Cornell University

Data Center Networks and Software-defined Networking

Hakim Weatherspoon

Associate Professor, Dept of Computer Science

CS 5413: High Performance Computing and Networking

March 20, 2017

Slides taken liberally from Jennifer Rexford's Computer Networks, COS 461
http://www.cs.princeton.edu/courses/archive/spr12/cos461/docs/lec24-sdn.ppt

Agenda for semester

- HW2 grading
  - Sign up interactive demo session
  - **This Wednesday and Thursday**

- Project
  - Continue to make progress.
  - **Intermediate project report due Mar 24. BOOM proposal due Mar 31.**
  - **Spring break, week of April 2nd**
  - **Intermediate project report 2 due April 12th.**
  - **BOOM, Wednesday, April 19**
  - **End of Semester, Wednesday, May 10**

- Check website for updated schedule

Where are we in the semester?

- Overview and Basics
  - Overview
  - Basic Switch and Queuing (today)
  - Low-latency and congestion avoidance (DCTCP)
- Data Center Networks
  - Data Center Network Topologies
  - Software defined networking
    - Software control plane (SDN)
    - Programmable data plane (hardware [P4] and software [Netmap])
  - Rack-scale computers and networks
  - Disaggregated datacenters
  - Alternative Switching Technologies
  - Data Center Transport
  - Virtualizing Networks
  - Middleboxes
- Advanced topics

Results from Survey

- Interested Topics:
  - SDN and programmable data planes
  - Disaggregated datacenters and rack-scale computers
  - Alternative switch technologies
  - Datacenter topologies
  - Datacenter transports
  - Advanced topics

- Interest in student presentations
  - Not really

Results from Survey

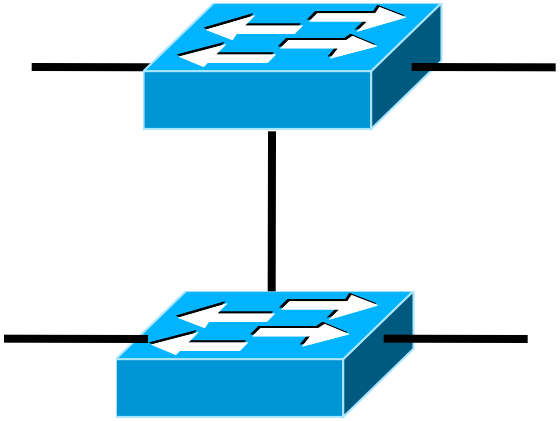
- Load
  - A bit High
- Interesting: lectures, Lab's, homeworks, projects
  - 4 out of 5 on average.
- How to make the class a 5 out of 5
  - Easier or no homework!
  - In class labs
  - Guidance and assistance on project
  - Linux kernel modules!
  - Labs about the cloud.

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 'Net: A Different Story…

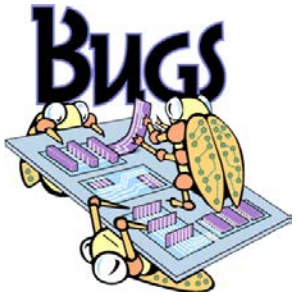
- Closed equipment
  - Software bundled with hardware
  - Vendor-specific interfaces
- Over specified
  - Slow protocol standardization
- Few people can innovate
  - Equipment vendors write the code
  - Long delays to introduce new features



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



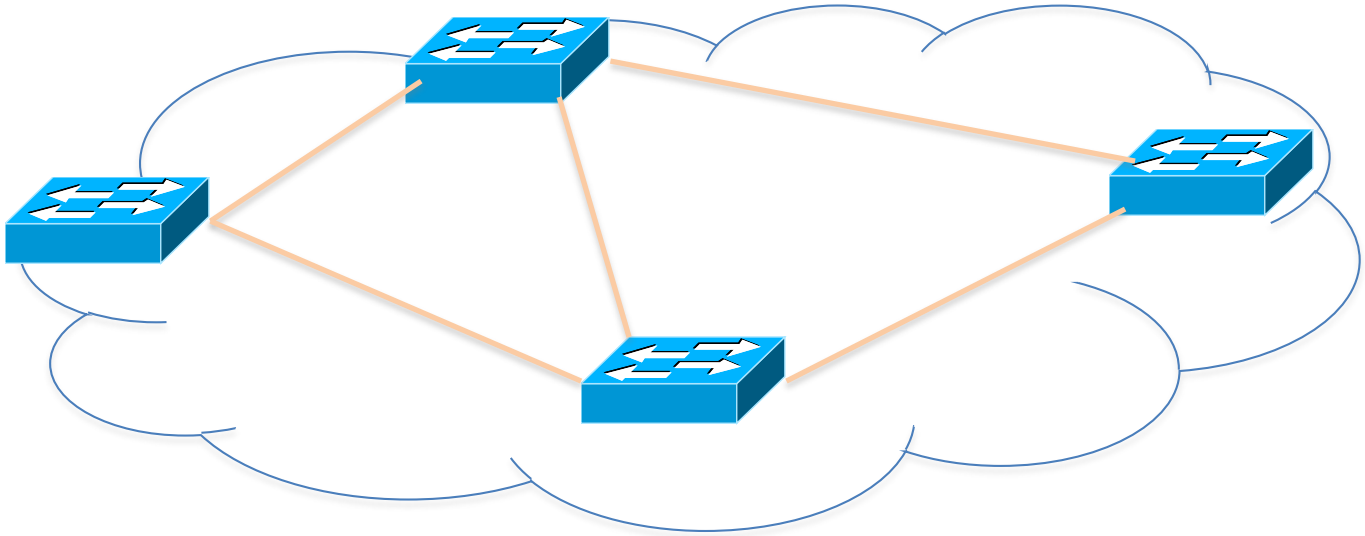
Creating Foundation for Networking

- 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

Rethinking the “Division of Labor”

