Data Center Challenges

Building Networks for Agility

Sreenivas Addagatla, Albert Greenberg,
James Hamilton, Navendu Jain,
Srikanth Kandula, Changhoon Kim,
Parantap Lahiri, David A. Maltz, Parveen Patel
Sudipta Sengupta

Microsoft Research

Agenda

- Brief characterization of "mega" cloud data centers based on industry studies
  - Costs
  - Pain-points with today's network
  - Traffic pattern characteristics in data centers
- VL2: a technology for building data center networks
  - Provides what data center tenants & owners want
    - Network virtualization
    - Uniform high capacity and performance isolation
    - Low cost and high reliability with simple mgmt
  - Principles and insights behind VL2 (aka project Monsoon)
  - VL2 prototype and evaluation

What's a Cloud Service Data Center?

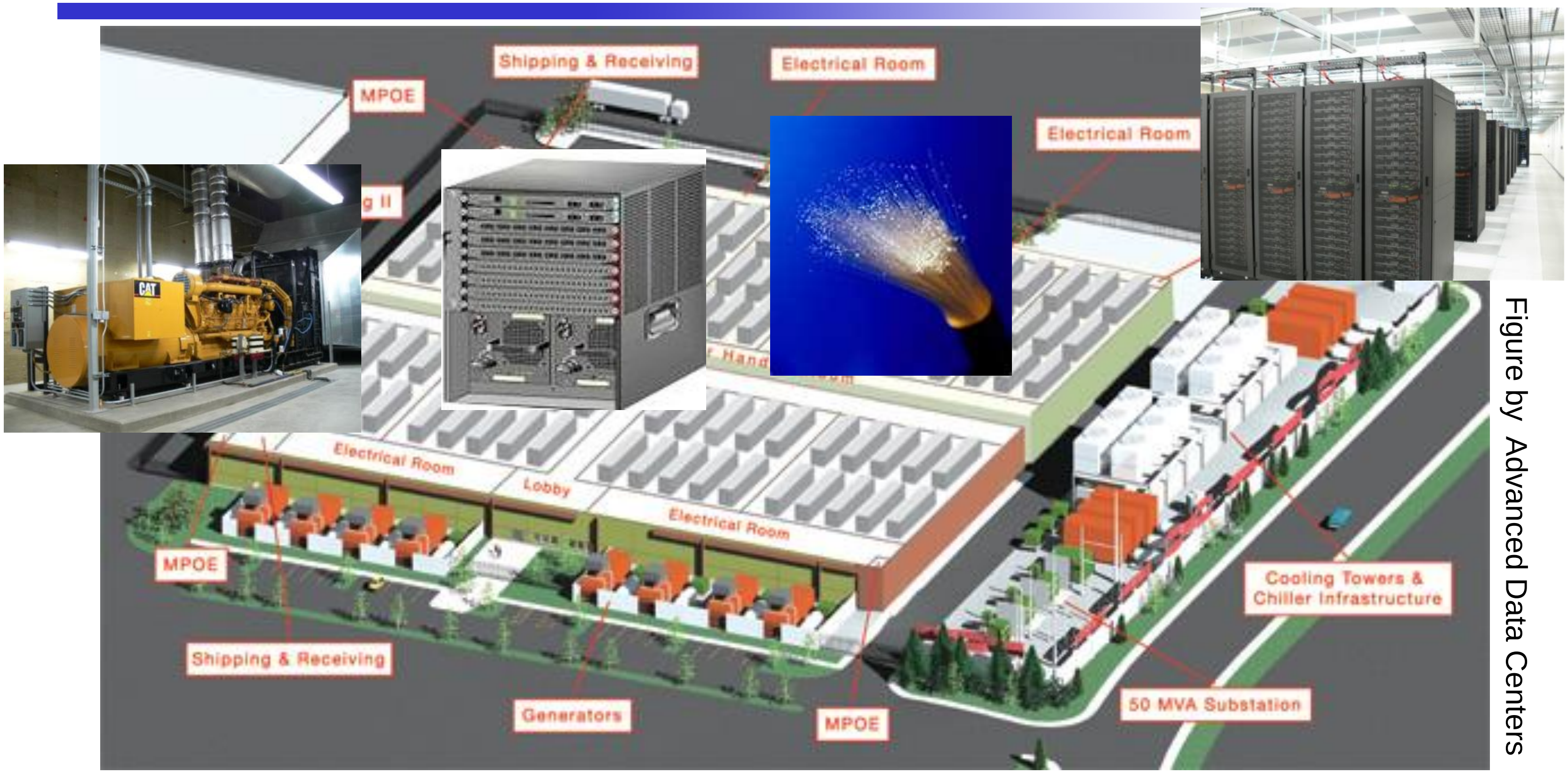


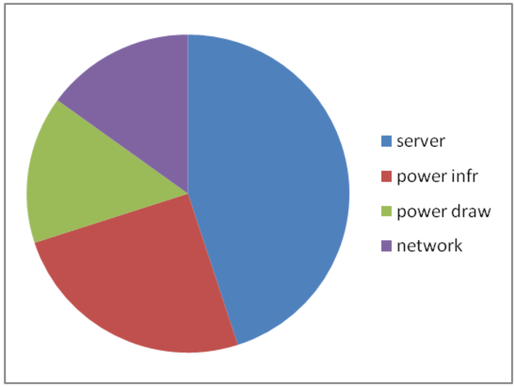
Figure by Advanced Data Centers

- 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

- Total cost varies
  - Upwards of $1/4 B for mega data center
  - Server costs dominate
  - Network costs significant



The Cost of a Cloud: Research Problems in Data Center Networks. Sigcomm CCR 2009. Greenberg, Hamilton, Maltz, Patel.

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

Data Centers are Like Factories

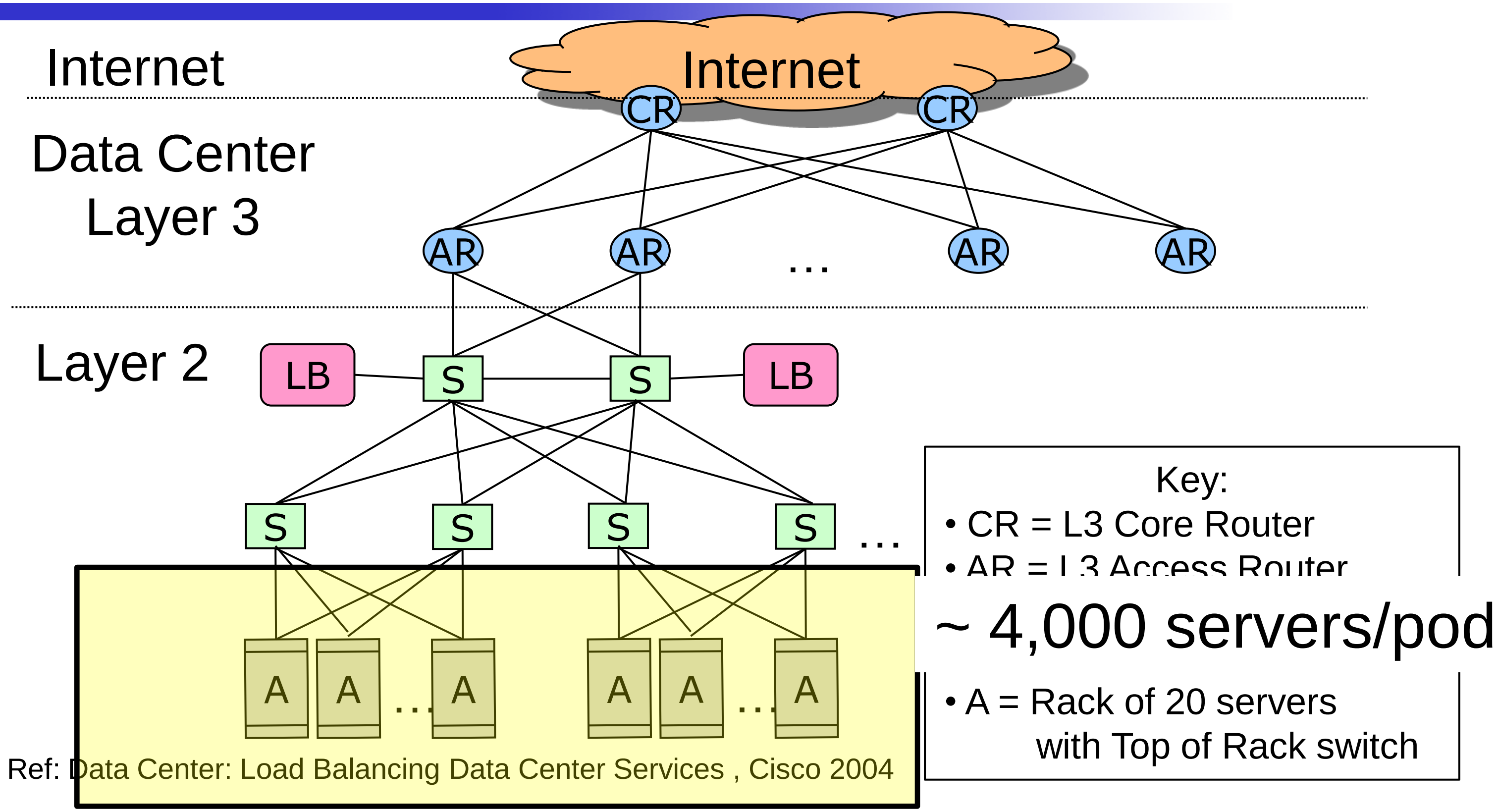
- Number 1 Goal:

  Maximize useful work per dollar spent

- Ugly secrets:
  - 10% to 30% CPU utilization considered “good” in DCs
  - There are servers that aren’t doing anything at all
- Cause:
  - Server are purchased rarely (roughly quarterly)
  - Reassigning servers among tenants is hard
  - Every tenant hoards servers

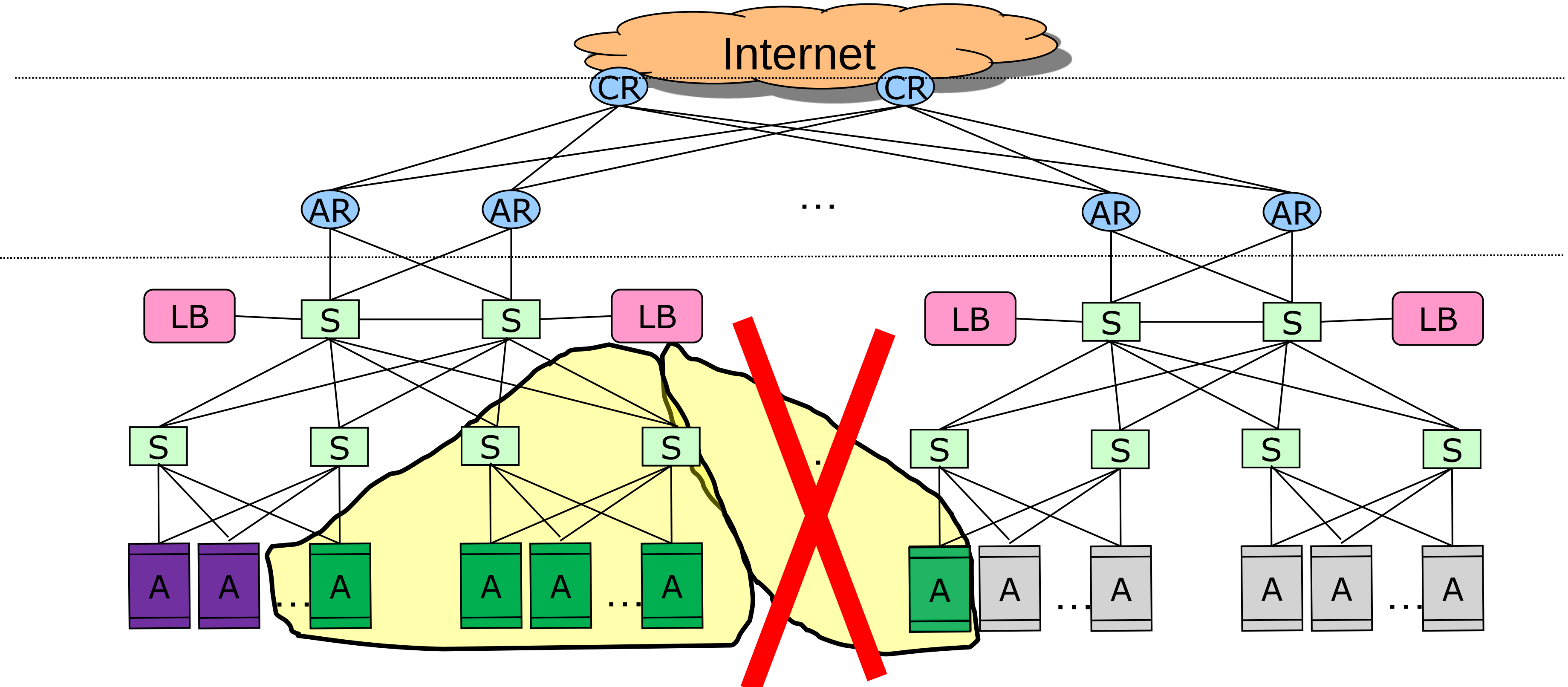
Solution: More agility: Any server, any service

