Datacenter Networking

Spring 2024
[cs168.io](cs168.io)

Rob Shakir

Thanks to Ankit Singla and Murphy McCauley for some of the material!

Recall – where is the Internet?

- Carrier hotel locations.
- Generally for interconnection between networks.
- Some smaller application hosting.
- Where do large applications live?

A Datacenter

Google datacenter in Belgium - https://www.google.com/about/datacenters/gallery/

Inside a (Google) Datacenter

Server racks in a Google datacenter - https://www.google.com/about/datacenters/gallery/

Infrastructure in a Google Datacenter

Cooling infrastructure in a Google datacenter - https://www.google.com/about/datacenters/gallery/

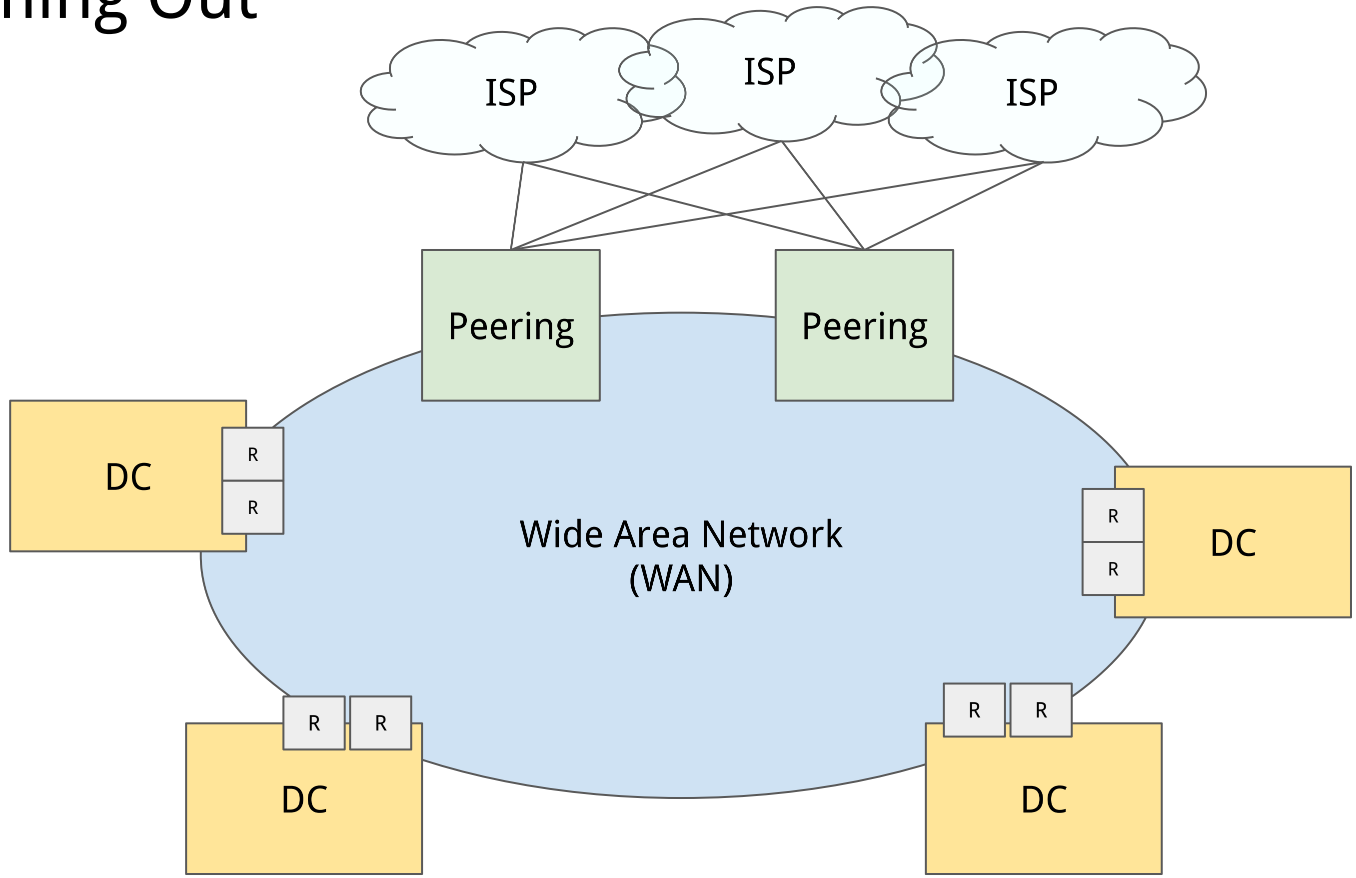
Datacenters

- Computing infrastructure, located in one physical location.

- Owned by one organisation.
- But used by multiple users and applications.

- Our focus: modern hyperscale datacenters.
  - Google, Facebook, Microsoft, Meta…
  - Concept scales down.

Zooming Out


ISP
ISP
ISP
Peering
Peering
DC
R
R
Wide Area Network
(WAN)
R
R
DC
R
R
DC
R
R
DC

Anatomy of an Application/Cloud Provider

- Data center locations – host servers and application infrastructure.
  - Often huge power requirements.
  - Does not need to be near other networks.

- Peering locations – host network <u>interconnection</u> infrastructure.
  - Typically mostly routers.
  - Needs to be near other networks.

- **W**ide **A**rea **N**etwork - connects the different locations together.
- Datacenter network – within a particular DC facility.

Our focuses

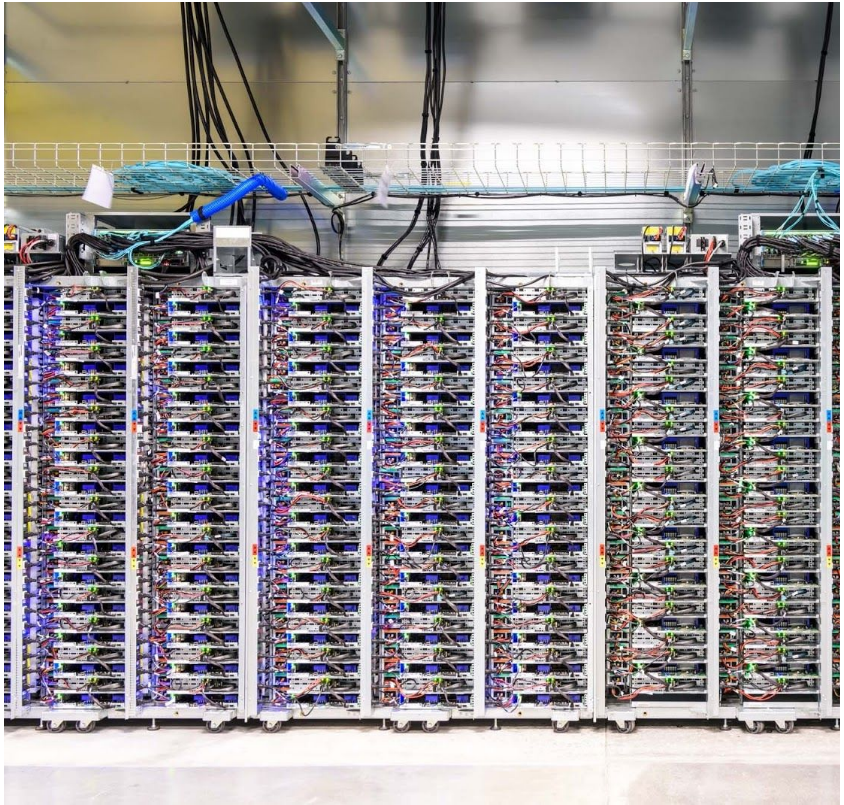
- What does a datacenter network look like?

- What makes a datacenter different to the wide area networks we have discussed thus far?

- Specific solutions for datacenter networking.
  - Congestion control.
  - Routing in datacenters [next time].

Questions?

Anatomy of a Datacenter



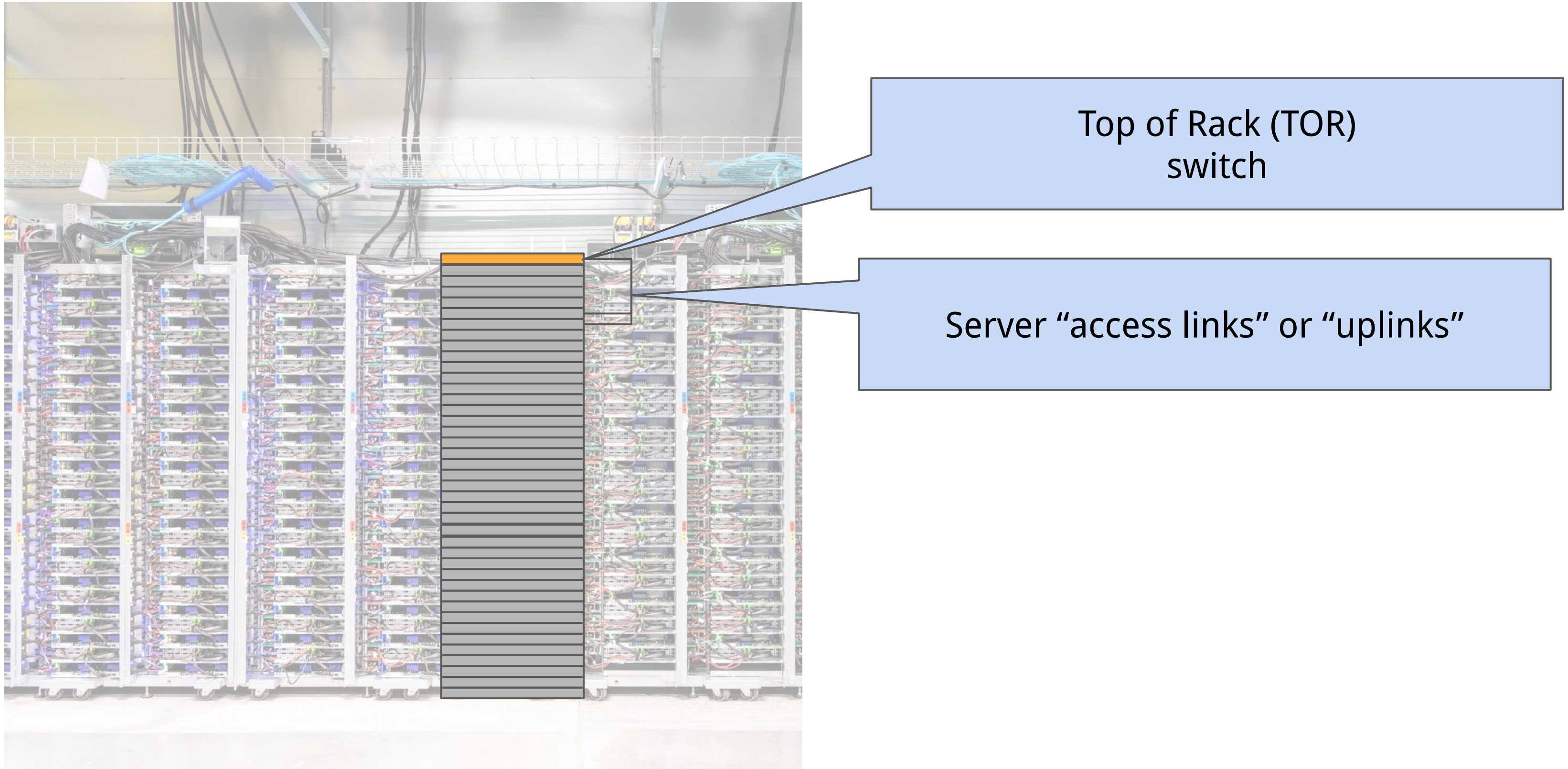
Anatomy of a Datacenter

1-2 servers per “U” [0]

Anatomy of a Datacenter

~40 “U” per rack.

