Distributed Flow-Table Cache for OpenFlow Switch

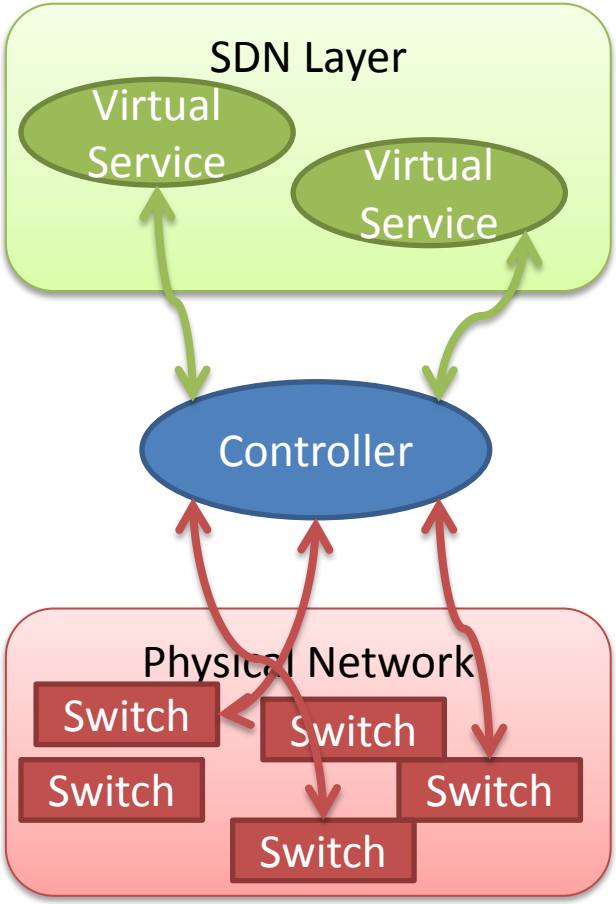
Chen-Yi Lin, Yu-Ting Zheng, Chih-Yi Y Lu, Youn-Long Lin
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan

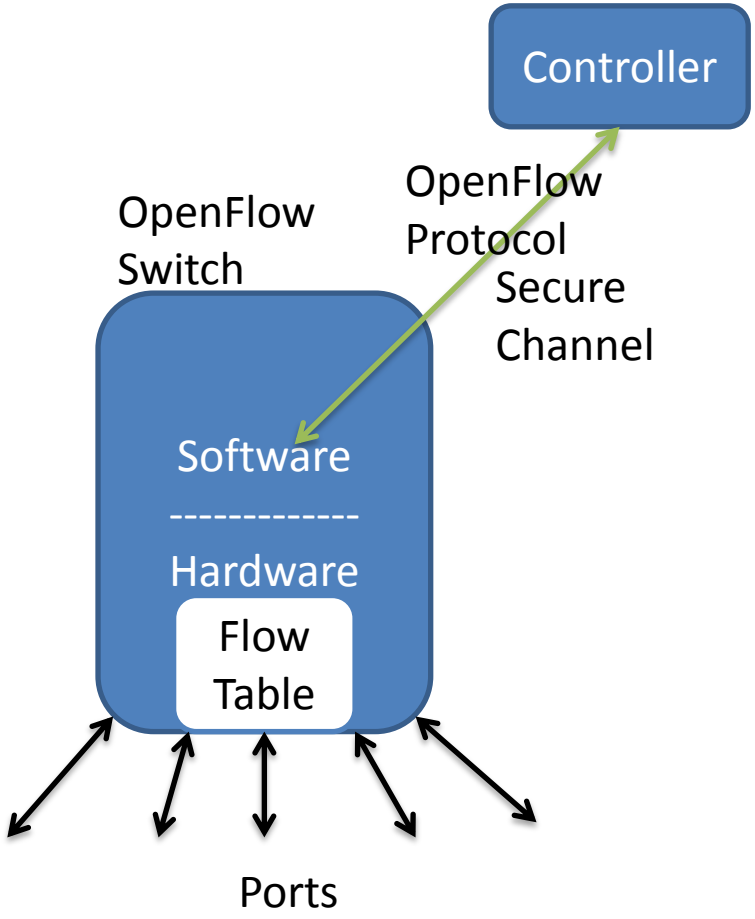
Outline

- Introduction to Software Defined Networking
- Bottleneck in Forwarding Table Access
- Proposed Distributed Per-Port Cache Architecture
- Experiments
- Conclusions

Software Defined Networking

Separation of Control from Data Plane





Essential Issues of SDN

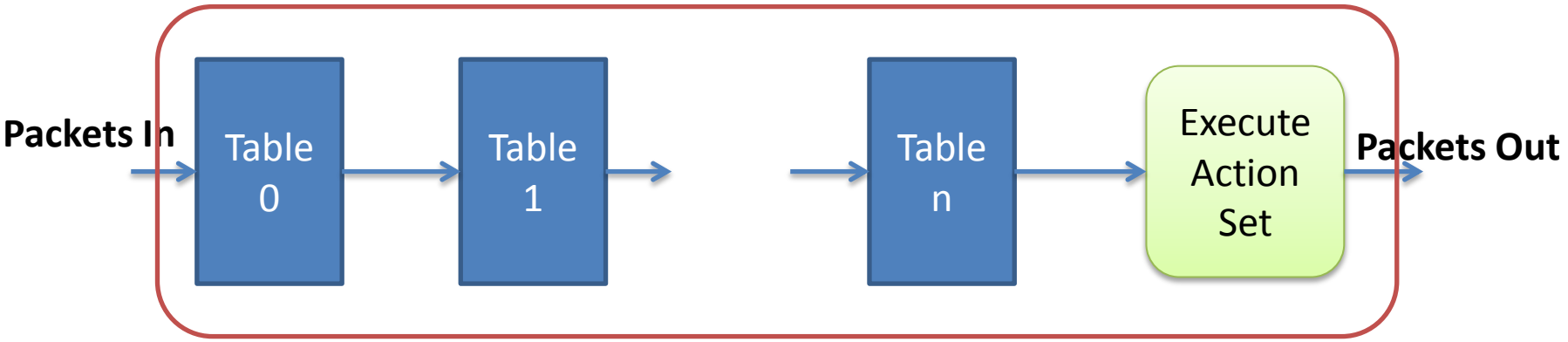
- Controller Loading, Reliability and Security
- Communication Between Controller and Routers/Switches
- Flow Table Update and Lookup
- Network Function Virtualization
- Scalability

Performance Requirement

- State-of-the-art OpenFlow Switch*
  - Throughput: 1.28Tbps
  - Forwarding Capacity: 960Mpps
  - Maximum MAC Addresses: 128K
  - Maximum Route: 12,000
  - Power Consumption: 400Watts
- Design goal
  - Multi-Billion Lookup Per Second @ Low Power