The Network of a Modern Data Center



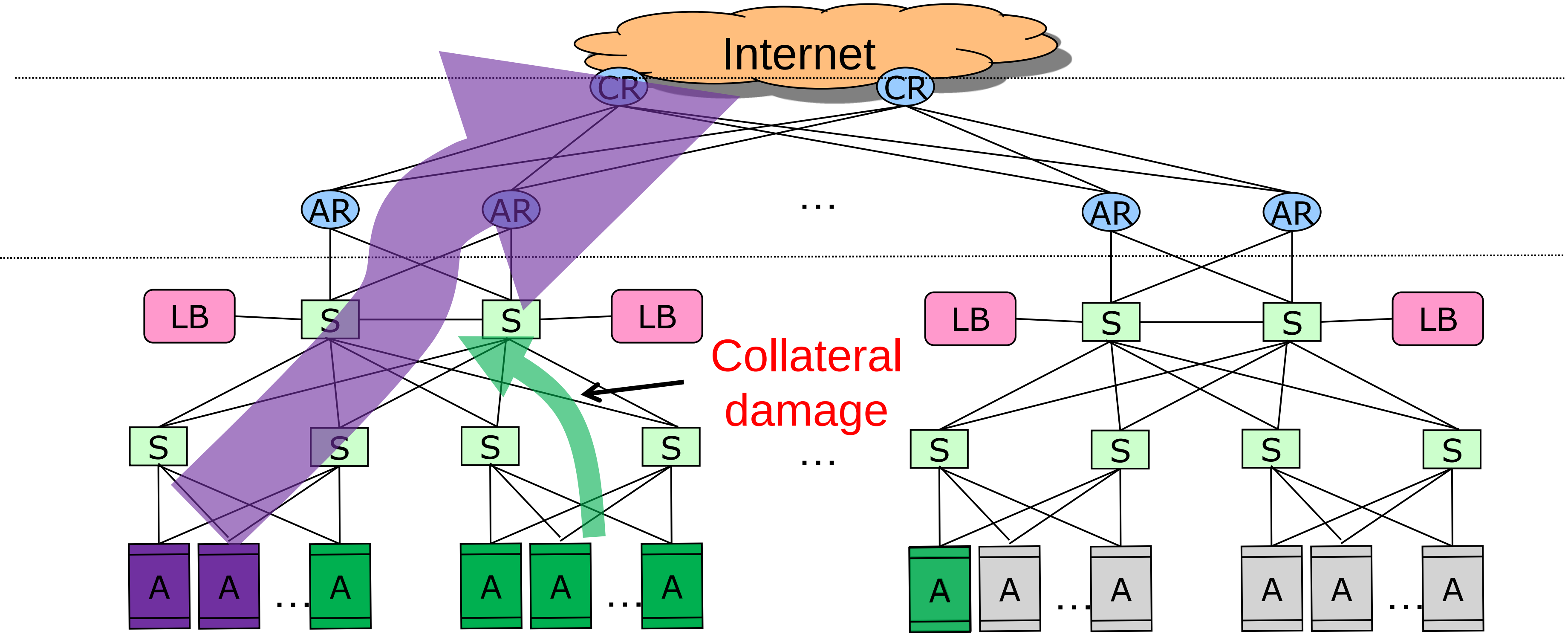
- Hierarchical network; 1+1 redundancy
- Equipment higher in the hierarchy handles more traffic, more expensive, more efforts made at availability ➔ *scale-up design*
- Servers connect via 1 Gbps UTP to Top of Rack switches
- Other links are mix of 1G, 10G; fiber, copper

Internal Fragmentation Prevents Applications from Dynamically Growing/Shrinking



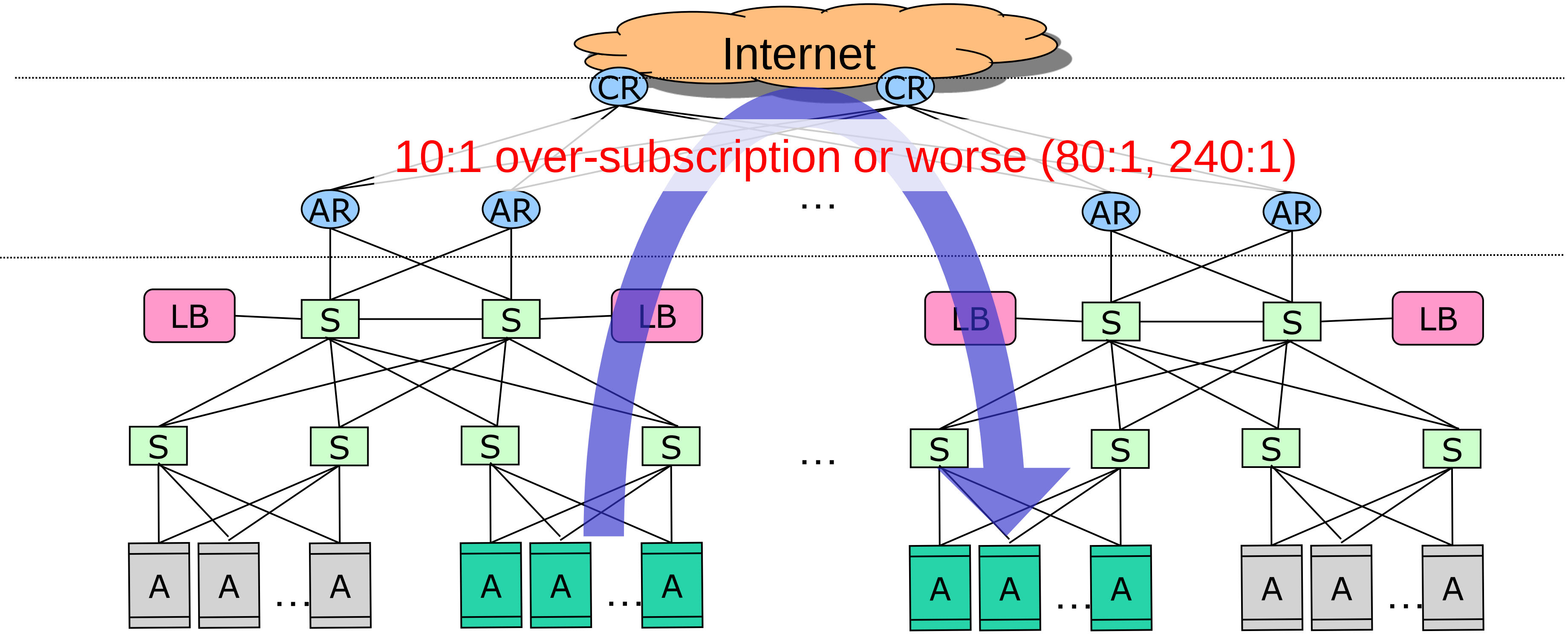
- VLANs used to isolate properties from each other
- IP addresses topologically determined by ARs
- Reconfiguration of IPs and VLAN trunks painful, error-prone, slow, often manual

No Performance Isolation



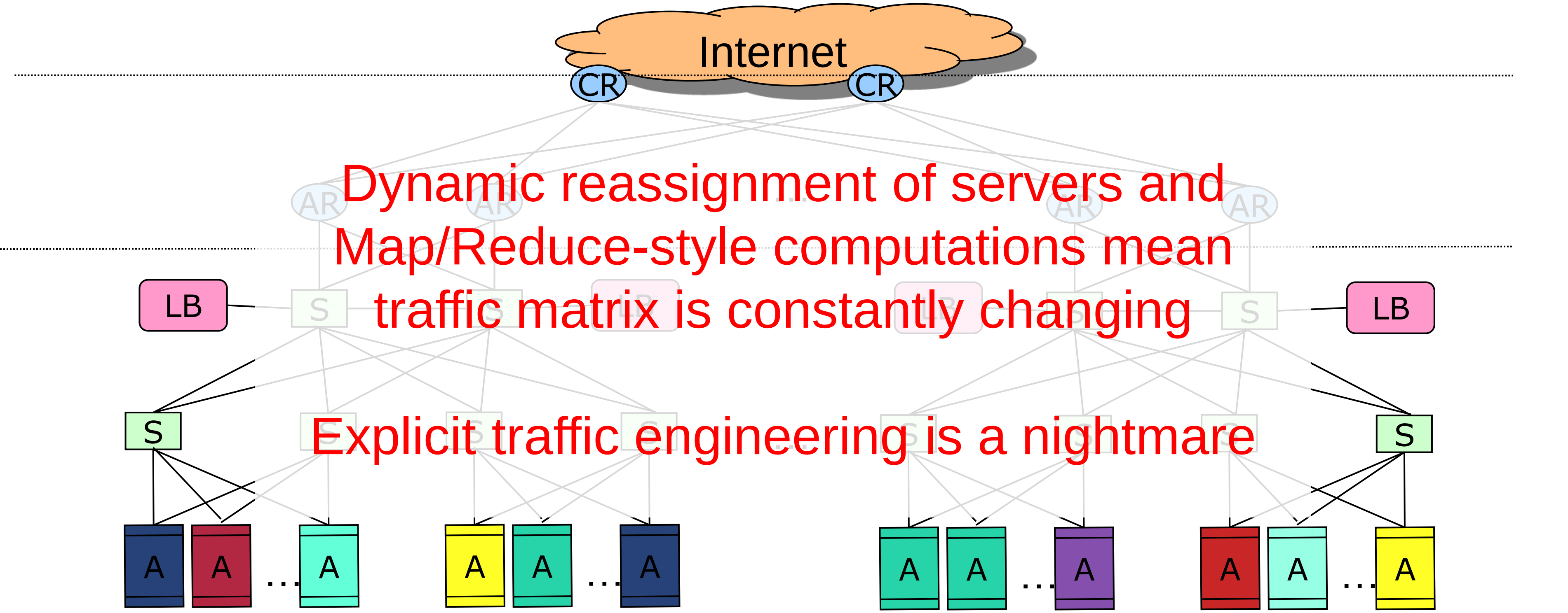
- VLANs typically provide only reachability isolation
- One service sending/recving too much traffic hurts all services sharing its subtree

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)

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

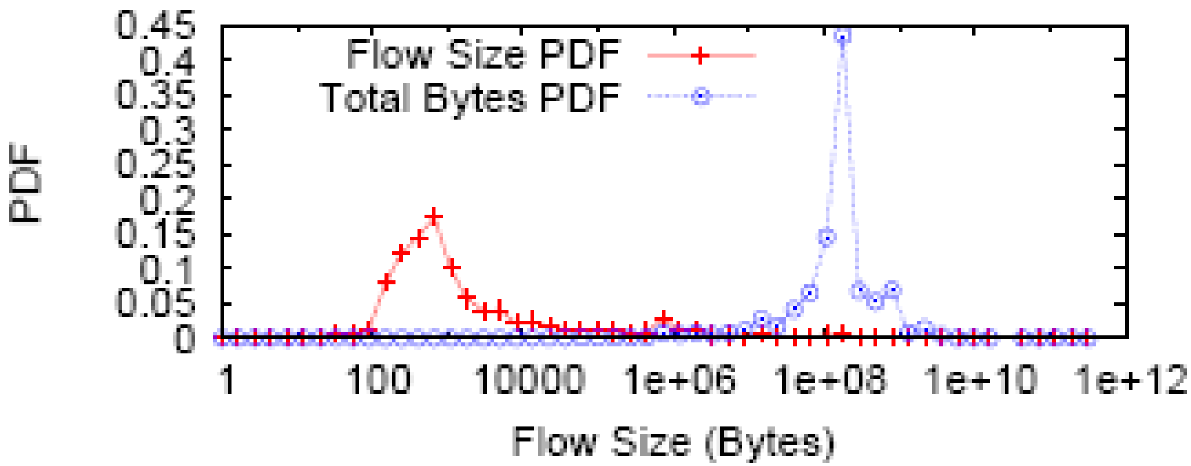
- Data centers run two kinds of applications:
  - Outward facing (serving web pages to users)
  - Internal computation (computing search index – think HPC)

Measuring Traffic in Today's Data Centers

- 80% of the packets stay inside the data center
  - Data mining, index computations, back end to front end
  - Trend is towards even more internal communication
- Detailed measurement study of data mining cluster
  - 1,500 servers, 79 ToRs
  - Logged: 5-tuple and size of all socket-level R/W ops
  - Aggregated in flows – all activity separated by < 60 s
  - Aggregated into traffic matrices every 100 s
    - Src, Dst, Bytes of data exchange

