Saratoga

a bundle convergence layer

draft-wood-dtnrg-saratoga-01.txt

Wood, Eddy, McKim, Ivancic, Jackson
Cisco Systems, NASA Glenn, SSTL.

Lloyd Wood
Cisco Systems

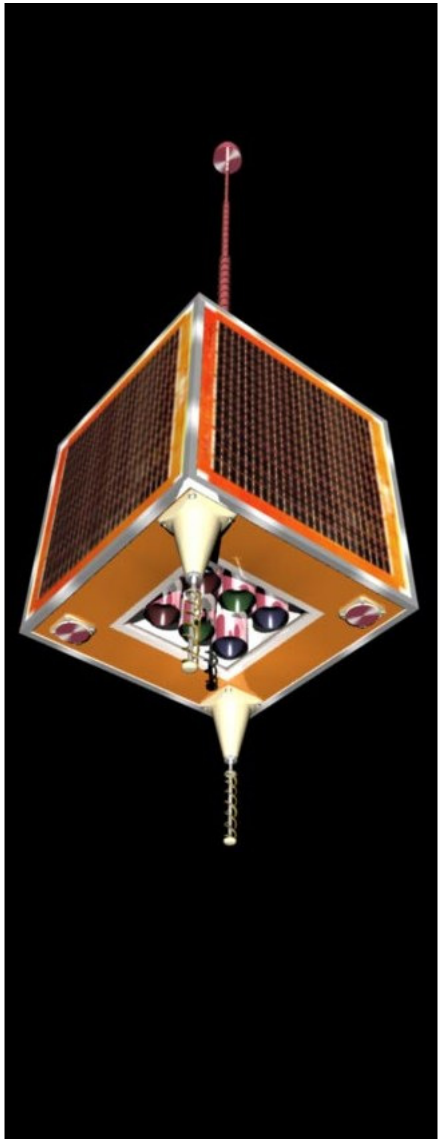
Delay-Tolerant Networking session
IETF 69, Chicago, July 2007

Changes from *Saratoga* -00 draft

- Added support for delivering errored content with UDP-Lite.
- Added explicit ‘bundle’ flag, so use of/need for processing with a bundle agent can be indicated.
- Added streaming support.
- Added timestamp support, which is useful for streaming and useful for measurements for sender rate-control algorithms *a la* TCP’s timestamps.
- Uses link-local multicast rather than broadcast for BEACONs. Also permits multicast delivery to multiple receivers. (Addresses have been requested from IANA.)
- Tidied packet formats for alignment and to allow more space for flags.

Short summary

- *Saratoga* is a simple file transfer protocol that can also be used to transfer DTN bundles.
- Developed and in use by Surrey Satellite Technology Ltd (SSTL) to transfer remote-sensing imagery from IP-based LEO satellites.
- NASA Glenn has cleaned up the *Saratoga* design to create a new version of *Saratoga* for file or bundle transfers. Described in **draft-wood-dtnrg-saratoga-01.txt**.
- We already have multiple implementations (in Perl, Python, and C, on Linux and RTEMS).
- Using the testbed first used for Cisco's CLEO router in orbit, we are preparing to fly the RTEMS code on the UK-DMC satellite.



