CS4450

Computer Networks:
Architecture and Protocols

Lecture 25
Beyond The Internet

Rachit Agarwal

Announcements

- **Final: 05/12 @ 7PM, Hollister Hall B14**
    - Any conflicts? I must know by Thursday
- **Quiz drop policy**
    - ¯_o_/¯
- **Lost sessions**
    - This is the last week, no lost sessions next week
    - Please email cs4450lost@gmail.com
- **Make-up projects**
    - Turns out to be extremely hard to make projects that:
        - Do not require enough time from you
        - And yet, let you apply your knowledge about the material
    - We are working on it full-time; we will give you enough time

Announcements

- **Extra practice problems**
  - I promise we are working full-time on this as well
  - ETA: Thursday
- **Problem Solving sessions**
  - As promised, we are going to organize these sessions
  - This week, and next week
- **Any thing else that you would like?**

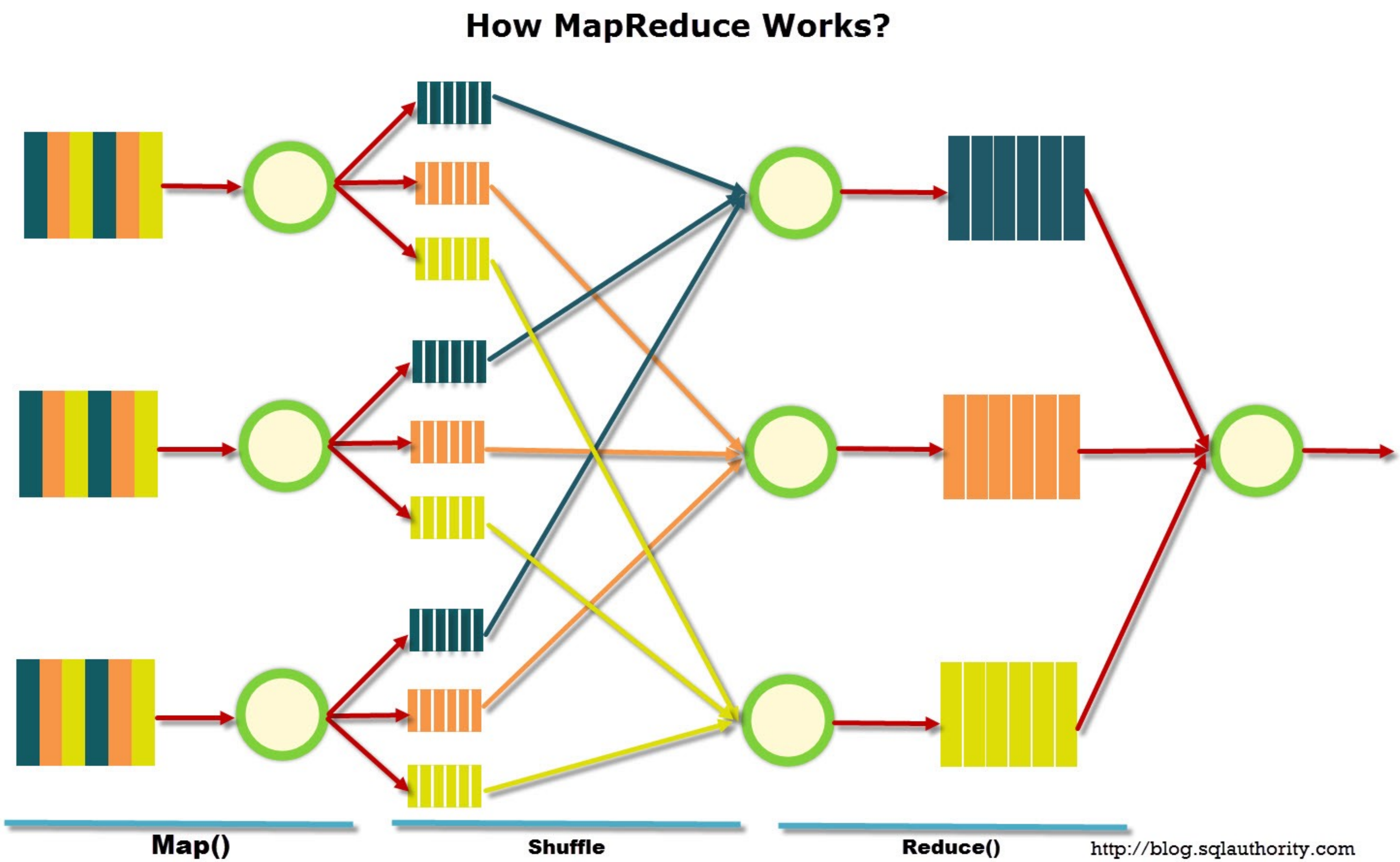
Goals for Today's Lecture

- **Understand how a new environment may lead to new design decisions**
- **Case study: Datacenter networks**

