國立清華大學
NATIONAL TSING HUA UNIVERSITY

Accelerating OpenFlow SDN Switches with Per-Port Cache

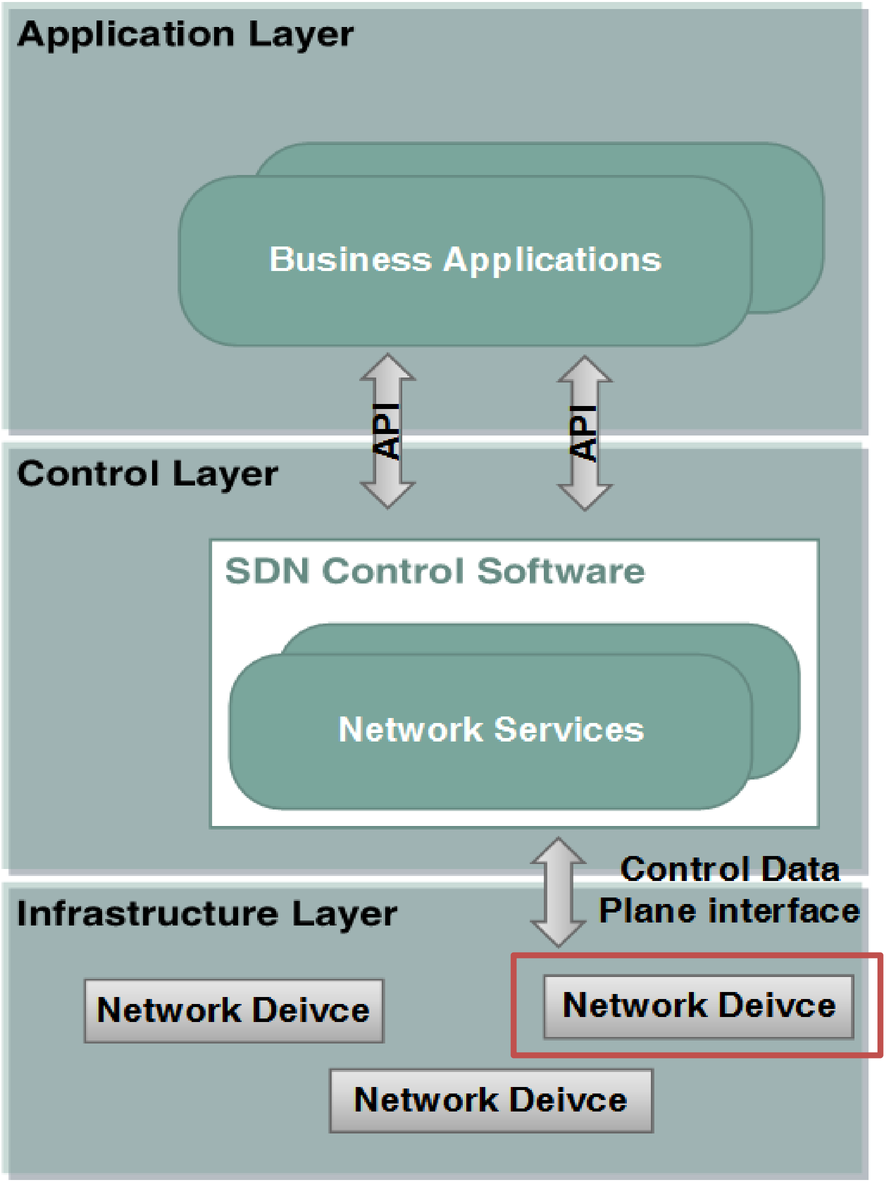
Cheng-Yi Lin Youn-Long Lin

Department of Computer Science
National Tsing Hua University

Outline

1. Introduction
2. Related Work
3. Per-Port Cache for OpenFlow Switch
4. Evaluation System Design
5. Performance Conclusion

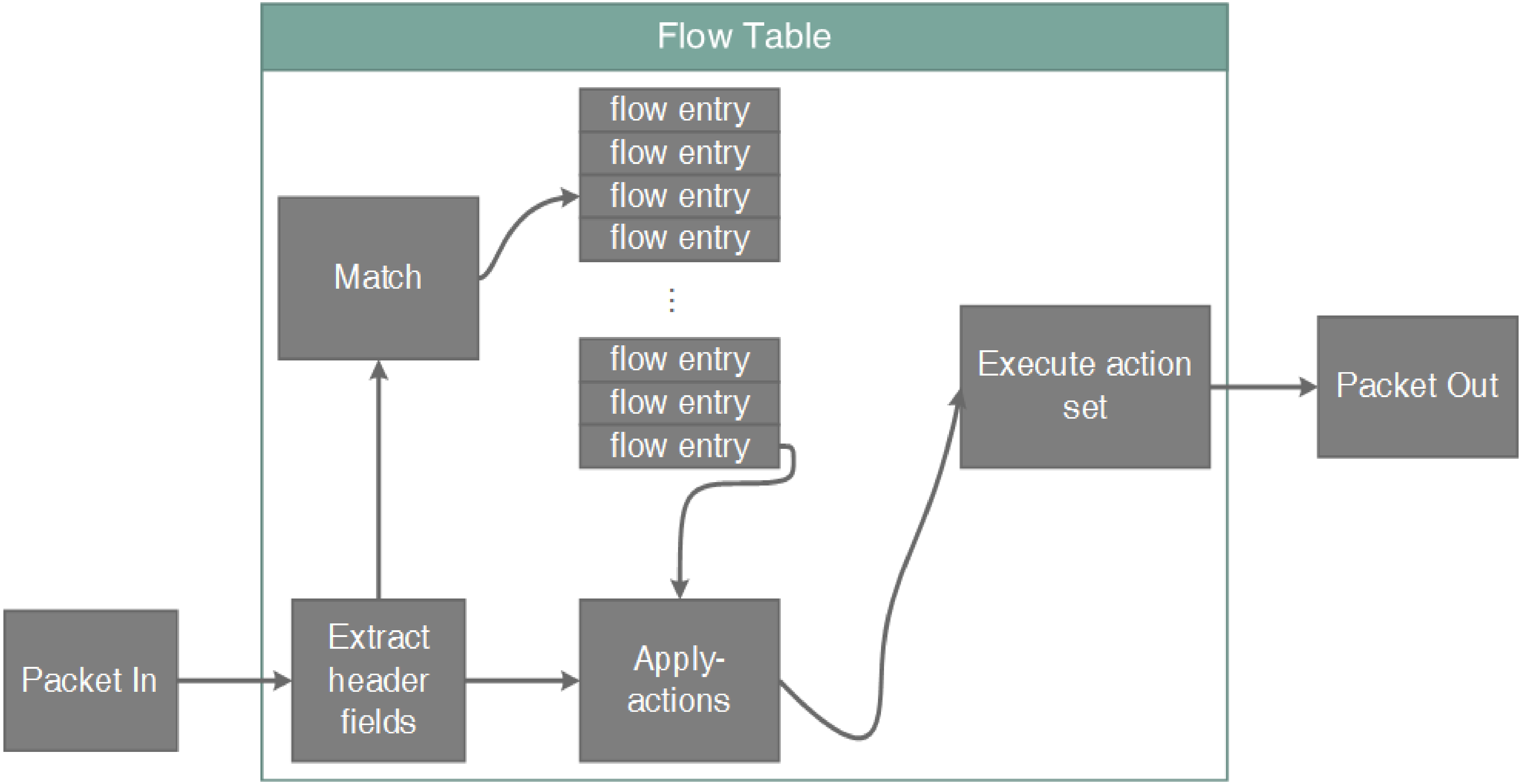
Software Defined Network



- **Definition:**
  - Software Defined Networking (SDN) is a network architecture where control is decoupled from forwarding and is directly programmable. (ONF, 2012, p. 7)
- **Open Networking Foundation (ONF) has dedicated to the promotion of SDN since 2011**

A Network Device stores instructions from the controller in a flow table.

Matching Inside a Flow Table



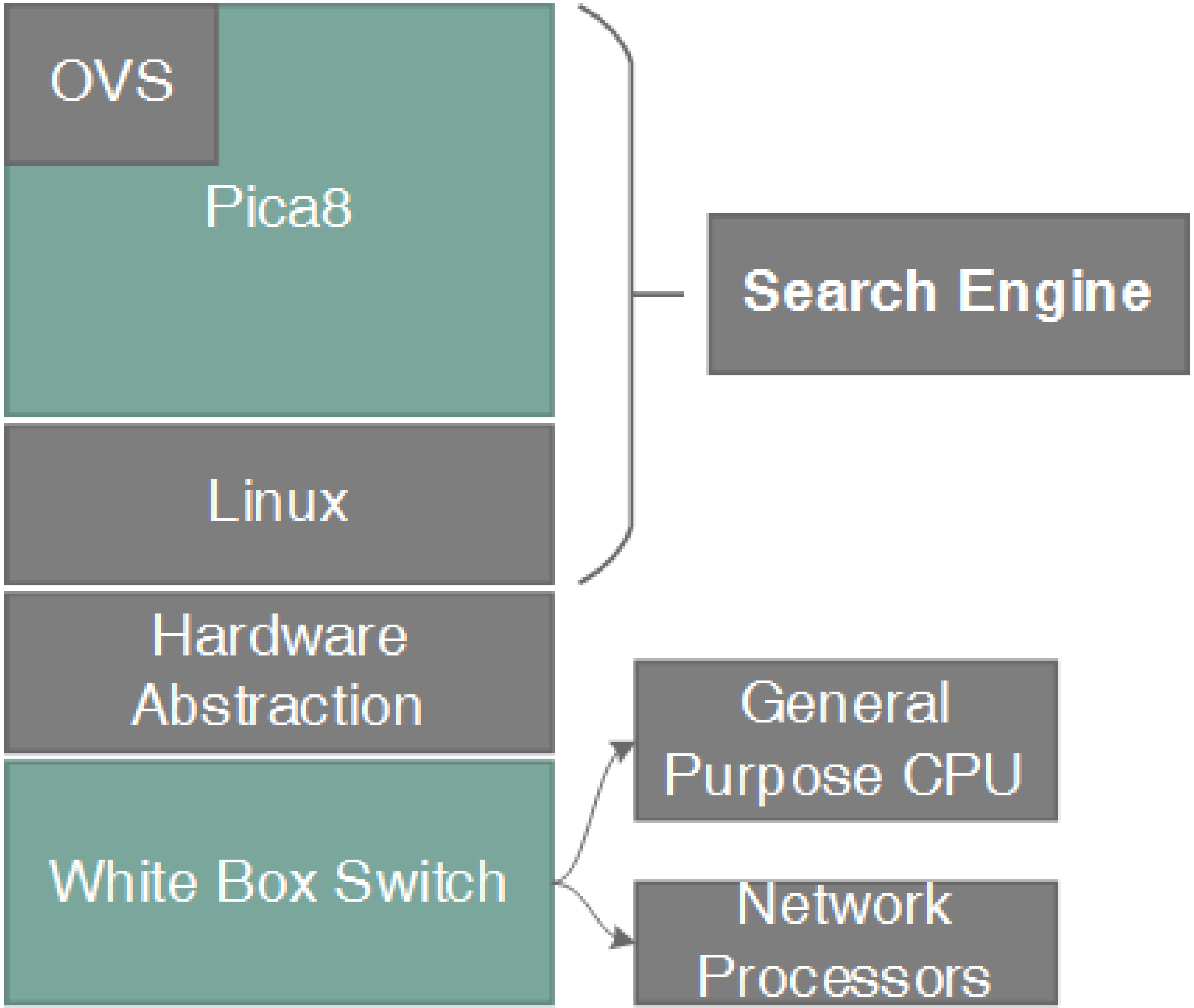
SDN Benefits and Issues

- **Benefits**
  - Reduced Network Complexity
  - Higher rate of innovation
  - Better user experience
- **Issues**
  - Network reliability
  - Communication latency and bottleneck between switch and controller
  - **OpenFlow switch forwarding speed✓**

