DTN

SCAWG
Network Technology Workshop

Reston, VA

10 August 2006

Scott Burleigh
Systems Engineering Section
Jet Propulsion Laboratory, California Institute of Technology
818.393.3353
Scott.Burleigh@jpl.nasa.gov

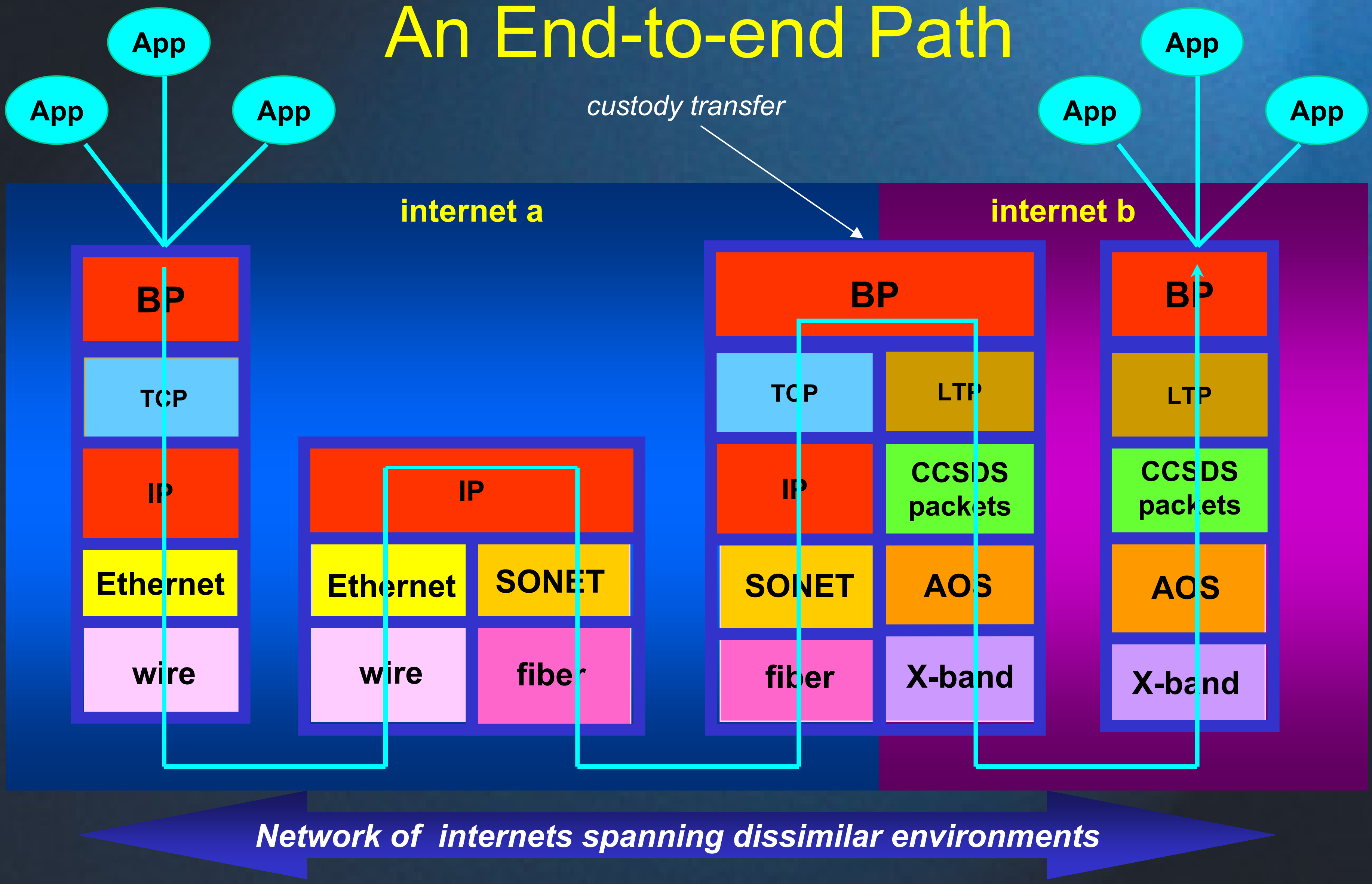
Delay-Tolerant Networking (DTN)

- An overlay network.
  - DTN “bundle protocol” (BP) is to IP as IP is to Ethernet.
  - A TCP connection within an IP-based network may be one “link” of a DTN end-to-end data path; a deep-space R/F transmission may be another.
- Reliability achieved by retransmission between relay points within the network, not end-to-end retransmission.
- Route computation has temporal as well as topological elements, e.g., a schedule of planned contacts.
- Forwarding at router is automatic but not necessarily immediate: store-and-forward rather than “bent pipe”.
- Contain DOS attacks: reciprocal inter-node suspicion.

