A Delay Tolerant Networking-based Approach to a High Data Rate Architecture for Spacecraft

Alan Hylton
Daniel Raible
Gilbert Clark
NASA Glenn

Delay Tolerant Networking - DTN

The goal is to enable networking despite disconnections and latency

- store, carry, and forward
- the architecture is defined in IETF RFC 4838

The unit of data is called the *bundle*:

- defined in a CCSDS Blue Book & IETF RFC 5050
- can have any size
- does not have fixed-width header fields
  - self-delimiting numeric values (SDNV)

NASA's reference implementation is the Interplanetary Overlay Network (ION)

High Data Rate Architecture - HiDRA

Our focus: Performance optimization for upcoming space systems

- enter *High-rate DTN*, or HDTN
- start *slower* : ≈ 1-10 Gbps
- get *faster* : ≈ 100-200Gbps
  - e.g., for trunk links

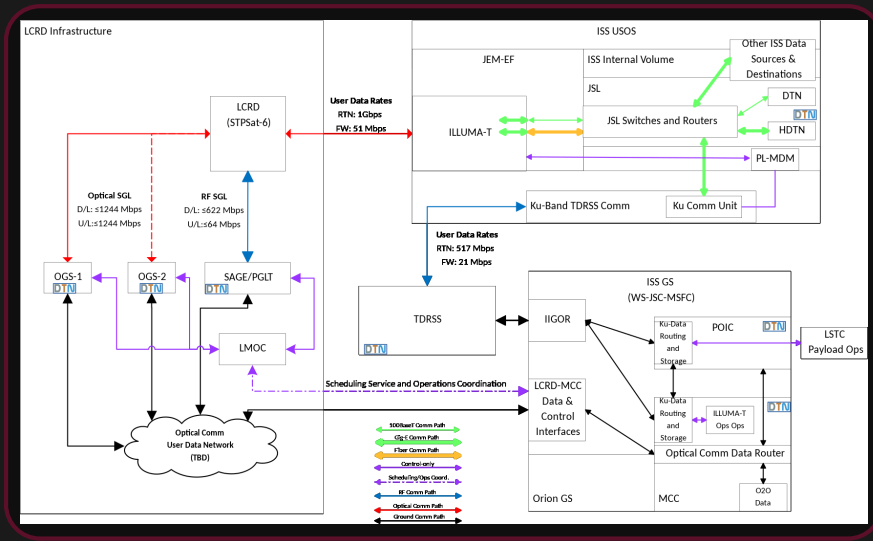
ION was created with deep space in mind

- slower systems
- slower buses

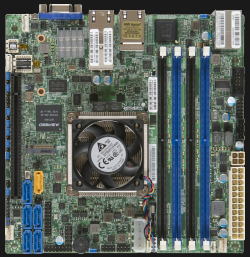
Each implementation has its focus

- research, academic, terrestrial, etc.

Upgrades and ILLUMA-T



ISS Computer Trade Study



- Xeon D
- ARM
- Atom

Setup

- Threadripper 2290WX *ipn:1*
  - 10GbE, 40GbE
- ARM Cortex A-72 *ipn:2*
  - 10GbE
- Xeon D-1540 *ipn:3*
  - 10GbE
- Atom E3826 *ipn:4*
  - 1GbE
- Intel i7-7700k *ipn:5*
  - 40GbE

Setup

- 1 & 10GbE links connected via 10GbE switch
- 40GigE links directly connected
- Network always connected
  - we have demonstrated store, carry, & forward capability
- Network-centric testing
  - all storage in RAM

Test Plan

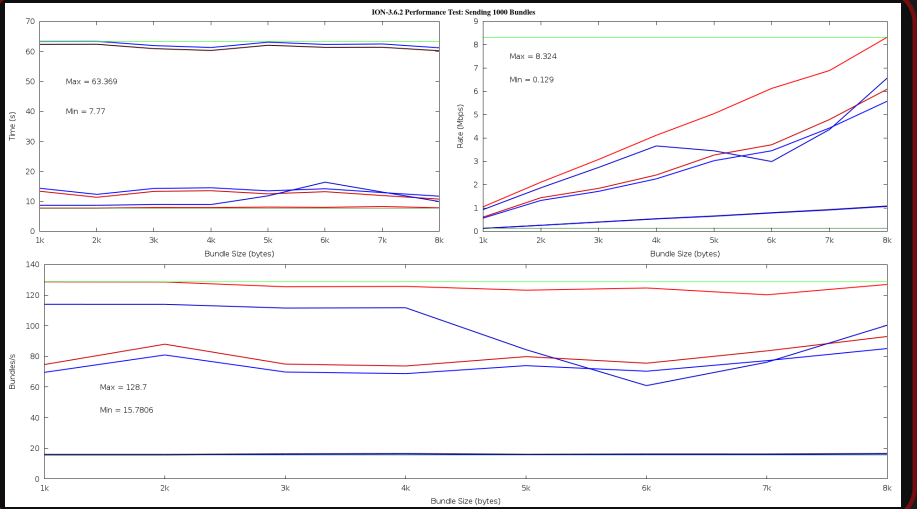
First tests:

