ONF CONNECT
DECEMBER 2018

Leveraging Stratum and Tofino Fast Refresh for Software Upgrades

Antonin Bas
Software Engineer, Barefoot Networks

Copyright 2018 - Barefoot Networks

Agenda

- Introduction to Tofino and programmability
- Synergy between Tofino & Stratum
- Current Tofino support for Stratum
- What is Tofino Fast Refresh and why use it?
- Demo: using Fast Refresh to change the switch role and optimize for latency & power

Copyright 2018 - Barefoot Networks

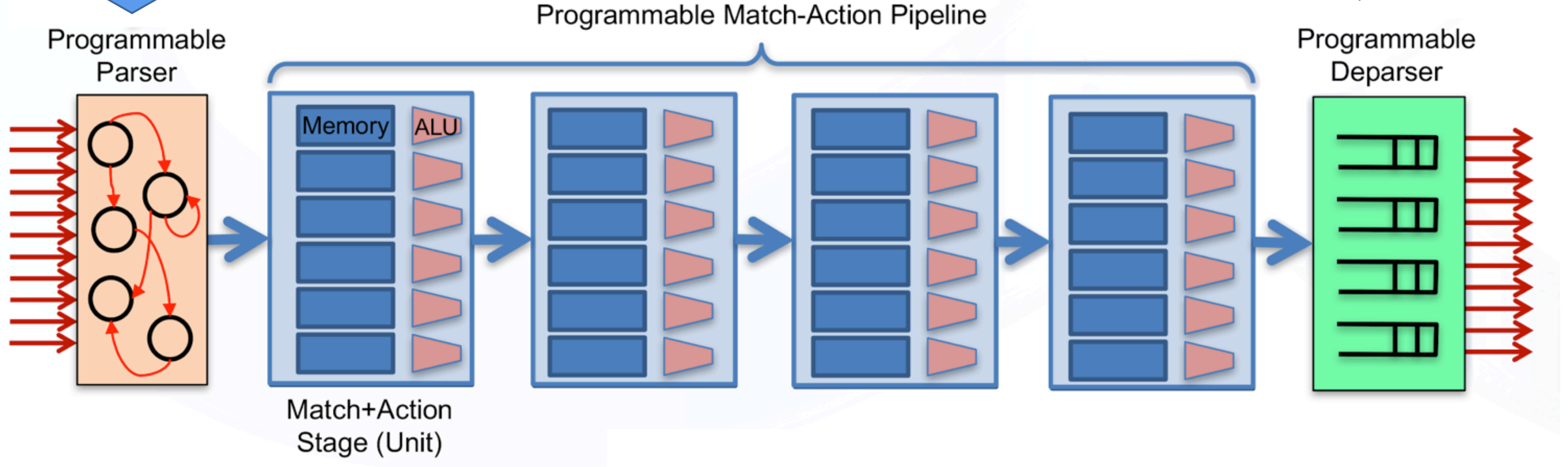
PISA: Protocol Independent Switch Architecture

Abstract machine model of a high-speed programmable switch architecture

Programmer declares which headers should be extracted and in which order

Programmer defines the tables and the exact processing algorithm

Programmer declares how the output packet will look on the wire



Copyright 2018 - Barefoot Networks

BAREFOOT

What is Barefoot Tofino?

- The first end-user programmable high speed Ethernet switch ASIC
- Modeled after the PISA architecture
- P4 programmable
  - Ships with a P4 compiler
  - If it compiles, it runs at line rate
- 65 x 100Gbps and several smaller SKUs
- No compromise on power consumption & speeds compared to fixed-function ASICs
- Integration with several existing Network Operating Systems, including Stratum!

Copyright 2018 - Barefoot Networks

Why data-plane programming?