OpenFlow Switch

- **Definition**
  - An OpenFlow switch is a software program or hardware device that forwards packets in a software-defined networking (SDN) environment.
- **Categories**
  - Hardware switch
    - Extreme Networks – BlackDiamond X8
  - Software switch
    - Open vSwitch (OVS)
  - Whitebox switch[1]
    - Pica8

White Box Switch Architecture (Pica8)



Motivation

- **Whitebox switch can do million packet lookups / sec [2]**
- **Future switches require billion packet lookups / sec**
  – Whitebox switch's performance is lower than the requirement
- **Goal**
  – Improve the white box switch performance without modifying the search engine

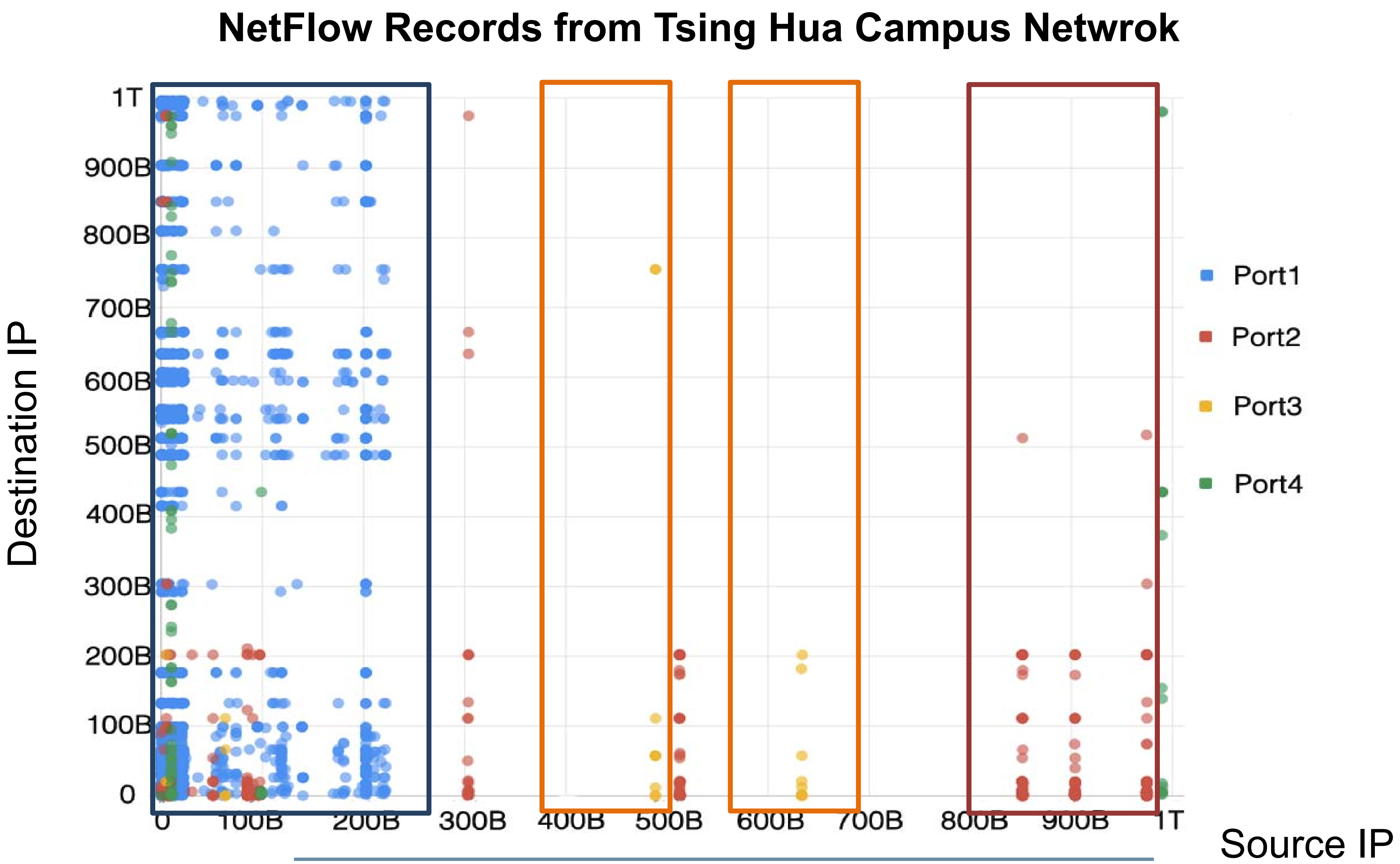
Related Research

- Cache memory design for network processors [3]
- Accelerating openflow switching with network processors [4]
- Using hardware classification to improve PC-based OpenFlow switching [5]
- Engineered Elephant Flows for Boosting Application Performance in Large-Scale CLOS Networks [6]

Locality in Network Traffic

- **Previous researches indicated:**
  - Network traffic has temporal locality
  - Using cache memory for NP can accelerate long-lived (elephant) flows
- **Little attention has been paid to spatial locality**

Analysis

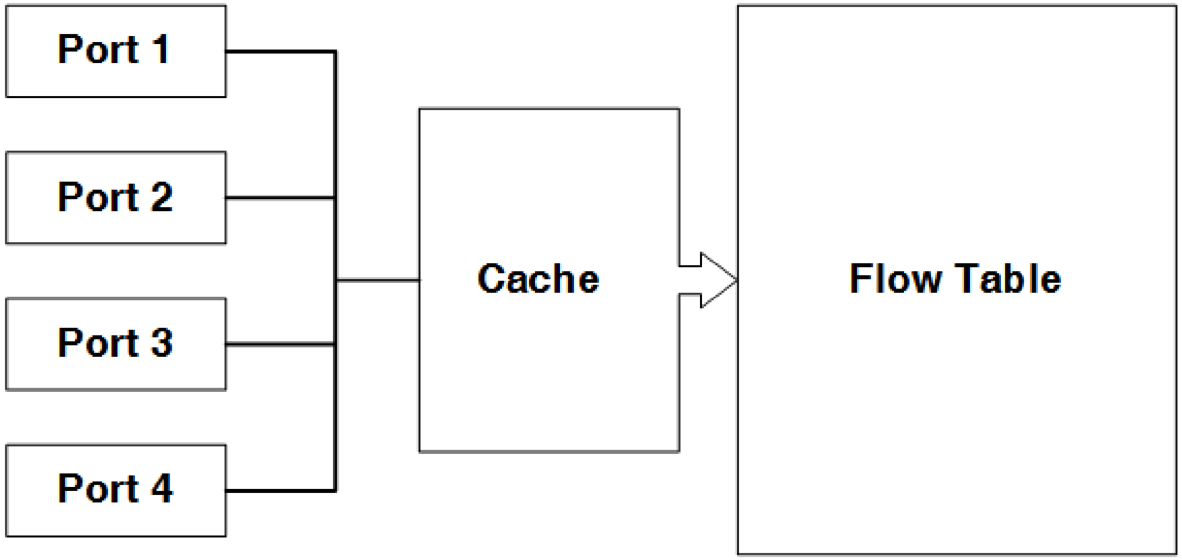

NetFlow Records from Tsing Hua Campus Netwrok
Destination IP
1T
900B
800B
700B
600B
500B
400B
300B
200B
100B
0
0
100B
200B
300B
400B
500B
600B
700B
800B
900B
1T
Source IP
Port1
Port2
Port3
Port4

Observation

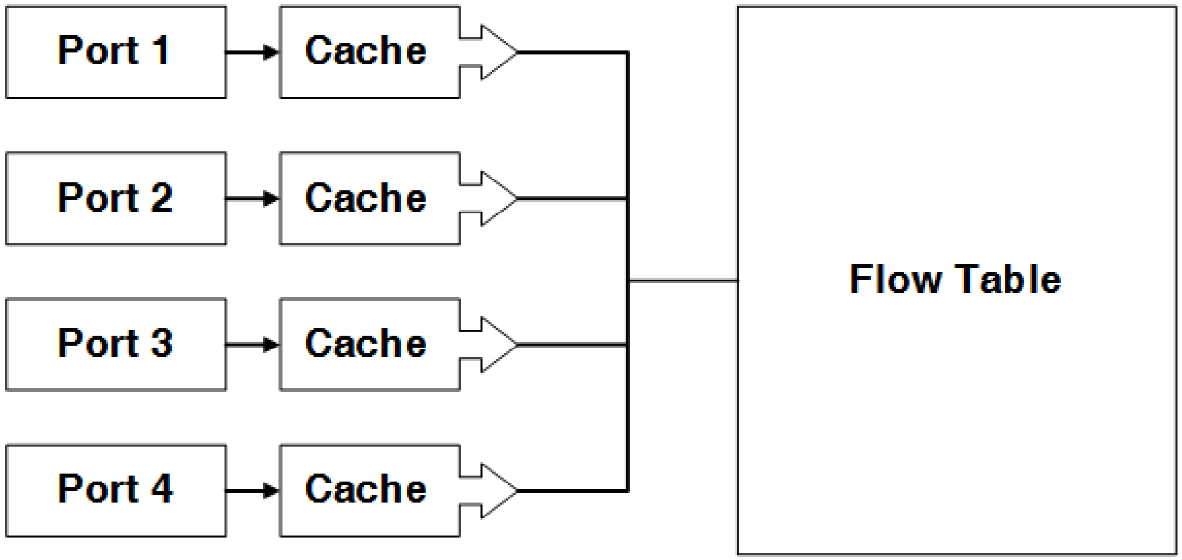
- We can group flows according to their source ports
  - A kind of spatial locality
- New design idea
  - Per-port cache
    - Based on the spatial locality

