JPL’s Bundle Protocol Implementation: Interplanetary Overlay Network (ION)

Constraints

- Interplanetary internet is a classic DTN scenario:
  - Long signal propagation times, intermittent links.
- Links are very expensive, always oversubscribed.
- Immediate delivery of partial data is often OK.
- Limited processing resources on spacecraft: slow (radiation-hardened) processors, but 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 small bundles.

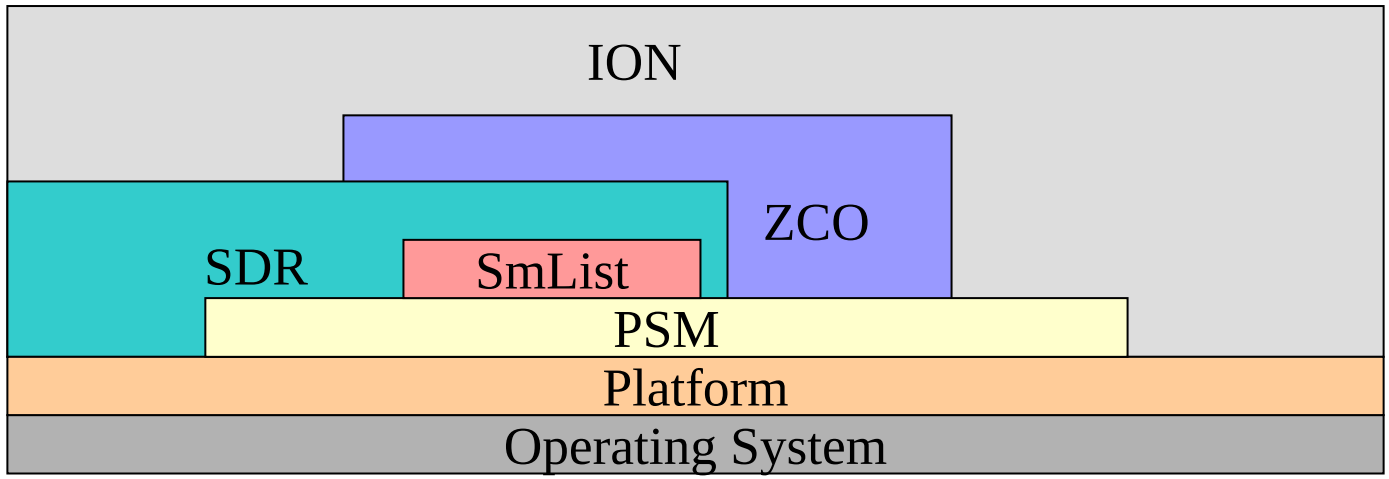
Outline

- Supporting infrastructure: sdr, psm, zco, platform.
  - Mostly inherited from Deep Impact flight software.
- Node architecture.
- Processing flow.
- Compressed Bundle Header Encoding (CBHE).
- The “ipn” scheme: EID structure, forwarding.
- Features implemented and not implemented.
- Ports to date.
- Performance.
- Distribution to date.

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 debugging usage.
- zco (Zero-Copy Objects): reduce protocol layer overhead.
- platform O/S abstraction layer for easy porting from Linux to VxWorks, Solaris, Interix.
- Written in C for small footprint, high speed.