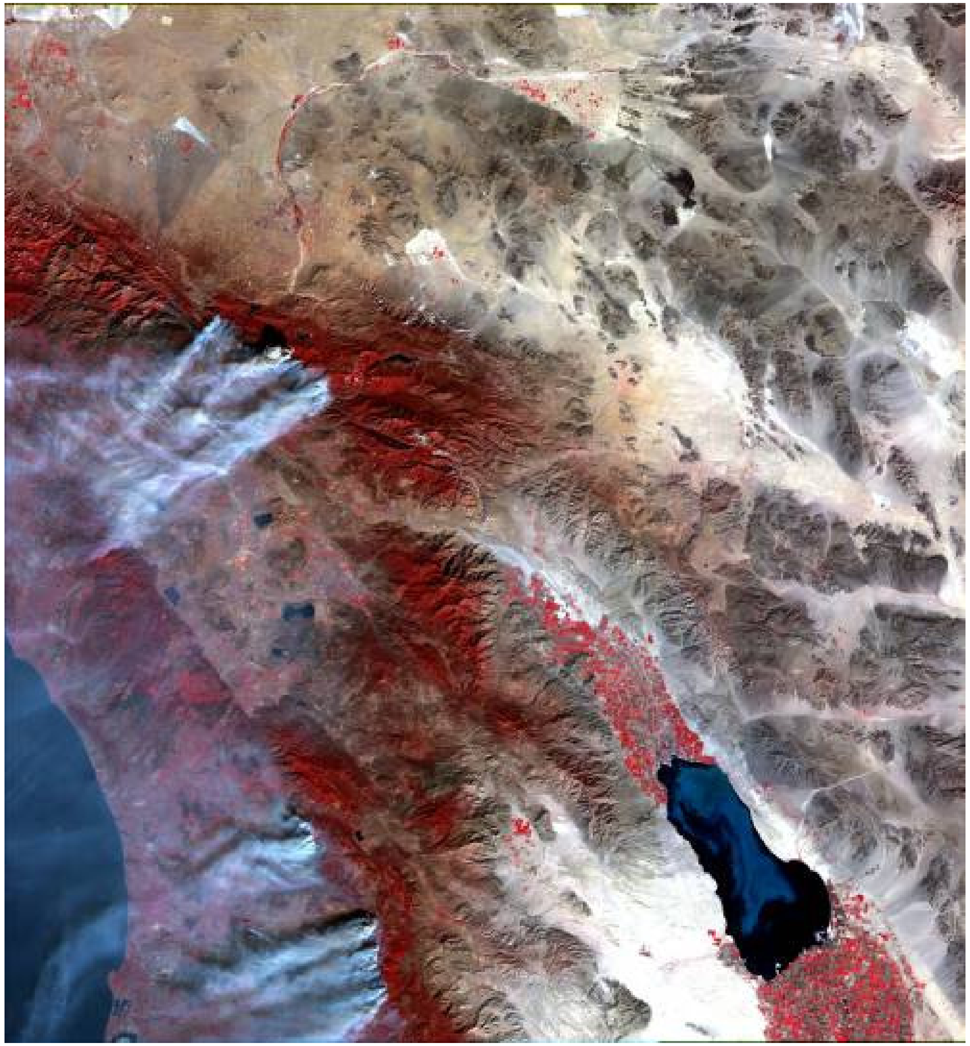
Disaster Monitoring Constellation (DMC)

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. Imaged the effects of Hurricane Katrina and the Indian Ocean Tsunami.

Government co-operation: Algeria, Nigeria, Turkey, United Kingdom, and China. Each government finances a ground station in its country and a satellite. Ground stations are networked together. Three more satellites have been announced and are being built.

www.dmcii.com



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

DMC in use: after Hurricane Katrina, 2005

In this false-color image, dry land is red. Flooded and damaged land is shown as brown.

Small part of an image taken by the Nigerian DMC satellite on Friday 2 September, for the US Geological Survey.

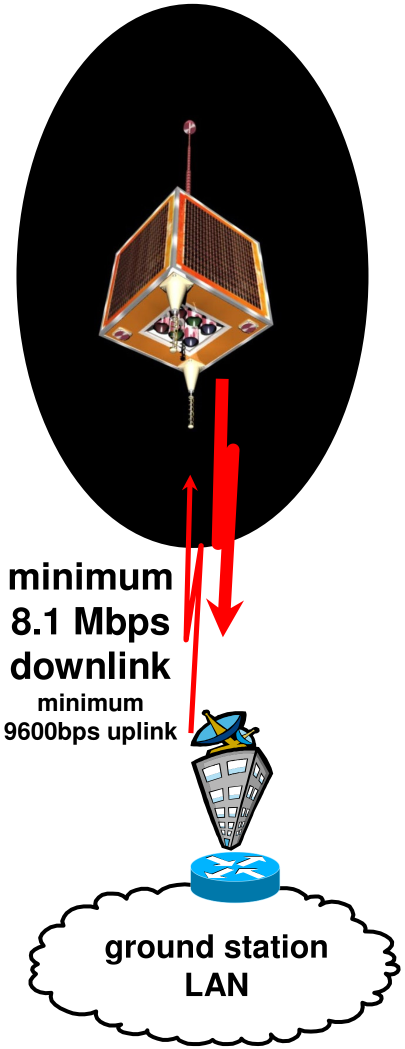
DMC is working as part of the United Nations International Charter for Space and Major Disasters.