Traditional Computer Networks

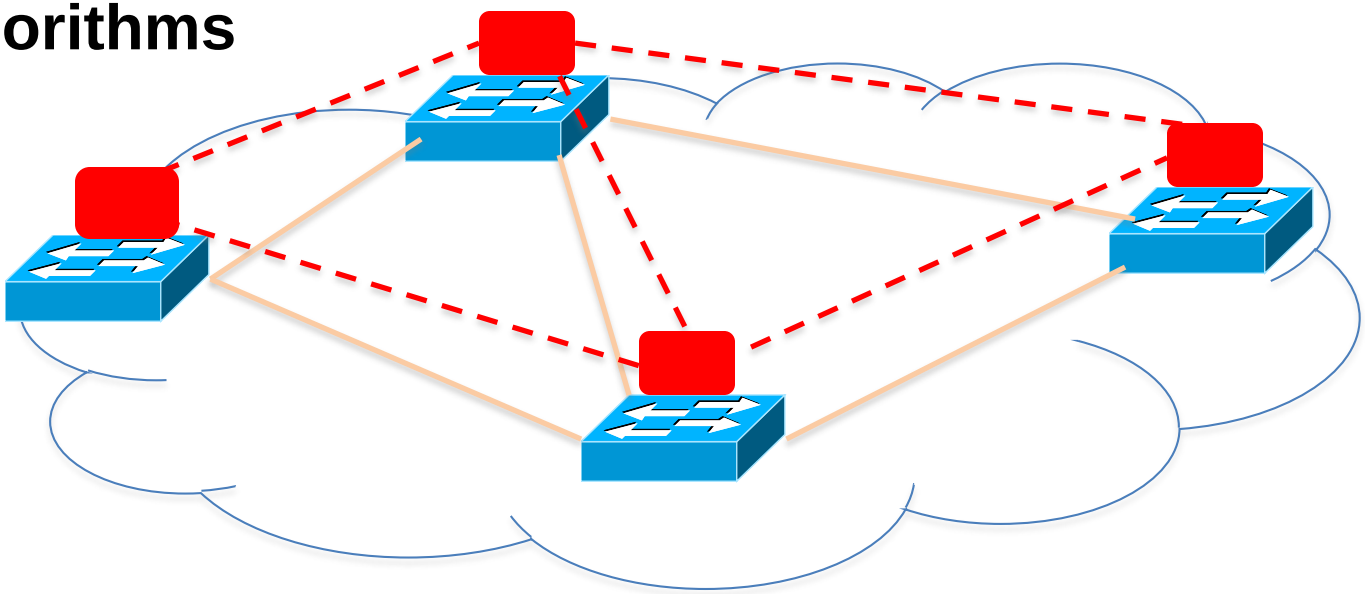
Data plane:
Packet streaming



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

Traditional Computer Networks

Control plane:
Distributed algorithms

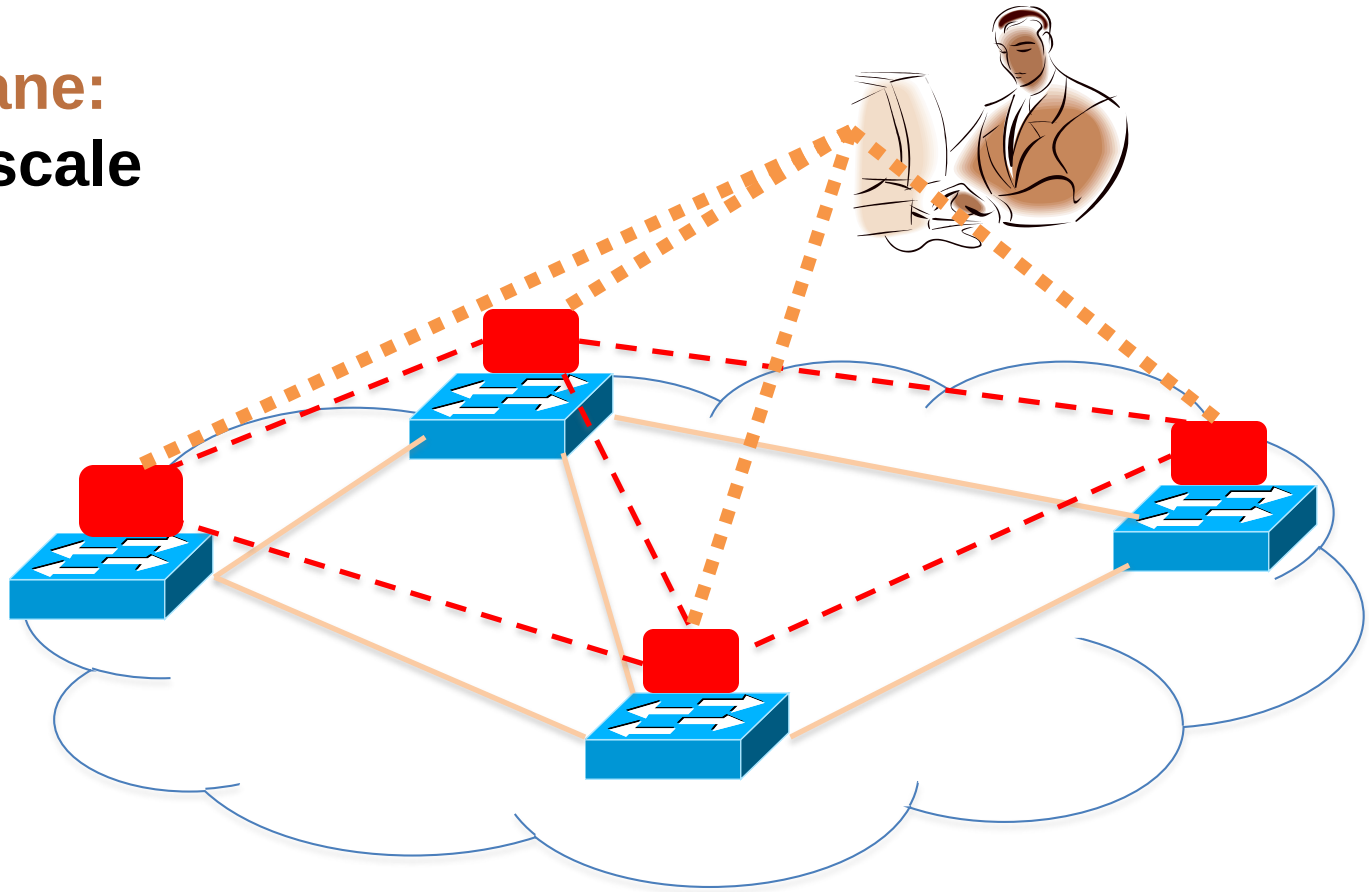


Track topology changes, compute routes, install forwarding rules

Traditional Computer Networks

Management plane:

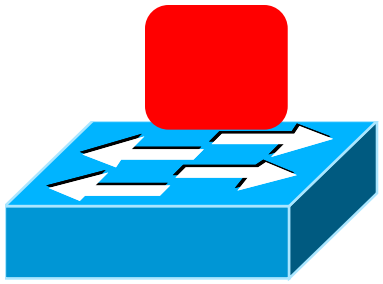
Human time scale



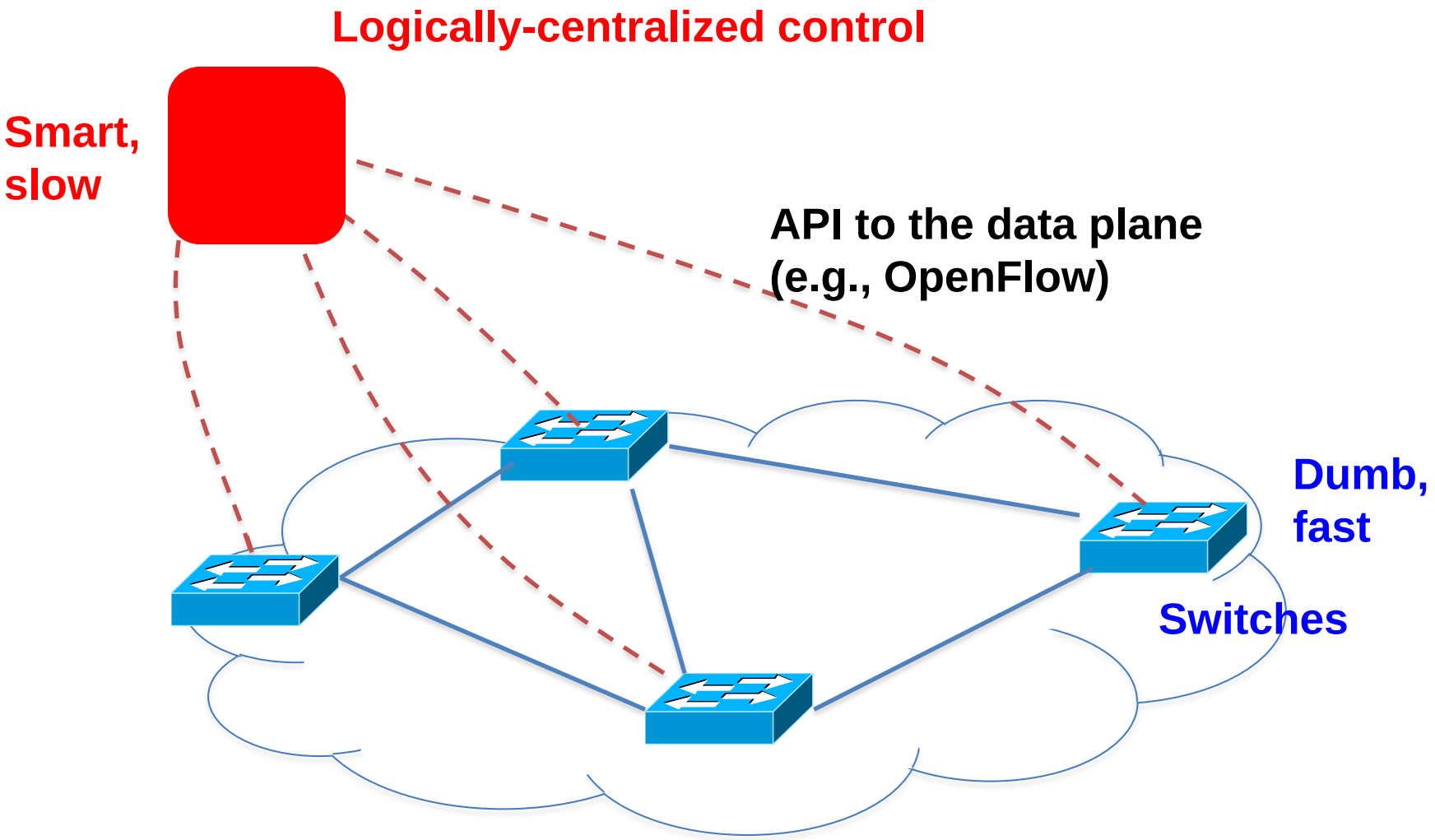
Collect measurements and configure the equipment

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)



OpenFlow Networks

Data-Plane: Simple Packet Handling

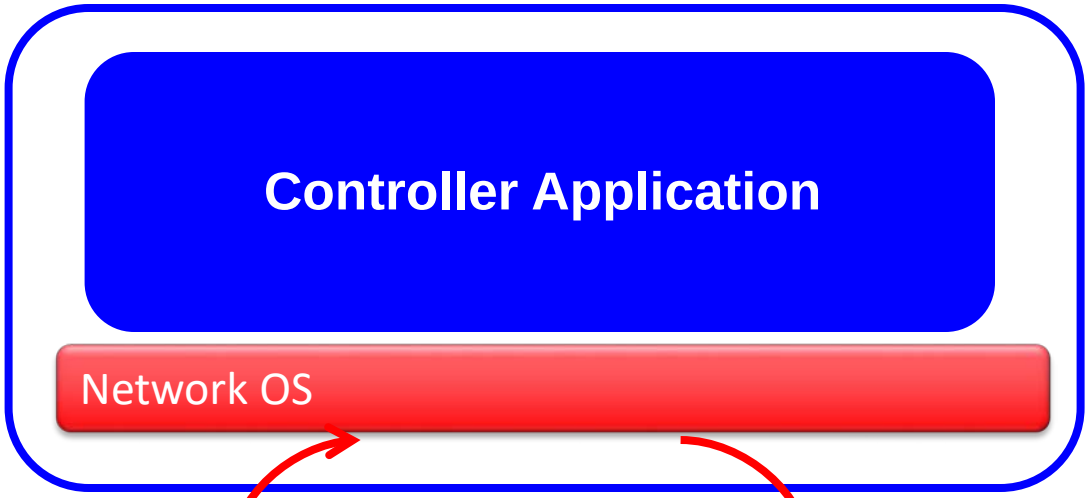
- Simple packet-handling rules
  - Pattern: match packet header bits
  - Actions: drop, forward, modify, send to controller
  - Priority: disambiguate overlapping patterns
  - Counters: #bytes and #packets