*PICA-8

SDN Flow Table Lookup:
Packets Matching Against Multiple Tables/Fields



Packet Fields:

- Ingress port
- Metadata
- Ether src
- Ether dst
- Ether type
- VLAN id
- VLAN priority
- MPLS label
- MPLS traffic class
- IPv4 src
- IPv4 dst
- IPv4 proto/ARP opcode
- IPv4 ToS bits
- TCP/UDP/SCTP src port
- ICMP type
- TCP/UDP/SCTP dst port
- ICMP code

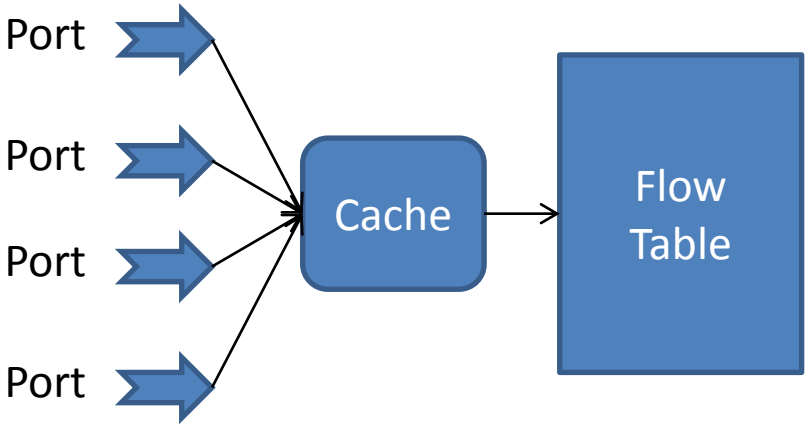
Possible Actions in a Stage:

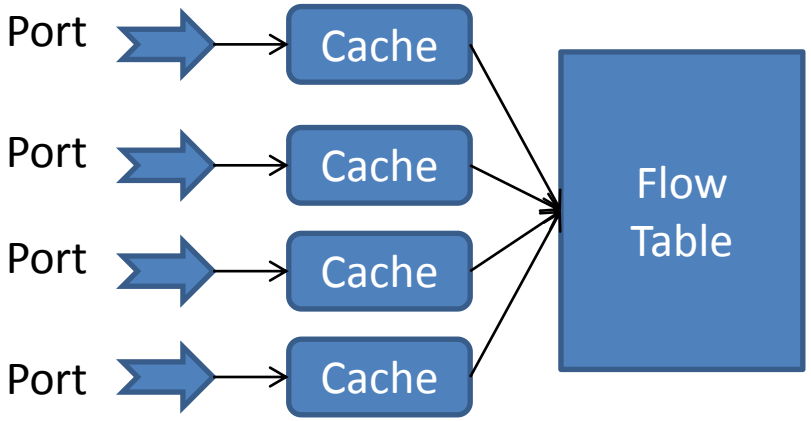
- **Update Counter**
- **Update Match Fields**
- **Update Metadata**
- **Update Action Set**
- **Goto Next Stage**
- **Send to Controller**
- **Drop**

Basic Idea

- Cache in Computer Architecture Research
  - Exploit Temporal and Spatial Localities in Memory Accesses
- For Flow Table Cache
  - Spatial locality
    - A flow mostly goes through same physical route
    - The centralized controller could so set up
  - Temporal locality
    - A flow consists of sequentially many packets
  - Issues
    - Centralized vs Distributed vs Hybrid
    - Size, Organization, Replacement Policy

Centralized Cache vs Distributed Cache





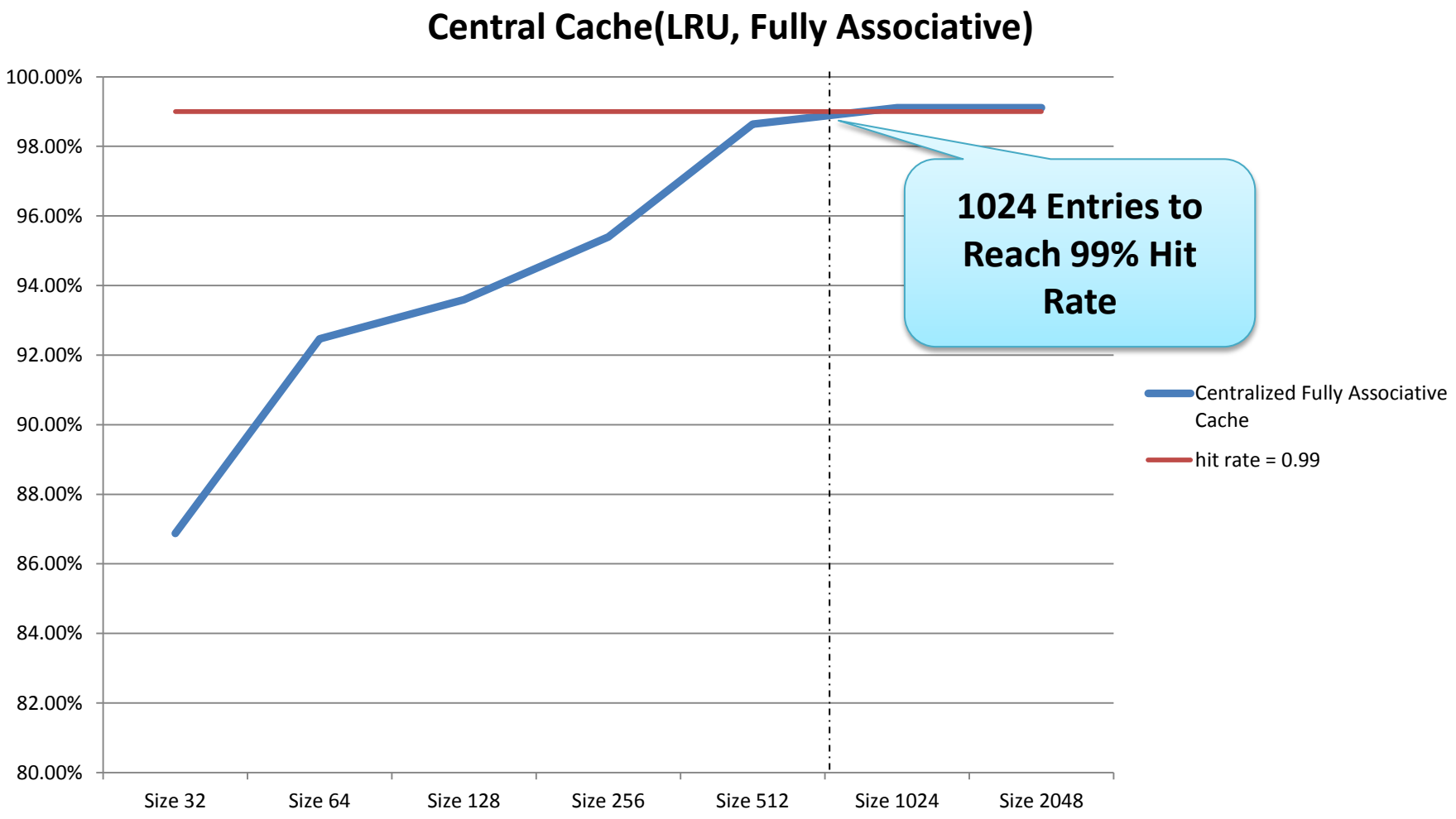
Experiment Setup I (Artificial Traffic)

