

Diving into the Data Plane of Delay Tolerant Networks

Dominick Ta

University of Washington - Seattle

Mentors:

Stephanie Booth & Rachel Dudukovich

Cognitive Communications Project and

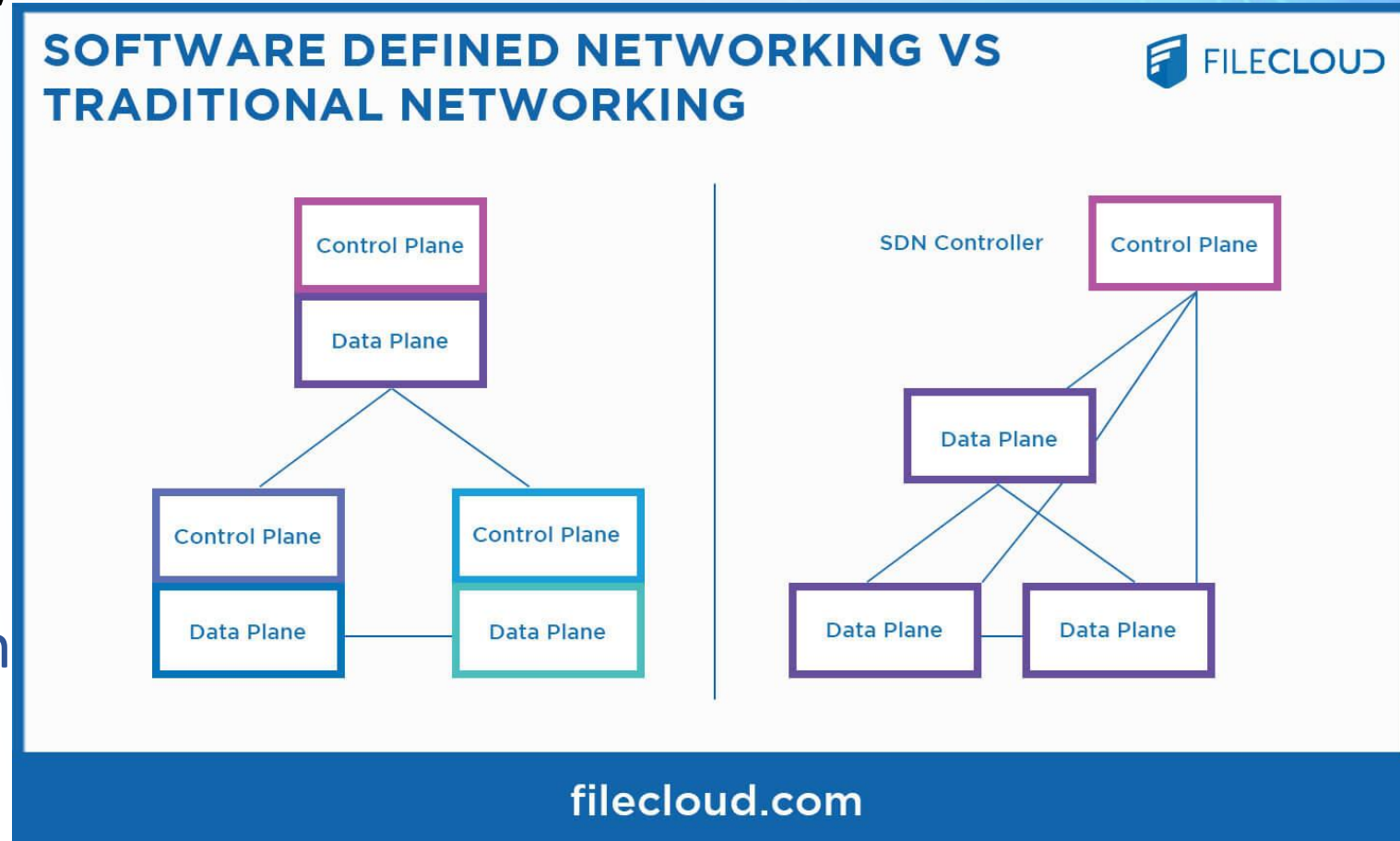
High-rate Delay Tolerant Networking Project

A Challenge with Space Networks

- Scalability in DTN is an unsolved challenge
 - DTN has been demonstrated to work in small scale
 - A Solar System Internet dramatically increases number of nodes
- Managing these large networks with unreliable links is difficult

