INTRODUCING OPTICAL SWITCHING INTO DATACENTER NETWORKS

George Porter (on behalf of many co-authors!)

UC San Diego

University of Washington
Feb 8, 2021
GROWTH OF “HYPERSONE” DATA CENTERS

Network problem: connecting >100,000 servers

1985 - 10 Mb/s
1990 - 100 Mb/s
1995 - 1 Gb/s
2000 - 10 Gb/s
2005 - 100 Gb/s
2010 - 100 Gb/s
HYPERSCALE IMPOSED NEW REQUIREMENTS ON NETWORKS

Can’t buy sufficiently fast core switches!

100,000 x 10 Gb/s = 1 Pb/s
2009: RISE OF “SCALE OUT” NETWORKS

Cheap Merchant silicon

Scaling bandwidth by scaling:
- # switches
- # cables
- # xceivers

64k servers mean 196,608 links

SCALING “TRADITIONAL” FATTREES IS BECOMING EXPENSIVE

Host

64 port switch

1 Gb/s links

Merchant Silicon

year: ~ 2004

Chip

64 ports × 1 Gb/s

~ 2004

Host

64 port switch

1 Gb/s links

Chip

64 ports × 10 Gb/s

~ 2010
SCALING “TRADITIONAL” FATTREES IS BECOMING EXPENSIVE

64 port switch

10 Gb/s links

~ 2010

64 ports × 10 Gb/s

32 port switch

40 Gb/s links

~ 2012

32 ports × 40 Gb/s

(128 ports × 10 Gb/s)
PROLIFERATION OF FAT TREE LAYERS A GROWING PROBLEM

**Optical links**
- 1,000+ Gb/s – 1,000+ meters
- 10 Gb/s – 2,000 meters

- Single mode fiber
- SFP+ transceiver

**Electrical links**
- 1 Gb/s – 100m
- 10 Gb/s – 10 meters

- CAT 5
- 10G DAC

**Datacenter Network**

For every device attached to the network, there are multiple transceivers in the network.

100k nodes: $O(100kW)$ and $O($$$)$
BANDWIDTH GROWTH CONTINUES HOWEVER...
BANDWIDTH GROWTH CONTINUES HOWEVER...

Today

More layers needed

Google's DC networks

Google's DC traffic

(normalized)

Single switch ASIC

Year

2005 2010 2015 2020 2025

Capacity (Tb/s)

10^0 10^2 10^4 10^6
SCALING LIMITATIONS OF CMOS-BASED PACKET SWITCH CHIPS

- Increasing difficulty getting data in/out of the chip
- Divergence between **link** rate and **channel** rate
  - E.g. 100G vs 4x25G
- More fabric layers = higher cost & power

```
Max. chip radix = 64 x 10G
Used chip radix = 16 x 40G
```

```
0.64 TB/s
5.12 TB/s
12.8 TB/s
```

"Hiding" layers
### MOVE TO “CHASSIS” BASED FAT TREES (FACEBOOK, GOOGLE)

**“Traditional” packet switch**

- **128 ports @ 100Gb/s**

**Facebook's “6-pack”**

- **16 ports @ 100Gb/s**

#### Multistage chassis switch

- **Fully-provisioned network – 8,192 end hosts**

<table>
<thead>
<tr>
<th>Architecture</th>
<th># Tiers</th>
<th># Hops</th>
<th># Transceivers</th>
<th># Switch chips</th>
<th># Switch boxes</th>
<th># Fibers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>3</td>
<td>5</td>
<td>49 k</td>
<td>1,280</td>
<td>1,280</td>
<td>25 k</td>
</tr>
<tr>
<td>Multistage Chassis</td>
<td>2</td>
<td>9</td>
<td>33 k</td>
<td>2,304</td>
<td>192</td>
<td>16 k</td>
</tr>
</tbody>
</table>

#### Improvement:

- 1.8× (power)
- 1.5× (cost)

#### Penalty:

- 1.6× (cost)
- 1.8× (latency)
TRENDS

• Conventional datacenter networks facing scaling limitations
  • Largely due to scaling limits of underlying packet switch chips

• Direction 1: Parallel network fabrics
  • Adopted thus far by Facebook and LinkedIn
  • Short-to-medium timeframe

• Direction 2: Replace packet switches with optical switches / circuit switches
  • Medium-to-longer timeframe
RESEARCH TIMELINE: DIRECTION 1: PARALLEL NETWORKS

Parallel Network Fabrics
PARALLEL NETWORK DESIGNS

Conventional architectures:

Chassis architecture still scaling the network up... just hiding the tiers in switch chassis.

