NetFPGA Summer Course

Presented by:

Andrew W Moore, Noa Zilberman, Gianni Antichi
Stephen Ibanez, Marcin Wojcik, Jong Hun Han,
Salvator Galea, Murali Ramanujam, Jingyun Zhang,
Yuta Tokusashi

University of Cambridge
July 24 – July 28, 2017

http://NetFPGA.org

The power of OpenSource: build your own proof of concept!

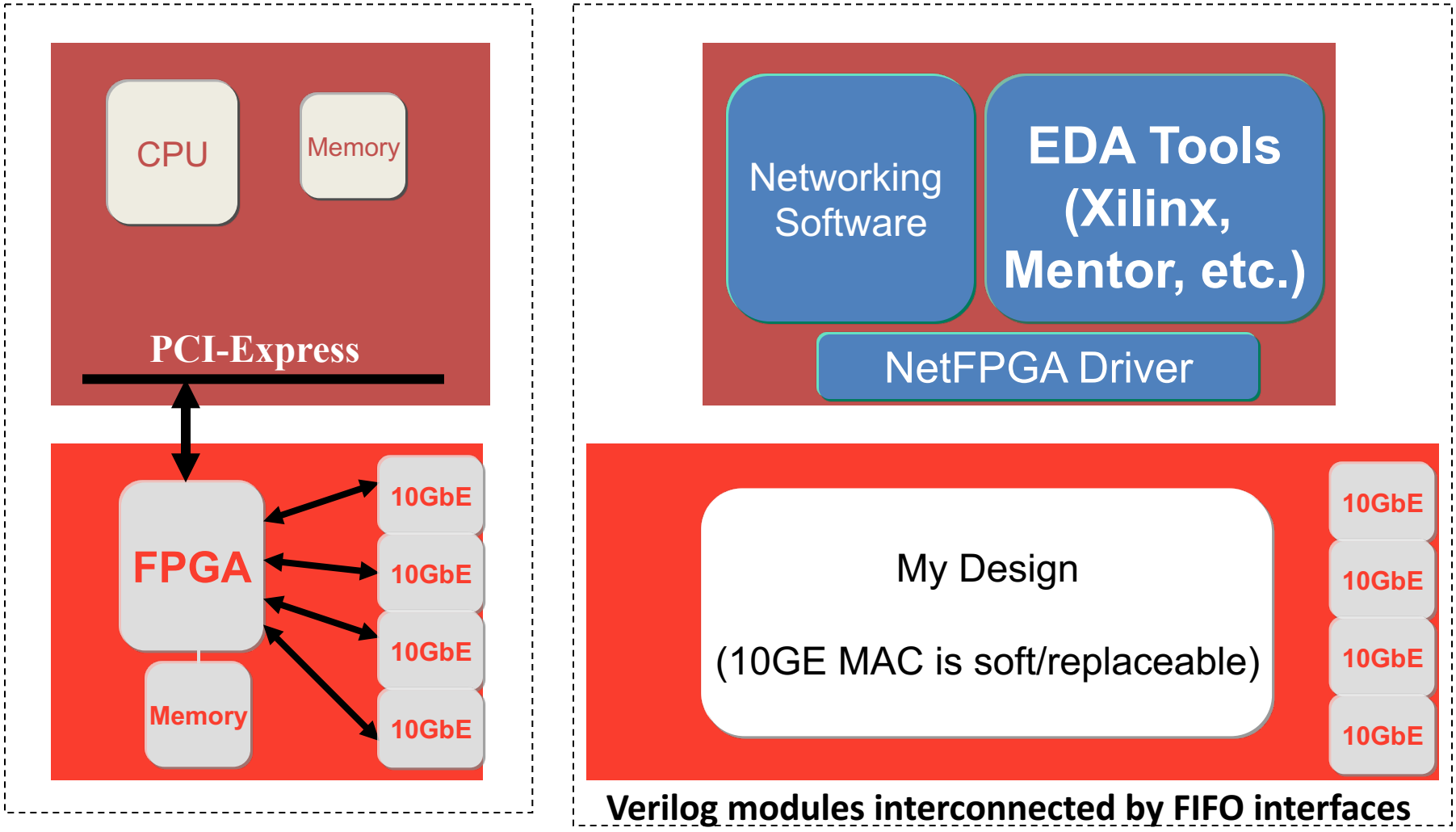
Gianni Antichi

Computer Laboratory
University of Cambridge

gianni.antichi@cl.cam.ac.uk
https://www.cl.cam.ac.uk/~ga288

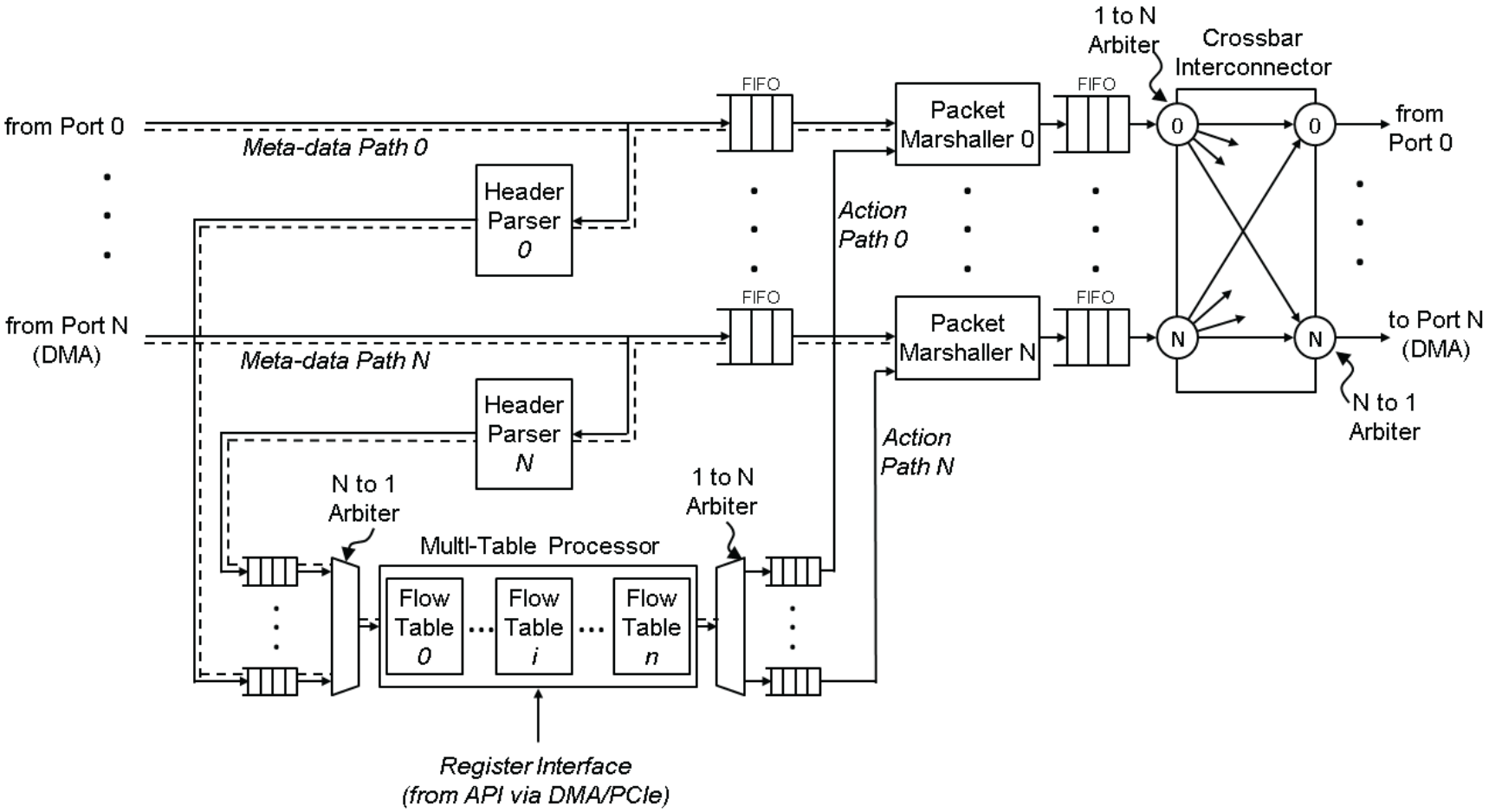
UNIVERSITY OF CAMBRIDGE

NetFPGA: Networked FPGA



BlueSwitch: A Multi Table OpenFlow Switch

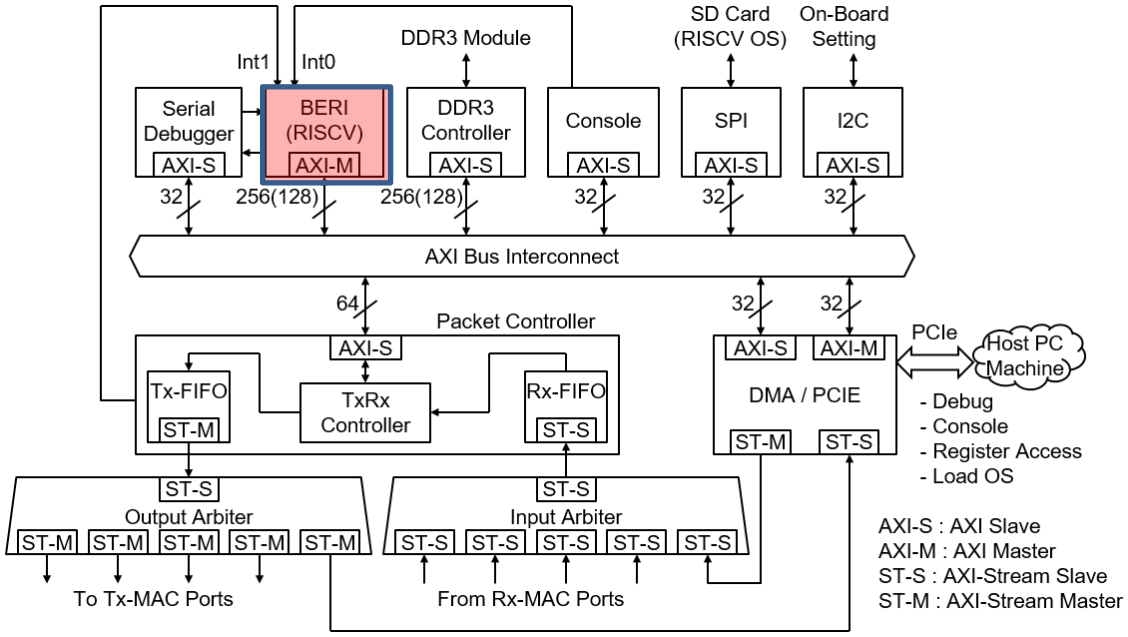
Your design can look completely different!



Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

NetSoc: NetFPGA + Open Source Processors

- Open source, RISC based SoC architectures
- *RISC-V – RISC-V ISA soft processor, Linux OS*
- *CHERI – 64bit MIPS soft processor, BSD OS*

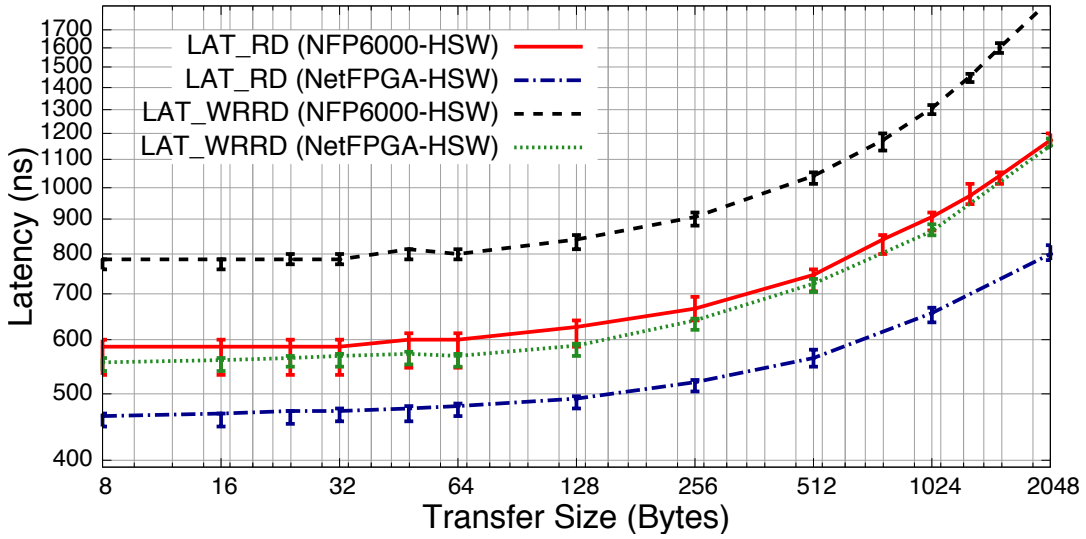


Prototyping RISC Based, Reconfigurable Networking Applications in Open Source, Han J.H., Zilberman N., Zeeb B.A, Fiessler A. and Moore A.W., Technical Report, CoRR abs/1612.05547, 2016

pciebench: an open source tool for benchmarking PCI Express

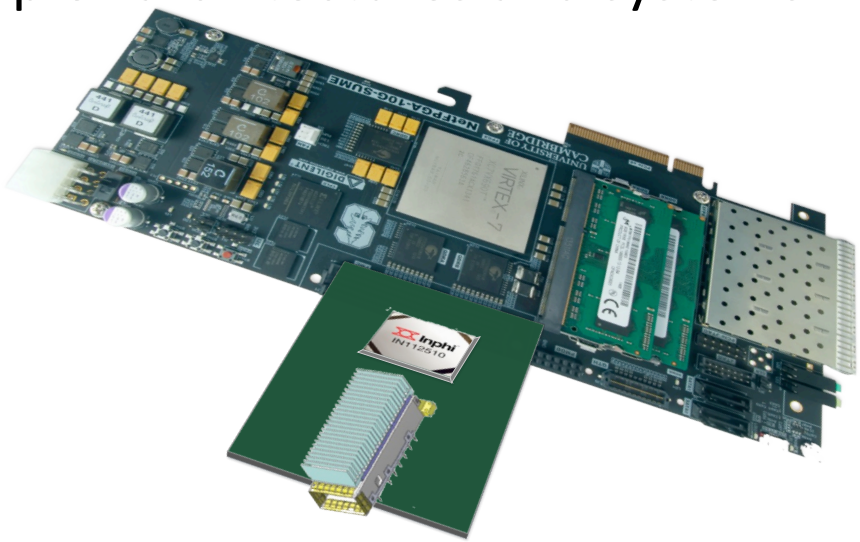
there is a limited understanding of PCIe functionality, nor the trade-offs that must be made to get best-performance from PCIe systems

- pciebench tool open source available
- It builds on NetFPGA and Netronome boards



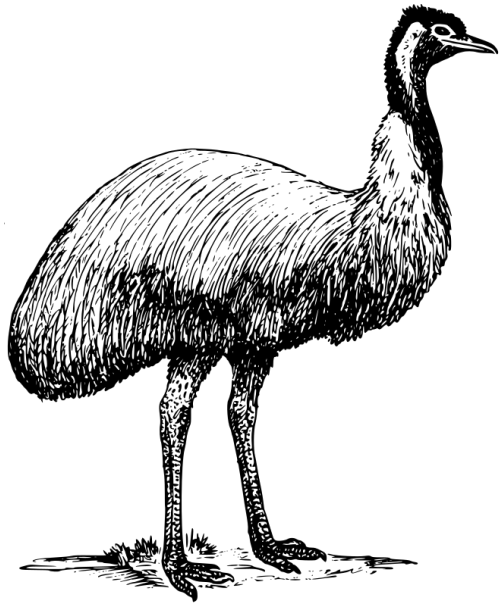
Power Efficient MAC

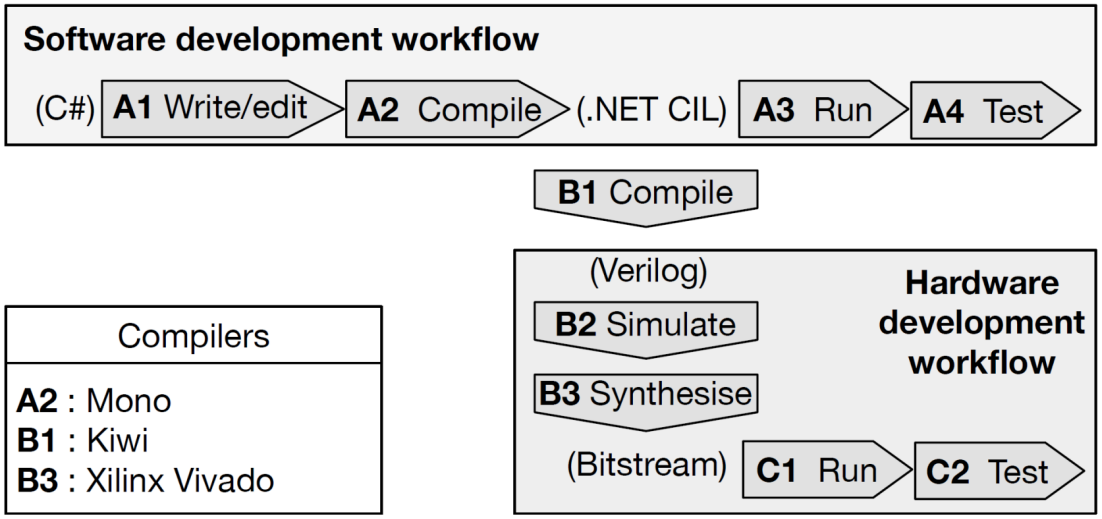
- A Platform for 100Gb/s power-saving MAC design (e.g. lights-out MAC)
- Porting MAC design to SUME permits:
  - Power measurements
  - Testing protocol's response
  - Reconsideration of power-saving mechanisms
  - Evaluating suitability for complex architectures and systems



Emu: *Rapid FPGA Prototyping of Networking Services in C#*

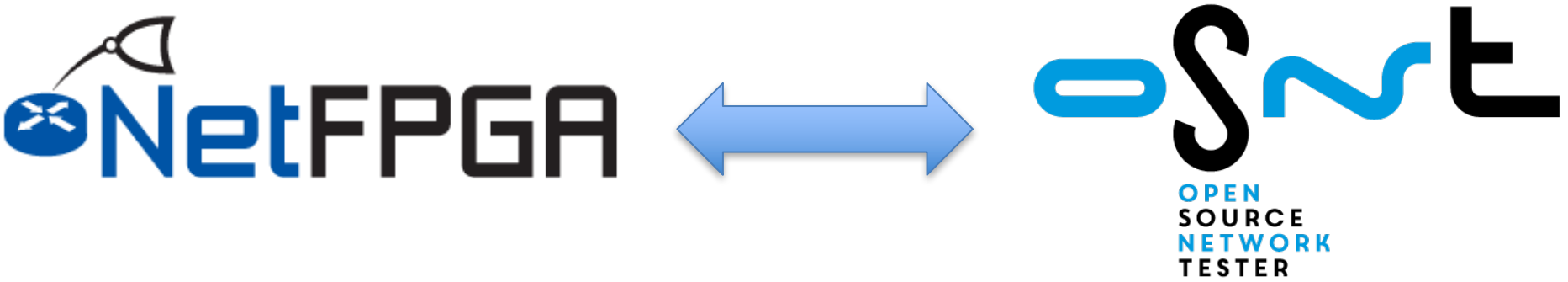
- Accelerating networking services
- Compiling .Net programs
  - To x86
  - To simulation environment
  - To multiple FPGA targets





Emu: Rapid Prototyping of Networking Services, Sultana N., et al., Usenix Annual Technical Conference (ATC), July 2017

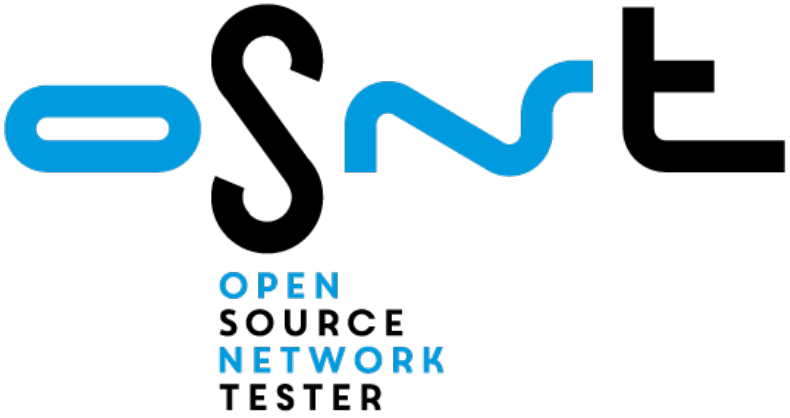
Open Source Network Tester



- NetFPGA platform enabled the first prototype of OSNT.
- The open nature of NetFPGA ecosystem represents the best starting point for open HW/SW community-oriented projects.
- OSNT aims to build a community as NetFPGA did.

OSNT: Open Source Network Tester, Antichi G. et al., IEEE Network Magazine, Special issue on Open Source for Networking: Tools and Applications, 2014

OSNT: Open Source Network Tester



Open source hardware and software platform for network monitoring and testing.

https://osnt.org

Low cost, flexible to update, scale-out, no CPU usage, nanosecond resolution measurements

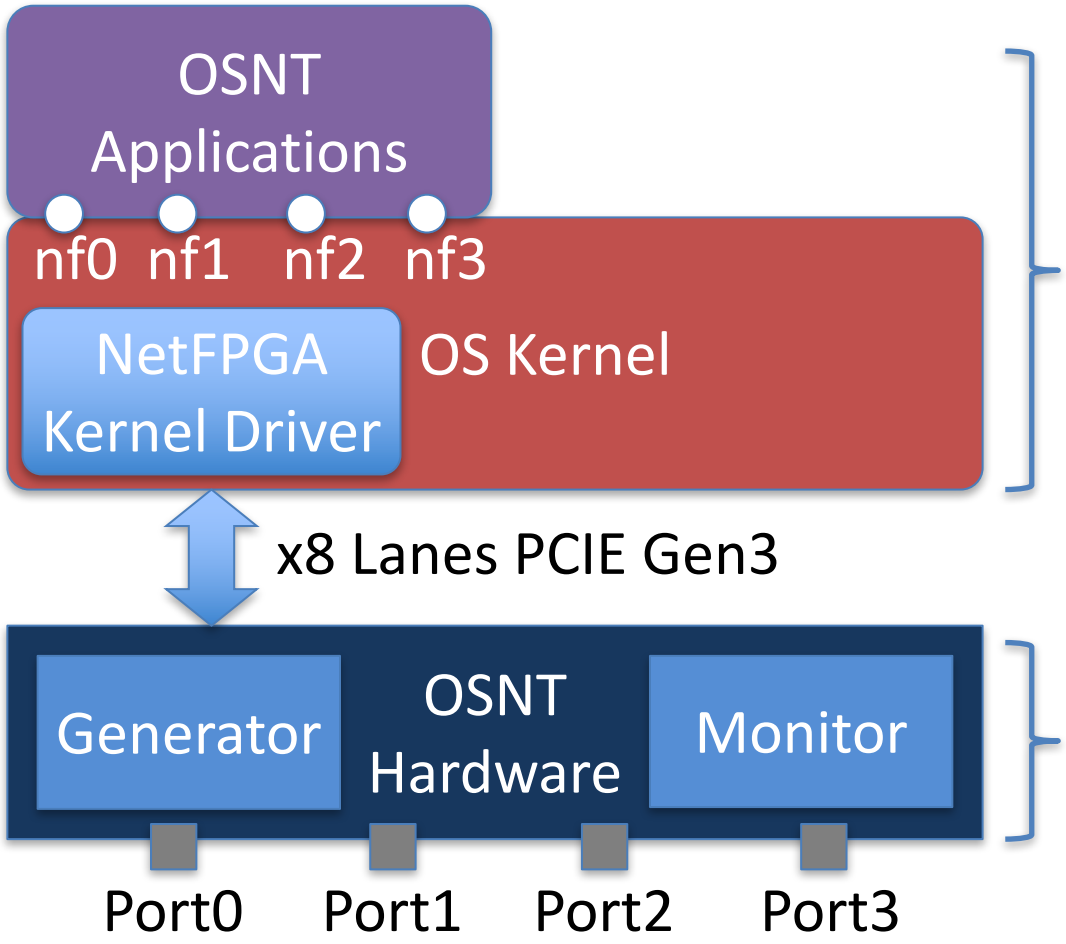
OSNT: Open Source Network Tester, Antichi G. et al., IEEE Network Magazine, Special issue on Open Source for Networking: Tools and Applications, 2014

Network Tester Comparison

	Cost	Flexibility	Precision	Line Rate
ixia SPIRENT endace power to see all	$$$ $$$	✗	✓	✓
DPDK, Moongen	[$]	✓	✗	[✓]
OSNT	$	✓	✓	✓

Open Source Network Tester

- OSNT is an open source HW/SW platform for network testing

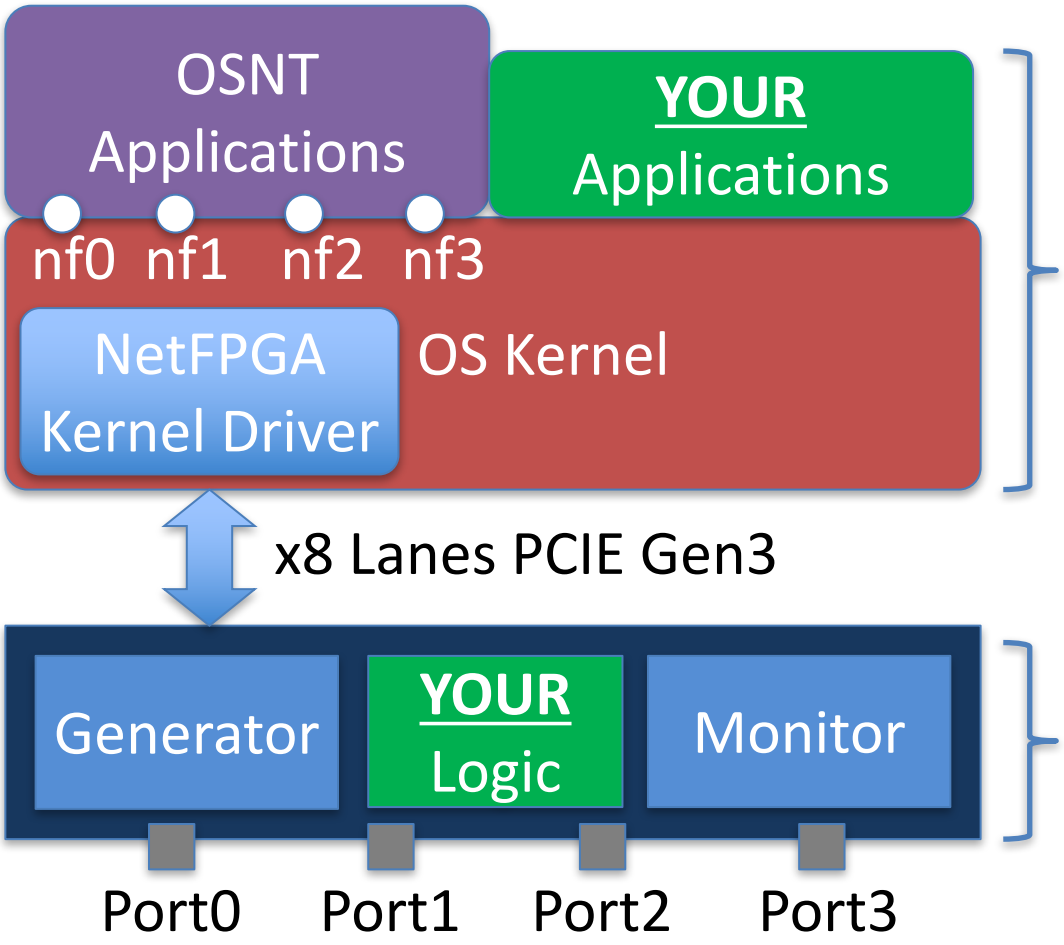


- Written in C, Python
- Open API and registers

- Written in Verilog HDL using standard Xilinx protocols
- Users can add and modify the modules

Open Source Network Tester

- OSNT is an open source HW/SW platform for network testing



- Written in C, Python
- Open API and registers

- Written in Verilog HDL using standard Xilinx protocols
- Users can add and modify the modules

Open Source Network Tester

OSNT **currently** is:

- 4x10Gbps traffic generator.
- Capture card with high resolution timestamp (6.4nsec).
- GPS-ready synchronized measurement kit.

