Data Center Networking

Prof. Andrzej Duda

Cloud Computing – Data Centers

What’s a Cloud Service Data Center?

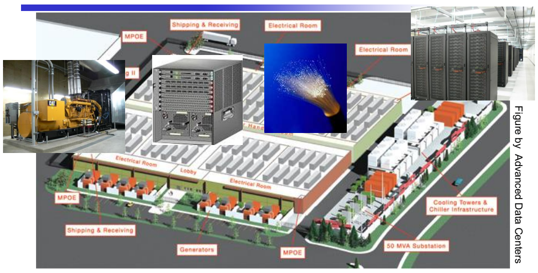


Figure by Advanced Data Centers

- Electrical power and economies of scale determine total data center size: 50,000 – 200,000 servers today
- Servers divided up among hundreds of different services
- Scale-out is paramount: some services have 10s of servers, some have 10s of 1000s

Data Center Costs

Amortized Cost*	Component	Sub-Components
~45%	Servers	CPU, memory, disk
~25%	Power infrastructure	UPS, cooling, power distribution
~15%	Power draw	Electrical utility costs
~15%	Network	Switches, links, transit

*3 yr amortization for servers, 15 yr for infrastructure; 5% cost of money

- Total cost varies
  - upwards of $1/4 B for mega data center
  - server costs dominate
  - network costs significant
- Long provisioning timescales:
  - new servers purchased quarterly at best

Overall Data Center Design Goal

Agility – Any service, Any Server

- Turn the servers into a single large fungible pool
  - Let services “breathe” : dynamically expand and contract their footprint as needed
    - We already see how this is done in terms of Google’s GFS, BigTable, MapReduce
- Benefits
  - Increase service developer productivity
  - Lower cost
  - Achieve high performance and reliability

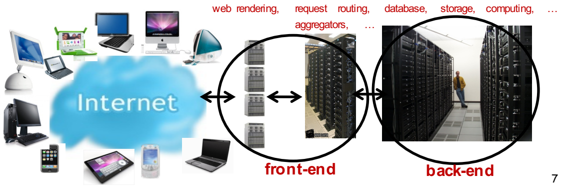
These are the three motivators for most data center infrastructure projects!

Cloud Computing

- Elastic resources
  - Expand and contract resources
  - Pay-per-use
  - Infrastructure on demand
- Multi-tenancy
  - Multiple independent users
  - Security and resource isolation
  - Amortize the cost of the (shared) infrastructure
- Flexibility service management
  - Resiliency: isolate failure of servers and storage
  - Workload movement: move work to other locations

Internet and Web …

- From "traditional" web to "web service" (or SOA)
  - no longer simply "file" (or web page) downloads
    - pages often dynamically generated, more complicated "objects" (e.g., Flash videos used in YouTube)
  - HTTP is used simply as a "transfer" protocol
    - many other "application protocols" layered on top of HTTP
  - web services & SOA (service-oriented architecture)
- A schematic representation of "modern" web services


web rendering, request routing, aggregators, …
database, storage, computing, …
Internet
front-end
back-end

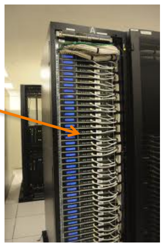
Data Center Network

Networking Objectives

1. Uniform high capacity
   - Capacity between servers limited only by their NICs
   - No need to consider topology when adding servers
     => In other words, high capacity between two any servers no matter which racks they are located !
2. Performance isolation
   - Traffic of one service should be unaffected by others
3. Ease of management: "Plug-&-Play" (layer-2 semantics)
   - Flat addressing, so any server can have any IP address
   - Server configuration is the same as in a LAN
   - Legacy applications depending on broadcast must work

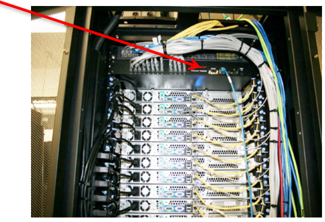
What goes into a datacenter (network)?

- Servers organized in racks



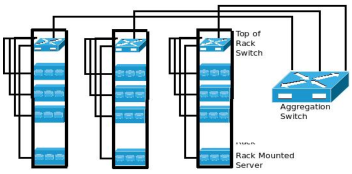
What goes into a datacenter (network)?

- Servers organized in racks
- Each rack has a `Top of Rack' (ToR) switch



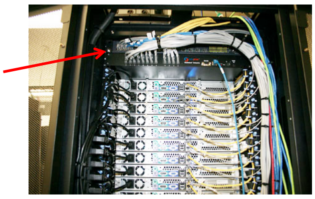
What goes into a datacenter (network)?

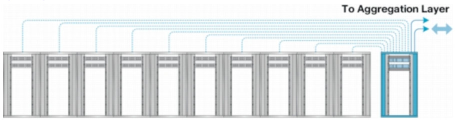
- Servers organized in racks
- Each rack has a `Top of Rack' (ToR) switch
- An `aggregation fabric' interconnects ToR switches



Top-of-Rack Architecture

- Rack of servers
  - –Commodity servers
  - –And top-of-rack switch
- Modular design
  - –Preconfigured racks
  - –Power, network, and storage cabling
- Aggregate to the next level





13

Top-of-Rack Architecture

- A rack has ~20-40 servers

Front of a rack

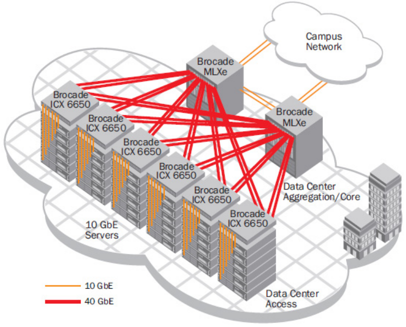
Rear of a rack

- Example of a TOR switch with 48 ports

"Top of Rack" switch

14

Example 1



Brocade reference design

15

SCALE!

16

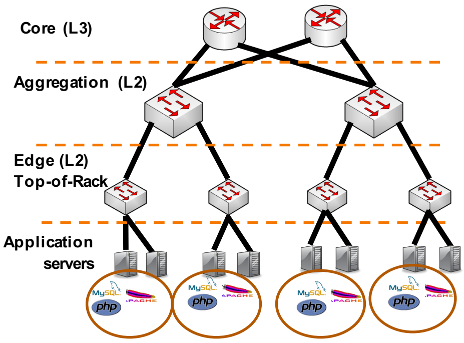
How big exactly?

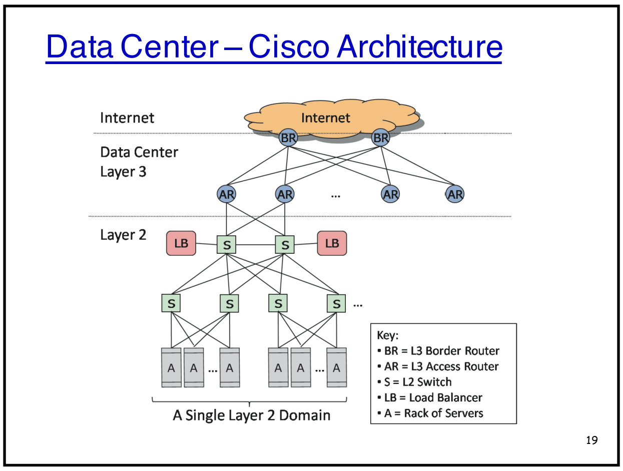
- 1M servers [Microsoft]
  - –less than Google, more than Amazon
- > $1B to build one site [Facebook]
- >$20M/month/site operational costs [Microsoft '09]