Alternative: Scale out via separate physical data planes

- Benefits: Reduced cost, power, and latency
- Tradeoff: Give up a single “fast” network abstraction
Ex. Broadcom’s Tomahawk switch:

32 ports @ 100 Gb/s

OR

128 ports @ 25 Gb/s

Conventional FatTrees
LINK CHANNEL COUNT IS INCREASING

Ex. 100 Gb/s optical link:

\[
\begin{array}{c}
\text{Tx} \\
100 \text{ G} \\
\text{Rx}
\end{array}
\quad = \quad
\begin{array}{c}
\text{Tx} \\
25 \text{ G} \\
\text{Rx} \\
\text{Tx} \\
25 \text{ G} \\
\text{Rx} \\
\text{Tx} \\
25 \text{ G} \\
\text{Rx}
\end{array}
\quad = \quad 4 \times 25 \text{ Gb/s}
\]

- Conventional FatTrees
PARALLEL NETWORKS IN INDUSTRY: FACEBOOK

- from 4 x 128p multi-chip 400G fabric switches
  - 48 FSW ASICs + Control Planes per Pod

- to 16 x 128p single-chip 100G fabric switches
  - 16 FSW ASICs + Control Planes per Pod

4 x 400G = 1.6T uplink per rack
16 x 100G = 1.6T uplink per rack

PARALLEL NETWORKS IN INDUSTRY: FACEBOOK

Simpler and Flatter

- **Over 3X less** switch ASICs and control planes in fabric
- **2.25X less** tiers of chips in the topology
- **Up to 2X less** host-to-host network hops intra-fabric
- **Up to 3X less** host-to-host network hops intra-region

---

**F4**
- 4 planes x 9 chip tiers
- 12 chips/fabric node

**Regional Fabric Aggregator (FA)**

**F16**
- 16 planes x 4 chip tiers
- 1 chip/fabric node

**Flat FA-DU tier**

**Regional Fabric Aggregator (FA)**

**Spine Switch**

**Fabric Switch**

---

RESEARCH TIMELINE: DIRECTION 2: OPTICAL NETWORKS

- Removing Transceivers
- Parallel Network Fabrics
MOTIVATION FOR OPTICAL NETWORKING

The faster the data rate of a cable, the shorter it has to be.

Switch 1
1 Gbps
O(100m)

Switch 1
10 Gbps
O(10m)

Switch 1
100 Gbps
O(1m)

Switch 1
100+ Gbps
1000m+

Transceivers:
• O($100)
• O(10 watts)

Two per each of the 196,608 cables...

Cable requires a transceiver at either end.
- No transceivers needed
- Supports unlimited bandwidth
  - Different service model
  - Not a drop-in replacement
  - Reconfiguration delay $\delta$
2009 – USING 3D MEMS TO REMOVE TRANSCEIVERS

- Technology: telecom-grade 3D-MEMs
- Scalability: 100s of ports
- Target: Inter-“pod”

- Model: 15ms switch time
- Observed: 12.1ms switch time

N. Farrington et al., SIGCOMM 2010
BOTTLENECKS IN NON-SWITCH COMPONENTS

- Telecom not designed for rapid reconfiguration
- Many non-switch bottlenecks in optical components

2-second link flap prevention

Packet-switch baseline

Observed circuit-switched bw

Transceiver
Electronic Dispersion Compensation
1. Collect counters from packet switches
2. Estimate “true” demand
3. Calculate max-weighted matching
4. Reconfigure packet and optical switches