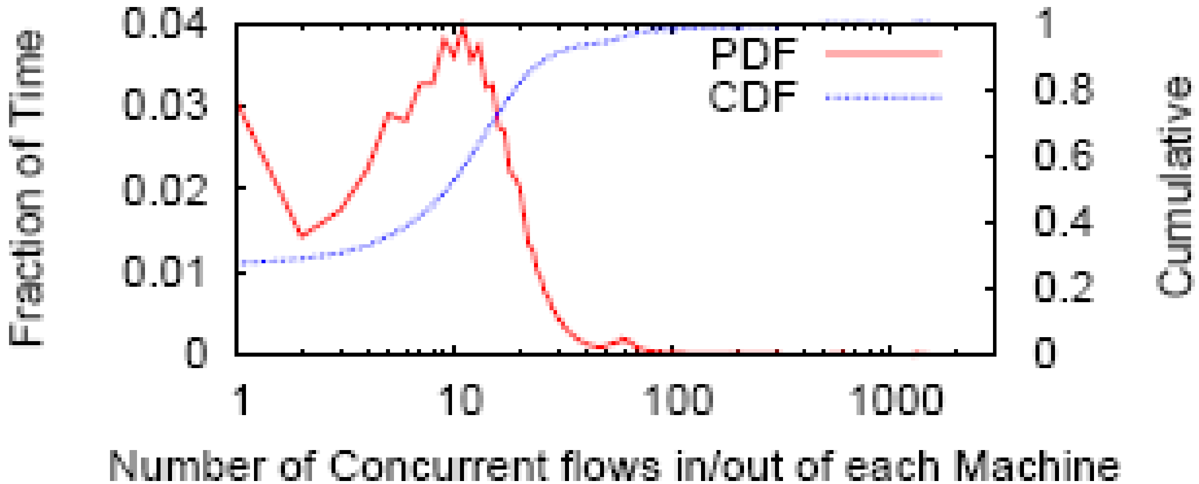
Flow Characteristics

DC traffic != Internet traffic



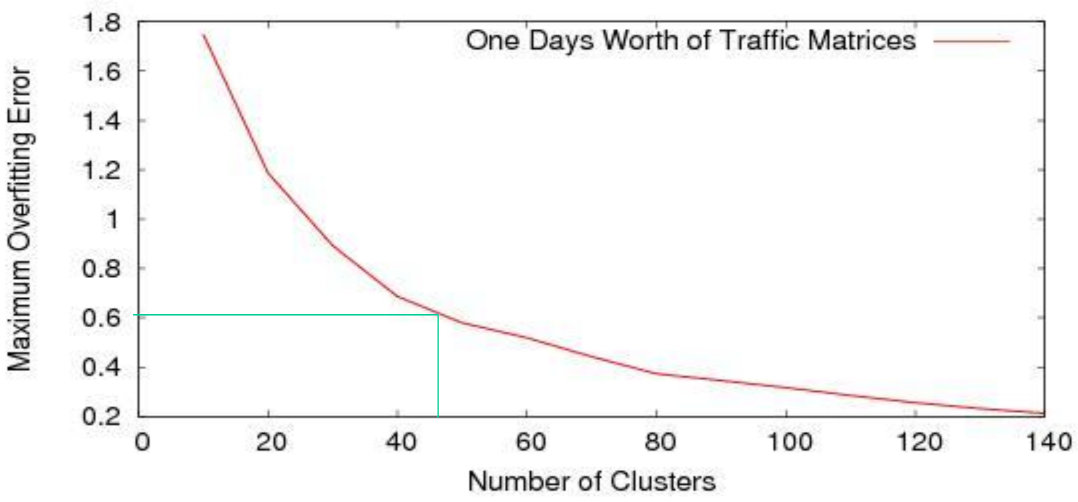
Most of the flows: various mice

Most of the bytes: within 100MB flows



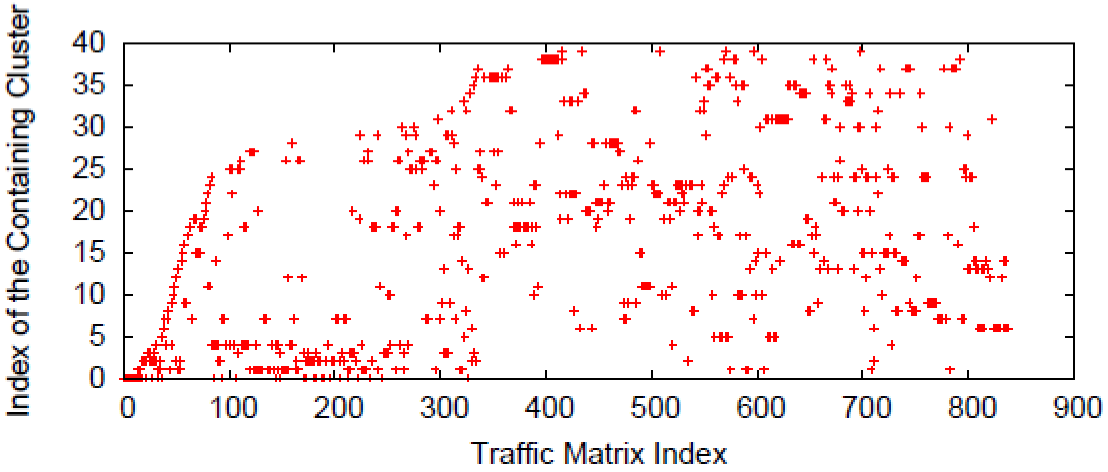
Median of 10 concurrent flows per server

Traffic Matrix Volatility



- Collapse similar traffic matrices into "clusters"

- Need 50-60 clusters to cover a day's traffic



- Traffic pattern changes nearly constantly

- Run length is 100s to 80% percentile; 99th is 800s

Today, Computation Constrained by Network*

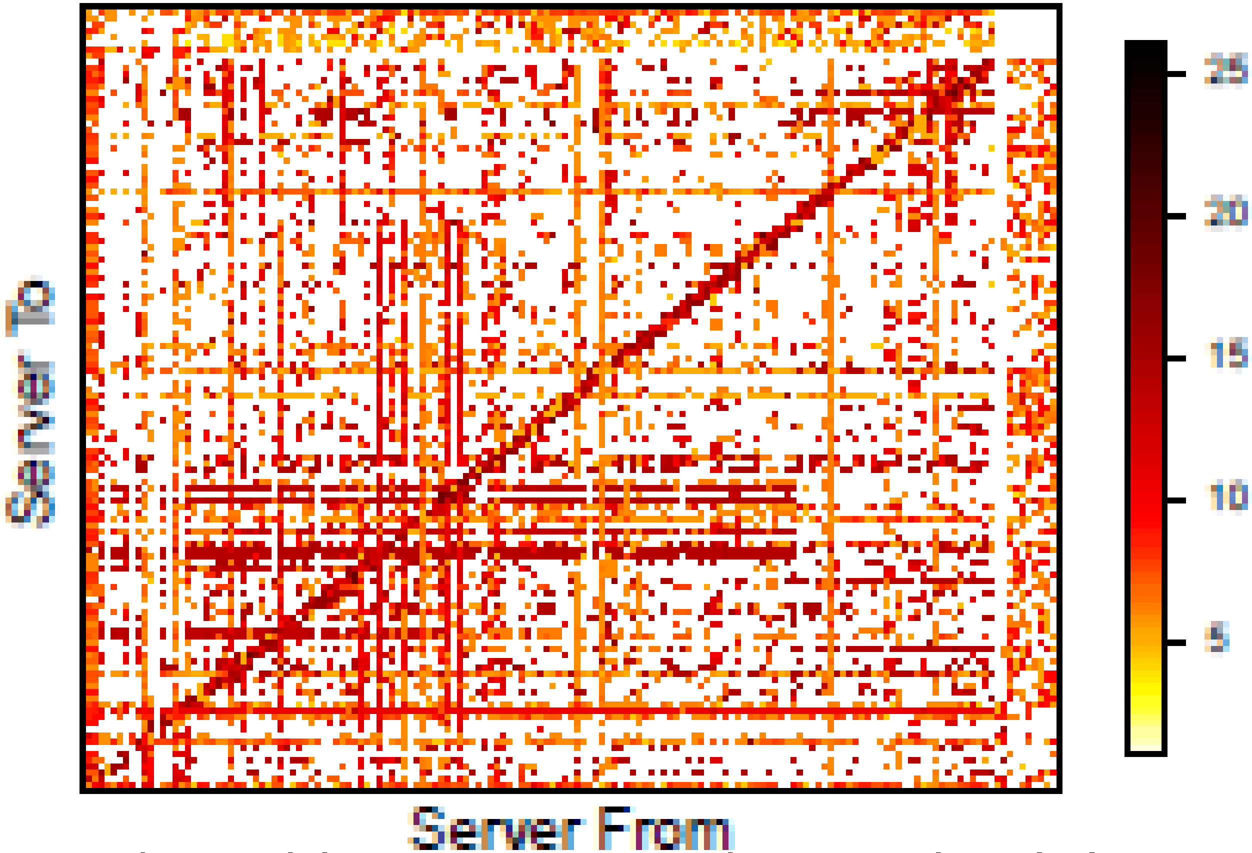


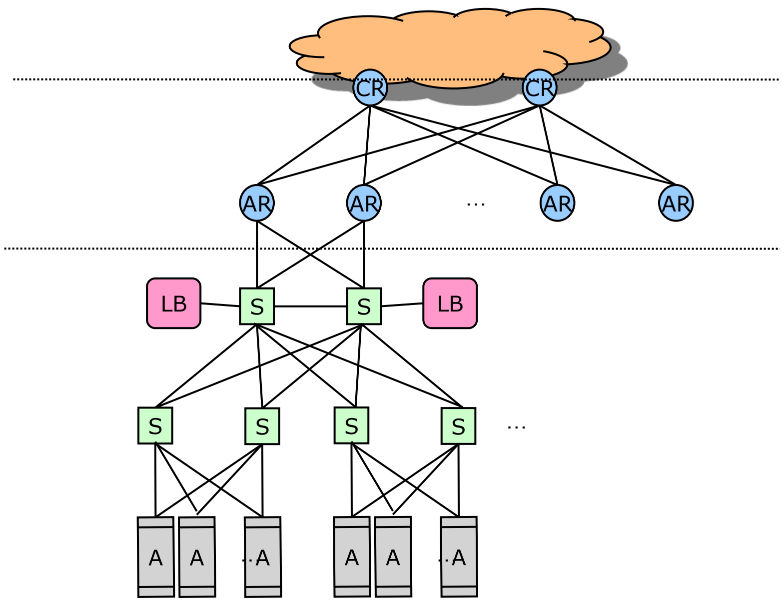
Figure: ln(Bytes/10sec) between servers in operational cluster

- Great efforts required to place communicating servers under the same ToR ➔ Most traffic lies on the diagonal
- Stripes show there is need for inter-ToR communication

*Kandula, Sengupta, Greenberg,Patel

What Do Data Center Faults Look Like?

- Need very high reliability near top of the tree
  - Very hard to achieve
    - Example: failure of a temporarily unpaired core switch affected ten million users for four hours
  - 0.3% of failure events knocked out all members of a network redundancy group
    - Typically at lower layers in tree, but not always



Ref: Data Center: Load Balancing Data Center Services , Cisco 2004

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

VL2: Distinguishing Design Principles

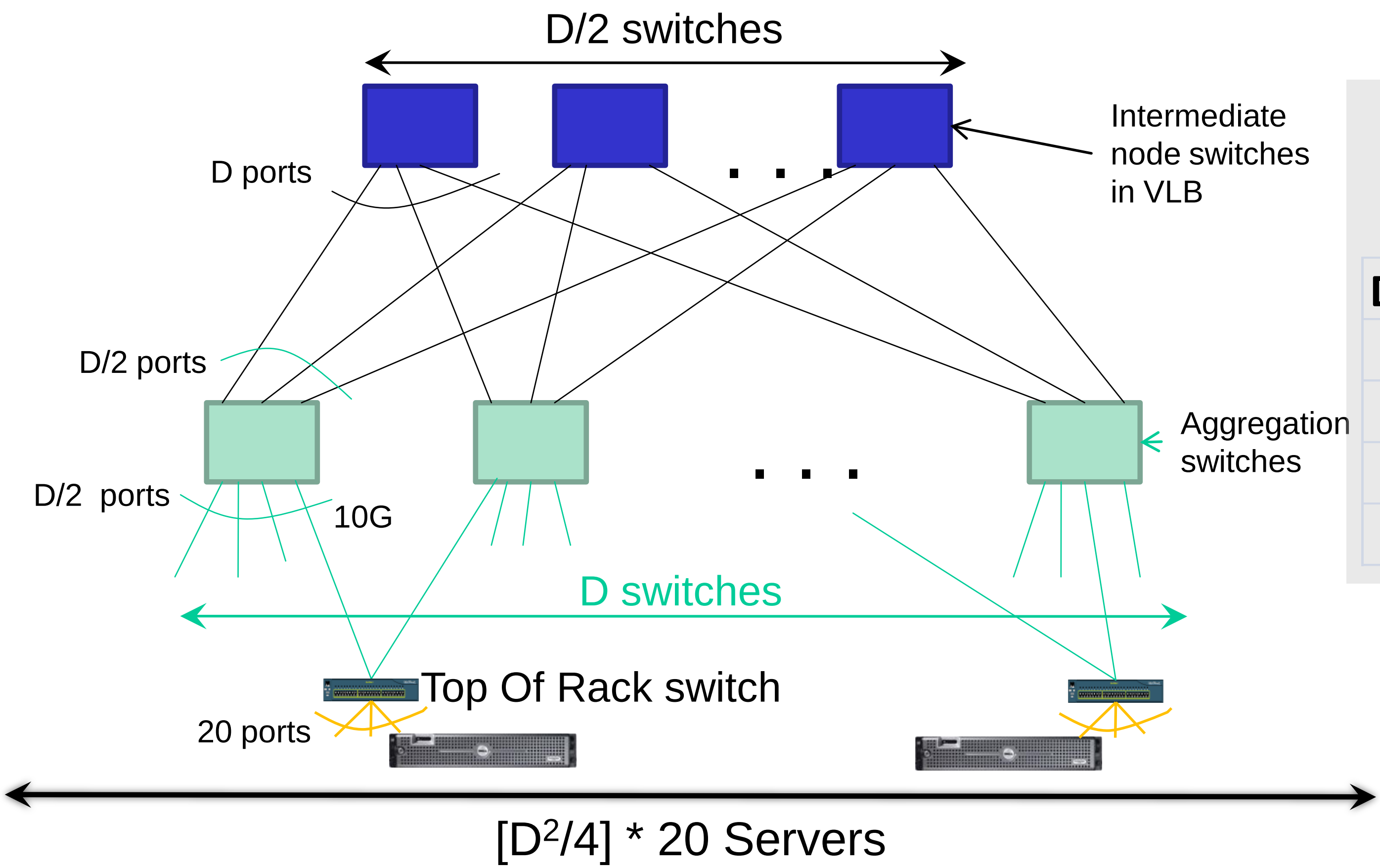
- Randomizing to Cope with Volatility
  - Tremendous variability in traffic matrices
- Separating Names from Locations
  - Any server, any service
- Leverage Strengths of End Systems
  - Programmable; big memories
- Building on Proven Networking Technology
  - We can build with parts shipping today
    - Leverage low cost, powerful merchant silicon ASICs, though do not rely on any one vendor
    - Innovate in software

What Enables a New Solution Now?

