Data Centers and SDNs

What are Data centers?

- What does a typical data center look like?
- How are data centers different from other enterprise networks?
- How should the network protocols adapt to these differences?

Why data centers?

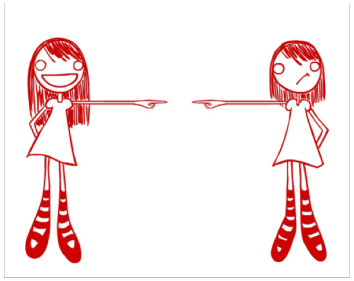
Before

- "Cloud companies" such as Google operated huge, sophisticated clusters
- To expand, they would **"scale up"**

Problems

- Starts to get expensive
- Cannot scale after a point

The Internet Made Me Do It



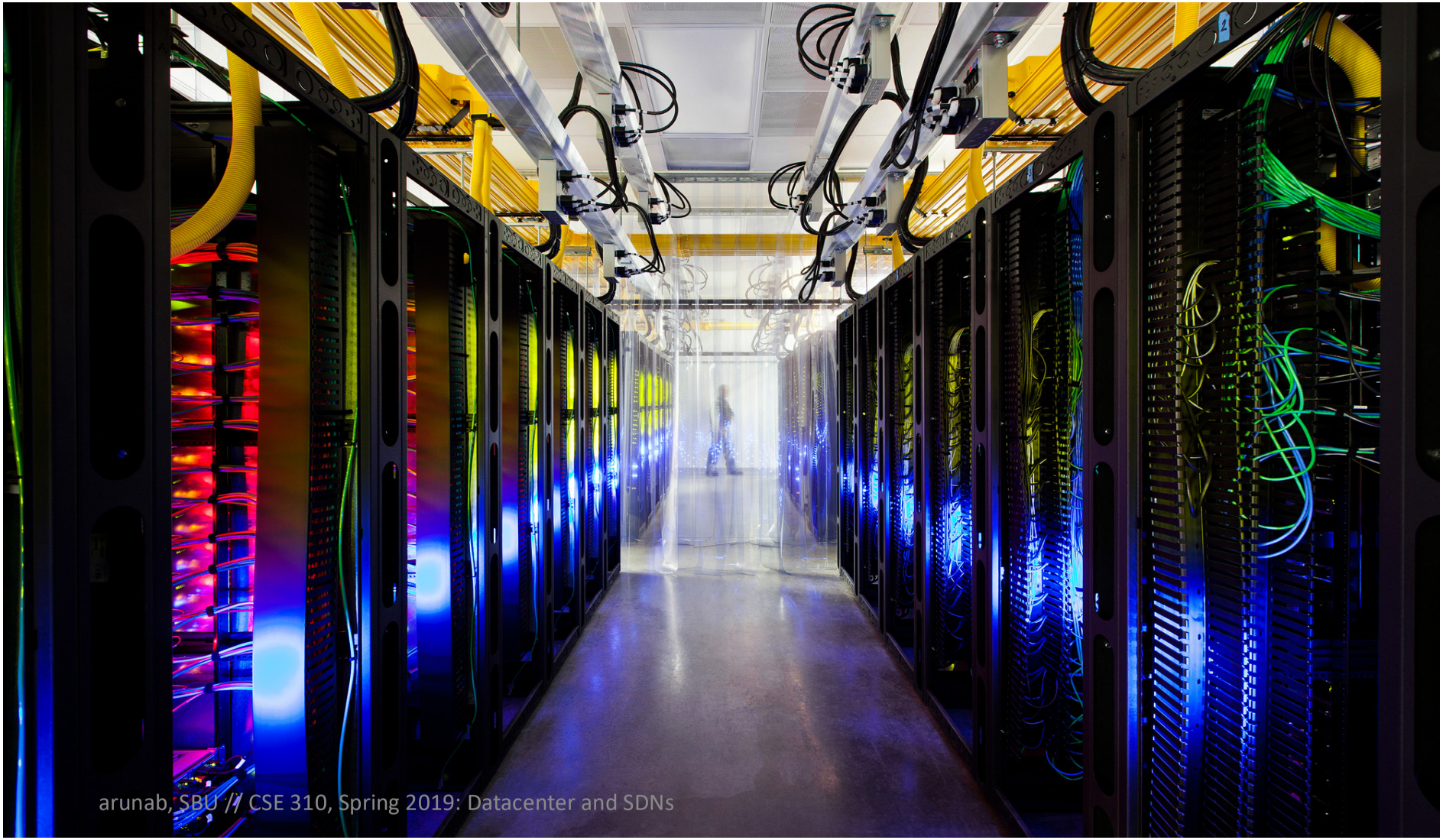
- Everyone wants to operate at Internet scale
  - Millions of users
  - Zetabytes of data to analyze
    - Webserver logs
    - Advertisement clicks
    - Social networks, blogs, Twitter, video…
- Not everyone has the expertise to build a cluster

.

