Datacenters
(Optional fun)

What we will cover

(Datacenter Topic 7 is not examinable in 2013-14)

- Characteristics of a datacenter environment
  - goals, constraints, workloads, *etc.*
- How and why DC networks are different (*vs.* WAN)
  - e.g., latency, geo, autonomy, …
- How traditional solutions fare in this environment
  - e.g., IP, Ethernet, TCP, ARP, DHCP
- Not details of *how* datacenter networks operate

Disclaimer

- Material is emerging (not established) wisdom
- Material is incomplete
  - many details on how and why datacenter networks operate aren’t public

Why Datacenters?

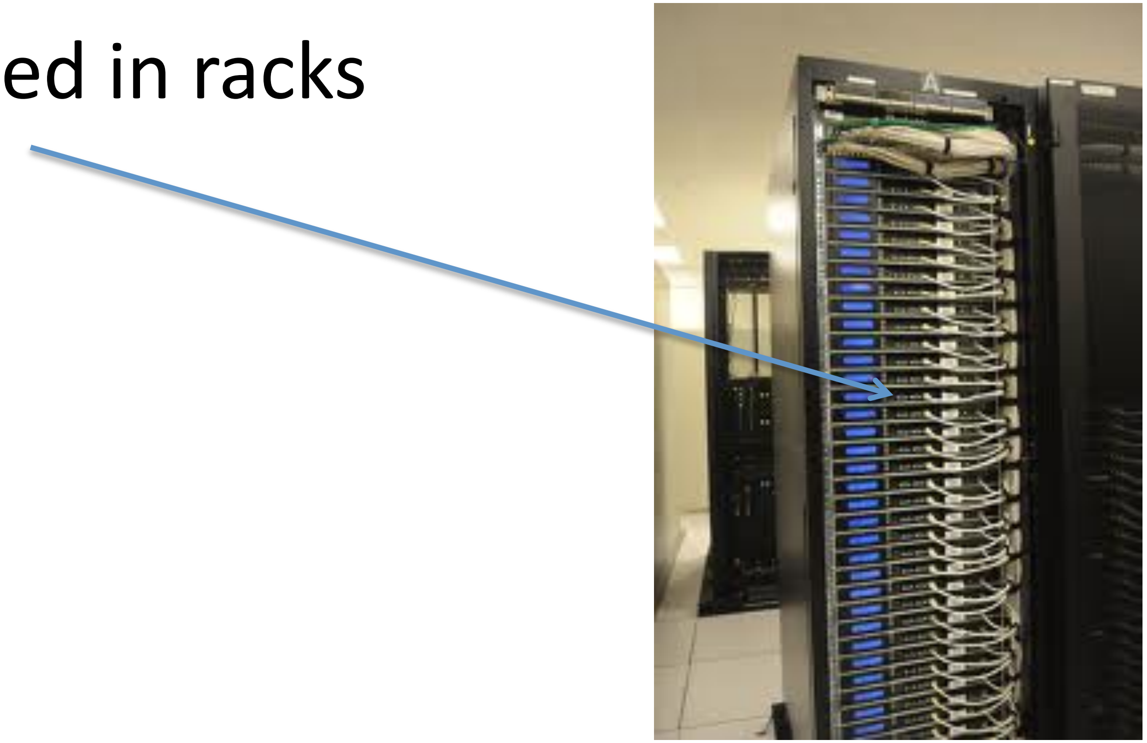
Your <public-life, private-life, banks, government> live in my datacenter.

Security, Privacy, Control, Cost, Energy, (breaking) received wisdom; all this and more come together into sharp focus in datacenters.

Do I need to labor the point?

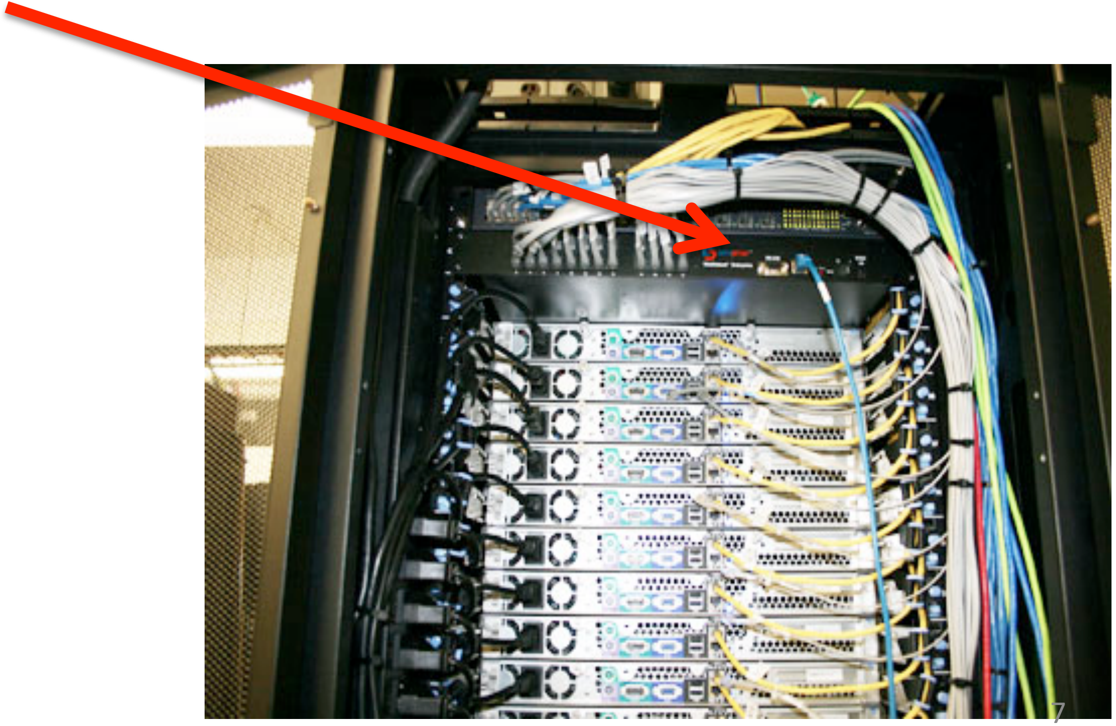
What goes into a datacenter (network)?

- Servers organized in racks



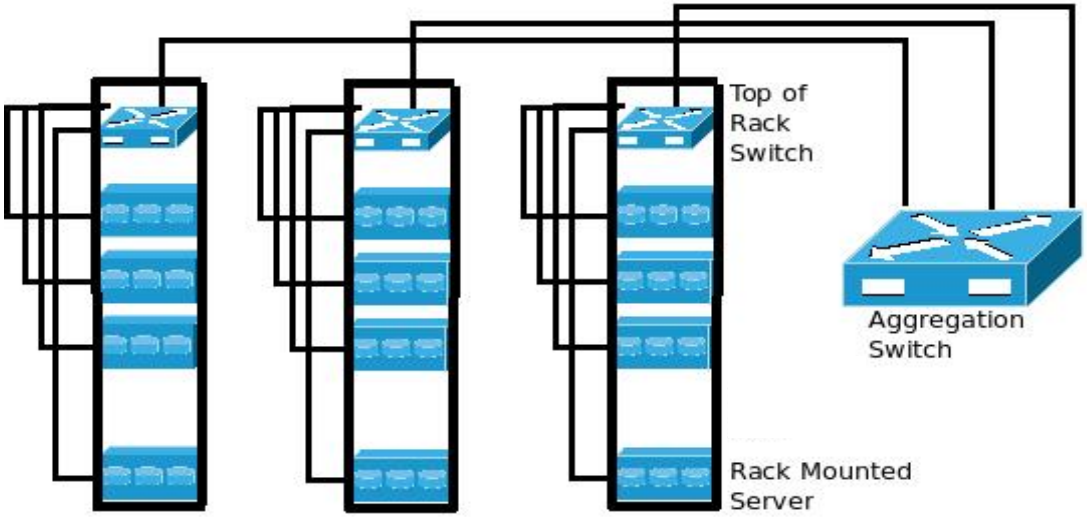
What goes into a datacenter (network)?

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



What goes into a datacenter (network)?