- Programmable switches with high port density
  - Fast: ASIC switches on a chip (Broadcom, Fulcrum, …)
  - Cheap: Small buffers, small forwarding tables
  - Flexible: Programmable control planes
- Centralized coordination
  - Scale-out data centers are not like enterprise networks
  - Centralized services already control/monitor health and role of each server (Autopilot)
  - Centralized directory and control plane acceptable (4D)

20 port 10GE switch. List price: $10K

An Example VL2 Topology: Clos Network

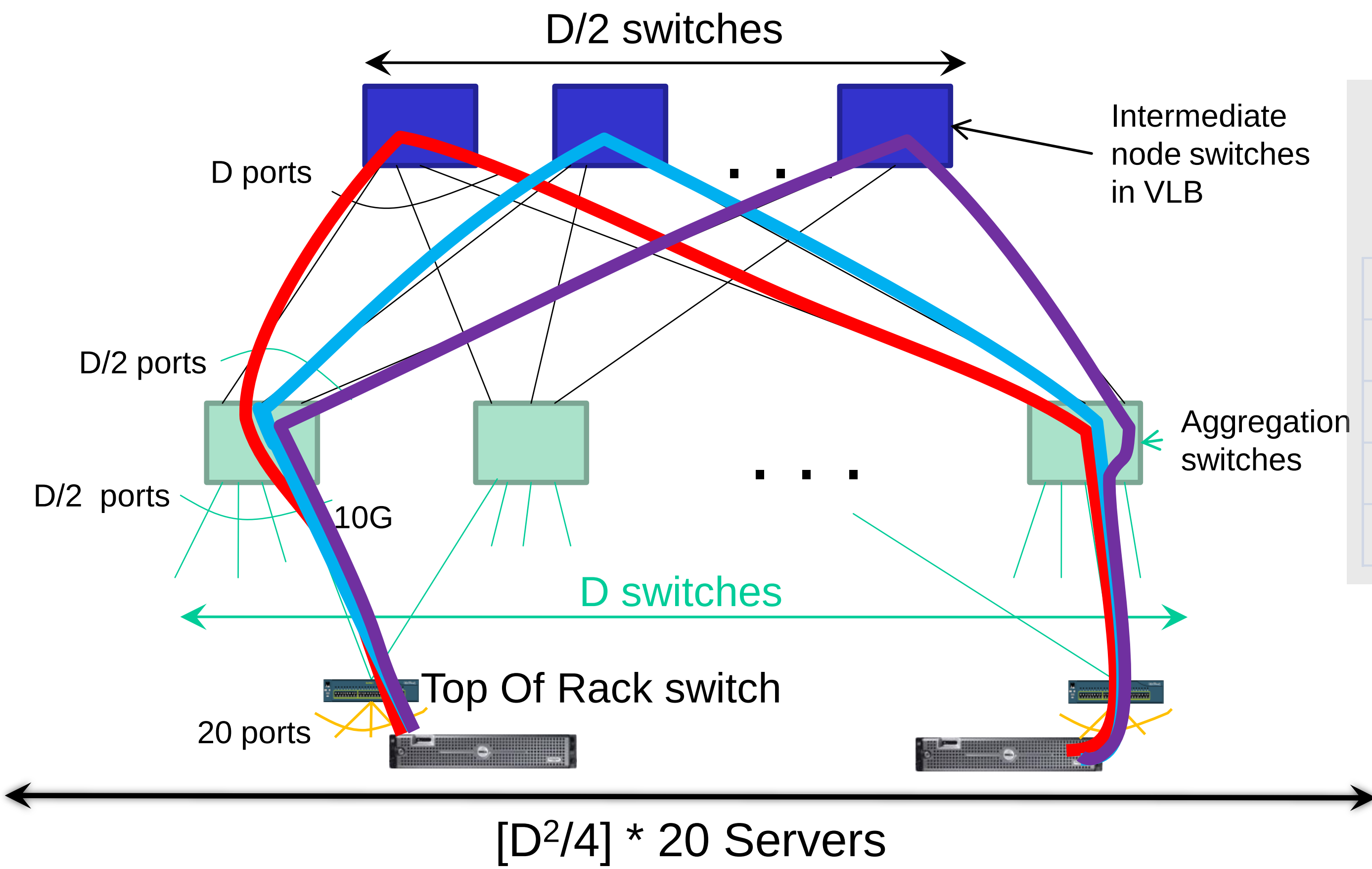


Node degree (D) of available switches & # servers supported

D	# Servers in pool
4	80
24	2,880
48	11,520
144	103,680

- A scale-out design with broad layers
  - Same bisection capacity at each layer ➔ no oversubscription
  - Extensive path diversity ➔ Graceful degradation under failure
  - ROC philosophy can be applied to the network switches

Use Randomization to Cope with Volatility

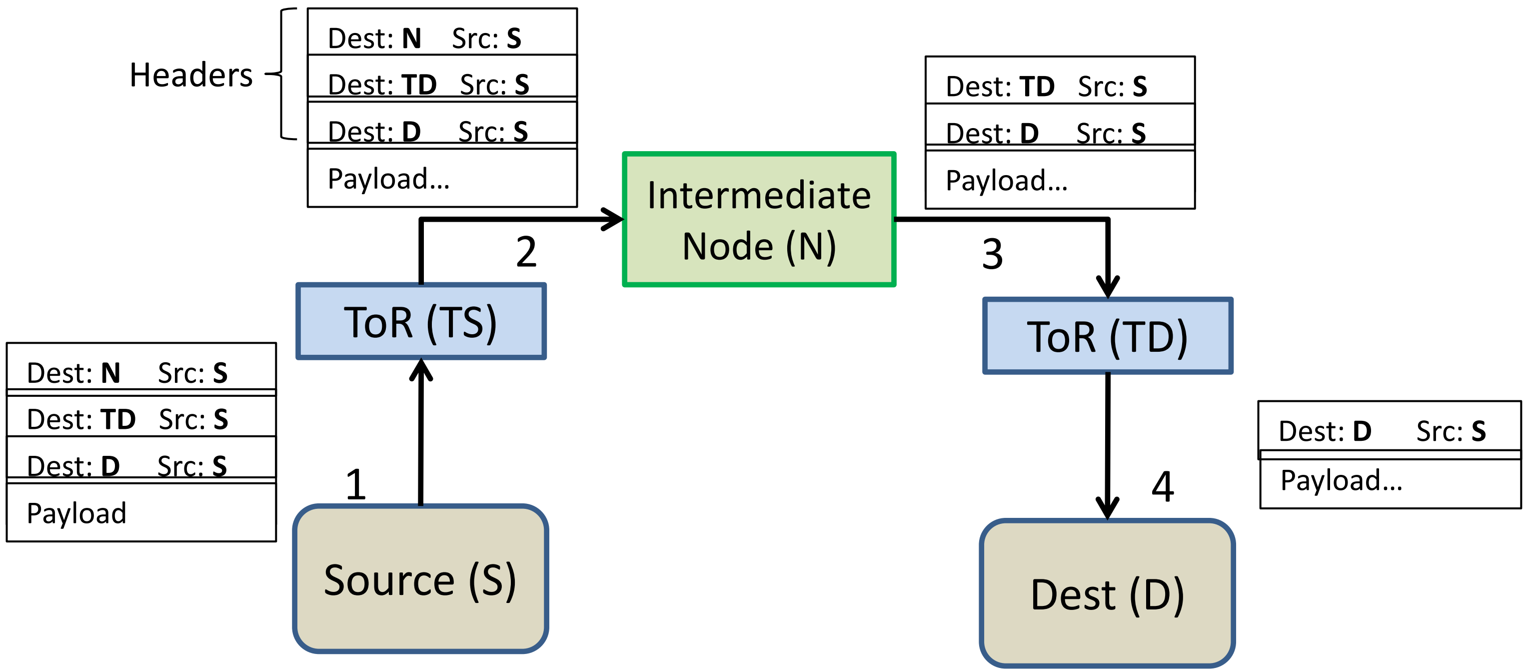


Node degree (D) of available switches & # servers supported

D	# Servers in pool
4	80
24	2,880
48	11,520
144	103,680

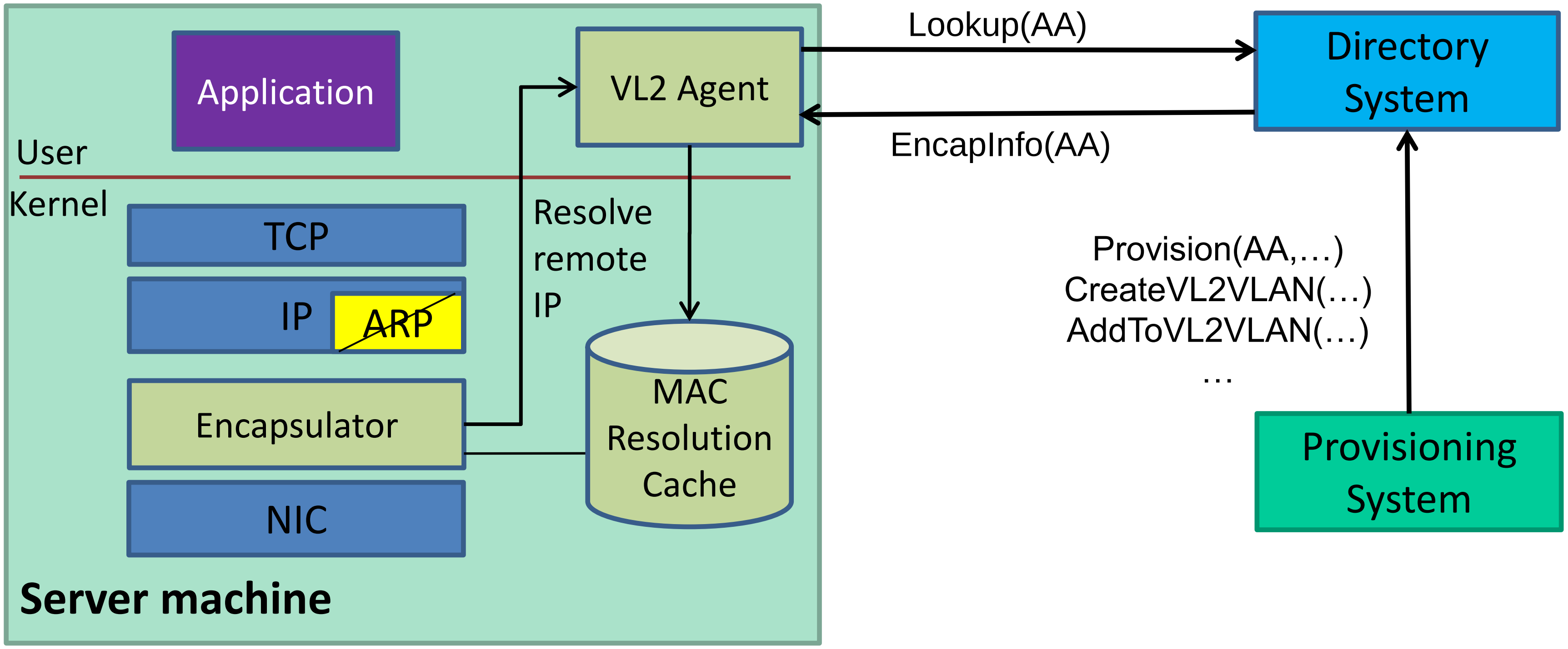
- Valiant Load Balancing
  - Every flow “bounced” off a random intermediate switch
  - Provably hotspot free for any admissible traffic matrix
  - Servers could randomize flow-lets if needed

Separating Names from Locations: How Smart Servers Use Dumb Switches



- Encapsulation used to transfer complexity to servers
  - Commodity switches have simple forwarding primitives
  - Complexity moved to computing the headers
- Many types of encapsulation available
  - IEEE 802.1ah defines MAC-in-MAC encapsulation; VLANs; etc.