Lets start with an application - MapReduce

- Large scale data analytics

- Ex: Google "crawls" the web, and creates search indexes



Performance of distributed systems

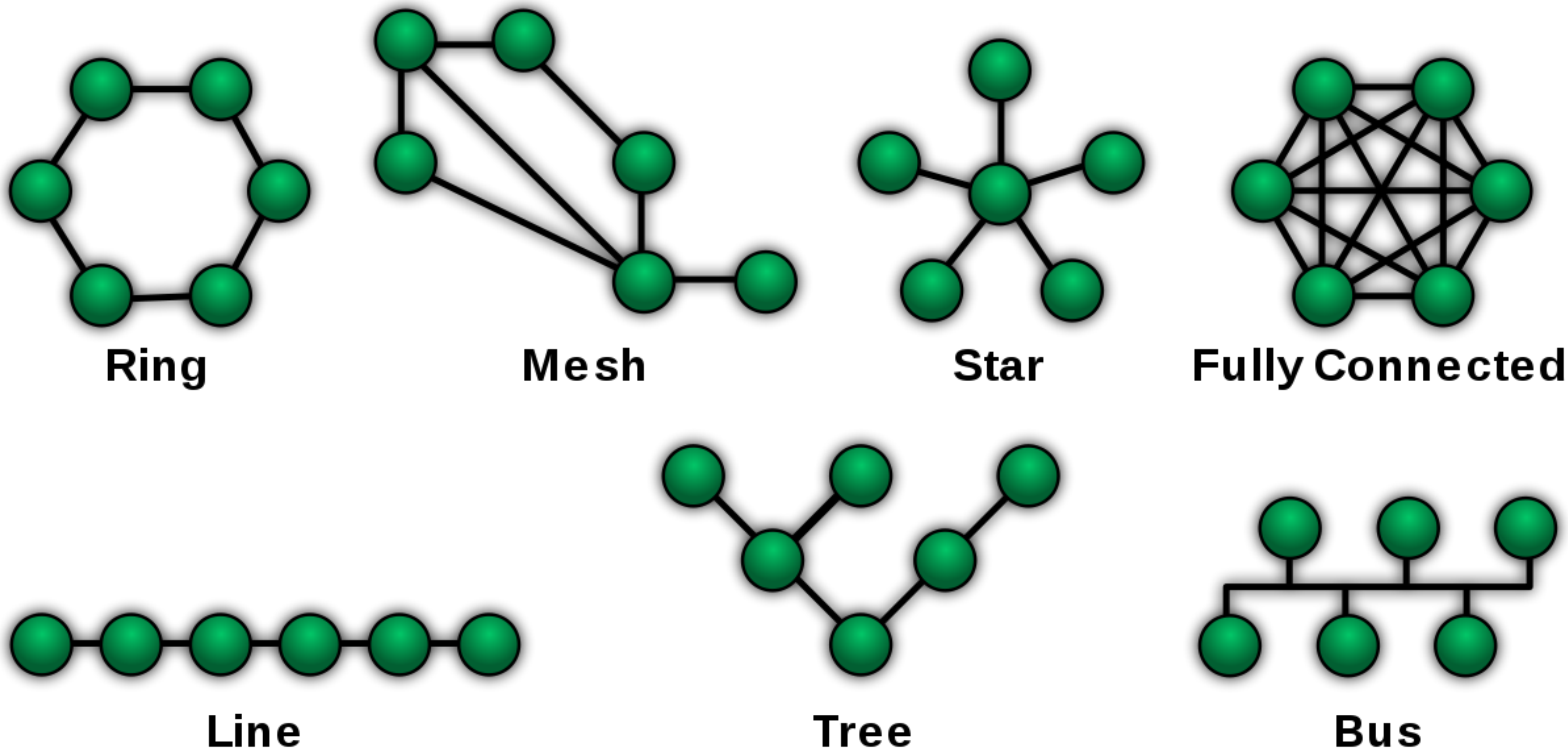
depends heavily on the

datacenter interconnect

Evaluation metrics for datacenter interconnects

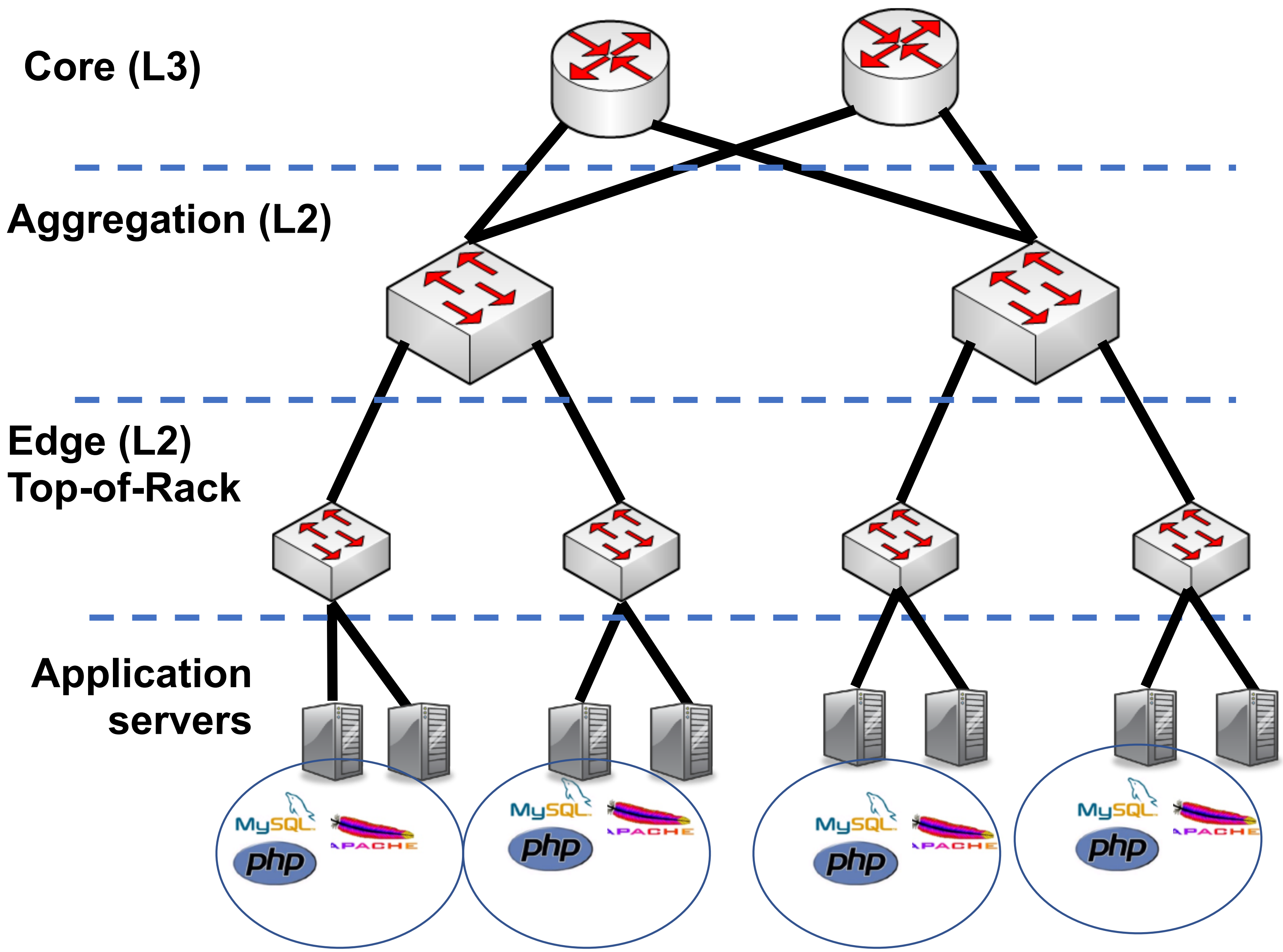
- **Diameter** –
  - Definition: max #hops between any 2 nodes
  - Importance: Speed-of-light latency (why not observed latency?)
    - Answer: queueing delays dependent on traffic as well