Per-Port Cache

- Central Cache



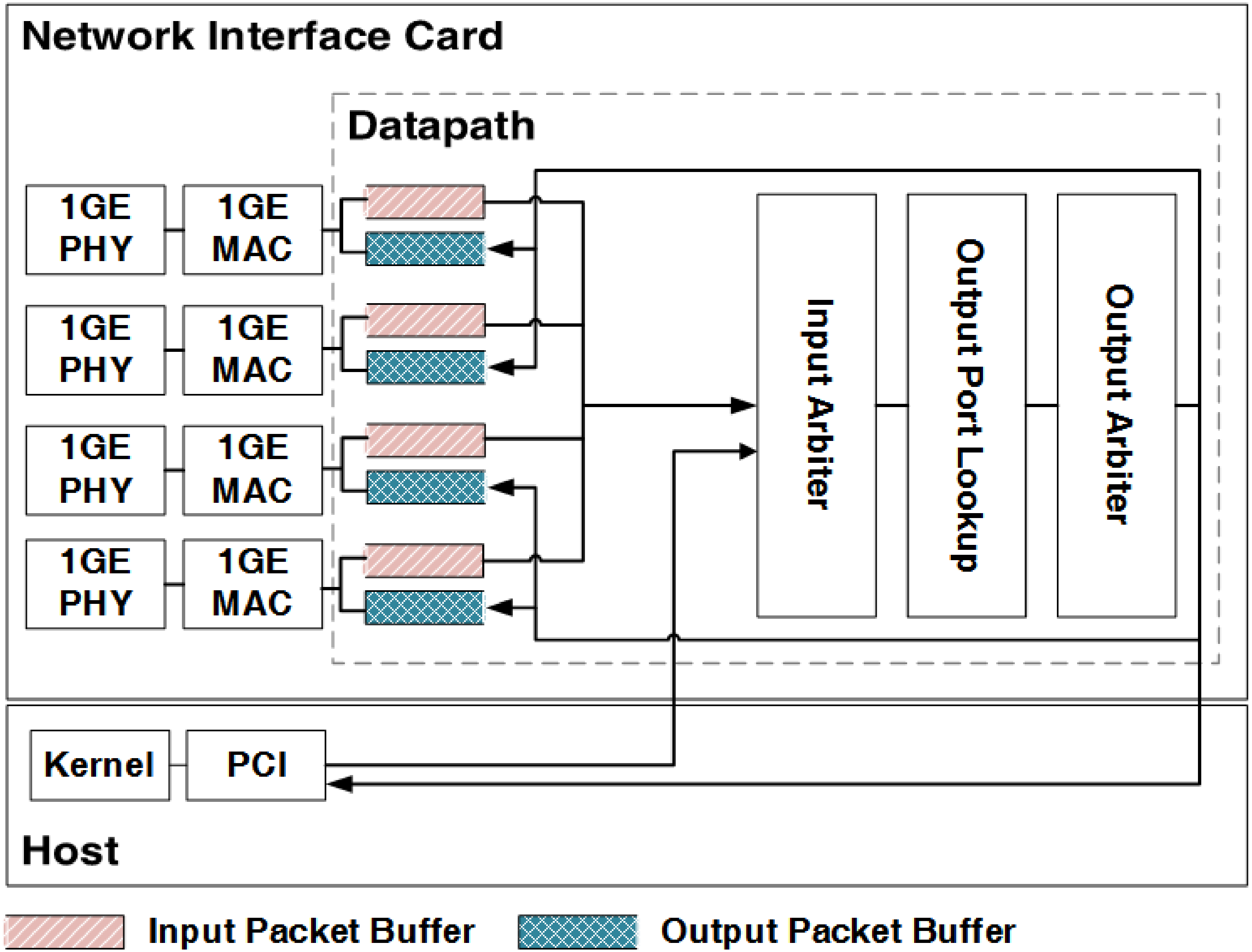
- Per-port Cache



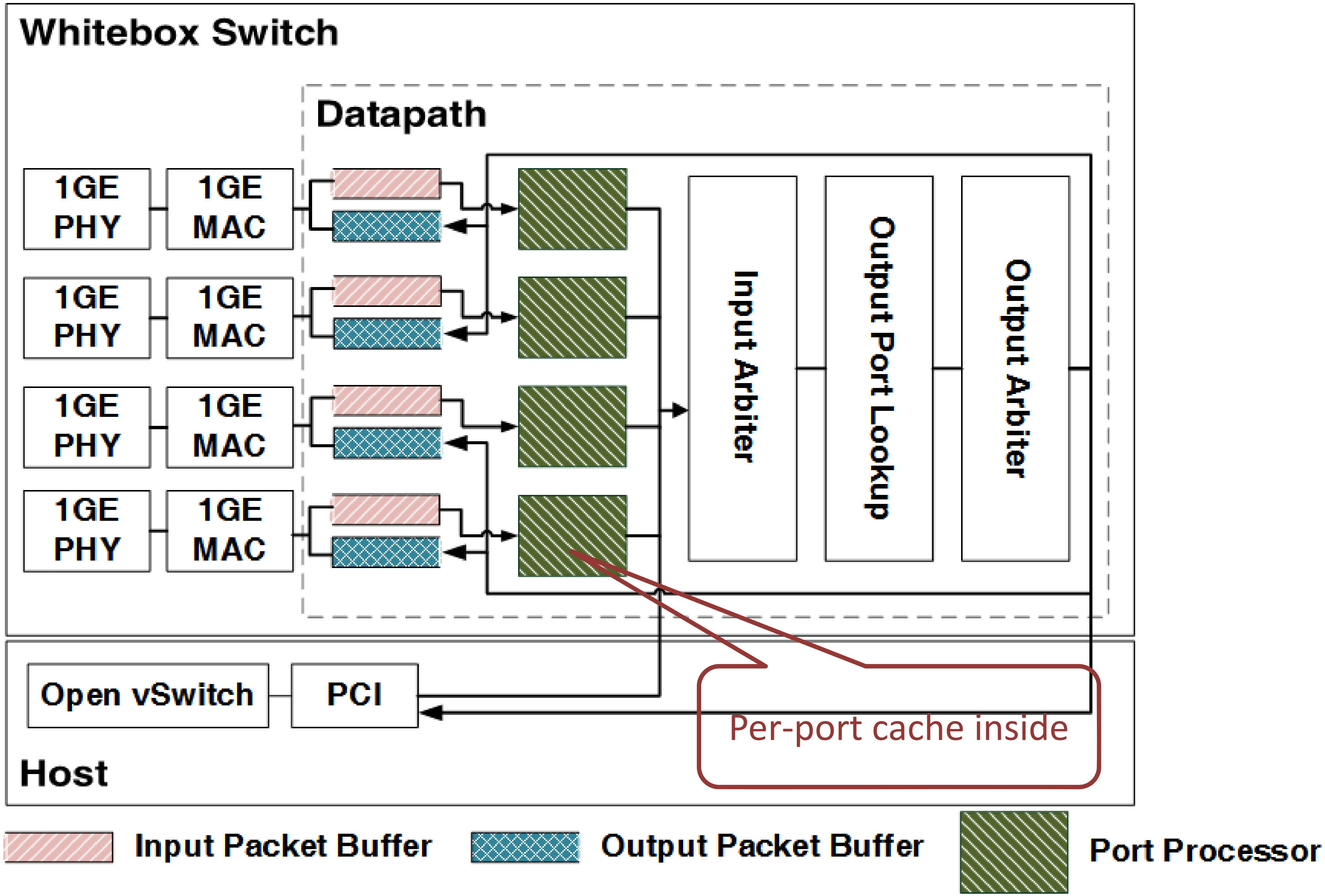
Per-port Cache Implementation

- **We employ a Whitebox Switch simulation equipped with our per-port cache**
- **Whitebox Switch**
  - Hardware Datapath with Per-port Cache
    - Modify from an NIC Verilog project in NetFPGA
  - Software Search Engine
    - A C-based Open vSwitch

