Use of the Delay-Tolerant Networking Bundle Protocol from space

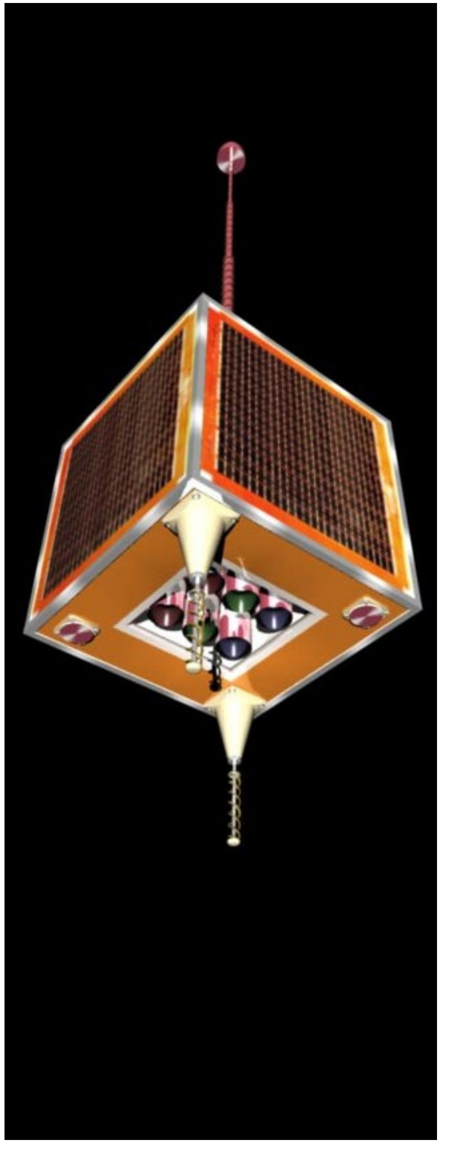
Cisco Systems,
NASA Glenn Research Center,
Surrey Satellite Technology Ltd.

Alex da Silva Curiel
30 September 2008

59th International Astronautical Congress, Glasgow
B2-3 Near-Earth and Interplanetary Communications Systems

Summary

- UK-DMC satellite launched with other DMC satellites into low Earth orbit, September 2003.
- All DMC satellites use the Internet Protocol (IP). IP used daily operationally for satellite and payload communication and control.
- CLEO Cisco router on UK-DMC tested by NASA/Cisco/SSTL team: IPv4 with mobile networking (2004), IPSec and IPv6 (2007).
- SSTL developed high-speed file transfer protocol, *Saratoga*, for moving imagery to ground. Runs on SSTL's imaging computers.
- Team now testing sending delay-tolerant networking (DTN) 'bundles' over *Saratoga*. Looking at bundle reliability and fragmentation. Tests in January and August 2008.



Overview

- The Disaster Monitoring Constellation.
- The network environment.
- *Saratoga* and its design choices – performance.
- The 'bundle' approach to delay-tolerant networking.
- Bundling over *Saratoga*. Tests and results.
- Experience with problems in bundling's design:
  - Reliability.
  - Reliance on a correct clock time.
  - Fragmentation.
- Alternatives to bundling?

Images shared by other organisations are used with thanks.

Disaster Monitoring Constellation (DMC)

