ServerSwitch: A Programmable and High Performance Platform for Data Center Networks

Guohan Lu, Chuanxiong Guo, Yulong Li, Zhiqiang Zhou†, Tong Yuan, Haitao Wu, Yongqiang Xiong, Rui Gao, Yongguang Zhang

Microsoft Research Asia

†Tsinghua University

NSDI 2011, Boston, USA

Motivations

- Lots of research and innovations in DCN
  - PortLand, DCell/BCube, CamCube, VL2, …
  - Topology, routing, congestion control, network services, etc.
- Many DCN designs depart from current practices
  - BCube uses self-defined packet header for source routing
  - Portland performs LPM on destination MAC
  - Quantized Congestion Notification (QCN) requires the switches to send explicit congestion notification
- Need a platform to prototype existing and many future DCN designs

Requirements

- Programmable and high-performance packet forwarding engine
  - Wire-speed packet forwarding for various packet sizes
  - Various packet forwarding schemes and formats
- New routing and signaling, flow/congestion control
  - ARP interception (PortLand), adaptive routing (BCube), congestion control (QCN)
- Support new DCN services by enabling in-network packet processing
  - Network cache service (CamCube), Switch-assisted reliable multicast (SideCar)

Existing Approaches

- Existing switches/routers
  - Usually closed system, no programming interface
- OpenFlow
  - Mainly focus on control plane at present
  - Unclear how to support new congestion control mechanisms and in-network data processing
- Software routers
  - Performance not comparable to switching ASIC
- NetFPGA
  - Not commodity devices and difficult to program

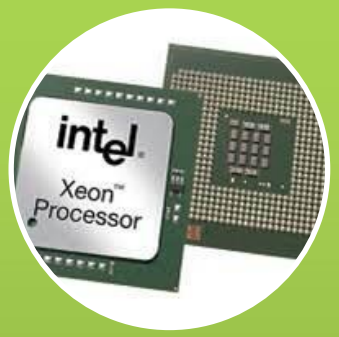
Technology Trends

Modern Switching Chip

- High switching capacity (640Gbps)
- Rich protocol support (Ethernet, IP, MPLS)
- TCAM for advanced packet filtering

PCI-E Interconnect

- High bandwidth (160Gbps)
- Low latency (<1us)