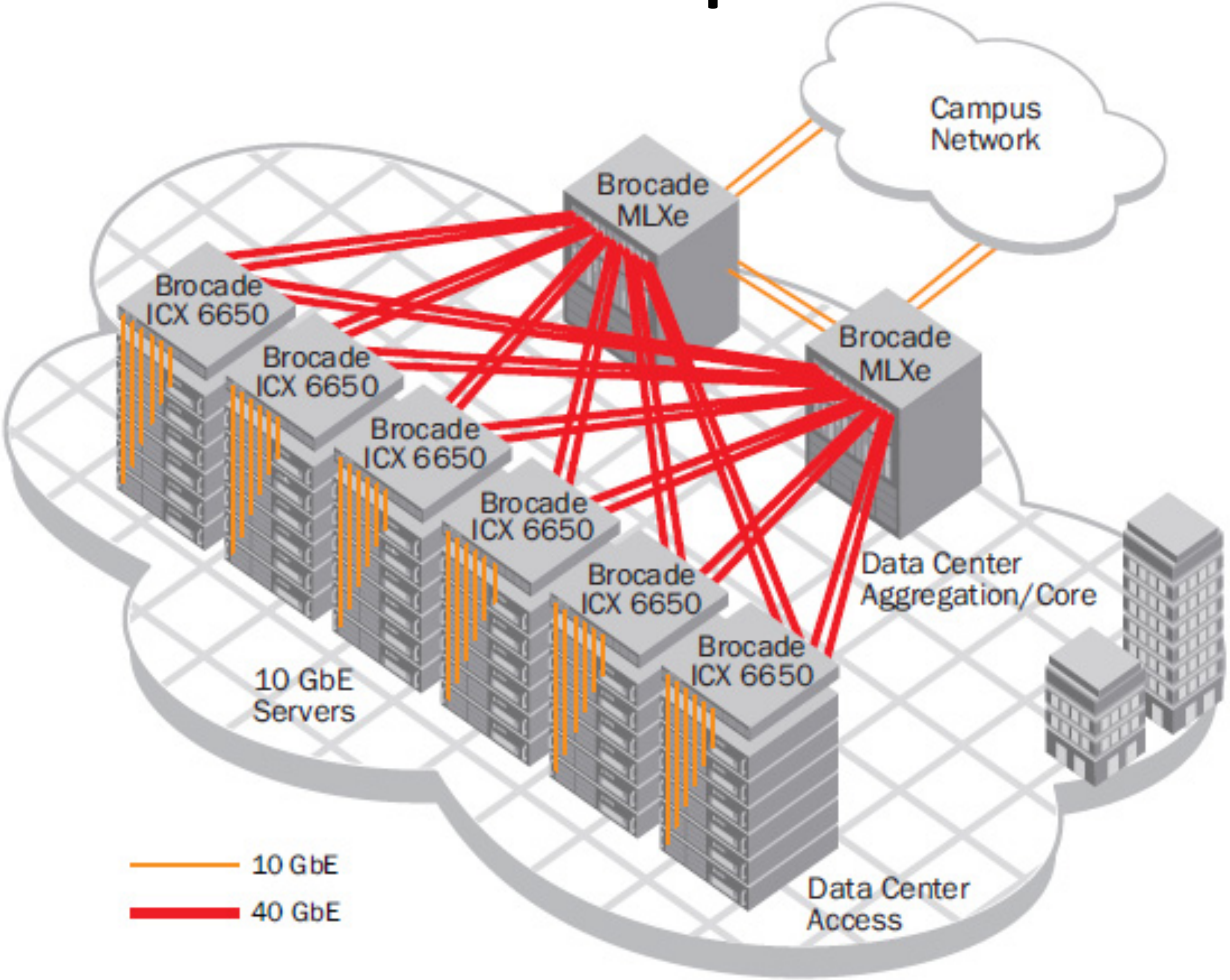
- Servers organized in racks
- Each rack has a `Top of Rack’ (ToR) switch
- An `aggregation fabric’ interconnects ToR switches



What goes into a datacenter (network)?

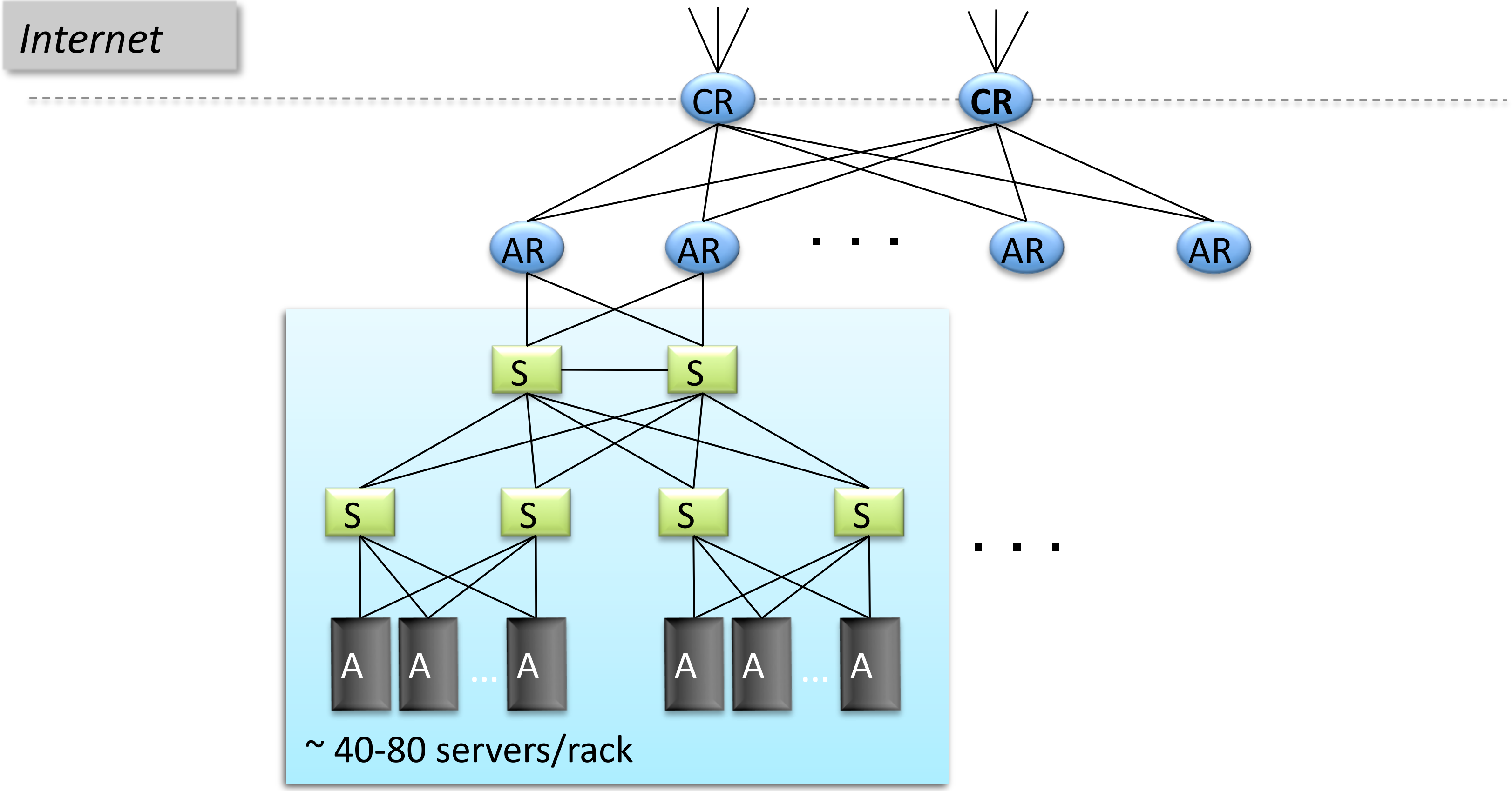
- Servers organized in racks
- Each rack has a `Top of Rack' (ToR) switch
- An `aggregation fabric' interconnects ToR switches
- Connected to the outside via `core' switches
  – note: blurry line between aggregation and core
- With network redundancy of ~2x for robustness

Example 1



Brocade reference design

Example 2



Cisco reference design

Observations on DC architecture

- Regular, well-defined arrangement
- Hierarchical structure with rack/aggr/core layers
- Mostly homogenous within a layer
- Supports communication between servers and between servers and the external world

Contrast: ad-hoc structure, heterogeneity of WANs

Datacenters have been around for a while

1949, EDSAC

What’s new?

SCALE!

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

What’s new?