But only O(10-100) sites

17

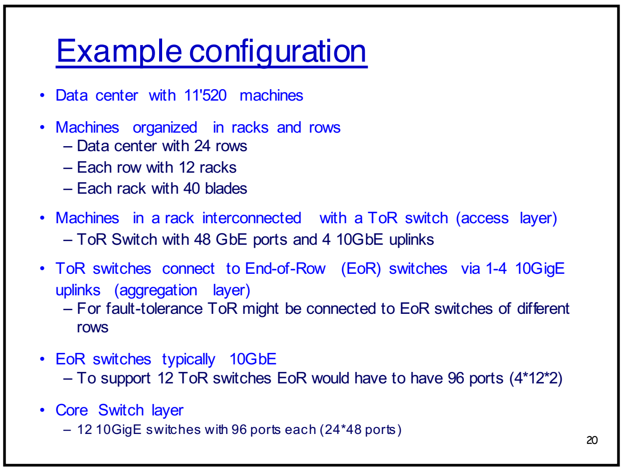
Canonical Data Center Architecture





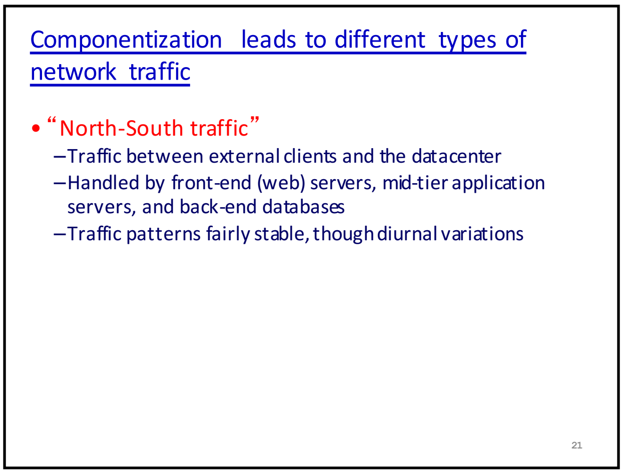
Data Center – Cisco Architecture

Key:
- BR = L3 Border Router
- AR = L3 Access Router
- S = L2 Switch
- LB = Load Balancer
- A = Rack of Servers



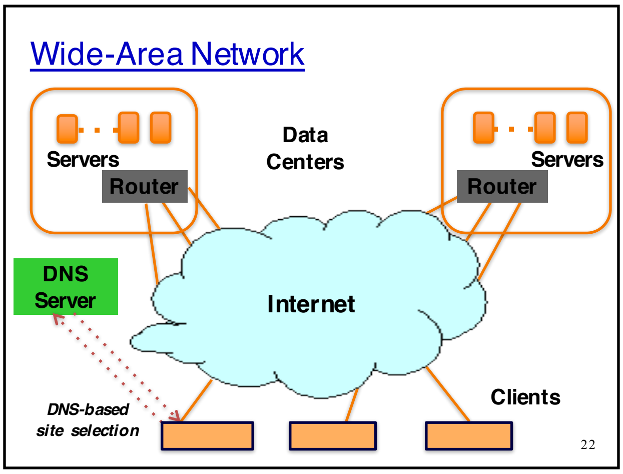
Example configuration

- Data center with 11'520 machines
- Machines organized in racks and rows
  - Data center with 24 rows
  - Each row with 12 racks
  - Each rack with 40 blades
- Machines in a rack interconnected with a ToR switch (access layer)
  - ToR Switch with 48 GbE ports and 4 10GbE uplinks
- ToR switches connect to End-of-Row (EoR) switches via 1-4 10GigE uplinks (aggregation layer)
  - For fault-tolerance ToR might be connected to EoR switches of different rows
- EoR switches typically 10GbE
  - To support 12 ToR switches EoR would have to have 96 ports (4*12*2)
- Core Switch layer
  - 12 10GigE switches with 96 ports each (24*48 ports)

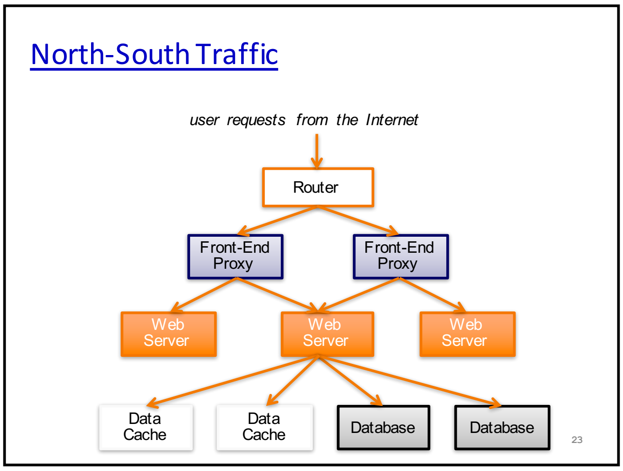


Componentization leads to different types of network traffic

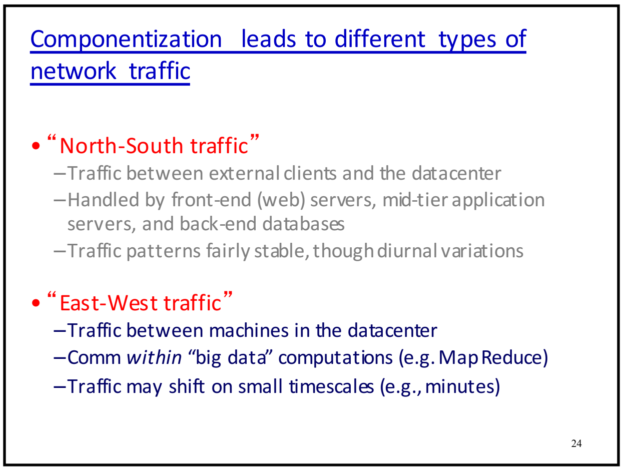
- "North-South traffic"
  - Traffic between external clients and the datacenter
  - Handled by front-end (web) servers, mid-tier application servers, and back-end databases
  - Traffic patterns fairly stable, though diurnal variations



Wide-Area Network

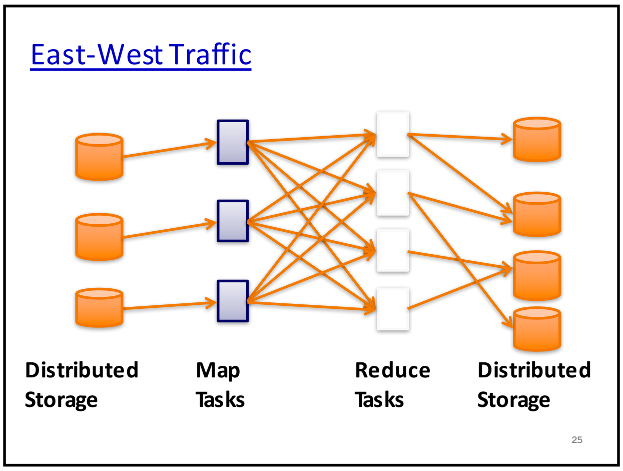


North-South Traffic



Componentization leads to different types of network traffic

- "North-South traffic"
  - Traffic between external clients and the datacenter
  - Handled by front-end (web) servers, mid-tier application servers, and back-end databases
  - Traffic patterns fairly stable, though diurnal variations
- "East-West traffic"
  - Traffic between machines in the datacenter
  - Comm *within* "big data" computations (e.g. Map Reduce)
  - Traffic may shift on small timescales (e.g., minutes)



