A Bing-Inspired View of Data Center Networking

David A. Maltz
and many collaborators
Azure/Autopilot Network Team
6/18/2013

What's a Cloud Service Data Center?

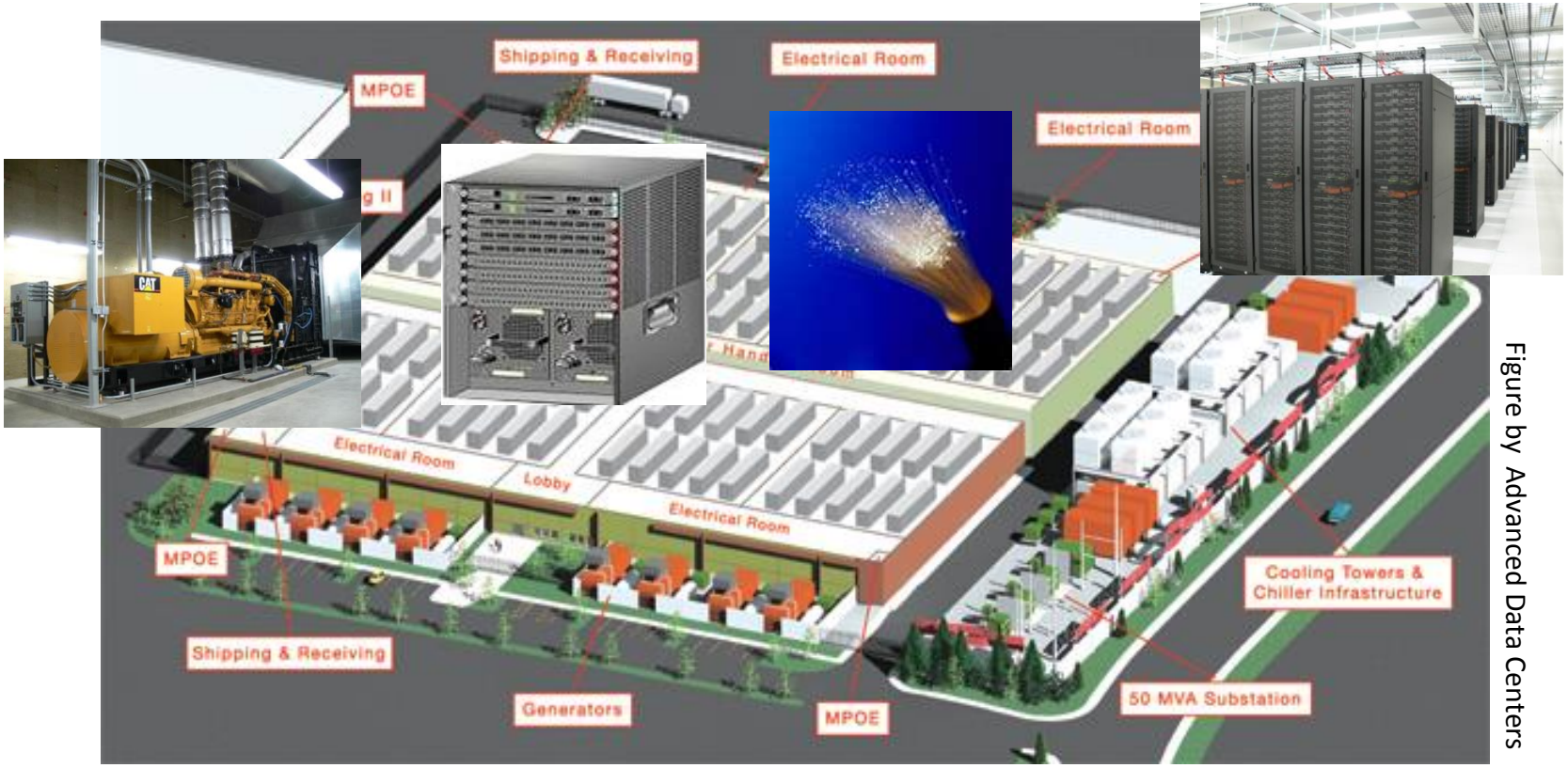


Figure by Advanced Data Centers

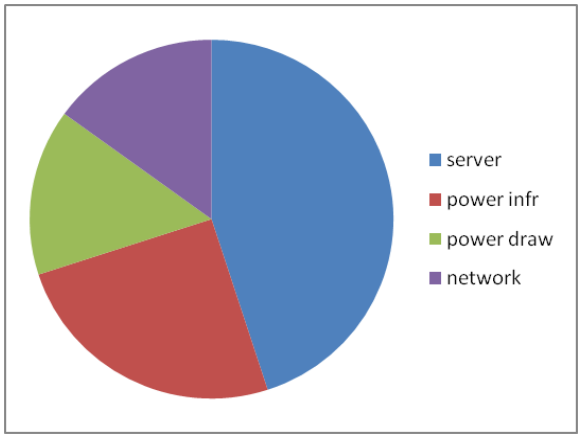
- Electrical power and economies of scale determine total data center size: 50,000 – 200,000 servers today
- Servers divided up among 1000s-100Ks of different services
- Scale-out is paramount: some services have 10s of servers, some have 10s of 1000s

Data Center Costs

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

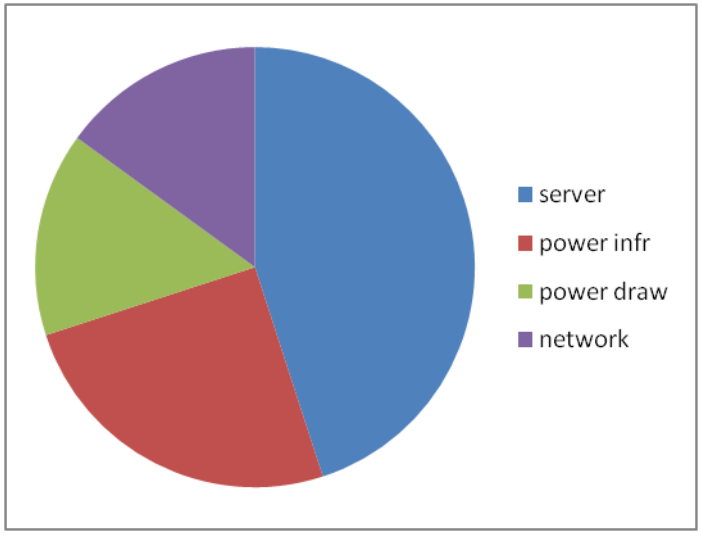
*3 yr amortization for servers, 15 yr for infrastructure; 5% cost of money

- Remarkably stable over time (sad)
  - Server costs dominate
  - Network costs significant
- Using only public numbers
  - Server = $2000/36 months = $55
  - Power draw = 240W @ $0.07KWhr = $12



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

Amdahl's Law: Always Good For Focusing the Mind



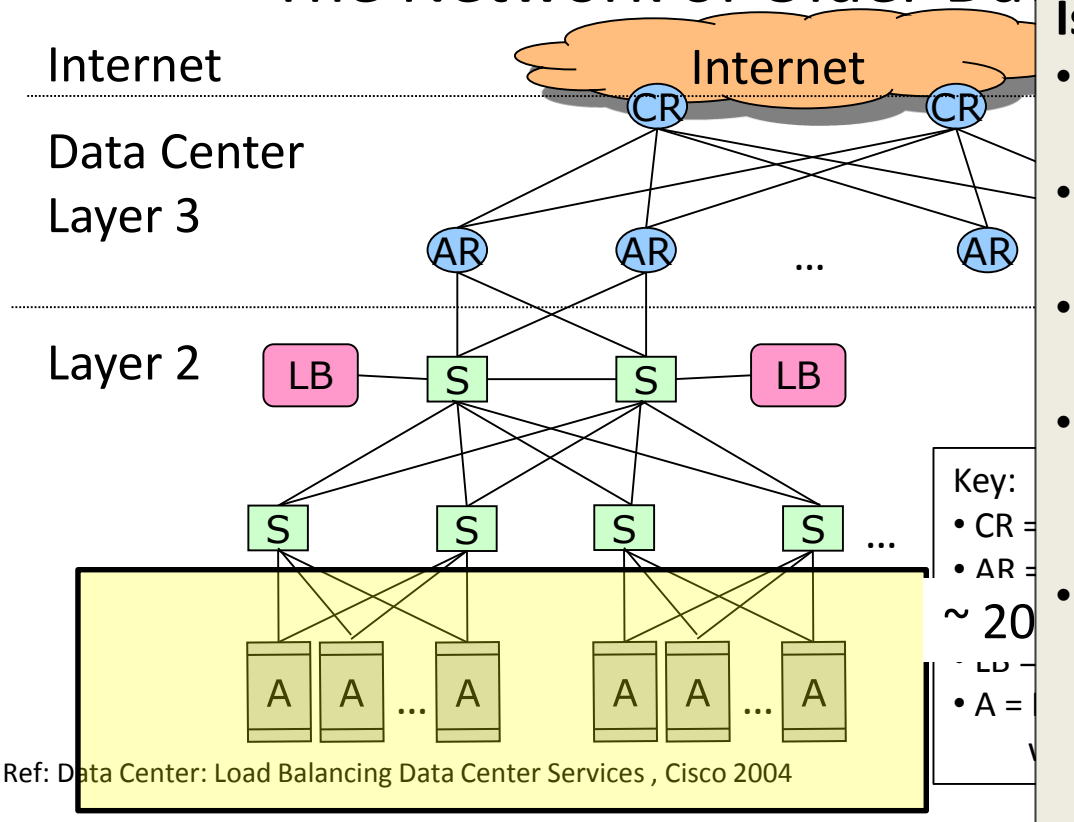
- Turn off servers?
  - Dude – I paid good money for those!
- Reduce network switch usage?
  - 48 servers/rack, 240W/server
  - 1 ToR switch/rack, 135W/switch
- Reduce server usage?
  - Big wins possible – eats into both utility draw and power infra
  - Caveat: power doesn't stat mux well – an overage is an outage
- Reduce wastage?
  - Green is good!

