The design of data center networks

Paolo Giaccone

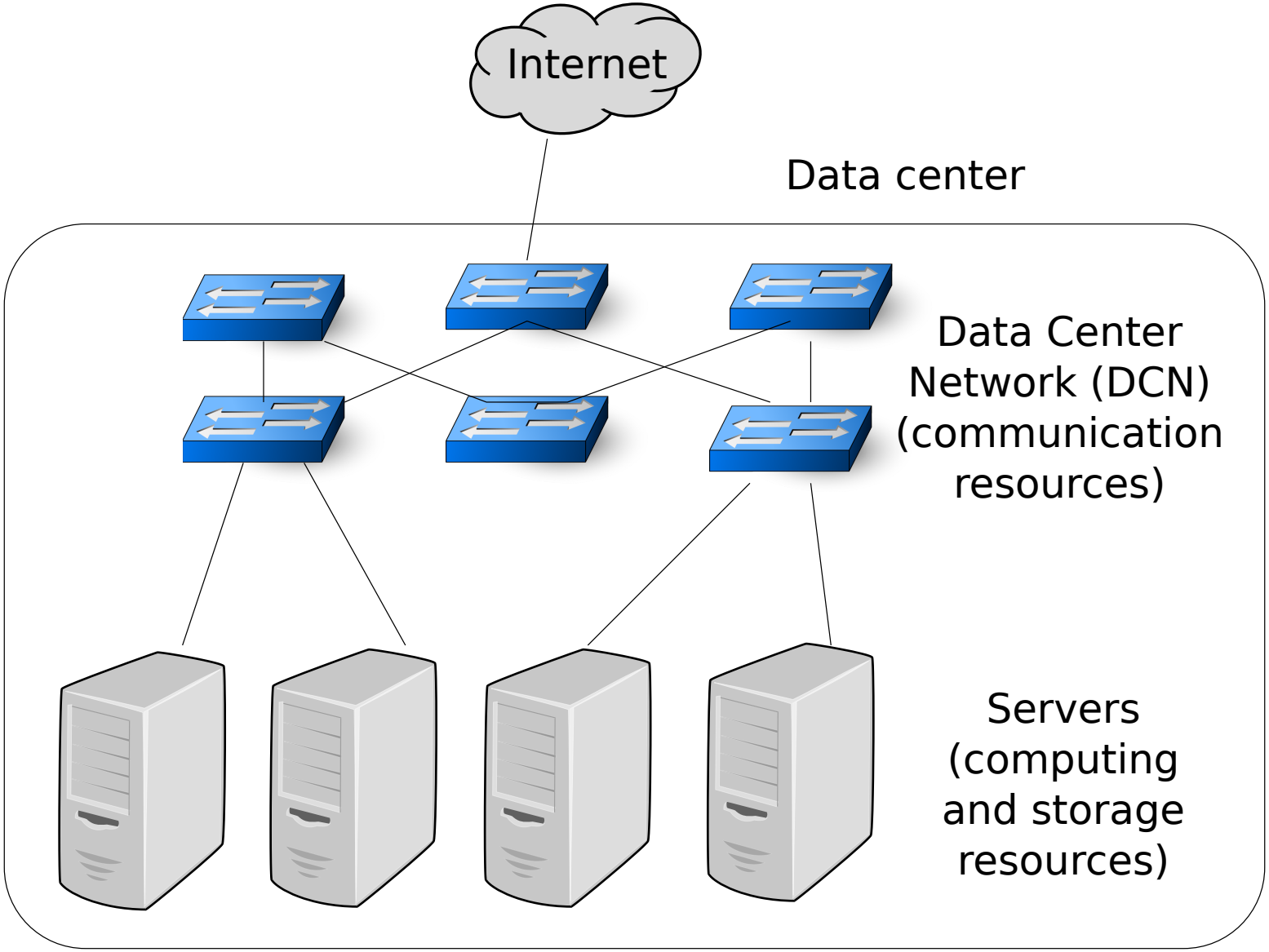
Notes for the class on "Switching technologies for data centers"

Politecnico di Torino

October 2021

Outline

Logical view of a data center



State-of-art design of Data Center Networks (DCN)

Three main pillars

- network is programmed via software
  - Software Defined Networking (SDN)
- network functions are virtualized and chained
  - Network Function Virtualization (NFV)
- physical interconnection network is based on variants of leaf-spine topology
  - based on Clos networks, factorized recursively

Section 1

SDN and Virtualization

Software Defined Networking (SDN)

SDN architecture

- network control (i.e., control plane) is decoupled from forwarding (i.e., data plane)
- network is directly programmable in software
- control plane is logically centralized in a server, where the SDN controller runs
- centralized control allows a coherent network view very useful to develop network applications
  - network automation and traffic engineering (e.g., load balancing, resource consolidation policies, restoration, etc.)
  - traffic monitoring and steering
- adopted to control Data Center Networks (DCN)

Virtualized Data Center

Resource virtualization

- Resources are virtual and decoupled from the physical hardware providing it
- Physical resource pooling
- Simple migration of resources to new/other physical hardware

Technologies

- server virtualization
- storage virtualization
- network virtualization
- network function virtualization (NFV)

Virtualization in computing and storage

Server virtualization

- run many VMs and applications on the same physical server
- hypervisor software
  - coordinate the VM access to the physical resources (CPU, memory, storage, etc.)
  - operating between the VMs and the physical server

Storage virtualization

- the users and the applications access the storage without knowing where it is located, its internal structure, how it is managed (e.g. backup, recovery)

Virtualization in networking

Network virtualization

- segmentation among VMs: each group of VMs "sees" its own network
- network slicing
  - very flexible definition of the logical topology on the DCN
  - btw, key technology for 5G networks
- implementations
  - traditional approach: VLAN, XVLAN, etc.
  - SDN approach: Openflow, OpenvSwitch, etc.

Virtualization in networking

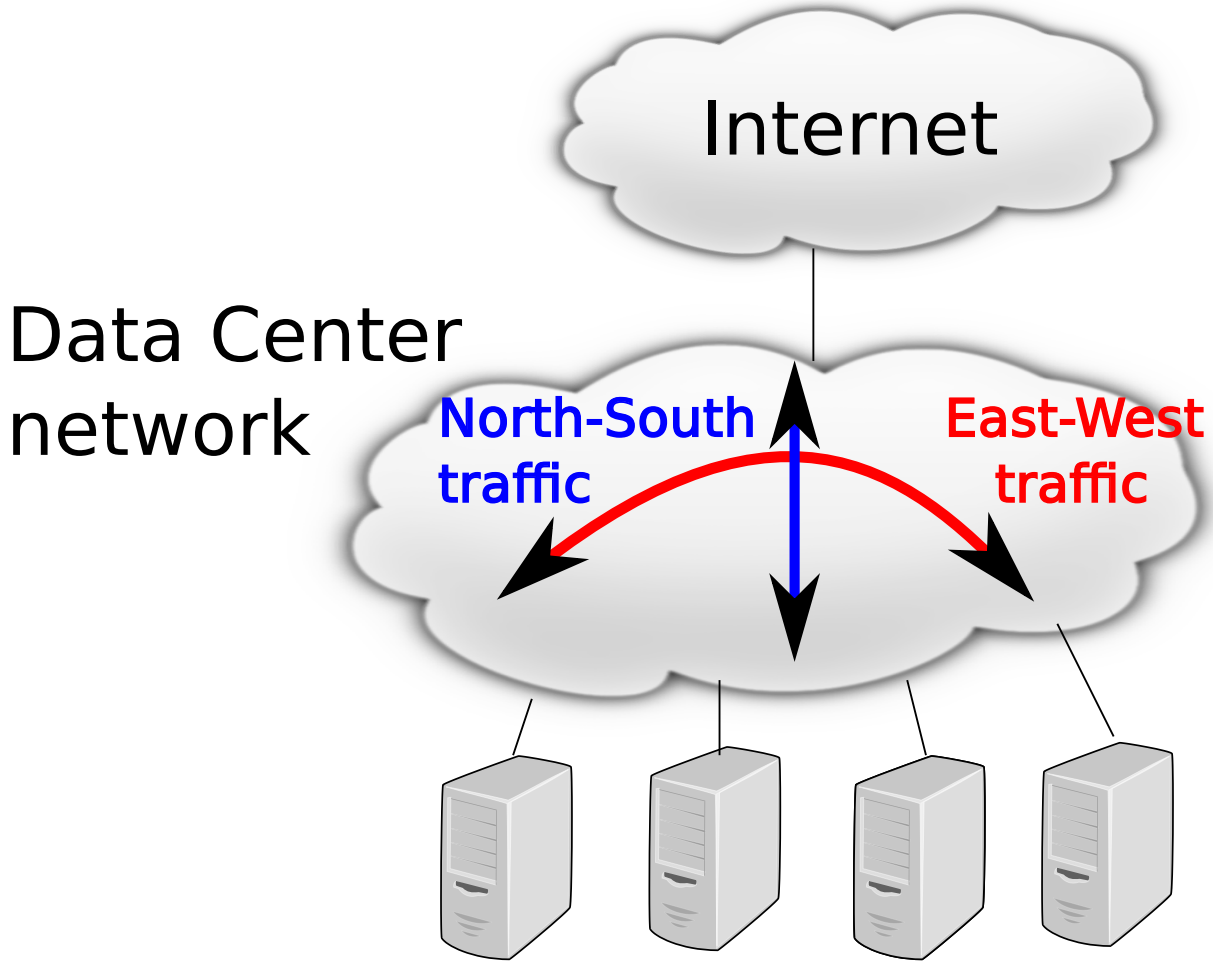
Network Function Virtualization (NFV)

