OCT 2018

P4 based Programmable Data Plane

Haitao Kang
CUSTOMER ENGINEER

BAREFOOT

P4 and Data Plane Programming

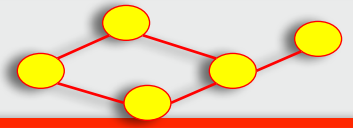
Software Defined Network (SDN)

Control Program

Control Program

Control Program

Centralized Network State



Control Plane

Packet

Packet Forwarding

Packet Forwarding

A network is a means to forward packets from one place to another, and modify them along the way.

If we cannot control how packets are forwarded and processed, we are not really in control.

"Software is eating the network"

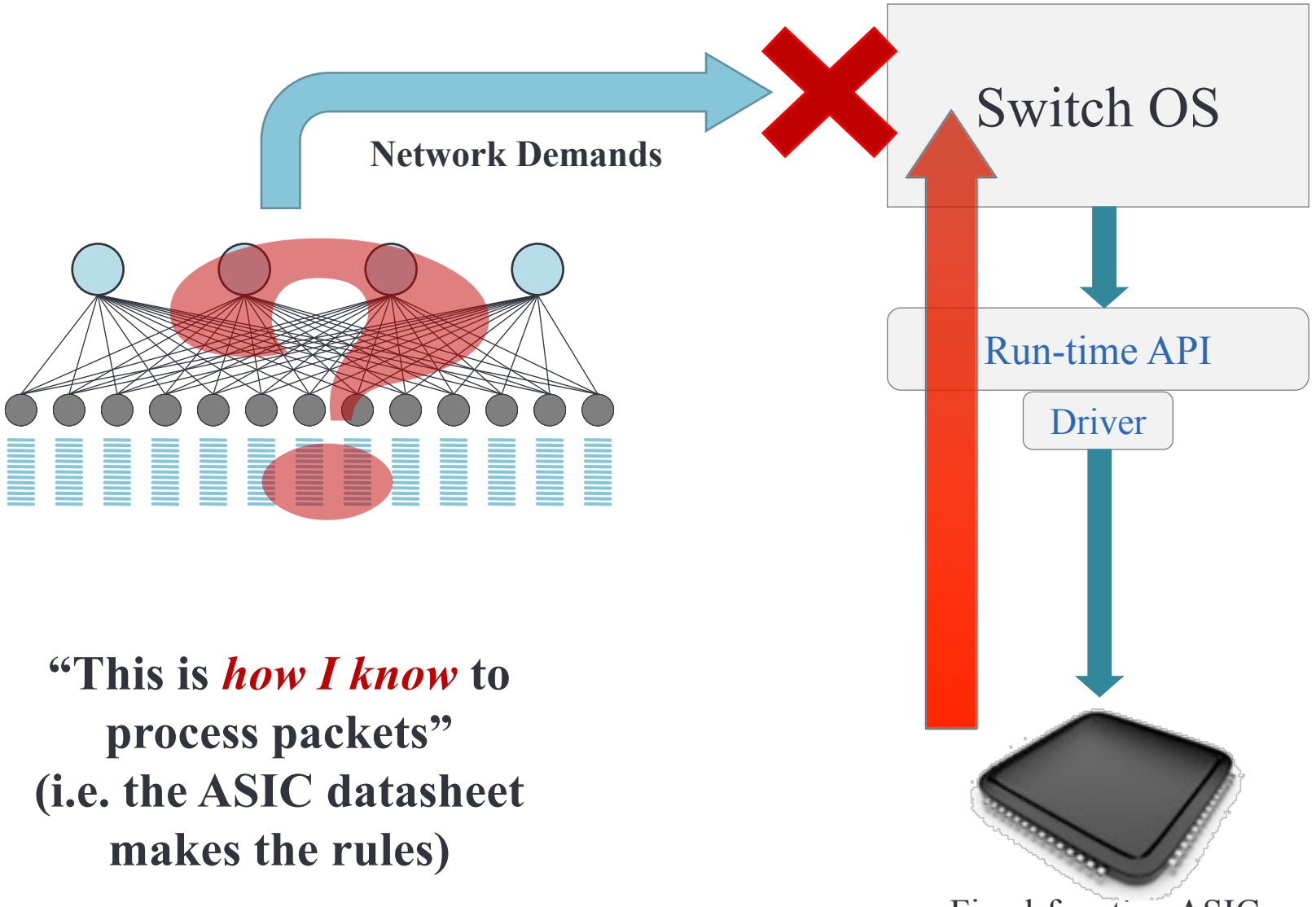
ONOS, ODL, Ryu

CORD, NSX, ONAP

Stagnation & No innovation

BAREFOOT

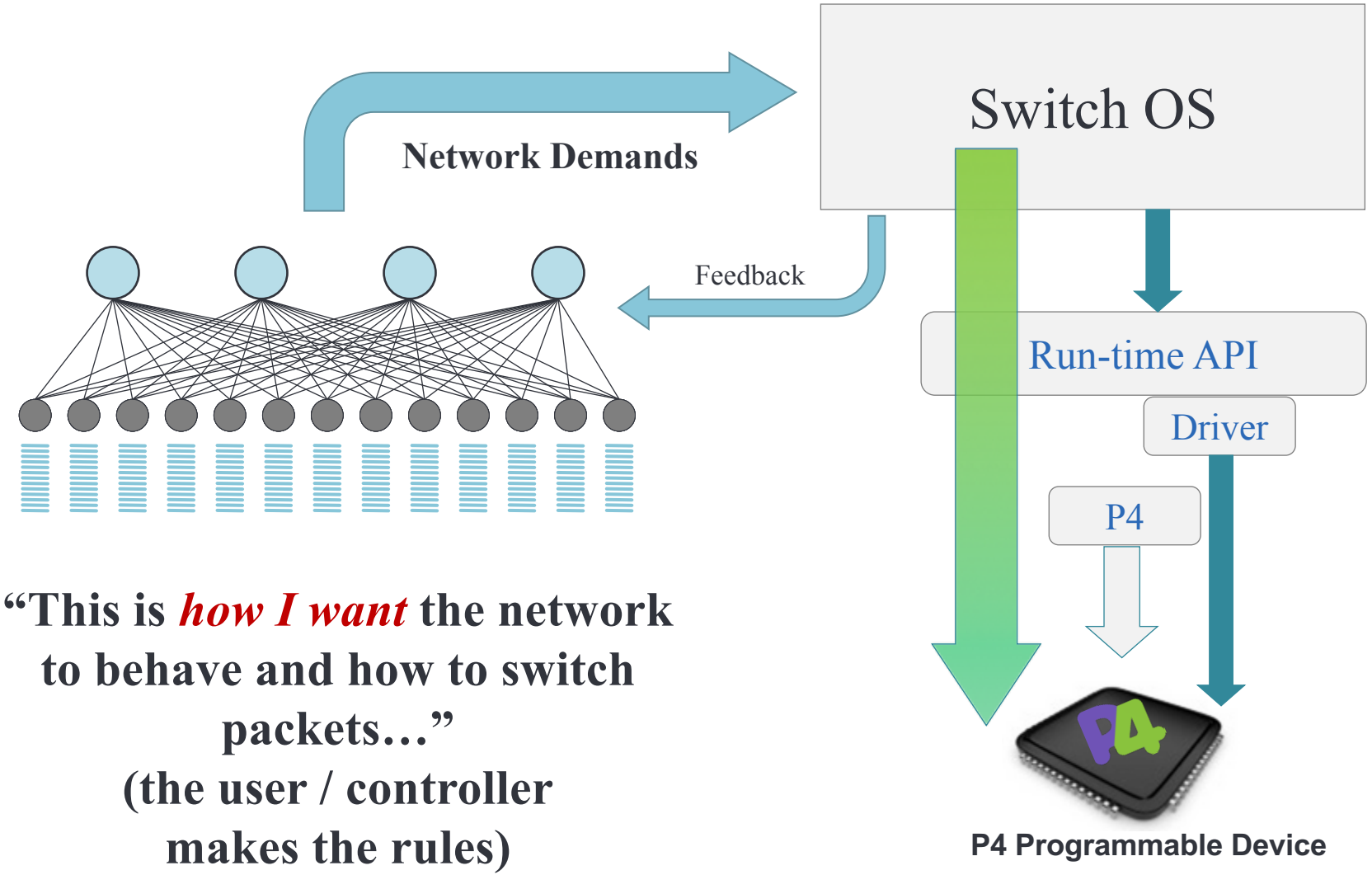
Bottoms-up network element design



"This is *how I know* to process packets" (i.e. the ASIC datasheet makes the rules)