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

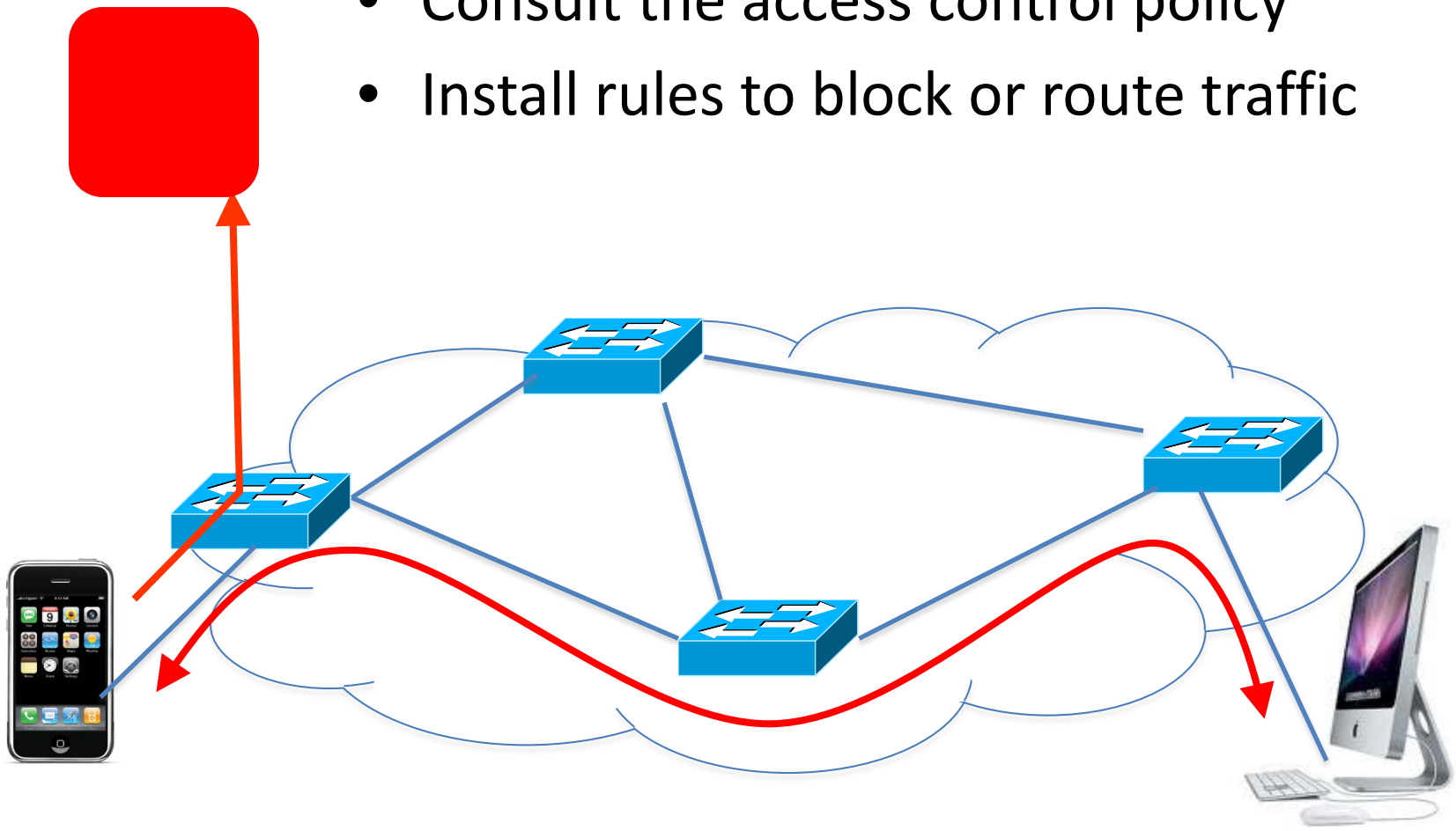
Example OpenFlow Applications

- **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/

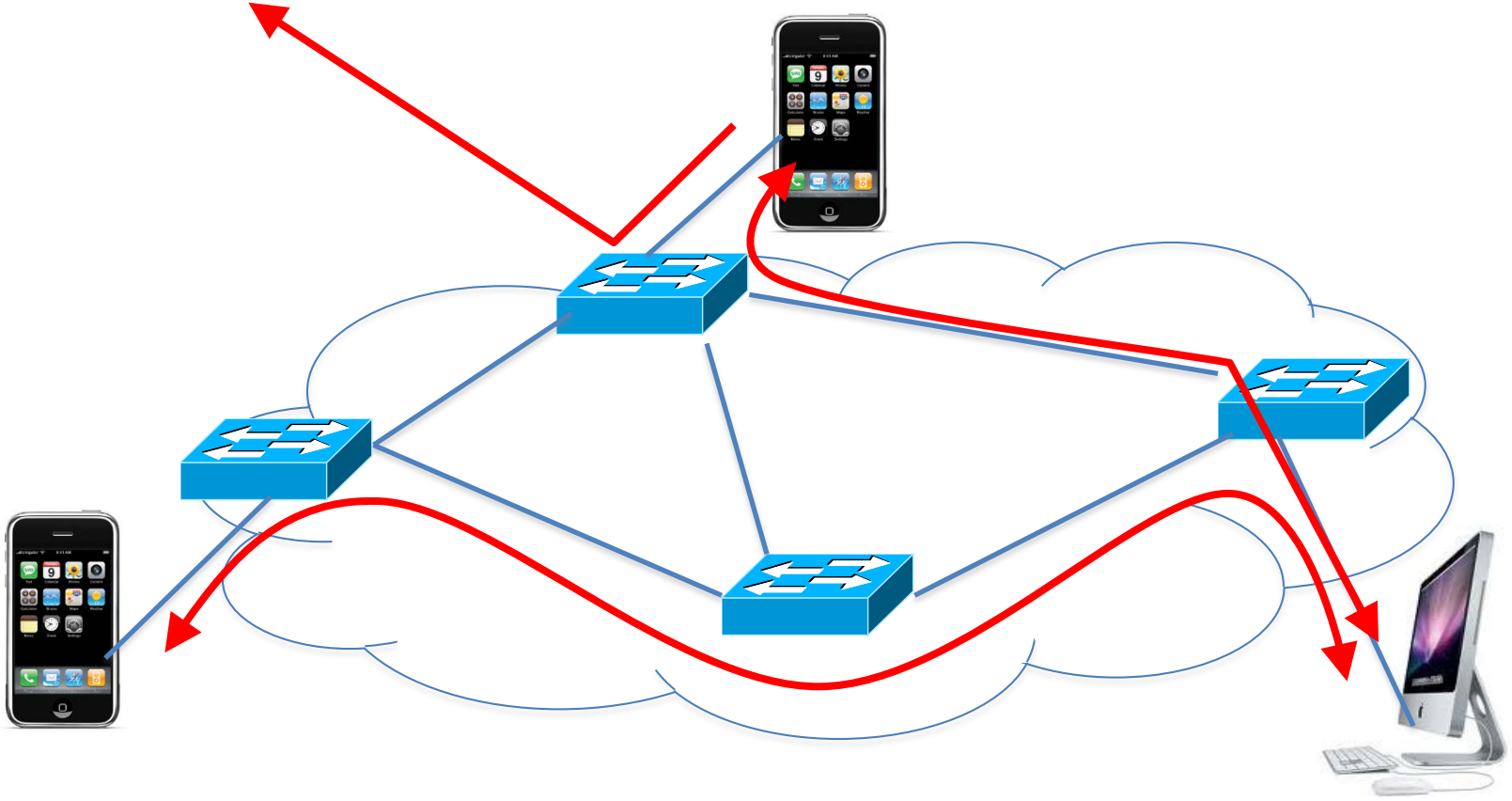
E.g.: Dynamic Access Control

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



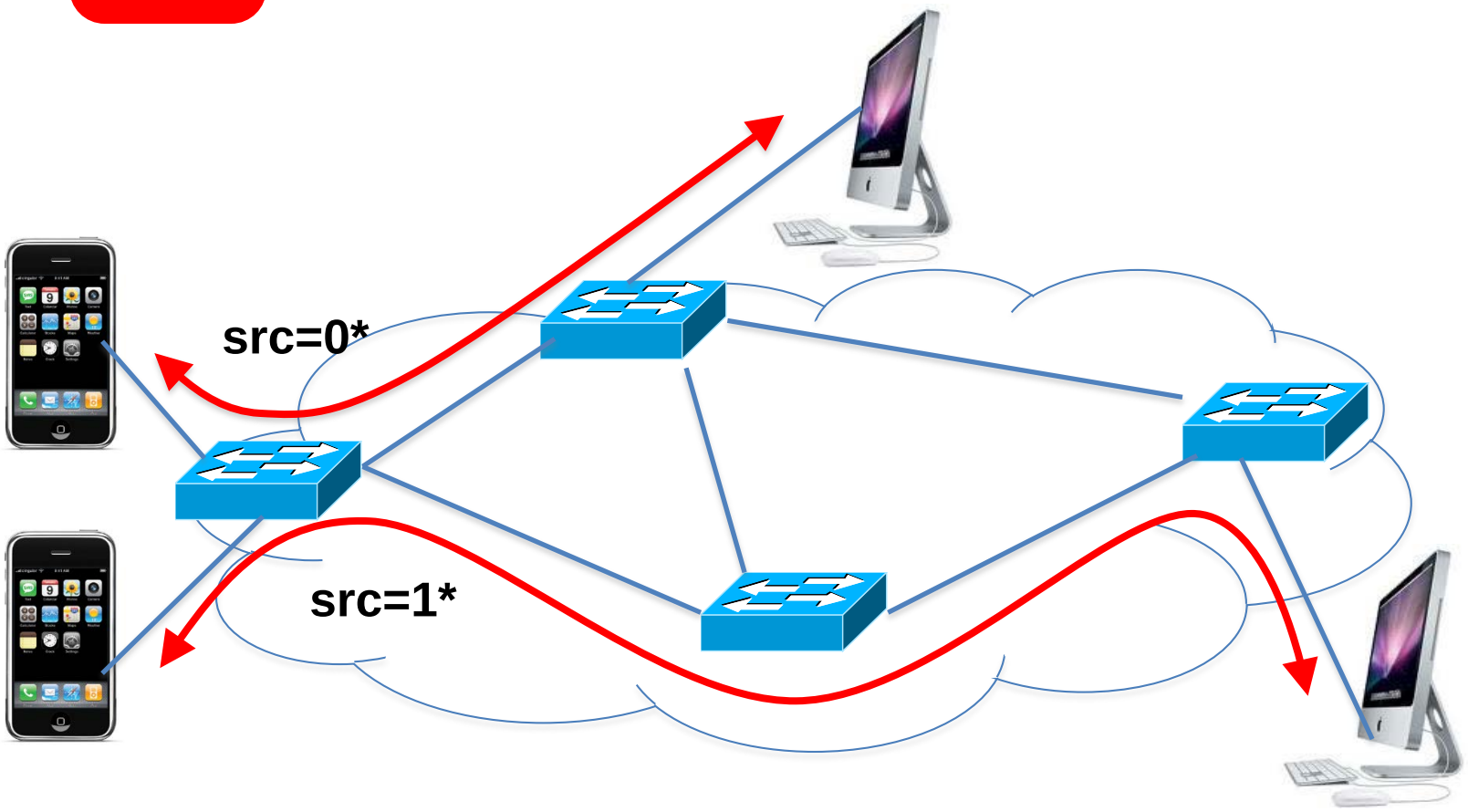
E.g.: Seamless Mobility/Migration