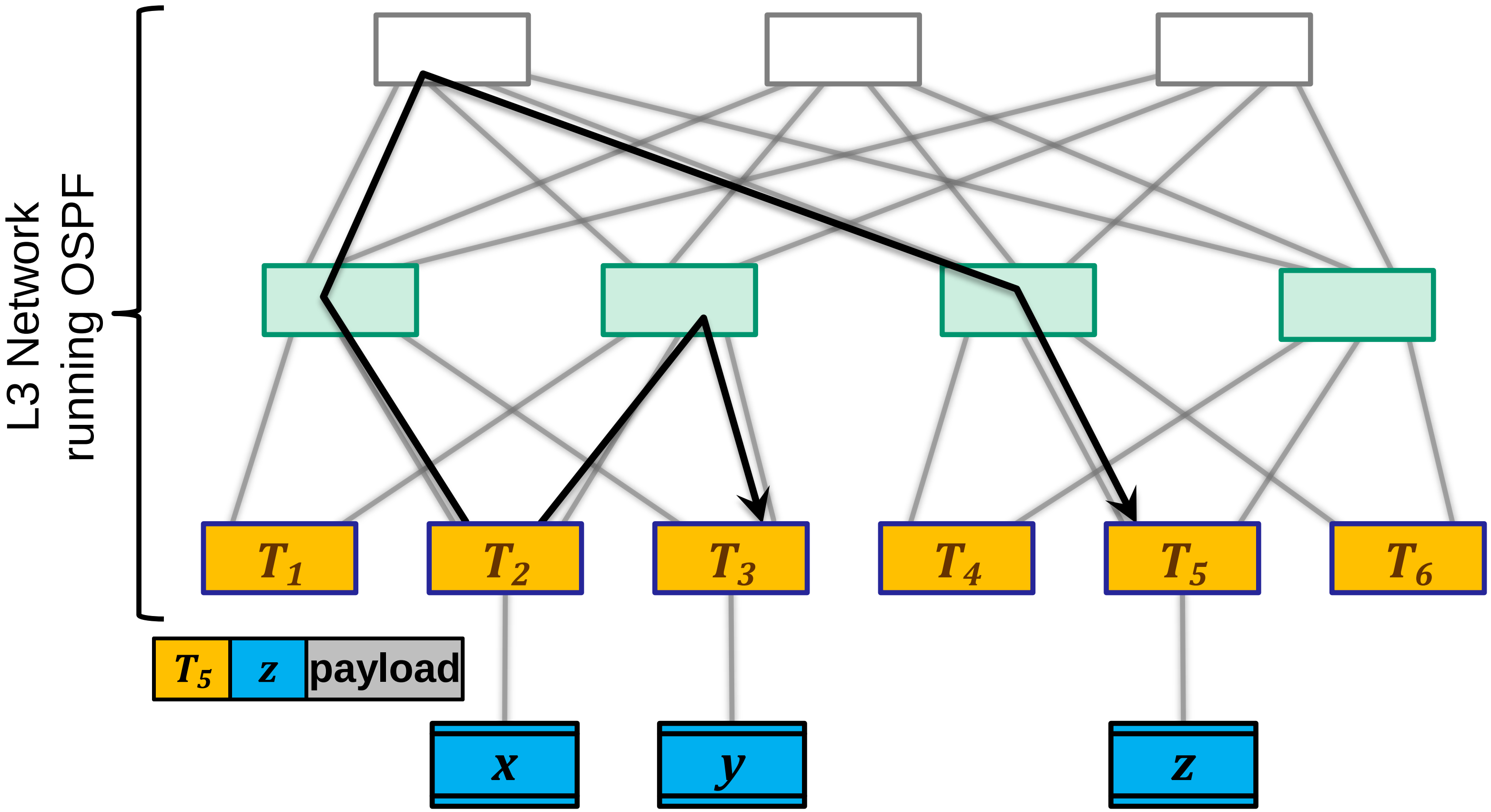
Leverage Strengths of End Systems



- Data center OSes already heavily modified for VMs, storage, etc.
  - A thin shim for network support is no big deal
- Applications work with Application Addresses
  - AA's are flat names; infrastructure addresses invisible to apps
- No change to applications or clients outside DC

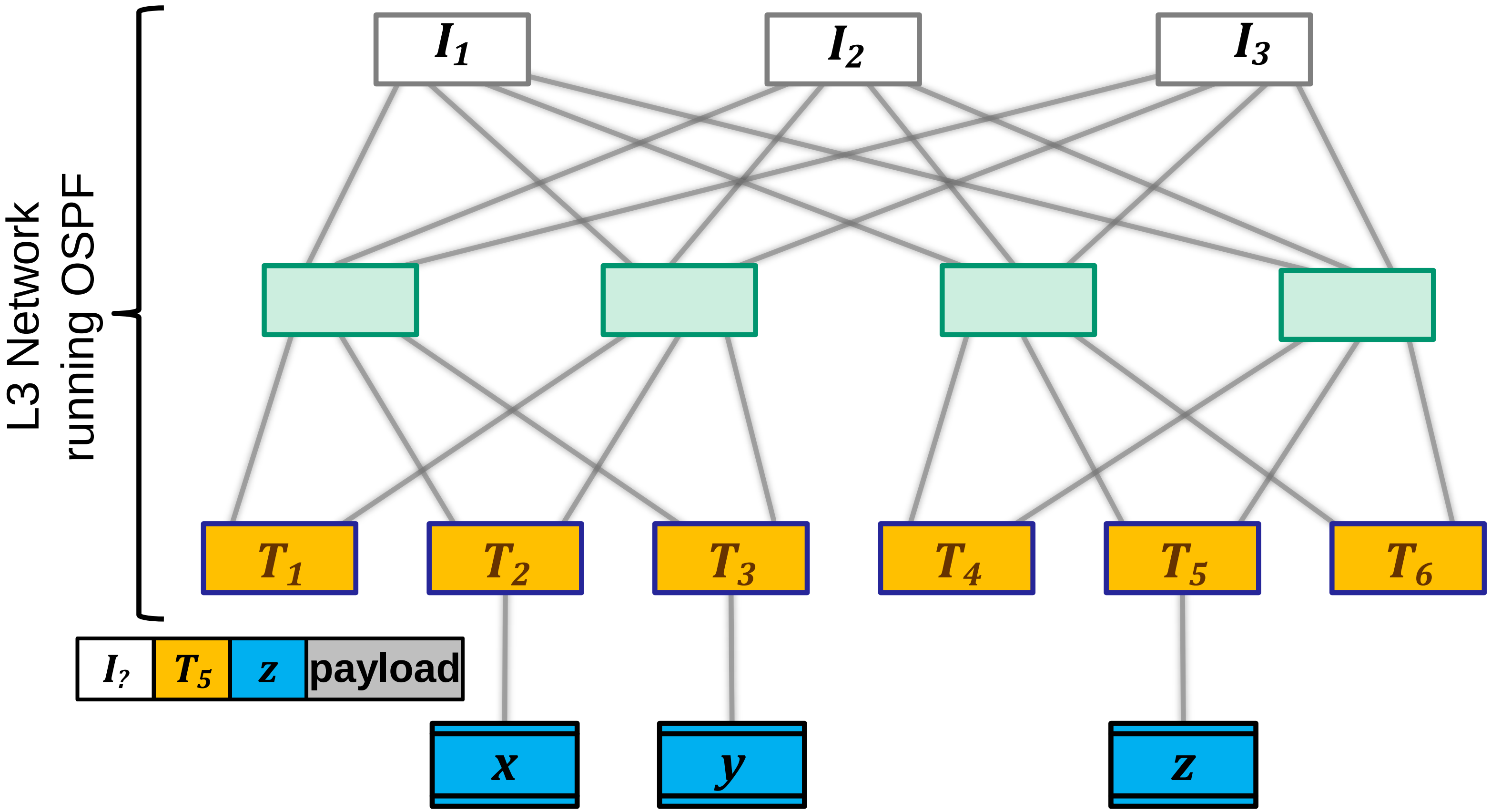
Separating Network Changes from Tenant Changes

How to implement VLB while avoiding need to update state on every host at every topology change?



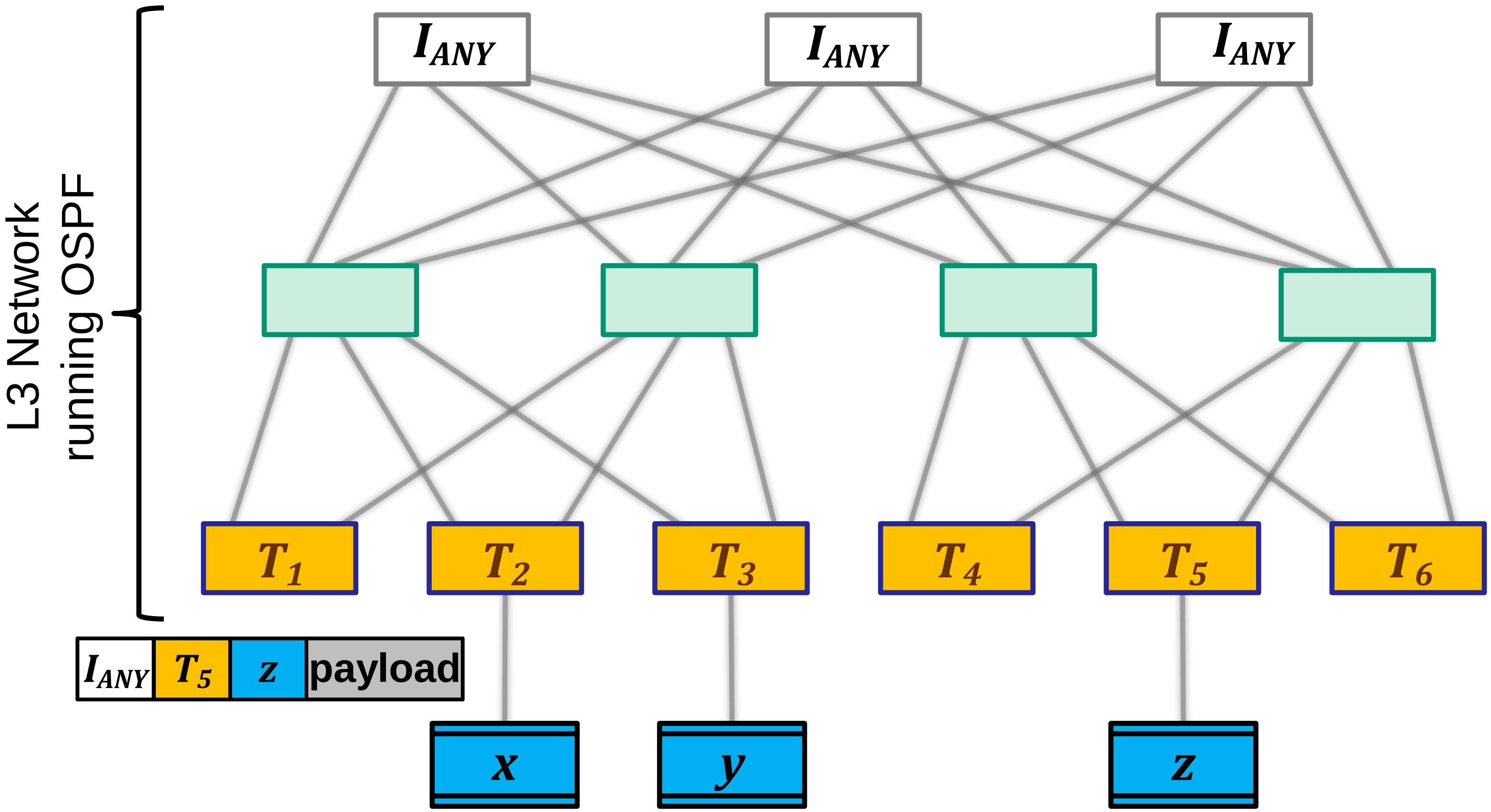
Separating Network Changes from Tenant Changes

How to implement VLB while avoiding need to update state on every host at every topology change?



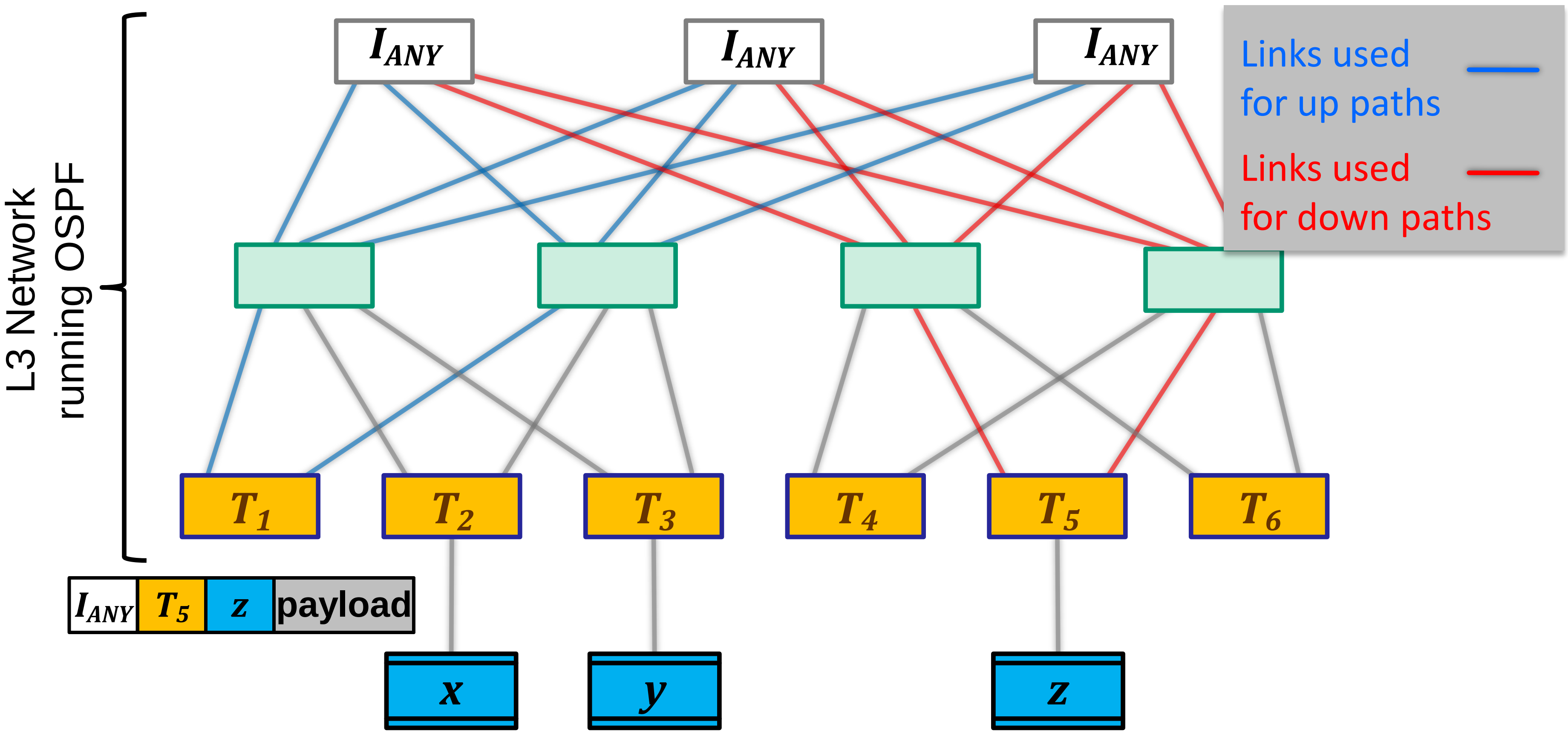
Separating Network Changes from Tenant Changes

How to implement VLB while avoiding need to update state on every host at every topology change?