BAREFOOT

Top-down network element design



"This is *how I want* the network to behave and how to switch packets…"
(the user / controller makes the rules)

```
table routing {
  key = { ipv4.dstAddr : lpm; }
  actions = { drop; route; }
  size : 2048;
}
control ingress() {
  apply {
    routing.apply();
  }
}
```

"P4: Programming Protocol-Independent Packet Processors"

BAREFOOT

P4 Community – Growing Momentum

Independent Consortium
Free to join
Apache 2.0 License

~**1500** developers
~ **5000** commits
~**1500** followers
~ **800** forks

~ **200** contributors
~ **30** Repositories
~ **12** teams
~ Multiple targets

~ **100** Industry and Academia Members
~ **4** Working Groups
~ **4** Bi-weekly face-to-face meetings
~ **8** Mailing Lists

BAREFOOT

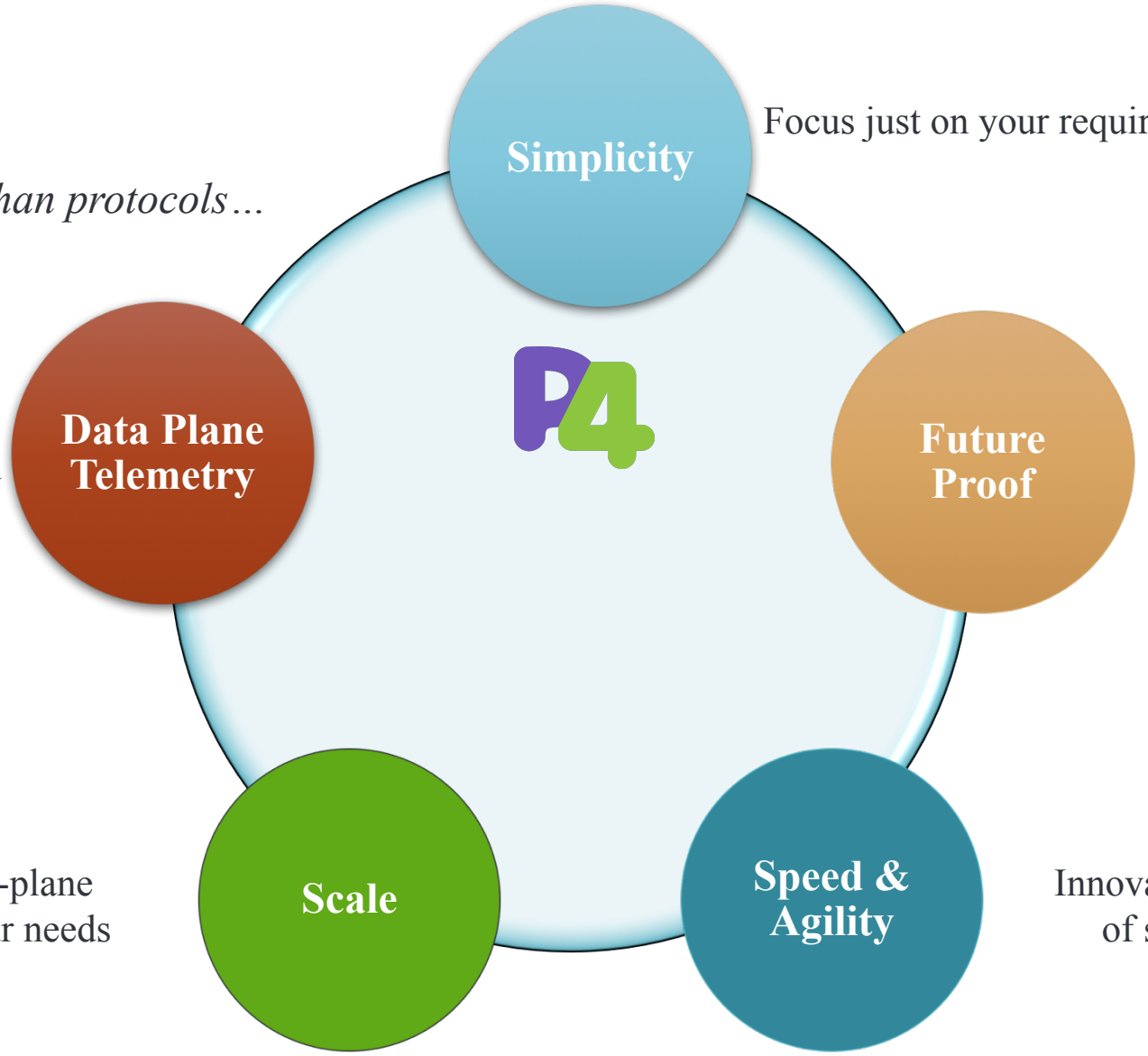
Programmable Network Devices

- PISA: Flexible Match+Action ASICs
  - Barefoot Tofino, Intel Flexpipe, Cisco Doppler, Cavium (Xpliant), …
- NPU
  - EZchip, Netronome, …
- CPU
  - Open Vswitch, eBPF, DPDK, VPP…
- FPGA
  - Xilinx, Altera, …

These devices let us tell them how to process packets.

Benefits of Data Plane Programmability

Think programming rather than protocols…



Focus just on your requirements

Monitor with Scalable, Programmable, Real-Time, Inband Network Telemetry (INT)

Continuously deliver features with no rip and replace

Maximize data-plane resources to your needs

Innovate at speed of software

BAREFOOT

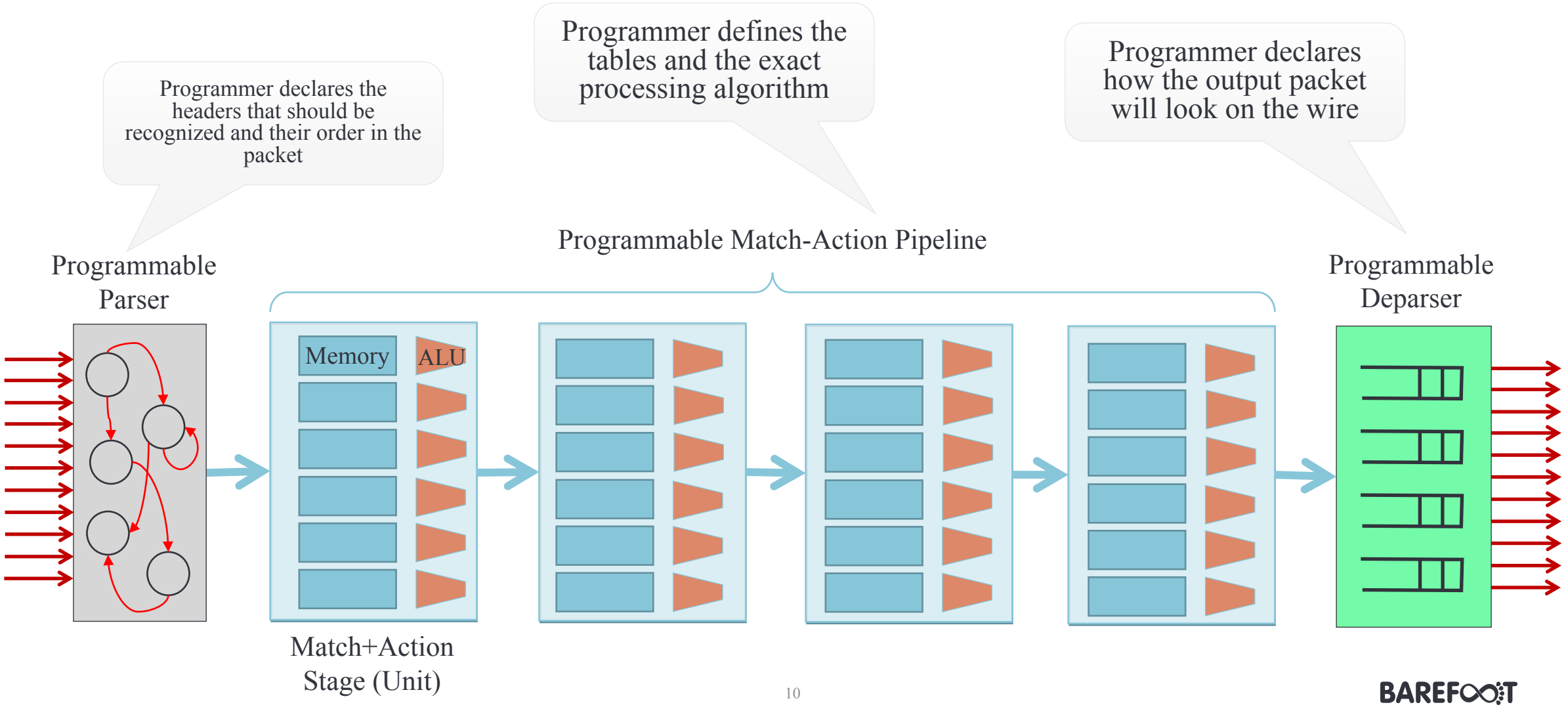
What can you do with P4?