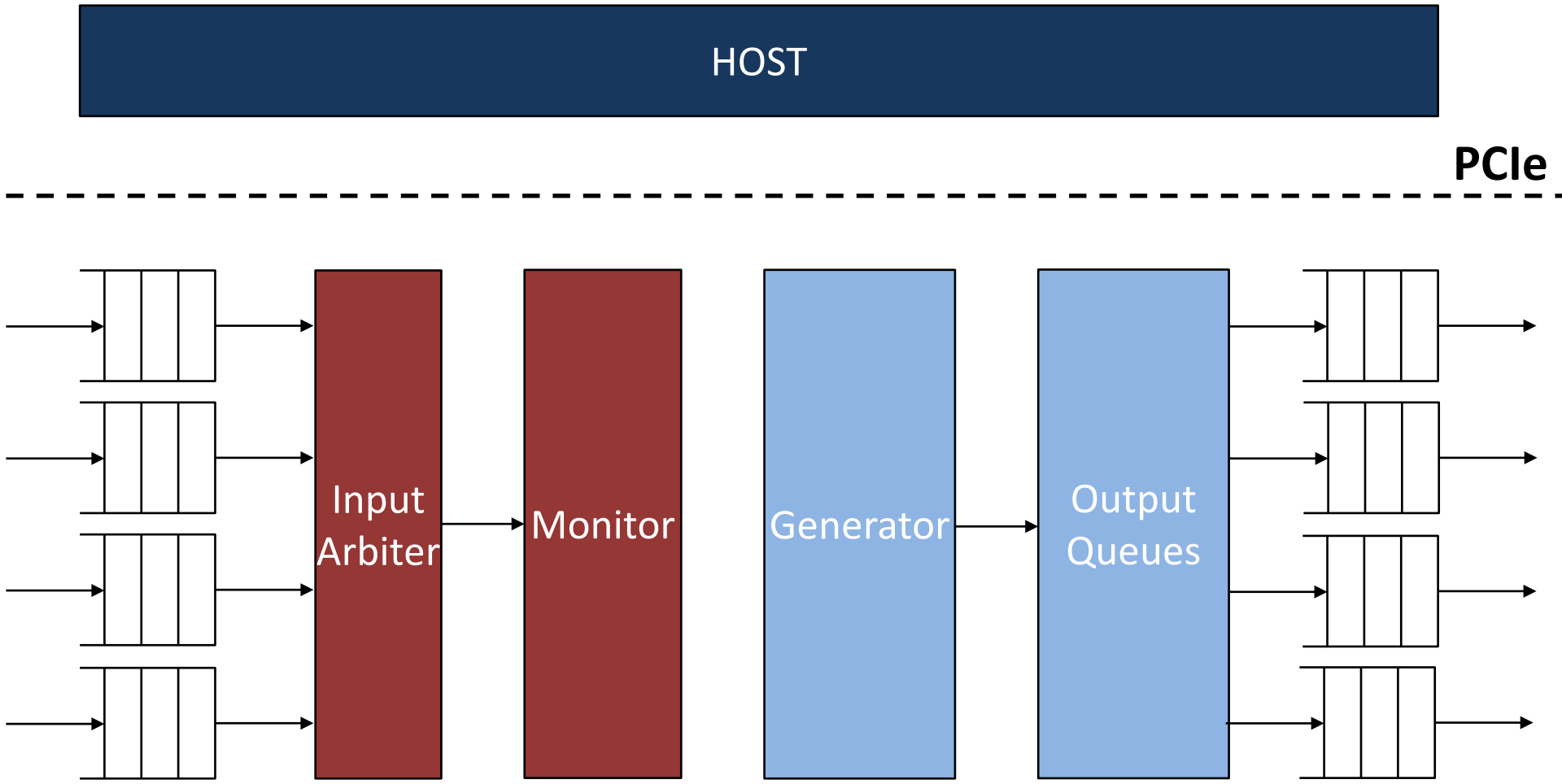
Open Source Network Tester

OSNT **currently** is:

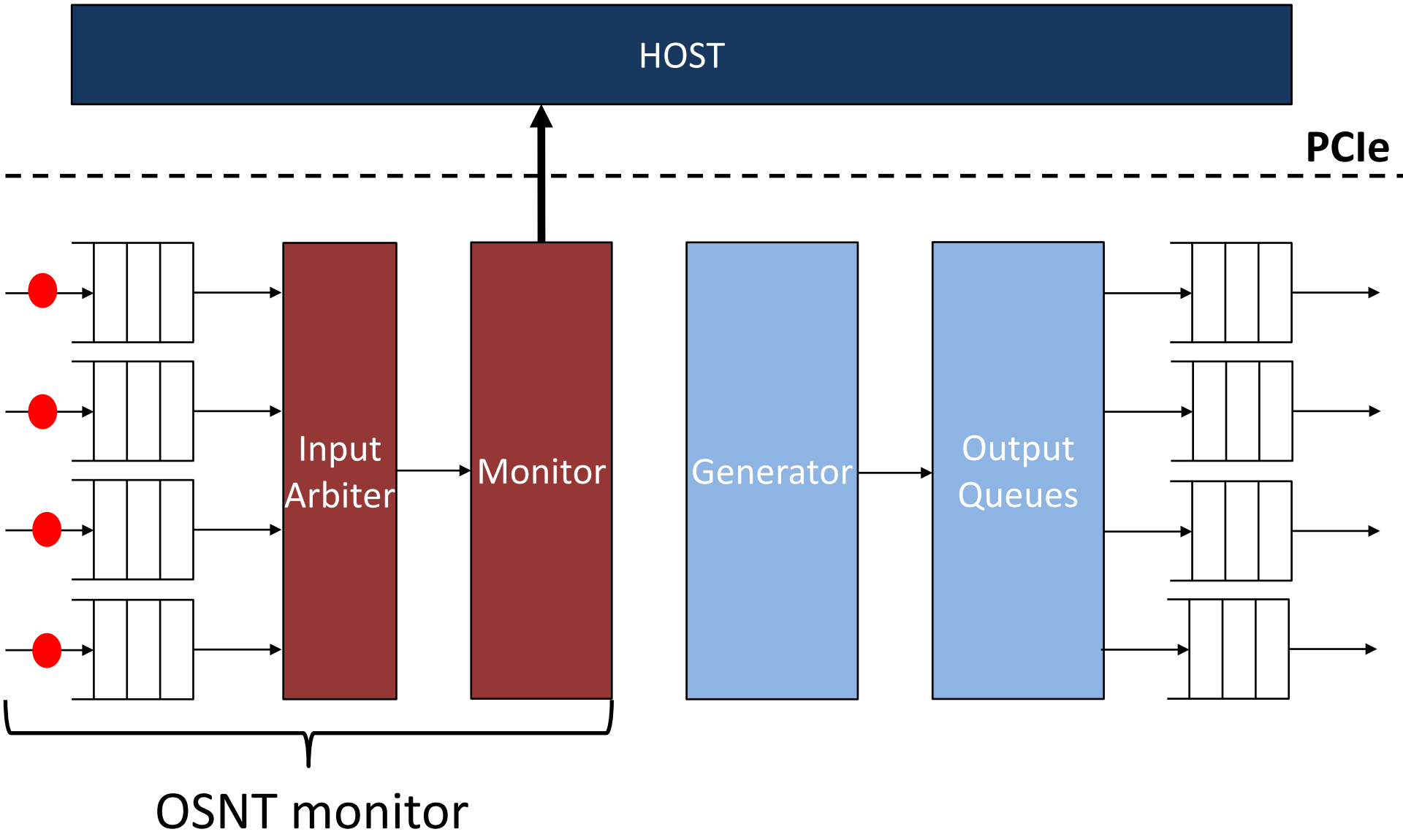
- 4x10Gbps traffic generator.
- Capture card with high resolution timestamp (6.4nsec).
- GPS-ready synchronized measurement kit.

a starting point

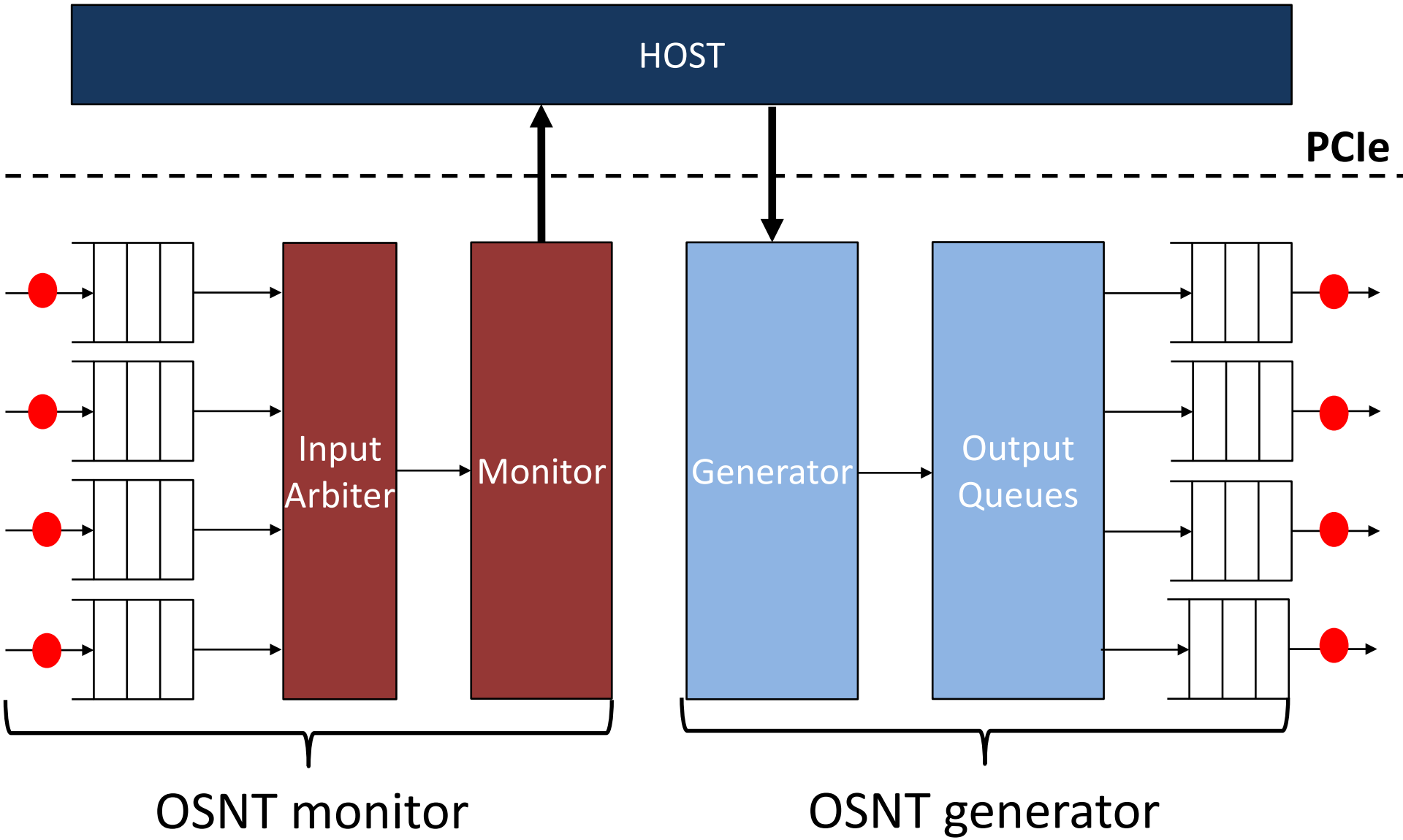
OSNT Architecture



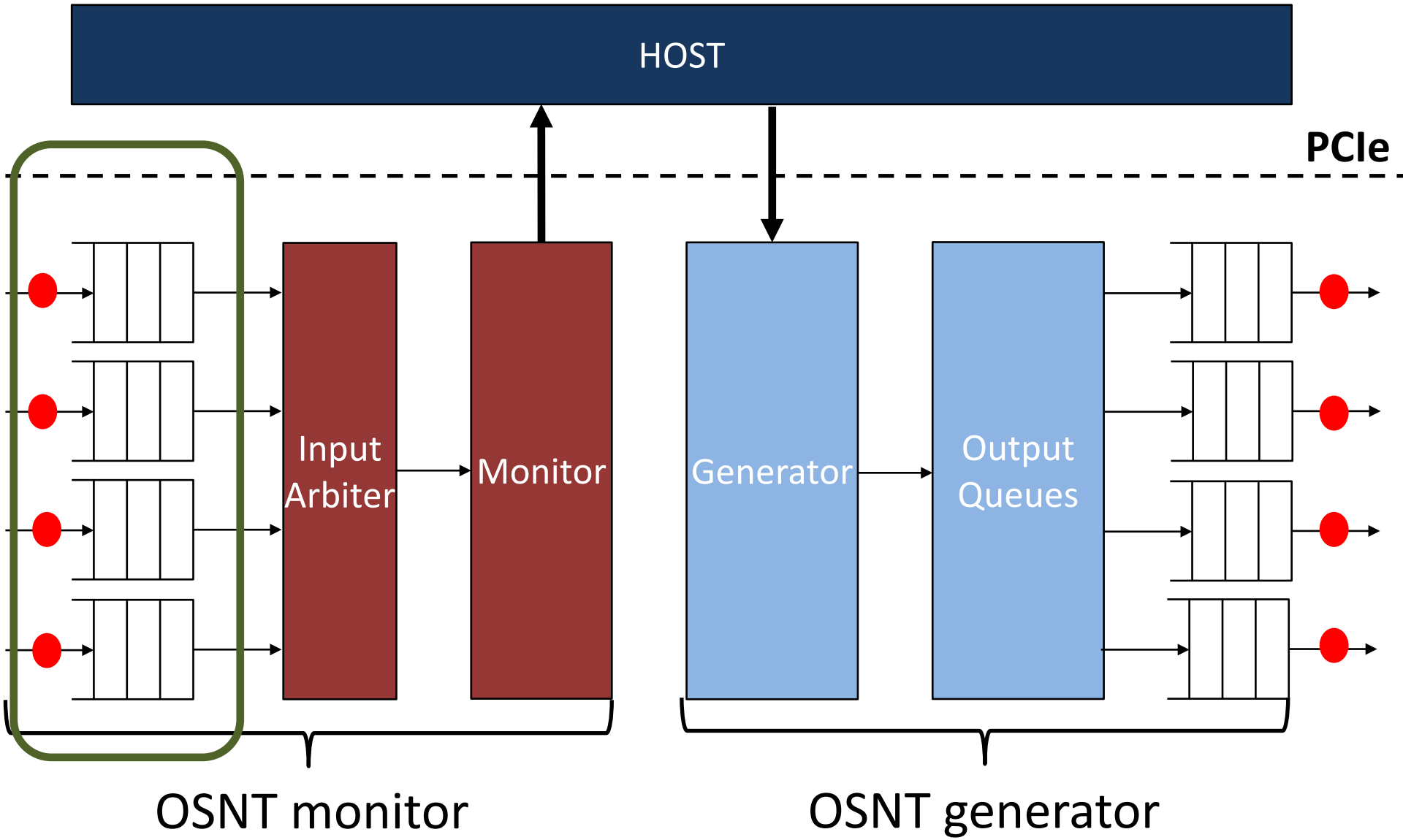
OSNT Architecture



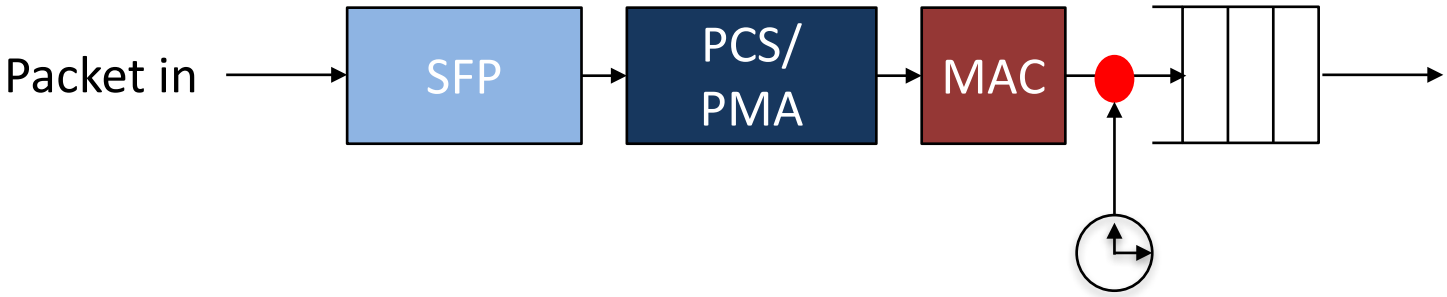
OSNT Architecture



OSNT Architecture

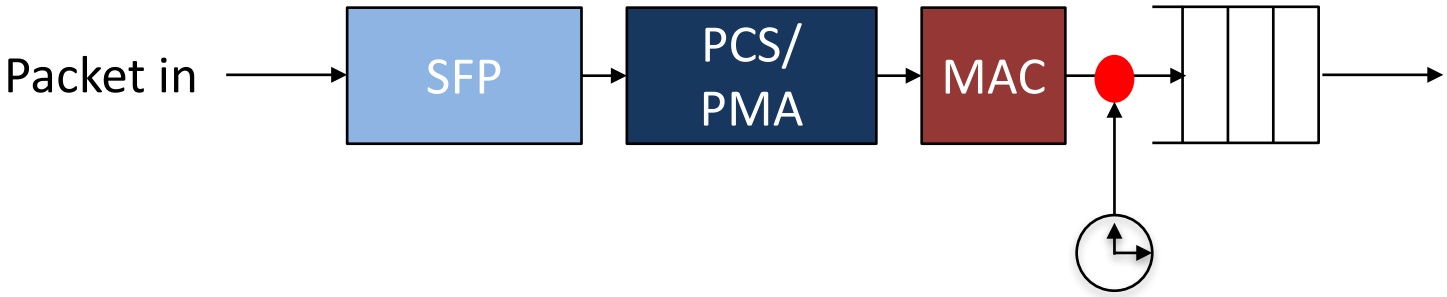


OSNT RX interface



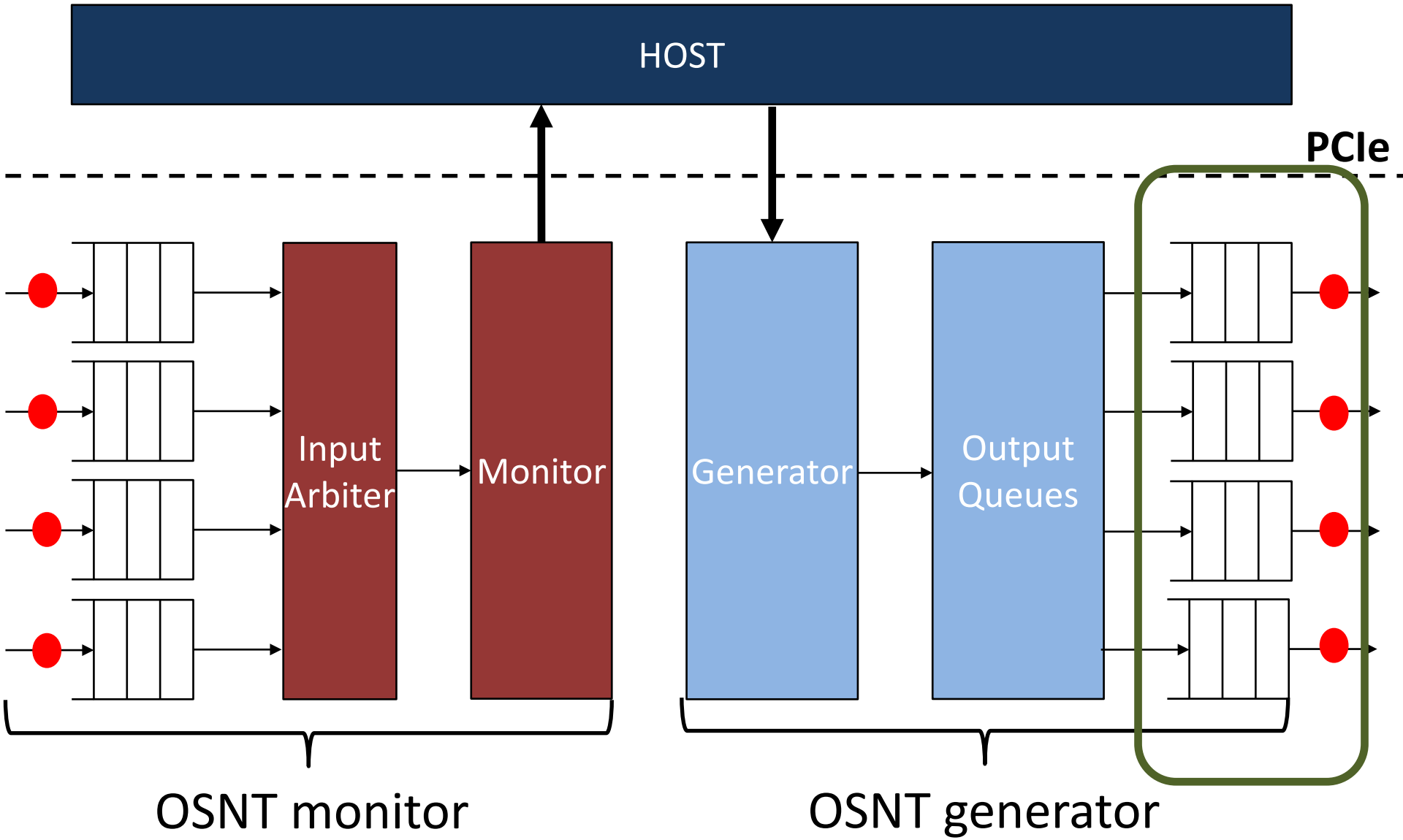
- Timestamp taken before RX queues to reduce FIFO-induced jitter

OSNT RX interface

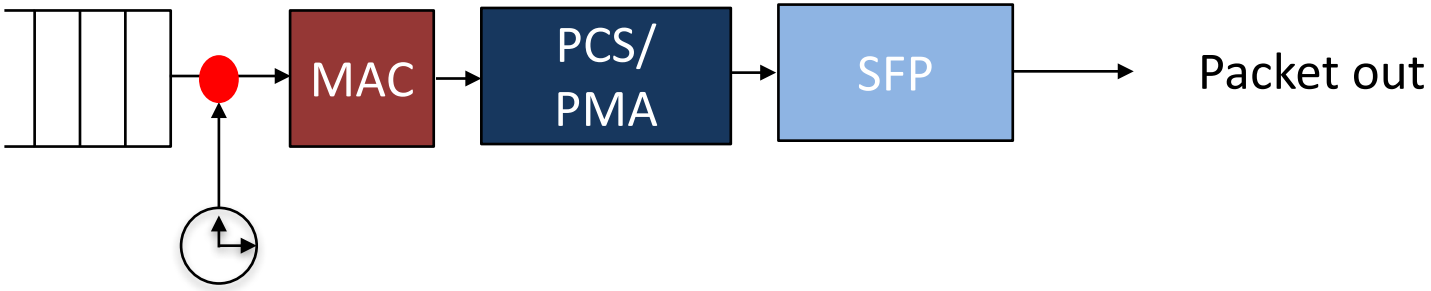


- Timestamp taken before RX queues to reduce FIFO-induced jitter
- Timestamp overwrites packet data at a configurable offset

OSNT Architecture

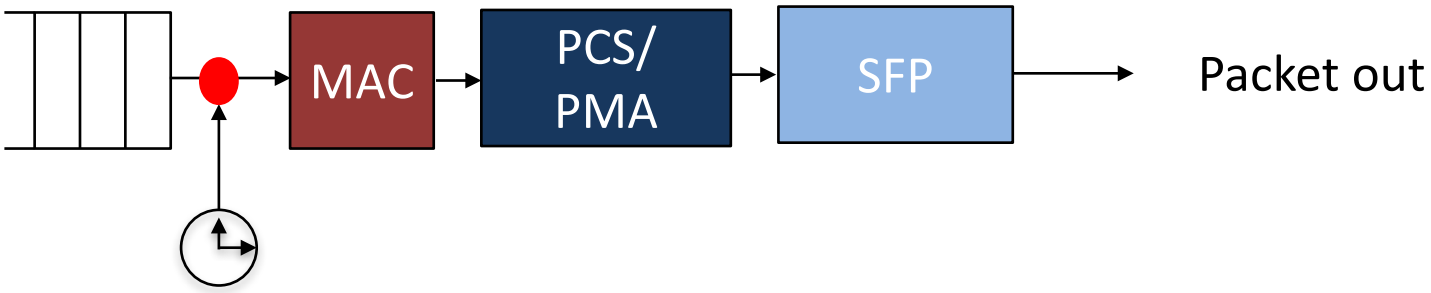


OSNT TX interface



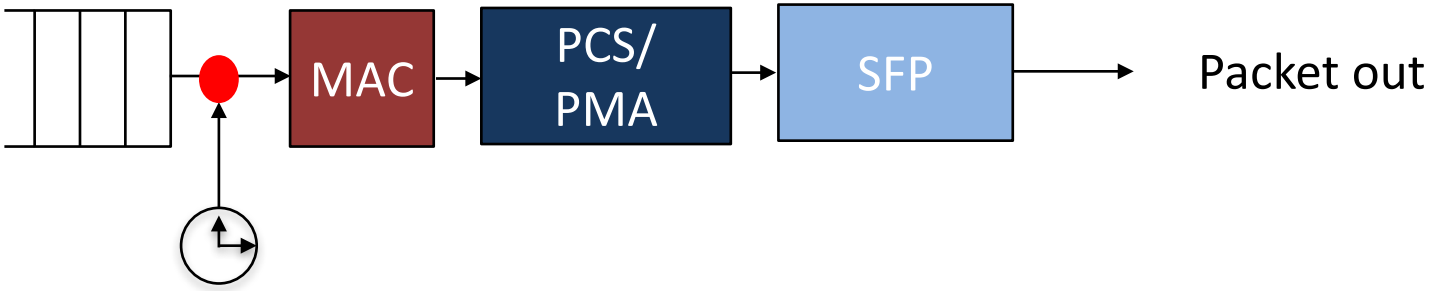
- Timestamp taken after TX queues to reduce FIFO-induced jitter

OSNT TX interface

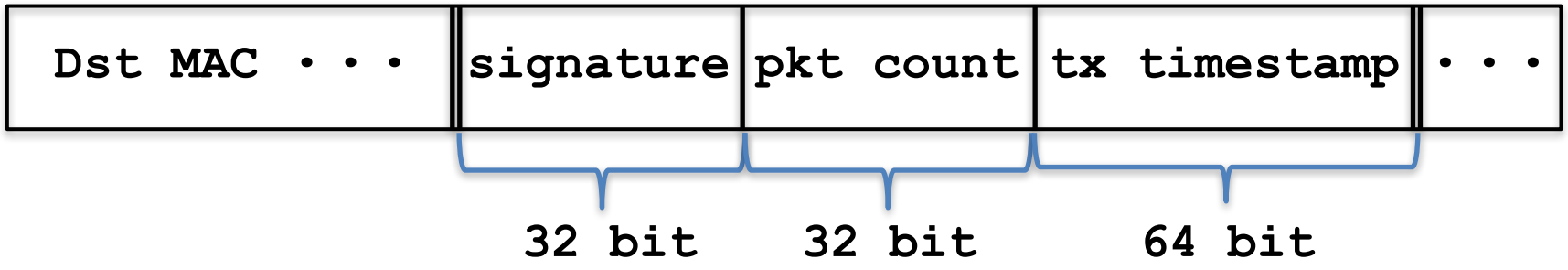


- Timestamp taken after TX queues to reduce FIFO-induced jitter
- Timestamp overwrites packet data at a configurable offset

OSNT TX interface



- Timestamp taken after TX queues to reduce FIFO-induced jitter
- Timestamp overwrites packet data at a configurable offset
- If enabled, it will overwrite 128bit data:



OSNT Timestamp

- Free running counter?