- One cycle ≈ one second
- Circuits try to “match” current network conditions
- Stateless in between assignments

Hedera demand estimator + Edmund’s Algorithm
APPLICABILITY LIMITED BY SLOW SWITCH TIME & CONTROL PLANE

- Model: 15ms switch time
- Reality: 1000ms control plane
- To “capture” more of the traffic in optics, need a faster switch and faster control plane

"Hardware Requirements for Optical Circuit Switched Data Center Networks", Farrington et al., OFC 2011
RESEARCH TIMELINE: DIRECTION 2: OPTICAL NETWORKS

- Removing Transceivers
- Parallel Network Fabrics
- Focus on Switch Time
REMOVING TRANSCEIVERS

- Millisecond switching
  - Helios, Sigcomm '10

- Microsecond switching
  - Mordia, Sigcomm '13

- Hybrid network & scheduling
  - ReacToR, NSDI '14
  - Solstice, CoNEXT '15

- Simple control
  - RotorNet, Sigcomm '17

- Low-latency for RotorNet
  - Opera, NSDI '20

Timeline:
- 2009
- 2019
USING 2D MEMS TO “CHASE MICE”

• Needed a faster switch
• 2D MEMS very fast…
  • 2 μs switch time + ringing
  • Approx 11.5 μs total
• …but not scalable (~24 ports)
  • Lots of ports → slow
  • Few ports → fast
HOW MICROSECOND SWITCHING CHANGES THE CONTROL PLANE

• Microsecond switching prevents scheduling with “fresh” data
  • Collecting demand a bottleneck!
  • Insight: amortize series of switch configurations across a single demand estimate:
    \[ TM' = \sum_{i}^{N} t_i P_i \]
  • Embodied by Solstice and Eclipse algorithms
  • Result: “Chasing” demand
  • Reactive and responsive

Step 1. Gather traffic matrix \( TM \)
Step 2. Scale \( TM \) into \( TM' \)
Step 3. Decompose \( TM' \) into schedule
Step 4. Execute schedule in hardware
RESEARCH TIMELINE: DIRECTION 2: OPTICAL NETWORKS

- Removing Transceivers
- Non Crossbar Switching
- Parallel Network Fabrics
- Focus on Switch Time
NON-CROSSBAR NETWORKS

Performance

Hybrid network & scheduling
ReacToR, NSDI '14
Solstice, CoNEXT '15

Microsecond switching
Mordia, Sigcomm '13

Millisecond switching
Helios, Sigcomm '10

Simple control
RotorNet, Sigcomm '17

Low-latency for RotorNet
Opera, NSDI'20

Practicality

2009

Time

2019
Toward 100+ Petabit/second datacenters

**Challenge:** deliver (very) low-cost bandwidth at scale

- **New protocols**
  Load balancing, congestion control, ...

- **New topologies**
  Jellyfish, Longhop, Slimfly, ...

- **New hardware**
  Optical circuit switching, RF/optical wireless, ...

- **Same switching model**
  New “Rotor” switching model

**Co-design:**
- Protocol
- Topology
- Hardware

**RotorNet** → “Future-proof” bandwidth (2× today) + simple control + ...
Optical switching – benefits & barriers

Data plane doesn’t scale to entire datacenter!

Electronic Packet Switch

Copper:
25 Gb/s

I/O limits bandwidth

ASIC

Fiber:
> 1 Tb/s

Cheap, future-proof bandwidth

Optical Circuit Switch

Sending racks/hosts

Receiving racks/hosts

Queue occupancy

Scheduling

Crossbar

Inputs

Outputs
Queue occupancy

Scheduling

Crossbar

UCSC CSE
Rotor switching model simplifies control

Crossbar model:
- $N$ input ports
- $N$ output ports

Real-time schedule

Matching 1

Queue occupancy

Scheduling

Rotor switch model:
- $N - 1$ matchings,
  - Fixed schedule
  - $1 \rightarrow 2$
  - $1 \rightarrow 3$
  - $1 \rightarrow 4$

No (central) control

→ Bounded reduction in throughput
$N$ input ports

$N$ output ports
Rotor switches have a simpler implementation

Optical Crossbar:

Optical Rotor switch:

• Cost and complexity scale with:

  Ports

Ex. 2,048 ports: 4,096 mirrors
2,048 directions

Matchings (<< Ports)

2 mirrors
16 directions
RotorNet architecture overview

- Forwarding?
  - Optical Rotor switch → More scalable
  - Rotor switching model → Simpler control

- Topology?

