

Using CPU as a Traffic Co-processing Unit in Commodity Switches

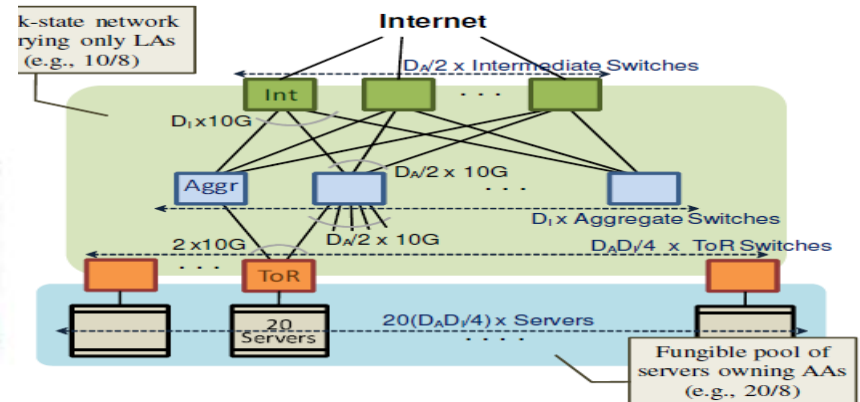
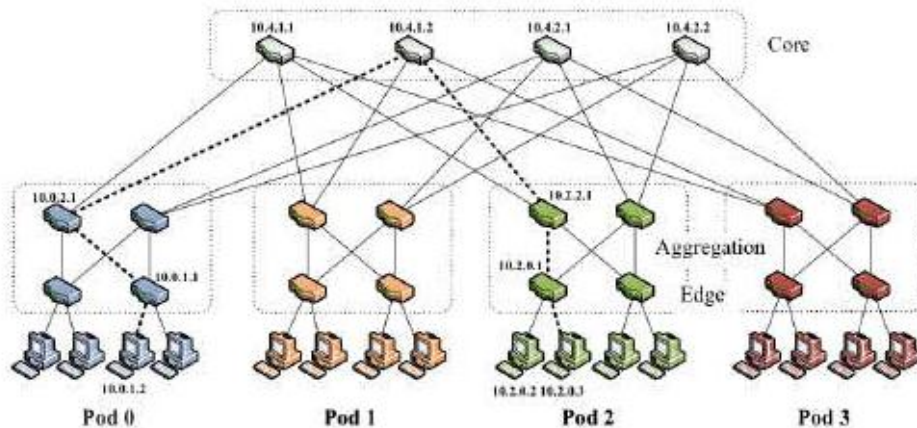
*Guohan Lu, Rui Miao⁺, Yongqiang Xiong
and Chuanxiong Guo*