Imagery delivered by using Saratoga over UDP.

Saratoga is in daily operational use.

www.dmcii.com

How is *Saratoga* used in operations?



Each DMC satellite has multiple onboard computers. For housekeeping (the On Board Computer, OBC), for image capture and packetised transmission (the 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. Each satellite is a custom-built local area network (LAN).

Newer satellites also have 20/40 Mbps X-band downlinks for added hi-res cameras; faster downlinks (100+ Mbps) are planned for future missions. Uplink is only 9600bps for command and control. Uplink speeds are also likely to increase… to 38400 bps. *Very* asymmetric; 850:1 or worse downlink/uplink ratio.

As much data as possible must be transferred during a pass over a ground station. Passes may be up to twelve minutes, depending on elevation. At 8Mbps, that's approximately 650MB of useful data (about a CD-ROM's worth) that can be transferred in a high pass – if you fill the downlink with back-to-back packets at line rate. Link utilization *really matters*. SSDRs take scheduled turns filling link.

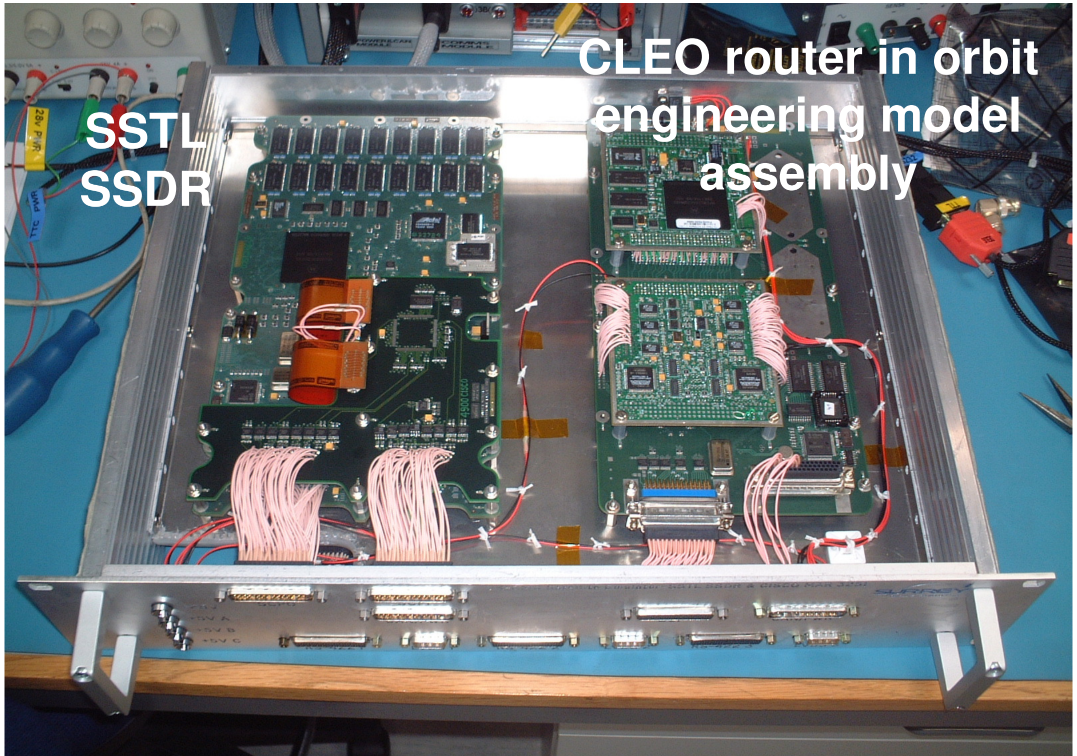
Ground-based testbed for development

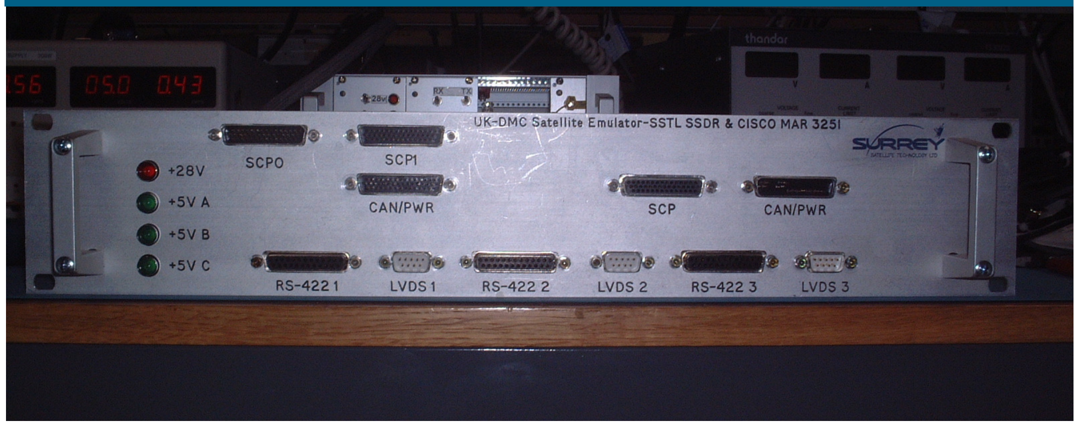
NASA Glenn needed to gain familiarity with operating and configuring SSTL's onboard computers.

Ground-based testbed allowed configuration changes to be tested on the ground at leisure before being made to CLEO router in orbit or SSDRs during a ten-minute pass over a ground station.

Built rack-mounted ground-based testbed ('flatsat') containing SSDR and networked it from NASA Glenn in Ohio, so NASA could get familiar with SSDR design and use.

Now using testbed in development role for flying *Saratoga* and DTN bundle code on UK-DMC satellite.





Why are we pursuing DTN with DMC?

- **We believe IP is useful for operational use of DTN** – not *just* convenient/cheap for prototyping DTN code. (Being convenient/cheap are compelling reasons to use IP for DTN. IP runs over many links already. Implementing support for LTP or convergence layers direct over all these links isn't scalable.)