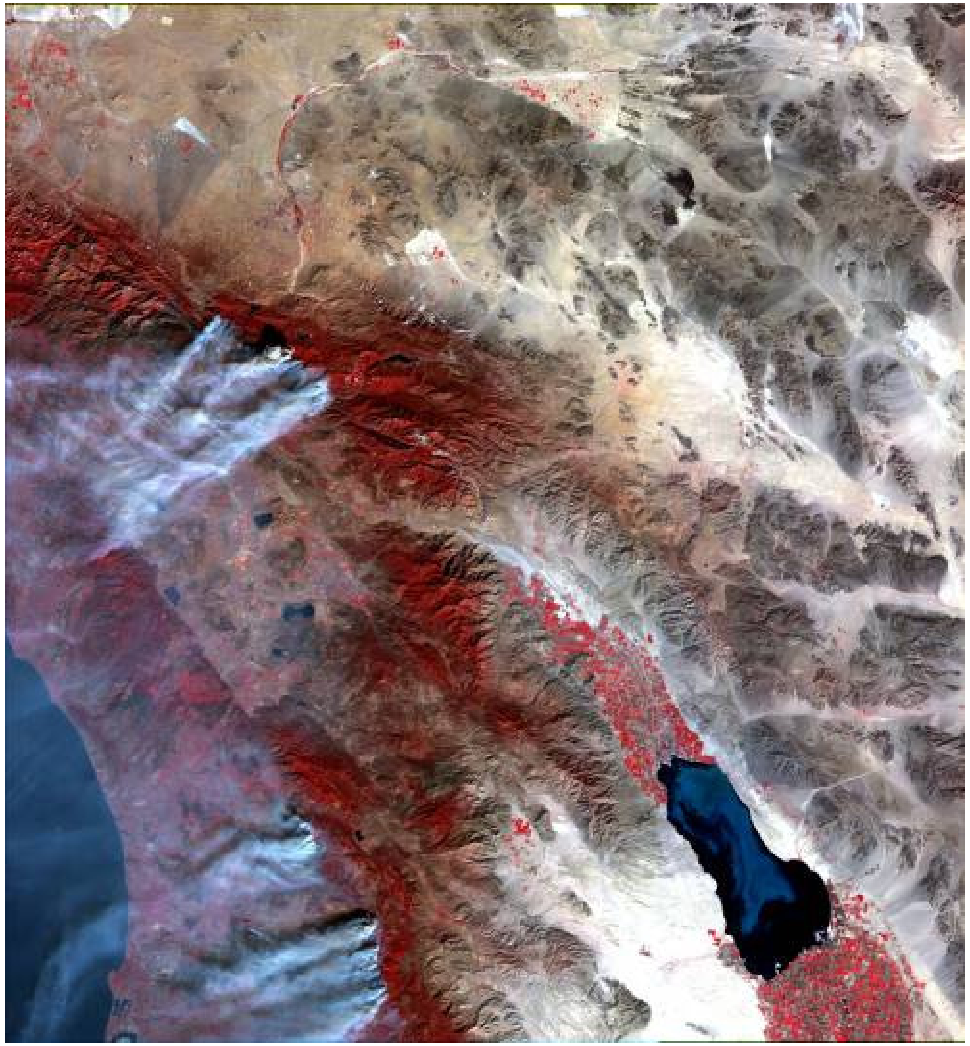
www.dmcii.com

Surrey Satellite Technology Ltd (SSTL) build and help operate an international constellation of small sensor satellites.

The satellites share a sun-synchronous orbital plane for rapid daily large-area imaging (640km swath width with 32m resolution). Can observe effects of natural disasters.

Government co-operation: Algeria, Nigeria, United Kingdom, Turkey and China.

Each government finances a ground station in its country and a satellite. Ground stations are networked together. Further satellites expected.



fires in California, 28 October 2003 (UK-DMC)