- Scale
- Service model
  - user-facing, revenue generating services
  - multi-tenancy
  - jargon: SaaS, PaaS, DaaS, IaaS, …

Implications

- Scale
  - need scalable solutions (duh)
  - improving efficiency, lowering cost is critical

  → *`scale out' solutions w/ commodity technologies*

- Service model
  - performance means $$
  - *virtualization* for isolation and portability

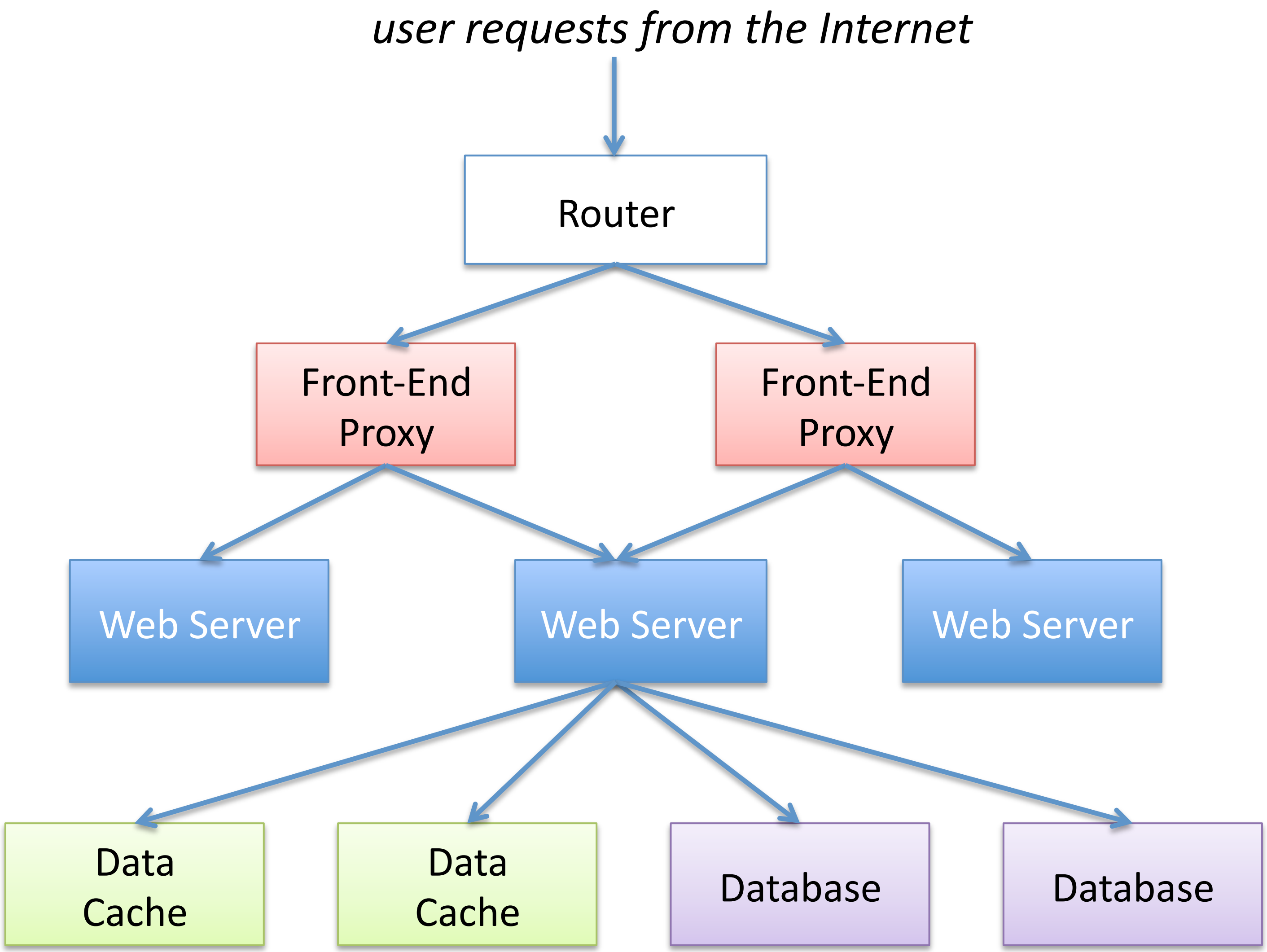
Multi-Tier Applications

- Applications decomposed into tasks
  - Many separate components
  - Running in parallel on different machines