- **Bisection Width** –
  - Definition: min #links cut to partition network into 2 equal halves
  - Importance: Fault tolerance
- **Bisection Bandwidth** –
  - Definition: min bandwidth between any 2 equal network halves
  - Importance: Bandwidth bottleneck
- **Oversubscription** –
  - Definition: ratio of worst-case achievable aggregate bandwidth between end-hosts to total bisection bandwidth

Legacy Interconnects



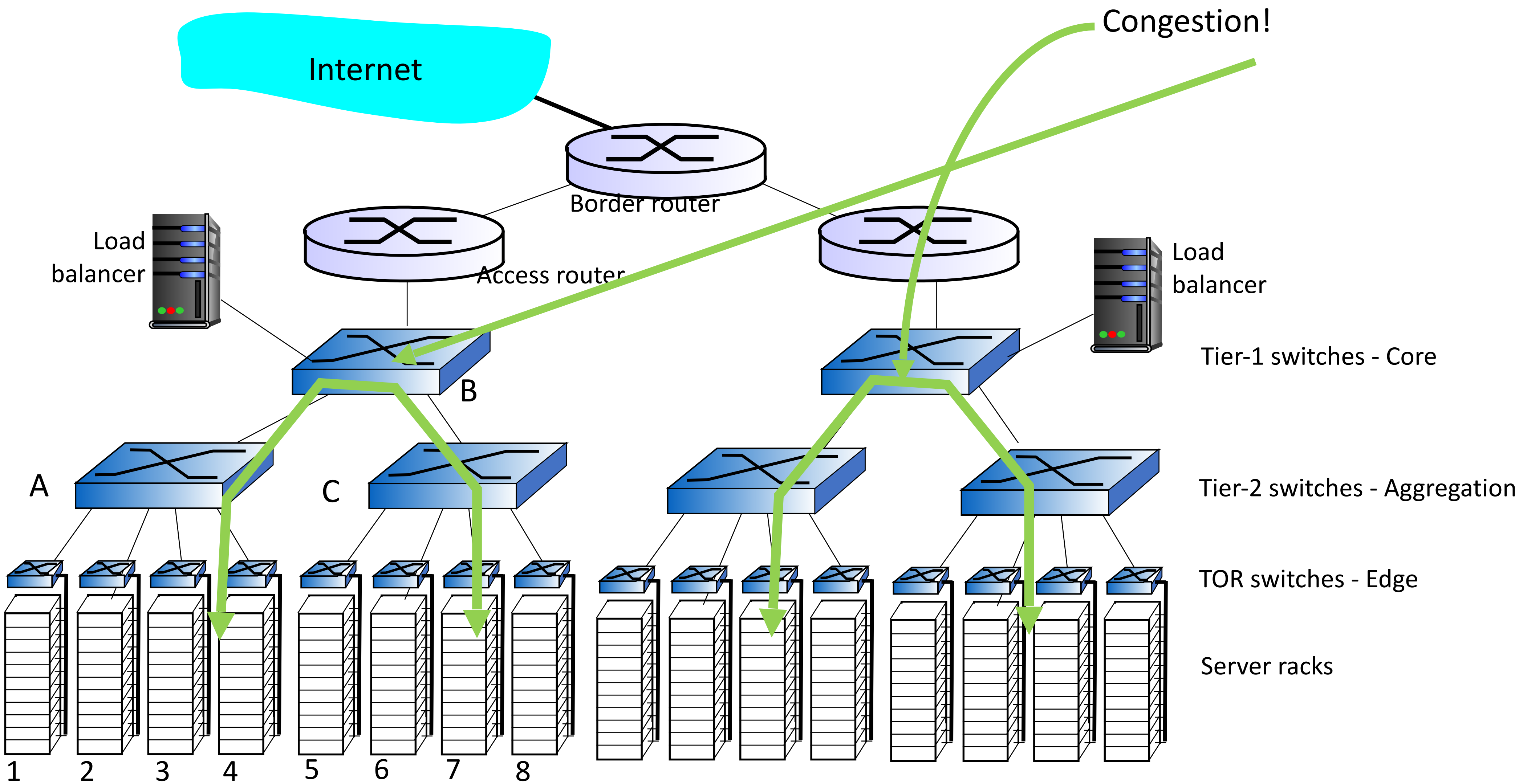
Diameter, Bisection Width, Bisection Bandwidth, Oversubscription

