# Control Planes for

# Optical Switching

## George Papen

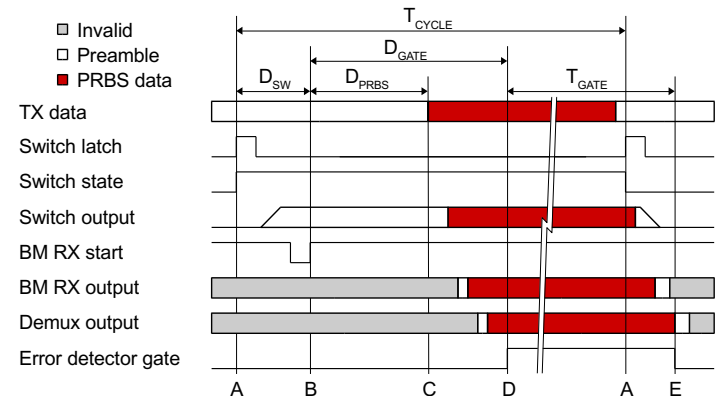## UC San Diego
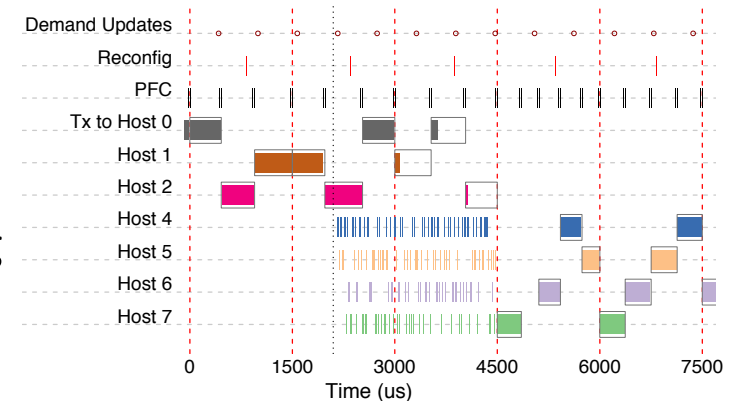
# Practical Optical Switching in Data Centers

- ## *Hardware Issues*
  - Link-level reconfiguration time dominates physical switch time
  - Synchronization is hard
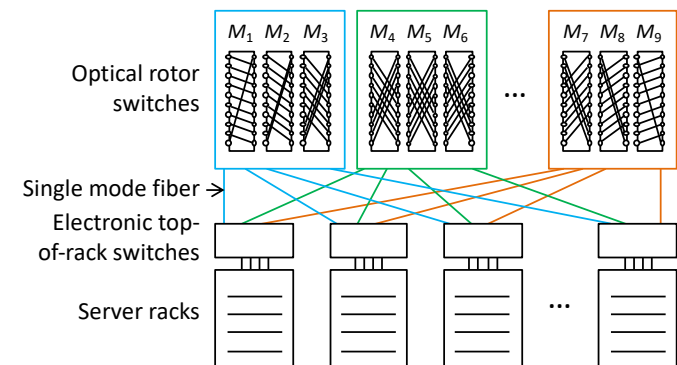  - Links w/burst-mode RXs are more complex

- ## *Software Issues*
  - Centralized scheduling does not scale
  - Software has no firm concept of ``Go now!"
  - Unacceptable delay for μs (or less) switching

- ## *A New Solution  RotorNet*
  - Decouple scheduling and routing
  - Decentralized control plane
  - Hide delay w/parallelism

# Hardware Issues: Reconfiguration Time

- Research at IBM w/UCSD intern using nanosecond Si-P switch

- Goal: minimize link-level reconfiguration time of the switch
  (Time during which data cannot reliably transit the network)