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

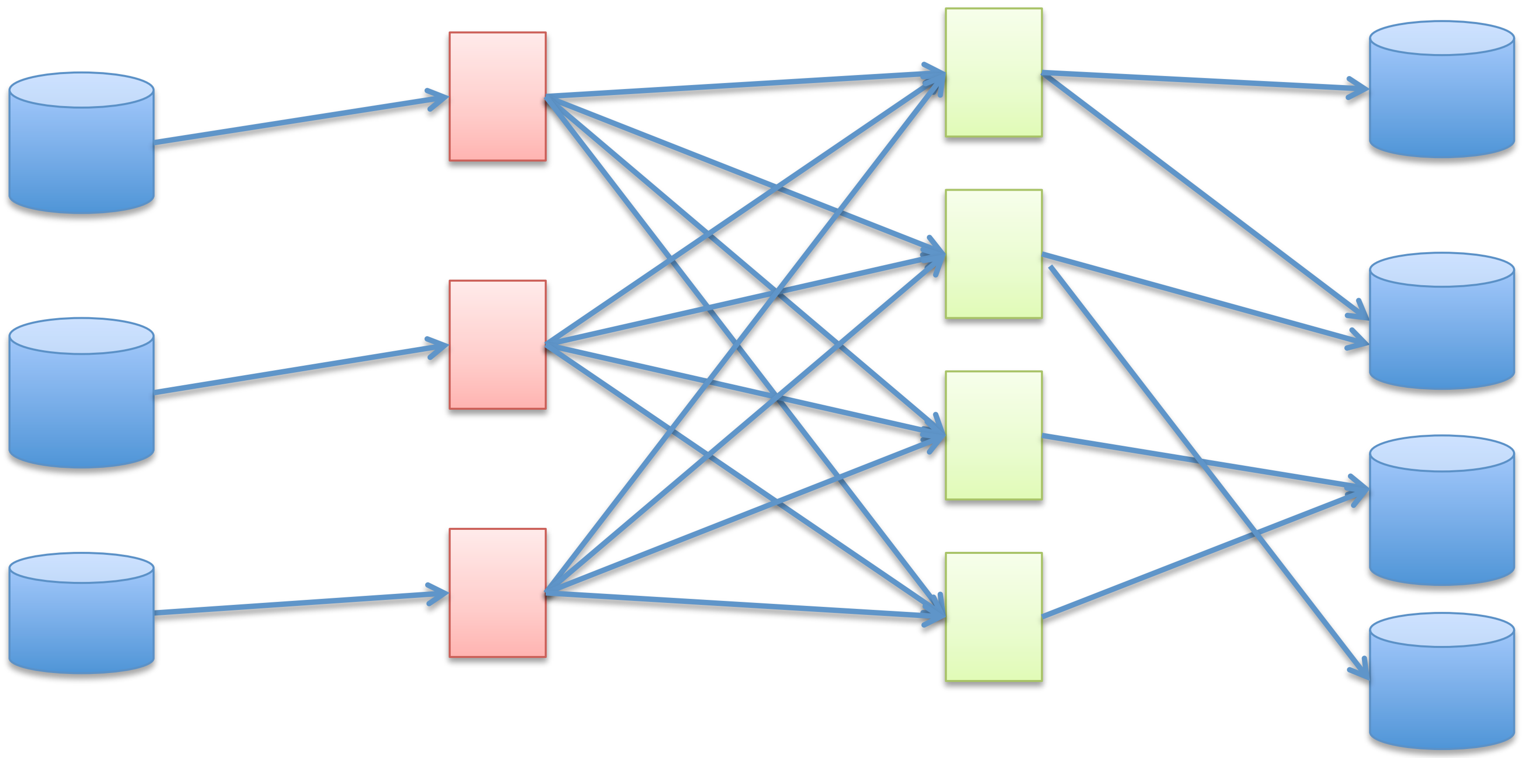
North-South Traffic



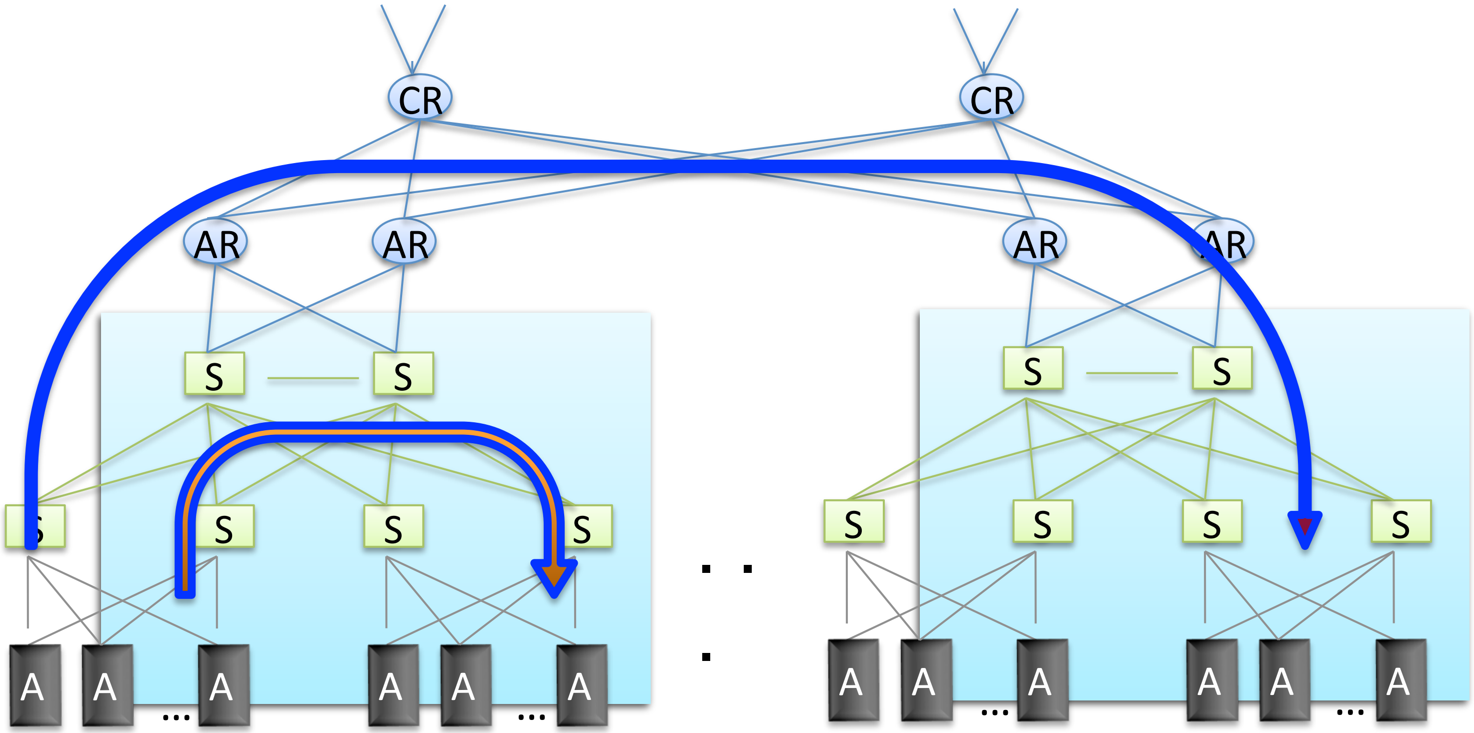
Componentization leads to different types of network traffic

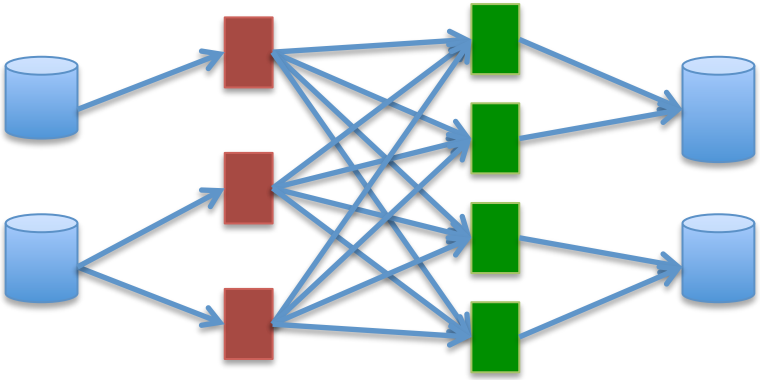
- "North-South traffic"
  - Traffic between external clients and the datacenter
  - Handled by front-end (web) servers, mid-tier application servers, and back-end databases
  - Traffic patterns fairly stable, though diurnal variations
- "East-West traffic"
  - Traffic between machines in the datacenter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