DMC satellite constellation launches

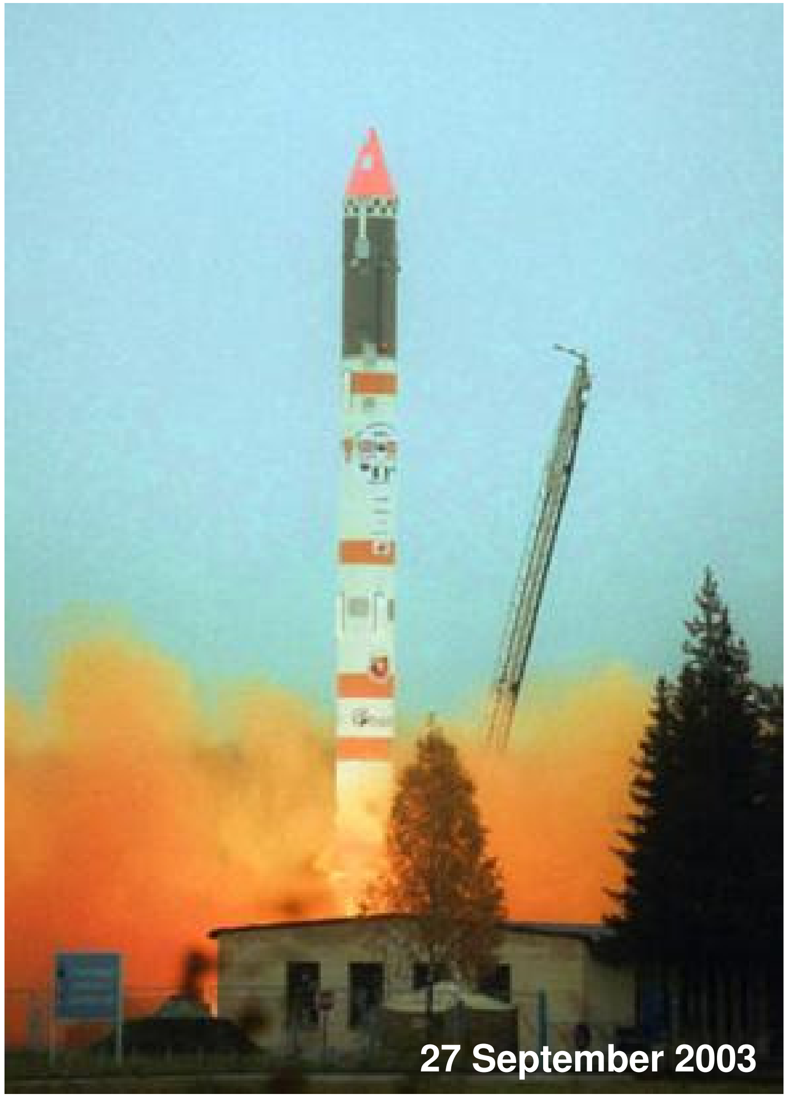
Five satellites launched so far. Similar base designs and subsystems, with custom modifications for each country.

Satellites launched from Plesetsk in Siberia on affordable shared Russian Kosmos-3M launches:

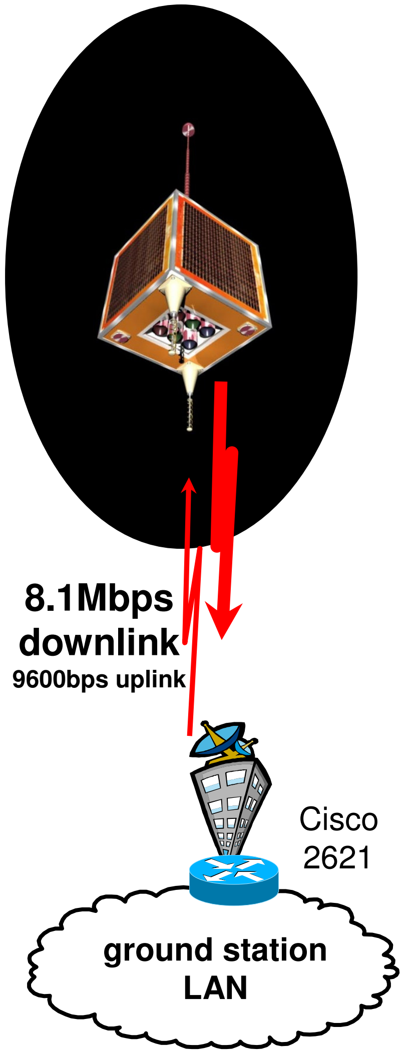
November 2002: AlSAT-1 (Algeria)
September 2003: UK-DMC, NigeriaSAT-1 and BilSat (Turkey)
October 2005: Beijing-1 (China)

Satellites and ground stations in each country use Internet Protocol (IP) to communicate. Earth images delivered to ground stations via UDP-based file transfer.

SSTL migrated from AX.25, as used on previous missions. First used CCSDS CFDP for image file transfers, but replaced CFDP with *Saratoga* to increase overall throughput.