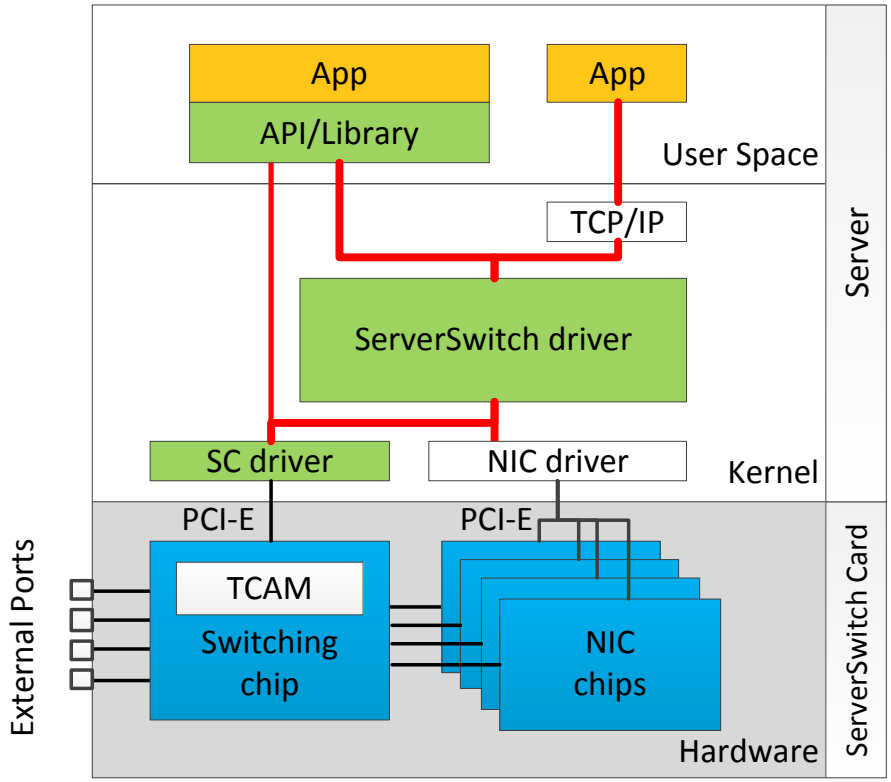
Commodity Server

- Multi-core
- Multi 10GE packet processing capability

Design Goals

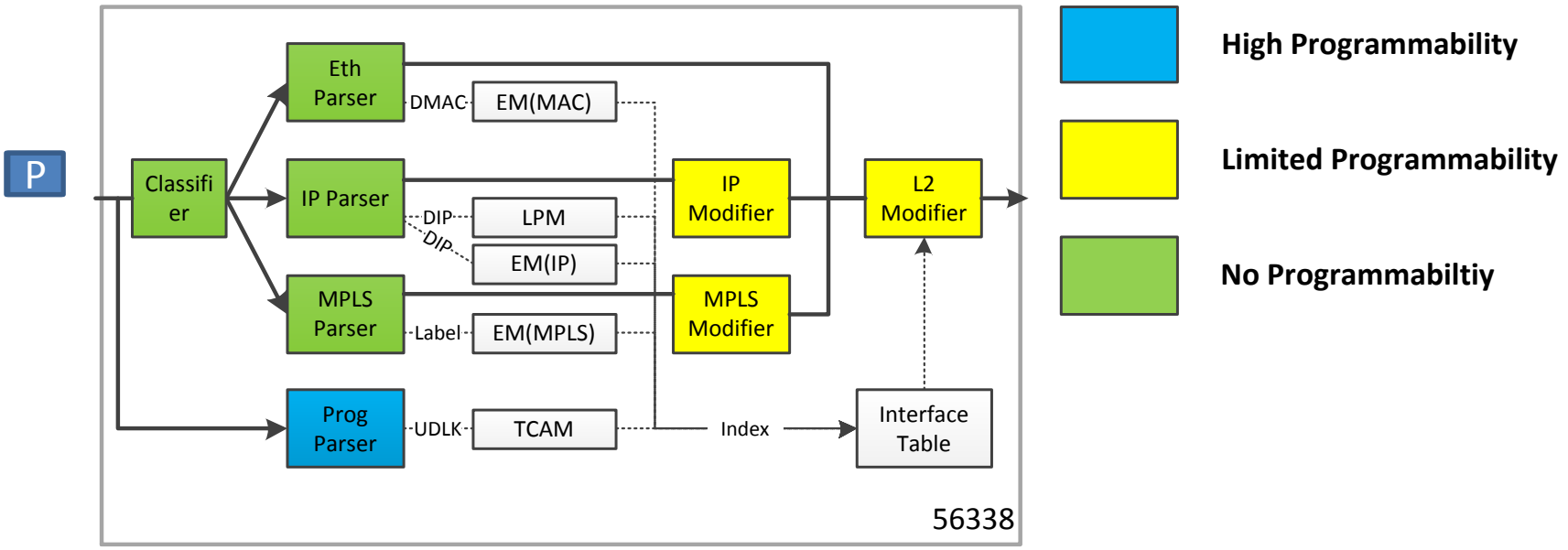
- Programmable packet forwarding engine in silicon
  - Leverage the high capacity and programmability within modern switching chip for packet forwarding
- Low latency software processing for control plane and congestion control messages
  - Leverage the low latency PCI-E interface for latency sensitive schemes
- Software-based in-network packet processing
  - Leverage the rich programmability and high performance provided by modern server

