

CS3640

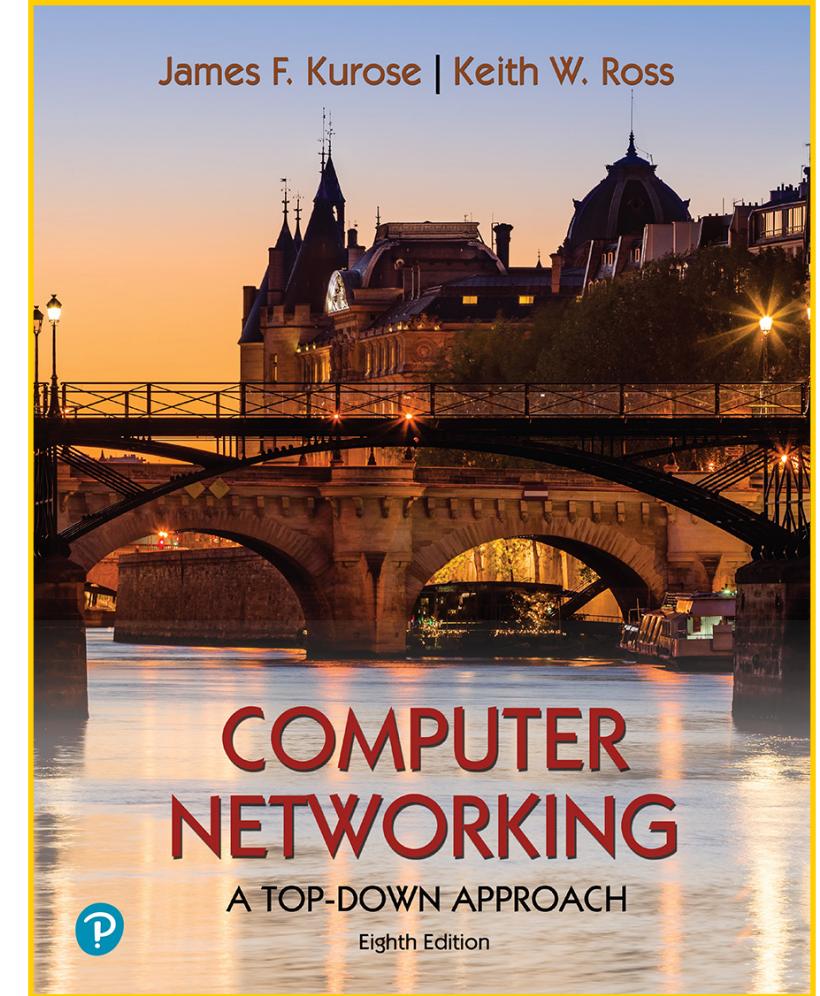
Research: Software Defined Networking

Prof. Supreeth Shastri
Computer Science
The University of Iowa

Lecture goals

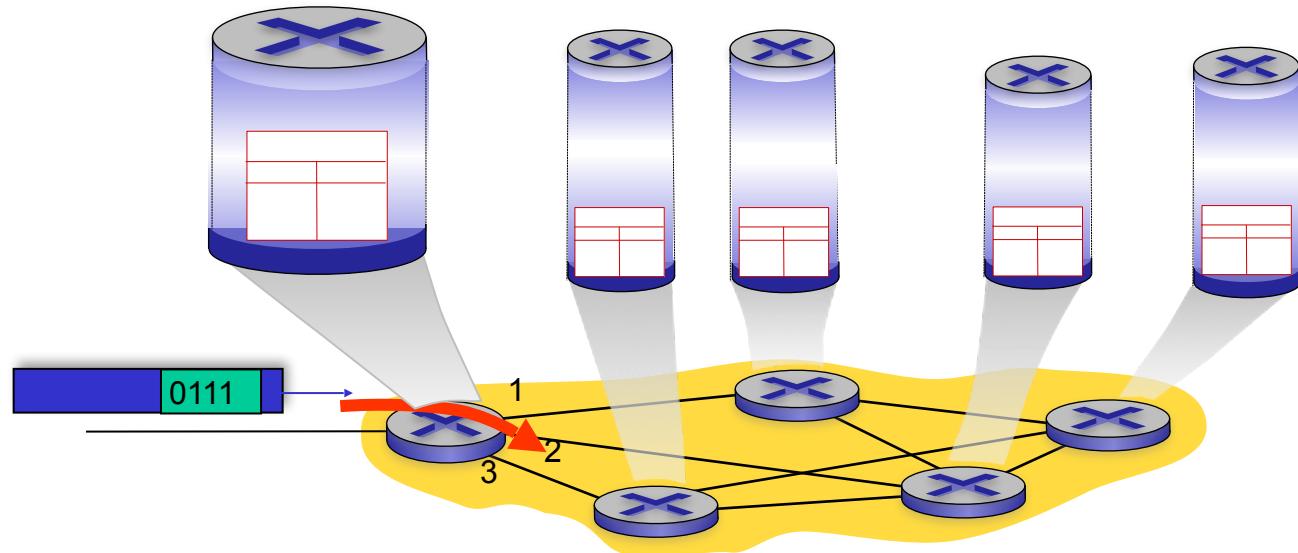
Evolution of core networking platform from being pre-programmed and hardware-driven to flexible and software-driven