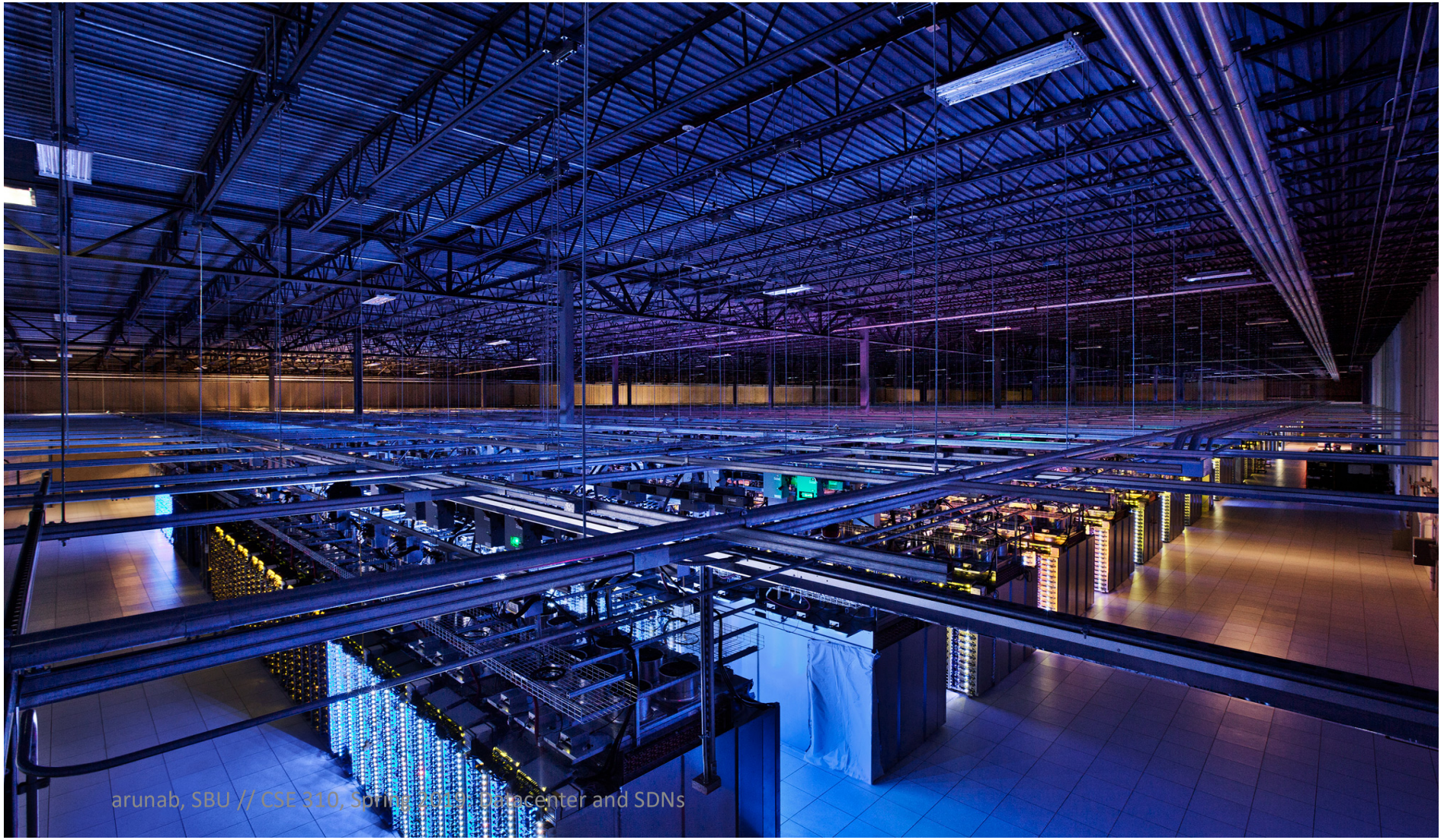
Today's data center model

- **Cheap, off the shelf, commodity parts**
  - **scale out**
  - Run standard software
- **Multi-tenancy**
  - Small companies "share" data centers

