Cloud and Datacenter Networking

Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione DIETI
Laurea Magistrale in Ingegneria Informatica

Prof. Roberto Canonico

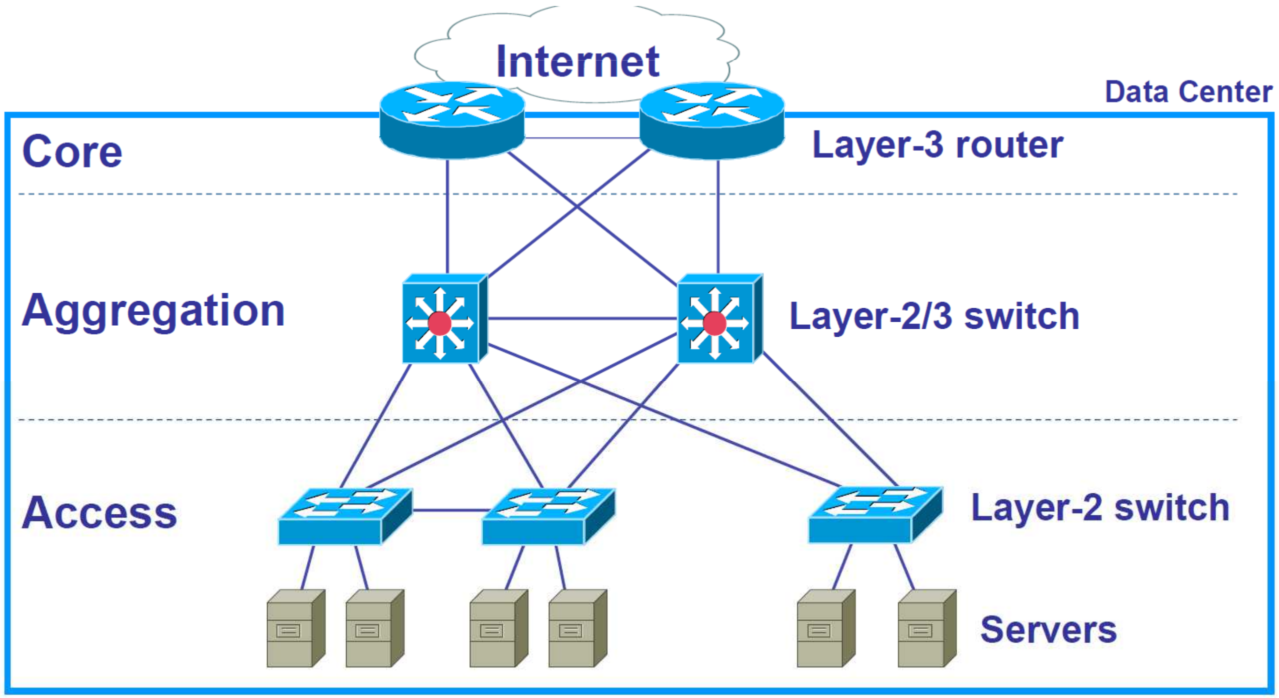
Datacenter networking infrastructure
Part II

Lesson outline

- Loops management in Ethernet networks: STP
- Limitations of traditional datacenter network topologies
- Datacenter networks and alternate paths exploitation
- TRILL protocol
- ECMP protocol

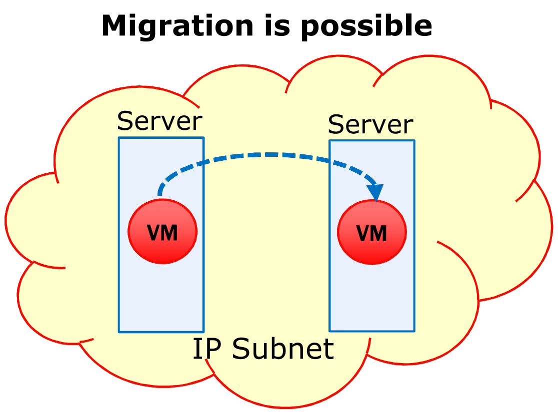
Traditional DC network architecture

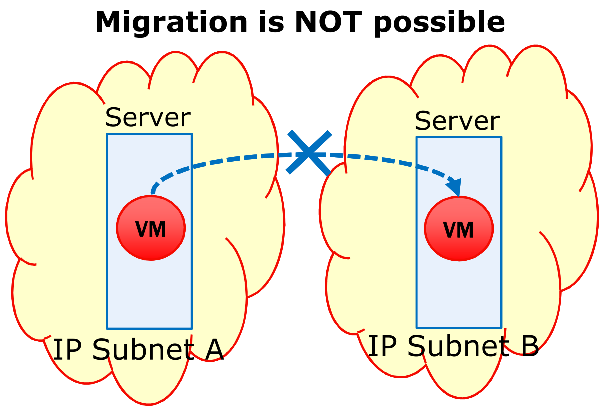
- In a datacenter computers are organized in racks to ease management and cabling and for a more efficient utilization of space
- Datacenter networks are typically organized in hierarchical structure
- Server NICs (2/4 per server) connected to an *access layer infrastructure*
- *Access layer* switches, in turn, are connected to an *aggregation layer infrastructure*
- The whole datacenter is connected to the outside world (e.g. the Internet, or a private WAN) through a *core layer infrastructure* operating at layer 3 (*IP routing*)



Datacenter networking and virtualization

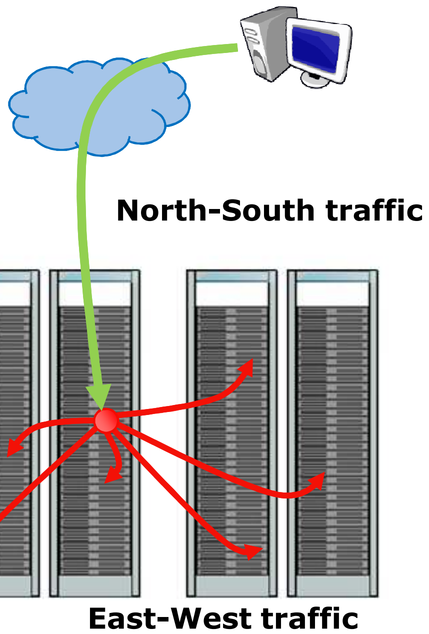
- In modern datacenters, servers' computational resources may be efficiently utilized thanks to a pervasive use of host virtualization technologies
  - KVM, Xen, Vmware, …
- A single server may host tens of Virtual Machines (VM) each configured with one or more virtual network interface cards (vNICs) with MAC and IP addresses of their own
- Modern virtualization technologies allow to migrate a VM from one server to another with a negligible downtime (*live migration*)
- Condition for live migration transparency to running applications:
  - a migrating VM must keep its own IP address in the new position
- If the datacenter network partitions the whole network infrastructure into clusters assigned to different IP subnets, a VM cannot migrate outside of its original cluster





Datacenter traffic analysis

- <u>Observation</u>: in large scale datacenters, traffic between hosts located in the DC (*East-West traffic* or *Machine-to-Machine traffic, m2m*) exceeds traffic exchanged with the outside world (*North-South traffic*)
  - Facebook: m2m traffic doubles in less than a year
- <u>Reasons</u>: modern cloud applications a single client-generated interaction produces multiple server-side queries and computation
  - Eg. Hints in a research textbox, customized ads, service mashups relying on multiple database queries, etc.
  - Only a fraction of server-side produced data is returned to the client
  - An example observed in FB network (*):
    a single HTTP request produced
    88 cache lookups (648 KB),
    35 database lookups (25.6 KB), and
    392 remote procedure calls (257 KB)
- <u>Conclusion</u>: the DC aggregation layer MUST NOT BE a bottleneck for communications

