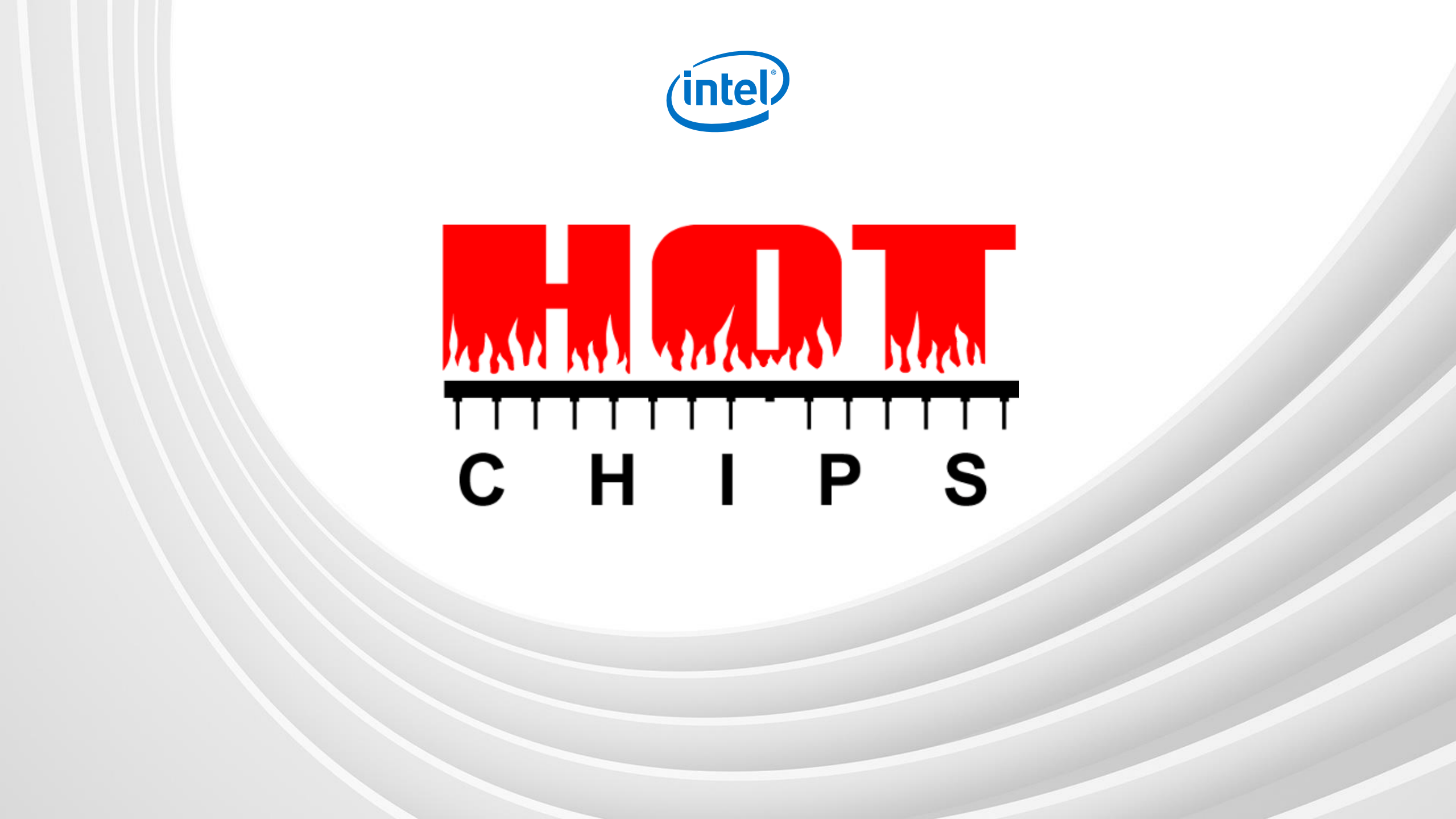
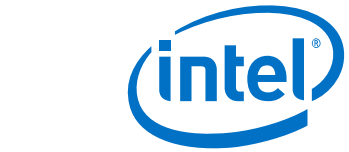

intel
HOT
CHIPS



Intel Tofino2 – A 12.9Tbps P4-Programmable Ethernet Switch

Anurag Agrawal & Changhoon Kim
Intel Corporation

Notices and Disclaimers

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available security updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Why do we need Programmable Switches?

"Future Computing Platforms Critically Require More Efficient, More Reliable, and More Flexible Networks"

Amin Vahdat – Engineering Fellow and Vice President – Google [ONF 2019]

Data plane programmability is critical to enable rapid innovation/experimentation to:

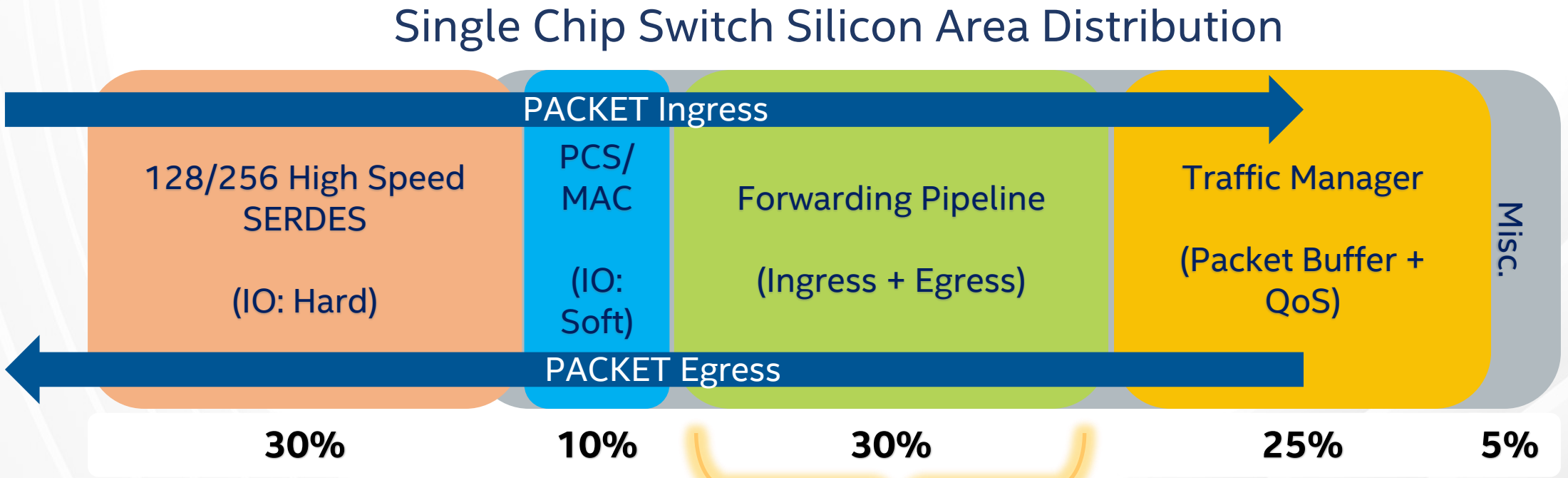
- Supports large scale disaggregation
- Non-blocking BW and "zero queuing" with new end-to-end congestion controls
- In network processing
- End-to-end visibility through data plane telemetry
- Application (load) aware networking
- …

Conventional Wisdom Against Designing One

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

—Conventional Wisdom

Our Observation



Only ~30% silicon needed Significant Attention

SRAM/TCAM make up large part of forwarding logic (Same for fixed or programmable design)

Even if logic increase by 2x → ~20-25% overall increase in chip Area/Power (~~not 10x~~)

PISA: Protocol Independent Switch Architecture

Multiple Match-Action Units for header transformation
(VLIW Instructions, ALUs, SRAM+TCAMs, counters, meters, …)

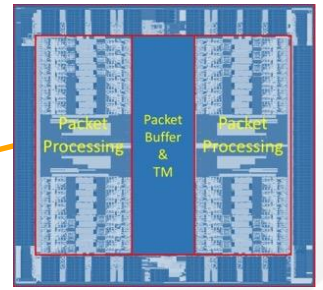
Packet Header fields to 'Registers'
(PHV: Packet Header Vector)

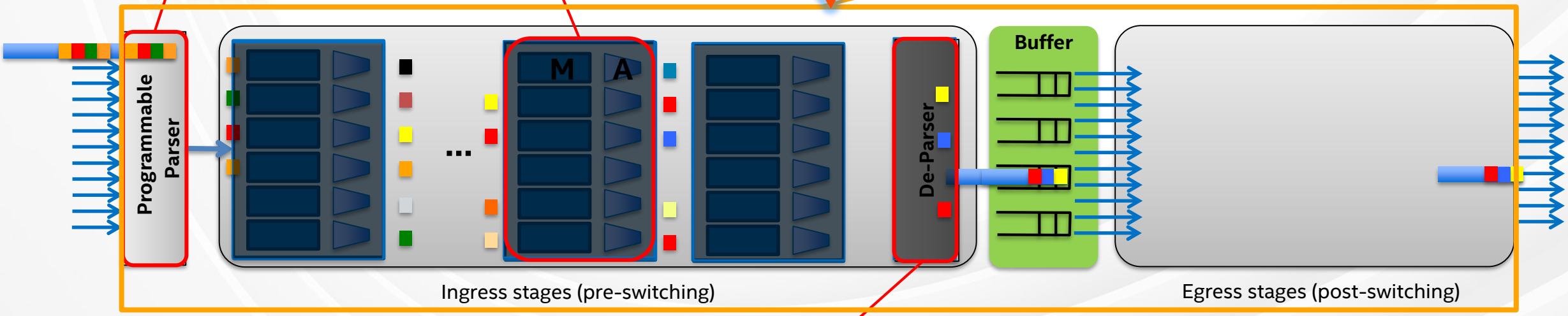
P4 Code

Compiled

User Programs
"This is how you must process packets"