- In-band Network Telemetry – INT[1]
- Low Latency Congestion Control – NDP[2]
- Layer 4 Load Balancer – SilkRoad[3]
- Fast In-Network cache for key-value stores – NetCache[4]
- Consensus at network speed – NetPaxos[5]
- Aggregation for MapReduce Applications [6]

[1] Kim, Changhoon, et al. "In-band network telemetry via programmable dataplanes." SIGCOMM. 2015.
[2] Handley, Mark, et al. "Re-architecting datacenter networks and stacks for low latency and high performance." SIGCOMM, 2017.
[3] Miao, Rui, et al. "SilkRoad: Making Stateful Layer-4 Load Balancing Fast and Cheap Using Switching ASICs." SIGCOMM, 2017.
[4] Xin Jin et al. "NetCache: Balancing Key-Value Stores with Fast In-Network Caching." To appear at SOSP 2017
[5] Dang, Huynh Tu, et al. "NetPaxos: Consensus at network speed." SIGCOMM, 2015.
[6] Sapio, Amedeo, et al. "In-Network Computation is a Dumb Idea Whose Time Has Come." *Hot Topics in Networks*. ACM, 2017.

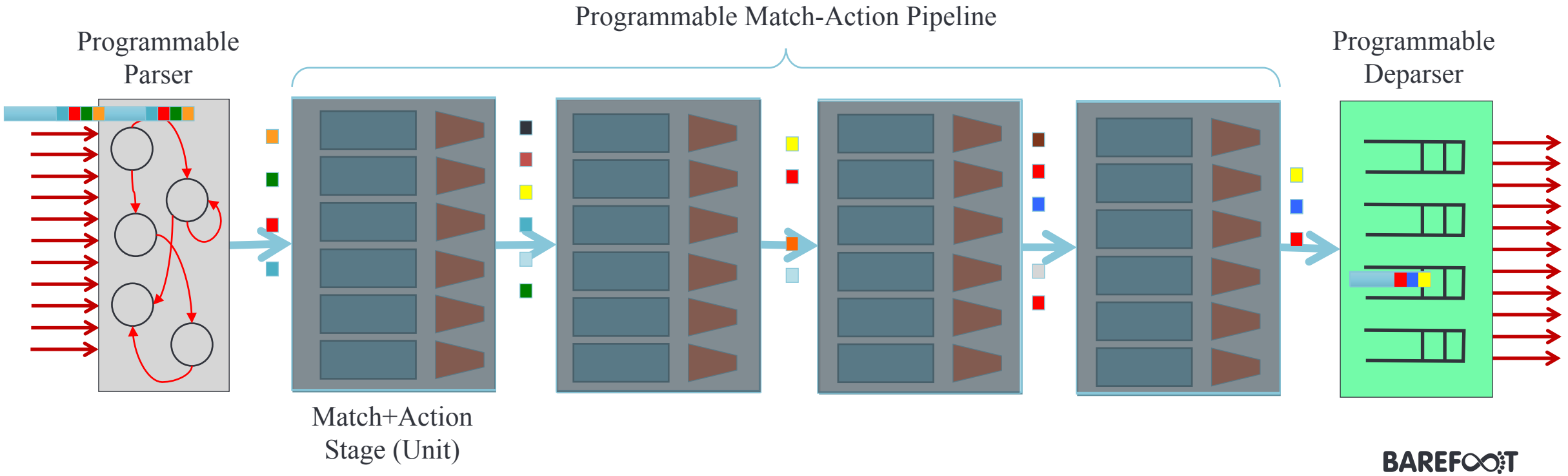
BAREFOOT

PISA: Protocol-Independent Switch Architecture

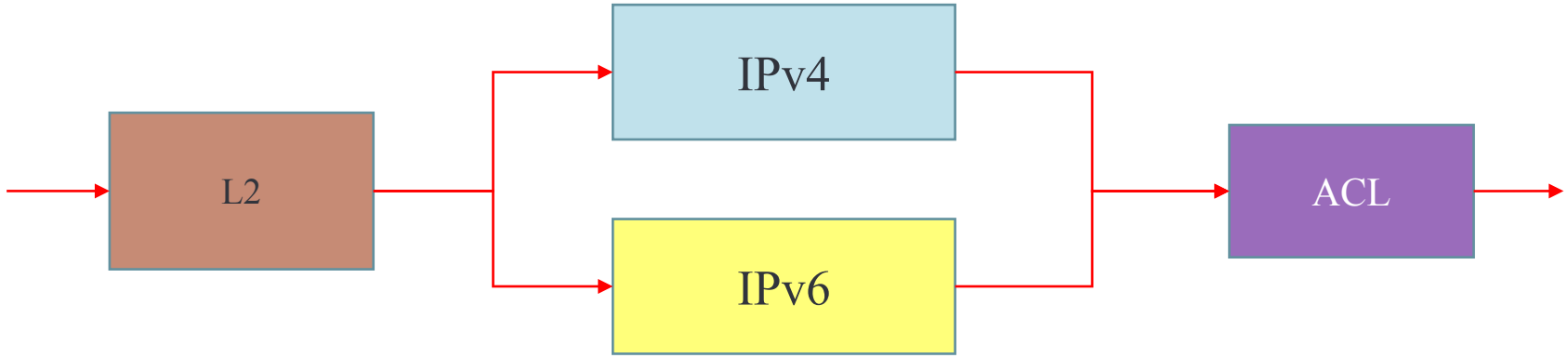


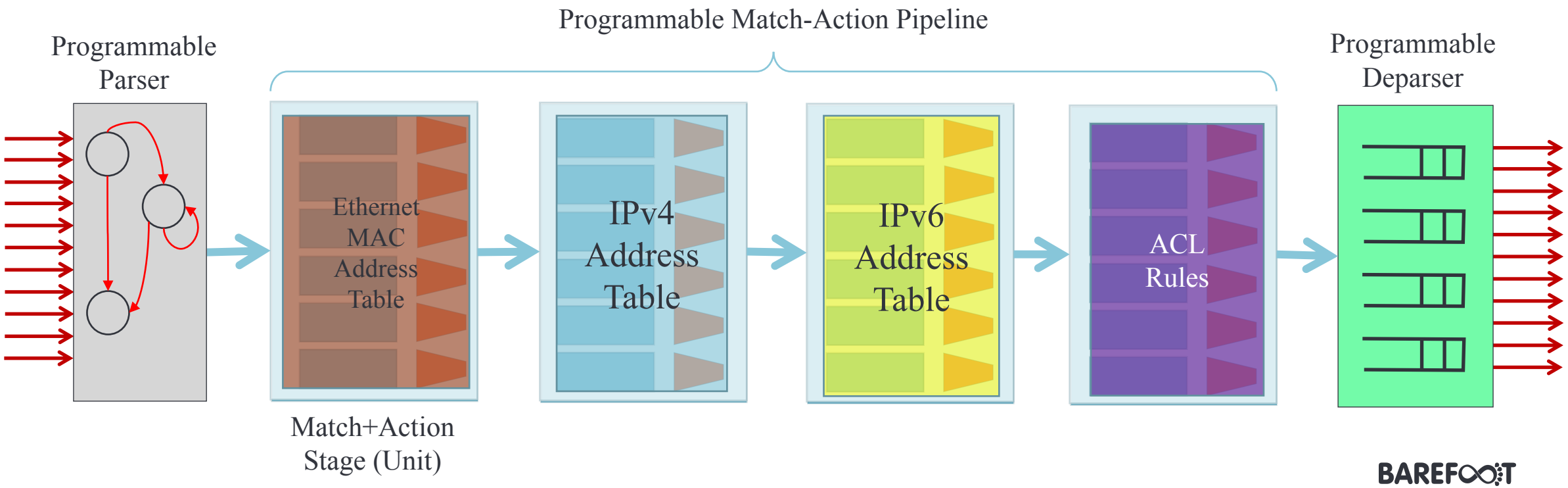
PISA in Action

- Packet is parsed into individual headers (parsed representation)
- Headers and intermediate results can be used for matching and actions
- Headers can be modified, added or removed
- Packet is deparsed (serialized)



Mapping a Simple L3 Data Plane Program on PISA





Example P4 Program

Parser Program

```
parser parse_ethernet {
    extract(ethernet);
    return switch(ethernet.ethertype) {
        0x8100 : parse_vlan_tag;
        0x0800 : parse_ipv4;
        0x8847 : parse_mpls;
        default: ingress;
}
```