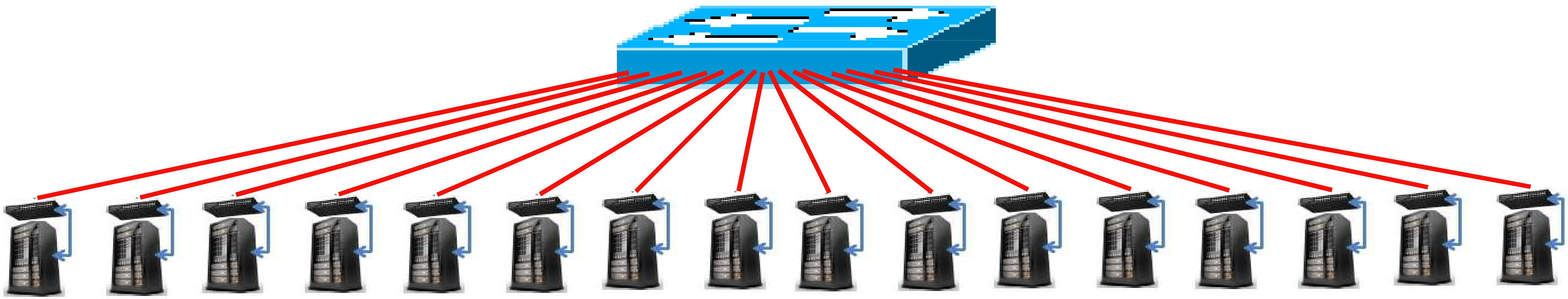


(*) N. Farrington and A. Andreyev. ***Facebook's data center network architecture***. In *Proc. IEEE Optical Interconnects*, May 2013

Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren. **Inside the Social Network's (Datacenter) Network.** *SIGCOMM Computer Communications Review,* 45, 4 (August 2015), pp. 123-137

Aggregation layer: simplest architecture

- To achieve the maximum communications throughput within the datacenter, ideally the aggregation layer could be made a single big non-blocking switch connecting all the access layer switches
  - This is an impractical non-scalable solution: the switch crossbar should guarantee a throughput too high to be achievable
  - This is the reason why DC network architectures are hierarchical
- <u>If the aggregation layer is not able to guarantee the required throughput, applications performance is affected</u>



- <u>Example</u>: a cluster of 1280 servers organized in 32 racks with 40 servers each, with an uplink from ToR switches formed by 4 x 10 Gb/s links and a single aggregation switch with 128 x 10 Gb/s ports (cost ≈ USD 700,000 in 2008)
  - If servers are equipped with 1 Gb/s NICs, total oversubscription is 1:1 $\rightarrow$ non-blocking network

Multi-layer tree topologies

- To avoid congestion probability, *oversubscription* must be kept as small as possible
  - A solution consists in connecting the various layers by means of multiple parallel links
- The picture shows a tree topology
  - Each switch connected to only one upper layer switch
  - Switches at the top of the hierarchy must have many ports and a very high aggregate bandwidth

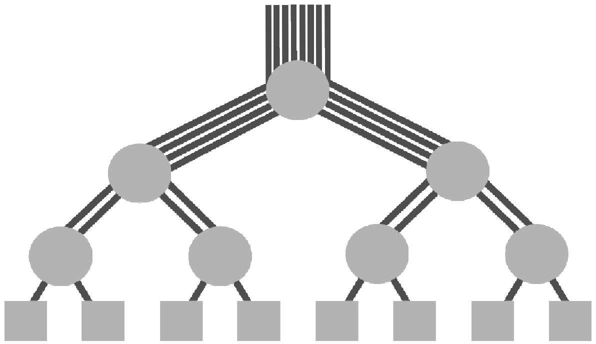
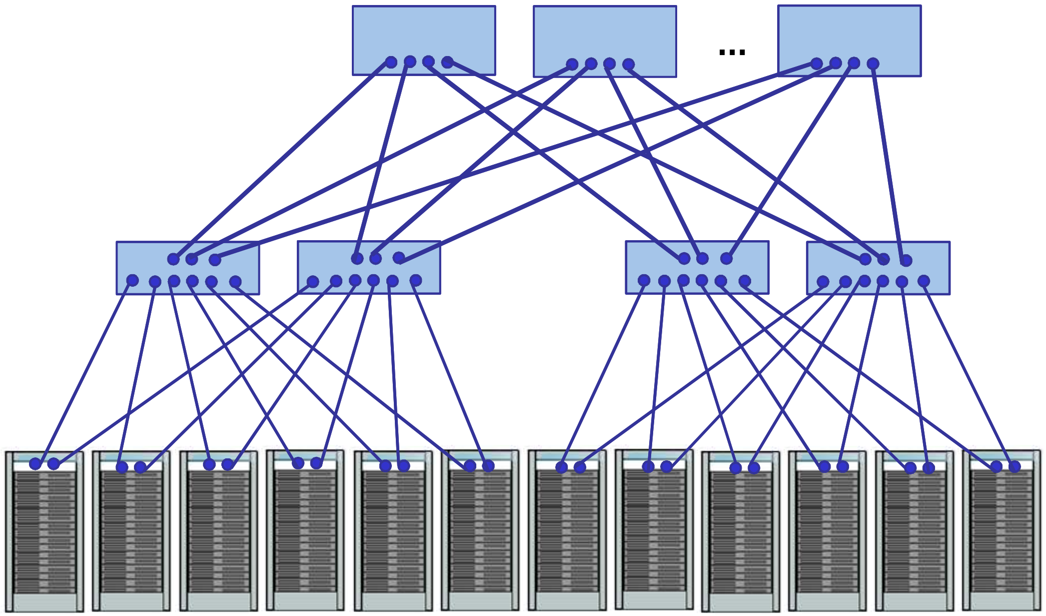


Image by Konstantin S. Solnushkin (www.clusterdesign.org)

- To effectively use parallel links bandwidth, link aggregation solutions are needed (eg. IEEE 802.3ad)
- In practice, links between layers are subject to *oversubscription* N:1 (N>1)
  - For particular traffic matrices, congestion probability is not negligible
- This leads to constraints in applications deployment
  - Servers that communicate more intensely must be located "more closely"
- <u>Requirement</u>: elasticity, i.e. no constraints in traffic distribution
- A greater number of layers also leads to greater latency
  - Negative impact on TCP throughput

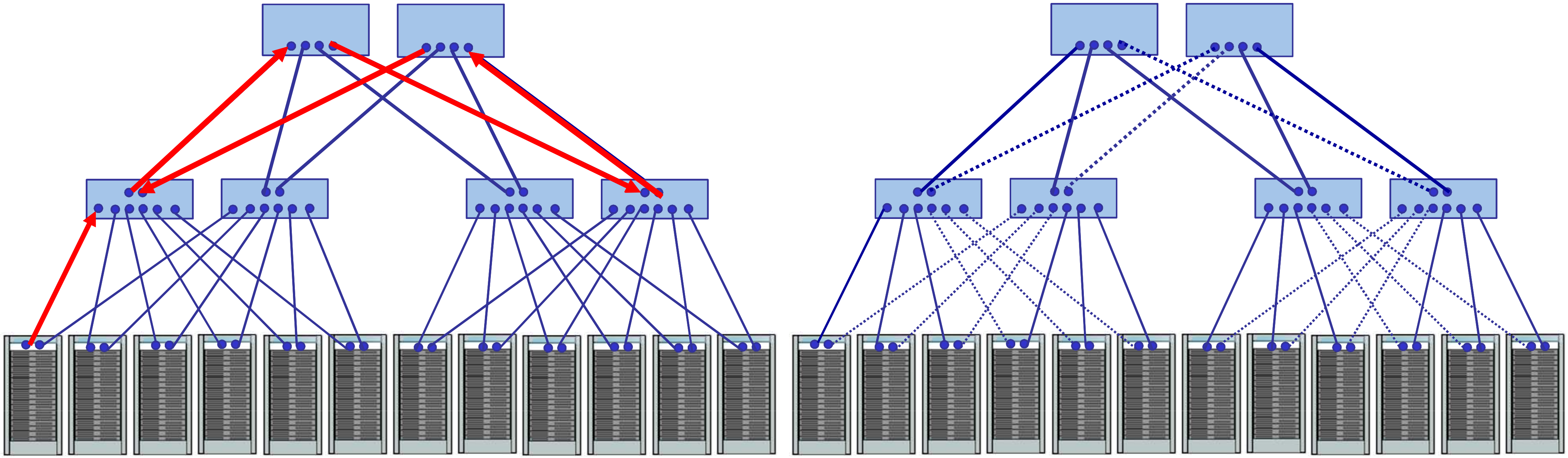
DC network arcitectures evolution

- The approach of creating an aggregation layer formed by a few “big” switches with a huge number of ports is not scalable
  - If oversubscription becomes too high, congestions may occur
- Evolution: from tree-like topologies to *multi-rooted topologies with multiple alternate paths between end-system*, where each switch is connected
  - to another upper-layer switch through multiple parallel uplinks
  - to multiple upper-layer switches
- In such a way:
  1. traffic may be split across multiple uplinks (i.e. *oversubscription* is kept as small as possible)
  2. the whole system is more robust to switch/link failures thanks to the existence of multiple paths