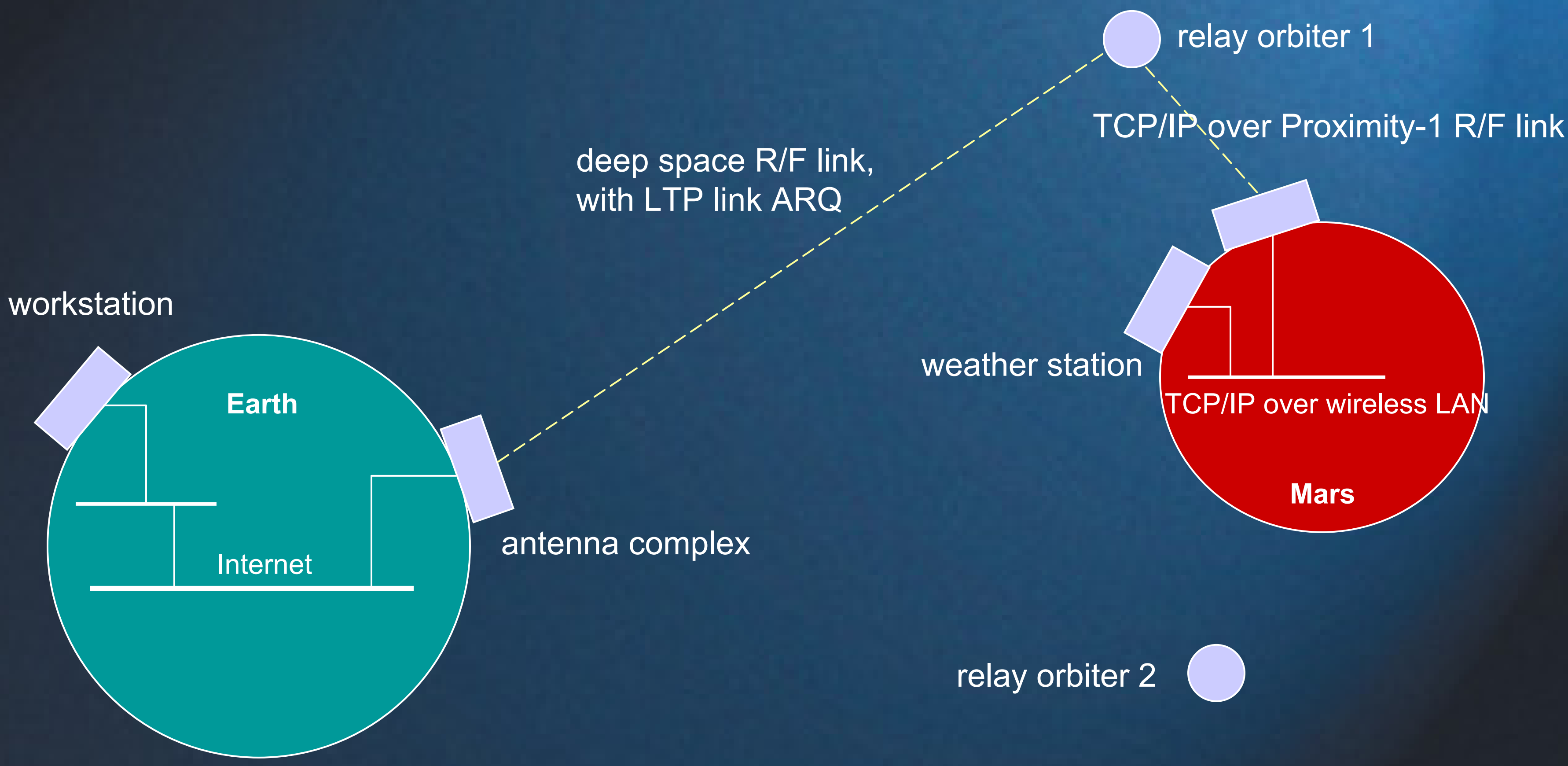
DTN Stack Elements for Deep Space

User application			
	File Transfer	Asynchronous Messaging	
Bundle Protocol **(intra-DTN routing)**			**Application layer**
LTP **(retransmission)**	**TCP (retransmission)**		**Transport layer**
CCSDS encapsulation packets	**IP (intra-Internet routing)**		**Network layer**
CCSDS AOS	**CCSDS Prox-1**	**Ethernet**	**Link layer**
R/F, optical	**wire**		**Physical layer**