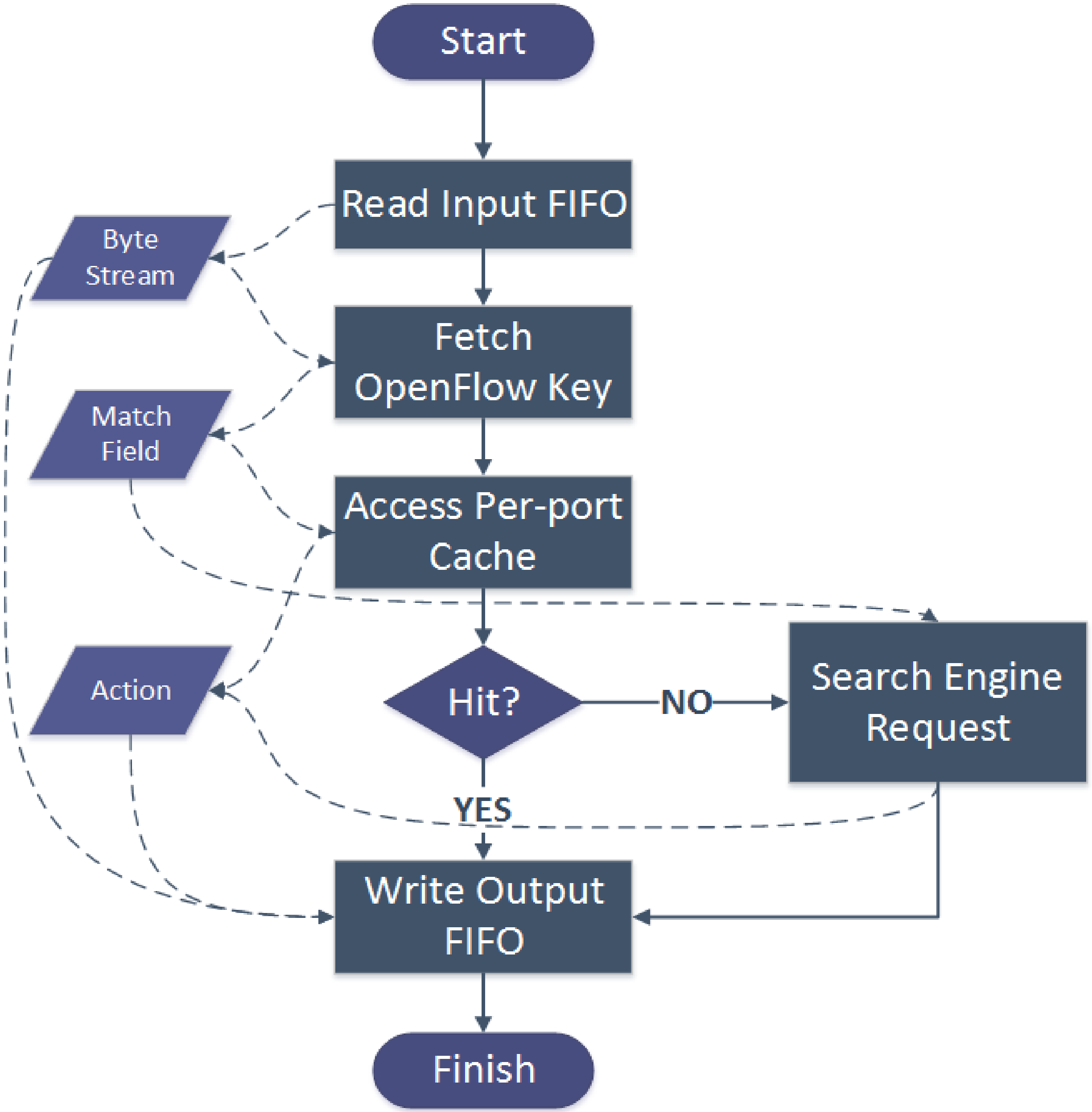
Hardware design - NIC



Hardware design – Switch



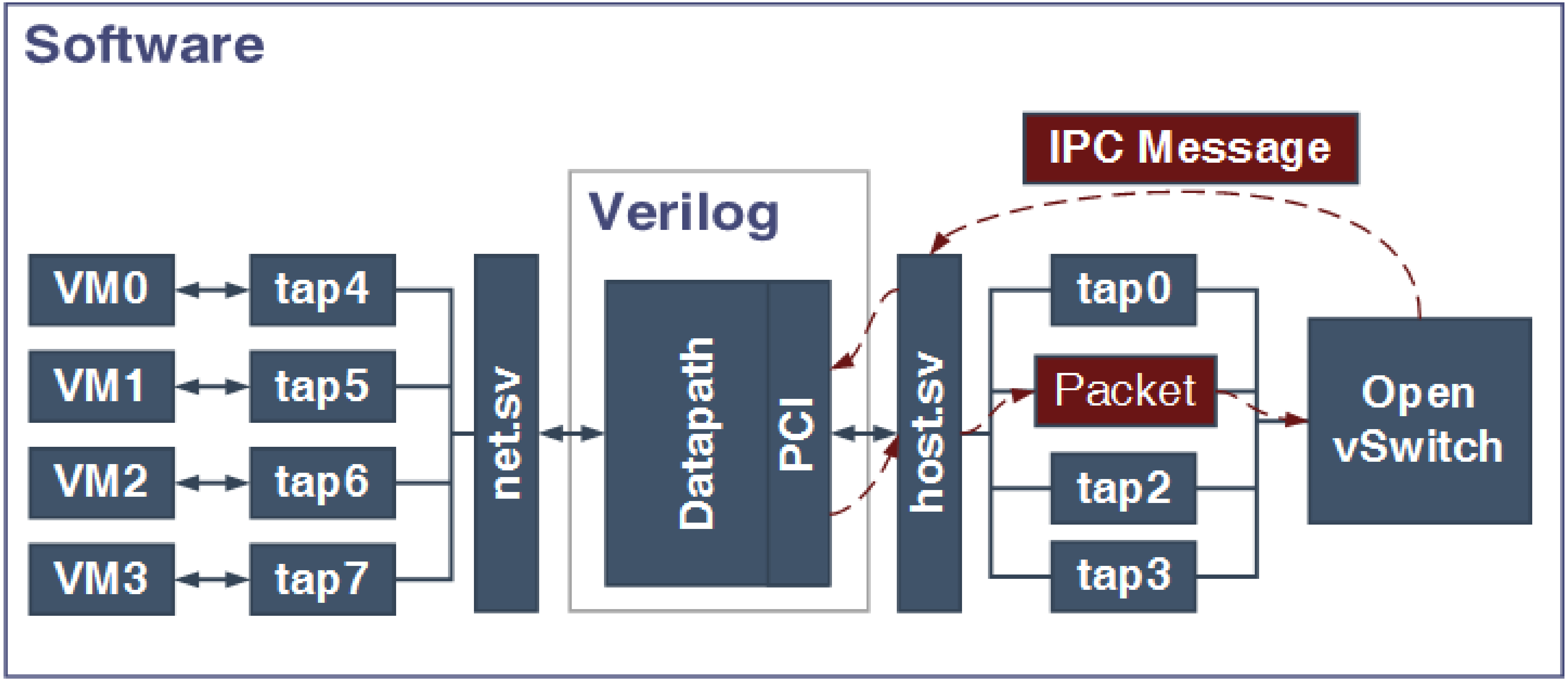
Port Processor Behavior



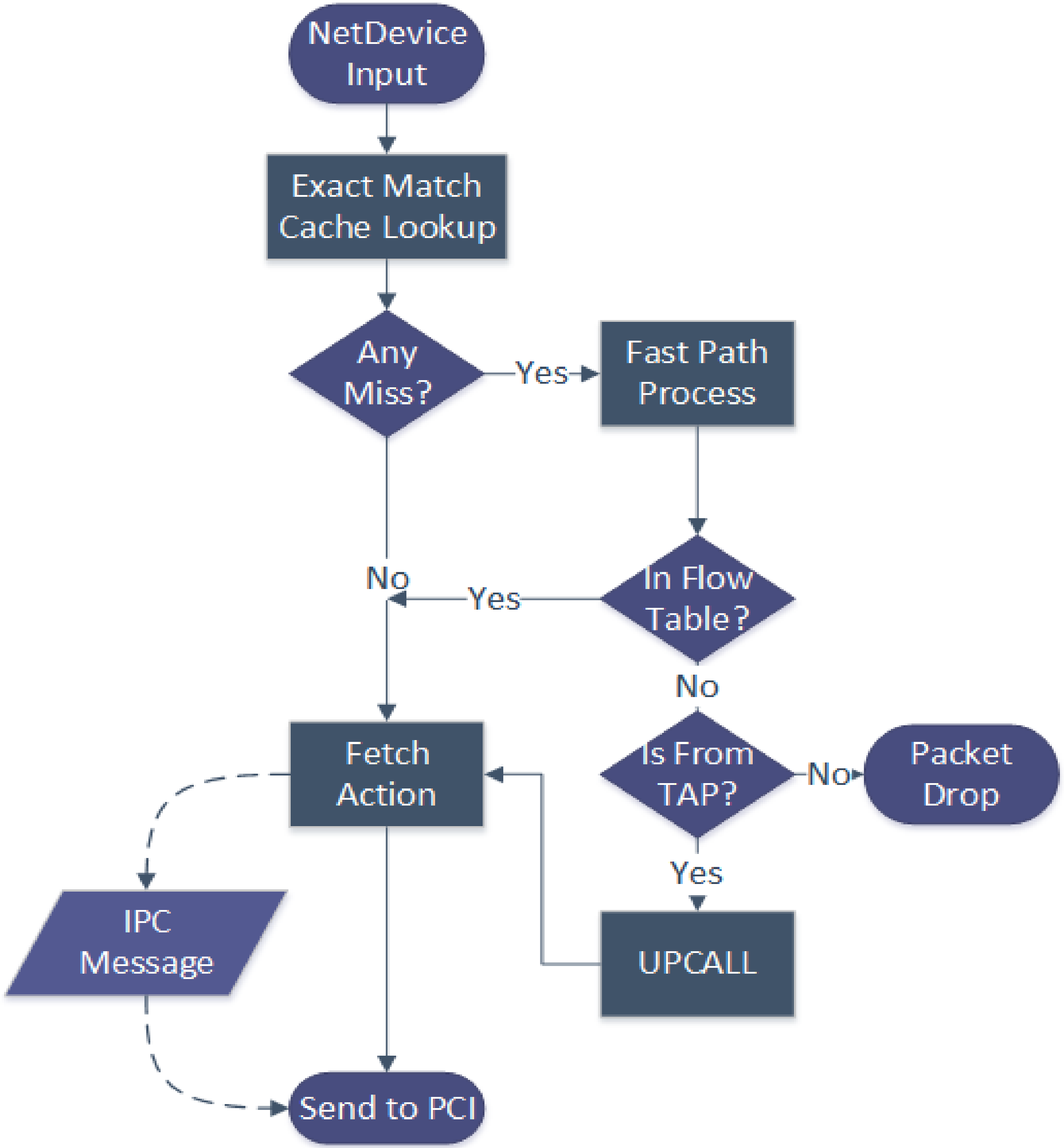
White Box Switch Co-Simulation

- **Hardware module**
  - Simulated using ModelSim
- **Software search engine**
  - Open vSwitch
- **Bridge**
  - Inter-process communication(IPC) message
  - TAP devices
    - A virtual net device collecting network packets.
  - SystemVerilog

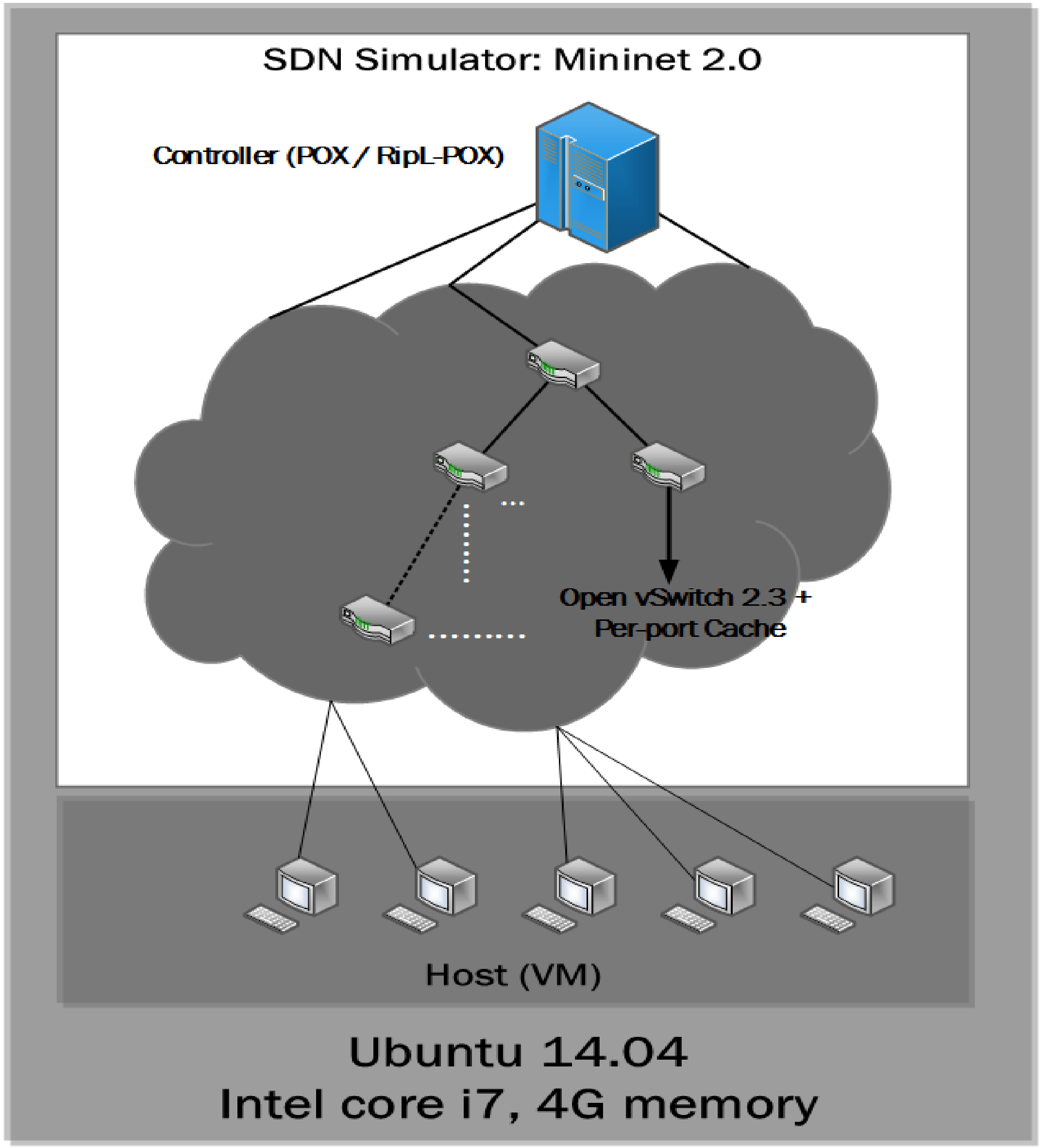
Co-Simulation Architecture



Open vSwitch Behavior



Experiment Setup



Measurement

- **Quality Metrics**
  - Hit rate of per-port cache
  - Average access time of flow table
- $Avg.AccessTime = R_{hit} \times T_{hit} + R_{miss} \times (T_{penalty} + T_{lookup})$
  - R_{hit} , R_{miss} = Hit rate and miss rate
  - T_{hit} = Hit time, T_{lookup} = Flow table lookup time
  - $T_{penalty}$ = miss penalty
    - (Cache access, waiting time for search engine and PCI transfer time)
- **Compare to Open vSwitch**