Microsoft Research Asia

⁺Tsinghua University

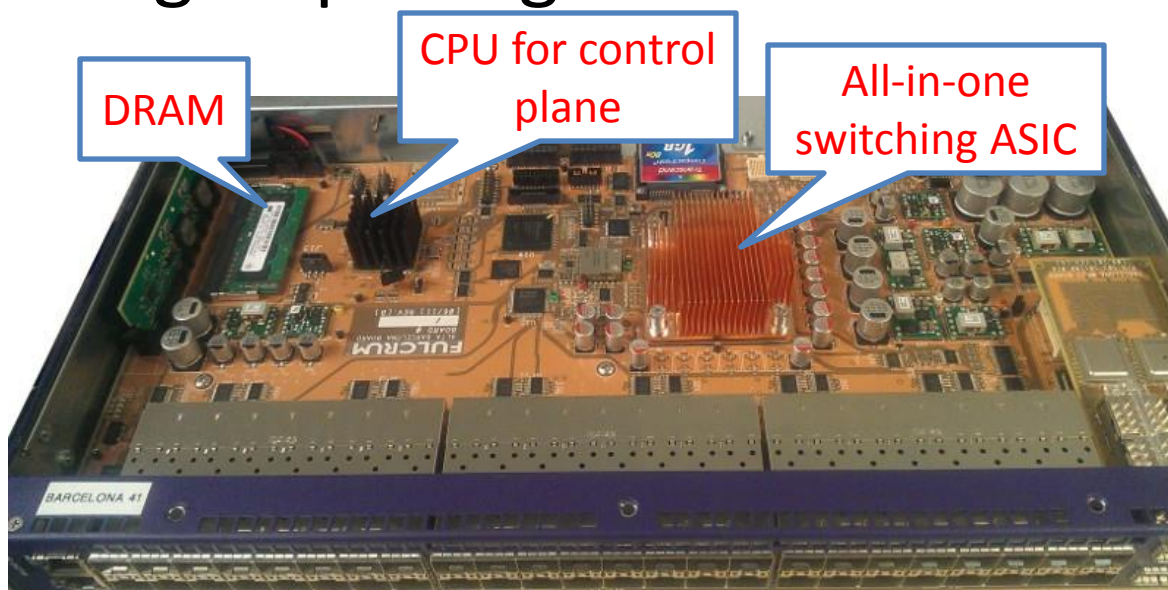
Background

- Commodity switches are the basic building blocks in enterprise and data center networks
 - PortLand and VL2 build entire DCN with 1U commodity switches



Background (cont')

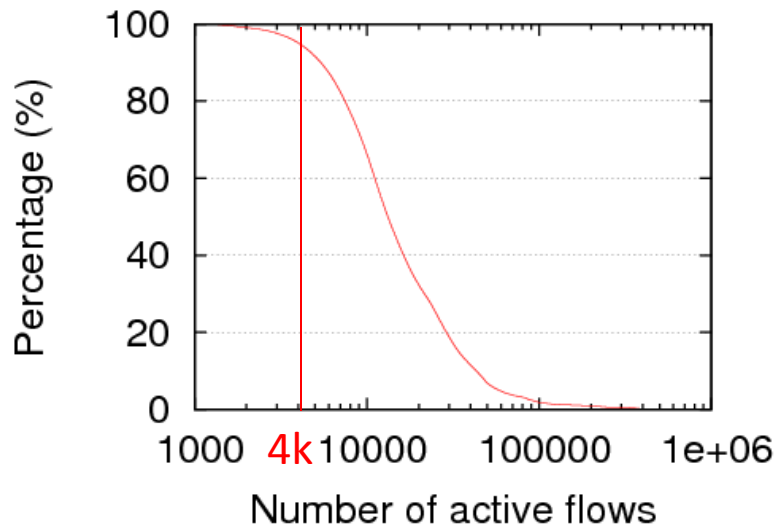
- Commodity switches now widely adopt single switching chip design



- Greatly simplifies switch design and lowers down the cost

Limitation (I)

- Limited forwarding table size for flow-based forwarding schemes, e.g. Openflow
 - Openflow provides finest granularity for better security (Ethane), traffic load balancing (Hedera), Energy saving (ElasticTree)
 - 4k flow entries for most recent BRCM switching chip

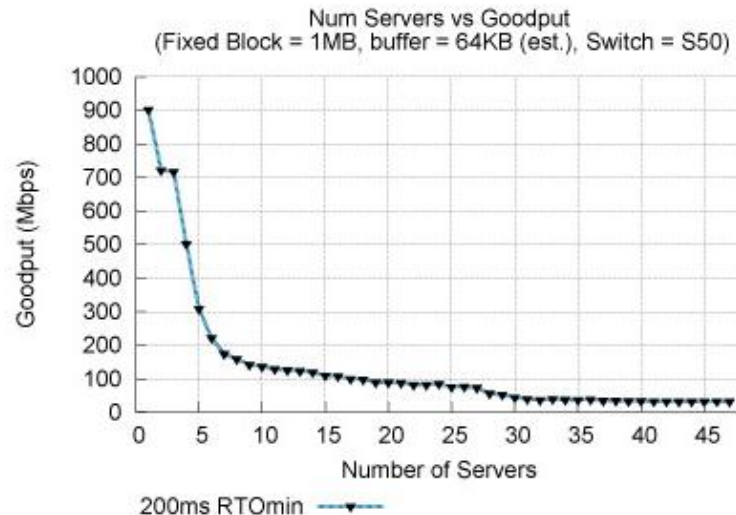
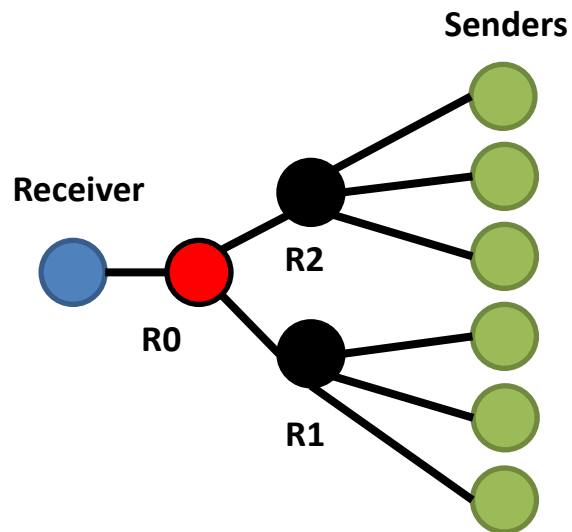