Power, Prices, and Contracts

"Big data centers have long term power contracts, so no savings possible" ← Not necessarily true…

- Contracts written by actuaries
  - Who have to out-predict the weather
  - They need to add a margin for profitability

The Network of Older Data Centers

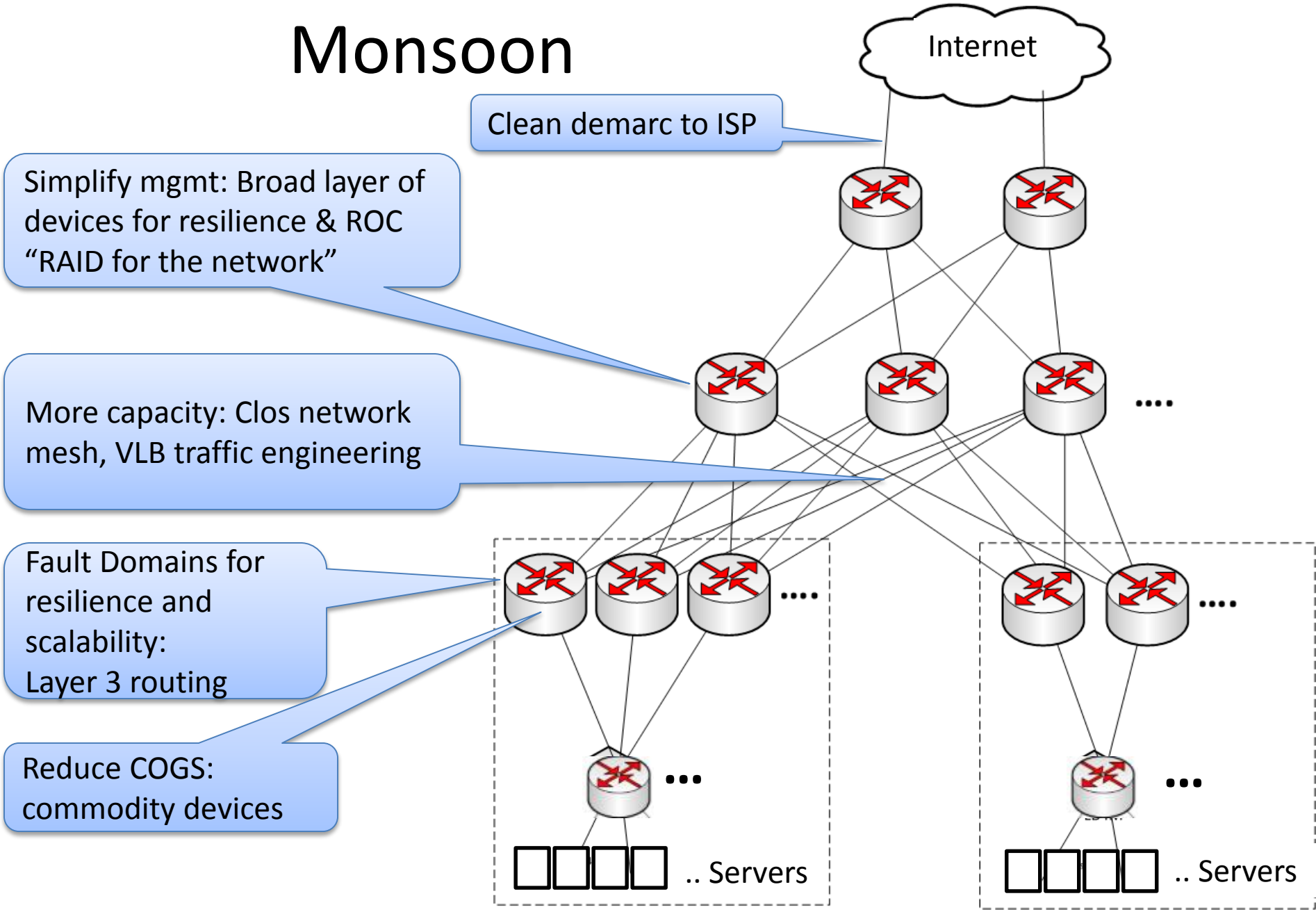


Issues:

- **Large Layer 2 regions**
  ➔ router CPU overwhelmed
- **Poor fault isolation**
  ➔20K servers at risk
- **Insufficient redundancy**
  ➔ mgmt is hard
- **Insufficient capacity**
  ➔wastes server resources
  ➔no performance isolation
- **Multiple complex bandwidth domains**
  ➔machine allocation impossibly painful&inefficient

- 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

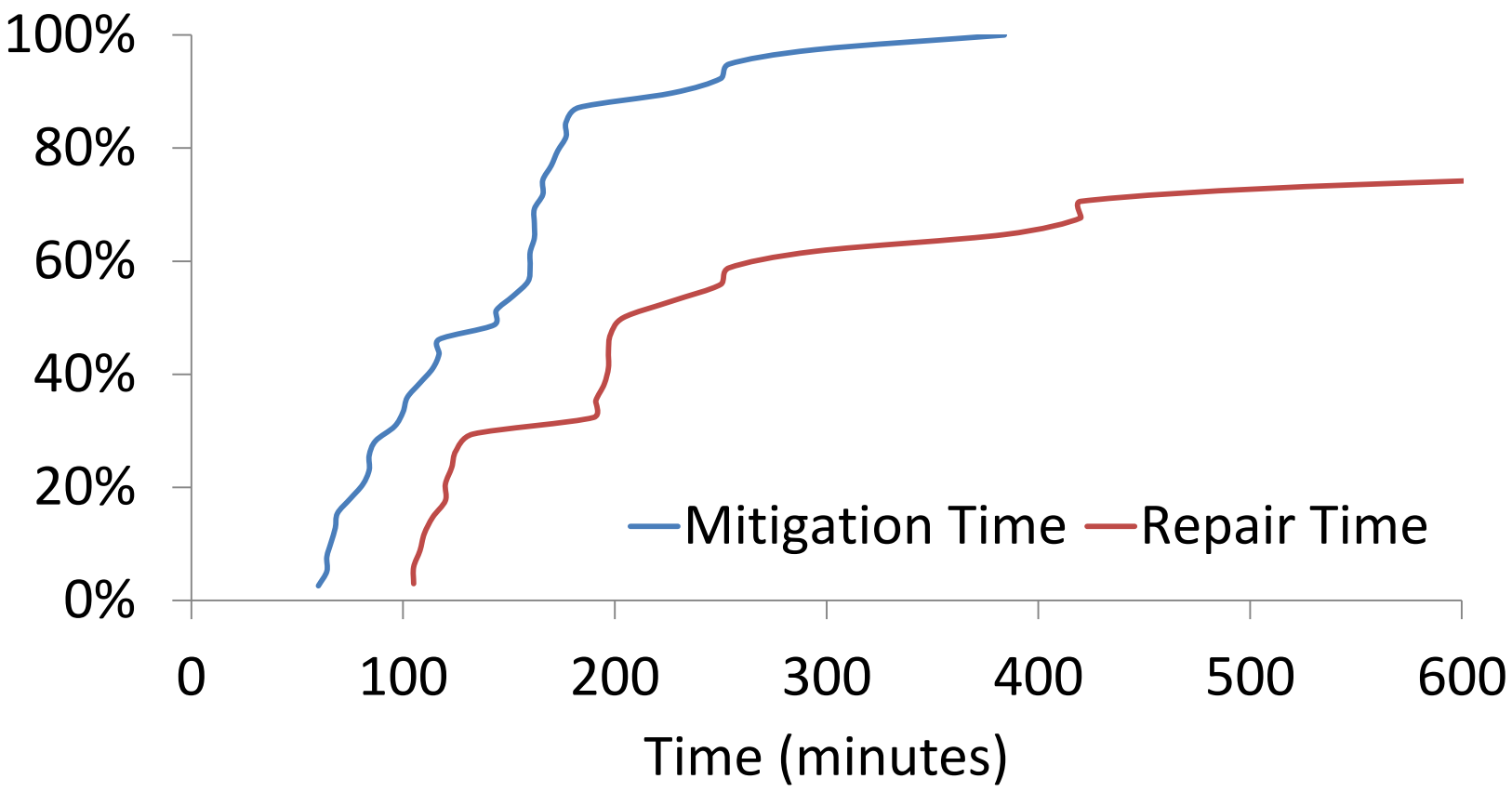
Target Physical Architecture: Monsoon



More Routers ➔ Good!
More Routers ➔ More Failures ➔ Bad!

