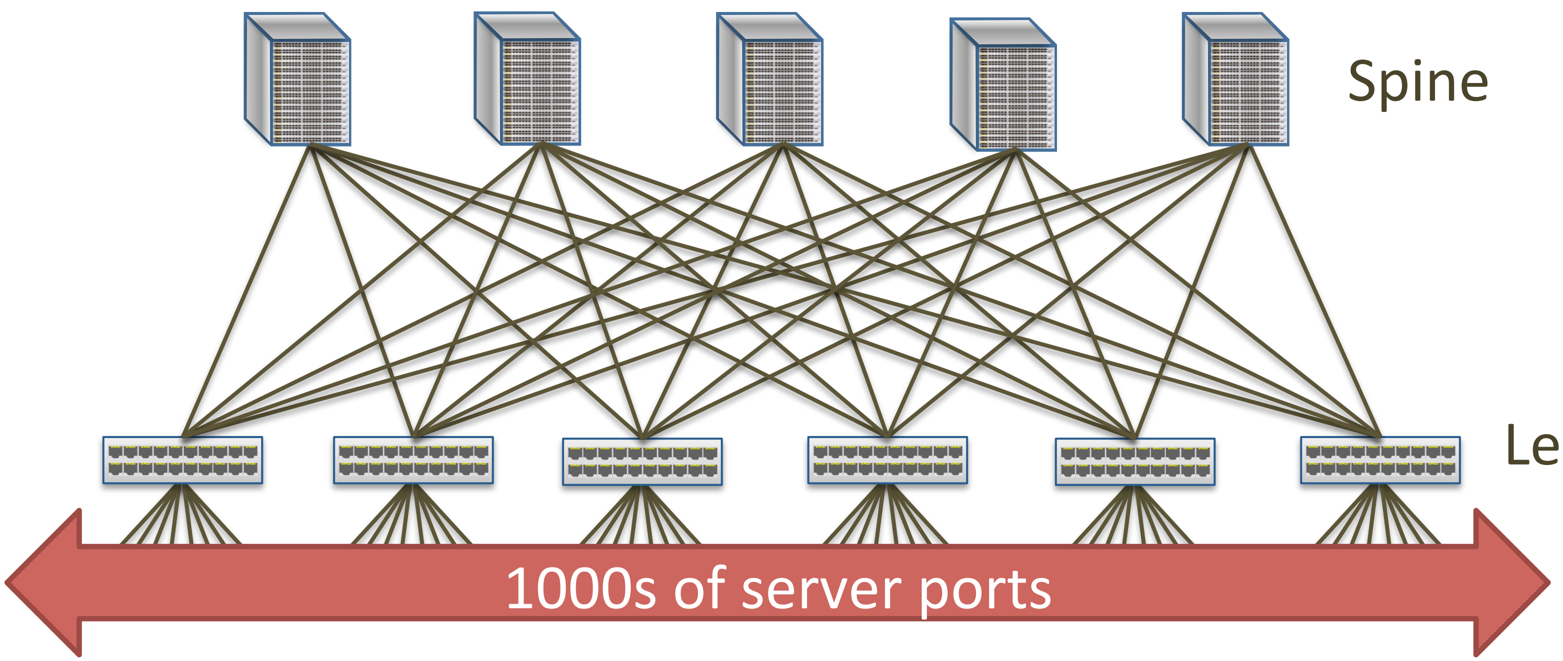
6.888
Lecture 9:
Wireless/Optical Datacenters

Mohammad Alizadeh and Dinesh Bharadia

✧ Many thanks to George Porter (UCSD) and Vyas Sekar (Berkeley)

Spring 2016

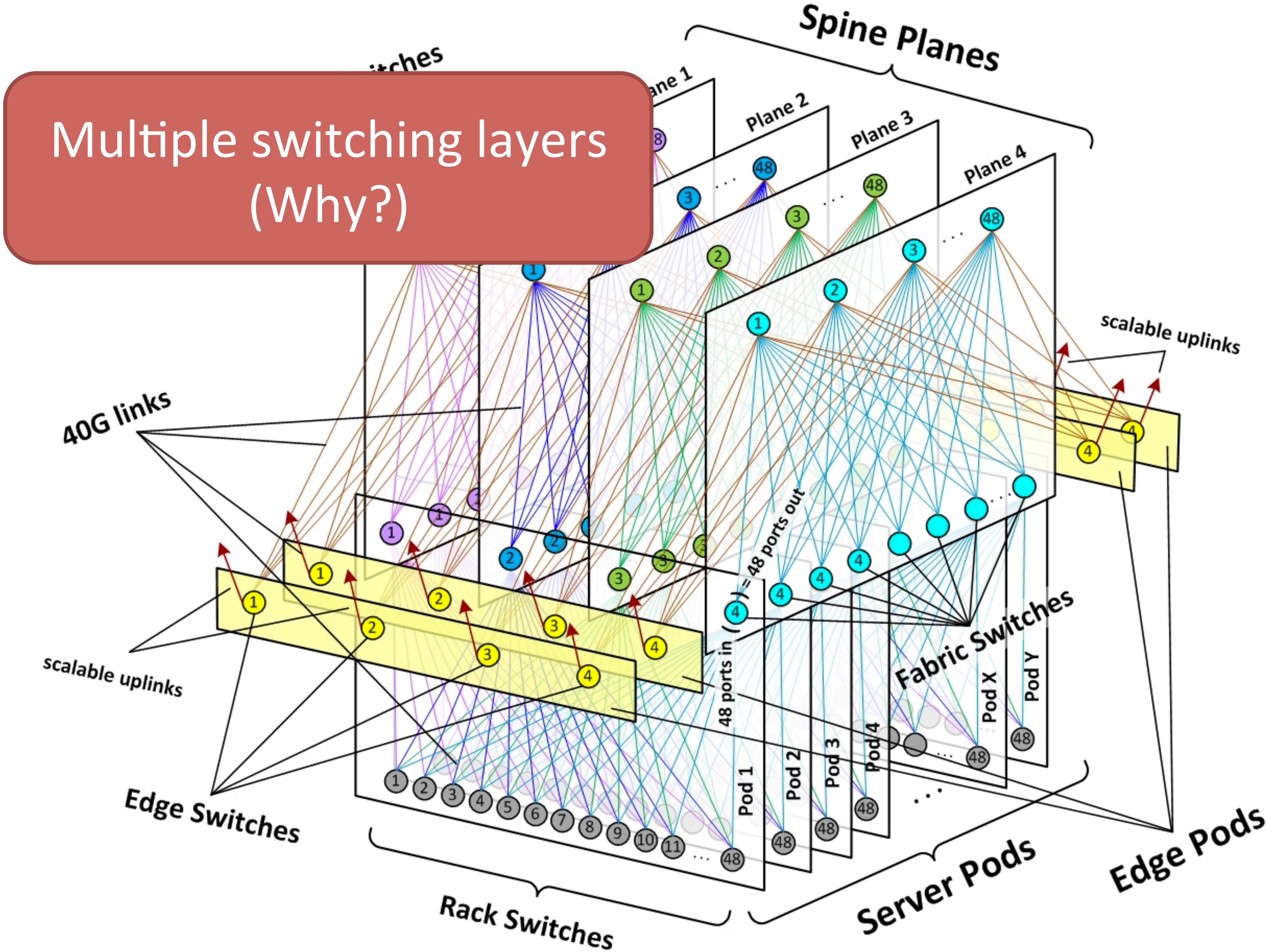
Datacenter Fabrics



Scale out designs (VL2, Fat-tree)

- ➢ Little to no oversubscription
- ➢ Cost, power, complexity

Multiple switching layers (Why?)