- 24-port OpenFlow Switch Simulator (Open vSwitch 2.1.0)
- Mininet 2.0.0 SDN Simulator
- Computing Platform
  - AMD A10-7850k APU with Radeon R7 graphics
  - Memory: 16GB
  - OS (Ubuntu 13.10)
- Traffic Characteristics
  - Next Page

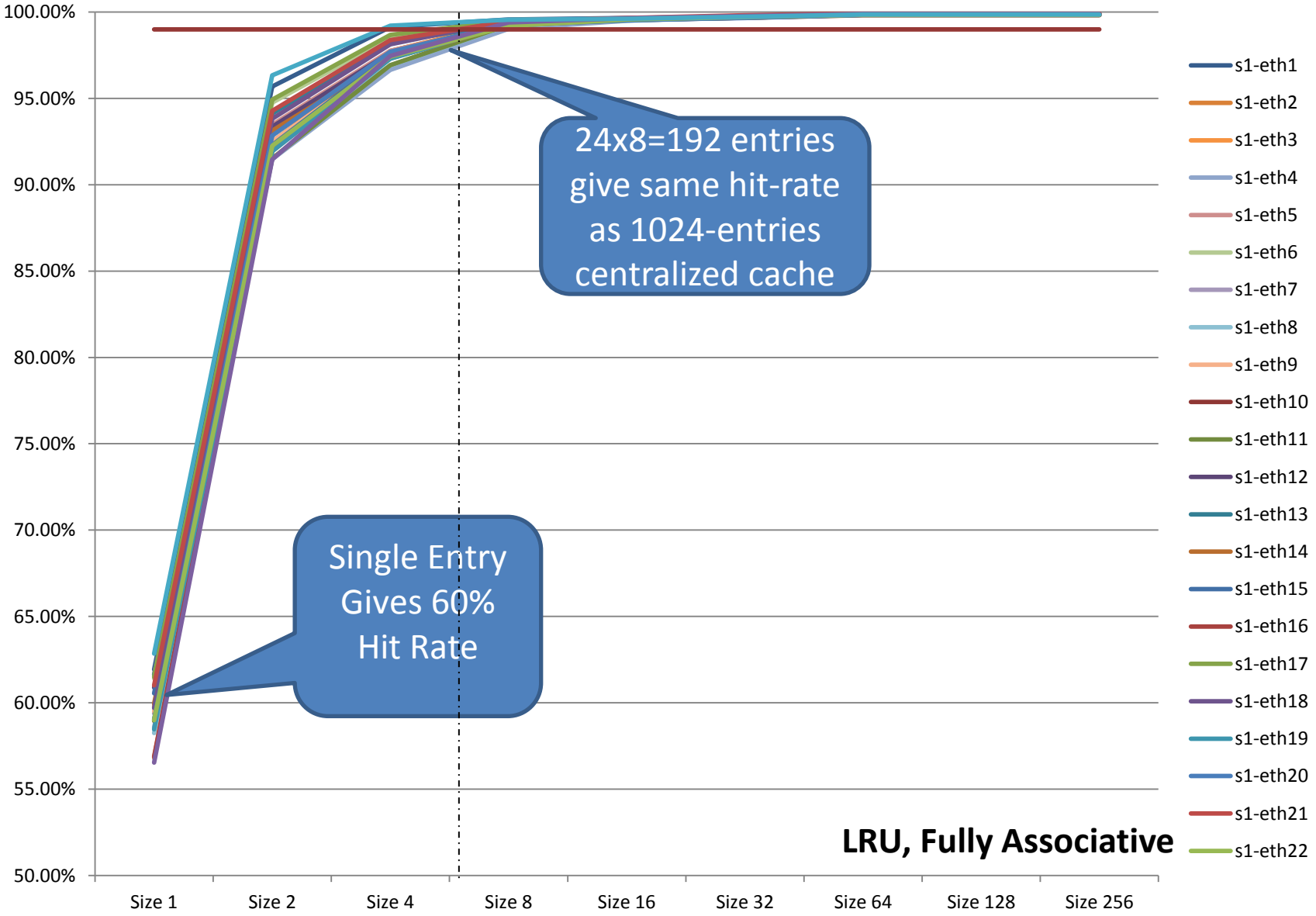
Artificial Traffic Characteristics

- Iperf Traffic Generator
  - TPC/UDP packets
  - Time interval between successive requests: Gamma Distribution(α = 2 ß = 3)
    - $y = \frac{x^{\alpha-1}e^{\frac{x}{\beta}}}{\Gamma(\alpha)\beta^{\alpha}}$
  - Number of packets: Normal Distribution
    - Mean = 7.5, Standard Deviation= 1.5
  - Packet size: Beta Distribution
    - $y = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$, $0 < x < 1$