Architecture



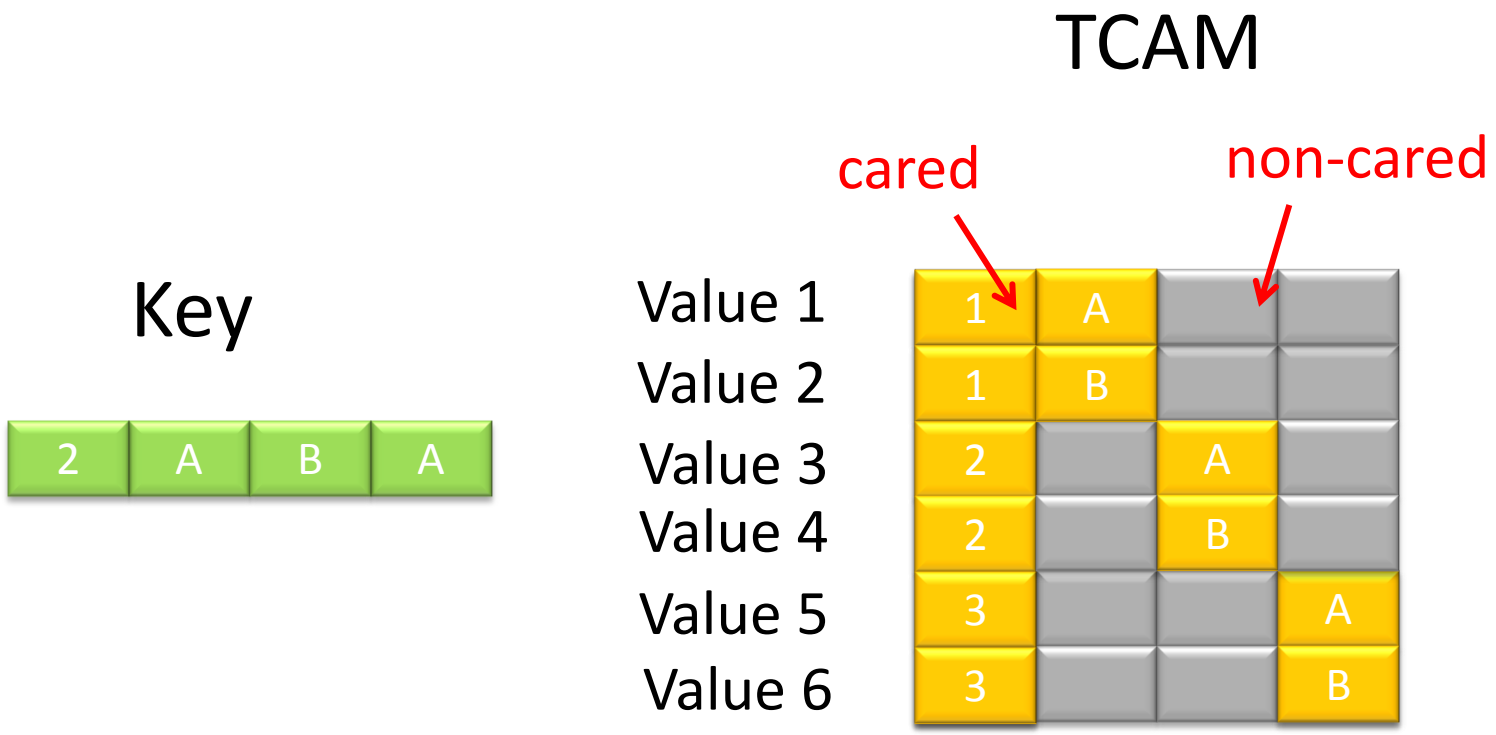
- Hardware
  - Modern Switching chip
  - Multi-core CPU
  - PCI-E interconnect
- Software Stack
  - C APIs for switching chip management
  - Packet Processing in both Kernel and User Space

Programmable Packet Forwarding Engine



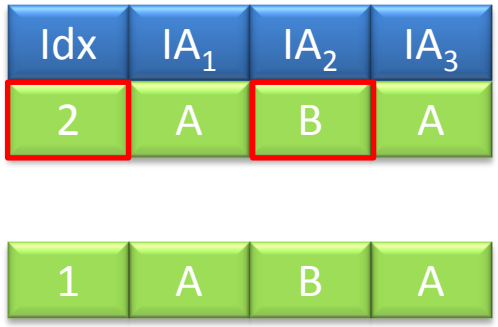
- Destination-based forwarding, *e.g.*, IP, Ethernet
- Tag-based forwarding, *e.g.*, MPLS
- Source Routing based forwarding, *e.g.*, BCube

