DATA COMMUNICATOIN NETWORKING

Instructor: Ouldooz Baghban Karimi

Course Book: Computer Networking, A Top-Down Approach, Kurose, Ross

Slides:

- Course book Slides
- Slides from Princeton University COS461 Spring 2012 offering, Jennifer Rexford

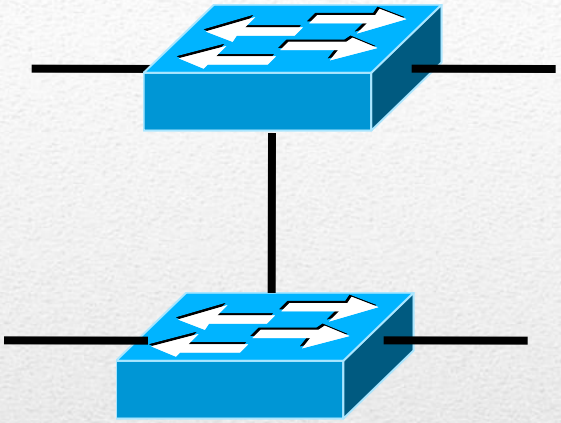
Course Overview

- **Basics of Computer Networks**
  - Internet & Protocol Stack
  - Application Layer
  - Transport Layer
  - Network Layer
  - Data Link Layer
- **Advanced Topics**
  - Case Studies of Computer Networks
  - Internet Applications
  - Network Management
  - Network Security