- We could use a 64-bit counter driven by the 156.25MHz system clock (naïve solution)

OSNT Timestamp

- Free running counter?

- We could use a 64-bit counter driven by the 156.25MHz system clock (naïve solution)
  - provides no means by which to correct oscillator frequency drift

OSNT Timestamp

- Free running counter?

- We could use a 64-bit counter driven by the 156.25MHz system clock (naïve solution)
  - provides no means by which to correct oscillator frequency drift
  - produces timestamps expressed in unit of 6.4 ns

OSNT Timestamp

- Free running counter?

- We could use a 64-bit counter driven by the 156.25MHz system clock (naïve solution)
  - provides no means by which to correct oscillator frequency drift
  - produces timestamps expressed in unit of 6.4 ns
  - fixed-point representation of time in seconds more useful to host

OSNT Timestamp

- Direct Digital Synthesis (DDS) is the **solution**!!

- DDS is a technique by which arbitrary variable frequencies can be generated

OSNT Timestamp

- Direct Digital Synthesis (DDS) is the **solution**!!

- DDS is a technique by which arbitrary variable frequencies can be generated
  - need a time reference to correct DDS rate (the GPS provides long-term stability)

OSNT Timestamp

- Direct Digital Synthesis (DDS) is the **solution**!!

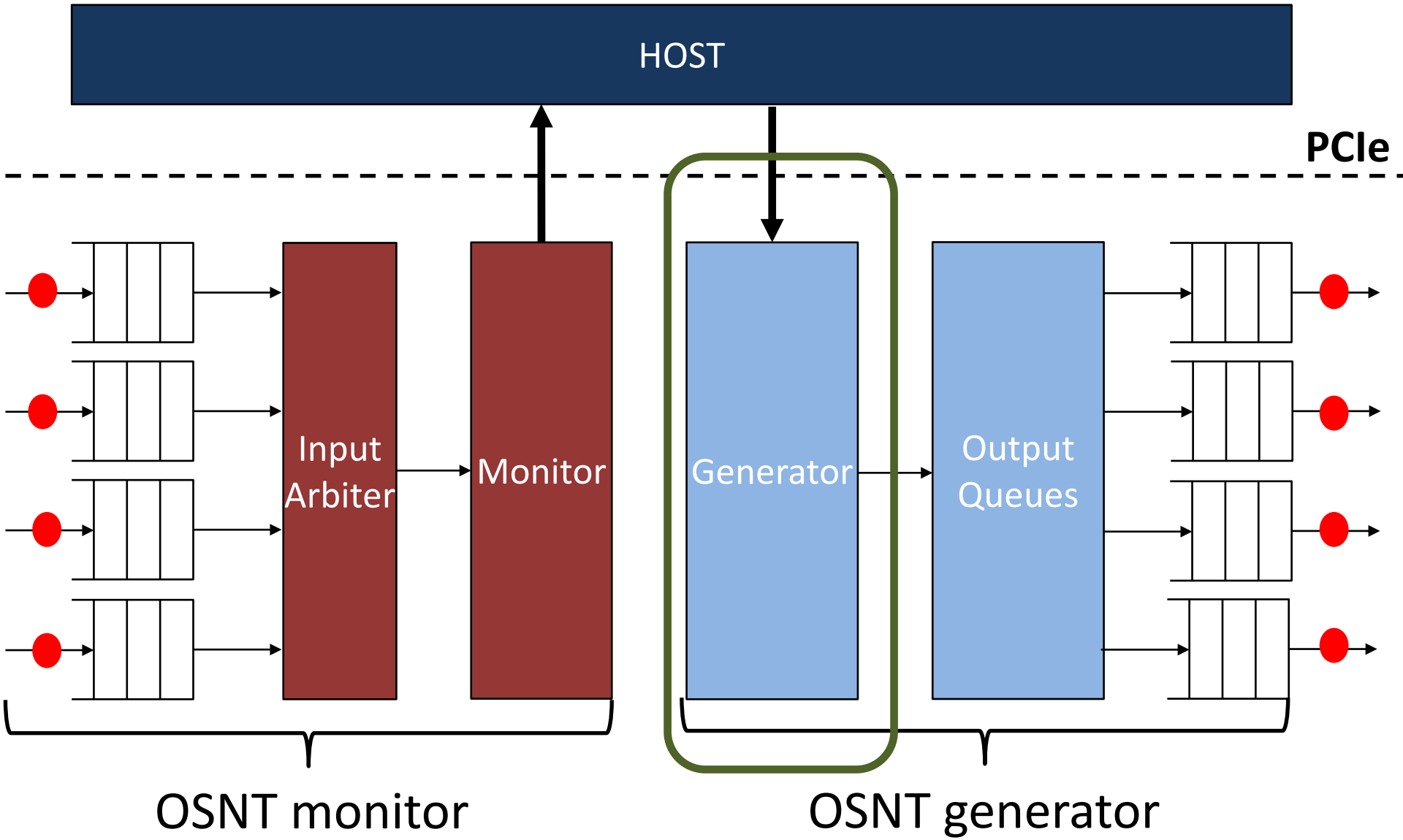
- DDS is a technique by which arbitrary variable frequencies can be generated
  - need a time reference to correct DDS rate (the GPS provides long-term stability)
  - allow 64 bit value in fixed-point representation

OSNT Timestamp

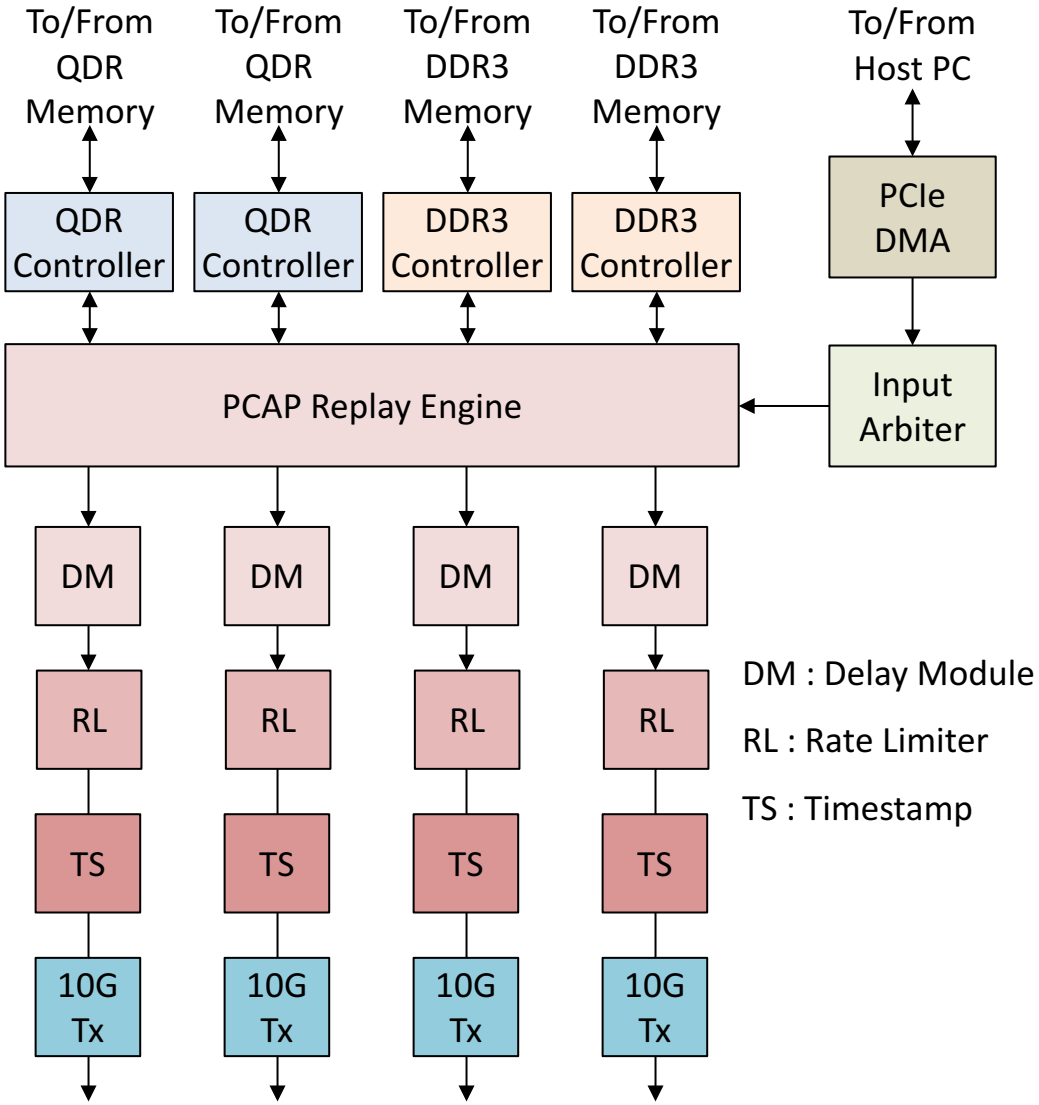
- Direct Digital Synthesis (DDS) is the **solution**!!

- DDS is a technique by which arbitrary variable frequencies can be generated
  - need a time reference to correct DDS rate (the GPS provides long-term stability)
  - allow 64 bit value in fixed-point representation
  - how Endace DAG card works!