- Rotor switching model

1-hop forwarding over Rotor switch

• Wait for direct path:

  Matching cycle 1

  Node 1 → 2, 3, 4
  Node 2 → 3, 4, 1
  Node 3 → 4, 1, 2
  Node 4 → 1, 2, 3

  Matching cycle 2

  ... 

  Uniform traffic → 100% throughput

• But datacenter traffic can be sparse ...
1-hop forwarding & sparse traffic = low throughput

- Wait for direct path:

  Problem: single flow $\rightarrow$ 33% throughput

- Hint at improvement: network is underutilized
2-hop forwarding better for sparse traffic

- Not new: Valiant ('82) & Chang et al. ('02)

![Diagram showing matching cycles and node connections]

**Throughput:**

- Single flow:
  - 33% (1-hop) → 100% (2-hop)
- Uniform traffic:
  - 100% (1-hop) → 50% (2-hop)

- Optimization: can we adapt between 1-hop and 2-hop forwarding?
RotorLB: adapting between 1 & 2-hop forwarding

RotorLB (Load Balancing) overview:

• Default to 1-hop forwarding

• Send traffic over 2 hops only when there is extra capacity

• Discover capacity using in-band pairwise protocol:

→ RotorLB is fully distributed
Throughput of forwarding approaches (256 ports)

[1] Ghobadi et al. Sigcomm '16
[2] Roy et al. Sigcomm '15

One connection

Uniform traffic

Ideal packet switch

3:1 packet switch

2-hop forwarding

1-hop forwarding

RotorLB

FB (Hadoop)[2]

FB (web)[2]

MSFT[1]
Throughput of forwarding approaches (256 ports)

Price of simple control

2× bandwidth (similar cost)

Ideal packet switch

3:1 packet switch

[1] Ghobadi et al. Sigcomm ’16

MSFT[1]
FB (web)[2]
FB (Hadoop)[2]
RotorNet architecture overview

- RotorLB → Distributed, bounded throughput
- Topology?
- Optical Rotor switch → More scalable
- Rotor switching model → Simpler control
How should we build a network from Rotor switches?

At large scale:

• **High latency:**
  Sequentially step through many matchings

• **Fabrication challenge:**
  Monolithic Rotor switch with many matchings

• **Single point of failure**
Distributing Rotor matchings = lower latency

Fault tolerant

Reduced latency:
• Access matchings in parallel

Simplifies Rotor switches:
• Matchings << ports
• More scalable, less expensive
Rotor switching is feasible today

Validated feasibility of entire architecture:
(8 endpoints)

100× faster switching than crossbar

Prototype Rotor switch

RotorLB
RotorNet topology
Optical Rotor switch
Rotor switch model

Inputs / Outputs
Matchings
Optics
SEQUENTIAL SWITCHING ENABLES NEW APPROACH TO BEAMSTEERING

"Pinwheel" sequential beam deflector

High-speed spindle
(e.g. commercial 3.5” 7200 RPM drive)

Diffracted beam

Faceted disk
(custom patterned with diffraction gratings)

Input beam
Initial results indicated that laser writing can produce the features needed.
PROTOTYPE PINWHEEL IN 3.5” HGST DESKSTAR NAS DRIVE

With encoder, encoder tracks, and clear cover
ROTOR SWITCH PROTOTYPE

Optical layout:

- Out 1
- Input
- Out 2

Grating pinwheel

Laser-written grating pinwheel

Diffracted beam

Input beam

(Crosstalk: < 30 dB)
(Operating spectrum: > 120 nm)
(2-pass insertion loss: 5 – 8 dB*)

