Software Defined Networks

Damiano Carra

Università degli Studi di Verona
Dipartimento di Informatica

Acknowledgements

- ❑ *Credits*
  - *Part of the course material is based on slides provided by the following authors*
    - *Data centers: Andreas Haeberlen, Zachary G. Ives*
    - *SDN: Jennifer Rexford, Nick McKeown*

Datacenter general overview

Scaling up

❑ What if one computer is not enough?

❑ What if the biggest computer is not enough?

❑ What if your cluster is too big to fit into your office building?

PC

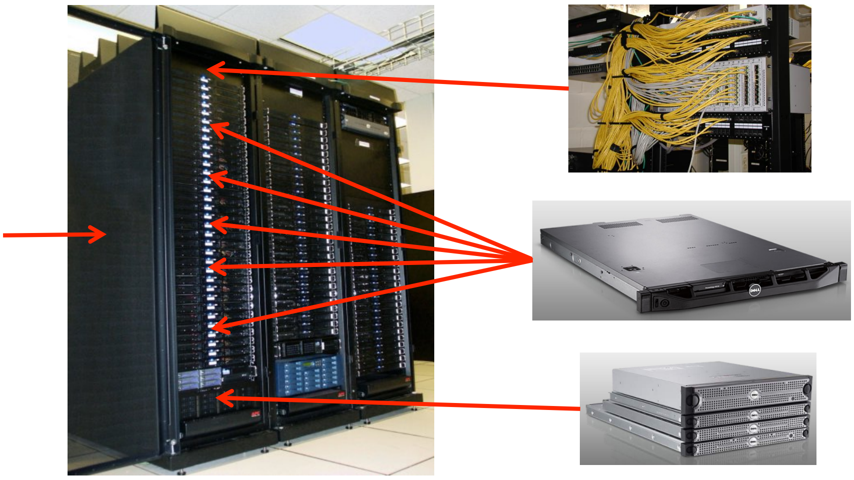
Server

Cluster

Data center

Clusters

- ❑ Characteristics of a cluster:
  - Many similar machines, close interconnection (same room?)
  - Often special, standardized hardware (racks, blades)
  - Usually owned and used by a single organization



Rack

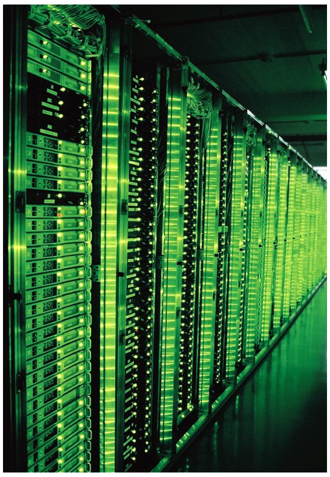
Network switch (connects nodes with each other and with other racks)

Many nodes/blades (often identical)

Storage device(s)

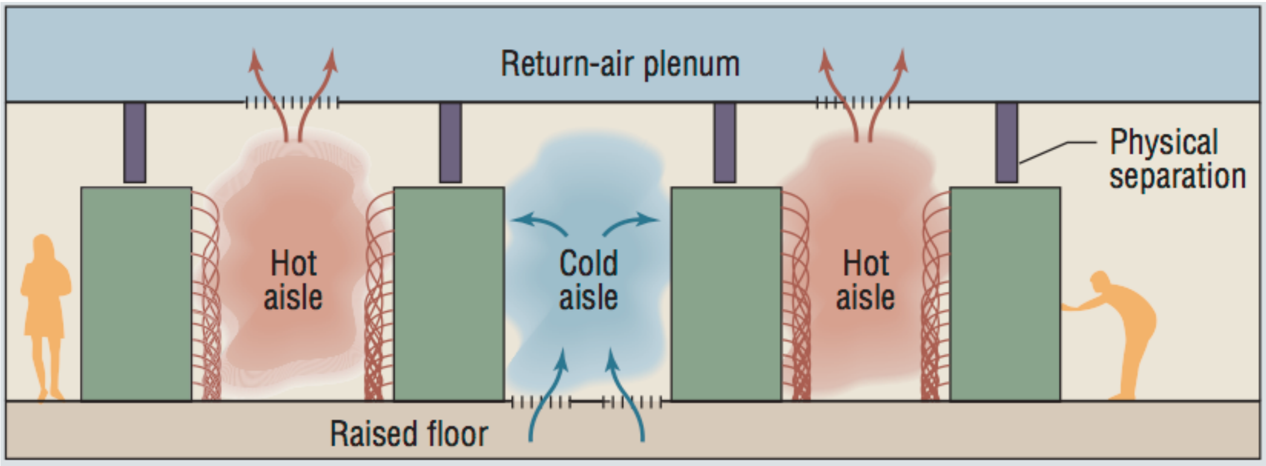
What's in a data center?

