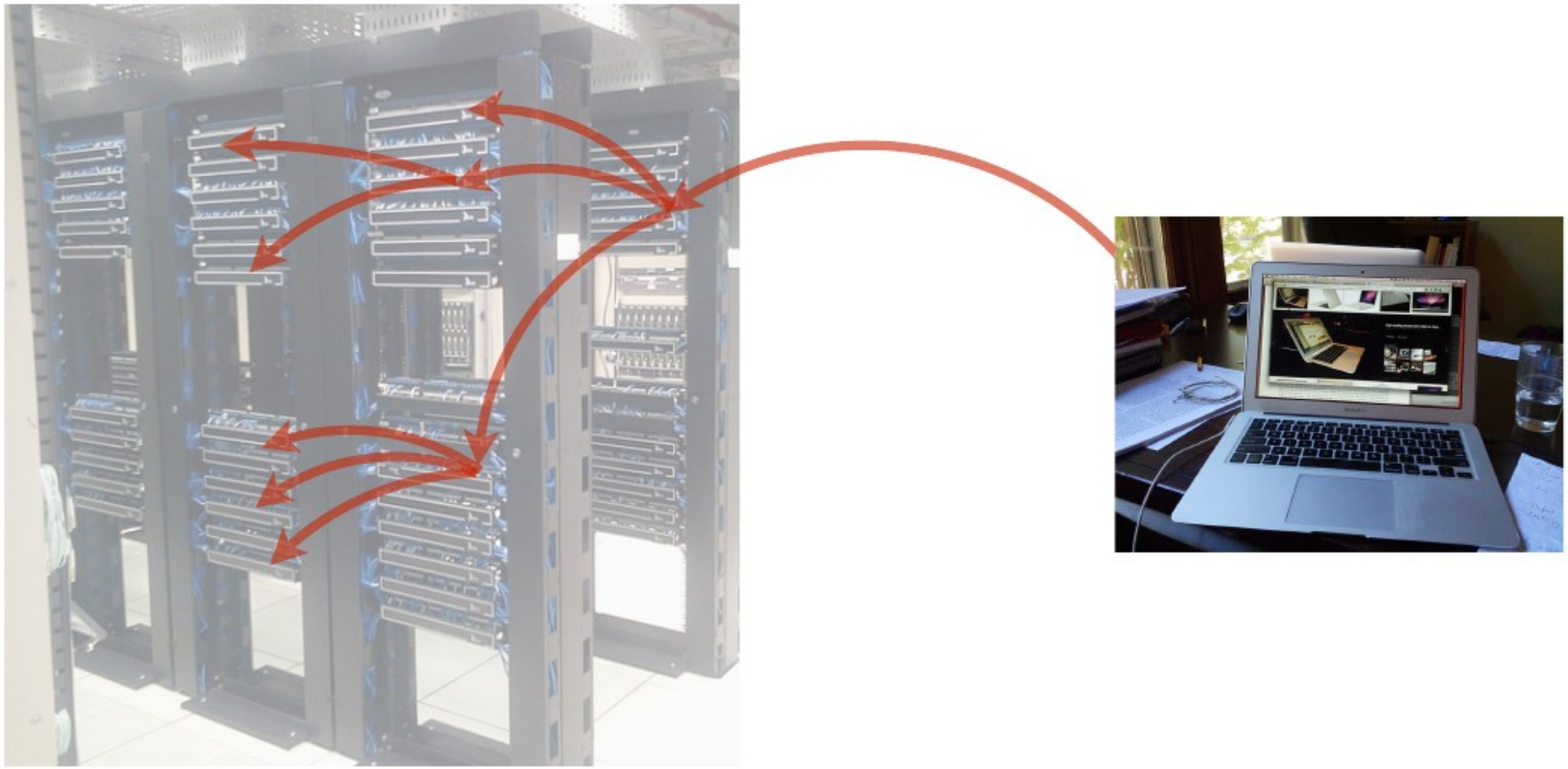
CLOUD COMPUTING
Network Structure

Zeinab Zali
Isfahan University Of Technology

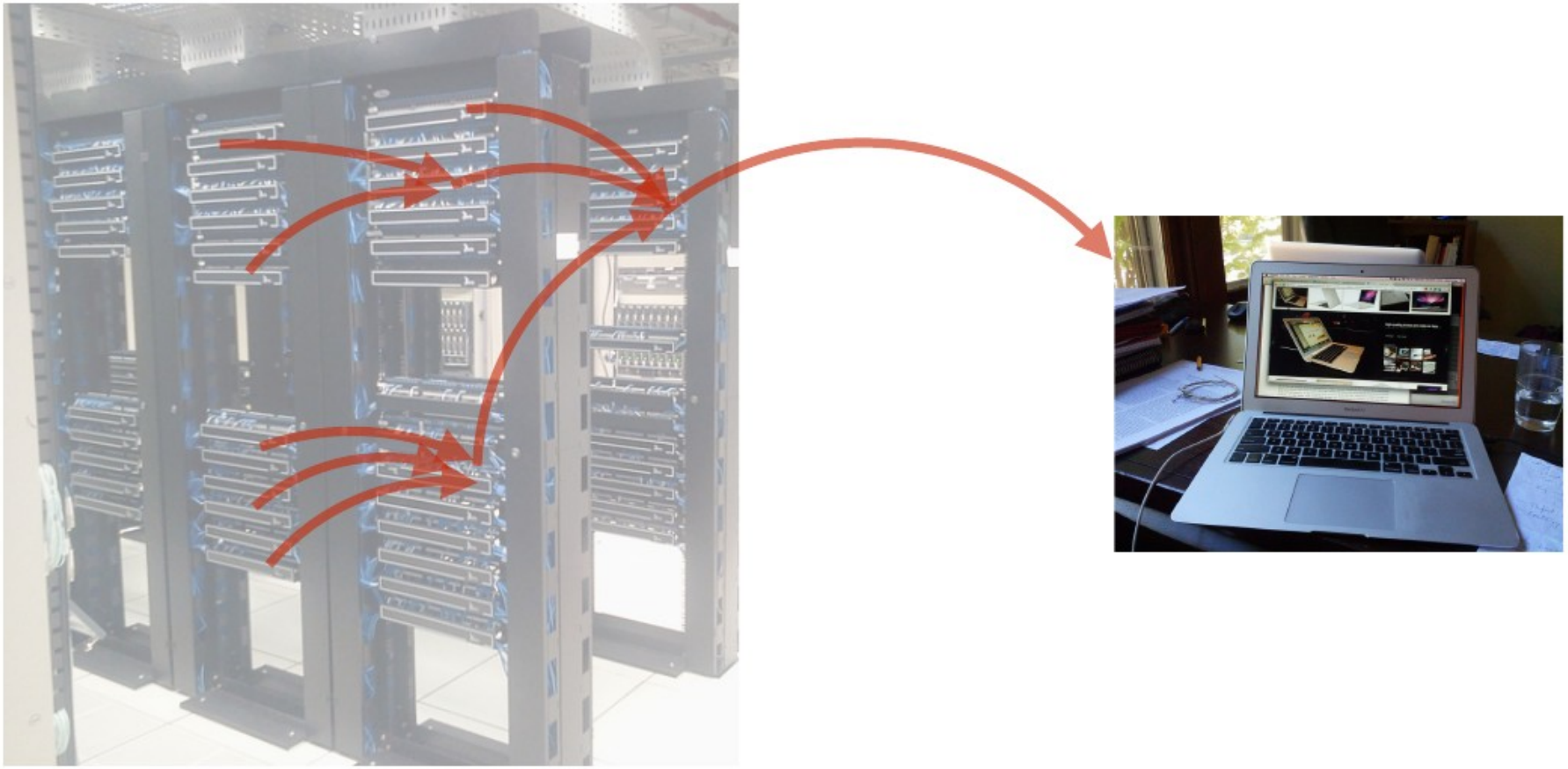
References: Cloud computing: Theory and practice, Chapter 5
Cloud Networking course, Illinois university