Existing network environment for the DMC



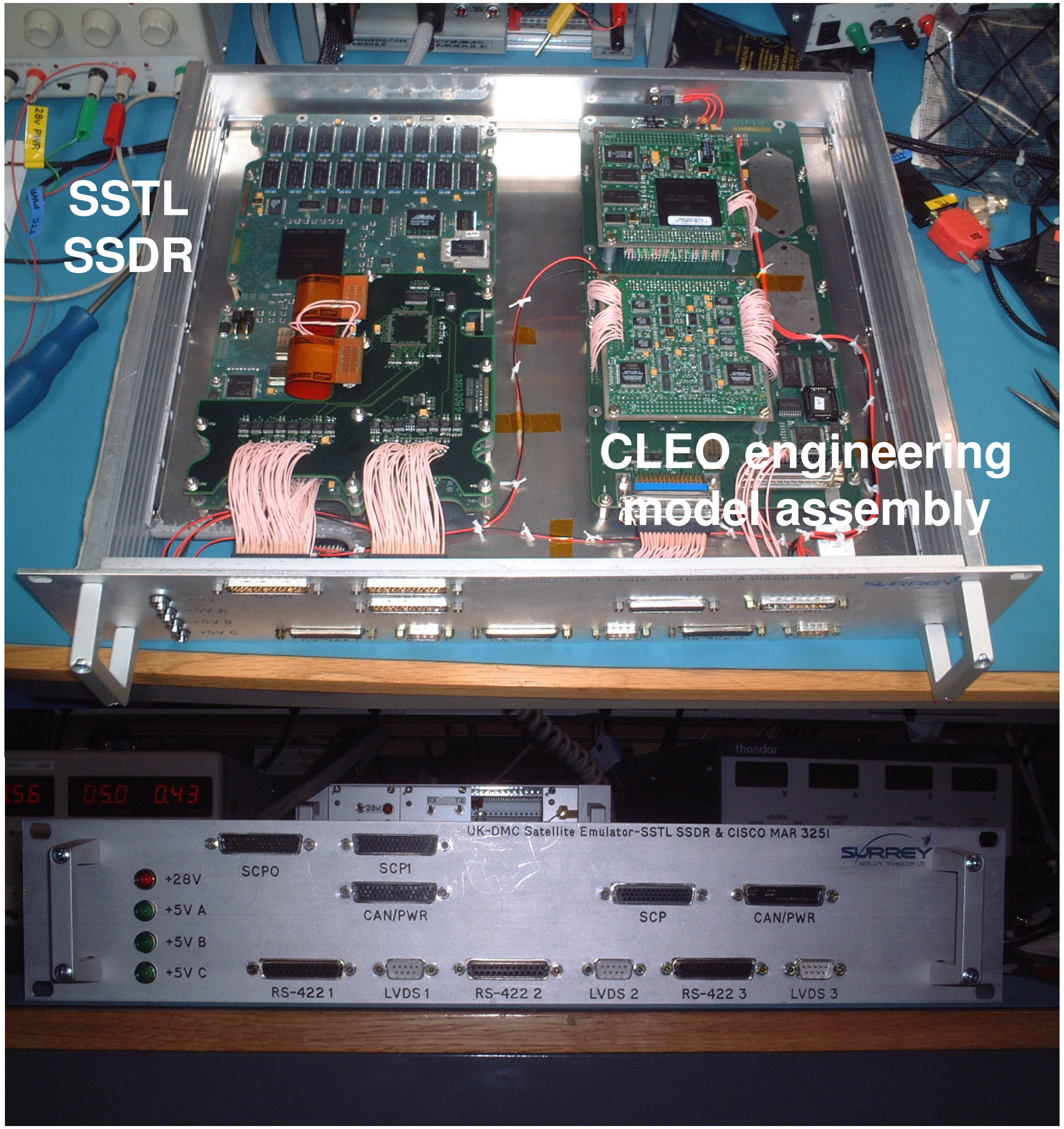
Satellites: each DMC satellite has multiple onboard computers. For housekeeping (On Board Computer, OBC), for image capture and packetised transmission (Solid State Data Recorders, SSDRs), for redundancy and survival. Interconnected by IP over 8.1Mbps serial links for data and slower CANbus for backup control; really a custom-built LAN. (CLEO router on UK-DMC only.)

S-band links: Very asymmetric design. Ten-minute passes over ground stations. *Saratoga* on 8.1Mbps downlink (faster on later DMCs) delivers imagery. Slow 9600bps uplink just for commands and reliable acks of image data. TCP is unfit for this environment – single scheduled *Saratoga* flow at any time (with low-rate telemetry stream multiplexed in), no competition.

Ground: SSTL's design for its ground stations' LANs uses IP. IP over 8.1 Mbps serial stream from downlink commercial modem goes into a rack-mounted Cisco 2621 router, which forwards IP packets onto the LAN. SSTL's ground station LAN is connected to and an integral part of SSTL's corporate IP network. Firewalled Internet access.