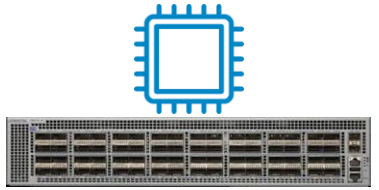
Intel® Tofino™: 6.5Tb/s, 16nm
(Dec'2016)





Any header anywhere

Intel Tofino Proof Point



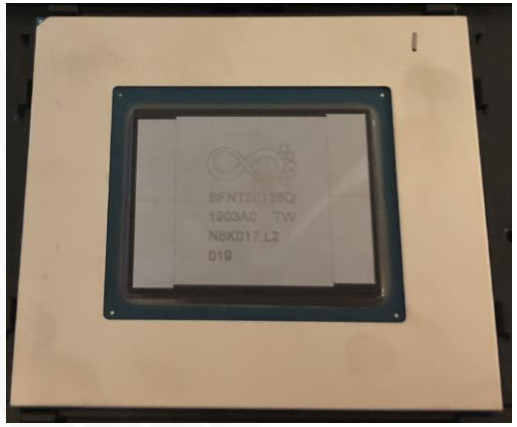
	P4 Programmable "Tofino"	Fixed Function ASIC	Benefits of P4-programmability
Technology	16nm	16nm	
L2/L3 Throughput	6.5Tb/s	6.4Tb/s	
Number of 100G Ports	64	64	
Availability	Yes	Yes	
Max Forwarding Rate	4.8B packets per sec	4.2B packets per sec	14% Greater Performance
Max 25G/10G Ports	256/258	128/130	
Programmability	Yes (P4)	No	
Typical System Power draw	4.2W per port	4.9W per port	14% Lower Power
Large Scale NAT	Yes (100k)	No	
Large scale stateful ACL	Yes (100k)	No	
Large Scale Tunnels	Yes (192k)	No	
Packet Buffer	Unified	Segmented	Better Burst Absorption
Segment Rtg/Bare Metal	Yes/Yes	No/No	
LAG/ECMP Hash Algorithm	Full entropy, programmable	Hash seed, reduced entropy	
ECMP	256 way	128 way	
Telemetry and Analytics	Line-rate per flow stats	Sflow (Sampled)	Real-time Visibility

Can This Be Repeated?

Was Tofino just a fluke?

Is there an evidence that switches can be programmable and meet speeds & feeds?

Intel Tofino2: The 2nd Generation of PISA



- 260x 56G SERDES
- PCIe Gen3 x4

More processing per packet, More efficiency,
Better QoS & Enhanced Advance applications

Performance

- 12.9Tb/s & 6B pkt/s of throughput

More Programmable PISA

- Fully programmable in P4 with more resources

Unified Traffic Manager

- Larger unified packet buffer

Power

- Significant better performance/watt

Industry-leading Process Node

- 7nm Technology

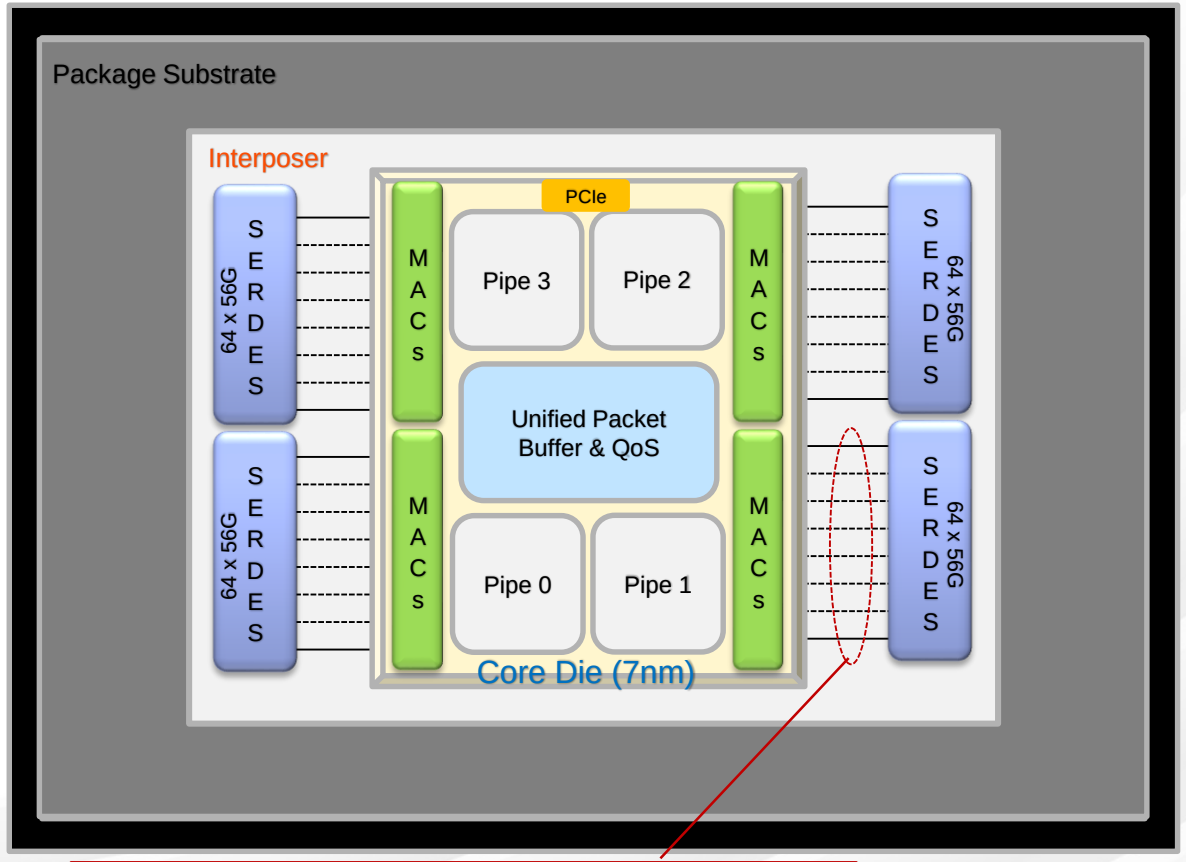
Intel Tofino2: Multi-die Package

Package

- 2.5D CoWoS Packaging
- 260 lanes of 56G-PAM4 SerDes Tiles (x4)
- PCIe Gen3 x4 lanes

Switch Die

- 4 Pipe x 3200Gb/sec/pipe Architecture
- 64MB Unified Packet Buffer
- 400G MACs (32x) + 100G MAC (1x)
  - Highest Radix as 256x10/25/50GE, 128x100GE, 64x200GE, 32x400GE
- Gen3 x4 PCIe Interface, GPIOs
- 7nm FF TSMC