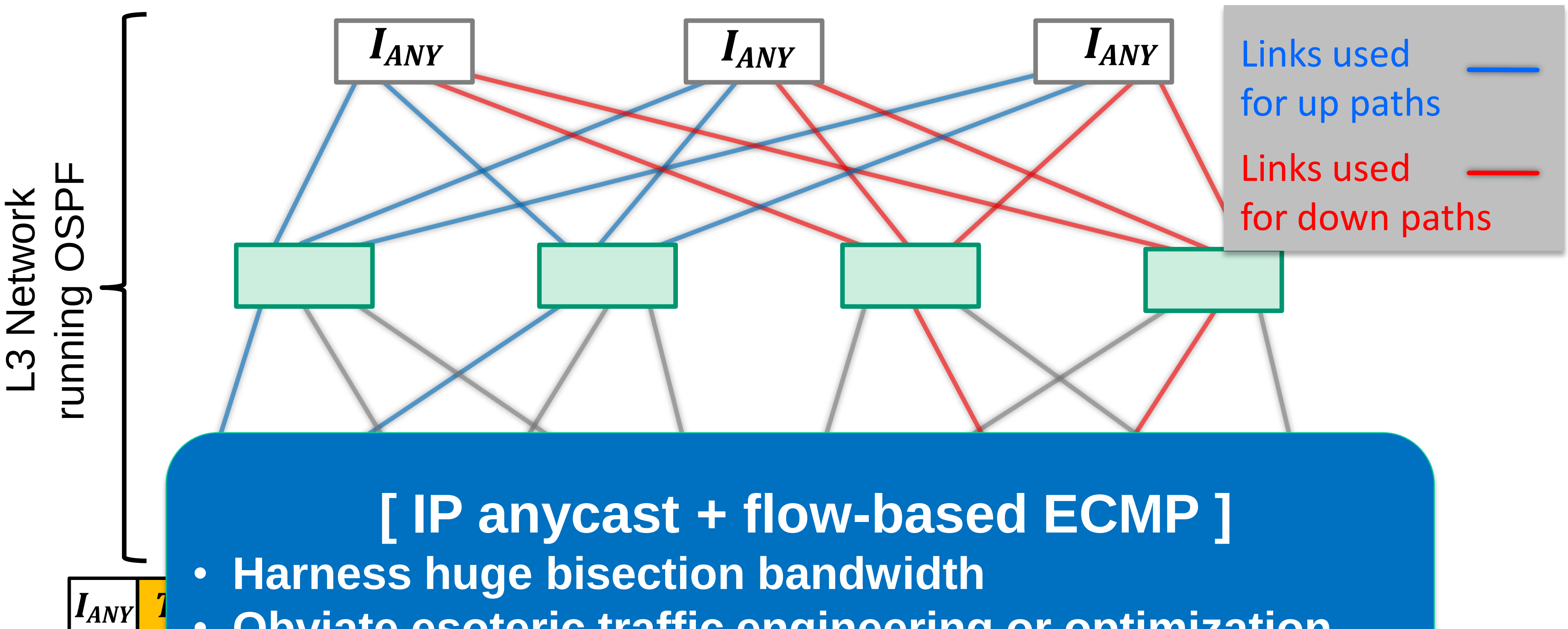
Separating Network Changes from Tenant Changes

How to implement VLB while avoiding need to update state on every host at every topology change?



Separating Network Changes from Tenant Changes

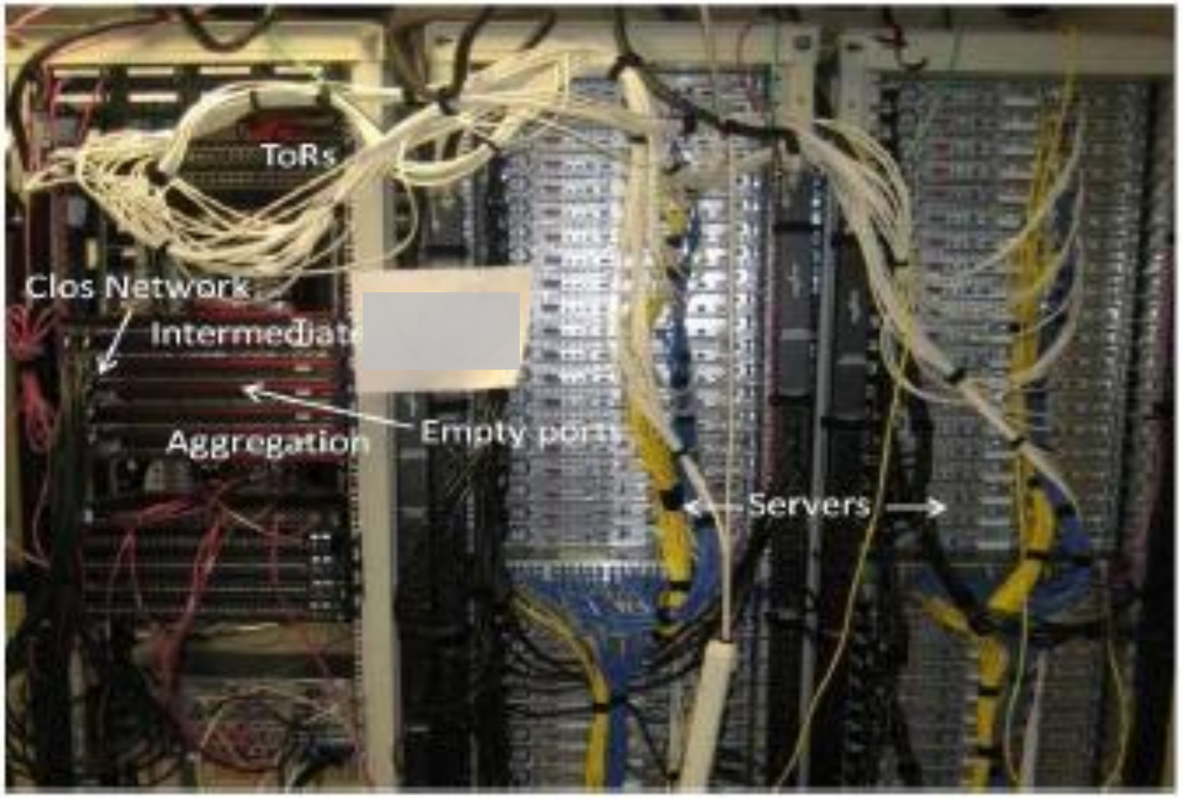
How to implement VLB while avoiding need to update state on every host at every topology change?



[IP anycast + flow-based ECMP]

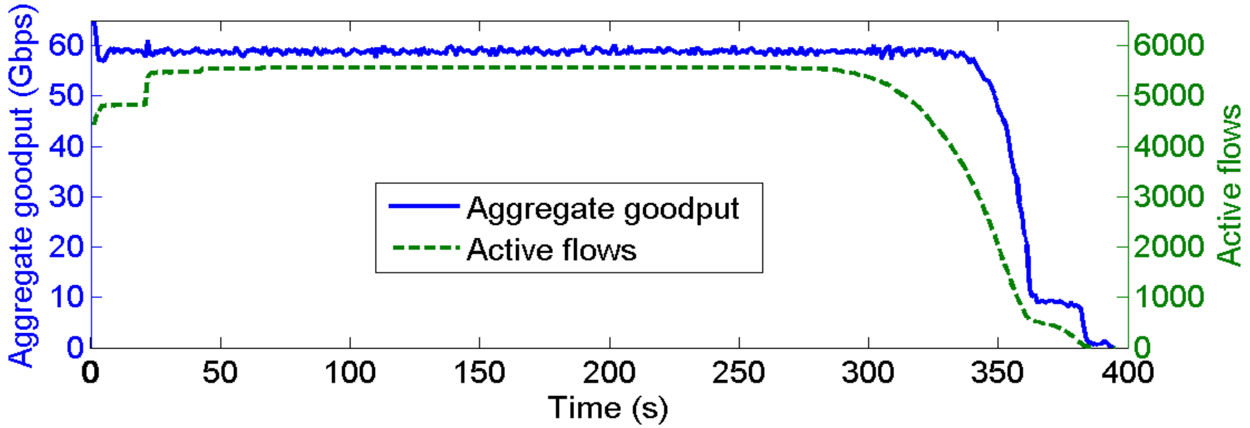
- **Harness huge bisection bandwidth**
- **Obviate esoteric traffic engineering or optimization**
- **Ensure robustness to failures**
- **Work with switch mechanisms available today**

VL2 Prototype



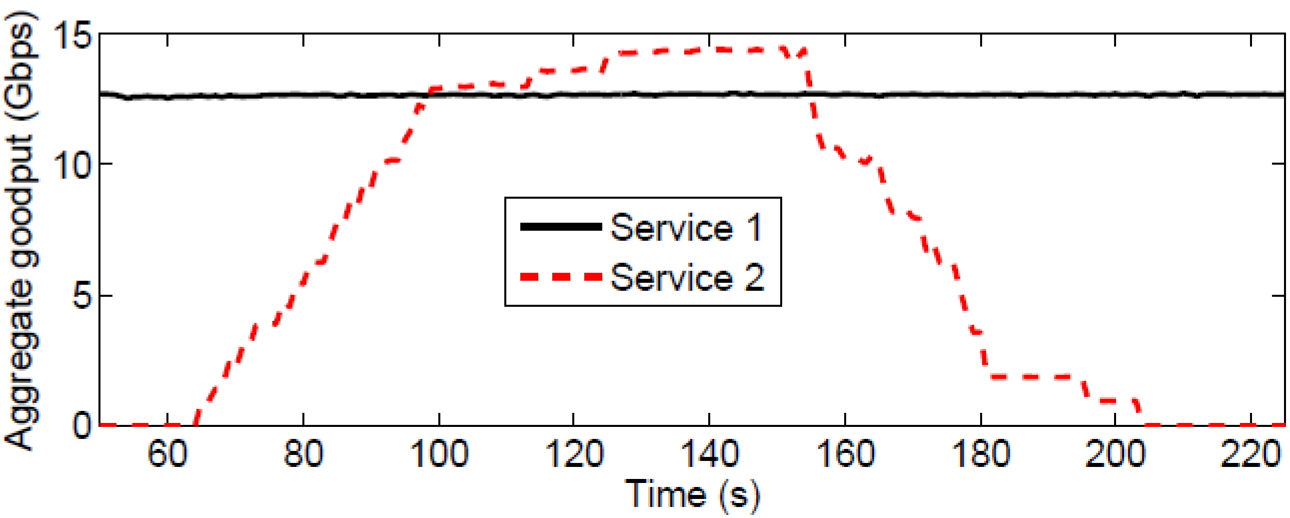
- 4 ToR switches, 3 aggregation switches, 3 intermediate switches
- Experiments conducted with 40, 80, 300 servers
  - Results have near perfect scaling
  - Gives us some confidence that design will scale-out as predicted