Header and Data Declarations

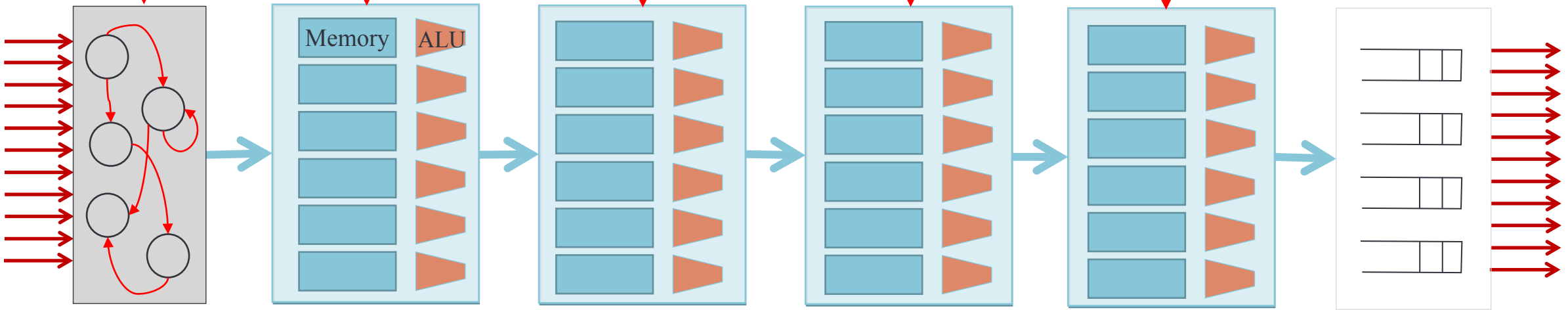
```
header_type  ethernet_t     { … }
header_type  l2_metadata_t { … }

header     ethernet_t     ethernet;
header     vlan_tag_t
vlan_tag[2];
metadata   l2_metadata_t l2_meta;
```

Tables and Control Flow

```
table port_table { … }

control ingress {
    apply(port_table);
    if (l2_meta.vlan_tags == 0) {
        process_assign_vlan();
    }
}
```



13

Barefoot Tofino and Applications

"Programmable switches are 10-100x slower than fixed-function switches. They cost more and consume more power."

CONVENTIONAL WISDOM IN NETWORKING

BAREFOOT

Barefoot Tofino – Programmability & Performance

6.5 Tb/s

3.3 Tb/s

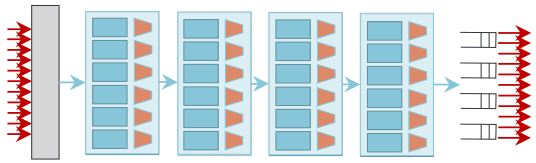
2.5 Tb/s

2.1 Tb/s

1.9 Tb/s

The world's **fastest** and most **programmable** Ethernet switch ASIC family.

Open Source **P4** Programming Language

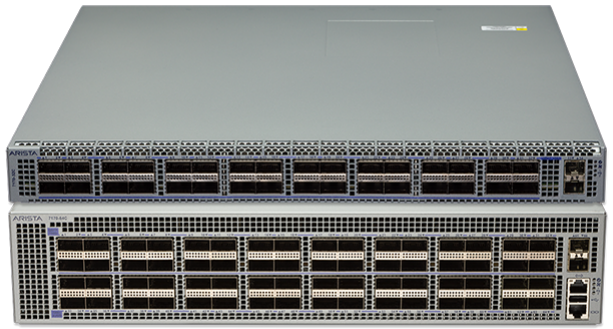


Open **PISA** Target Architecture

BAREFOOT

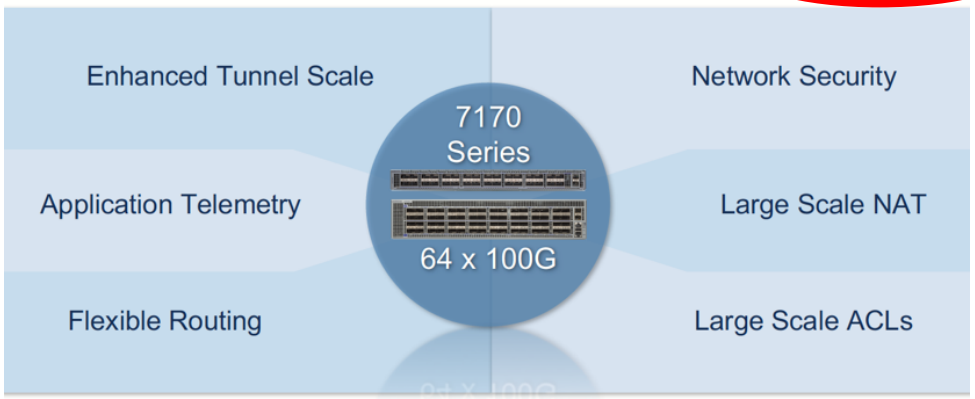
ARISTA

Arista 7170 Series of Multi-function Programmable Platforms



1U 7170-32C
32 ports of 100G
2.5B pkts/s. Multiple profiles.

2U 7170-64C
64 ports of 100G
5B pkts/s. Multiple profiles.



CISCO

Cisco Nexus® 34180YC programmable switch

1U Cisco Nexus® 34180YC programmable switch. High-speed, low-power, high-density data center switch. 48 ports of 10/25G and 6 ports of 100G. Multiple profiles.

BAREFOOT

Data-Plane Telemetry

THE NETWORK SHOULD ANSWER THESE QUESTIONS

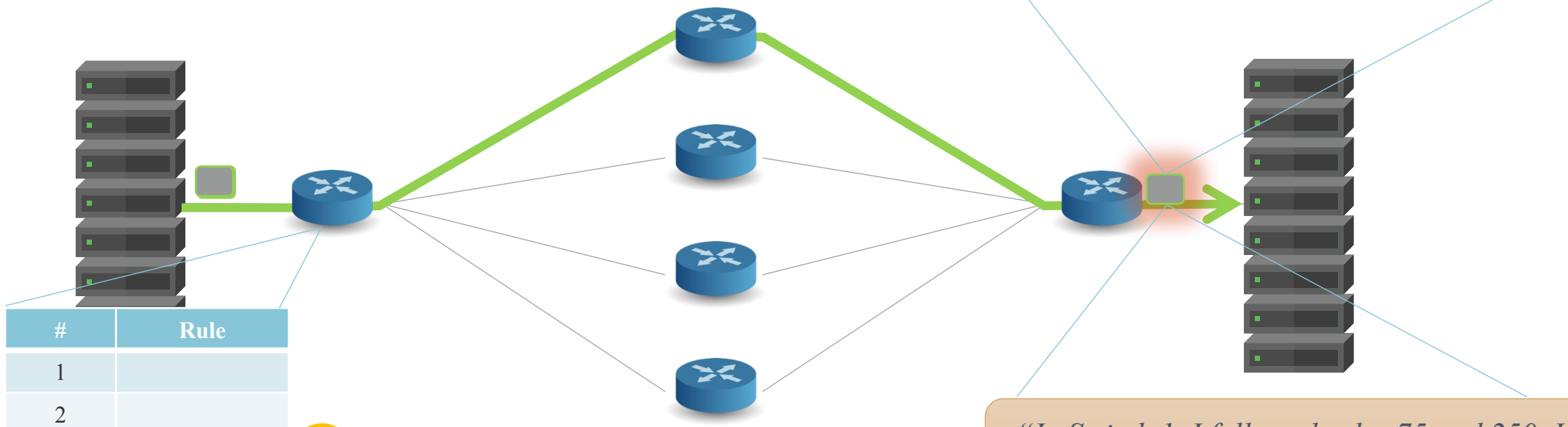
1. *"Which path did my packet take?"*
2. *"Which rules did my packet follow?"*
3. *"How long did it queue at each switch?"*
4. *"Who did it share the queues with?"*

Tofino + Deep Insight can answer all four questions. At full line rate. Without generating any additional packets!