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



E.g.: Server Load Balancing

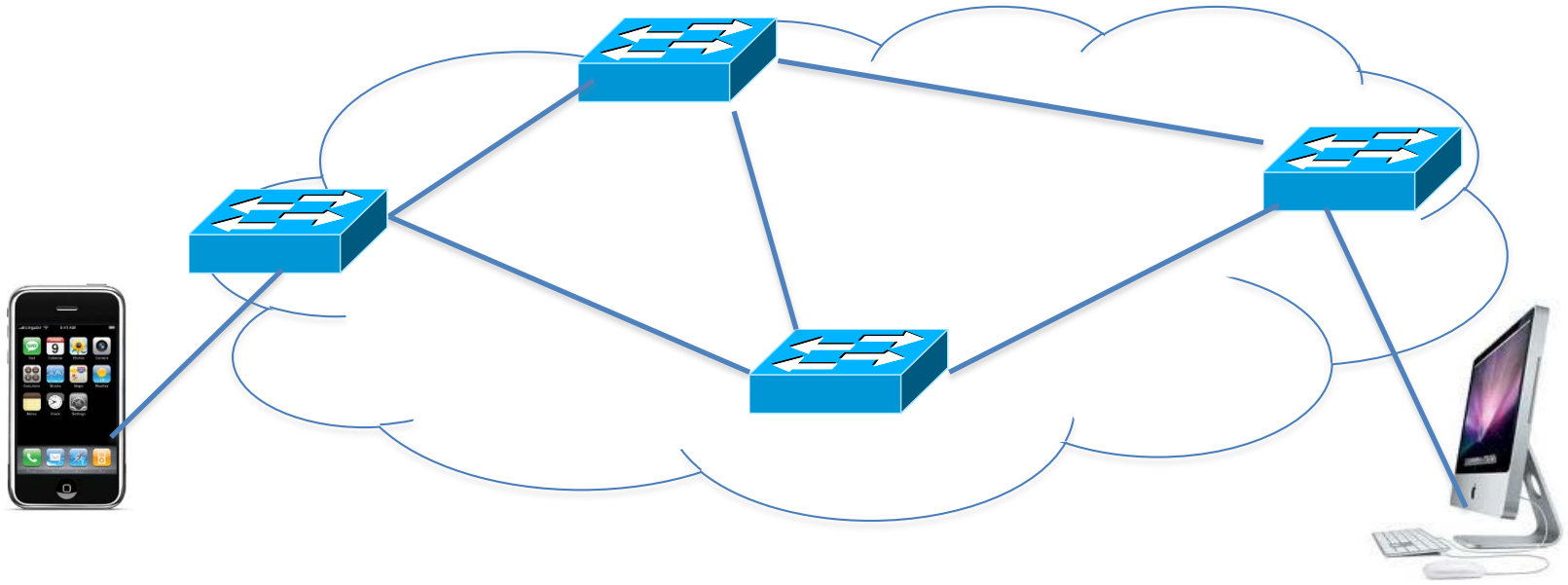
- Pre-install load-balancing policy
- Split traffic based on source IP



E.g.: Network Virtualization

Controller #1 **Controller #2** **Controller #3**

Partition the space of packet headers



OpenFlow in the Wild