- Includes:
  - Switch reconfiguration time
  - Clock Data Recovery (CDR) locking time
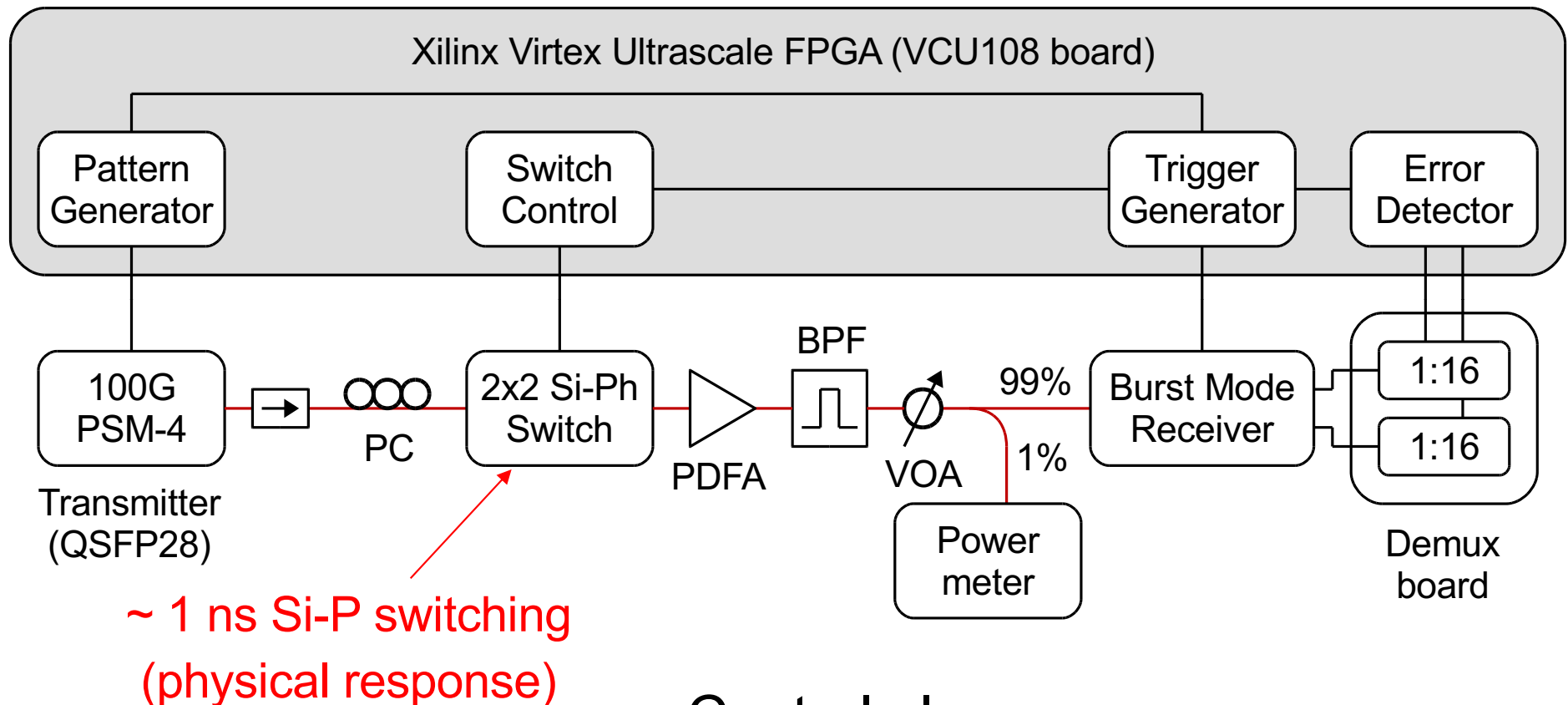  - Synchronization guard delays
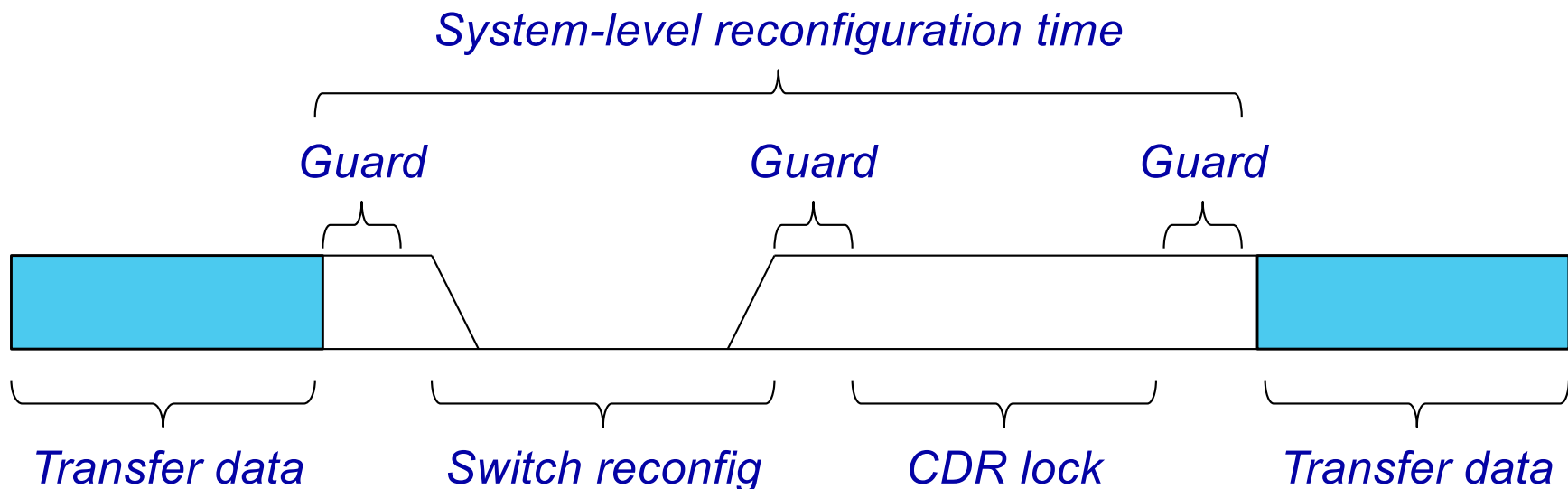