East-West Traffic

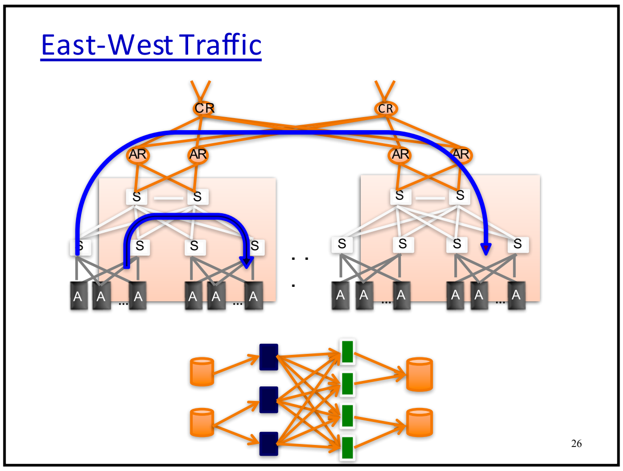
Distributed Storage

Map Tasks

Reduce Tasks

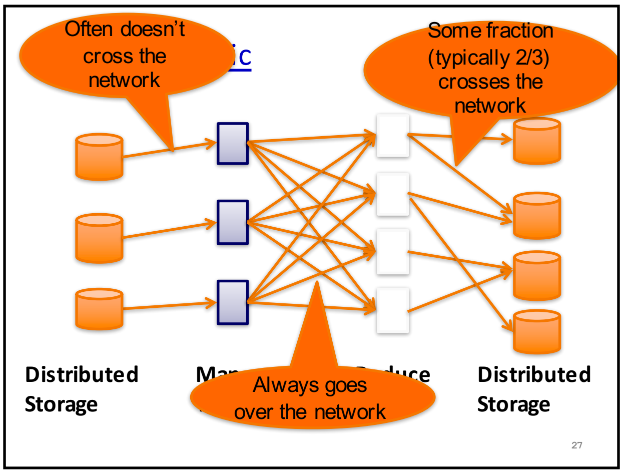
Distributed Storage

25



East-West Traffic

26



Often doesn't cross the network

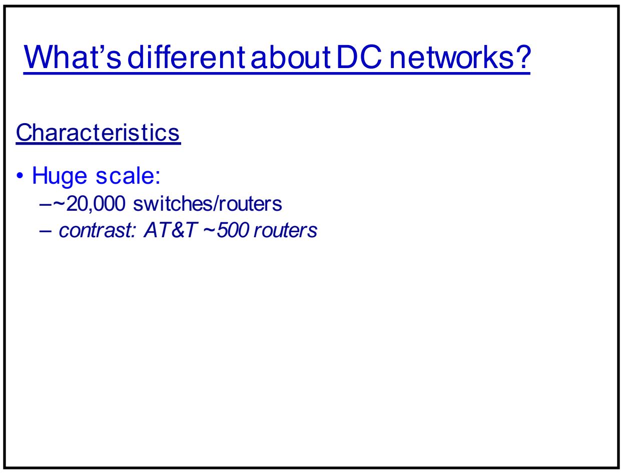
Some fraction (typically 2/3) crosses the network

Always goes over the network

Distributed Storage

Distributed Storage

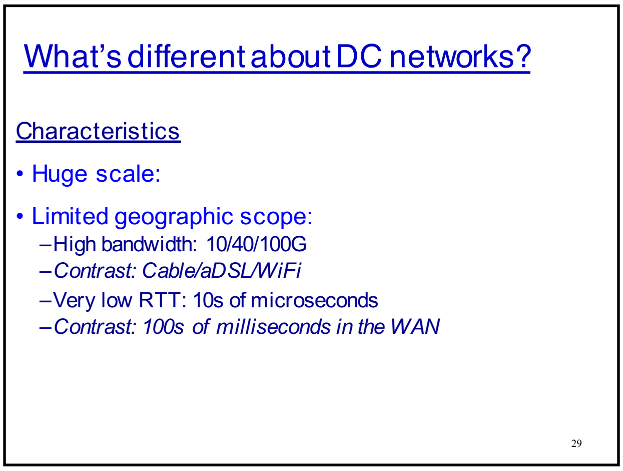
27



What's different about DC networks?

Characteristics

- Huge scale:
  - –~20,000 switches/routers
  - – *contrast: AT&T ~500 routers*

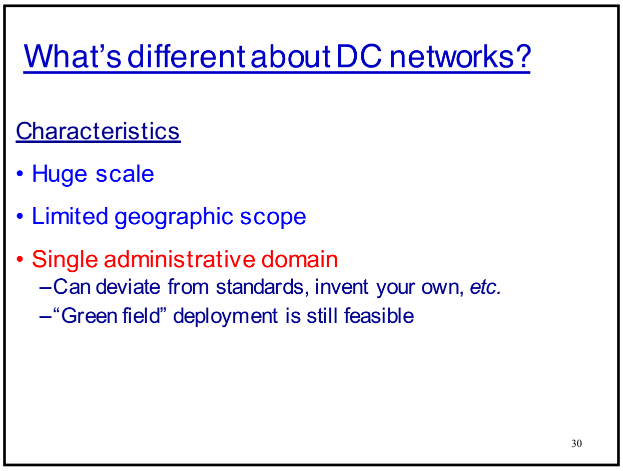


What's different about DC networks?

Characteristics

- Huge scale:
- Limited geographic scope:
  - –High bandwidth: 10/40/100G
  - –*Contrast: Cable/aDSL/WiFi*
  - –Very low RTT: 10s of microseconds
  - –*Contrast: 100s of milliseconds in the WAN*

29



What's different about DC networks?

Characteristics

- Huge scale
- Limited geographic scope
- Single administrative domain
  - –Can deviate from standards, invent your own, *etc.*
  - –"Green field" deployment is still feasible

30

What's different about DC networks?

Characteristics

- Huge scale
- Limited geographic scope
- Single administrative domain
- Control over one/both endpoints
  - –can change (say) addressing, congestion control, *etc.*
  - –can add mechanisms for security/policy/etc. at the endpoints (typically in the hypervisor)

What's different about DC networks?

Characteristics

- Huge scale
- Limited geographic scope
- Single administrative domain
- Control over one/both endpoints
- Control over the *placement* of traffic source/sink
  - –e.g., map-reduce scheduler chooses where tasks run
  - –alters traffic pattern (what traffic crosses which links)

What's different about DC networks?

Characteristics

- Huge scale
- Limited geographic scope
- Single administrative domain
- Control over one/both endpoints
- Control over the *placement* of traffic source/sink
- Regular/planned topologies (e.g., trees/fat-trees)
  - –Contrast: ad-hoc WAN topologies (dictated by real-world geography and facilities)

What's different about DC networks?

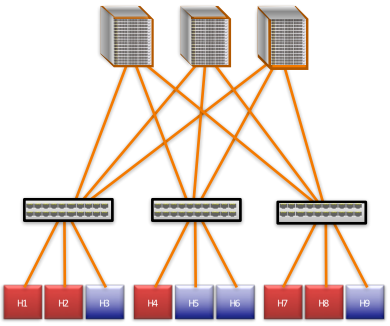
Characteristics

- Huge scale
- Limited geographic scope
- Single administrative domain
- Control over one/both endpoints
- Control over the *placement* of traffic source/sink
- Regular/planned topologies (e.g., trees/fat-trees)
- Limited heterogeneity
  - –link speeds, technologies, latencies, …

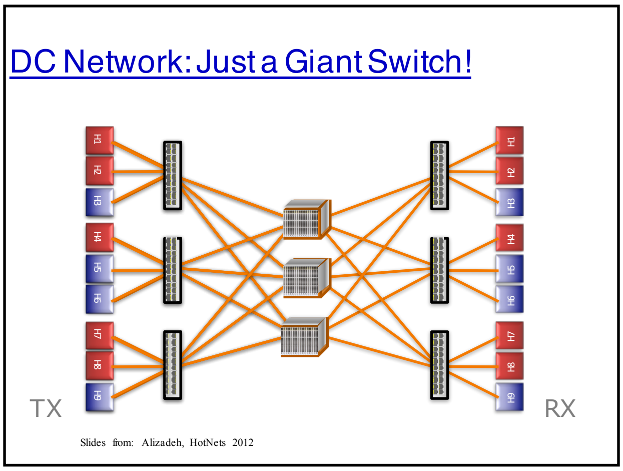
High Bandwidth

- Ideal: Each server can talk to any other server at its full access link rate
- Conceptually: DC network as one giant switch

DC Network: Just a Giant Switch!

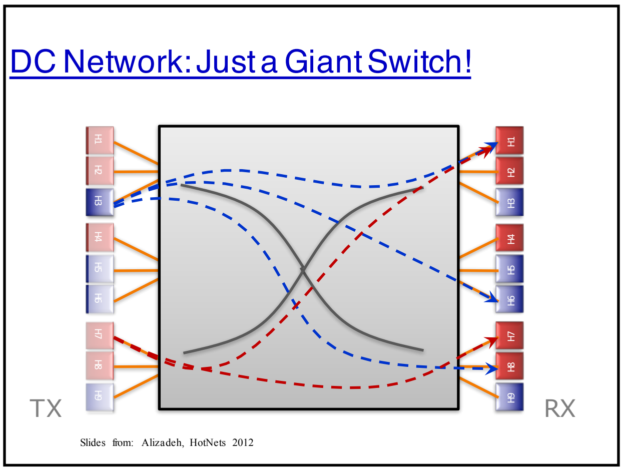


Slides from: Alizadeh, HotNets 2012



DC Network: Just a Giant Switch!