Implementation Layers



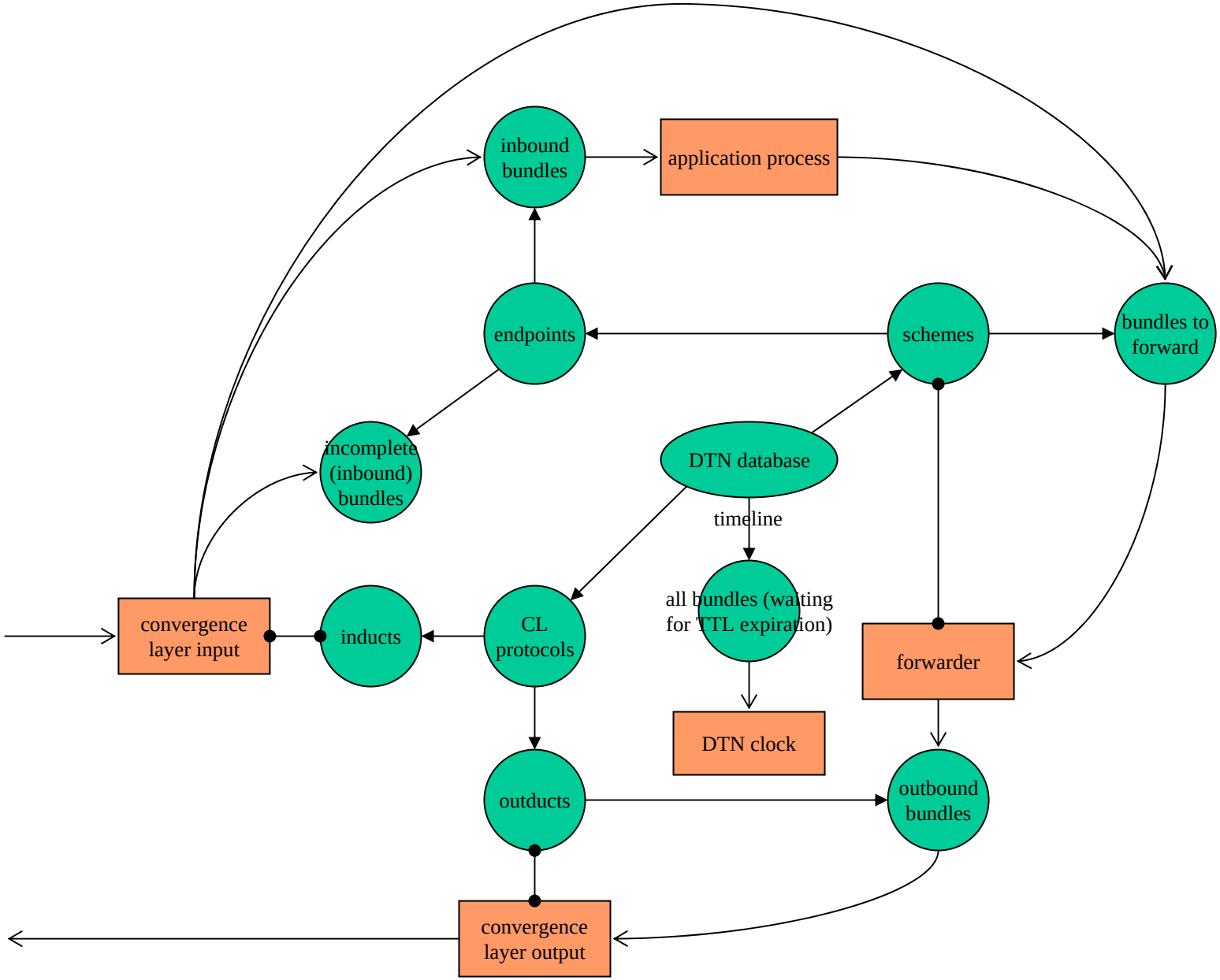
ION	Interplanetary Overlay Network libraries and daemons
ZCO	Zero-copy objects capability: minimize data copying up and down the stack
SDR	Spacecraft Data Recorder = persistent object database in shared memory, using PSM and SMList
SmList	linked lists in shared memory using PSM
PSM	Personal Space Management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

Node architecture

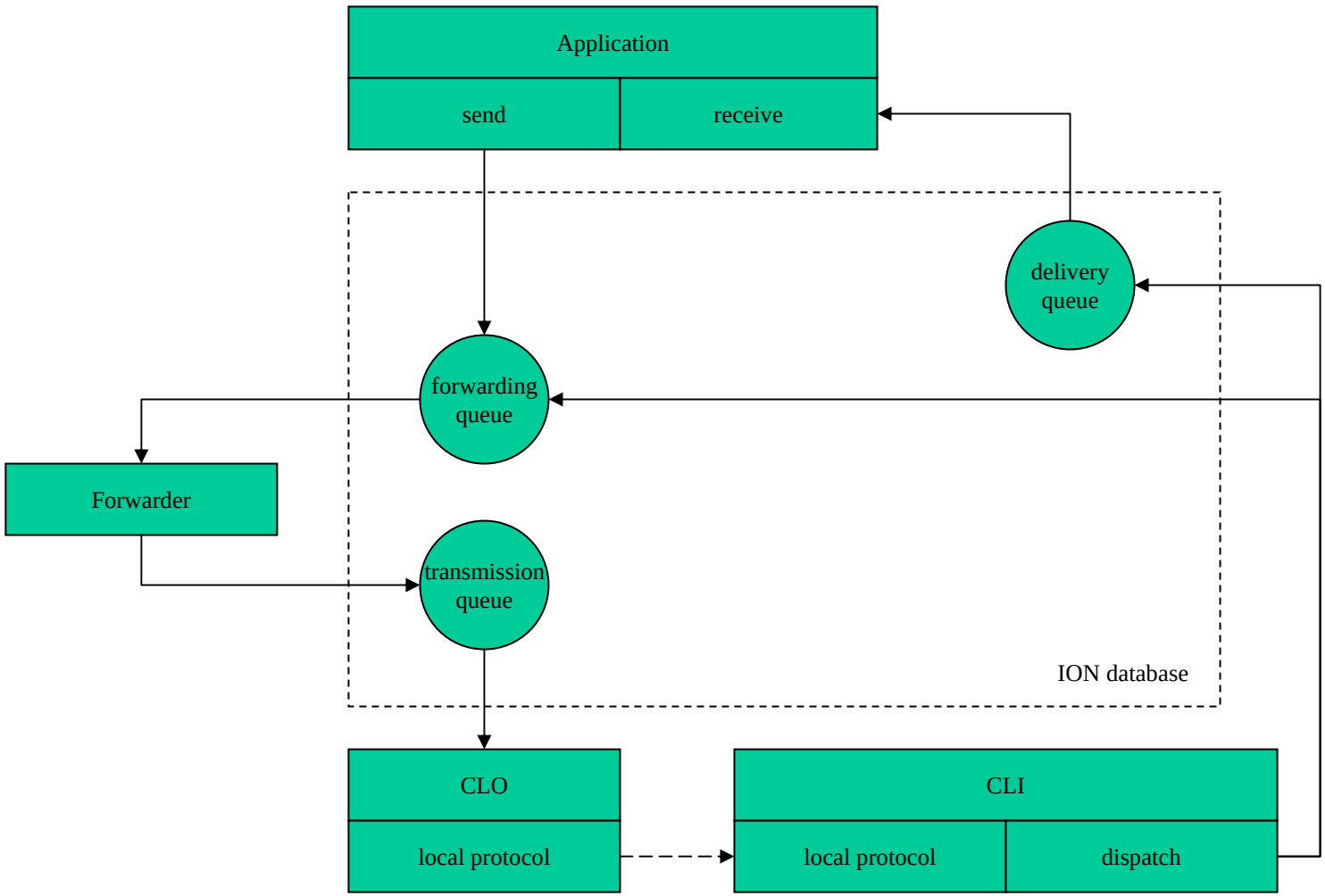
- ION is database-centric rather that daemon-centric.
  - A node is a database.
- Bundle protocol API is local functions in shared libraries, rather than inter-process communication channels.
- Multiple independent processes – daemons and applications, as peers – share direct access to the node state (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
convergence layer input
inducts
CL protocols
all bundles (waiting for TTL expiration)
forwarder
DTN clock
outducts
outbound bundles
convergence layer output