- Establish ION baseline using UDP
- Measure sender & receiver stats for each test
  - 1000 bundles/test
  - 1k - 8k bundle sizes
- Run tests on both ION and HDTN

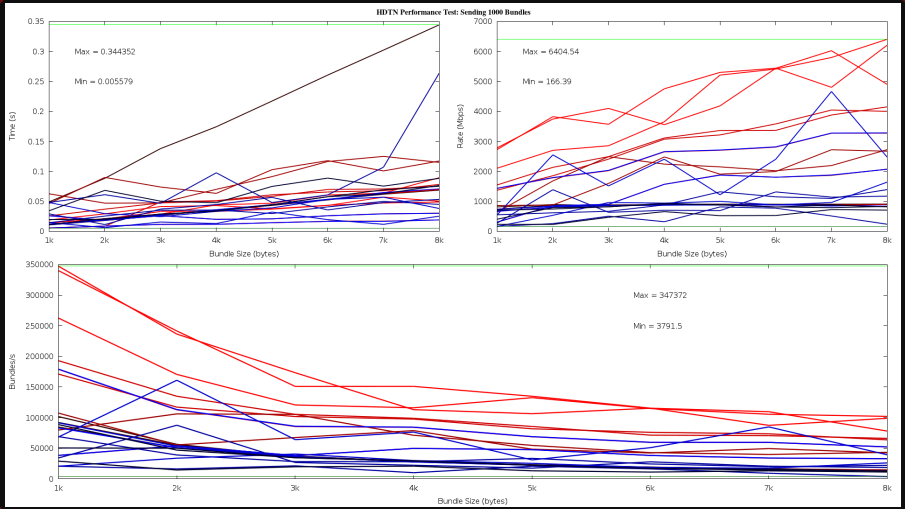
Second tests:

- Send similar bundles sizes as quickly as possible (HDTN only)
  - UDP & TCP

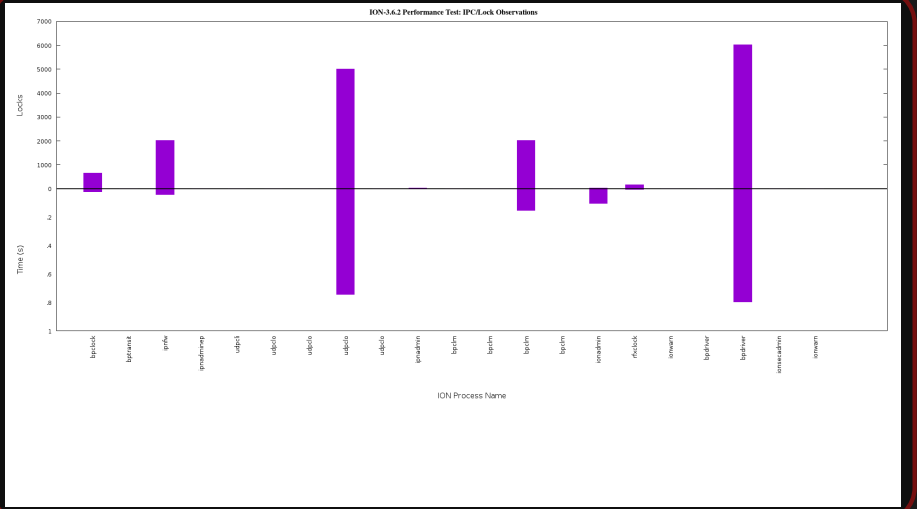
Baseline - Interplanetary Overlay Network (ION)