Slides from: Alizadeh, HotNets 2012

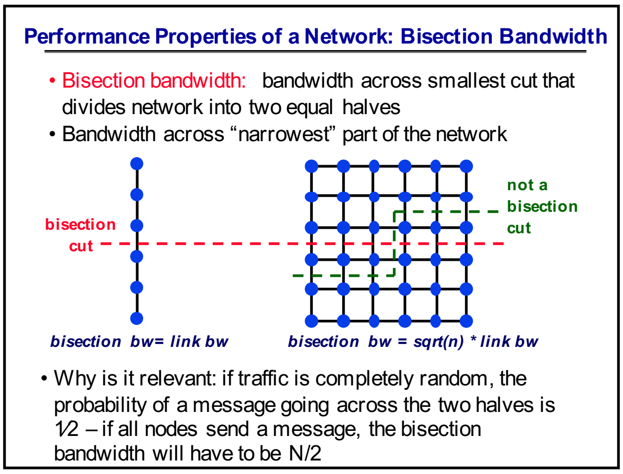


DC Network: Just a Giant Switch!

Slides from: Alizadeh, HotNets 2012

High Bandwidth

- Ideal: Each server can talk to any other server at its full access link rate
- Conceptually: DC network as one giant switch
  - Would require a 10 Pbits/sec switch!
    - 1M ports (one port/server)
    - 10Gbps per port
- Practical approach: build a network of switches ("fabric") with high "bisection bandwidth"
  - Each switch has practical #ports and link speeds



Performance Properties of a Network: Bisection Bandwidth

- Bisection bandwidth: bandwidth across smallest cut that divides network into two equal halves
- Bandwidth across "narrowest" part of the network

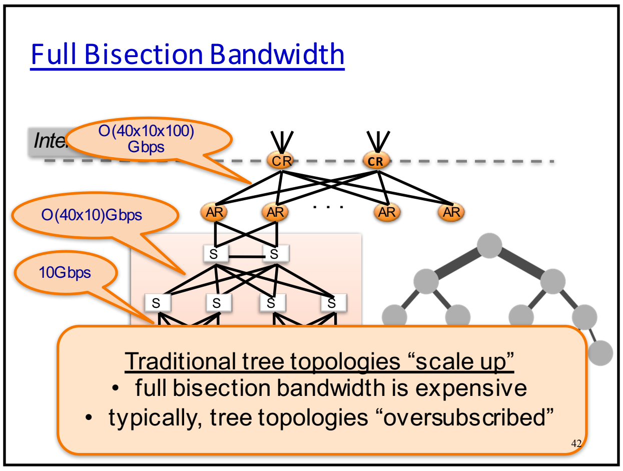
- Why is it relevant: if traffic is completely random, the probability of a message going across the two halves is 1/2 – if all nodes send a message, the bisection bandwidth will have to be N/2

What's different about DC networks?

Goals

- Extreme bisection bandwidth requirements
  - recall: all that east-west traffic
  - target: any server can communicate at its full link speed
  - problem: server's access link is 10Gbps!

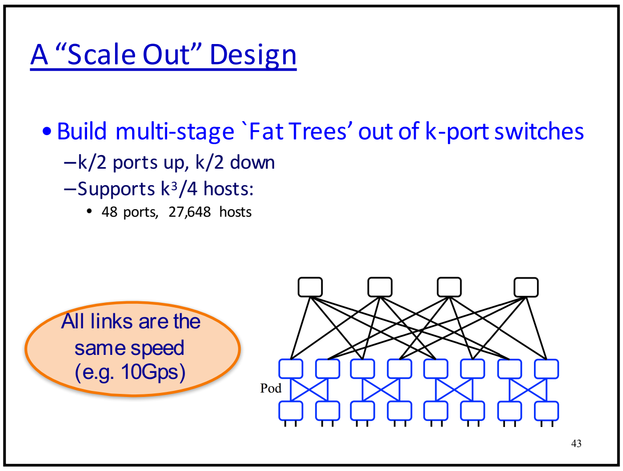
41



Full Bisection Bandwidth

Traditional tree topologies "scale up"
- full bisection bandwidth is expensive
- typically, tree topologies "oversubscribed"

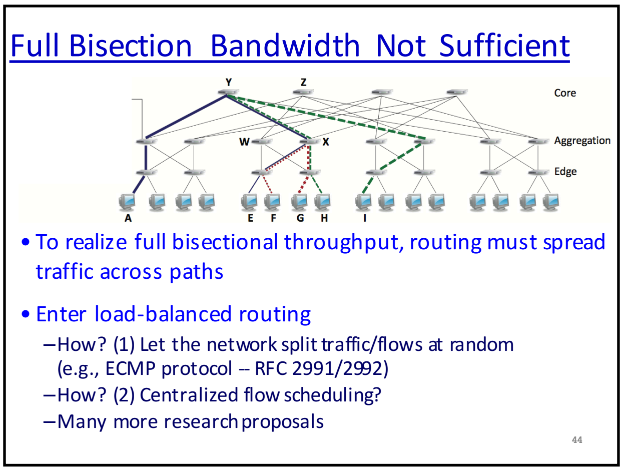
42



A “Scale Out” Design

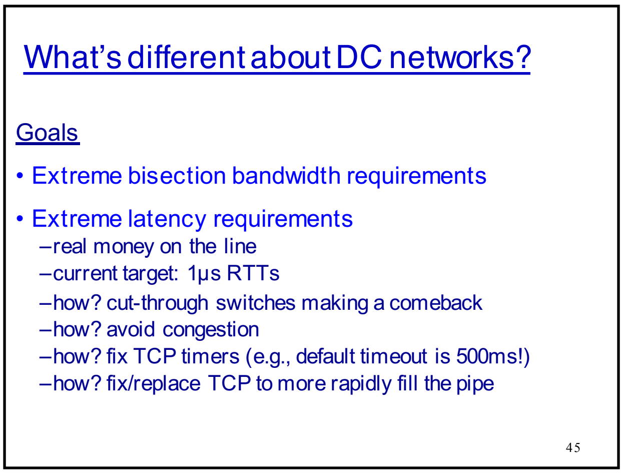
- Build multi-stage `Fat Trees’ out of k-port switches
  - –k/2 ports up, k/2 down
  - –Supports $k^3/4$ hosts:
    - 48 ports, 27,648 hosts

All links are the same speed (e.g. 10Gps)



Full Bisection Bandwidth Not Sufficient

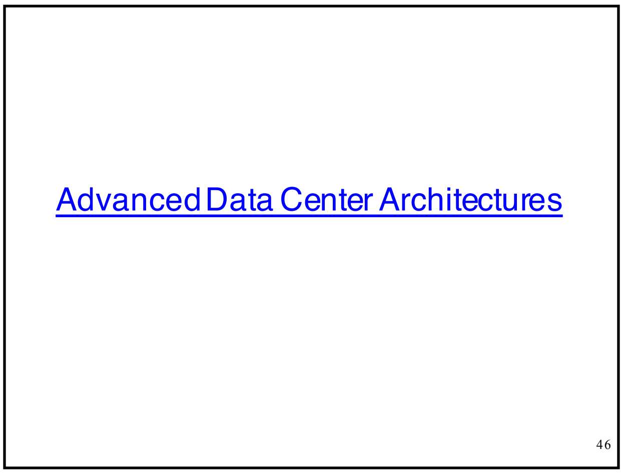
- To realize full bisectional throughput, routing must spread traffic across paths
- Enter load-balanced routing
  - –How? (1) Let the network split traffic/flows at random (e.g., ECMP protocol – RFC 2991/2992)
  - –How? (2) Centralized flow scheduling?
  - –Many more research proposals



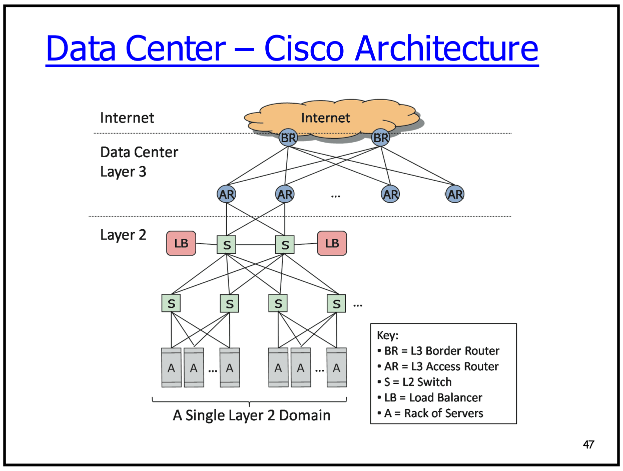
What’s different about DC networks?