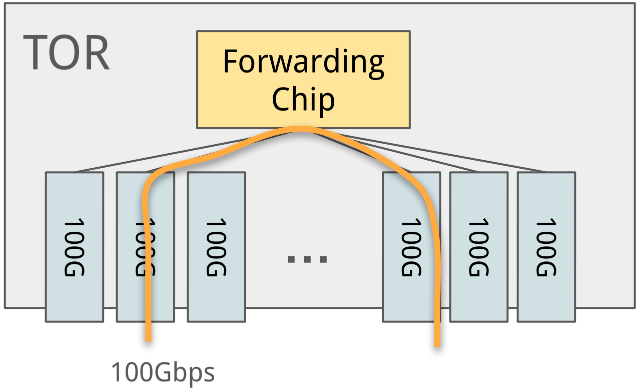
Anatomy of a Datacenter



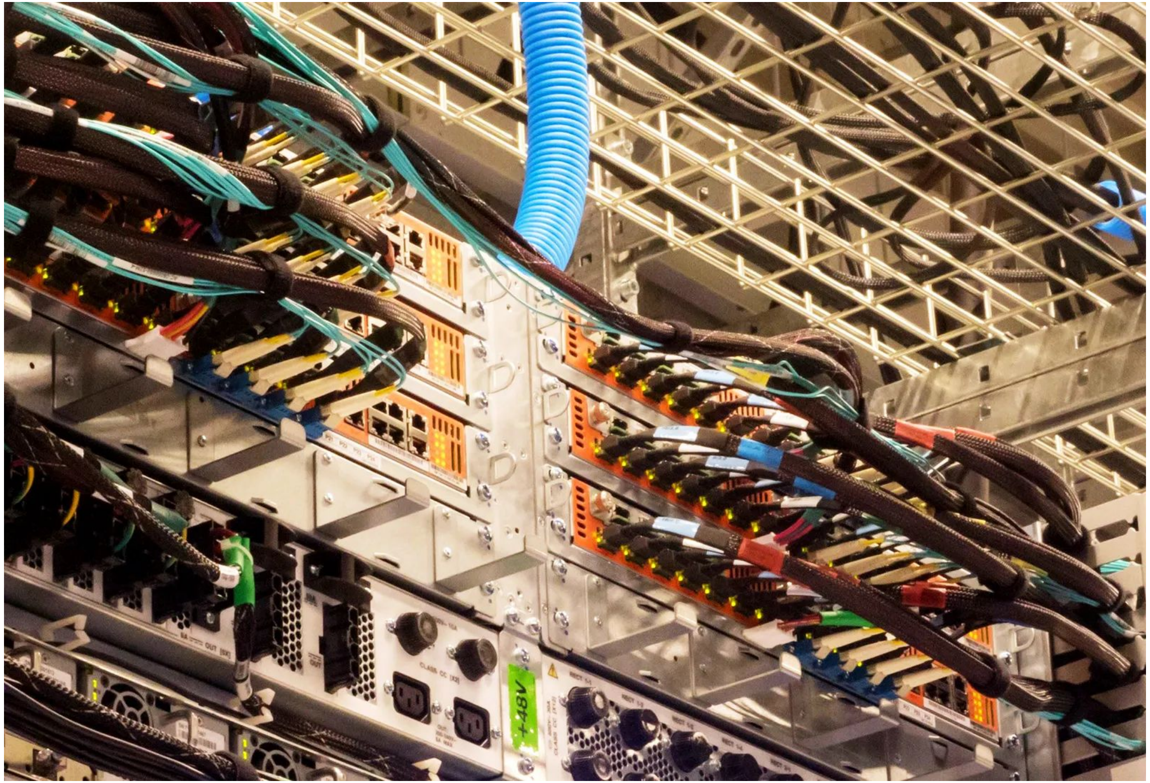
Anatomy of a Datacenter

Top of Rack (TOR) switch

Server “access links” or “uplinks”

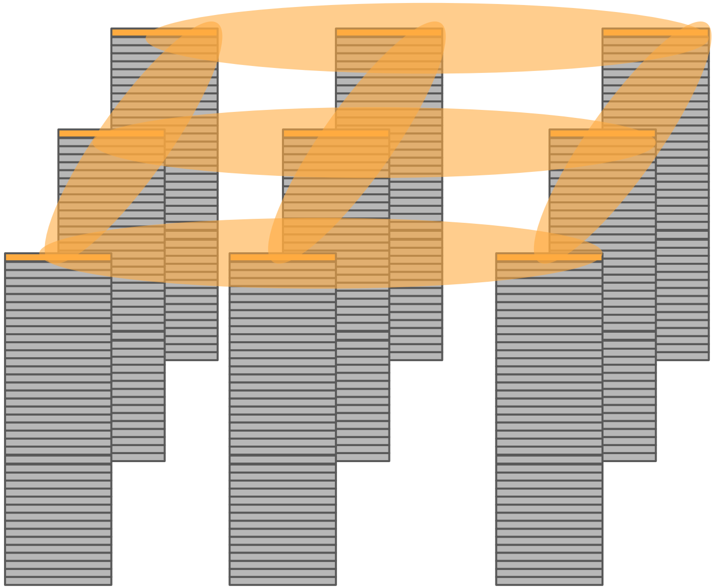


Top-of-Rack Switch



Google “pluto” TOR - - ~2015 – [Wired]

Anatomy of a Datacenter



- 40-80 servers per rack.
- 100Gbps per server.
- Many racks per datacenter!
- How do we connect racks together?

Why is the datacenter different?

- We have generally been thinking about Wide Area Networks.
- These WANs interconnect to make up the Internet.
- Why might datacenter networks be different?

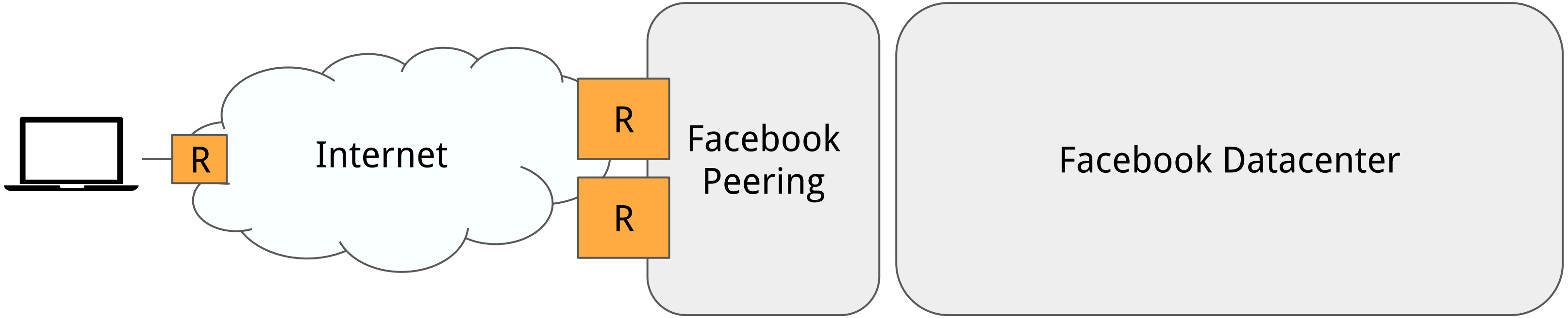
Why is the datacenter different?

- We have generally been thinking about Wide Area Networks.

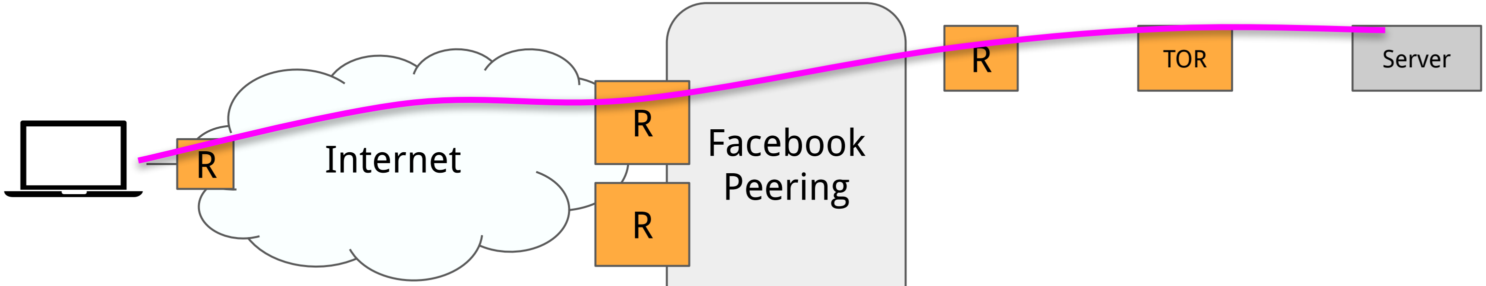
- These WANs interconnect to make up the Internet.

- Why might datacenter networks be different?
    - Run by a single organisation
    - Exist in a single physical location
    - High scale (in that single location!)
    - More control over network and hosts (to some degree)
    - Homogeneous
    - Performance, performance, performance!

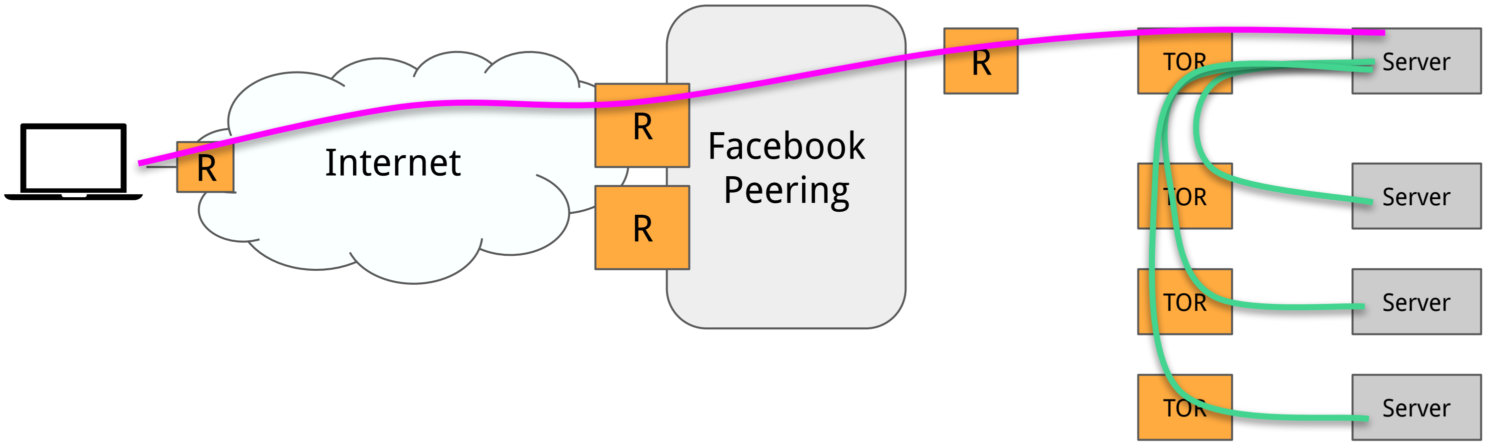
Accessing an Application



Accessing an Application



Accessing an Application



Accessing an Application

USENIX NSDI, 2013

Scaling Memcache at Facebook

Rajesh Nishtala, Hans Fugal, Steven Grimm, Marc Kwiatkowski, Herman Lee, Harry C. Li, Ryan McElroy, Mike Paleczny, Daniel Peek, Paul Saab, David Stafford, Tony Tung, Venkateshwaran Venkataramani

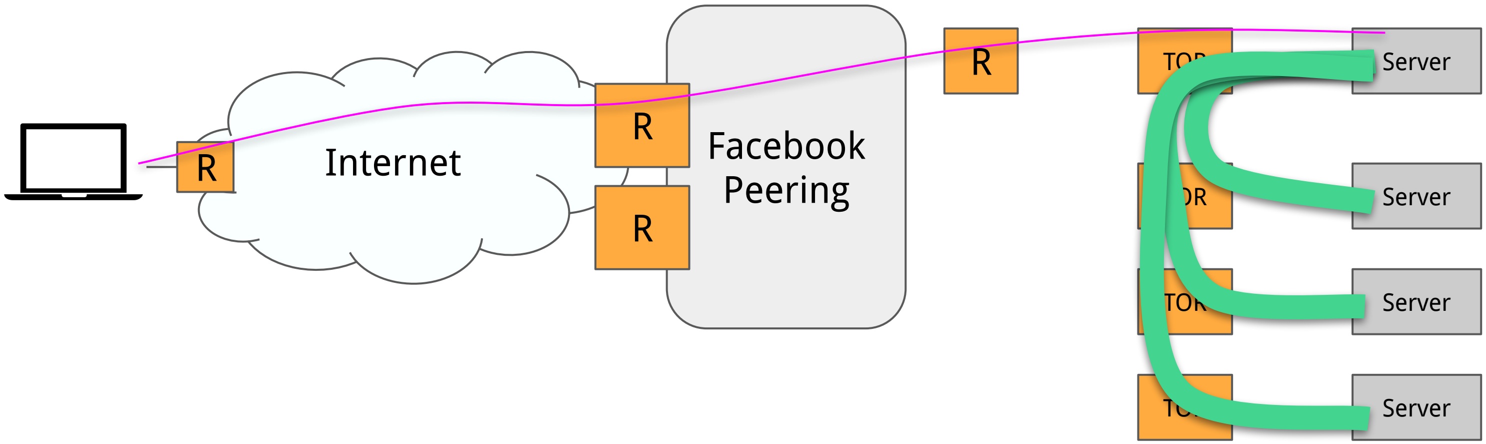
{rajeshn,hans}@fb.com, {sgrimm, marc}@facebook.com, {herman, hcli, rm, mpal, dpeek, ps, dstaff, ttung, veeve}@fb.com

Facebook Inc.