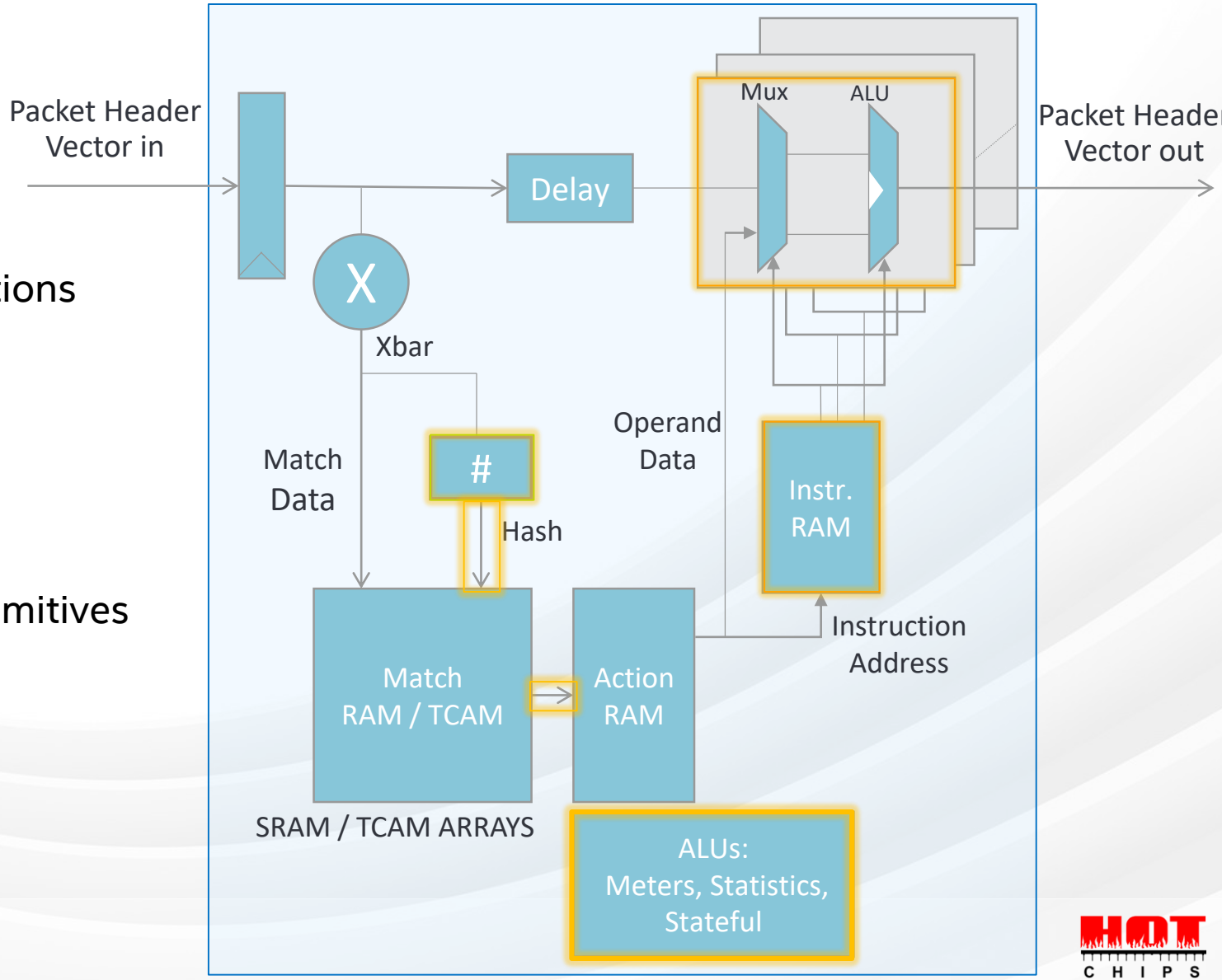


Main Die ←→ Tile

- Bunch-of-Wires (BoW)
- Parallel Source Synchronous bus per SerDes lane

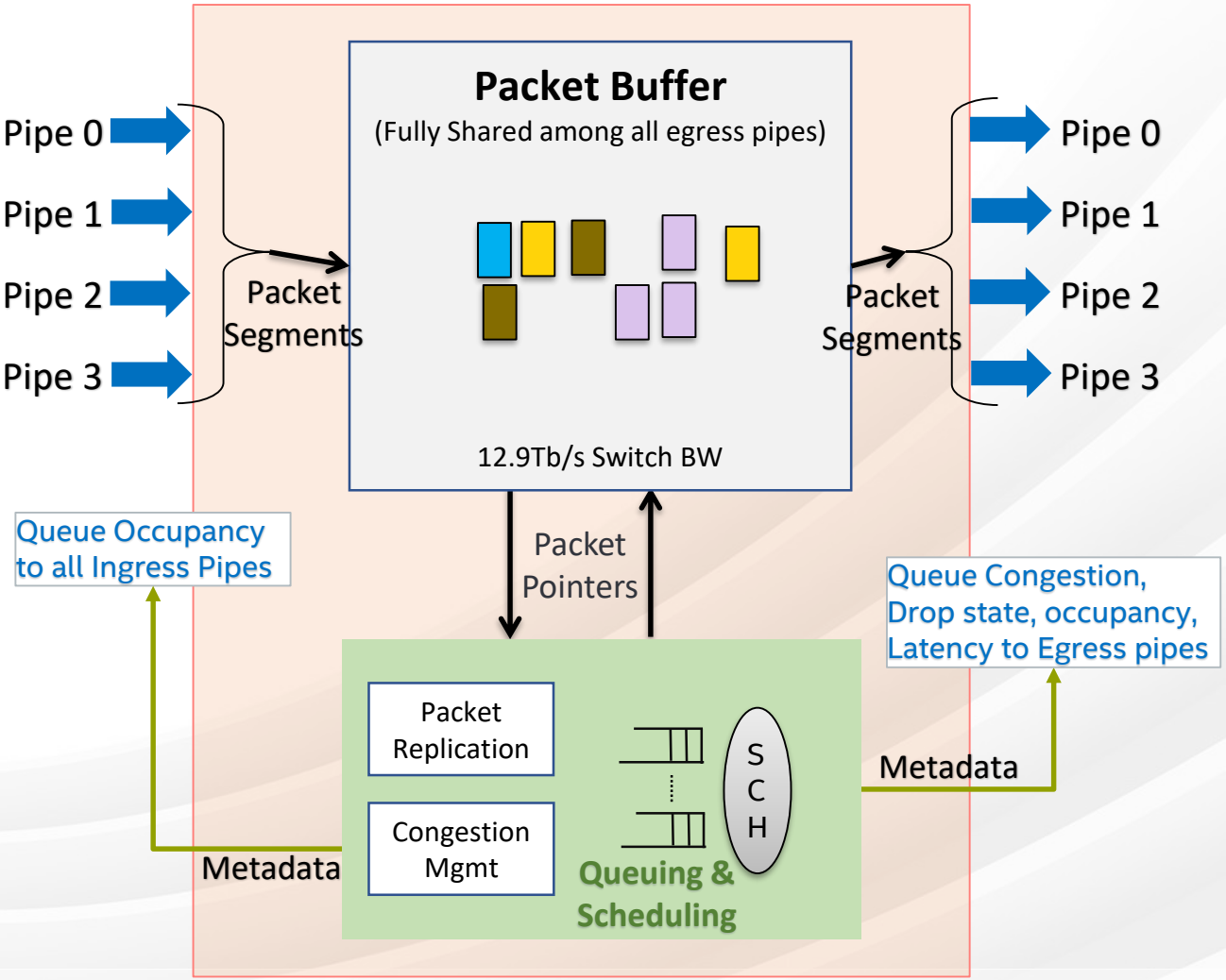
Intel Tofino2: Match-Action Unit (MAU)

- New VLIW instructions, Multiple instructions per table
- Enhanced Stateful ALU capabilities
  - Max/min selection, 16b signed divide, learning/match, and more
- HW learning and Stateful FIFO/Stack primitives

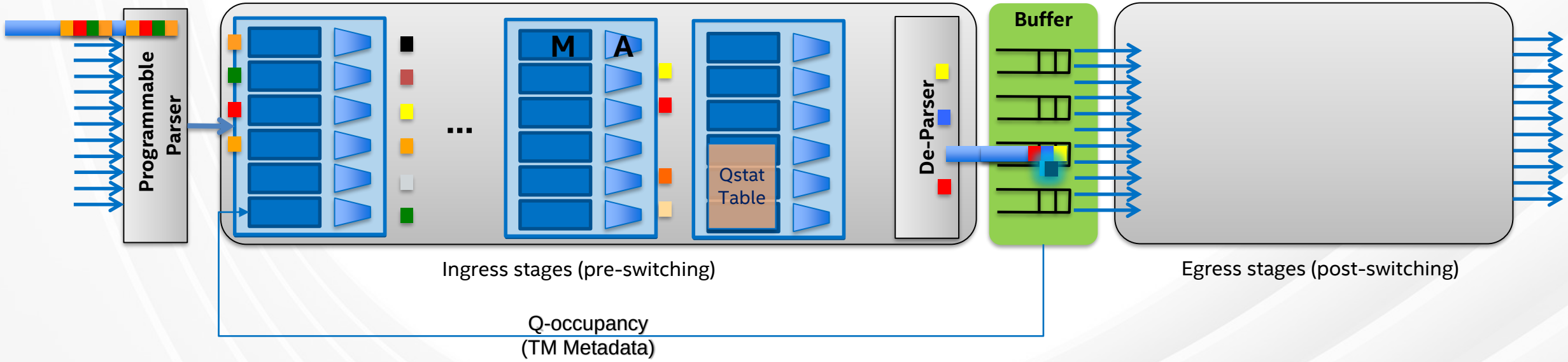


Intel Tofino2: Traffic Manager

- Non-blocking any-to-any communication
  - Radix: 32x400GE … 256x50GE
- ~25Tb/s of write and read from 4 pipes
  - Like a 25000 bits wide memory accessed at 1 GHz
- ~6B pkt/s of enqueue & dequeue operations
- 128Q per 400G port with Hierarchical-QoS
  - 1.68ns for 64B packet @ 400G
- Exporting rich metadata to forwarding pipeline