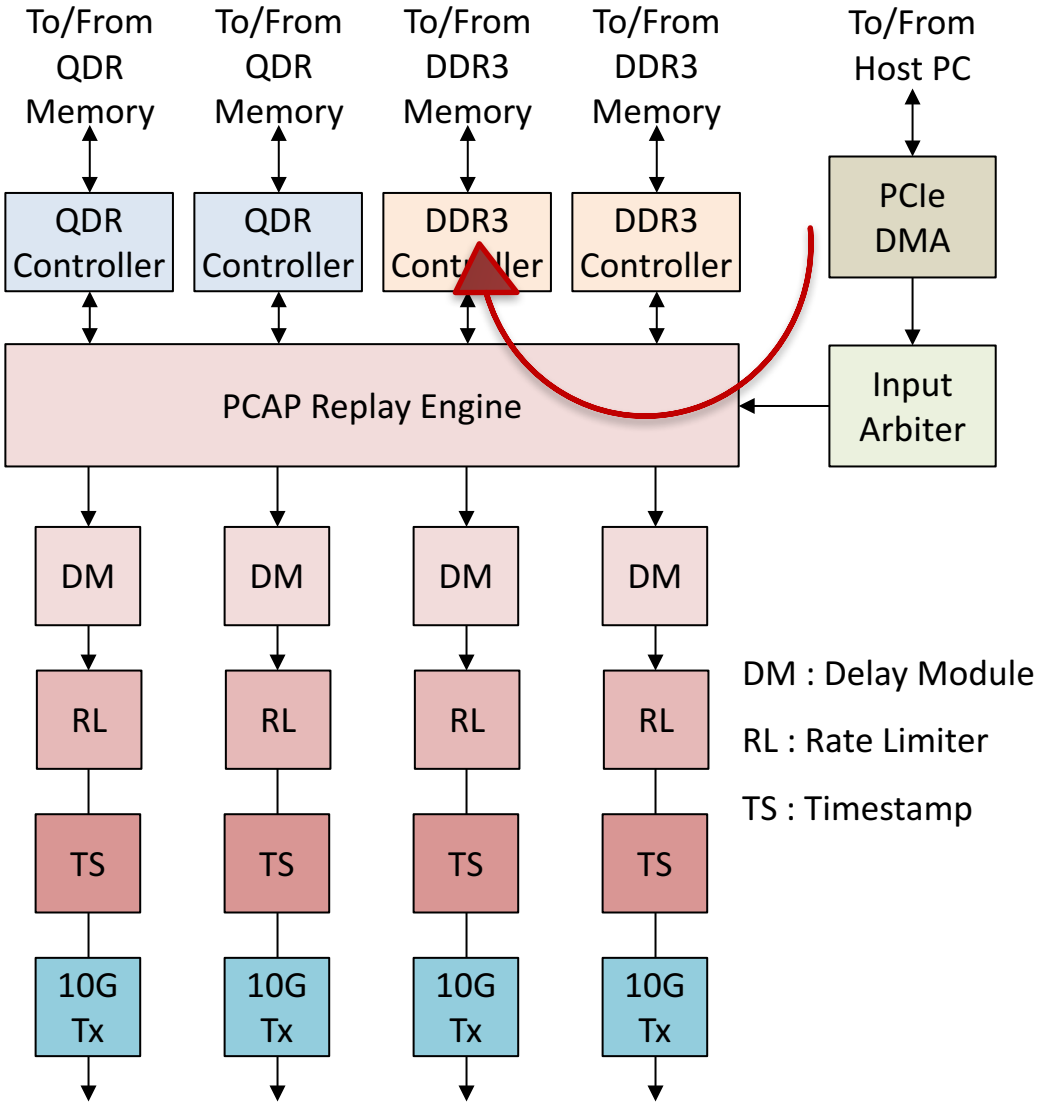
OSNT Architecture



OSNT Generator

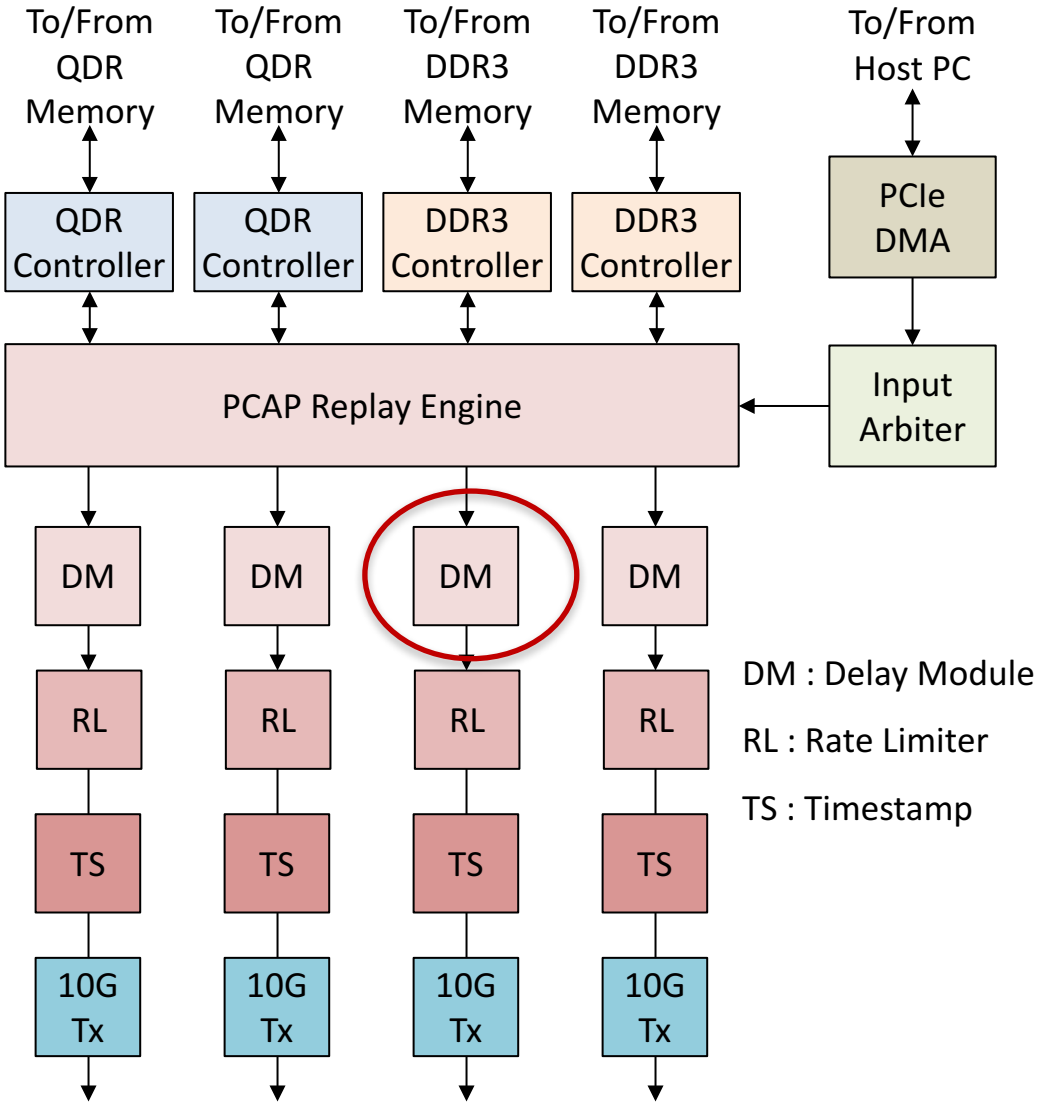


OSNT Generator



- 4x10G PCAP replay engine
- SRAM: 27MB
- DRAM: 8GB

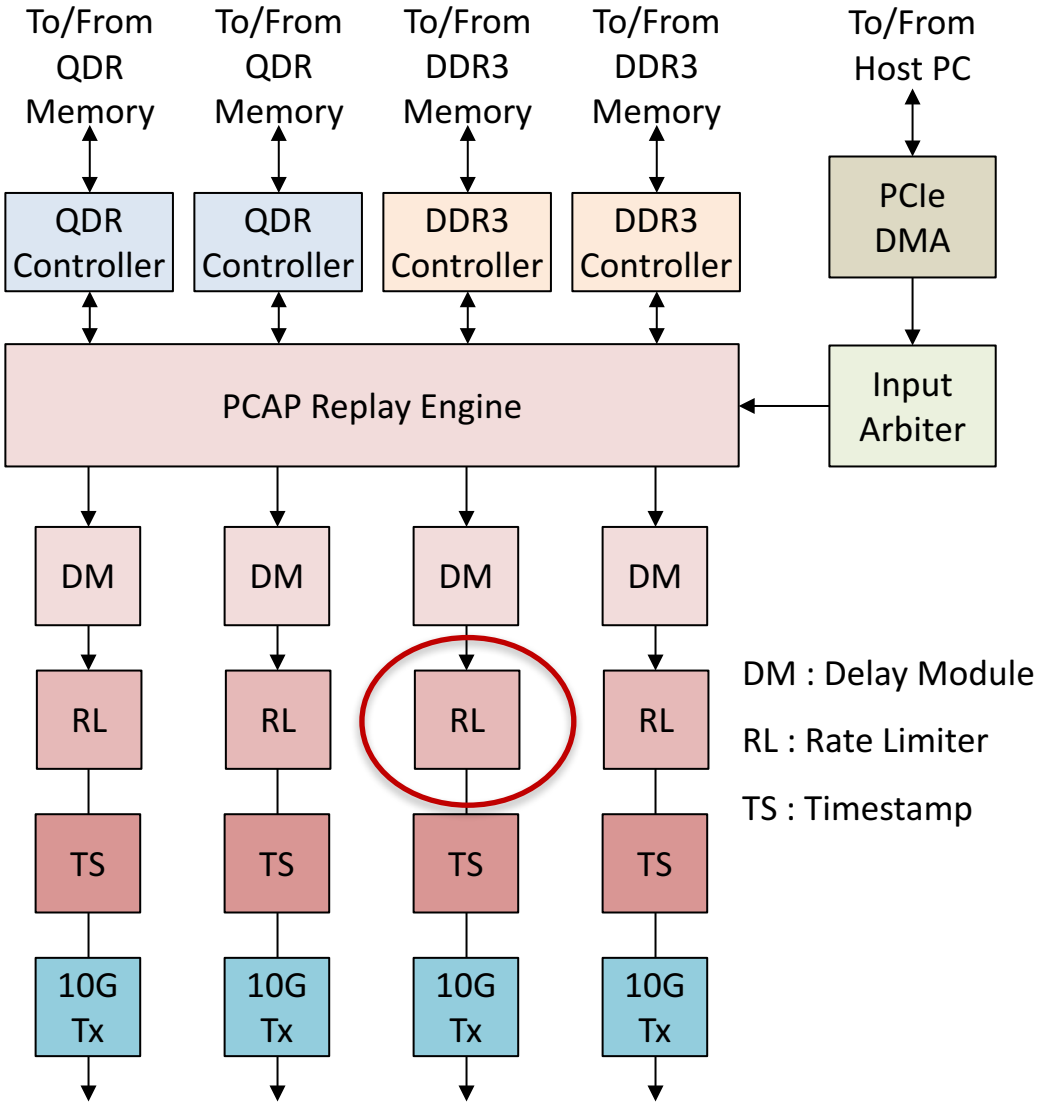
OSNT Generator



- 4x10G PCAP replay engine
- SRAM: 27MB
- DRAM: 8GB

- Delay module

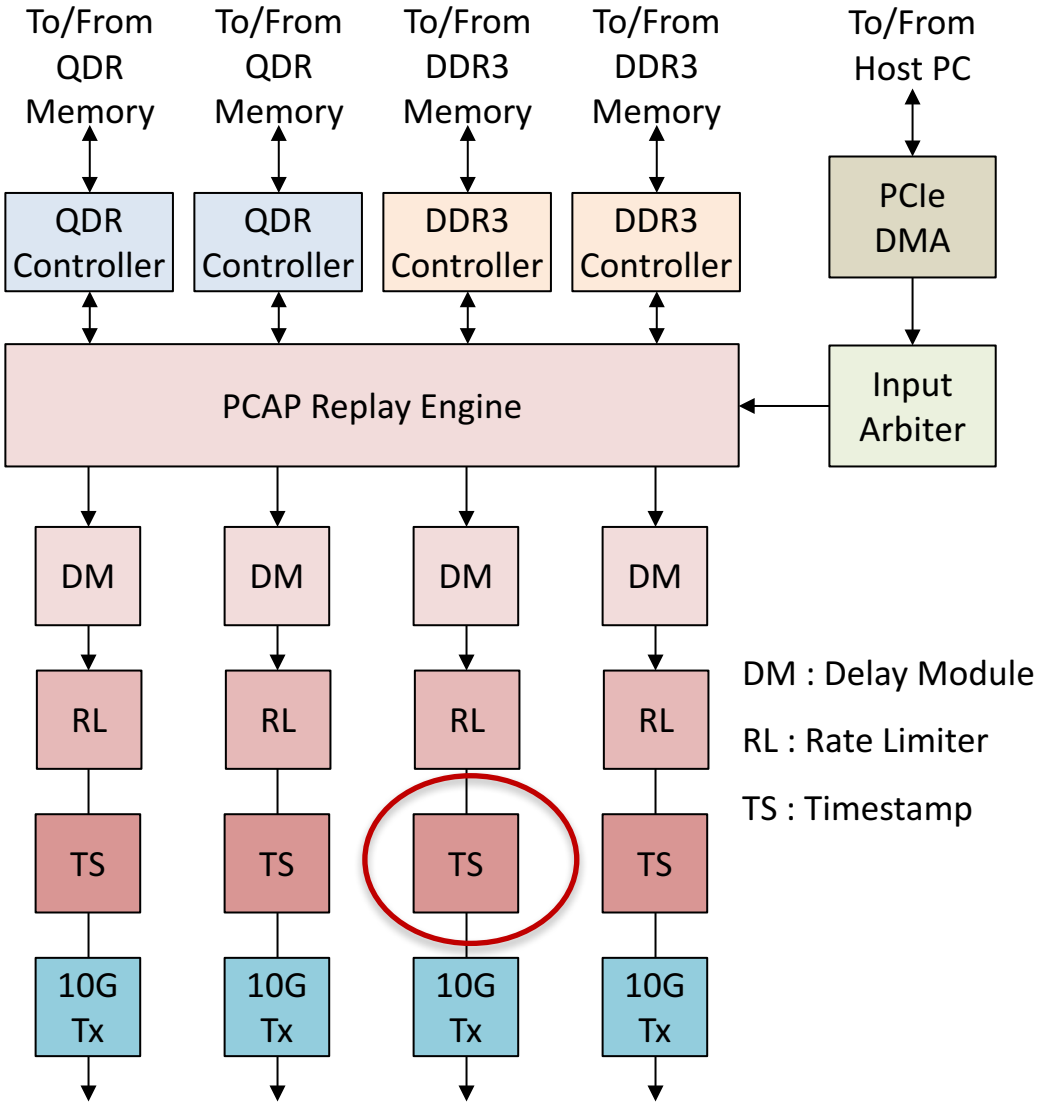
OSNT Generator



- 4x10G PCAP replay engine
- SRAM: 27MB
- DRAM: 8GB

- Delay module
- Rate limiter

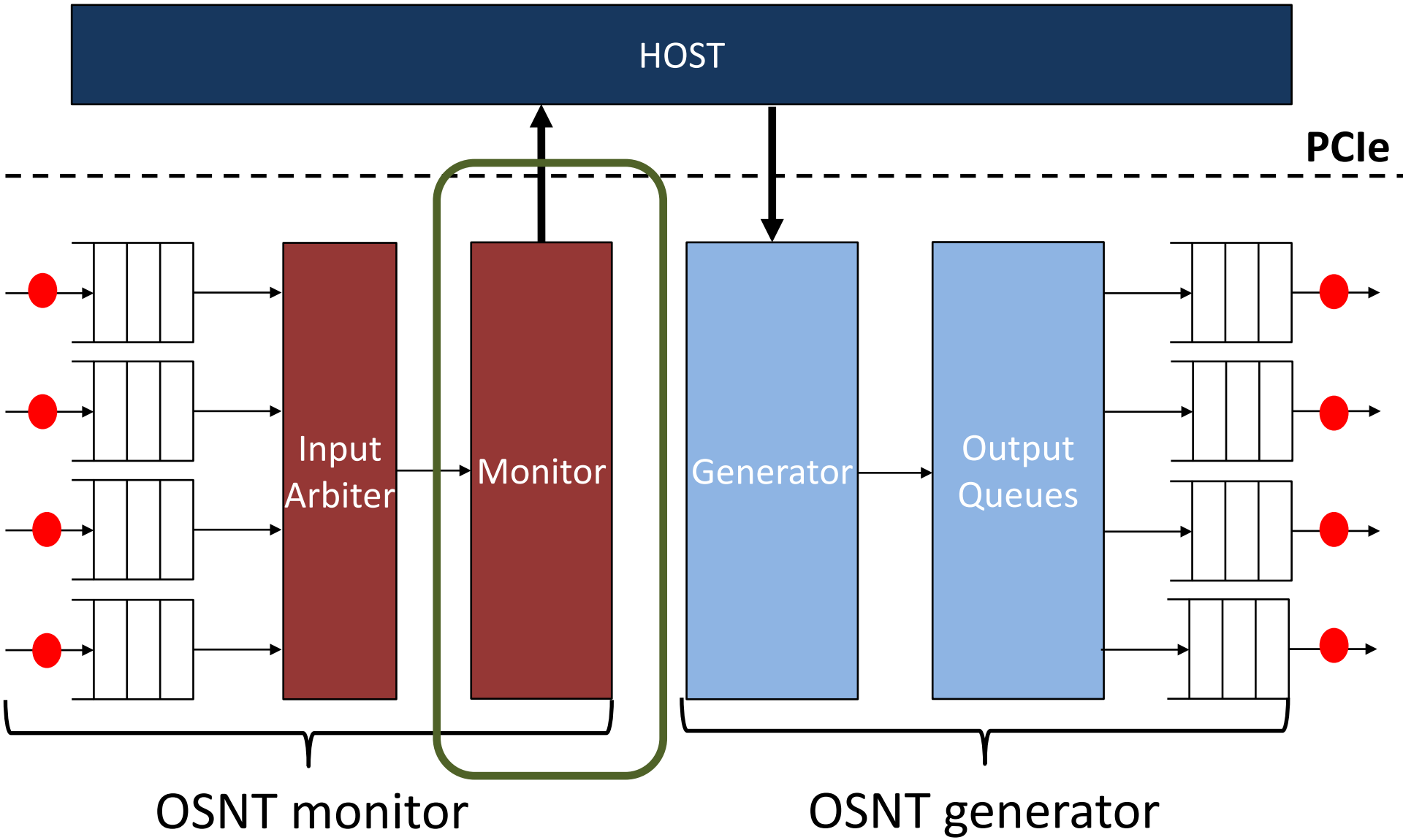
OSNT Generator



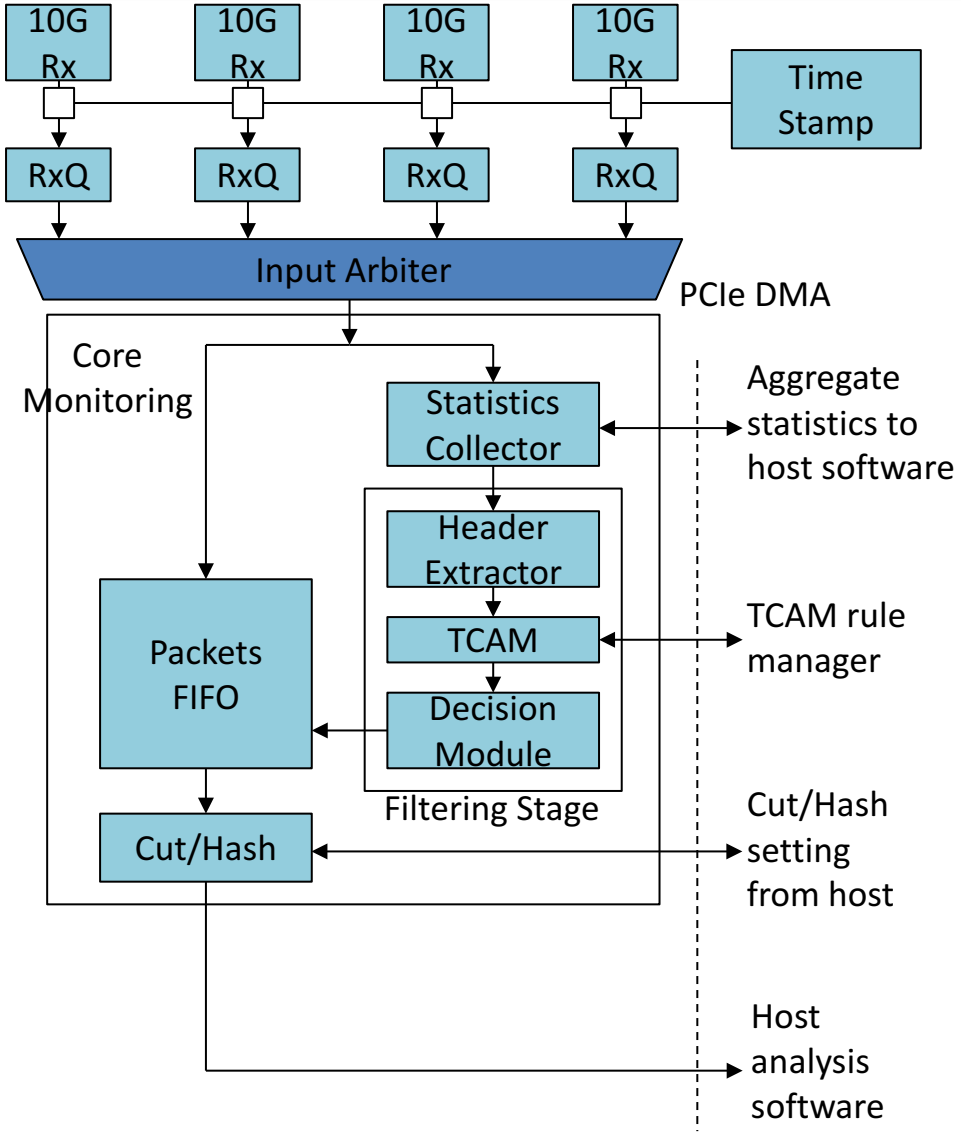
- 4x10G PCAP replay engine
- SRAM: 27MB
- DRAM: 8GB

- Delay module
- Rate limiter
- TX timestamping

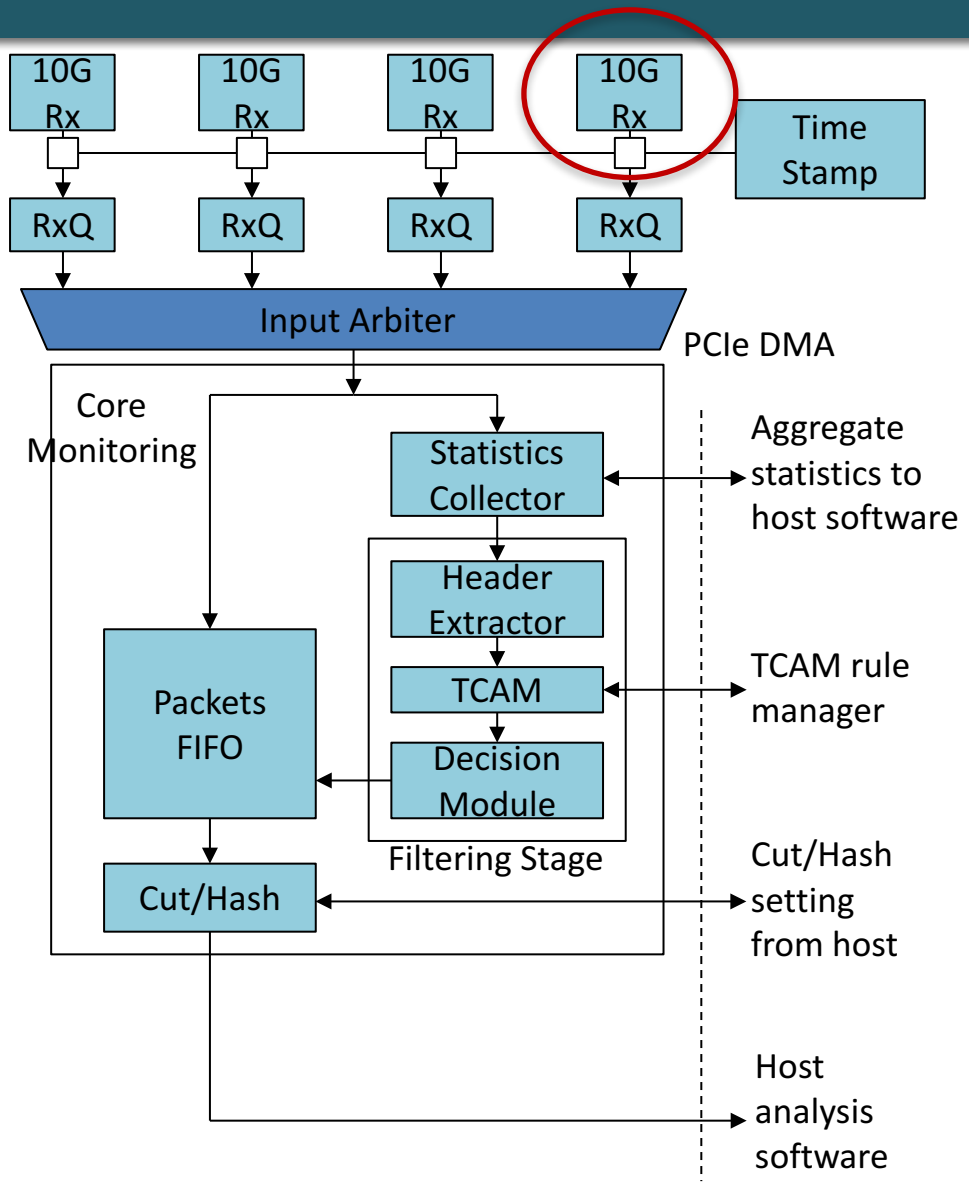
OSNT Architeture



OSNT Monitor

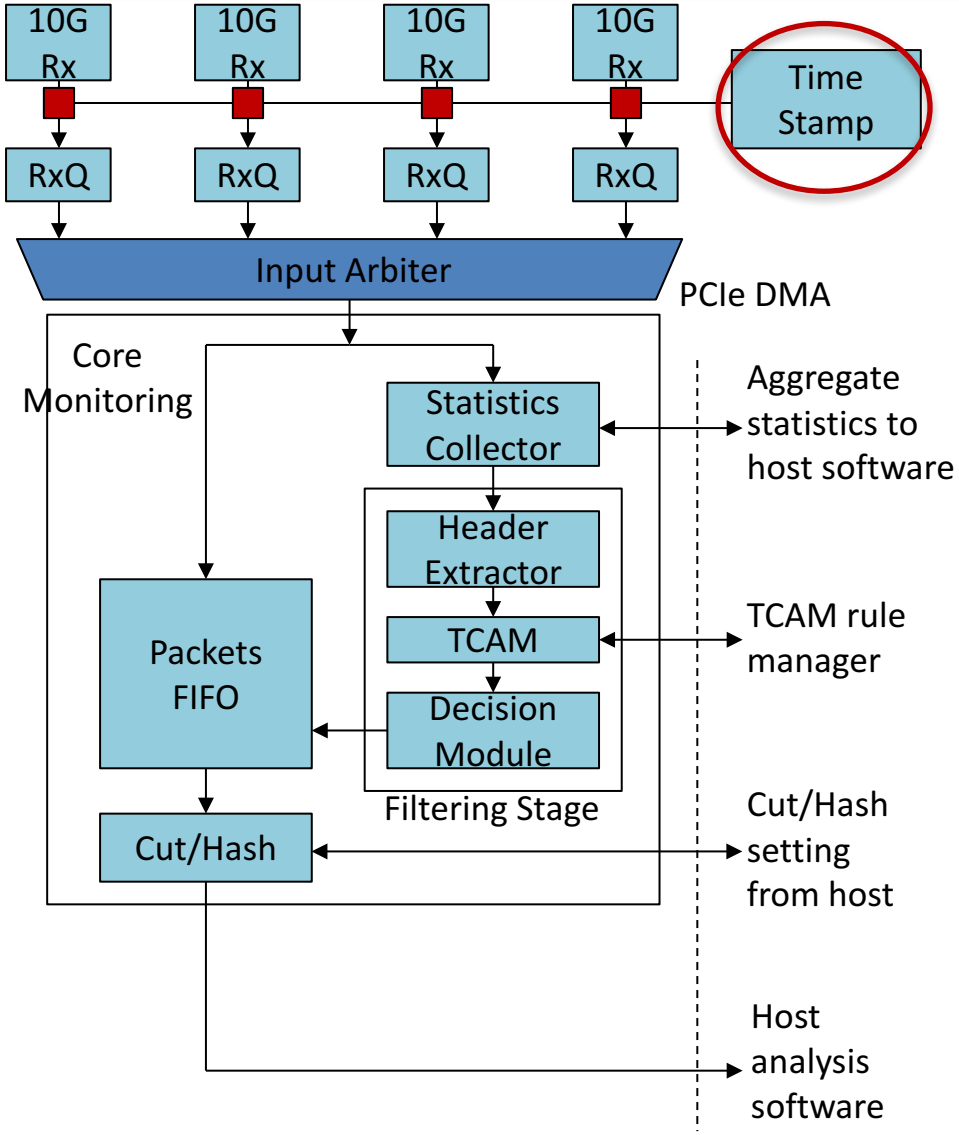


OSNT Monitor



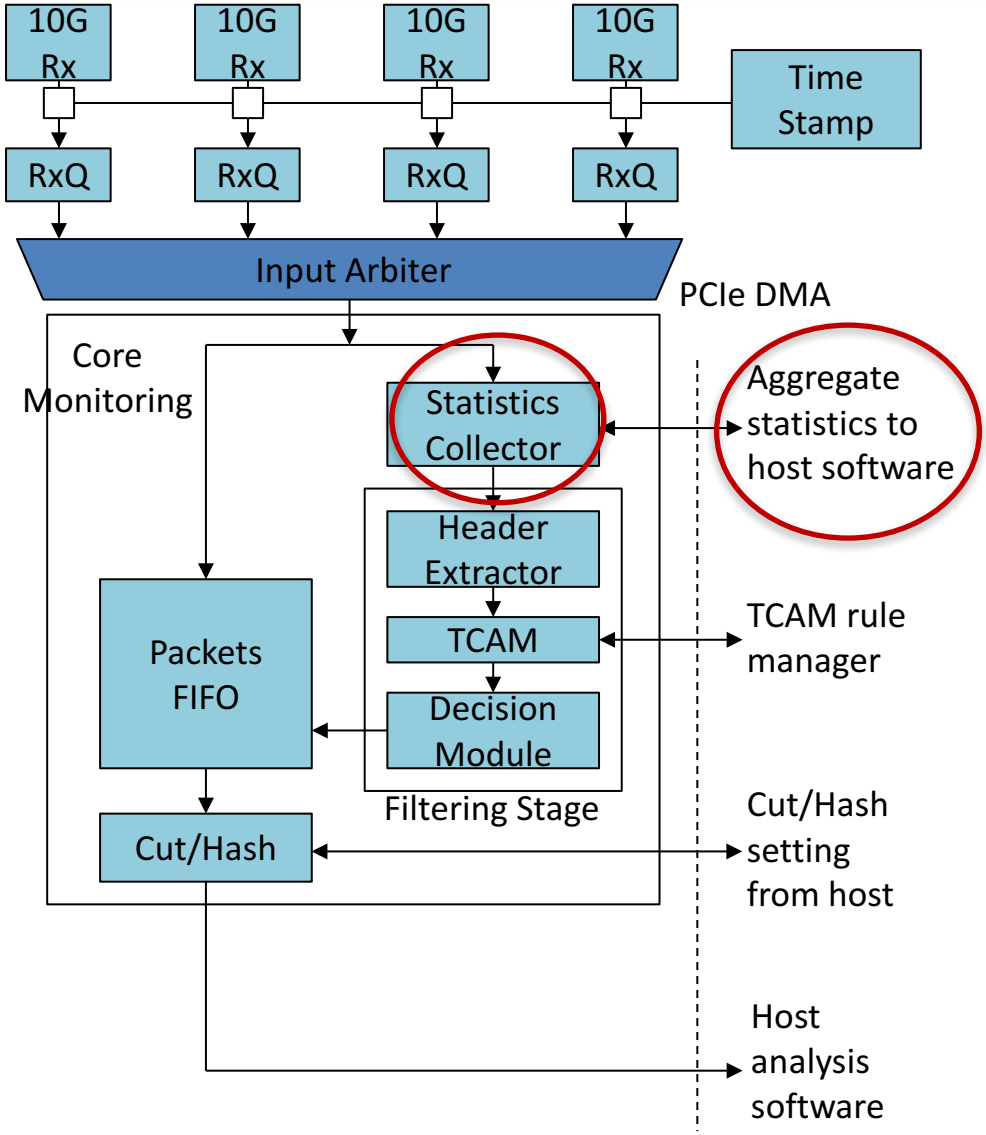
packet entering the board

OSNT Monitor



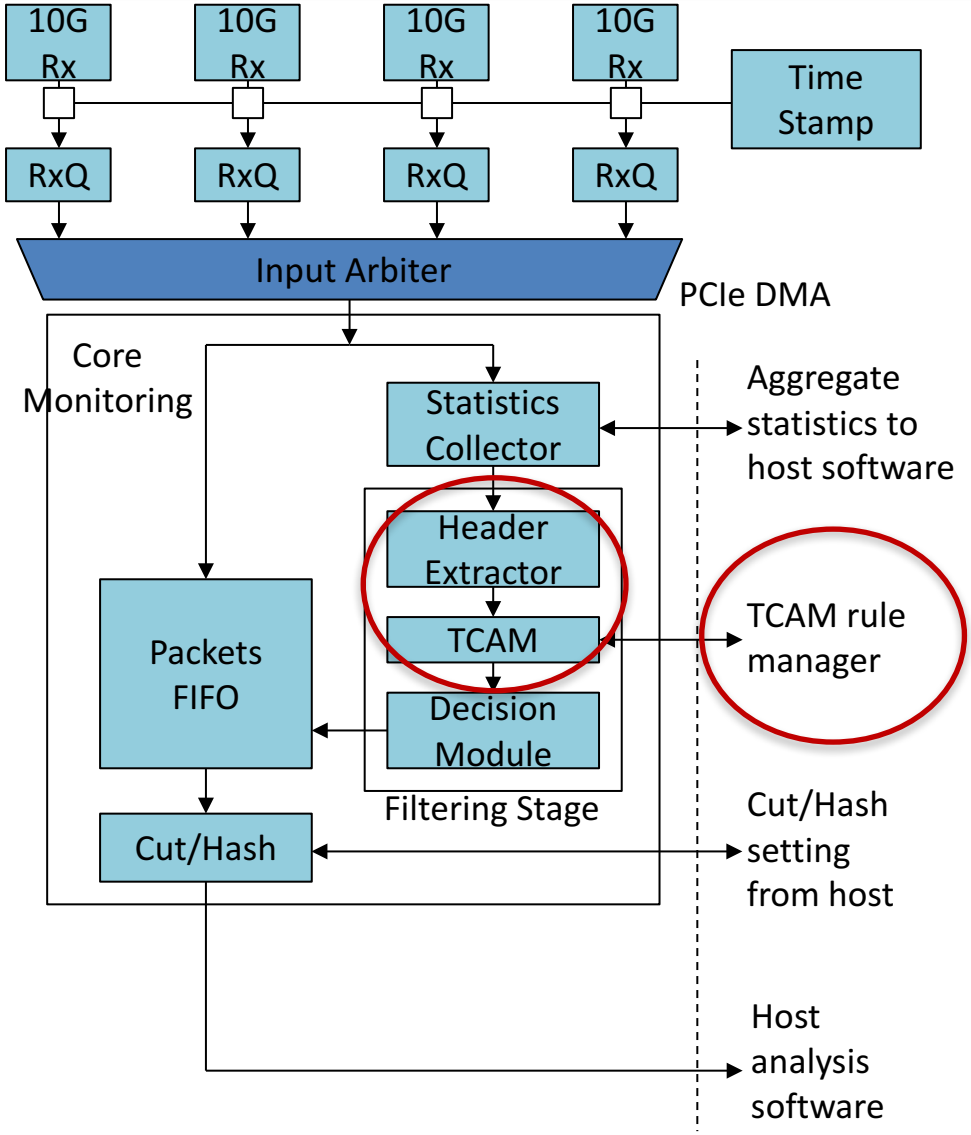
- RX timestamp

OSNT Monitor



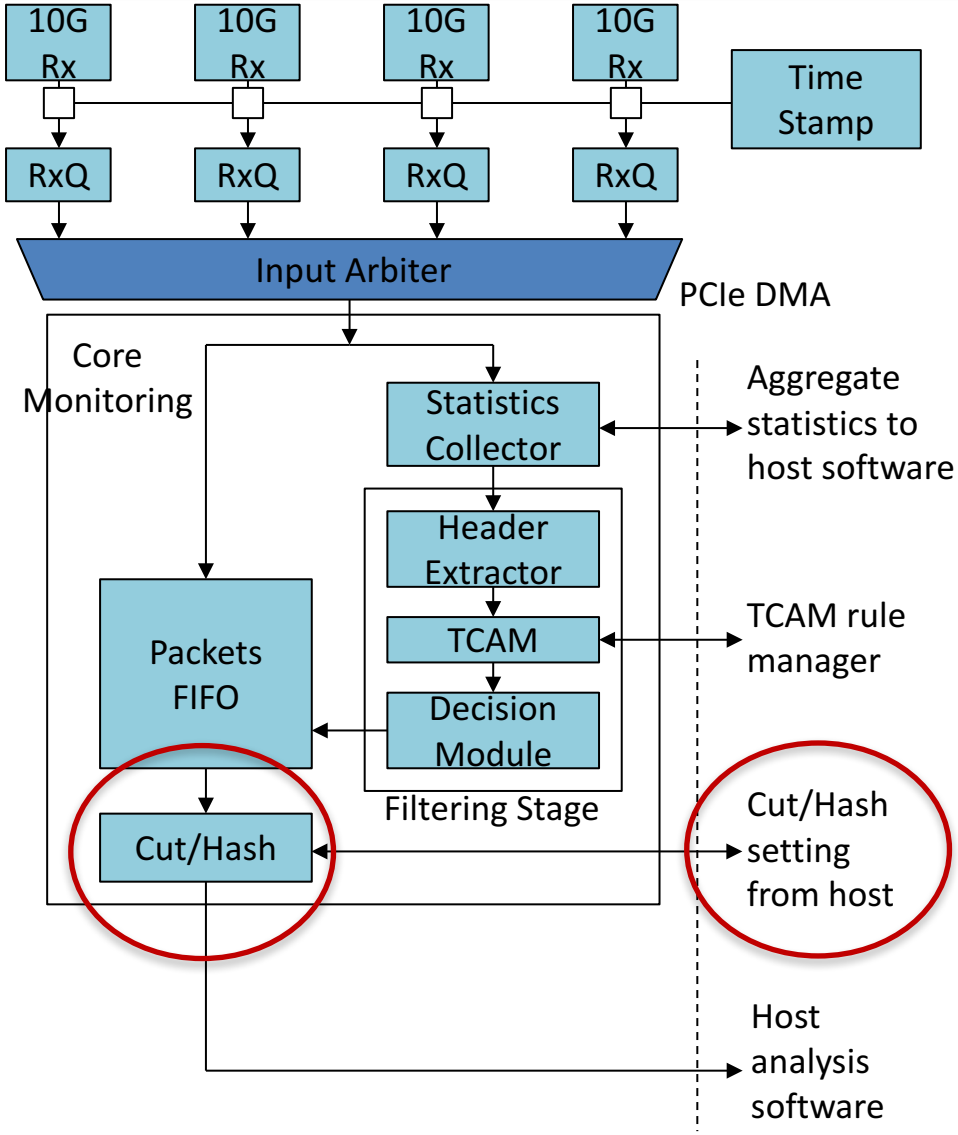
- RX timestamp
- Stats collector

OSNT Monitor



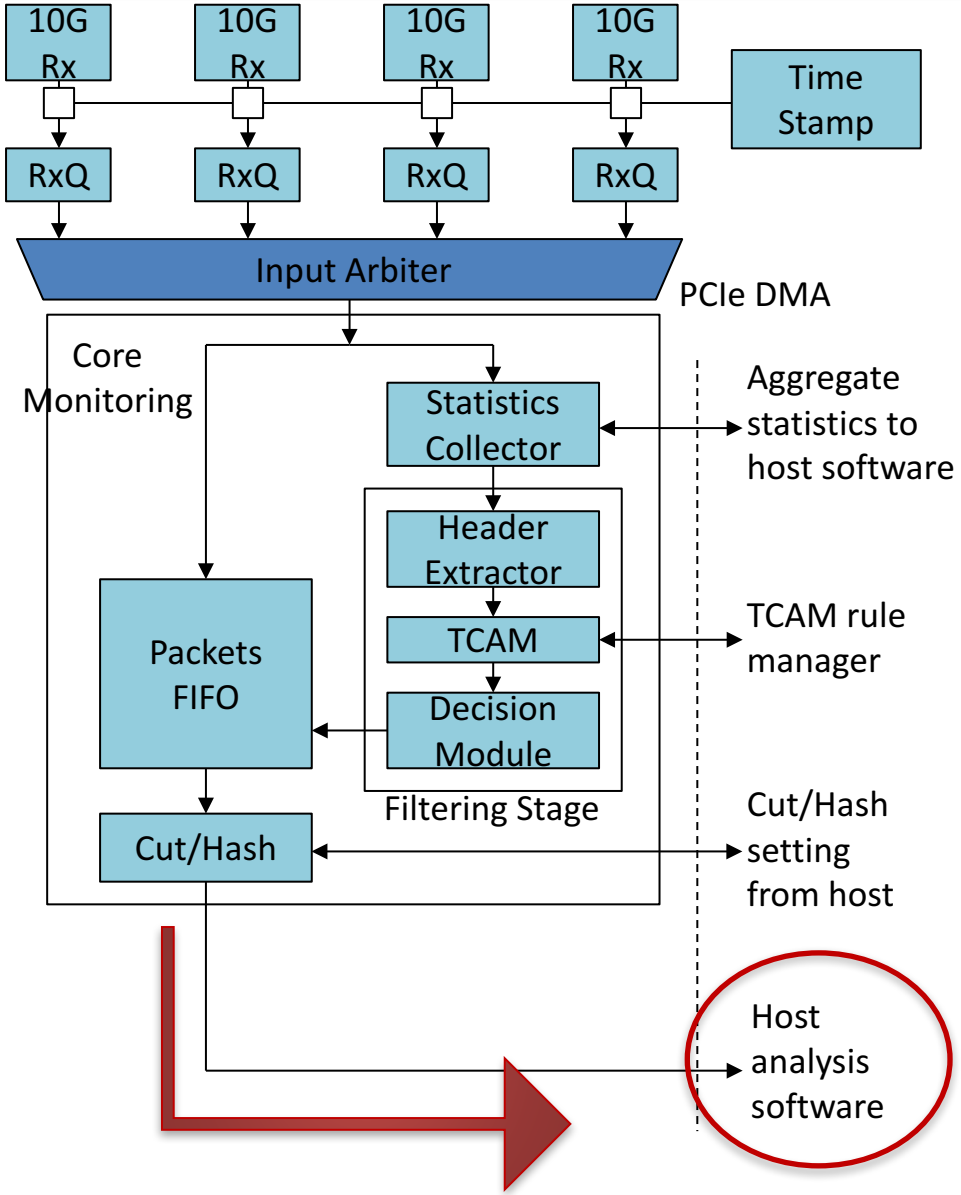
- RX timestamp
- Stats collector
- TCAM-based Packet Filter (5-tuple)