- network functions (NFs) (as firewall, load balancer, traffic classifier, etc.) run on a VM
  - no need for specialized hardware, just commodity servers are enough
- Service Function Chaining (SFC)
  - usually integrated with SDN to route the traffic across the different NFs
    - Segment Routing (SRv6) as possible solution, based on source routing

Section 2

Data center traffic

Data traffic within a data center



Intra-data center traffic

East-West traffic

- storage replication (few flows, many data)
  - in Hadhoop distributed filesystem, at least 3 copies of the same data, usually two in the same rack and one in another rack
- VM migration
- Network Function Virtualization (NFV)
  - data is processed through a sequence of VMs (e.g., firewall, web server, parental control, accounting server)

East-West traffic usually larger than North-South traffic. Some citations:

- A 1 byte transaction in North-South traffic generates on average a 100 bytes transaction in East-West traffic
- According to Cisco's Global Cloud Index: In a data center: East-West traffic (76%), North-South traffic (17%), inter-data center traffic (7%). In campus networks: North-South traffic (>90%).

  (`http://blogs.cisco.com/security/trends-in-data-center-security-part-1-traffic-trends`, May 2014)

East-West traffic patterns

Unicast

- point-to-point communication
- e.g., VM migration, data backup, stream data processing

Multicast

- one-to-many communication
- e.g., software update, data replication (≥ 3 copies per content) for reliability, OS image provision for VM

Incast

- many-to-one communication
- e.g., reduce phase in MapReduce, merging tables in databases

Section 3

Basic routing and addressing schemes

Addressing and routing in a data center

Scenario

100,000 servers, 32 VM each $\rightarrow 3 \cdot 10^6$ MAC and IPs

Addressing and routing schemes

- challenging
- standard schemes do not provide efficient solutions
- e.g. consider two possible alternative options:
  1. layer-2 addressing
  2. layer-3 addressing

Routing design

Requirements taken (verbatim) from IETF RFC 7938

- "Select a topology that can be scaled horizontally by adding more links and network devices of the same type without requiring upgrades to the network elements themselves."
- "Define a narrow set of software features/protocols supported by a multitude of networking equipment vendors."
- "Choose a routing protocol that has a simple implementation in terms of programming code complexity and ease of operational support."
- "Minimize the failure domain of equipment or protocol issues as much as possible."
- "Allow for some traffic engineering, preferably via explicit control of the routing prefix next hop using built-in protocol mechanics."

Layer-2 addressing

One single LAN

Drawbacks

- very large forwarding tables in L2 switches
- shared broadcast domain and lots of broadcast traffic (e.g., ARP)
- routing loops
  - (unfortunately) no TTL in Ethernet packets
  - standard Spanning Tree Protocol (STP) not suitable
    - routing across multiple paths cannot be exploited
    - slow to converge
- large failures
  - due to possible improper cabling, misconfiguration, or flawed software on a single device
  - affect the entire spanning-tree domain

Layer-2 design

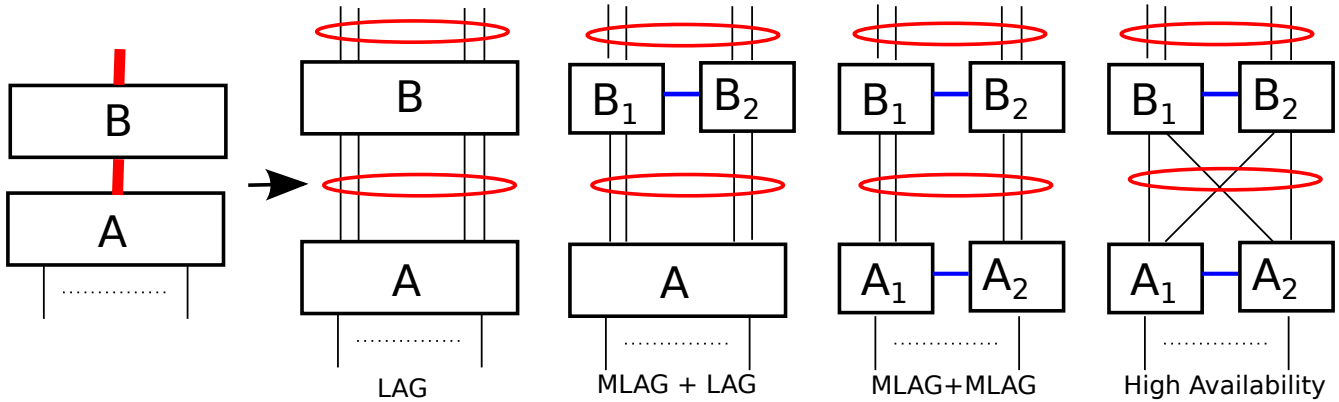
STP enhancements

- Rapid Spanning Tree Protocols (RSTP)
  - to converge faster than STP
- Multiple Spanning Tree Protocol (MST)
  - to exploit multiple paths
- Multi-Chassis Link-Aggregation (MC-LAG)
  - to virtualize multiple links and switches into a single link and switch
  - lack of ability to scale to > 2 links or switches
  - lack of standard implementations
  - failure risk of syncing the states between the switches

Layer-2 design

Multi-Chassis Link-Aggregation (MC-LA)

- Link Aggregation (LA) allows one or more links to be aggregated together to form a Link Aggregation Group (LAG), such that they are seen as a single link
  - automatically distributes and load balances the traffic across the working links within a LAG, thus high throughput
- Multi-Chassis (MC) allows two or more switches to share a common LAG endpoint, as they were a single virtual switch
  - switches in an MC-LAG cluster communicate to synchronize and negotiate automatic switchovers, thus high throughput and redundancy



Layer-3 addressing

One subnet per VLAN

Drawbacks

- many DHCP servers and VLANs
- very large number of switches and routers (around 10,000)
- Interior Gateway Protocol IGP (e.g., OSPF) in each router
  - manual administrator configuration and oversight
  - flood prefix advertisements, controlled only by a basic partition into areas
- routing loops
  - (fortunately) TTL in IP packet
- VM migration
  - when changing LAN, a new IP address is required and existing TCP connections break

Practical solutions

Addressing in data centers

- VXLAN
  - scalable tunneling scheme similar to VLAN
- LISP
  - provides IP address mobility across layer-3 subnets
- IPv6
- BGP
- but many other solutions: FabricPath, TRILL, NVGRE, OTV, Shortest Path Bridging (SPB), Segment Routing (SRv6), etc.