Intel Tofino2: Traffic Manager Metadata to Ingress Pipe



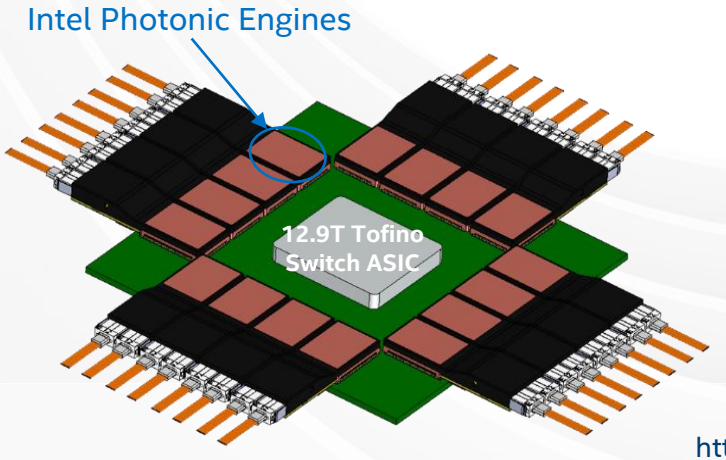
- Egress queues' info at all ingress pipes
- Control plane configures queues to monitor

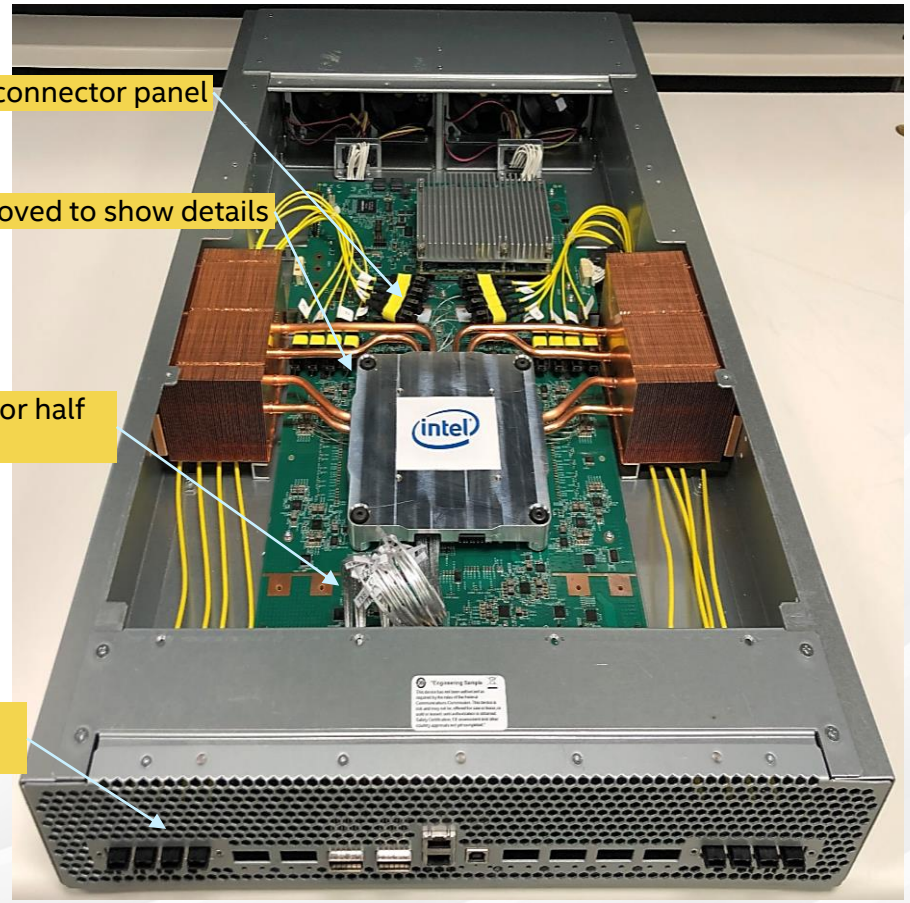
- Use Cases:
  - Congestion-aware routing
  - Congestion Notification from ingress
  - ...

Co-packaged Optics (CPO) Demonstration at 12.9Tb/s

March 2020 Demonstration

- Intel Tofino2 P4-programmable Ethernet switch co-packaged with integrated photonic engines
- Fully functional switch demonstrating 400G Ethernet traffic interoperability
- Compliant with applicable standards for I/O interfaces

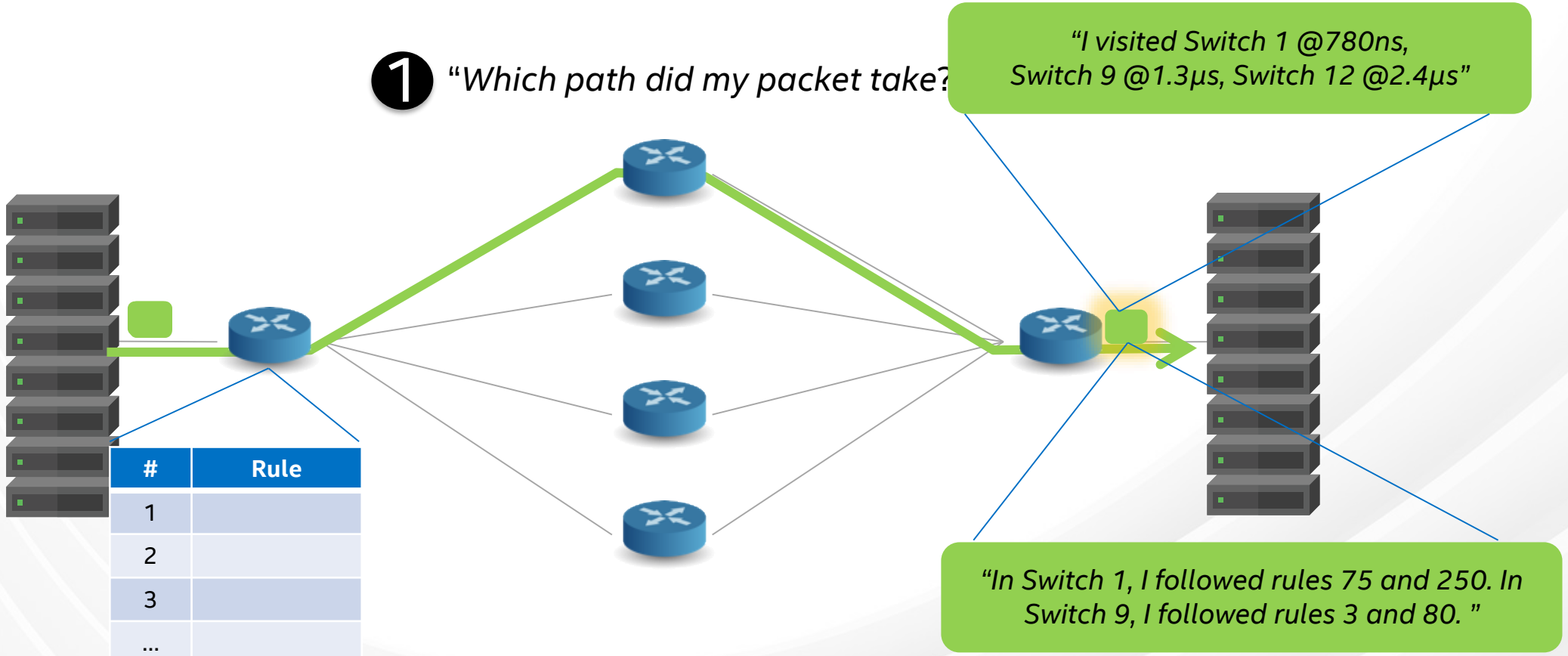




https://newsroom.intel.com/news/intel-demonstrates-industry-first-co-packaged-optics-ethernet-switch/

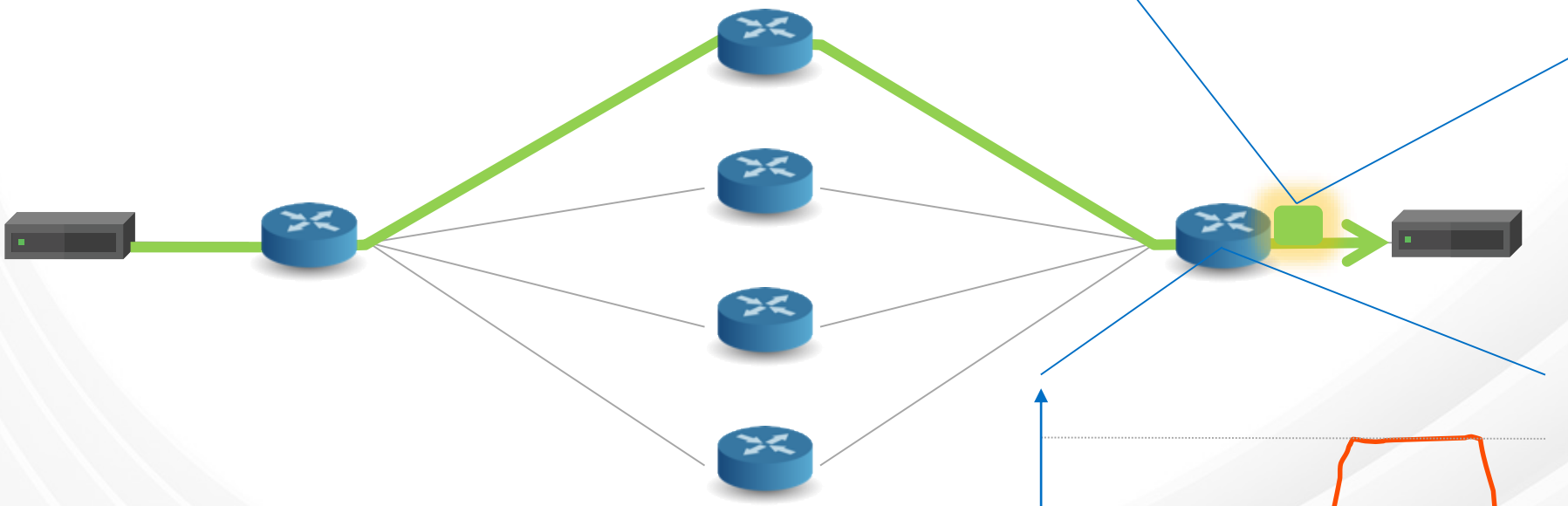
How Is Programmability Used?