- ❑ Hundreds or thousands of racks
  - Each rack has 20-60 servers

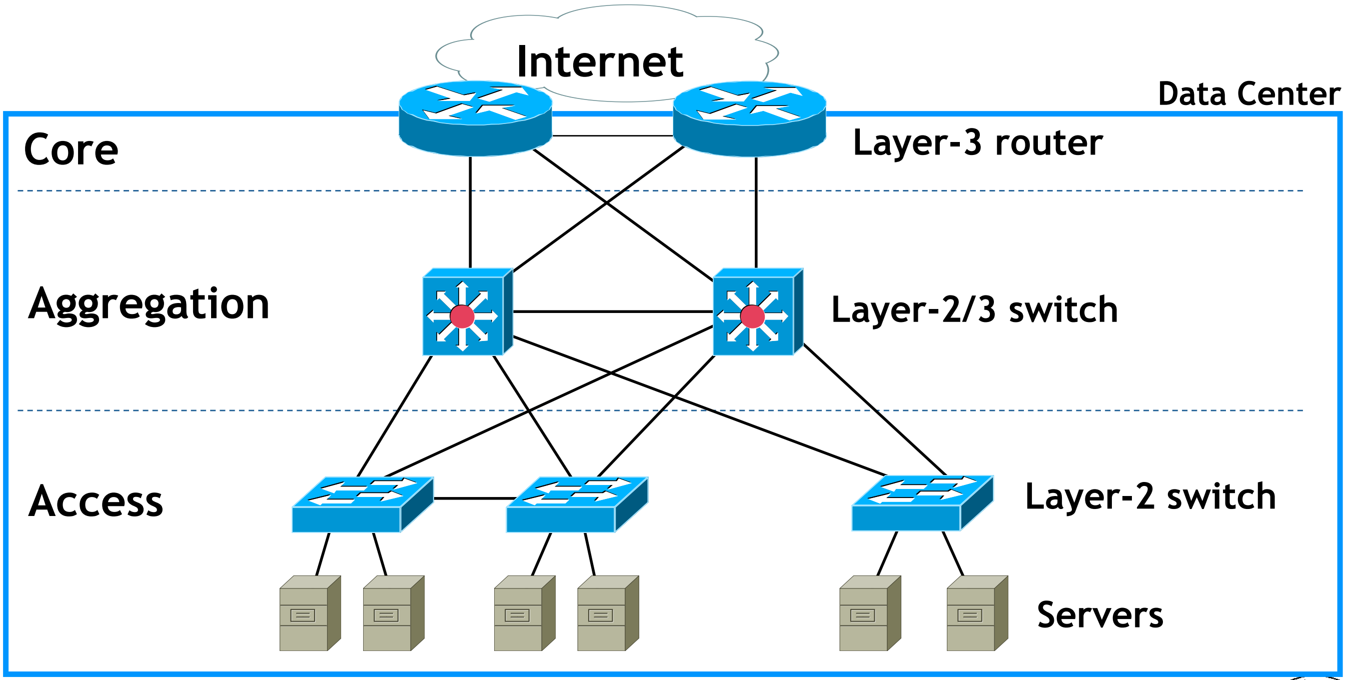


Source: 1&1

Rack organization



Common data center topology



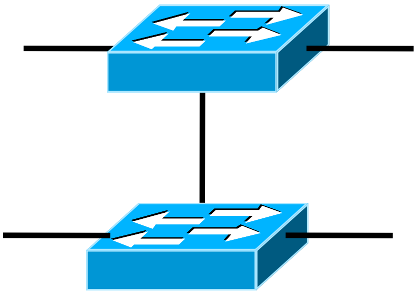
Software Defined Networks (SDN)

The Internet: A Remarkable Story

- ❑ Tremendous success
  - From research experiment to global infrastructure
- ❑ Brilliance of under-specifying
  - Network: best-effort packet delivery
  - Hosts: arbitrary applications
- ❑ Enables innovation in applications
  - Web, P2P, VoIP, social networks, virtual worlds
- ❑ But, change is easy only at the edge…

Inside the Network: A Different Story…

- ❑ Closed equipment
  - Software bundled with hardware
  - Vendor-specific interfaces
- ❑ Over specified
  - Slow protocol standardization
- ❑ Few people can innovate
  - Equipment vendors write the code
  - Long delays to introduce new features

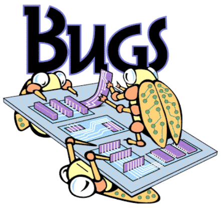


→ Impacts performance, security, reliability, cost…

Networks are Hard to Manage

- ❑ Operating a network is expensive
  - More than half the cost of a network
  - Yet, operator error causes most outages

- ❑ Buggy software in the equipment
  - Routers with 20+ million lines of code
  - Cascading failures, vulnerabilities, etc.