Network topologies and loops

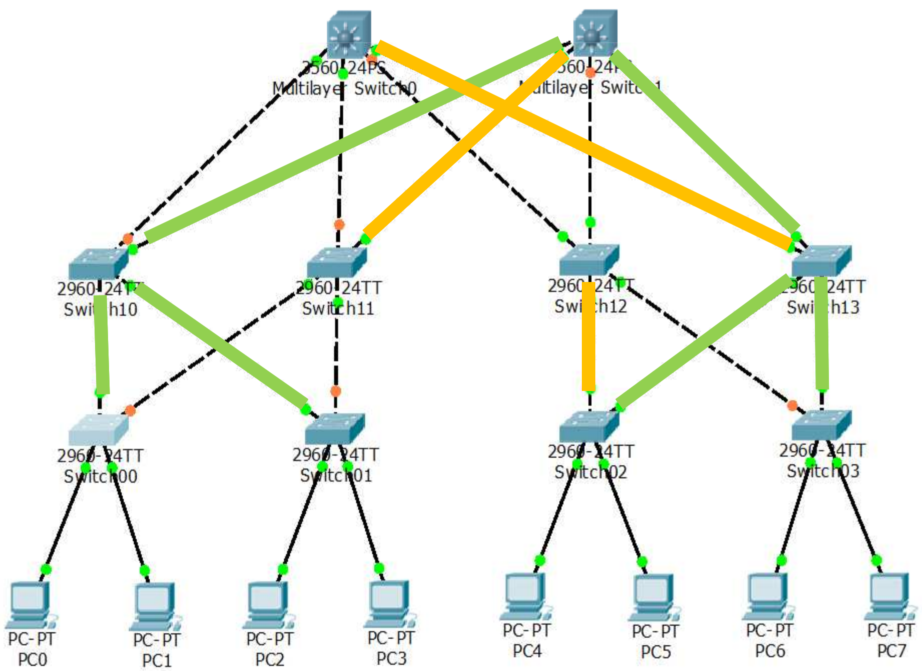
- This network topology has a problem: loops !

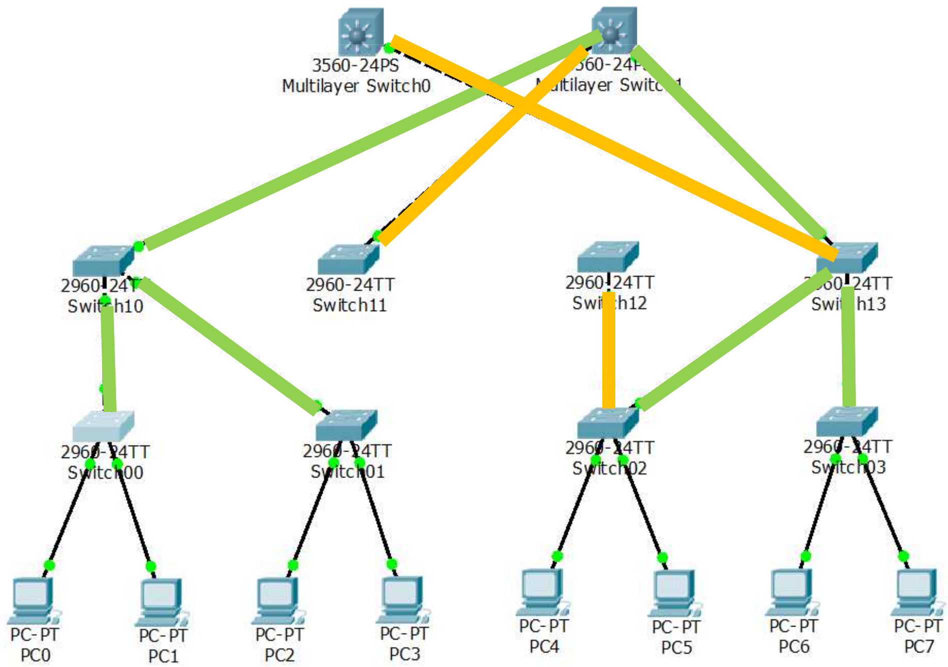


- Switches operate at layer 2 (no TTL)
- Mechanisms are needed to avoid that packets entering a loop are forwarded forever
- Traditional loop-management approach in Ethernet networks: STP
  - IEEE 802.1D standard protocol invented by Radia Perlman
  - To cut loops, network topology is transformed into a tree (*loop-free topology*) by disabling a subset of links
  - <u>Inconvenience</u>: only a fraction of network capacity may be utilized in this way and oversubscription is not reduced
- Alternative solutions: TRILL, FabricPath (Cisco), VCS (Brocade), M-LAG, QFabric, SPB, ...

Example of a DC network with STP in action

- Switches decide which interfaces should be switched off to prevent loops
- One end of a disabled link is turned off while the other is still on
- This results in a *spanning tree* connecting all the end systems as well as all the switches without any loop