BGP in the Internet

- reliable (runs on TCP) and secure
- routing protocol used by Internet Autonomous Systems (AS)
  - iBGP designed to route traffic within an AS
  - eBGP designed to route traffic between ASs
- AS-path
  - routing path to the network prefixes
  - e.g., 1.1.0.0/16 through AS1$\rightarrow$AS2$\rightarrow$AS3
- support traffic engineering
  - path attribute for each AP-path
  - flexible tag routing to set the preference for different paths
- AS-paths allow to construct a graph of AS connectivity
  - remove routing loops
  - enforce policy decisions on routing

BGP vs OSPF

Scalability and state management

- OSPF
  - periodic refresh of routing information
- BGP
  - simpler internal data structures and state machines
  - routing state does not expire

Failures

- OSPF
  - the event propagation scope is an entire area
- BGP
  - less information flooding overhead
  - every BGP router calculates and propagates only the selected best-path
  - network failure masked as soon as BGP finds an alternate path, which exists for Clos topologies

BGP in DCN

ECMP (Equal Cost Multiple Path) support

- multiple best paths to the same destination
- allows load balancing, crucial to keep low congestion and reduce delays

Two distinct options:

iBPG

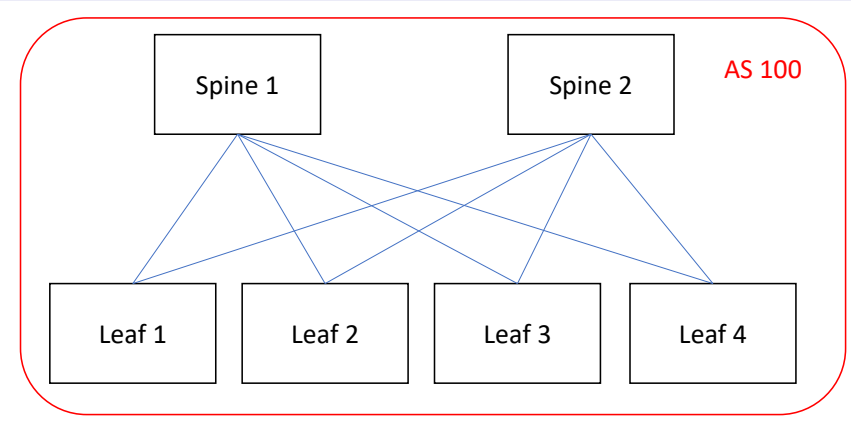
- one AS for all the DCN
- all the network nodes (switches/routers) are under the same AS
- full mesh of connected routers to allow the full distribution of prefixes
  - limited scalability, but this issue can be addressed with route reflectors

eBGP

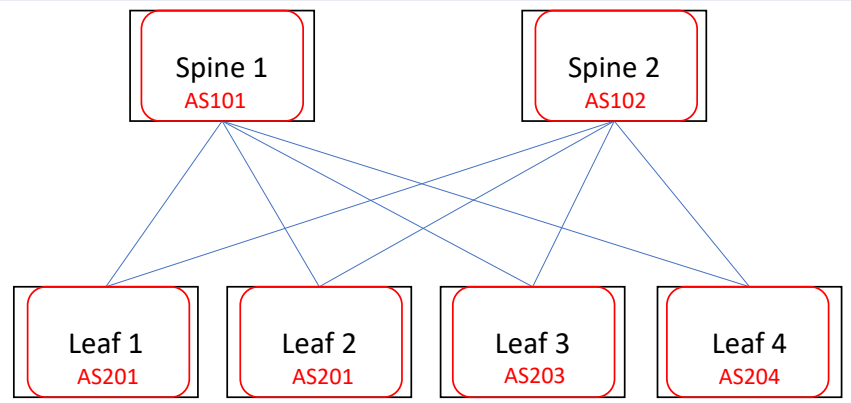
- each network node appears as a distinct AS
- at most 65k ASN (AS number), thus network node

iBGP vs eBGP

iBGP



eBGP

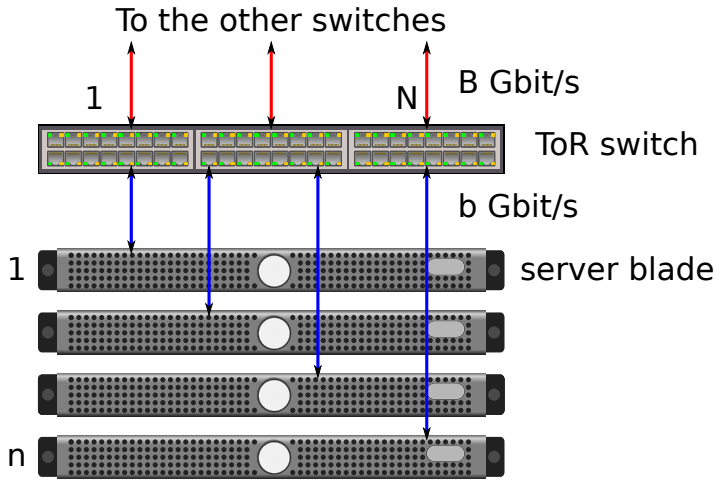


Section 4

Interconnection topologies

Server packing in a rack

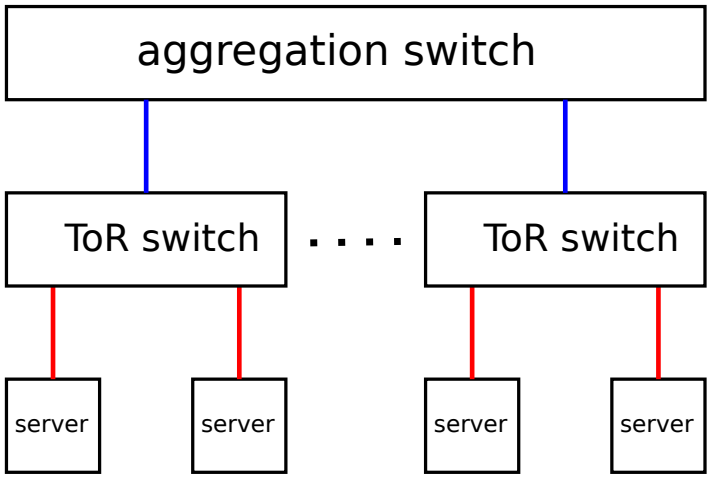
- Standard 19 inch rack
- 42 EIA Units (pizza box)
  - 40 server blades
    - possible single /26 subnet
  - 1 ToR (Top of Rack) switch
- without oversubscription: $NB = nb$
  example
  - 40 ports @ 1 Gbit/s to the servers
  - 4 ports @ 10 Gbit/s to the other switches
- with oversubscription: $NB < nb$
  example with oversubscription 1:4
  - 40 ports @ 1 Gbit/s to the servers
  - 1 ports @ 10 Gbit/s to the other switches



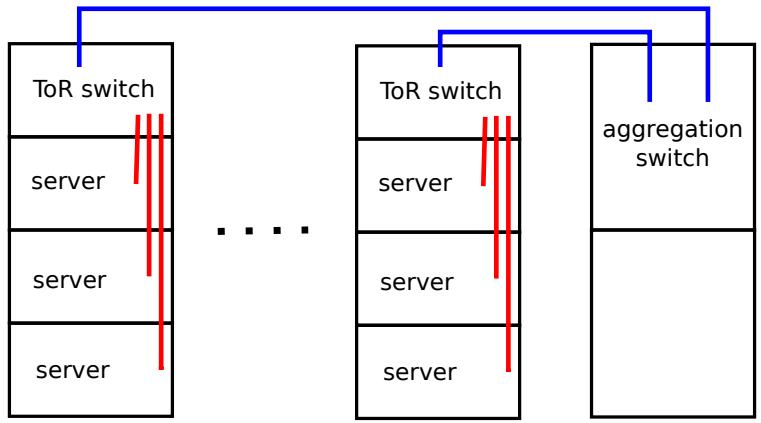
ToR vs EoR architectures

ToR (Top-of-Rack) architecture

- in a rack, all servers are connected to a ToR switch
- the servers and the ToR switch are colocated in the same rack
- aggregation switches in dedicated racks or in shared racks with other ToR switches and servers
- simpler cabling, but higher complexity for switch management