- ❑ The network is “in the way”
  - Especially a problem in data centers
  - … and home networks

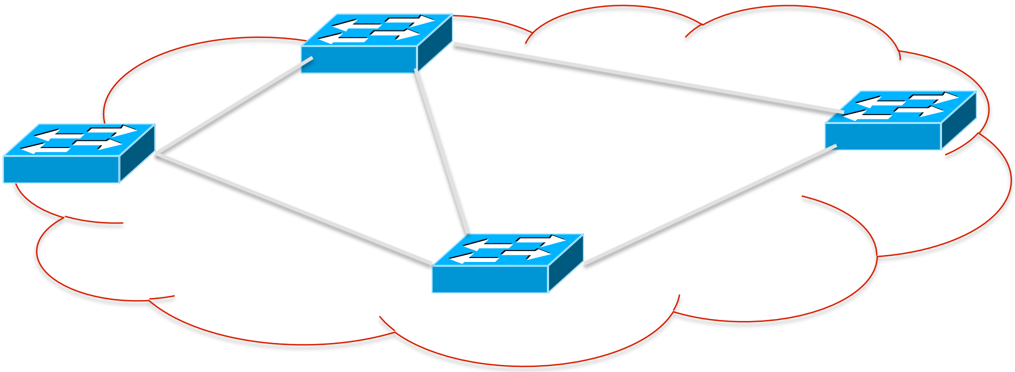
Creating Foundation for Networking

- ❑ A domain, not (yet?) a discipline
  - Alphabet soup of protocols
  - Header formats, bit twiddling
  - Preoccupation with artifacts
- ❑ From practice, to principles
  - Intellectual foundation for networking
  - Identify the key abstractions
  - … and support them efficiently
- ❑ To build networks worthy of society' s trust

Traditional Computer Networks

The "Division of Labor"

Data plane: Packet streaming

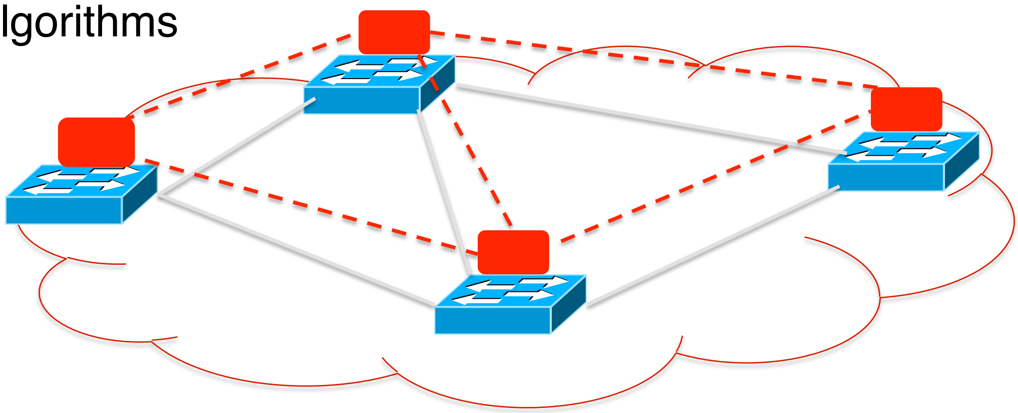


Forward, filter, buffer, mark, rate-limit, and measure packets

Traditional Computer Networks

The “Division of Labor”

Control plane:
Distributed algorithms

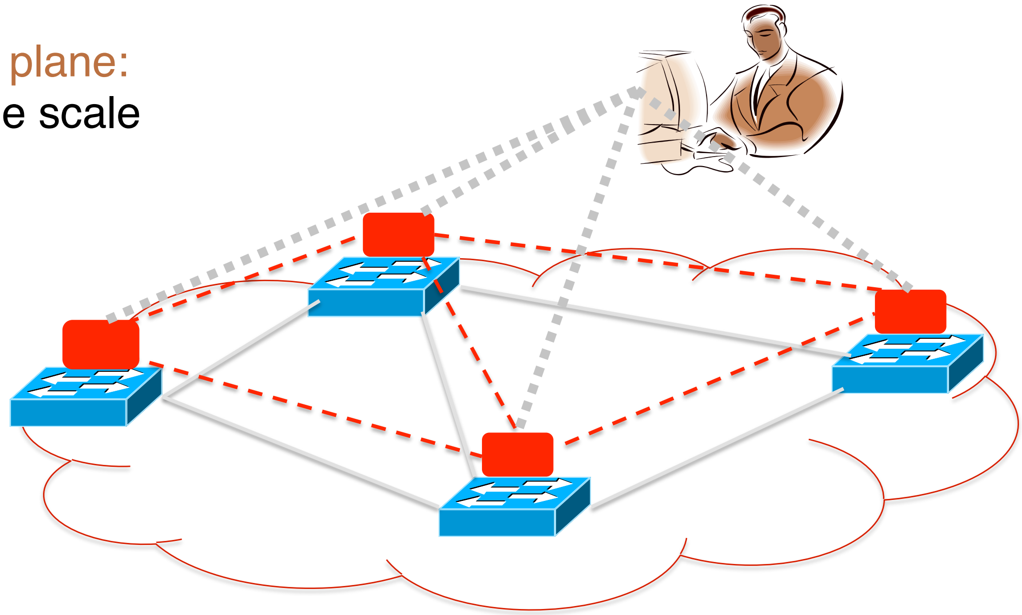


Track topology changes, compute routes, install forwarding rules

Traditional Computer Networks

The “Division of Labor”