STP: Spanning Tree Protocol

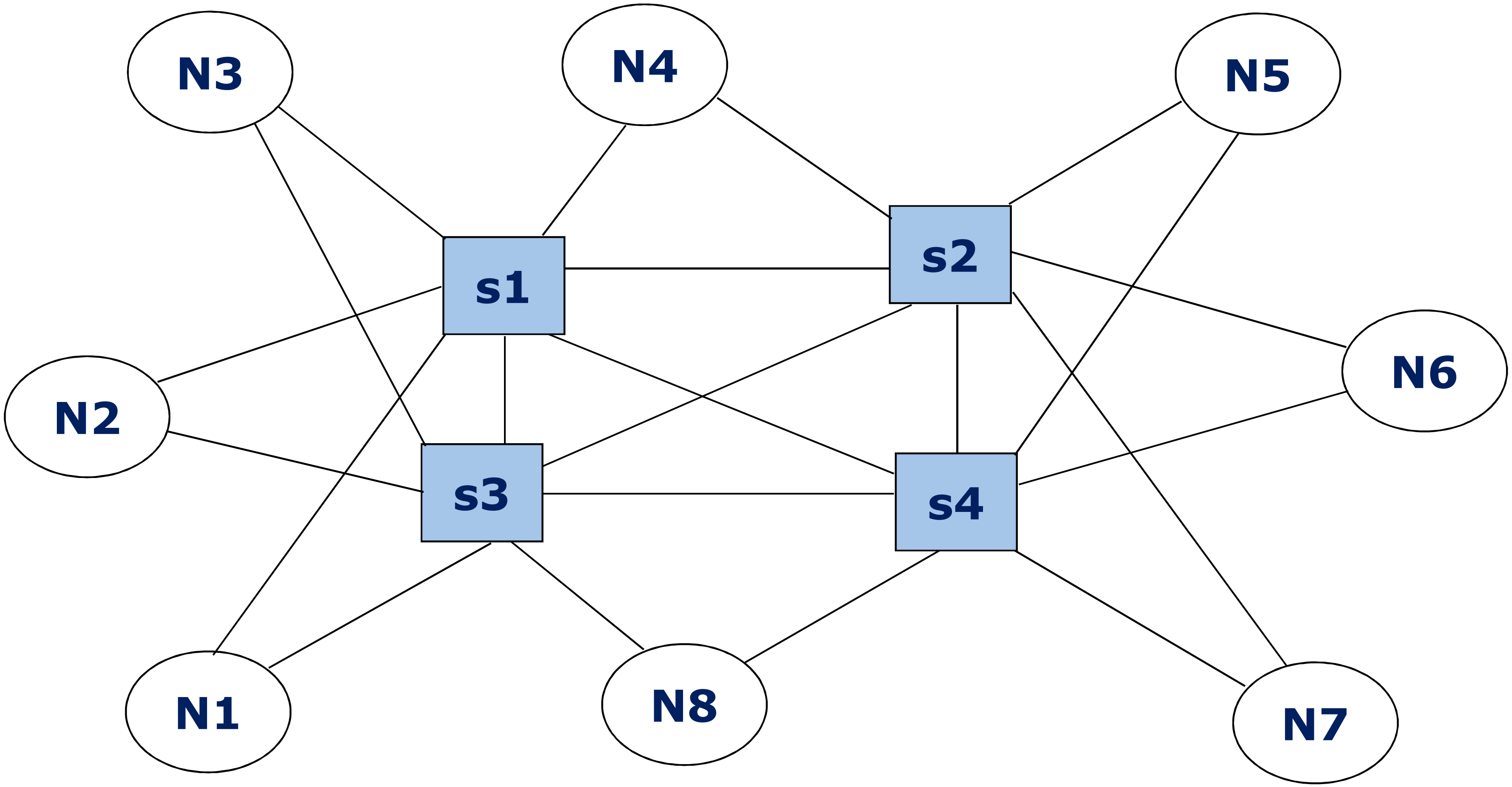
- Switches periodically exchange Configuration *Bridge Protocol Data Units* (BPDUs) to build the topology database
  - BPDU's are forwarded out all ports every 2s, to the dedicated MAC multicast address of 01:80:C2:00:00:00
  - Configuration BPDUs contain the switch *bridge ID*
- Each bridge starts out thinking it is the Root bridge
- Eventually, all switches agree that the Root bridge is the switch with smallest bridge ID
- Through BPDU exchanges, tree converges, which means all switches have same view of the spanning tree
- Each port of a switch may be in one of the following states:
  - Forwarding, Blocking, Listening, Learning

STP: how it works

STP creates a tree that provides a single unique path to each destination as follows:

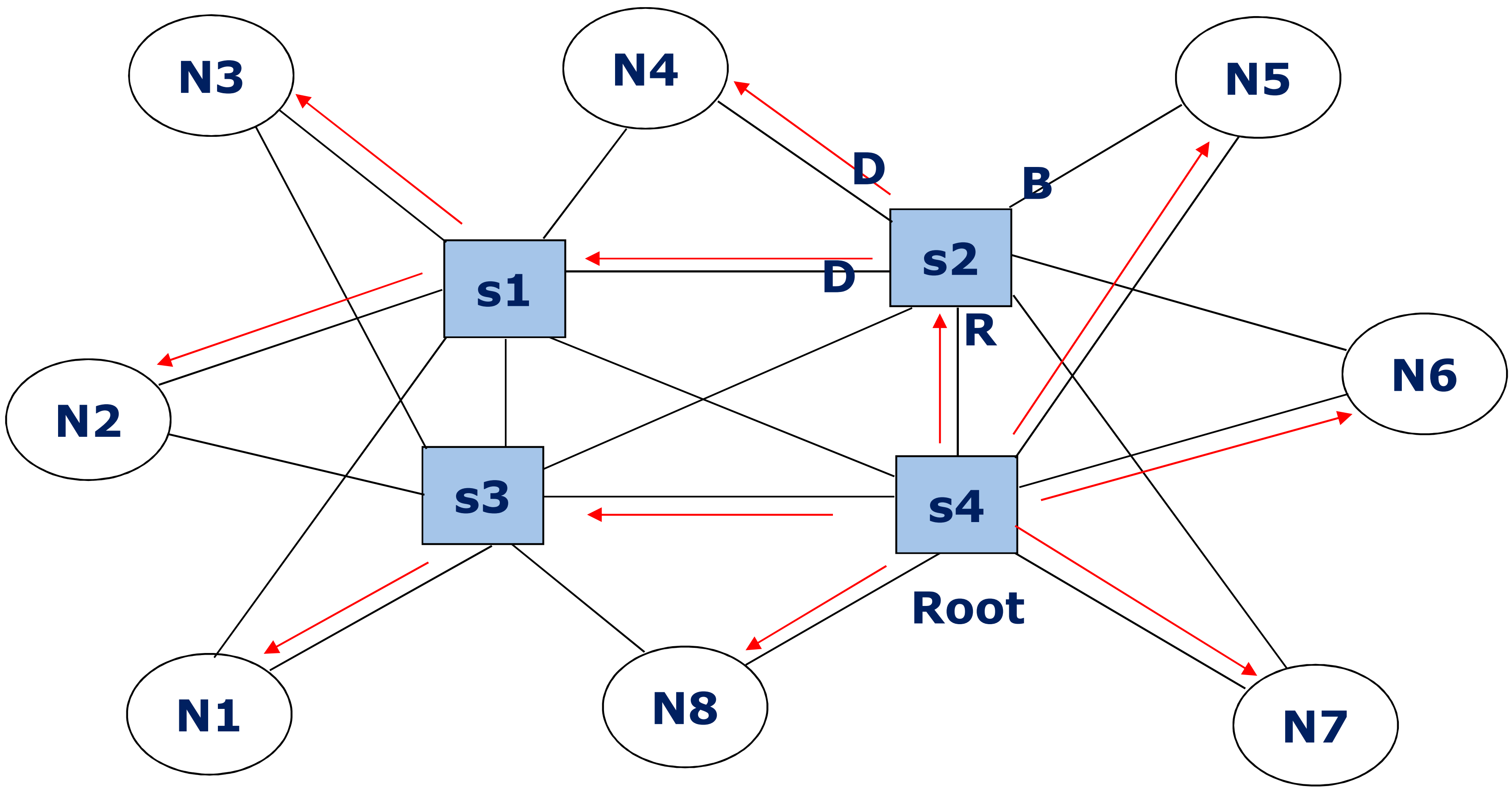
- switches elect a root bridge acting as the root of the spanning tree
- each bridge calculates the distance of the shortest path to the root bridge
- each bridge determines a *root port*, which will be used to send packets to root
- for each segment, a *designated port* is identified, i.e. the port closest to the root
- root ports and designated ports are set to *forwarding* state;
  all other ports are set to *blocking* state
  - Packets will not be received or forwarded on blocked ports

Spanning Tree Protocol in action (1)