East-West Traffic

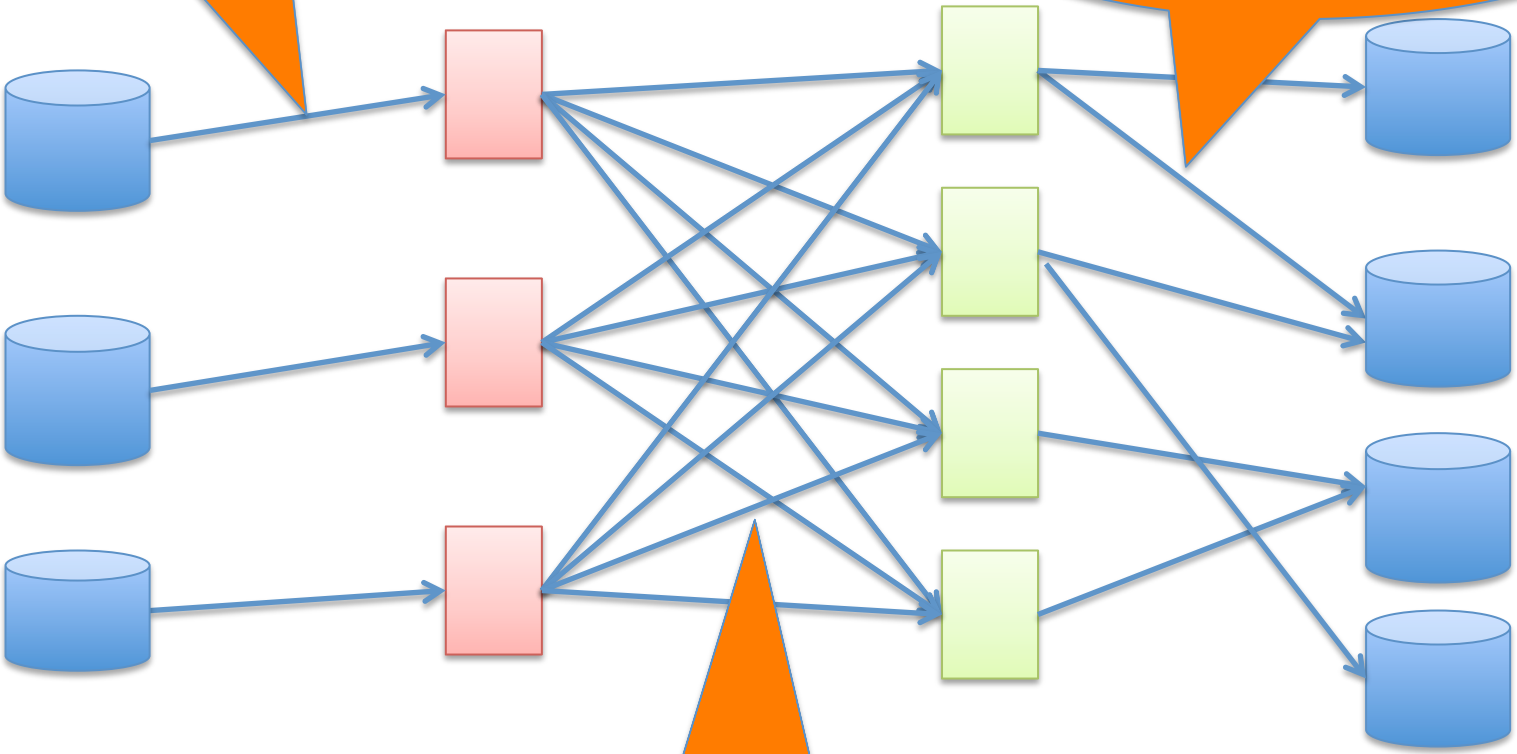


East-West Traffic





st-West T



Often doesn’t cross the network

Some fraction (typically 2/3) crosses the network

Always goes over the network

Distributed Storage

Ma

uce

Distributed Storage

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*

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

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

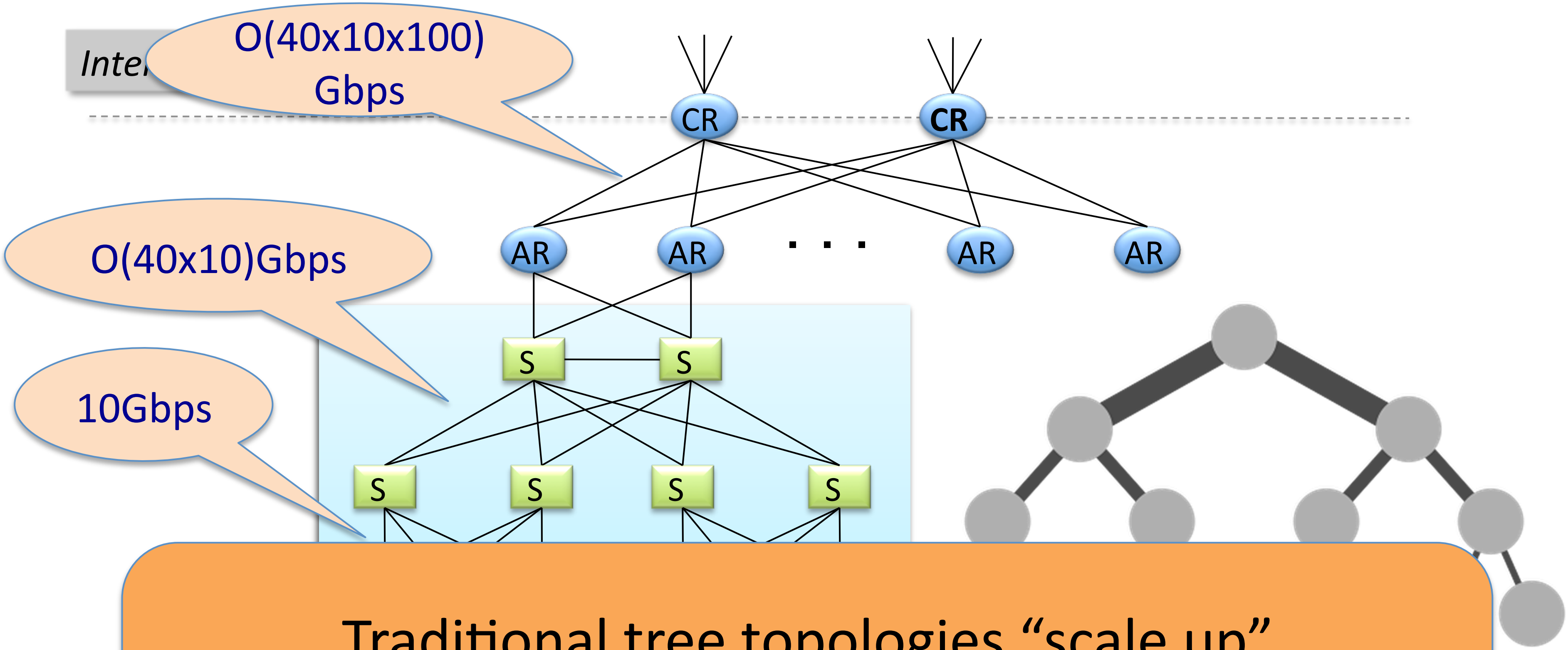
- Huge scale
- Limited geographic scope
- Single administrative domain
- Control over one/both endpoints
- Control over the *placement* of traffic source/sink
- Regular/planned topologies (e.g., trees/fat-trees)
- Limited heterogeneity
  - – link speeds, technologies, latencies, …

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!

Full Bisection Bandwidth



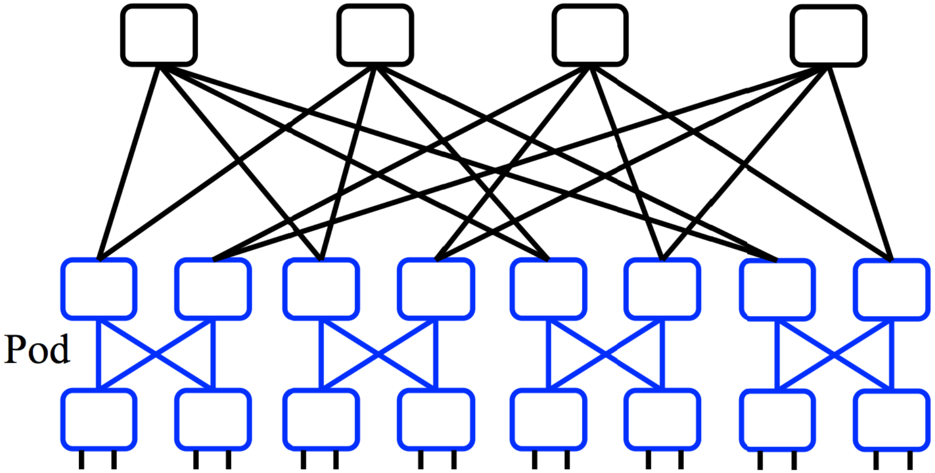
Traditional tree topologies "scale up"

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

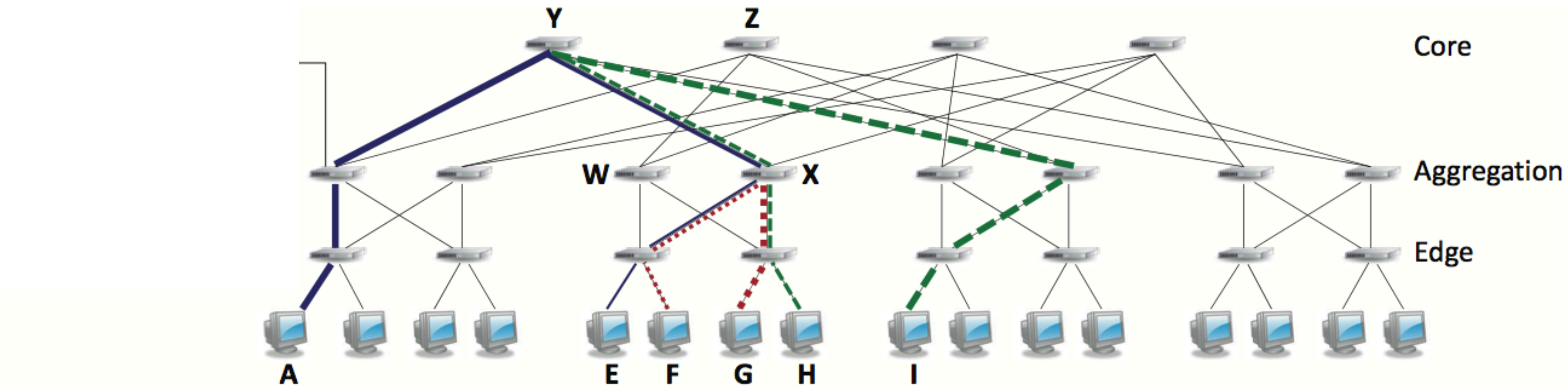
A “Scale Out” Design

- 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
    - reduces switching time

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
    - reduces queuing delay

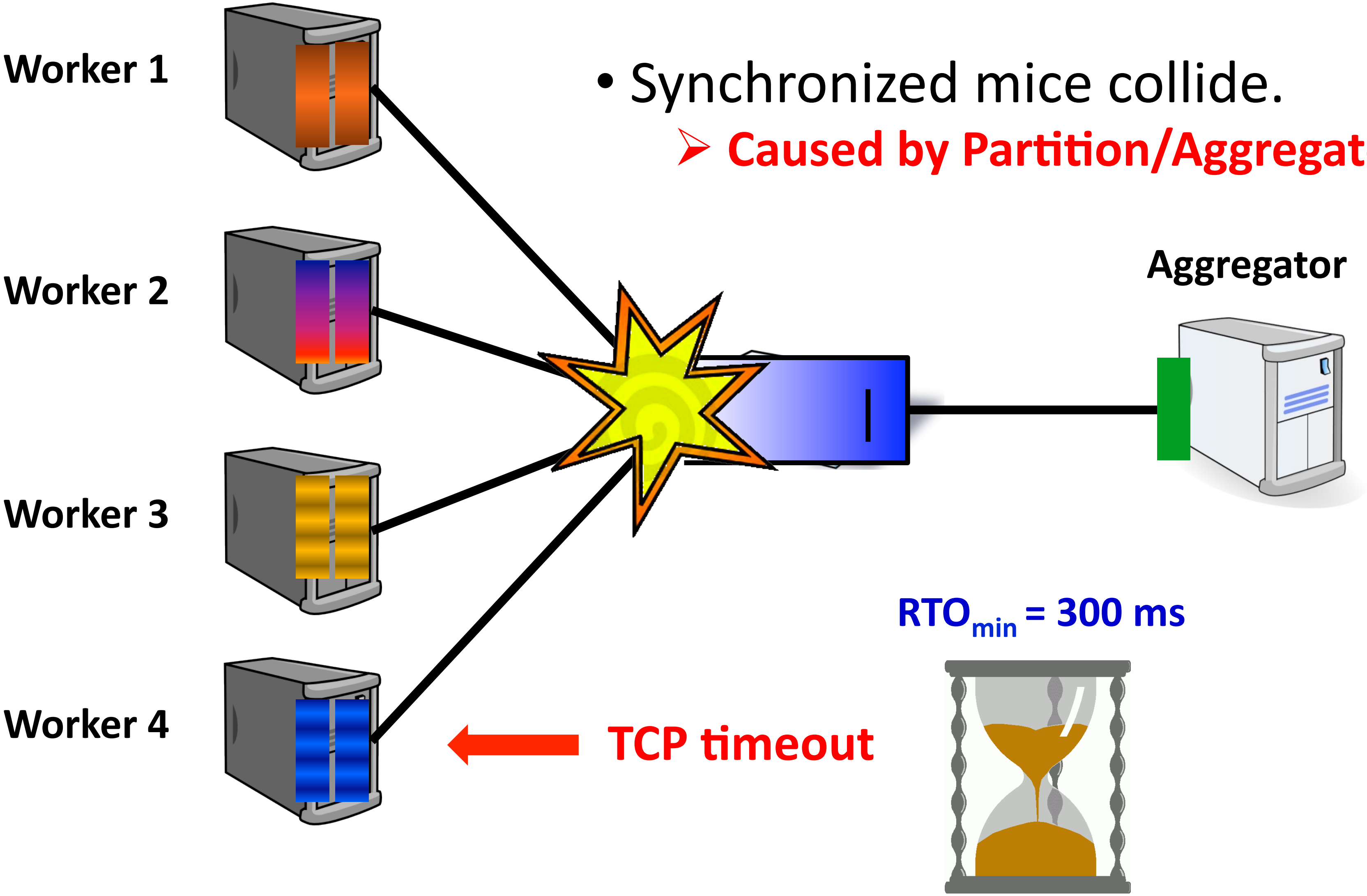
What’s different about DC networks?