Google datacenter

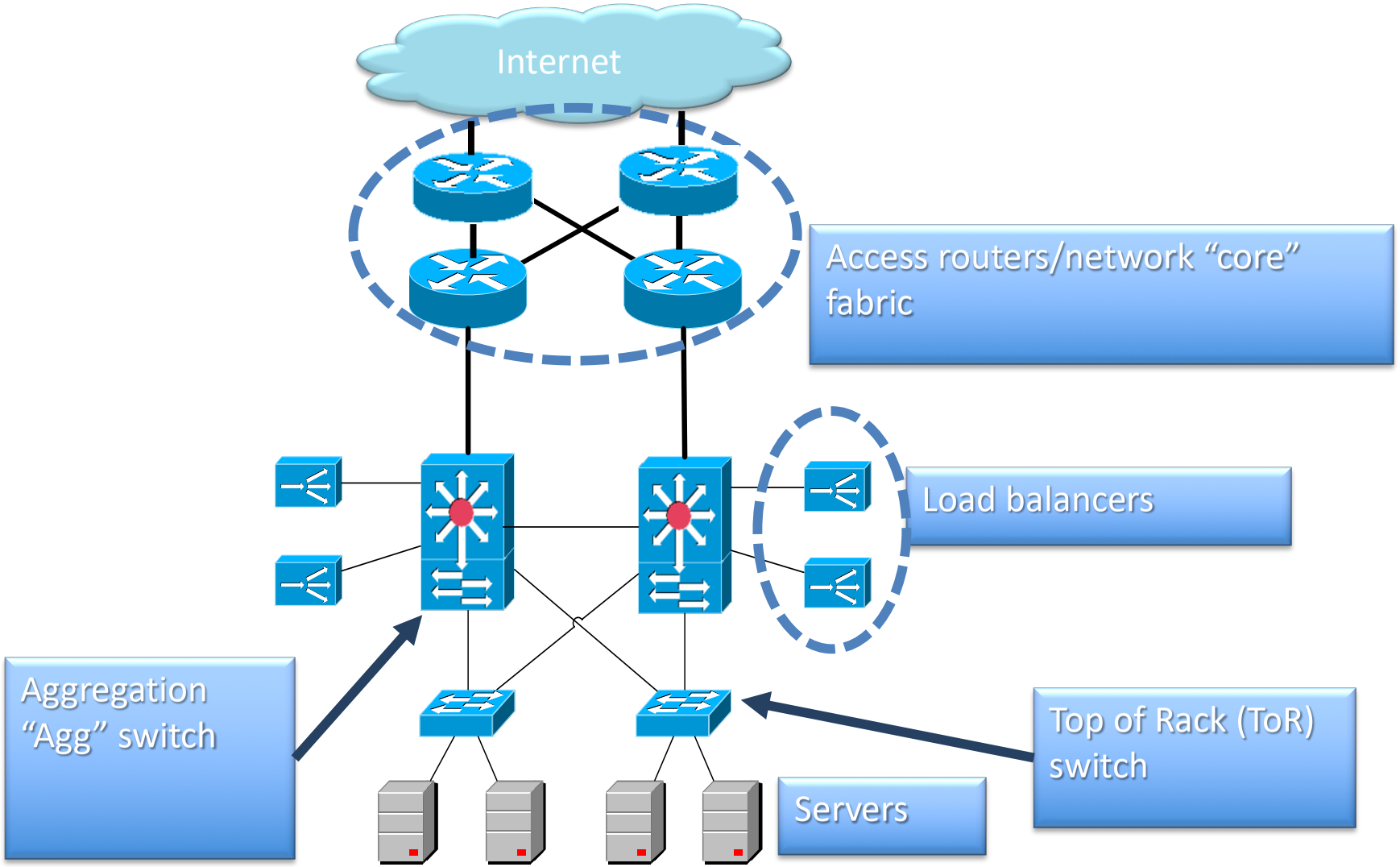


Google datacenter



Cooling unit

Datacenter networks



The one problem: OMG the wires

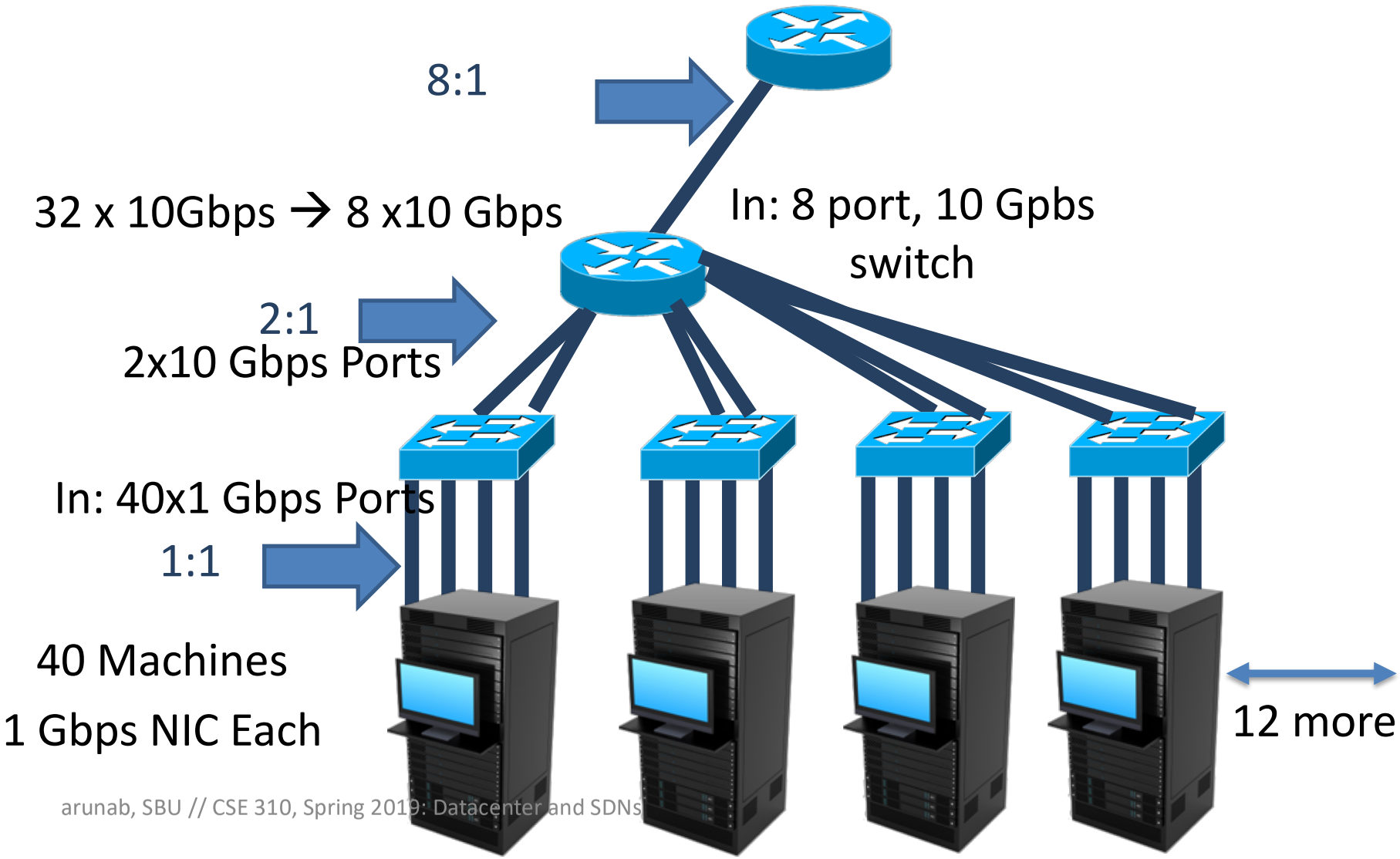
Networking problems in Datacenters: Oversubscription

- Datacenters are **data intensive**
- Most hardware can handle this
  - CPUs scale with Moore's Law
  - RAM is fast and cheap
  - RAID and SSDs are pretty fast
- Current networks cannot handle it
  - Slow, not keeping pace over time
  - Requires specialized switches that are very expensive

Oversubscription problem

- Datacenter requires full bisection bandwidth, but networking infrastructure is expensive
  - 10 GB server NICs are becoming more affordable, but 128 port 10 GB routers are not
  - $7000 for 48 port 1 GB switch, $700,000 for 128 port 10 GB switch
- Many of the routers have to be configured by hand, so cannot adapt to traffic

Oversubscription Example



Consequences of Oversubscription

- Oversubscription limits scalability: Sometimes, the oversubscription as high as 240:1

- Problem is about to get worse
  - 10 GigE servers are becoming more affordable
  - 128 port 10 GigE routers are not