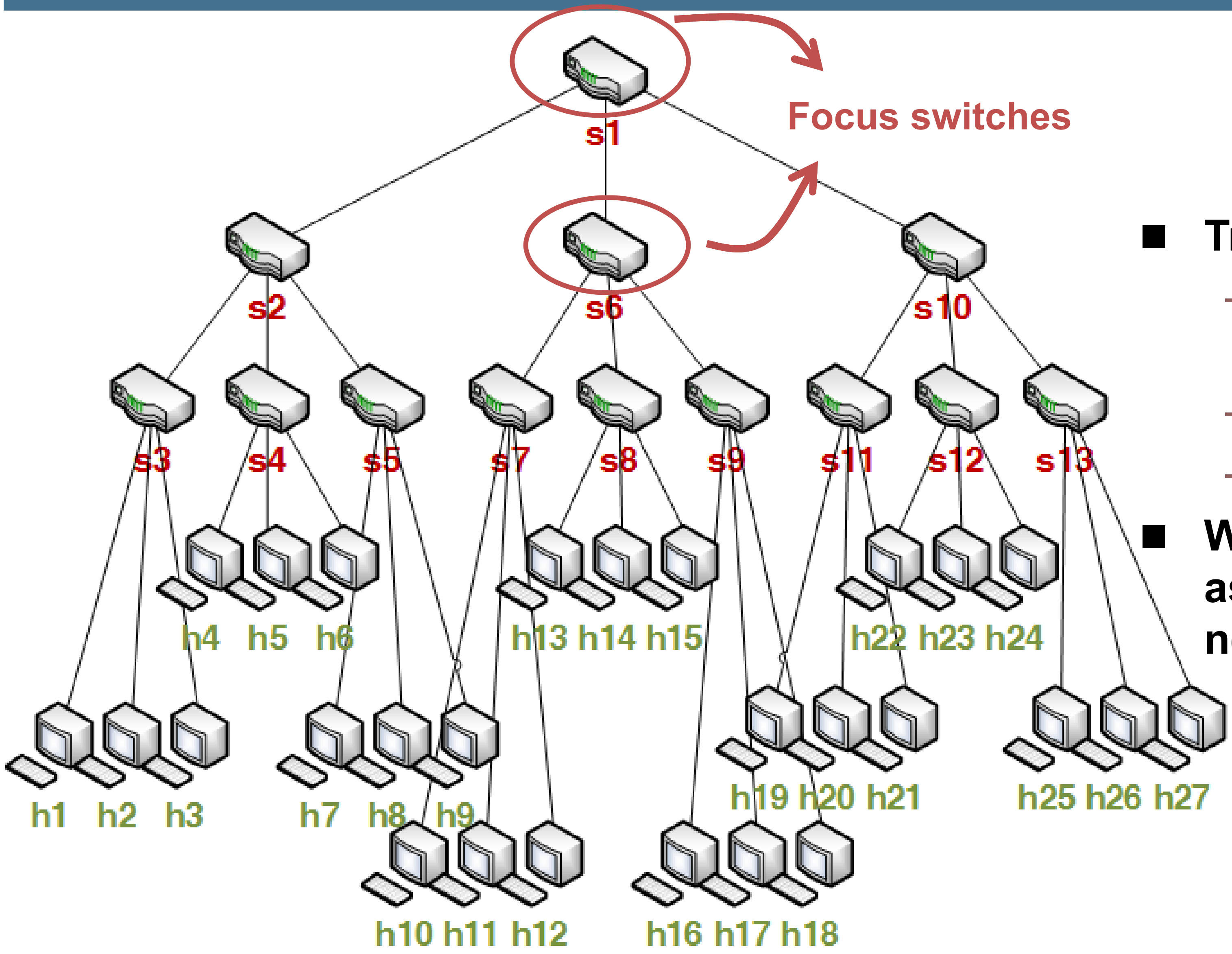
Experiment Parameters

- **Cache Configuration**
  - Cache size and replacement policy.
- **✓Traffic Pattern**
  - ICMP packets. (By Ping)
- **✓Topology**
  - Equal cost-multi path(ECMP) and Tree topologies

Traffic Pattern

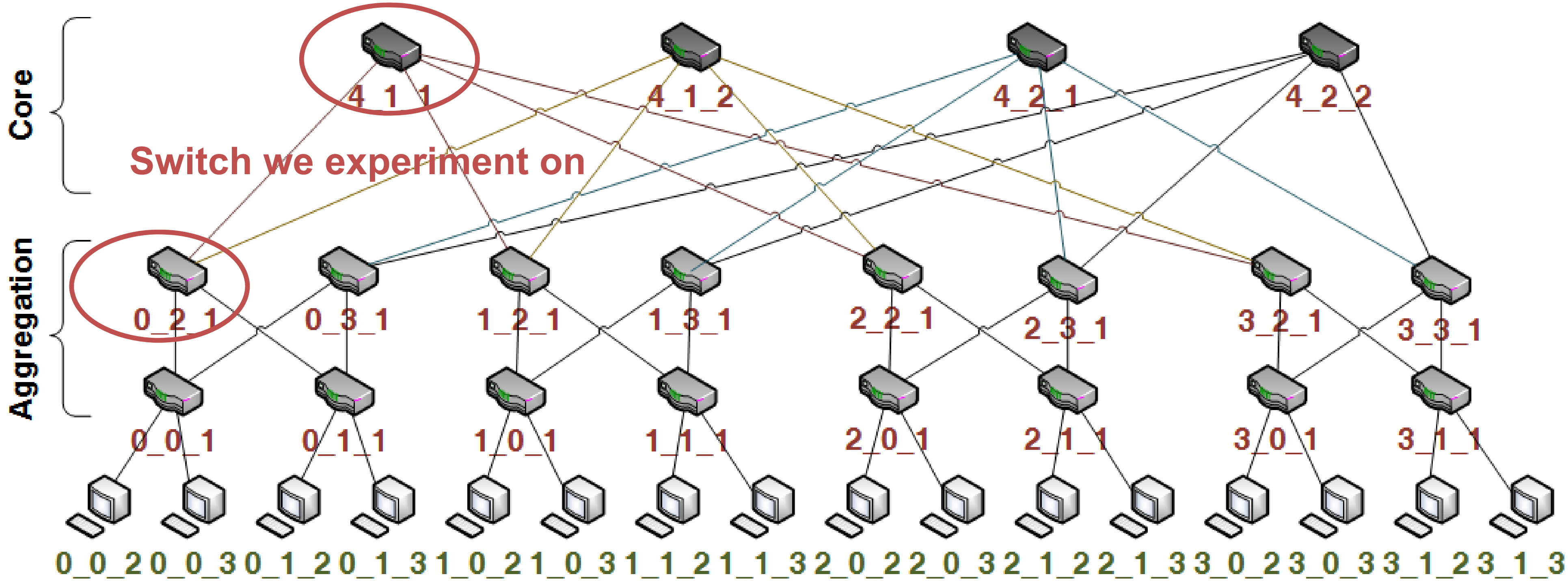
- **Understanding data center traffic characteristics (T. Benson et al. in 2010)**
- **Parameters of flow**
  - Inter-arrival Time
  - On-Period
  - Off-Period
- **Lognormal distribution**
  - $\frac{1}{2} erfc\left(-\frac{\ln x-\mu}{\sigma\sqrt{2}}\right) = \Phi\left(\frac{\ln x-\mu}{\sigma}\right), where\ \mu = 0.9\ and\ \sigma = 2.3$
    - Most values before saturation are between 1 to 10 (s)

Network Topology - Tree



- **Tree**
  - Widely used in campus network
  - Easy to understand
  - Good scalability
- **We use this topology as our simulated network**

Network Topology - ECMP

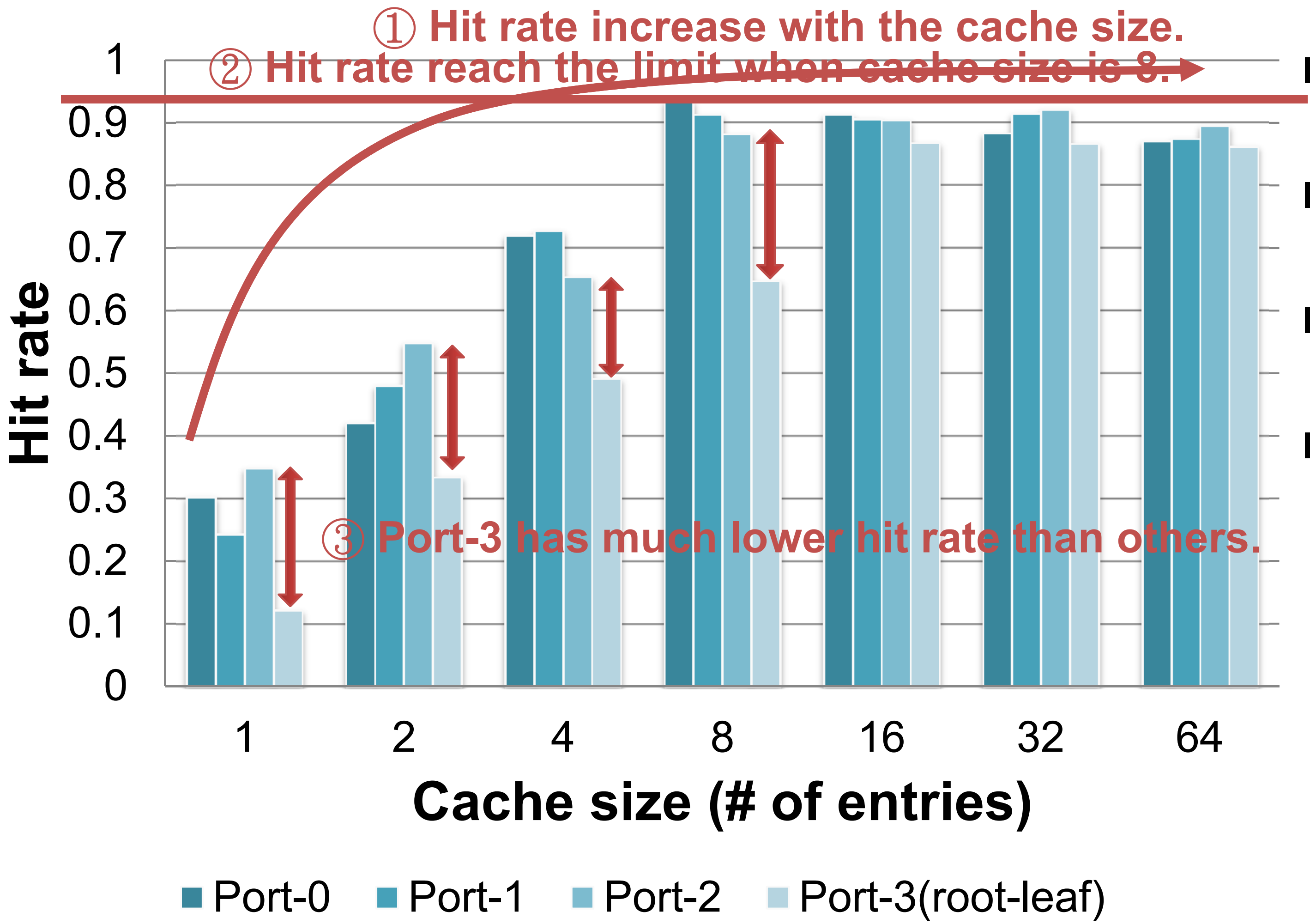


- **Equal Cost Multi Path (ECMP)**
  - Widely used in data center
  - Routing algorithm is simple
  - Easy to do utilization
- **We use this topology as our simulated network**