OSNT Monitor



- RX timestamp
- Stats collector
- TCAM-based Packet Filter (5-tuple)
- Cut-Hash function

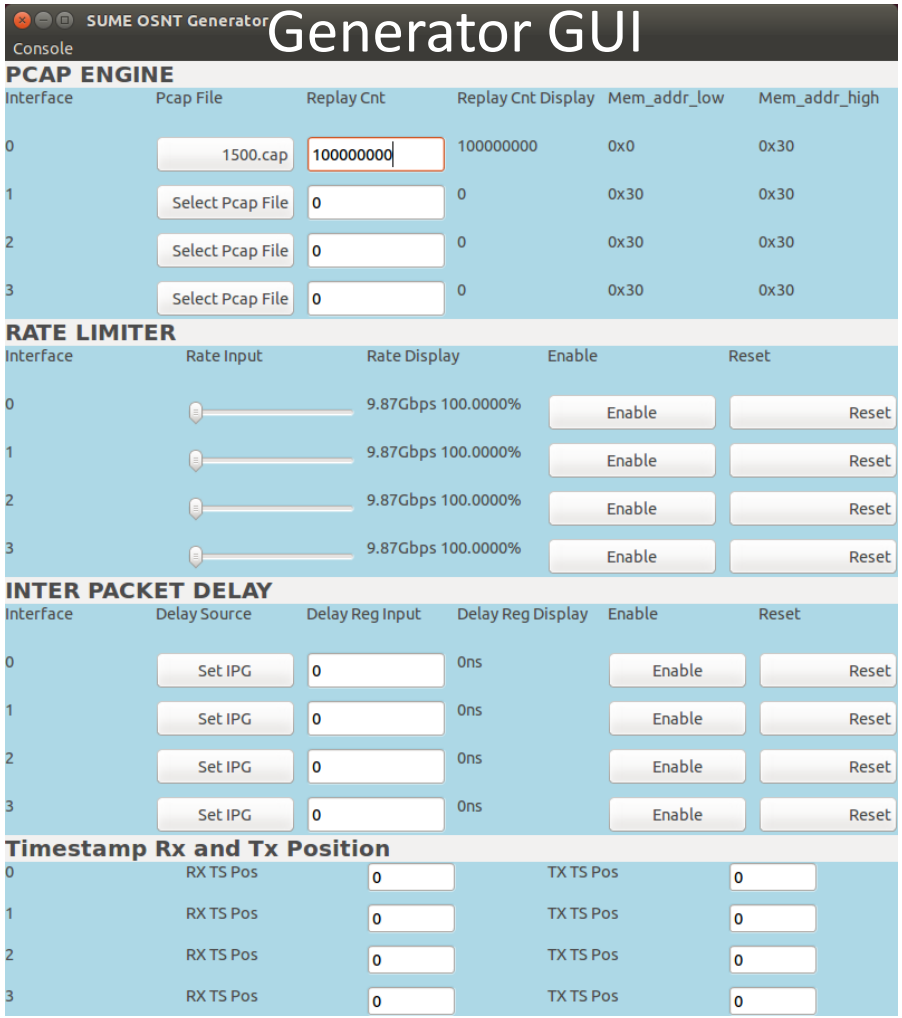
OSNT Monitor

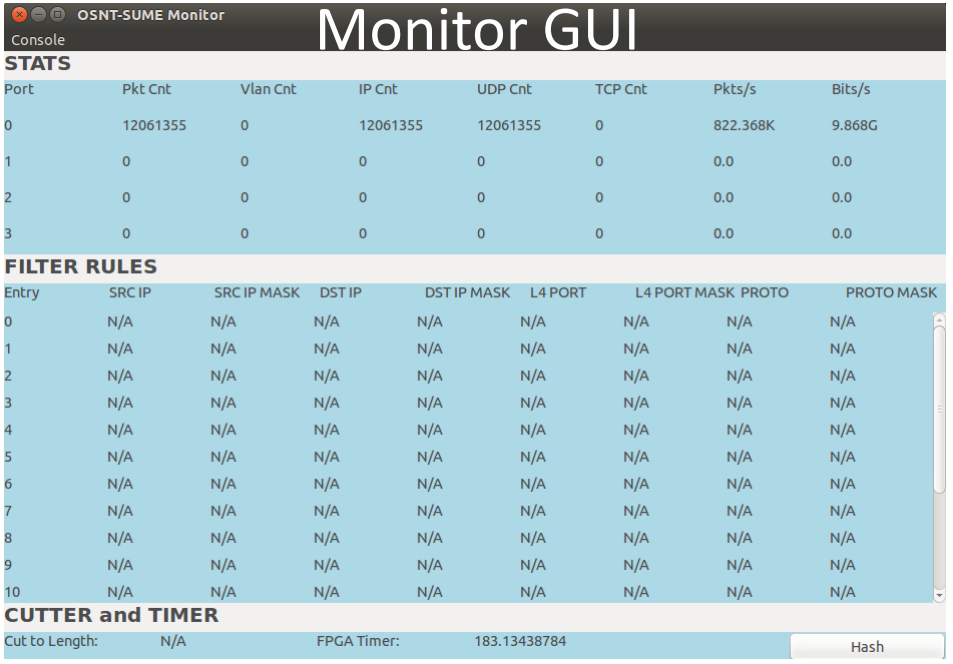


- RX timestamp
- Stats collector
- TCAM-based Packet Filter (5-tuple)
- Cut-Hash function

OSNT GUI

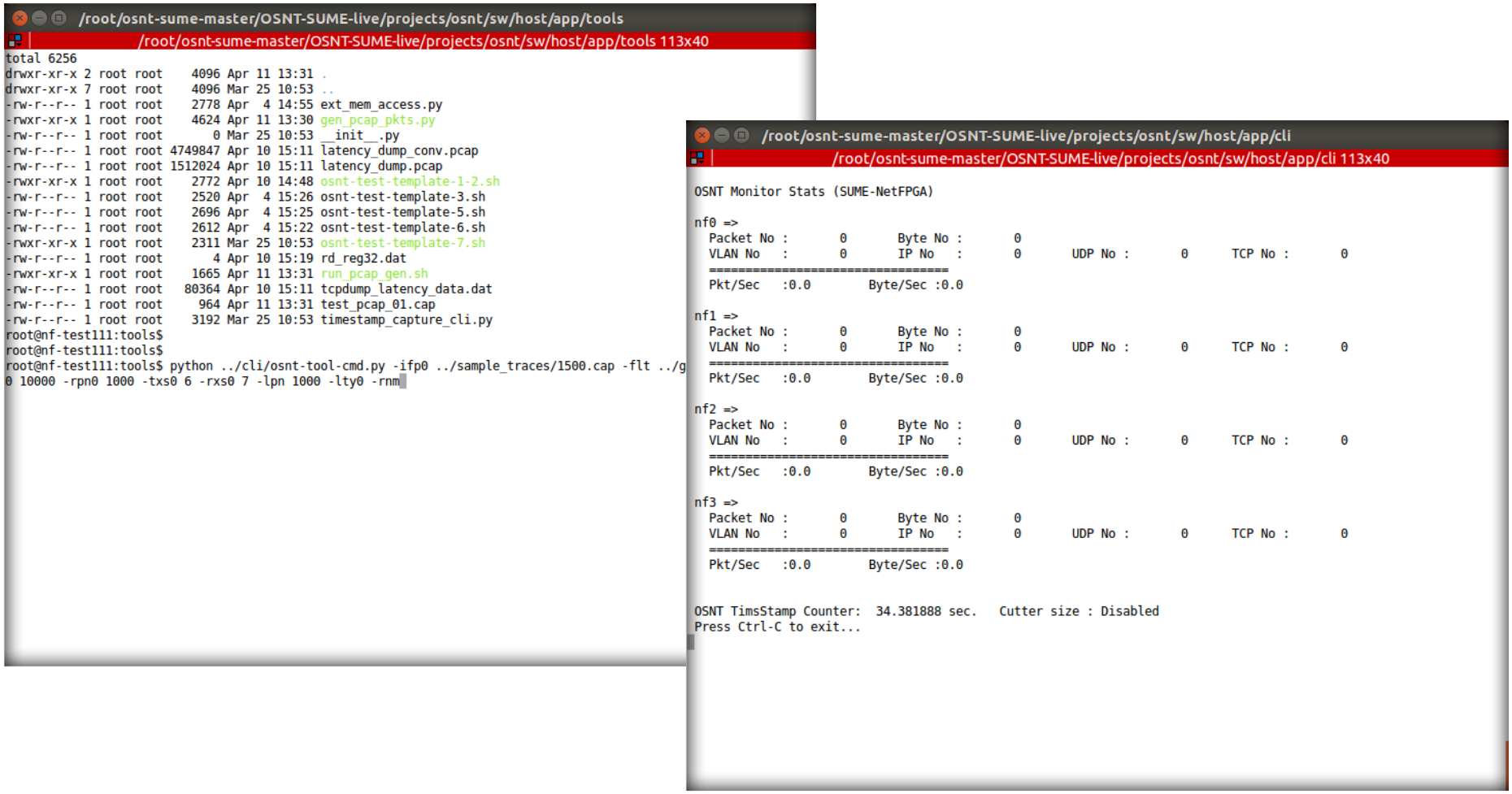
- OSNT GUI – Extensible Generator and Monitor GUI in Python.
- Command-Line-Interface is also available.





OSNT command line

- Command-Line-Interface is available to create a script automating the test process.

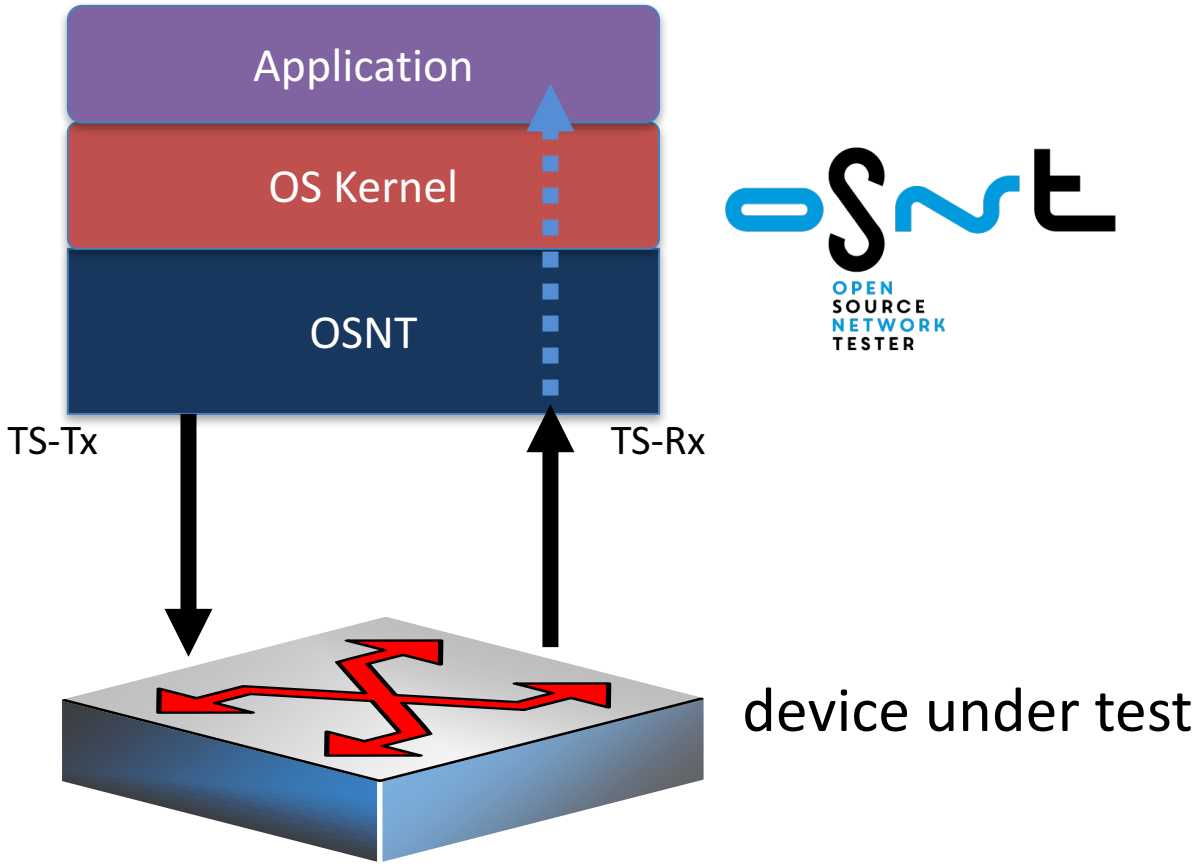


OSNT in action

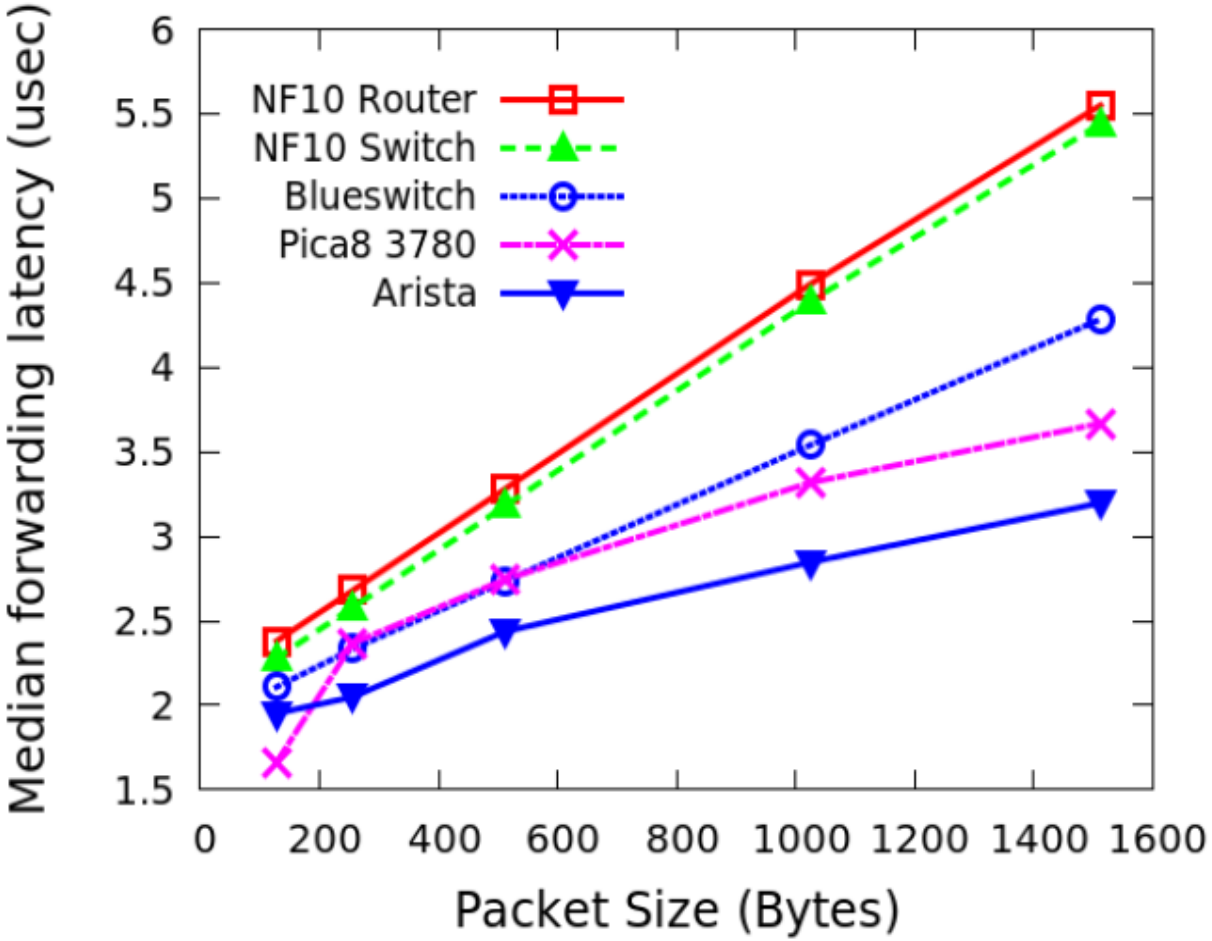
*Enabling **network innovation** with accurate **networking systems characterization***

Forwarding latency measurement

- Unloaded switches baseline latency no cross traffic



Forwarding latency measurement

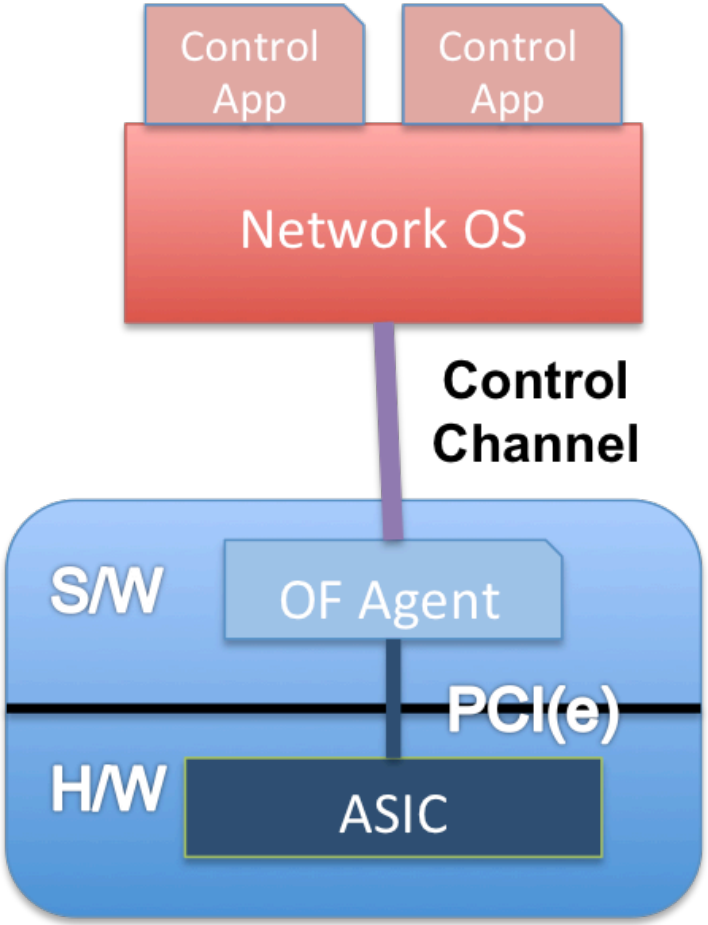


Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

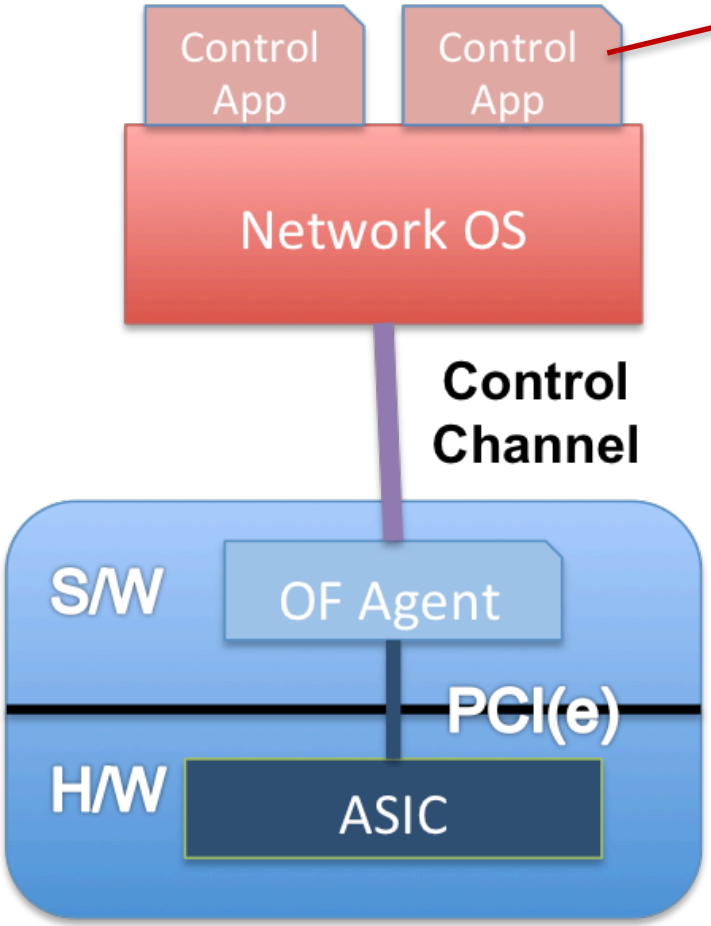
SDN Testing Suite

- SDN enables unprecedented flexible and extensible network control
- OpenFlow specifications lack performance semantics
  - What does a barrier reply signifies?
- OpenFlow performance aspects are yet to be explored
  - How do you compare two OpenFlow switches?
- OpenFlow flexibility is not always portable on switch ASIC