logical topology

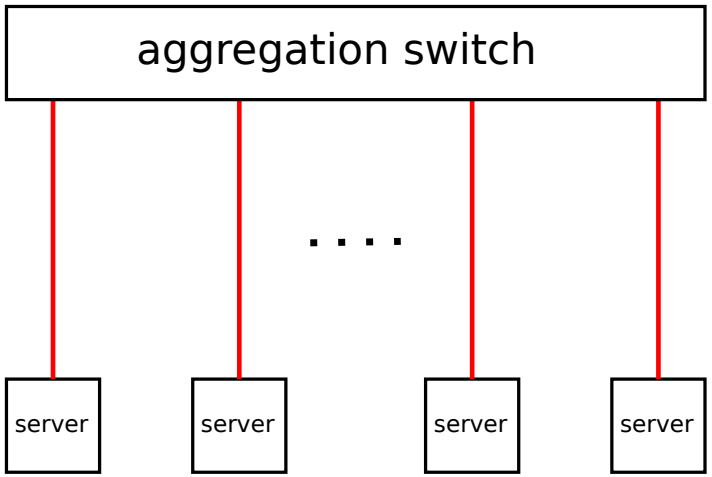


physical topology

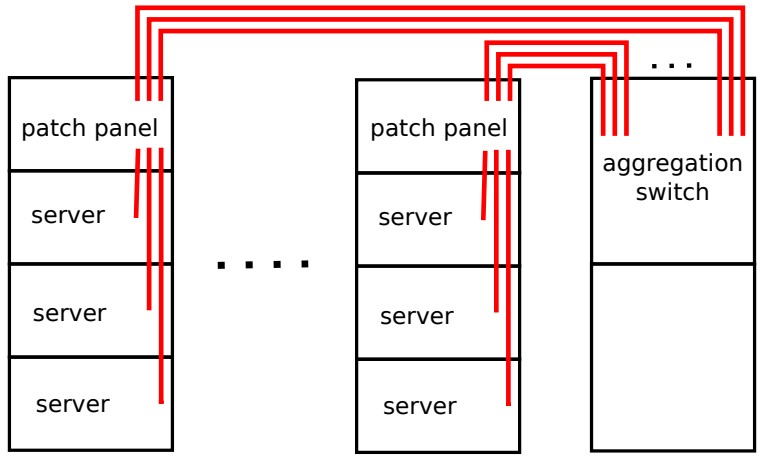
ToR vs EoR architectures

EoR (End-of-Row) architecture

- servers in a racks are connected directly to the aggregation switch in another rack
- patch panel to connect the servers to the aggregation switch
- simpler switch management, but more complex cabling



logical topology



physical topology

Interconnection among racks

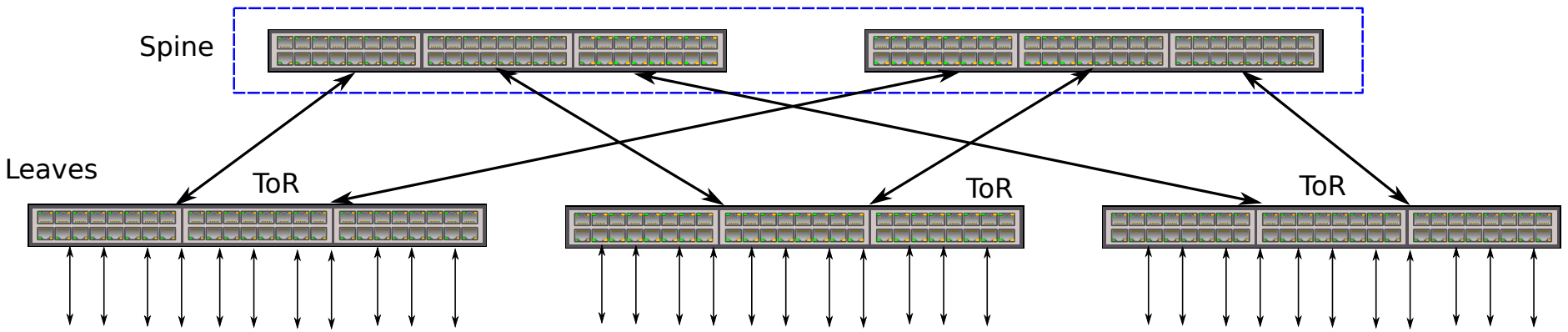
Leaf and spine

Two stage interconnections

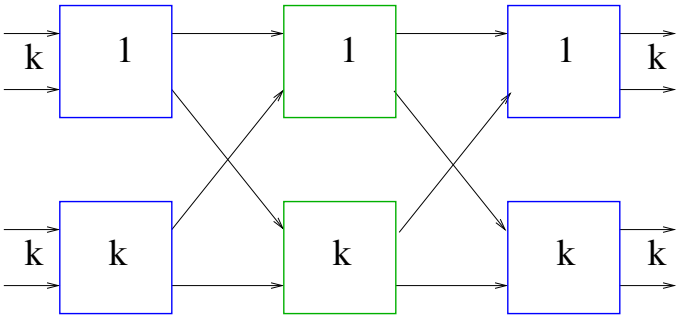
- Leaf: ToR switch
- Spine: dedicated switches (aggregation switches)

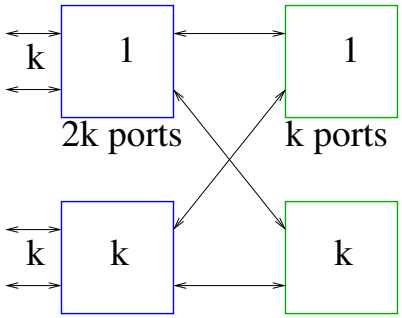
In practice

- servers with two interfaces connected to two ToR switches to provide fault-tolerance



From Clos to "Leaf and Spine" topology





Clos topology

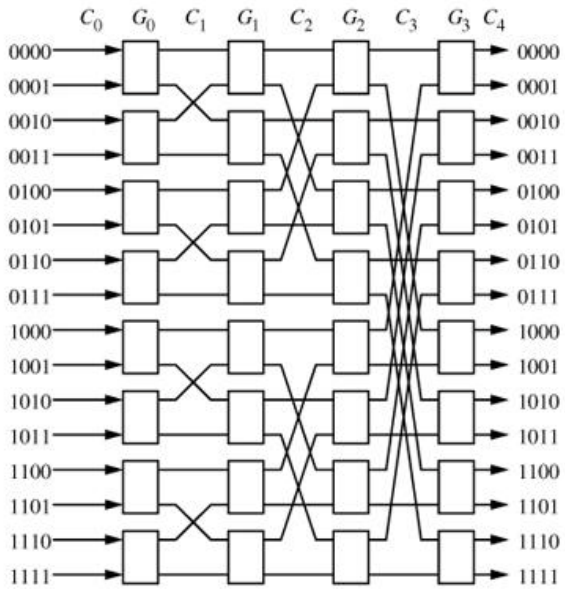
- each switching module is unidirectional
- each path traverses 3 modules

Leaf and spine topology

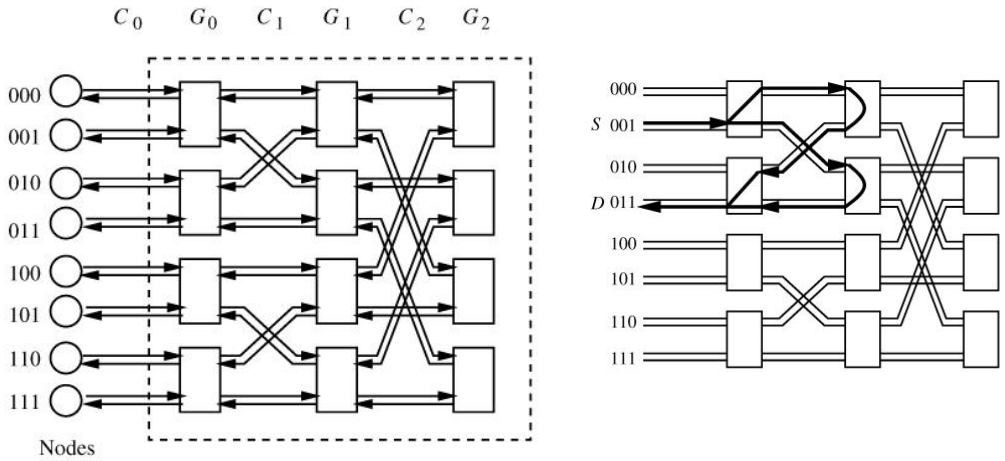
- each switching module is bidirectional
- each path traverses either 1 or 3 modules