- *Software Defined Networking*
- *OpenFlow*
- *Case study: Google SDN*



Chapters 4.4, 5.5

Previously on CS3640...



*Each router contains a **forwarding table** and uses it to implement **destination-based** forwarding of IP datagrams*

Research question: *Could this abstraction be generalized?*

- ▶ IP datagram header contain many fields (in addition to destination address)
- ▶ Many actions could be taken on the packet: forward, drop, copy, modify, log, etc.

New Abstraction: match-plus-action

Flow: defined by header field values (in link-, network-, transport-layer fields)

Generalized Forwarding: simple packet-handling rules

- *match: pattern values in packet header fields*
- *actions: drop, forward, modify or send to controller*
- *priority: disambiguate overlapping patterns*

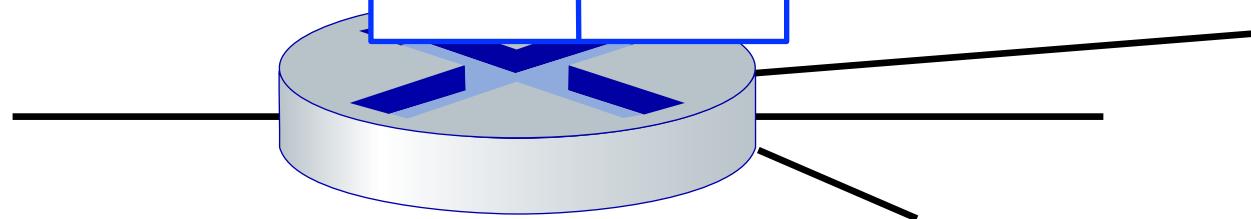
an example flow table

Flow table defines
the router's **match-
plus-action** rules

| Flow table | |
|------------|--------|
| match | action |
| | |

src = *.*.*.* , dest=3.4.*.*
src=1.2.*.* , dest=*.**.*.*
src=10.1.2.3 , dest=*.**.*.*