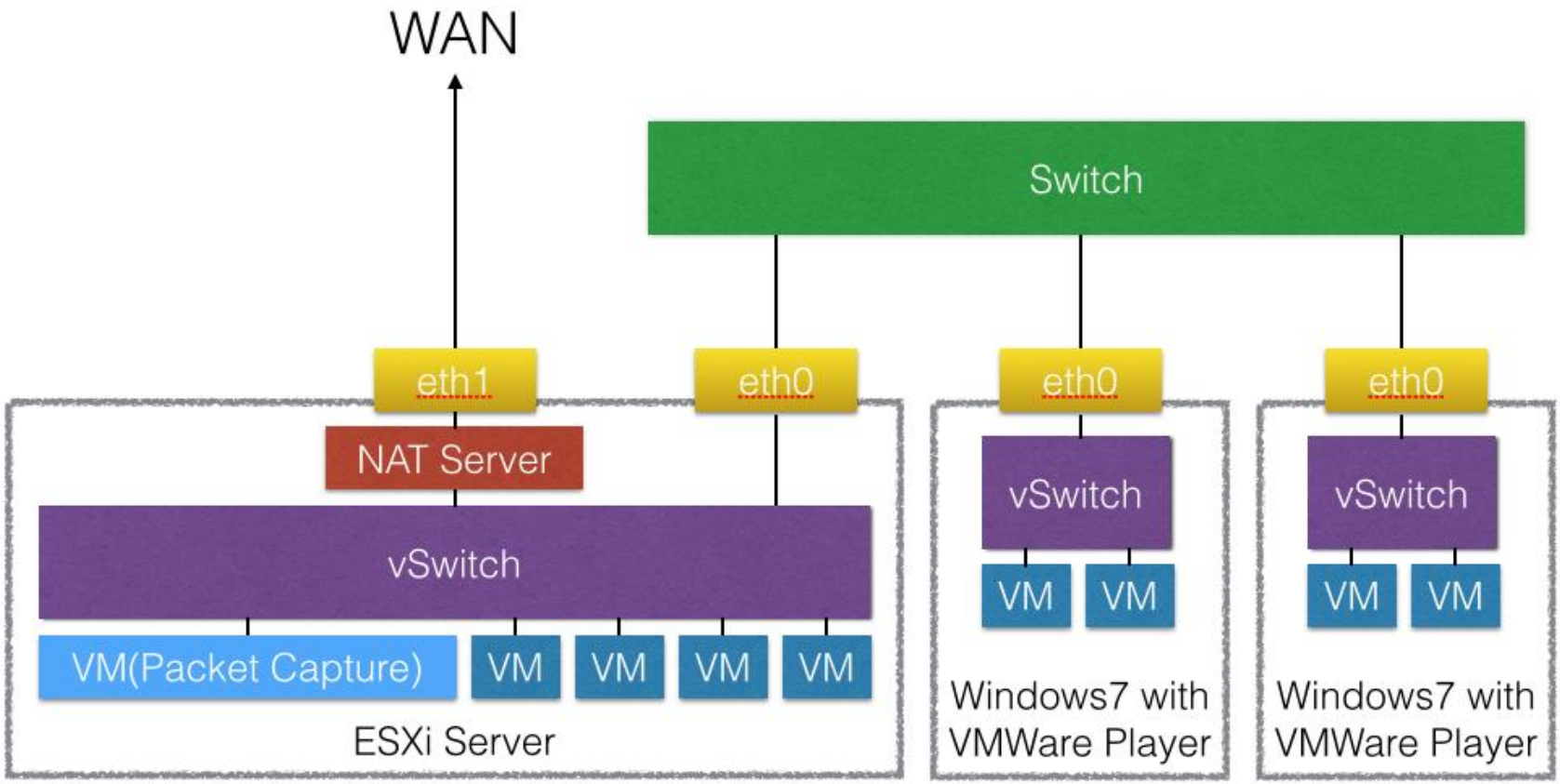
Hit Rate vs Cache Size (Centralized)



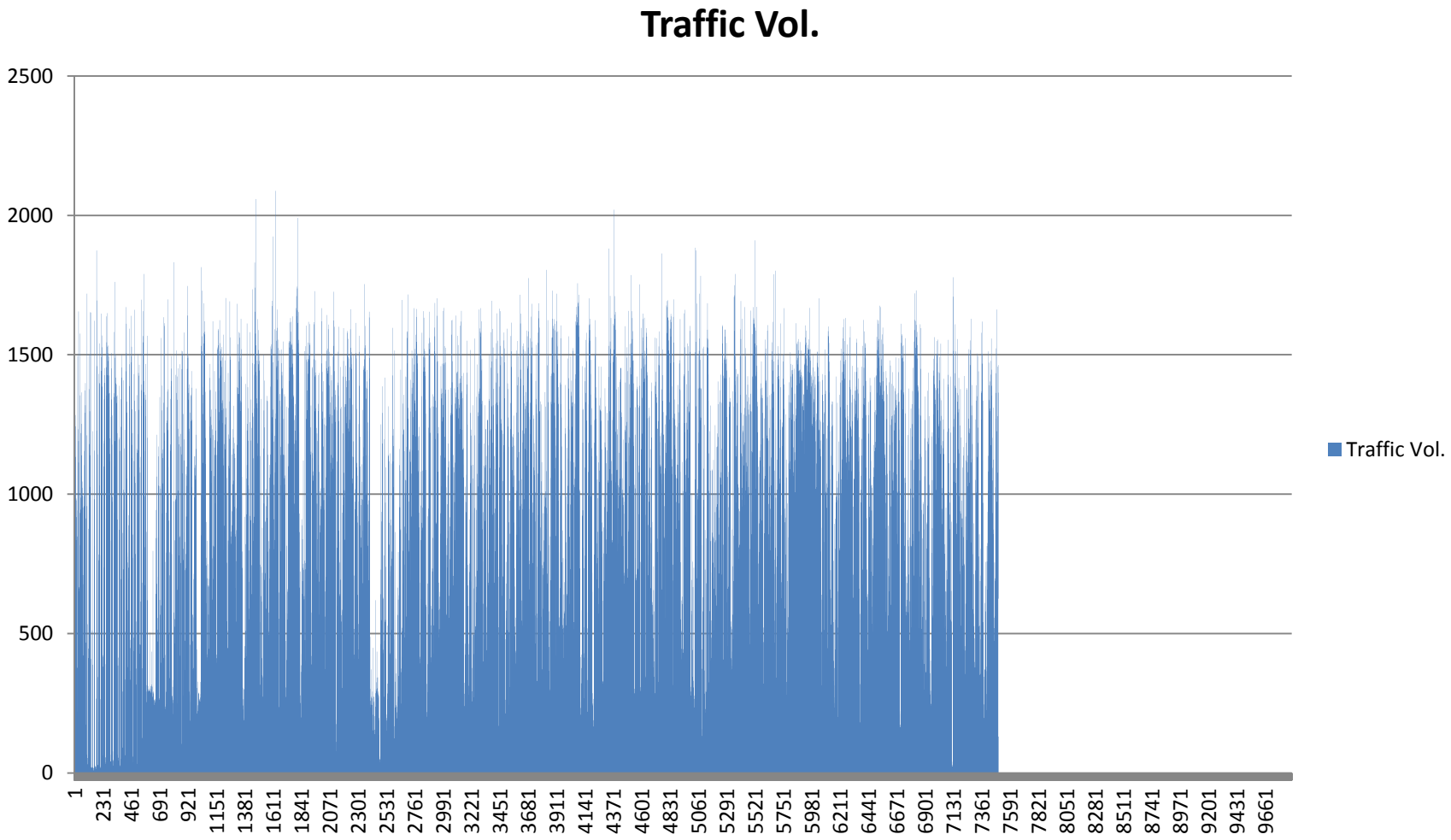
Hit Rate vs Cache Size (Distributed)