An End-to-end Path



DTN Operations In Deep Space



DTN Current Status

- Specifications and documentation
  - Internet Draft for the DTN architecture
  - Advanced Internet Drafts for both the BP and LTP protocol specifications
    - Plan to submit these as Experimental RFCs within IETF in 2006
- Implementations
  - BP implementations
    - DTN2: open source reference implementation (Intel, UC Berkeley)
    - ION: designed for space flight (JPL)
  - LTP implementations
    - Reference implementation in Java (Ohio University)
    - C++ implementation for terrestrial applications (Trinity College)
    - C implementation designed for space flight (JHU/APL)

Remaining Problems

- Route computation algorithms
  - Very different types of contacts
    - Scheduled
    - Opportunistic
    - Predicted
  - Traditional metrics (distance vector, link state) don’t work.
    - They don’t take timing into account: a two-hop path available in 10 minutes may be better than a one-hop path available tomorrow.
    - Topology may change too rapidly for protocols to track.
- Congestion control
  - TCP congestion window and ICMP source quench are end-to-end, may not reduce data injection rate at source until congestion collapse has already occurred.

ION

- JPL's implementation of the DTN Bundle Protocol, designed for operations in deep space – Interplanetary Internet.
  - Static routing tables are practical for now, because the number of communicating nodes will remain small for decades.
    - Link initiation and termination remain the job of flight software, <u>not</u> the DTN router.
    - Outbound bundle handling:
      - Automatically issued on the appropriate links during the time the links are enabled.
      - Queued up for future transmission while the links are dormant.
  - Includes a congestion control system based on BP custody transfer.

Constraints

- Interplanetary internet is a classic DTN scenario:
  - Long signal propagation times, intermittent links.
- Links are very expensive, usually oversubscribed.
- Immediate delivery of partial data is often OK.
- Limited processing resources on spacecraft:
  - Slow (radiation-hardened) processors
  - Relatively ample memory
  - Solid-state storage
- For inclusion in flight software:
  - Processing efficiency is important.
  - Must port to VxWorks real-time O/S.
  - No malloc/free; must not crash other flight software.

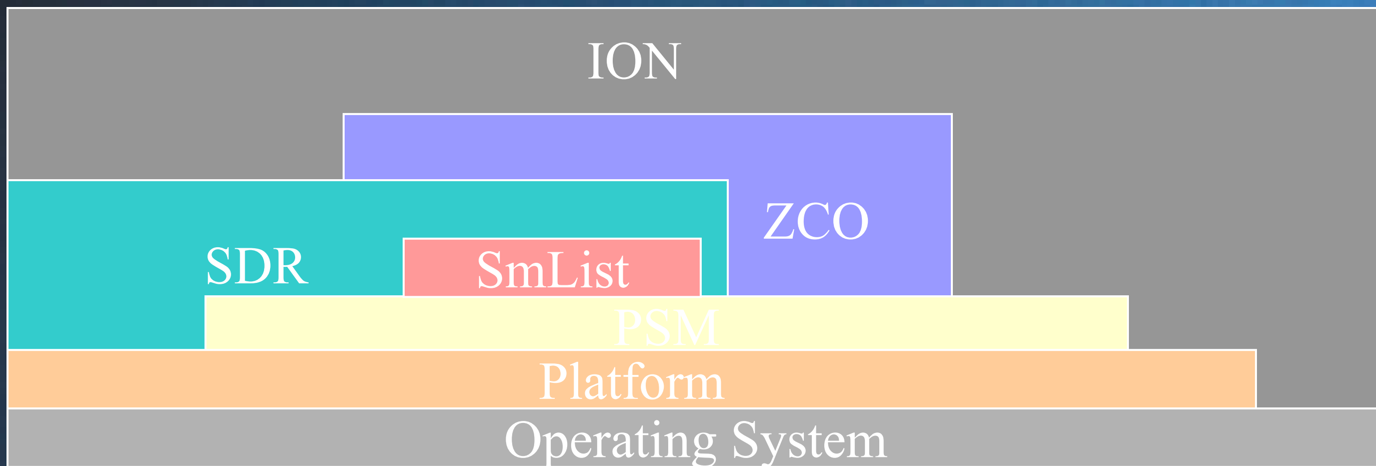
Applications

- Brief messages (typically less than 64 KB).
  - One bundle per message.
  - CCSDS Asynchronous Message Service (AMS) is being considered.
- Files, often structured in records.
  - Need to be able to deliver individual records as they arrive, so most likely one bundle per record.
  - CCSDS File Delivery Protocol (CFDP) is the standard.
- Streaming voice and video for Constellation.
- In general, we expect relatively <u>small bundles</u>.