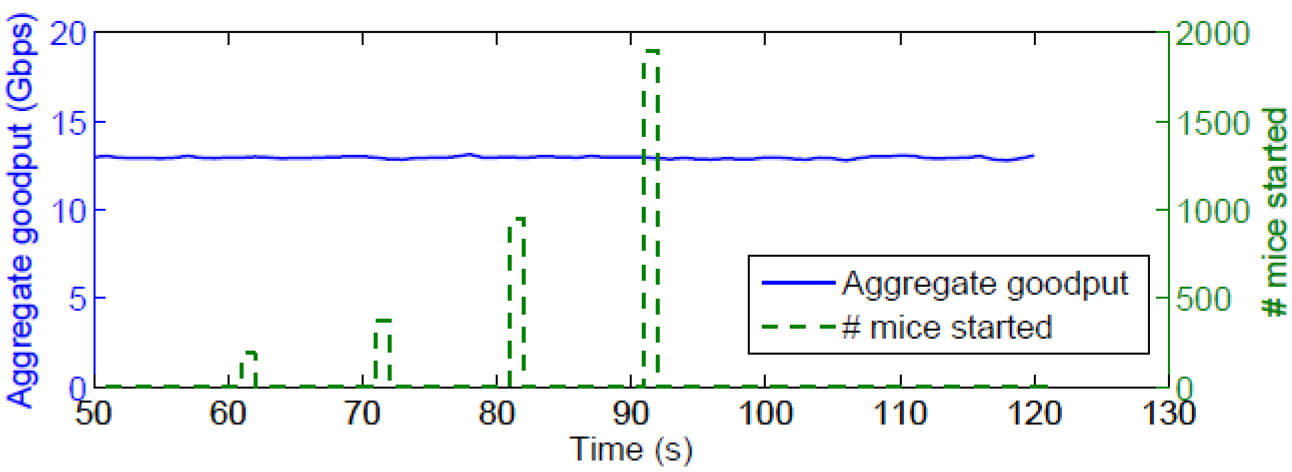
VL2 Achieves Uniform High Throughput



- Experiment: all-to-all shuffle of 500 MB among 75 servers – 2.7 TB
  - Excellent metric of overall efficiency and performance
  - All2All shuffle is superset of other traffic patterns
- Results:
  - Ave goodput: 58.6 Gbps; Fairness index: .995; Ave link util: 86%
- Perfect system-wide efficiency would yield aggregate goodput of 75G
  - Monsoon efficiency is 78% of perfect
  - 10% inefficiency due to duplexing issues; 6% header overhead
  - VL2 efficiency is <u>94% of optimal</u>

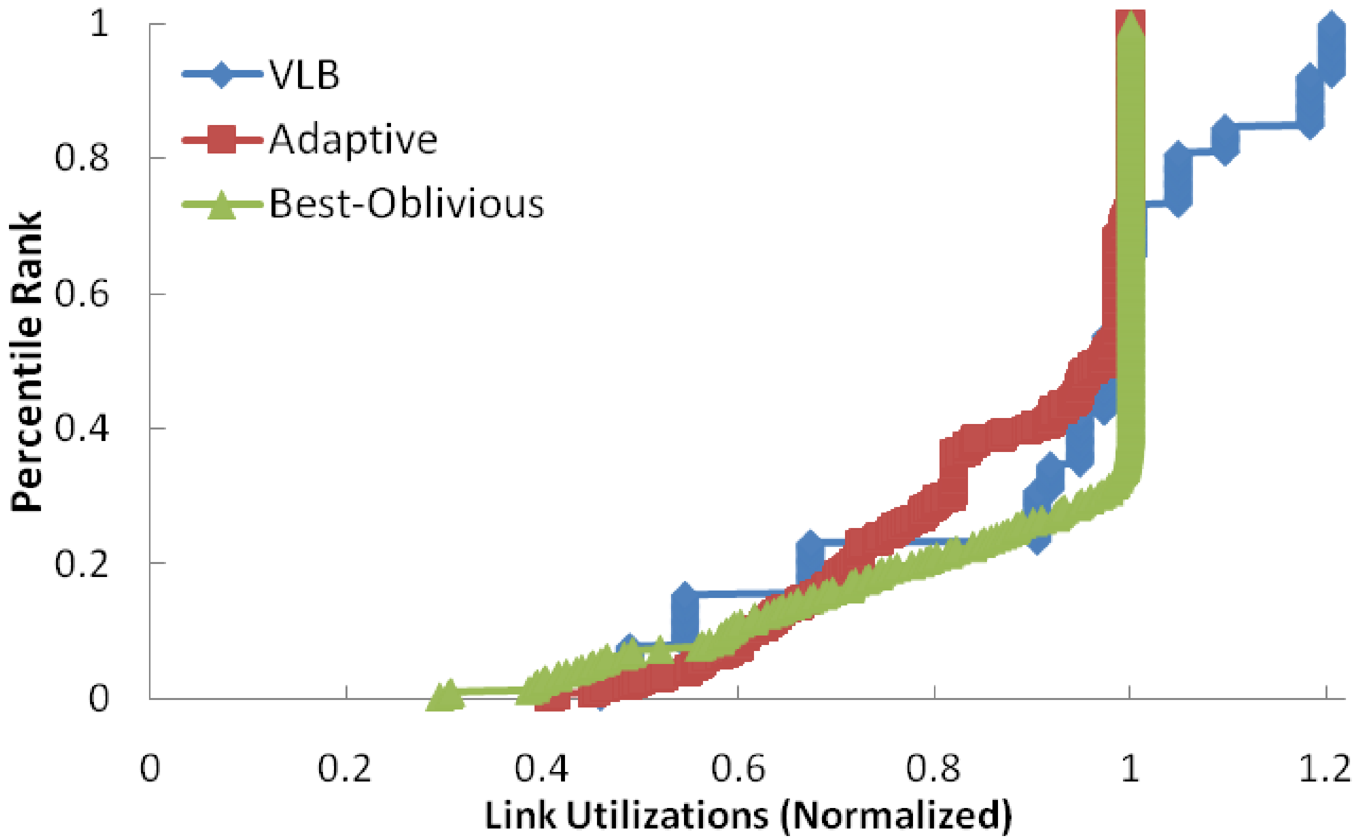
VL2 Provides Performance Isolation





- Service 1 unaffected by service 2's activity

VLB vs. Adaptive vs. Best Oblivious Routing



- VLB does as well as adaptive routing (traffic engineering using an oracle) on Data Center traffic
    - Worst link is 20% busier with VLB, median is same

Related Work

- OpenFlow
  - Shares idea of simple switches controlled by external SW
  - VL2 is a philosophy for how to use the switches
- Fat-trees, PortLand [Vahdat, et al., SIGCOMM'08,'09]
  - Shares a preference for a Clos topology
  - Monsoon provides a virtual layer 2 using different techniques: changes to servers, an existing forwarding primitive, directory service
- Dcell, BCube [Guo, et al., SIGCOMM'08,'09]
  - Uses servers themselves to forward packets
- SEATTLE [Kim, et al., SIGCOMM'08]
  - Shared goal of a large L2, different approach to directory service
- Formal network theory and HPC
  - Valiant Load Balancing, Clos networks
- Logically centralized routing
  - 4D, Tesseract, Ethane

Summary

- Key to economic data centers is agility
  - Any server, any service
  - Today, the network is the largest blocker
- The right network model to create is a virtual layer 2 per service
  - Uniform High Bandwidth☑
  - Performance Isolation☑
  - Layer 2 Semantics☑
- VL2 implements this model via several techniques
  - Randomizing to cope with volatility (VLB) ➔uniform BW/perf iso
  - Name/location separation & end system changes ➔ L2 semantics
  - End system changes & proven technology ➔ deployable now
  - Performance at 40, 80, 300 servers is excellent, and looks scalable

VL2: Any server/any service agility via scalable virtual L2 networks that eliminate fragmentation of the server pool

More Information

- **The Cost of a Cloud: Research Problems in Data Center Networks**
  - http://research.microsoft.com/~dmaltz/papers/DC-Costs-CCR-editorial.pdf
- **VL2: A Scalable and Flexible Data Center Network**
  - http://research.microsoft.com/apps/pubs/default.aspx?id=80693
- **Towards a Next Generation Data Center Architecture: Scalability and Commoditization**
  - http://research.microsoft.com/~dmaltz/papers/monsoon-presto08.pdf
- **The Nature of Datacenter Traffic: Measurements and Analysis**
  - http://research.microsoft.com/en-us/UM/people/srikanth/data/imc09_dcTraffic.pdf
- **What Goes into a Data Center?**
  - http://research.microsoft.com/apps/pubs/default.aspx?id=81782
- **James Hamilton's Perspectives Blog**
  - http://perspectives.mvdirona.com
- **Designing & Deploying Internet-Scale Services**
  - http://mvdirona.com/jrh/talksAndPapers/JamesRH_Lisa.pdf
- **Cost of Power in Large Scale Data Centers**
  - http://perspectives.mvdirona.com/2008/11/28/CostOfPowerInLargeScaleDataCenters.aspx

BACK UP SLIDES

Abstract (this won't be part of the presented slide deck – I'm just keeping the information together)

Here's an abstract and slide deck for a 30 to 45 min presentation on VL2, our data center network. I can add more details on the Monsoon design or more background on the enabling HW, the traffic patterns, etc. as desired. See http://research.microsoft.com/apps/pubs/default.aspx?id=81782 for possibilities. (we could reprise the tutorial if you'd like – it ran in 3 hours originally) We can do a demo if that would be appealing (takes about 5 min)

To be agile and cost effective, data centers must allow dynamic resource allocation across large server pools. Today, the highest barriers to achieving this agility are limitations imposed by the network, such as bandwidth bottlenecks, subnet layout, and VLAN restrictions. To overcome this challenge, we present VL2, a practical network architecture that scales to support huge data centers with 100,000 servers while providing uniform high capacity between servers, performance isolation between services, and Ethernet layer-2 semantics.

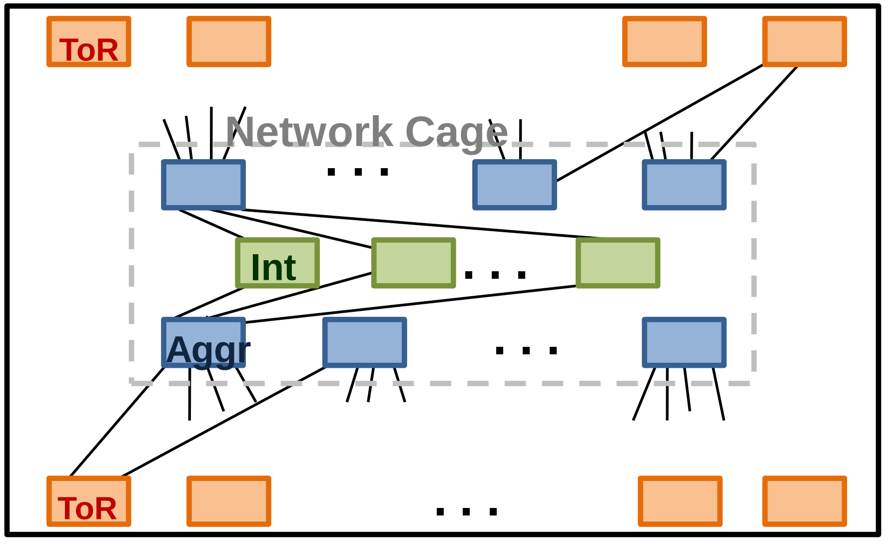
VL2 uses (1) flat addressing to allow service instances to be placed anywhere in the network, (2) Valiant Load Balancing to spread traffic uniformly across network paths, and (3) end-system based address resolution to scale to large server pools, without introducing complexity to the network control plane. VL2's design is driven by detailed measurements of traffic and fault data from a large operational cloud service provider. VL2's implementation leverages proven network technologies, already available at low cost in high-speed hardware implementations, to build a scalable and reliable network architecture. As a result, VL2 networks can be deployed today, and we have built a working prototype with 300 servers. We evaluate the merits of the VL2 design using measurement, analysis, and experiments. Our VL2 prototype shuffles 2.7 TB of data among 75 servers in 395 seconds – sustaining a rate that is 94% of the maximum possible.

Cost of a Monsoon Network

- For a typical 35K server data center
  - A monsoon network provides 1:1 oversubscription at the same cost as conventional network (with 1:240 oversubscription)
  - A conventional network with 1:1 oversub would cost 14x the cost of the monsoon network
- Conventional network costs 14-16x monsoon cost for oversubscription ratios between 1:1 and 1:23
- Monsoon networks do get cheaper if some oversubscription is acceptable
  - Monsoon network with 1:23 oversubscription is 30% the cost of a monsoon network connecting the same servers at 1:1 oversubscription