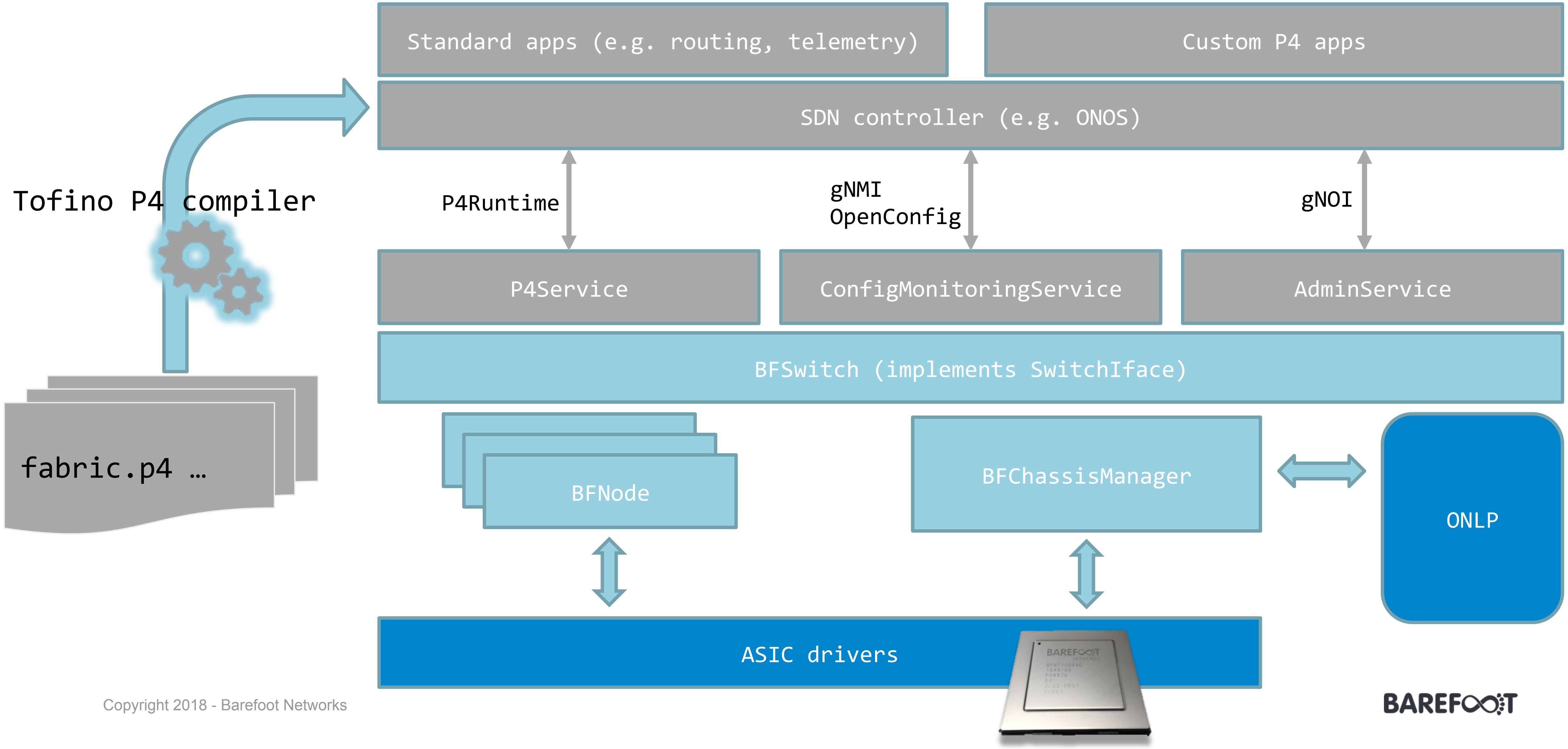
1. **New features**: Realize your beautiful ideas very quickly
2. **Reduce complexity**: Remove unnecessary features and tables
3. **Efficient use of H/W resources**: Achieve biggest bang for buck
4. **Greater visibility**: New diagnostics, telemetry, OAM, etc.
5. **Modularity**: Compose forwarding behavior from libraries
6. **Portability**: Specify forwarding behavior once; compile to many devices
7. **Own your own ideas**: No need to share your ideas with others

"Protocols are being lifted off chips and into software"

– Ben Horowitz

Copyright 2018 - Barefoot Networks

Tofino support in Stratum



Copyright 2018 - Barefoot Networks

BAREFOOT

Current status of Tofino support

- Supported today
  - Most of P4Runtime features
    - Packet IO
    - Match-action programming (direct & indirect)
    - All standard externs (counters, trTCM meters, learning, …) save for stateful registers
  - Port operational status and port stats for gNMI `Set` & `Subscribe`
- Upcoming support (Q1 2019)
  - Port configuration through gNMI
  - P4Runtime stateful register support

Copyright 2018 - Barefoot Networks

Why use Tofino with Stratum?