- Because the DMC is an example of using IP both on the ground and in space, with the ground station acting as a gateway between types of use.
- Assumptions governing IP use (link use, shared contention *vs* dedicated scheduling models) differ between ground/space, but the protocol used remains the same. DMC can be seen as a prototypical DTN scenario, with long disruptions between passes over ground stations.

Transport protocol matrix – where this fits

Characteristic / Reliability factor	always congestion controlled	can be uncontrolled to fill dedicated links
permits delivery of errored content	DCCP (still uses checksum across headers for reliability)	*Saratoga* (reliable headers) UDP-Lite (reliable headers) LTP? (**but** unreliable headers)
unguaranteed packet delivery	DCCP SCTP (with 'partial reliability' support)	*Saratoga* (streaming/no acks) UDP/UDP-Lite LTP (green packets, unacked)
error-rejecting guaranteed packet delivery	SCTP TCP	*Saratoga* LTP (red **only** when using security/NULL authentication)

Reliability must include error detection!

- Saratoga always uses the UDP checksum to cover header and payload. This is consistent but not that strong (one's-complement), and not end-to-end. An end-to-end MD5 checksum over the file/bundle compensates and increases confidence that a reliable copy has been made. A strong link-layer checksum is optional.
- UDP-Lite checksum covers a *minimum* of IP/UDP-Lite/Saratoga headers, so there's always some checking of header content.
- LTP doesn't include any checksums, and needs to use NULL authentication extension or full security framework. Without any security, LTP relies solely on error-checking of link layer.
- Bundle protocol lacks end-to-end reliability, too. So we've written a new draft proposing a block checksum – but that won't cover the entire bundle format.
- Reliability is discussed in Stone, Saltzer and our new paper: *Checksum Coverage and Delivery of Errored Content*.
- **This needs much more discussion.**

