Automated Contact Graph Generation with SPSCGR

Philip Tsao
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109

Abstract

A tool has been developed to generate Contact Graph data structures (that describe predicted channel capacity and connectivity of links between nodes of a Delay/Disruption Tolerant Network) from antenna scheduling information provided by the JPL Service Preparation Subsystem webservice. Capabilities and limitations of the current implementation are presented and prospective future directions are speculated.

Background

In the near future, the JPL Deep Space Network (DSN) faces the challenge of meeting requirements of increasingly ambitious deep space missions in the face of continued fiscal austerity. One of the mechanisms for addressing this challenge lies in transitioning the DSN from a largely "virtual circuit" infrastructure to more of a "packet-oriented" service. Terrestrial communications providers have largely completed this transition by the first decade of the twenty first century. The relative advantages and disadvantages of both paradigms have been discussed at length in the literature (Baran 1962; Tanenbaum 1996).

The protocols employed by terrestrial packet-switched service providers are a poor match for the requirements of the DSN. Terrestrial protocols typically assume propagation times of at most a few hundred milliseconds (on the order of a round trip light-time to and from a geostationary satellite) while the signal propagation times of interplanetary distances several orders of magnitude higher (minutes to hours). Furthermore, spacecraft orbiting other planets may be "eclipsed" from antennas on the surface on the Earth for a significant fraction of their orbital periods requiring that data be "held" until a subsequent communications opportunity.

Delay/Disruption Tolerant Networking (DTN) is a "packet-oriented" technology that has been designed to satisfy the requirements of the DSN (among other applications). Generally speaking, DTN provides a standardized mechanism for overcoming long light time delays and connection outages (Cerf et al. 2007).

DTN implementations fall into two major categories: 1. those optimized for terrestrial sensor networks and highly unreliable network infrastructure and 2. those designed for deep space applications. The Interplanetary Overlay Network (ION) is a JPL-developed DTN implementation specifically optimized for the radiation-hardened microprocessors on board spacecraft destined for deep space (Burleigh et al. 2003).

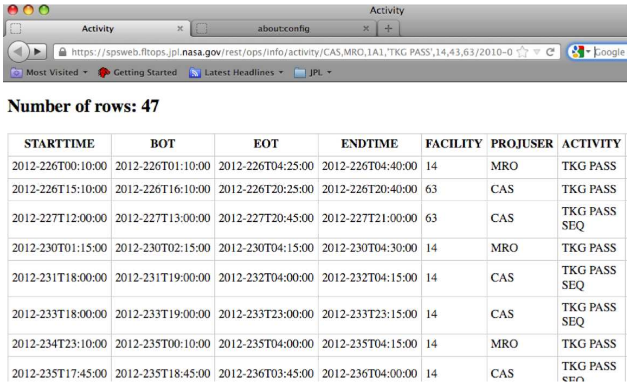
The principal difference (among others) between the two categories of DTN implementations is the mechanism by which traffic is routed. Terrestrial DTN implementations may have little a priori information about the location and connectivity of individual nodes on the network and may rely on aggressive discovery mechanisms and statistical modeling of estimated topology. On the other hand, for deep space applications the trajectories of individual nodes are very well known and future connectivity and link latency can be estimated very accurately many months in advance (these estimates would have to be recalculated after trajectory correction maneuvers, unscheduled antenna maintenance or other unexpected events).

ION uses a data structure known as a *Contact Graph* to represent present and future node availability, capacity and latency (from the point of view of ION, a "node" may be a spacecraft or an antenna on the ground). The contact graph is replicated to all nodes on the DTN network so that proper routing decisions may be made at each hop and is updated when the assumptions underlying forecasts change. Prior to the work that led to this article, contact graphs were usually generated by hand (not too bothersome for simplistic scenarios but error prone for large scale modeling). The following sections describe an automated mechanism for generating a contact graph from an existing DSN scheduling database suitable for modeling and simulation purposes.

Implementation

SPSCGR parses HTML output from the DSN Service Preparation Subsystem (SPS) web service. SPS is a web service (see Figure 1) that permits end users to obtain schedules for each DSN antenna and every mission that uses DSN antennas. From this information it is possible to construct a syntactically correct contact graph (an example is in Figure 2) for use by one or more nodes running the ION DTN implementation.

Note that the parameters currently extracted from the SPS portal are necessary but insufficient for a complete contact



Number of rows: 47

STARTTIME	BOT	EOT	ENDTIME	FACILITY	PROJUSER	ACTIVITY
2012-226T00:10:00	2012-226T01:10:00	2012-226T04:25:00	2012-226T04:40:00	14	MRO	TKG PASS
2012-226T15:10:00	2012-226T16:10:00	2012-226T20:25:00	2012-226T20:40:00	63	CAS	TKG PASS
2012-227T12:00:00	2012-227T13:00:00	2012-227T20:45:00	2012-227T21:00:00	63	CAS	TKG PASS SEQ
2012-230T01:15:00	2012-230T02:15:00	2012-230T04:15:00	2012-230T04:30:00	14	MRO	TKG PASS
2012-231T18:00:00	2012-231T19:00:00	2012-232T04:00:00	2012-232T04:15:00	14	CAS	TKG PASS SEQ
2012-233T18:00:00	2012-233T19:00:00	2012-233T23:00:00	2012-233T23:15:00	14	CAS	TKG PASS SEQ
2012-234T23:10:00	2012-235T00:10:00	2012-235T04:00:00	2012-235T04:15:00	14	MRO	TKG PASS
2012-235T17:45:00	2012-235T18:45:00	2012-236T03:45:00	2012-236T04:00:00	14	CAS	TKG PASS SEQ

Figure 1: Web browser rendering of SPS HTML output.