Supporting infrastructure

- psm (Personal Space Management): high-speed dynamic allocation of memory within a fixed pre-allocated block.
  - Built-in memory trace functions for debugging.
- sdr (Spacecraft Data Recorder): robust embedded object persistence system; database for non-volatile state.
  - Performance tunable between maximum safety, maximum speed.
  - Again, built-in trace functions for usage debugging.
- zco (Zero-Copy Objects): reduce protocol layer overhead.
- platform O/S abstraction layer for ease of porting.
- Written in C for small footprint, high speed.
- Mostly inherited from Deep Impact flight software – flight proven.

Implementation Layers



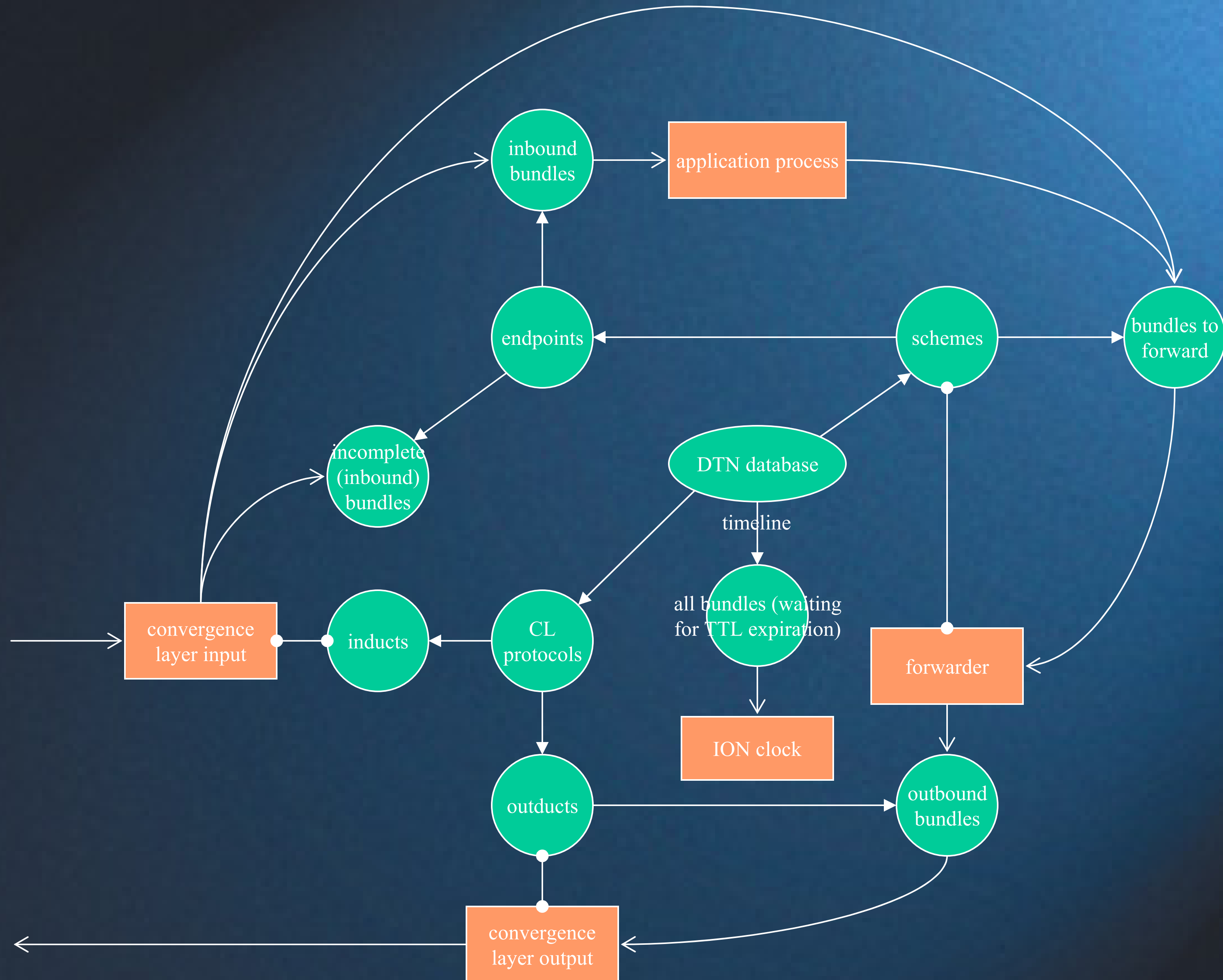
ION	Interplanetary Overlay Network libraries and daemons
ZCO	Zero-copy objects capability: minimize data copying up and down the stack
SDR	Spacecraft Data Recorder: persistent object database in shared memory, using PSM and SMList
SmList	linked lists in shared memory using PSM
PSM	Personal Space Management: memory management within a pre-allocated memory partition
Platform	common access to O/S: shared memory, system time, IPC mechanisms
Operating System	POSIX thread spawn/destroy, file system, time

Node architecture

- ION is database-centric rather than daemon-centric.
  - Each node is a single SDR database.
- Bundle protocol API is local functions in shared libraries, rather than inter-process communication channels.
- Multiple independent processes – daemons and applications, as peers – share direct access to the node state (database and shared memory) concurrently.