OFC 2018: A. Forencich et. al, "System-Level Demonstration of a Dynamically Reconfigured Burst-Mode Link Using a Nanosecond Si-Photonic Switch"

# System Reconfiguration Time Testbed

- ## Data Plane

Xilinx Virtex Ultrascale FPGA (VCU108 board)

Pattern Generator

Switch Control

Trigger Generator

Error Detector

100G PSM-4

Transmitter (QSFP28)

PC

2x2 Si-Ph Switch

PDFA

BPF

VOA

99%

1%

Power meter

Burst Mode Receiver

1:16

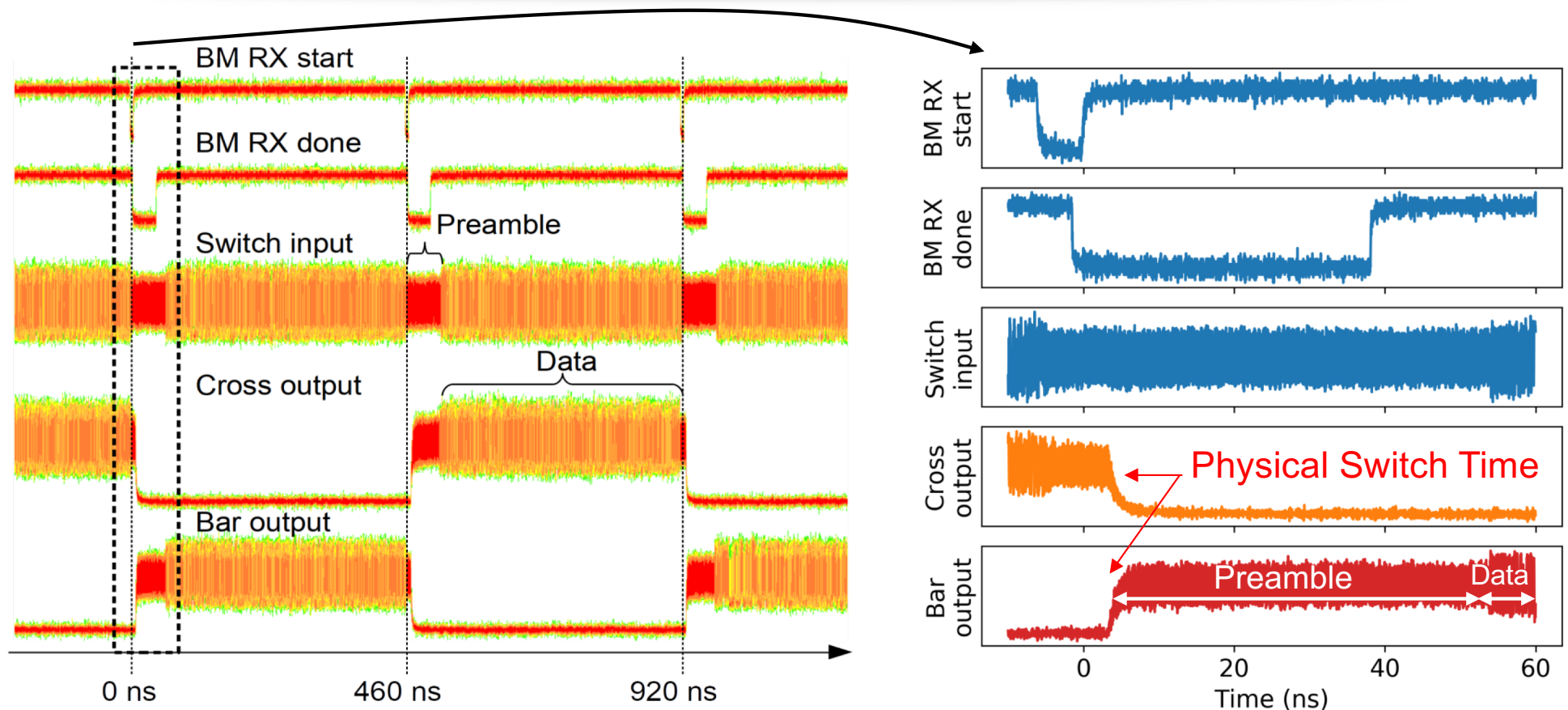1:16

Demux board

~ 1 ns Si-P switching (physical response)