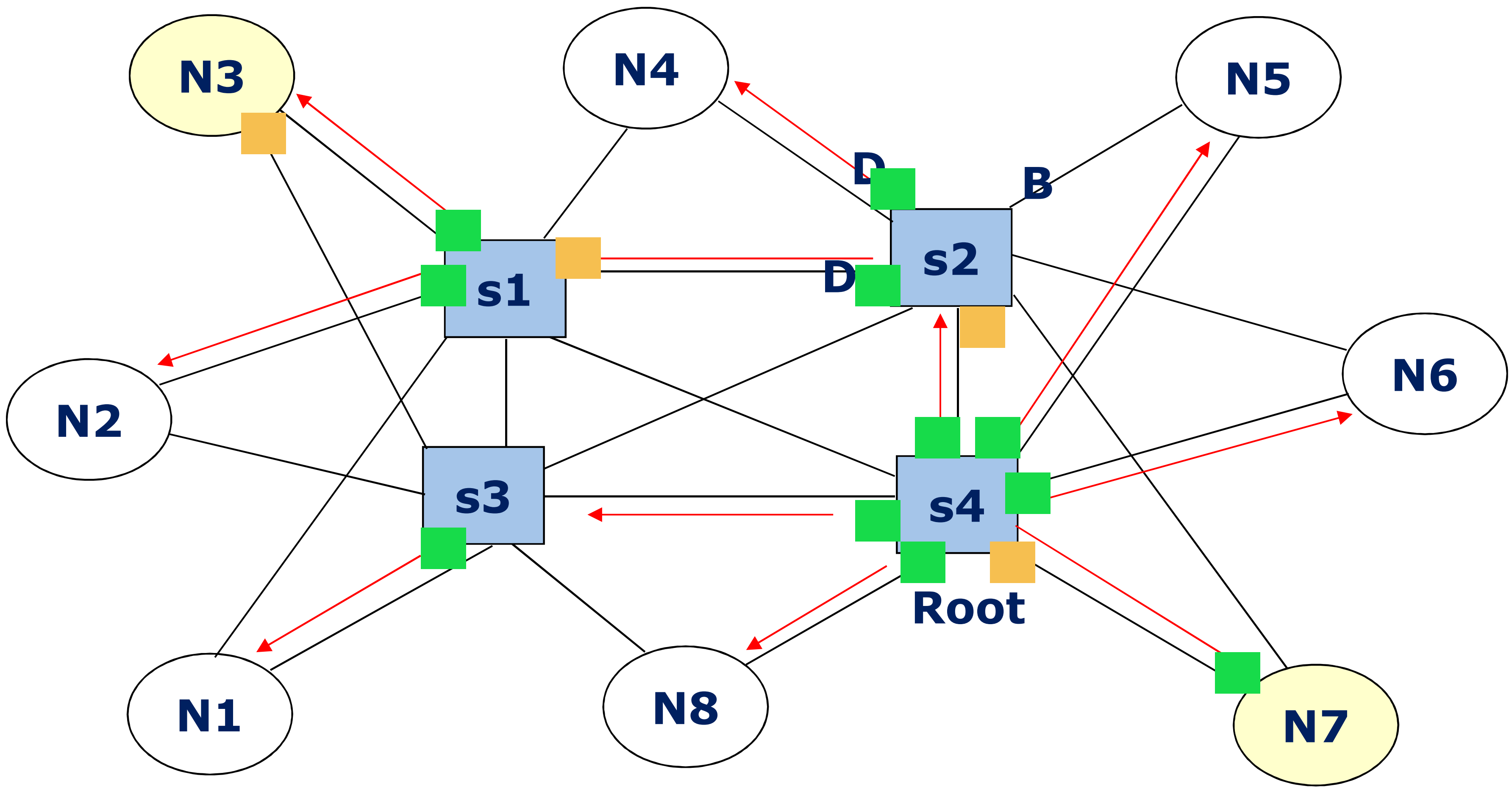
Spanning Tree Protocol in action (2)

- Having elected s4 as the Root Bridge, this is the resulting spanning tree
- Red arrows show the path of a broadcast packet from the root to any possible destinations
- Links not marked with an arrow are not traversed by any packets



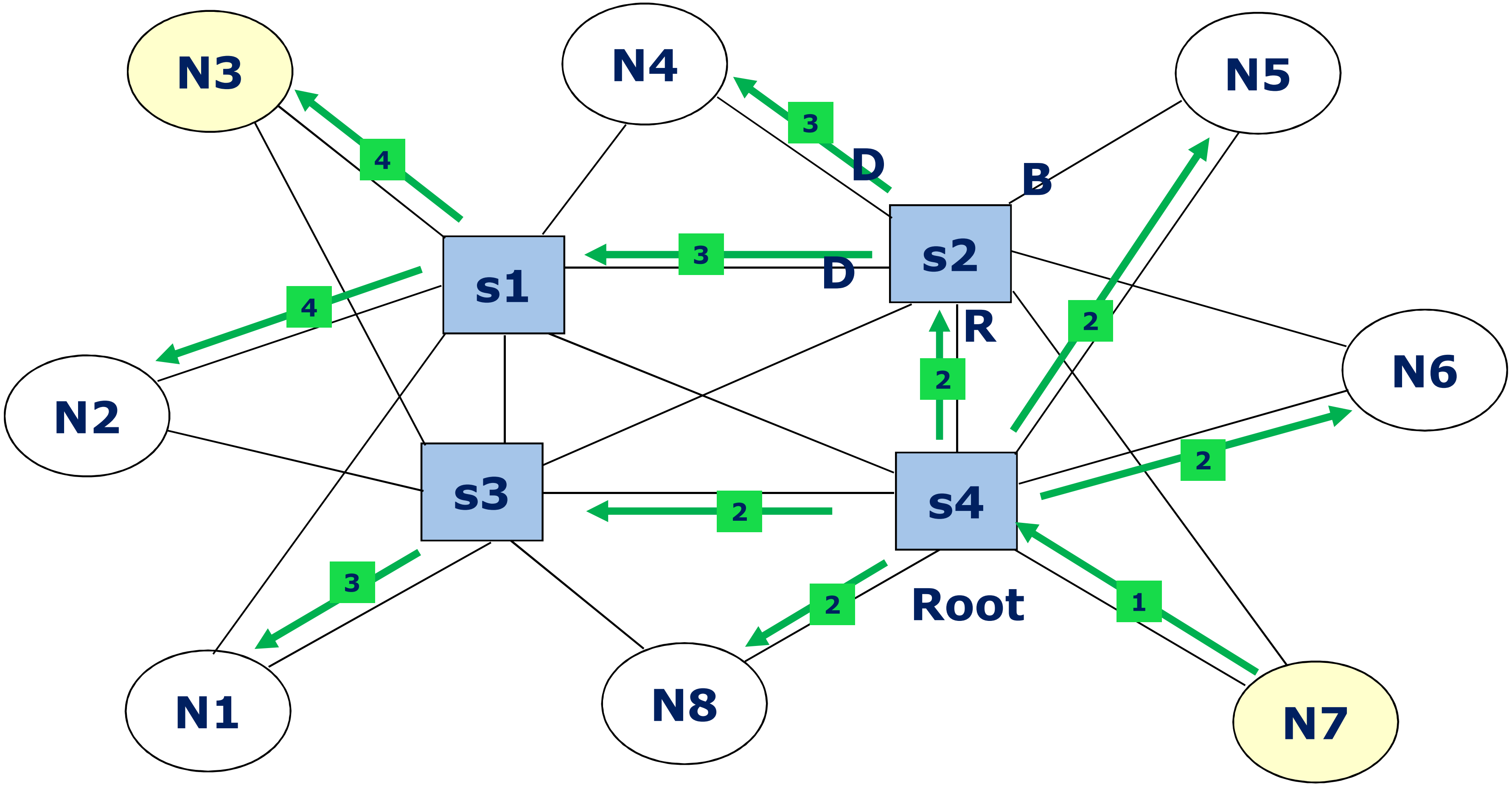
Spanning Tree Protocol in action (3)

- N7 sends in broadcast (i.e. to FF:FF:FF:FF:FF) an ARP request querying for N3's MAC
  - Packet is sent to the root s4 and from the root down the spanning tree towards all destinations
- N3 replies to N7 with its own MAC address
  - switches have learned where N7's MAC is located from the previous transmission