1 popular page loaded = **521** distinct memcache loads

(95th percentile = 1740!)

Accessing an Application

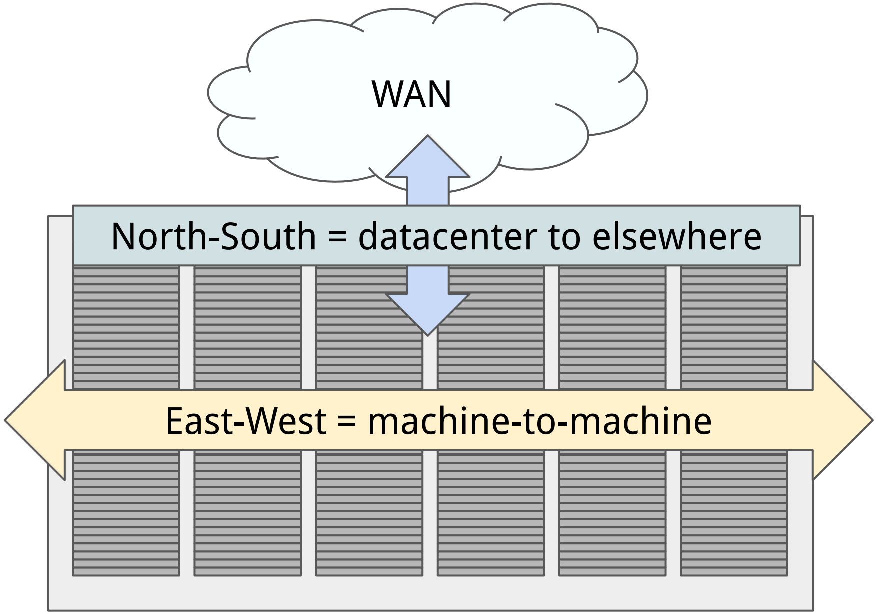


Significantly more inter-machine traffic than “user” to “machine”.

Other Applications

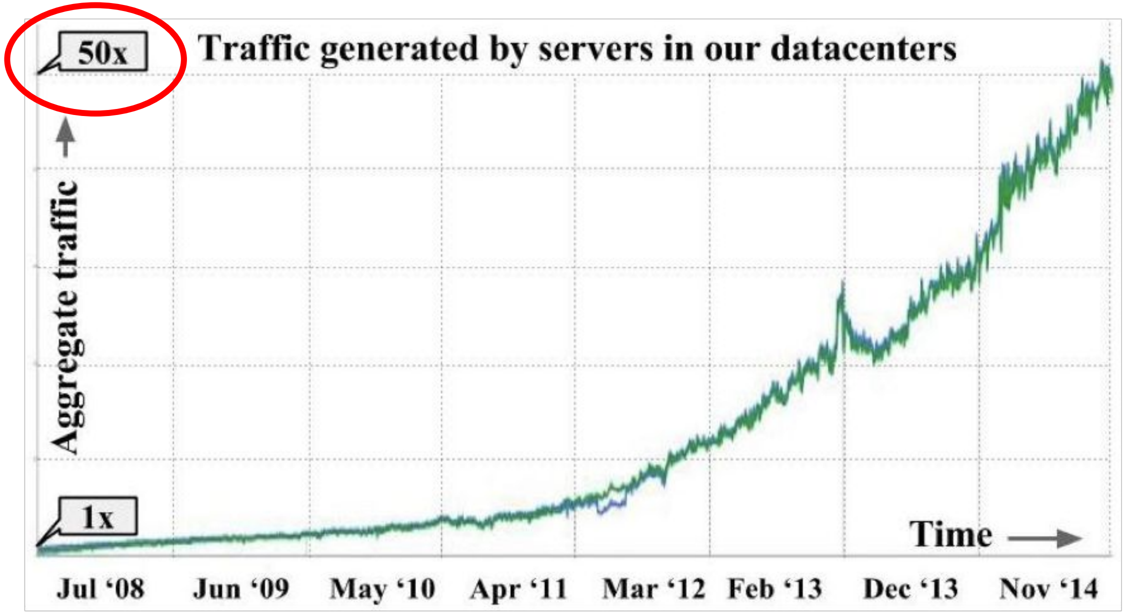
- Big data analytics
  - e.g., mapreduce
- Significantly more traffic between machines - maybe *no* user-facing traffic.

Datacenter Traffic Patterns



East-West traffic is several orders of magnitude larger than North-South.

East-West Traffic Volume



"Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network", Arjun Singh et al. @ Google, ACM SIGCOMM'15

Questions?

How do we support East-West bandwidth?

- Ideally any server can talk to any server at line rate.

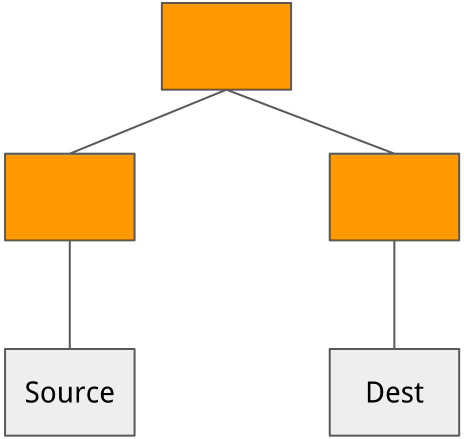
- We want a network with high **bisection bandwidth**.

Bisection Bandwidth

- Pick the number of links we must cut in order to partition a network into two halves.
- Bisection bandwidth is the sum of those bandwidths.

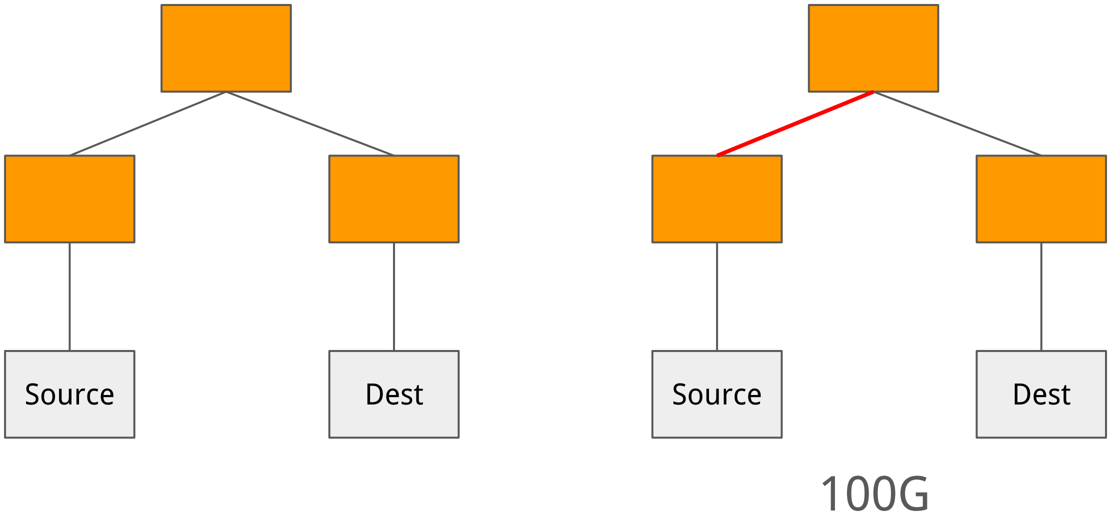
Bisection Bandwidth

- Pick the number of links we must cut in order to partition a network into two halves.
- Bisection bandwidth is the sum of those bandwidths.



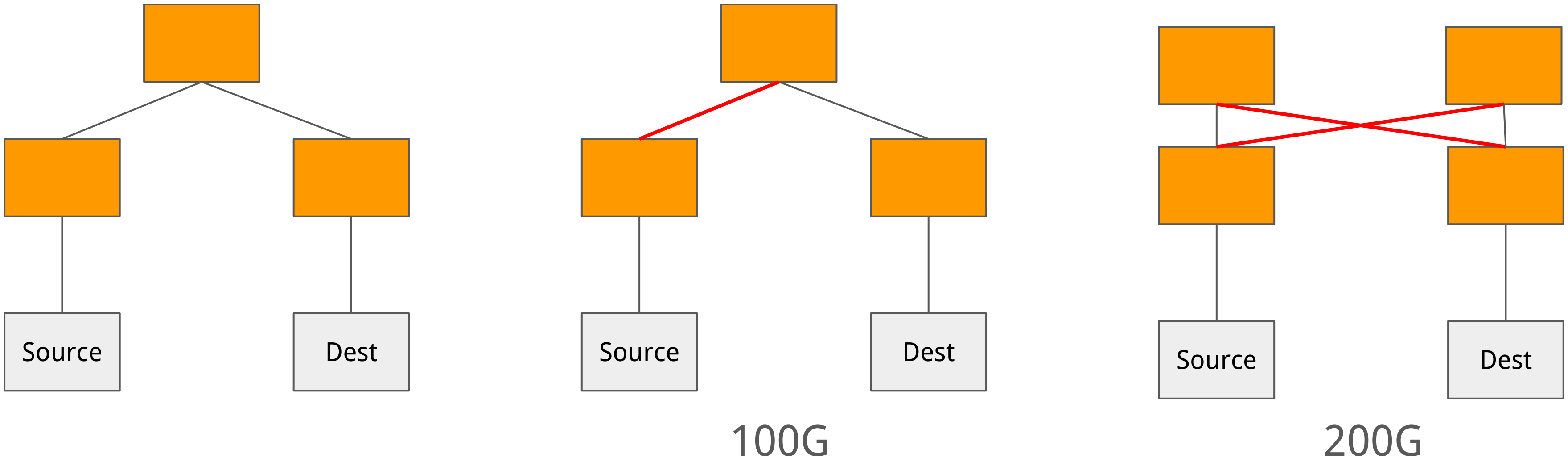
Bisection Bandwidth

- Pick the number of links we must cut in order to partition a network into two halves.
- Bisection bandwidth is the sum of those bandwidths.



Bisection Bandwidth

- Pick the number of links we must cut in order to partition a network into two halves.
- Bisection bandwidth is the sum of those bandwidths.



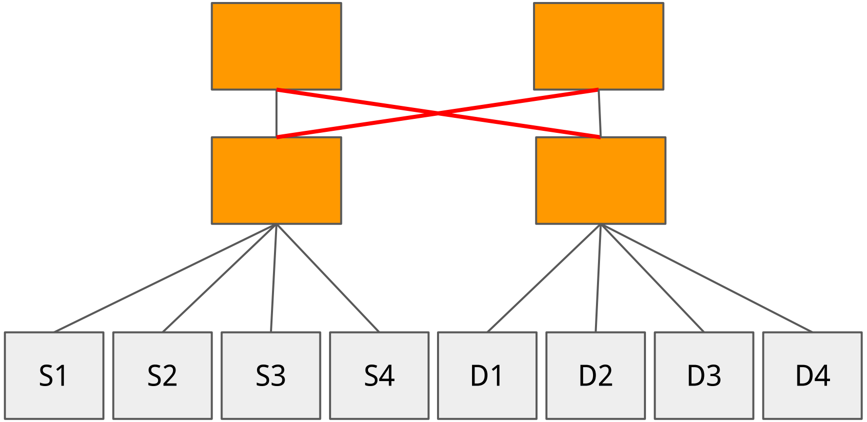
Bisection Bandwidth

- Pick the number of links we must cut in order to partition a network into two halves.
- Bisection bandwidth is the sum of those bandwidths.

- **Full** bisection bandwidth: Nodes in one partition can communicate simultaneously with nodes in the other partition at full rate.
  - Given N nodes, each with access link capacity R, bisection bandwidth = N/2 x R

- <u>Oversubscription</u>, informally, how far from the full bisection bandwidth we are.
  - Formally: ratio of worst-case achievable bandwidth to full bisection bandwidth.

Bisection Bandwidth



Bisection Bandwidth: 200G

Full Bisection Bandwidth: (8/2)*100G = 400G

Oversubscription: 200/400 = 2x

Questions?