Data center for map-reduce style applications with 120 ToR and ~5k servers

of active flows \geq 4096 for 95%+ time

Limitation (II)

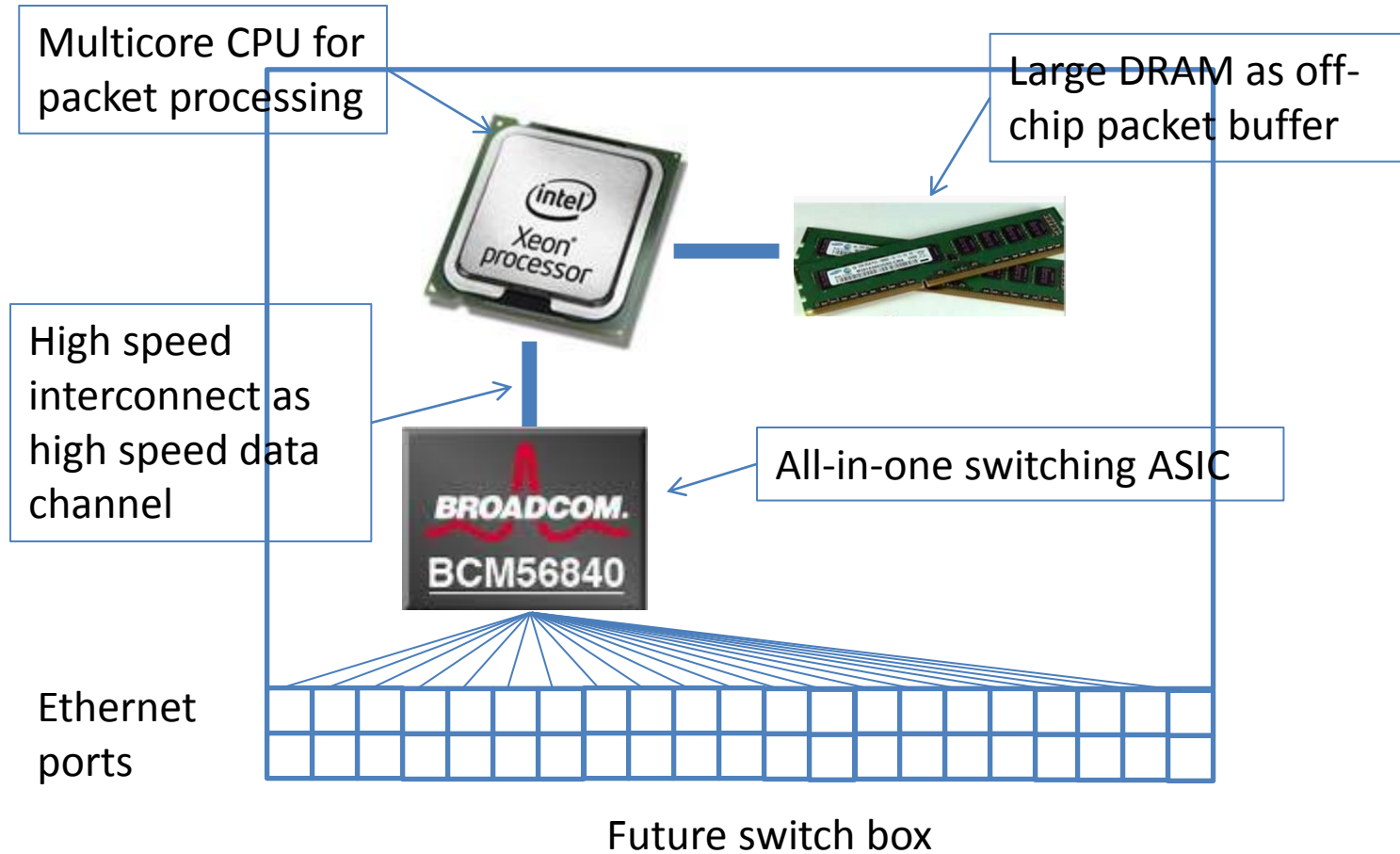
- Shallow packet buffer for bursty traffic
 - Switching ASIC has only several MB buffer
 - Bursty traffic pattern, e.g. TCP incast, TCP flash crowds
 - Packet drops lead to degraded network performance