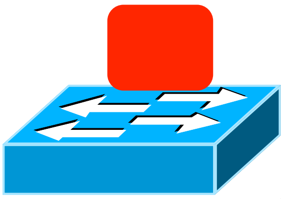
Management plane:
Human time scale



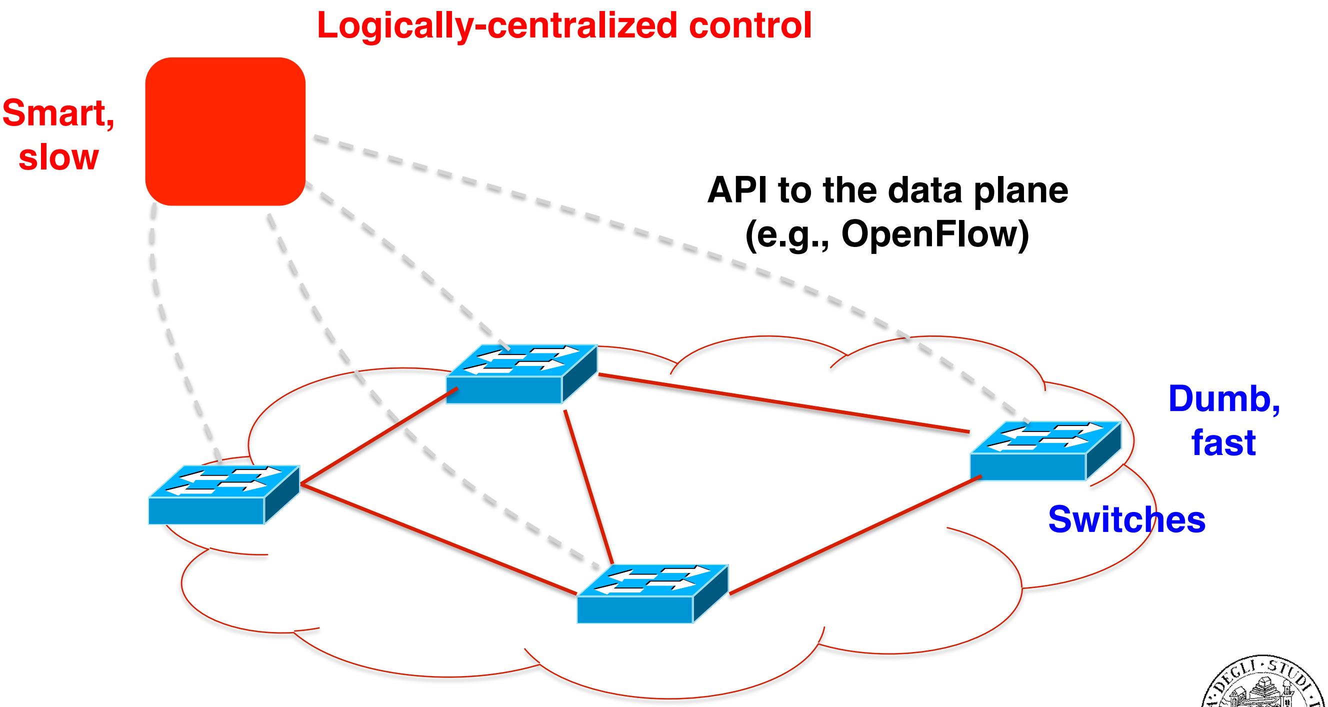
Collect measurements and configure the equipment

Death to the Control Plane!