TCAM Basic

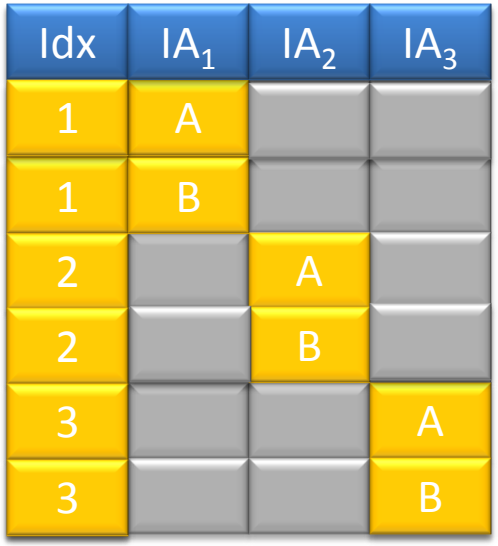

TCAM
cared
non-cared
Key
2
A
B
A
Value 1
Value 2
Value 3
Value 4
Value 5
Value 6
1
A
1
B
2
A
2
B
3
A
3
B

TCAM Based Source Routing

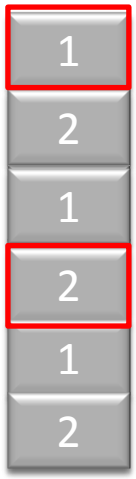
Incoming Packet



TCAM



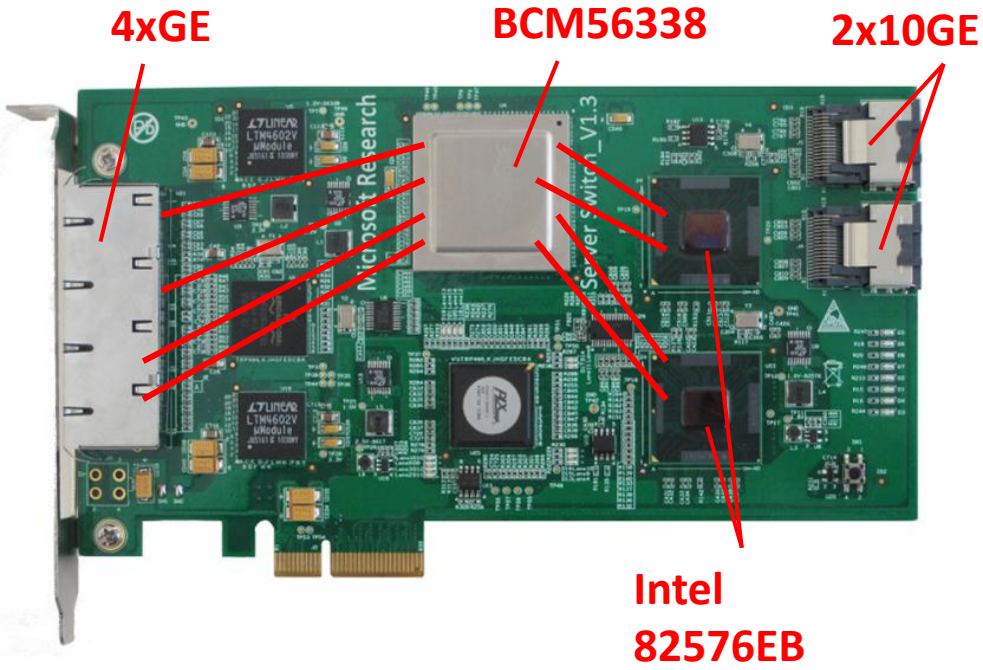
Output Port



ServerSwitch API

- Switching chip management
  - User defined lookup key extraction
  - Forwarding table manipulation
  - Traffic statistics collection
- Examples:
  - `SetUDLK(1, (B0-5))`
  - `SetLookupTable(TCAM, 1, 1, "000201000000", "FFFFFF000000", {act=REDIRECT VIF, vif=3})`
  - `ReadRegister(OUTPUT_QUEUE_BYTES_PORT 0)`