- ## Control plane
  - Xilinx XVCU095 FPGA
  - 25 Gbps pattern generator
  - Trigger generator, switch interface
  - 25 Gbps gated error detector
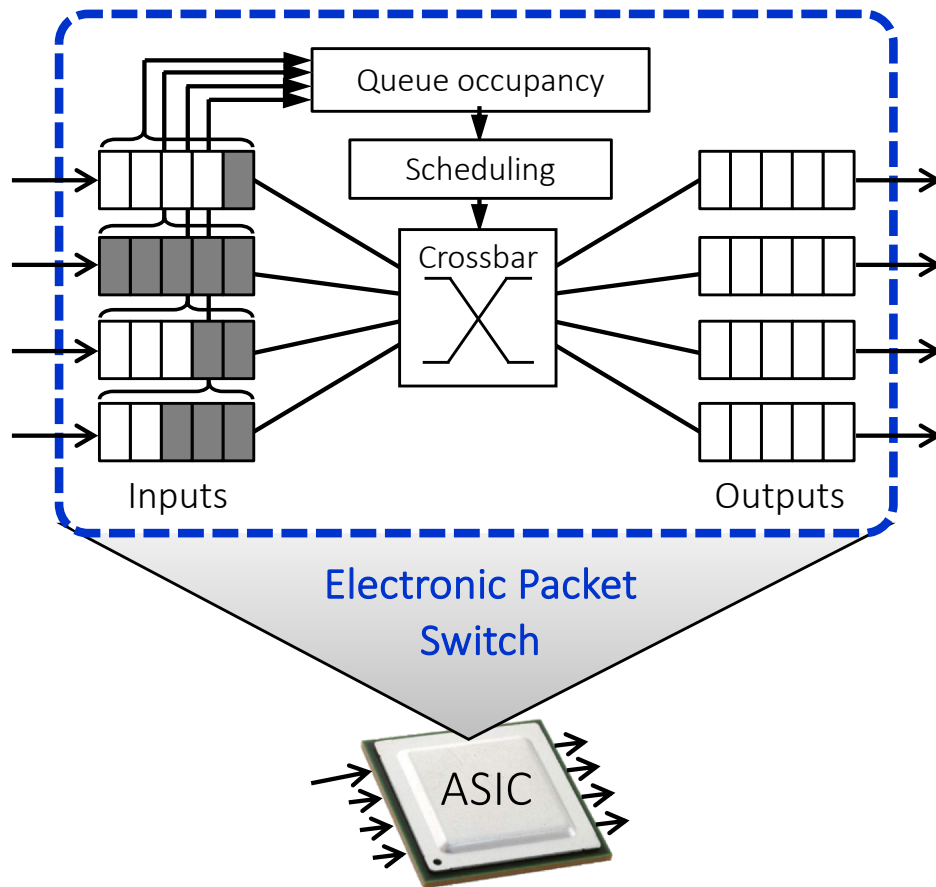
# Measured Waveforms of Photonic Switch and BM-RX



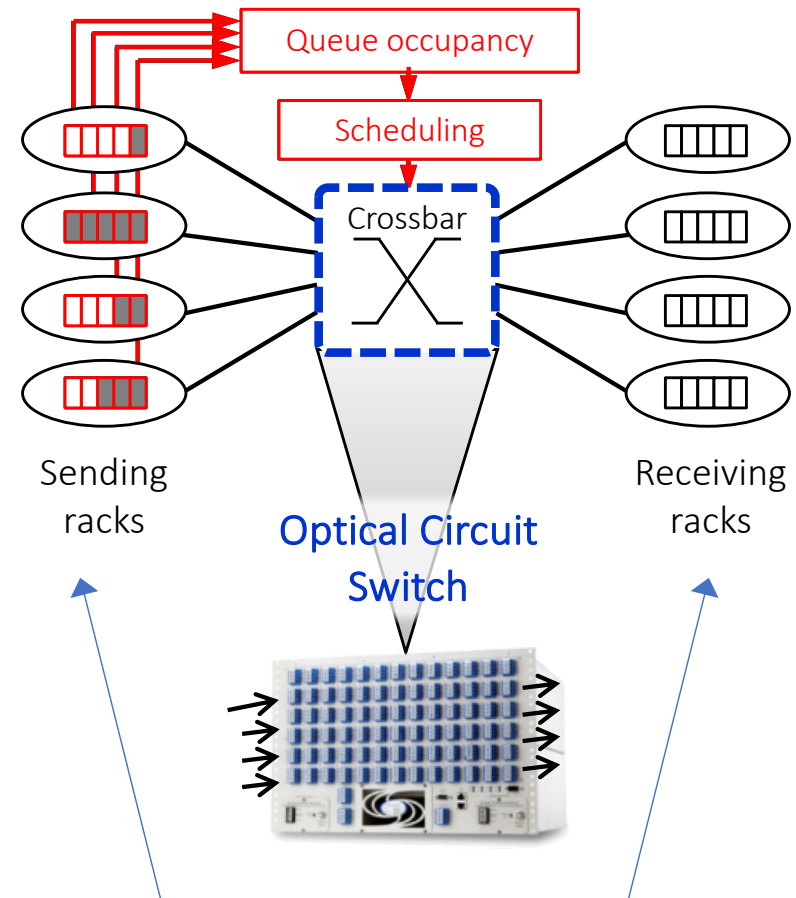| Payload size (B) | 2048 | 1024 | 2048 | 1024 |
|---|---|---|---|---|
| Data rate (Gbps) | 12.5 | 12.5 | 20 | 20 |
| Cycle time (ns) | 1366 | 730 | 858 | 460 |
| BM-Switch time (ns) | 90 | 90 | 60 | 60 |
| Duty cycle (%) | 93 | 87 | 93 | 87 |

Reconfiguration 60-90x slower than physical switch time !