- Network failures are a headache for AP team
  - Impact multiple servers across the data center
  - Require quick mitigation – network team has to drop what it's doing and jump to the problem
  - Extremely difficult to pinpoint the root cause – grey failures cause *some* packets to get impacted *some* times
- Current approach
  - Manually investigate and mitigate
  - Fix the root cause offline
  - Network team needs someone on call 7*24
  - Hours, days, weeks of diagnosis

Network failure recovery is slow



Network failures are difficult to handle

Category	Detection	Mitigation	Repair	%
Configuration 38%	Connectivity problem	Deactivate switch	Update configuration	38%
Layer 2 Software 21%	Layer 2 loop	Deactivate port	Update firmware	14%
	Broadcast storm	Deactivate port		5%
	Imbalance-triggered overload	Restart switch		2%
Hardware 18%	Bit corruption error	Deactivate port	Replace cable	13%
	Unstable power	Deactivate switch	Repair power	5%
Layer 3 Software/ Unknown 23%	Switch stops forwarding	Restart switch	Update firmware	9%
	Imbalance-triggered overload	Restart switch		7%
	Lost configuration	Restart switch		5%
	High CPU utilization	Restart switch		2%

FIXED by automated configuration management

Data from NetPilot, SIGCOMM2012. 82 Incidents selected from a 6 month period.

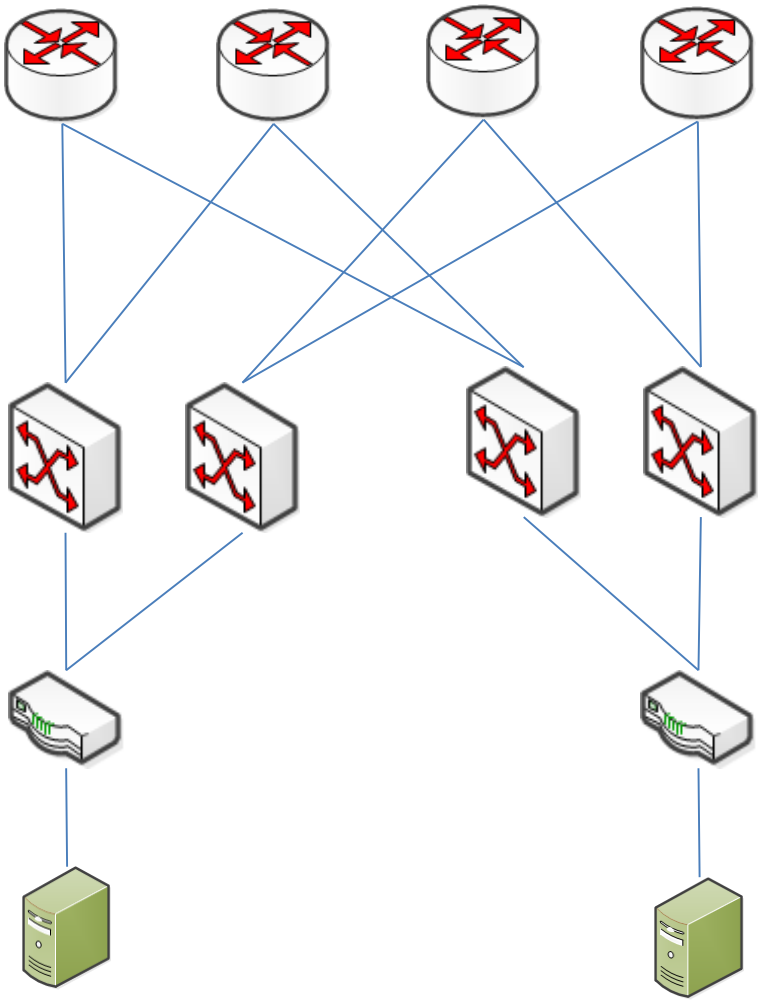
RAID for the Network: Enabling Automatic Network Issue Mitigation

- Autopilot shows Recovery Oriented Computing (ROC) works great for servers and services
  - Watchdogs continuously assess health of service
  - Repair workflow initiated on errors

- Can we bring automated ROC to the network?
  - ROC requires that reboots aren't fatal
  - The new network architecture supports this! (Netpilot, SIGCOMM2012)
- Still need a robust way to determine when a switch or service isn't healthy

Multi-path routing

- Tens to hundreds of equivalent paths between any server pair
- ECMP for load-balanced routing
- Question: how to identify faulty/flaky links/paths?



Fault Domains

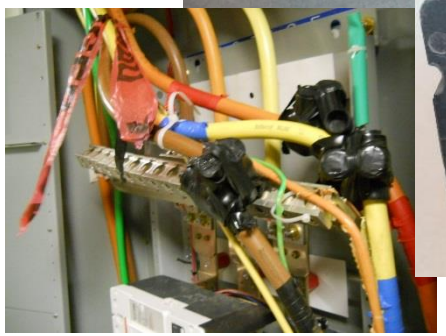
- A whole data center is a lot to lose
  - **Goal: keep failures/overload/problems localized**
- Many stressors seen in the wild
  - Bugs in protocol implementations
  - Byzantine behavior
  - L2 convergence issues
  - Table space exhaustion in commodity devices
- Control plane stressors are the worst
  - But also “mundane issues:” power, smoke, flood

Fault Domains: Solutions/Challenges

- Today's solution: L3 routing
  - Using BGP as the IGP with success
    - (Lapukhov, et al., NANOG55)
    - BGP well tested for inter-op
    - Still: lots of code, hence lots of bugs…
  - Minimizing L2 spans was big win
    - Hypothesis: it's auto-discovery from no assumed knowledge that's the issue
- Tomorrow's solutions: Direct-Control SDN
  - Logically centralized controller publishes route information to all switches
  - *Will failure modes be better or worse?*

Physical Redundancy Mechanisms Will Fail You

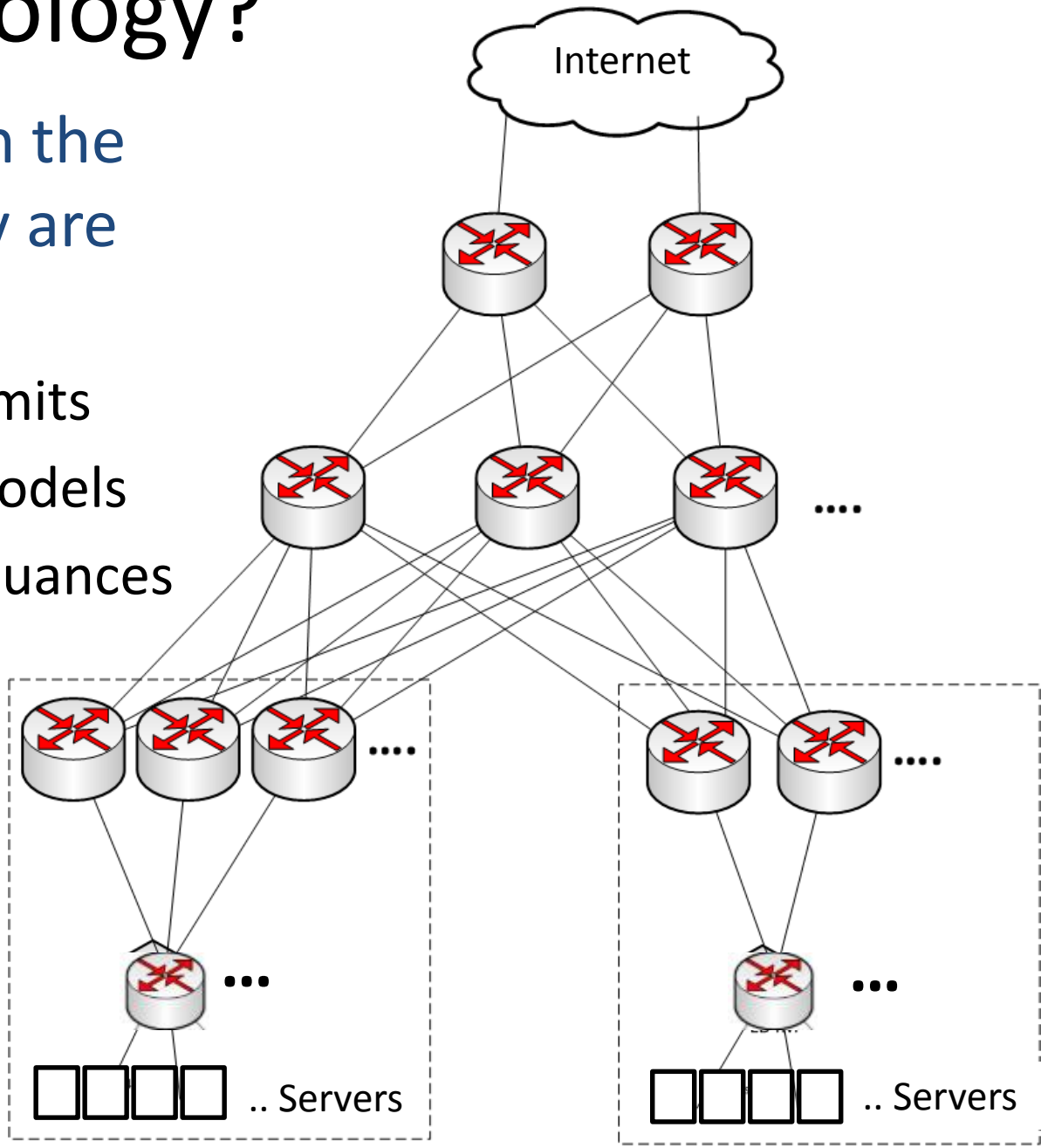
- Data centers are large, dangerous data factories
  - Industrial accidents happen
- 99.999% available power doesn't exist (often promised)
- Software failover and data replication are the only ways to build reliable services
  - Spread servers out
  - A challenge to network capacity and server placement



Exact Topology?