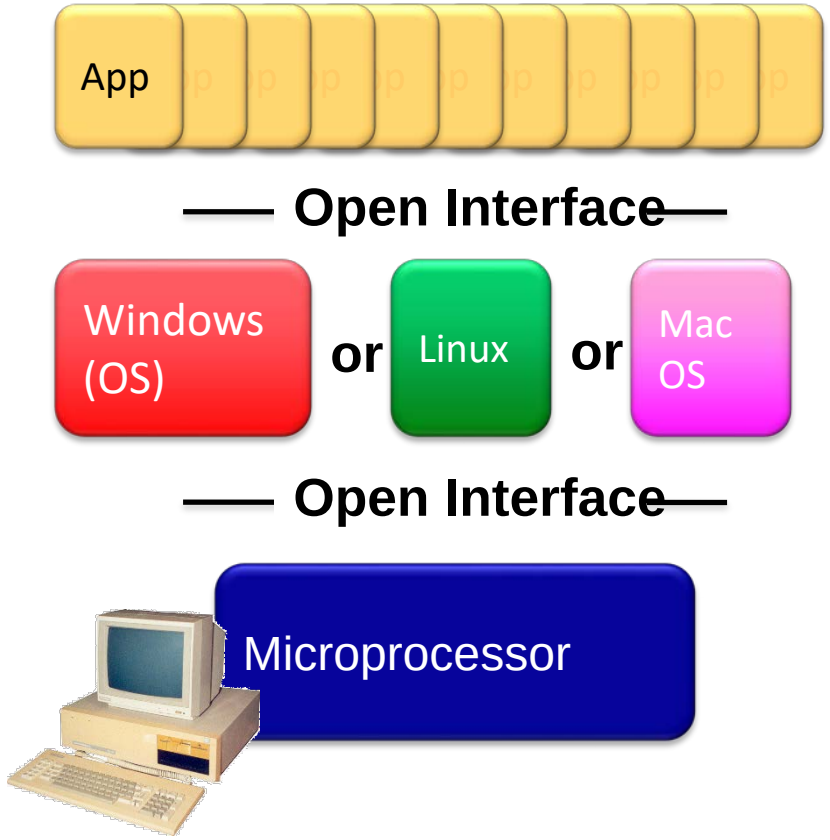
- Open Networking Foundation
  - Google, Facebook, Microsoft, Yahoo, Verizon, Deutsche Telekom, and many other companies
- Commercial OpenFlow switches
  - HP, NEC, Quanta, Dell, IBM, Juniper, …
- Network operating systems
  - NOX, Beacon, Floodlight, Nettle, ONIX, POX, Frenetic
- Network deployments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