✧ https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

Building Block: Merchant Silicon Switching Chips

Switch ASIC

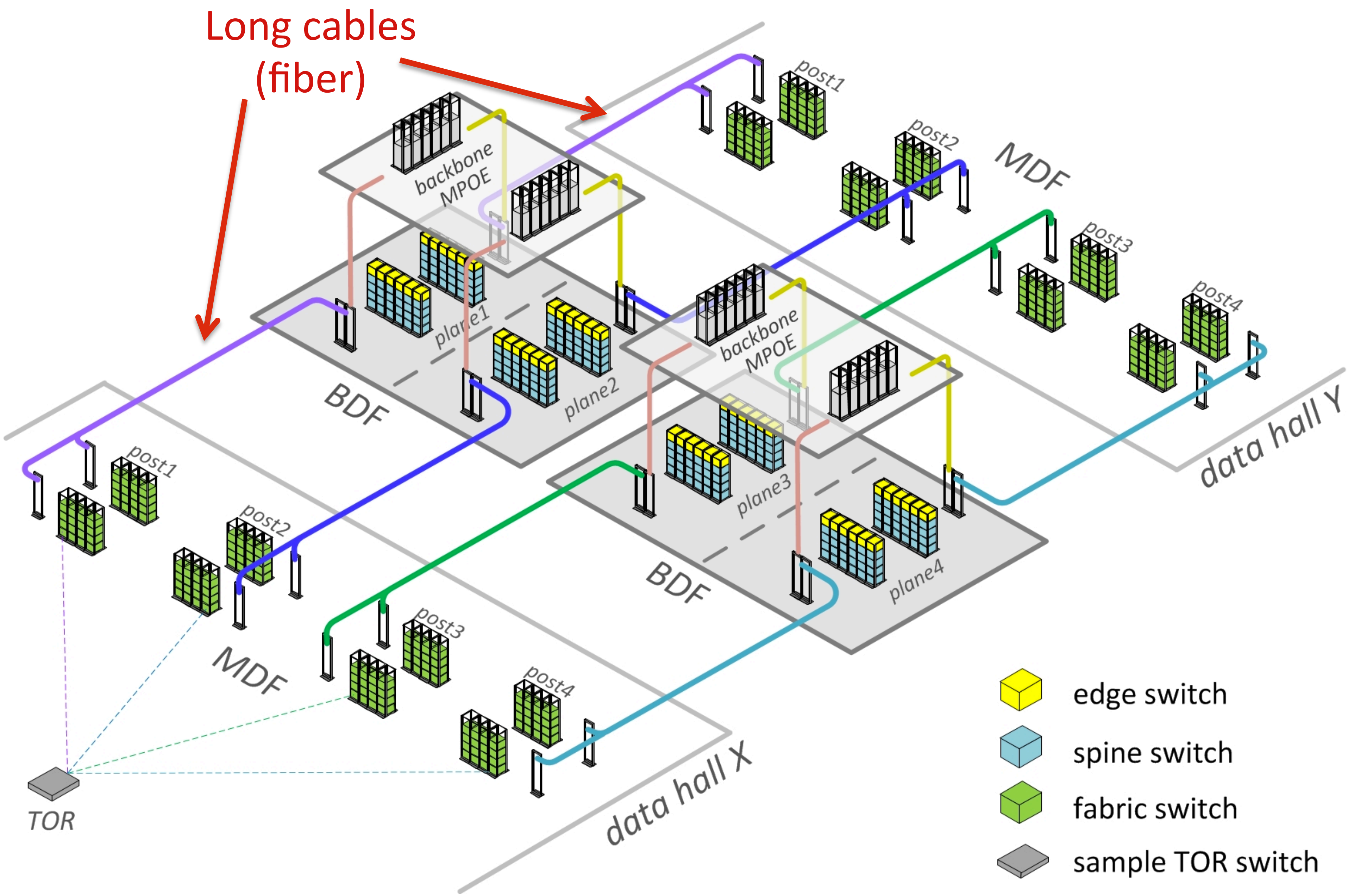
6 pack

Facebook Wedge

Limited radix: 16x40Gbps

High power: 17 W/port

✧ Image courtesy of Facebook



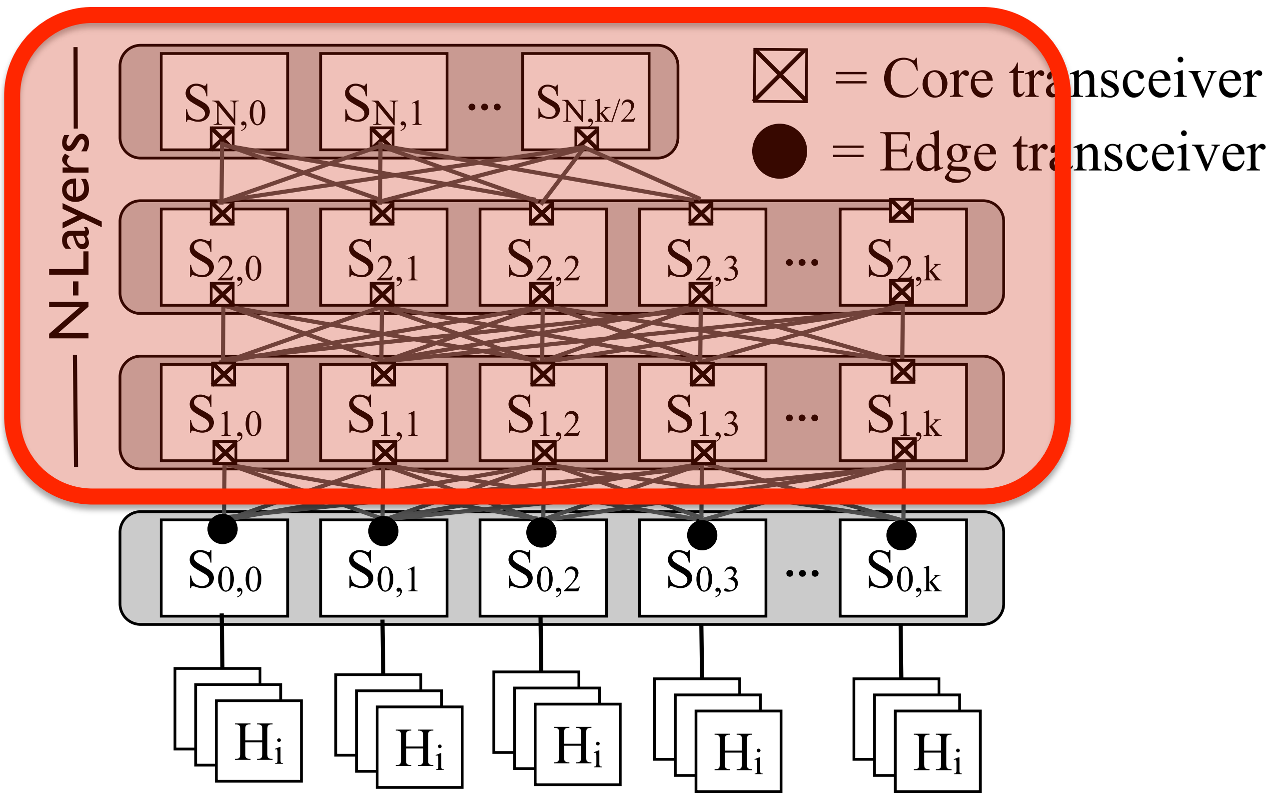
✧ https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

Scale-out packet-switch fabrics

Large number of switches, fibers, optical transceivers

Power hungry

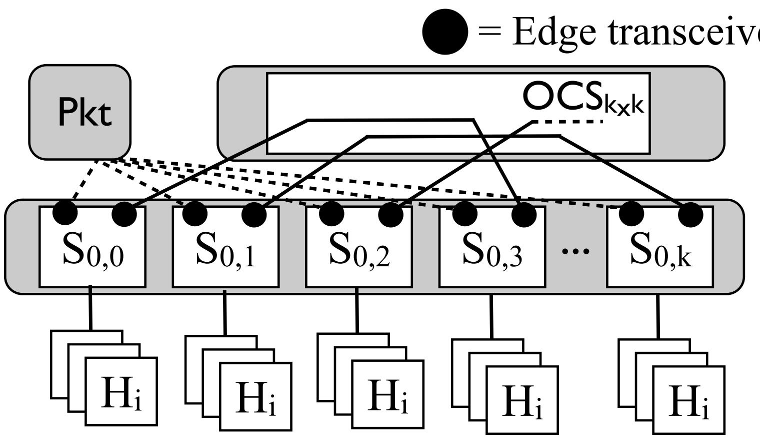
Hard to expand



Beyond Packet-Switched DC Fabrics

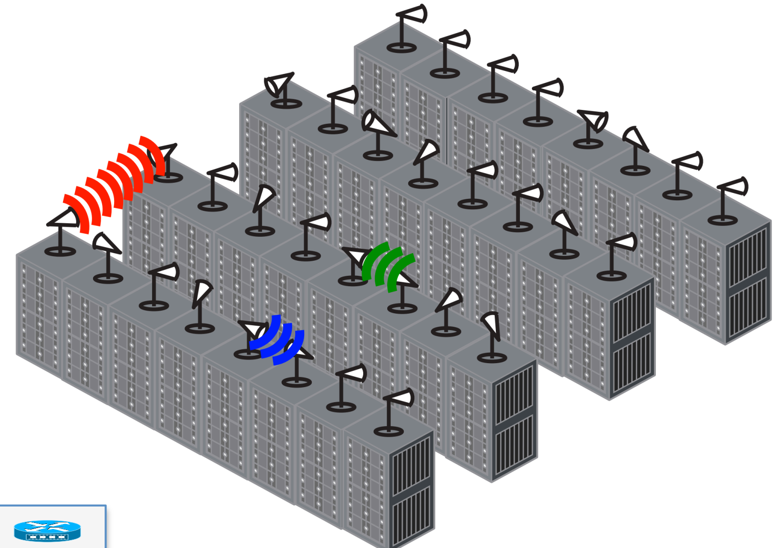
Optical circuit switching

[Helios, cThrough, Mordio, ReacTor, …]



60 GHz RF

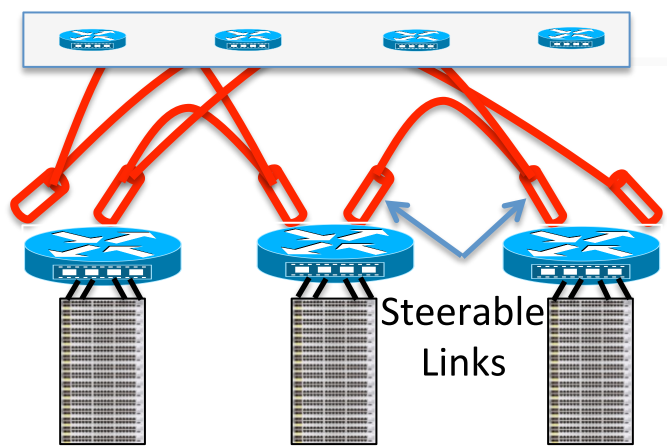
[Flyways, MirrorMirror]