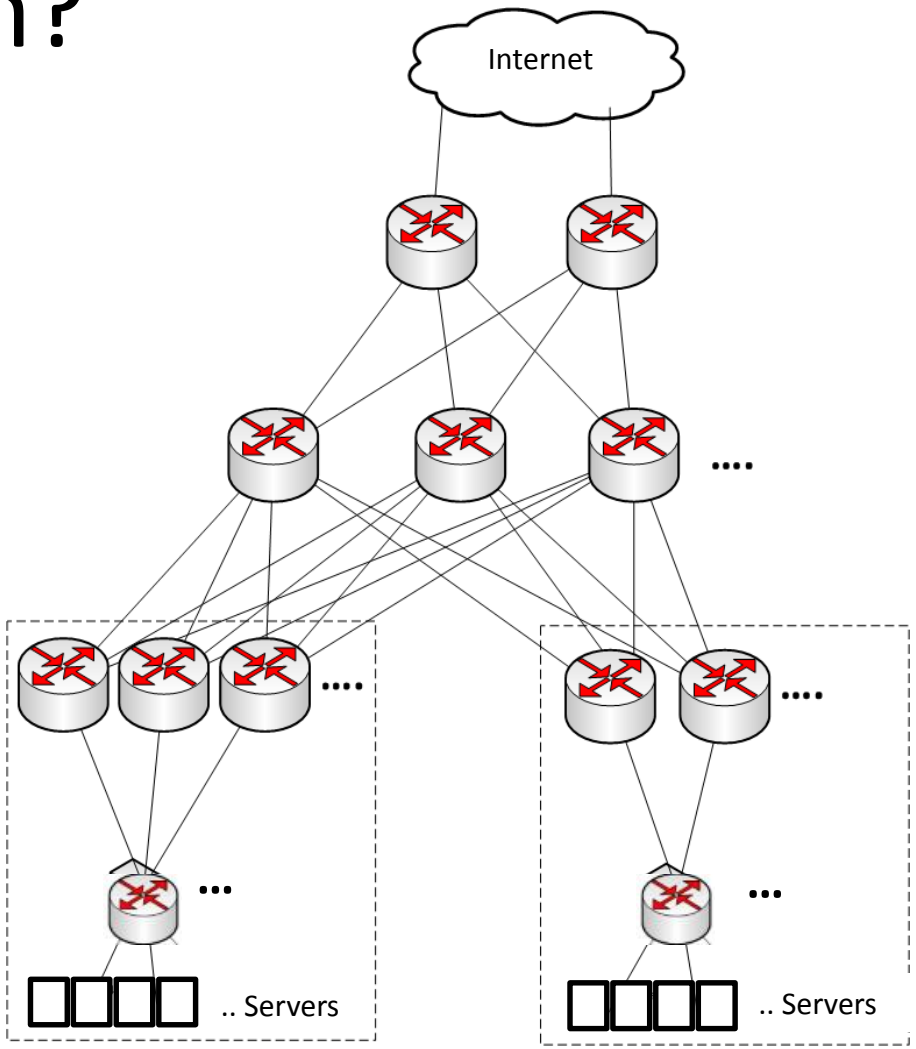
- Many variations on the Monsoon topology are possible
  - Different scaling limits
  - Different failure models
  - Different routing nuances

Have built several



Oversubscription?

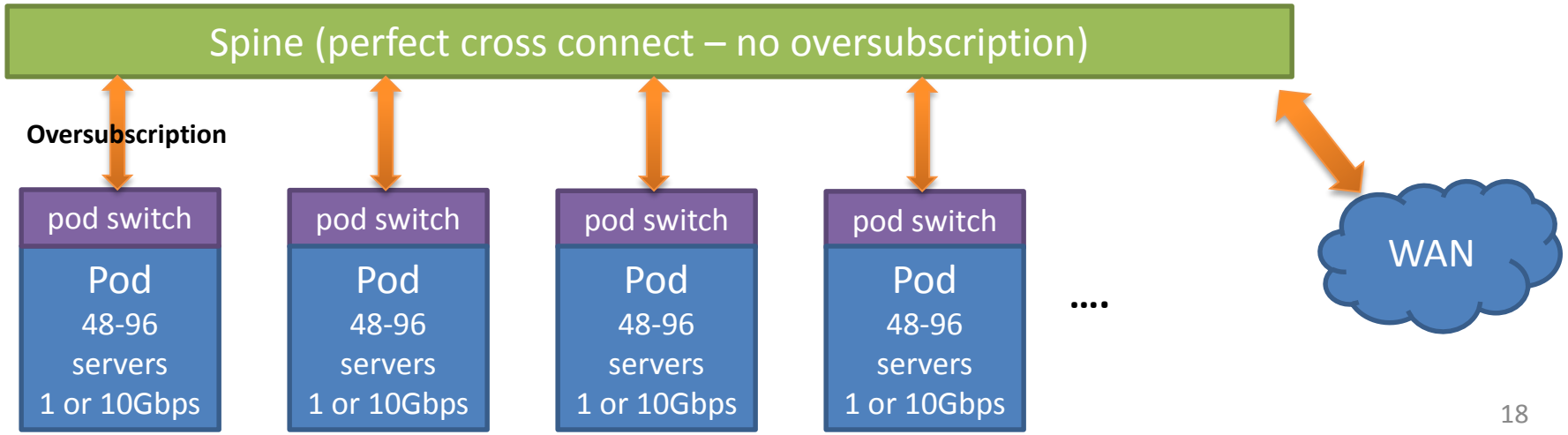
- Contentious point:
  - Oversubscription at all?
  - Where to put it?
- Higher in network
  - More complex BW domains
  - Good for apps with traffic locality
- Lower in network
  - Simpler BW domains
  - Servers more fungible
  - Easier to place applications



Modeling the Network

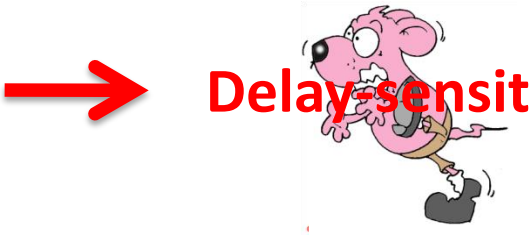
- Personal goal: move to a simple network capacity model
  - Full bandwidth inside pod
  - Oversubscription at the pod-to-spine connection
  - *All pods in the data center are network equivalent*

    ➔Defragments pool of servers, eases allocation



Workload is a mix of Traffic Types

- Partition/Aggregate
  (Query)



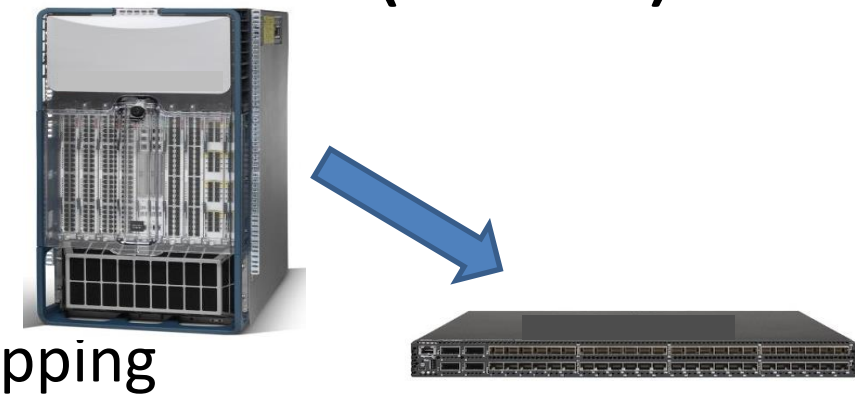
- Short messages [50KB-1MB]
  (Coordination, Control state)

- Large flows [10MB-1GB]
  (Data update)

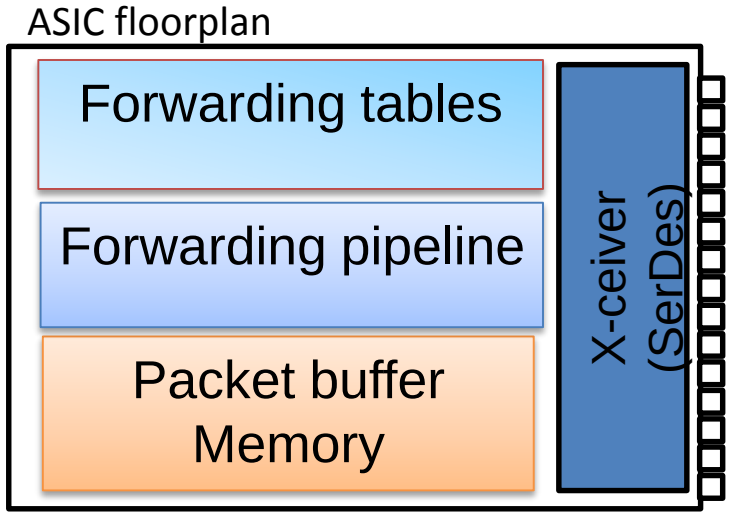
- Each server participates in all three

Drawings by: Mohammad Alizadeh

Reducing Cost of Goods Sold (COGS)