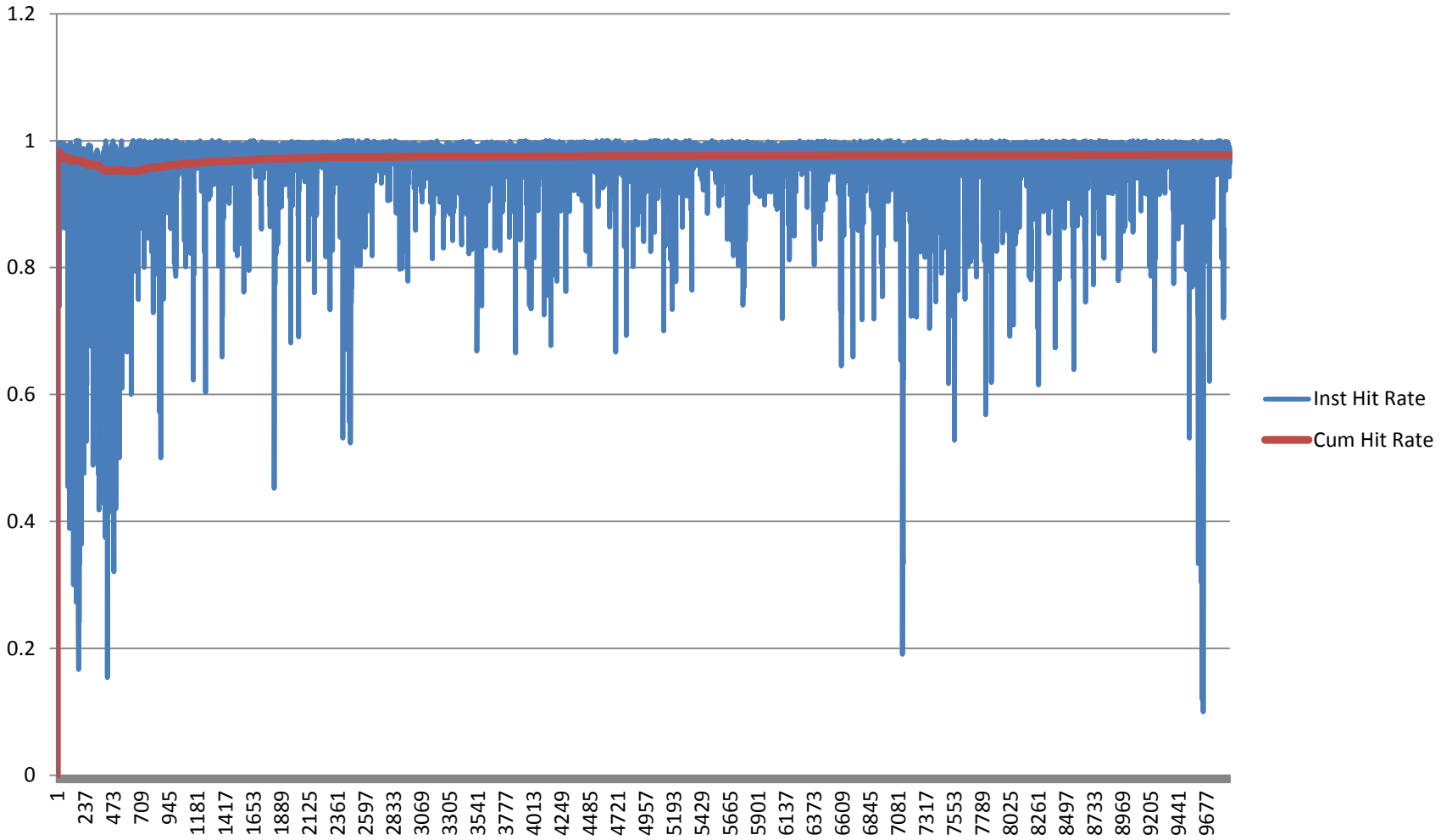
Experiment Setup II (1+8-Port Real Traffic)