# Software Issues

# The Centralized Control Issue



Electronic Packet Switch diagram: Inputs → Queue occupancy → Scheduling → Crossbar → Outputs, implemented in ASIC.

Data plane doesn't scale to entire datacenter!

Optical Circuit Switch diagram: Sending racks → Queue occupancy → Scheduling → Crossbar → Receiving racks.

Information required for scheduling not locally available

# Steps in Centralized Scheduling
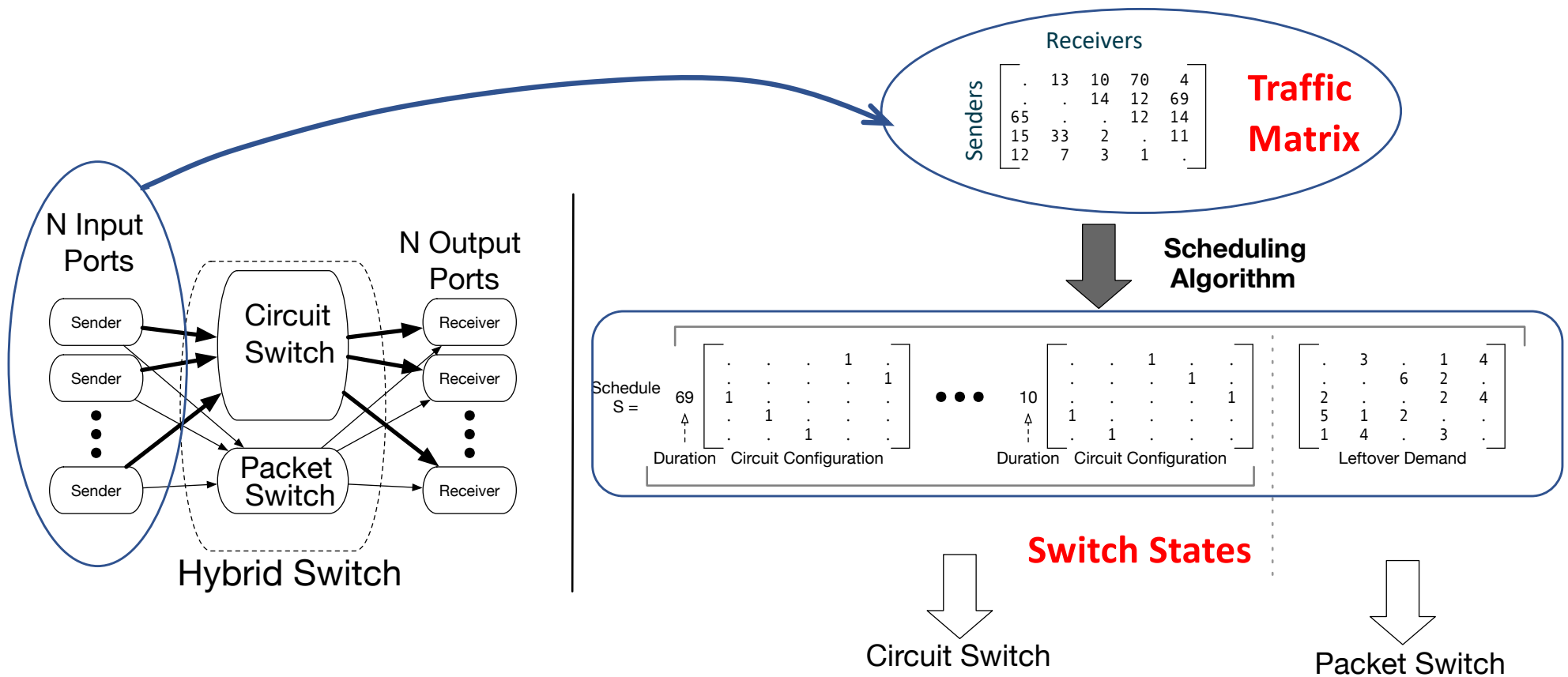


- Collect demand information from endhosts
- Send demand information over a network to central location
- Form the traffic matrix
- Factor traffic matrix into a sequence of switch states
- Finally! Set the switch