Goals

- Extreme bisection bandwidth requirements
- Extreme latency requirements
  - –real money on the line
  - –current target: 1μs RTTs
  - –how? cut-through switches making a comeback
  - –how? avoid congestion
  - –how? fix TCP timers (e.g., default timeout is 500ms!)
  - –how? fix/replace TCP to more rapidly fill the pipe

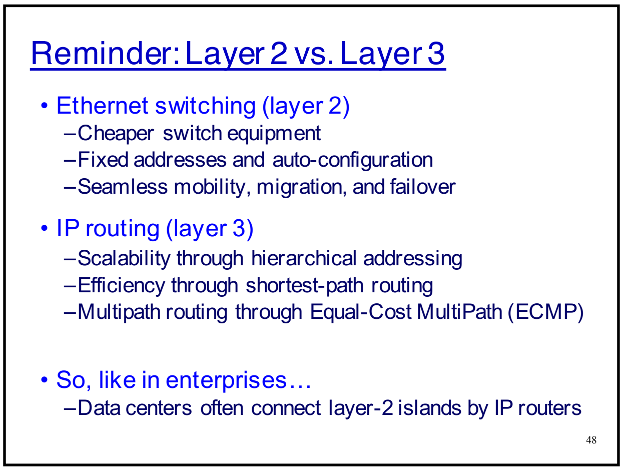


Advanced Data Center Architectures



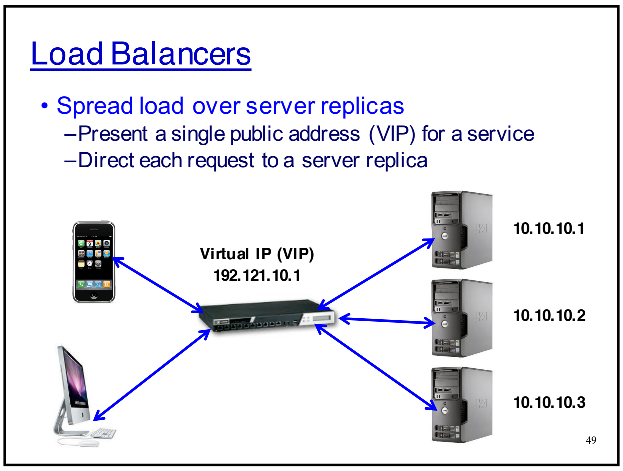
Data Center – Cisco Architecture

Key:
- BR = L3 Border Router
- AR = L3 Access Router
- S = L2 Switch
- LB = Load Balancer
- A = Rack of Servers



Reminder: Layer 2 vs. Layer 3

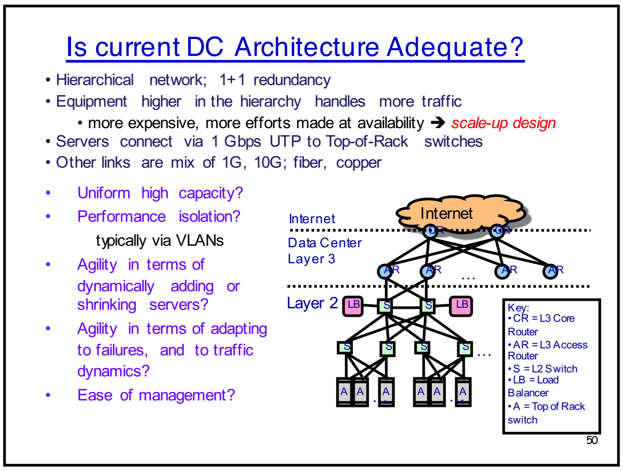
- Ethernet switching (layer 2)
  - –Cheaper switch equipment
  - –Fixed addresses and auto-configuration
  - –Seamless mobility, migration, and failover
- IP routing (layer 3)
  - –Scalability through hierarchical addressing
  - –Efficiency through shortest-path routing
  - –Multipath routing through Equal-Cost MultiPath (ECMP)
- So, like in enterprises…
  - –Data centers often connect layer-2 islands by IP routers



Load Balancers

- Spread load over server replicas
  - –Present a single public address (VIP) for a service
  - –Direct each request to a server replica

49



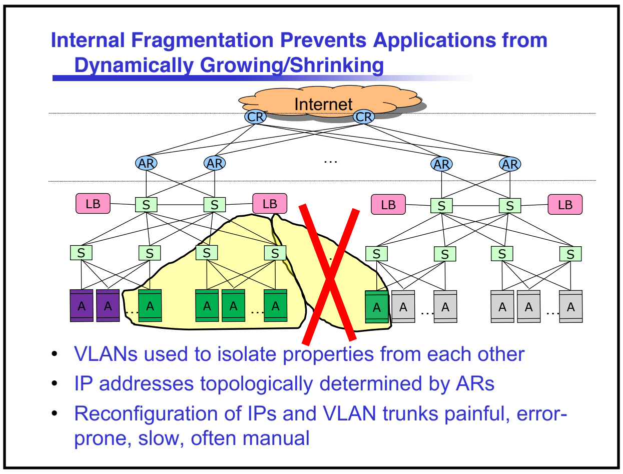
Is current DC Architecture Adequate?

- Hierarchical network; 1+1 redundancy
- Equipment higher in the hierarchy handles more traffic
  - more expensive, more efforts made at availability ➔ *scale-up design*
- Servers connect via 1 Gbps UTP to Top-of-Rack switches
- Other links are mix of 1G, 10G; fiber, copper

- Uniform high capacity?
- Performance isolation?
  typically via VLANs
- Agility in terms of dynamically adding or shrinking servers?
- Agility in terms of adapting to failures, and to traffic dynamics?
- Ease of management?

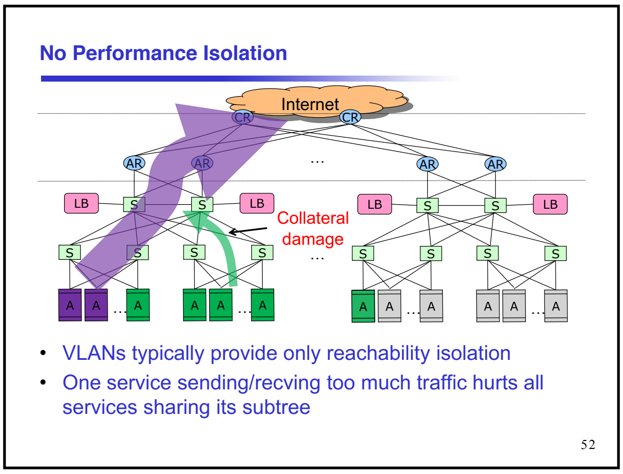
Key:
- CR = L3 Core Router
- AR = L3 Access Router
- S = L2 Switch
- LB = Load Balancer
- A = Top of Rack switch

50



Internal Fragmentation Prevents Applications from Dynamically Growing/Shrinking

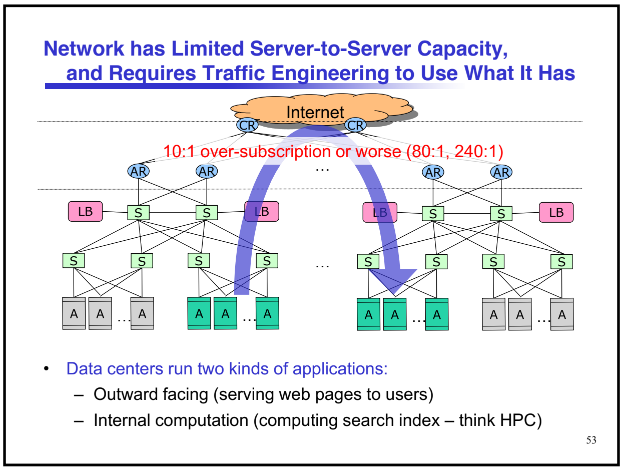
- VLANs used to isolate properties from each other
- IP addresses topologically determined by ARs
- Reconfiguration of IPs and VLAN trunks painful, error-prone, slow, often manual