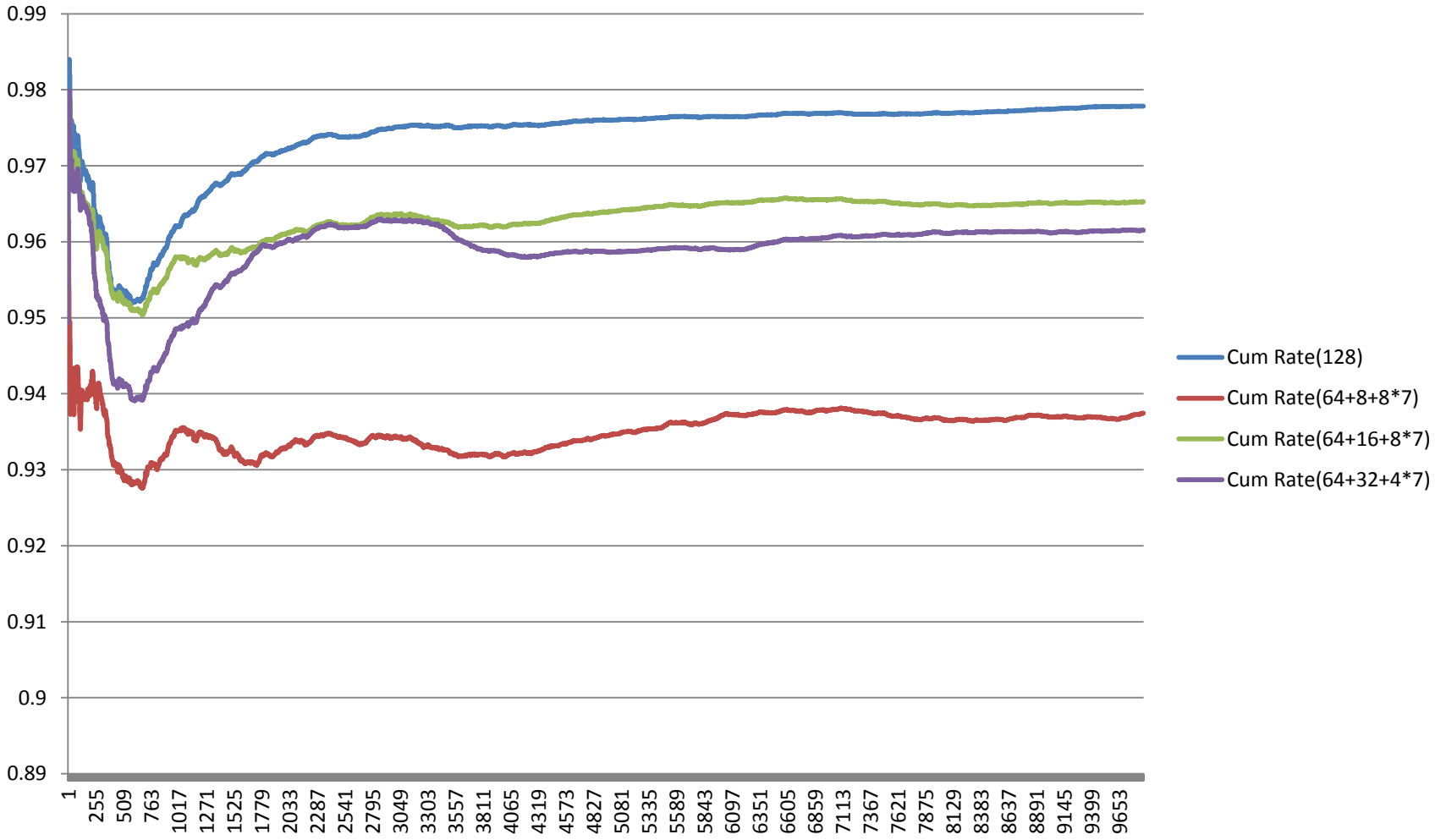
Captured Traffic



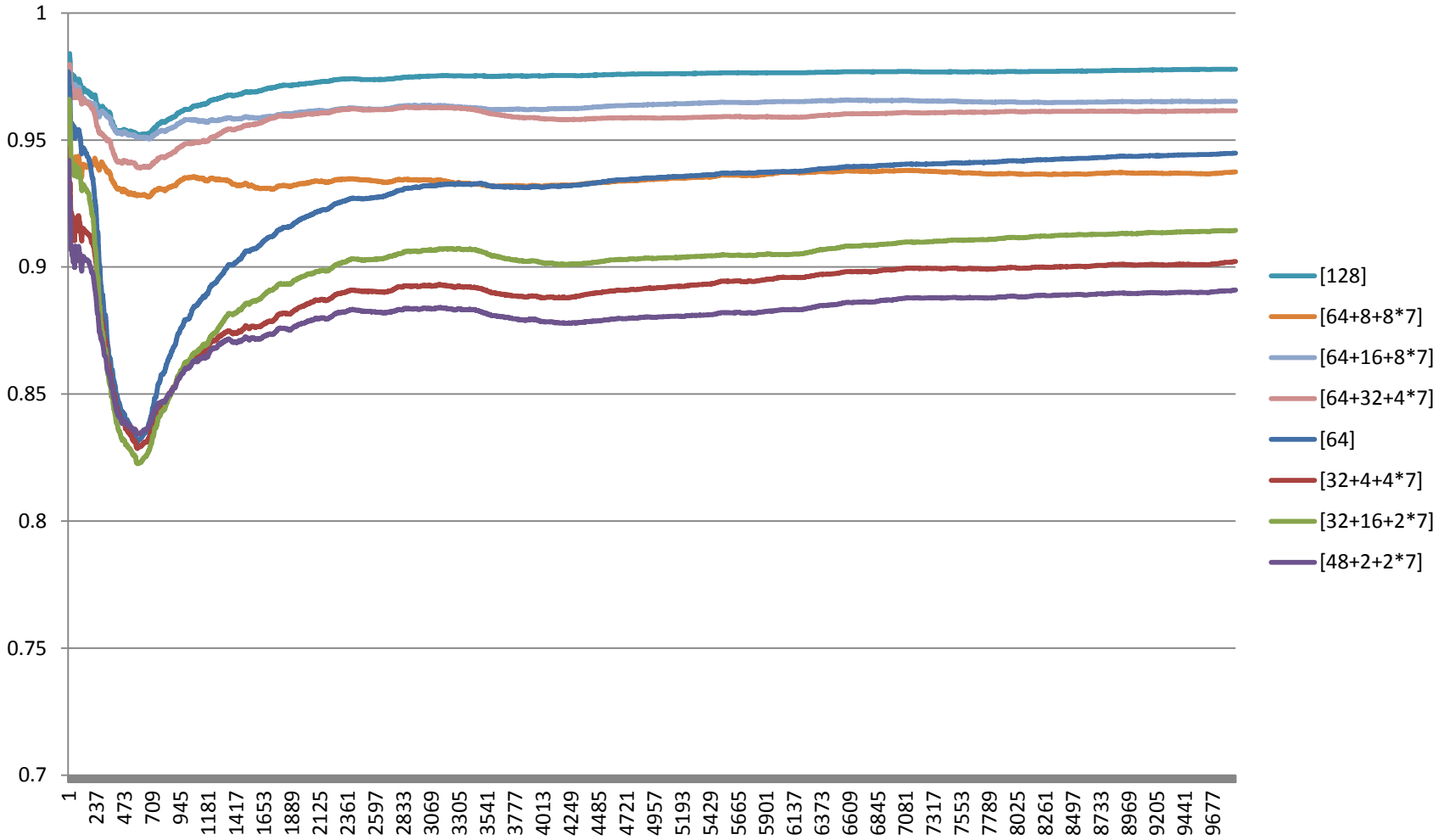
Instant and Cumulative Hit Rate



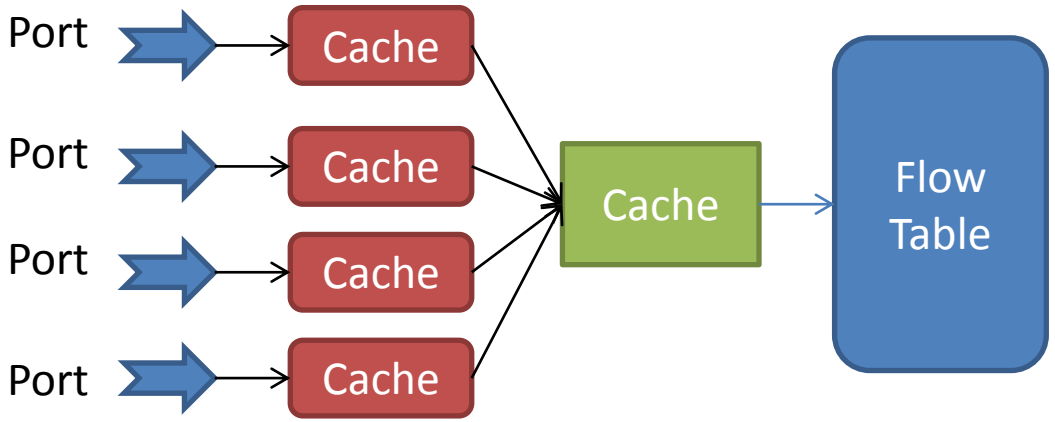
Hit Rate vs Cache Configurations (Size 128)