Goals

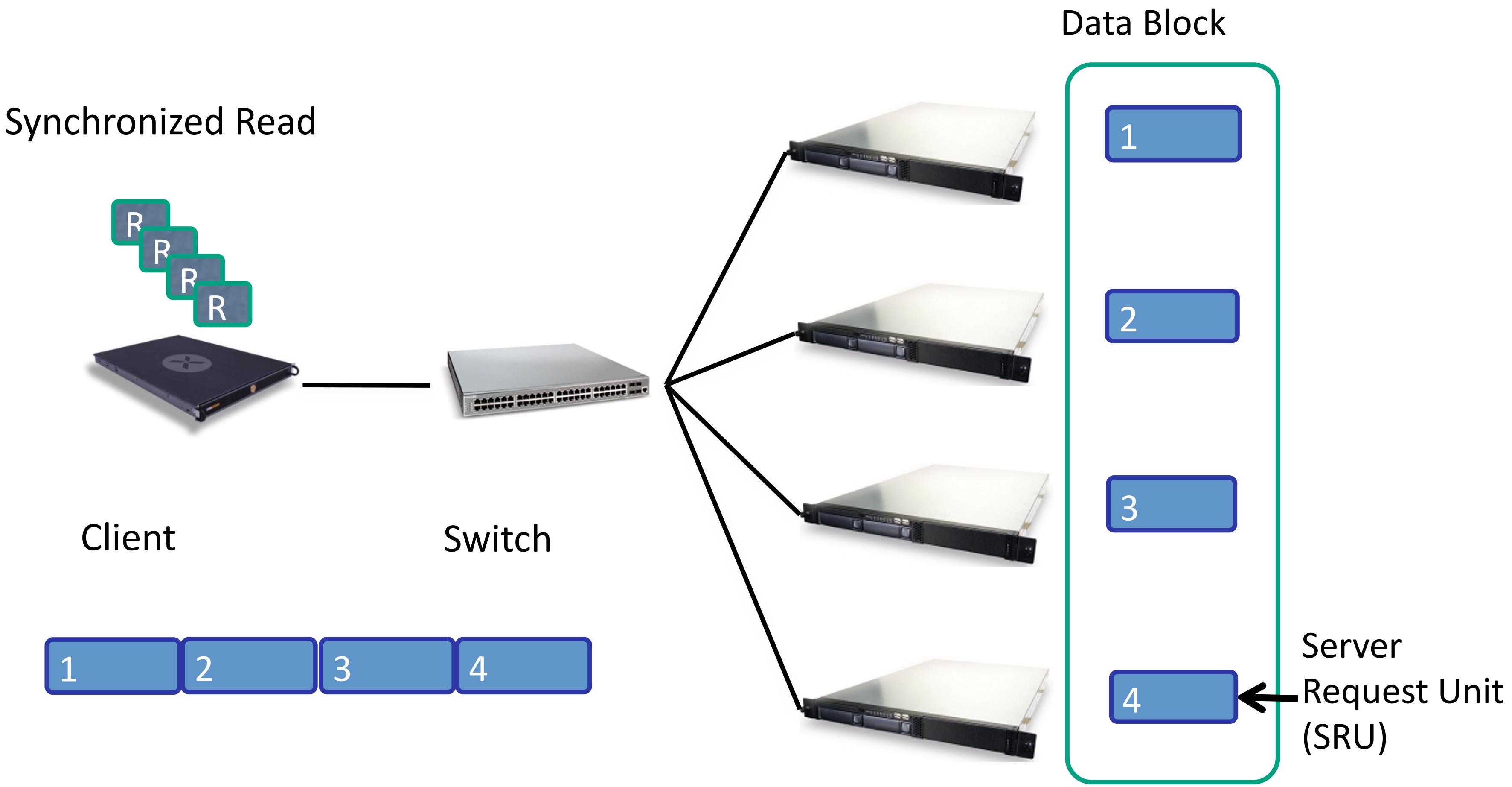
- Extreme bisection bandwidth requirements
- Extreme latency requirements
  - real money on the line
  - current target: 1μs RTTs
  - how? cut-through switches making a comeback (lec. 2!)
  - how? avoid congestion
  - how? fix TCP timers (e.g., default timeout is 500ms!)
  - how? fix/replace TCP to more rapidly fill the pipe

An example problem at scale - INCAST

- Synchronized mice collide.
  - **Caused by Partition/Aggregate.**



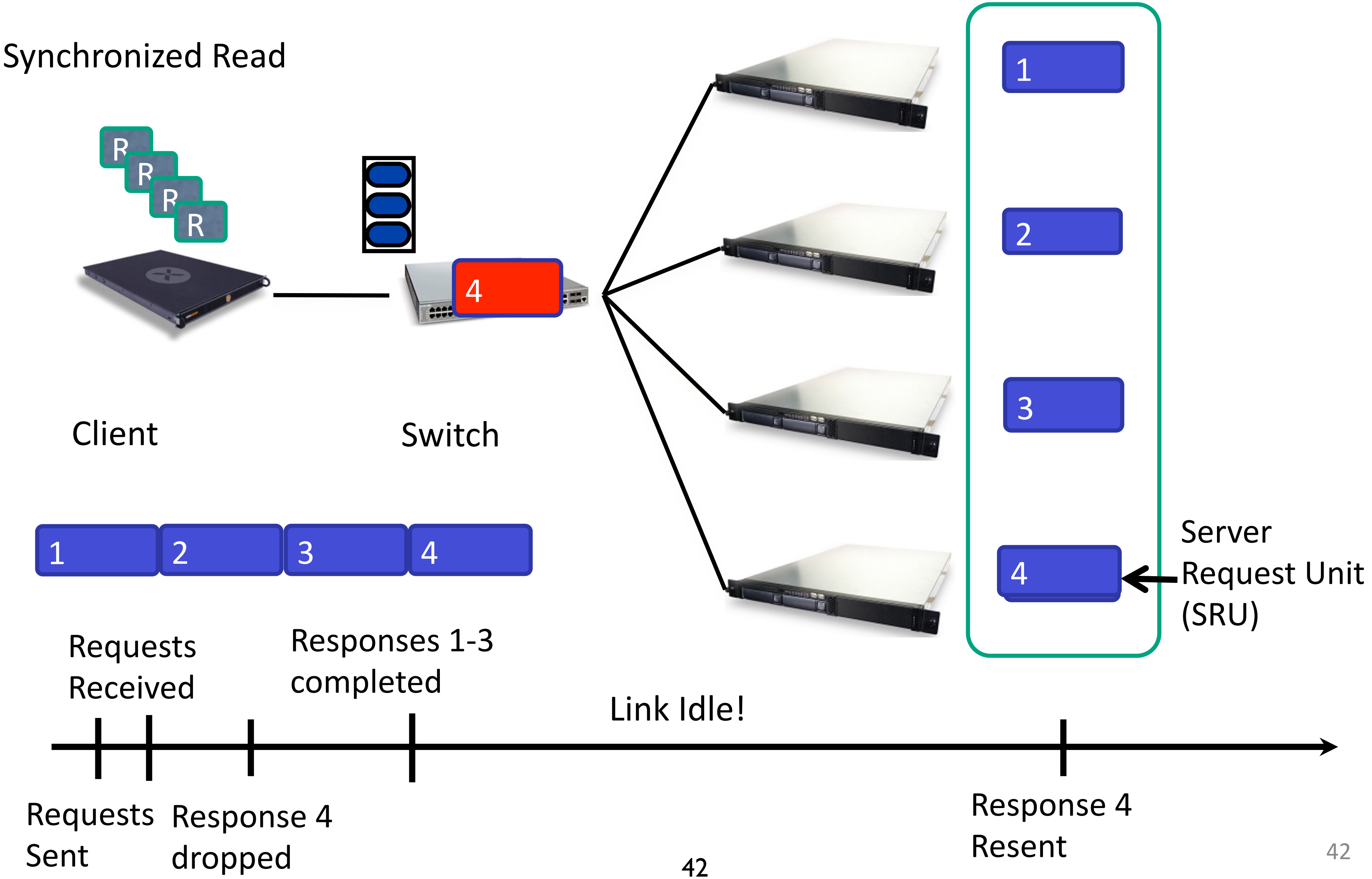
The Incast Workload



Client now sends
next batch of requests

Storage Servers

Incast Workload Overfills Buffers



Synchronized Read

Client

Switch

Server Request Unit (SRU)