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.

CBHE

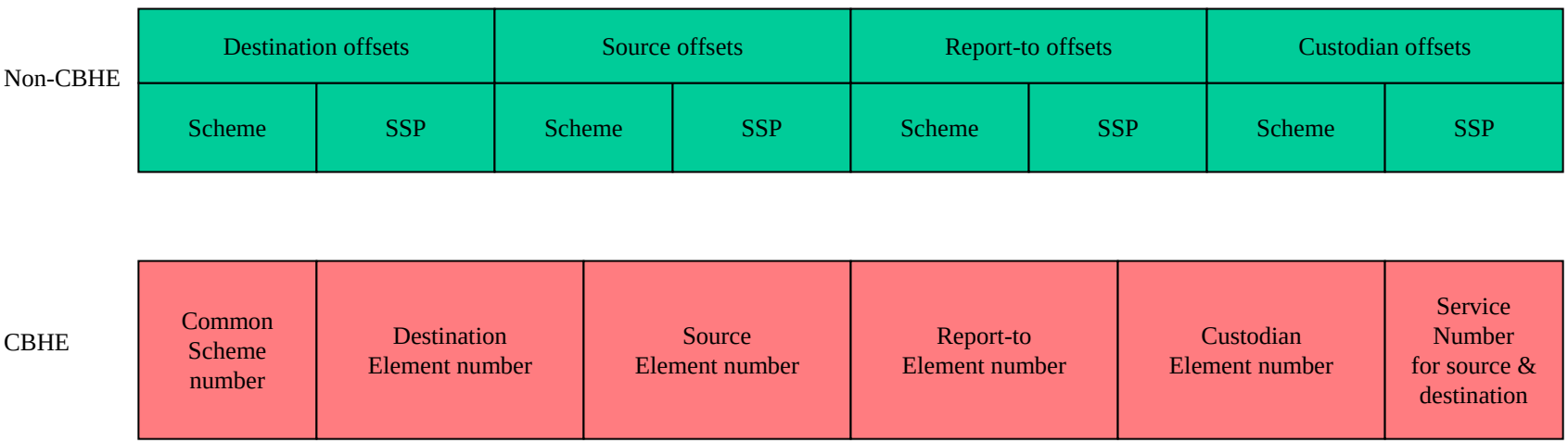
- For a CBHE-conformant scheme, every endpoint ID is

  scheme_name:*element_nbr*.*service_nbr*

- 65,535 schemes supported.
- Up to 16,777,215 elements in each scheme.
  - Element ~= node.
  - 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 is not needed, so it is omitted. All administrative bundles are service number 0.

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 continua. (Element number might also serve as AMS continuum number.)

ipn-specific forwarder

- Use proximate-destination element number as index into plans array; 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, destination induct name):
  - Insert bundle into the transmission queue for that outduct, noting the associated destination induct name [if any].

Features implemented (and not)

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

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)

Performance

- Maximum data rate clocked to date is 300 Mbps.
  - Over a Gigabit Ethernet 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 with custody transfer requested and sdr tuned to maximum safety, only 3 to 4 Mbps.

Evaluation copies distributed to date

- ESA (European Space Agency)
- CNES (the French space agency)
- Johns Hopkins University Applied Physics Laboratory
- NASA Constellation project