Overview

- DTN convergence layer protocol that uses TCP as a transport
- Motivation:
  - message framing, app-level acks, reactive fragmentation
- draft-demmer-tcp-clayer-00.txt
Basic Protocol Operation

- Contact header for session setup
- Data segments / acks exchanged in each direction
- Connection close due to idle timer
- Shutdown msg to convey reason code
Contact Header

- Magic + version to establish session
- Flags: enable acks, enable reactive fragmentation, enable negative acks
- Keepalive interval
- Local eid for bundle layer identification
CL Message Format

- Preamble byte with four bit type code and four bits of flags
- Defined message types:
  - 0x1: DATA_SEGMENT
  - 0x2: ACK_SEGMENT
  - 0x3: REFUSE_BUNDLE
  - 0x4: KEEPALIVE
  - 0x5: SHUTDOWN
Data / Ack Segments

• Bundle data sent in configurable chunks
  – Minimal overhead: 1 byte + SDNV length
  – First & last data segments marked with corresponding flags
• Cumulative acks returned per-segment
  – E.g. Three 10k segments => ack 10k, 20k, 30k
• Enables reactive fragmentation if the connection unexpectedly breaks
Keepalive

- Contact header exchange negotiates a keepalive interval
- Enables peering nodes to distinguish idleness from network partition
- Keepalives should not stop a node from closing an idle link
• In response to a data segment, node may send refuse bundle message
• Enables early duplicate detection (i.e. before payload transmitted)
• Optionally enabled (not implemented by DTN2 reference implementation)
Shutdown

- App-level shutdown byte to precede TCP FIN
- Optional reason code
  - Idle timeout, version mismatch, busy, others?
- Optional reconnection delay request
Design Questions

- **Protocol version management:**
  - Current spec relies on newer version to "downgrade" or disconnect

- **Alternative:** future extension messages ignored with common way of specifying length