Requests Received

Responses 1-3 completed

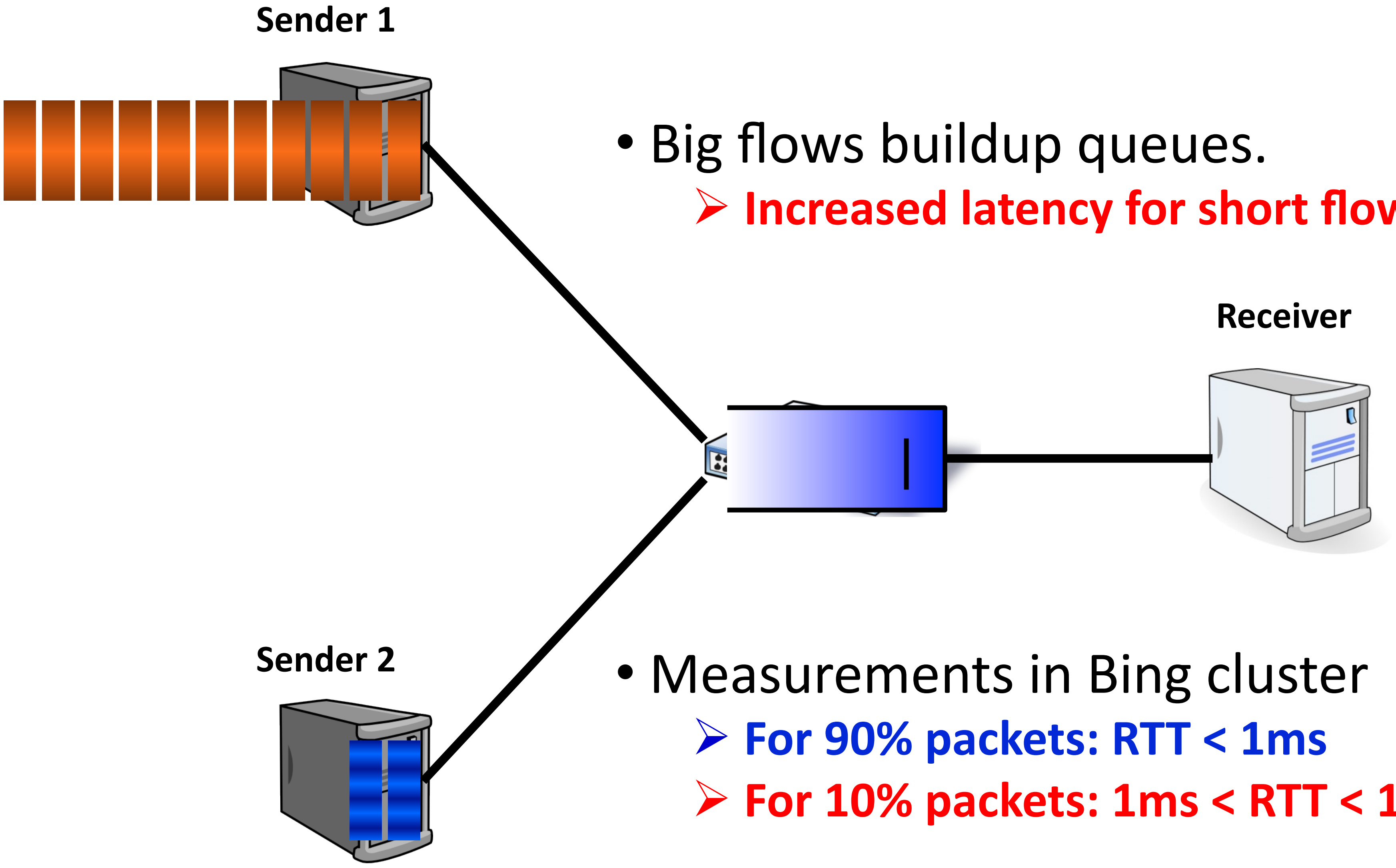
Link Idle!

Requests Sent

Response 4 dropped

Response 4 Resent

Queue Buildup



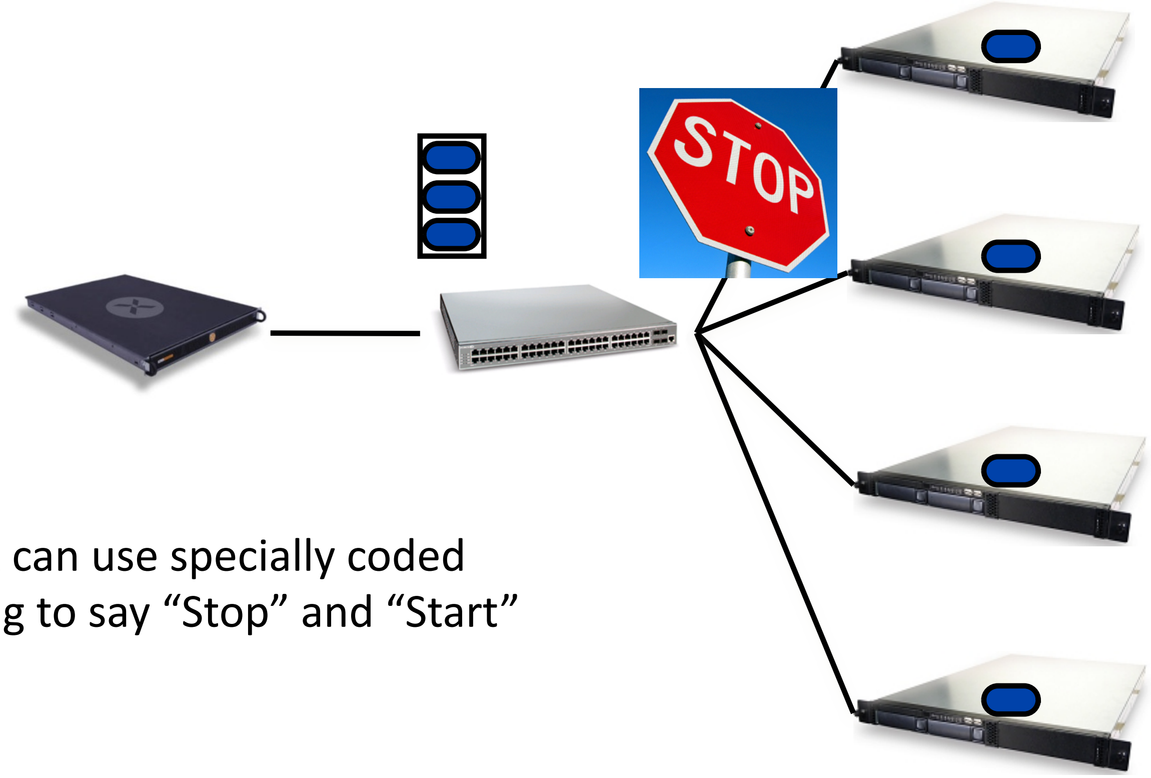
- Big flows buildup queues.
  - ➢ **Increased latency for short flows.**

- Measurements in Bing cluster
  - ➢ **For 90% packets: RTT < 1ms**
  - ➢ **For 10% packets: 1ms < RTT < 15ms**