Cabling Costs and Issues

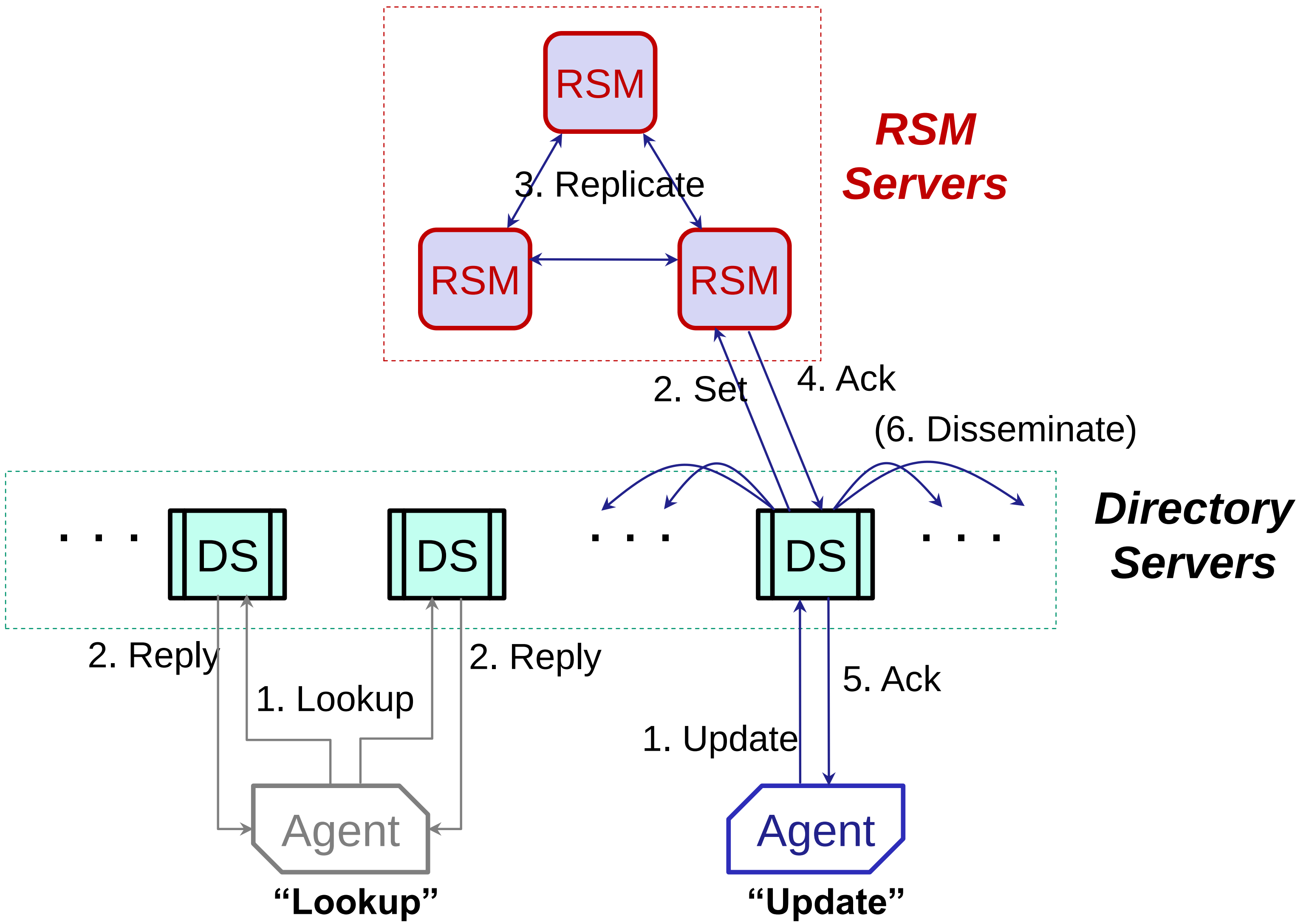
- Cabling complexity is not a big deal
  - Monsoon network cabling fits nicely into conventional open floor plan data center
  - Containerized designs available
- Cost is not a big deal
  - Computation shows it as 12% of total network cost
  - Estimate: SFP+ cable = $190, two 10G ports = $1K, cabling should be ~19% of switch cost



Directory System Performance

- Key issues:
  - Lookup latency (SLA set at 10ms)
  - How many servers needed to handle a DC’s lookup traffic?
  - Update latency
  - Convergence latency

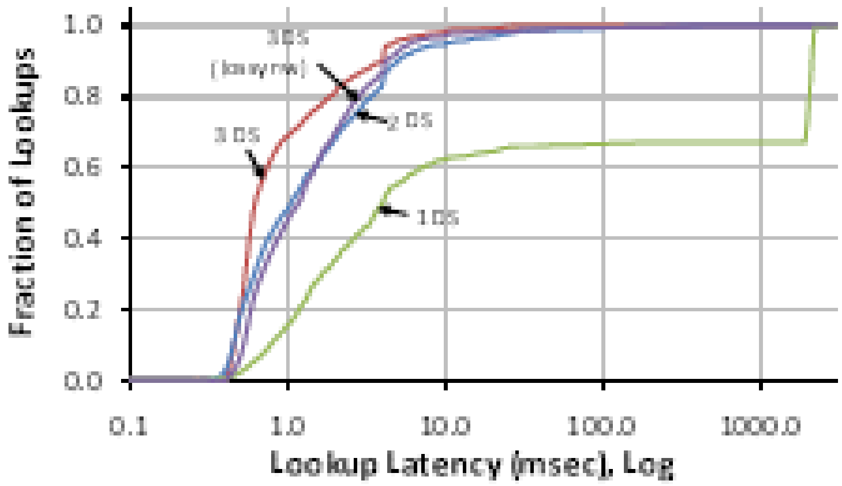
Directory System

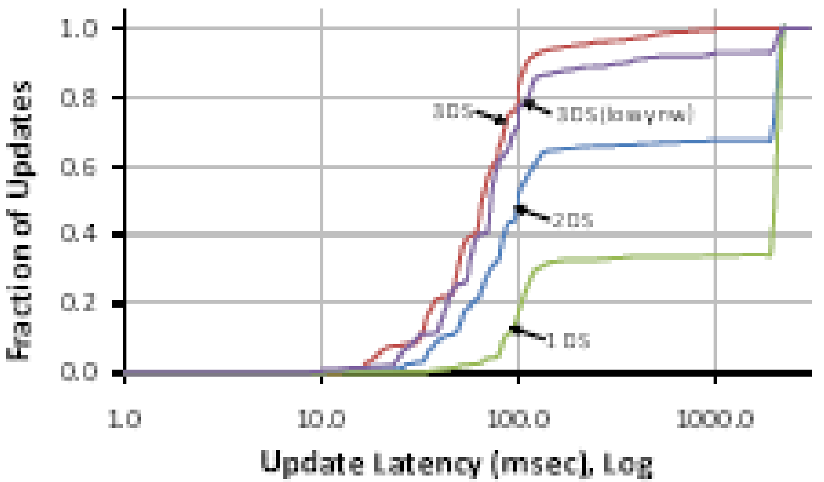


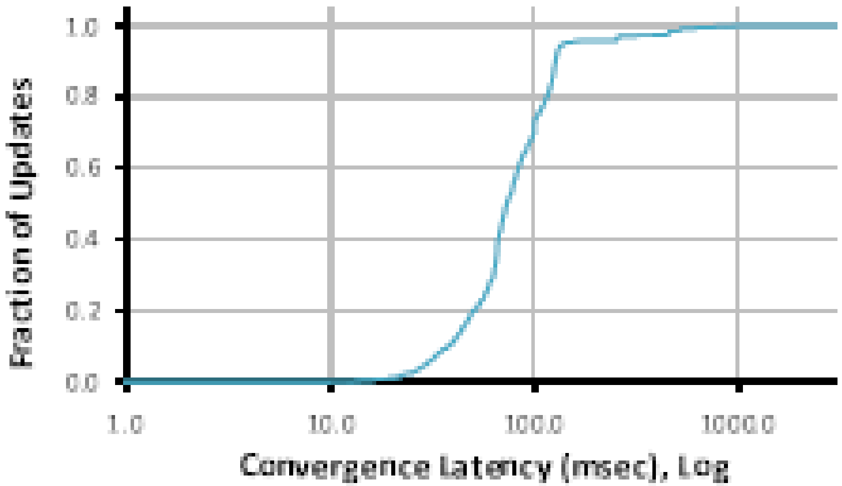
Directory System Performance

- Lookup latency
  - Each server assigned to the directory system can handle 17K lookups/sec with 99^{th} percentile latency < 10ms
  - Scaling is linear as expected (verified with 3,5,7 directory servers)
- Directory System sizing
  - How many lookups per second?
    - Median node has 10 connections, 100K servers = 1M entries
    - Assume (worst case?) that all need to be refreshed at once
  - 64 servers handles the load w/i 10ms SLA
  - Directory system consumes 0.06% of total servers

Directory System Performance

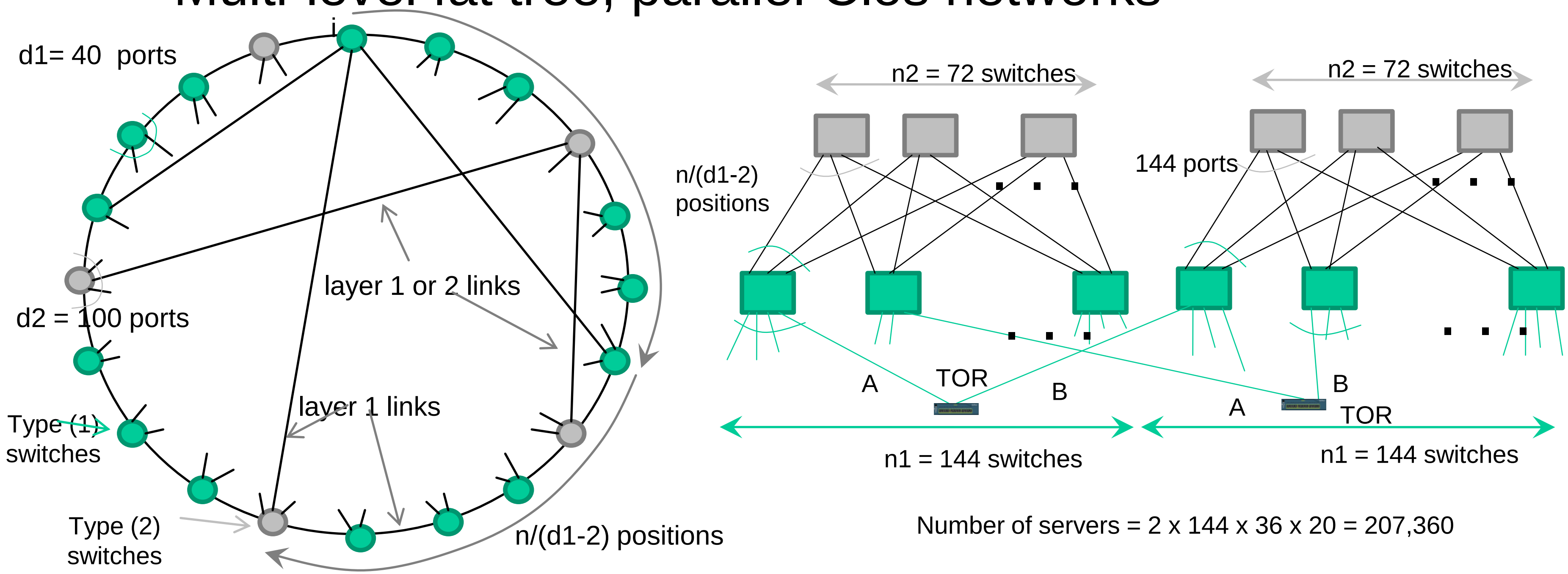






The Topology Isn't the Most Important Thing

- Two-layer Clos network seems optimal for our current environment, but …
- Other topologies can be used with Monsoon
  - Ring/Chord topology makes organic growth easier
  - Multi-level fat tree, parallel Clos networks



Number of servers = 2 x 144 x 36 x 20 = 207,360

Server Costs

Ugly secret: 10% to 30% utilization considered “good” in DCs

Causes:

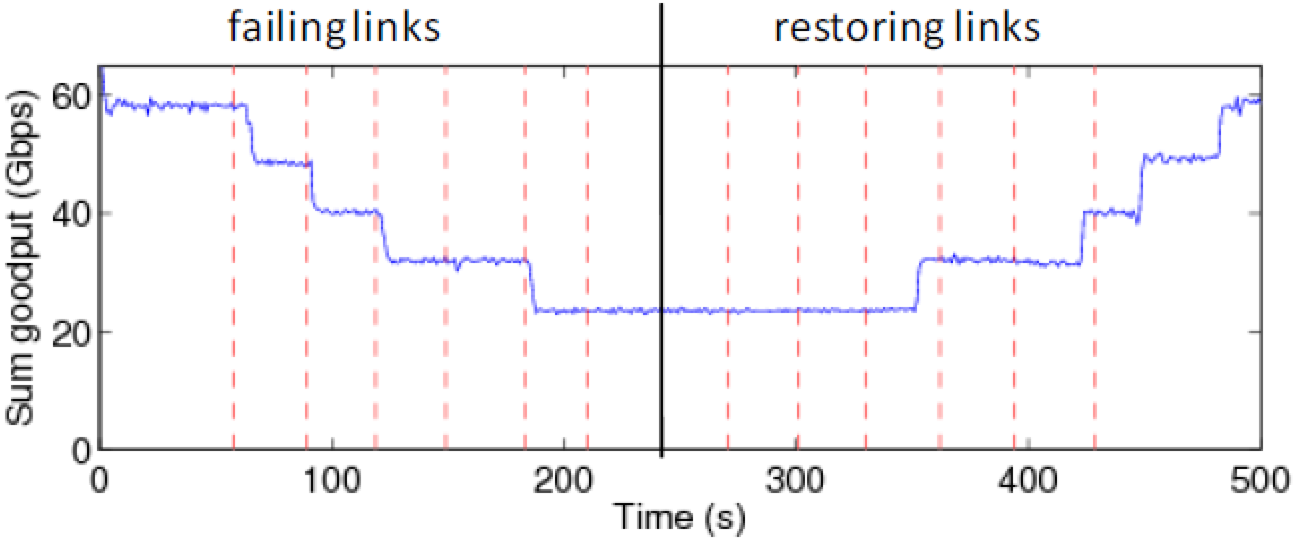
- Uneven application fit:
  - Each server has CPU, memory, disk: most applications exhaust one resource, stranding the others
- Long provisioning timescales:
  - New servers purchased quarterly at best
- Uncertainty in demand:
  - Demand for a new service can spike quickly
- Risk management:
  - Not having spare servers to meet demand brings failure just when success is at hand

If each service buys its own servers, the natural response is hoarding

Improving Server ROI: Need Agility

- Turn the servers into a single large fungible pool
  - Let services “breathe” : dynamically expand and contract their footprint as needed
- Requirements for implementing agility
  - Means for rapidly installing a service’s code on a server
    - *Virtual machines, disk images* ☑
  - Means for a server to access persistent data
    - Data too large to copy during provisioning process
    - *Distributed filesystems (e.g., blob stores)* ☑
  - Means for communicating with other servers, regardless of where they are in the data center
    - *Network* ☐

VL2 is resilient to link failures



- Performance degrades and recovers gracefully as links are failed and restored