- ❑ Simpler management
  - No need to “invert” control-plane operations
- ❑ Faster pace of innovation
  - Less dependence on vendors and standards
- ❑ Easier interoperability
  - Compatibility only in “wire” protocols
- ❑ Simpler, cheaper equipment
  - Minimal software



Software Defined Networking (SDN)



Logically-centralized control
Smart, slow
API to the data plane (e.g., OpenFlow)
Dumb, fast
Switches

Data-Plane: Simple Packet Handling

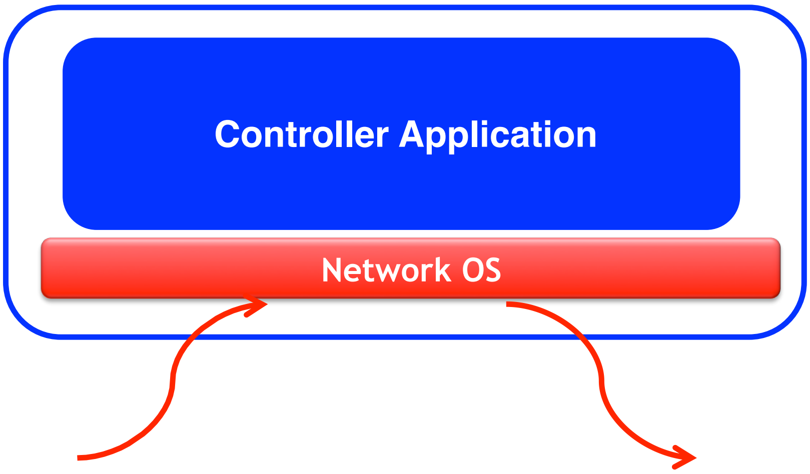
- ❑ Simple packet-handling rules
  - Pattern: match packet header bits
  - Actions: drop, forward, modify, send to controller
  - Priority: disambiguate overlapping patterns
  - Counters: #bytes and #packets

1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

Unifies Different Kinds of Boxes

- ❑ Router
  - Match: longest destination IP prefix
  - Action: forward out a link
- ❑ Switch
  - Match: destination MAC address
  - Action: forward or flood
- ❑ Firewall
  - Match: IP addresses and TCP/UDP port numbers
  - Action: permit or deny
- ❑ NAT
  - Match: IP address and port
  - Action: rewrite address and port

Controller: Programmability



Events from switches
Topology changes,
Traffic statistics,
Arriving packets

Commands to switches
(Un)install rules,
Query statistics,
Send packets

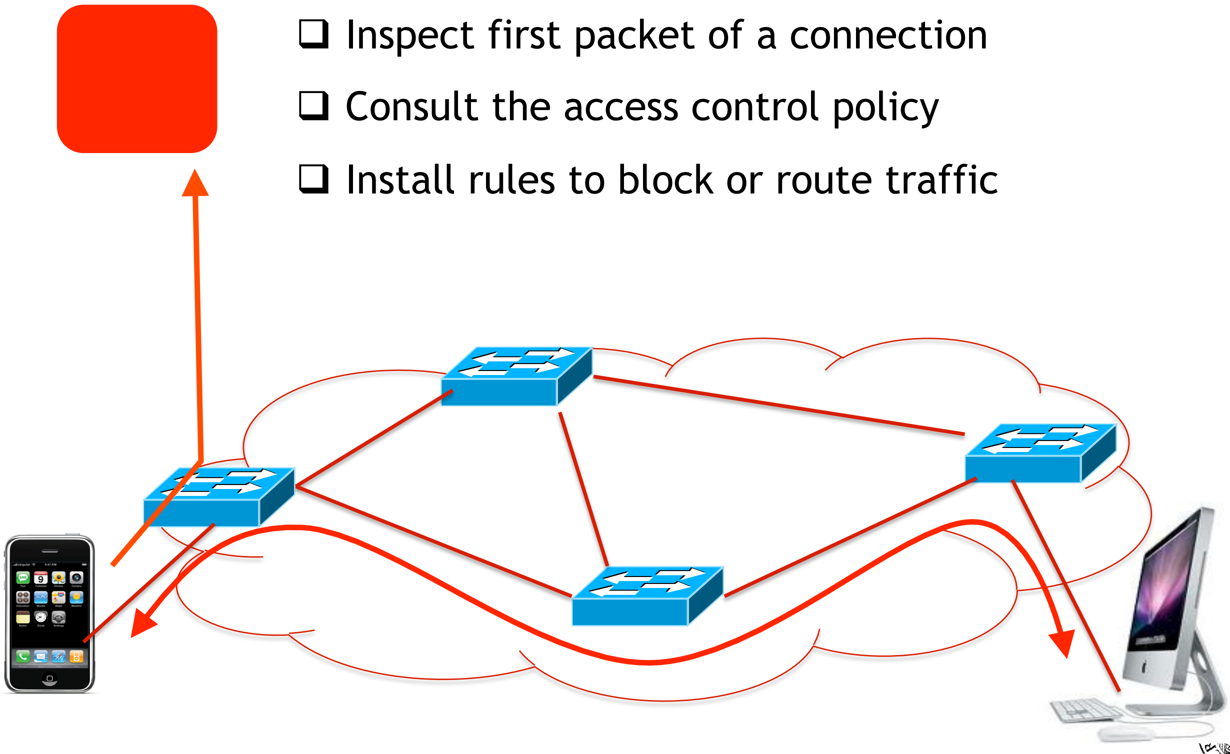
Example OpenFlow Applications

- ❑ **Dynamic access control**
- ❑ **Seamless mobility/migration**
- ❑ **Server load balancing**
- ❑ **Network virtualization**
- ❑ Using multiple wireless access points
- ❑ Energy-efficient networking
- ❑ Adaptive traffic monitoring
- ❑ Denial-of-Service attack detection