Implementation



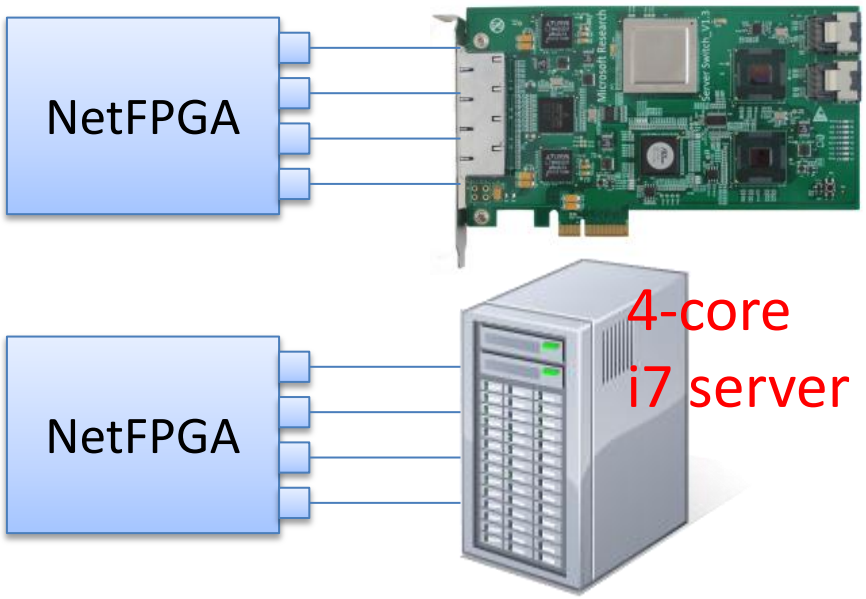
- Hardware
  - 4 GE external ports
  - x4 PCI-E to server
  - 2x10GE board-to-board interconnection
  - Cost: 400$ in 80 pieces
  - Power consumption: 15.7W
- Software
  - Windows Server 2008 R2
  - Switching chip driver (2670 lines of C)
  - NIC driver (binary from Intel)
  - ServerSwitch driver (20719 lines of C)
  - User library (Based on Broadcom SDK)

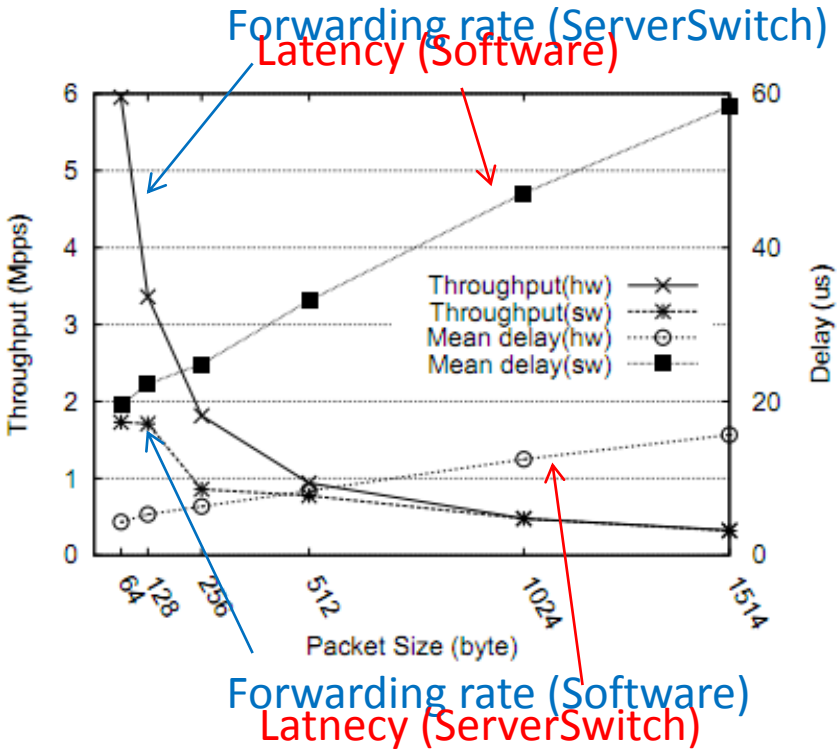
Example 1: BCube

B14-17	Version	HL	Tos	Total length
B18-21	Identification		Flags	Fragment offset
B22-25	TTL	Protocol	Header checksum	
B26-29	Source Address			
B30-33	Destination Address			
B34-37	NHA_1	NHA_2	NHA_3	NHA_4
B38-41	NHA_5	NHA_6	NHA_7	NHA_8
B42-45	BCube Protocol	NH	Pad	