✧ Fig. from presentation by Xia Zhou

Free-space Optics

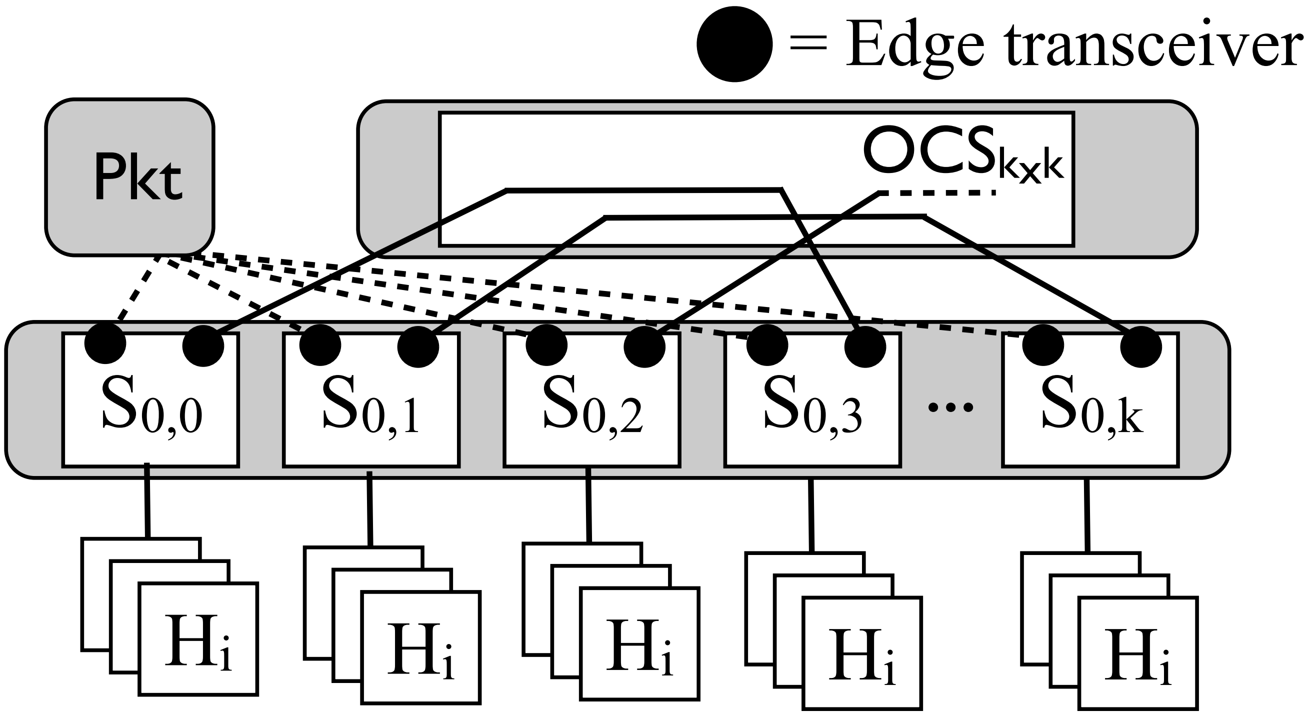
[FireFly]



Integrating Microsecond Circuit Switching into the Data Center

- Slides based on presentation by George Porter (UCSD)

Key idea:
Hybrid Circuit/Packet Networks



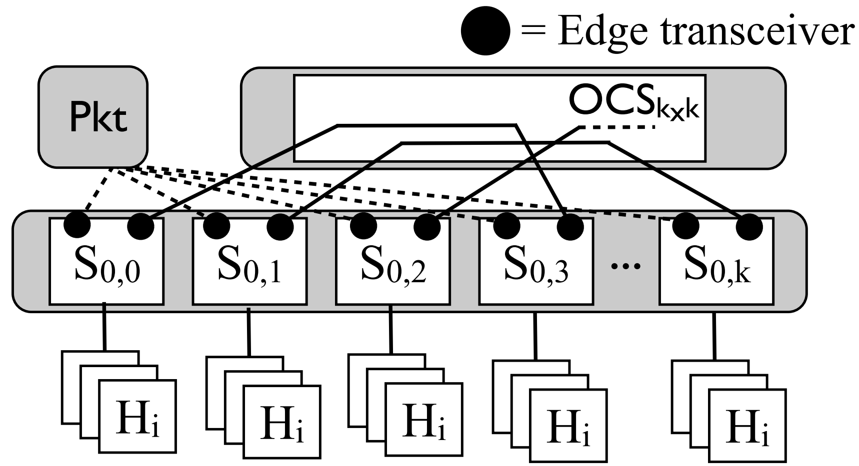
Why build hybrid switch?

Circuit vs. Packet Switching

Observation: Correlated traffic → Circuits

Electrical Packet	**Optical Circuit**
$500/port	$500/port
10 Gb/s fixed rate	Rate free (10/40/100/400/+)
12 W/port	240 mW/port
Transceivers (OEO)	No transceivers
Buffering	No buffering
Per-packet switching	Duty cycle overhead
In-band control	Out-of-band control

Disadvantages of Circuits



Despite advantages, circuits present different service model:

- Point-to-point connectivity
- Must wait for circuit to be assigned } *affects throughput, latency*
- Circuit "down" while being reconfigured } *affects network duty cycle; overall efficiency*

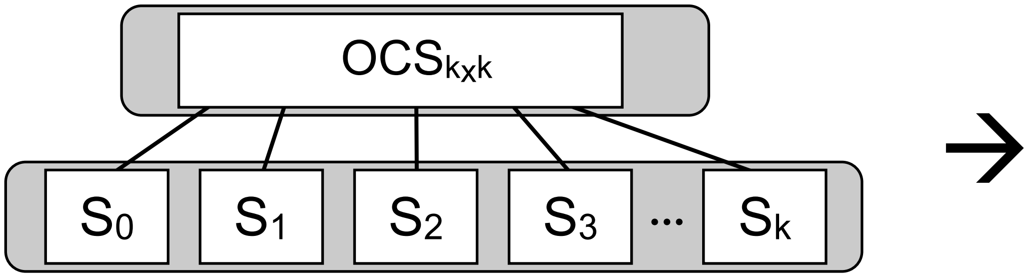
Stability Increases with Aggregation

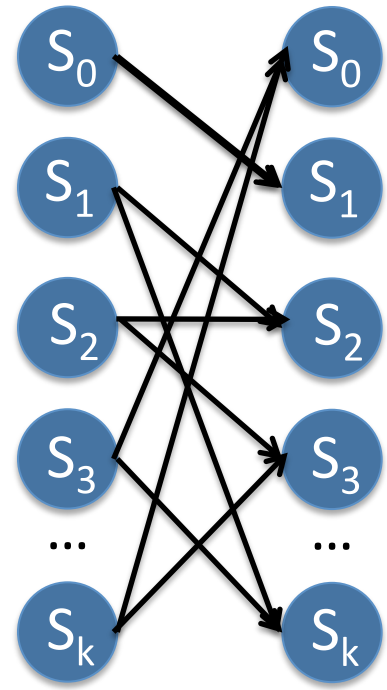
Inter-Data Center
Inter-Pod
Inter-Rack
Inter-Server
Inter-Process
Inter-Thread

Where is the Sweet Spot?