CLEO testbed creates code for space use

- Ground-based testbed loaned to NASA Glenn was key to initial success of testing CLEO router, and later IPv6 testing.
- Now used for software development on SSTL's Solid-State Data Recorder (SSDR) computer: RTEMS talking *Saratoga*.
- Code is written and tested here before uploading for on-orbit tests.

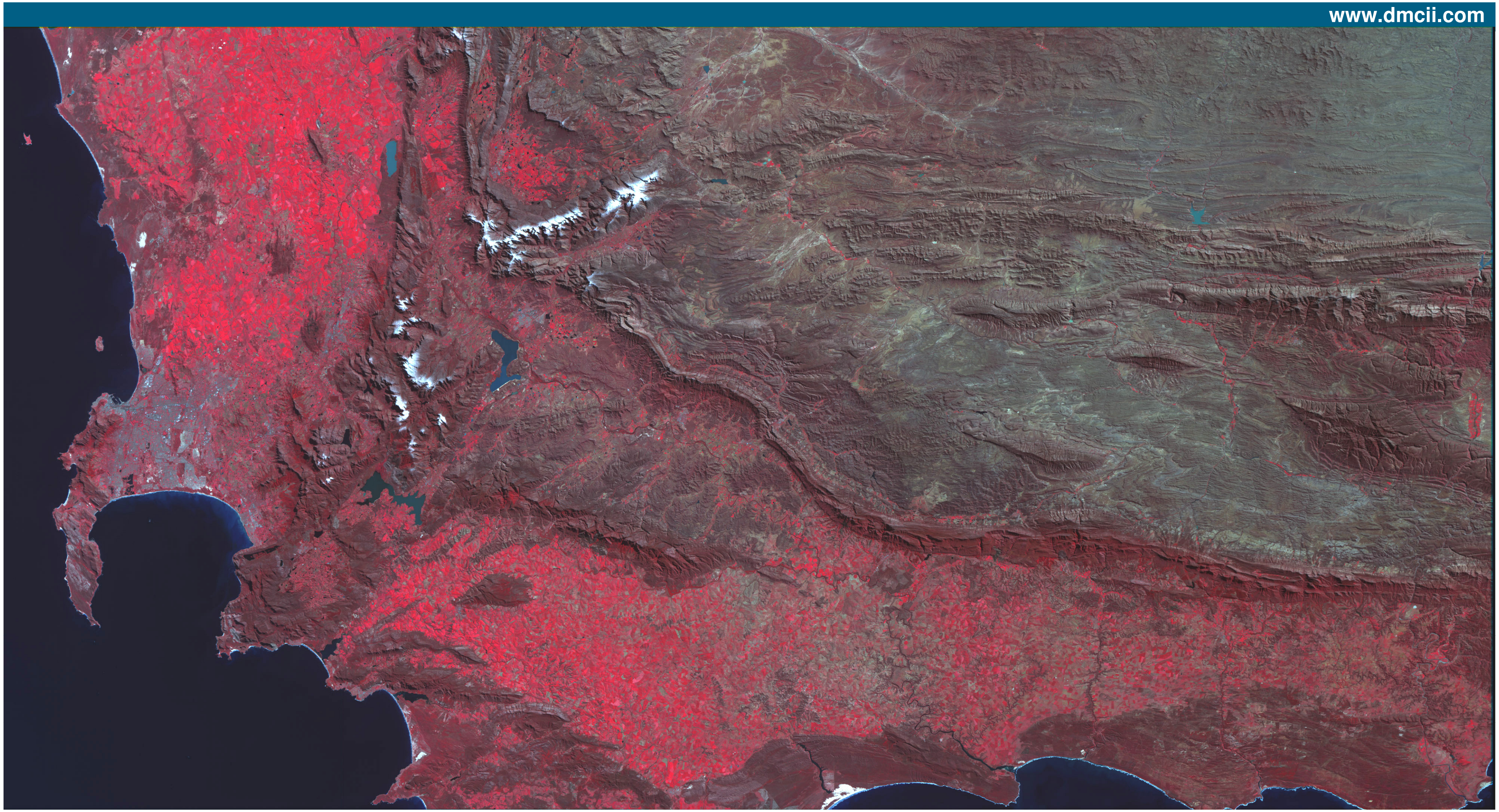


Saratoga and its design choices

- Performance across private links – a single flow can run at line speed, sending packets back-to-back. Link capacity is not wasted.
- Copes with high link asymmetry (>850:1) with selective negative acknowledgements.
- Provides file metadata for flexibility; allows push/get file transfers, directory browsing.
- Simple, clear UDP-based design. Internet-drafts document *Saratoga* (IETF tsvwg) and optionally carrying bundles with *Saratoga* (IRTF dtnrg).