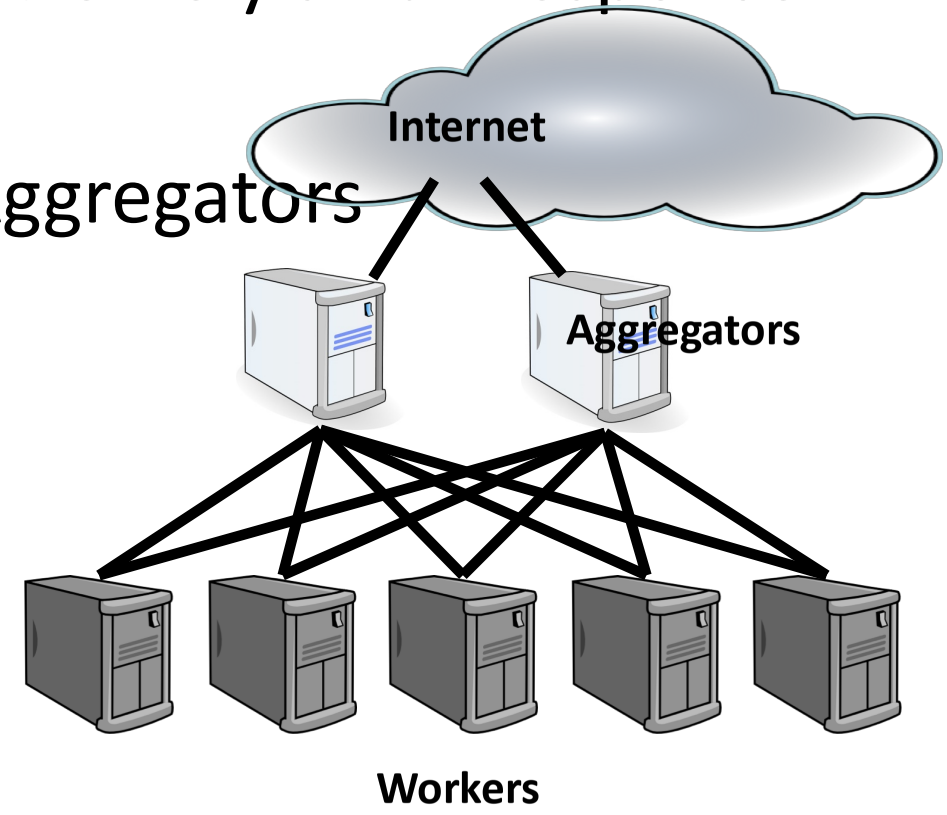
How to solve oversubscription?

- Use a large number of “commodity” switches to solve the oversubscription problem

- Example in class.

Second problem: TCP

- Datacenters need extremely small response time
- Example: Workers/Aggregators



Second problem: TCP

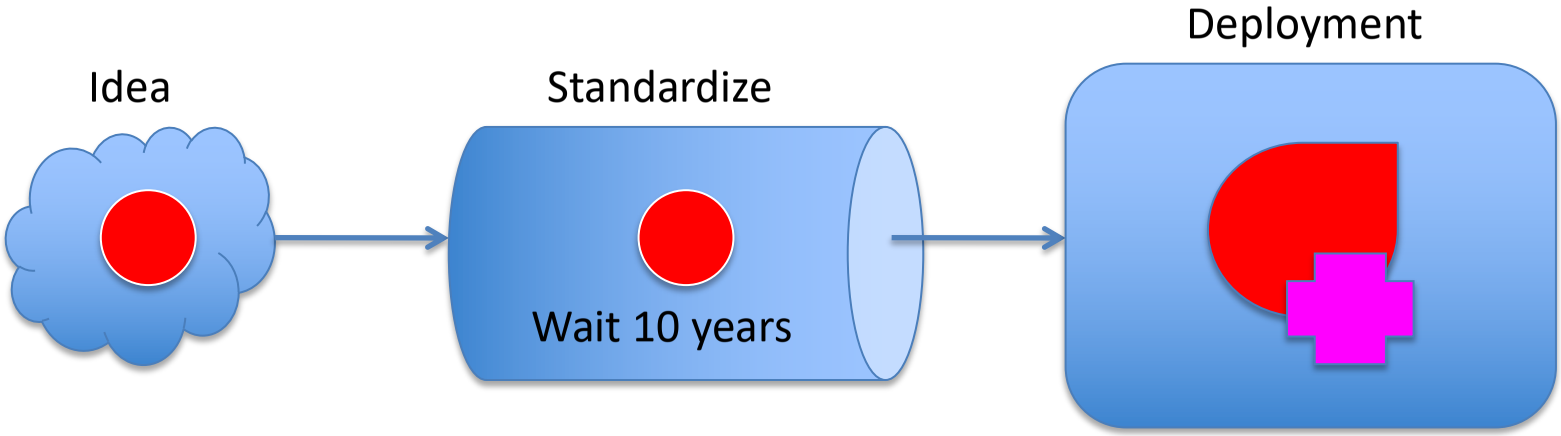
- Most of the traffic is within the datacenter
  - Most flows are short, from workers to aggregators
  - However, the latency expectation is very small (10s of milliseconds)
  - Datacenter RTTs are in the order of microseconds
- Existing TCP cannot handle this traffic

So what do we do about it?

- New TCP called DCTCP to support fan out
- DCTCP requires changes to the routers in the data centers
  - This design is not well-suited for Internet were routers cannot be modified easily.

Programmable networks

Why programmable networks?



- L3 innovations driven by vendors
- Consumers largely locked out
- Lowest common denominator features
- Glacial innovation

What about hardware switches/routers

- The data plane and the control plane on routers (and switches) are combined
- CISCO has pre-defined network algorithms that are hard to change

Why Programmable network is hard?