Tofino is the best fit for Stratum

- Most feature-complete & compliant P4Runtime implementation
  - 18+ months of development
  - First demo @ SDN NFV World Congress – October 2017
  - Support for advanced features such as dynamic reconfiguration and “rollback-on-error” batch semantics
- Tofino’s “native” support for P4 enables high-performance P4Runtime implementation
  - Up to 100,000 new flow rules per second using batching
- Barefoot is an active contributor to Stratum and is committed to keep releasing code and open-sourcing top-level SDK interfaces

Copyright 2018 - Barefoot Networks

What is Tofino Fast Refresh?

A fresh start for your data plane

- Reset your switch state: start from a clean slate (new or same P4)
- Simple 3 step sequence:
  1. Begin: P4 compiler outputs are given to the drivers
  2. Forwarding state is built / replayed through usual API calls
  3. End: Drivers are told to perform a core reset and all memories (including forwarding state) are written through batched DMA
- **Minimal traffic interruption during step 3: < 50 ms for *any* P4 program and *any* set of flow rules**
- Facilitated by a new generation of **program-independent APIs**: P4Runtime, Barefoot Runtime Interface (BRI)
- Can be leveraged by all Network Operating Systems!

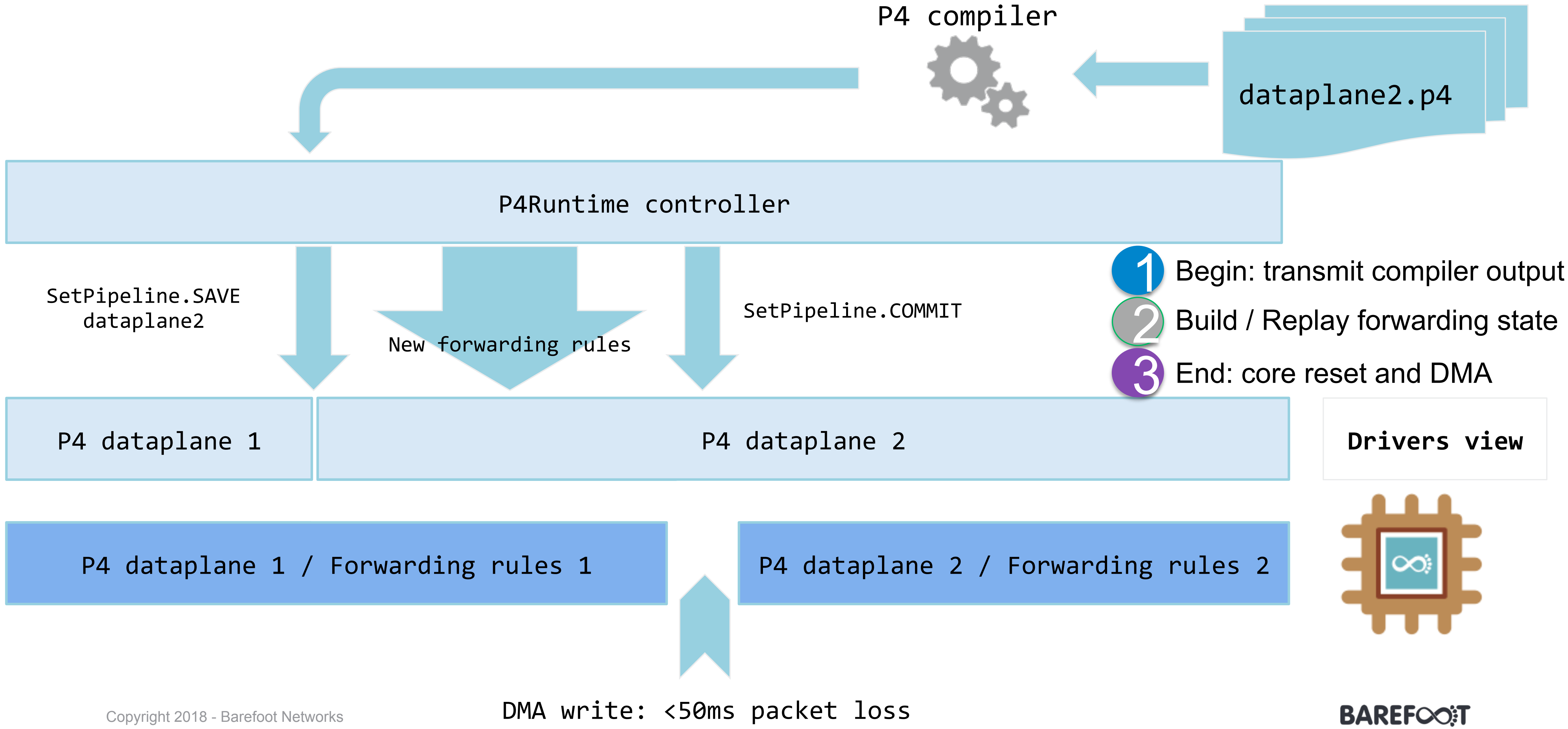
Copyright 2018 - Barefoot Networks

Why use Tofino Fast Refresh?

- Unified and resilient mechanism to **upgrade software**
  - Upgrade data plane, driver stack, control plane or even Linux OS
  - “Bug-free” upgrade: data plane & forwarding state re-built from scratch