Data Center Traffics

A Web Search



A Web Search



Extremely short response deadlines for each server — 10ms

DataCenter Traffic

- loading one of facebook popular pages results in an average of 521 distinct items fetched from memcache
  - Thus web servers have to routinely communicate with many memcached servers to satisfy a user request.
    - all web servers communicate with every memcached server in a short period of time.
    - This all-to-all communication pattern can cause high congestion or allow a single server to become the bottleneck for many web servers

Big data analytics

- Hadoop
- Spark
- Dryad
- Database joins
- ...

A Sample DataCenter

- Facebook’s network (2015) consists of multiple datacenter sites and a backbone connecting these sites.
- Each datacenter site contains one or more datacenter buildings
- Each datacenter contains multiple clusters.
- A cluster is considered a unit of deployment in Facebook datacenters

Traffic characteristics

- What does data center traffic look like?
  - It depends on applications, scale, network design, ...
- Facebook: “machine to machine” traffic is several orders of magnitude larger than what goes out to the Internet

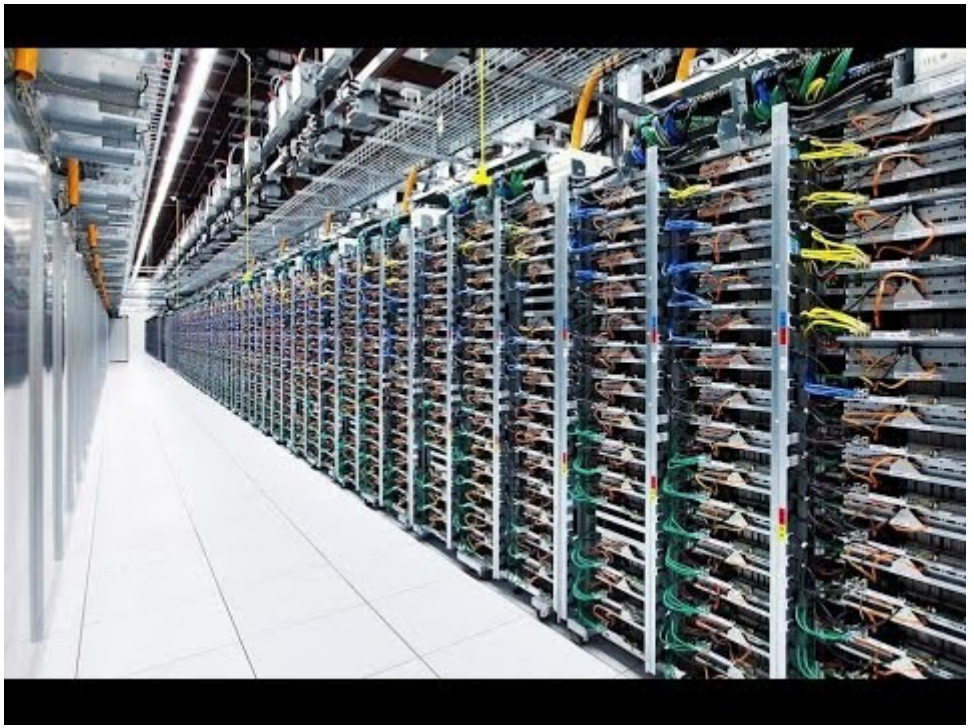
Locality	All	Hadoop	FE	Svc.	Cache	DB
Rack	12.9	13.3	2.7	12.1	0.2	0
Cluster	57.5	80.9	81.3	56.3	13.0	30.7
DC	11.9	3.3	7.3	15.7	40.7	34.5
Inter-DC	17.7	2.5	8.6	15.9	16.1	34.8
Percentage		23.7	21.5	18.0	10.2	5.2

TCP incast

- Applications running in data center networks exhibits barrier-synchronized and many-to-one communication pattern
  - multiple workers simultaneously transmit bulk of data to a single aggregator using TCP protocol
- This synchronized transmission may overload aggregator's switch buffer, which leads to severe packet loss and overall throughput degradation.
- This overall throughput degradation is called as TCP Incast problem in data center network

Data Center traffic implications

- Data center internal traffic is BIG
- Tight deadlines for network I/O
  - Having multiple stages for a request response workflow
- Congestion and TCP incast
- Need for isolation across applications
  - Applications with different objects sharing the network