- Commodity switching
  - Driven by cheap ASICS
  - Switches $50/10Gbps and dropping
- Merchant silicon providers get more transistors, spend them on (in decreasing order)
  - More ports, more speed (SERDES)
  - Buffer space, FWDing tables
  - Data plane pipeline
- Results in switches that have:
  - Small TCAMs, fewer microseconds of buffer per port
  - Lots of ports at high speed

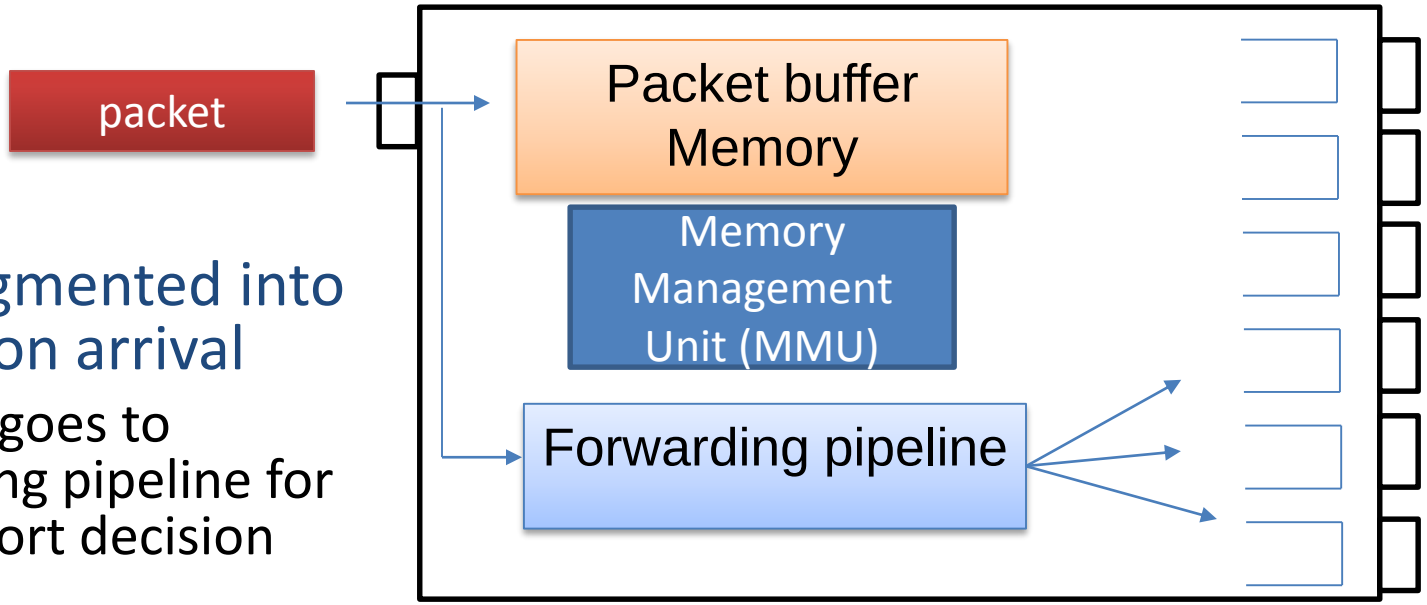


Example ASIC Generations

Year	Chip	Ports	Buffer size	Fwding Entries
1999	Firebolt-II	48x1G, 4x10G	2MB	16K
2010	Scorpion	24x106	2MB	16K
2011	Trident	64x10G	3-4MB	16K
2013	Trident2	128x10G	9MB	16K?

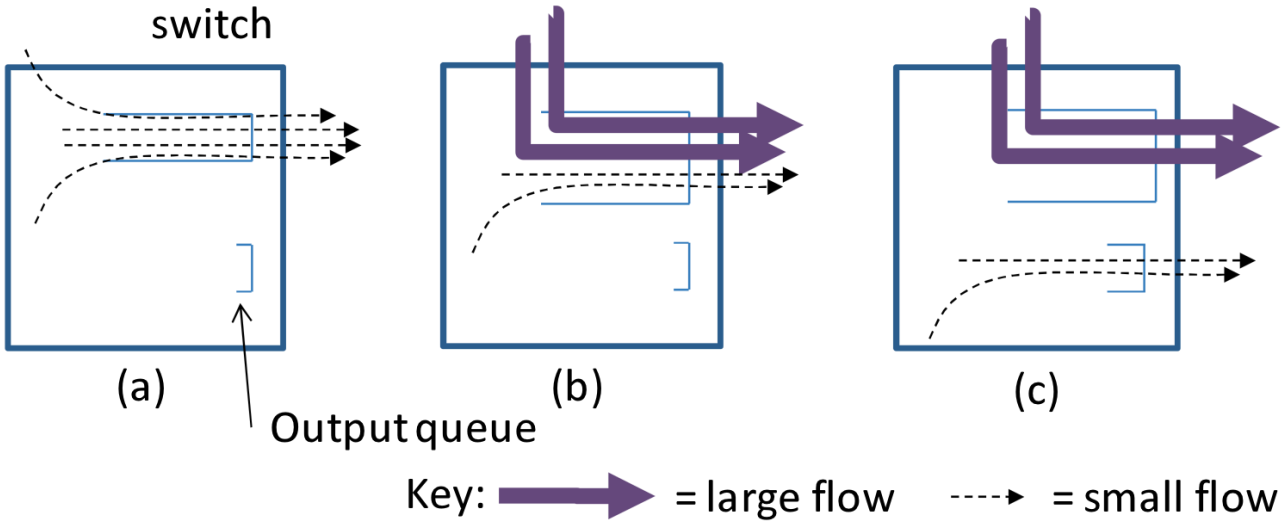
Apologies if this table has errors.
Written out from memory.

Most Cheap Switches Use Shared Memory



- Packets segmented into 128B cells on arrival
  - First cell goes to forwarding pipeline for output port decision
- MMU decides whether to allocate cells to the rest of the packet
  - *Dynamically* sets the max length of each output queue
  - Goals: handle bursts w/o dropping pkts, preserve inter-port fairness
  - Many knobs (see patents by Broadcom for examples)
  - Includes current queue lengths, QoS

Impairments Due to Shared Memory



(a) Incast: many (typically small) synchronized flows collide at an interface.

- Bursty packet drops, TCP timeouts (RTO_{min} = 300 ms).

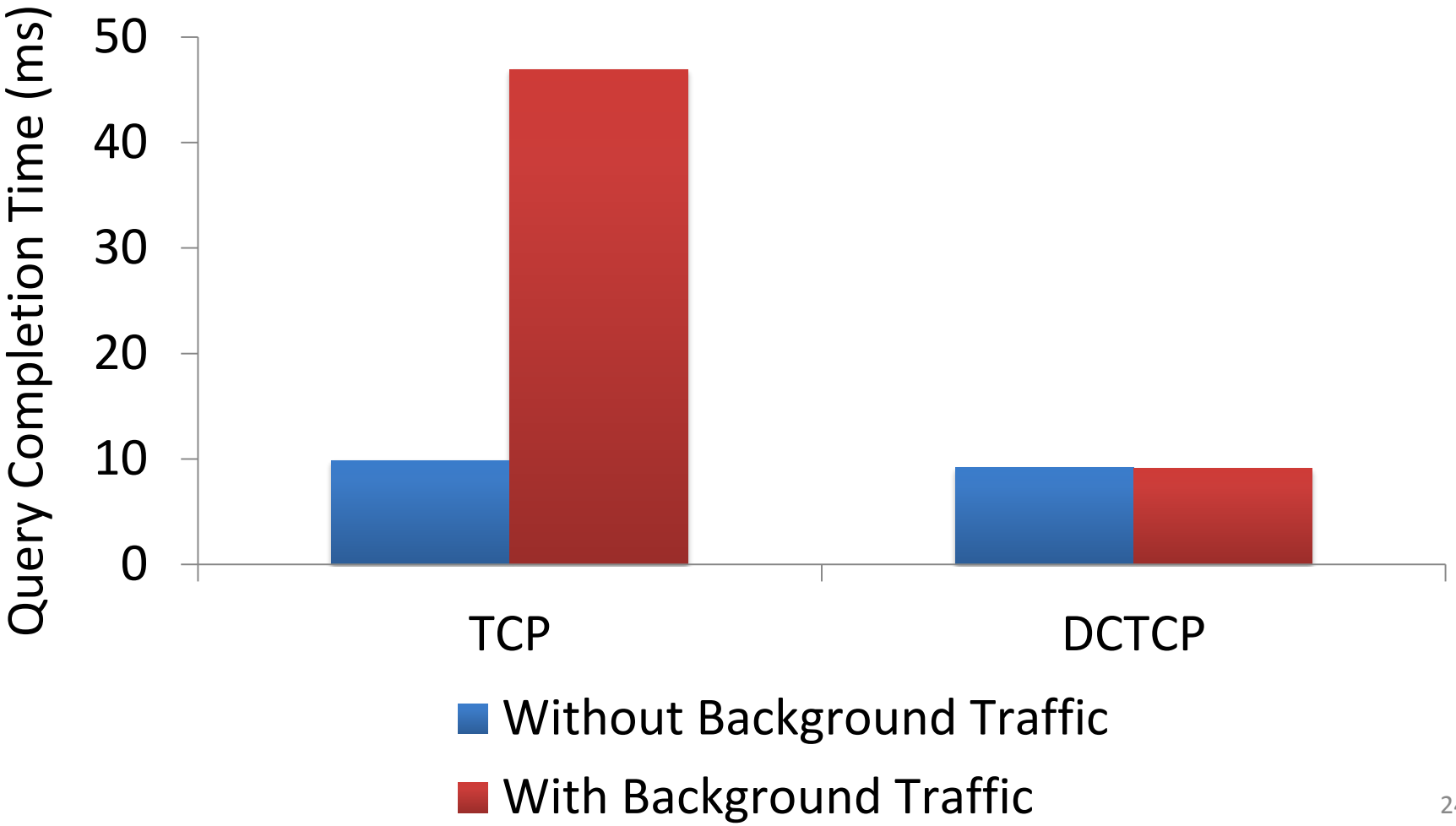
(b) Queue Buildup: Long TCP flows buildup queues.