BAREFOOT

1 *"Which path did my packet take?"*

"I visited Switch 1 @780ns, Switch 9 @1.3µs, Switch 12 @2.4µs"



#	Rule
1	
2	
3	
…	
75	192.168.0/24
…	

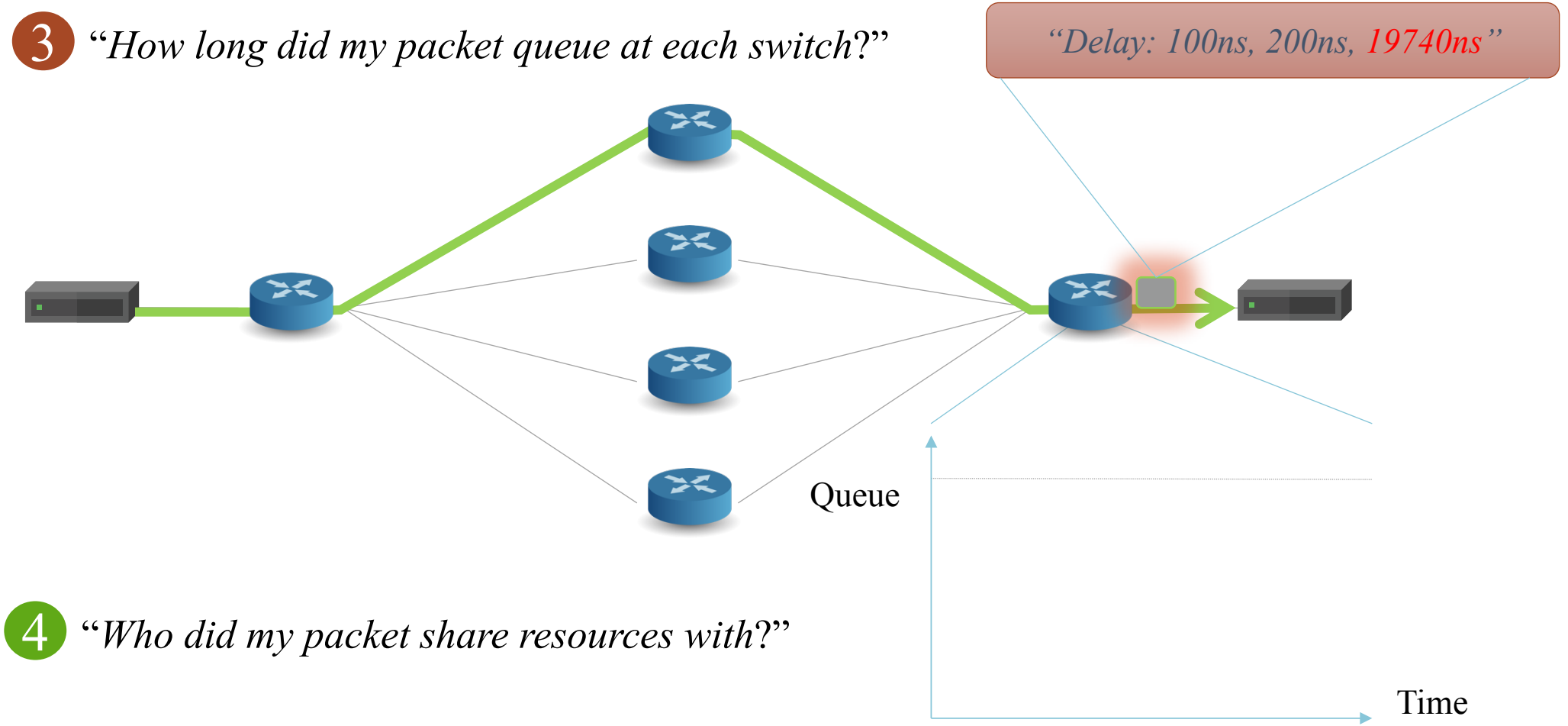
2 *"Which rules did my packet follow?"*

"In Switch 1, I followed rules 75 and 250. In Switch 9, I followed rules 3 and 80. "

19

BAREFOOT

3 *"How long did my packet queue at each switch?"*



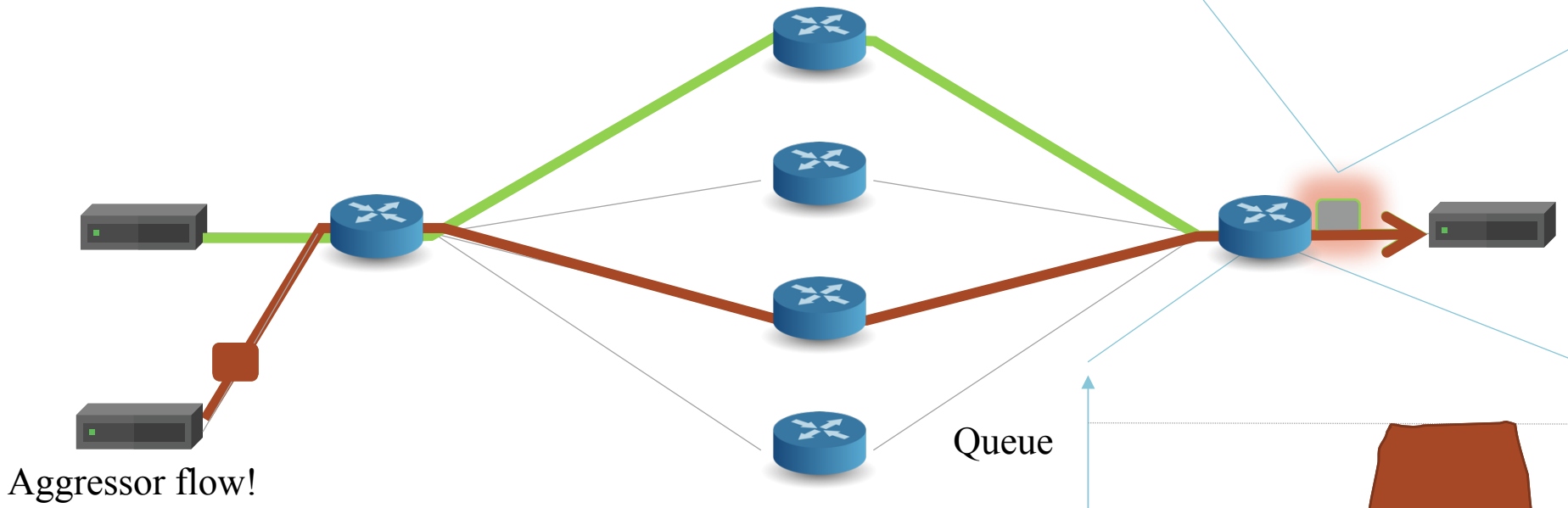
4 *"Who did my packet share resources with?"*

20

BAREFOOT

3 *"How long did my packet queue at each switch?"*

"Delay: 100ns, 200ns, 19740ns"



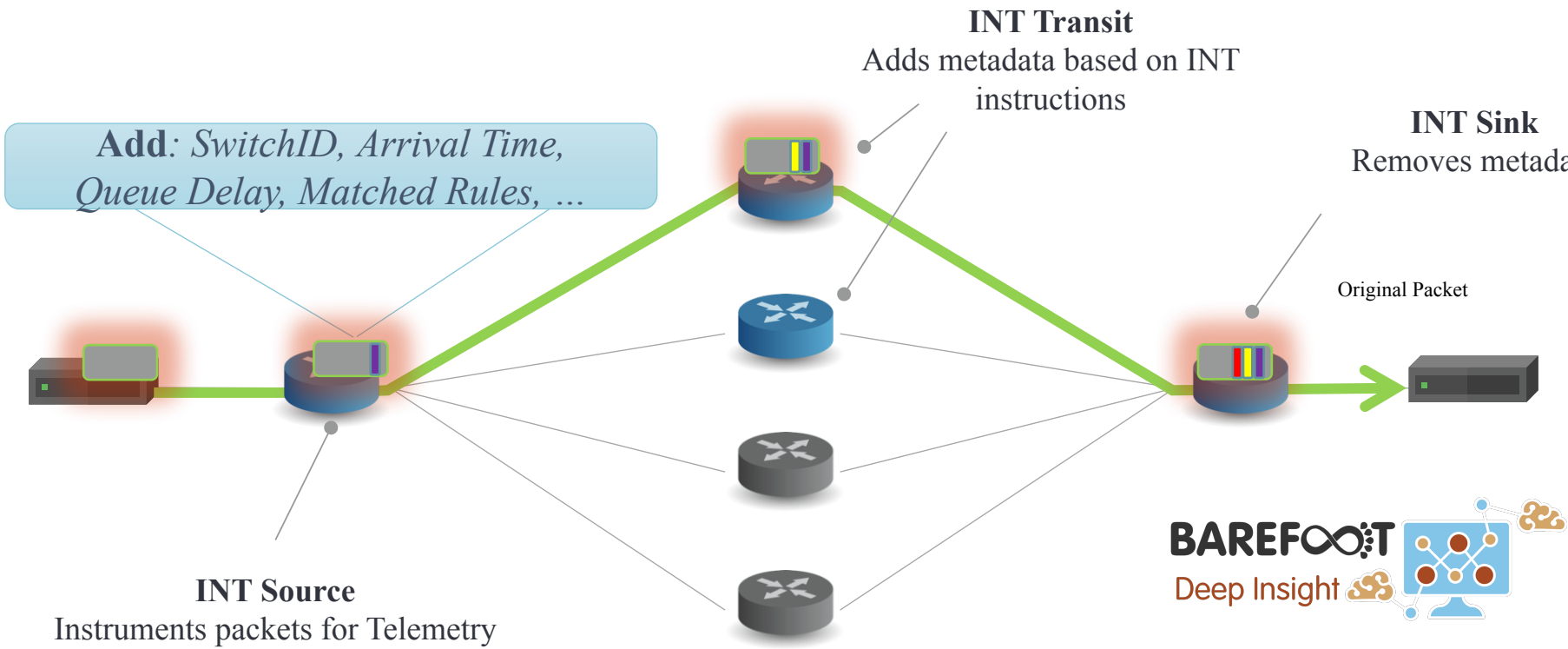
4 *"Who did my packet share resources with?"*