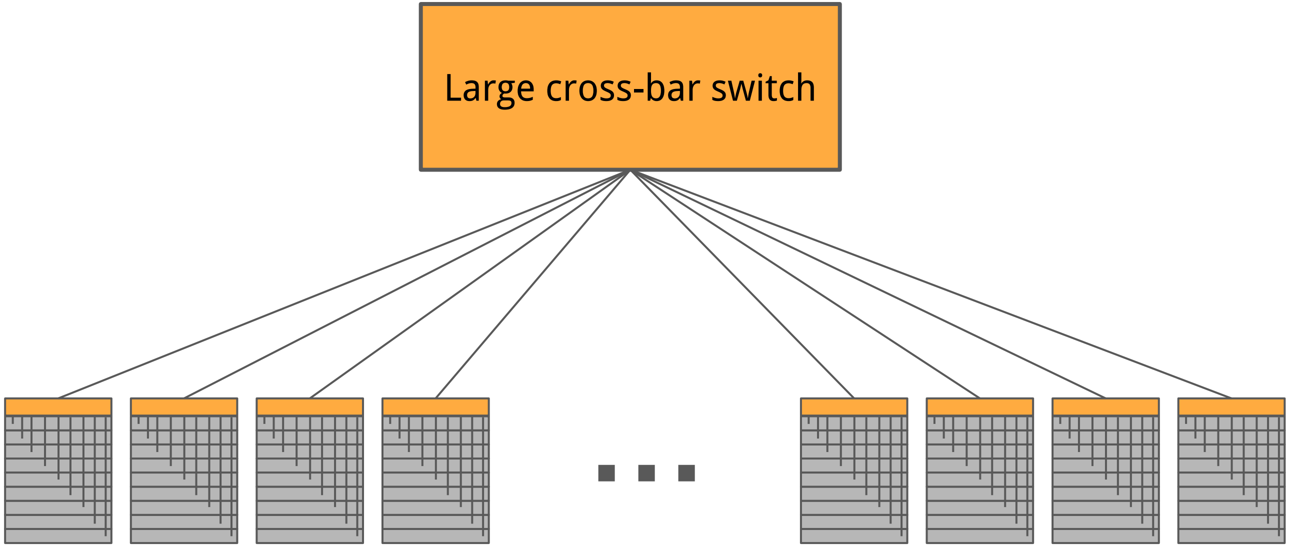
Maximising Bisection Bandwidth

- As we’ve seen, bisection bandwidth is a function of the topology of the network.

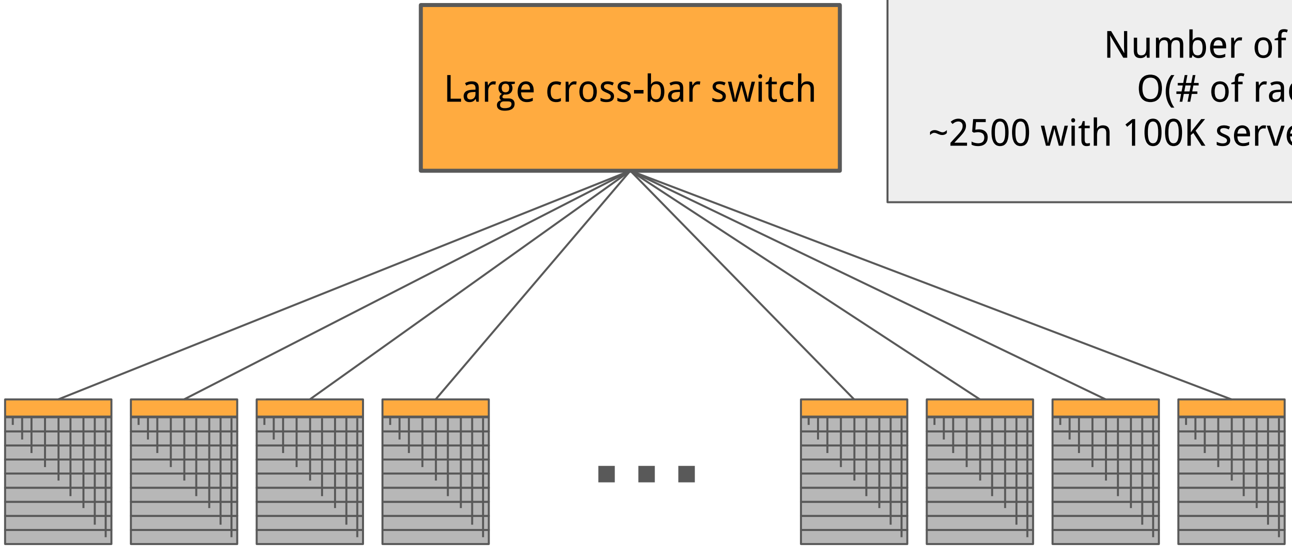
- In the datacenter we can choose our topology relatively easily.
  - Run more cables (fibre, electrical)

- What topology do we build?

“Big Switch” Approach for DC Networking

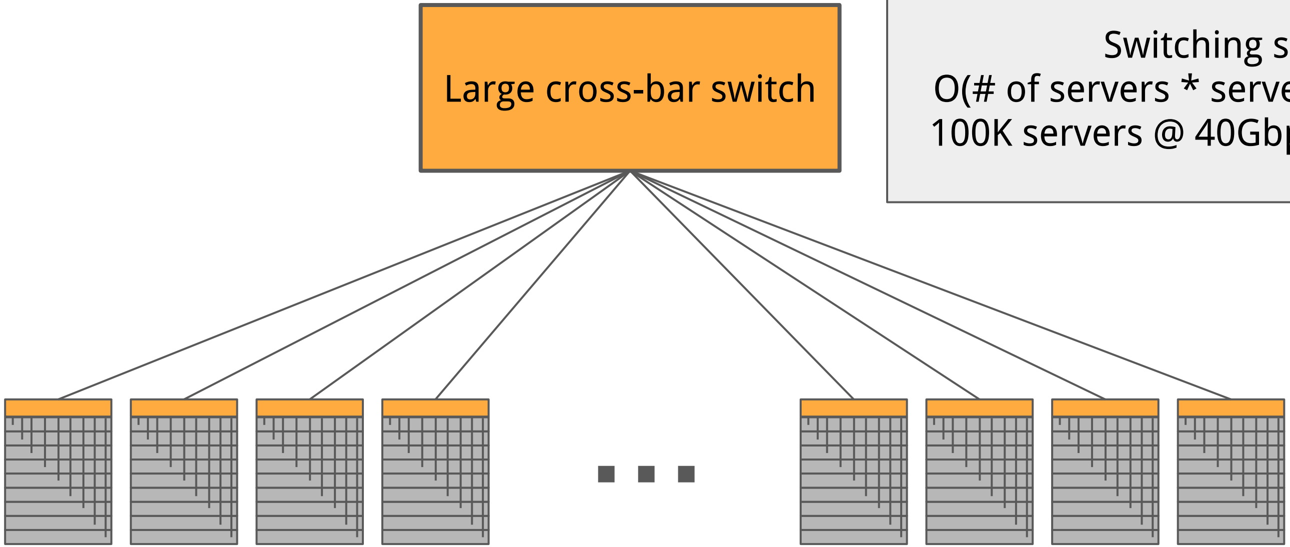


“Big Switch” Approach for DC Networking



Number of ports
O(# of racks)
~2500 with 100K servers – large **radix**

“Big Switch” Approach for DC Networking



Switching speed:
O(# of servers * server access speed)
100K servers @ 40Gbps = O(Petabits)!

Does not scale (and if it did, would be $$$$)

We tried to do this!

10K Gig-E Switch

10K Gigabit Ethernet Switch

Request for Proposal

Google Part 900190
version 1.0

But what we needed was a 10,000-port switch that cost $100/port. So, almost exactly 20 years ago, we sent this five-page RFP to four different switch vendors (IIRC: Cisco, Force10, HP, and Quanta) and tried to interest them in building such a switch. They politely declined because “nobody is asking for such a product except for you”, and they anticipated margins to be low.

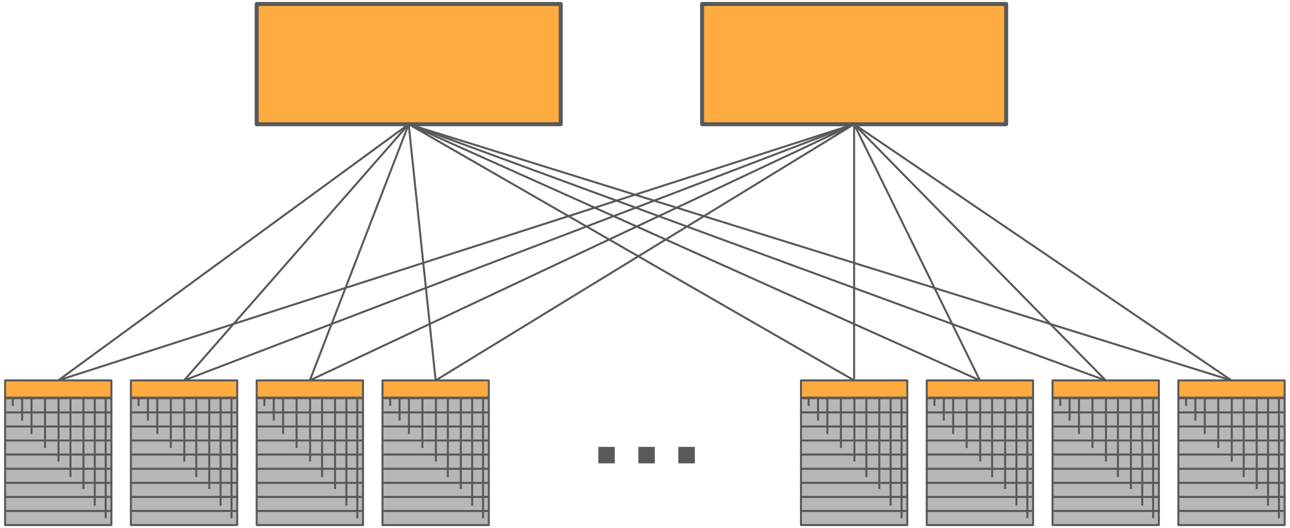
6 Implementation Ideas

This section attempts to explain why we believe it is possible to build a 10,000 port non-blocking switch for $100/port. This section does not imply any requirements for a specific design and should be thought of as one or more potential paths to a solution.

The Broadcom BCM5670 and BCM5671 provide what appear to be ideal solutions for our problem. The BCM5670 can be configured with 2 BCM5690 chips to create a 20 Gig-E port switch with 2 10G uplinks and 2 10G cross links to another box with 20 Gig-E ports. The BCM5670 can be configured with 4 BCM5690 chips to create a 40 Gig-E port switch with 4 10G uplinks. The total chip cost is $1338. Then we just need to aggregate the uplinks into a non-blocking mesh. We can build such a mesh by configuring 1125 BCM5670s into a CLOS network at a chip cost of $416250. This leads to a total chip cost of $75/port. This leaves $25/port for the phy layers, uplinks, circuit boards, power supplies, CPU or configuration system, etc.