Canonical Datacenter Interconnect



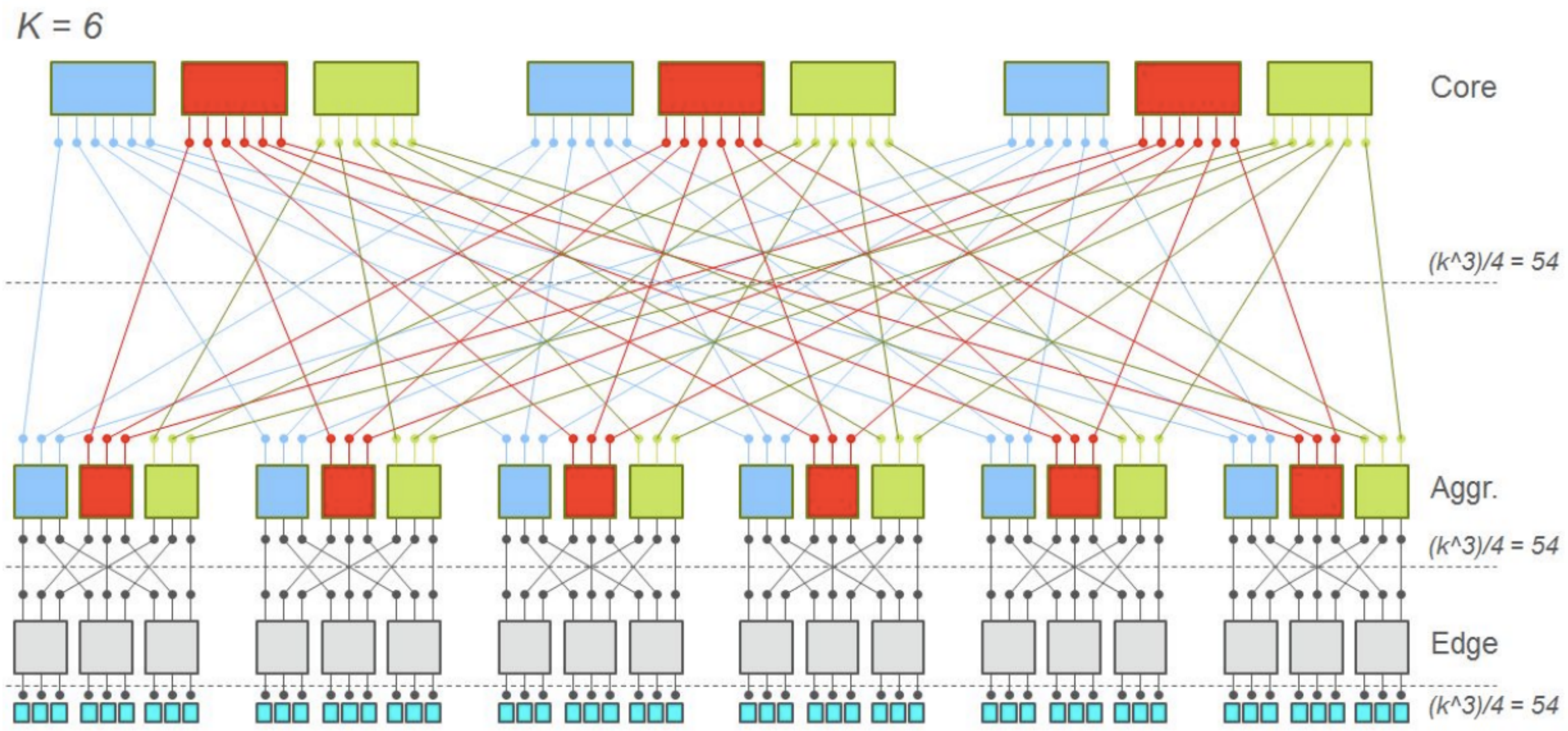
Diameter, Bisection Width, Bisection Bandwidth, Oversubscription

What a real Datacenter Interconnect really looks like?