See http://www.openflow.org/videos/

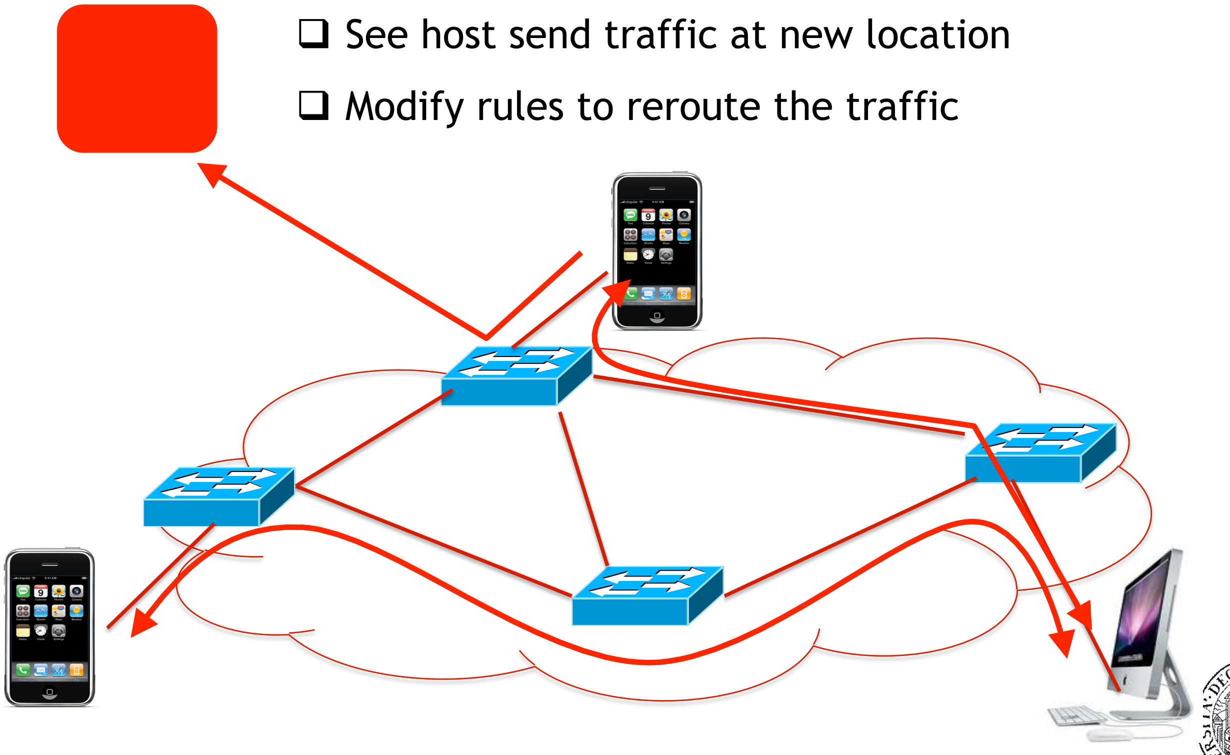
E.g.: Dynamic Access Control

- ❑ Inspect first packet of a connection
- ❑ Consult the access control policy
- ❑ Install rules to block or route traffic