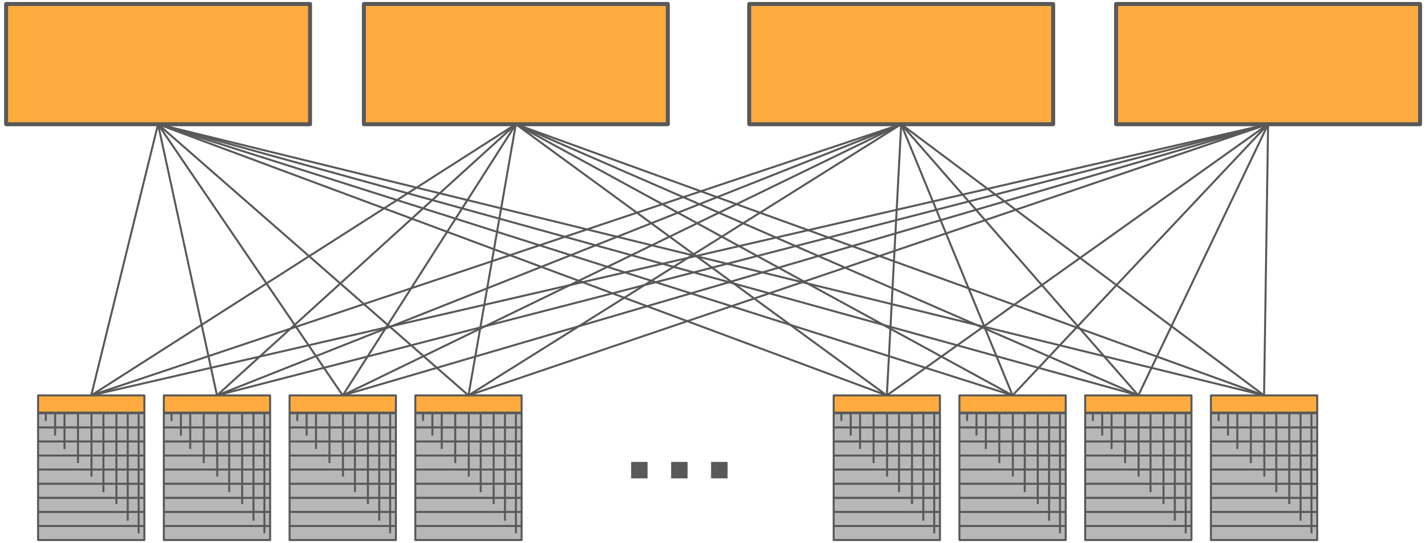
Urs Hözle (Google) on LinkedIn

Avoiding a “Big Switch”



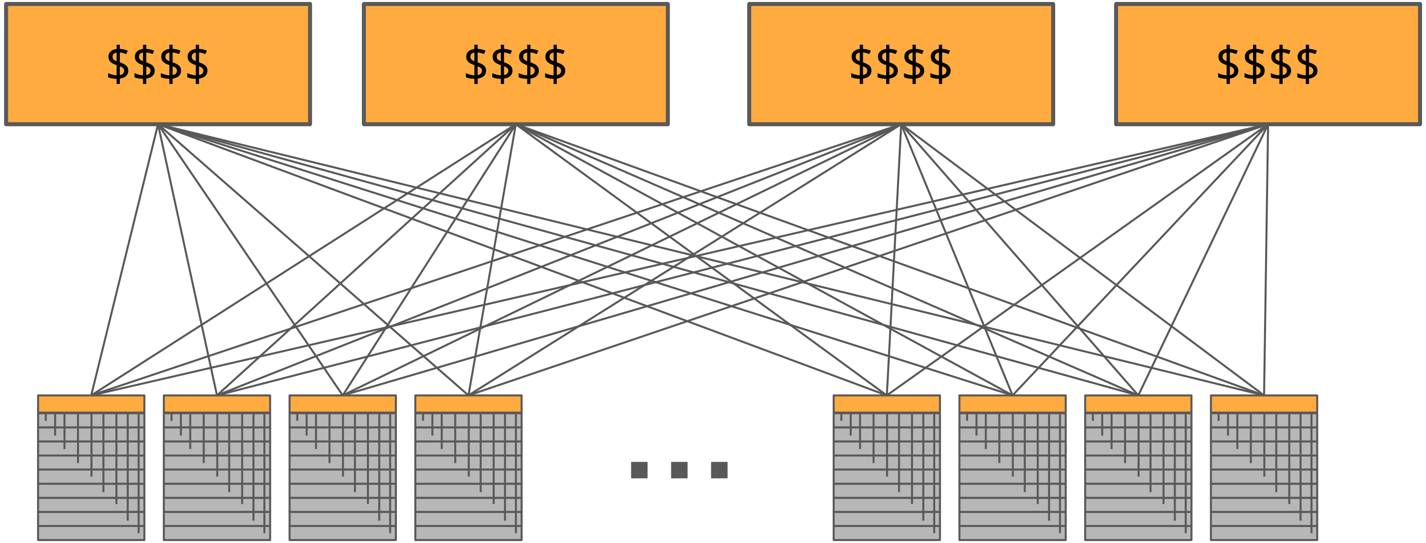
Reduced radix and bandwidth *if we don’t care about failures*

Avoiding a “Big Switch”



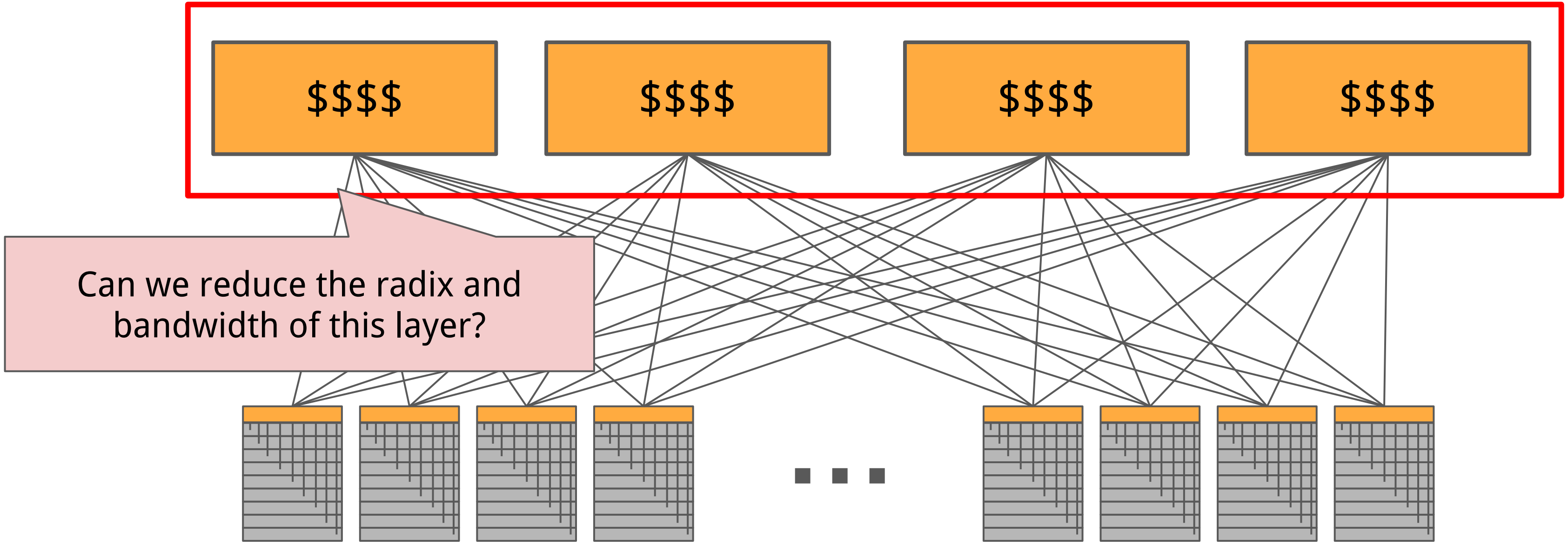
Reduced radix and bandwidth per switch - *if we can use multiple paths*

Building a DC network



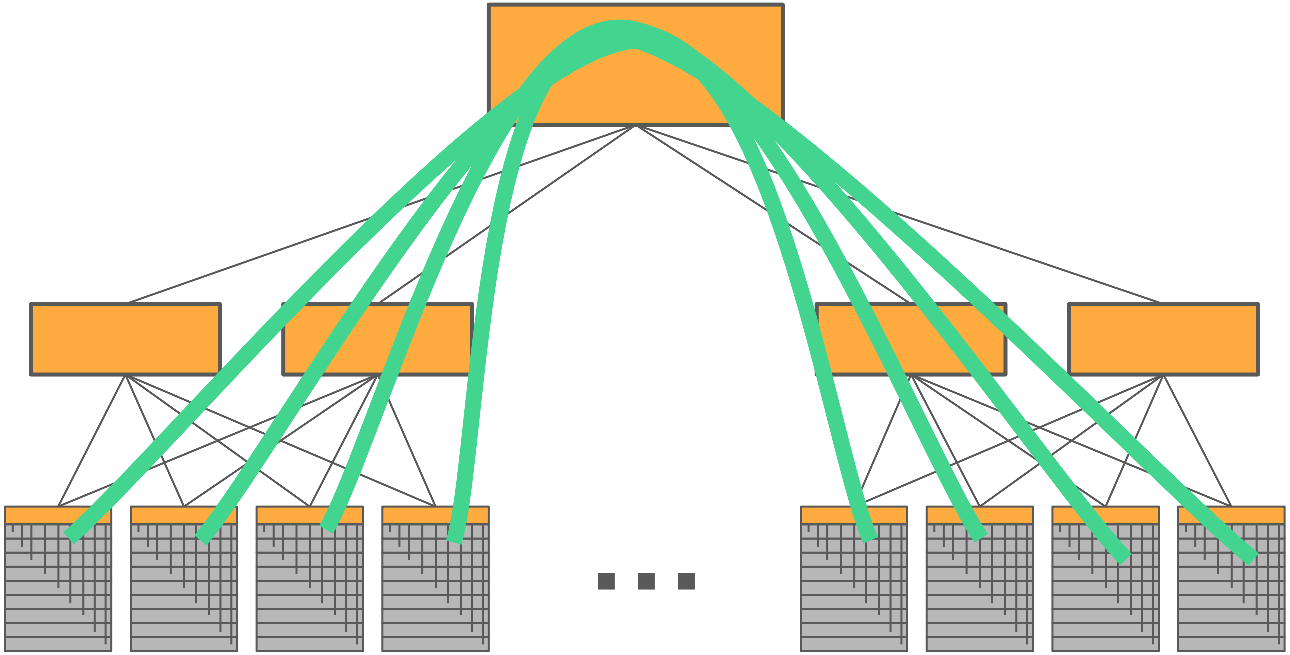
This topology works (and has been used).

Building a DC network



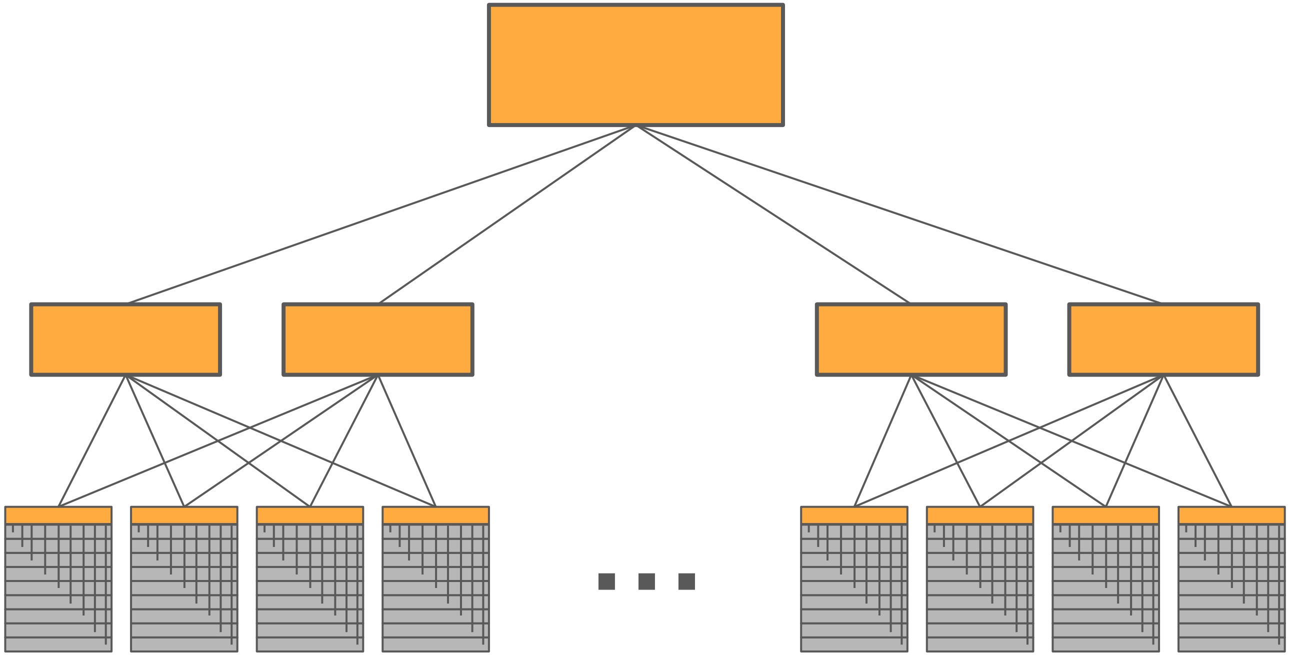
This topology works (and has been used).