From unidirectional to bidirectional networks

Unidirectional Banyan (butterfly) network



Bidirectional butterfly network

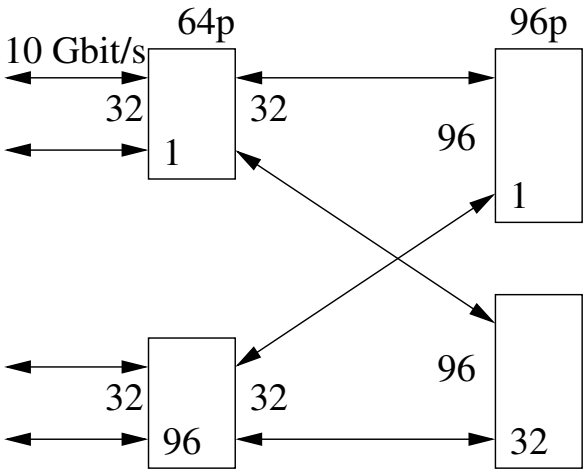


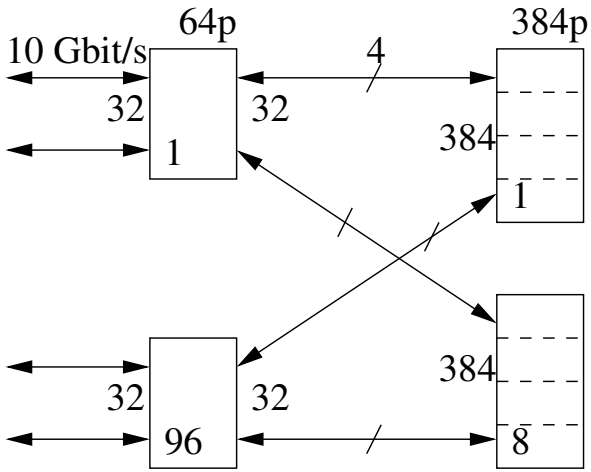
Pictures taken from "Interconnection Networks: An Engineering Approach", by Duato, Yalamanchili and Ni, 2003

Example of DCN design

3 072 servers

- 3 072 ports at 10 Gbit/s $\Rightarrow$ 30.72 Tbit/s
- alternative designs
  - 96 switches with 64 ports and 32 switches with 96 ports
  - 96 switches with 64 ports and 8 switches with 384 ports



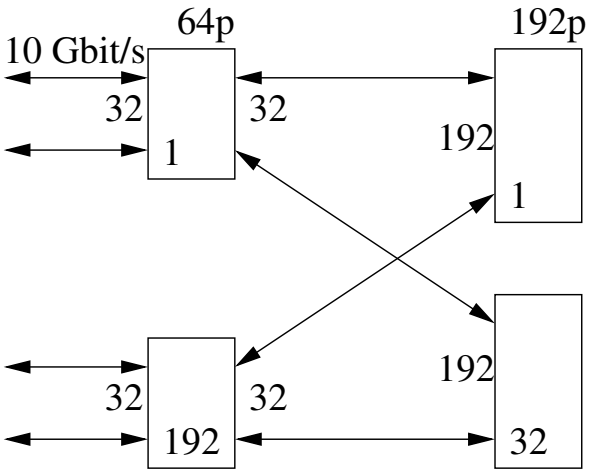


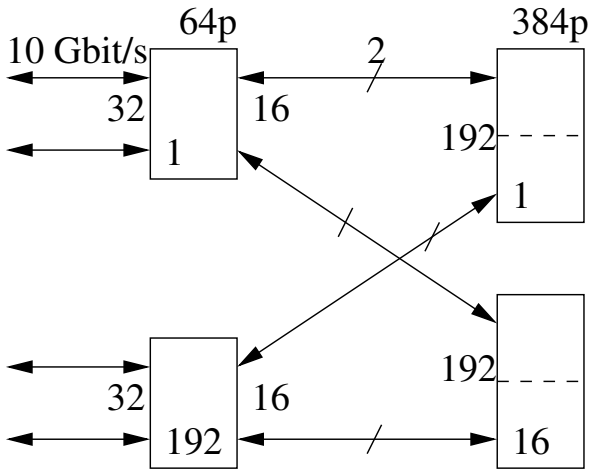
Example taken from "Cisco's Massively Scalable Data Center", 2009

Example of DCN design

6 144 servers

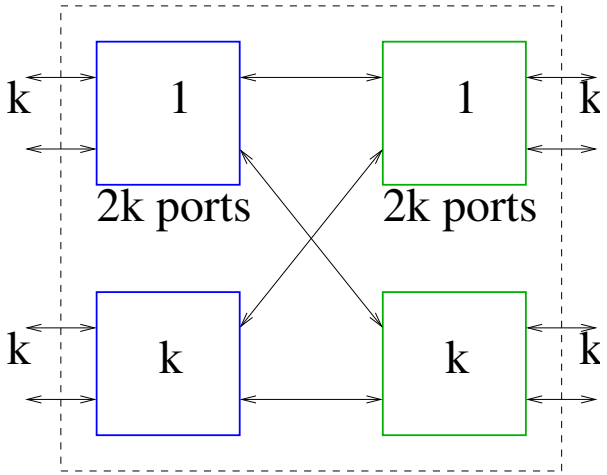
- 6 144 ports at 10 Gbit/s $\Rightarrow$ 61.44 Tbit/s
- alternative designs
  - 192 switches with 64 ports and 32 switches with 192 ports
  - 192 switches with 64 ports and 16 switches with 384 ports





Example taken from "Cisco's Massively Scalable Data Center", 2009

Recursive Leaf and Spine



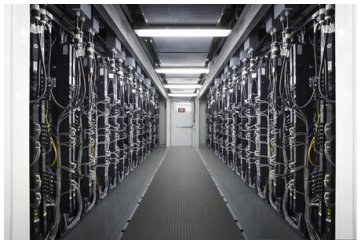
Leaf with $2k^2$ bidirectional ports

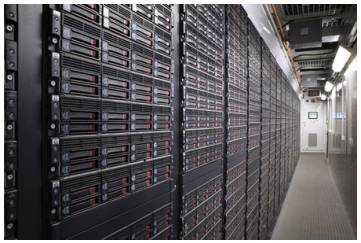
- k^2 ports to the servers and k^2 ports to the data center network
  - note that this cannot be used to interconnect directly $2k^2$ servers since the network would be blocking
- built with $2k$ switches with $2k$ ports

Physical infrastructure

POD (Point of Delivery)

A module or group of network, compute, storage, and application components that work together to deliver a network service. The PoD is a repeatable pattern, and its components increase the modularity, scalability, and manageability of data centers. *(taken from Wikipedia)*

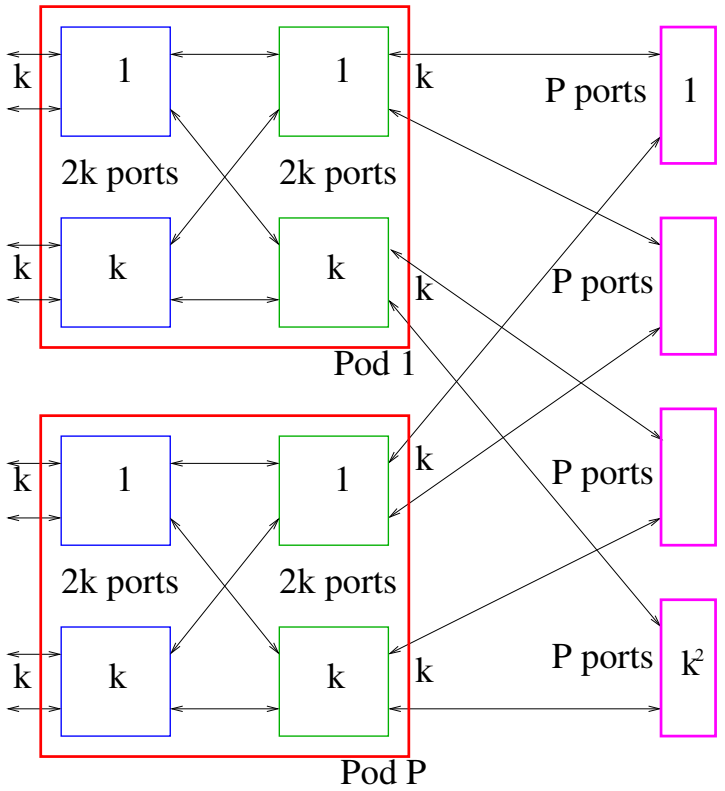




Intra-pod and inter-pod communications

Design

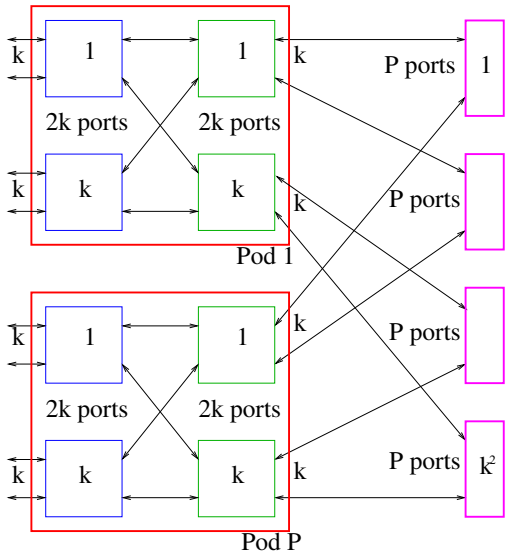
- $k^2 P$ servers
  - $2kP$ switches with $2k$ ports
  - k^2 switches with P ports
- choose $P = 2k$
  - $2k^3$ servers
  - $5k^2$ switches with $2k$ ports