forward(2)
drop
send to controller



OpenFlow

How it started

OpenFlow: Enabling Innovation in Campus Networks

Nick McKeown
Stanford University

Tom Anderson
University of Washington

Hari Balakrishnan
MIT

Guru Parulkar
Stanford University

Larry Peterson
Princeton University

Jennifer Rexford
Princeton University

Scott Shenker
University of California,
Berkeley

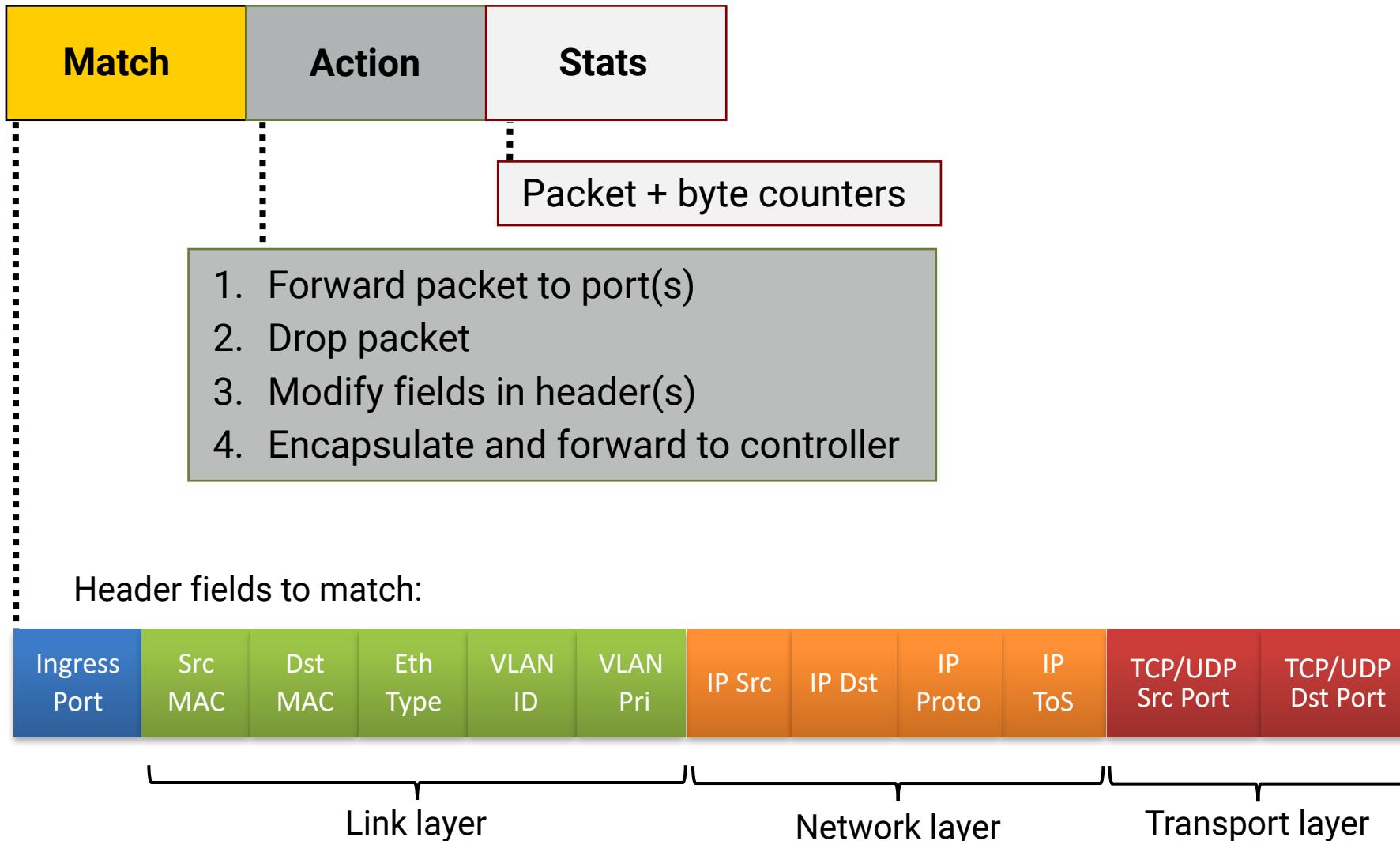
Jonathan Turner
Washington University in
St. Louis

How it's going

**\$32+ Billion Worldwide Software-Defined Networking
Market to 2025 - Featuring Cisco, Huawei & VMware
Among Others**

August 18, 2020 06:33 ET | Source: [Research and Markets](#)

OpenFlow: flow table entries



OpenFlow: Examples

Destination-based forwarding:

| Switch Port | MAC src | MAC dst | Eth type | VLAN ID | VLAN Pri | IP Src | IP Dst | IP Prot | IP ToS | TCP s-port | TCP d-port | Action |
|-------------|---------|---------|----------|---------|----------|--------|----------|---------|--------|------------|------------|--------|
| * | * | * | * | * | * | * | 51.6.0.8 | * | * | * | * | port6 |

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

| Switch Port | MAC src | MAC dst | Eth type | VLAN ID | VLAN Pri | IP Src | IP Dst | IP Prot | IP ToS | TCP s-port | TCP d-port | Action |
|-------------|---------|---------|----------|---------|----------|--------|--------|---------|--------|------------|------------|--------|
| * | * | * | * | * | * | * | * | * | * | * | 22 | drop |

Block (do not forward) all datagrams destined to TCP port 22 (ssh port #)

| Switch Port | MAC src | MAC dst | Eth type | VLAN ID | VLAN Pri | IP Src | IP Dst | IP Prot | IP ToS | TCP s-port | TCP d-port | Action |
|-------------|---------|---------|----------|---------|----------|--------|-------------|---------|--------|------------|------------|--------|
| * | * | * | * | * | * | * | 128.119.1.* | * | * | * | * | drop |