DC Physical Structure

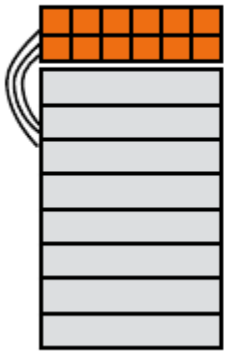
Additional References:

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

Inside the Social Network's (Datacenter) Network, Facebook

A Scalable, Commodity Data Center Network Architecture, University of california

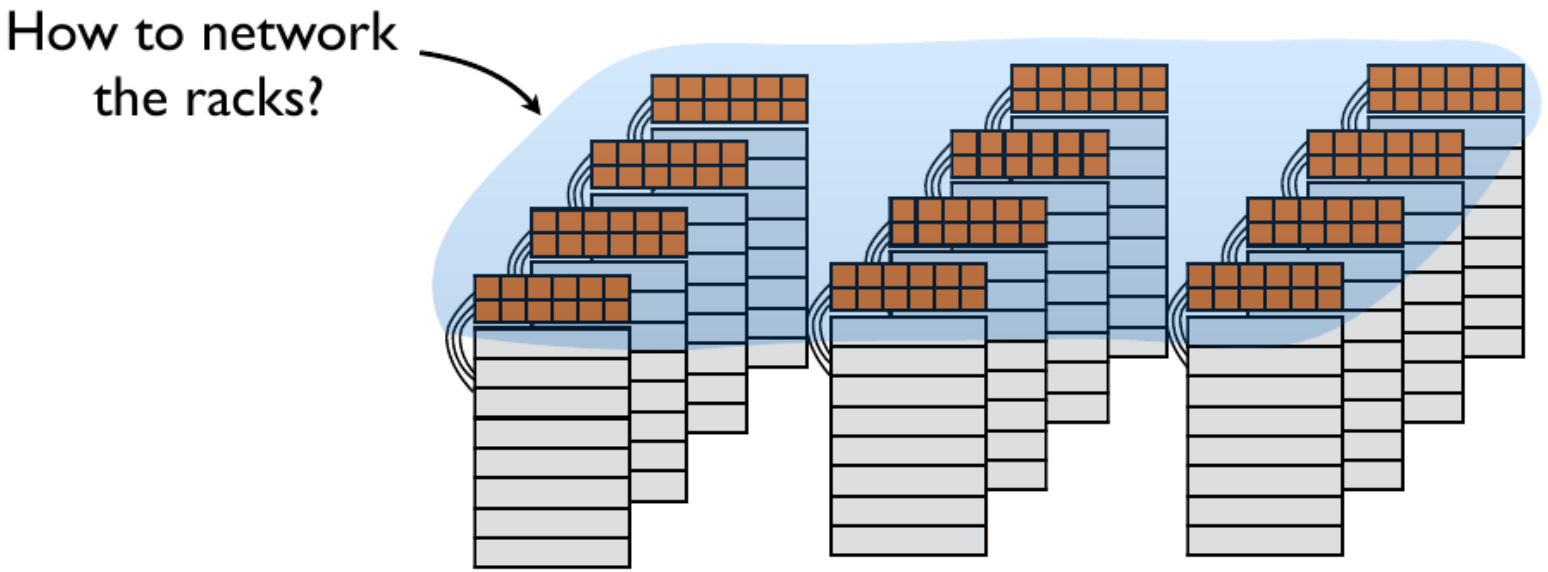
A rack server



A top-of-rack switch

A rack of servers

Lots of racks

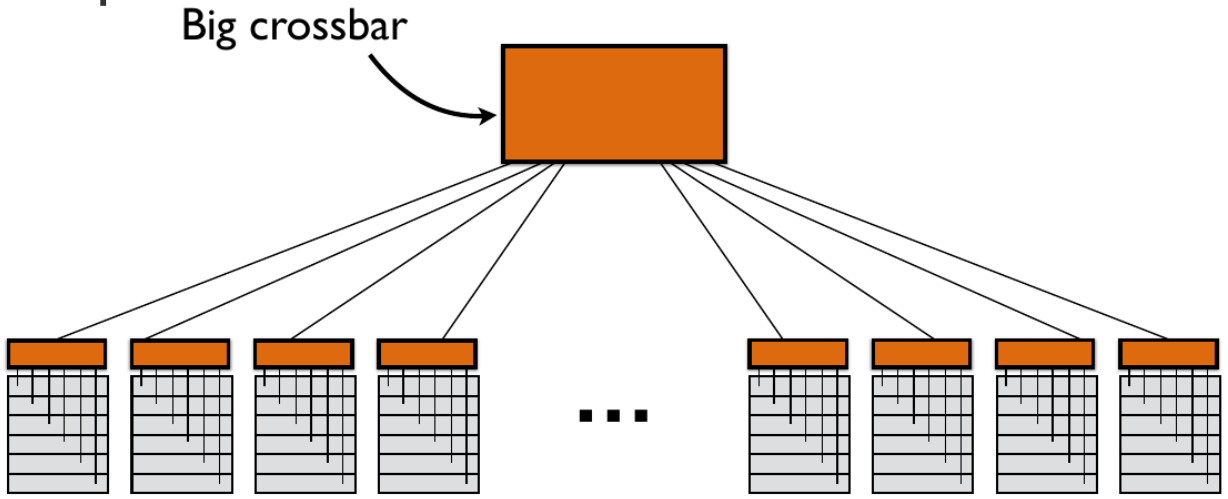


topology related concepts

- The topology of an interconnection network determines
  - Network diameter: the average distance between all pairs of nodes
  - Bisection width: the minimum number of links cut to partition the network into two halves
  - Bisection bandwidth: the bandwidth available between the 2 partitions of a bisected network
    - It accounts for the bottleneck bandwidth of the entire network, so it is a good metric for bandwidth characteristics of the network
  - Cost and power consumption