Diameter, Bisection Width, Bisection Bandwidth, Oversubscription

Modern Datacenter Interconnect



Diameter, Bisection Width, Bisection Bandwidth, Oversubscription

Case study:

The evolution of Google's datacenter interconnect

Google datacenter interconnect principles

- High bisection bandwidth and graceful fault tolerance
- Low Cost
- Centralized control

Centralized Control

- **Custom control plane**
  - Existing protocols did not support routing along multiple paths
  - Protocol overhead of running distributed protocols on large scale
  - Easier network manageability
- Treat the network as a **single switch with O(10,000) ports**
- Anticipated some of the principles of **Software Defined Networking**

Issues

High congestion as utilization approached **just 25%**

- Bursty flows
- Limited buffer on commodity switches
- Intentional oversubscription for cost saving
- Imperfect flow hashing

Congestion Solutions

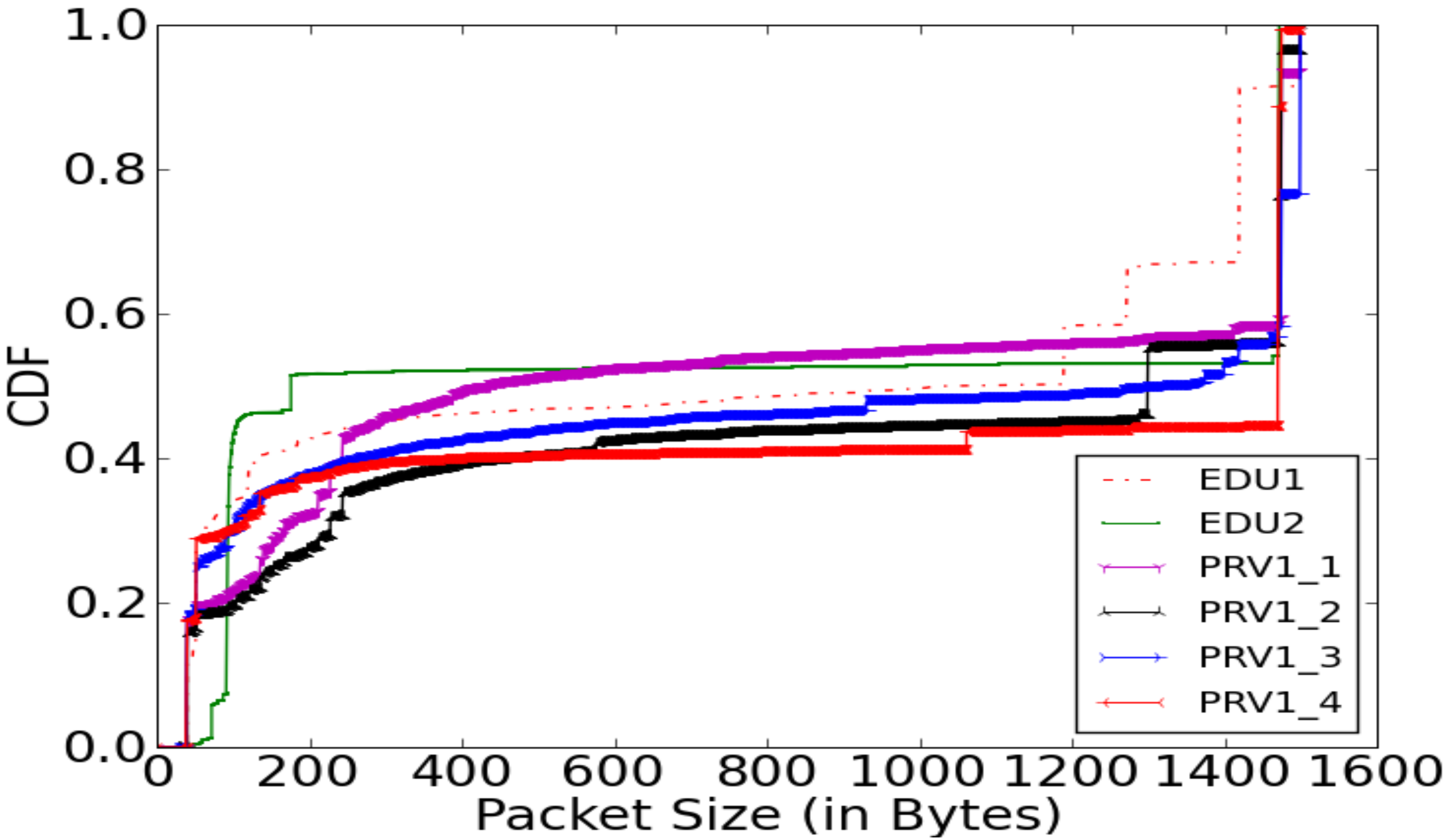
We will see more later

- Configure switch hardware schedulers to drop packets based on QoS
- Tune host congestion window
- Explicit Congestion Notification
- Dynamic buffer sharing on merchant silicon to absorb bursts
- Carefully configure switch hashing to support ECMP load balancing

Case study:

The traffic in Google's datacenter interconnect

Packet Size Distribution



Any interesting observations?