Basic *Saratoga* design

- Flood data packets out as fast as you can. No specified congestion control, since you're only going one hop. (No timers, so good for long delays.) Any multiplexing of flows is done by the *Saratoga* peer.
- Every so often, ask for an acknowledgement from the file receiver. Receiver can also send acks if it thinks it needs to, or to start/restart/finish transfer.
- Acks are Selective Negative Acknowledgements (SNACKs) indicating left edge and any gaps to fill with resent data (and with enough information so that intelligent sender rate control can be added).
- That's it. But just how big is a file/bundle?

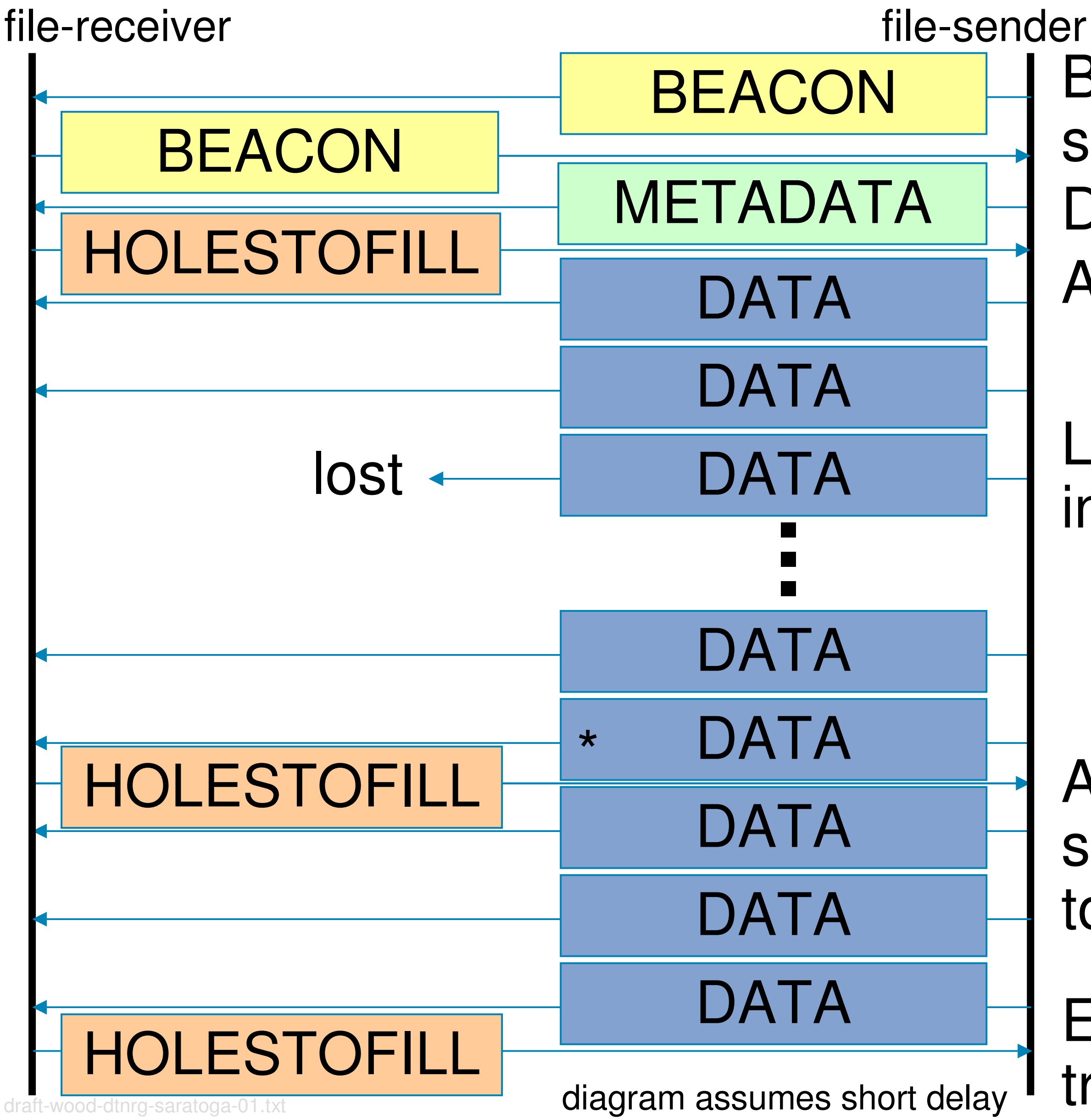
Filesizes can be *large*

- For the DMC, imaging files are big – typically up to a few gigabytes at 32m resolution; larger for newer cameras. So we think bundles will also be *large* – gigabytes and up.
- But ad-hoc/sensor nets also need to transfer small files/bundles; guessing a range limits use.
- So we allow a range of file-descriptor pointers to be advertised: 16/32/64/128-bit file descriptors.
- If file is less than 64KiB, use 16-bit offsets. If file is larger but less than 4GiB, use 32-bit offsets…
- 16-bit is *always* supported. Others are optional. Draft diagrams are 32-bit, which fits 80 columns.

Saratoga packets

BEACON	Sent periodically. Describes the *Saratoga* peer: Identity (e.g. EID) capability/desire to send/receive packets. max. file descriptor handled (16/32/64/128-bit).
REQUEST	Asks for a file via ‘get’, directory listings, deletes.
METADATA	Sent at start of transaction. Describes the file/bundle: identity for transaction file name/details, including size. descriptor size to be used for this file (one of 16/32/64/128-bit pointer sizes.)
DATA	Uses descriptor of chosen size to indicate offset for data segment. May request an ack.
HOLESTOFILL	Ack. Can use the descriptor size to indicate offsets for missing ‘holes’ in data.

Saratoga transactions: 'put'



Beacons (optional); show filesize capabilities.

Describes chosen file.

Ack indicates reception.

Lost data creates hole in copy at receiver.

Ack requested and sent describing hole to be filled.

Empty ack indicates transaction is complete.

Saratoga transactions: ‘get’