A **Tree**

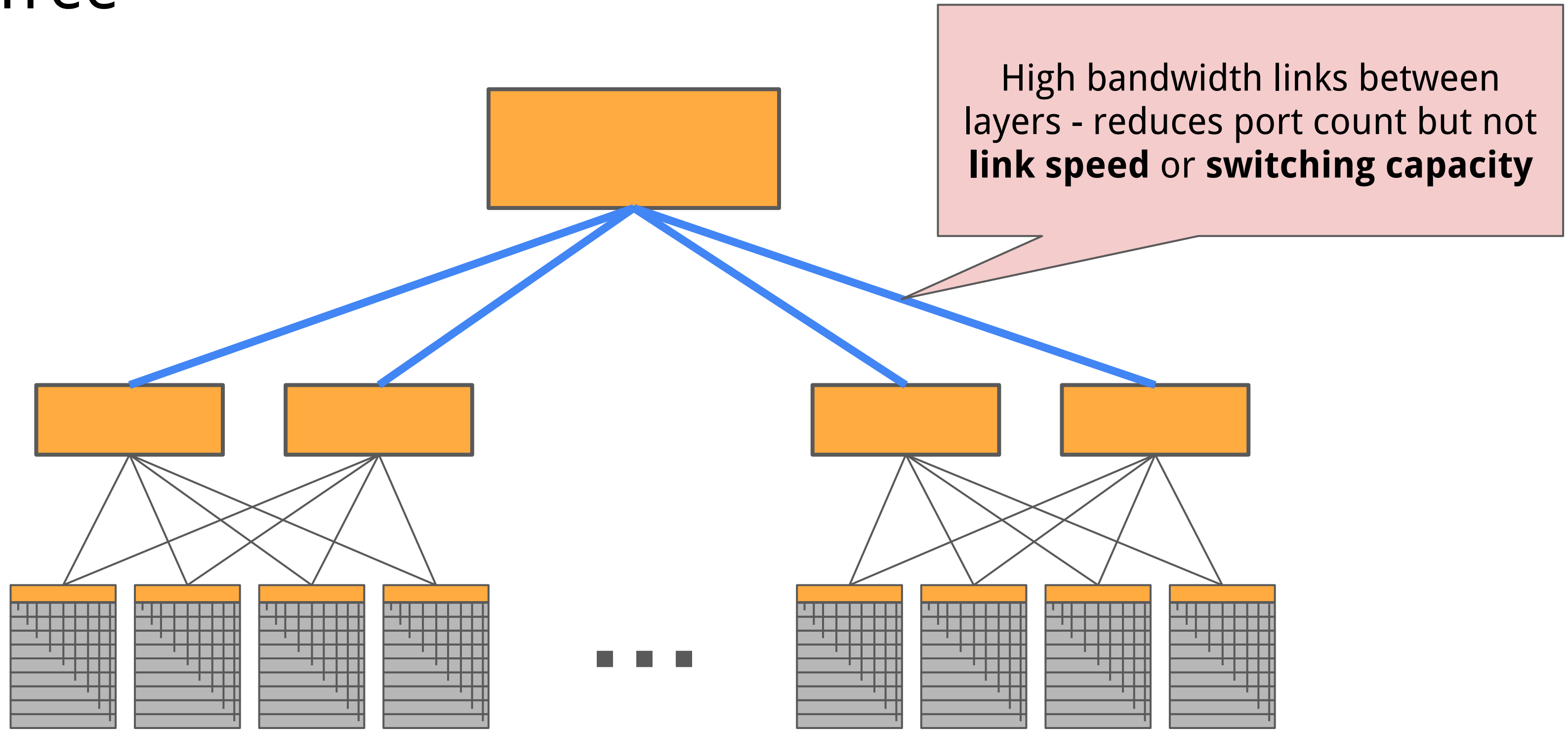


Problem: low bisection bandwidth → congestion

A **Tree**

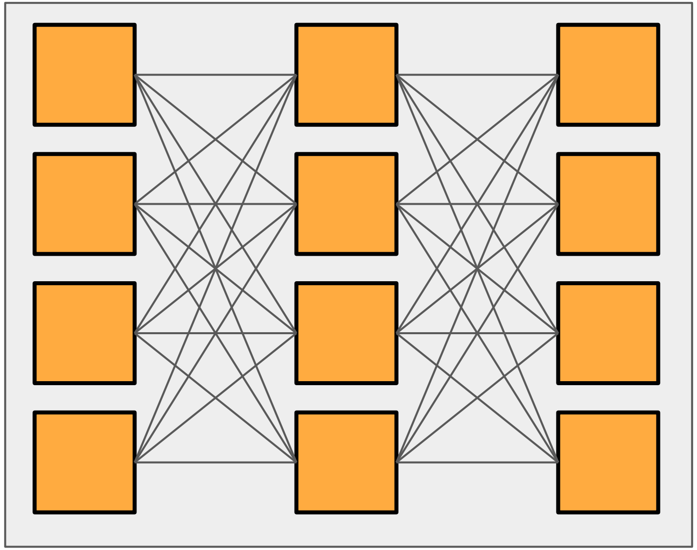


A **Fat** Tree



Still not scalable – or very expensive

Clos Networks



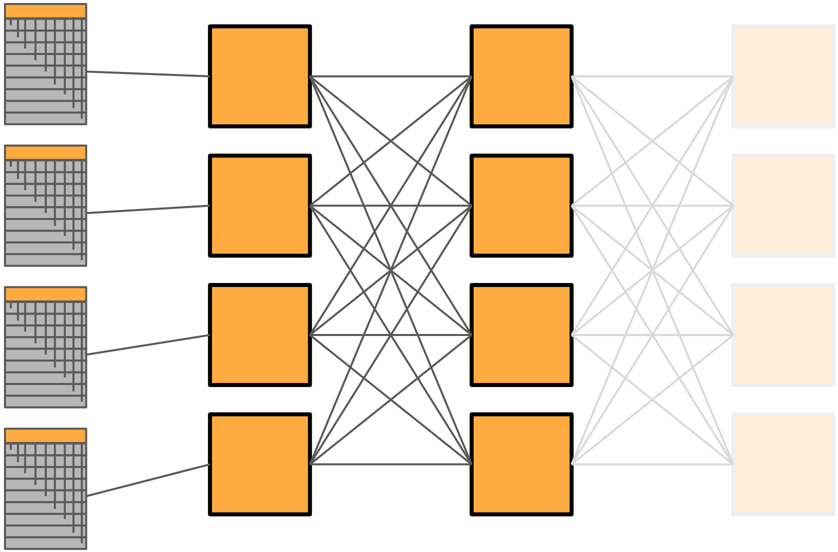
- All switches have same # of ports.
- # of ports per switch is low.
- All link speeds are the same.
- Highly multi-path.

Using small (commodity, cheap!) elements to build large capacity-rich networks.

Clos Networks

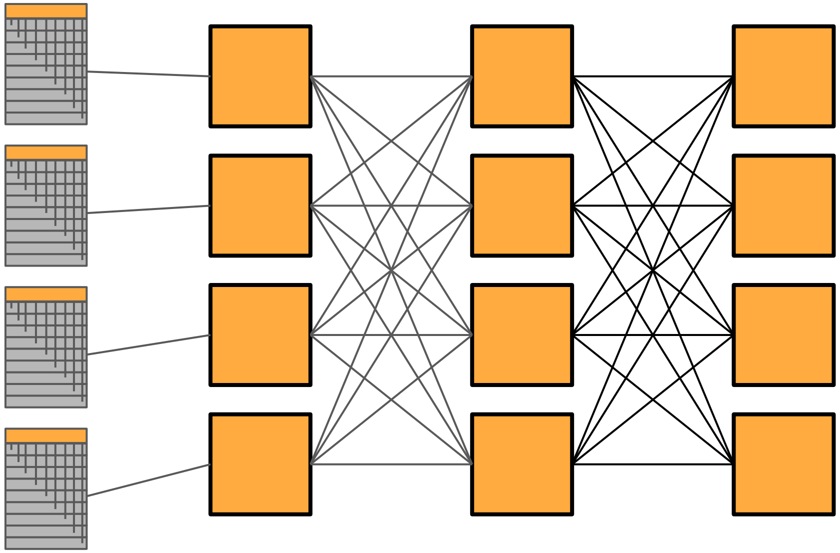
- Not a new idea!
- Formalised by Charles Clos in 1952.
- Networks can be scaled by adding *stages*.

Clos Networks



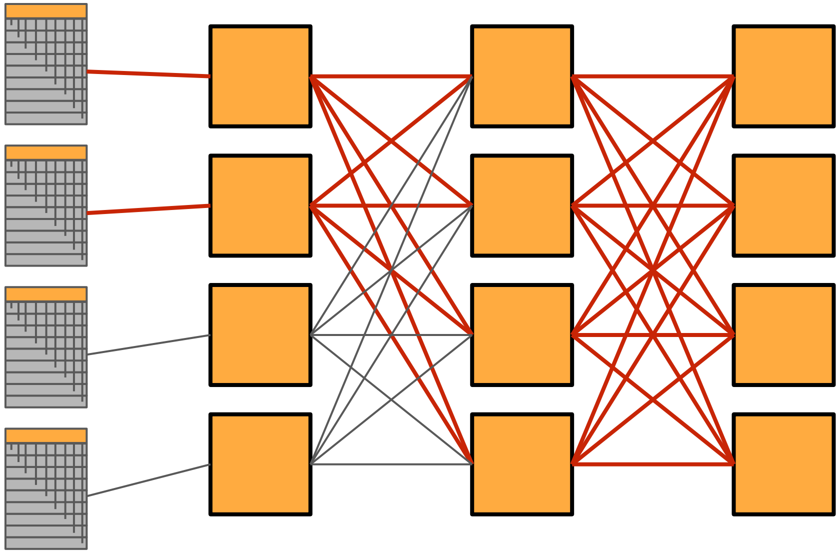
- DC networks tend to be *folded Clos*.
- Input and output switches are the same.
  - Network links are bidirectional

Clos Networks



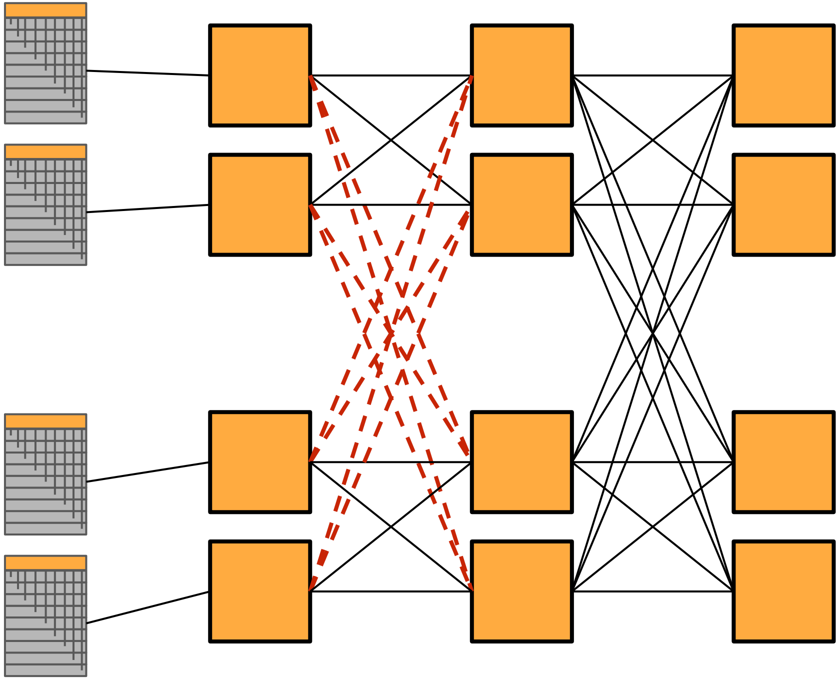
- DC networks tend to be *multi-stage*.
- Allows scaling beyond the radix of the commodity switch platforms being used.

Clos Networks

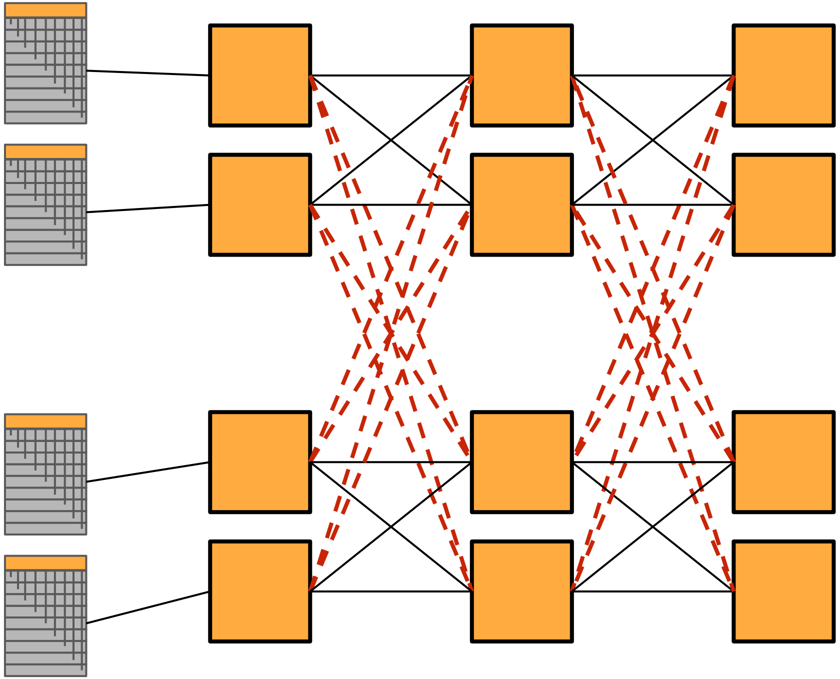


- DC networks tend to be *multi-stage*.
- Allows scaling beyond the radix of the commodity switch platforms being used.