Observations from the Interconnect

- **Link utilization low at edge and aggregate level**
- **Core most utilized**
    - Hot-spots exists (> 70% utilization)
    - < 25% links are hotspots
    - Loss occurs on less utilized links (< 70%)
        - Implicating momentary bursts
- **Time-of-Day variations exists**
    - Variation an order of magnitude larger at core

Insights from the Interconnect

- **75% of traffic stays within a rack (Clouds)**
  - Applications are **not uniformly** placed
- **Half packets are small (< 200B)**
  - **Keep alive integral** in application design
- **At most 25% of core links highly utilized**
  - Need effective routing algorithms to reduce utilization
  - Load balance across paths and migrate applications
- **Questioned popular assumptions**
  - Do we need more bisection? **No**
  - Is centralization feasible? **Yes**

What is REALLY different when compared to the Internet?

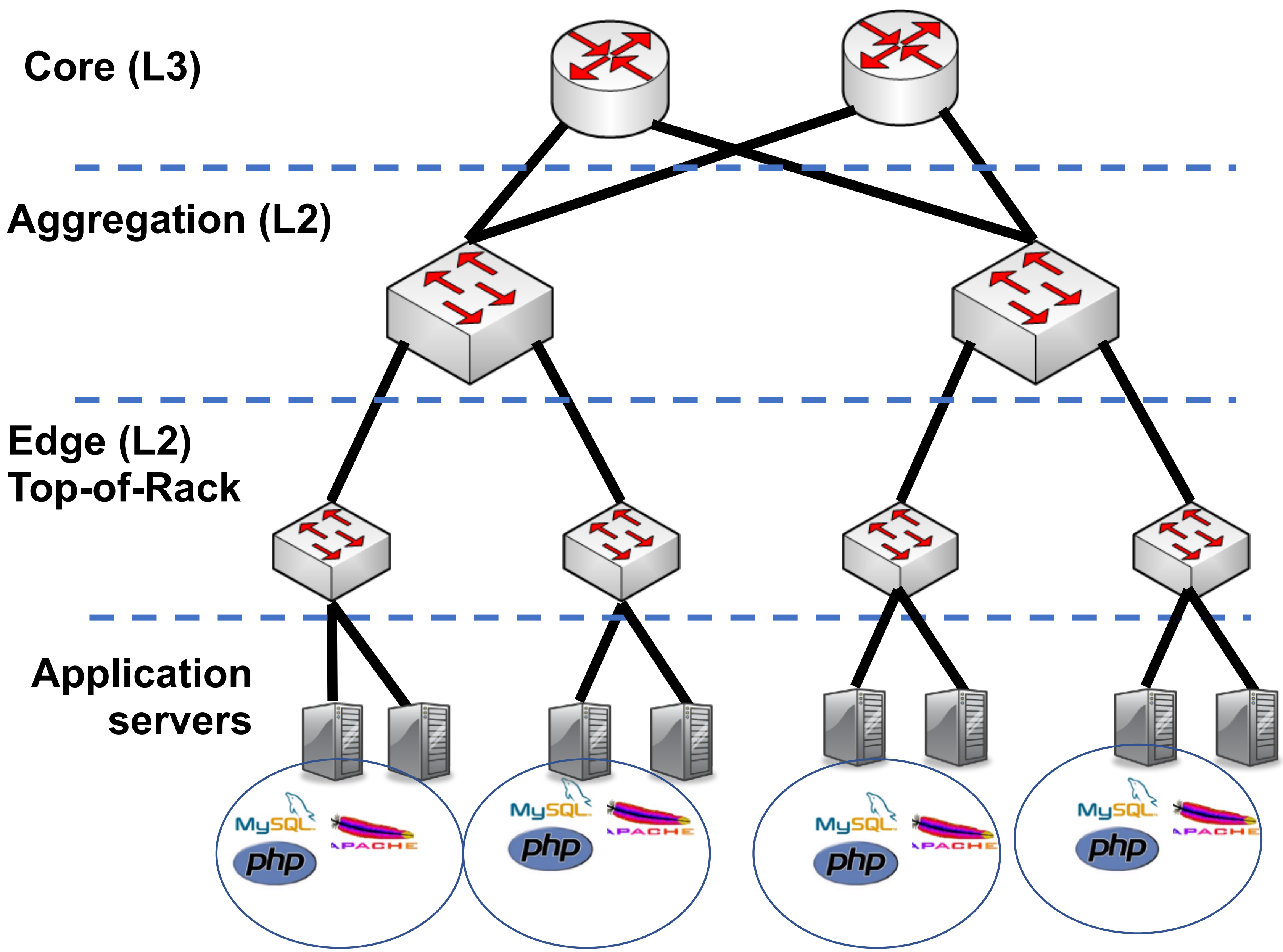
What is REALLY different from the Internet

- **Single entity**
  - **Google owns everything, from the OS to the network hardware**
  - Discussion: how could we exploit this property?
- **Link Layer?**
  - We never exploited anything about "ASes" in link layer
- **Network Layer?**
  - Do we still **need** BGP?
  - Could we still **use** BGP?
- **Transport Layer?**
  - A lot of failure modes of TCP go away (OS owned by Google)
  - Is TCP still a good solution?
- **<u>Reality:</u> Increasingly less separation between link and network layers**

What is REALLY different from the Internet

- **Fixed (structured) topology, complete control and knowledge**
  - **The topology is designed, owned, and managed by Google**
  - Discussion: how could we exploit this property?
- **Link Layer and Network Layer**
  - More efficient algorithms for route computation

Example: Simplification of Routing



Can you think of a simple mechanism?