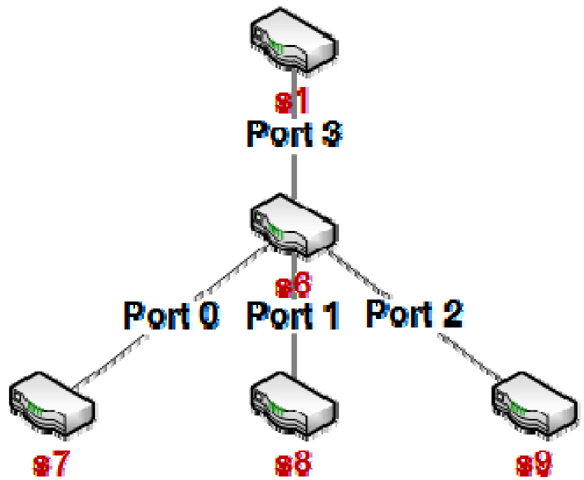
Experiment Result Overview

- Hit rate as function of cache size (4)
- Average hit rate and cache size (1)
- Hit rates for different cache replacement policy (1)
- Average flow table access time (2)

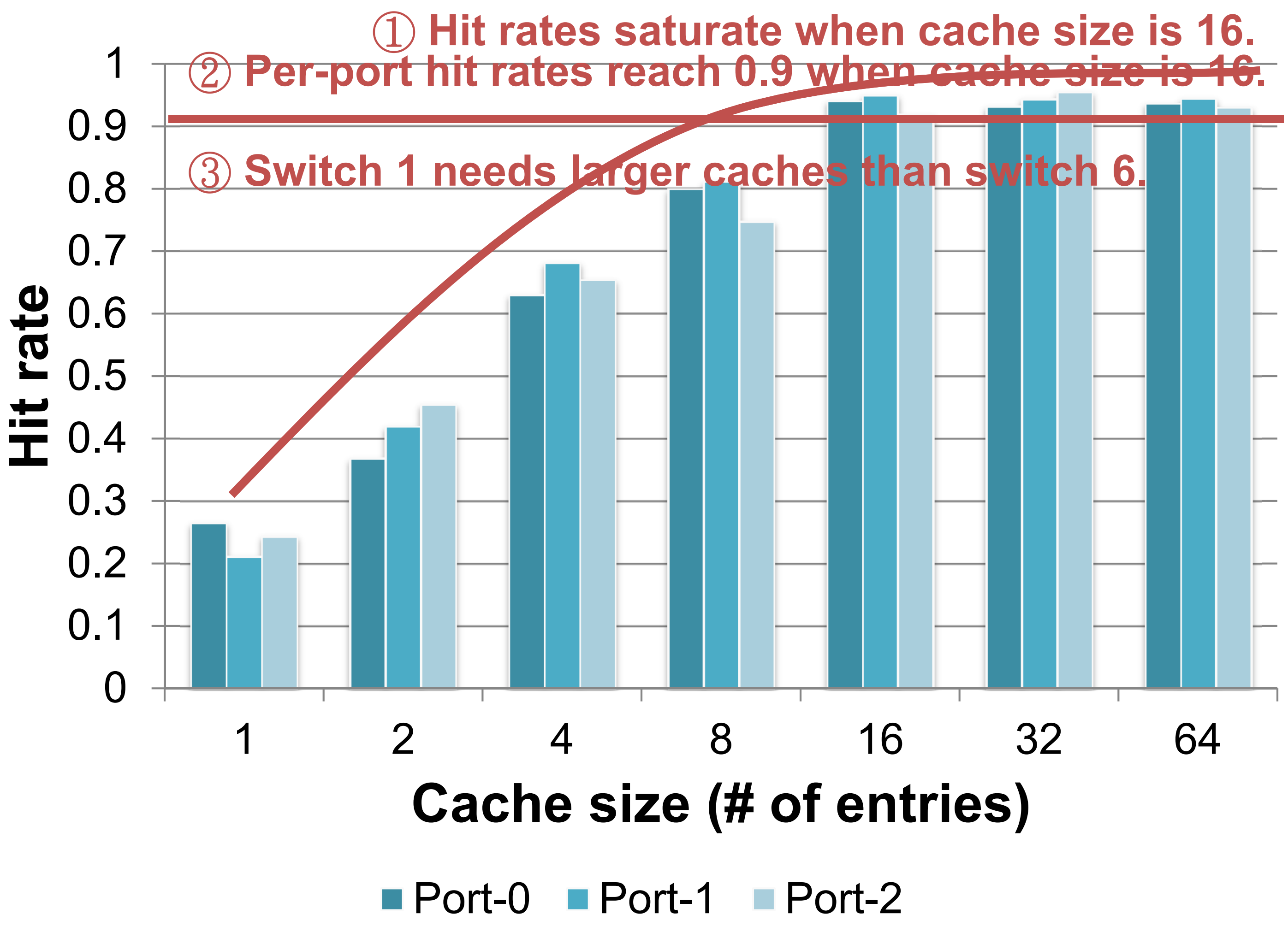
Switch 6 Hit Rate as Function of Cache Size



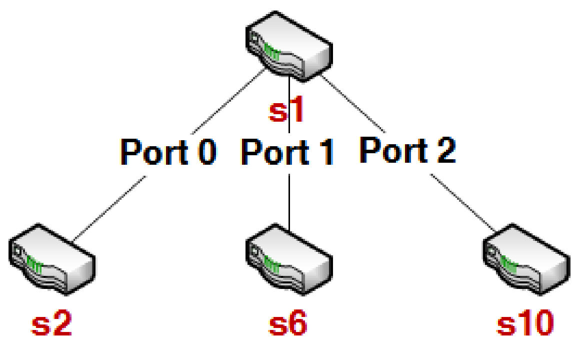
- **Topology**
  - Tree
- **Routing Algorithm**
  - L3 learning switch
- **Traffic generator**
  - Ping
- **Replacement policy**
  - LRU



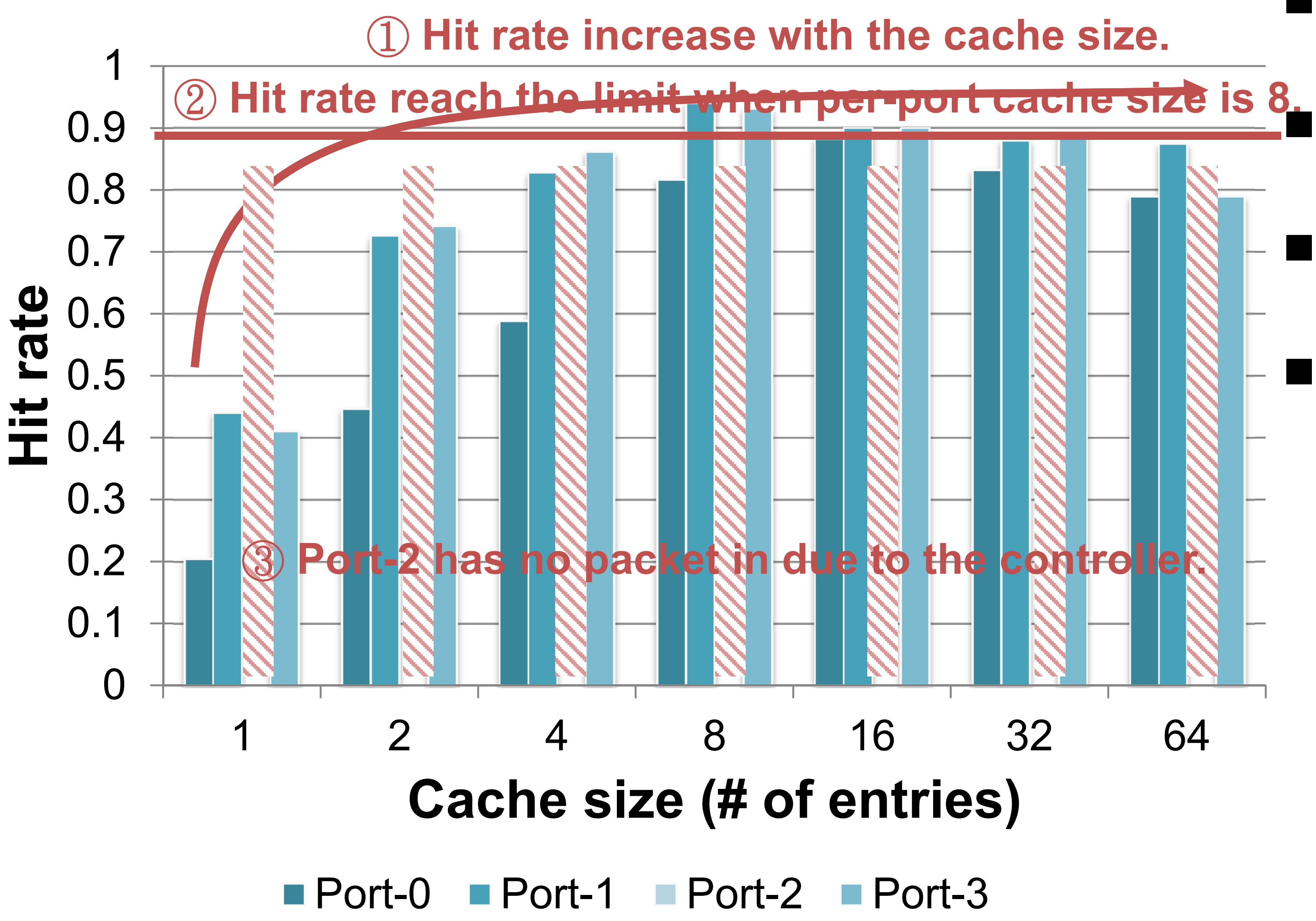
Switch 1 Hit Rate as Function of Cache Size



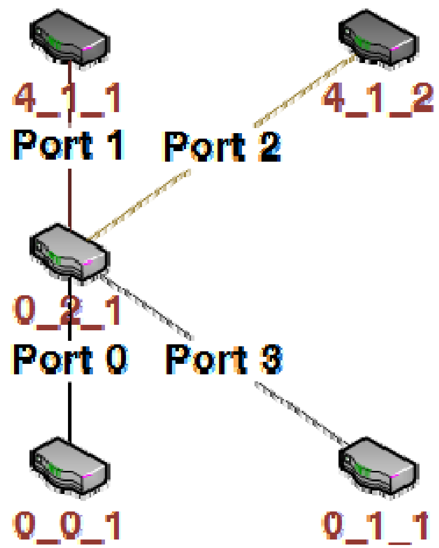
- **Topology**
  - Tree
- **Routing Algorithm**
  - L3 learning switch
- **Traffic generator**
  - Ping
- **Replacement policy**
  - LRU