Our Solution: SDN and Data Plane Programming

- Apply SDN techniques to delay tolerant networks
 - Requires networking hardware to understand bundle protocol (BP)
- We can use data plane programming to achieve this
- Success implicates:
 - network hardware that can natively forward BP
 - process BP faster
 - ability to control network behavior based on BP logic

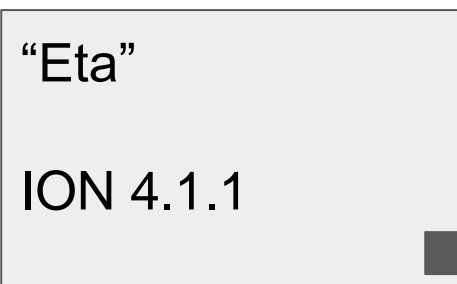


Our Approach

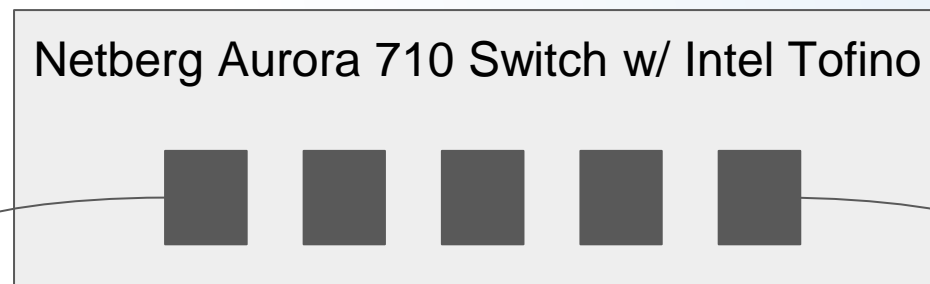
- Utilize P4 data plane programming language to program a networking switch to understand BP
 - Hardware knows nothing: ethernet, IP, TCP/UDP, BP
 - Plan is to write a P4 program and teach it everything it needs to know about BP



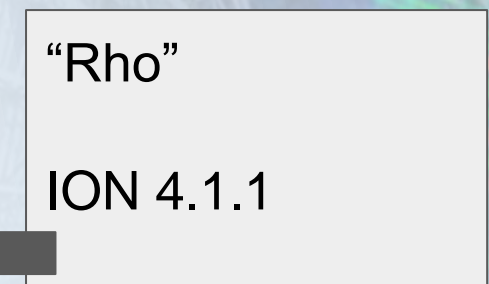
DTN Node



DTN Router and Translator



DTN Node



Difficulties with Bundle Protocol and P4/Tofino

- BP is complex but P4 expects simple protocols
 - BPv6: Self-Delimiting Numeric Values (SDNV)
 - BPv7: Concise Binary Object Representation (CBOR)
- P4 language limitations
 - Limited support for variable-width fields (see Meta4)
 - Parsing stage
 - has recursion
 - can check boolean conditions on parser state transitions
 - Control stage
 - no looping or recursion
 - can check boolean conditions
 - more read/write
- Hardware/Intel Tofino limitations
 - Compiler bugs & inconsistent behavior
 - Limited memory

```
Using SDE_INSTALL /home/dta/sde/bf-sde-9.7.0/install
Using SDE version bf-sde-9.7.0

OS Name: "Open Network Linux OS ONL-master, 2020-08-03.13
This system has 16GB of RAM and 8 CPU(s)
Parallelization: Recommended: -j8 Actual: -j8

Compiling for p4_16/tna
P4 compiler path: /home/dta/sde/bf-sde-9.7.0/install/bin/bf-p4c
P4 compiler version: 9.7.0 (SHA: da5115f) (p4c-based)
Build Dir: /home/dta/sde/bf-sde-9.7.0/build/p4-build/tofino/parse_bpv7
Logs Dir: /home/dta/sde/bf-sde-9.7.0/logs/p4-build/tofino/parse_bpv7

Building parse_bpv7 CLEAR CMAKE MAKE ... FAILED
===== make.log =====
/home/dta/sde/bf-sde-9.7.0/parse_bpv7/parse_bpv7.p4(141): warning: Parser "verify" is currently unsupported
verify(major_type = 4, error.WrongMajorType);
error: invalid SuperCluster was formed: SUPERCLUSTER Uid: 193
slice lists:
[ ingress::meta.debug_metadata.test_overwrite<8> ^0 ^bit[0..575] meta solitary exact_containers [0:4] ]
rotational clusters:
[[ingress::meta.debug_metadata.test_overwrite<8> ^0 ^bit[0..575] meta solitary exact_containers [0:4]]]
because this slice list is not byte-sized: [ ingress::meta.debug_metadata.test_overwrite<8> ^0 ^bit[0..575] meta solitary exact_containers [0:4] ] has 5 bits.
This is either a compiler internal bug or you can introduce padding fields around them by @padding or @flexible
Number of errors exceeded set maximum of 1

CMakeFiles/parse_bpv7-tofino.dir/build.make:60: recipe for target 'parse_bpv7/tofino/bf-rt.json' failed
make[2]: *** [parse_bpv7/tofino/bf-rt.json] Error 3
make[2]: *** Deleting file 'parse_bpv7/tofino/bf-rt.json'
CMakeFiles/Makefile2:132: recipe for target 'CMakeFiles/parse_bpv7-tofino.dir/all' failed
make[1]: *** [CMakeFiles/parse_bpv7-tofino.dir/all] Error 2
Makefile:127: recipe for target 'all' failed
make: *** [all] Error 2
===== make.log =====

ERROR: For the details and to obtain technical support see the file
/home/dta/sde/bf-sde-9.7.0/logs/p4-build/tofino/parse_bpv7/make.log

root@aurora:/home/dta/sde/bf-sde-9.7.0#
```