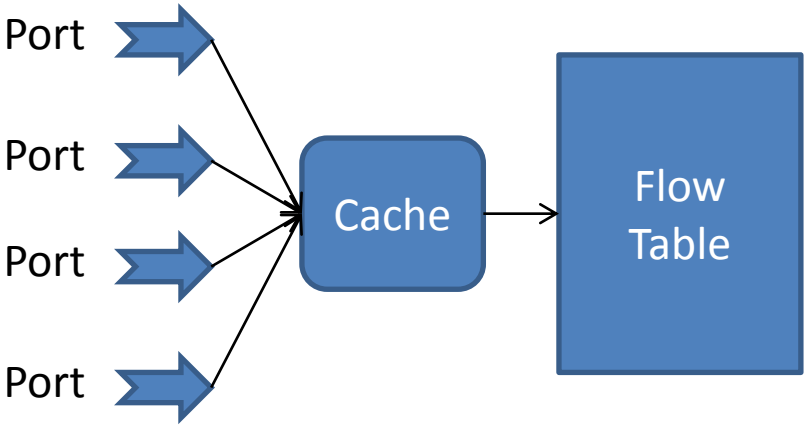


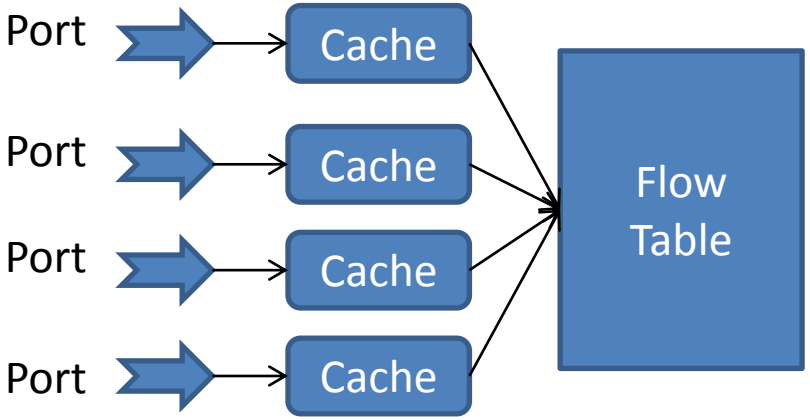
Four Configurations Each of Sizes 128 & 64



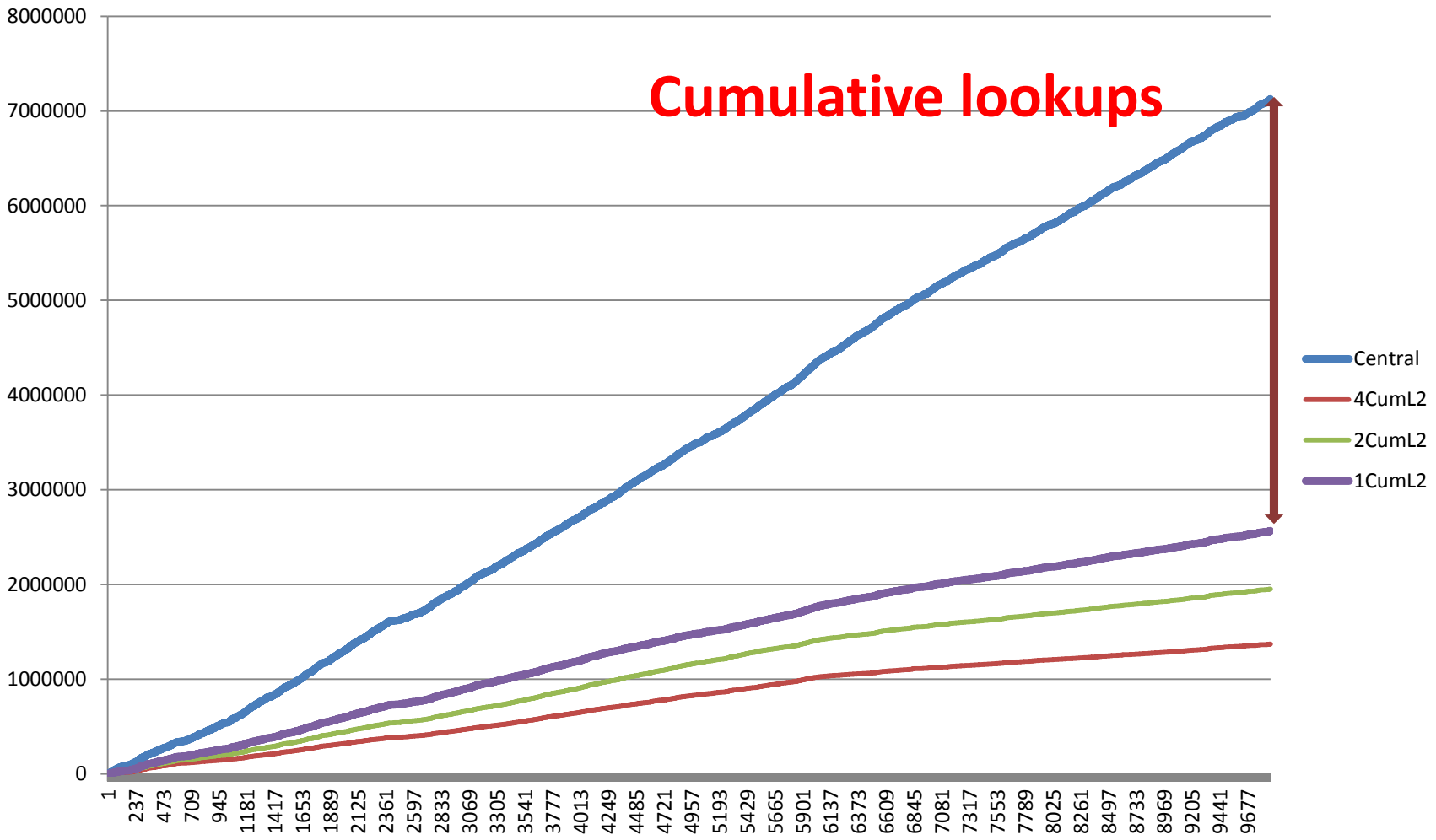
Hybrid Centralized&Per-Port Organization
(Kind of L1-L2 Cache)