Example of DCN design

Data center with 65 536 servers in 64 pods

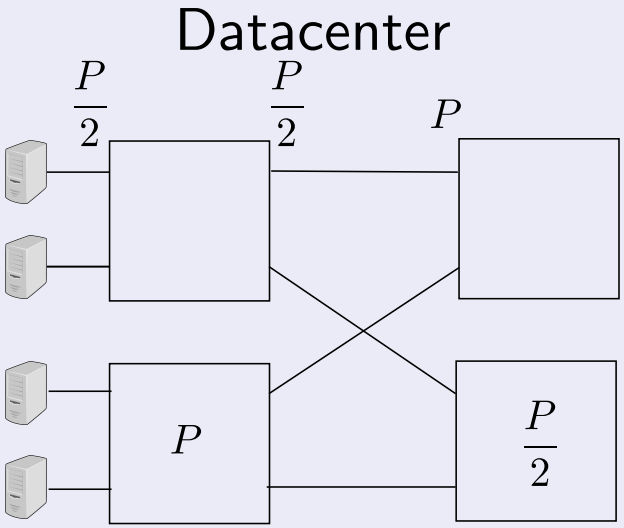
- 65 536 ports at 10 Gbit/s $\Rightarrow$ 655 Tbit/s
- $P = 64$ pods, $k = 32$
- in total 5 120 switches with 64 ports

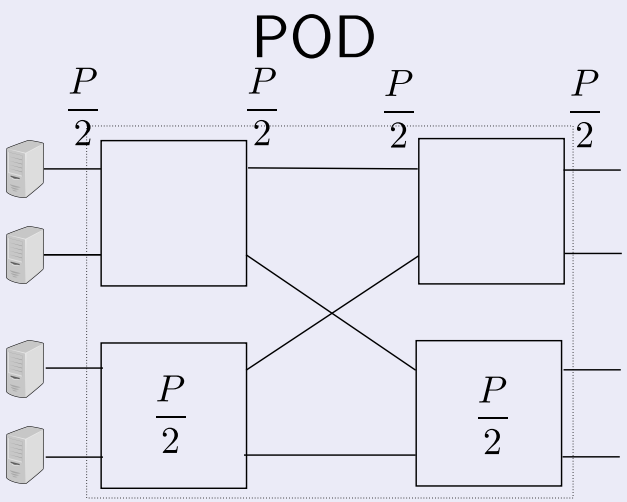


Recursive construction of a data center

- Assume only switches with P ports
- C_x is the number of switches in a data center with x servers
- $C_{x,y}$ is the number of switches in a POD connecting x servers to y spine switches

2-layers



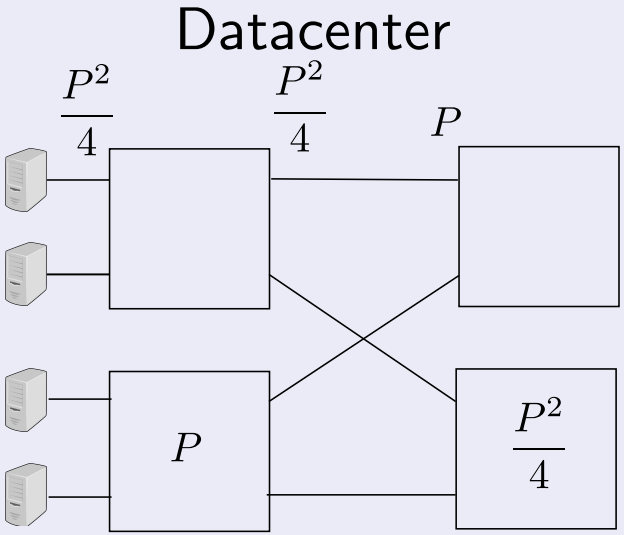


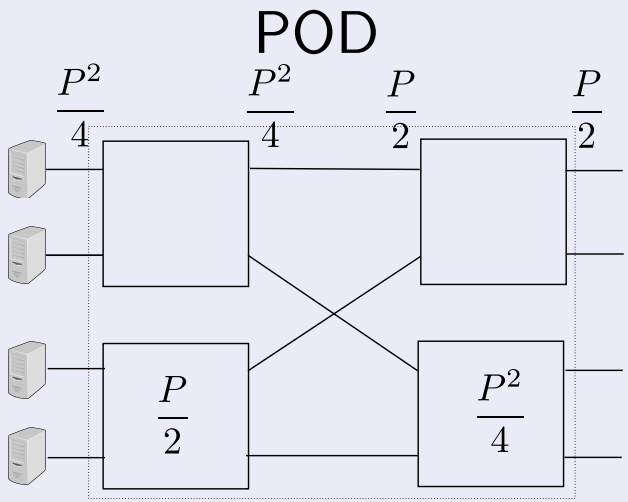
$$C_{\frac{P^2}{2}} = \frac{3}{2}P$$

$$C_{\frac{P^2}{4},\frac{P^2}{4}} = P$$

Recursive construction of a data center

3-layers



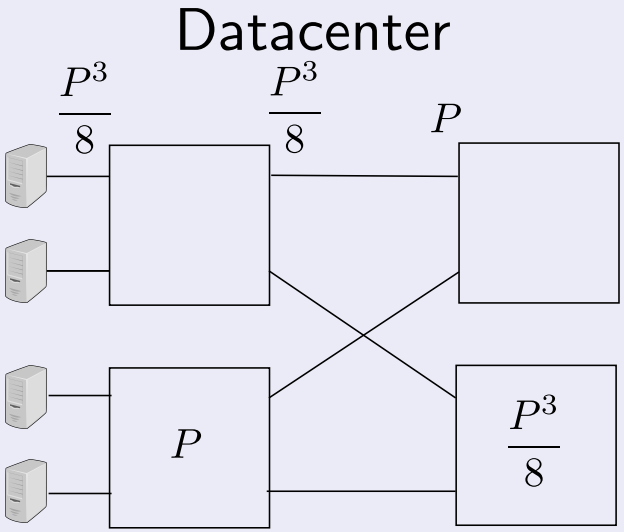


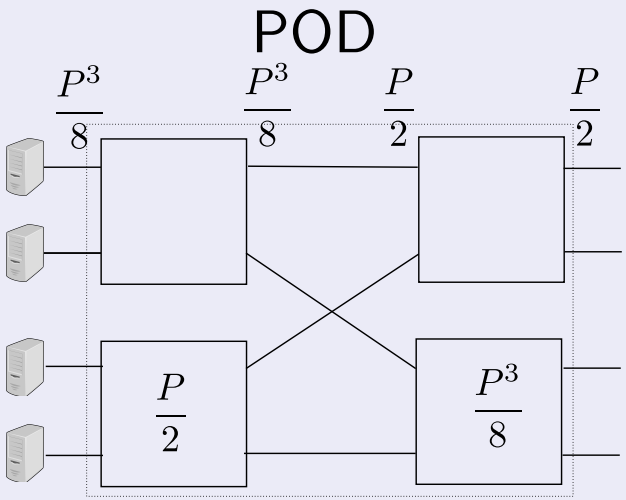
$$C_{\frac{P^3}{4}} = \frac{5}{4}P^2$$

$$C_{\frac{P^3}{8},\frac{P^3}{8}} = \frac{3}{4}P^2$$

Recursive construction of a data center

4-layers





$$C_{\frac{P^4}{8}} = \frac{7}{8}P^3$$

$$C_{\frac{P^4}{16},\frac{P^4}{16}} = \frac{1}{2}P^3$$

Recursive construction of a data center

Layers	Servers	Switches
1	P	1
2	$\frac{1}{2}P^2$	$\frac{3}{2}P$
3	$\frac{1}{4}P^3$	$\frac{5}{4}P^2$
4	$\frac{1}{8}P^4$	$\frac{7}{8}P^3$
5	$\frac{1}{16}P^5$	$\frac{9}{16}P^4$
L	$\frac{1}{2^{L-1}}P^L$	$\frac{(2L-1)}{2^{L-1}}P^{L-1}$