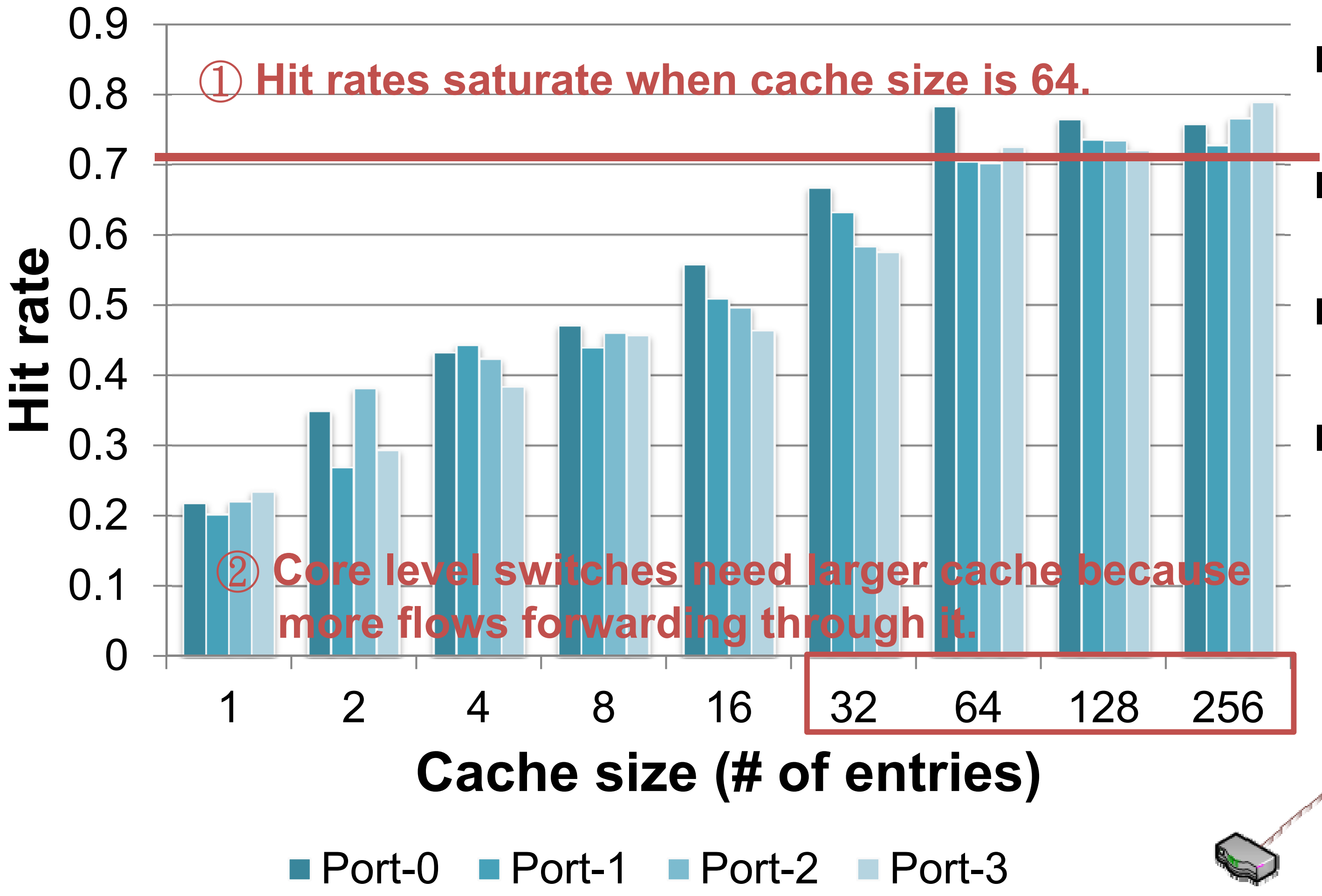
Switch 0_2_1 Hit Rate as Function of Cache Size



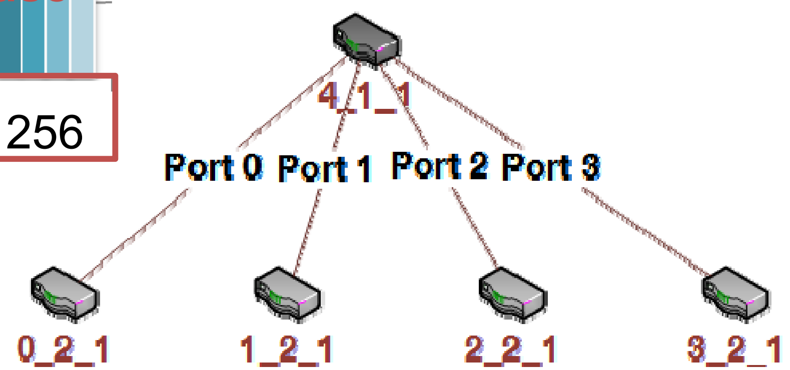
- **Topology**
  - ECMP
- **Routing Algorithm**
  - Spanning tree (Hash)
- **Traffic generator**
  - Ping
- **Replacement policy**
  - LRU



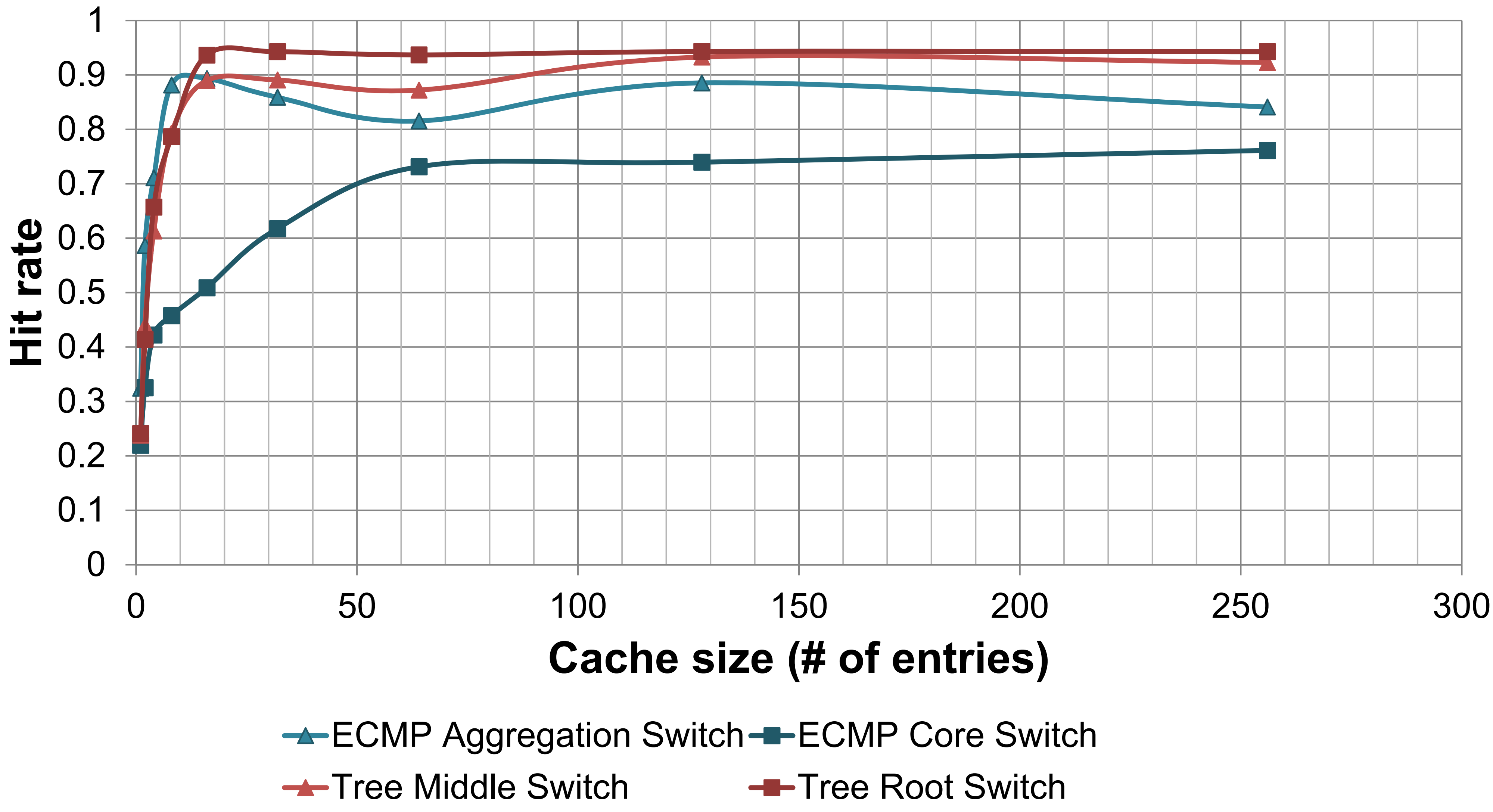
Switch 4_1_1 Hit Rate as Function of Cache Size



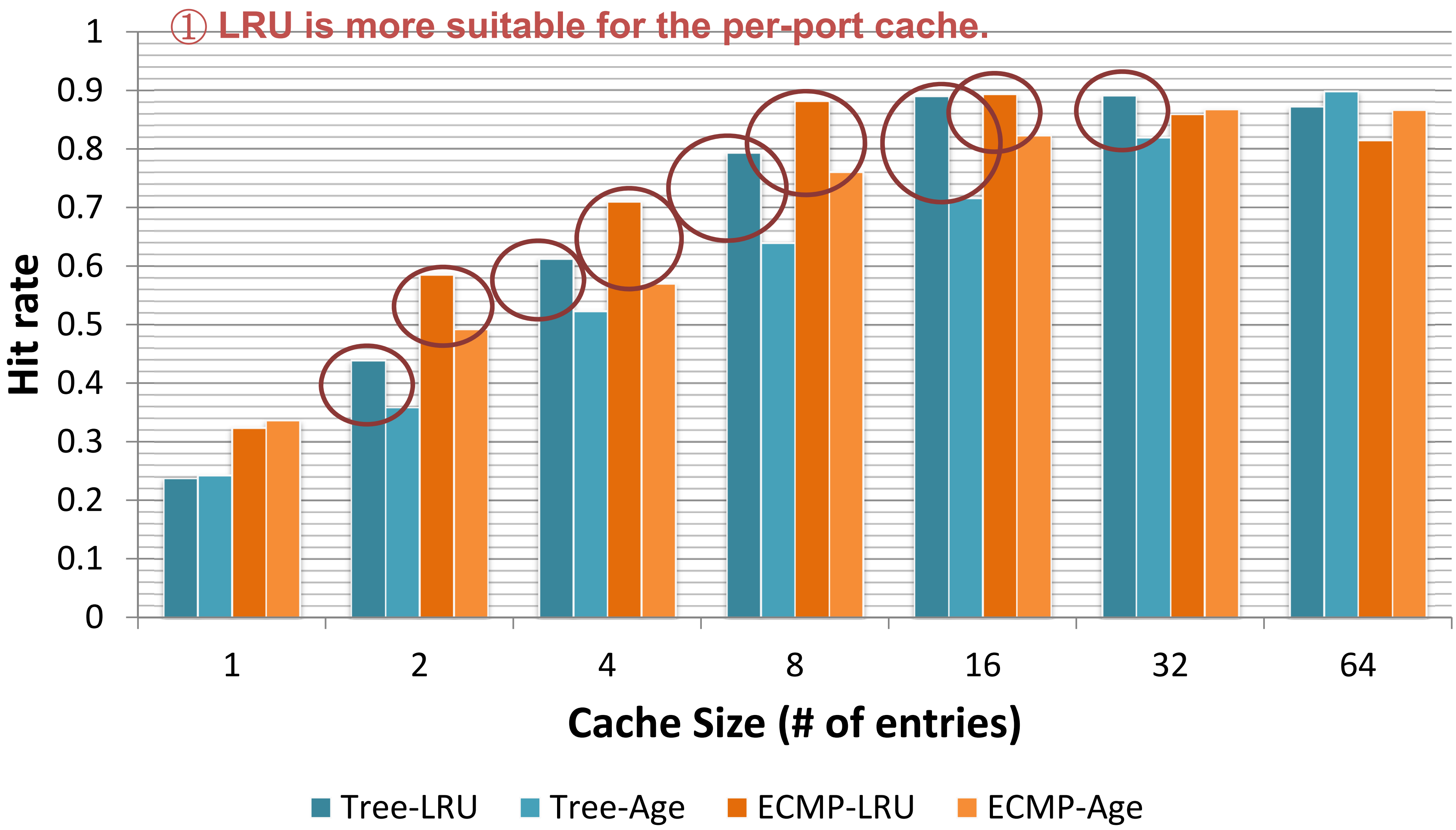
- **Topology**
  - ECMP
- **Routing Algorithm**
  - Spanning tree (Hash)
- **Traffic generator**
  - Ping
- **Replacement policy**
  - LRU