A Helpful Analogy

From Nick McKeown's talk "Making SDN Work" at the Open Networking Summit, April 2012

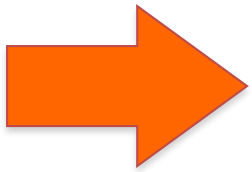
Mainframes

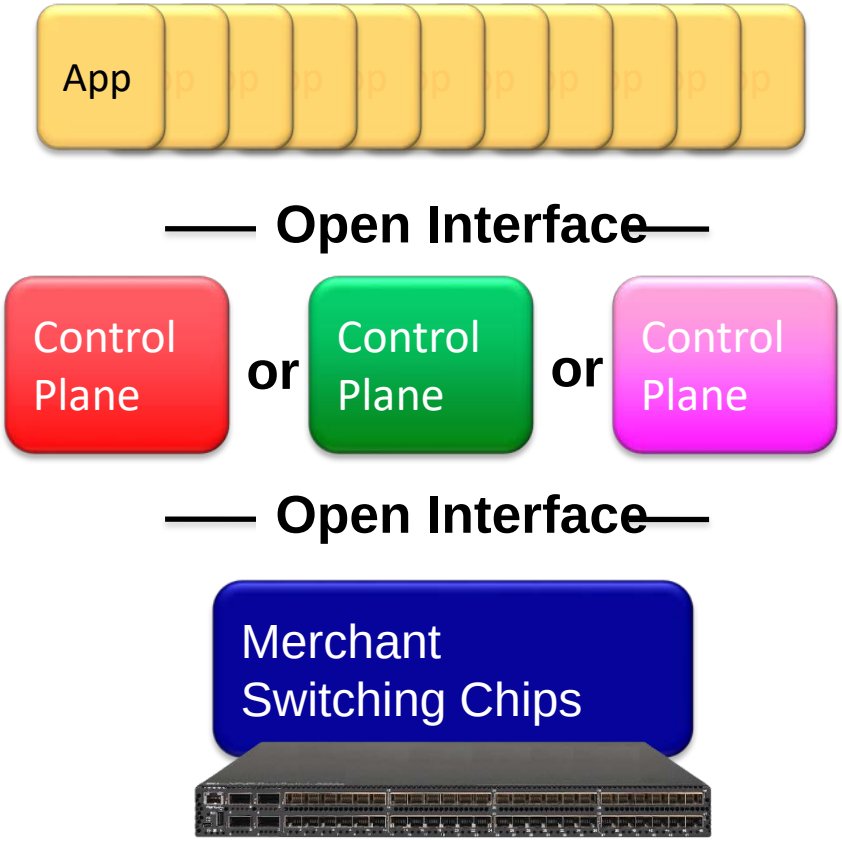
Vertically integrated
Closed, proprietary
Slow innovation
Small industry



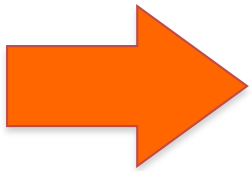
Horizontal
Open interfaces
Rapid innovation
Huge industry