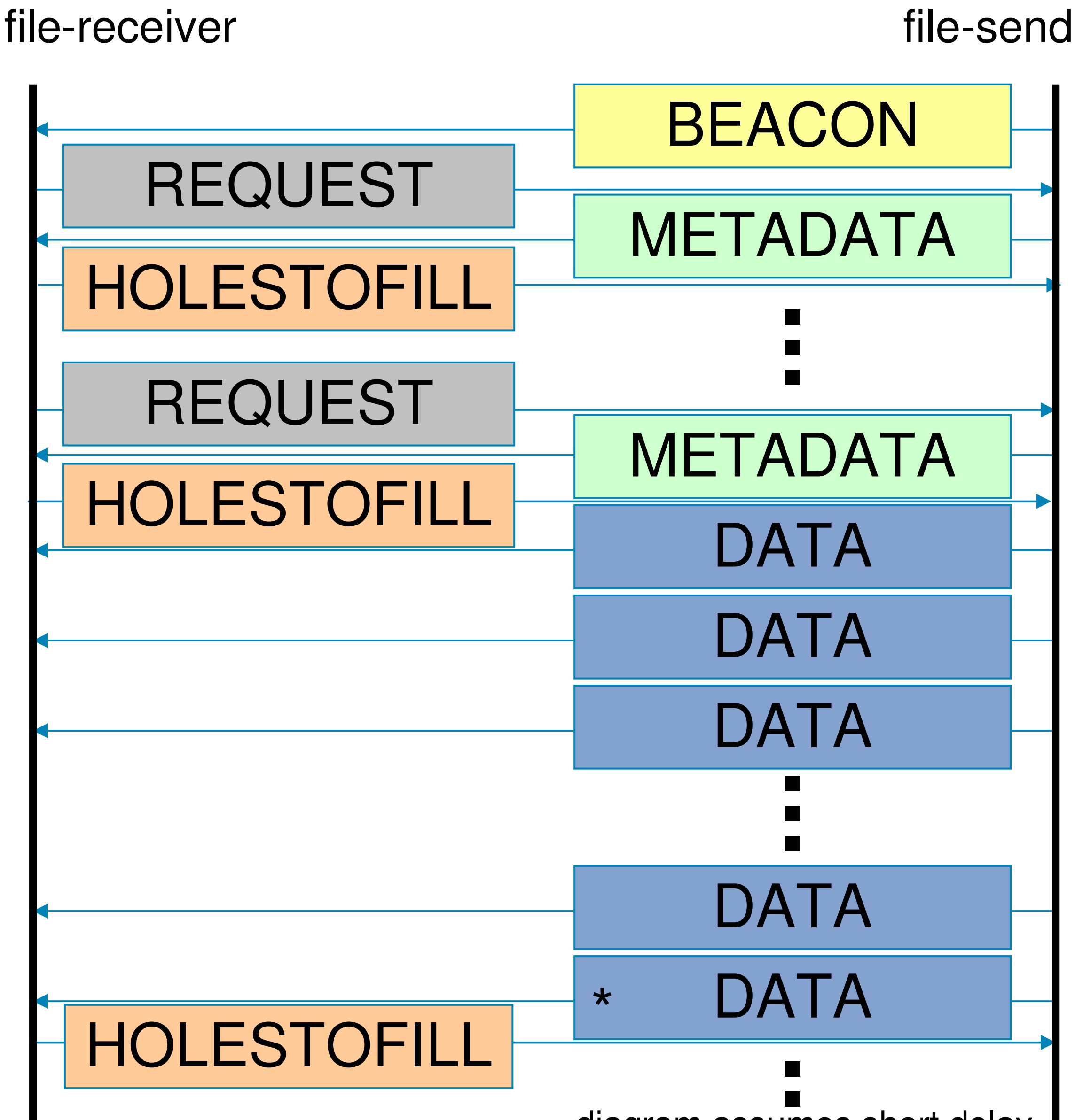


diagram assumes short delay

Beacon heard (optional).
‘getdir’ can request file list.

File list sent as file…
HOLETOFILL/DATA transaction omitted

‘get’ requests a file.
File is described.
METADATA is acked.
File data is streamed out directly after METADATA, without waiting for ack.

Ack requested and sent. Sender continues to send DATA.

Why *Saratoga* instead of TCP?

- Link utilization and throughput on dedicated links.
- Assumptions about loss/congestion/competition.
- Able to cope with high link asymmetry (>850:1).
- Simplicity. TCP is really for a conversation between two hosts; needs a lot of code on top to make it transfer files. We're just interested in moving files; makes *e.g.* sequence nos. simpler.
- Long delay use – eventually TCP will fail to open a connection because its SYN/ACK exchange won't complete. TCP has many unwanted timers.

Licklider LTP and *Saratoga* – comparison

Feature	*Licklider LTP*	*Saratoga*
large object transfers	yes (SDNV)	yes (descriptors)
works under high latency	yes	yes
robust checksummed format	no (add authentication)	**yes**
header integrity checks	no (left to link layer)	**yes**, always
end-to-end integrity checks	no (add bundle security)	**yes** (can use MD5)
supports delivery of errored data	yes ('doesn't happen')	**yes** (UDP-Lite)
includes object metadata	no (left to bundle)	**yes** (optional)
directory listings for file selection	no	**yes** (optional)
supports 'push' transfers	yes	**yes**
supports 'pull' transfer requests	no	**yes**
beacons for discovery and automated transfers	no	**yes** (optional)
multicast to many receivers	no	**yes** (optional)
handles asymmetry	yes	**yes, very well**

Future plans

- We plan to fly RTEMS *Saratoga* code with bundle support on the UK-DMC satellite later this year.
- First bundles in space?

> Certainly, if the equipment were already developed, **reliable**, and available, it would be used.
>
> - J. C. R. Licklider, *Man-Computer Symbiosis*, 1960.

draft-wood-dtnrg-saratoga-01.txt

Questions?

thankyou

with thanks to

Will Ivancic, Wes Eddy, Jim McKim

and Chris Jackson