Node architecture (cont'd)

- Separate process for each scheme-specific forwarder.
  - Forwarder is tailored to the characteristics (endpoint naming, topology) of the environment implied by the scheme name.
- Separate process for each convergence-layer input and output.
  - No assumption of duplex connectivity.
- Schemes (forwarders) and convergence-layer adapter points can be added while the node is running.



inbound bundles
application process
endpoints
schemes
bundles to forward
incomplete (inbound) bundles
DTN database
timeline
all bundles (waiting for TTL expiration)
convergence layer input
inducts
CL protocols
forwarder
ION clock
outducts
outbound bundles
convergence layer output

Compressed Bundle Header Encoding (CBHE)

- For a CBHE-conformant scheme, every endpoint ID is
  scheme_name:*element_nbr*.*service_nbr*
- 65,535 schemes supported.
- Up to 16,777,215 elements in each scheme.
  - Element ~= node.
  - So the number of nodes addressable by scheme/element is 256 times the size of IPv4 address space.
- Up to 65,535 services in each scheme.
  - Service ~= "demux token" or IP protocol number.

CBHE (cont'd)

- For bundles traveling exclusively among nodes whose IDs share the same CBHE-conformant scheme name, primary bundle header length is fixed at 34 bytes.
  - Dictionary not needed, so it's omitted.
  - All administrative bundles are service number zero.

	Destination offsets		Source offsets		Report-to offsets		Custodian offsets	
Non-CBHE	Scheme	SSP	Scheme	SSP	Scheme	SSP	Scheme	SSP

CBHE	Common Scheme number	Destination Element number	Source Element number	Report-to Element number	Custodian Element number	Service Number for source & destination

Features implemented (and not)

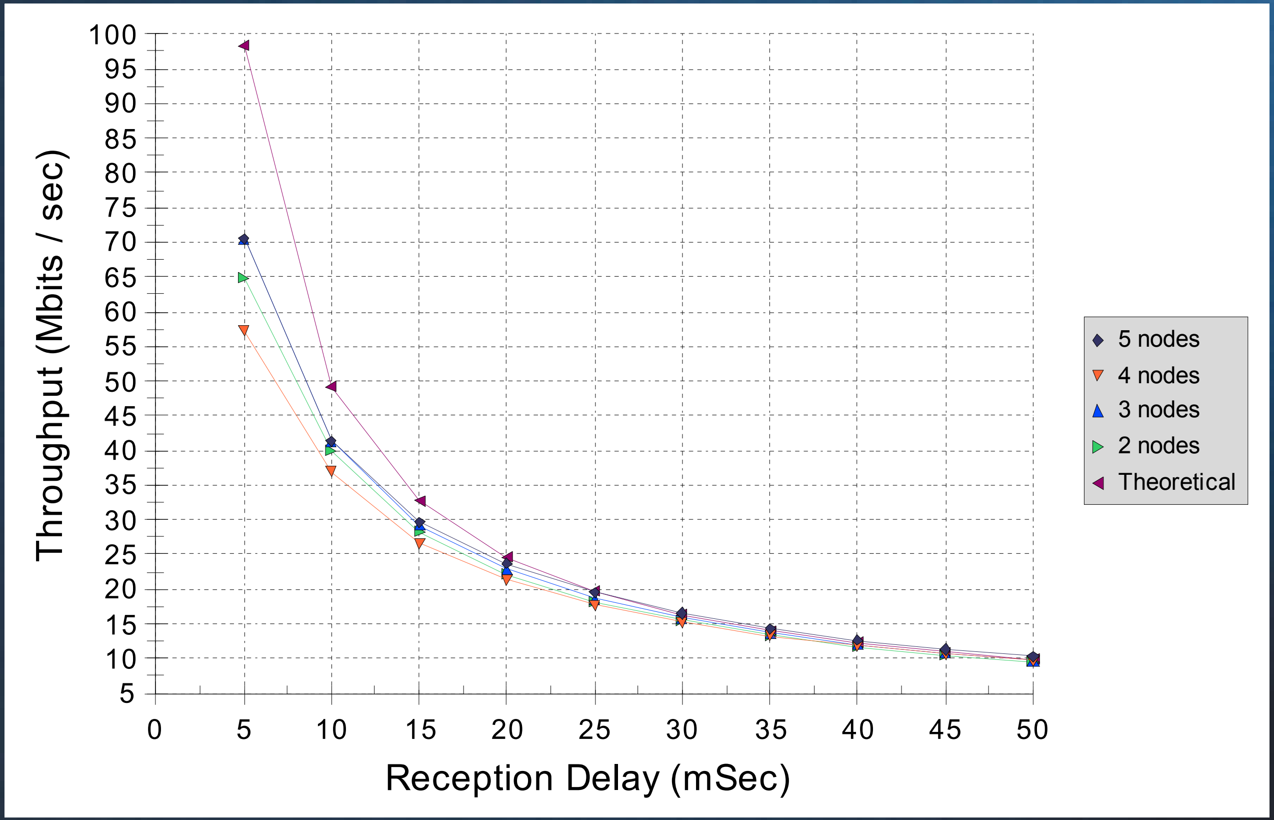
- Conforms to current BP specification (version 4, December 2005).
- Implemented: custody transfer, status reports, delivery options, priority, reassembly from fragments, for both CBHE and non-CBHE bundles.
  - Forwarder for the ipn scheme.
  - Convergence-layer adapters for TCP, “SPOF”.
  - Congestion control based on custody transfer.
- Partially implemented: flooding.
- Not implemented: fragmentation, application-initiated acknowledgements, security, multicast.