Optimality of recursive construction

- C_S is the total number of switches with P ports
- S is the number of servers, being $S = \frac{1}{2^{L-1}} P^L$

Total cost

$$C_S = \frac{S}{P}(2L-1) = \frac{S}{P}\frac{(2\log_2 S - \log_2 P - 1)}{(\log_2 P - 1)}$$

For very large number of servers ($S \to \infty$):

$$C_S \to \frac{2}{(P\log_2 P - P)} S \log_2 S = \Theta(S \log S)$$

which can be shown to be asymptotically optimal

Other topologies

Many other topologies have been devised. See for example:

- A. Hammadi, L. Mhamdi, "A survey on architectures and energy efficiency in Data Center Networks", Computer Communications, March 2014,

  `http://www.sciencedirect.com/science/article/pii/S0140366413002727`

- M.F. Bari, R. Boutaba, E. Esteves, L.Z. Granville, M. Podlesny, M.G. Rabbani, Qi Zhang, M.F. Zhani, "Data Center Network Virtualization: A Survey", IEEE Communications Surveys & Tutorials, 2013

Section 5

Google data center

Google scenario

World-wide coverage with tens of sites

Data center traffic

Bandwidth demand doubles every 12-15 months (faster than Internet)

- larger datasets (photo/video content, logs, Internet-connected sensors, etc.)
- web services
- internal applications (index generation, web search, serving ads, etc.)

Goggle's data center

Design approach

- multistage Clos topologies on commodity switch silicon
- centralized control
  - one configuration pushed to all the switches
  - SDN approach
- modular hardware design with simple, robust software

Reference paper:

[*Google*] A. Singh, et al., "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network", ACM SIGCOMM Computer Communication Reviews, Oct. 2015

Hardware alternative choices

Commodity switches

- cheap and simple, fast evolving
- intermittent capacity
- few available protocols, suitable for a single operator scenario

WAN switches

- complex and expensive, slow evolving
- highest availability, but here intermittent capacity is allowed
- many available protocols to support interoperability among multivendor WANs

Google's choice

- general-purpose, off-the-shelf switch components for commodity switches
- 5 generations in the period 2004-2015

Topology

Clos topology

- basic switching module with small number of ports
- scale to any size
  - limited by the control plane scalability
  - path diversity and redundancy
  - complexity of multiple equal cost paths
- complex fiber interconnection

Jupiter data center

Jupiter is the latest generation of operational data centers

Design

- multistage, recursively factorized Clos network
- basic block: 16 ports @ 40 Gbit/s, but each interface can be split in 4 ports @ 10 Gbit/s
  - e.g. 48 ports @ 10 Gbit/s plus 4 ports @ 40 Gbit/s
- overall bisection bandwidth: $512 \times 64 \times 40 = 1\,310\,720$ Gbit/s (1.3 Pbit/s)
- server: $192 \times 32 \times 64 = 393\,216$ servers @ 10 Gbit/s
- 3:1 oversubscription ratio between server capacity and network capacity

Jupiter topology

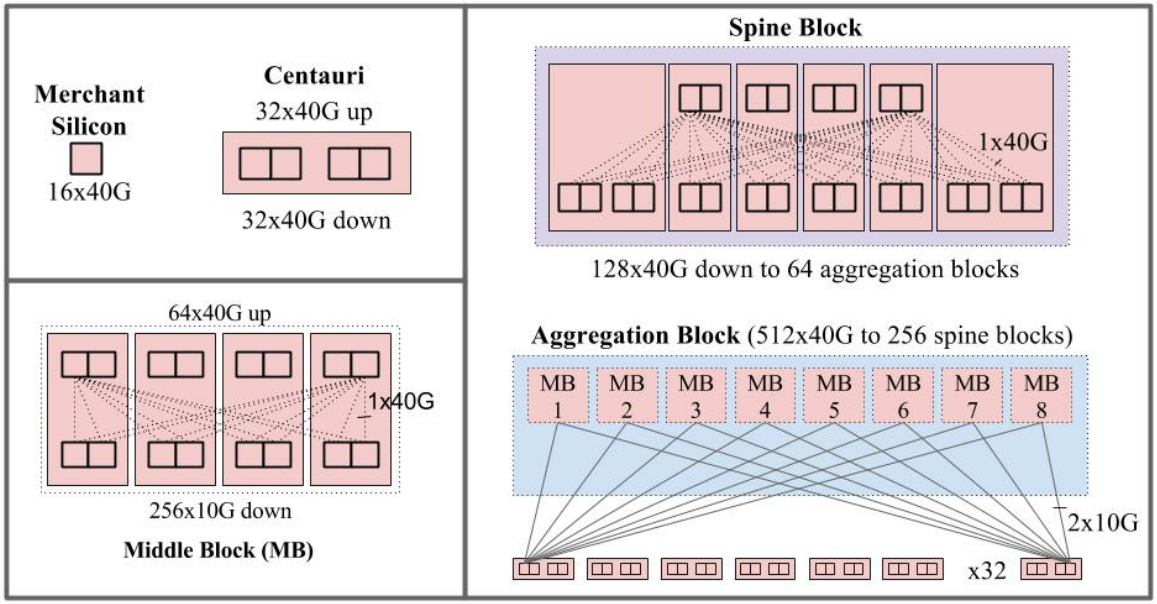


Figure 13: Building blocks used in the Jupiter topology.

Figure 14: Jupiter Middle blocks housed in racks.

Figures reproduced from [*Google*].

Connecting Jupiter to Internet

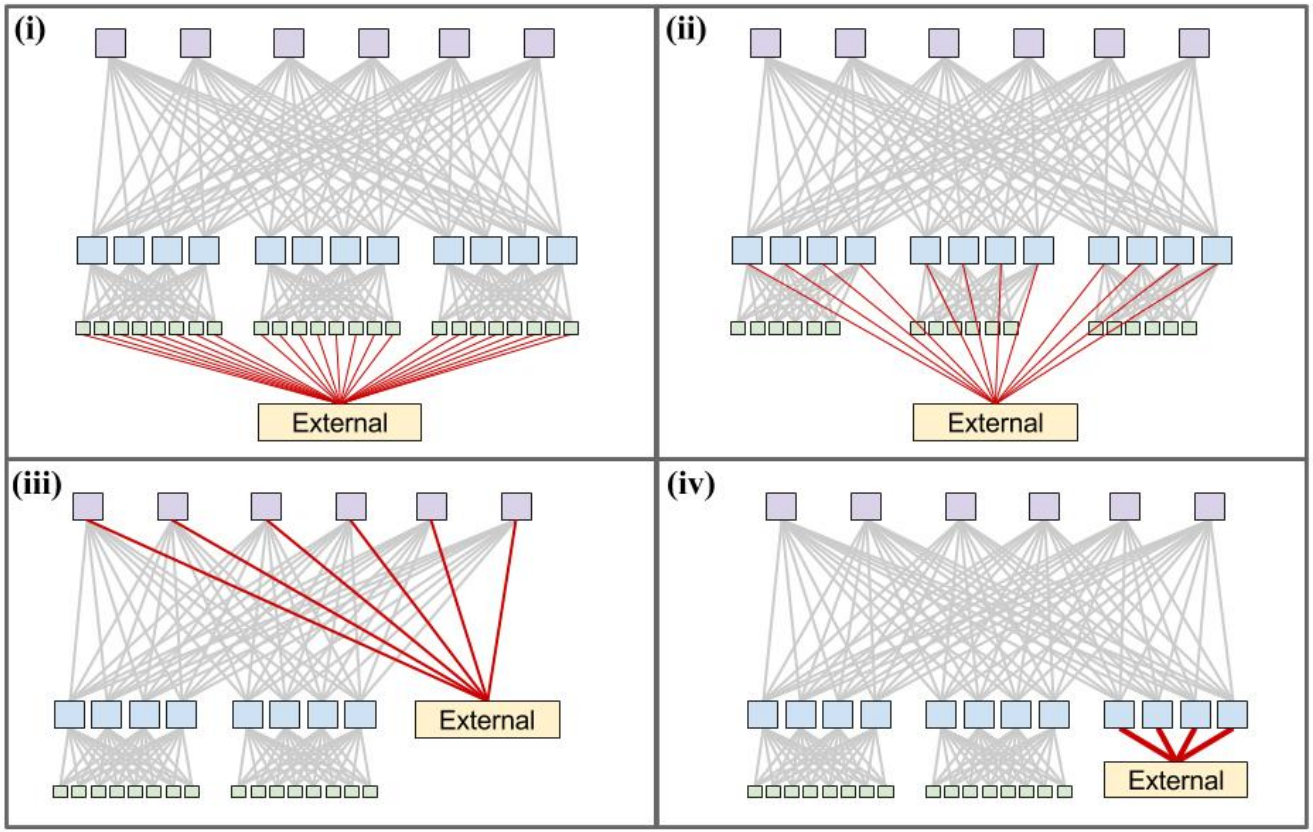


Figure 15: Four options to connect to the external network layer.