- Use for **scheduled maintenance** & to solve mysterious data plane issues: just refresh it!
- Also use to **change the role** of the switch by reconfiguring it to use a new P4 program (*aka* “Fast Reconfig”).
  - Optimize your data plane for a specific feature set
  - Optimize your data plane for low latency or power consumption
- Support multiple data plane profiles and upgrade scenarios in your NOS!

Copyright 2018 - Barefoot Networks

Fast Refresh with P4Runtime


P4 compiler
dataplane2.p4
P4Runtime controller
SetPipeline.SAVE dataplane2
New forwarding rules
SetPipeline.COMMIT
P4 dataplane 1
P4 dataplane 2
Drivers view
P4 dataplane 1 / Forwarding rules 1
P4 dataplane 2 / Forwarding rules 2
DMA write: <50ms packet loss

1. Begin: transmit compiler output
2. Build / Replay forwarding state
3. End: core reset and DMA

Copyright 2018 - Barefoot Networks

Demo: Power and latency saving by reducing complexity

Using Fast Refresh to run 4 different P4 programs on Tofino

P4 program	P4 architecture	# MA entries	Description
I) fabric-spgw.p4 (ONF)	PSA	262,396	ONF's fabric.p4 with SPGW-u offload and PCC gating • 120K on-chip subscriber connections • 4K arbitrary IPv4 prefix routes • 100K IPv4 host routes • …
II) fabric.p4 (ONF)	PSA	113,824	ONF's fabric.p4 without SPGW-u offload • 4K arbitrary IPv4 prefix routes • 100K IPv4 host routes • …
III) L3.p4	TNA	277,824	Simple L3 IPv4 forwarding • 12K arbitrary prefix routes • 200K host routes • 65K next hops
IV) L3_heavy.p4	TNA	1,343,744	Heavy L3 IPv4 forwarding • **1M+ host, /28, /24, /20, /16, /8 routes**