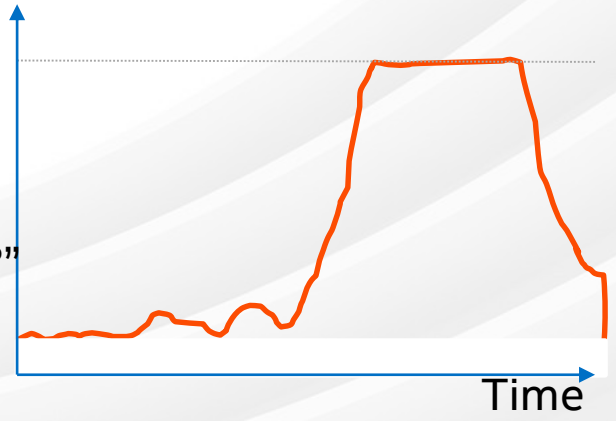
1. Data-plane Telemetry and Real-time Control


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. "

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

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

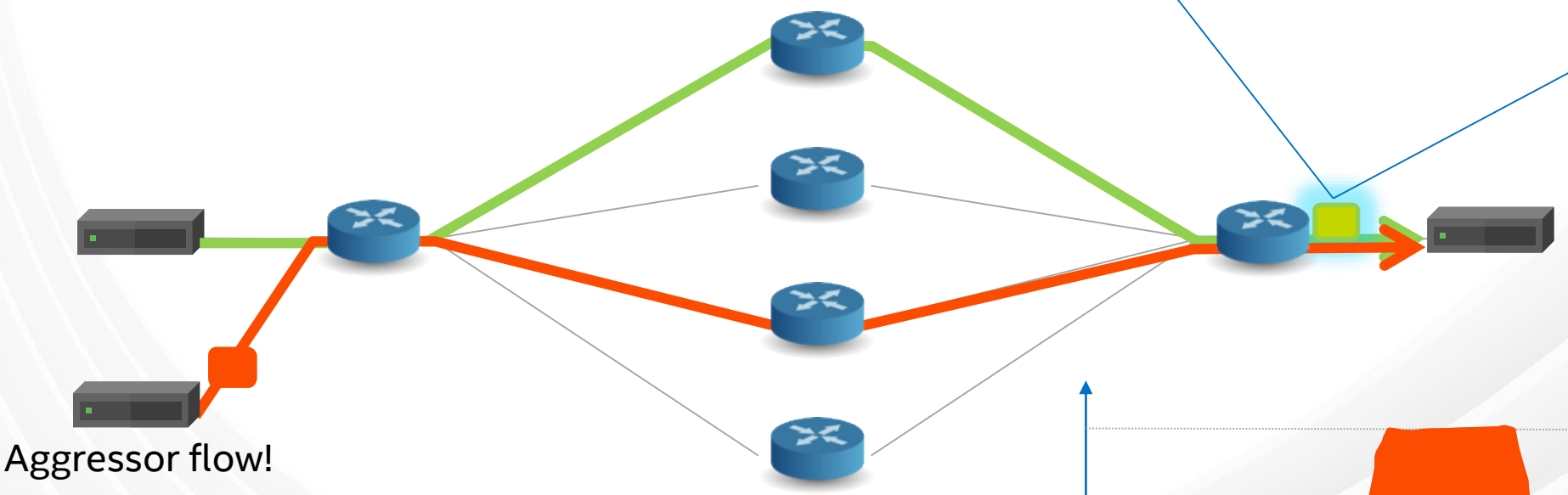




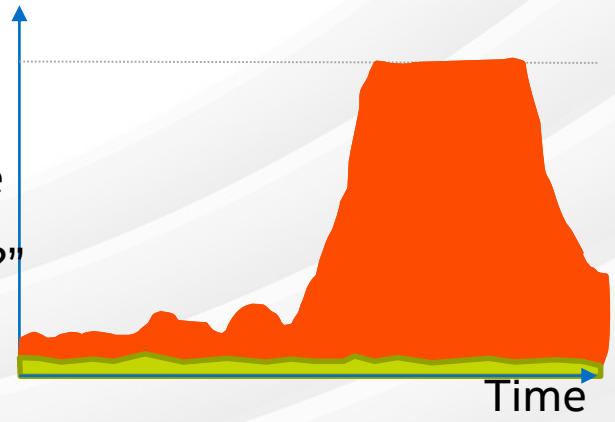
❹ *"Who did my packet share the queue with?"*

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

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



Aggressor flow!



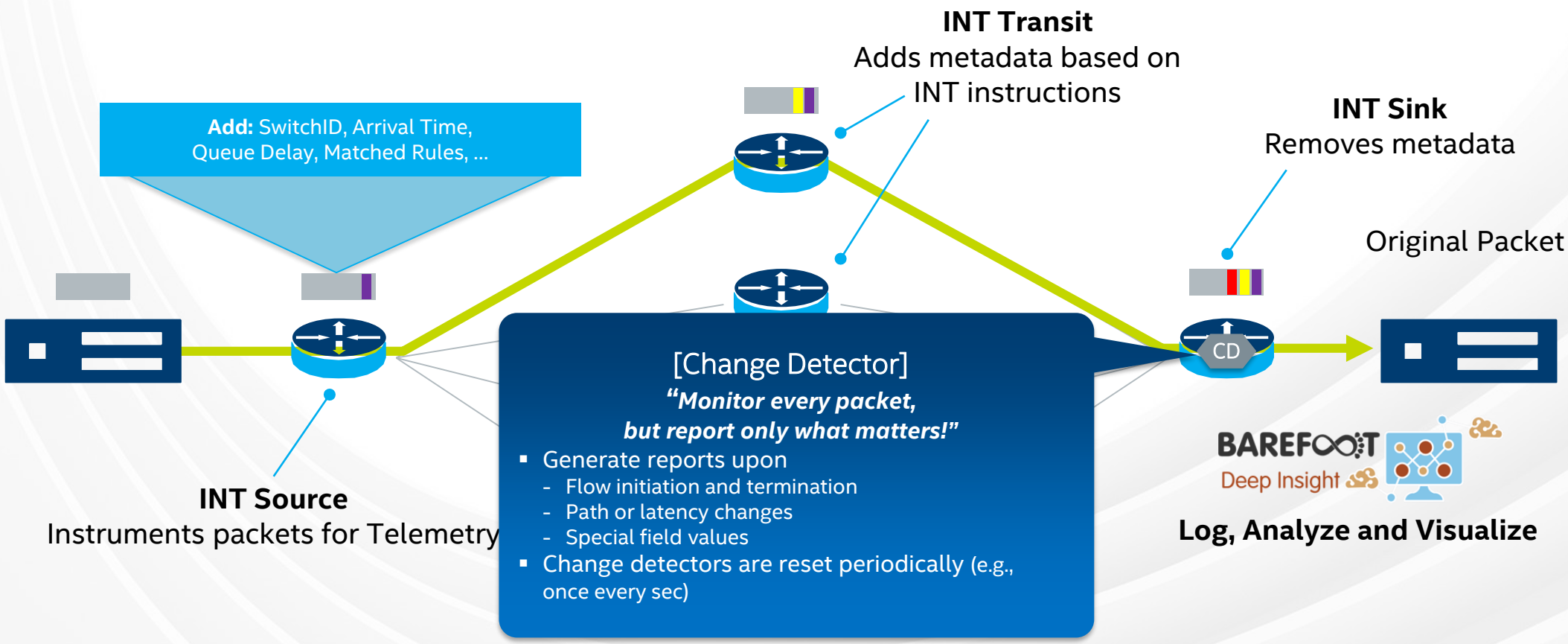
❹ *"Who did my packet share the queue with?"*

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?”