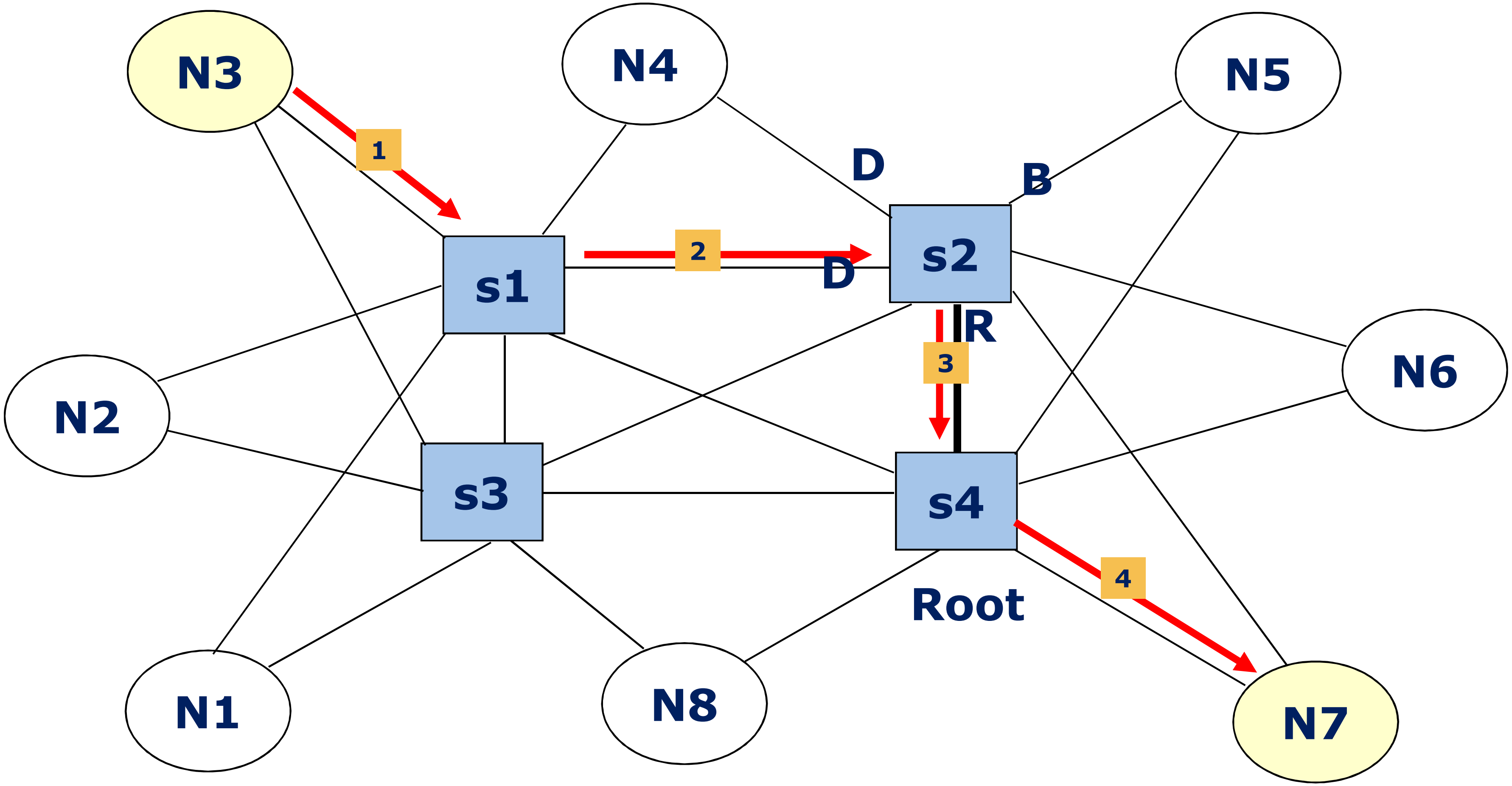
Spanning Tree Protocol in action (3)

- N7 sends in broadcast (i.e. to FF:FF:FF:FF:FF) an ARP request querying for N3's MAC
  - Packet is sent to the root s4 and from the root down the spanning tree towards all destinations
- N3 replies to N7 with its own MAC address
  - switches have learned where N7's MAC is located from the previous transmission



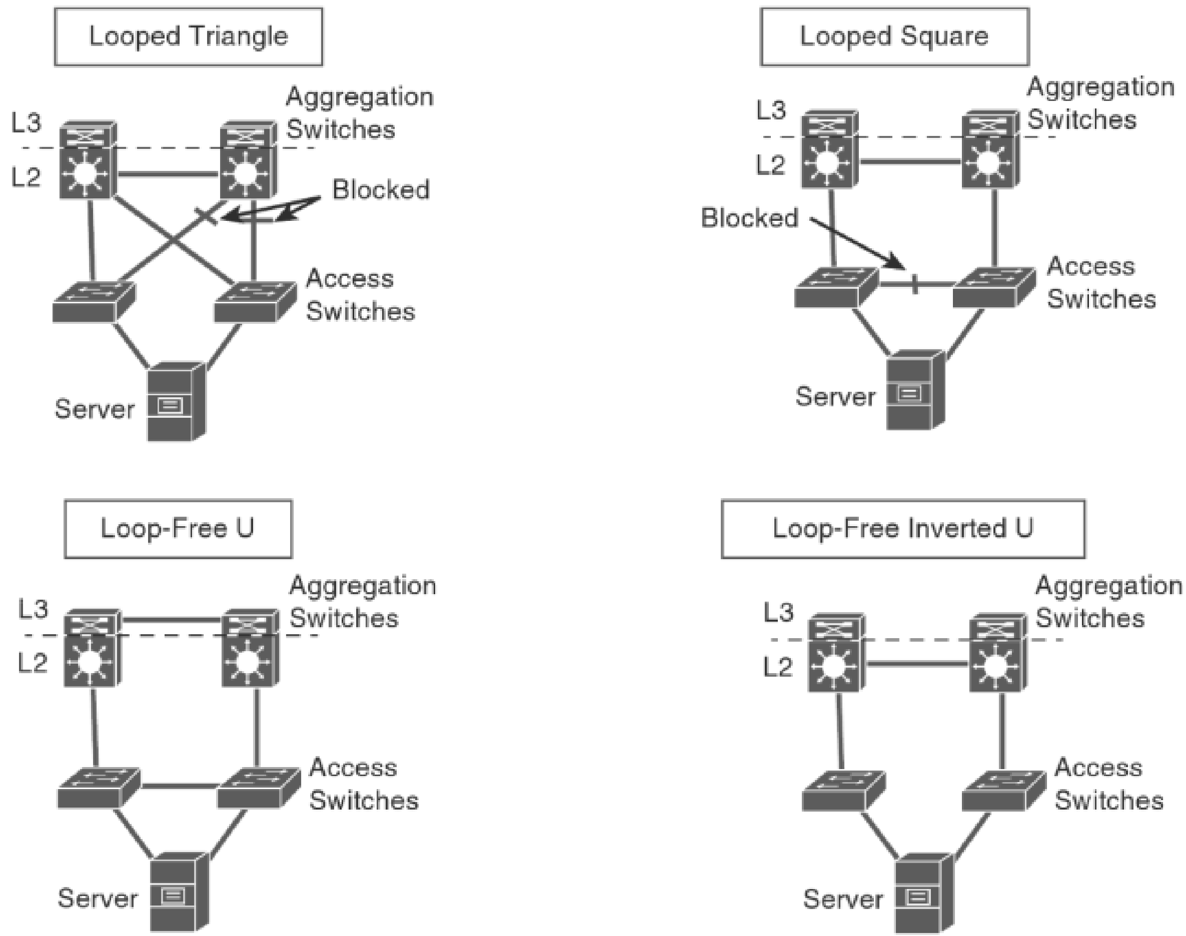
Spanning Tree Protocol in action (4)

- N7 sends in broadcast (i.e. to FF:FF:FF:FF:FF) an ARP request querying for N3's MAC
  - Packet is sent to the root s4 and from the root down the spanning tree towards all destinations
- N3 replies to N7 with its own MAC address
  - switches have learned where N7's MAC is located from the previous transmission