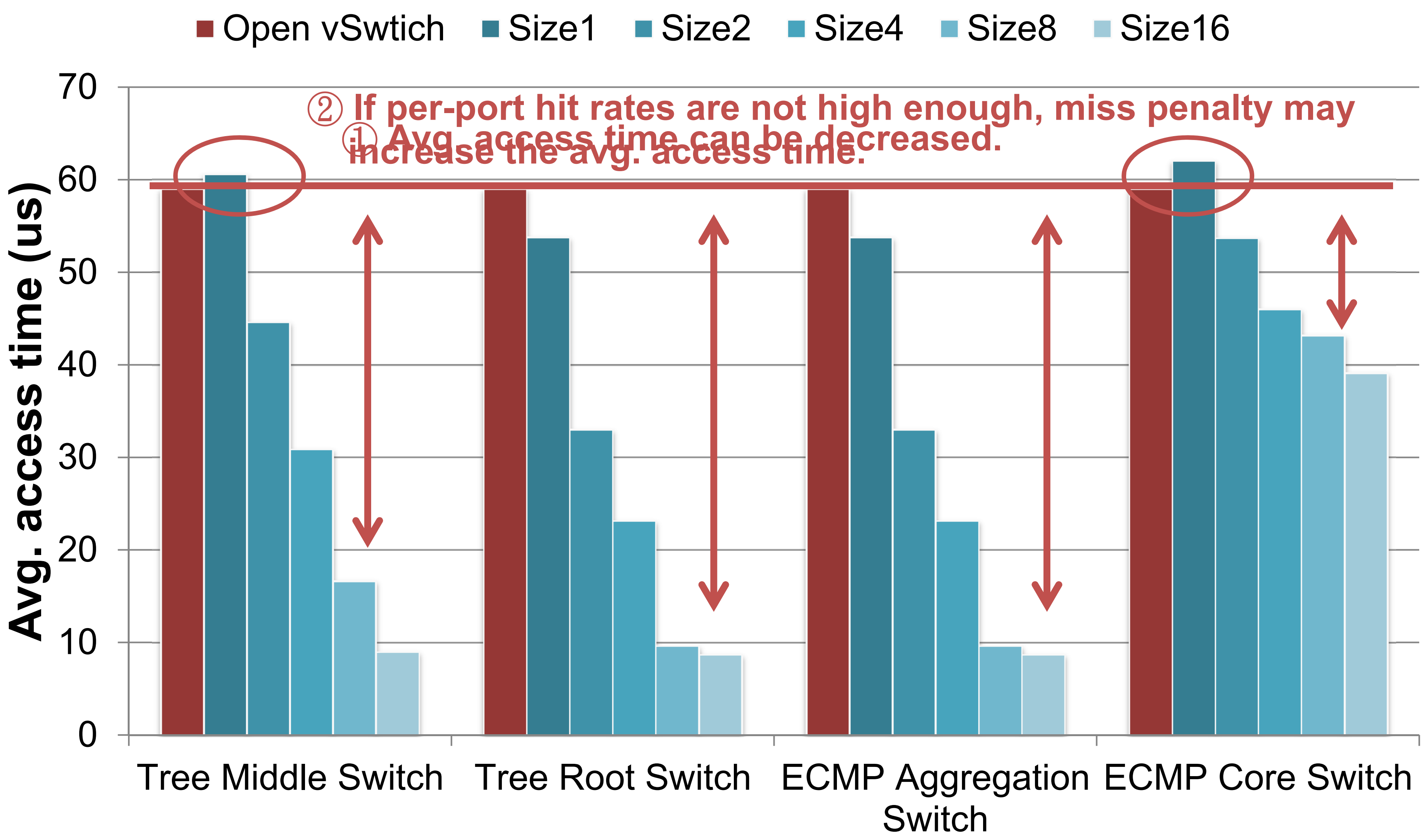
Average Hit Rate vs. Cache Size



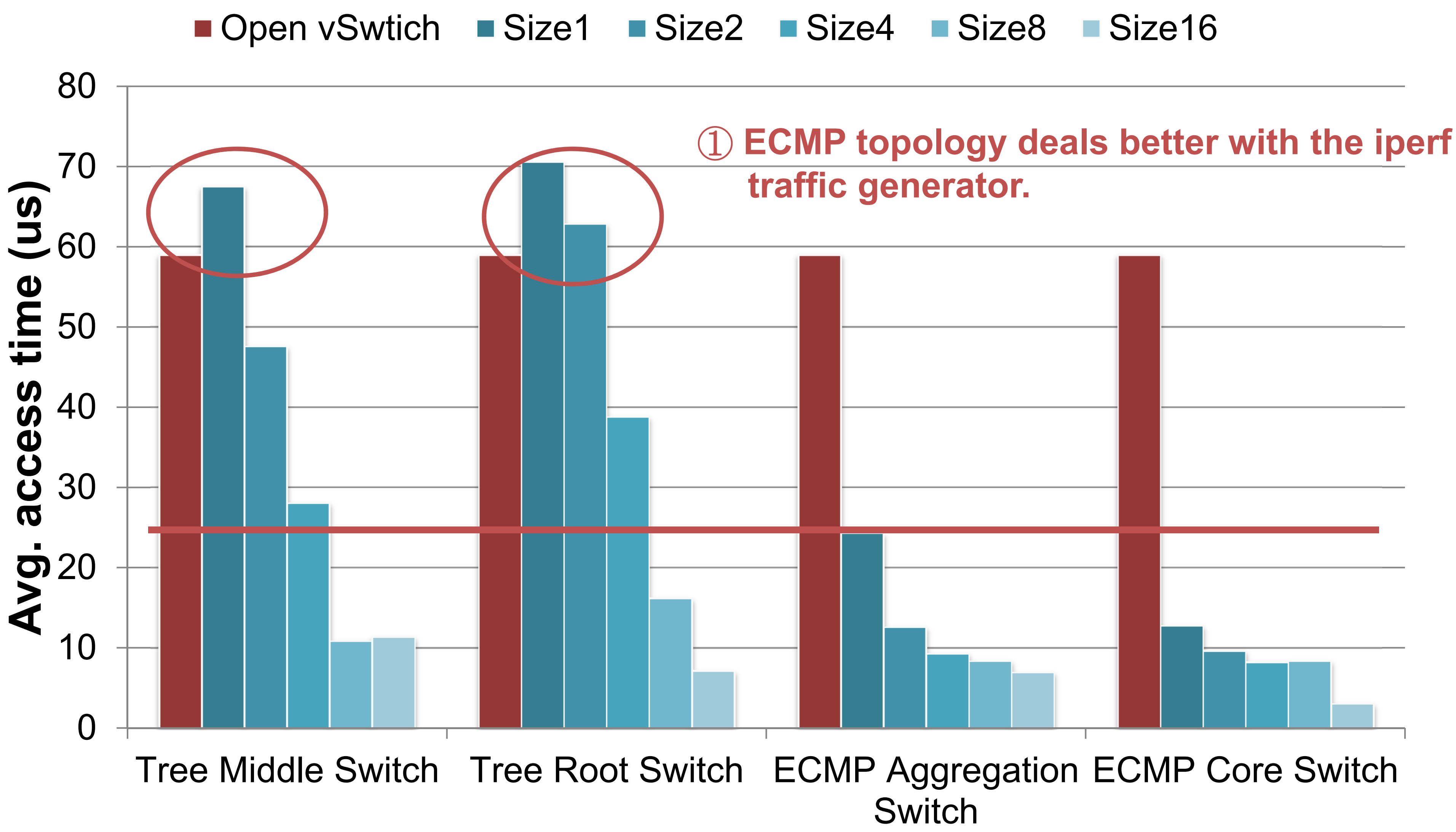
Hit Rates vs Replacement Policies



Average Access Time for Ping Traffic



Average Access Time for iperf Traffic



Summary

- Network flows have spatial locality
- Traffic pattern and cache replacement policy affect the per-port cache performance
  - LRU is suitable for our per-port cache design
- Our per-port cache design can significantly improve the switch performance with little overhead

Future Work

- **Routing policy has potential to affect per-port cache performance**
- **Distributing different entries size to per-port cache may be reasonable**
- **Building a real test bed to receive more accurate information**
  - With NetFPGAs

References

- [1]" What are white box switches? ", Available at https://www.sdxcentral.com/resources/white-box/what-is-white-box-networking/.
- [2] M. Zec, L. Rizzo, and M. Mikuc, "Dxr: Towards a billion routing lookups per second in software," SIGCOMM Comput. Commun. Rev., vol. 42, pp. 29-36, Sept. 2012.
- [3] T. Chiueh and P. Pradhan, " Cache memory design for network processors," in Proceedings. Sixth International Symposium on High-Performance Computer Architecture, 2000. HPCA-6, pp. 409-418, 2000.
- [4] Y. Luo, P. Cascon, E. Murray, and J. Ortega, "Accelerating openflow switching with network processors," in Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS '09, (New York, NY, USA), pp. 70-71, ACM, 2009.
- [5] V. Tanyingyong, M. Hidell, and P. Sjodin, "Improving pc-based openflow switching performance," in 2010 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), pp. 1-2, Oct 2010.
- [6] " Engineered elephant ows for boosting application performance in largescale clos networks," Available at https://www.broadcom.com/collateral/wp/OF-DPA-WP102-R.pdf.