- No buffer room left for transient bursts (as in Incast).
- Increased latency for short flows.

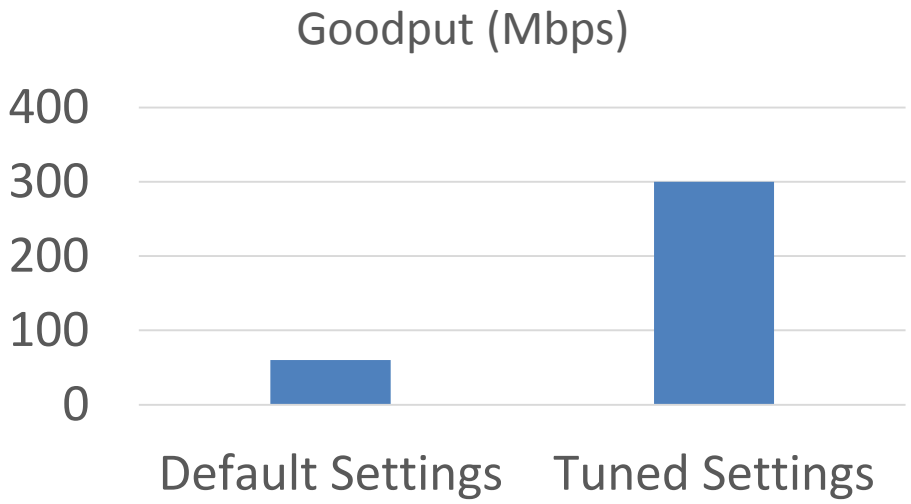
(c) Buffer Pressure: Short flows on one port are hurt by long flows on other ports due to shared memory buffer space.

Effects on Applications are Significant: Buffer Pressure

- 1 Rack: 10-to-1 Incast, Background traffic between other 30 servers.



MMU Settings are Significant




```
driv "s dyncelllimit.ge,hg,xe
DYNCELLSETLIMIT=16383
DYNCELLRESETLIMIT=15563"

driv "s totaldyncelllimit setlimit=16383"

driv "s totaldyncellresetlimit
resetlimit=15563"
```

- Experiment: 70 ms RTT, backlogged TCP
- The importance of queue management is not a new problem
  - Less explored: MMU algorithms, parameter setting, etc
- Lots of room to make big impacts
  - Few roadblocks for researchers to make progress

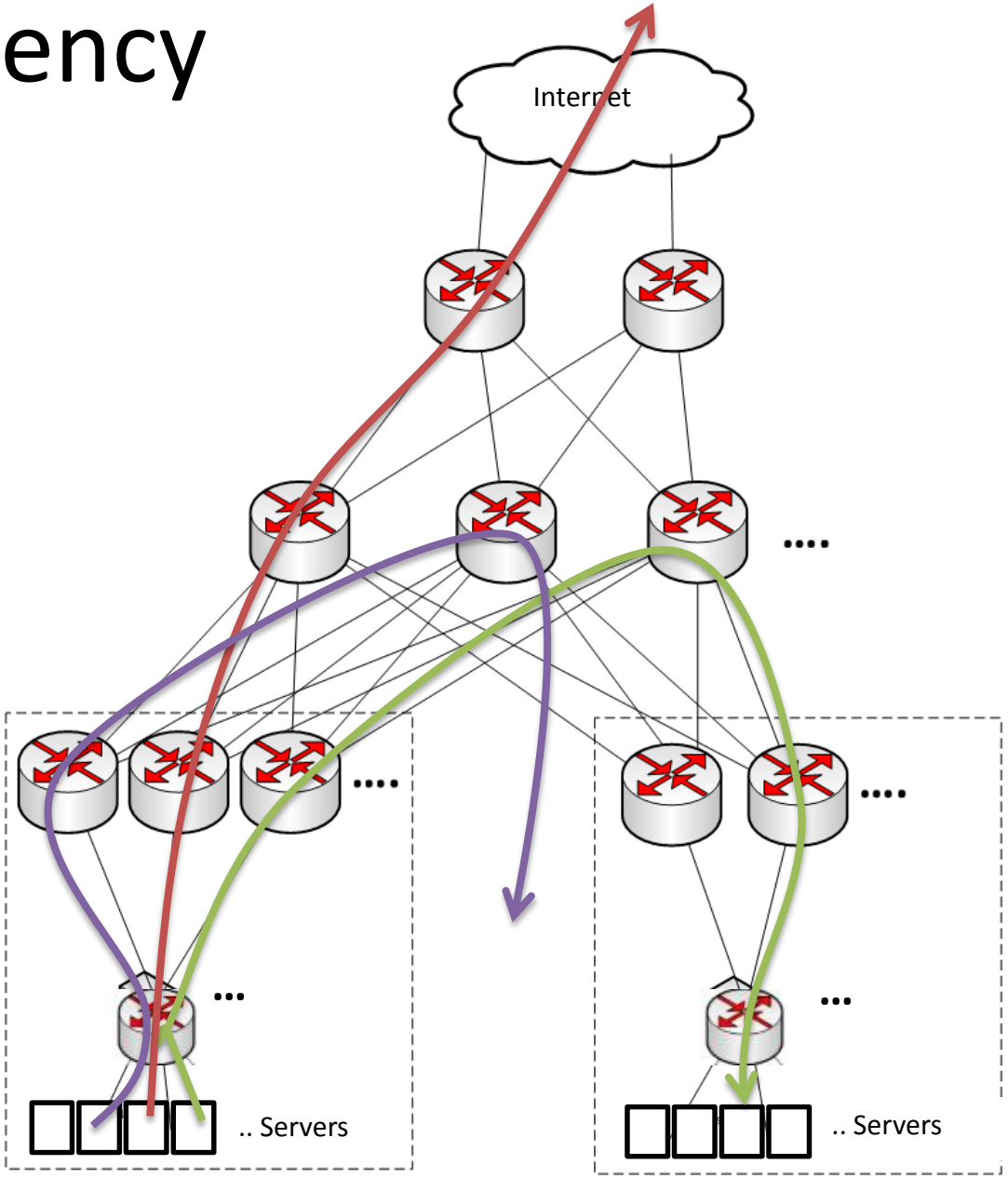
Time, Time, Time …

- Many apps want SLAs on the order of a few milliseconds
  - And at the 99.9th percentile or higher
  - Working in this “pseudo-realtime” region is hard
- Many activities require coordinating at the scale of 1ms, and this is hard
  - Issues like this exist in both *nix and Windows
  - DelAck timer in Windows doesn’t actually fire on 10ms intervals
  - Some kernel traffic pacing APIs burst to 700 Mbps for 15 ms when you asked for 300Mbps

Ack delay (ms)	Count
0~10	712
10~20	295
20~30	105
30~40	8
40~50	8
50~60	10
60~70	2
70~80	0
80~90	1
>90	3