Copyright 2018 - Barefoot Networks

BAREFOOT

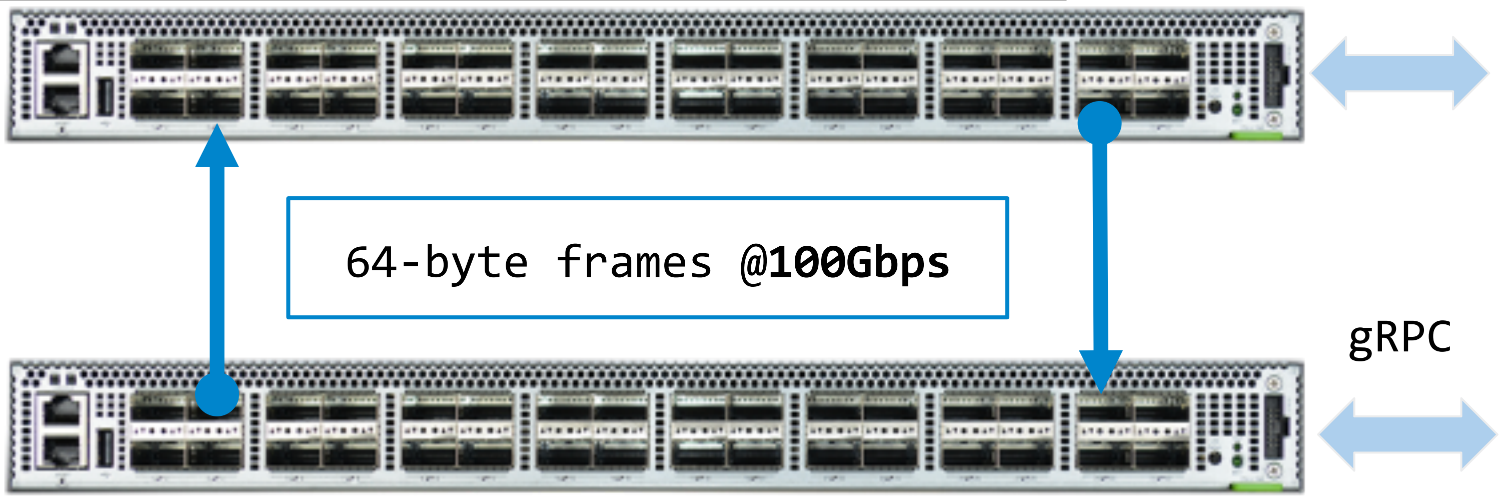
Fast Refresh Demo Setup

Tofino 3.2T Switch Under Test

Stratum NOS

Running the following P4s with static flow rules:

- L3.p4
- L3_heavy.p4
- ONF’s fabric.p4
- ONF’s fabric.p4 with SPGW-u & PCC gating