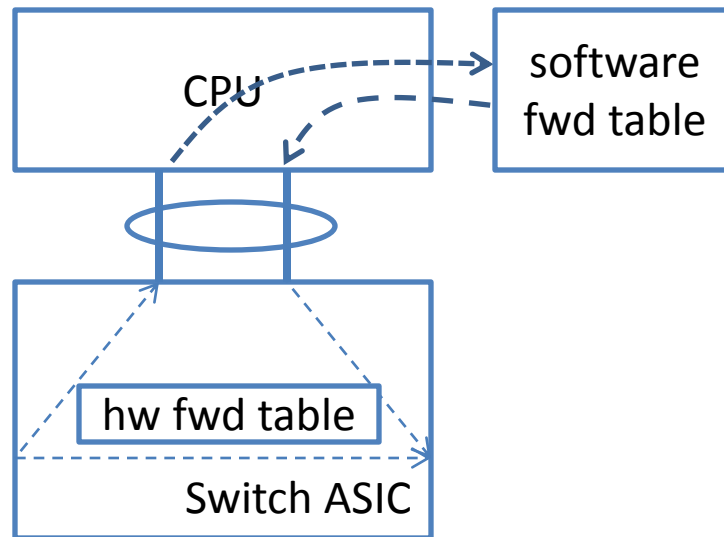
Design Goals

- Large forwarding table
 - Support large forwarding table for forwarding schemes such as OpenFlow
- Deep packet buffer
 - Absorb temporary traffic bursts, e.g., TCP incast, TCP flash crowds

Assumptions for Commodity Switches

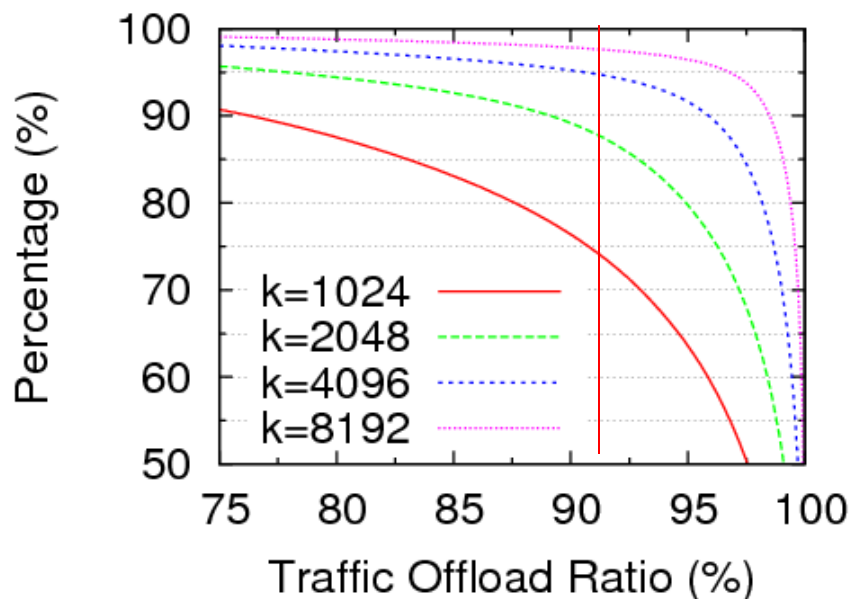


Large forwarding table



- Complete forwarding table in software
- Partial forwarding table in hardware

Traffic Offloading Ratio (TFOR)

