Control Planes for Optical Switching

George Papen

UC San Diego

Opportunities and Challenges for Optical Switching in the Data Center

OFC Workshop 2019
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 $\mu$s (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

![Diagram of system-level reconfiguration time with guards and phases]

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

- 100G PSM-4 Transmitter (QSFP28)
- 2x2 Si-Ph Switch
- Pattern Generator
- Switch Control
- Xilinx Virtex Ultrascale FPGA (VCU108 board)
- BPF
- PDFA
- VOA
- Power meter
- Error Detector
- Burst Mode Receiver
- Demux board (1:16)
- Trigger Generator
- Control plane
  - Xilinx XVCU095 FPGA
  - 25 Gbps pattern generator
  - Trigger generator, switch interface
  - 25 Gbps gated error detector

~ 1 ns Si-P switching (physical response)
Measured Waveforms of Photonic Switch and BM-RX

<table>
<thead>
<tr>
<th>Payload size (B)</th>
<th>2048</th>
<th>1024</th>
<th>2048</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data rate (Gbps)</td>
<td>12.5</td>
<td>12.5</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Cycle time (ns)</td>
<td>1366</td>
<td>730</td>
<td>858</td>
<td>460</td>
</tr>
<tr>
<td>BM-Switch time (ns)</td>
<td>90</td>
<td>90</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>Duty cycle (%)</td>
<td>93</td>
<td>87</td>
<td>93</td>
<td>87</td>
</tr>
</tbody>
</table>

Reconfiguration 60-90x slower than physical switch time!
Software Issues
The Centralized Control Issue

Data plane doesn’t scale to entire datacenter!

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
A Centralized Control Plane - ReacToR

“A multiport microsecond optical circuit switch for data center networking,” PTL 2013

![Diagram of control plane and circuit switch](image-url)
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

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)