Scheduling Techniques for Hybrid Circuit/PacketNetworks, CoNEXT 2015

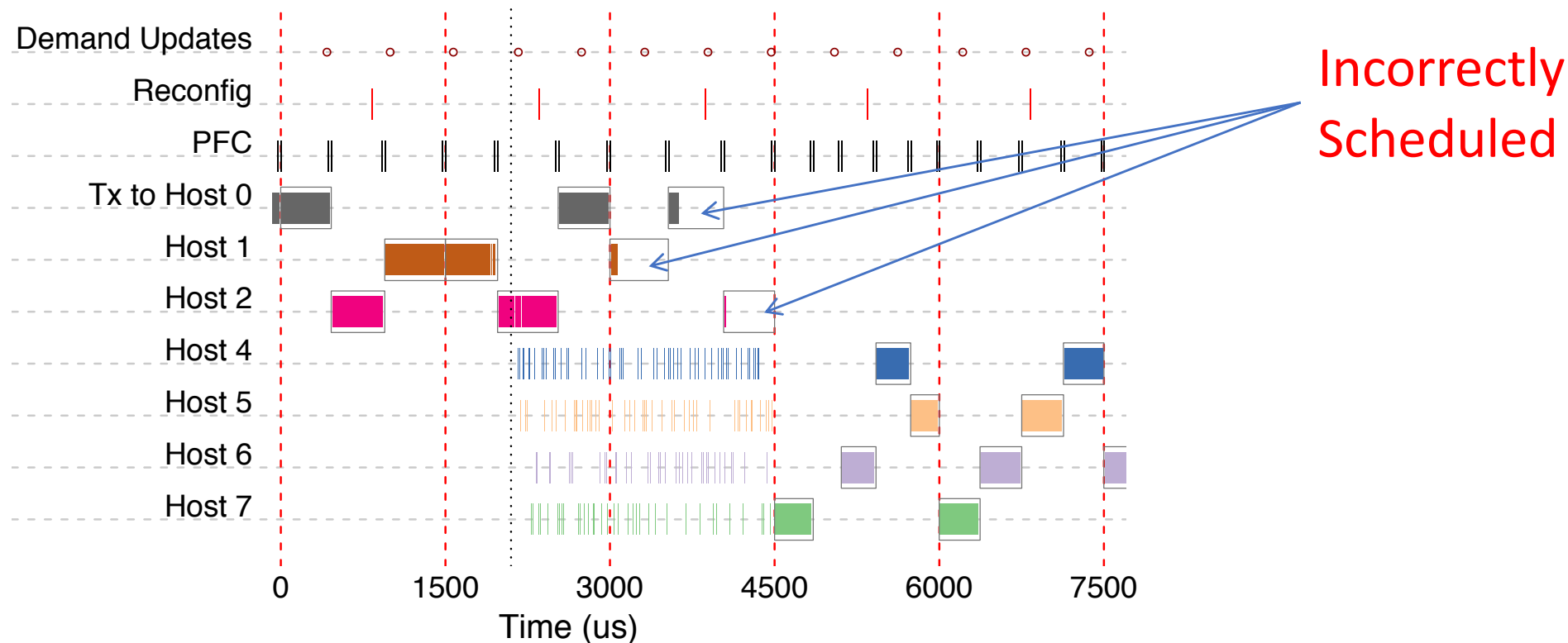# A Centralized Control Plane -ReacToR