1. **Enough Stability**
2. **Enough Traffic**

Mordia OCS model

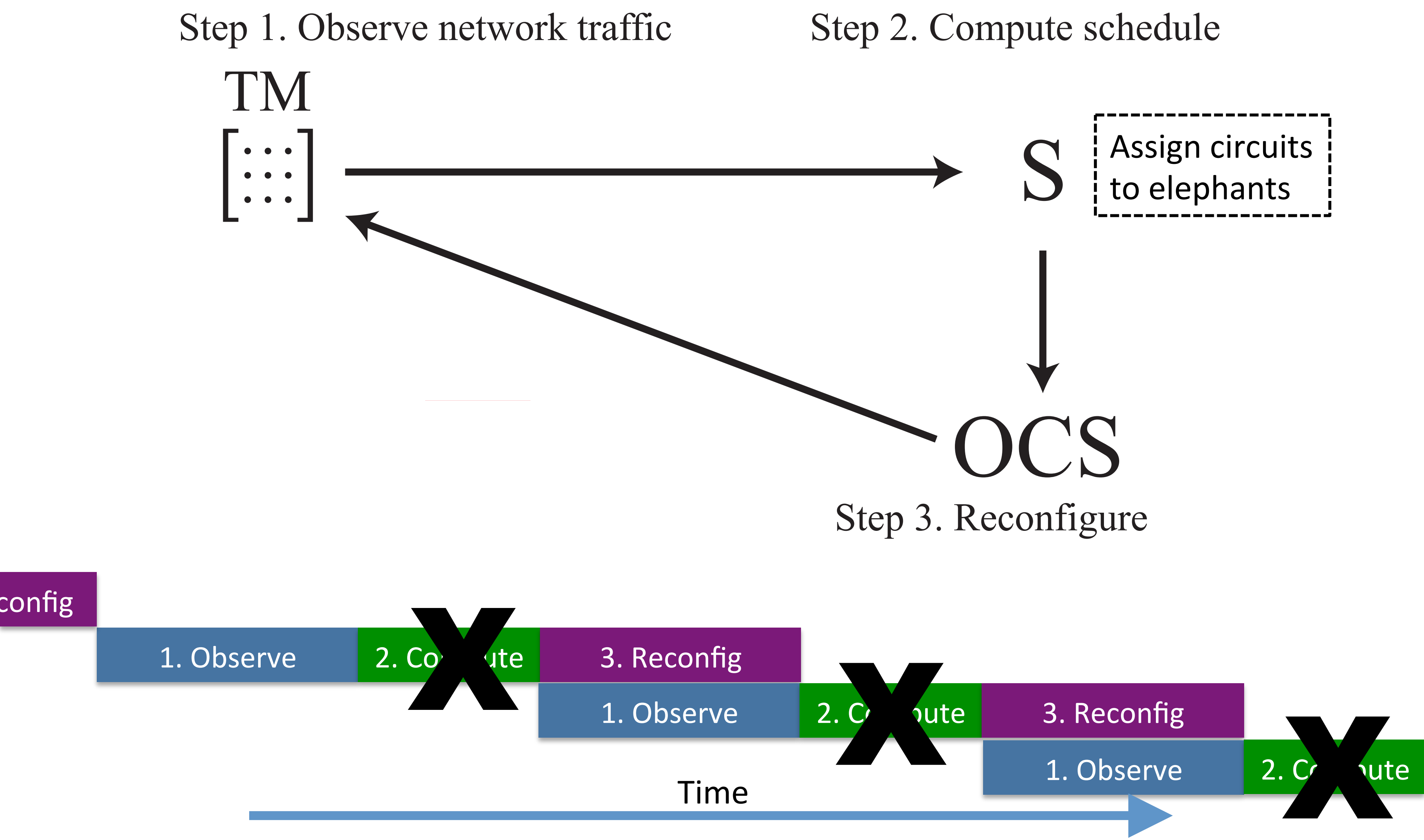




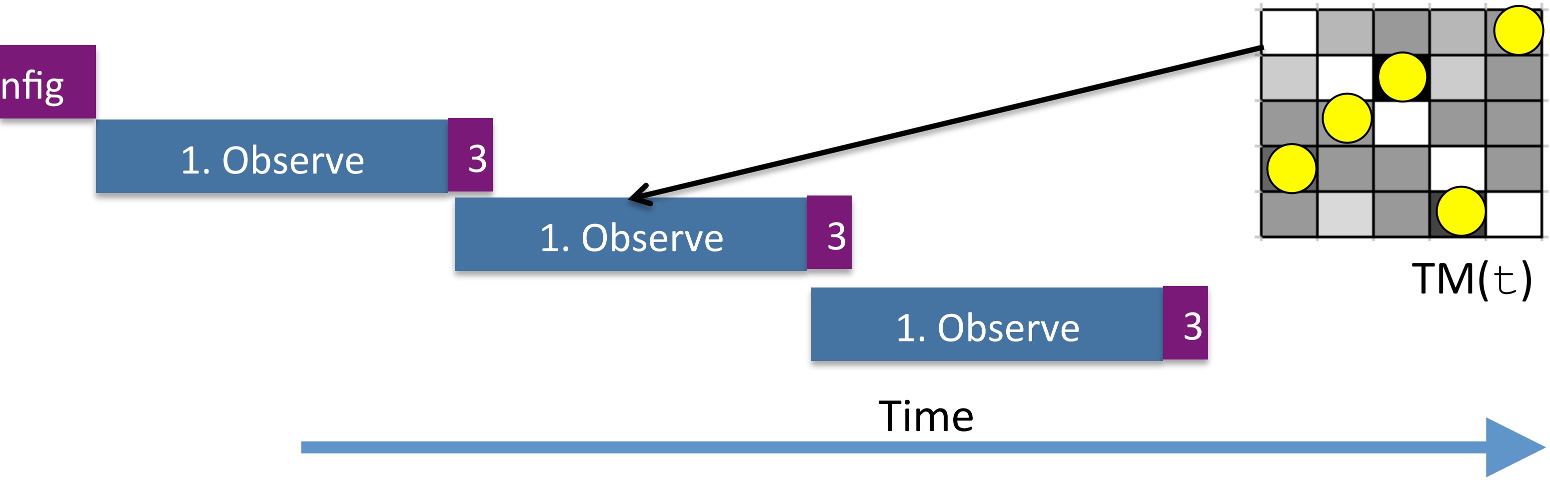
Bi-partite graph

- Directly connects inputs to outputs
- Reconfiguration time: 10us
  - "Night" time (Tn): no traffic during reconfiguration
  - "Day" time (Td): circuits/mapping established
- Duty cycle: Td / (Td+Tn)

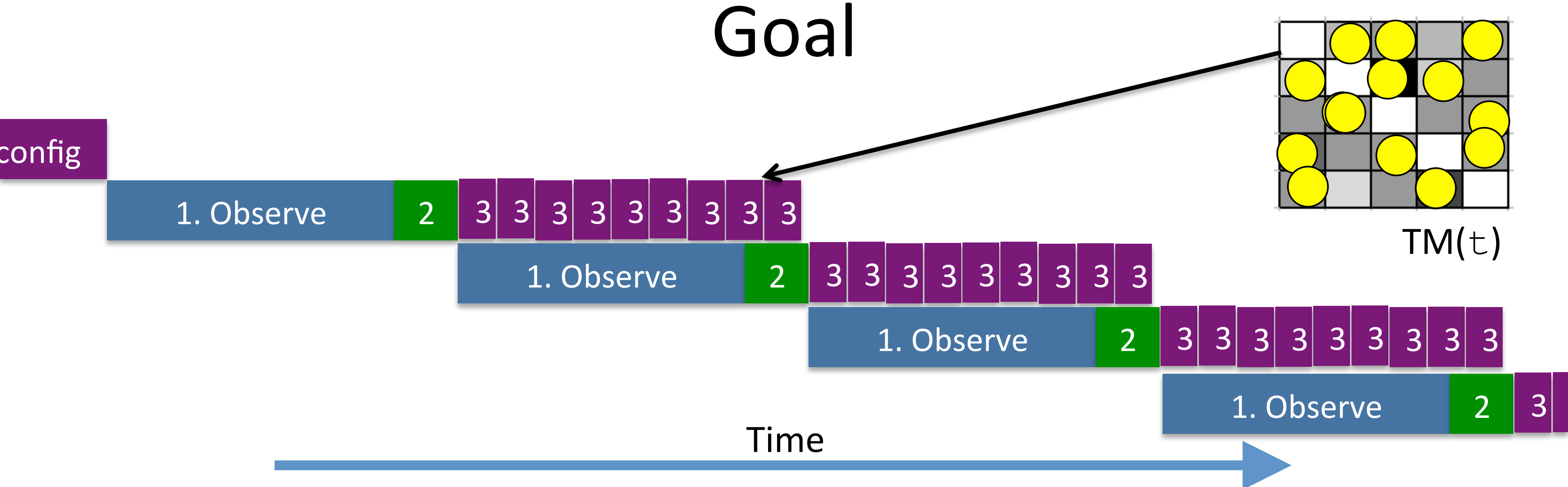
Previous approaches: Hotspot Scheduling



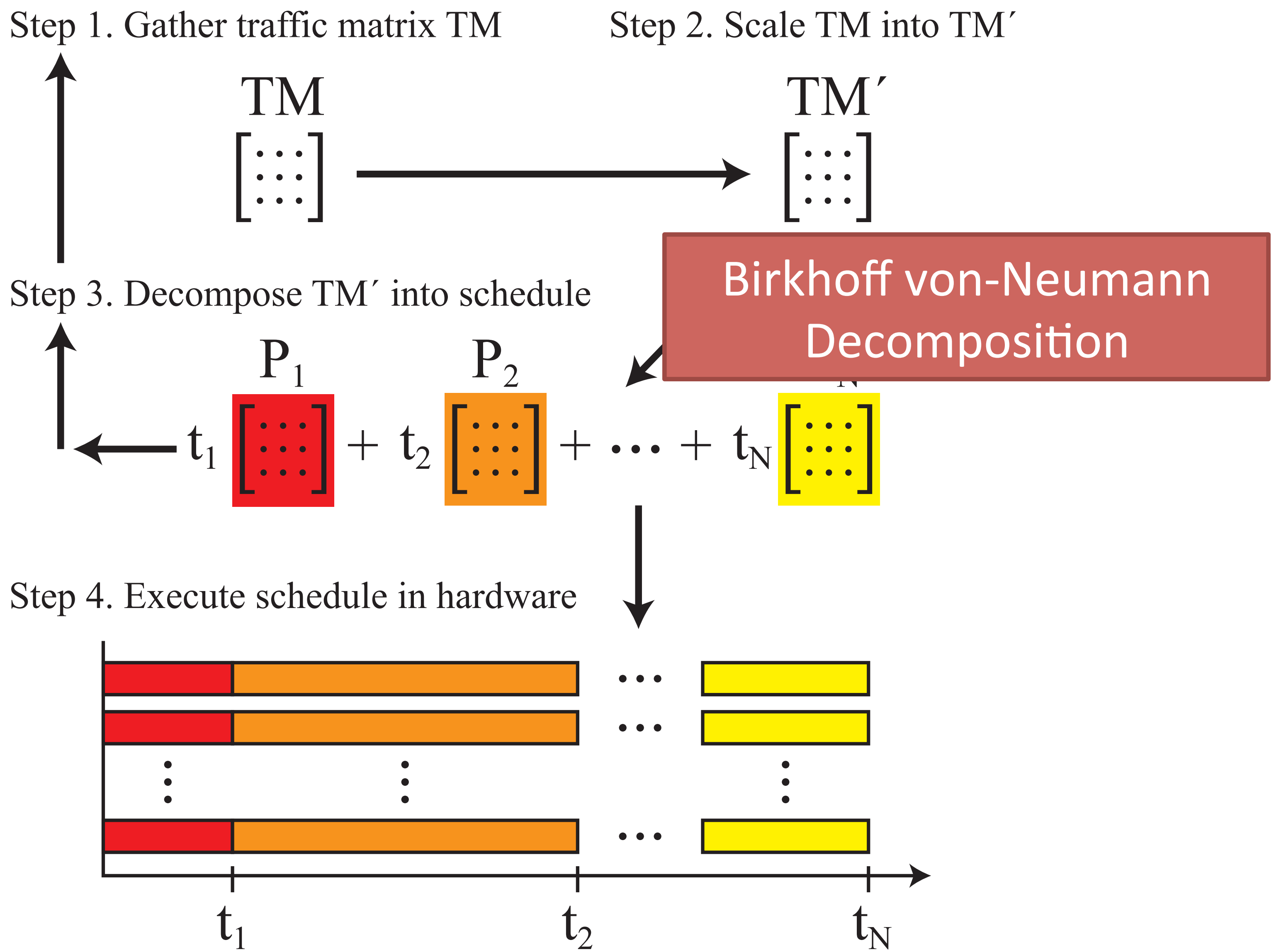
Limitations of Hotspot Scheduling



Goal



Traffic Matrix Scheduling



BvN Decomposition

$\exists \quad (\alpha_1, P_1), (\alpha_2, P_2), \ldots, (\alpha_{k'}, P_{k'})$ s.t.

$$T = \alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_{k'} P_{k'}$$

T has to be doubly-stochastic

k' could be large ($\Omega(n^2)$in worst case)