No Performance Isolation

- VLANs typically provide only reachability isolation
- One service sending/recving too much traffic hurts all services sharing its subtree

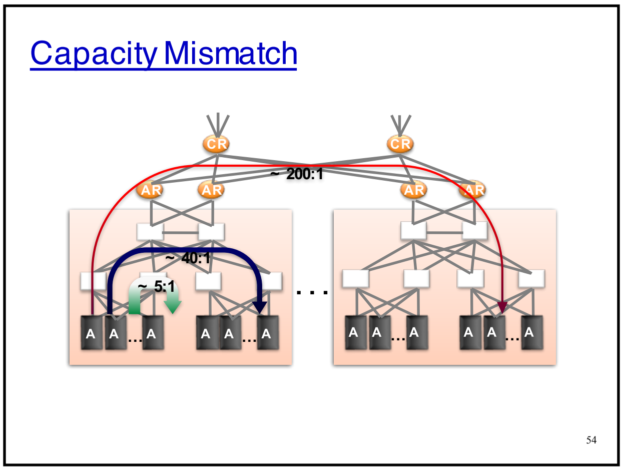
52



Network has Limited Server-to-Server Capacity, and Requires Traffic Engineering to Use What It Has

- Data centers run two kinds of applications:
  - Outward facing (serving web pages to users)
  - Internal computation (computing search index – think HPC)

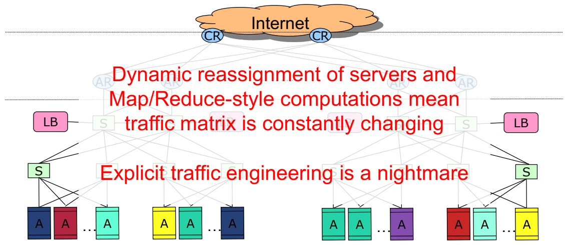
53



Capacity Mismatch

54

Network Needs Greater Bisection BW, and Requires Traffic Engineering to Use What It Has



Dynamic reassignment of servers and Map/Reduce-style computations mean traffic matrix is constantly changing

Explicit traffic engineering is a nightmare

- Data centers run two kinds of applications:
  - Outward facing (serving web pages to users)
  - Internal computation (computing search index – think HPC)

Objectives for the Network of Single Data Center

Developers want ***network virtualization***: a mental model where all their servers, and only their servers, are plugged into an Ethernet switch

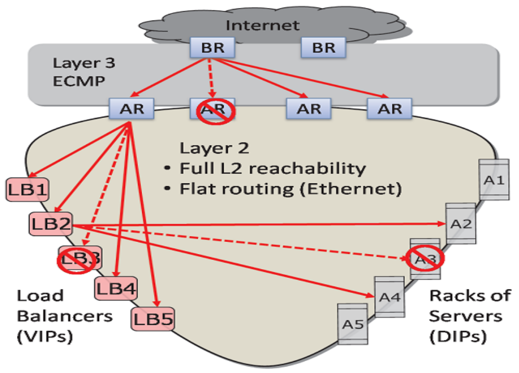
- Uniform high capacity
  - Capacity between two servers limited only by their NICs
  - No need to consider topology when adding servers
- Performance isolation
  - Traffic of one service should be unaffected by others
- Layer-2 semantics
  - Flat addressing, so any server can have any IP address
  - Server configuration is the same as in a LAN
  - Legacy applications depending on broadcast must work

Monsoon

Monsoon approach

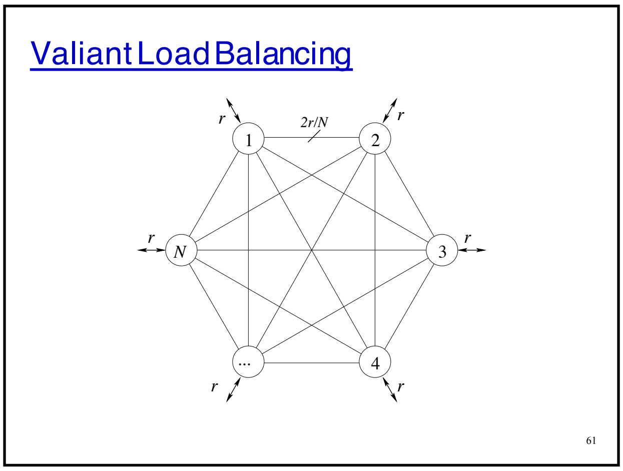
- Layer 2 based using commodity switches
- Hierarchy has 2 types of switches:
  - access switches (top of rack)
  - load balancing switches
- Eliminate spanning tree
  - Flat routing
  - Allows network to take advantage of path diversity
- Prevent MAC address learning
  - Monsoon Agent distribute data plane information
  - TOR: Only need to learn address for the intermediate switches
  - Core: learn for TOR switches
- Support efficient grouping of hosts (VLAN replacement)

Moonson



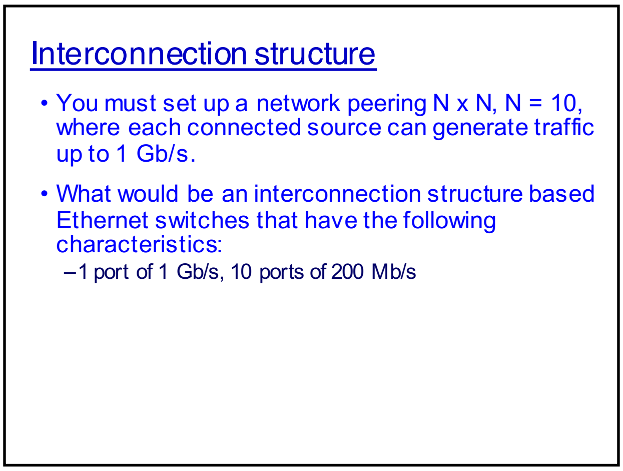
Monsoon Components

- Top-of-Rack switch:
  - Aggregate traffic from 20 end host in a rack
  - Performs IP to MAC translation
- Intermediate Switch
  - Disperses traffic
  - Balances traffic among switches
  - Used for Valiant load balancing
- Decision Element
  - Places routes in switches
  - Maintain a directory services of IP to MAC
- Endhost
  - Performs IP to MAC lookup



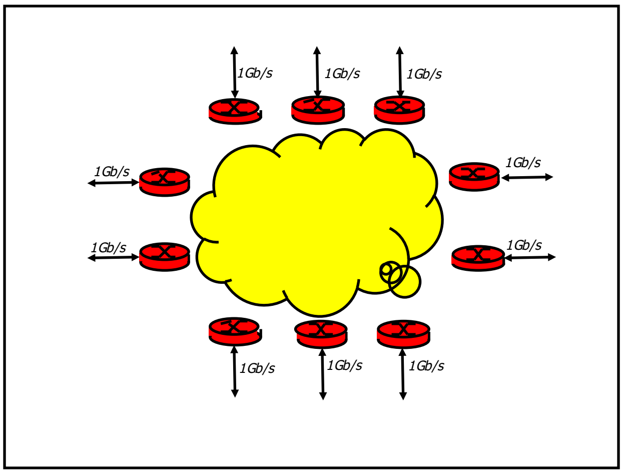
Valiant Load Balancing

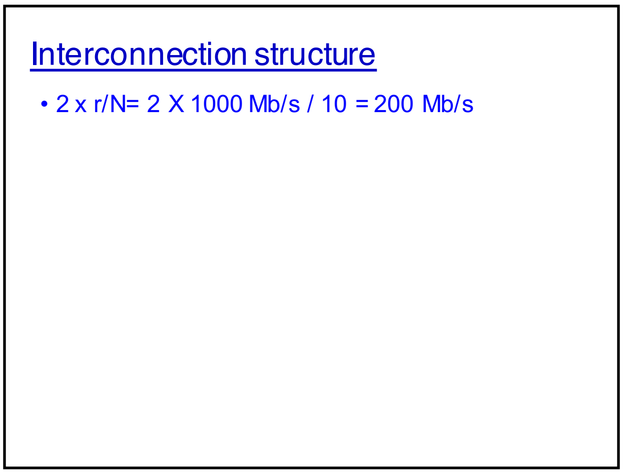
61



Interconnection structure

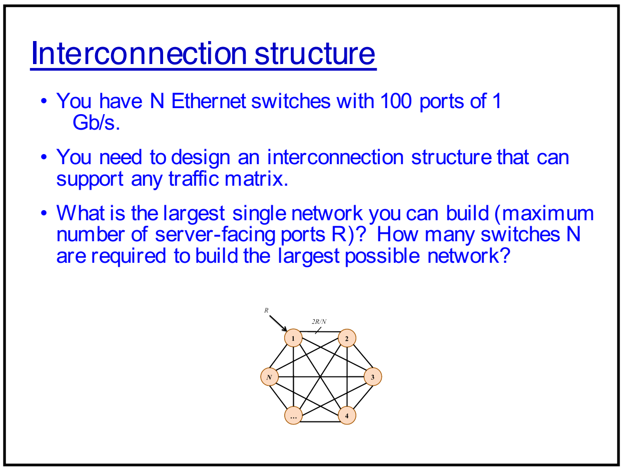
- You must set up a network peering N x N, N = 10, where each connected source can generate traffic up to 1 Gb/s.
- What would be an interconnection structure based Ethernet switches that have the following characteristics:
  - –1 port of 1 Gb/s, 10 ports of 200 Mb/s