Performance

- Maximum data rate clocked to date is 352 Mbps.
  - Over a Gigabit Ethernet (single hop) between two dual-core 3GHz Pentium-4 hosts running Fedora Core 3, each with 800 MHz FSB, 512MB of DDR400 RAM, 7200 rpm hard disk.
  - sdr tuned to maximum speed and minimum safety.
  - No custody transfer.
- At the other extreme: running over a two-hop path on a 100-Mbps Ethernet between older Pentiums, with custody transfer over each hop:
  - With sdr tuned to maximum speed, about 40 Mbps.
  - With sdr tuned to maximum safety, only 3 to 4 Mbps.

Congestion Control Results

- No data loss and no router failure in any test.
- With zero artificial delay, the throughput rate measured between two nodes with no intervening routers was 300 Mbps.
- Throughput rates for other topologies and imposed delays are as shown:



Ports to date

- Linux (Red Hat 8+, Fedora Core 3)
  - 32-bit Pentium
  - 64-bit AMD Athlon 64
- Interix (POSIX environment for Windows)
- VxWorks (but not tested yet)

Evaluation copies distributed to date

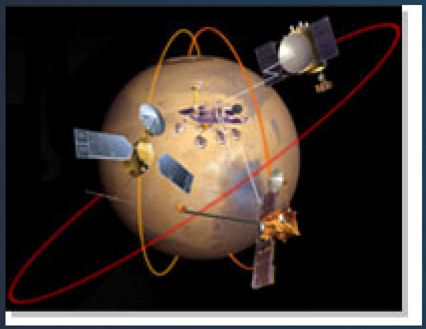
- NASA
  - Goddard Space Flight Center
  - Marshall Space Flight Center
  - Ames Research Center
  - Glenn Research Center
  - Constellation project
- ESA (European Space Agency)
- CNES (the French national space agency)
- Johns Hopkins University Applied Physics Laboratory
- MITRE Corporation
- Interface & Control Systems

Backup slides

Deep Space Communications Today

- Communication opportunities are scheduled, based on orbit dynamics & operations plans.
- Transmission initiation is manual, per schedule.
- Transmission direction is manual: point antenna, start transmitting when the right spacecraft is listening.
- Retransmission is manual: on loss of data, command repeat.
- More recently (MER), manual forwarding through relay point: command to Odyssey or MGS.

What's Wrong With That?



- This mission communications model has worked fine for over forty years; we've done a lot of good science.
- But the status quo is:
  - Labor-intensive
    - Communication operations cost is a large fraction of the budget for each mission.
    - Risk of human error mandates mitigations that further increase cost.
  - Program-limiting
    - Cost and risk increase with the number of links between communicating entities.
    - As cross-links among spacecraft become common (Mars network, lunar exploration Constellation), cost and risk increases are non-linear with increase in the number of spacecraft.

An Alternative

- The Internet has come to be widely used to conduct scientific investigations, for both science and engineering telemetry.
  - For example, the High-Performance Wireless Research and Education Network (HPWREN) in southern California.
    - Astronomy.
    - Ecology.
    - Geophysics.
- So why not use it for deep space science missions too?
  - Minimize cost (automation, COTS).
  - Minimize risk (huge installed base).