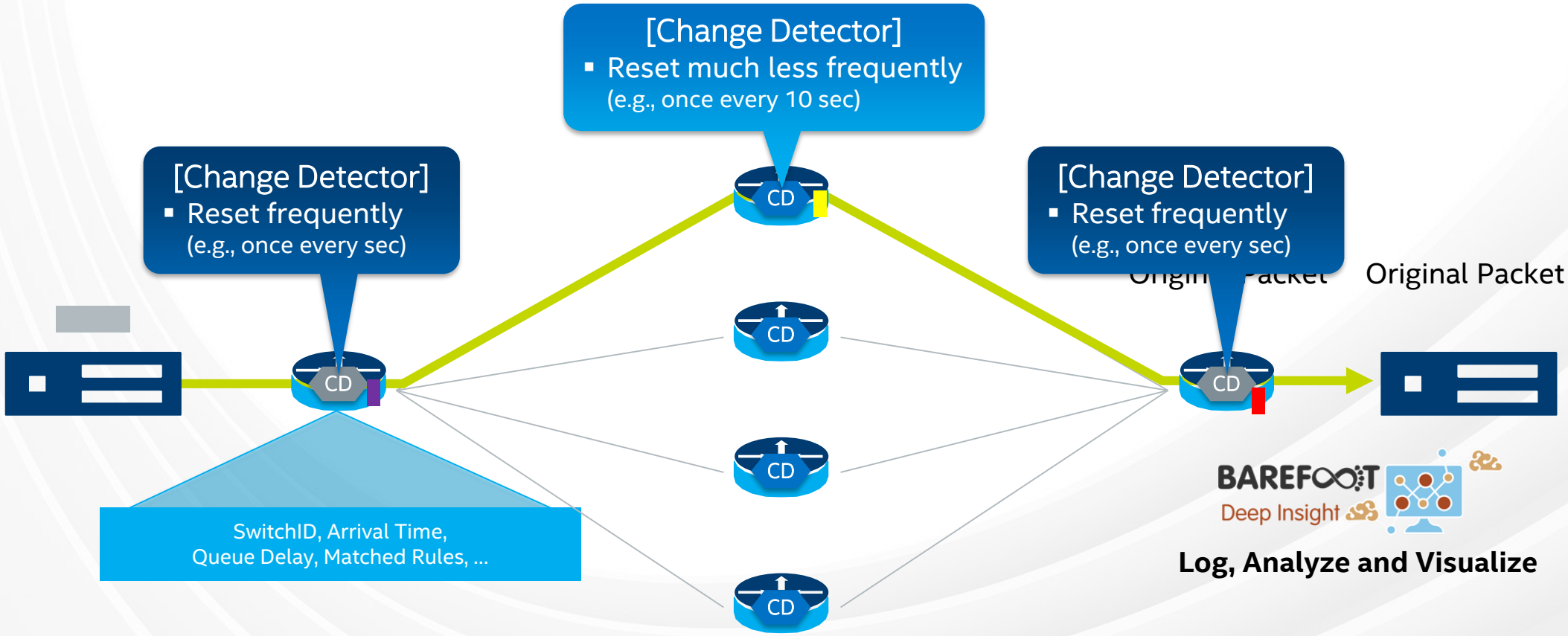
Intel Tofino™ + Deep Insight™ can answer all four questions.
At full line rate. Without generating any additional packets!

Flow Reporting: INT-MD Mode



Flow Reporting: INT-XD Mode

Increase of report-traffic volume is marginal!



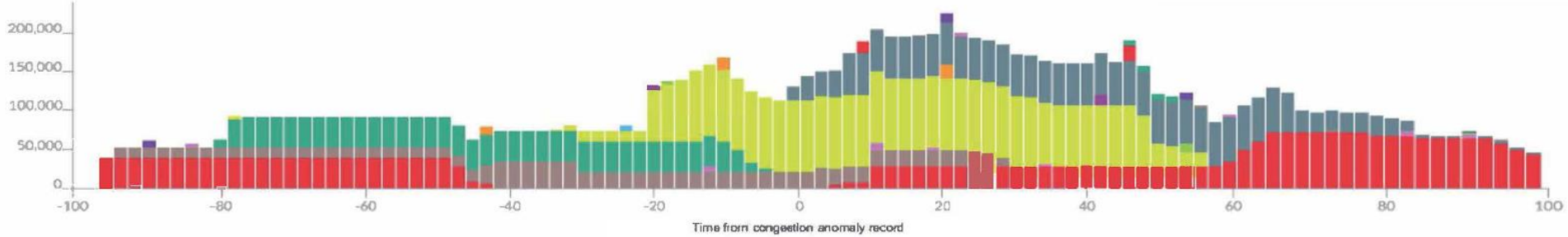
Viewing Microbursts (to the nanosecond)

Anomaly Records — BAREFOOT Deep Insight

Timestamp	Switch Id	Queue

July 25, 2017 - 18:17:51.513 UTC

Queue Occupancy Over Time (bytes)



17 Affected Flows

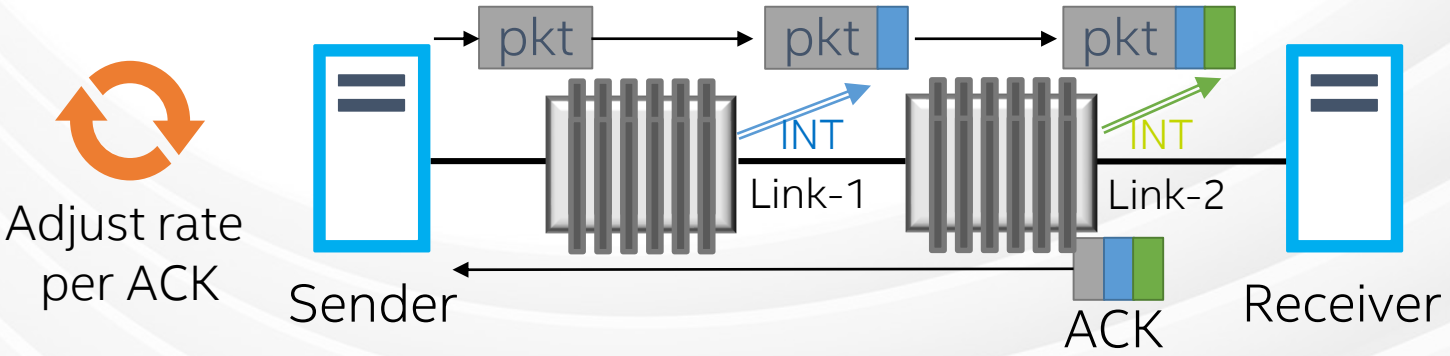
Flow	kB in Queue	% of Queue Buildup	Packet Drops
10.32.2.2:46380 -> 10.36.1.2:5101 TCP	3282	29	0
10.32.2.2:46374 -> 10.36.1.2:5101 TCP	3073.5	27	25
10.32.2.2:46386 -> 10.36.1.2:5101 TCP	2092.5	18	27
10.32.2.2:46388 -> 10.36.1.2:5101 TCP	1456.5	13	0
10.32.2.2:46390 -> 10.36.1.2:5101 TCP	1227	11	36
10.32.2.2:46372 -> 10.36.1.2:5101 TCP	45	0	0
10.32.2.2:46392 -> 10.36.1.2:5101 TCP	37.5	0	39
10.35.1.2:34256 -> 10.36.1.2:5102 TCP	34.5	0	0

HPCC: INT-based High Precision Congestion Control

Published at SIGCOMM 2019 by Alibaba, Harvard, Univ. of Cambridge, and MIT

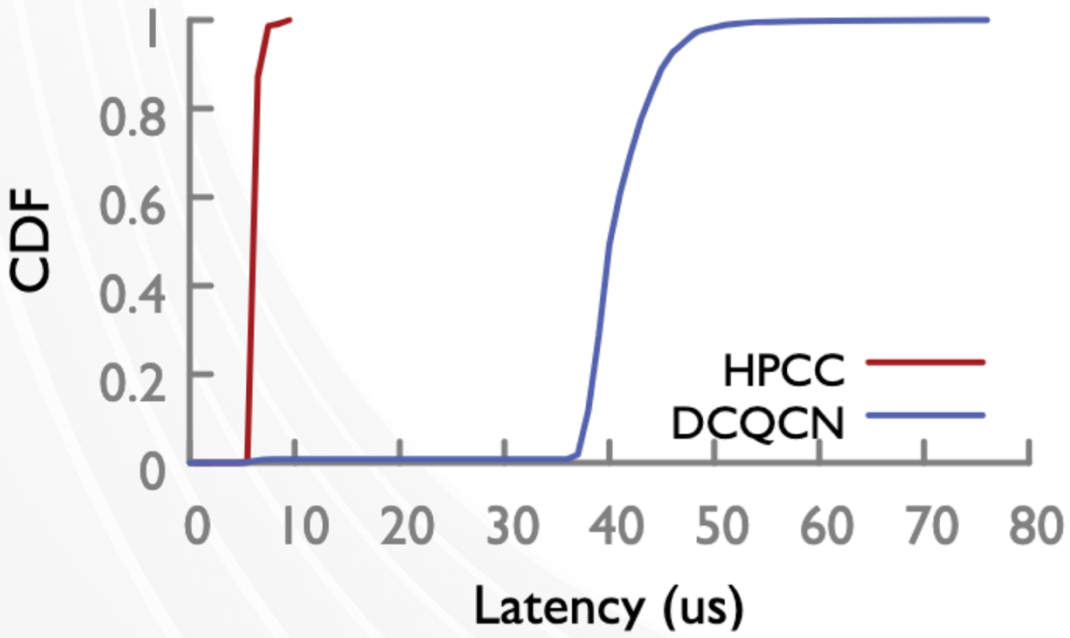
Using INT as explicit and precise feedback