21

BAREFOOT

How it works and how we use the data

Leverages In-Band Network Telemetry (INT)
https://github.com/p4lang/p4-applications/tree/master/telemetry/specs



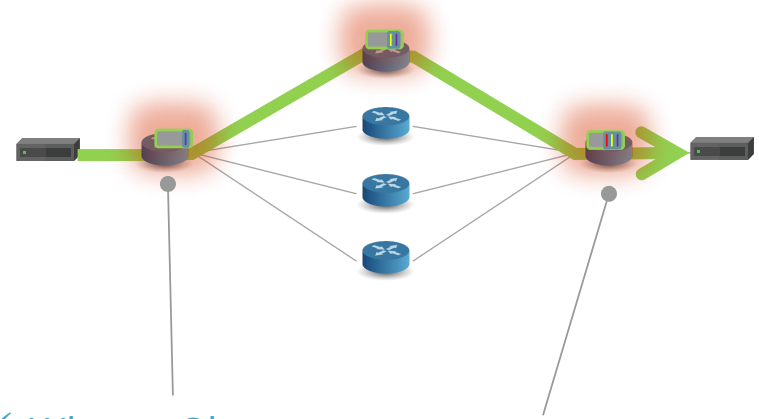
22

BAREFOOT

SPRINT: A Fully Featured, High-Performance INT

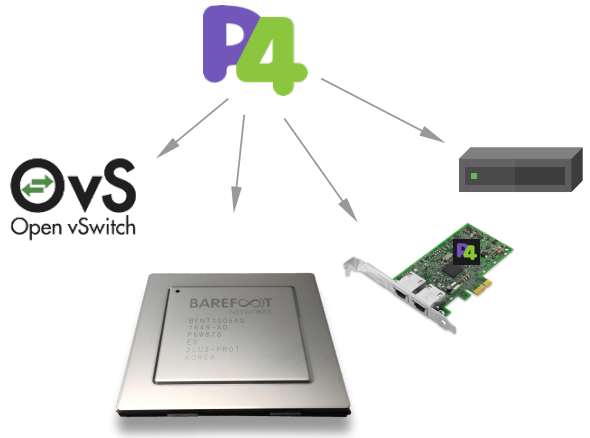
FULLY COMPATIBLE SUPERSET OF A VANILLA INT IMPLEMENTATION

S
Smart



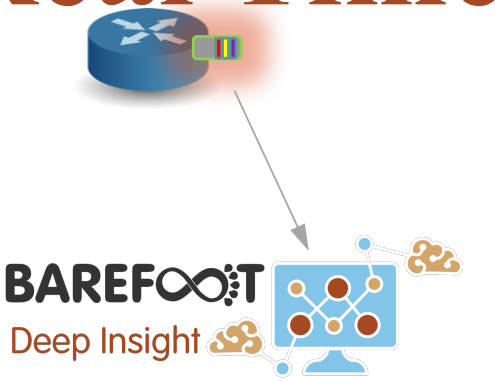
- ✓ What to Observe
- ✓ What to Collect
- ✓ Intelligent Triggers
- ✓ Built-in Load Balancing

P
Programmable



- ✓ Adapt to customers requirements
- ✓ Flexible encapsulation through P4
- ✓ Open specifications and ecosystem

R
Real Time



- ✓ Data-plane Streaming
- ✓ Packet-by-packet Anomaly detection
- ✓ Real time Analytics with Deep Insight

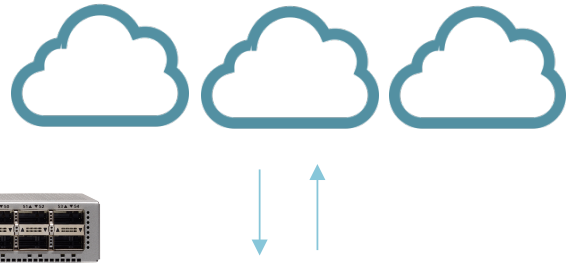
BAREFOOT

Barefoot Deep Insight Analytics

Barefoot SPRINT Data-Plane Telemetry

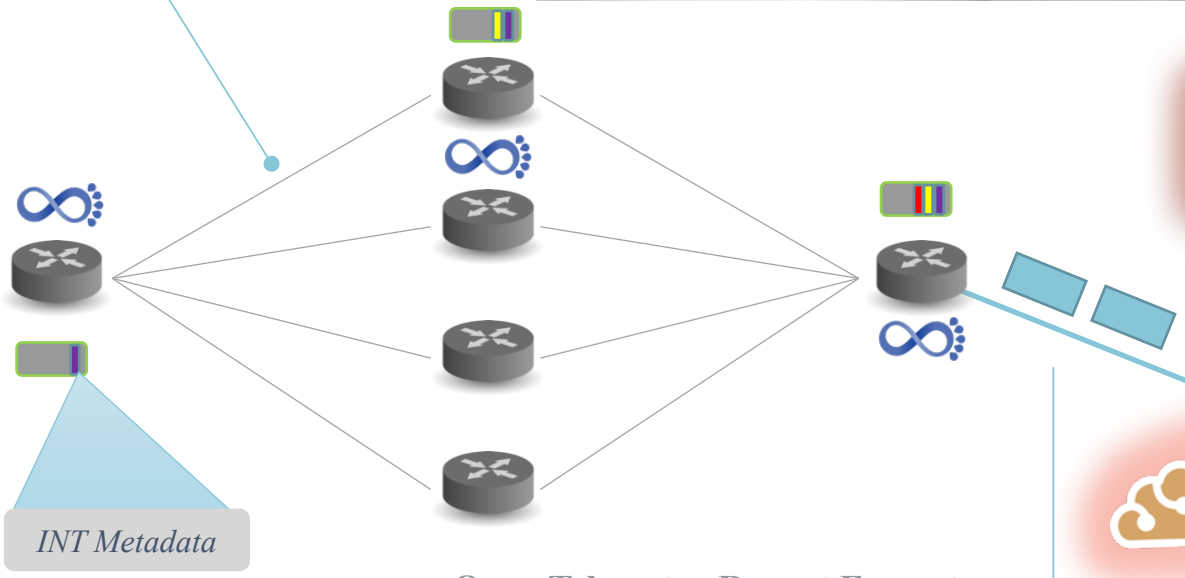
- In-Band Network Telemetry (**INT.P4**)
- Intelligent Deduplication and Triggers
- Hardware Primitives
- Line Rate Monitoring

3rd Party Network Management Solutions

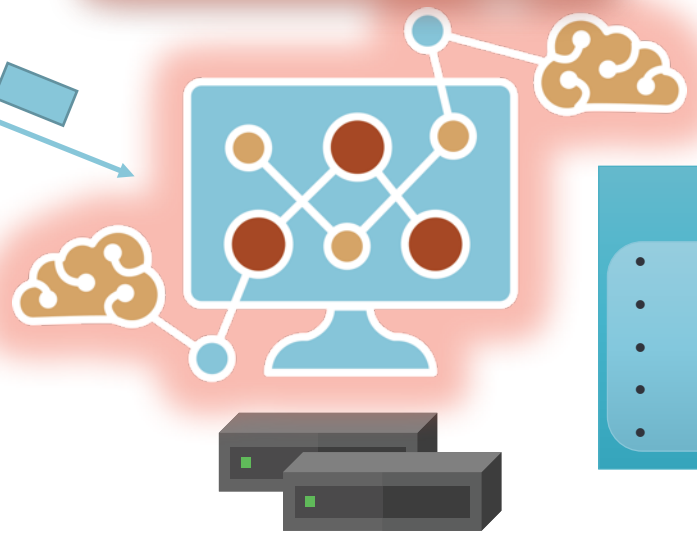


Answer for **Every** Packet...