Block (do not forward) all datagrams sent by host 128.119.1.*

OpenFlow: Examples

Layer 2 destination-based forwarding:

| Switch Port | MAC src | MAC dst | Eth type | VLAN ID | VLAN Pri | IP Src | IP Dst | IP Prot | IP ToS | TCP s-port | TCP d-port | Action |
|-------------|---------|-----------------------|----------|---------|----------|--------|--------|---------|--------|------------|------------|--------|
| * | * | 22:A7:23: 11:E1:02 | * | * | * | * | * | * | * | * | * | port3 |

layer 2 frames with destination MAC address 22:A7:23:11:E1:02 should be forwarded to output port 3

Generality of OpenFlow

match-plus-action abstraction unifies different kinds of devices

Router

- **match:** longest IP prefix
- **action:** forward out a link

Firewall

- **match:** IP addr and/or TCP/UDP ports
- **action:** permit or drop

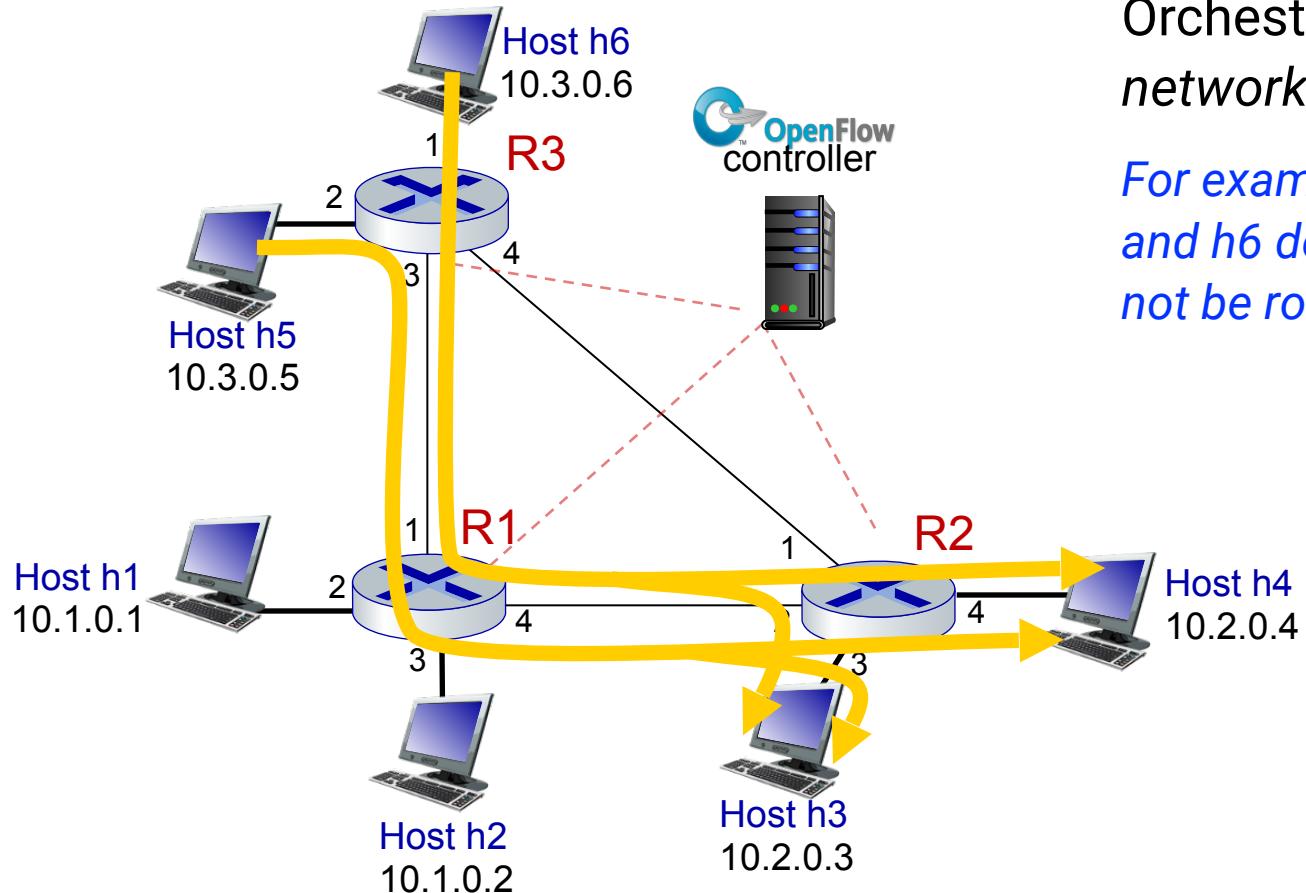
Switch

- **match:** destination MAC
- **action:** forward or flood

NAT

- **match:** IP addr and/or TCP/UDP ports
- **action:** rewrite IP addr and ports