Link-Layer Flow Control

Common between switches but this is flow-control to the end host too…

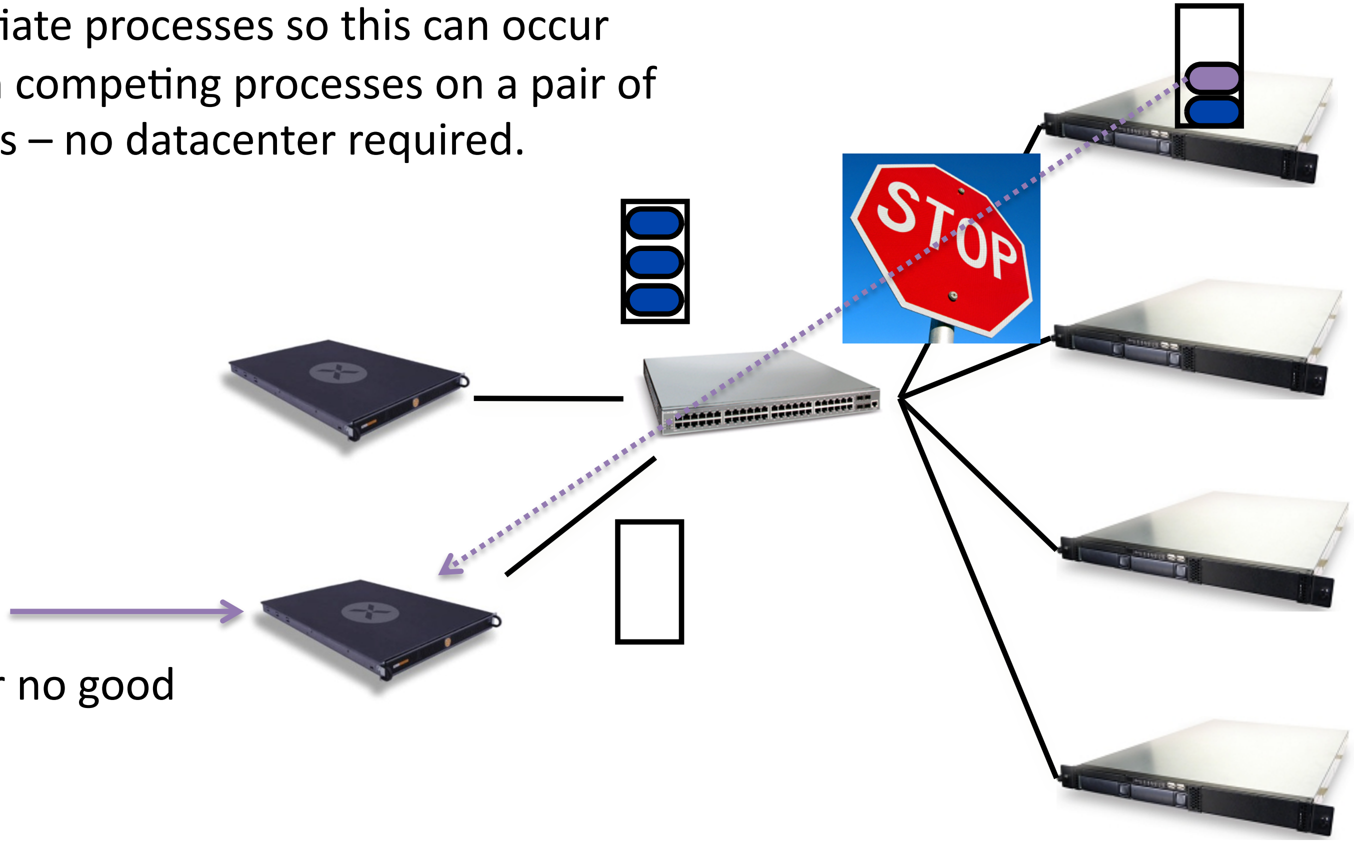
- Another idea to reduce incast is to employ Link-Layer Flow Control…..



Recall: the Data-Link can use specially coded symbols in the coding to say “Stop” and “Start”

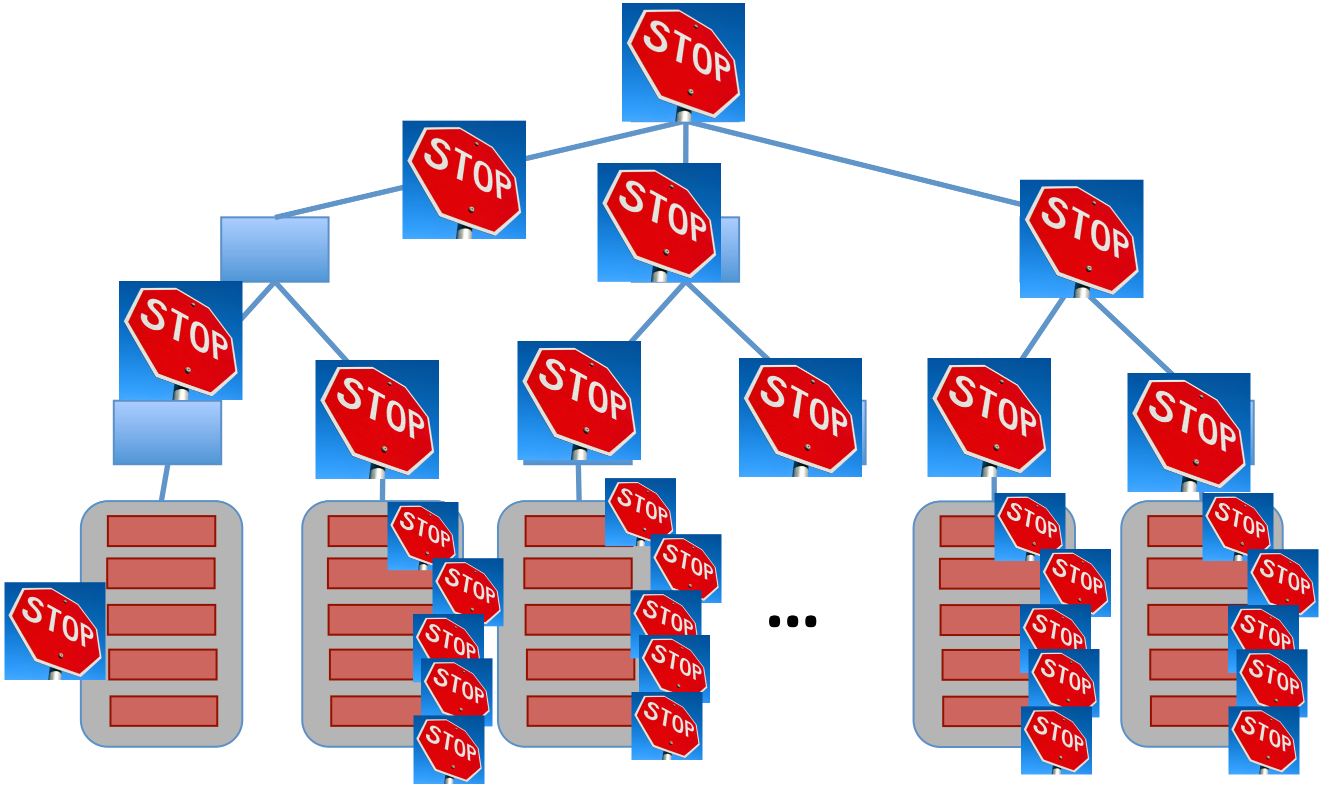
Link Layer Flow Control – The Dark side Head of Line Blocking….

Such HOL blocking does not even differentiate processes so this can occur between competing processes on a pair of machines – no datacenter required.



Waiting for no good reason….

Link Layer Flow Control
But its worse that you imagine….



Double down on trouble….

Did I mention this is Link-Layer!

That means no (IP) control traffic, no routing messages….

a whole system waiting for one machine

Incast is very unpleasant.

Reducing the impact of HOL in Link Layer Flow Control can be done through priority queues and *overtaking*….

What’s different about DC networks?

Goals

- Extreme bisection bandwidth requirements
- Extreme latency requirements
- *Predictable, deterministic* performance
  - “your packet will reach in Xms, or not at all”
  - “your VM will always see at least YGbps throughput”
  - Resurrecting `best effort’ vs. `Quality of Service’ debates
  - How is still an open question

What’s different about DC networks?

Goals

- Extreme bisection bandwidth requirements
- Extreme latency requirements
- *Predictable, deterministic* performance
- Differentiating between tenants is key
  - e.g., “No traffic between VMs of tenant A and tenant B”
  - “Tenant X cannot consume more than XGbps”
  - “Tenant Y’s traffic is low priority”

What's different about DC networks?

Goals

- Extreme bisection bandwidth requirements
- Extreme latency requirements
- *Predictable, deterministic* performance
- Differentiating between tenants is key
- Scalability (of course)
  - Q: How's Ethernet spanning tree looking?

What’s different about DC networks?

Goals

- Extreme bisection bandwidth requirements
- Extreme latency requirements
- *Predictable, deterministic* performance
- Differentiating between tenants is key
- Scalability (of course)
- Cost/efficiency
  - focus on commodity solutions, ease of management
  - some debate over the importance in the network case

Summary

- new characteristics and goals
- some liberating, some constraining
- scalability is the baseline requirement
- more emphasis on performance
- less emphasis on heterogeneity
- less emphasis on interoperability