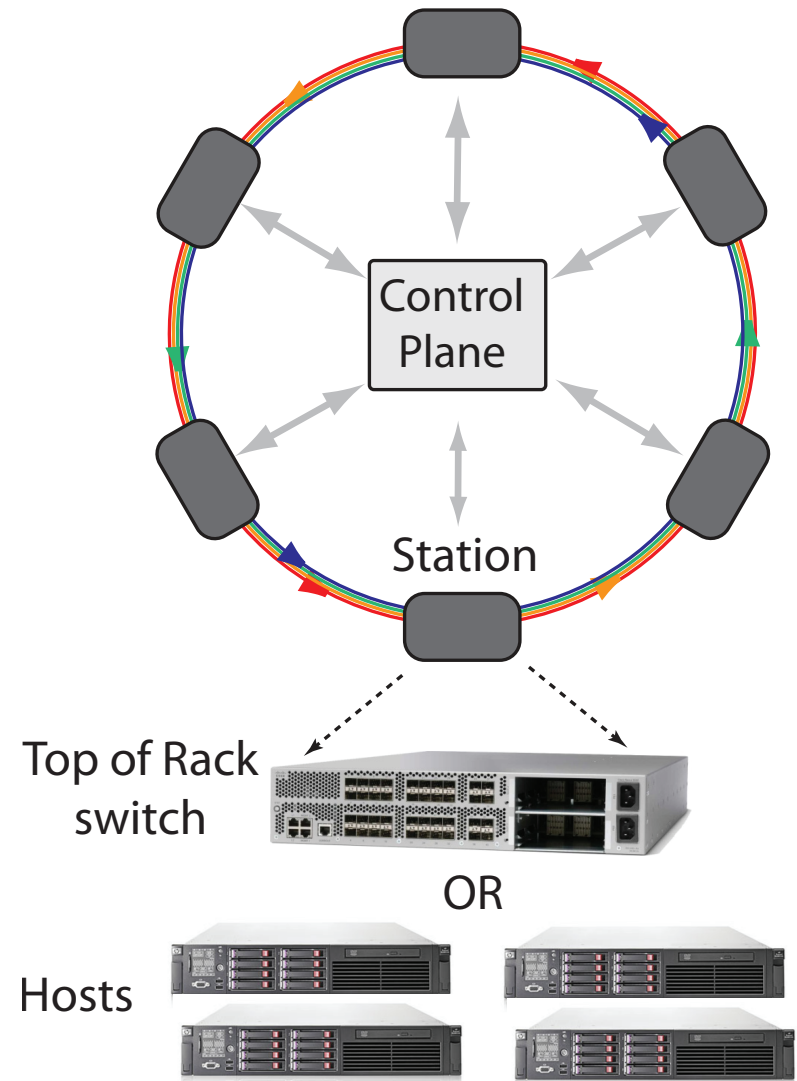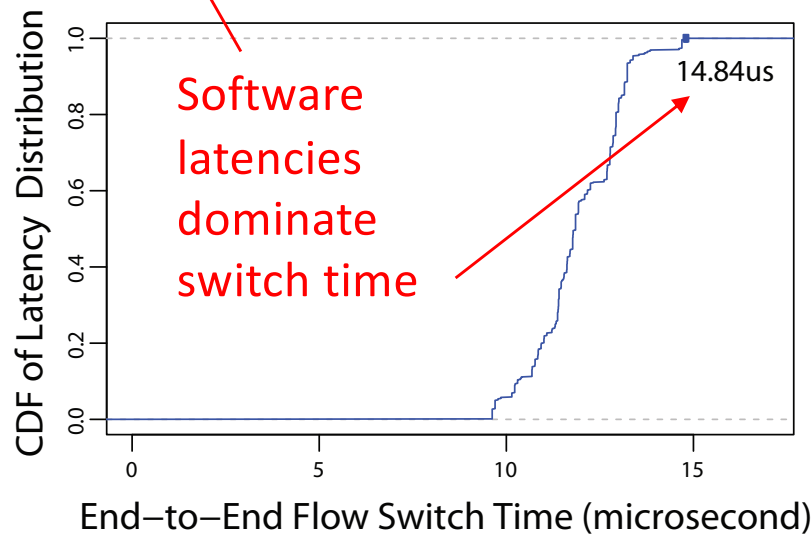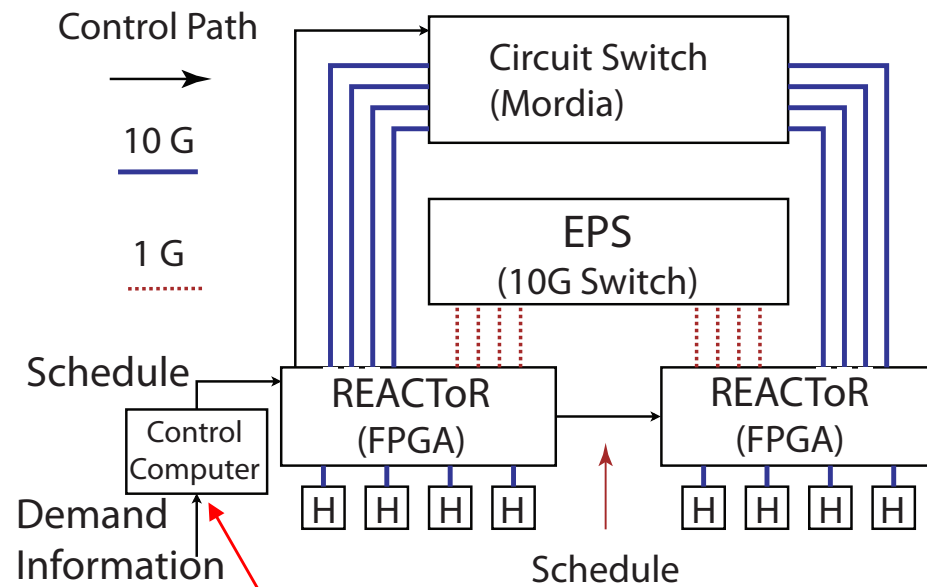