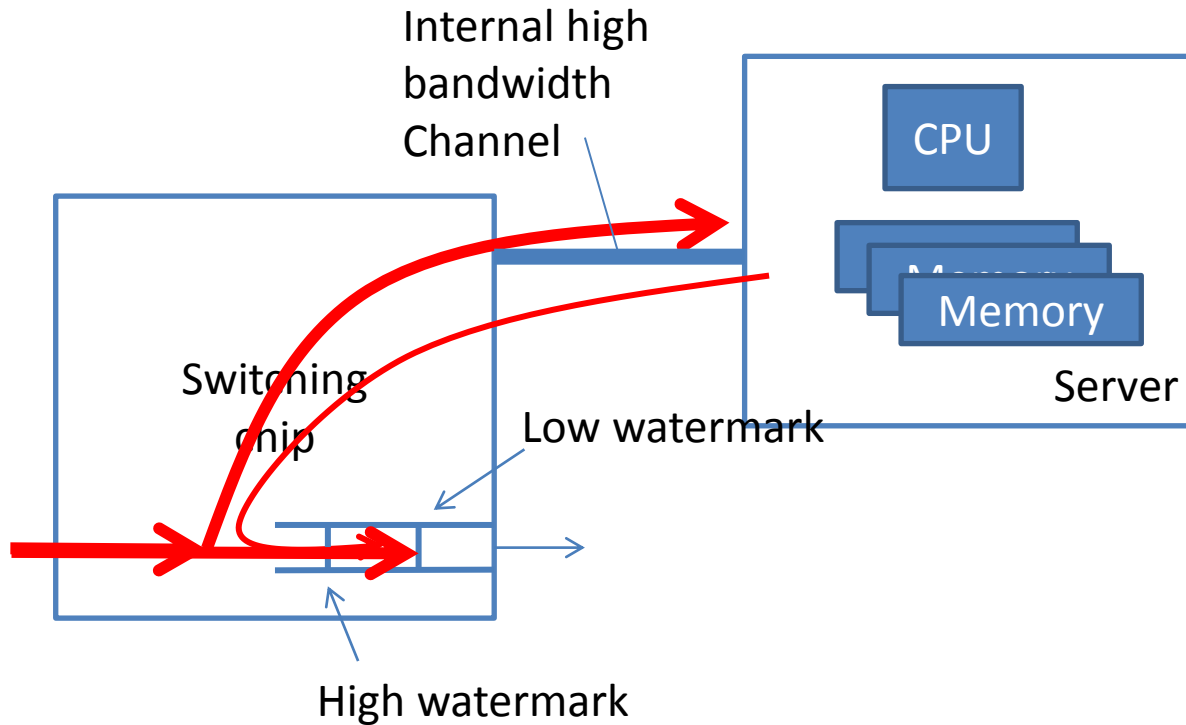


- TFOR: Traffic forwarded by HW v.s. all traffic
- Obtain TFOR: For every minute, get flow rates, sort the flows based on the rates, put k fastest flows in HW.
- **TFOR $\geq 92\%$ for 95%+ time when $k = 4096$**