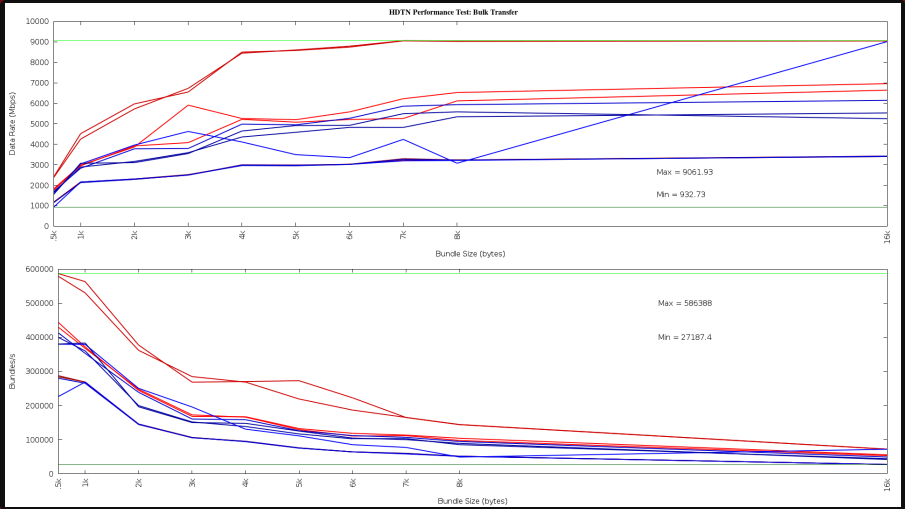
HDTN


HDTN Performance Test: Sending 1000 Bundles
Max = 0.344352
Min = 0.005579
Time (s)
Bundle Size (bytes)
Max = 6404.54
Min = 166.39
Rate (Mbps)
Bundle Size (bytes)
Max = 347372
Min = 3791.5
Bundles/s
Bundle Size (bytes)

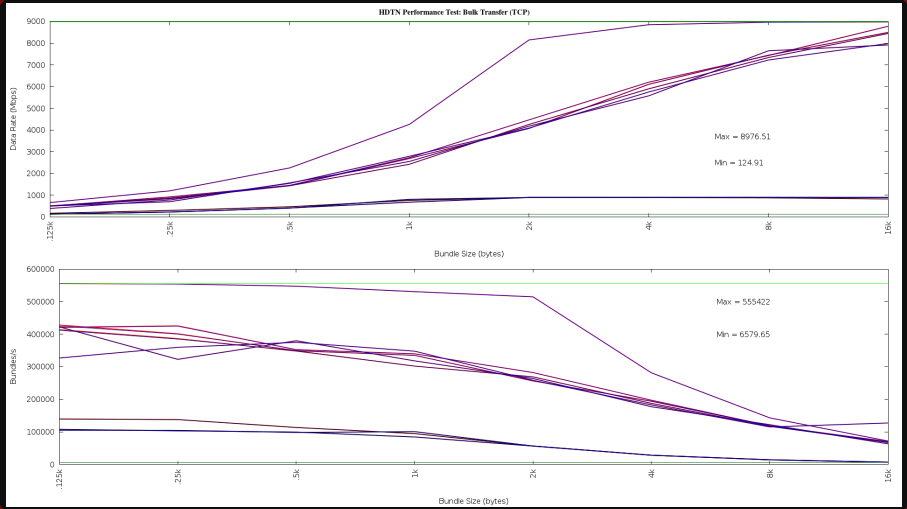
ION Revisited



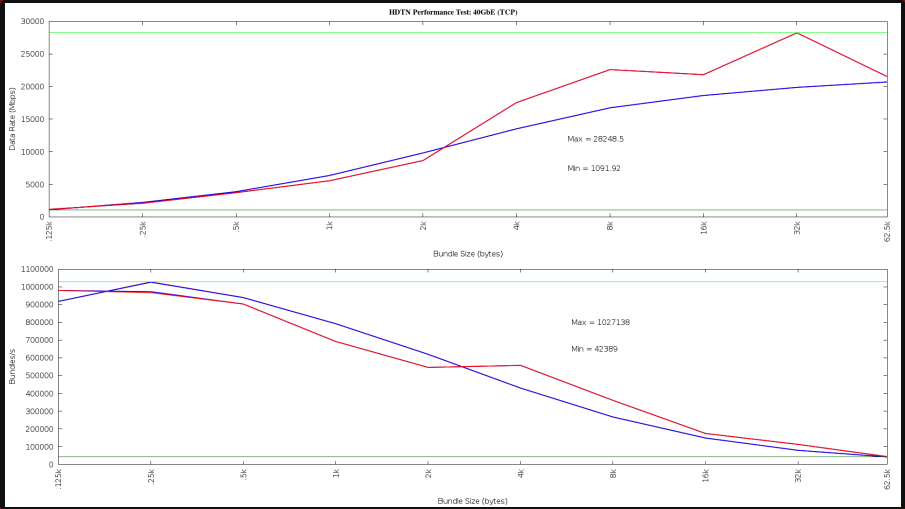
HDTN → HDTN: 10GbE Tests



HDTN → HDTN: 10GbE Tests (TCP)