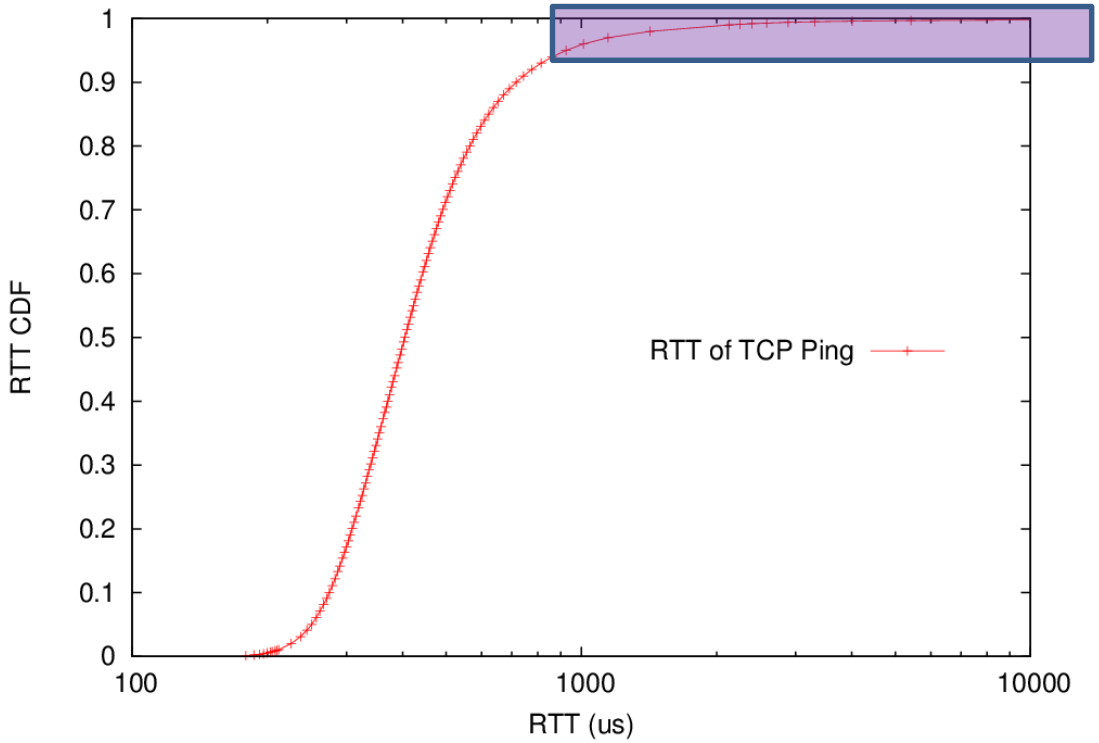
Network Latency

- How to measure it?
  - Big mesh of pings



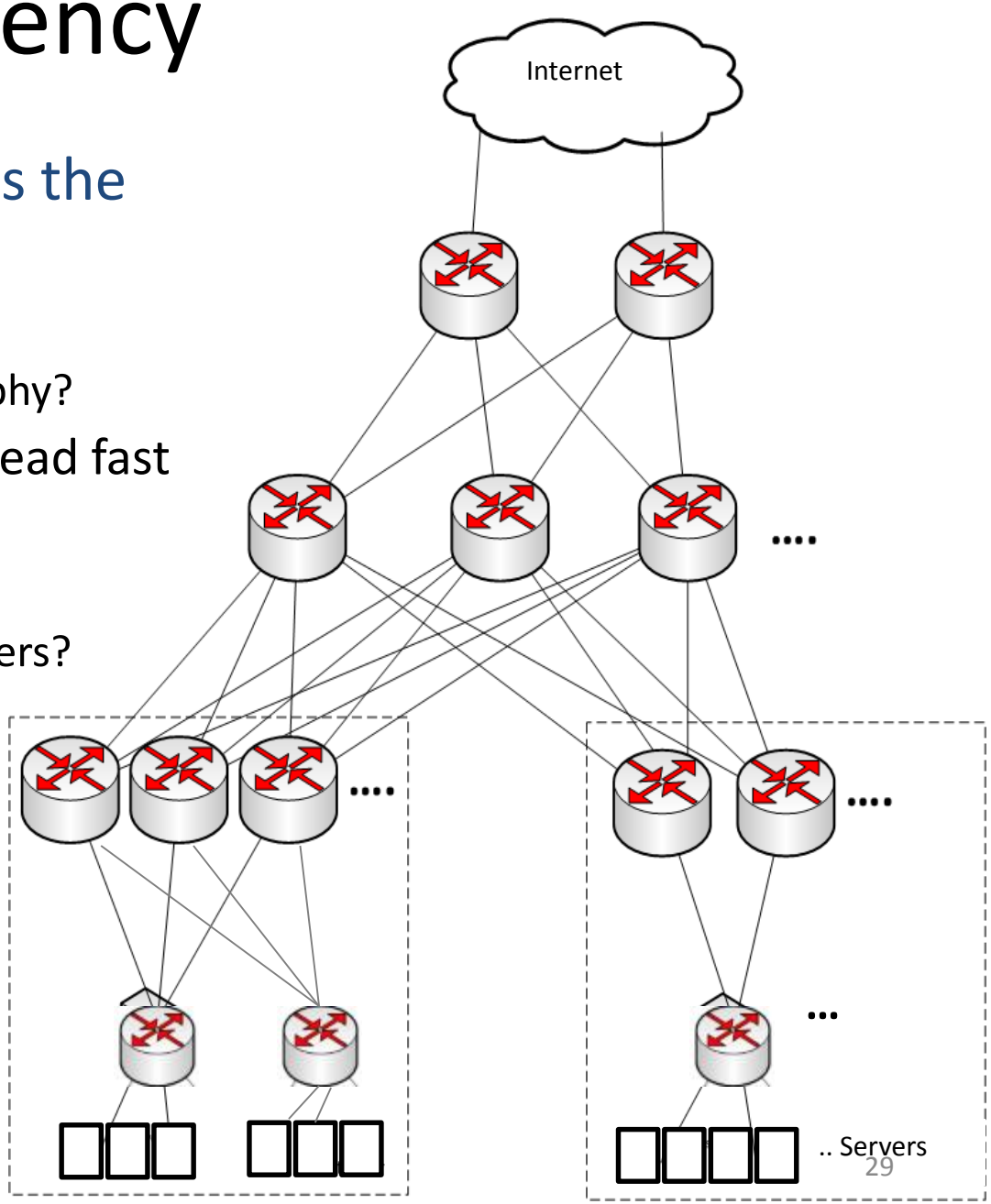
RTT Long Tail

- Hundreds of billions RTT measurement a day from hundreds of thousands servers
  - Within a DC, long tail RTT values with tens of ms
- Questions:
  - Where does the large RTT come from?
  - How to reduce?



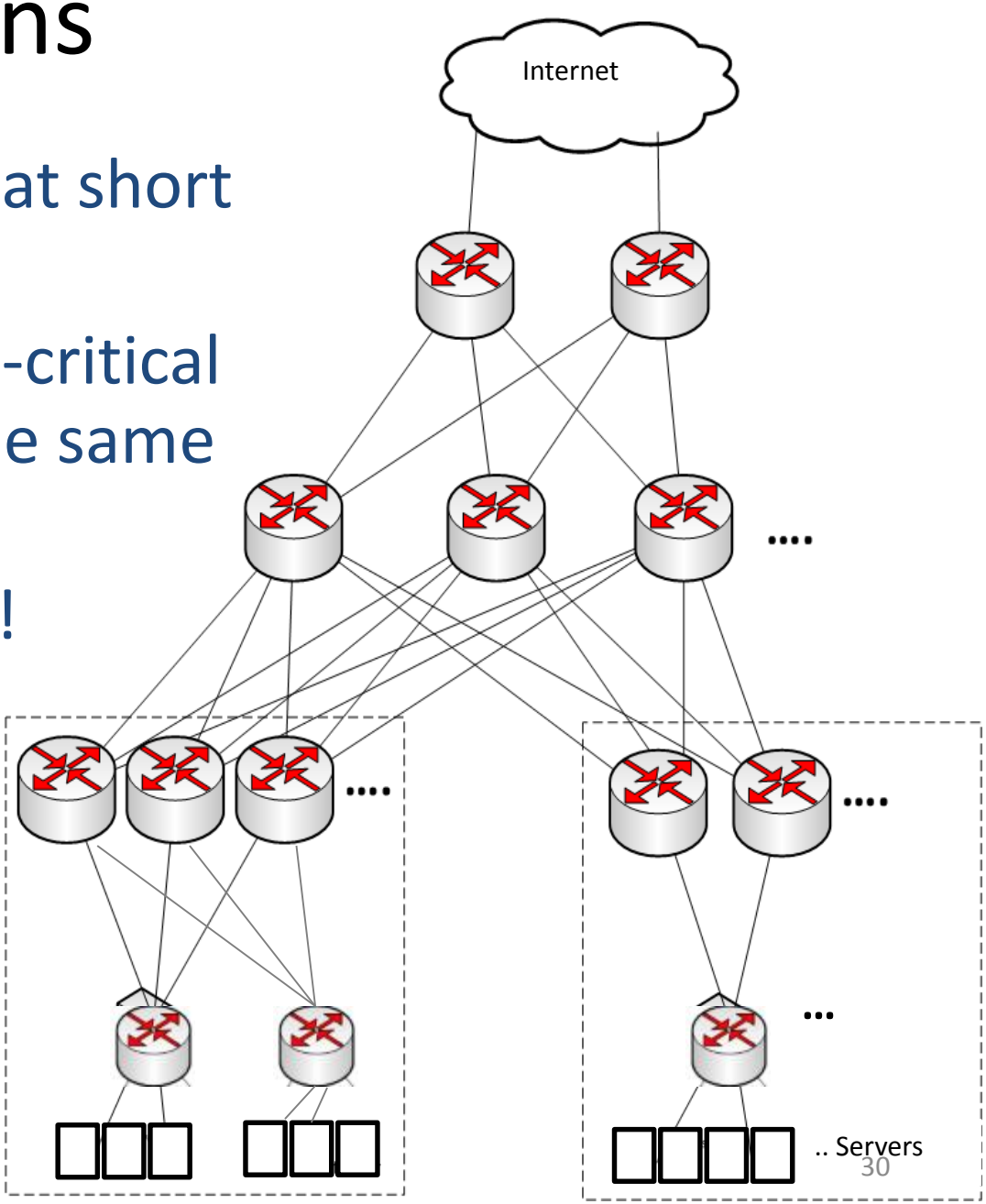
Network Latency

- Where in the network is the queuing?
  - Very transient
    - Triangulation/tomography?
  - ASIC counters can't be read fast enough
  - Better ASIC circuits?
    - Hi/Lo watermark counters?