Flow Management

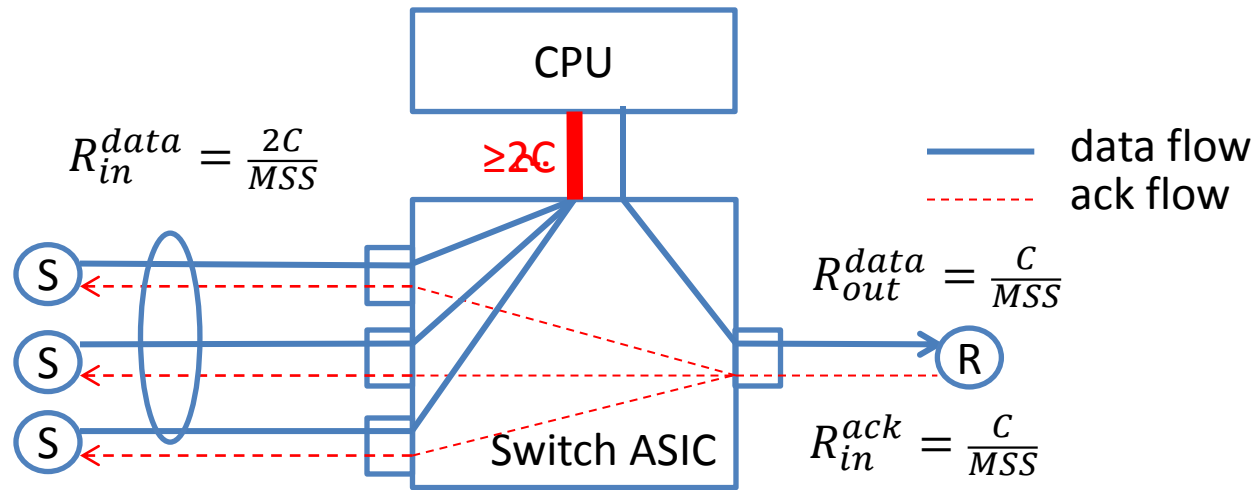
- k fastest flows are forwarded by hardware, rest are forwarded by software
- Assume one byte counter per flow in hardware
- Procedures
 - Count software-forwarded flow bytes, periodically read the counters from hardware
 - Rank flows based on their rates and determine k fastest flows
 - Offload fast flows to hardware and onload slow flows to software

Deep Packet Buffer



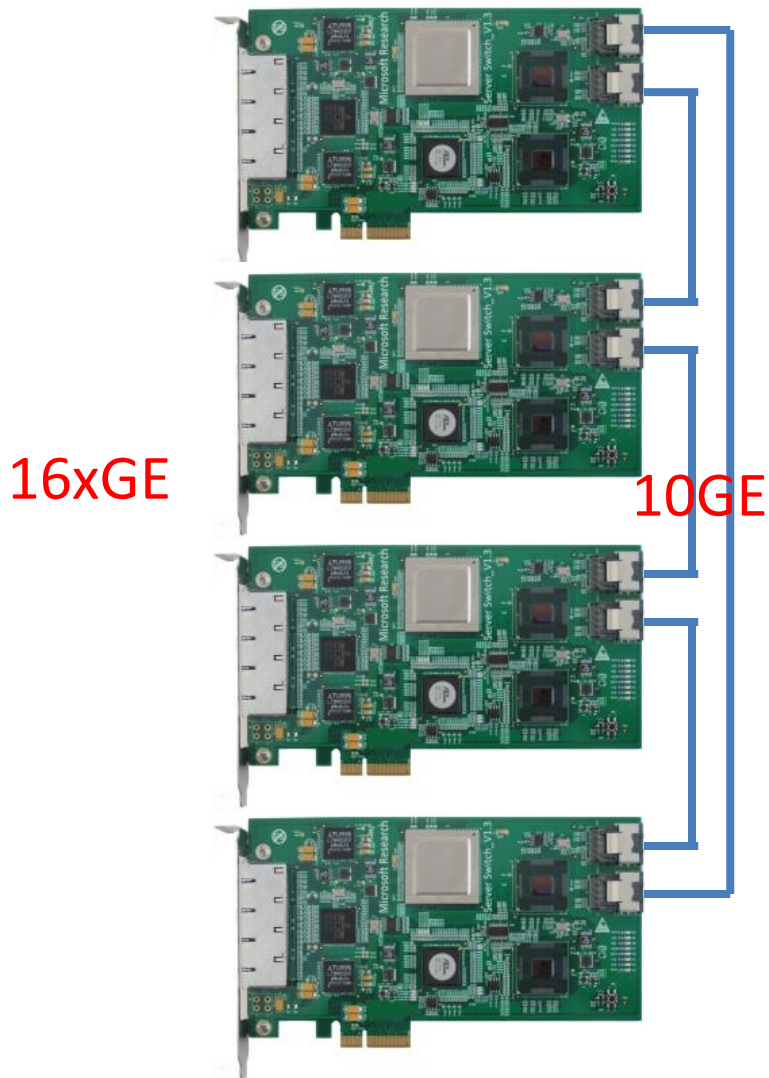
- Phase 1: Traffic redirection
- Phase 2: Cancel redirection