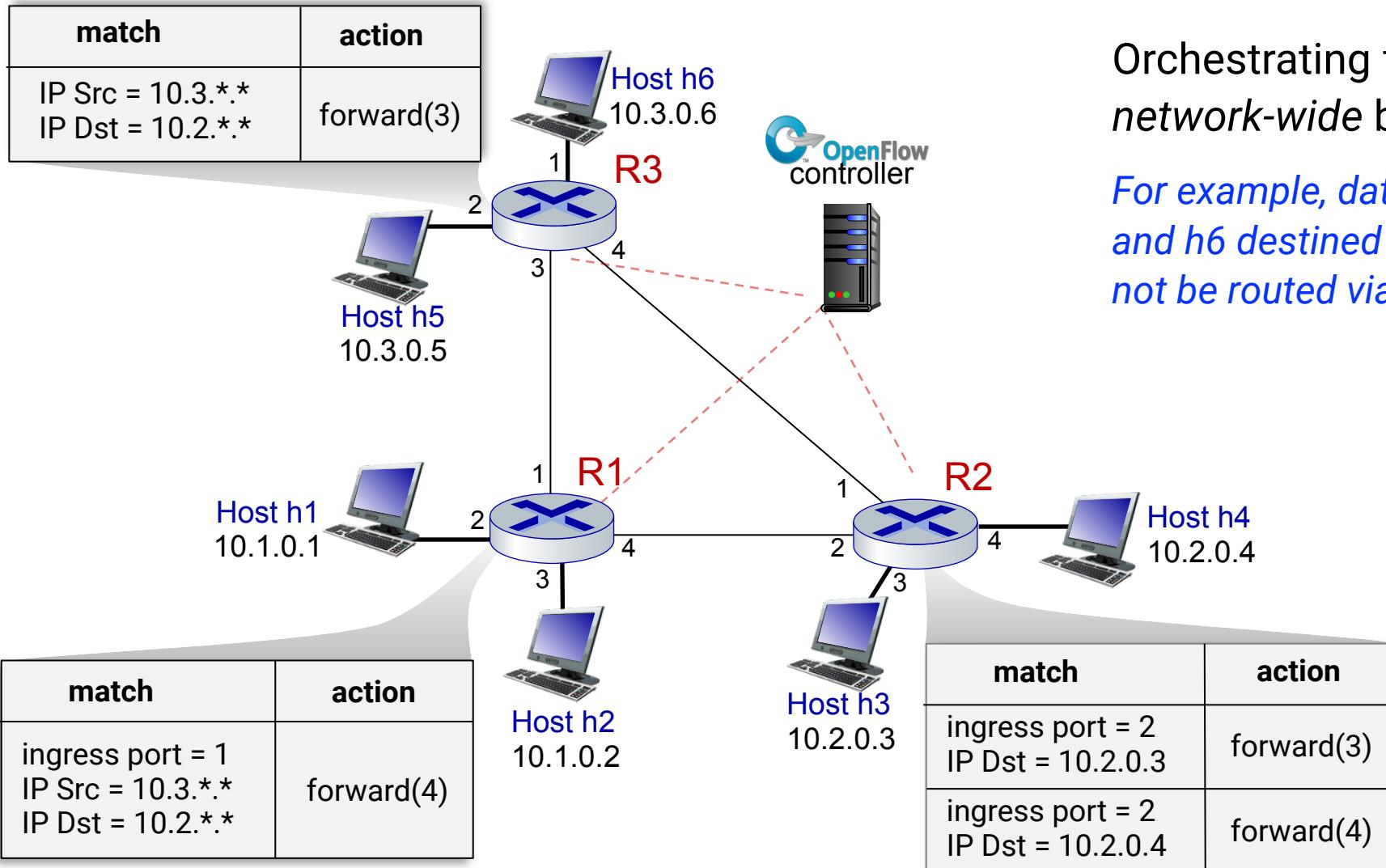
OpenFlow Controller



Orchestrating flow tables to create
network-wide behavior

For example, datagrams from hosts h5 and h6 destined for h3 or h4, should not be routed via direct link R3-R2