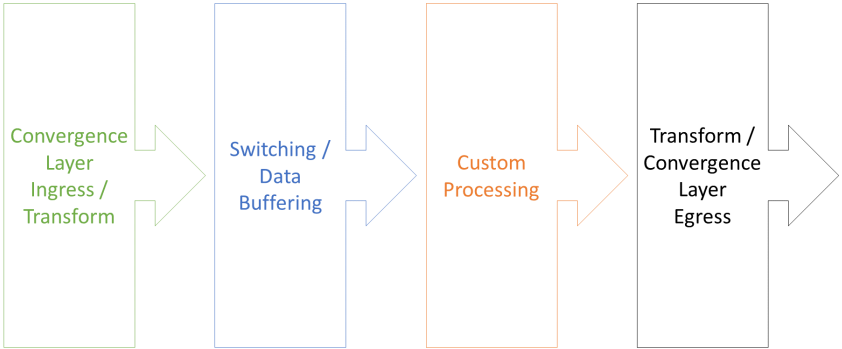


HDTN → HDTN: 40GbE Tests (TCP)



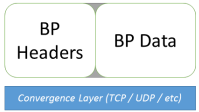
The Prototype HDTN Pipeline

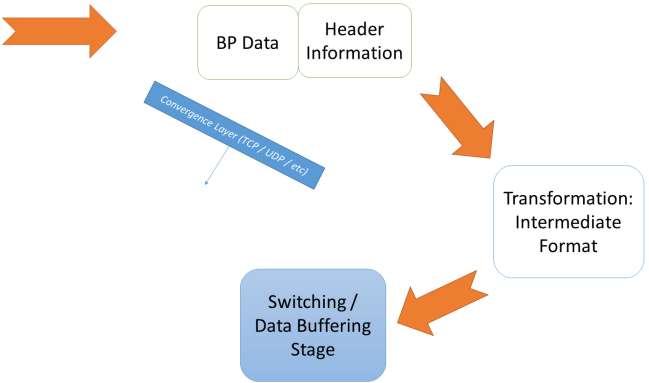
A bundle starts at the left and works its way toward the right, with multiple custom processing steps possible