Internal bandwidth Needed



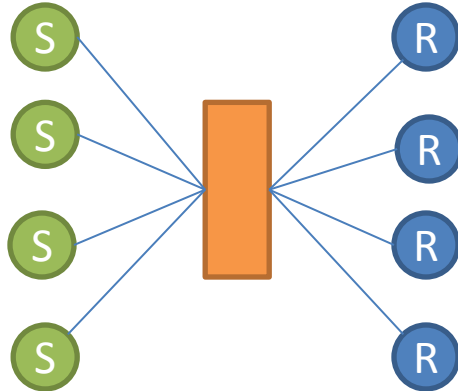
- Receiver: delayed ack disabled
- Senders: TCP slow start
- No packet drops when internal bandwidth is larger than $2C$.

Prototype



- A 16xGE port switch using 4 ServerSwitch cards
- HP z800 workstation
 - 8 CPU cores
 - 48GB DRAM
- Kernel code for packet forwarding
- User space code for switch ASIC management

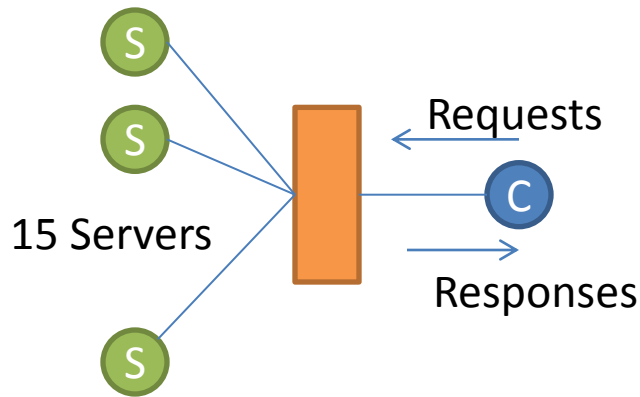
Large Forwarding Table



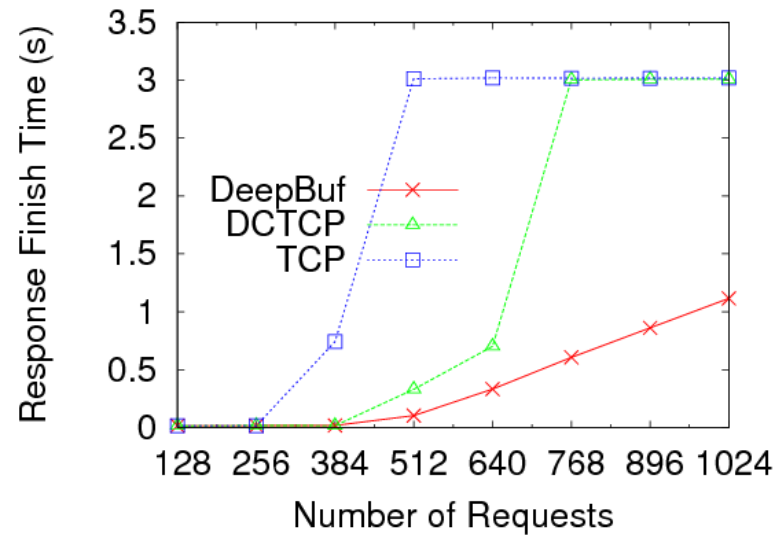
- 10 min synthesized traffic using flow size distribution from DCN measurements
- 1,792 HW flow entries

Interval ratio	Total bytes (GB)	# of active flows	TFOR
1x	33.6	10,644	96.1%
1/10x	336	106,544	90.5%

Deep Packet Buffer



TCP Flash Crowds last for 1 second



1024 Requests	SYN/ACK timeout	Data timeout	Fast Recovery	Packet drops
TCP	109	180	690	15962
DCTCP	23	395	173	3302
DeepBuf	0	0	0	0

Conclusions

- Two major limitations of current commodity switches
 - Limited forwarding table for Openflow
 - Shallow packet buffer for bursty traffic pattern
- Use CPU as traffic co-processor to address these two limitations

QUESTIONS?