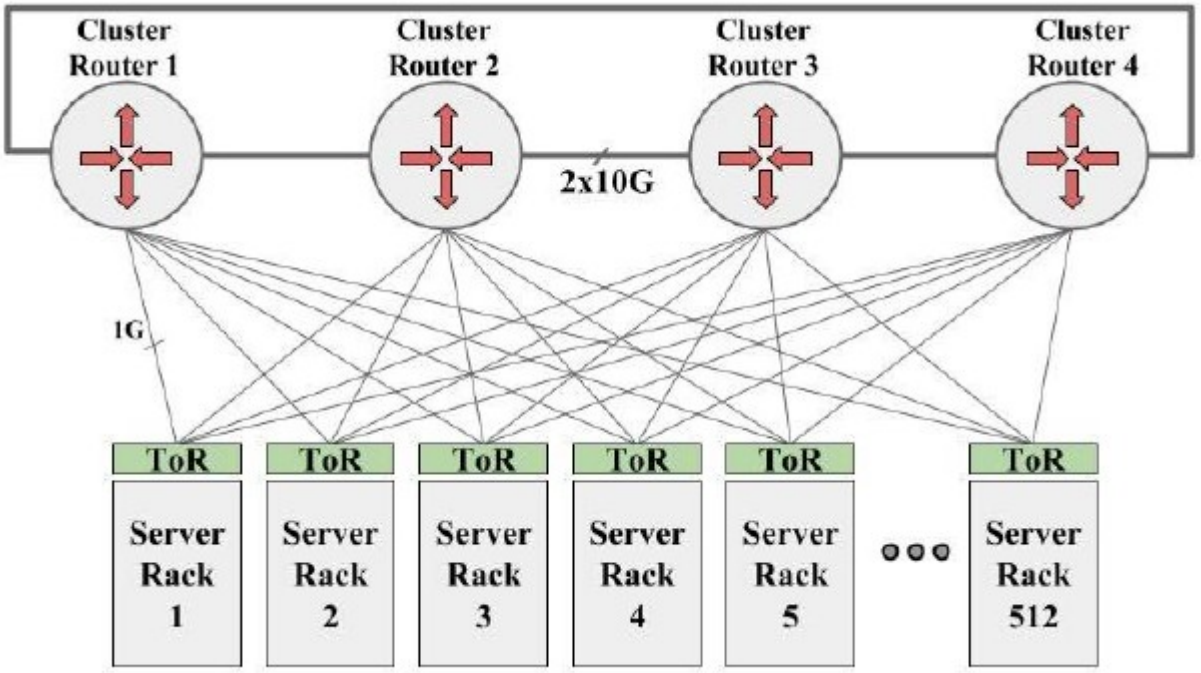
Big switch approch (I)

- a crossbar switch with M inputs and N outputs has a matrix with M × N cross-points
- At each crosspoint is a switch; when closed, it connects one of the inputs to one of the outputs.
- A crossbar is a non-blocking switch: concurrent connections do not prevent connecting other inputs to other outputs



Big switch approch (II)

- A traditional 2Tbps four-post cluster of google (2004). Top of Rack (ToR) switches serving 40 1G-connected servers were connected via 1G links to four 512 1G port Cluster Routers (CRs) connected with 10G sidelinks



Big switch problems

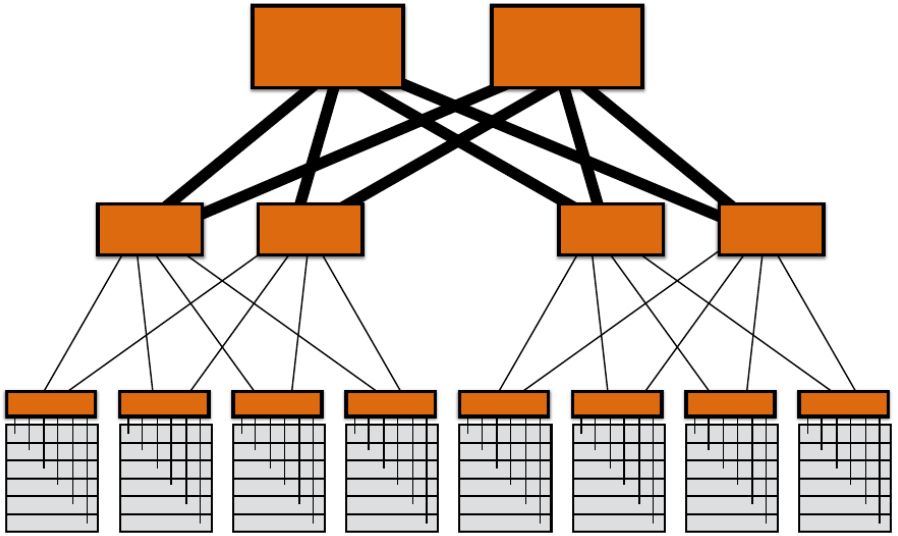
- If all these servers wanted to talk to the rest of the network, there's only a tenth of the capacity available.
  - As a result, the traffic has to be very local.
  - This restricts application deployment and skill.
  - High bandwidth applications had to fit under a single ToR to avoid the heavily oversubscribed ToR uplinks
- The large switches are very expensive because of their high port density.

Big switch problems

- Non-compatible next generation hardwares :if you're looking at this as a 1GB network, the next generation might be 10GB, and that will not be available at the same port density
- the most serious limitation to this approach is scalability
  - You're limited fundamentally in how many servers you can have by the port count of these large switches

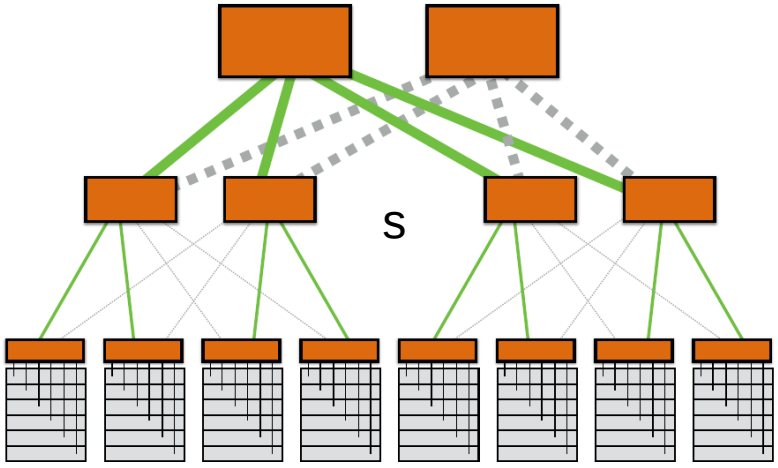
Tree Network (I)

