northam, J., and C. Jackson, "Experience with delay-tolerant networking from orbit", International Journal of Satellite Communications and Networking, Special Issue on best papers of the Fourth Advanced Satellite Mobile Systems Conference (ASMS 2008), vol. 28, issues 5-6, pp. 335-351, September-December 2010.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, November 2007.

[RFC5325] Burleigh, S., Ramadas, M., and S. Farrell, "Licklider Transmission Protocol - Motivation", RFC 5325, September 2008.

[RFC5327] Farrell, S., Ramadas, M., and S. Burleigh, "Licklider Transmission Protocol - Security Extensions", RFC 5327, September 2008.

[SSTL-2008]
"UK-DMC satellite first to transfer sensor data from space using 'bundle' protocol", press release, Surrey Satellite Technology Ltd , September 2008.

Authors' Addresses

Lloyd Wood
University of Surrey alumni
Sydney, New South Wales
Australia

Email: L.Wood@society.surrey.ac.uk

Jim McKim
RS Information Systems
NASA Glenn Research Center
21000 Brookpark Road, MS 142-1
Cleveland, OH 44135
USA

Phone: +1-216-433-6536
Email: James.H.McKim@grc.nasa.gov

Wesley M. Eddy
MTI Systems
MS 500-ASRC
NASA Glenn Research Center
21000 Brookpark Road
Cleveland, OH 44135
USA

Phone: +1-216-433-6682
Email: weddy@grc.nasa.gov

Will Ivancic
NASA Glenn Research Center
21000 Brookpark Road, MS 54-5
Cleveland, OH 44135
USA

Phone: +1-216-433-3494
Email: William.D.Ivancic@grc.nasa.gov

Chris Jackson
Surrey Satellite Technology Ltd
Tycho House
Surrey Space Centre
20 Stephenson Road
Guildford, Surrey GU2 7YE
United Kingdom

Phone: +44-1483-803-803
Email: C.Jackson@sstl.co.uk