OpenFlow toolstack X-Ray

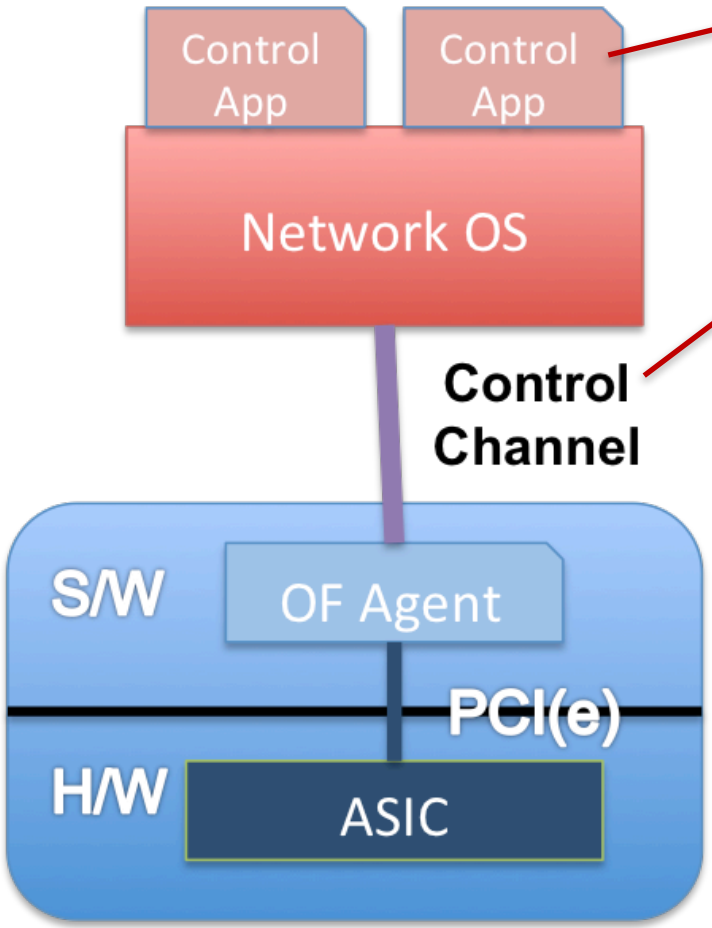


OpenFlow toolstack X-Ray



Control Application Complexity

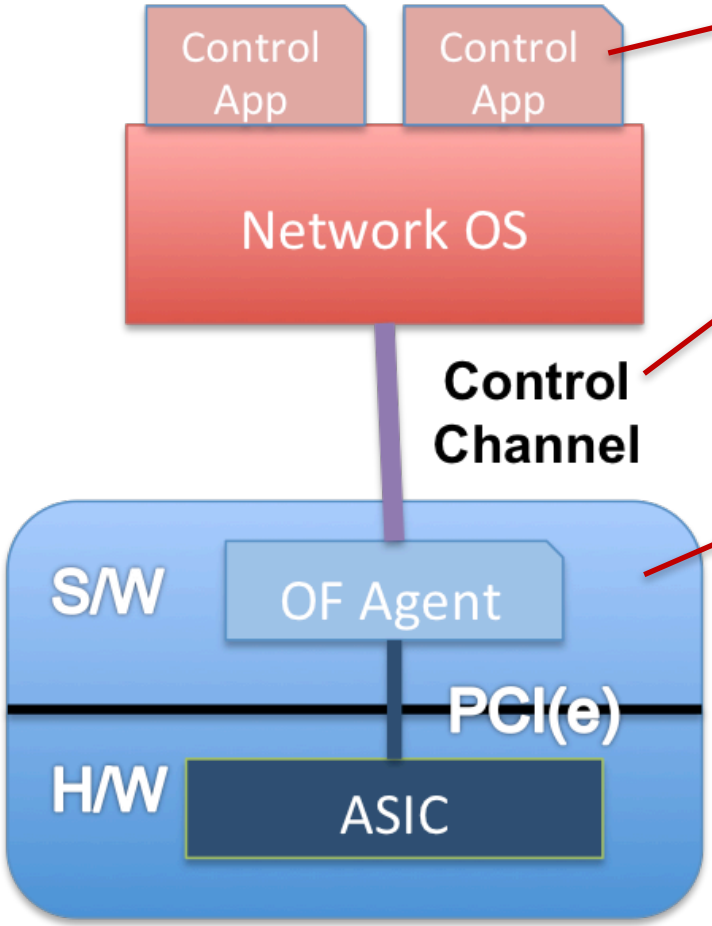
OpenFlow toolstack X-Ray



Control Application Complexity

Control Channel Capacity

OpenFlow toolstack X-Ray



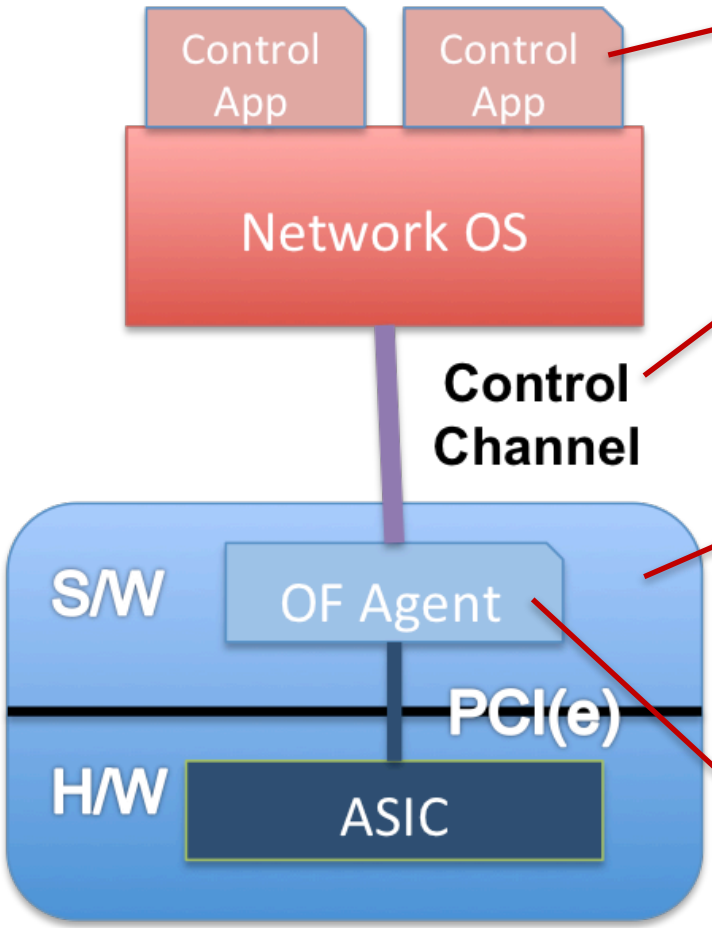
Control Application Complexity

Control Channel Capacity

Scarce co-processor resources

Switch OS scheduling is non trivial

OpenFlow toolstack X-Ray



Control Application Complexity

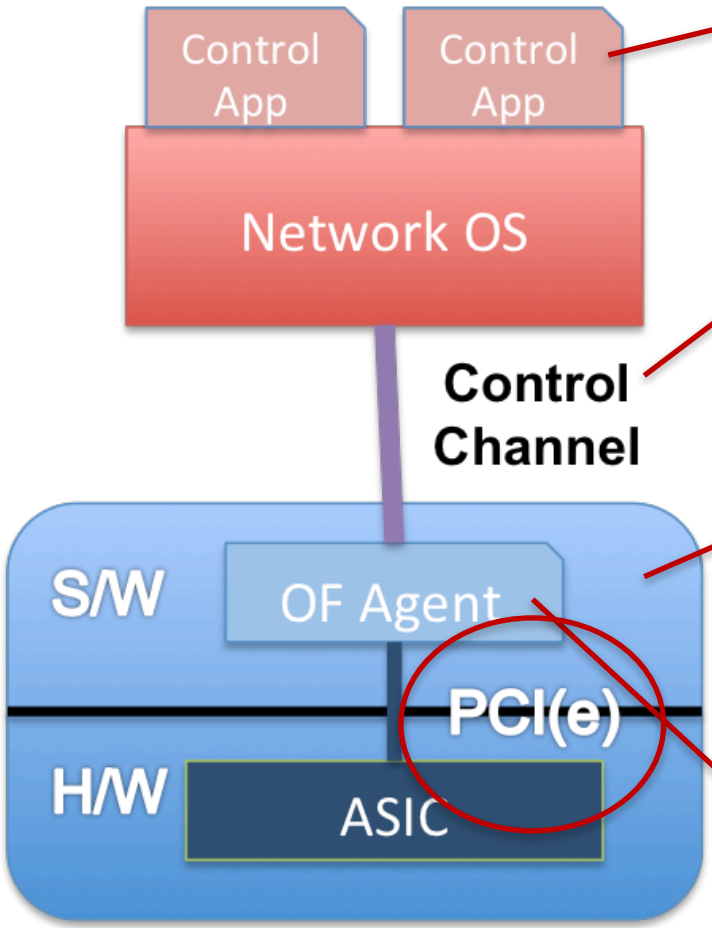
Control Channel Capacity

Scarce co-processor resources

Switch OS scheduling is non trivial

ASIC driver -> policy configuration

OpenFlow toolstack X-Ray



Control Application Complexity

Control Channel Capacity

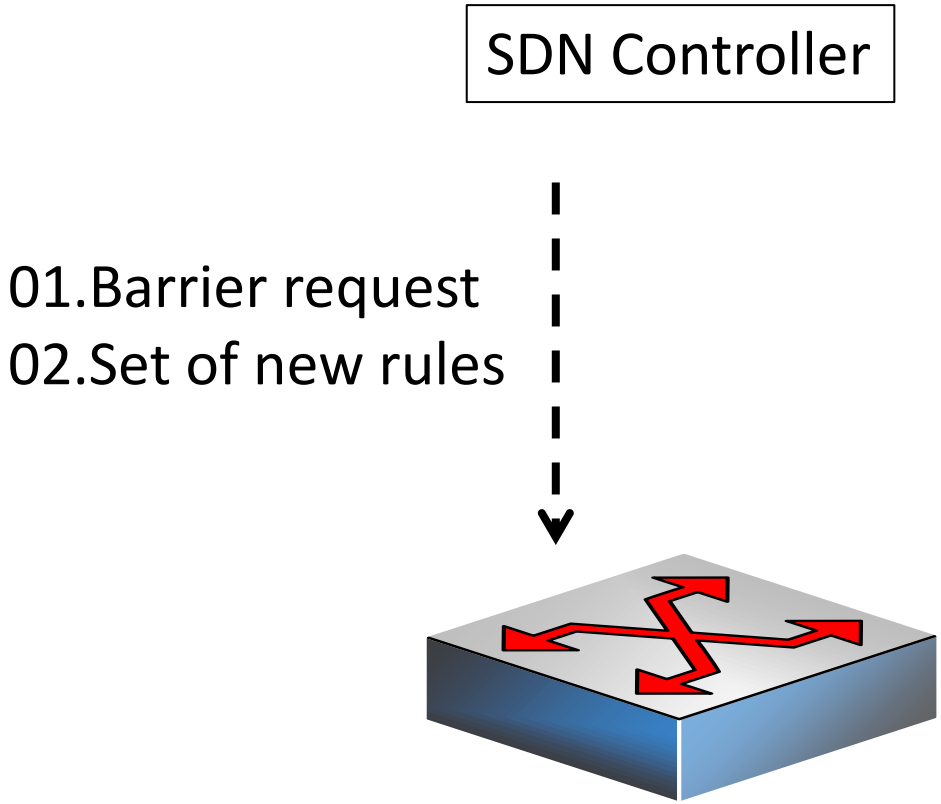
Scarce co-processor resources

Switch OS scheduling is non trivial

ASIC driver -> policy configuration

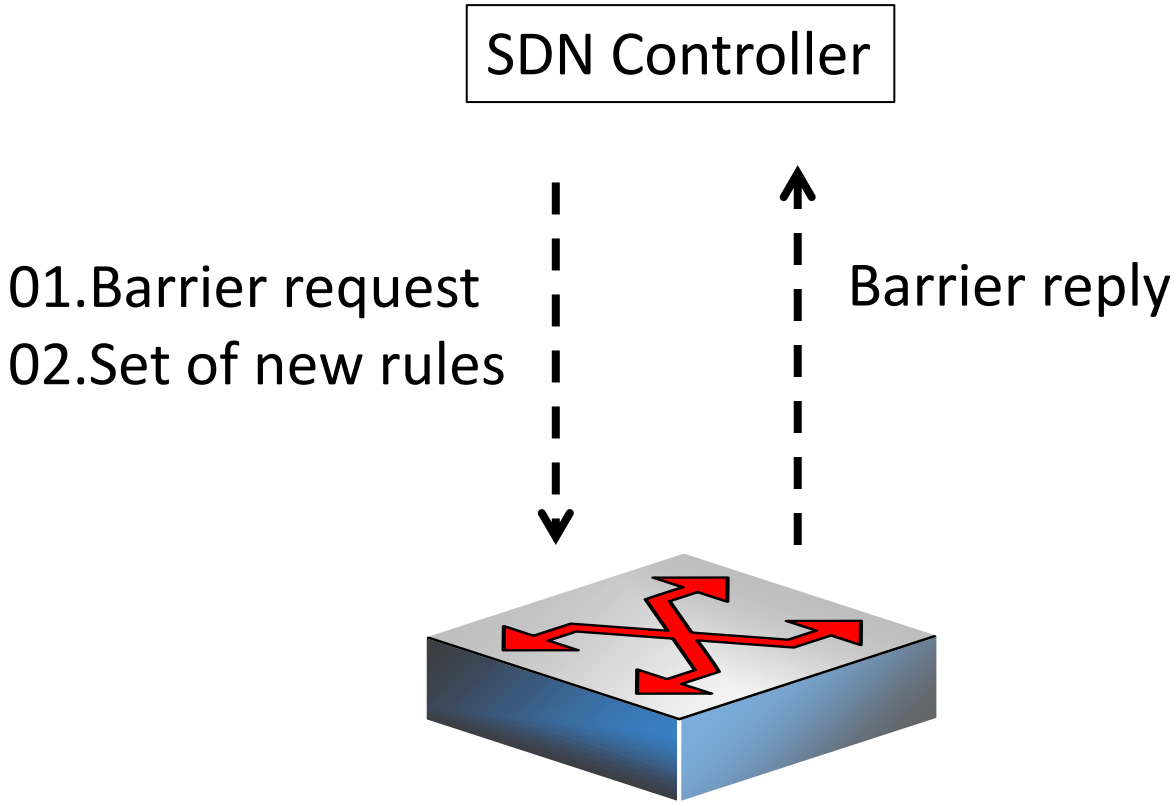
SDN networks performances

THE PROACTIVE CASE



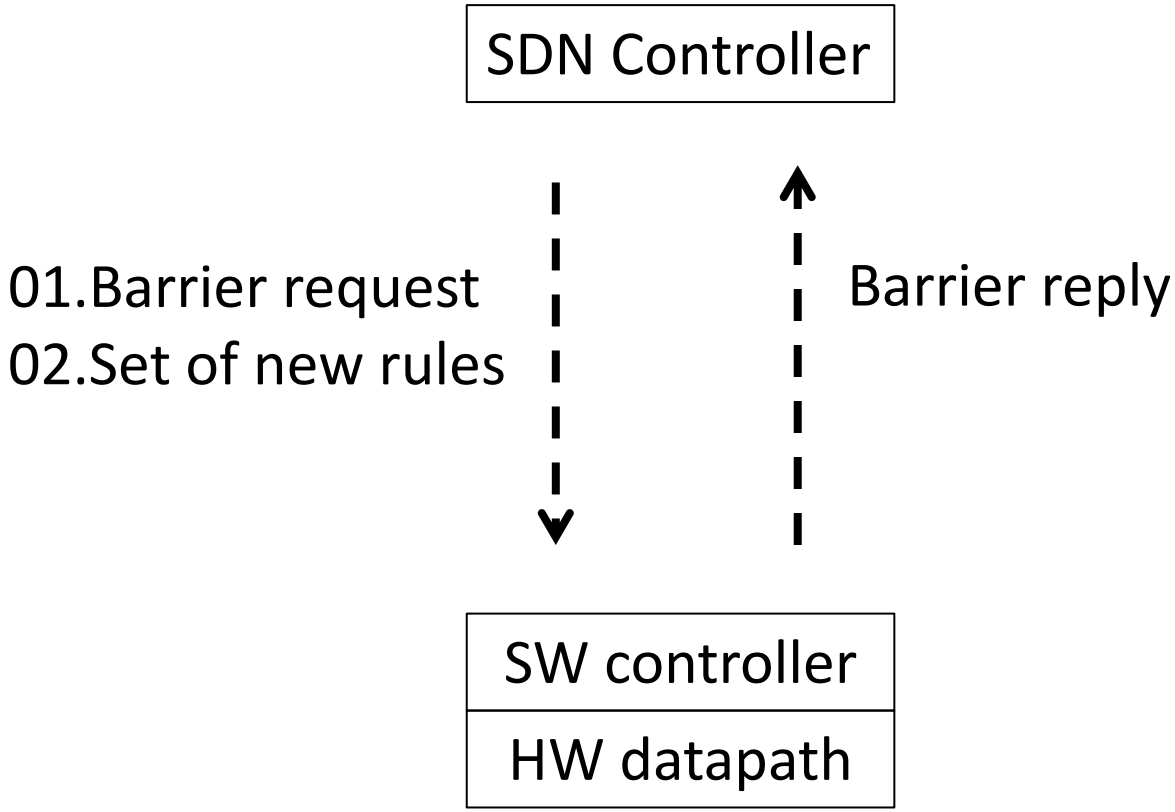
SDN networks performances

THE PROACTIVE CASE



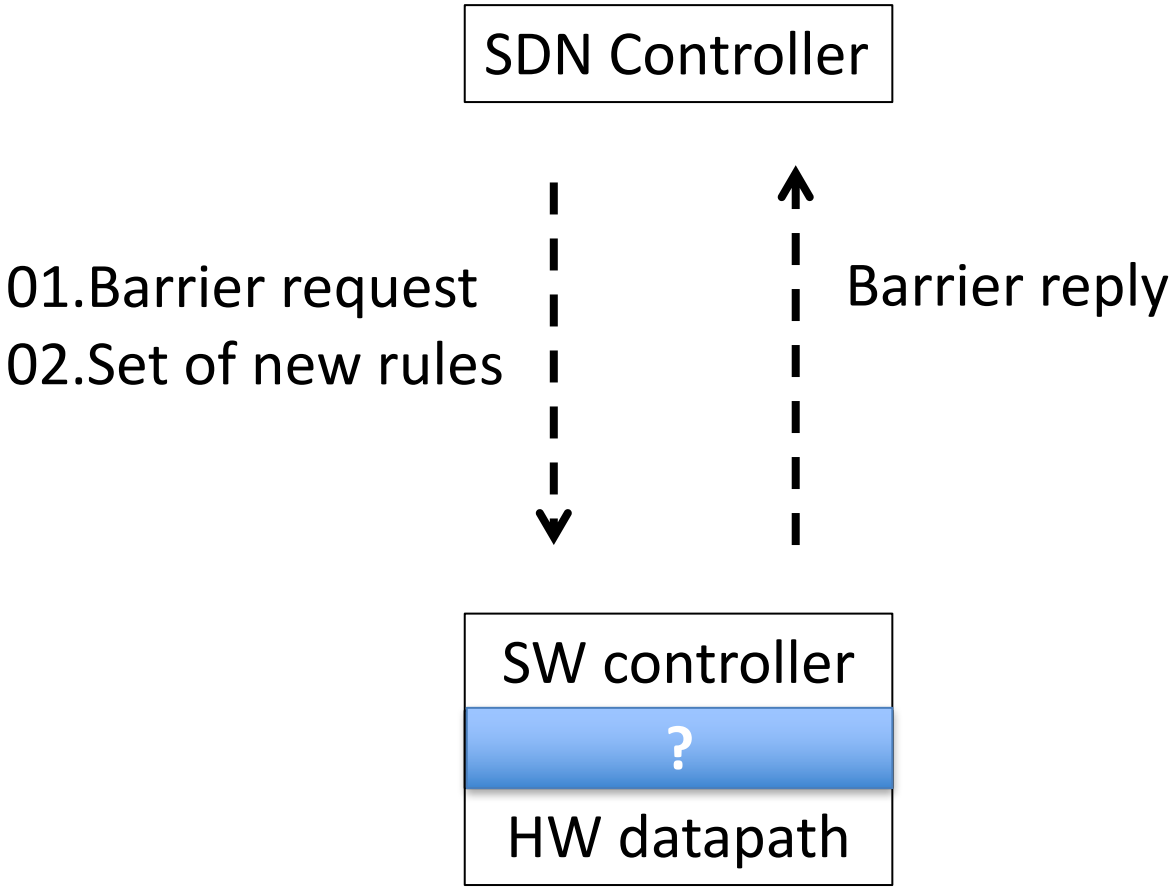
SDN networks performances

THE PROACTIVE CASE



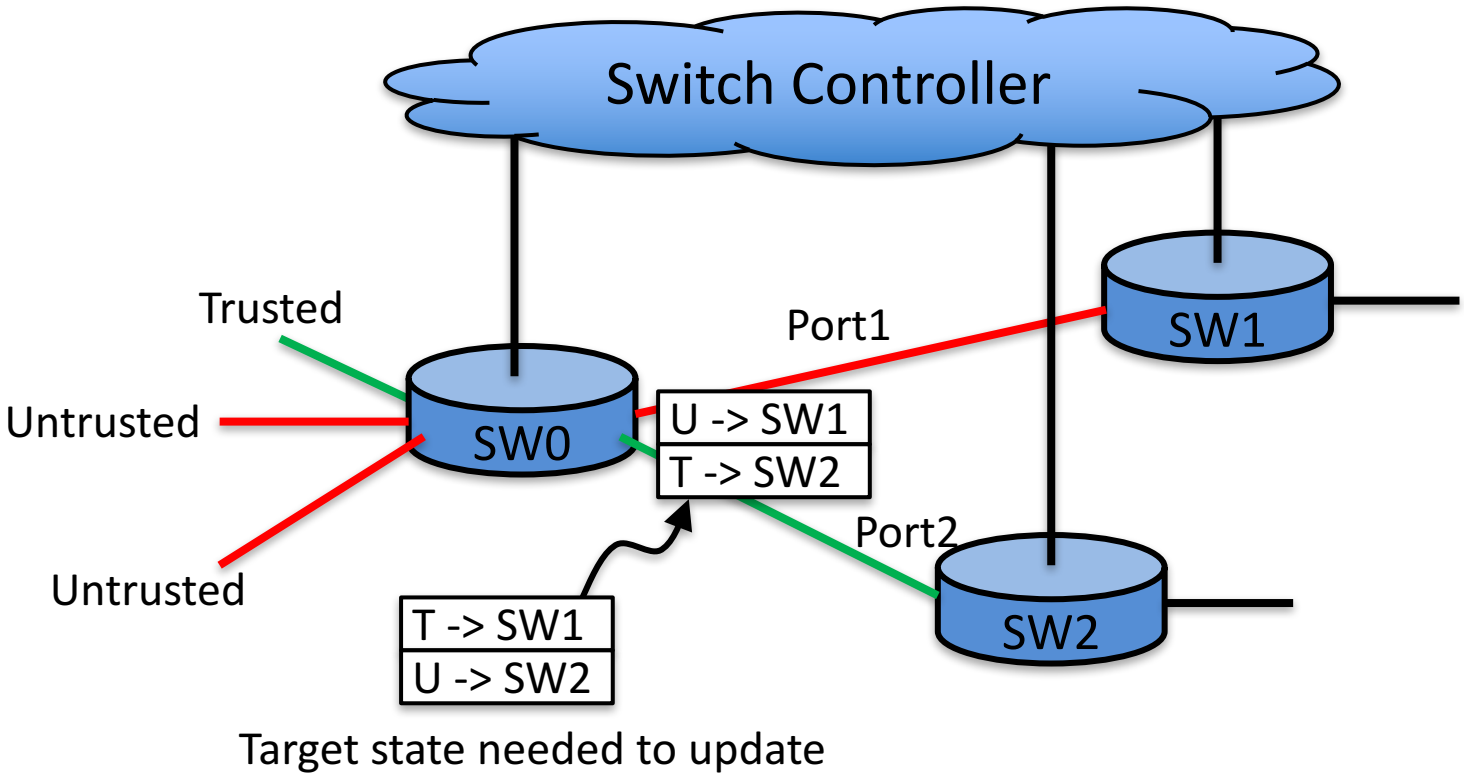
SDN networks performances

THE PROACTIVE CASE

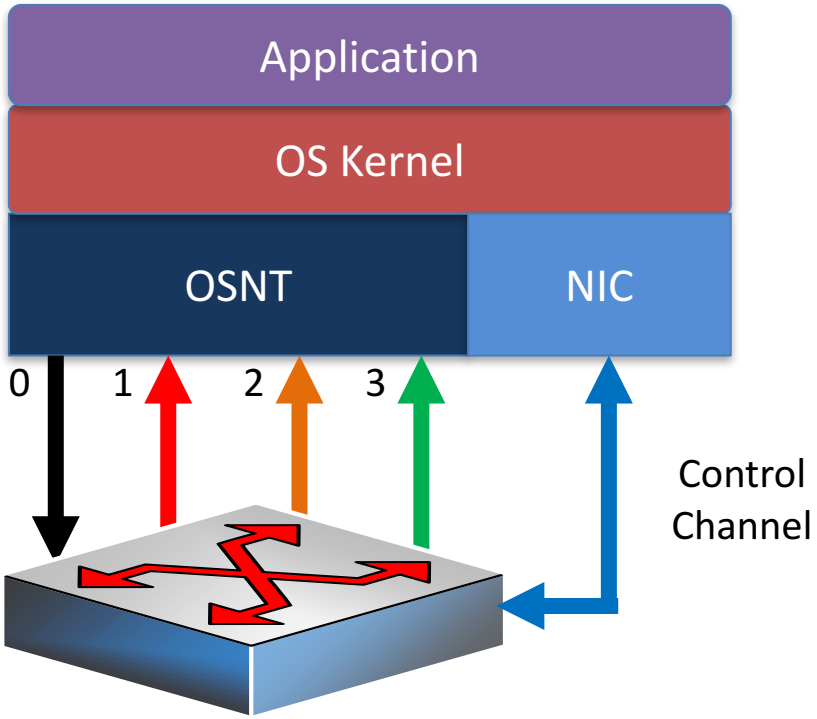


Control/Data plane consistency

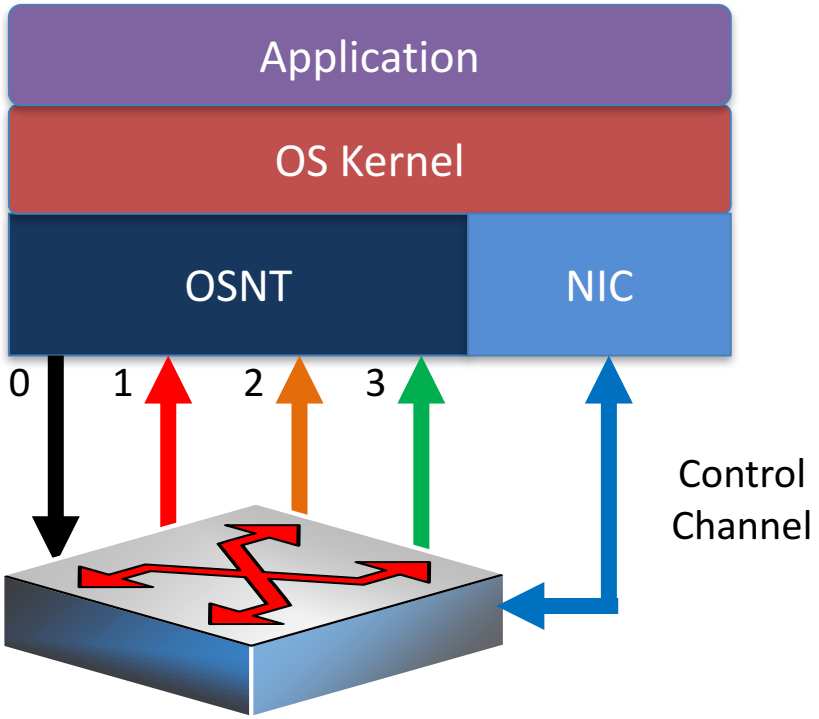
Consistent policy update affects security in SDN.