- Suppose: T is a scaled doubly-stochastic matrix

Scheduling

circuit switch configuration: bipartite graph matching

Traffic Matrix: T

$$\begin{bmatrix} \circ & 1 & \circ & 4 & \circ \\ \circ & \circ & 1 & \circ & 4 \\ \circ & 4 & \circ & 1 & \circ \\ 4 & \circ & \circ & \circ & 1 \\ 1 & \circ & 4 & \circ & \circ \end{bmatrix}$$

n = 5 nodes

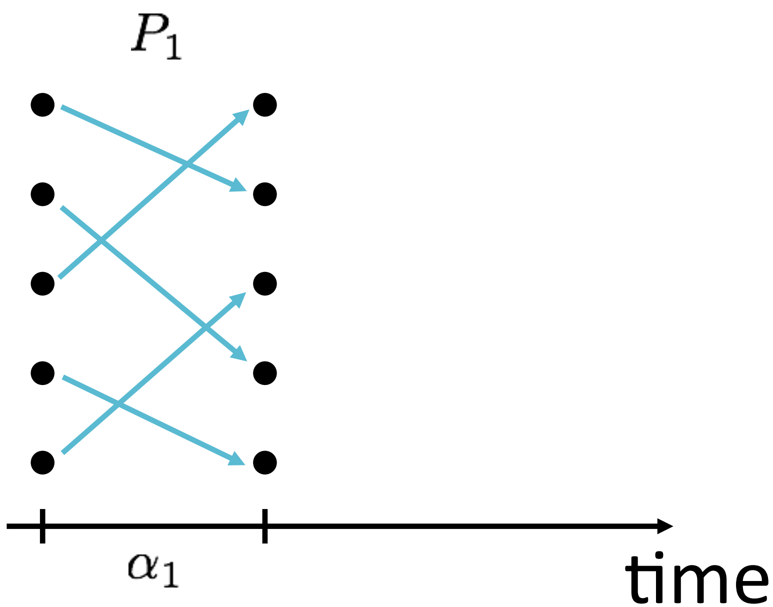
Scheduling

configuration of circuit switch modeled as bipartite graph matching

Traffic Matrix: T

$$\begin{bmatrix} \circ & 1 & \circ & 4 & \circ \\ \circ & \circ & 1 & \circ & 4 \\ \circ & 4 & \circ & 1 & \circ \\ 4 & \circ & \circ & \circ & 1 \\ 1 & \circ & 4 & \circ & \circ \end{bmatrix}$$

n = 5 nodes



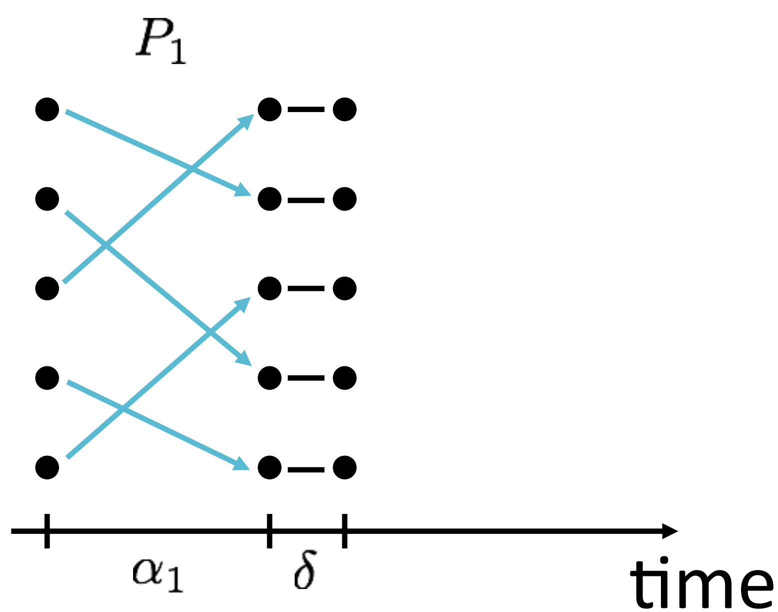
Scheduling

configuration of circuit switch modeled as bipartite graph matching

Traffic Matrix: T

$$\begin{bmatrix} \cdot & 1 & \cdot & 0 & \cdot \\ \cdot & \cdot & 1 & \cdot & 0 \\ \cdot & 0 & \cdot & 1 & \cdot \\ 0 & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & 0 & \cdot & \cdot \end{bmatrix}$$

n = 5 nodes



reconfiguration delay

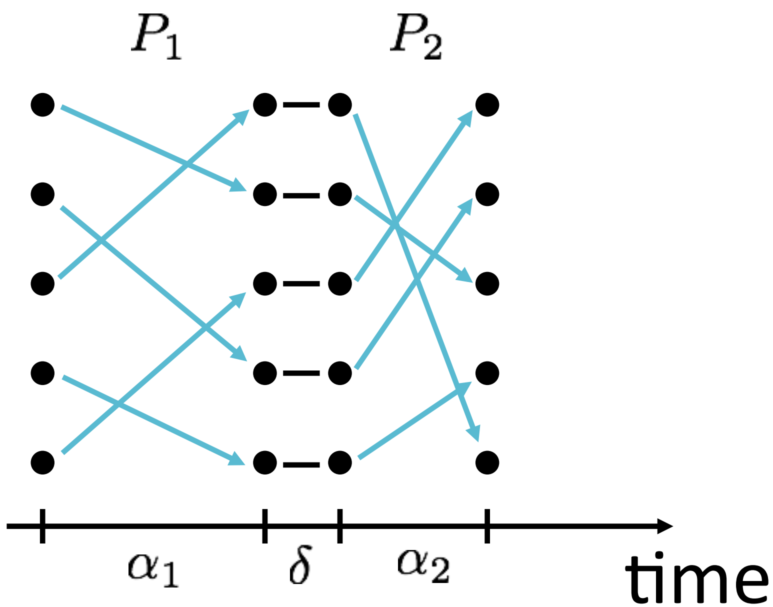
Scheduling

configuration of circuit switch modeled as bipartite graph matching

Traffic Matrix: T

$$\begin{bmatrix} \cdot & 1 & \cdot & 0 & \cdot \\ \cdot & \cdot & 1 & \cdot & 0 \\ \cdot & 0 & \cdot & 1 & \cdot \\ 0 & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & 0 & \cdot & \cdot \end{bmatrix}$$

n = 5 nodes



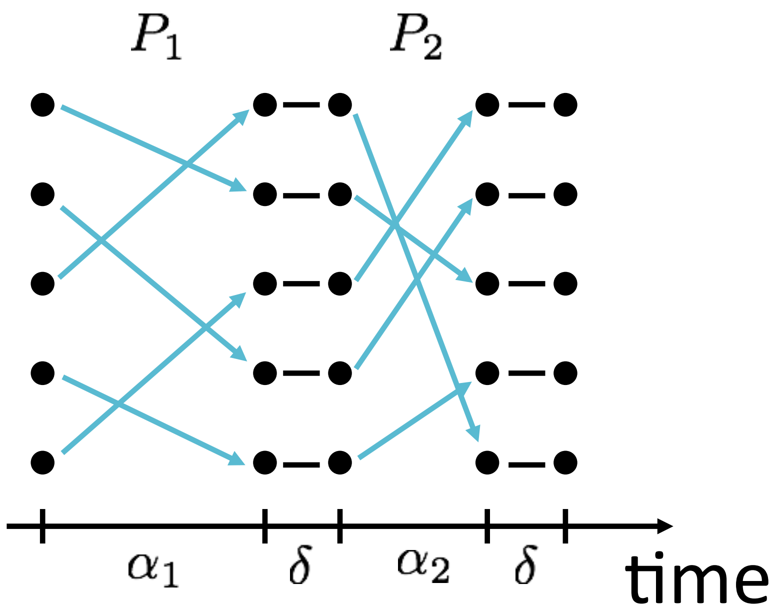
Scheduling

configuration of circuit switch modeled as bipartite graph matching

Traffic Matrix: T

$$\begin{bmatrix} \cdot & \mathbf{0} & \cdot & 0 & \cdot \\ \cdot & \cdot & \mathbf{0} & \cdot & 0 \\ \cdot & 0 & \cdot & \mathbf{0} & \cdot \\ 0 & \cdot & \cdot & \cdot & \mathbf{0} \\ \mathbf{0} & \cdot & 0 & \cdot & \cdot \end{bmatrix}$$

n = 5 nodes



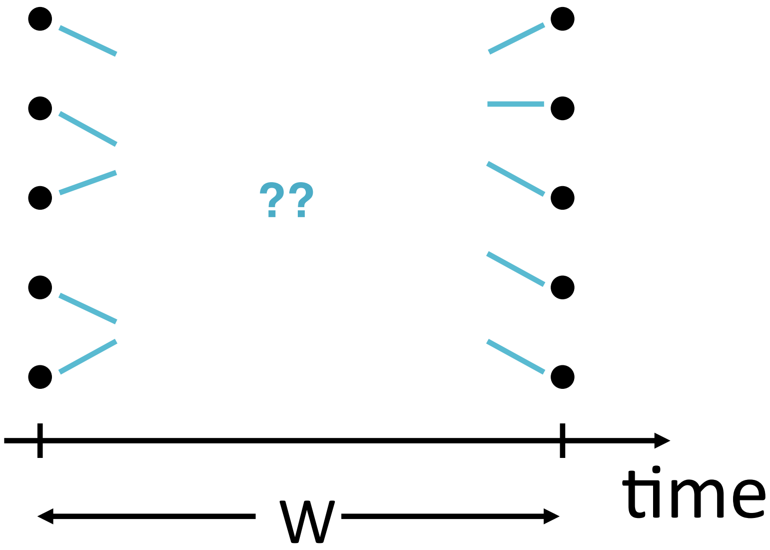
Scheduling

maximize throughput in time-window W

Traffic Matrix: T

$$\begin{bmatrix} \cdot & 1 & \cdot & 4 & \cdot \\ \cdot & \cdot & 1 & \cdot & 4 \\ \cdot & 4 & \cdot & 1 & \cdot \\ 4 & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & 4 & \cdot & \cdot \end{bmatrix}$$