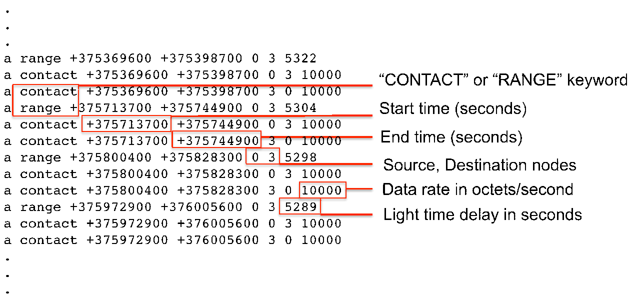


Figure 2: Excerpt of contact graph generated from SPS data depicted in Figure 1.

graph. For example, parameters such as track start and end times do not necessarily bracket the communications windows available for DTN bundle transmission. Furthermore the duplex and data rates of the links are not yet available from SPS and must be obtained from each individual mission. DSN currently has no knowledge of the configuration of the spacecraft (radio mode, antenna pointing, etc.) and therefore cannot publish this information in SPS. The current version of SPSCGR does not estimate data rates from link budgets. Duplex defaults to "bidirectional" and data rate defaults to 100 kilobits per second.

The SPSCGR object code links against the JPL-developed SPICE toolkit to estimate light time delays and confirm "visibility" (some tracks may include intervals in which the spacecraft isn't visible for the purposes of ground testing and troubleshooting). SPICE is a library that can compute solutions to space geometry problems and is often used for purposes such as mission modeling and frequency predict generation (Acton Jr 1996).

For best results, SPICE should be periodically updated with the latest mission ephemeris kernels available at `ftp://naif.jpl.nasa.gov` to keep up with the latest spacecraft trajectory maneuvers and measurements. Other information such as Earth orientation and leap second data may also be regularly updated but is less important as the LTP [1] implementation in ION 3.x is less sensitive to these parameters.

Because of an unfortunate lack of standardization, a "translation-table" between SPS and SPICE names is required (`14` maps to `dss-14` and `CAS` maps to `cassini`). A preliminary translation table has been constructed but is not yet complete.

SPSCGR's running time is currently dominated by routines in the SPICE toolkit. The HTML file from SPS is parsed twice, once to assign node numbers and once to assemble the actual contact graph (this could have been done in one pass but the author chose to keep the current design simple at the expense of a modest performance hit). SPSCGR has been tested on HTML files representing actual DSN schedules for up to three antennas for up to 24 hours and the current implementation should have no problem scaling up to days or weeks. Should scalability become a concern, there is room for optimization.

SPSCGR was derived from a previously developed prototype that generated an approximate contact graph from information provied by the MaROS (Mars Relay Operations Service) system. MaROS is analogous to SPS for the rovers and spacecraft on the surface of and in orbit around Mars. Missions use MaROS to schedule communication relay opportunities between surface assets (rovers and landers) and ground station on Earth via one or more Mars orbiters (Allard 2010).

SPSCGR was written in C and therefore should be fairly portable. It has been tested on Linux, Microsoft Windows (cygwin) and Mac OS X. The only dependencies are the C runtime (provided by the operating system) and SPICE libraries (freely downloadable from JPL).

Summary and Concluding Remarks

An automated tool for generating contact graphs for the ION DTN implementation was presented. SPSCGR in its current form may be useful for the purpose of modeling and simulating a fully DTN-enabled Deep Space Network. Future versions may be used to generate actual contact graphs for routing of mission data.

It should be noted that contact graphs may be useful to the DSN even for non-DTN purposes. The information encoded within a contact graph may be instrumental for efficiently provisioning resources in communications networks subject to Quality of Service constraints. In fact, there is no reason why a contact graph can't be used to better schedule resources for the current "virtual-circuit" implementation of the DSN.

Acknowledgments

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology and was funded by grants from the National Aeronautics and Space Administration. The author would like to thank Daniel Allard, Loren Clare, Esther Jennings and Shan Malhotra for their assistance and insight.

[1] Licklider Transmission Protocol is used by ION to carry DTN traffic across Solar System distances (Ramadas, Burleigh, and Farrell 2008).

References

Acton Jr, C. H. 1996. Ancillary data services of NASA's navigation and ancillary information facility. *Planetary and Space Science* 44(1):65–70.

Allard, D. A. 2010. MaROS strategic relay planning and coordination interfaces. In *Aerospace Conference, 2010 IEEE*, 1–11. IEEE.

Baran, P. 1962. *On Distributed Communication Networks*. Rand Corporation.

Burleigh, S.; Hooke, A.; Torgerson, L.; Fall, K.; Cerf, V.; Durst, B.; Scott, K.; and Weiss, H. 2003. Delay-tolerant networking: an approach to interplanetary internet. *Communications Magazine, IEEE* 41(6):128–136.

Cerf, V.; Burleigh, S.; Hooke, A.; Torgerson, L.; Durst, R.; Scott, K.; Fall, K.; and Weiss, H. 2007. Delay-Tolerant Networking Architecture. RFC 4838 (Informational).

Ramadas, M.; Burleigh, S.; and Farrell, S. 2008. Licklider Transmission Protocol Specification. RFC 5326 (Experimental).

Tanenbaum, A. S. 1996. Computer networks. *Prentice Hall* 263–264.