Figures reproduced from [*Google*].

Section 6

Hot issues

Traffic control in data centers

- End-users' perceived performance (QoE) depends heavily from the latency of experienced by the intra-DC traffic
- Flow completion time is the main performance metric to minimize for the intra-DC traffic
  - New transport protocols
  - New queueing and scheduling algorithms at both the servers and the switches
  - Advanced traffic engineering schemes

Hybrid optical-electronic data center

Optical networks

- slow switching (only at flow level, at $>$ *ms* scale)
- very high bandwidth and low energy consumption

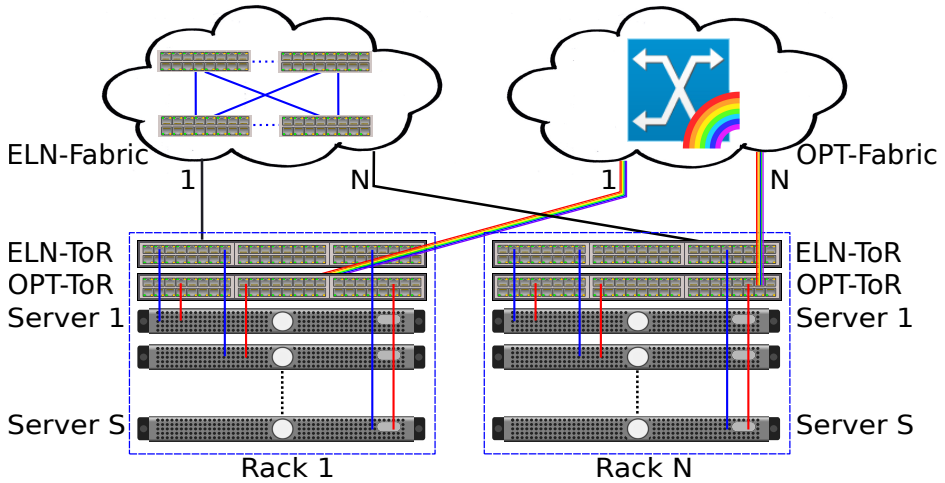
Electronic networks

- fast switching (at packet level, at *ns* scale)
- high bandwidth but high energy consumption

Main idea

Deploy two interconnection networks in parallel

- optical network for elephant flows (i.e., fast circuit switching)
- electronic network for mice flows (packet switching)



Open compute project

- `http://www.opencompute.org/`
- open data center architecture, sponsored by Facebook
  - mechanical specifications (connectors)
  - electric powering, cooling methodology
  - storage and server implementation
- leaf-and-spine architecture

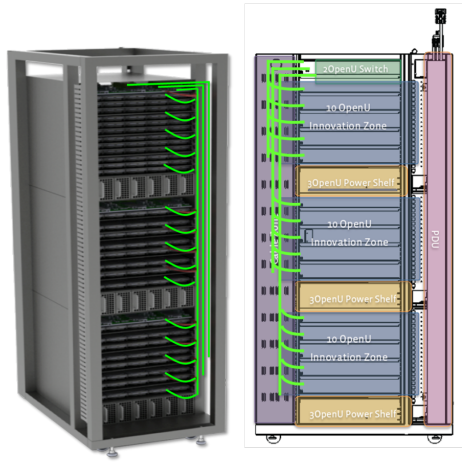


Figure 3.1: Open Rack with Optical Interconnect. In this architectural concept the green lines represent optical fiber cables terminated with the New Photonic Connector. They connect the various compute systems within the rack to the Top of Rack (TOR) switch. The optical fibers could contain up to 64 fibers and still support the described New Photonic Connector mechanical guidelines.

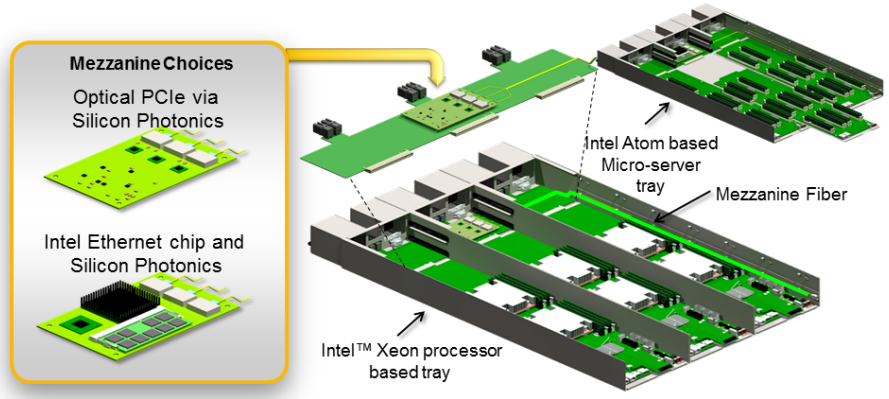


Figure 3.4: An example of a Photonically Enabled Architecture in an Open Compute mechanical arrangement using a Mezzanine Fiber – In this concept the New Photonic Connector cable concept is used to enable a reduced cable burden, and front panel access, through the use of silicon photonics modules and the modular architectural concepts which were discussed earlier.

Energy consumption

Data centers are one of the largest and fastest growing consumers of electricity

- *In 2010, collectively data centers consumed around 1.5% of the total electricity used world-wide* (J. Koomey. Growth in Data Center Electricity Use 2005 to 2010, 2011. Analytic Press)

"Green" data centers

Data centers partially or completely powered by renewables (e.g., solar, wind energy)

- self-generation: use their own renewable energy
- co-location: use the energy from existing nearby plants

VNF migration

Migrating NVFs (or containers) is challenging for classical data centers and for 5G edge clouds (exacerbated by the users' mobility)

Non-live migration

- Requires to temporarily stop the service
- Phases
  1. Shutdown or suspend the VM/container
  2. The whole VM/container instance (data and state) are copied to the destination server
  3. Restart the VM/container in the new server
  4. Make available the service again
- Simple implementation

VNF migration

Live migration

- The VNF instance (MV or container) is copied to the destination server while the service is running
- Limited downtime

Challenges for live migration

- Memory Data Migration
  - To run a migrated VNF from the suspended point, all the active states of the migrated VNF must be transmitted to the destination server
- Storage Data Migration
  - Virtual disk of the migrated VNF must be transmitted to the target site
- Network Connection Continuity
  - Once a VNF is migrated to the new location, some strategies are required to make the corresponding VMs/containers reachable to the end users

Live VNF migration

Memory data migration

- Phases for the migration
  1. Push phase: the memory pages are transferred iteratively to the destination server, while the VM/container is still running on the source server
  2. Stop and copy phase: VM/container running on the source server is halted and all the related memory data are transmitted to the destination server
  3. Pull phase: VM/container starts running on the destination server and, if a page fault happens, the required pages are fetched from the source server

VM migration and the network

Network connection connectivity

- Layer 2 Solution
  - one LAN for the whole data center
  - extend the LAN to multiple data centers using proprietary solutions
- Layer 3 Solution
  - IP tunneling and Dynamic DNS
  - problems: cannot migrate existing connections
  - preserve existing connections and redirect new connections
- Layer 4 Solution
  - reestablishing the TCP connection by sending a SYN packet at the host with the updated IP address of the server
  - application layer is also referred