n = 5 nodes



Problem Statement

maximize $\left\| \min\left(\sum_{i=1}^{k} \alpha_i P_i, T \right) \right\|_1$

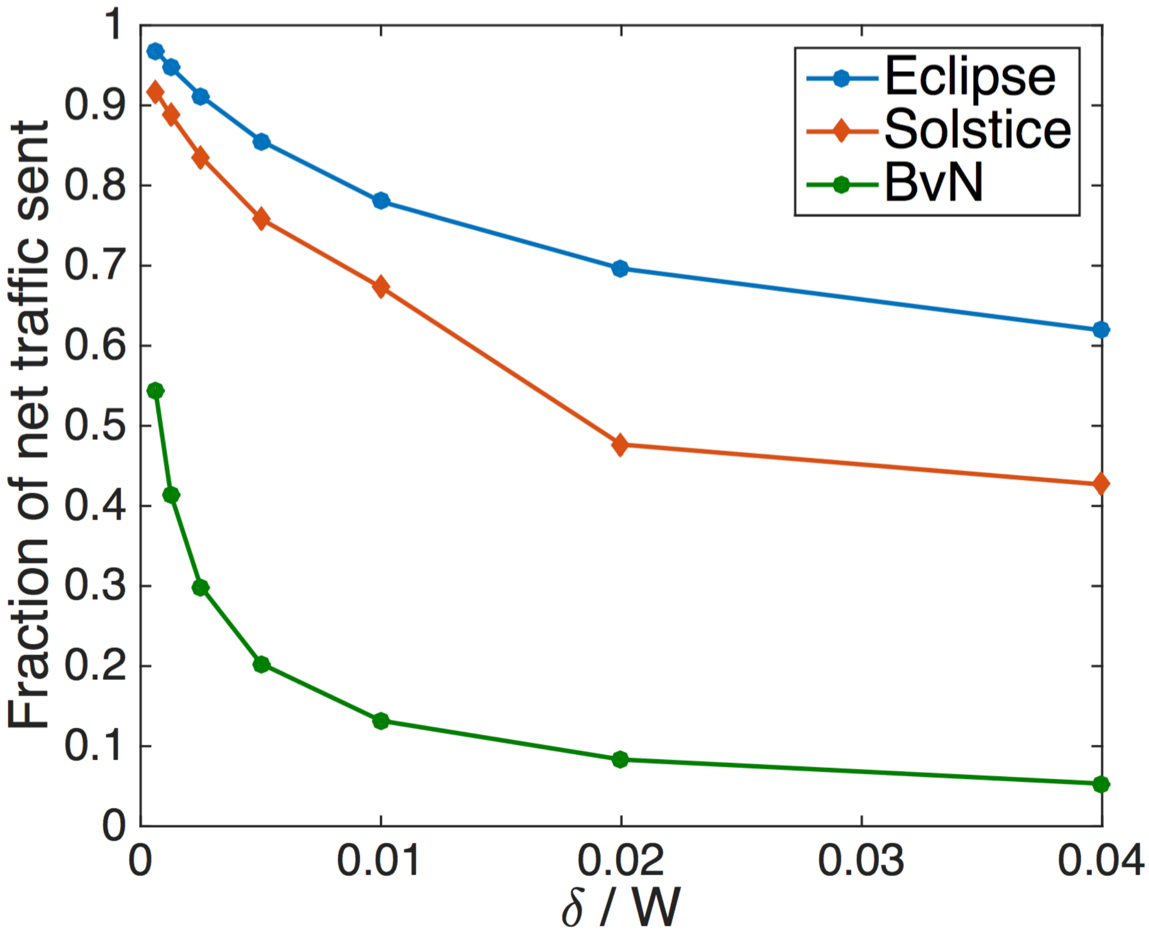
s.t. $\alpha_1 + \alpha_2 + \ldots + \alpha_k + k\delta \leq W$

number of matchings → $k \in \mathbb{N}$

permutation matrices → $P_1, \ldots, P_k \in \mathcal{P}$

duration → $\alpha_1, \ldots, \alpha_k \geq 0$

Eclipse: Greedy Algorithm (with provable guarantees)



- Venkatakrishnan et al., "Costly Circuits, Submodular Schedules, Hybrid Switch Scheduling for Data Centers", To appear in SIGMETRICS 2016.

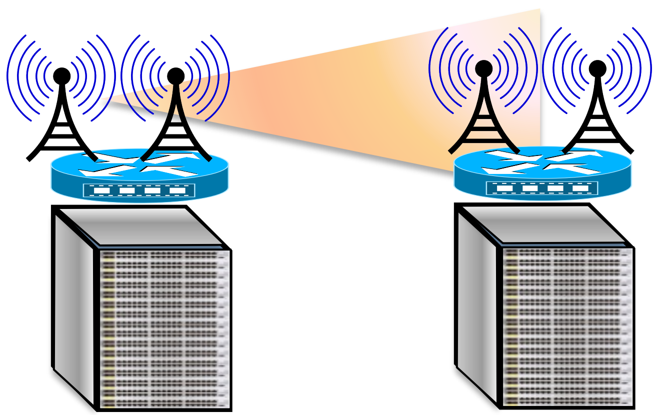
Discussion

Firefly

✧ Slides based on presentation by Vyas Sekar (CMU)

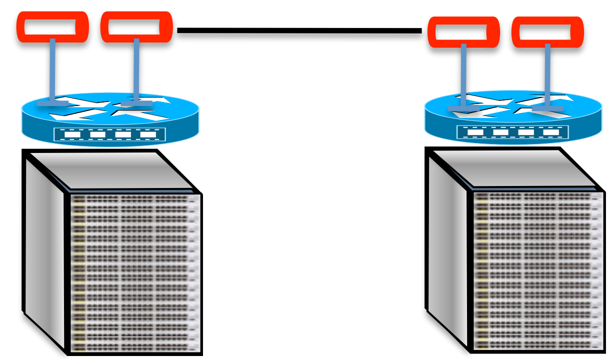
Why FSO instead of RF?

RF (e.g. 60GHZ)



Wide beam ➔
Faster steering of beams
High interference
Limited active links
Limited Throughput

FSO (Free Space optical)



Narrow beam ➔
Slow steering of beams
Zero interference
No limit on active links
High Throughput

Today's FSO

Cost: $15K per FSO

Size: 3 ft^3

Power: 30w

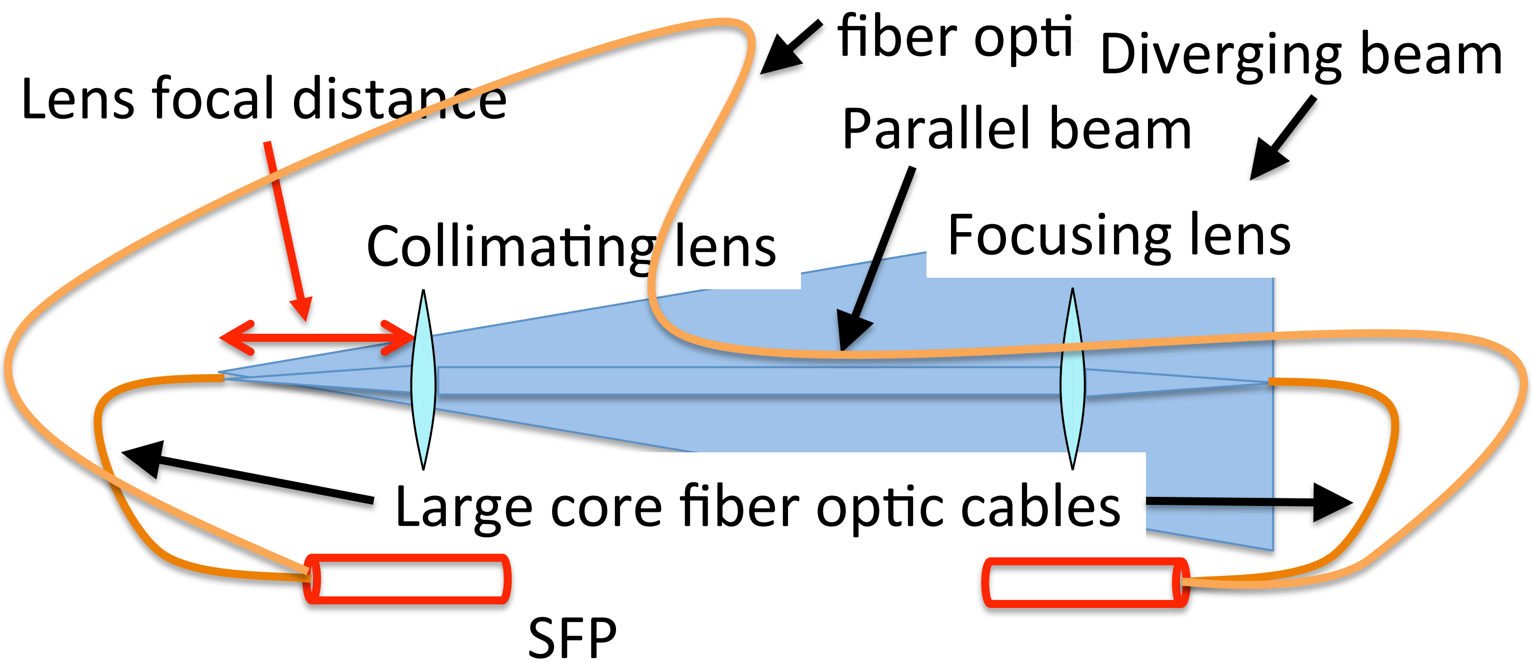
Non steerable

- Current: bulky, power-hungry, and expensive
- <u>Required</u>: small, low power and low expense

Why Size, Cost, Power Can be Reduced?