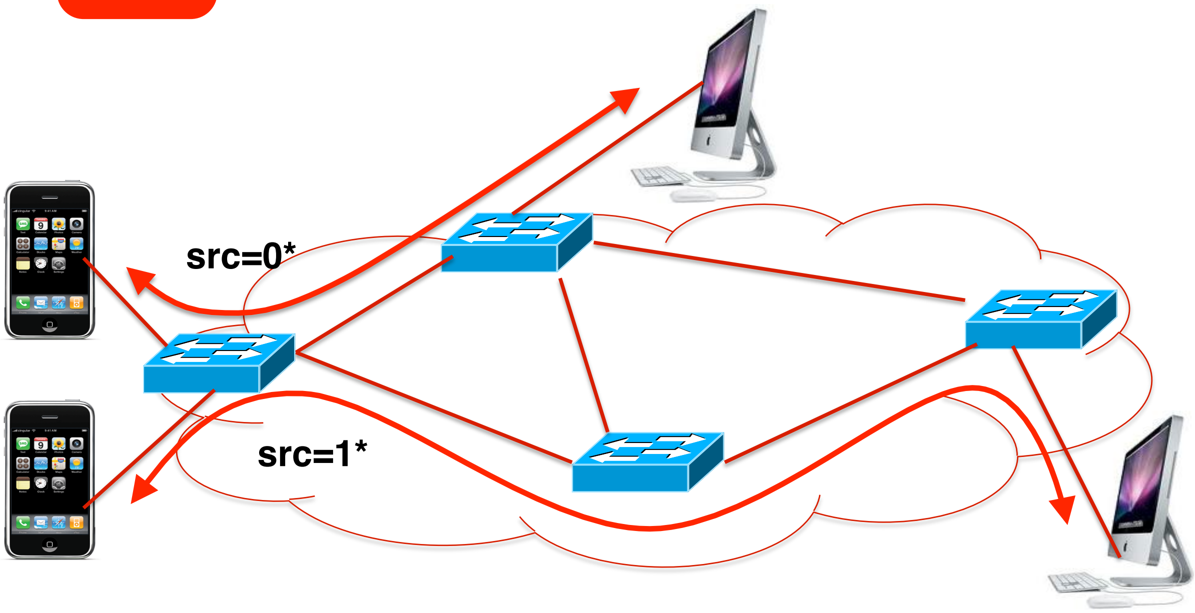
E.g.: Seamless Mobility/Migration

- ❑ See host send traffic at new location
- ❑ Modify rules to reroute the traffic



E.g.: Server Load Balancing

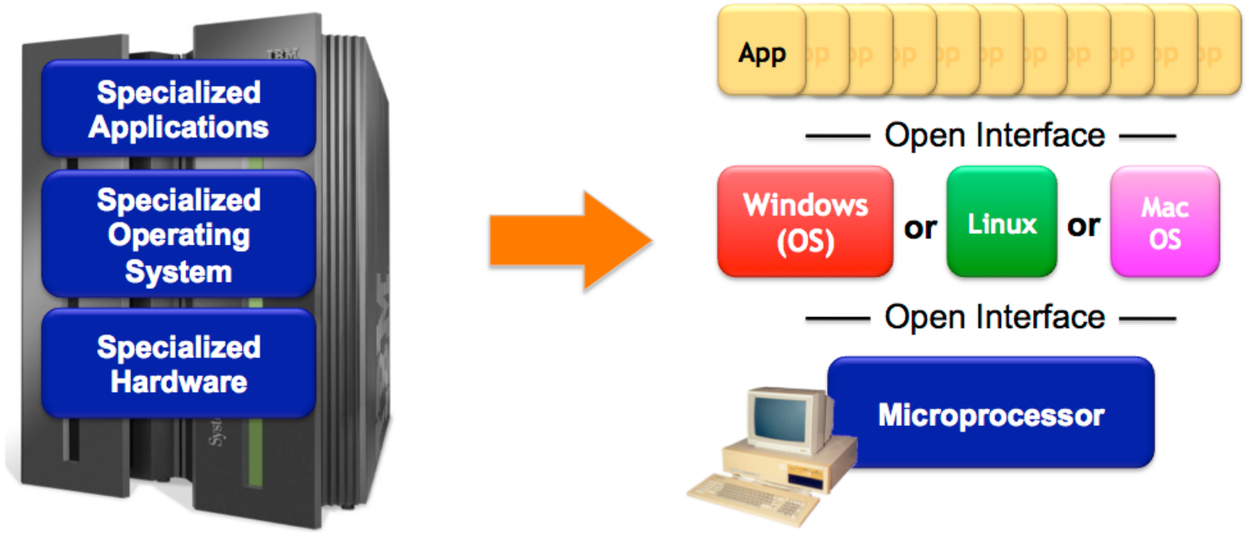
- ❑ Pre-install load-balancing policy
- ❑ Split traffic based on source IP



OpenFlow in the Wild

- ❑ Open Networking Foundation
  - Google, Facebook, Microsoft, Yahoo, Verizon, Deutsche Telekom, and many other companies
- ❑ Commercial OpenFlow switches
  - HP, NEC, Quanta, Dell, IBM, Juniper, …
- ❑ Network operating systems
  - NOX, Beacon, Floodlight, Nettle, ONIX, POX, Frenetic
- ❑ Network deployments
  - Eight campuses, and two research backbone networks
  - Commercial deployments (e.g., Google backbone)