Current Results: Proof of Concept “BP Translator”

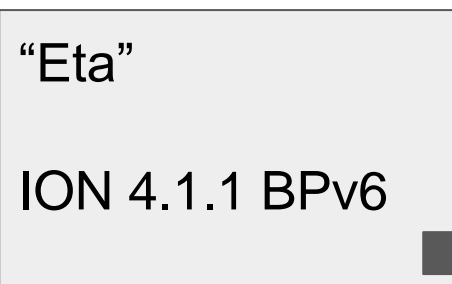
- **BP Translator**

- Able to convert between BPv6 and BPv7 bundles (*with limitations)
- Could be useful for certain backwards compatibility situations (e.g. ISS currently uses BPv6)

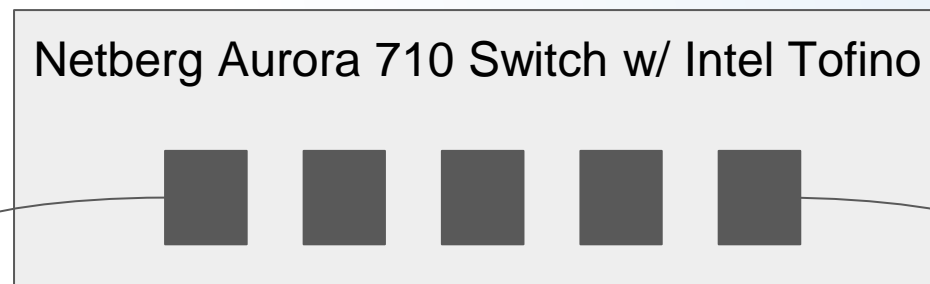
- **Generalizable features (beyond translation):**

- Switch can parse BP fields
- Switch can modify fields at multiple layers (e.g. Ethernet, IP, Transport, Bundle)
- Switch can send BP specific metric data to a controller (local or remote)
- Controller (local or remote) can modify switch behavior as needed (e.g. BP routing rules)