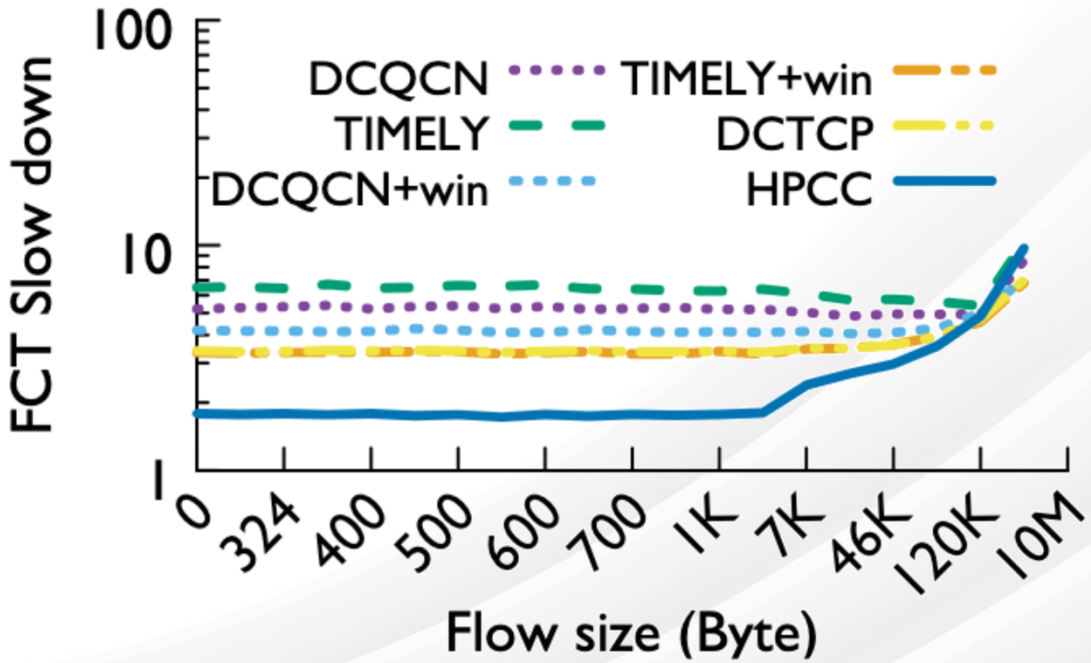
- Very fast convergence via MIMD (multiplicative increase & multiplicative decrease)
- Near-zero queue
- Few parameters



Key Benefits of HPCC:

Extremely Low Latency And Very High Throughput At The Same Time





How Is Programmability Used?

2. Accelerating Apps Beyond Networking

Cambrian Explosion Of Beautiful New Apps

Graphics

OpenCL™

Compiler

GPU & GPGPU

- More…
- ML/DL - Inference
- ML/DL - Training
- Self-driving Vehicles
- HPC & Big Data
- Virtual Reality
- Augmented Reality
- Gaming/Professional Graphics
- 2D/3D Graphics
- Basic Graphics

Networking

P4

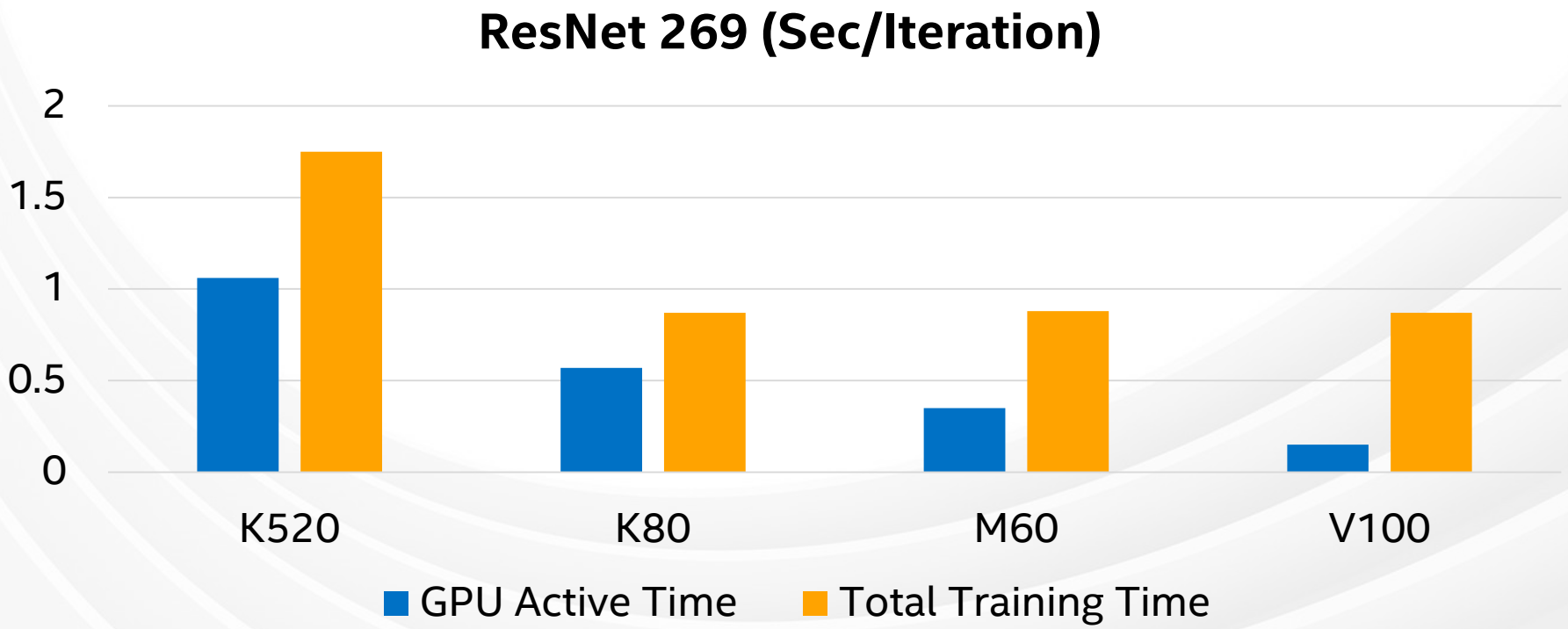
Compiler

Programmable Data Planes

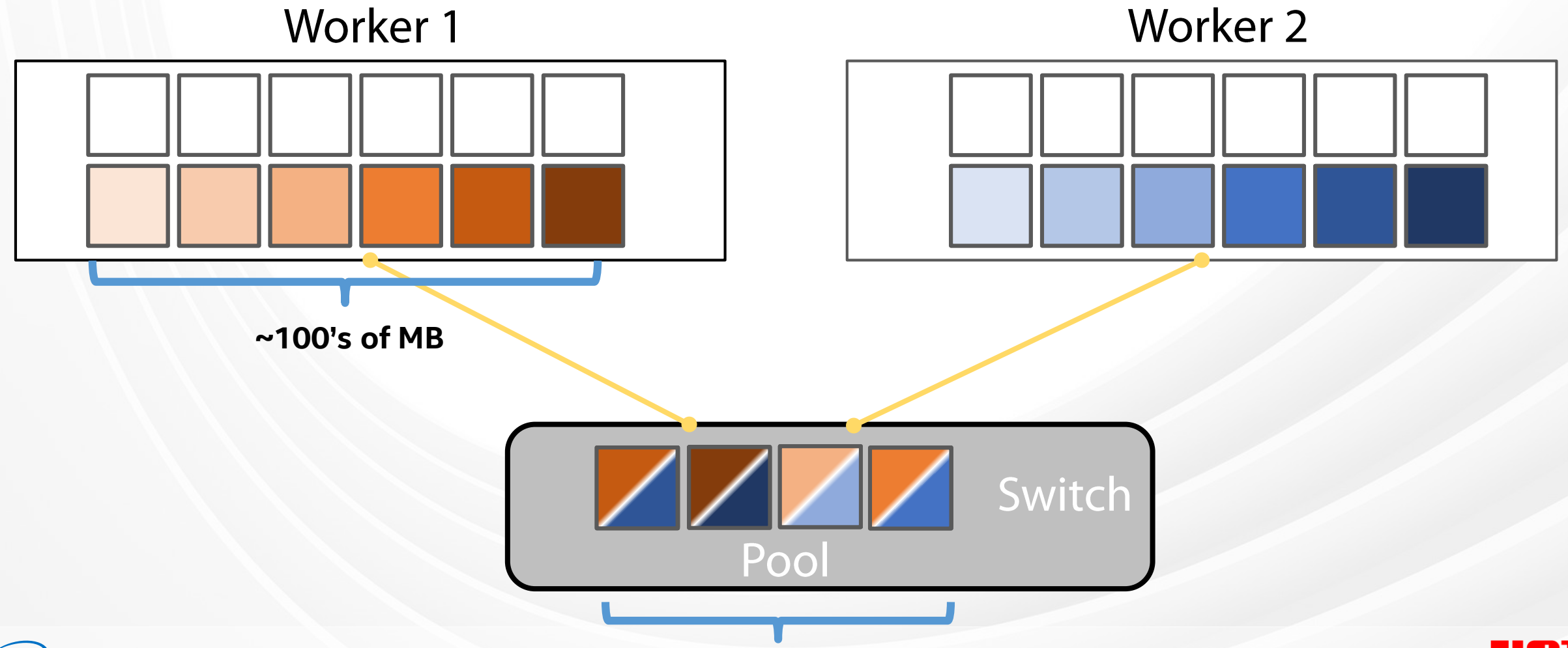
- More…
- Storage/Memory/ML Interconnect
- In-line Crypto
- DNS Cache
- KV Caching & Replication
- Extremely-Accurate Time Sync
- Pattern Matching & Stream Processing
- Consensus Acceleration
- Network Packet Broker
- Firewall & DDoS
- Accelerated VNFs
- E2E Telemetry & Analytics
- Adv Congestion Control
- Enterprise/Telco/SP Switching
- Cloud Switching