- Because in most cases forwarding data plane is in hardware
  - If you make the network programmable in software, then it will be too slow
  - If it remains in hardware, then it is not easy to program

OpenFlow: An example programmable network

- Key innovation: Use the same hardware for switching/forwarding, but use software for control

- Key idea: Allow software to manipulate the flow table, but use hardware to switch using the flow table

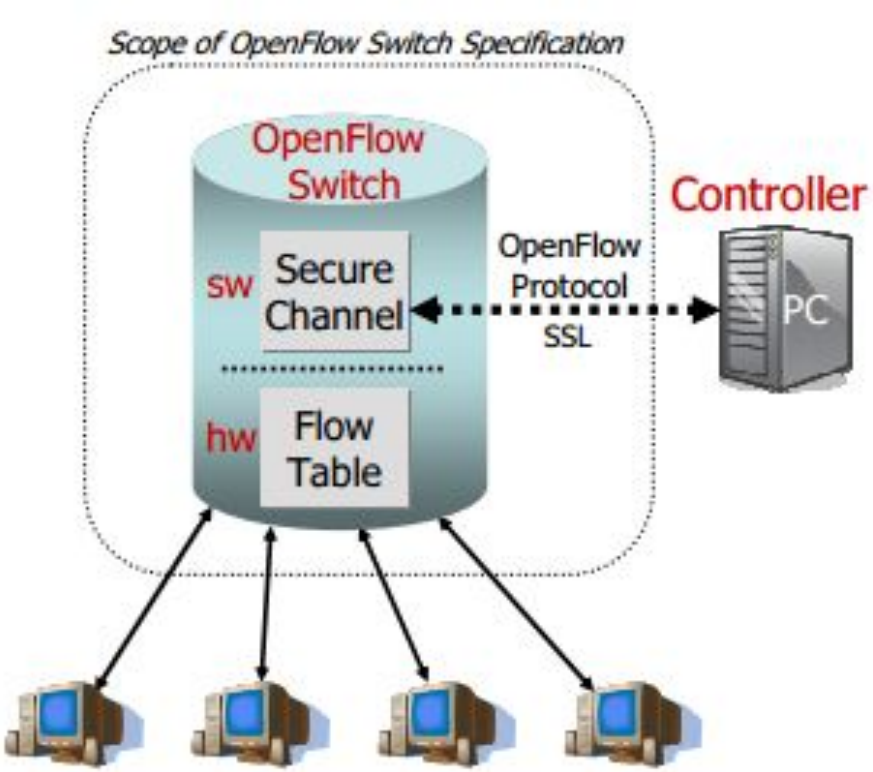


Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

Combine Programmable Network with a Network Operating System (NOX)