It Works Fine in Near-Earth Space

- Space Communication Protocol Standards (SCPS)
  - TCP options that improve performance on satellite links, where data loss is more often due to corruption than to congestion
  - international standard
- Operating Missions as Nodes on the Internet (OMNI)
  - UoSAT-12, an HTTP server in orbit
  - CHIPSat, used Internet protocols on all communication links
  - CANDOS on STS-107, used mobile IP
- IP stack would likely also work well in cislunar space and in surface networks on other planets.

So What's the Problem?

- Interplanetary space is a qualitatively different communication environment.
  - Internet, near-Earth, and planetary surface networks are all characterized by:
    - Very short distances between communicating nodes, therefore very brief signal propagation delays (up to about a second).
    - Continuous end-to-end connectivity. A lapse in connectivity on any single link is treated as an anomaly and allowed to terminate communication.
  - Any network spanning interplanetary space would be characterized by:
    - Long distances between communicating nodes, lengthy signal propagation delays (e.g., 8-20 minutes from Earth to Mars).
    - Routine lapses in connectivity on all links of end-to-end path.

It’s All About Delay

- Network disruption is, essentially, unpredictable delay.
  - Case 1: continuous connectivity but client is 56 million miles from server. *Response to query arrives 10 min. after query is issued*.
  - Case 2: client and server are in adjacent offices but router is powered off for 10 minutes. *Response to query arrives 10 min. after query is issued*.
- Key effect of delay: reliable transmission of a given byte of data can take an arbitrarily long time.
  - Transmission can be lost due to corruption, N times.
  - NAK can be lost due to corruption, N times.
  - Disruption can delay transmission of NAK (or retransmission of data) by an arbitrarily long time.

Effects of Long and/or Variable Delay

- Connection establishment could take more time than entire communication opportunity.
  - So protocols must be connectionless.
- Transmission history can't be used to predict round-trip times.
  - So communication timeout interval computation must rely on link state information rather than timing statistics.
- End-to-end retransmission would reserve resources (retransmission buffer) at originator for entire duration of the transaction – possibly days or weeks.
  - So retransmission should be between relay points within the network rather than end-to-end: custody transfer.

Effects of Delay (cont'd)

- In-order stream delivery could be stuck for a long time, waiting for byte N to arrive before delivering byte N + 1.
  - So out-of-transmission-order delivery is needed – multiple concurrent transmissions.
  - So data must be structured in transmission blocks (e.g., messages) for concurrent retransmission – *not streams*.
- But reliable transmission of any single block can take an arbitrarily long time.
  - So any number of message transmissions might be in progress at the moment a computer is rebooted or power cycled.
  - So retransmission buffers should reside in non-volatile storage – not memory – to minimize risk of massive transmission failure.