- Instead of having just one big switch or a few big switches, we can organize the network hierarchy of switches of increasing size
  - at the bottom still set the racks and their top of rack switches, which further connect to somewhat larger switches and further on to bigger switches



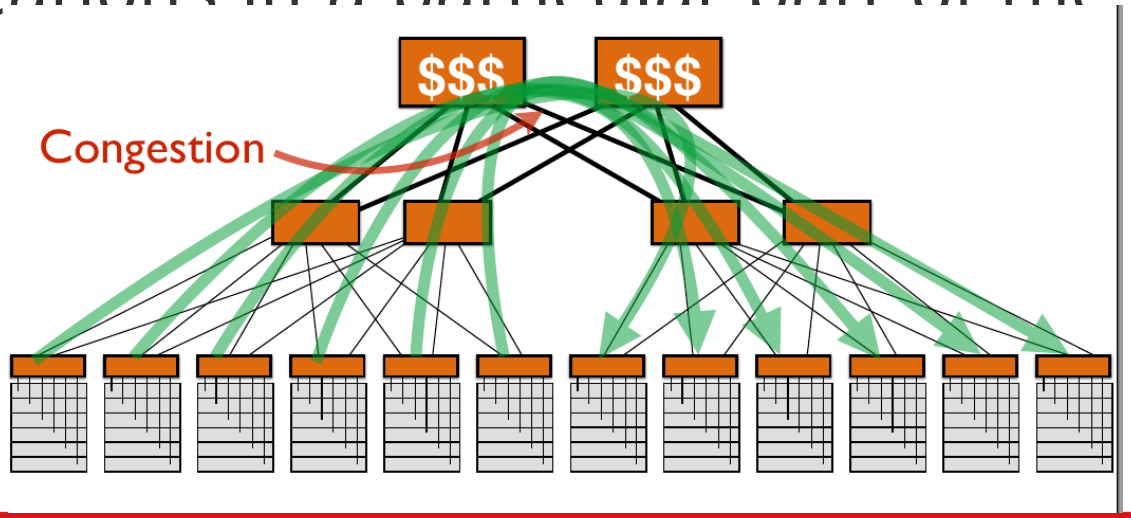
Tree Network (II)

- we might have more leaves or more racks here at the expense of limited capacity within racks
- It is actually a tree and some redundancy



Tree Network problems (I)

- As machine to machine traffic increases, if half of the top of rack switches send all their traffic to the other half, there will be congestion at the root of this network.
  - This restricts application deployment, in that you have to carefully plan to deploy applications in a particular part of the network

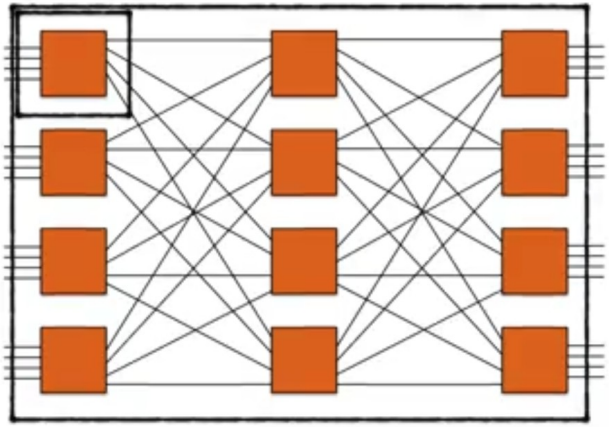


Tree Network problems (II)

- A couple of failures can partition this entire network, leaving it dysfunctional.
- It also suffers from some of the same drawbacks that we saw with the big switch approach
  - The switches at the top of the hierarchy are still quite expensive and your skill or capacity are quite limited