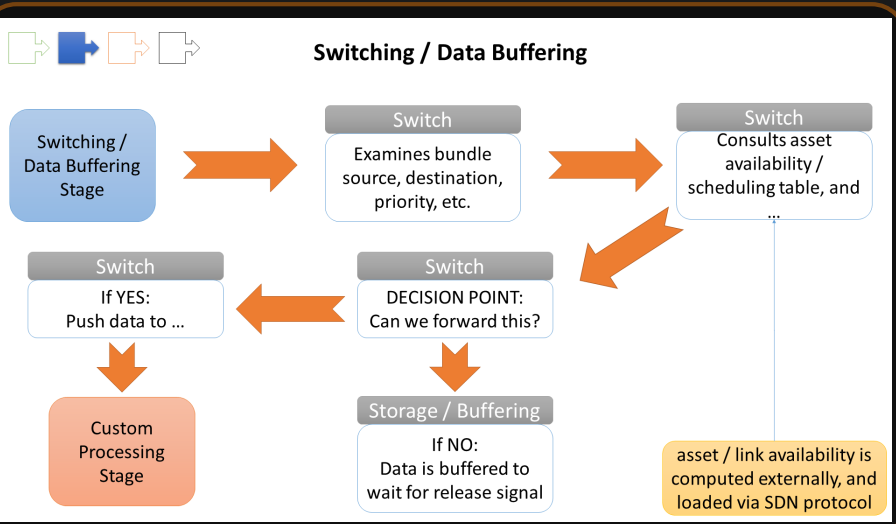


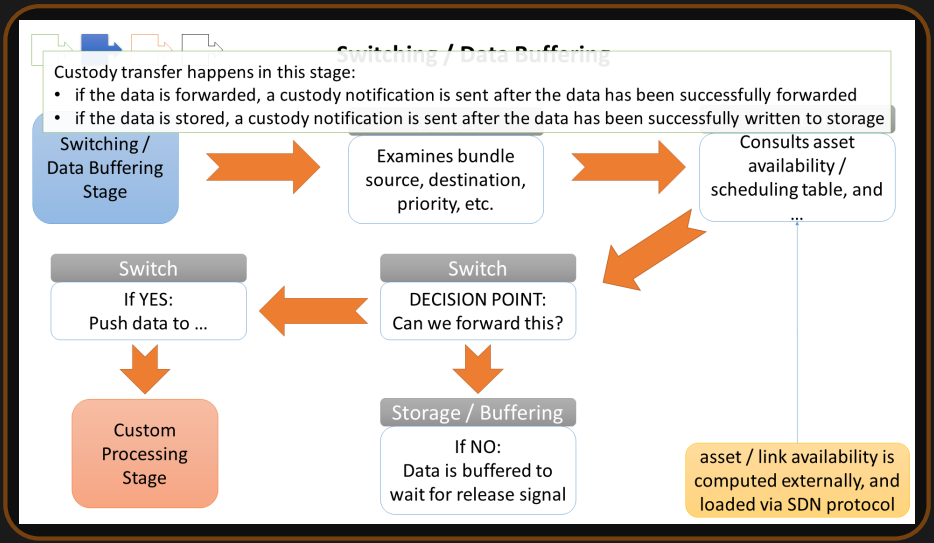
Ingress, Switching, and Egress lend themselves to parallelization and support 1 .. N worker processes / threads
This can be combined with SR-IOV, hardware queues, and other elements to achieve high throughput in software

Convergence Layer Ingress / Transform






Switching / Data Buffering
Switching / Data Buffering Stage
Switch
Examines bundle source, destination, priority, etc.
Switch
Consults asset availability / scheduling table, and ...
Switch
DECISION POINT: Can we forward this?
Switch
If YES: Push data to ...
Custom Processing Stage
Storage / Buffering
If NO: Data is buffered to wait for release signal
asset / link availability is computed externally, and loaded via SDN protocol


Switching / Data Buffering
Custody transfer happens in this stage:
• if the data is forwarded, a custody notification is sent after the data has been successfully forwarded
• if the data is stored, a custody notification is sent after the data has been successfully written to storage
Switching / Data Buffering Stage
Examines bundle source, destination, priority, etc.
Consults asset availability / scheduling table, and ...
Switch
If YES: Push data to ...
Switch
DECISION POINT: Can we forward this?
Custom Processing Stage
Storage / Buffering
If NO: Data is buffered to wait for release signal
asset / link availability is computed externally, and loaded via SDN protocol

Custom Processing

Custom Processing Stage

Registration Table

User-defined expressions / rules are applied to fields in the bundle, and bundle may be forwarded to user applications