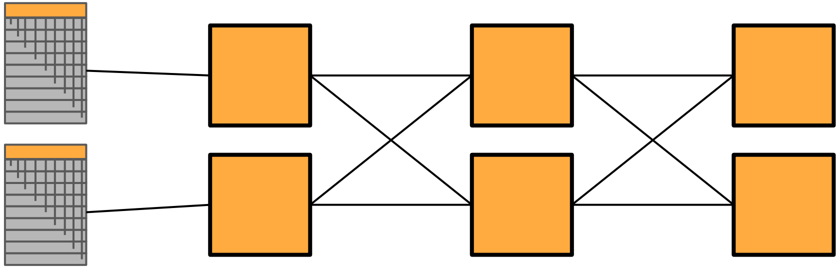
Clos Networks - Bisection Bandwidth



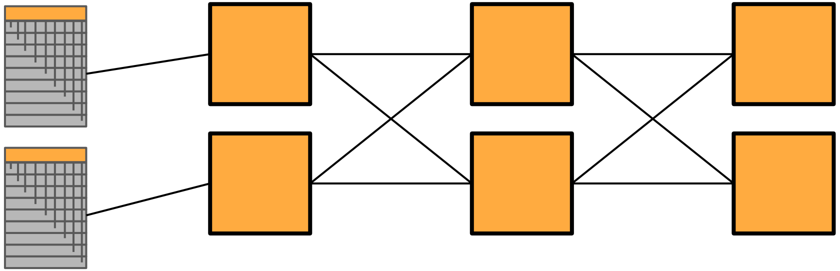
Clos Networks - Bisection Bandwidth



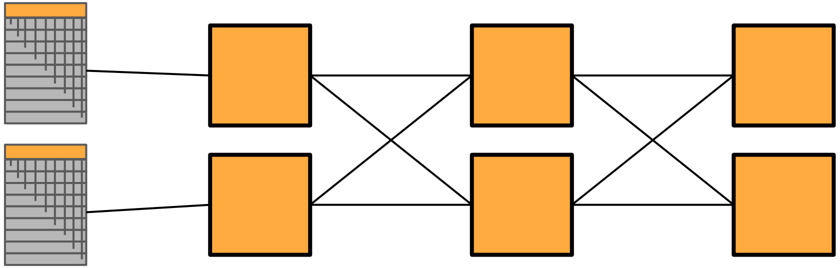
Clos Networks - Bisection Bandwidth



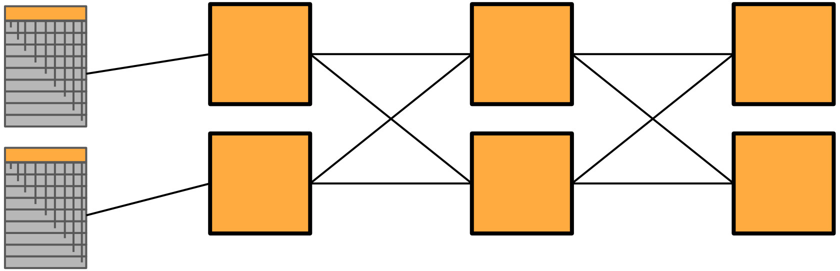
16*100G links failed to partition = 1600Gbps bisection bandwidth



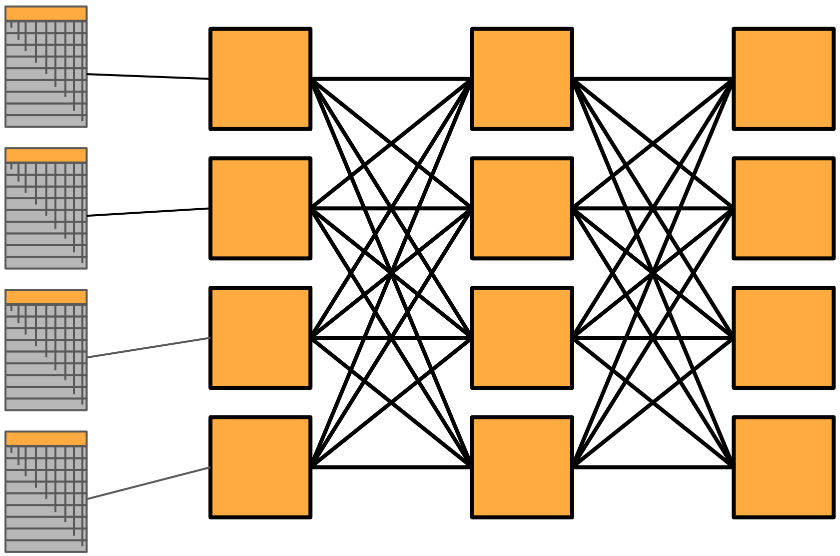
Clos Networks - Bisection Bandwidth



Full bisection bandwidth = (4*80)/2 * 100G = **1600G**

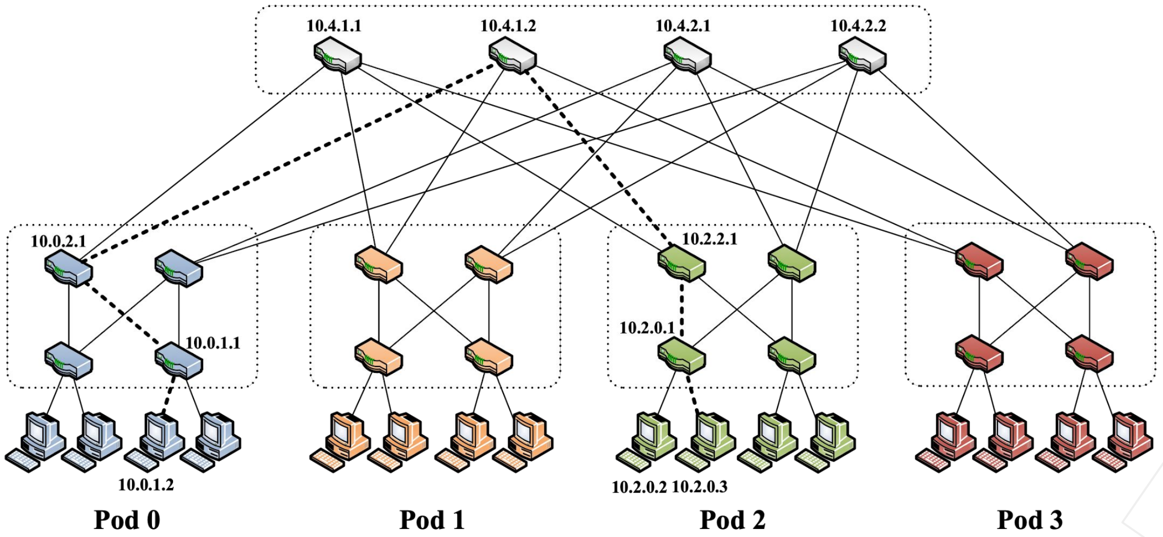


Mixing Link Speeds



- Need not have all the links be exactly the same capacity.
- Server uplinks/access links can be lower bandwidth than switch to switch links.
- Easy to accomplish where switch chips allow “breaking out” of individual ports.
- e.g., 200G server uplink, 400G switch-to-switch

Evolution of Clos Networks for DC

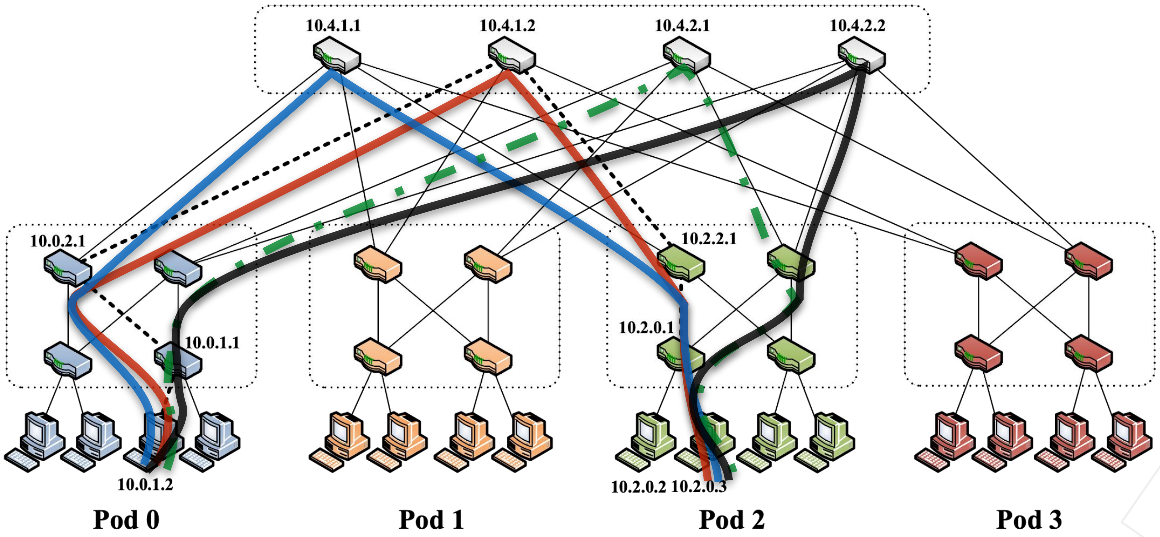


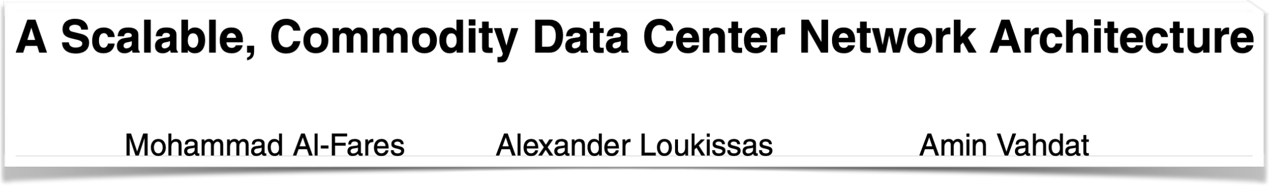
A Scalable, Commodity Data Center Network Architecture