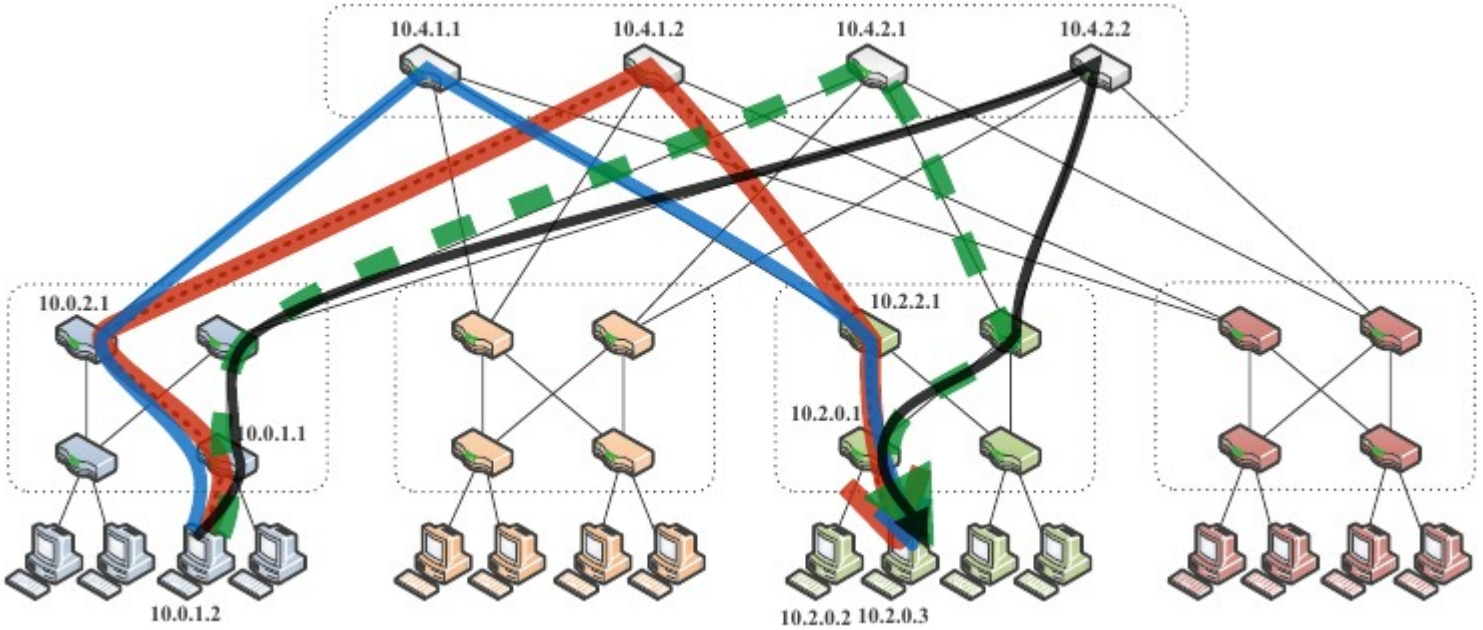
Clos networks

- A Clos network is a multistage non-blocking network with an odd number of stages
  - Difference with big switch: all the building blocks of the network are quite small and we're putting them together to build a larger switch
- There are two ways scaling such a topology
  - increase the no. of layers
  - increase the no. of devices' ports
- Results:
  - increase the scale/bisection of datacenter
  - Supports fault tolerance



Fat tree

- Fat tree: a folded Clos topology that the input and the output networks share switch modules
- Fat-trees have additional links to increase the bandwidth near the root of the tree

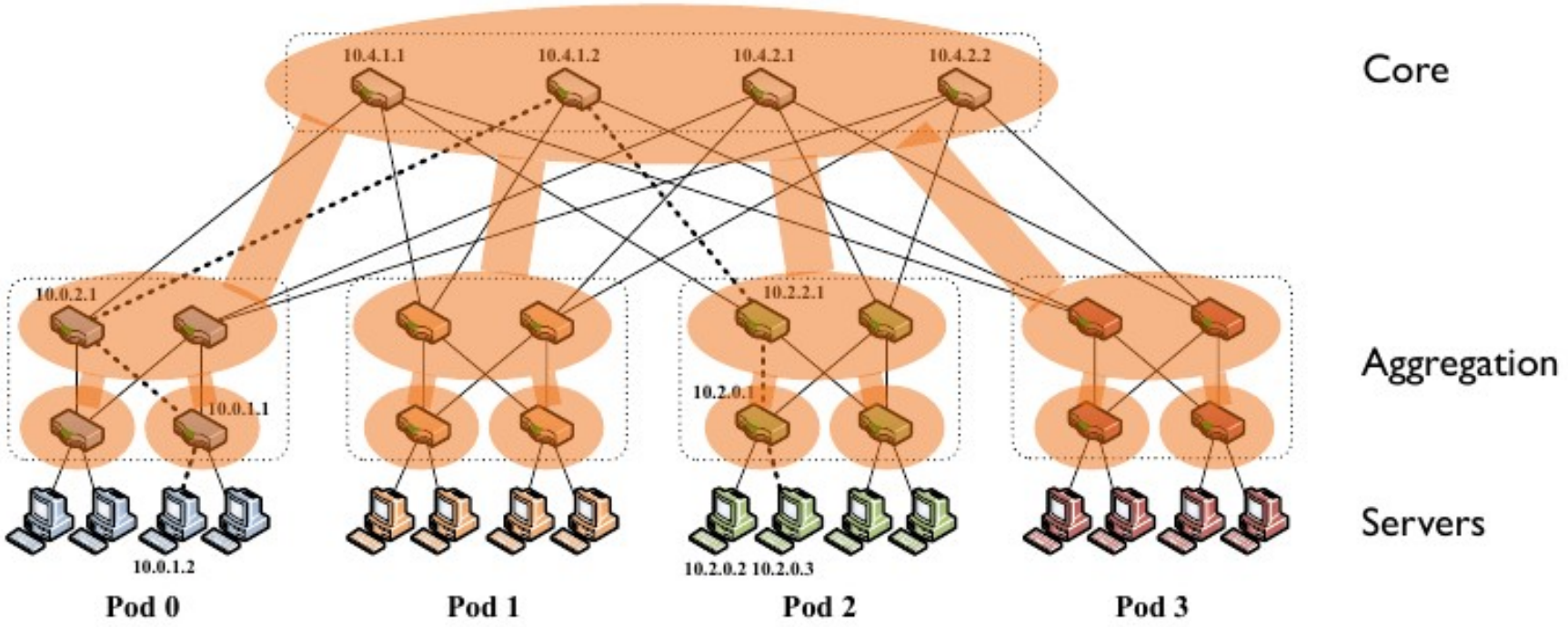


Fat Tree

- Some set of paths in a fat-tree will saturate all bandwidth available to the end hosts for arbitrary communication patterns
- A fat-tree communication architecture can be built with cheap commodity parts as all switching elements of a fat-tree are identical