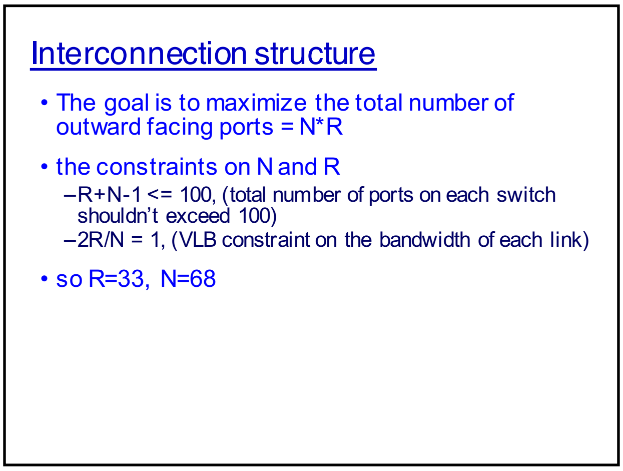
Interconnection structure

- 2 x r/N= 2 X 1000 Mb/s / 10 = 200 Mb/s



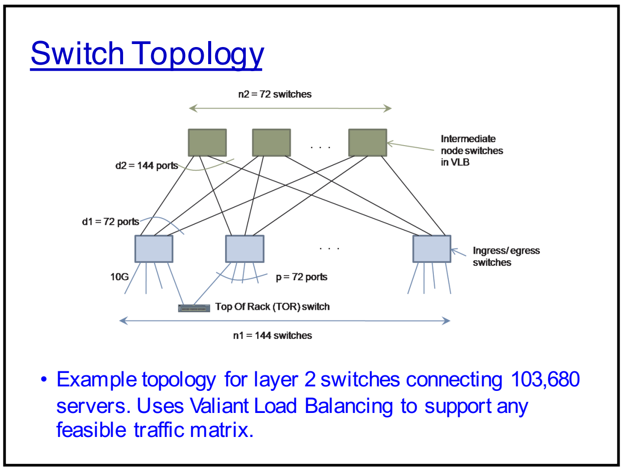
Interconnection structure

- You have N Ethernet switches with 100 ports of 1 Gb/s.
- You need to design an interconnection structure that can support any traffic matrix.
- What is the largest single network you can build (maximum number of server-facing ports R)? How many switches N are required to build the largest possible network?



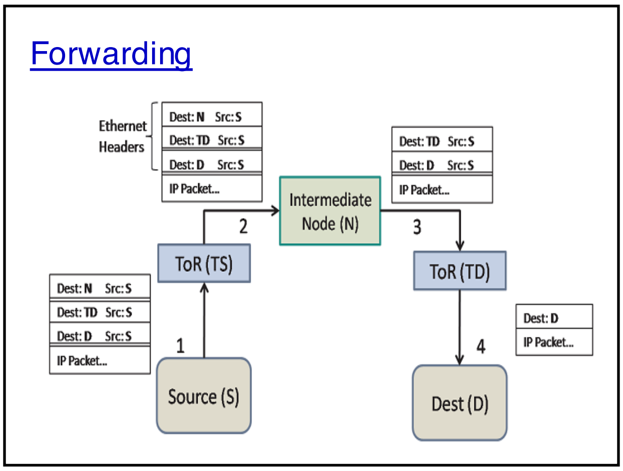
Interconnection structure

- The goal is to maximize the total number of outward facing ports = N*R
- the constraints on N and R
  - –R+N-1 <= 100, (total number of ports on each switch shouldn’t exceed 100)
  - –2R/N = 1, (VLB constraint on the bandwidth of each link)
- so R=33, N=68

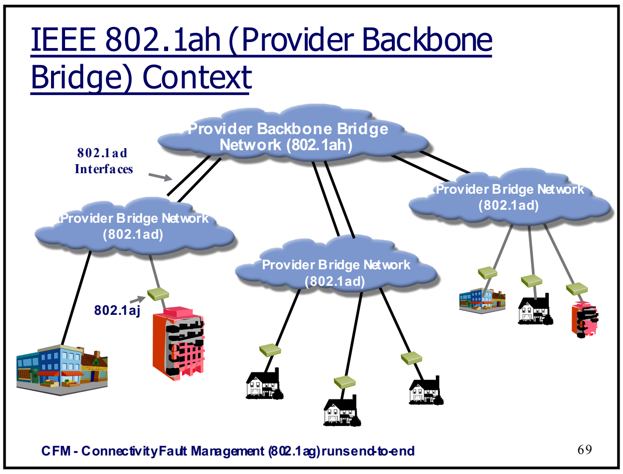


Switch Topology

- Example topology for layer 2 switches connecting 103,680 servers. Uses Valiant Load Balancing to support any feasible traffic matrix.



Forwarding

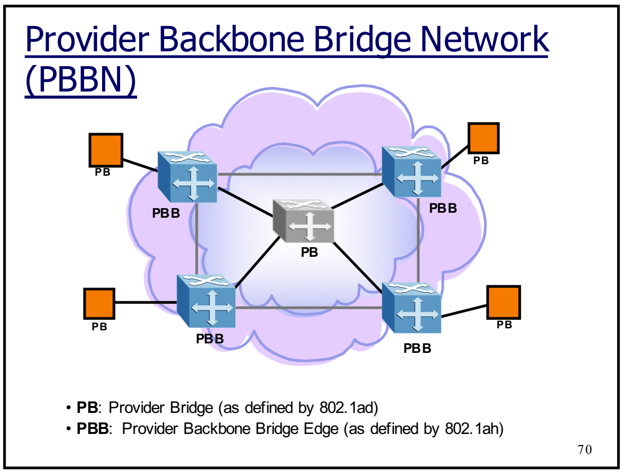


IEEE 802.1ah (Provider Backbone Bridge) Context

Provider Backbone Bridge Network (802.1ah)
802.1ad Interfaces
Provider Bridge Network (802.1ad)
Provider Bridge Network (802.1ad)
Provider Bridge Network (802.1ad)
802.1aj

CFM - Connectivity Fault Management (802.1ag) runs end-to-end

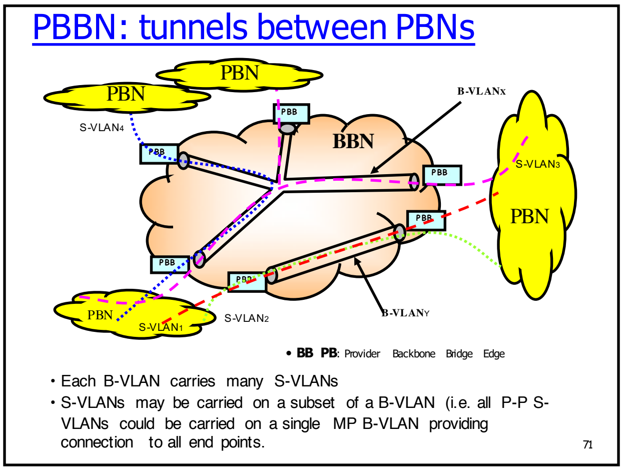
69



Provider Backbone Bridge Network (PBBN)