- Self-defined packet header for BCube source routing
- Easy to program: Less than 200 LoC to program the switching chip

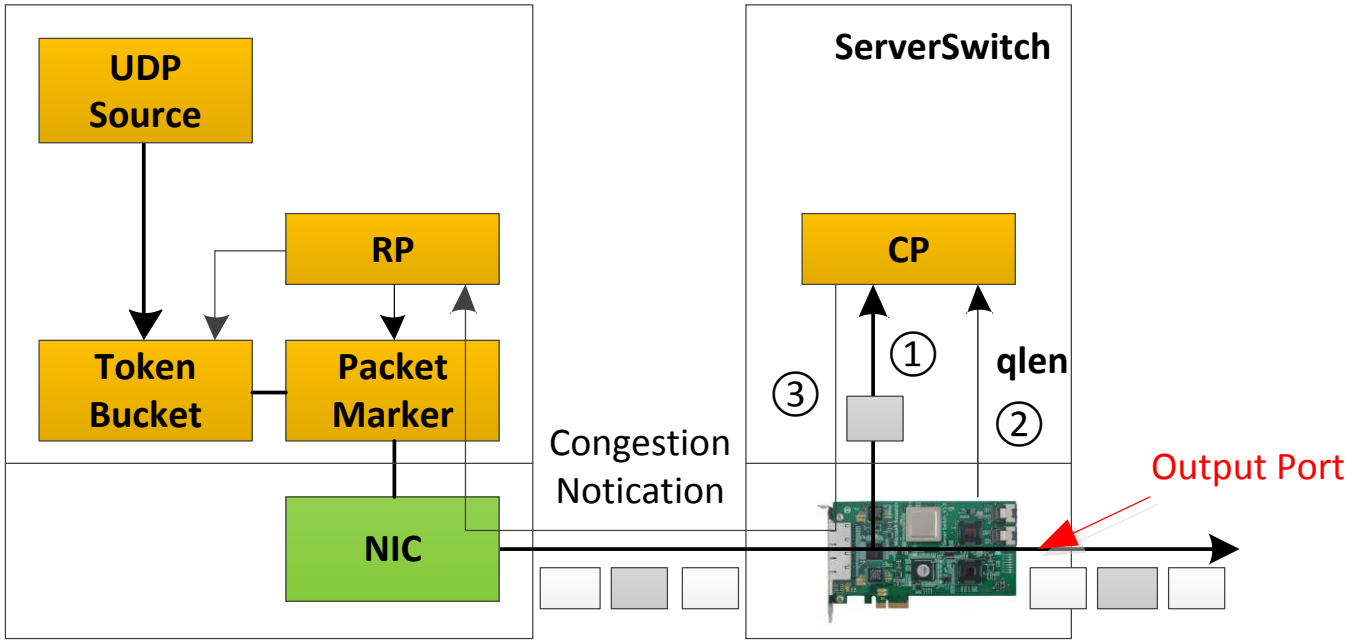
BCube Experiment





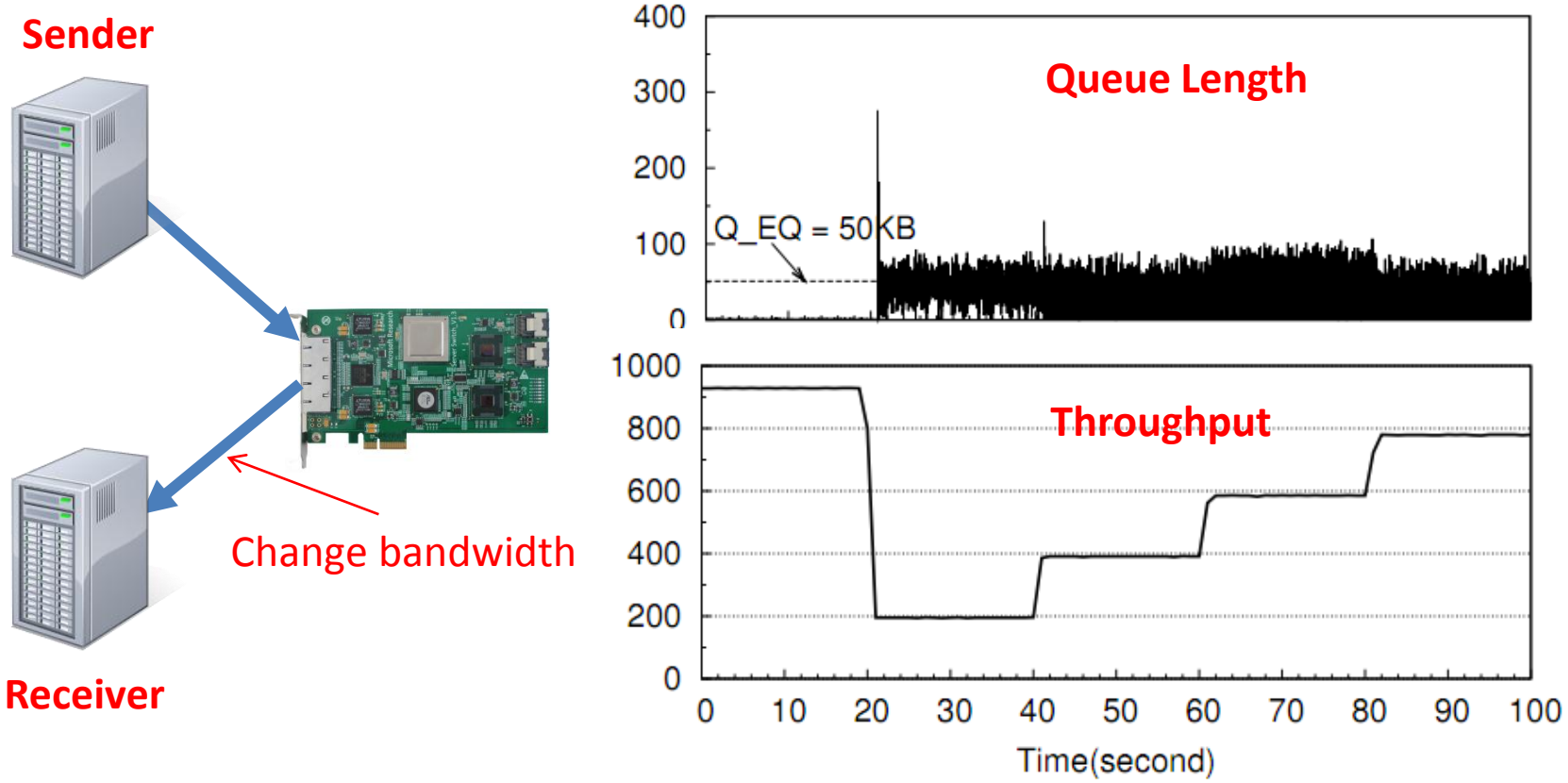
- ServerSwitch: wire-speed packet forwarding for 64B
- ServerSwitch: 15.6us forwarding latency, ~1/3 of software forwarding latency

Example 2: Quantized Congestion Notification



- Congestion notification generation requires very low latency

QCN Experiment



- Queue fluctuates around equilibrium point (Q_EQ)

Limitations

- Only support modifications for standard protocols
  - Ethernet MACs, IP TTL, MPLS label
- Not suitable for low-latency, per-packet processing
  - XCP
- Limited number of ports and port speed
  - Cannot be directly used for fat-tree and VL2
  - 4 ServerSwitch cards form a 16-port ServerSwitch, still viable for prototyping fat-tree and VL2

Summary

- ServerSwitch: integrating a high performance, limited programmable ASIC switching chip with a powerful, fully programmable server
  - Line-rate forwarding performance for various user-defined forwarding schemes
  - Support new signaling and congestion mechanisms
  - Enable in-network data processing
- Ongoing 10GE ServerSwitch

Thanks.

Q&A