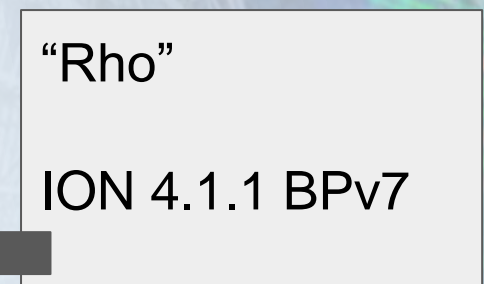
DTN Node



DTN Router and Translator



DTN Node



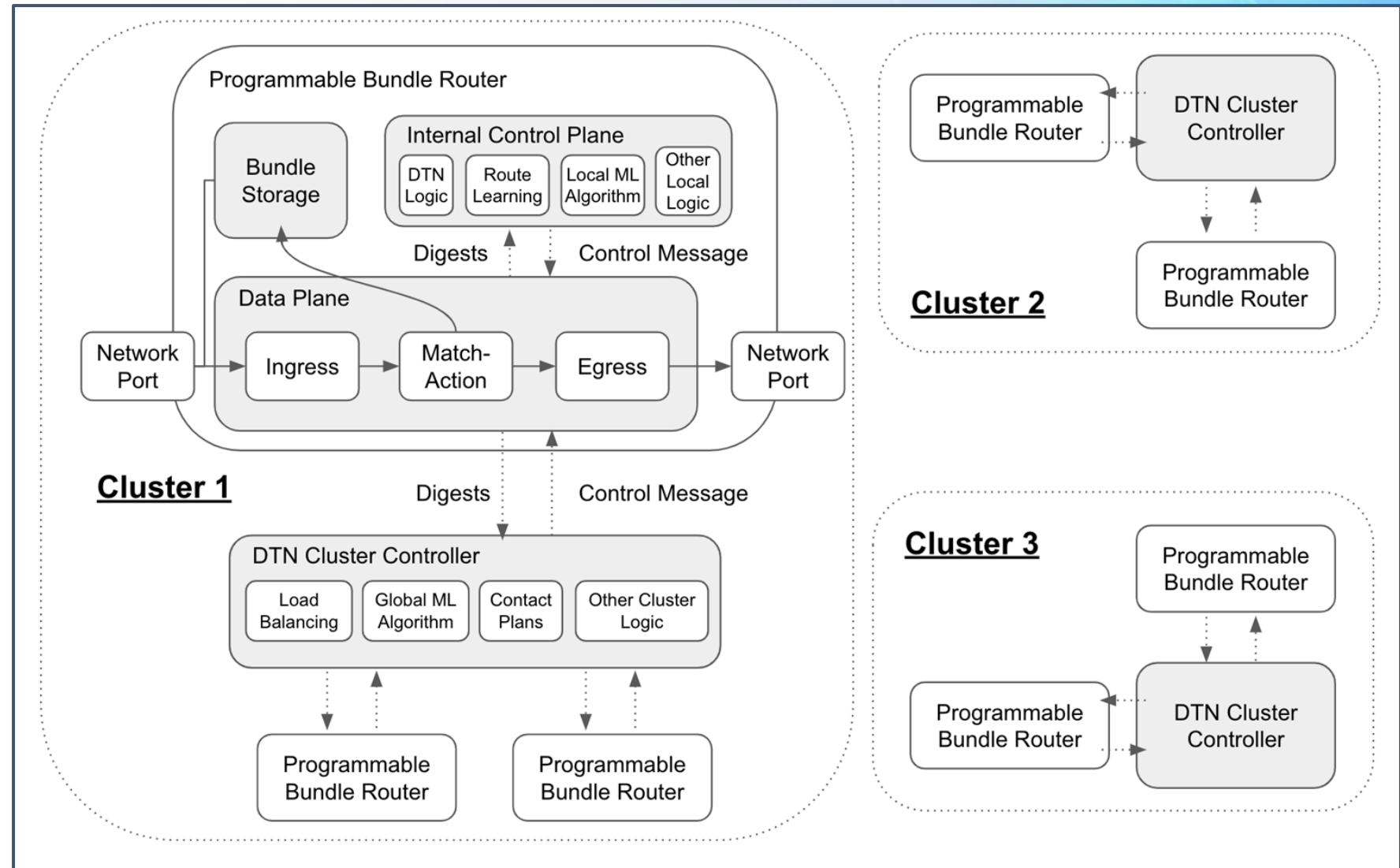
A Proposal: Software-Defined DTN (SDDTN) Architecture

Features:

- DTN network split into clusters
- Each cluster is led by a “DTN Cluster Controller”
- Each bundle node implemented as a programmable router
 - has internal control plane
 - sends telemetry to cluster controller
- Clusters can be dynamic

Goals:

- Ability to manage large DTN networks in a scalable manner
- Nodes can work independently as needed



Conclusion

- Proposed a Software-Defined DTN (SDDTN) Architecture as a way to manage large DTN networks
- Showcased proof of concept that can process BP in hardware
- Detailed challenges of working with BP encoding schemes in hardware

Thanks for listening!

Happy to answer any questions

Contact Info:

domta@cs.washington.edu