- **PB**: Provider Bridge (as defined by 802.1ad)
- **PBB**: Provider Backbone Bridge Edge (as defined by 802.1ah)

70

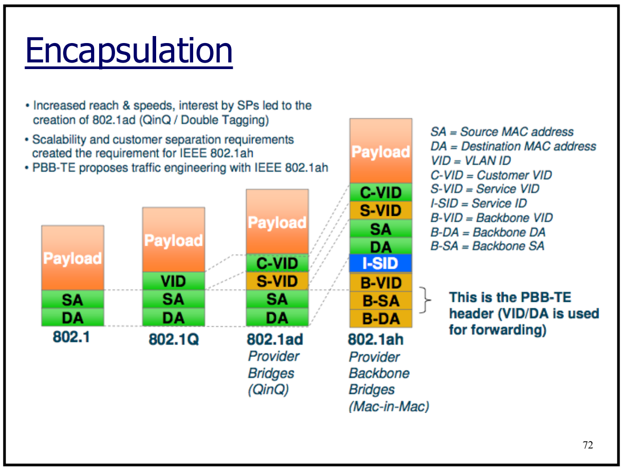


PBBN: tunnels between PBNs

- **BB PB**: Provider Backbone Bridge Edge

- Each B-VLAN carries many S-VLANs
- S-VLANs may be carried on a subset of a B-VLAN (i.e. all P-P S-VLANs could be carried on a single MP B-VLAN providing connection to all end points.

71



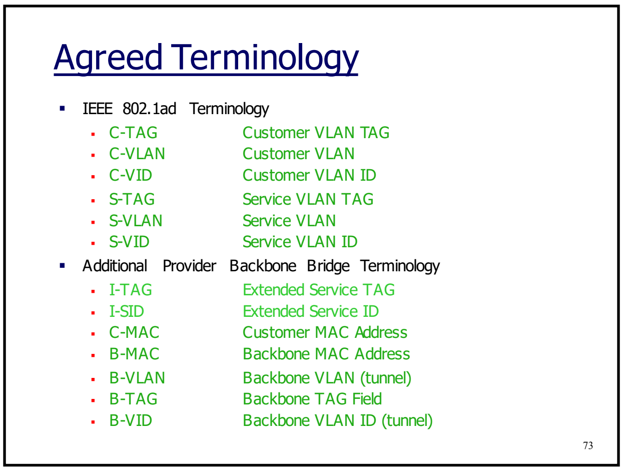
Encapsulation

- Increased reach & speeds, interest by SPs led to the creation of 802.1ad (QinQ / Double Tagging)
- Scalability and customer separation requirements created the requirement for IEEE 802.1ah
- PBB-TE proposes traffic engineering with IEEE 802.1ah

SA = Source MAC address
DA = Destination MAC address
VID = VLAN ID
C-VID = Customer VID
S-VID = Service VID
I-SID = Service ID
B-VID = Backbone VID
B-DA = Backbone DA
B-SA = Backbone SA

This is the PBB-TE header (VID/DA is used for forwarding)

72



Agreed Terminology

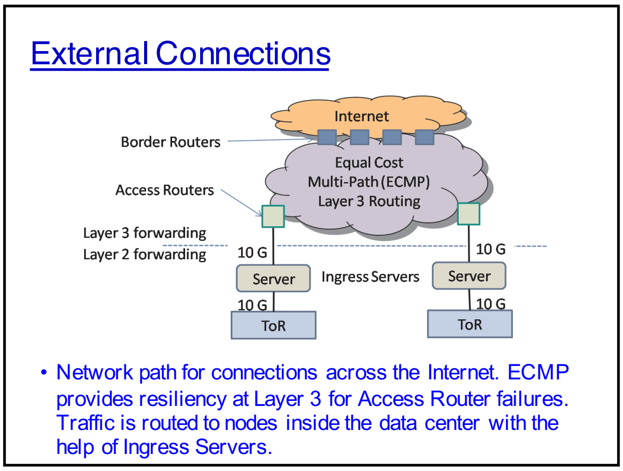
- IEEE 802.1ad Terminology

C-TAG	Customer VLAN TAG
C-VLAN	Customer VLAN
C-VID	Customer VLAN ID
S-TAG	Service VLAN TAG
S-VLAN	Service VLAN
S-VID	Service VLAN ID

- Additional Provider Backbone Bridge Terminology

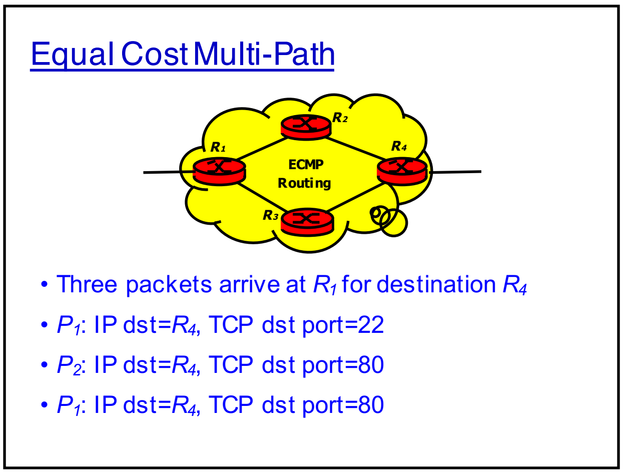
I-TAG	Extended Service TAG
I-SID	Extended Service ID
C-MAC	Customer MAC Address
B-MAC	Backbone MAC Address
B-VLAN	Backbone VLAN (tunnel)
B-TAG	Backbone TAG Field
B-VID	Backbone VLAN ID (tunnel)

73



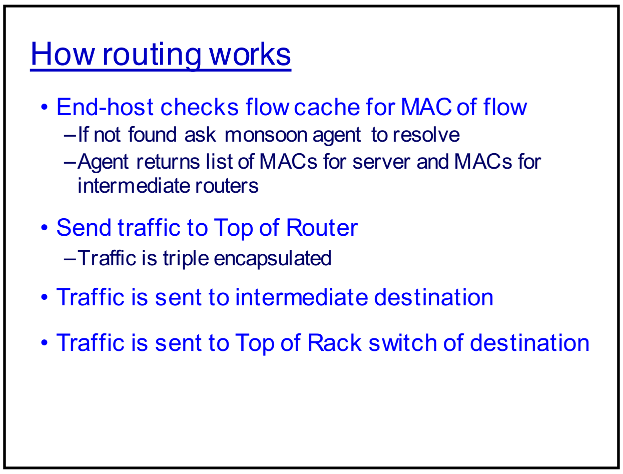
External Connections

- Network path for connections across the Internet. ECMP provides resiliency at Layer 3 for Access Router failures. Traffic is routed to nodes inside the data center with the help of Ingress Servers.



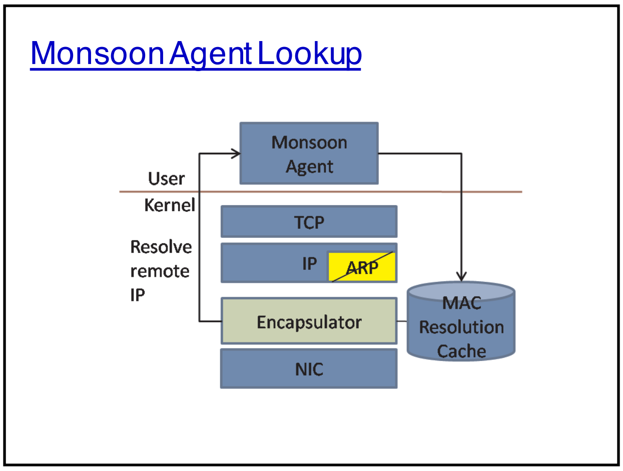
Equal Cost Multi-Path

- Three packets arrive at R_1 for destination R_4
- P_1: IP dst=R_4, TCP dst port=22
- P_2: IP dst=R_4, TCP dst port=80
- P_1: IP dst=R_4, TCP dst port=80



How routing works

- End-host checks flow cache for MAC of flow
  - If not found ask monsoon agent to resolve
  - Agent returns list of MACs for server and MACs for intermediate routers
- Send traffic to Top of Router
  - Traffic is triple encapsulated
- Traffic is sent to intermediate destination
- Traffic is sent to Top of Rack switch of destination



Monsoon Agent Lookup