- Traditional use : outdoor, long haul
  - High power
  - Weatherproof
- Data centers: indoor, short haul
- Feasible roadmap via commodity fiber optics
  - E.g. Small form transceivers (Optical SFP)

FSO Design Overview



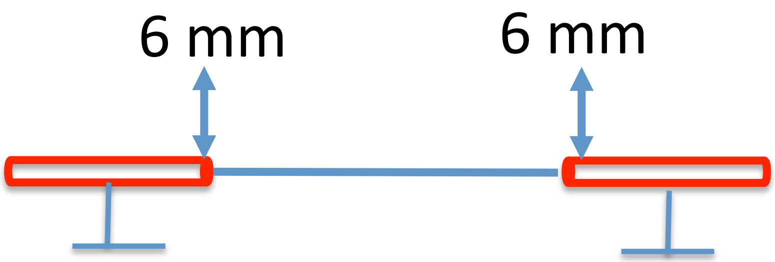
- large cores (> 125 microns) are more robust

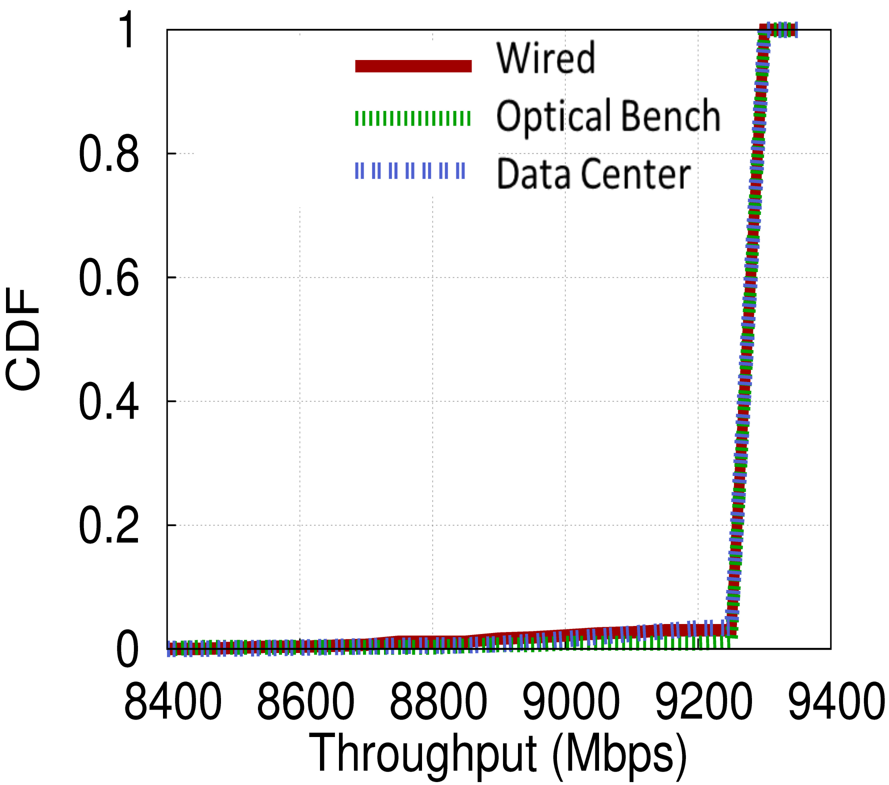
FSO Link Performance

Effect of vibrations, etc.