Internet

- **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 Internet



- **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…

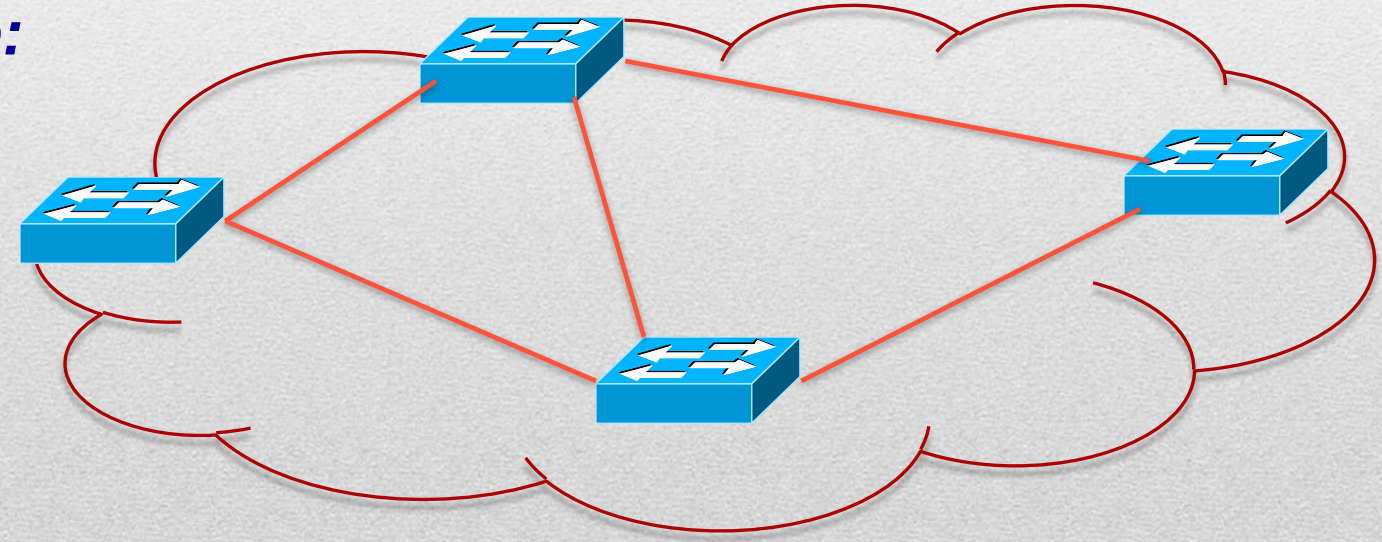
Networks are Hard to Manage

- 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

Traditional Computer Networks

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

Data plane:
Packet streaming

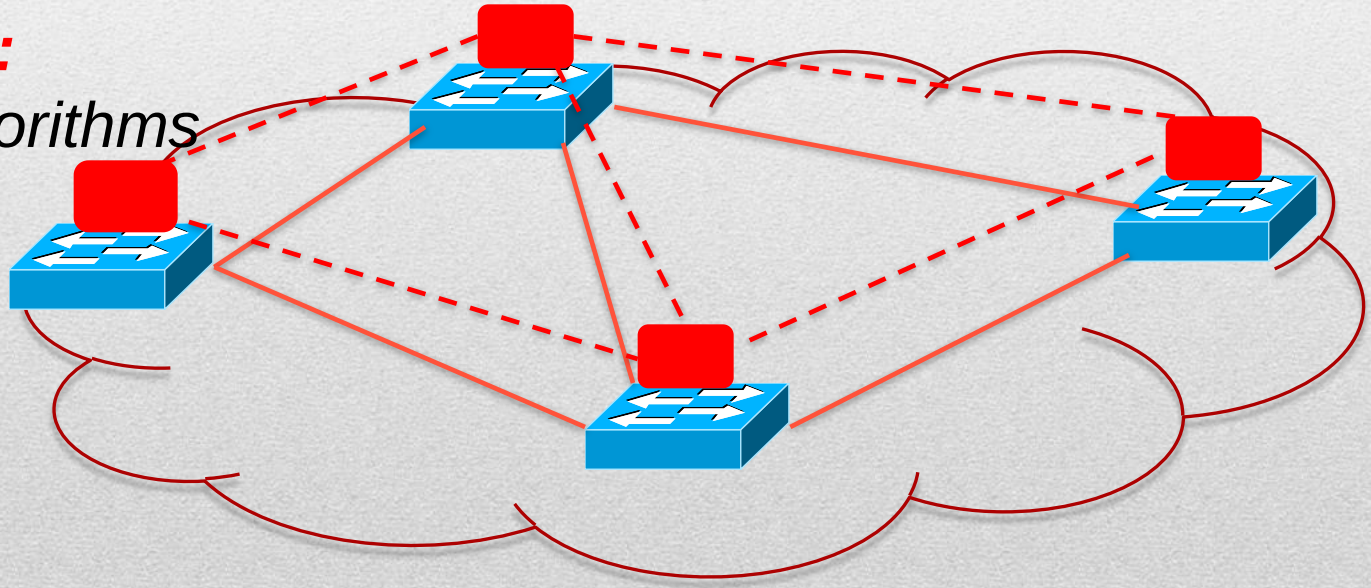


Traditional Computer Networks

Track topology changes, compute routes, install forwarding rules

Control plane:

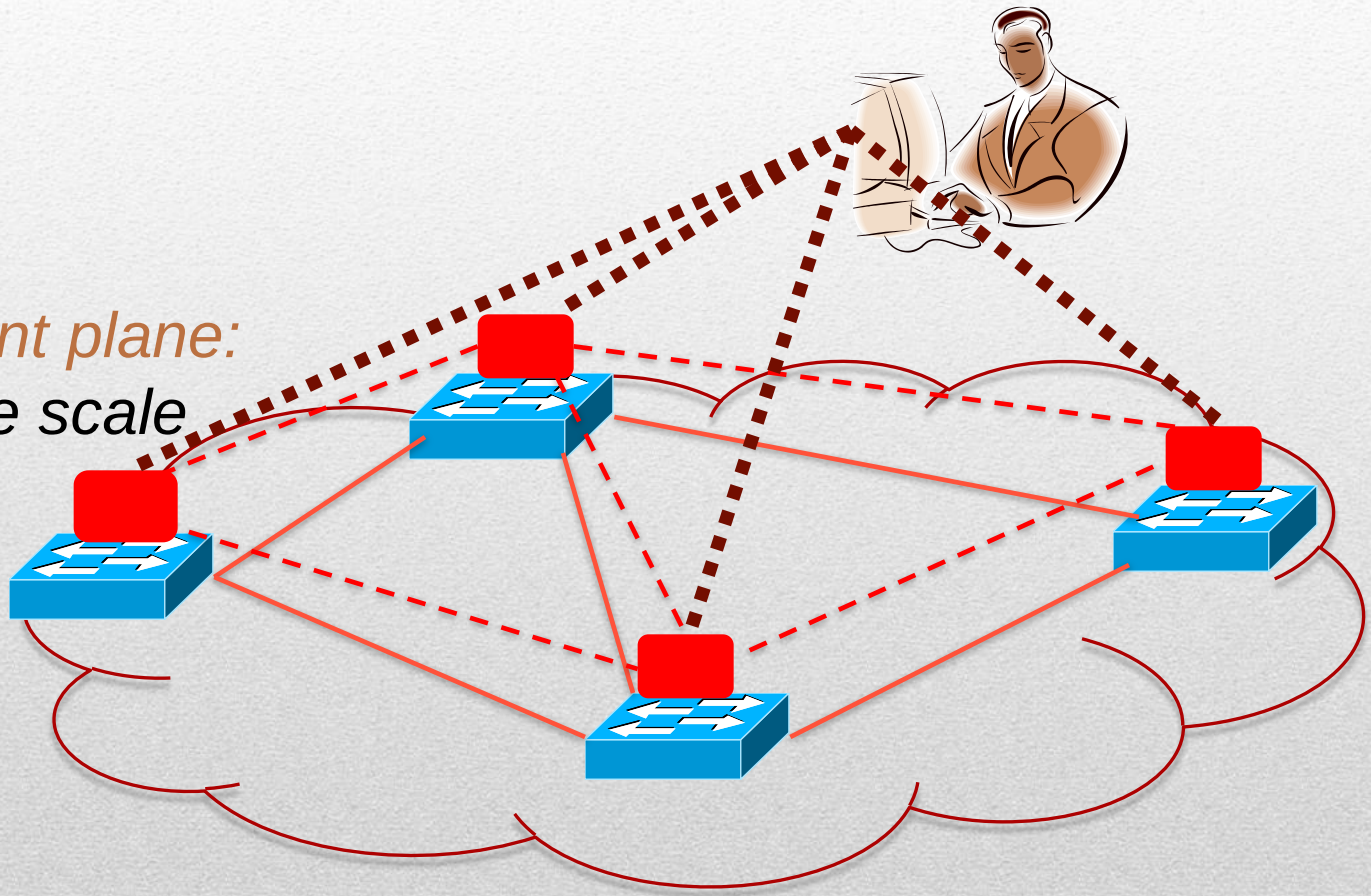
Distributed algorithms



Traditional Computer Networks

Management plane:

Human time scale

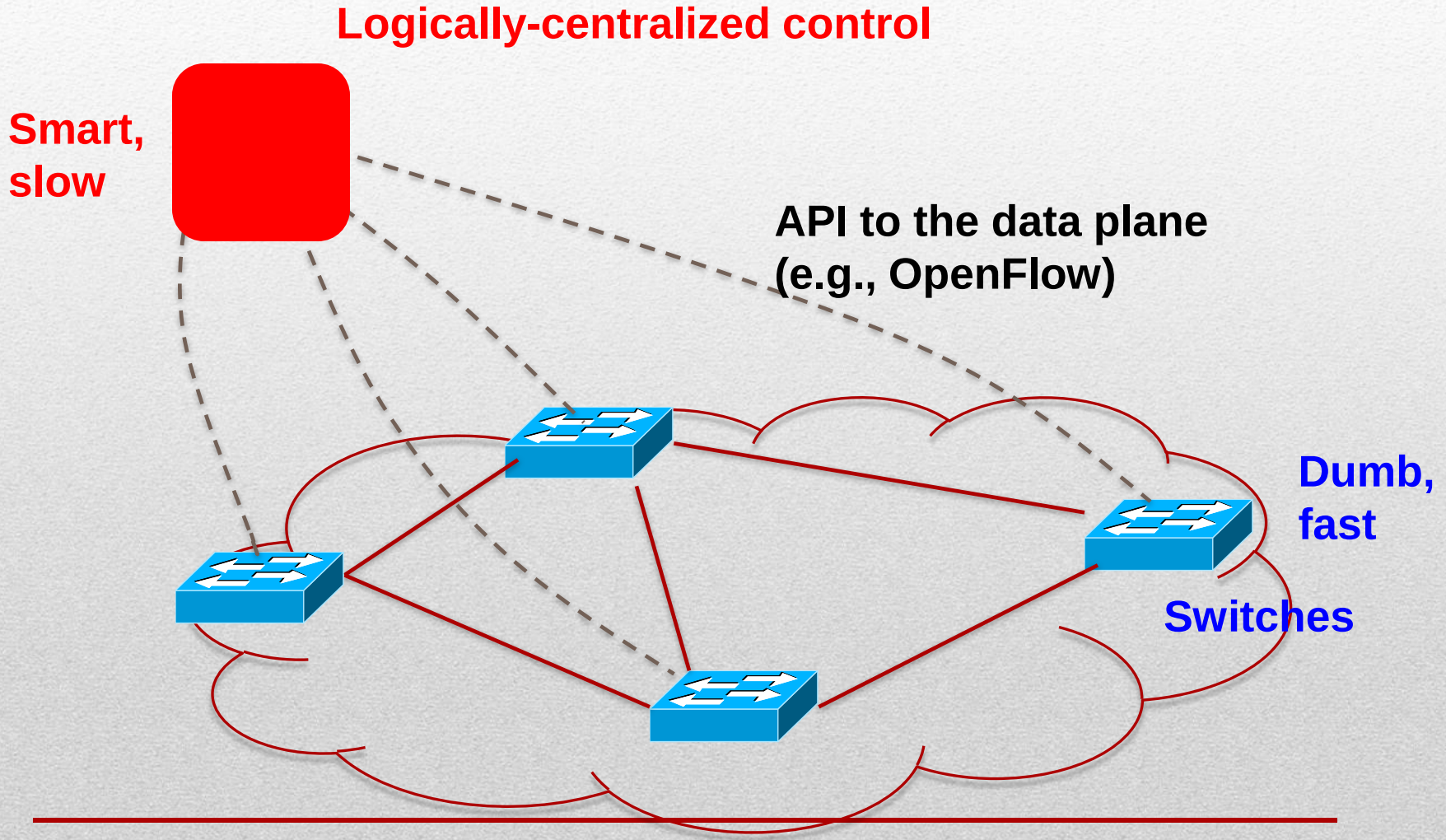


Collect measurements and configure the equipment

No 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

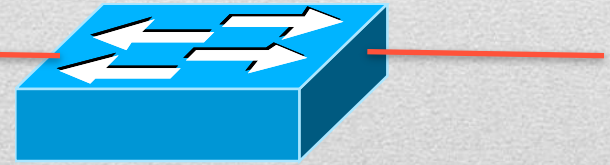


OpenFlow Networks

- **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

OpenFlow Networks

- Simple packet-handling rules
  - Pattern
    - Match packet header bits
  - Actions
    - Drop, forward, modify, send to controller
  - Priority
    - Disambiguate overlapping patterns
  - Counters
    - #bytes and #packets