OpenFlow Controller

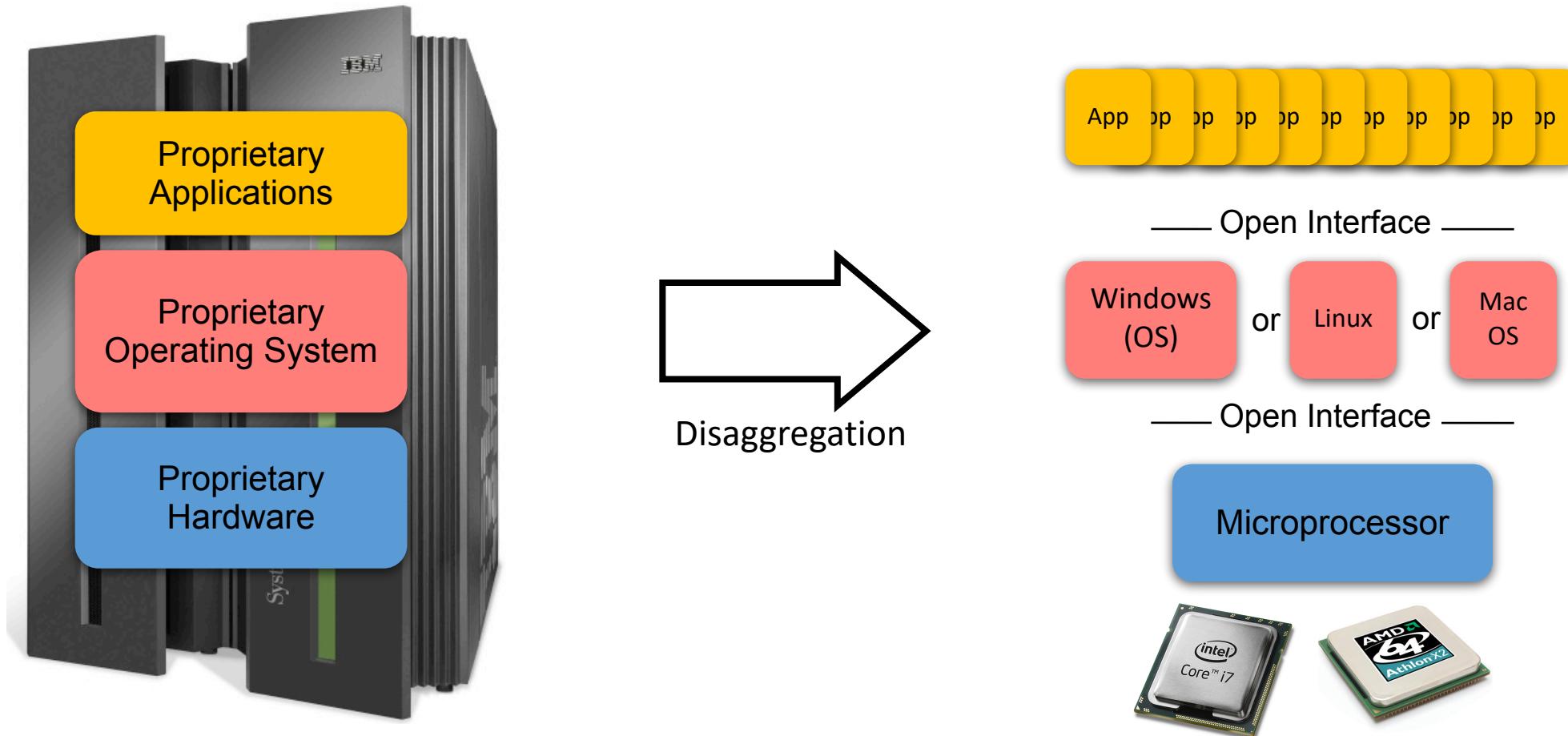


Orchestrating flow tables to create network-wide behavior

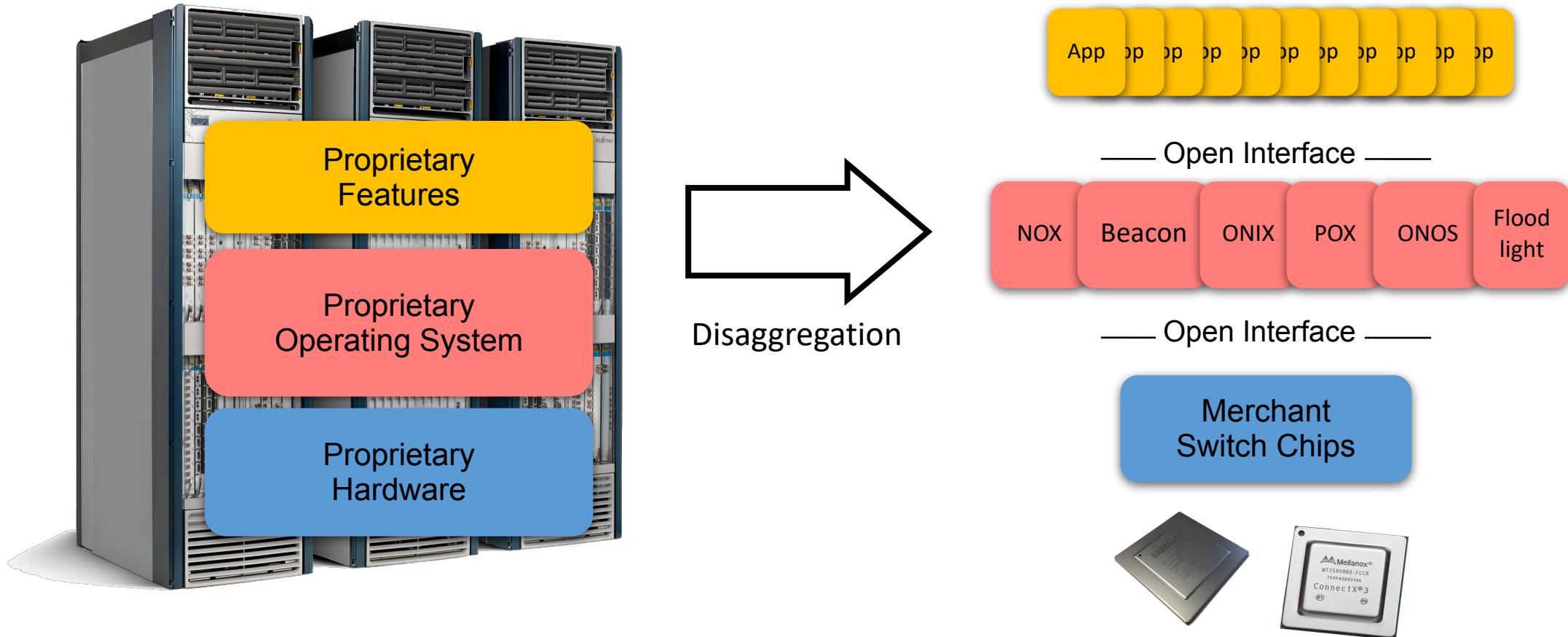
For example, datagrams from hosts h5 and h6 destined for h3 or h4, should not be routed via direct link R3-R2

Google + SDN

Transformation in Computing Industry (circa 1980)



Transformation in Networking Industry (circa 2010)