What is REALLY different from the Internet

- **Fixed (structured) topology, complete control and knowledge**
  - **The topology is designed, owned, and managed by Google**
  - Discussion: how could we exploit this property?
- **Link Layer and Network Layer**
  - More efficient algorithms for route computation
  - Could “bake in” routing **results** into switch routing tables
  - Software-defined networks, centralized control
  - **Other benefits:**
    - Better control over “load balancing”
    - Avoid convergence issues (but new issues come up)
- **Transport Layer?**
  - We never made any assumptions about topology in L4 design
  - Is TCP still a good idea?

What is REALLY different from the Internet

- **Small-scale, within a single geographic location**
  - **The entire datacenter is may be 1M machines, in a single location**
  - Discussion: how could we exploit this property?
- **Link Layer and Network Layer?**
  - Another motivating factor for centralized control
  - Routes can be computed and "installed" quickly
- **Transport layer?**
  - Next slide …

What is REALLY different from the Internet

- **Tiny round trip times**
  - **Less than 5 microseconds (for a single packet)**
  - Discussion: how could we exploit this property?
- **Link Layer and Network Layer?**
  - Millisecond-level convergence times no longer “sufficient”
  - Even more motivation for software-defined, centralized control
- **Transport layer?**
  - Most flows small; can be completed within a couple of RTT
  - Even 3-way hand-shake takes 7.5microseconds
  - TCP is not going to work well!

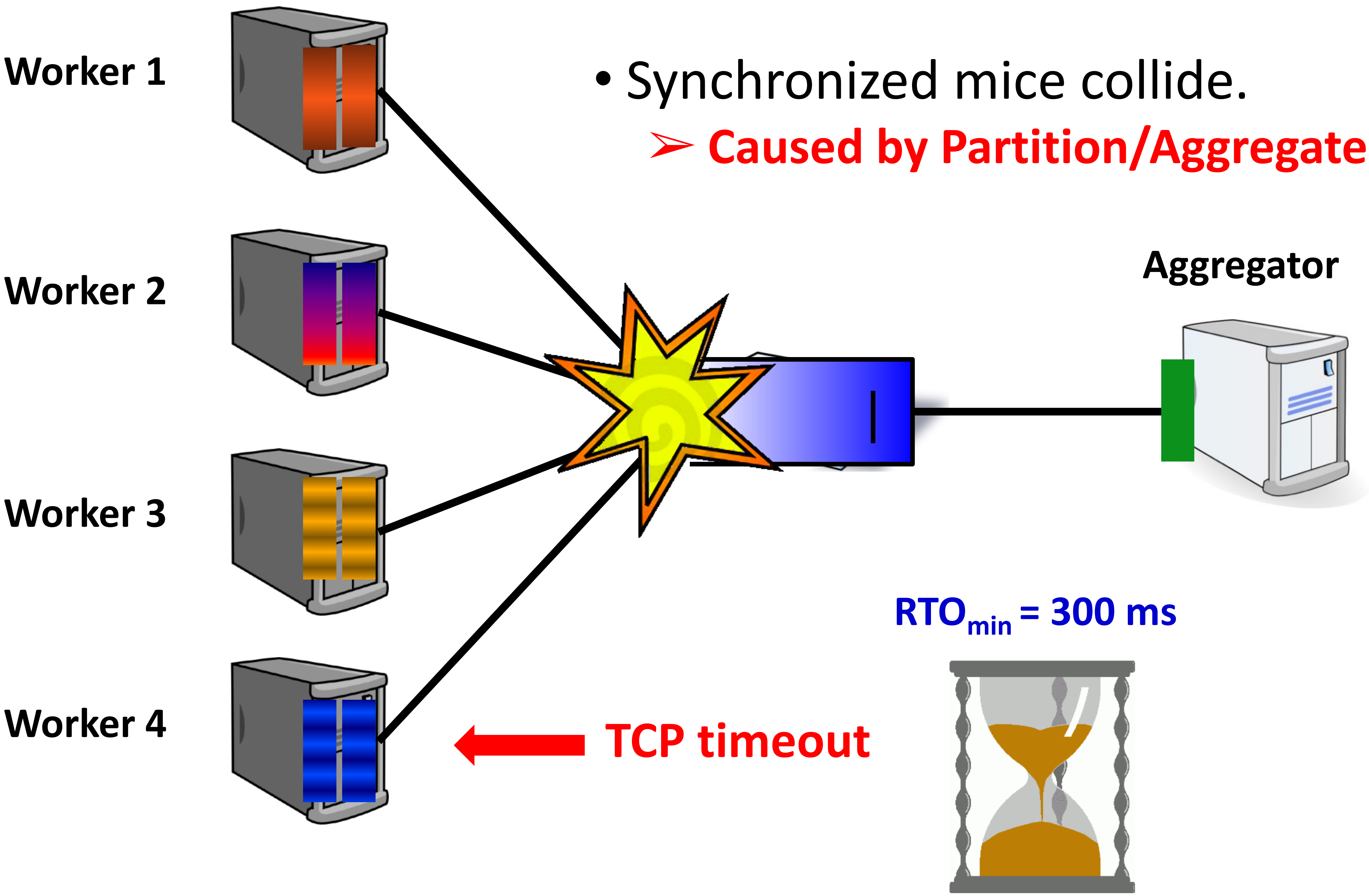
Datacenter Transport Design:
One of the most active research areas