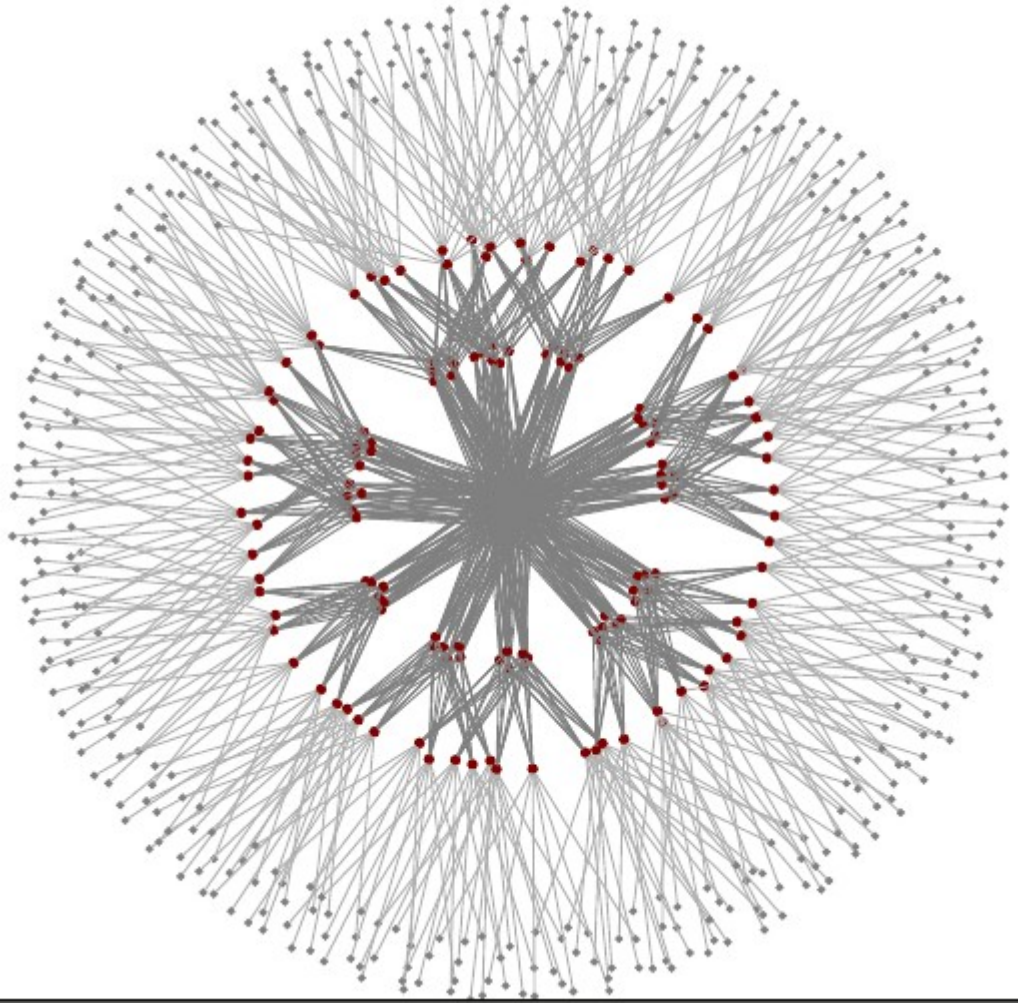
Fat tree

- This topology is in a sense trying to achieve what the tree hierarchy fails to achieve
  - scalability with high capacity throughout