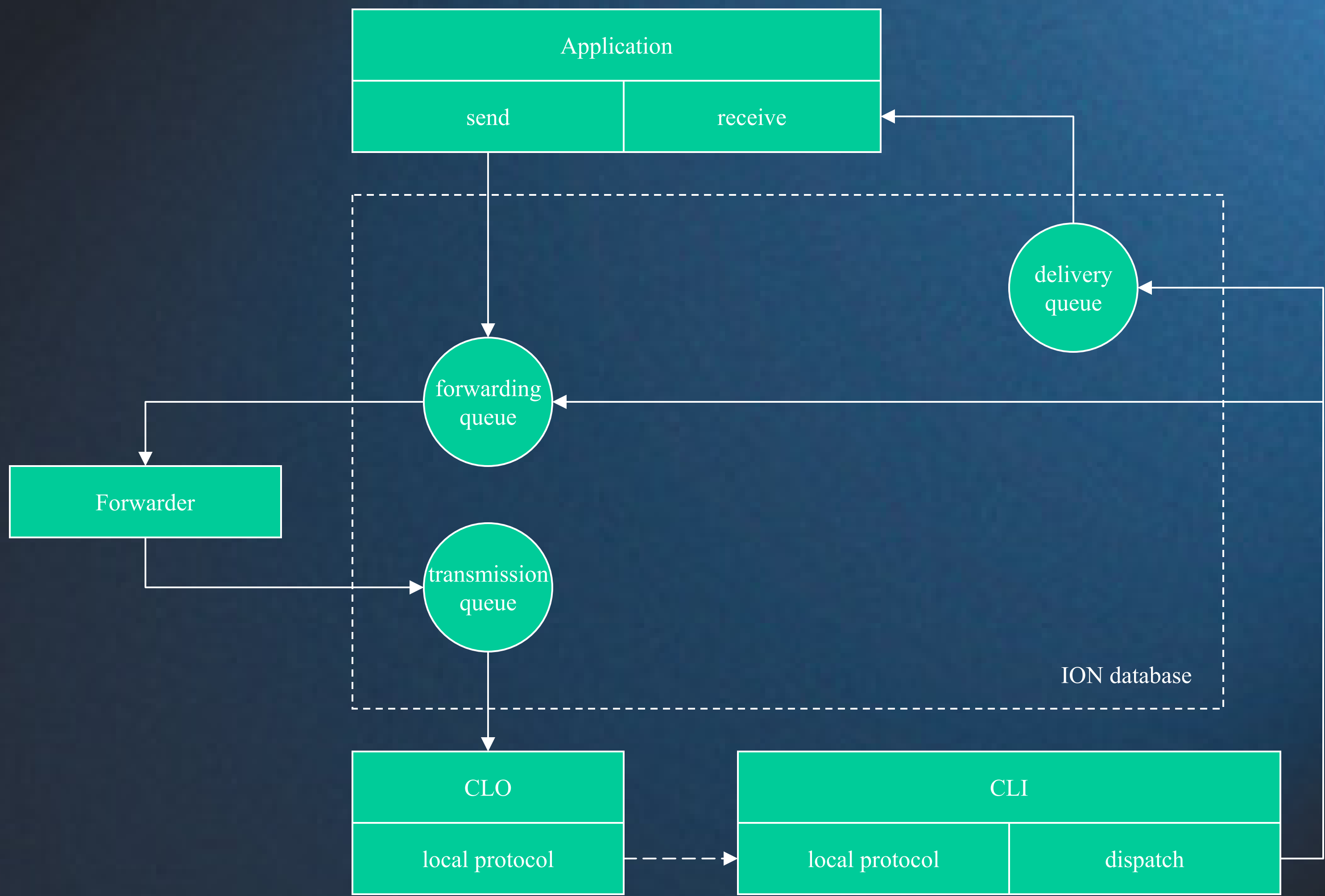
Interplanetary IP – the Bottom Line

- None of these effects preclude the use of the IP network protocol (IP datagram transmission) itself.
- But:
  - TCP isn't suitable.
    - Connections, streaming, end-to-end retransmission, in-order delivery.
    - Retransmission buffers are in memory.
    - Timeout intervals are computed from transmission history.
  - The BGP external routing protocol uses TCP, so it's not suitable.
  - Internal routing protocols use history-based timeouts to detect route failures, so routine loss and re-establishment of connectivity would incorrectly cause route failure to be inferred and propagated to routing tables. Not suitable.
- The off-the-shelf IP stack doesn't work for deep space.

Where Does That Leave Us?

- We could simply use IP anyway.
  - Omit routing protocols; just manage static routes.
  - Omit TCP, leave reliability to the applications and/or ops.
- But this would be functionally the same as status quo.
  - TCP-reliant Internet applications wouldn't work.
  - Would still be labor-intensive and program-limiting.
- Alternatively: develop a new automated network architecture that is tolerant of long and/or arbitrary delay.
  - TCP-reliant Internet applications still won't work, but in some cases we can proxy them into the new infrastructure.
  - Reduce cost and risk: automate network functions, automate retransmission, integrate easily with Internet.

Processing Flow



CLI

- Acquire bundle from sending CLO, using the underlying CL protocol.
- Dispatch the bundle.

dispatch

- Local delivery: if an endpoint in the database (that is, an endpoint in which the node is registered) matches the destination endpoint ID, append bundle to that endpoint's delivery queue.
- Forwarding: append bundle to forwarding queue based on scheme name of bundle's destination endpoint ID, with "proximate destination EID" initially set to the bundle's destination EID.
  - Forwarder later appends it to outduct's transmission queue; see ipn forwarder below.

CLO

- Pop bundle from outduct’s transmission queue.
- As necessary, map the associated destination duct name to a destination SAP in the namespace for the duct’s CL protocol. (Otherwise use the default destination SAP specified for the duct.)
- Invoke that protocol to transmit the bundle to the selected destination SAP.

The “ipn” scheme

- CBHE-conformant, so every EID is:

  ipn:*element_nbr.service_nbr*

  - “Elements” notionally map to Constellation elements, such as the Crew Exploration Vehicle.
  - Services:
    - 1 currently used for test.
    - 2 could be CFDP traffic.
    - 3 to N could be traffic for Remote AMS applications.
      - Element number might additionally serve as AMS continuum number.

ipn-specific forwarder

- Use proximate-destination element number as index into array of “plans”; use source element number and/or service number to select rule in that plan (or use default rule).
- If rule cites another EID:
  - If non-ipn scheme, append (with proximate destination EID changed) to that scheme’s forwarding queue.
  - Else, iterate with new proximate-destination element number.
- Otherwise (rule is outduct reference and, possibly, name of destination induct):
  - Insert bundle into the transmission queue for that outduct, noting the associated destination induct name [if any].