6mm movement tolerance

Range up to 24m tested





FSO link is as robust as a wired link

Steerability

<u>Shortcomings of current FSOs</u>

✓Cost

✓Size

✓Power

•Not Steerable

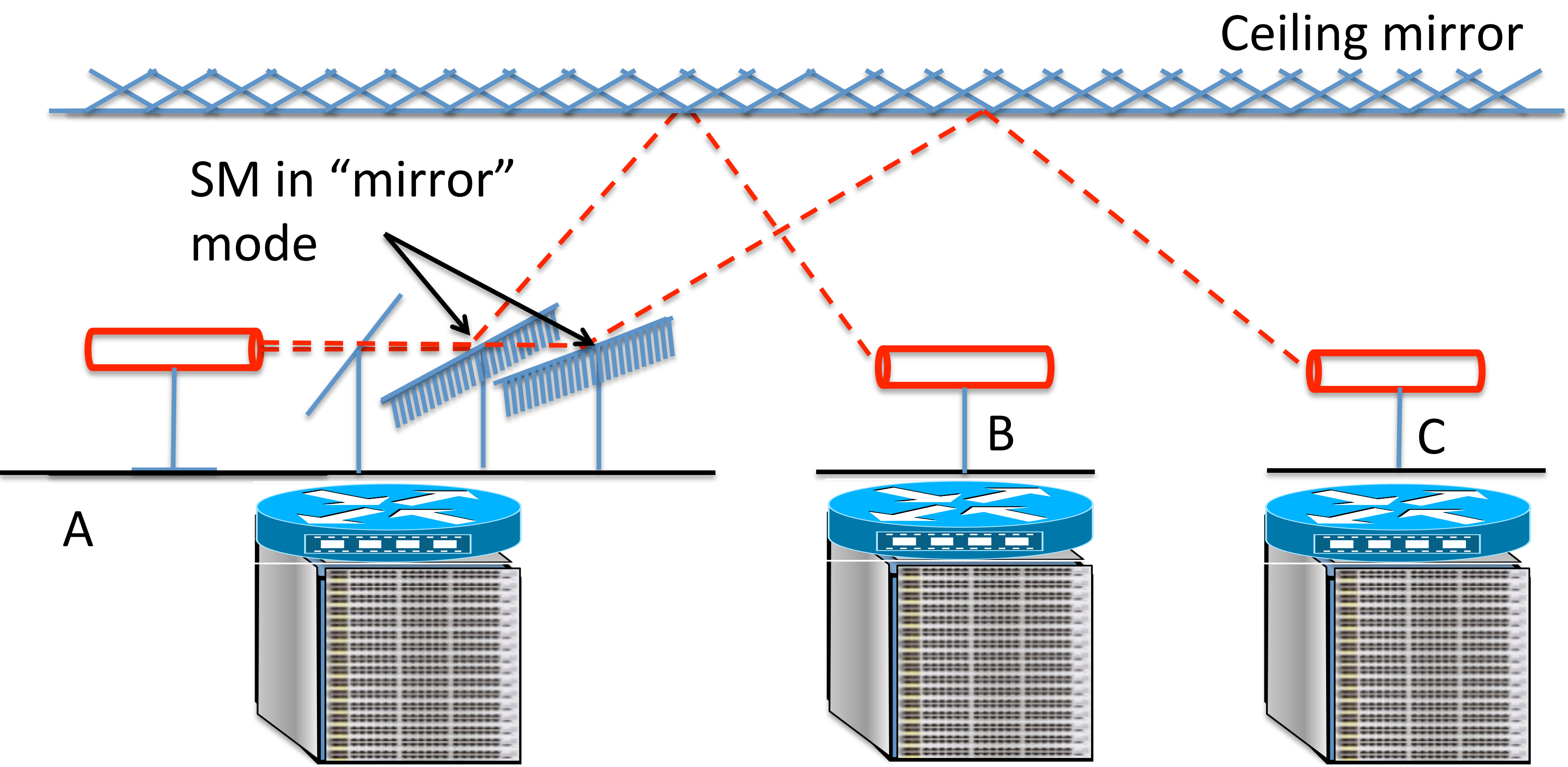
FSO design using SFP

Via Switchable mirrors or Galvo mirrors

<u>Shortcomings of current FSOs</u>

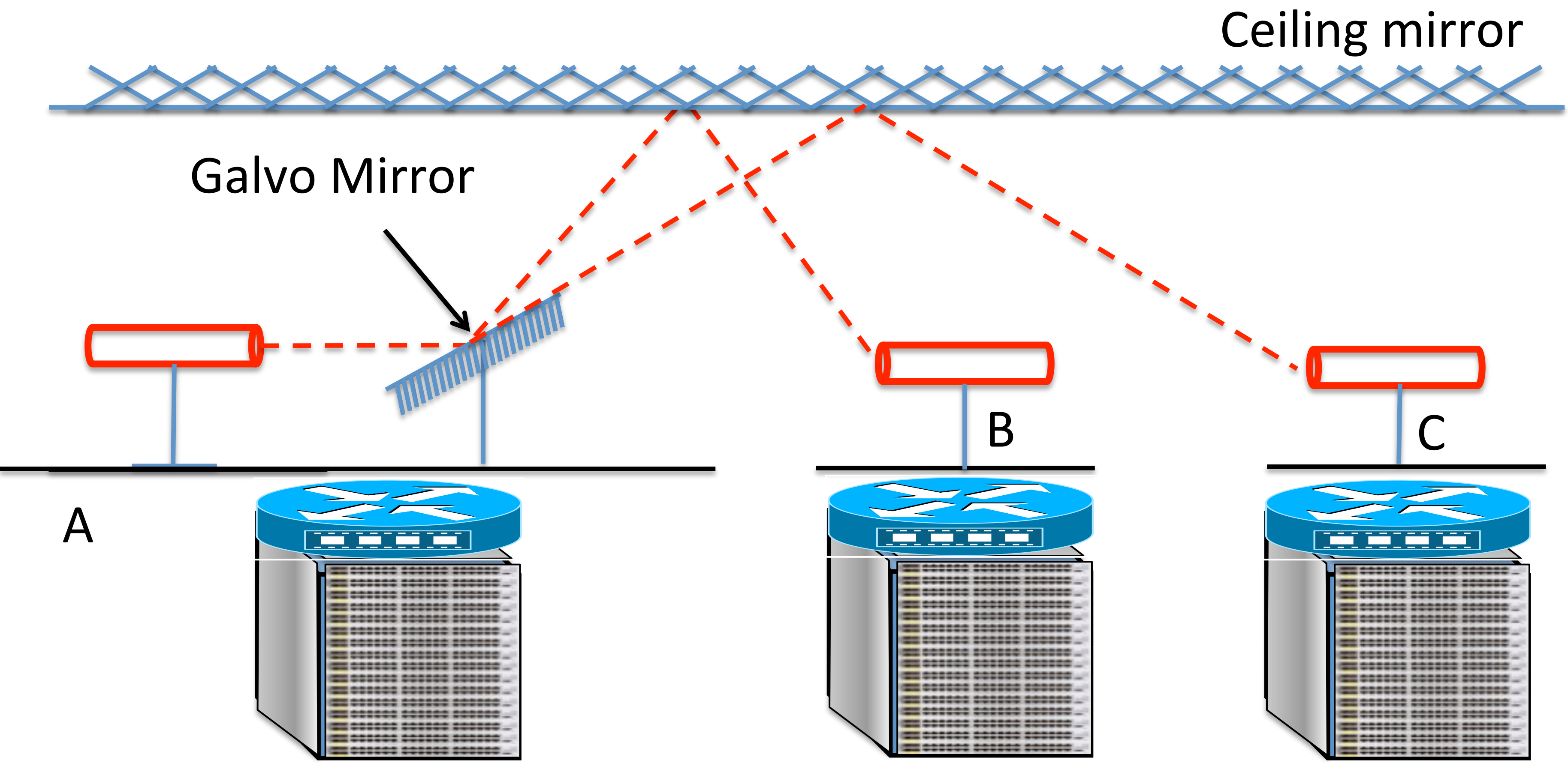
Steerability via Switchable Mirror

- Switchable Mirror: glass ⟷ mirror
- Electronic control, low latency



Steerability via Galvo Mirror

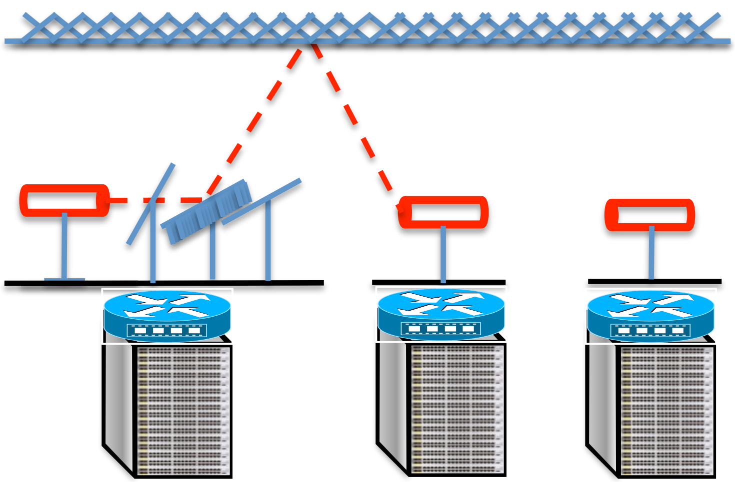
- Galvo Mirror: small rotating mirror
- Very low latency



How to design FireFly network?

Goals: Robustness to current and future traffic

Budget & Physical Constraints



Design parameters

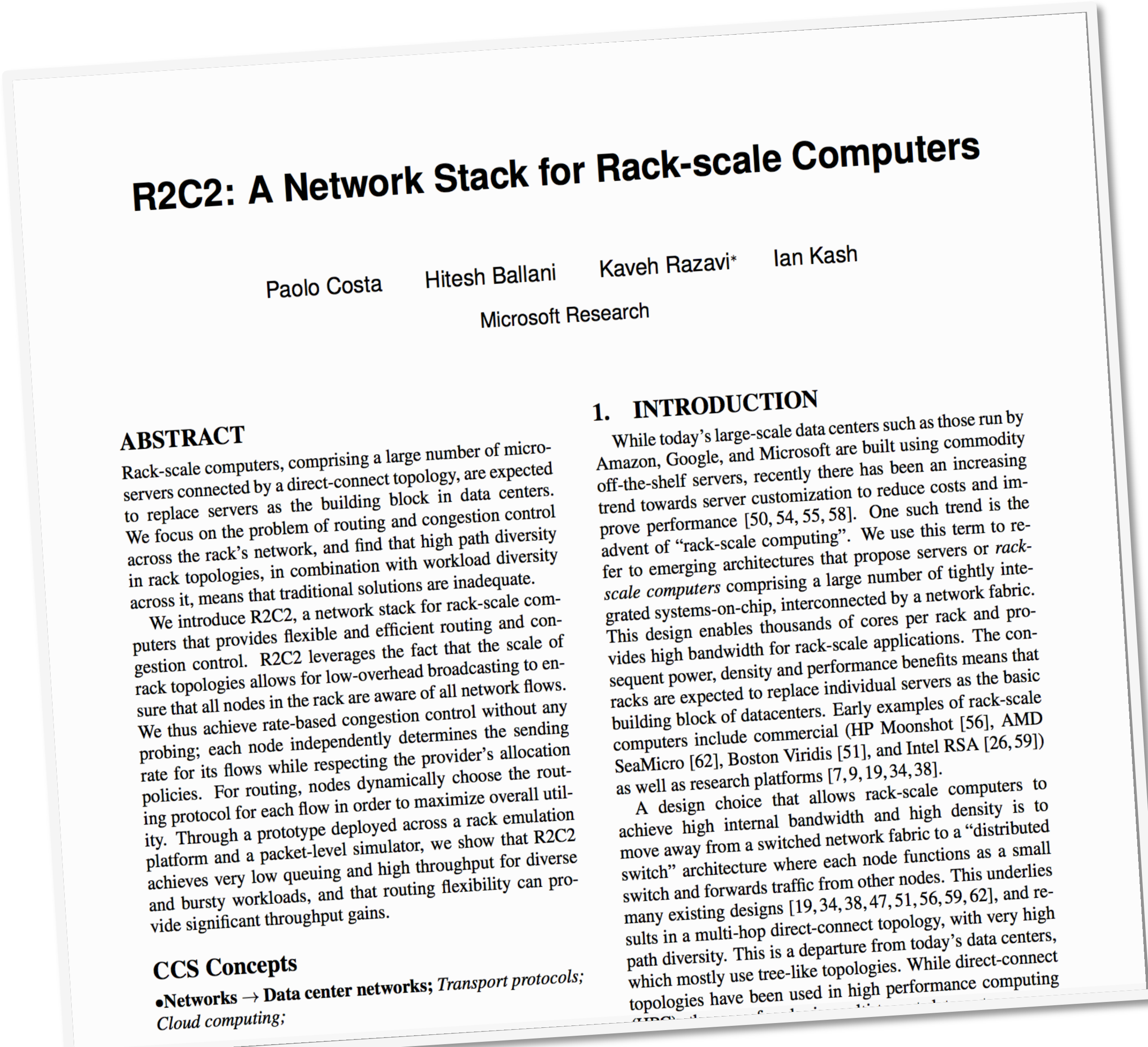
- Number of FSOs?
- Number of steering mirrors?
- Initial mirrors' configuration

Performance metric

- Dynamic bisection bandwidth

Discussion

Next Time: Rack-Scale Computing



R2C2: A Network Stack for Rack-scale Computers

Paolo Costa Hitesh Ballani Kaveh Razavi* Ian Kash

Microsoft Research

ABSTRACT

Rack-scale computers, comprising a large number of micro-servers connected by a direct-connect topology, are expected to replace servers as the building block in data centers. We focus on the problem of routing and congestion control across the rack's network, and find that high path diversity in rack topologies, in combination with workload diversity across it, means that traditional solutions are inadequate.

We introduce R2C2, a network stack for rack-scale computers that provides flexible and efficient routing and congestion control. R2C2 leverages the fact that the scale of rack topologies allows for low-overhead broadcasting to ensure that all nodes in the rack are aware of all network flows. We thus achieve rate-based congestion control without any probing; each node independently determines the sending rate for its flows while respecting the provider's allocation policies. For routing, nodes dynamically choose the routing protocol for each flow in order to maximize overall utility. Through a prototype deployed across a rack emulation platform and a packet-level simulator, we show that R2C2 achieves very low queuing and high throughput for diverse and bursty workloads, and that routing flexibility can provide significant throughput gains.

CCS Concepts

•**Networks → Data center networks;** *Transport protocols; Cloud computing;*

1. INTRODUCTION

While today's large-scale data centers such as those run by Amazon, Google, and Microsoft are built using commodity off-the-shelf servers, recently there has been an increasing trend towards server customization to reduce costs and improve performance [50, 54, 55, 58]. One such trend is the advent of "rack-scale computing". We use this term to refer to emerging architectures that propose servers or *rack-scale computers* comprising a large number of tightly integrated systems-on-chip, interconnected by a network fabric. This design enables thousands of cores per rack and provides high bandwidth for rack-scale applications. The consequent power, density and performance benefits means that racks are expected to replace individual servers as the basic building block of datacenters. Early examples of rack-scale computers include commercial (HP Moonshot [56], AMD SeaMicro [62], Boston Viridis [51], and Intel RSA [26, 59]) as well as research platforms [7, 9, 19, 34, 38].

A design choice that allows rack-scale computers to achieve high internal bandwidth and high density is to move away from a switched network fabric to a "distributed switch" architecture where each node functions as a small switch and forwards traffic from other nodes. This underlies many existing designs [19, 34, 38, 47, 51, 56, 59, 62], and results in a multi-hop direct-connect topology, with very high path diversity. This is a departure from today's data centers, which mostly use tree-like topologies. While direct-connect topologies have been used in high performance computing