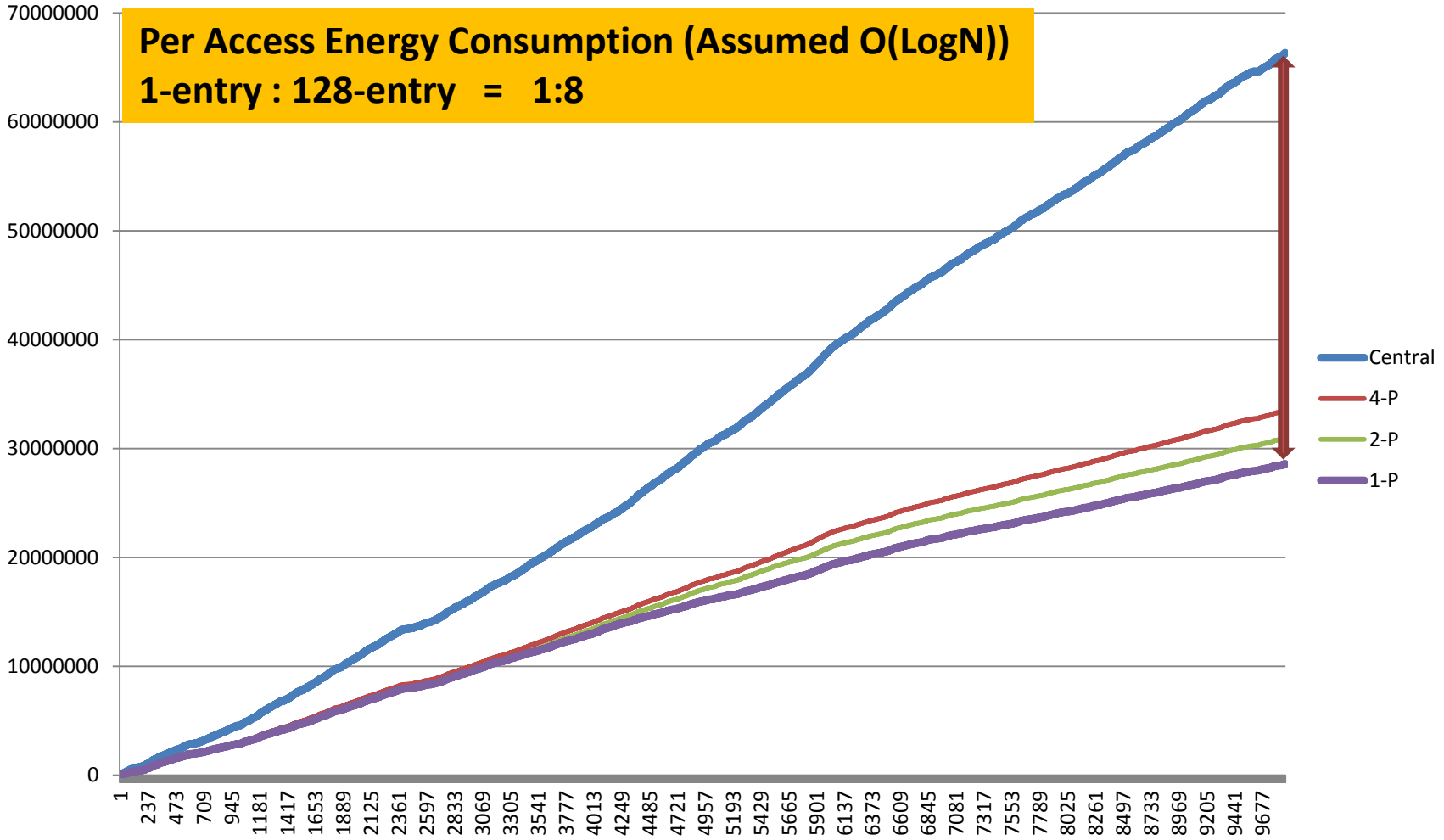




1-entry per-Port Cache Offload 65% Lookups



60% Reduction in Memory Energy Consumption

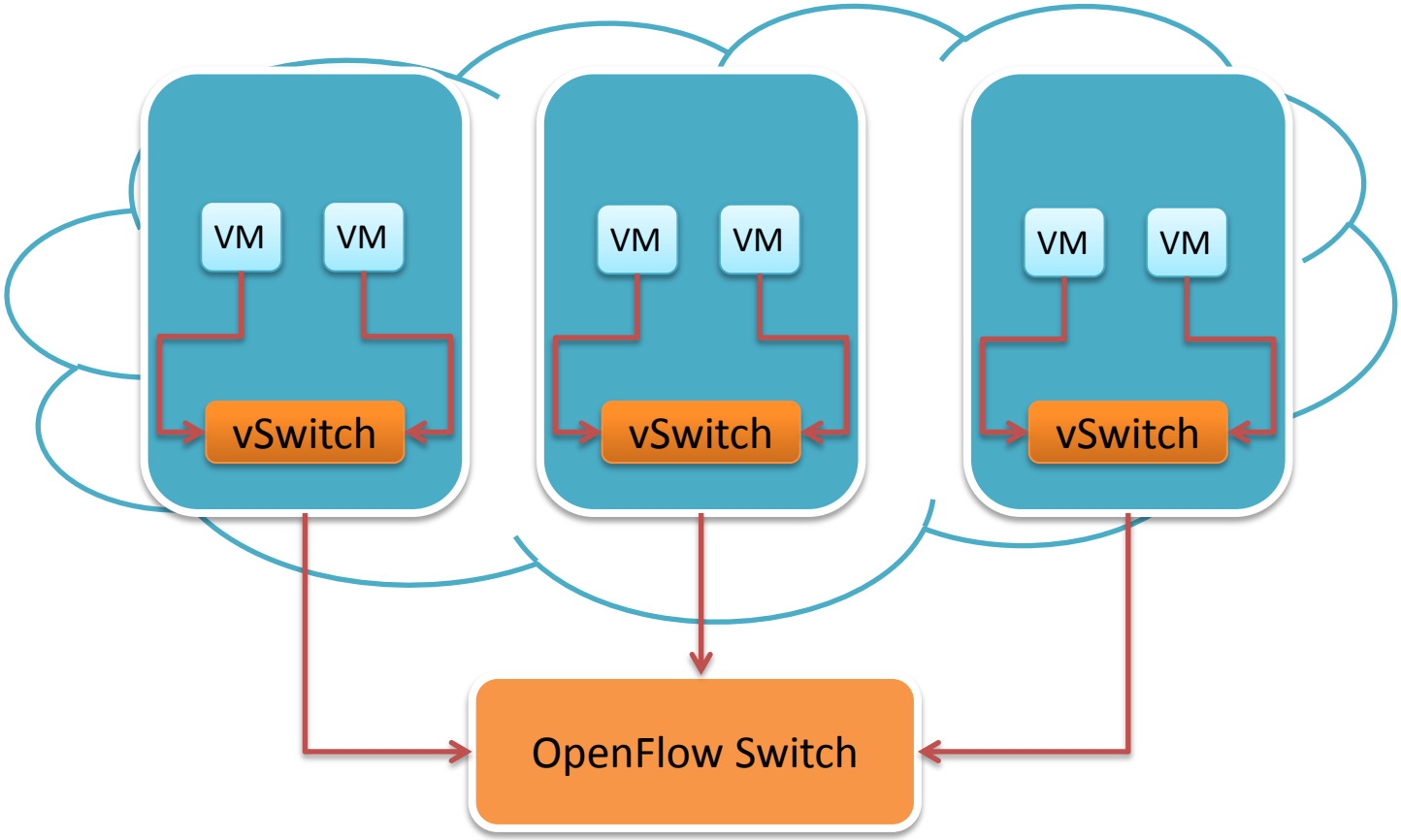


Conclusions and Future Works

- Distributed Per-Port Cache Shown Very Promising for OpenFlow Switch Design
- 192 Entries vs 1024 Entries for Same 99% Hit Rate
- 1-Entry Per-Port Cache Offloads 65% of Traffic and Saves 60% Memory Energy Consumption
- Future Directions
  - Large Scale (Campus-Wide)Traffic
  - Adaptive Cache Organizations and Replacement Policies
  - Speeding Up Software OpenFlow Switches with Multicores for Multiple Virtual Machines

Thank you for your attention!!

Implementing Software Switch with Multicores



Thank you for your attention!!