Access-Aggregation connection options

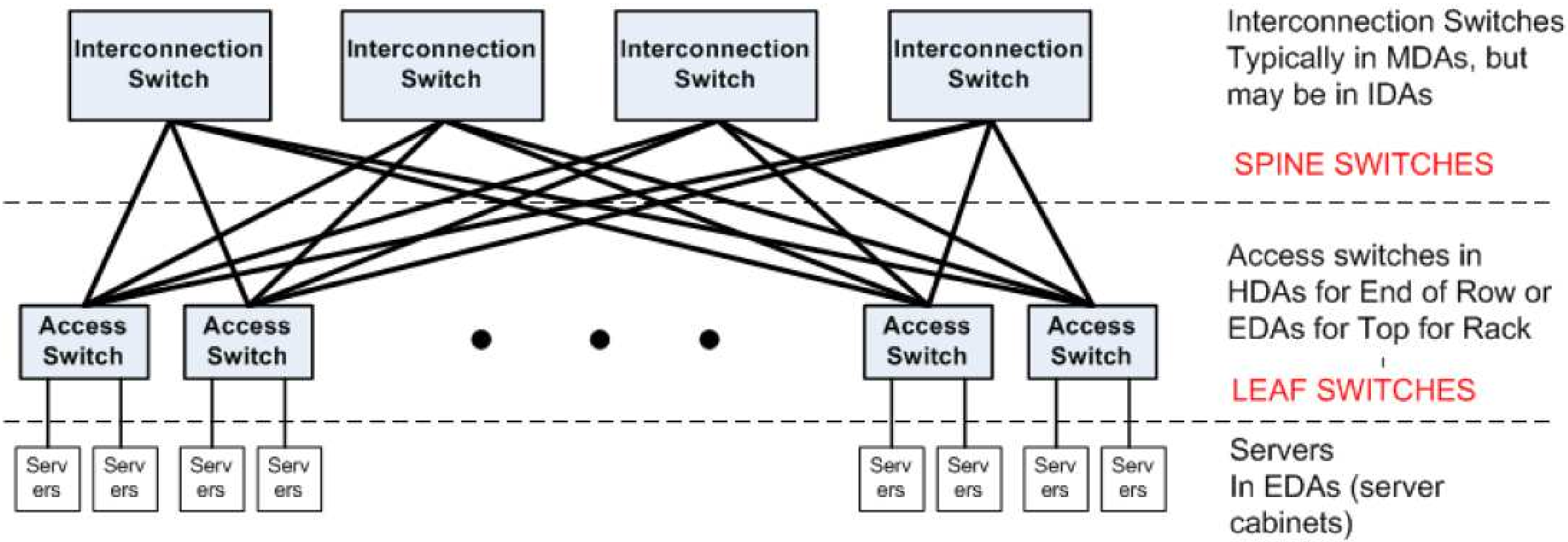
- Looped Triangle topology
  - STP blocks 2 uplinks out of 4
- Looped Square
  - STP blocks 1 horizontal link
  - In case of failure of an uplink, traffic is routed to the adjacent access switch → oversubscription doubles
- Loop-Free U
  - Communication between aggregation switches is L3
  - No loops → no links blocked by STP
- Loop-Free Inverted U



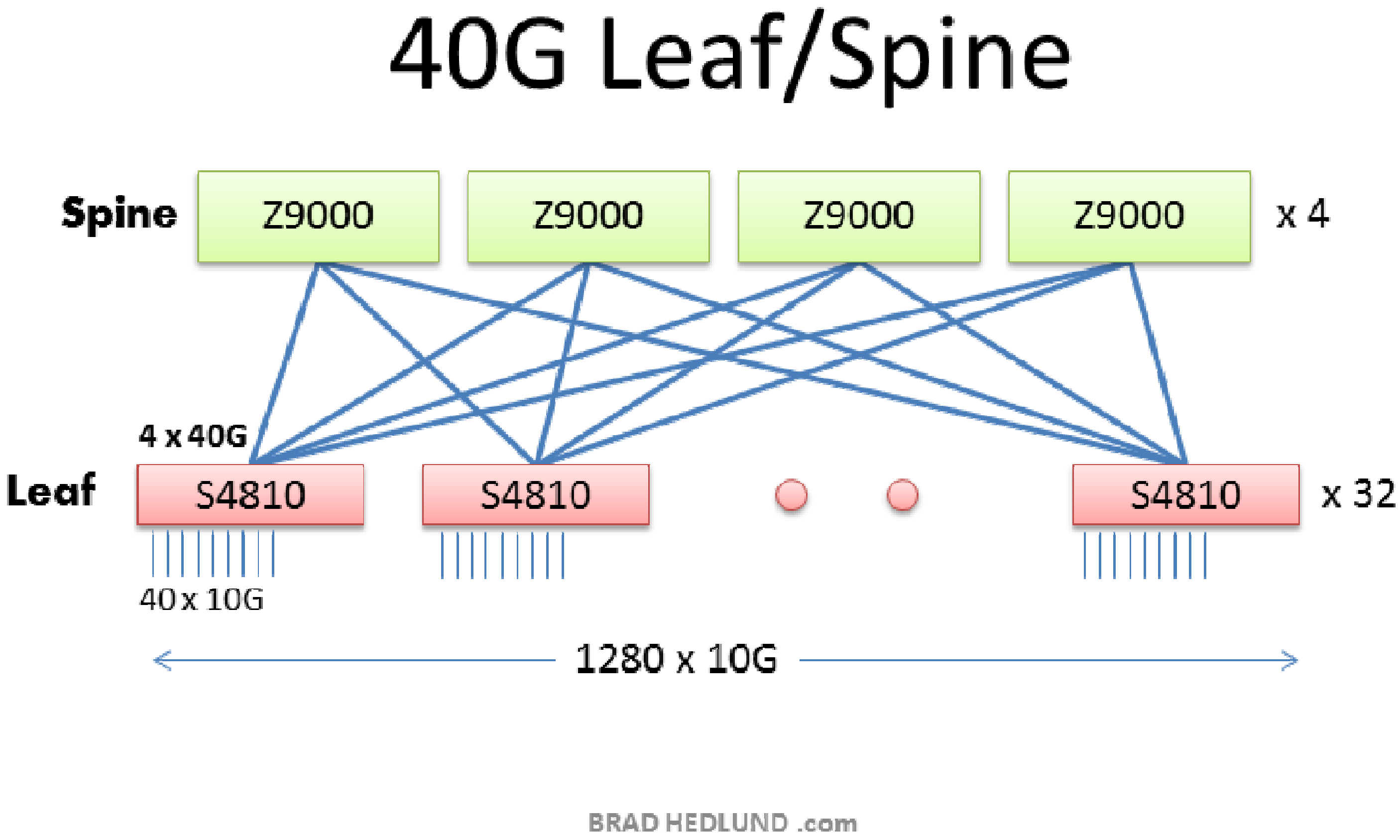
Source: Cloud Computing: automating the virtualized data center. Gustavo A. A. Santana. CISCO Press (2014)

Multi-rooted Leaf-Spine topologies

- Also known as fat-tree, they derive from Clos networks (*folded Clos*)
- Two levels hierarchy, each *leaf* switch is connected to all *spine switches*
- Advantage: elasticity
  - If the number of racks increases, the number of leaf switches increases
  - If network capacity is to be increased, the number of spine switches is to be increased