Routers/Switches





Vertically integrated
Closed, proprietary
Slow innovation

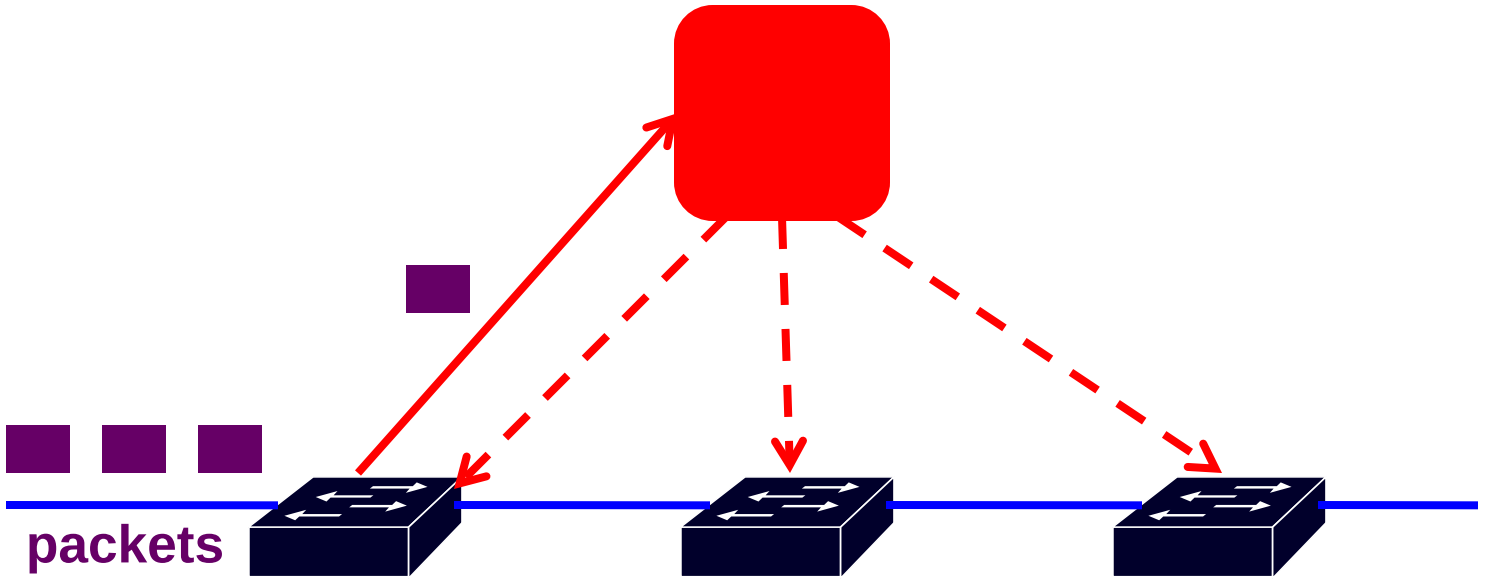


Horizontal
Open interfaces
Rapid innovation

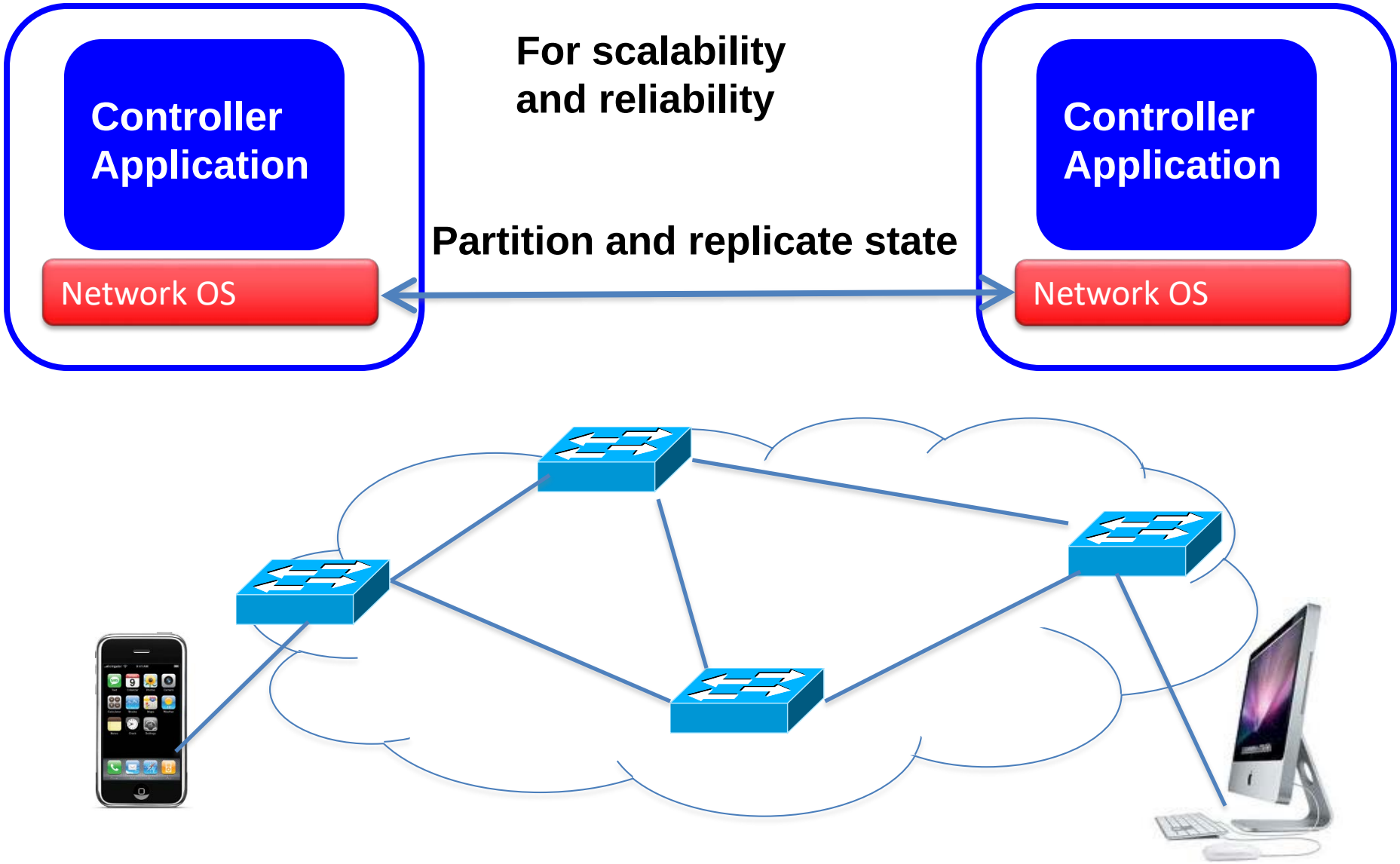
Challenges

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”



Distributed Controller

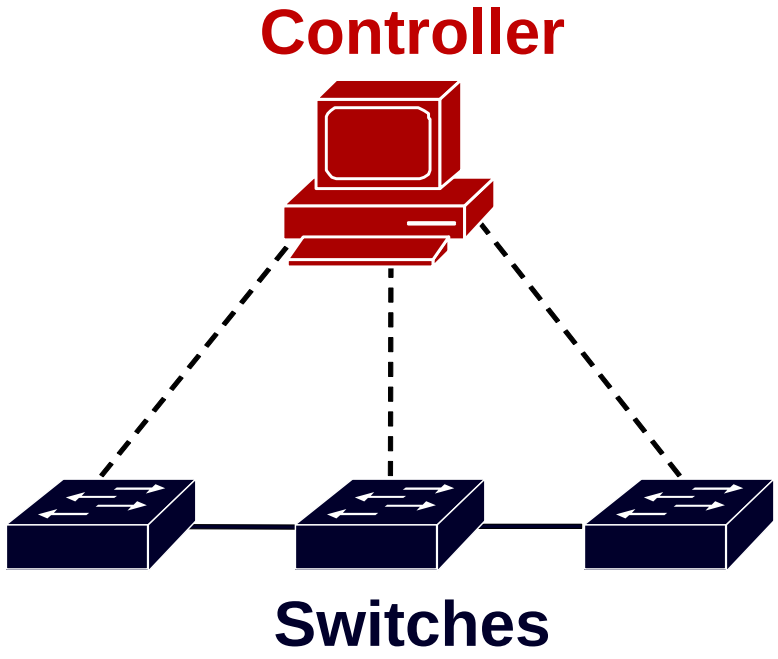


Testing and Debugging

- 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

Programming Abstractions

- Controller APIs are low-level
  - Thin veneer on the underlying hardware
- Need better languages
  - Composition of modules
  - Managing concurrency
  - Querying network state
  - Network-wide abstractions
- Ongoing at Princeton
  - http://www.frenetic-lang.org/



Perspective

- 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