Delay-tolerant networking (DTN)

- DTN began intended as 'Interplanetary Internet' for deep-space connectivity, but is now also used for opportunistic ad-hoc networks.
- Data is moved like store-and-forward email messages in 'bundles' between nodes, when limited connectivity becomes available and links are up.
- NASA Glenn has ported DTN bundling code to SSTL's onboard computers, using CLEO testbed.
- Many 'convergence' (transport) layers for bundling – UDP more useful here than TCP; SSTL's custom *Saratoga*/UDP is a simple, high-performing choice.
- Bundles downloaded from UK-DMC satellite to NASA computer in SSTL ground station. Forwarded via bundle over TCP to NASA Glenn. January and August 2008.



The Cape of Good Hope and False Bay. False colours – red is vegetation. Taken by UK-DMC satellite at 08:27 UTC, Wednesday, 27 August 2008.

Downloaded using bundling over *Saratoga*, with proactive fragmentation. Fragments assembled at NASA Glenn, then postprocessed at SSTL.

First sensor imagery delivered by bundles from space.

Why run bundling over *Saratoga*?

- A lot of research effort on bundling; IETF DTNRG community views it as *the* chosen way to handle delay-tolerant and disrupted networks.
- A LEO satellite passing over a ground station has disrupted connectivity. Seems a natural fit with bundling, which should handle disruption.
- Bundling is one way to split end-to-end path and set up separate control loops to increase performance.
- Evaluating bundling for space use; NASA is also considering bundling for its deep-space missions (possibility of experimenting with bundling and CFDP on *Deep Impact* comet mission.)

Value of testing bundling over *Saratoga*

- First use of bundles for sensor data from space.
- Demonstrated problems with lack of reliability checks in bundles. We have implemented an MD5 checksum in *Saratoga* to help compensate.
- Demonstrated problems with keeping machines in good clock sync so that bundles with misset times aren't expired and dropped.
- Demonstrated working proactive fragmentation.
- Problems we encountered suggest that bundling design is not ready for operational deployment.
- *Saratoga* meets SSTL's operational needs by itself.

Known problems with bundling

- IRTF DTN Research Group adopted bundling as ‘universal’ way to deal with DTNs. Not investigating alternatives.
- Bundling is not mature or ready for *mission-critical* use.
- Bundle design ignores *end-to-end principle*. **No built-in reliability checks.** Security protocols are emphasised.
- We are now retrofitting reliability as an add-on to security.
- Assumes all nodes know current UTC time and are up-to-date with leap seconds. (This matches NASA JPL deep-space network use.) Bundles that are too old are discarded. Can’t learn the time using bundling. Needs regular updates for new leap seconds to avoid gradual clock drift from sync.
- Reliability, reactive fragmentation and delivery of fragments across disparate hops, routing across bundle networks, key management, etc. are still open problems.
- **Bundling is really still a research effort.**

An alternative to bundling: HTTP*-DTN*

- MIME describes the things we move around the network.
- The most successful protocols support MIME.
- HTTP is the simplest MIME wrapper.
- HTTP provides infinitely-flexible text metadata.
- Uses HTTP hop-by-hop between neighbouring DTN nodes. No proxying, no intercepting. Proxy cache model is not relevant here.
- Allow HTTP to be run over different transports: TCP, SCTP, *Saratoga*… HTTP can be separated from TCP's limitations.
- Divide HTTP from transport to make a true session layer. What HTTP requires from transport isn't that onerous.
- **HTTP has what bundling doesn't: content identification (via MIME), reliability, well-understood security, fragmentation.**
- Described in an internet-draft. Worth further investigation.

further information:

http://www.ee.surrey.ac.uk/Personal/L.Wood/dtn/

or just google ***Saratoga*** **UK-DMC**

Questions?

thankyou