A Helpful Analogy: Mainframes



Specialized Applications

Specialized Operating System

Specialized Hardware

App

Open Interface

Windows (OS) or Linux or Mac OS

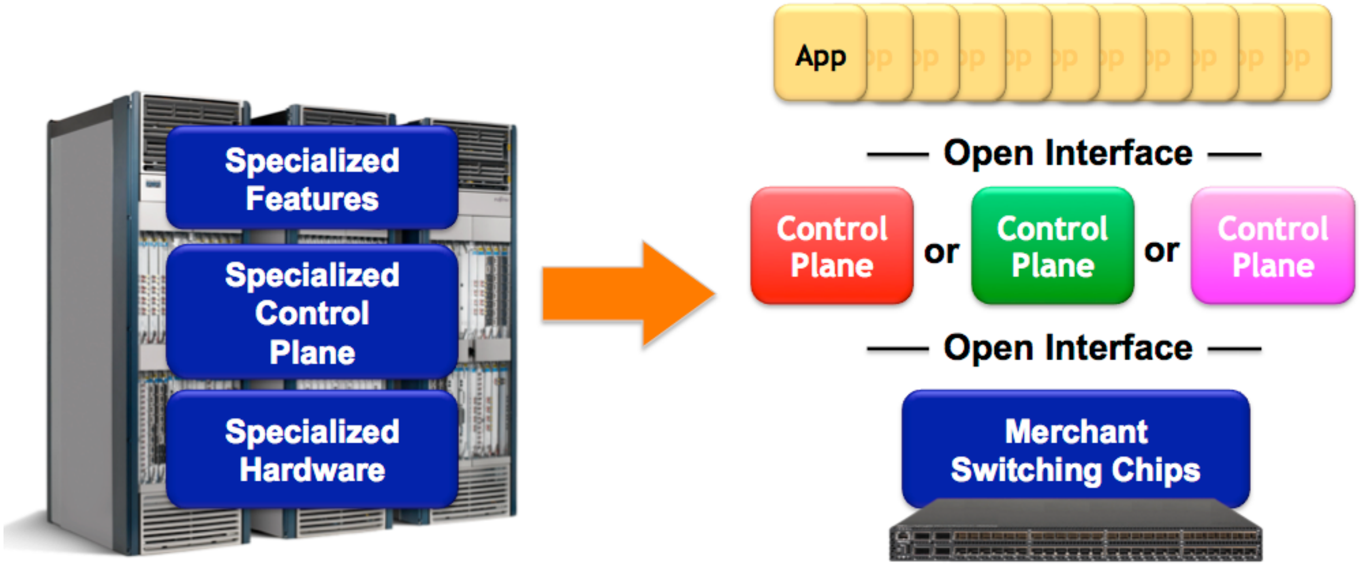
Open Interface

Microprocessor

Vertically integrated
Closed, proprietary
Slow innovation
Small industry

Horizontal
Open interfaces
Rapid innovation
Huge industry

A Helpful Analogy (cont’d): Routers/Switches



Specialized Features

Specialized Control Plane

Specialized Hardware

App

Open Interface

Control Plane or Control Plane or Control Plane

Open Interface

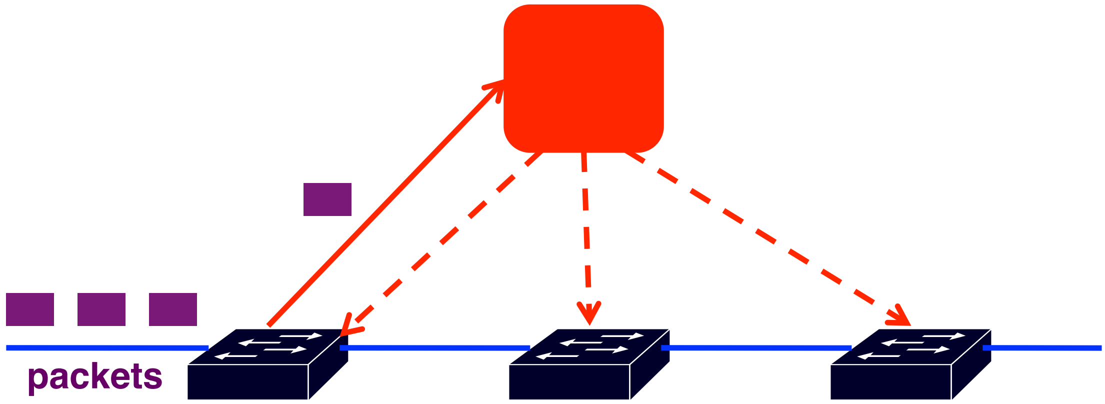
Merchant Switching Chips

Vertically integrated
Closed, proprietary
Slow innovation

Horizontal
Open interfaces
Rapid innovation

Challenges: Controller Delay and Overhead

- ❑ Controller is much slower the the switch
- ❑ Processing packets leads to delay and overhead
- ❑ Need to keep most packets in the “fast path”



Challenges: Testing and Debugging

- ❑ OpenFlow makes programming possible
  - Network-wide view at controller
  - Direct control over data plane
- ❑ Plenty of room for bugs
  - Still a complex, distributed system
- ❑ Need for testing techniques
  - Controller applications
  - Controller and switches
  - Rules installed in the switches

SDN – Conclusion

- Rethinking networking
  - Open interfaces to the data plane
  - Separation of control and data
  - Leveraging techniques from distributed systems

- Significant momentum
  - In both research and industry