Control/Data plane consistency



Control/Data plane consistency

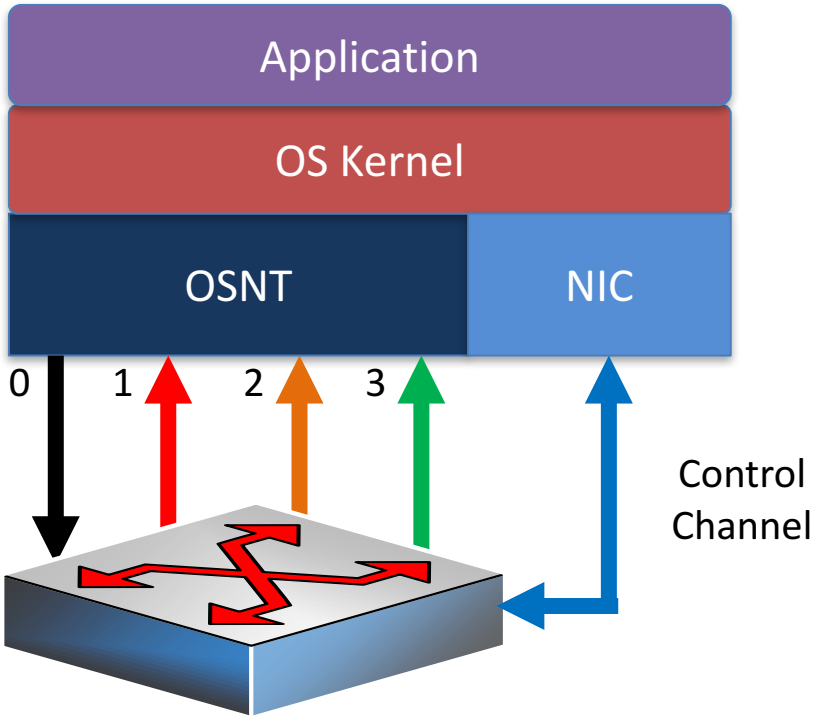


1. Initial rule : 0 → 1 (as a set of different IP rules)

2. Rule update : 0 → 2

Control/Data plane consistency

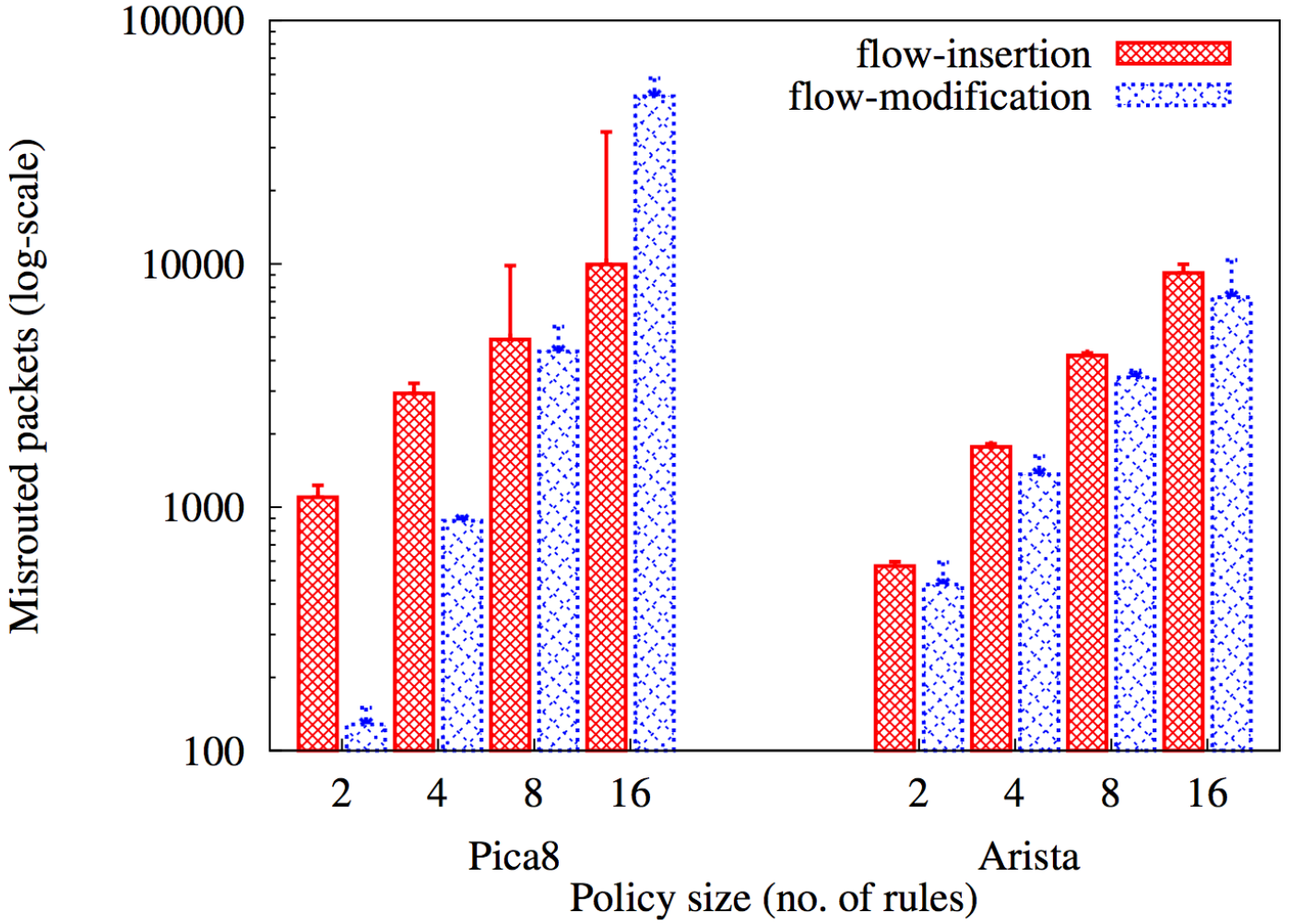
We generate an aggregate 2Gbps with 150B packets



1. Initial rule : $0 \rightarrow 1$ (as a set of different IP rules)

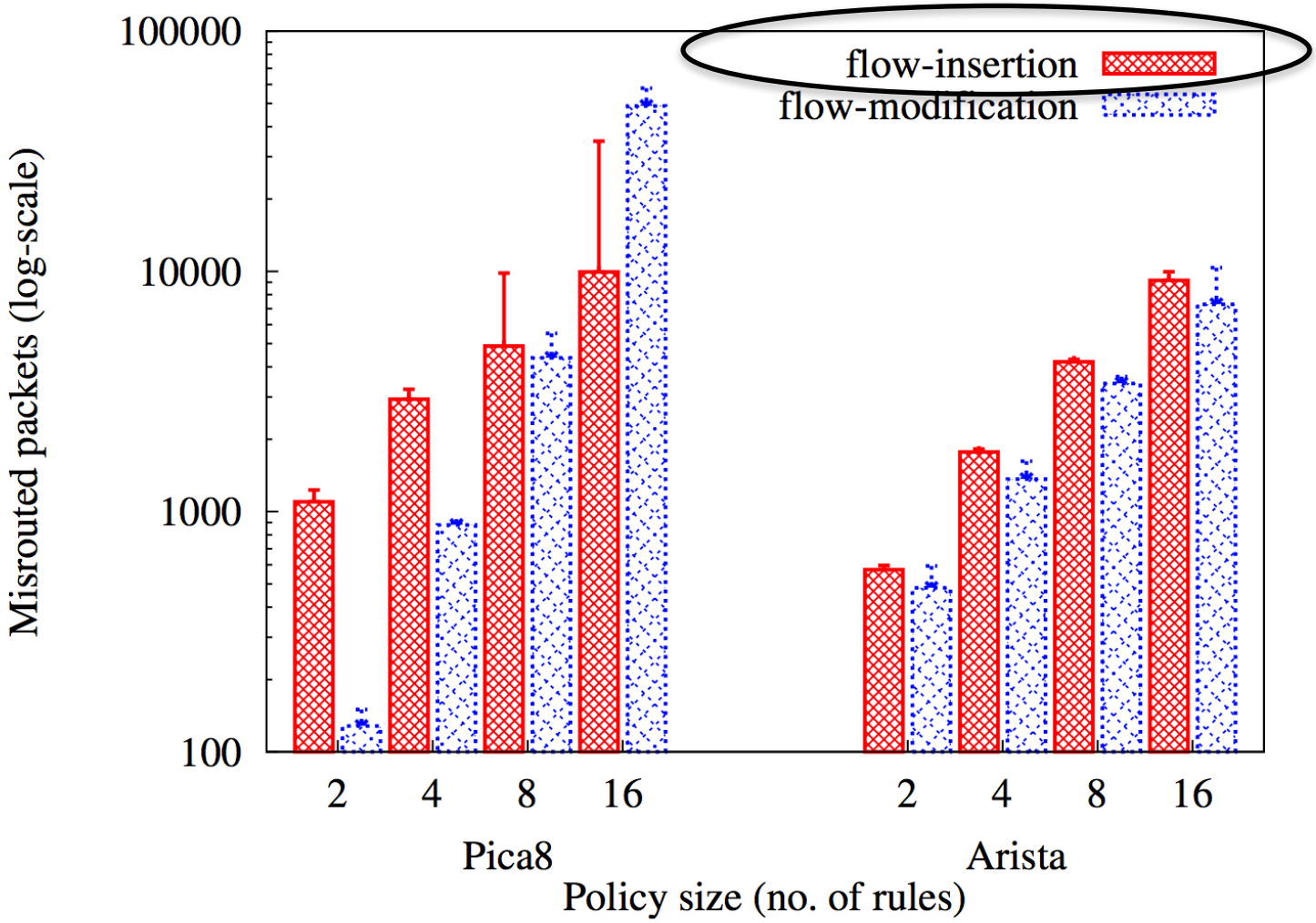
2. Rule update : $0 \rightarrow 2$

Control/Data plane consistency



Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

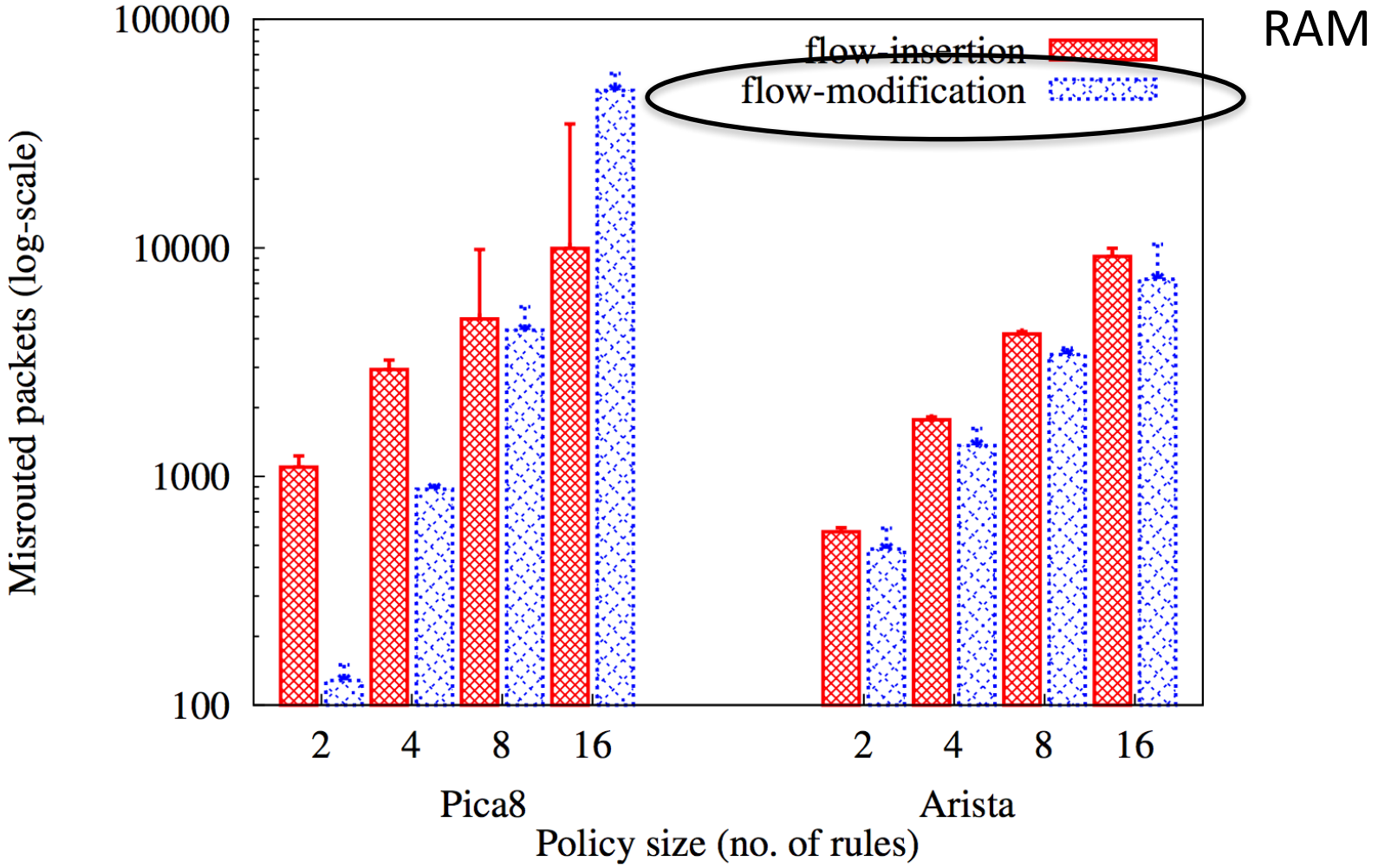
Control/Data plane consistency



TCAM + RAM

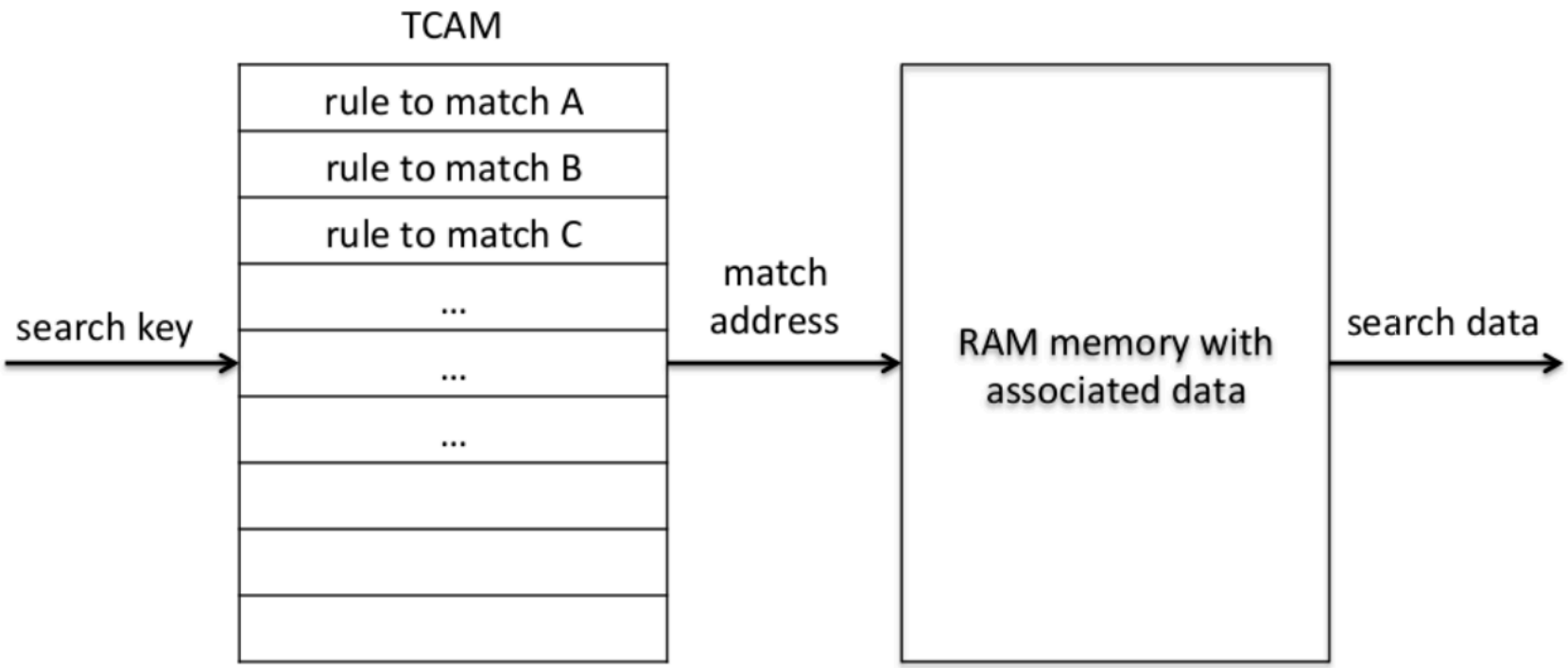
Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

Control/Data plane consistency



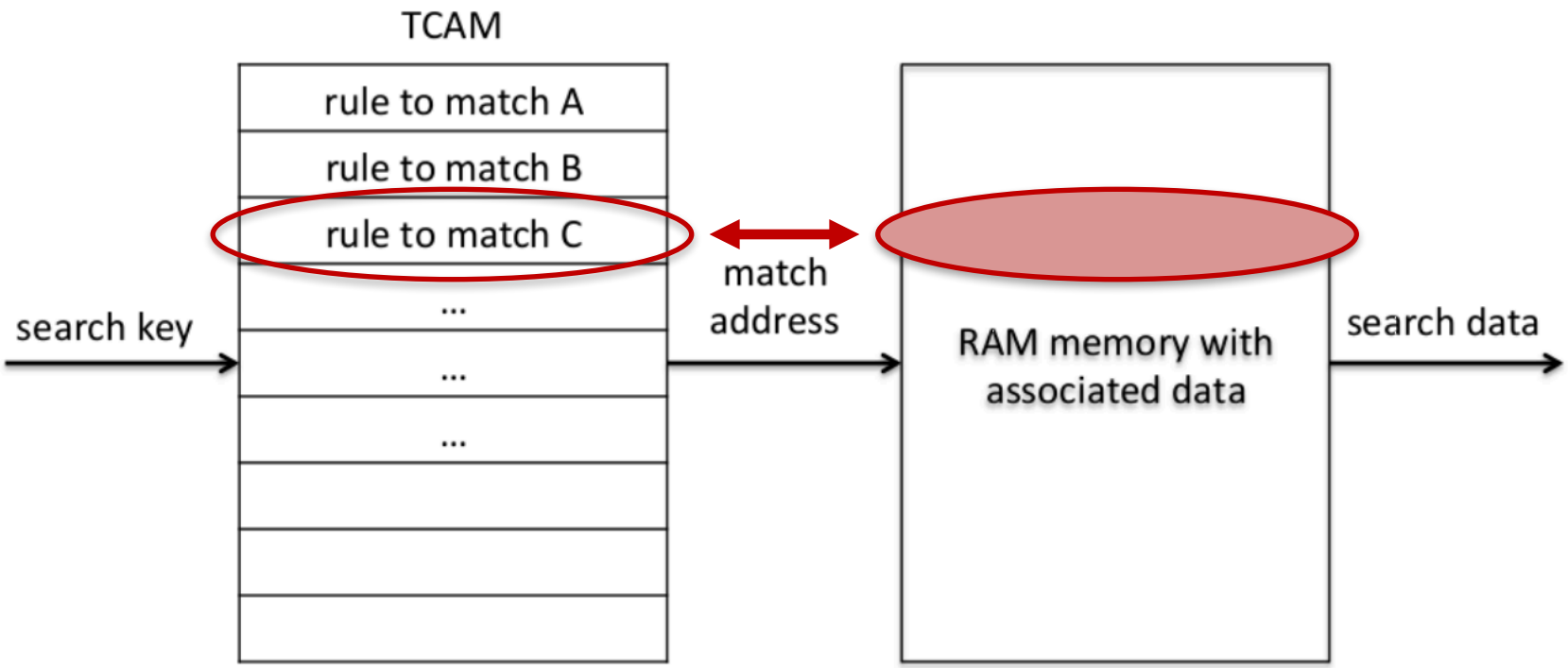
Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

Control/Data plane consistency



Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

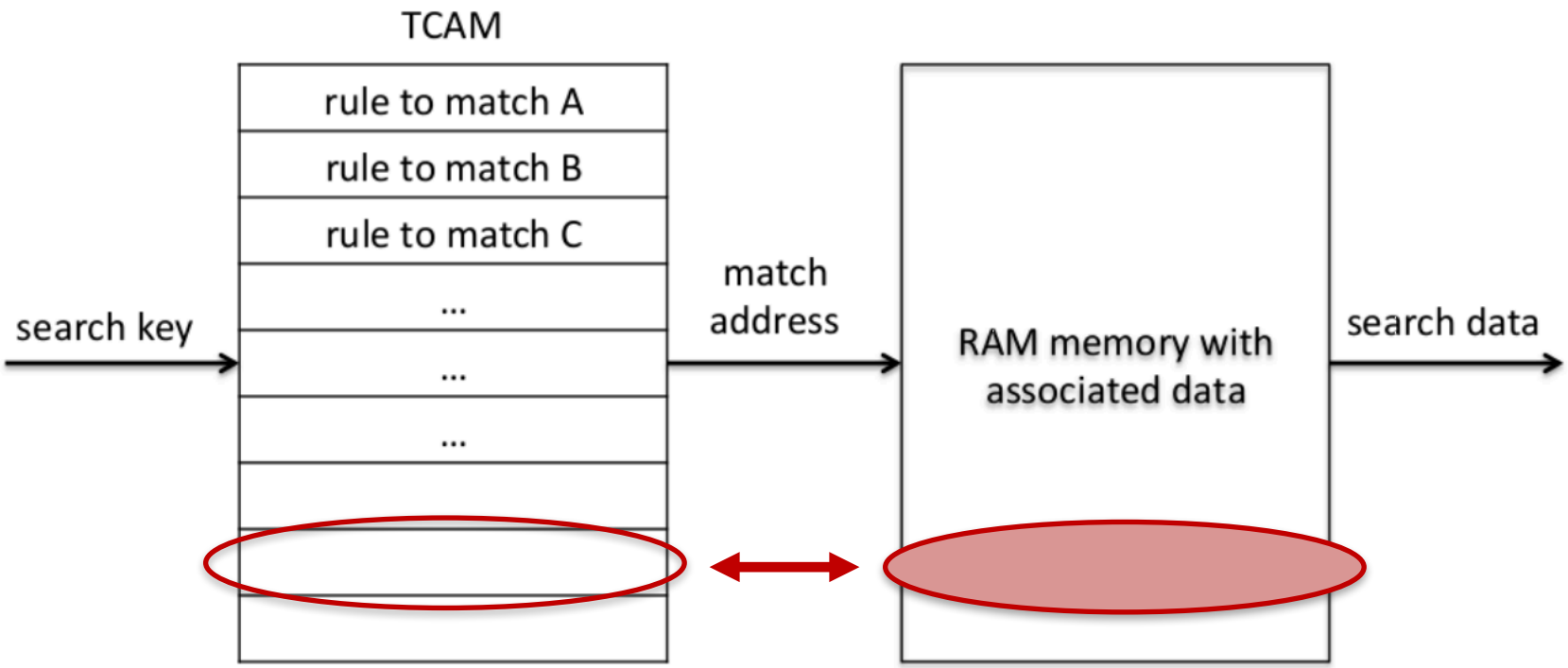
Control/Data plane consistency



A flow modification requires only to change the “action” in the RAM. The flow is already present in the TCAM.

Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

Control/Data plane consistency

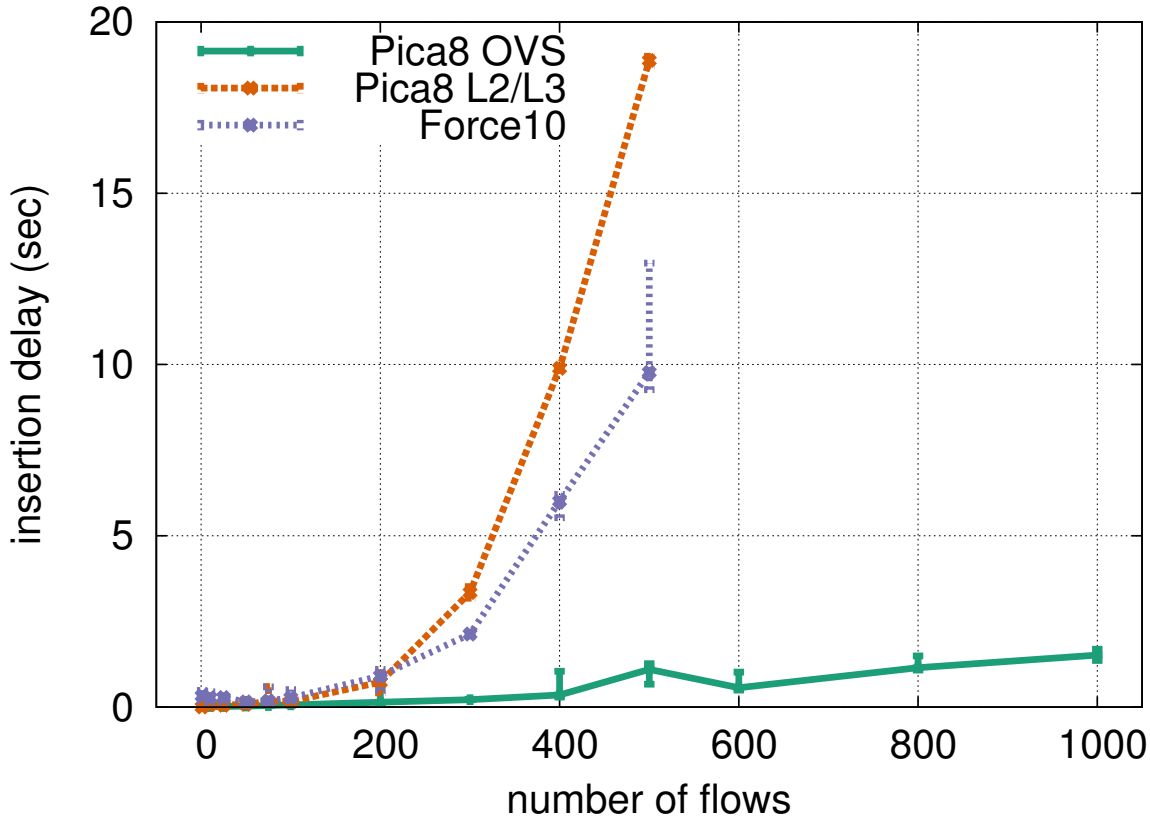


A flow insertion requires also to write the matching fields in the TCAM.