update_config.py

Performs Fast Refresh
Replays flow rules

PktGen CLI

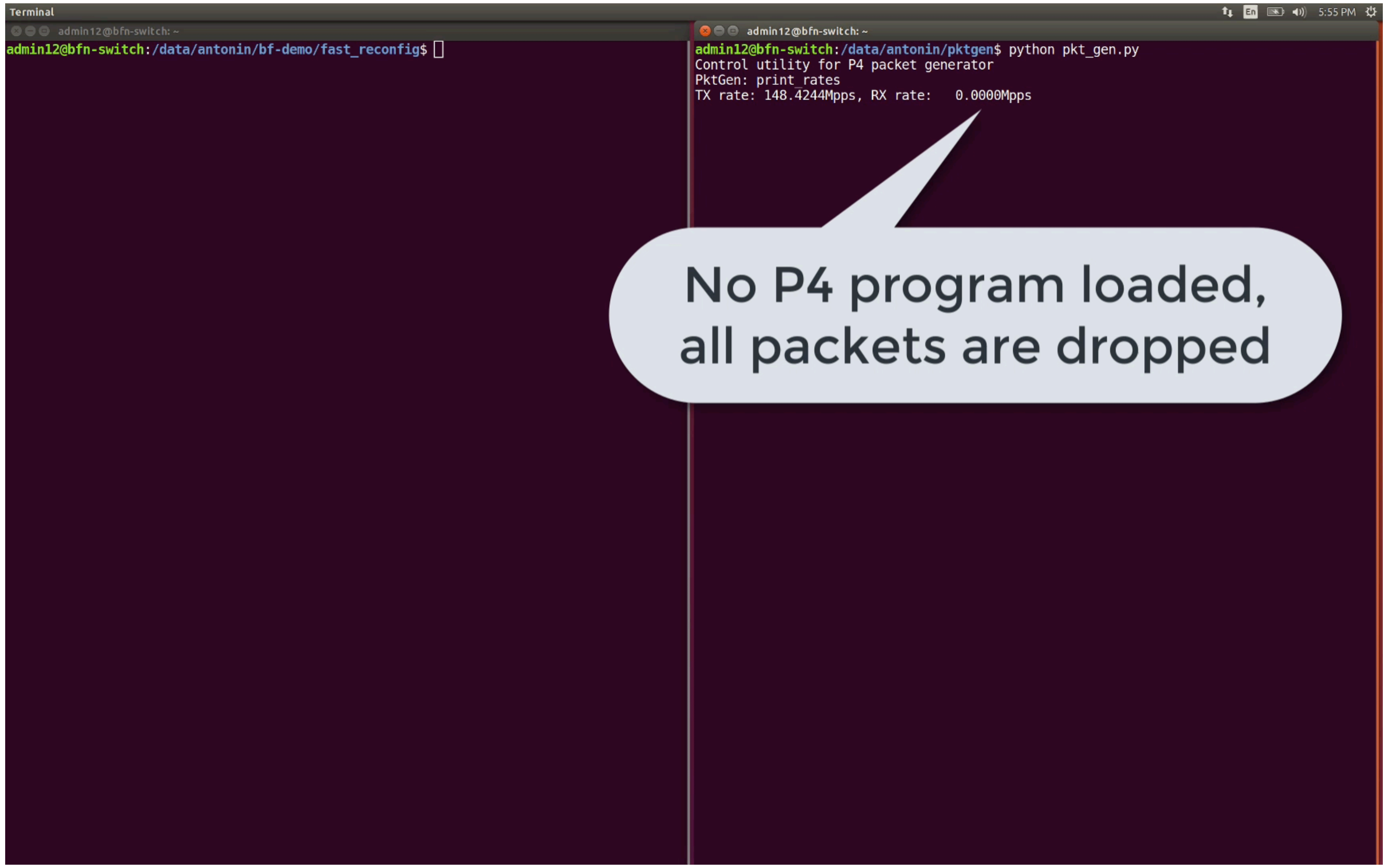
Displays rates, latency and packet drops

Tofino 3.2T as packet generator

Generates line rate packet stream
Computes packet drops through MAC counters
Computes real-time latency of stream

Copyright 2018 - Barefoot Networks

Demo video



```
admin12@bfn-switch:/data/antonin/bf-demo/fast_reconfig$
```

```
admin12@bfn-switch:/data/antonin/pktgen$ python pkt_gen.py
Control utility for P4 packet generator
PktGen: print rates
TX rate: 148.4244Mpps, RX rate:   0.0000Mpps
```

No P4 program loaded, all packets are dropped

Copyright 2018 - Barefoot Networks

Demo: Power and latency saving by reducing complexity

Demo results

P4 program	# MA entries	Measured latency	Estimated worst-case power usage (MA pipeline only)	Packet drop during Fast Refresh
I) fabric-spgw.p4 (ONF)	262,396	681 ns	53.1%	
II) fabric.p4 (ONF)	113,824	644 ns	27.8% 47.7% savings compared to I)	< 31 ms
III) L3.p4	277,824	370 ns	9.6%	< 31 ms
IV) L3_heavy.p4	1,343,744	365 ns	17.8%	< 31 ms

Copyright 2018 - Barefoot Networks

Takeways

- Use Fast Refresh on Tofino to update P4 programs, upgrade software and reconcile state with minimum traffic interruption
  - And stay within SLA!
- Change your P4 program without modifying any x86 code on the switch thanks to program-independent APIs (P4Runtime, BRI)
- Use Fast Refresh in Stratum, SONIC, …
- Optimize your program for specific features, or for latency / power
- Power of programmability: use Tofino as a packet generator to evaluate another switch!

Copyright 2018 - Barefoot Networks

BAREFOOT

Thank You

Antonin Bas
antonin@barefootnetworks.com

Copyright 2018 - Barefoot Networks