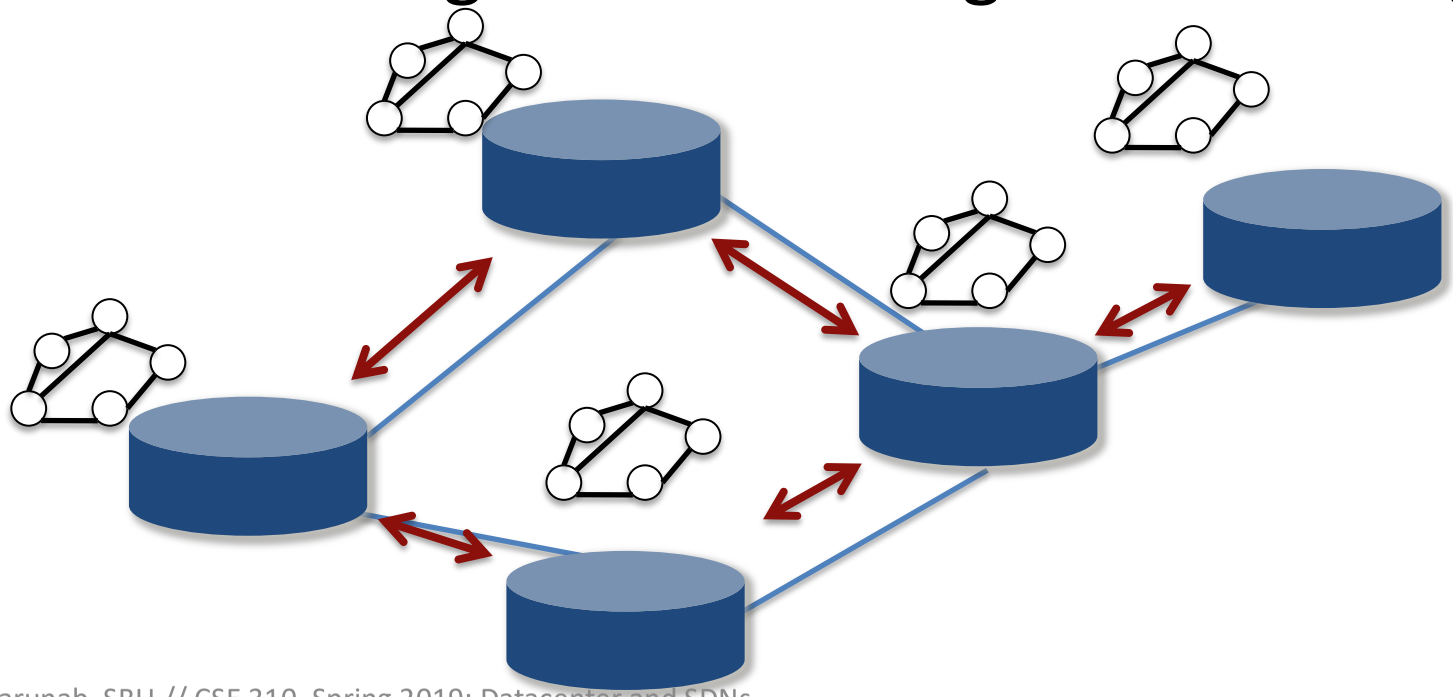
There are no control plane abstractions

- Control plane tasks: routing, isolation, traffic engineering
- All done from scratch
- Programming language analogy