Blueswitch: Enabling provably consistent configuration of network switches, Han J.H., Mundkur P., Rotsos C., Antichi G., Dave N., Moore A.W., Neumann P.G., ACM/IEEE ANCS, Oakland, CA, USA, 7-8 May, 2015

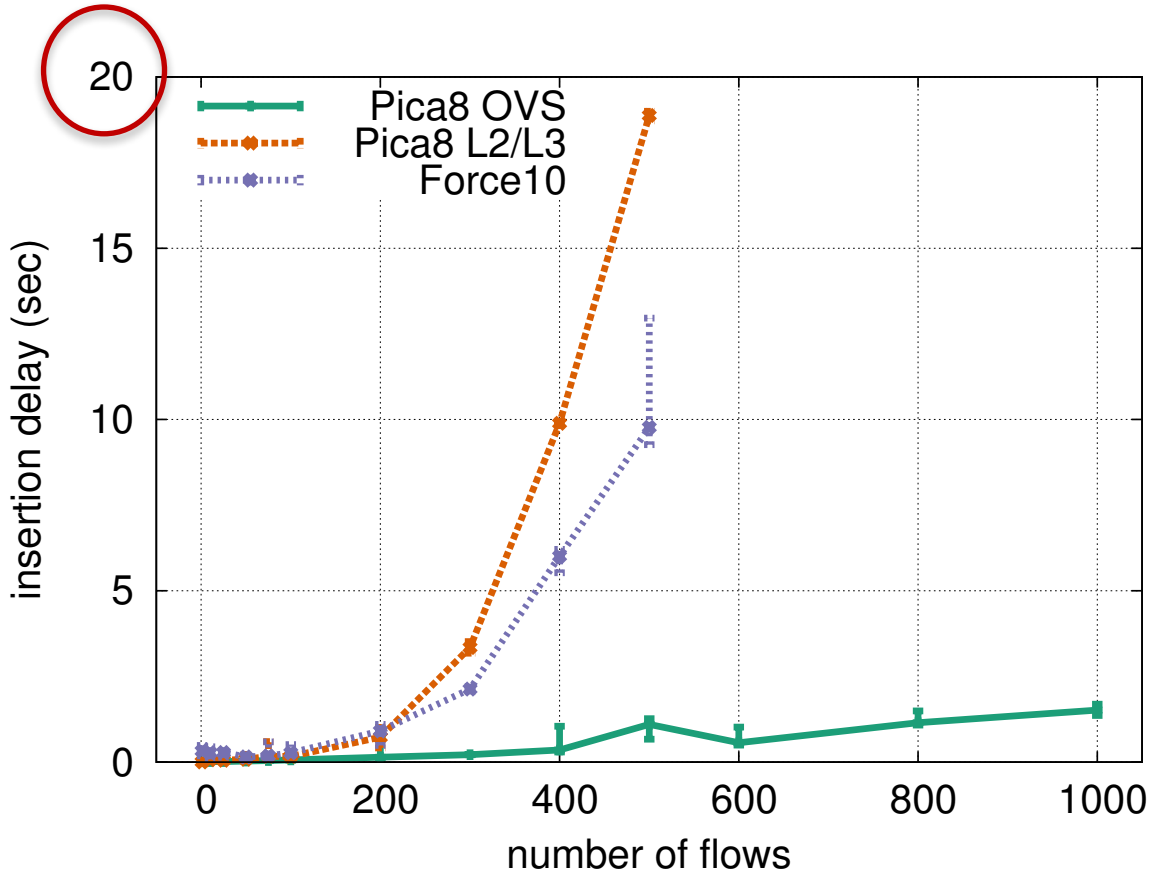
Dataplane performances

Flow addition delay



OFLOPS-Turbo: Testing the Next-Generation OpenFlow Switch, Rostos C., Antichi G., Bruyere M., Owezarski P., Moore A.W., IEEE ICC, London, UK, 8-12 June, 2015

Dataplane performances

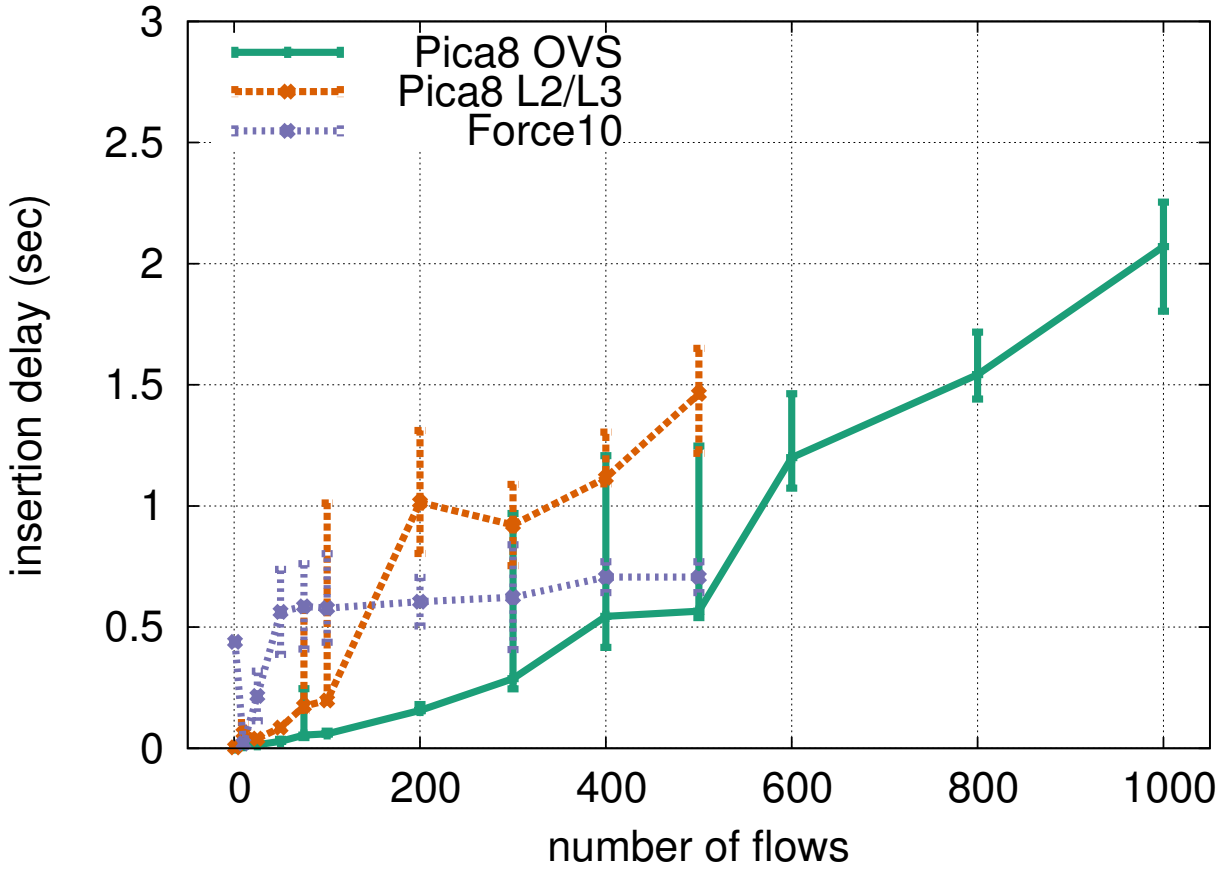


Flow addition delay

OFLOPS-Turbo: Testing the Next-Generation OpenFlow Switch, Rostos C., Antichi G., Bruyere M., Owezarski P., Moore A.W., IEEE ICC, London, UK, 8-12 June, 2015

Dataplane performances

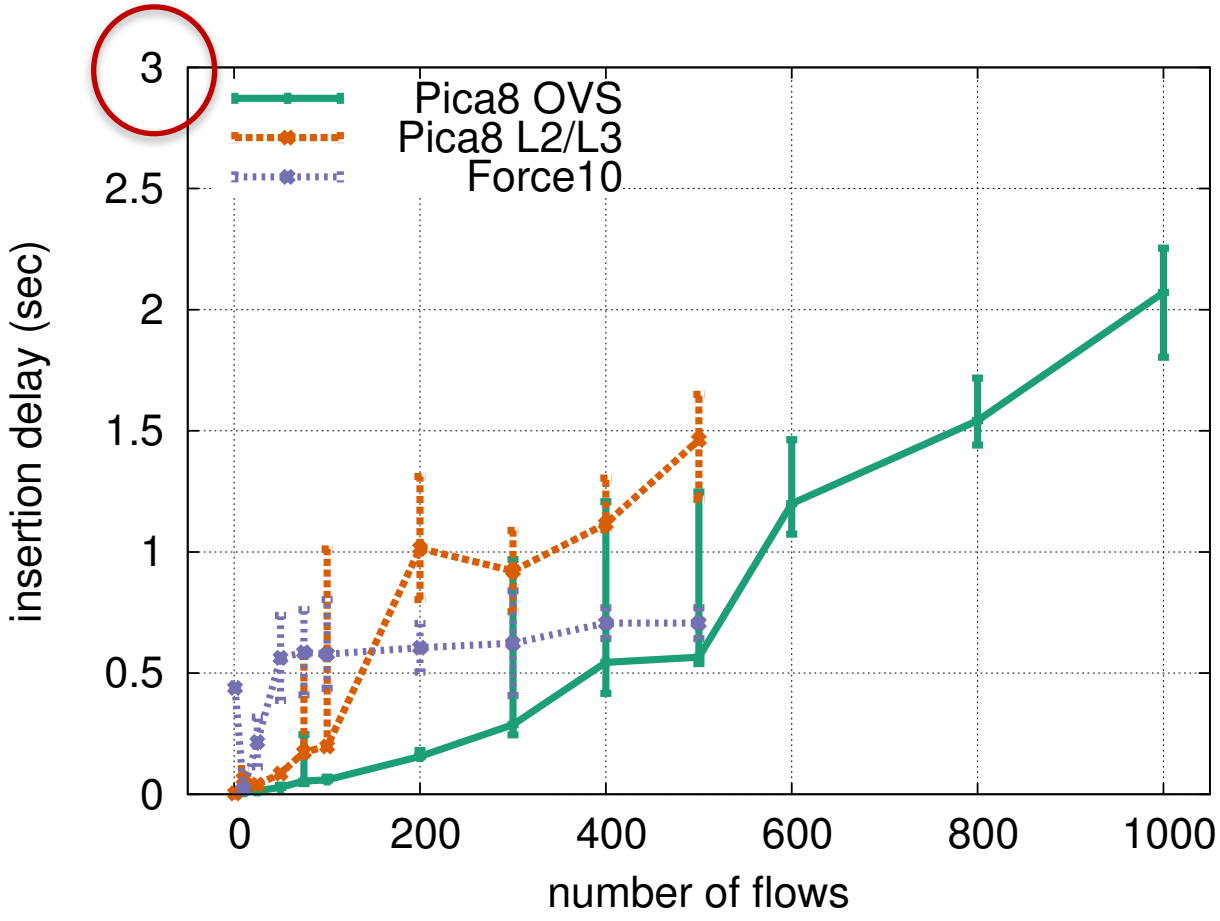
Flow modification delay



OFLOPS-Turbo: Testing the Next-Generation OpenFlow Switch, Rostos C., Antichi G., Bruyere M., Owezarski P., Moore A.W., IEEE ICC, London, UK, 8-12 June, 2015

Dataplane performances

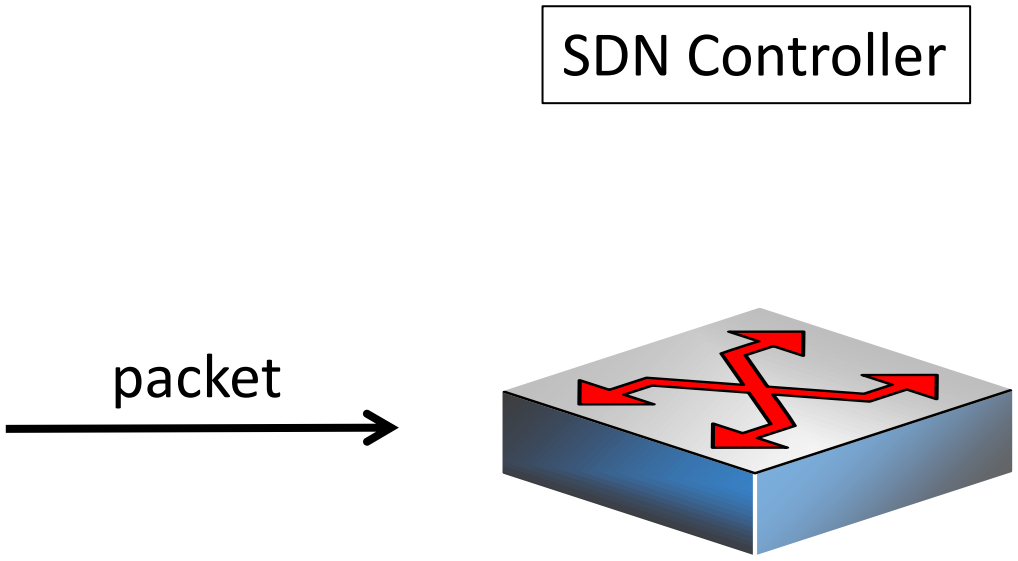
Flow modification delay



OFLOPS-Turbo: Testing the Next-Generation OpenFlow Switch, Rostos C., Antichi G., Bruyere M., Owezarski P., Moore A.W., IEEE ICC, London, UK, 8-12 June, 2015

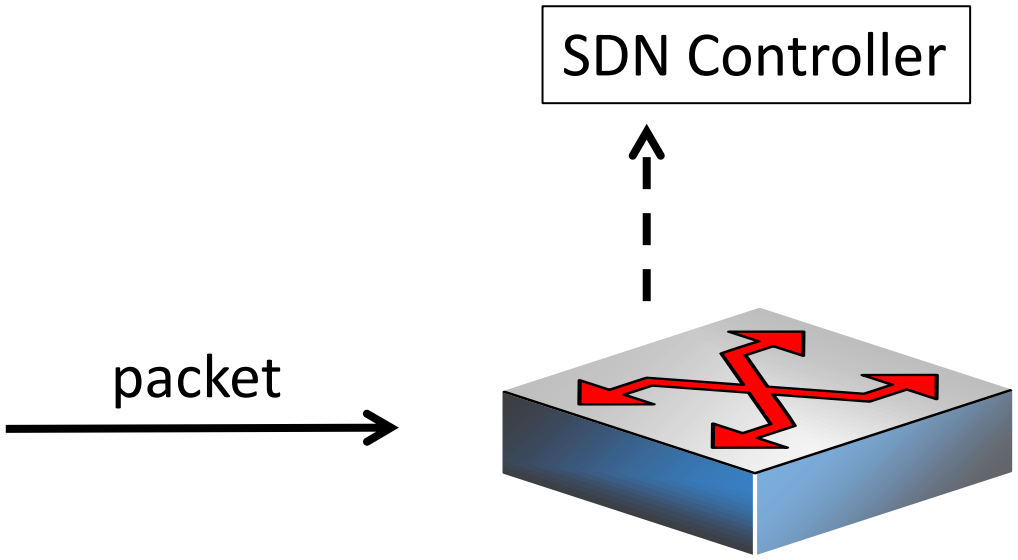
SDN networks performances

THE REACTIVE CASE



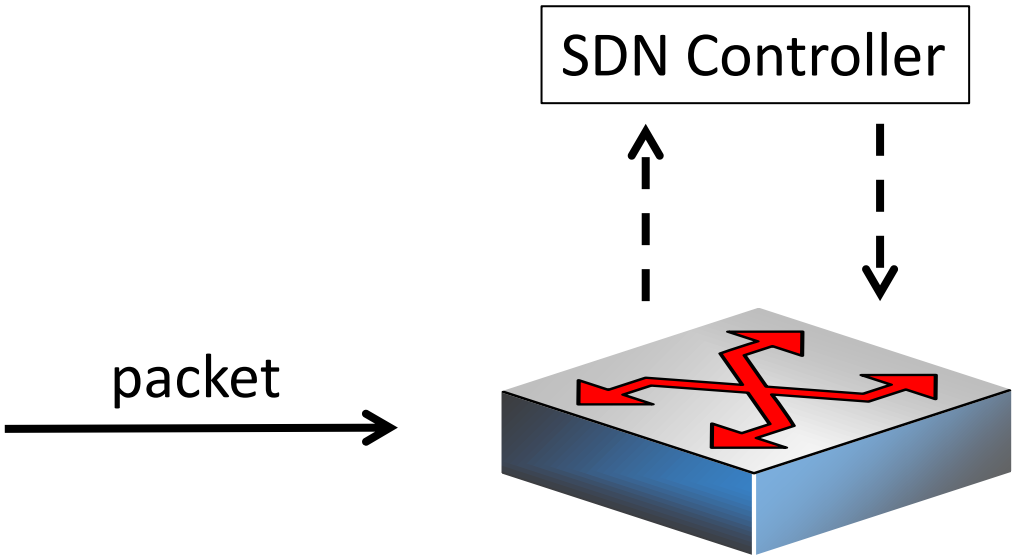
SDN networks performances

THE REACTIVE CASE



SDN networks performances

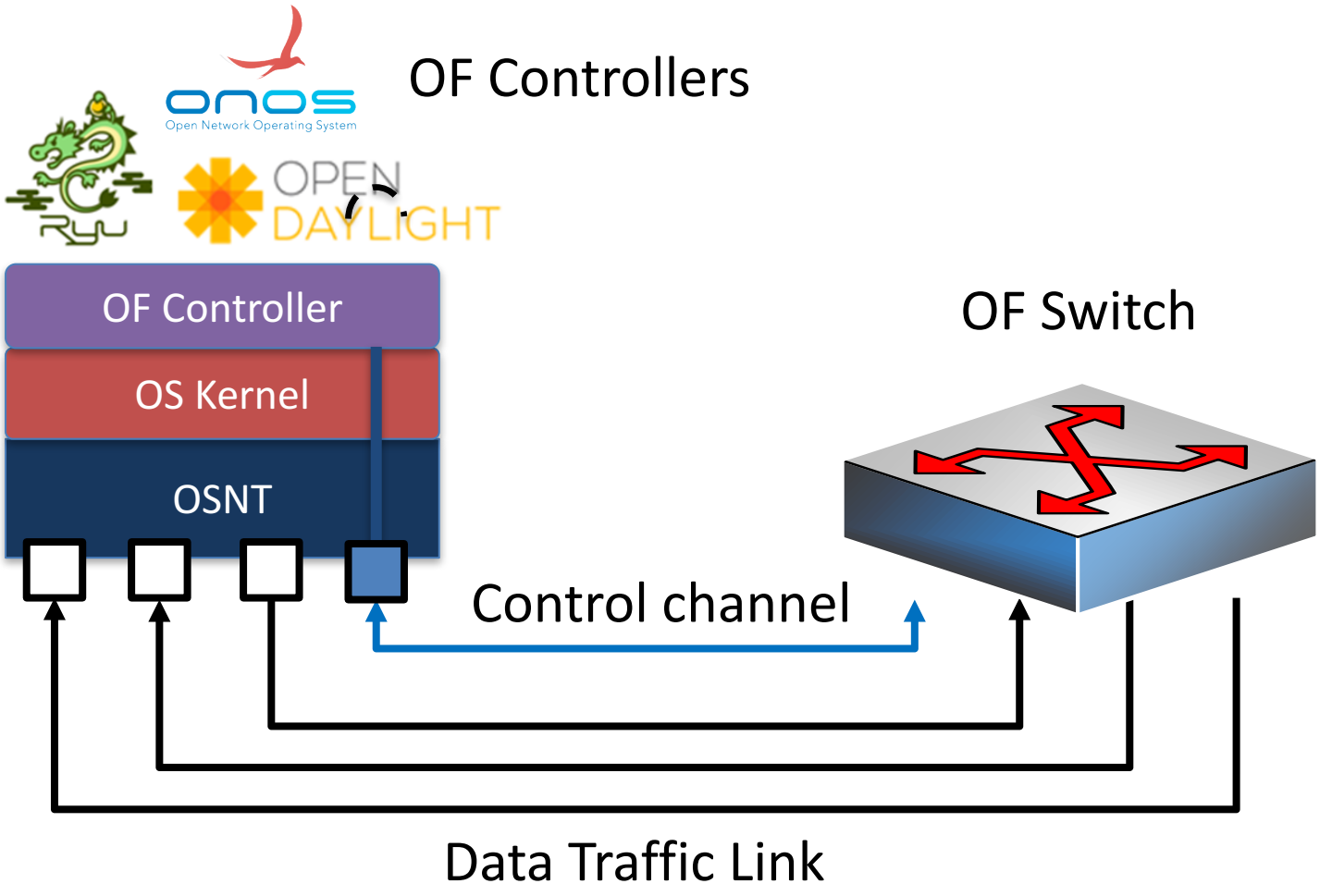
THE REACTIVE CASE



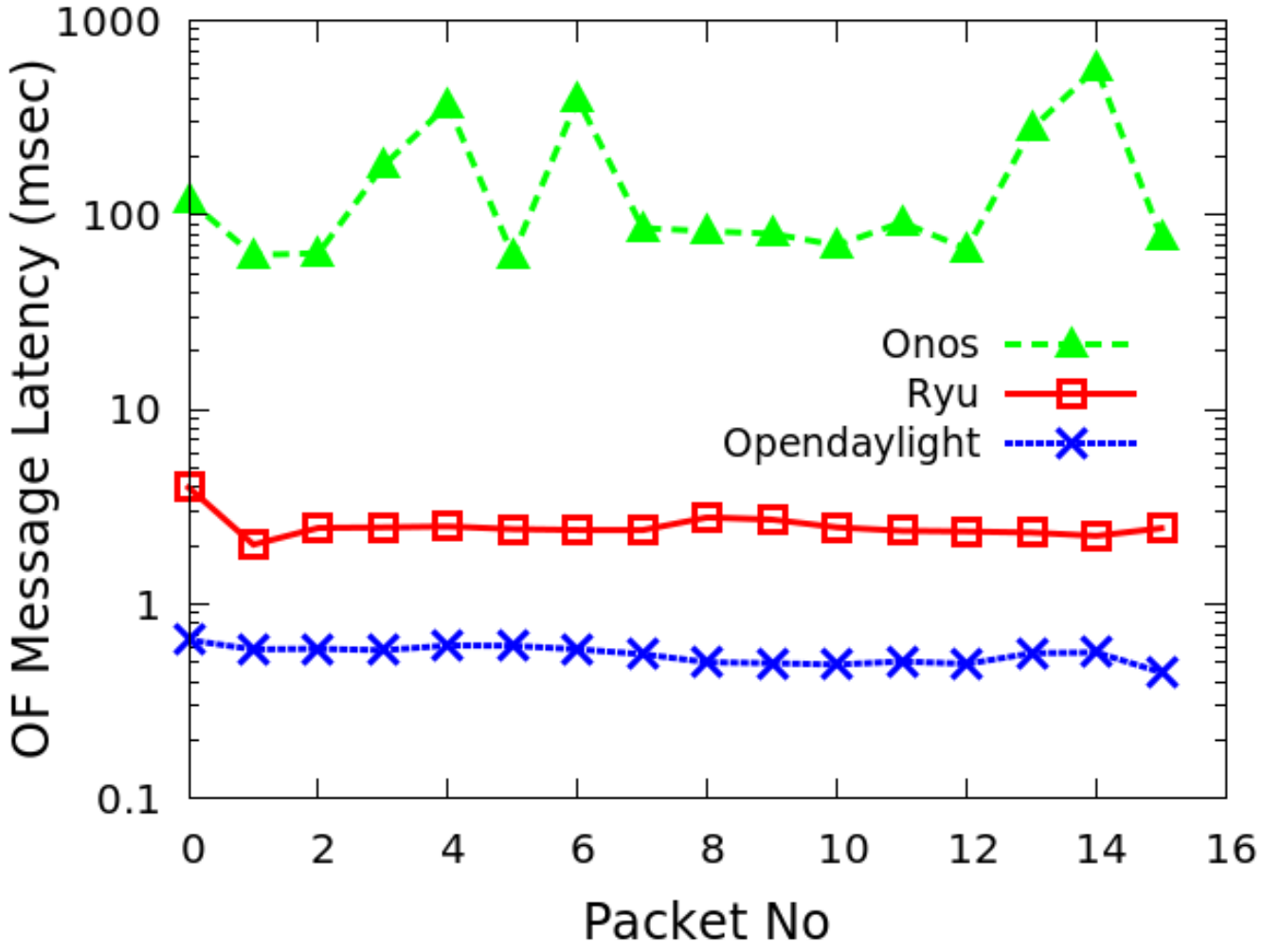
SDN networks performances

THE REACTIVE CASE

Controller performances in SDN networks



Controller performances in SDN networks



OSNT-SUME-live Github

- OSNT-SUME-live is publicly available.

NetFPGA / OSNT-SUME-live Private

Unwatch 20 | Star 0 | Fork 2

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

OSNT for NetFPGA-SUME board

Edit

New Add topics

47 commits | 1 branch | 4 releases | 3 contributors

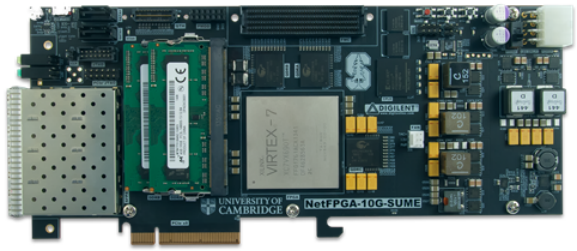
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

jhhan Merge branch 'master' of https://github.com/NetFPGA/OSNT-SUME-live — Latest commit 1a058a2 7 days ago

contrib/challenge2017	NetFPGA design challenge 2017, added the first test	12 days ago
lib	Update the extmem packet replay controller and add the qdr i/f contro...	7 days ago
projects	Update minor in 10g rx and tx cores. Add comments to the timestamp mo...	a month ago
scripts	Add the first clean repo.	8 months ago
util	update pcap_gen to support configurable source and dest MAC addresses	12 days ago
Makefile	Add Makefile for ip core and sw driver generation.	
README.md	Update the release not link.	

README.md

OSNT



Acknowledgments (I)

NetFPGA Team at University of Cambridge (Past and Present):

Andrew Moore, David Miller, Muhammad Shahbaz, Martin Zadnik, Matthew Grosvenor, Yury Audzevich, Neelakandan Manihatty-Bojan, Georgina Kalogeridou, Jong Hun Han, Noa Zilberman, Gianni Antichi, Charalampos Rotsos, Hwanju Kim, Marco Forconesi, Jinyun Zhang, Bjoern Zeeb, Robert Watson, Salvator Galea, Marcin Wojcik, Diana Andreea Popescu, Murali Ramanujam

NetFPGA Team at Stanford University (Past and Present):

Nick McKeown, Glen Gibb, Jad Naous, David Erickson, G. Adam Covington, John W. Lockwood, Jianying Luo, Brandon Heller, Paul Hartke, Neda Beheshti, Sara Bolouki, James Zeng, Jonathan Ellithorpe, Sachidanandan Sambandan, Eric Lo, Stephen Gabriel Ibanez

All Community members (including but not limited to):

Paul Rodman, Kumar Sanghvi, Wojciech A. Koszek, Yahsar Ganjali, Martin Labrecque, Jeff Shafer, Eric Keller, Tatsuya Yabe, Bilal Anwer, Yashar Ganjali, Martin Labrecque, Lisa Donatini, Sergio Lopez-Buedo , Andreas Fiessler, Robert Soule, Pietro Bressana, Yuta Tokusashi

Steve Wang, Erik Cengar, Michael Alexander, Sam Bobrowicz, Garrett Aufdemberg, Patrick Kane, Tom Weldon

Patrick Lysaght, Kees Vissers, Michaela Blott, Shep Siegel, Cathal McCabe

Acknowledgements (II)