Accelerating DNN Training

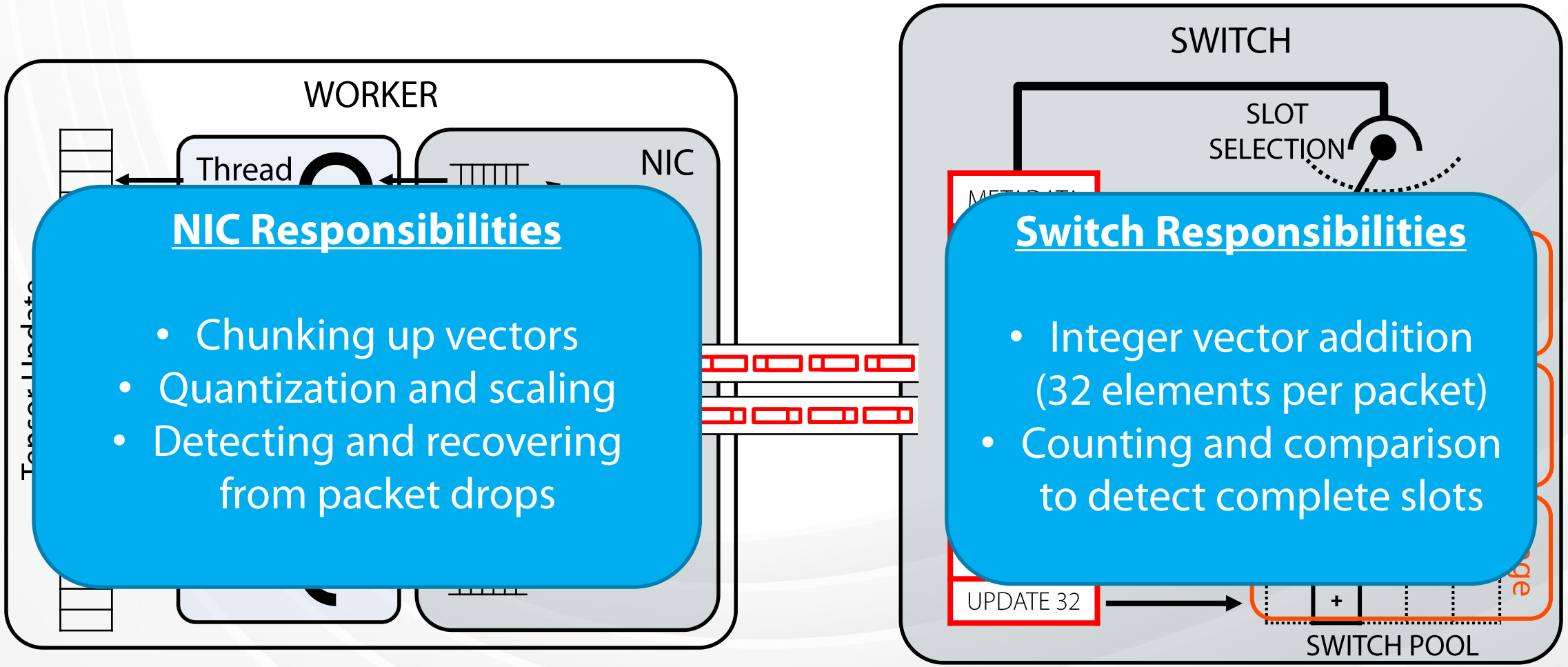
- Training over huge data requires distributed processing
- With faster workers, sharing learned parameters becomes a bottleneck



SwitchML: Streaming Aggregation via Switches

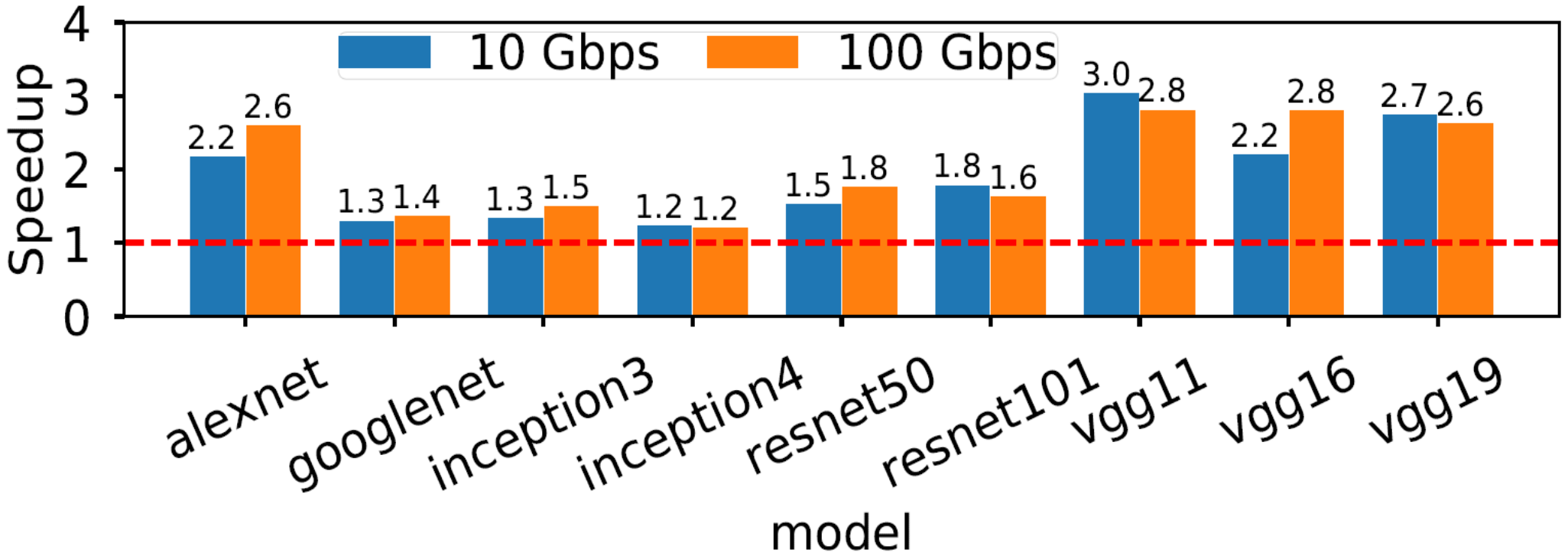


Combined Switch-NIC Architecture

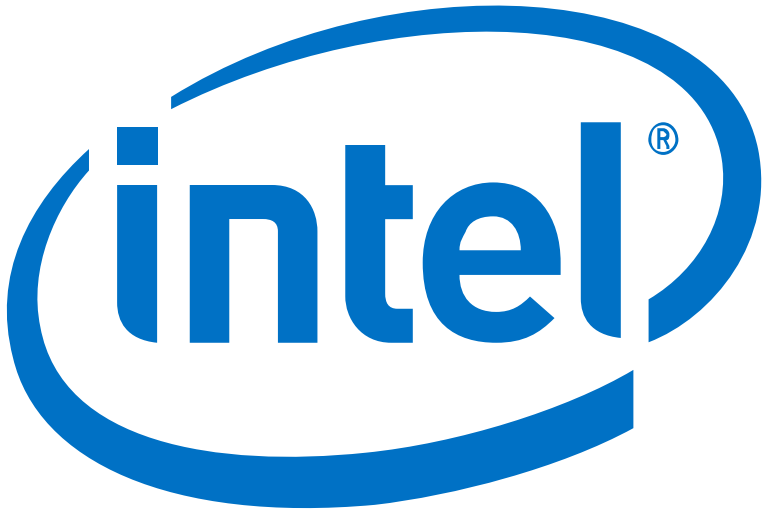


How Much Faster is SwitchML?

SwitchML provides a speedup from **20% to 300%** compared to Tensorflow with NCCL (with direct GPU memory access)



Combined Switch-NIC Architecture Leads To Innovations


intel