(*can be improved with better grating)

(WD) HGST Deskstar NAS drive
THE PINWHEEL ENABLES MICROSECOND-SCALE SWITCHING

Image of fiber I/O array

Switching transitions

15 µs reconfiguration @ 7200 RPM
(1,000 x faster than commercial MEMS OXC)
**IMPROVED PERFORMANCE WITH NEW PROTOTYPE**

**1st Prototype:** MEMS selector switch
- Higher loss optics on enclosed ½” breadboard
- 150 µs switching
- I/O to external connection patch panels

**2nd Prototype:** “rotor” switch with pinwheel
- Lower loss optics mounted on vibration-isolated rail
- 15 µs switching (@ 7200 RPM)
- I/O with 4x internal connection patch panels
RACK MOUNTED TESTING OF NEW ROTOR SWITCH PROTOTYPE
RotorNet scales to 1,000s of racks

- Rotor switch design point: 2,048 ports, 1,000× faster switching than crossbar
  Details in: W. Mellette et al., *Journal of Lightwave Technology ’16*
  W. Mellette et al., *OFC ’16*

- 2,048-rack data center: → Latency (cycle time) = 3.2 ms

- Faster than 10 ms crossbar reconfiguration time

- Hybrid network for low-latency applications
RotorNet component comparison

<table>
<thead>
<tr>
<th>Network</th>
<th># Packet switches</th>
<th># Transceivers</th>
<th># Rotor switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:1 Fat Tree</td>
<td>2.6 k</td>
<td>103 k</td>
<td>0</td>
</tr>
</tbody>
</table>

RotorNet delivers:

- Today: Bandwidth 2× less expensive
- Future: Cost advantage grows with bandwidth
- Benefits of optical switching without control complexity
RESEARCH TIMELINE: DIRECTION 2: OPTICAL NETWORKS

- Removing Transceivers
- Non-Crossbar Switching
- Parallel Network Fabrics
- Focus on Switch Time
- All-optical Network
REMOVING TRANSCEIVERS

Performance

Time

2009

Practicality

Millisecond switching
Helios, Sigcomm '10

Microsecond switching
Mordia, Sigcomm '13

Hybrid network & scheduling
ReacToR, NSDI '14
Solstice, CoNEXT '15

Simple control
RotorNet, Sigcomm '17

Low-latency for RotorNet
Opera, NSDI'20

2019
Expander graph networks – an alternative to Fat Tree topologies

- Similar hardware, cost, and power savings to an oversubscribed Fat Tree
- Improved throughput vs oversubscribed Fat Tree at low load

“Bandwidth tax” – Reduction in throughput at high traffic loads
  – Proportional to average path length
Bandwidth tax limits throughput in expander networks

Bandwidth tax = 2 → Throughput = 50% at high load

→ Is it possible to support high loads while reducing cost and power?
Reconfigurable networks enable higher throughput

Reconfigured direct links: bandwidth tax = 1

Multi-hop links bandwidth tax = 2

Reconfigure links

Flow 1: 100% Flow 2: 100%

✓ Reconfiguration permits high throughput at high load

Added complexity: how do we decide which links to reconfigure and when?

→ “RotorNet” (Sigcomm ‘17) – fixed schedule of direct circuits

Today’s circuit switching technologies reconfigure too slowly → high latency
Our contribution: we can have the best of static *and* reconfigurable

Reconfigurable networks: high latency

Expander networks: low latency

Workload = Short flows + Long flows

Latency-bound + Throughput-bound

“Opera” – combining expanders and reconfiguration in a single, unified network
Opera’s design – part 1: providing low-bandwidth-tax connectivity

Full, direct inter-rack connectivity with $N$ matchings:
Opera’s design – part 1: providing low-bandwidth-tax connectivity

Full, direct inter-rack connectivity with $N$ matchings:

$\begin{align*}
\text{M}_1 & \quad \text{M}_2 & \quad \cdots & \quad \text{M}_N
\end{align*}$