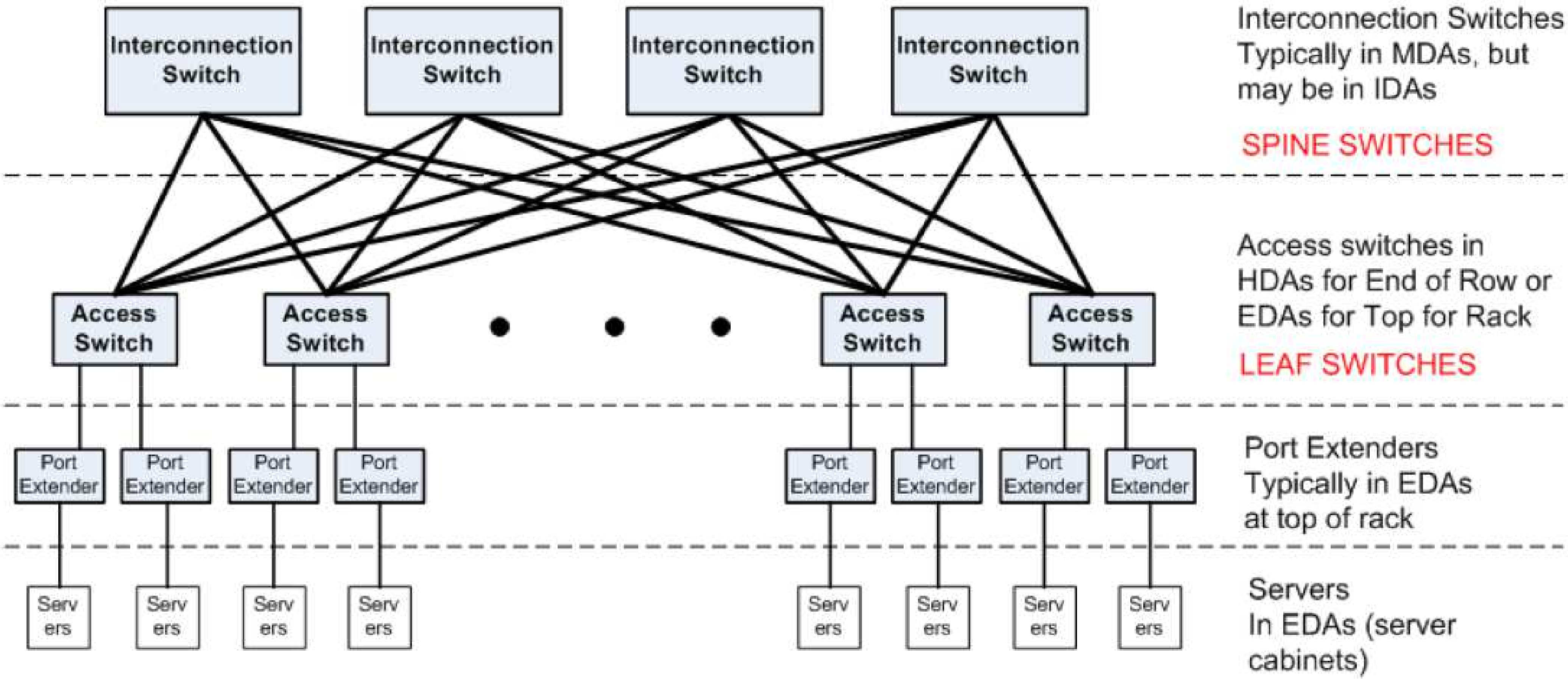


A Leaf-Spine network with 40GbE links



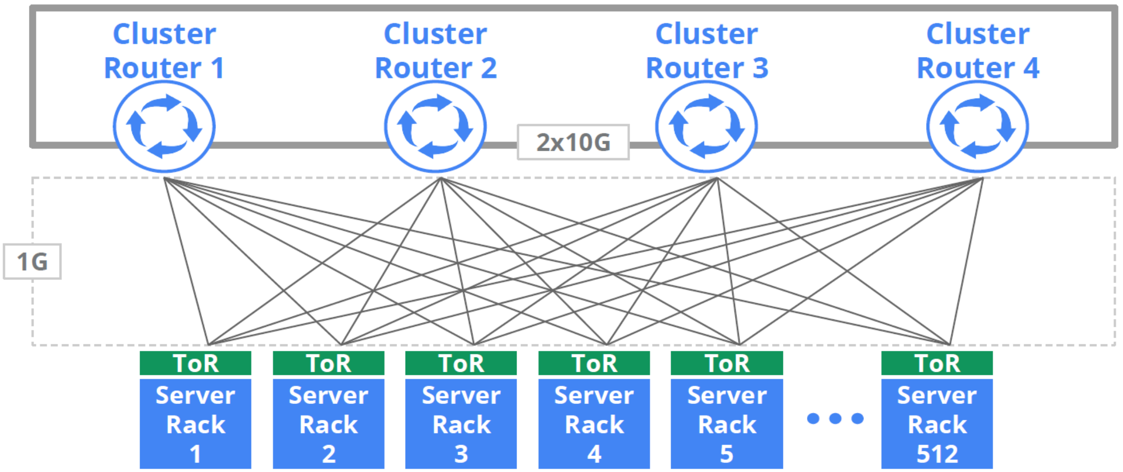
https://s3.amazonaws.com/bradhedlund2/2012/40G-10G-leaf-spine/clos-40G.png

Leaf-spine network with Port Extenders



Google DC architecture in 2004

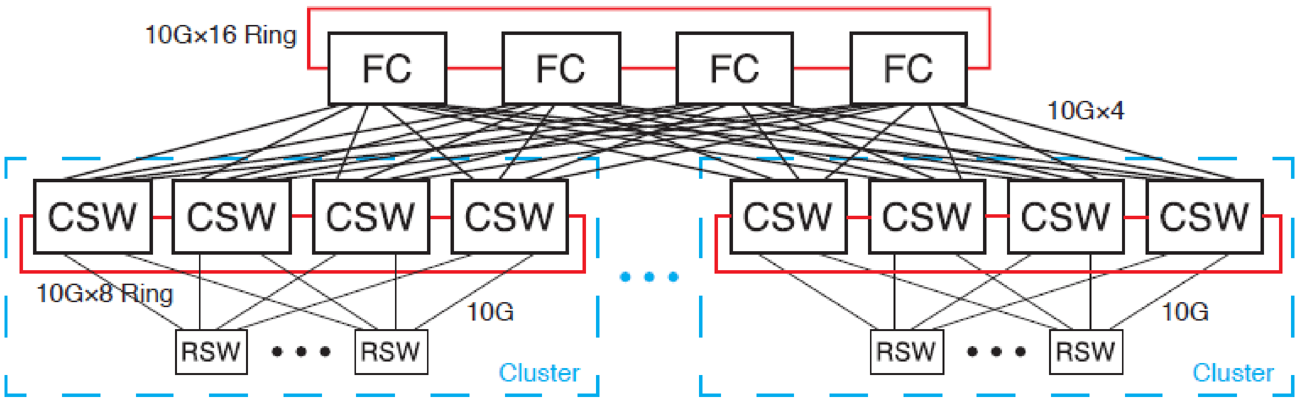
- ToR switches connected by 1 Gb/s links to an upper aggregation layer made of 4 routers, in turn connected to form a ring by means of couples of 10 Gb/s links
- Each rack included 40 servers, equipped with 1 Gb/s NICs
- A whole cluster included $512 \cdot 40 \approx 20000$ servers
- Aggregate bandwidth of a cluster: $4 \cdot 512 \cdot 1$ Gb/s = 2 Tb/s
- Each rack could produce up to 40 Gb/s of aggregate traffic but racks were connected to the upper layer router with a link capacity of $4 \cdot 1$ Gb/s = 4 Gb/s
  - Congestions were possible if all the servers of a rack needed to communicate with the rest of DC
  - Traffic needed to be kept as local as possible within a rack



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.
Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network.
SIGCOMM Computer Communications Review, 45, 4 (August 2015), pp. 183-197

Facebook DC architecture in 2013

- Servers connected by 10 Gb/s links to a ToR switch(RSW) in each rack
- RSW switches connected by 4 x 10 Gb/s uplinks to an *aggregation layer* formed by 4 *cluster switches* (CSW) connected to form a ring
  - Oversubscription: 40 servers · 10 Gb/s : 4 uplinks · 10 Gb/s = 10 : 1
- A single ring of 4 CSWs identifies a cluster (e.g. including 16 racks)
  - The 4 CSW switches are connected in a ring topology by means of 8 x 10 Gb/s links
- CSW switches connected by 4 x 10 Gb/s uplinks to a *core* layer formed by 4 *Fat Cat* (FC) switches connected to form a ring by means of 16 x 10 Gb/s links
  - Oversubscription: 16 rack · 10 Gb/s : 4 uplink · 10 Gb/s = 4 : 1



N. Farrington and A. Andreyev. ***Facebook's data center network architecture***. In *Proc. IEEE Optical Interconnects*, May 2013