Traditional Control Mechanisms

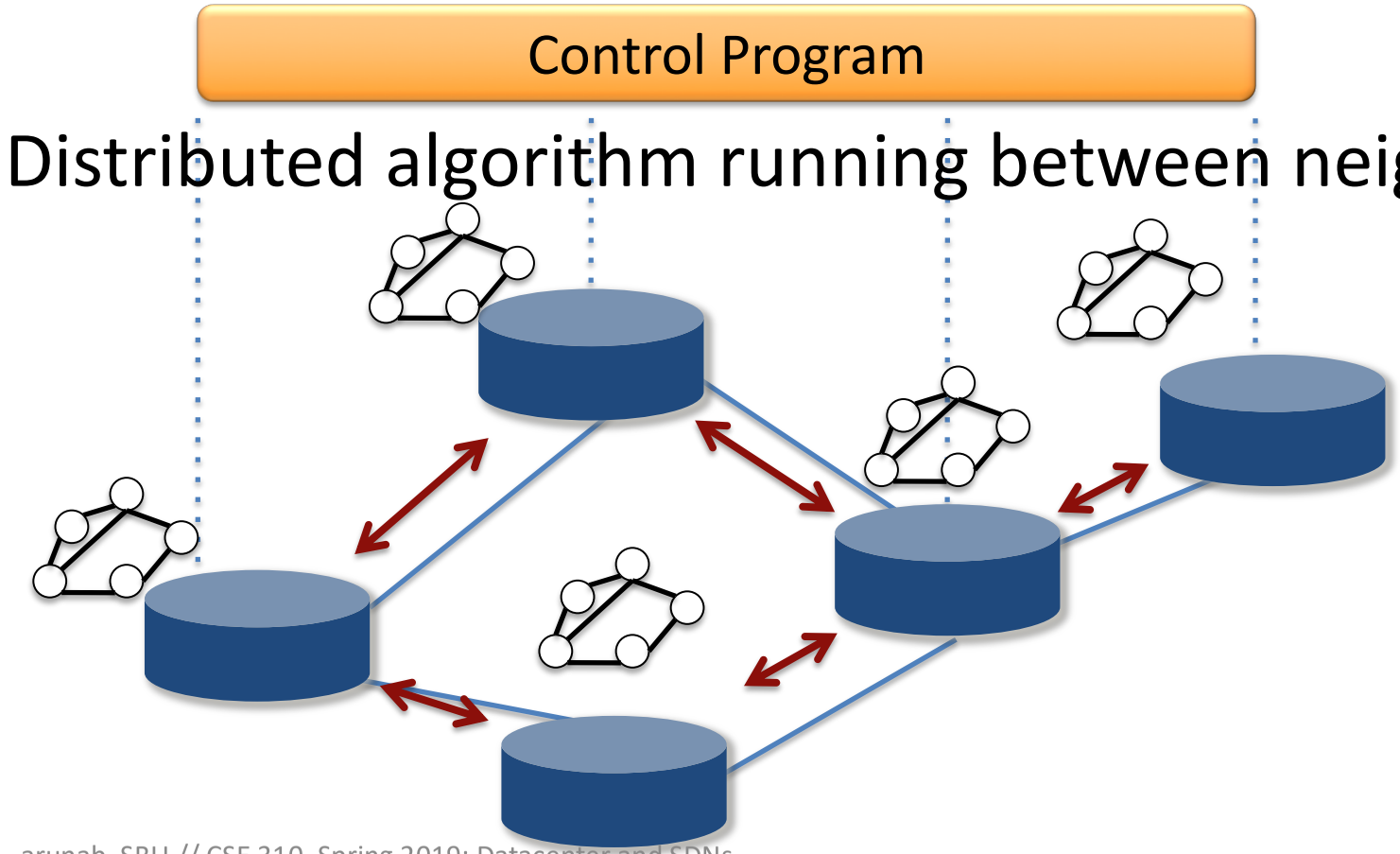
e.g. routing, access control

Distributed algorithm running between neighbors

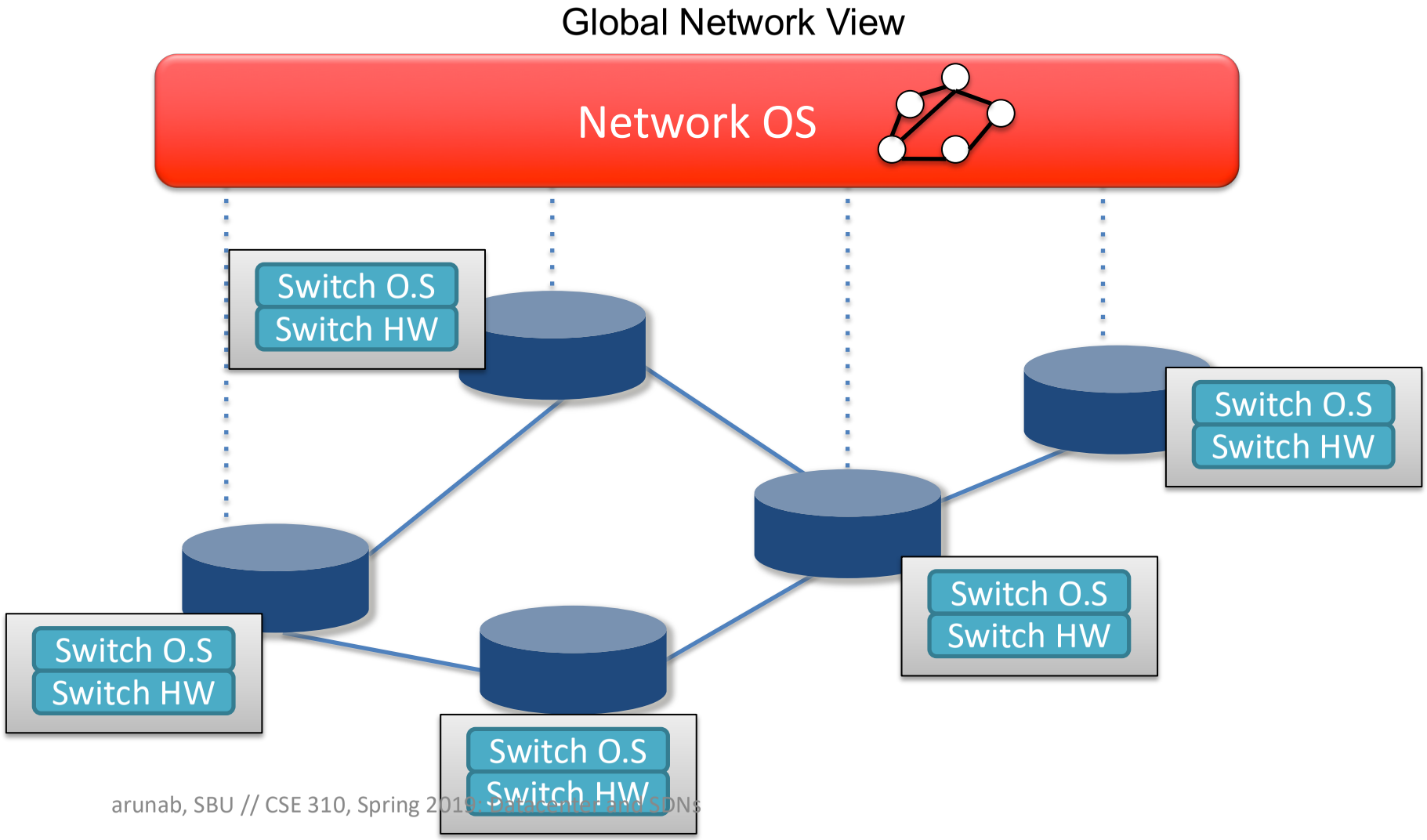


Instead abstract a control program

e.g. routing, access control



NOX + OpenFlow



What does NOX do?

- Abstracts routing layer functionality
  - Gets topology information SDN is an architecture that allows network operators to:
  - Abstract control plane functionality
  - Make the network more programmable

- OpenFlow + Nox is one mechanism that will let you realize the SDN architecture

So, What is SDN?

SDNs

- SDN is an architecture that allows network operators to:
  - Abstract control plane functionality
  - Make the network more programmable
- OpenFlow + Nox is one mechanism that will let you realize the SDN architecture