Google's network footprint

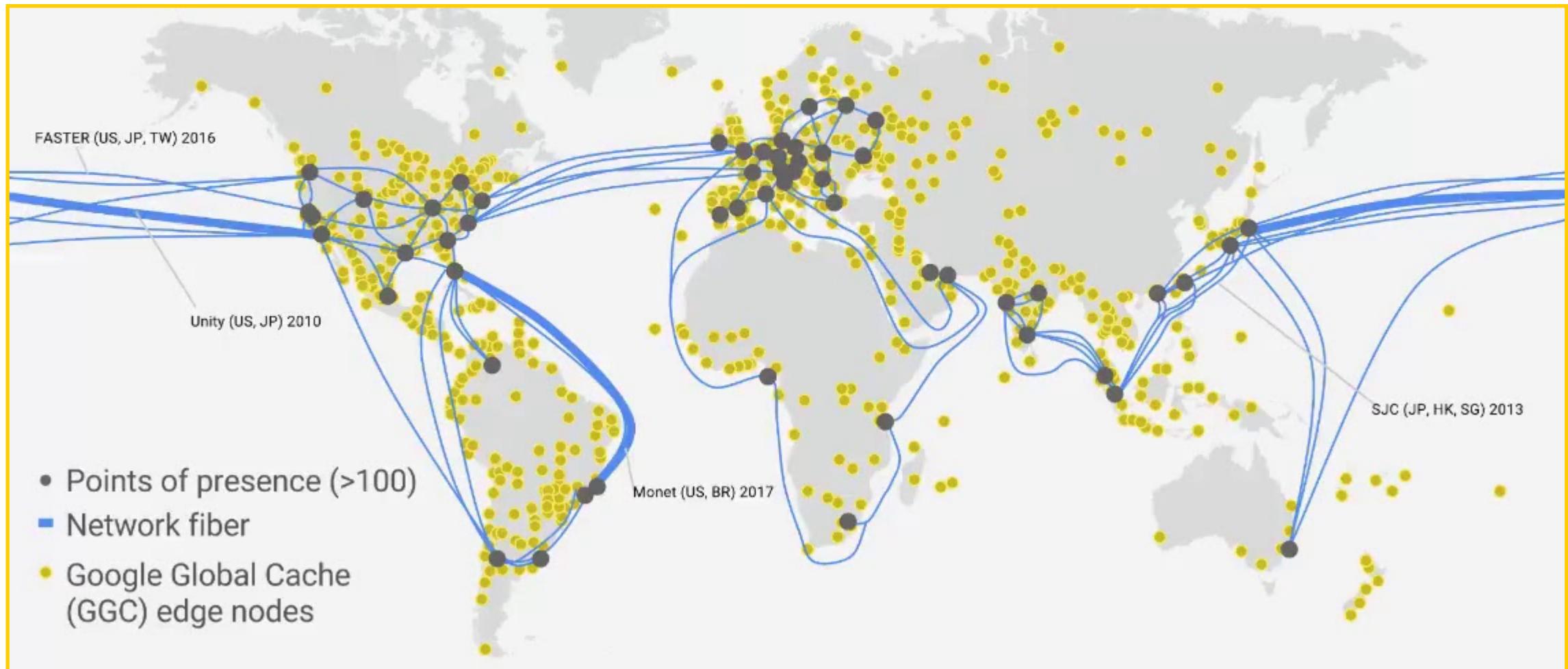
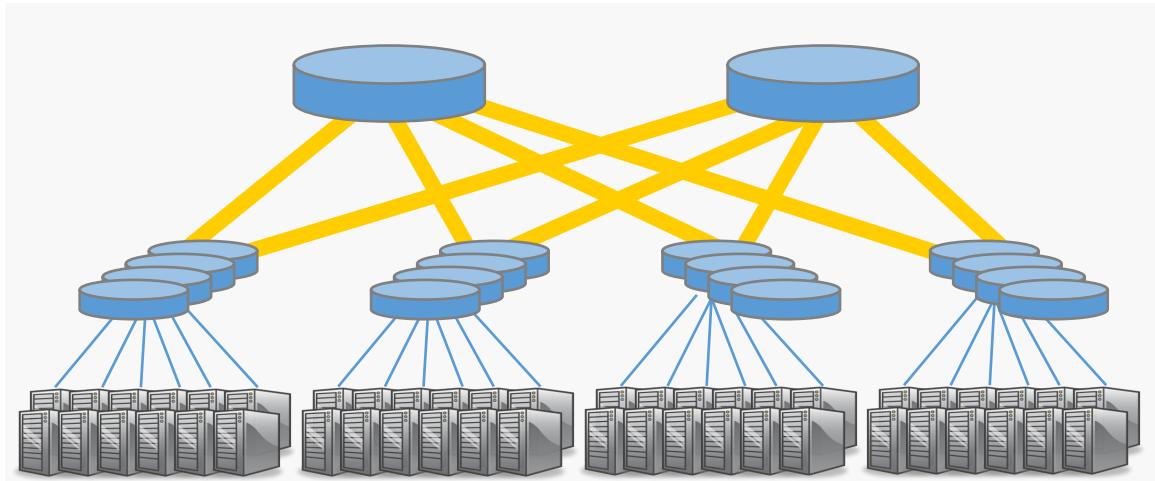


Image courtesy: Google cloud

SDN's Impact on Datacenters



Cost

500,000 servers & 25,000 switches

\$10k per legacy switch = \$250M

\$2k per SDN switch = \$50M

Savings in 5 data centers = \$1Bn

Control

Centralized control is easier to manage

Customized, differentiated network

Home grown traffic engineering

50% utilization → 95% utilization

Suggested Reading

A Purpose-Built Global Network: Google's Move to SDN [CACM 2016]

- *Perspectives from Vahdat, Clark, and Rexford*
- *Transitioning networking research into real world*
- *How SDN solved Google's scale challenges*
- *Also: <https://youtu.be/FaAZAll2x0w&t=90s>*



Spot Quiz (ICON)