Time
Opera’s design – part 2: providing low-latency connectivity

- Short, latency-bound flows can be sent immediately over multi-hop paths (high BW tax)
- Long, throughput-bound flows can wait for direct paths (low BW tax)

Key property: Opera only pays a bandwidth tax for short flows → lower average tax
Choosing matchings

i. **Expansion**  
Union of 3 or more **randomly-structured matchings** is an expander \([1]\)


ii. **Direct connectivity between all racks over time**

Factor complete graph into \(N\) **randomly-structured** & **disjoint** matchings:

\[
\text{Source Rack} = \text{Factored complete graph} = M_1 + M_2 + \cdots + M_N
\]
Offsetting reconfigurations for continuous connectivity

Circuit sw 1:

M₁ → M₅

Circuit sw 2:

M₂ → M₆

Circuit sw 3:

M₃ → M₇

Circuit sw 4:

M₄ → M₈

Cycle: ~ 1-10 ms

Dwell ~ 100 µs

Reconfig. ~ 10 µs

Time to wait for direct path → cutoff between “short” & “long” flows
Quantifying the cutoff

For 10 Gb/s – 100 Gb/s links:

- Long flows ≥ 15-30 MB can afford to wait for direct paths
- Short flows < 15-30 MB cannot wait for direct paths

Published data center flow distributions:

Most flows must pay bandwidth tax

> 90% of bytes
Packet simulations: throughput and latency for 100-rack network

**Workload 1:** All-to-all shuffle (favorable)

→ 4x higher throughput & faster completion

**Workload 2:** Shuffle + MSFT websearch workload (challenging)

→ 2-4x higher throughput & equivalent completion times for short flows

**Workload 3:** MSFT datamining (100 B – 1 GB flows)

→ 60% higher admissible load with equivalent FCTs
Practical considerations

**Fault tolerance:**
- Full connectivity maintained with 4% of links, 7% of ToRs, or 40% of circuit switches failed
  (Better than oversubscribed Fat Tree, not as good as static expander)
- Failures detected and disseminated within $O(10 \text{ ms})$

**Prototype implementation:**
- Time-synchronized routing implemented on programmable Barefoot Tofino switch with P4
- Opera scales to 1,000’s of racks, 10,000’s of servers with commodity switch table sizes
CREDITS AND THANKS TO MY COLLEAGUES/STUDENTS

- Alex C. Snoeren
- Alex Forencich
- Amin Vahdat
- Andrew Grieco
- Anthony Lentine
- Arjun Roy
- Chang-Heng Wang
- Christopher DeRose
- Chunming Qiao
- Conglong Li
- David G. Andersen
- Douglas C. Trotter
- Feng Lu
- Geoffrey M. Voelker
- George C. Papen
- George Papen
- Glenn M. Schuster
- Guohui Wang
- Hamid Bazzaz
- He Liu
- Jordan Davis
- Joseph Ford
- Joshua Zhu
- Kai Chen
- Li Chen
- Malveeka Tewari
- Matthew K. Mukerjee
- Michael A. Kozuch
- Michael Kaminsky
- Minlan Yu
- Nathan Farrington
- Nicolas Feltman
- Pang-Chen Sun
- Rajdeep Das
- Richard Strong
- Rishi Kapoor
- Rob McGuinness
- Ryan Aguinaldo
- Shan Zhong
- Shaya Fainman
- Shayan Mookherjea
- Sivasankar Radhakrishnan
- Srinivasan Seshan
- Stefan Savage
- T. S. Eugene Ng
- Tajana Rosing
- Tara Javidi
- Vikram Subramanya
- William M. Mellette
- Yeshaiahu Fainman
- Yibo Guo
THE FUTURE OF OPTICAL SWITCHING FOR DATACENTERS/HPC

• Reaching the limits of CMOS-based packet switching
  • In terms of cost, power, performance...

• Direction 1: scale bandwidth by adding parallel dataplanes

• Direction 2: scale bandwidth by replacing packet switches with optical ones
  • Unique opportunity to incorporate novel optical devices such as spinning pinwheel/hard drive based switches

• Thank you for your time and attention!
UC San Diego