"A multiport microsecond optical circuit switch for data center networking," *PTL 2013*

# Time-Varying Demand



- Control plane tries to allocate circuits based on calculated schedule
- Control plane prototype was slow, did not always schedule correctly, and does not scale – hard lesson learned!

"Circuit Switching Under the Radar with REACToR," NSDI 2014
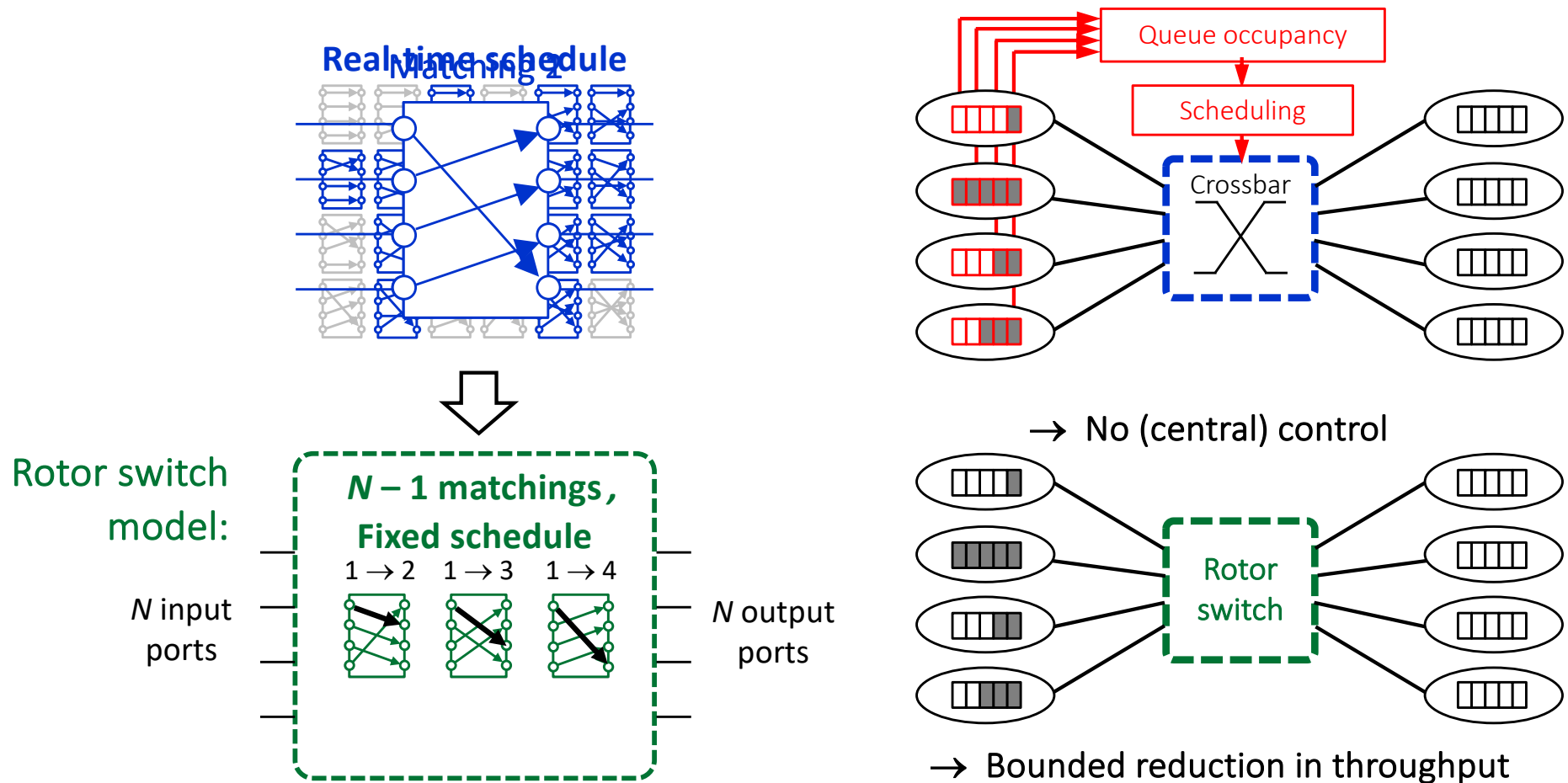
# A New Solution - RotorNet

- No centralized control – inherently more scalable!

- Co-design of optical switch and network
  *Why build a large, fast crossbar that you cannot control?*

- Parallelism decouples minimum latency from switching time

TOMORROW: Max Mellette, invited talk M2C.3, Monday 11 AM, Room 3.
"A Practical Approach to Optical Switching in Datacenters"

# RotorNet has no Central Control



**Real-time schedule**

Matching 1

Rotor switch model:

$N$ input ports

$N − 1$ matchings, Fixed schedule

$1 \rightarrow 2$   $1 \rightarrow 3$   $1 \rightarrow 4$

$N$ output ports

Queue occupancy

Scheduling

Crossbar

→ No (central) control

Rotor switch

→ Bounded reduction in throughput

RotorNet: A Scalable, Low-complexity, Optical Datacenter Network, Sigcomm '17

# Summary

- Mimicking the electronic packet-switched network control plane leads to an optical circuit switch that does not scale

- Even w/o centralized scheduling, the control plane is hard
  - Must reduce/hide system reconfiguration time
  - Must synchronize "asynchronous" end hosts

- RotorNet addresses control plane issues by:
  - No centralized scheduler
  - Using parallelism to bypass system reconfiguration delay
  - *Still must address synchronization with end hosts*

# Contributing Researchers

*Systems*

- **UC San Diego:** Max Mellette, George Papen, George Porter, Alex C. Snoeren, Geoffrey M. Voelker, Amin Vahdat, with students Nathan Farrington, Rishi Kapoor, He Liu, Feng Lu, Rob McGuinness, Arjun Roy, Malveeka Tewari

- **CMU**: David G. Andersen, Srinivasan Seshan, with students Matthew K. Mukerjee , Conglong Li, Nicolas Feltman

- **Intel**: Michael Kaminsky

*Hardware*

- **UC San Diego:** Joe Ford, Max Mellette, George Papen, P.-C. Sun, with students Alex Forencich, Max Mellette, Glenn M. Schuster

- **IBM:** Nicolas Dupuis, Christian Baks, Benjamin G Lee, Laurent Schares

- **Technical University of Denmark:** Valerija Kamchevska (student)