1. How did it get here?
2. Why is it here?
3. How long was it delayed?
4. Why was it delayed?



Deep Insight
Open Northbound APIs



Deep Insight Analytics Software

- Real-time Anomaly Detection
- Rich Analytics
- Dashboards and Drill down workflows
- Modular Architecture for scale-out
- Runs on Commodity Servers

Open Telemetry Report Format
defined by the **P4.org**
Applications Working Group

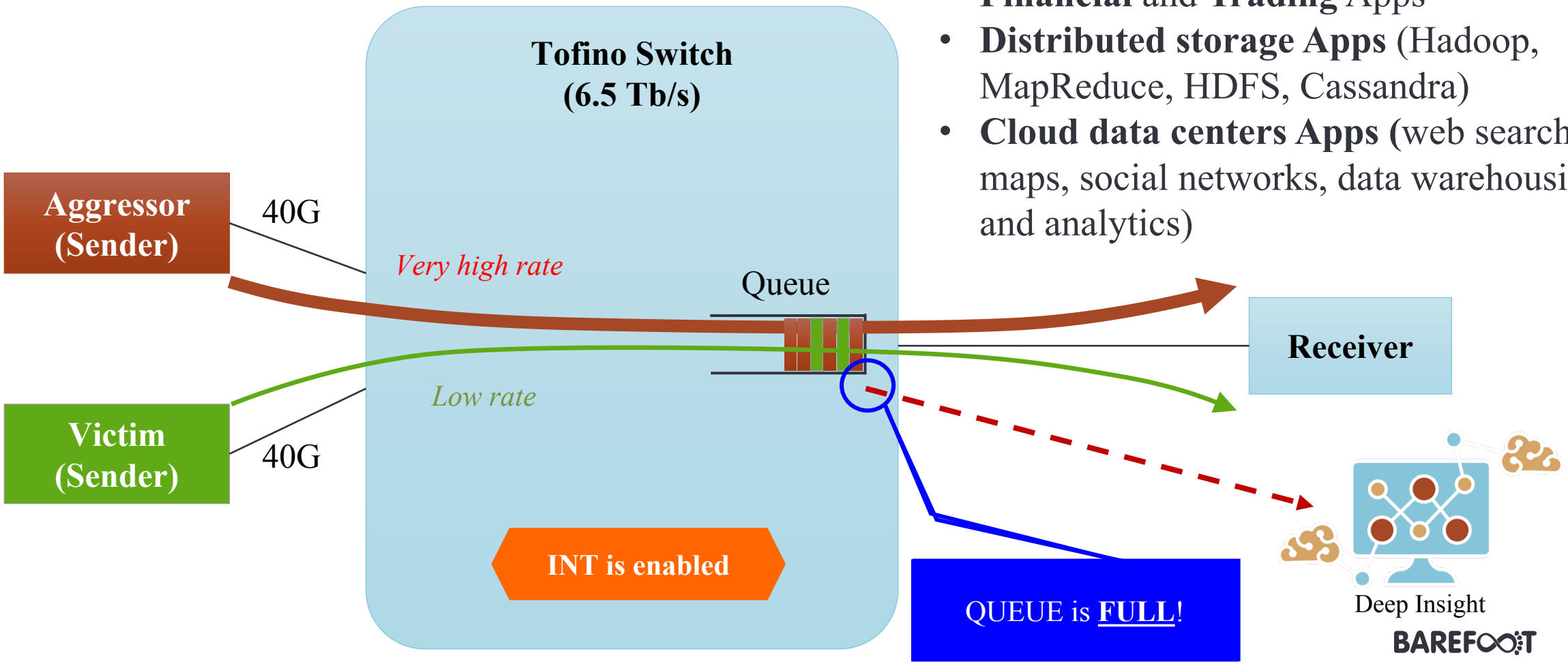
BAREFOOT

Financial and Enterprise – Congestion analysis

TCP INCAST PROBLEMS (MICRO-BURSTS)

Applications

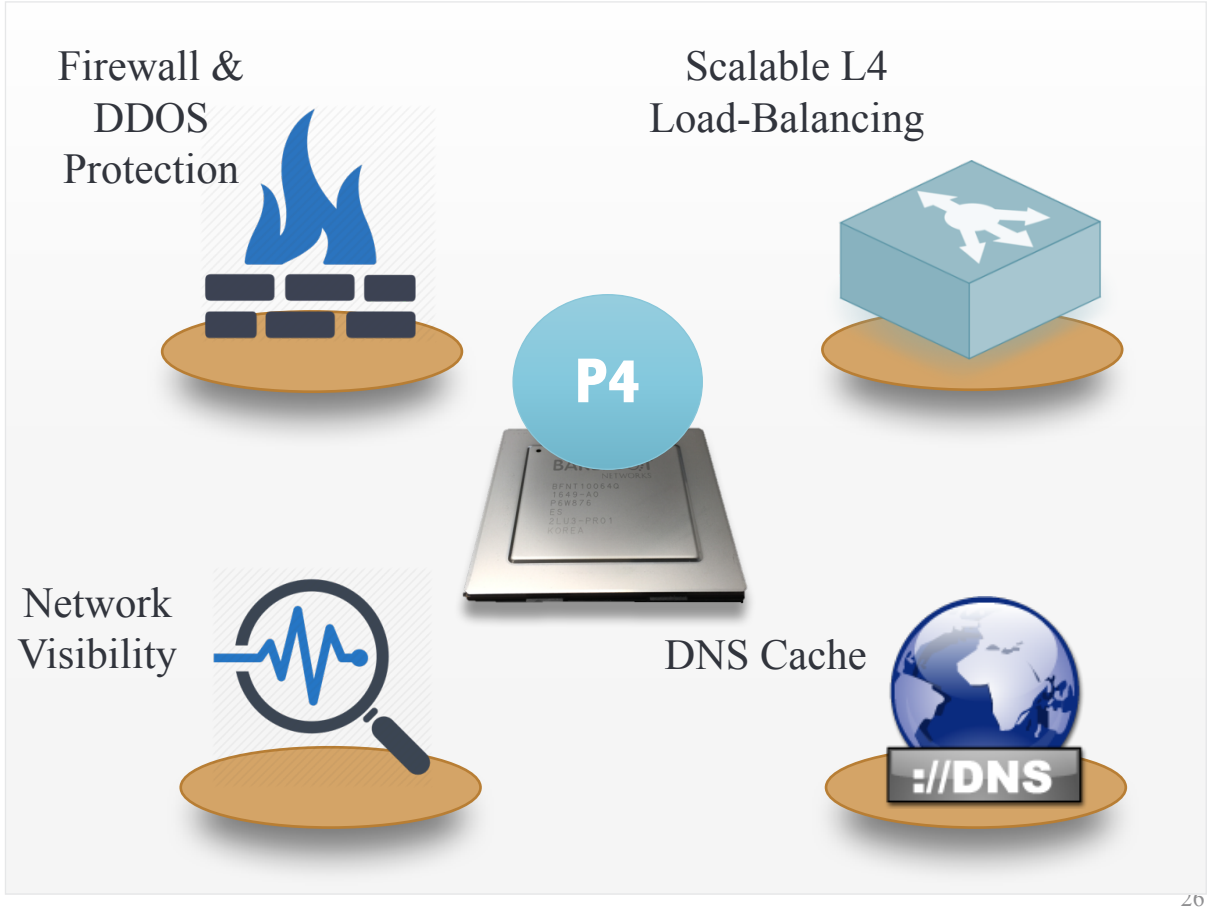
- **Financial** and **Trading** Apps
- **Distributed storage Apps** (Hadoop, MapReduce, HDFS, Cassandra)
- **Cloud data centers Apps (**web search, maps, social networks, data warehousing and analytics)



In-Network and Smart Appliance Model for the Cloud Era

Problem:

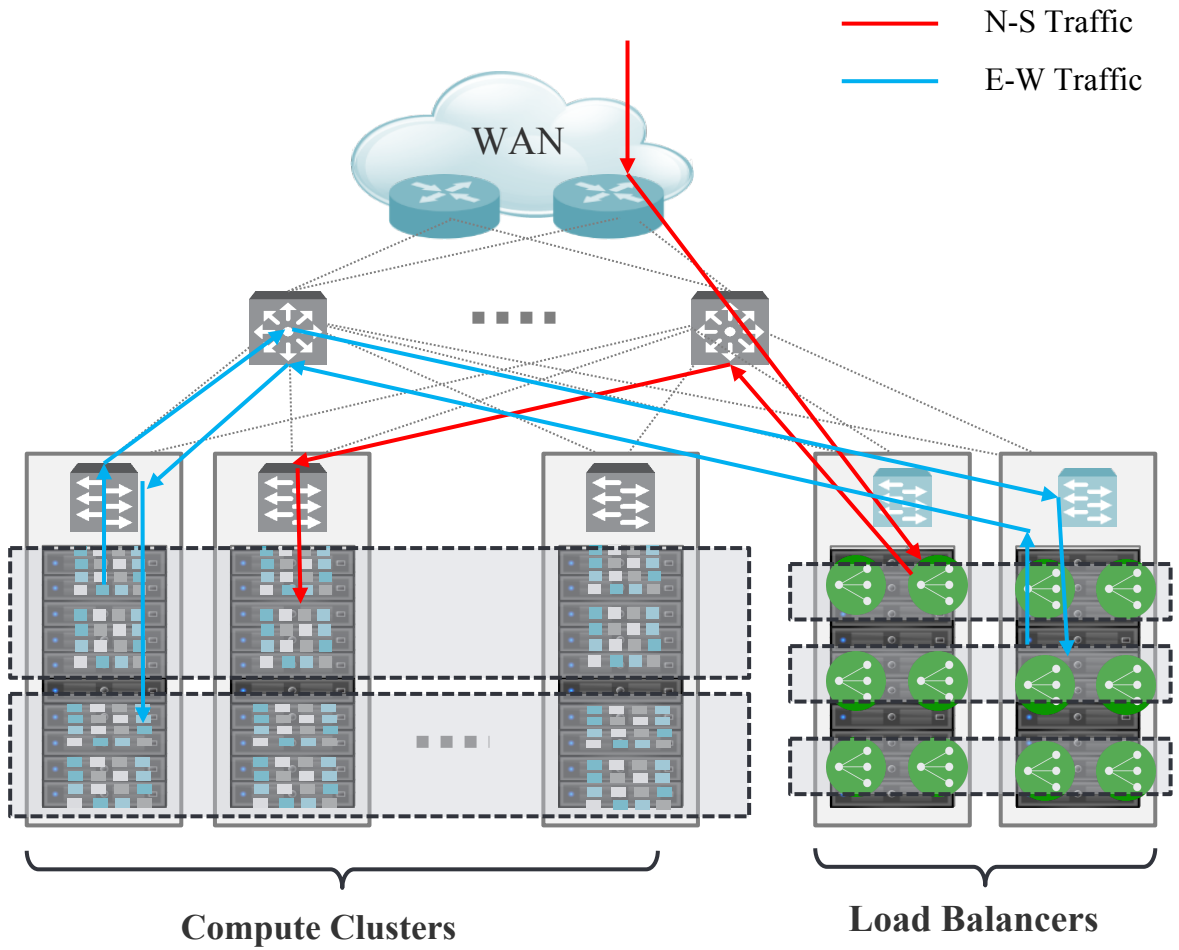
- Customers and Operators devoting thousands of x86 servers for network and security functions
- High CAPEX, Poor efficiency (sized for worst case) and Application performance (Latency)



Accelerate Network, Security and Apps efficiency

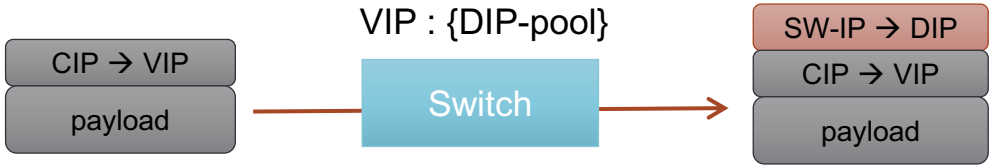
- 100x better performance
- 1000x lower latency (nanoseconds)
- 100x lower power

L4 Load-balancing at every ToR (1 of 2)

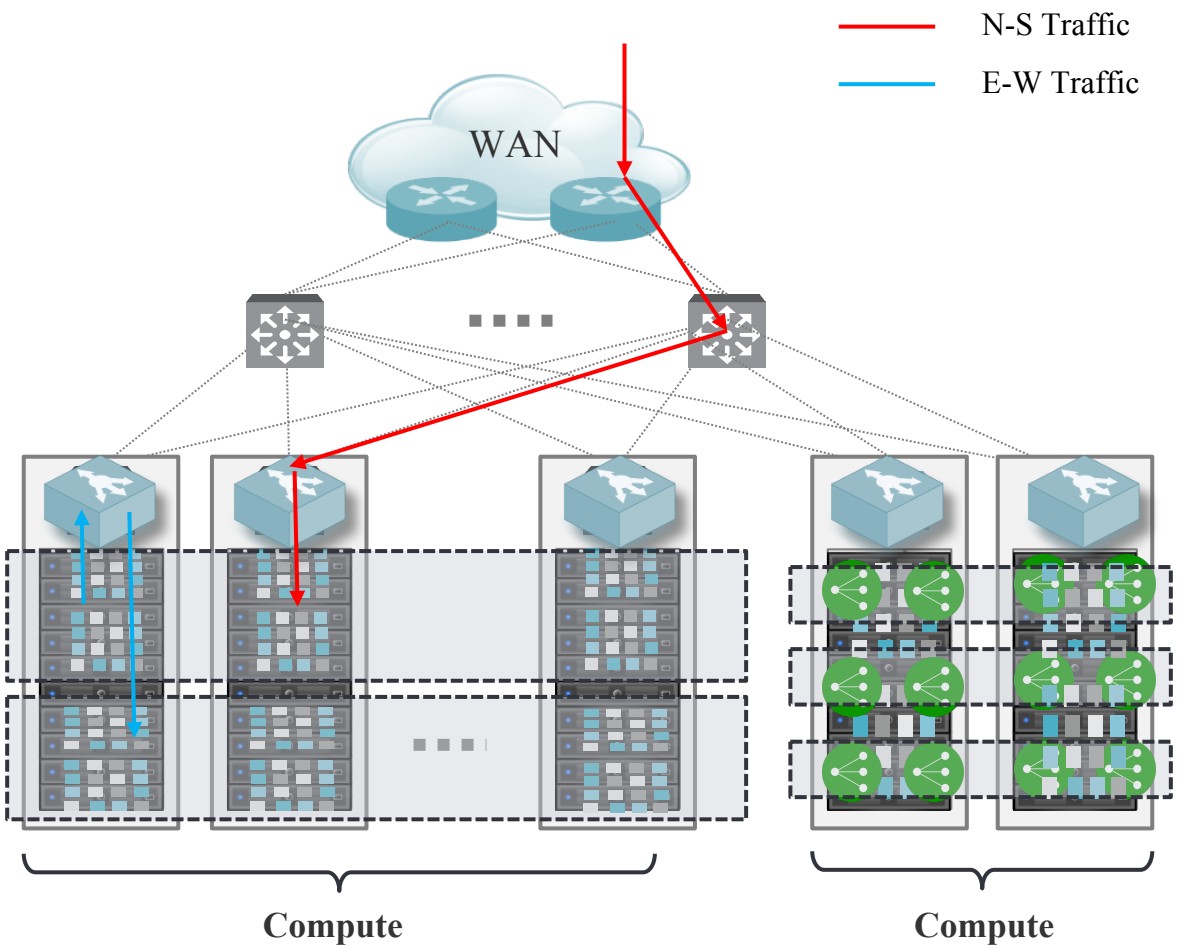


Problem in Today's Networks

- Centralized Hardware or Software Load Balancing with high Infrastructure **Cost and Latency**
- **Hair-pinning** of E-W and N-S Traffic to Virtual or Physical Load-Balancers
- **Hard to scale** to Millions of Connections at Line Rate



L4 Load-balancing at every ToR (2 of 2)



Problem in Today's Networks

- Centralized Hardware or Software Load Balancing with high Infrastructure **Cost and Latency**
- **Hair-pinning** of E-W and N-S Traffic to Virtual or Physical Load-Balancers
- **Hard to scale** to Millions of Connections at Line Rate

P4 Tofino Solution

- **High Scale** (Millions of Flows) with frequent DIP Pool updates
- Per-connection **Consistency**
- **Optimized** Traffic **Flow** and **Latency** (Ideal for E-W LB)
- **Consolidation** of Middle-boxes or x86 to Lower Cost
- **Robustness** and DDOS protection built into Tofino

Flexible Tofino Deployment Model

- **Service Appliance:** L4 LB appliances using Tofino
- **Distributed:** Embed L4 LB capabilities into regular switches

BAREFOOT

In summary…

1. **SDN is about who is in control!**
   Part 1: Network owners decided how their networks are controlled.
   Part 2: With P4, they could decide how packets are processed.
2. **Chip technology**: Programmable switch now has the same power, performance and cost as fixed function.
3. **Line-rate telemetry now possible**. Per-packet, flexible. 100% in data plane at line rate.
4. **New ideas**: Beautiful new ideas are now owned by the programmer, not the chip designer.

BAREFOOT

Thank You!

Haitao Kang
BAREFOOTNETWORKS.COM