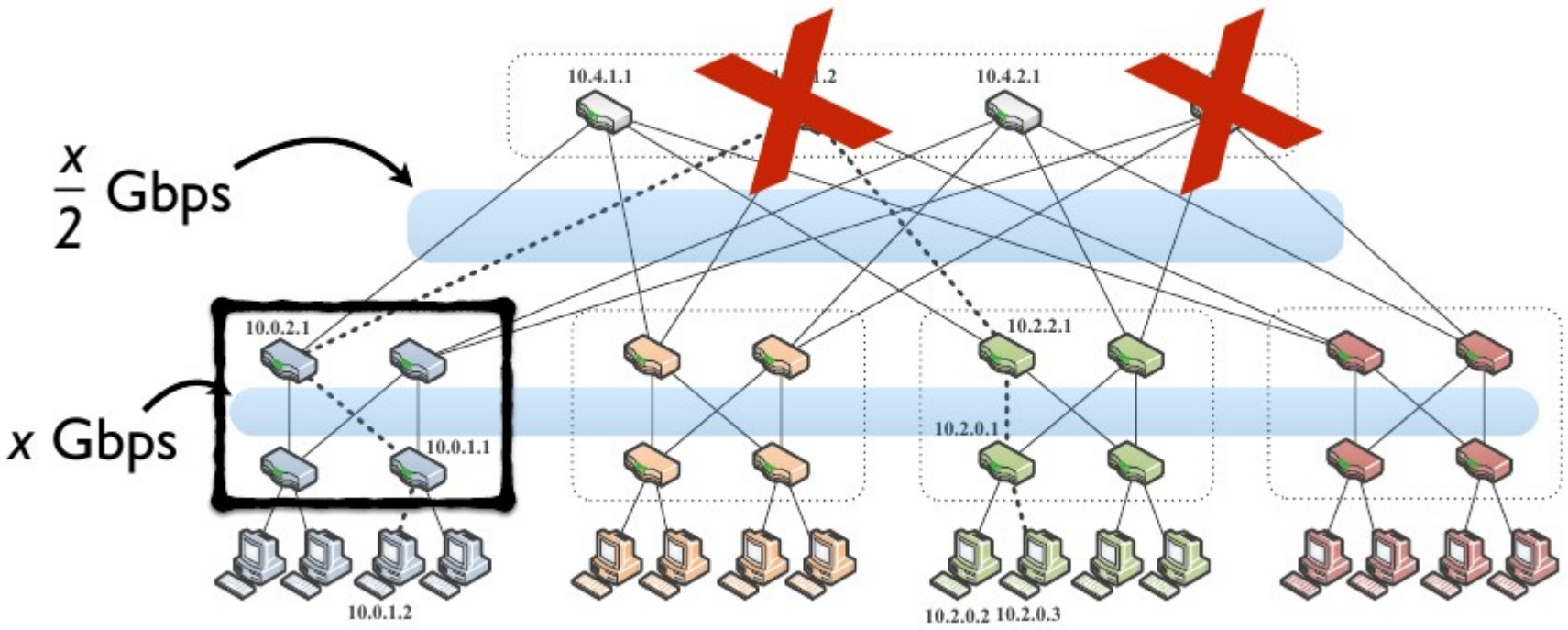


Fat tree example

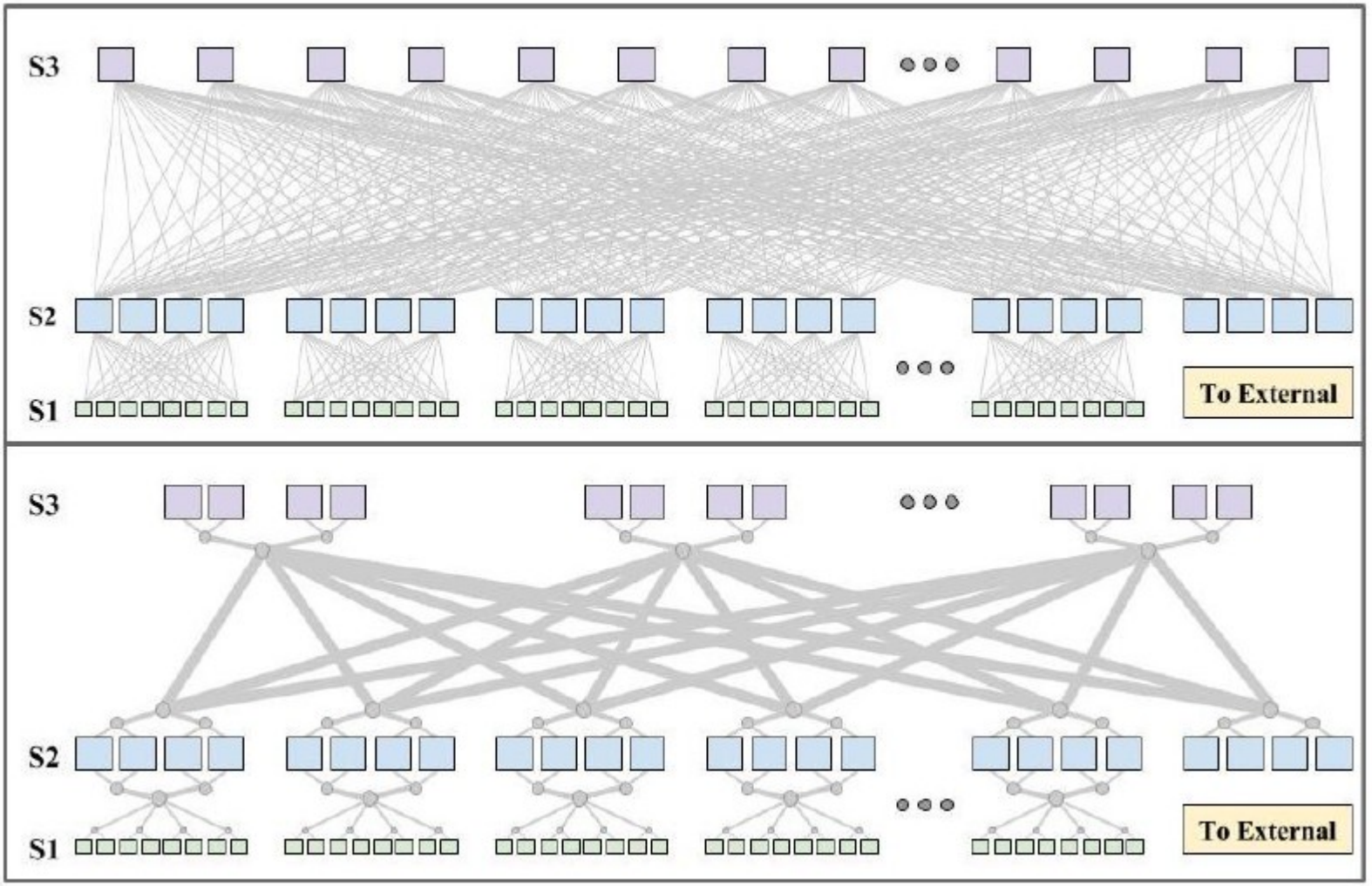
- A large number of redundants



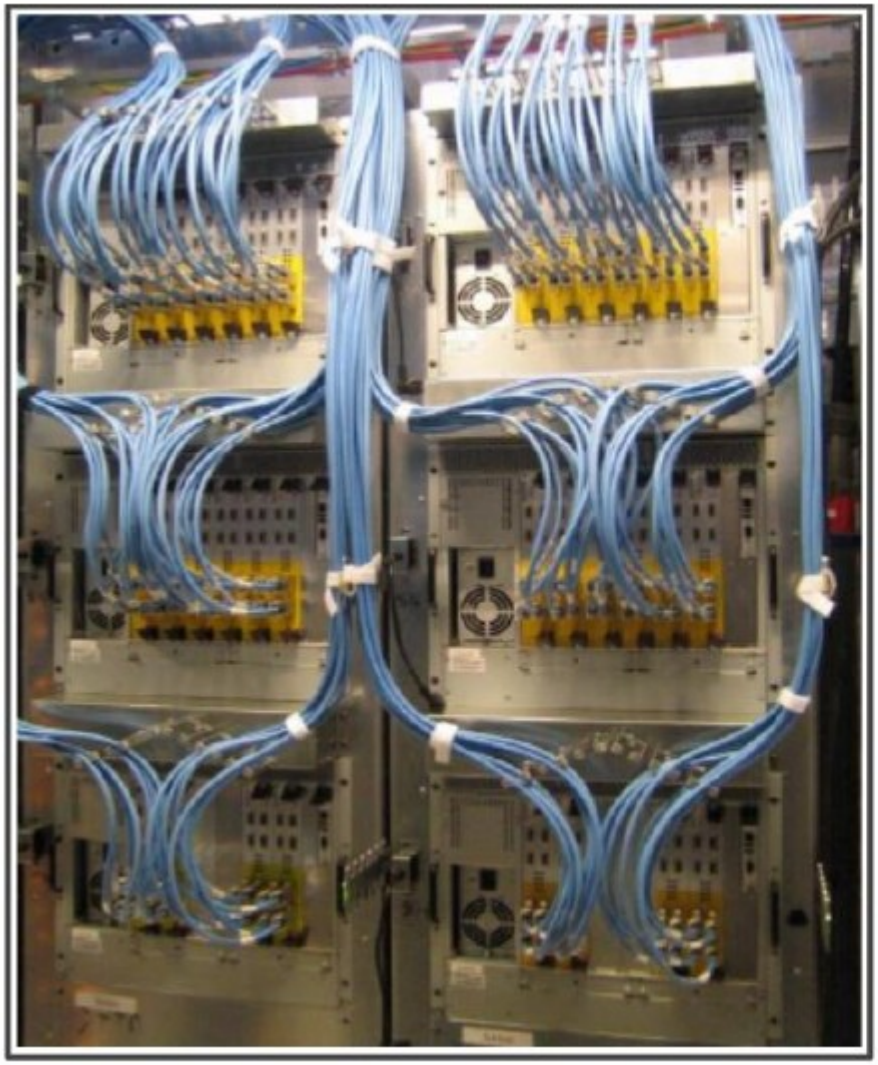
Fat tree: half the root capacity



Google



Google



Google