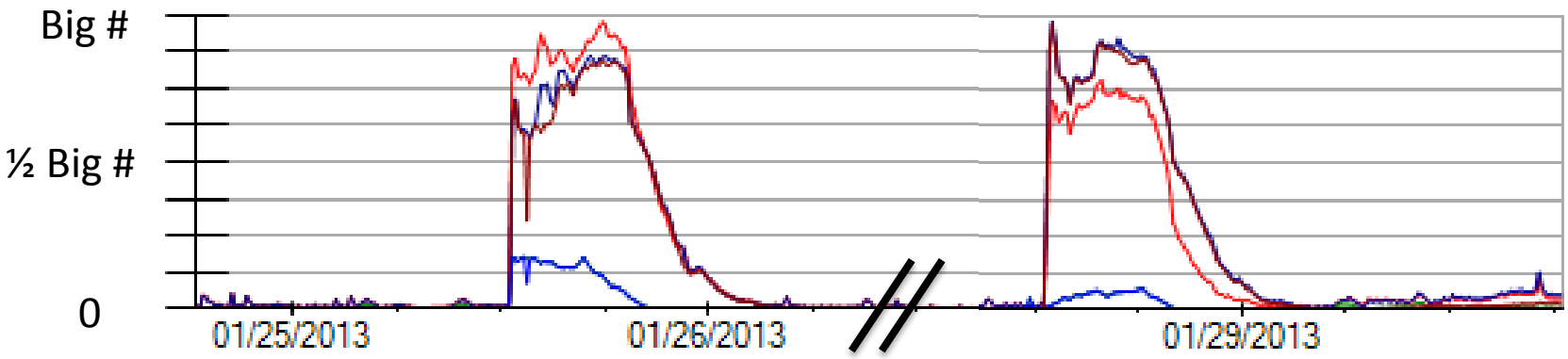
Implications

- Do services interact at short timescales?
- Can two bursty time-critical services be under the same switches?
- Need better models!



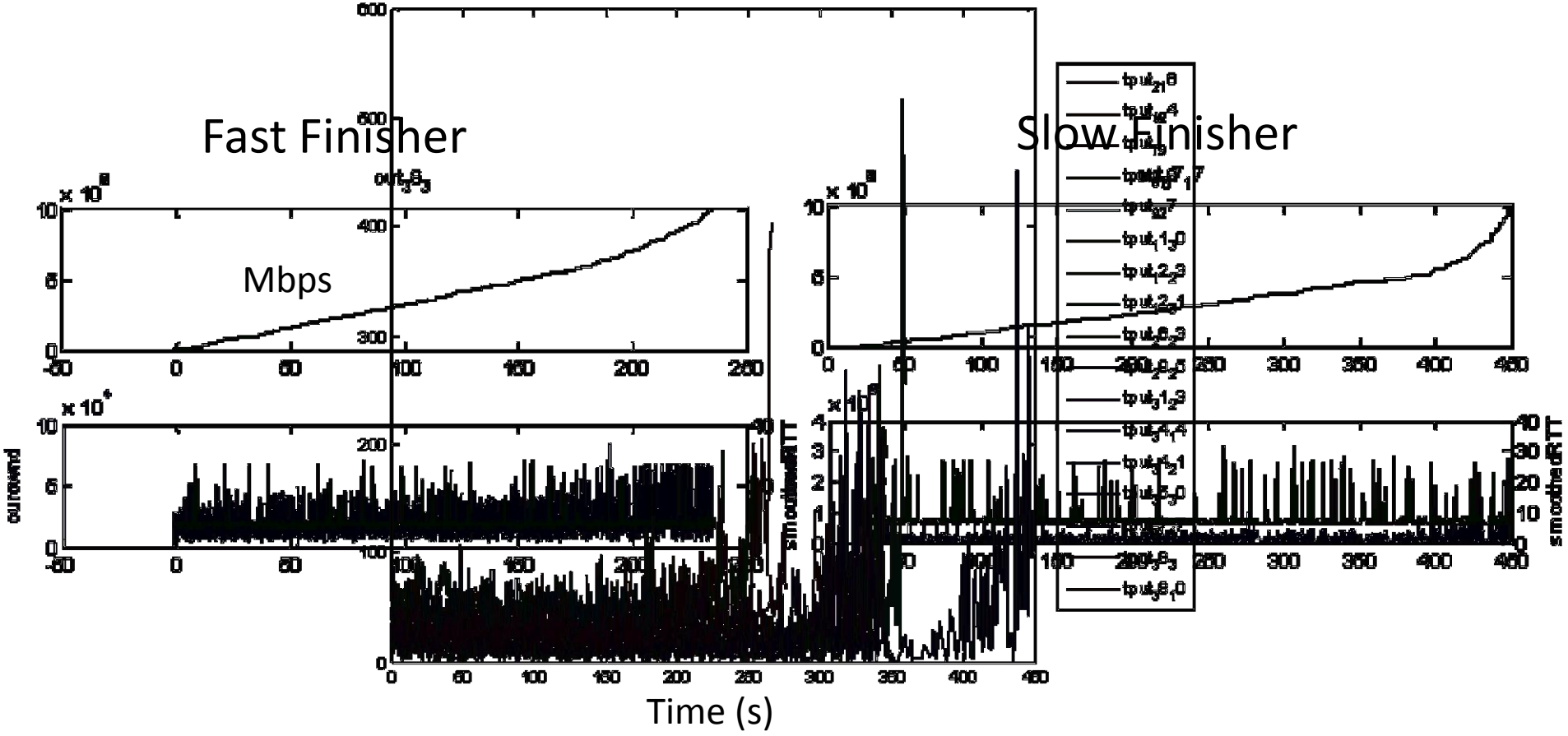
Slow TCP connections

- Very common scenario
  - Transfer hundreds of TB or more data from one DC to another DC (the big data biz has lots of big data…)
  - Thousands or more machines involved
  - Hundreds of thousands to millions of TCP connections
  - Most TCP connections achieve tens MB/s throughput and can transmit 1GB in minutes
  - But there are always slow TCP connections (e.g., need hours to transmit 1GB)



Slow TCP connections

- Question
  - Why some of the TCP connections are so slow?
  - Large ensembles of TCP connections behave strangely



TCP: Love It/Hate It

- Most applications need what TCP provides
  - Control of: flow, congestion, error, segmentation
- Attempts to improve by reimplementing have checkered history
  - Just one example: memcached - TCP, UDP, TCP, …
- But performance critical apps continually fall down on TCP issues
  - Incast, small buffers in Top of Rack switches
  - Large delay*BW links coupled with HW like Large Send Offload
  - 10Gbps servers talking to 1Gbps servers
  - A request/response structure on a stream-based transport

Data Center Applications: Minivans through Indy Race Cars

- Many applications are not very performance critical
  - Optimize for time to write and maintain them
  - Networking needs of a minivan
    - Should just work without networking experts
- Some applications are very performance critical
  - Run on thousands of servers
  - Would do anything to get 5% more efficiency
  - Networking needs of an Indy race car
    - Tweak every TCP parameter, change app, pace pkts
- Don't always know which you're building when you start…

<shameless plug>

Want to design some of the biggest data centers in the world?

Want to experience what “scalable” and “reliable” really mean?

Think measuring compute capacity in millions of MIPs is small potatoes?

Microsoft’s AutoPilot team is hiring!

</shameless plug>

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
- **DCTCP: Efficient Packet Transport for the Commoditized Data Center**
  - http://research.microsoft.com/en-us/um/people/padhye/publications/dctcp-sigcomm2010.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

Backup

Virtual Networking Topology

- Each service/tenant needs its own virtual network
  - In cloud-hosting, competing companies need to run over the same shared network infrastructure
  - Connected back to other locations (e.g., home DC)
  - Seamlessly integrated with network in other location
  - Performance isolation between virtual networks
  - A describable SLA
- Today, implemented via shim layers on the servers
  - Virtual switches with the needed features to segregate traffic

Trends in Cloud

- More, More, More
  - Faced with increasing scale, automation is the only path to survival
  - "Human-touch" creeps into processes – constant vigilance required to stamp it out
- Network time is significant and increasing(?) component of total Page Load Time
  - Lots of causes, lots of solutions – all important
- More powerful servers need a more powerful network
  - Server costs dominate, so network spend that unlocks server capacity pays back quickly

How to Get Started

- Talk to your own enterprises
  - Your campus IT folks
  - Companies nearby
- Talk to anyone that runs applications or networks
  - NANOG, LISA
- Get your hands dirty
  - Build a cloud app
  - Set up a cloud – emulab? your own scripts? Try provisioning 10 servers, or 1 server 10 times

How to Do Good Work

- Experiment!
  - Do microbenchmarks
  - Get time on cloud services (Azure, AWS, …) there are programs to help pay for it
  - Simulations alone don't cut it
- Make friends with HCI/Sociologist colleagues
  - Management processes have a human element
  - Do a real study of them
  - There are fields that know how to do this