Mohammad Al-Fares   Alexander Loukissas   Amin Vahdat

ACM SIGCOMM 2008

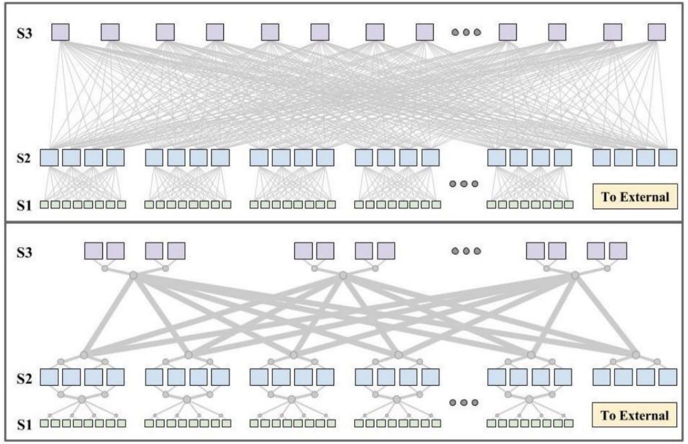
Evolution of Clos Networks for DC

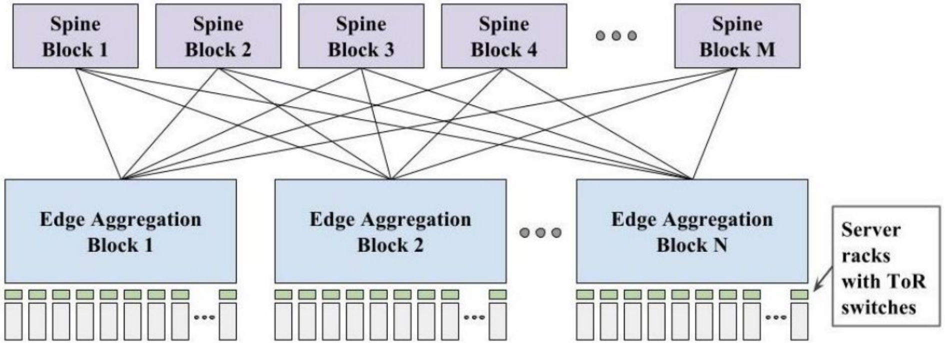




ACM SIGCOMM 2008

Evolution of Clos Networks for DC



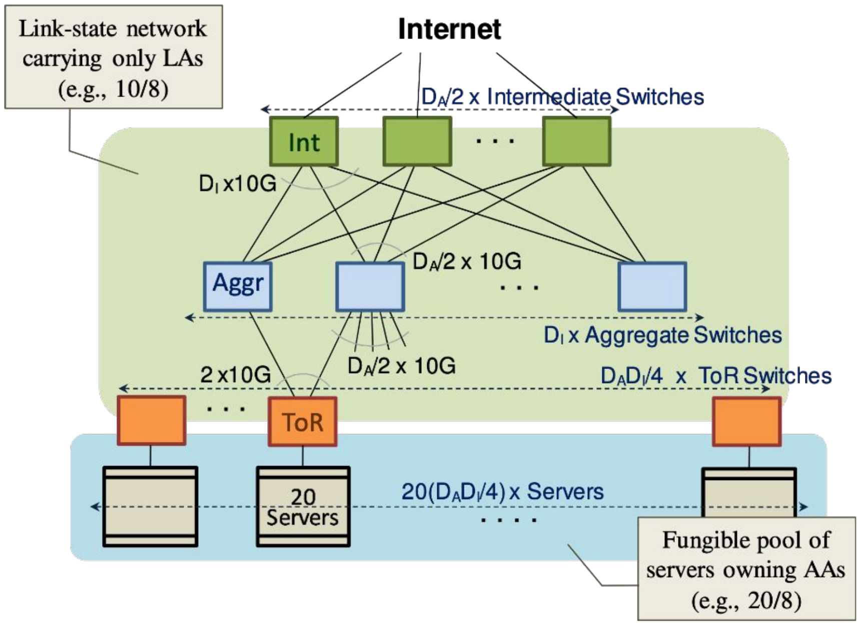


Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network

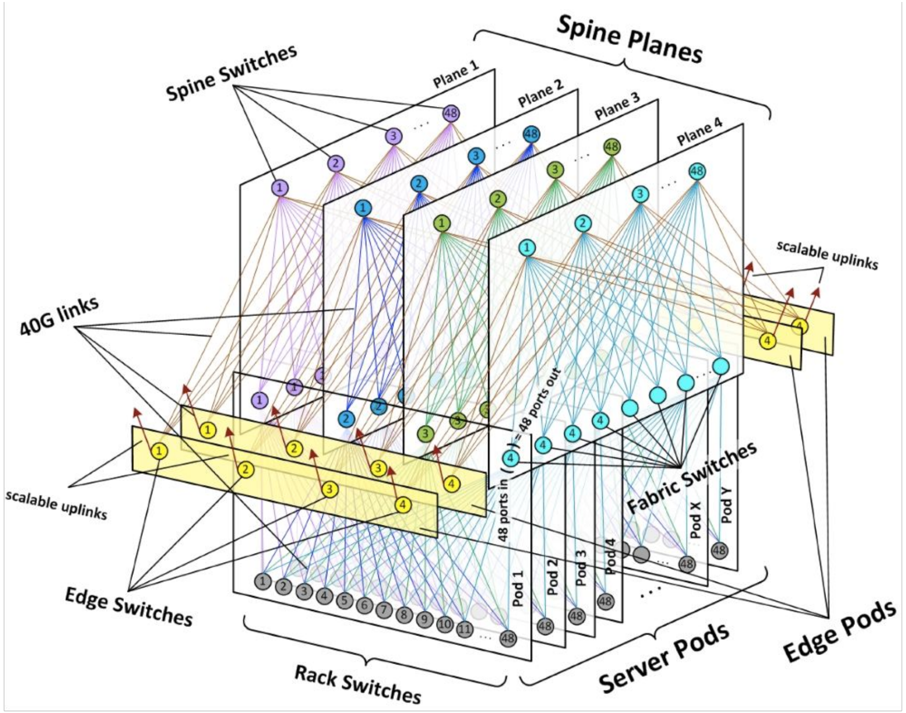
Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon,
Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost,
Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat
Google, Inc.
jupiter-sigcomm@google.com

ACM SIGCOMM 2015

Design Variants are Common



VL2 @ **Microsoft**, ACM SIGCOMM'09
Greenburg, Hamilton, Jain, Kandula, Kim, Lahiri, Maltz, Patel, Sengupta



"Introducing data center fabric, the next-generation **Facebook** data center network", Alexey Andreyev, 2015

Questions?

Congestion Control in Datacenters

- Datacenters are constrained environments – owned by a single operator.

- Leads to the opportunity for innovation to exploit the characteristics of the network.

Queuing Delay

- Packet delay = transmission delay + propagation delay + queueing delay

- Assume, 10Gbps links and 1000 byte packets
  - Transmission delay (at one hop) = 0.8 μsecs
- Assuming an average queue size of 10 packets, then per hop:
  - Per hop: avg. queuing delay = avg #pkts in queue x transmission delay = 8 μsecs
  - If we have 5 hops: queueing delay = 40 μsecs

- In the wide-area Internet, propagation delay is ~10-100s of milliseconds
- In a datacenter, propagation delay is ~10s μsecs

- Hence: packet delay may be dominated by queueing!

Improving TCP congestion control in datacenters

- **Problem:** TCP deliberately tries to fill up queues.
  - Increases the rate until the queue overflows.
- Problem is <u>worse</u> in datacenters, where there are limited types of flows.
- Most flows are short and latency-sensitive (mice).
  - e.g., queries for web search.
- Some flows are very large, and throughput-sensitive (elephants).
  - e.g., storage backups
- Elephant flows fill up buffers, delaying the mice…

Datacentre Congestion Control

- Congestion control solution must avoid filling up queues.

- Option #1: react to explicit feedback from routers (ECN).
    - Idea behind DCTCP (Microsoft).

- Option #2: react to delay instead of loss.
    - Idea behind BBR (Google).

- Both are possible because of <u>constrained environments.</u>
    - Control of the host, and the network.
    - Active area of research and development.

DCTCP

ACM SIGCOMM, 2010

DCTCP: Efficient Packet Transport for the Commoditized Data Center

Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitu Padhye, Parveen Patel, Balaji Prabhakar,
Sudipta Sengupta, Murari Sridharan

Published in 2010, in use in multiple environments.
Standardised as RFC8257, and implemented in the Linux kernel.

DCTCP

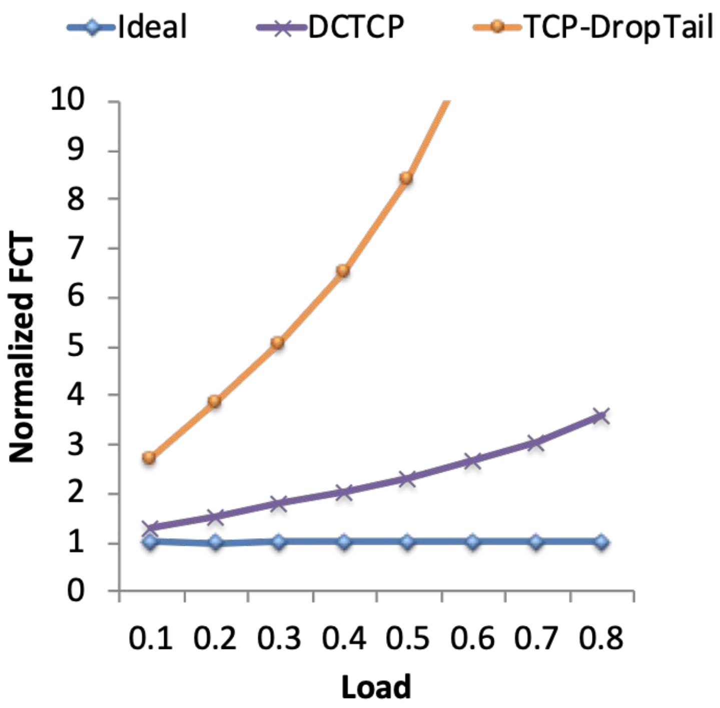
- ECN: Explicit Congestion Notification
  - Routers mark packets when queue length exceeds a threshold.
  - Sources cut their rate.
  - <u>Not</u> widely deployed in WAN routers.

- DCTCP uses ECN with modifications:
  - Routers start marking packets earlier
  - Senders cut rate in proportion to number of packets with ECN markings
    - Adapt earlier but more gently.

- Trivial change at hosts and routers.
  - But needed control of the environment → well suited for the DC!

DCTCP Performance Improvements

- **FCT**: flow completion time
    - Time from flow starting to last byte being received at the destination.
- Ideal FCT:
    - FCT using a omniscient scheduler that has global knowledge, and schedules flows to minimise FCT.
- Normalised FCT: FCT/Ideal-FCT.
    - How much longer am I than ideal?



pFabric

- Packets carry a single priority number.
  - Priority = remaining flow size (# number of unacknowledged bytes).
  - Low number means high priority.

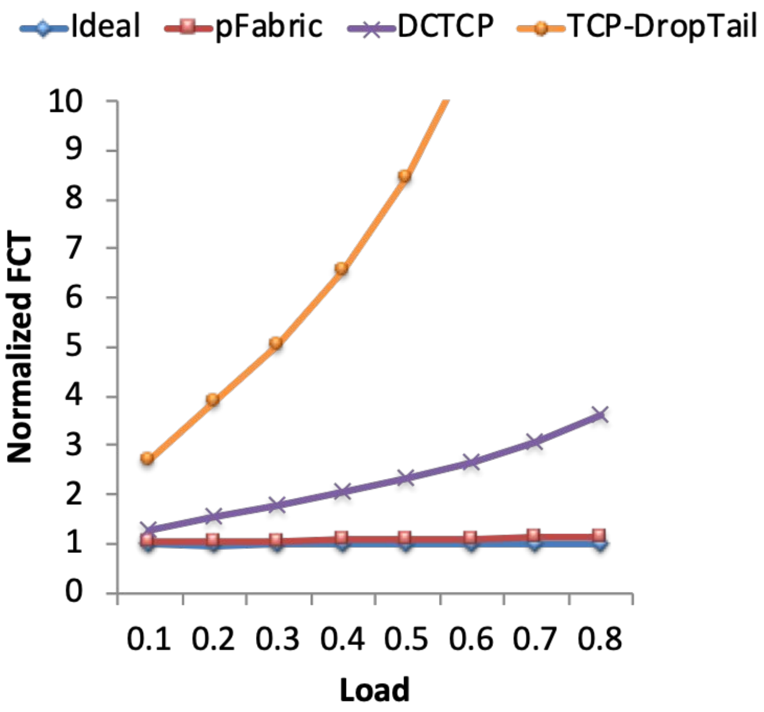
- Switches send highest priority packet.
  - Drop lowest priority packet.

- Senders: transmit/retransmit at line rate.
  - Only drop transmission rate under extreme loss (timeouts).

- Requires non-trivial changes at switches <u>and</u> end hosts.

How well does pFabric do?

- **FCT**: flow completion time
  - Time from flow starting to last byte being received at the destination.
- Ideal FCT:
  - FCT using a omniscient scheduler that has global knowledge, and schedules flows to minimise FCT.
- Normalised FCT: FCT/Ideal-FCT.
  - How much longer am I than ideal?



Why does pFabric work so well?

- Elephant and mice travel together (hence, high throughput).
- Mice get priority (hence, low latency for mice).
- A sender just transmits at full rate (no wasting time on slow start)
    - But if it's sending a large flow, most of those packets are low priority (avoids collapses).
- Nice example of clean-state network and host co-design!
- But, practically harder to realise – since it requires full control.

Summary

- Datacenters are single organisation, multi-application environments.

- A key criteria is high any-to-any bandwidth.
  - We characterise this as bisection bandwidth.

- The topology of the datacenter must be designed to both be scalable, and cost efficient.

- Some technologies - e.g., congestion control - can be optimised based on the characteristics of datacenters.

Next Time

- What else is different in datacenters?
  - Particularly, how does routing work in these topologies?

- How do we address the multi-tenant nature of a DC?