TCP in datacenter context

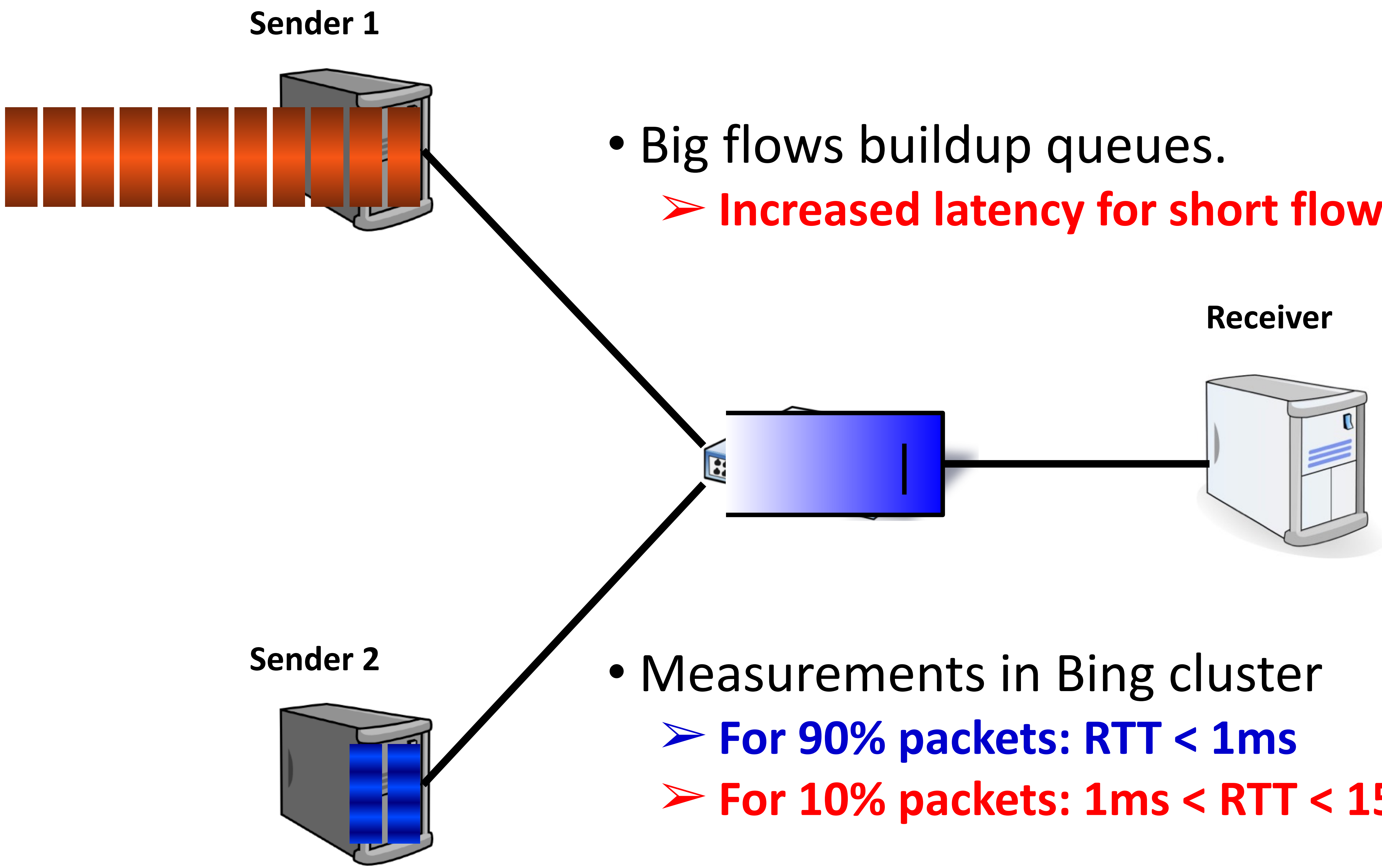
- **TCP is too inefficient**
    - Three-way handshake takes **too long**
    - Does not work well with **short flows**
    - Not designed for **low latency**
    - Has no notion of **deadlines**
    - **Does NOT work well with "Incast"**
    - **Queue build-up due to long flows; short flows suffer**

Incast

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



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**

TCP in datacenter context

- **TCP is too inefficient**
  - Three-way handshake takes **too long**
  - Does not work well with **short flows**
  - Not designed for **low latency**
  - Has no notion of **deadlines**
  - **Does NOT work well with “Incast”**
  - **Queue build-up due to long flows; short flows suffer**
- **How would you solve these problems?**