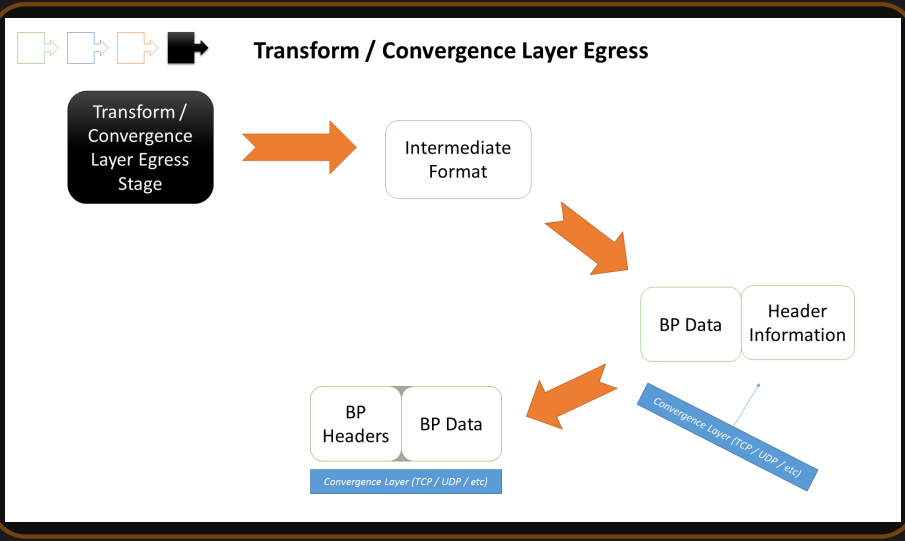
Registration Table

Once all expressions have been satisfied, the bundle is forwarded to …

Fields	Action	Priority
ipn.src == 32	DROP	40000
ipn.src == me && ipn.svc == 3147	FORWARD TO <my_application>	32768
ipn.dst < 8	FORWARD TO CONTROLLER	16384
ipn.dst == 255	FLOOD ALL AVAILABLE	16384
payload.size > 32768	TAG <FLAG_LARGE>	16384

Transform / Convergence Layer Egress Stage

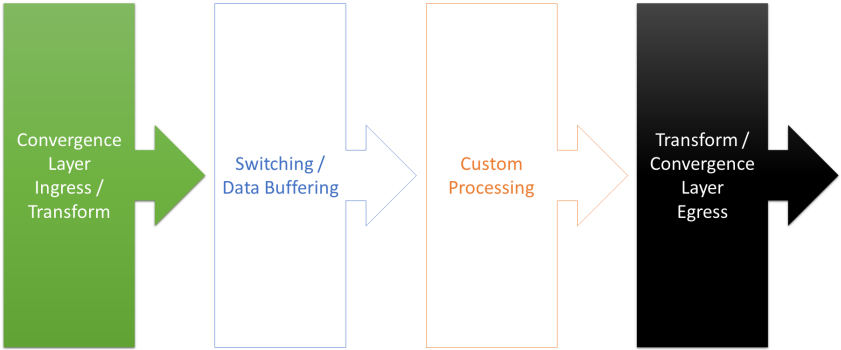
We're working toward a well-defined set of headers and actions based on OpenFlow and similar protocols.


Transform / Convergence Layer Egress
Transform / Convergence Layer Egress Stage
Intermediate Format
BP Data
Header Information
BP Headers
BP Data
Convergence Layer (TCP / UDP / etc)
Convergence Layer (TCP / UDP / etc)

What We Discussed Today

We tested two elements: how fast per-thread ingress can go, and how fast per-thread egress can go.

These two pieces, as well as switching aspects, are in good shape: other elements are a work in progress.



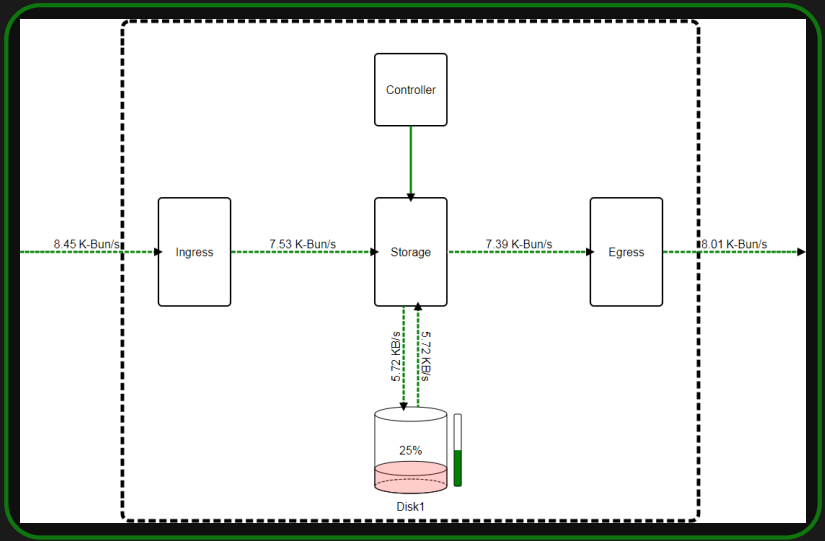
Next Steps - Software

- Prototype storage component
  - Toward an efficient, persistent key/value bundle store
- Work on a suite of killer test applications
  - e.g. bpping, bpsource, bpsink
- Move toward HDTN 2.0
  - Transition from a prototype-driven feasibility assessment toward an actual project

Next Steps - Hardware

- Bundle Protocol co-processor / acceleration
  - Largely FPGA-driven, with hope to eventually move to ASIC
- Explore SDN platforms
  - Identify extensions to e.g. OpenFlow to allow interaction with our systems
- Build / acquire additional development platforms for characterization

Next Steps - Web Interface



Discussion Action