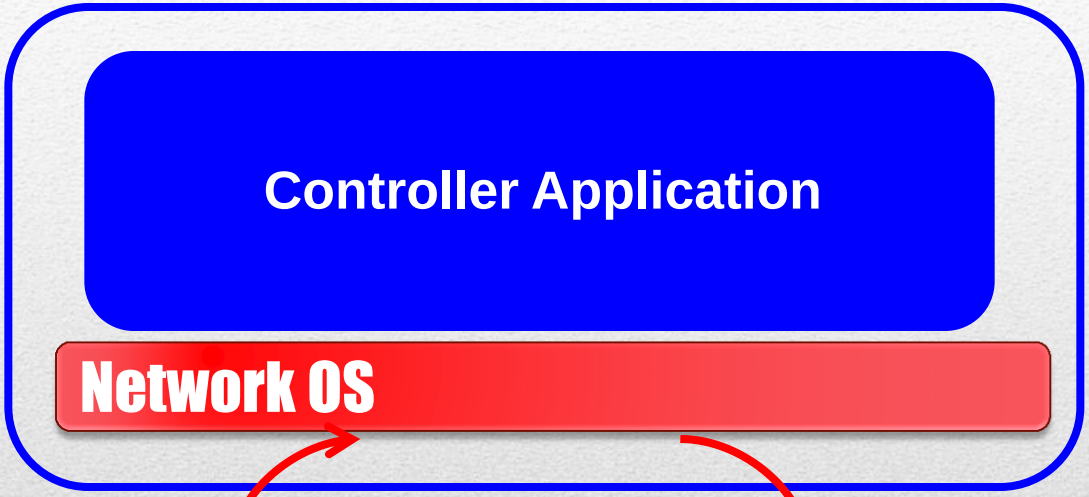


1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

OpenFlow Networks

- 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

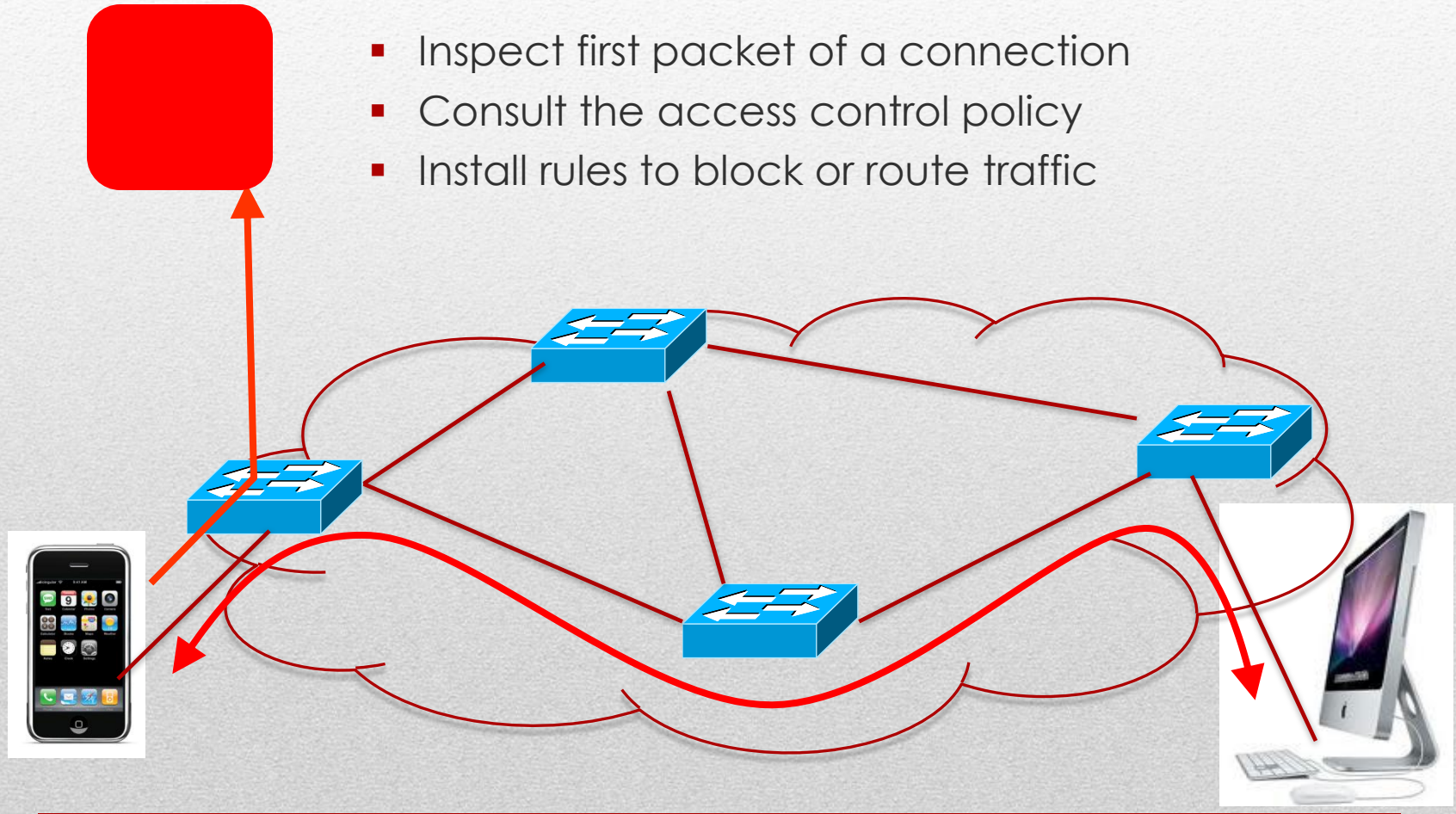
Example OpenFlow Applications

- **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

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

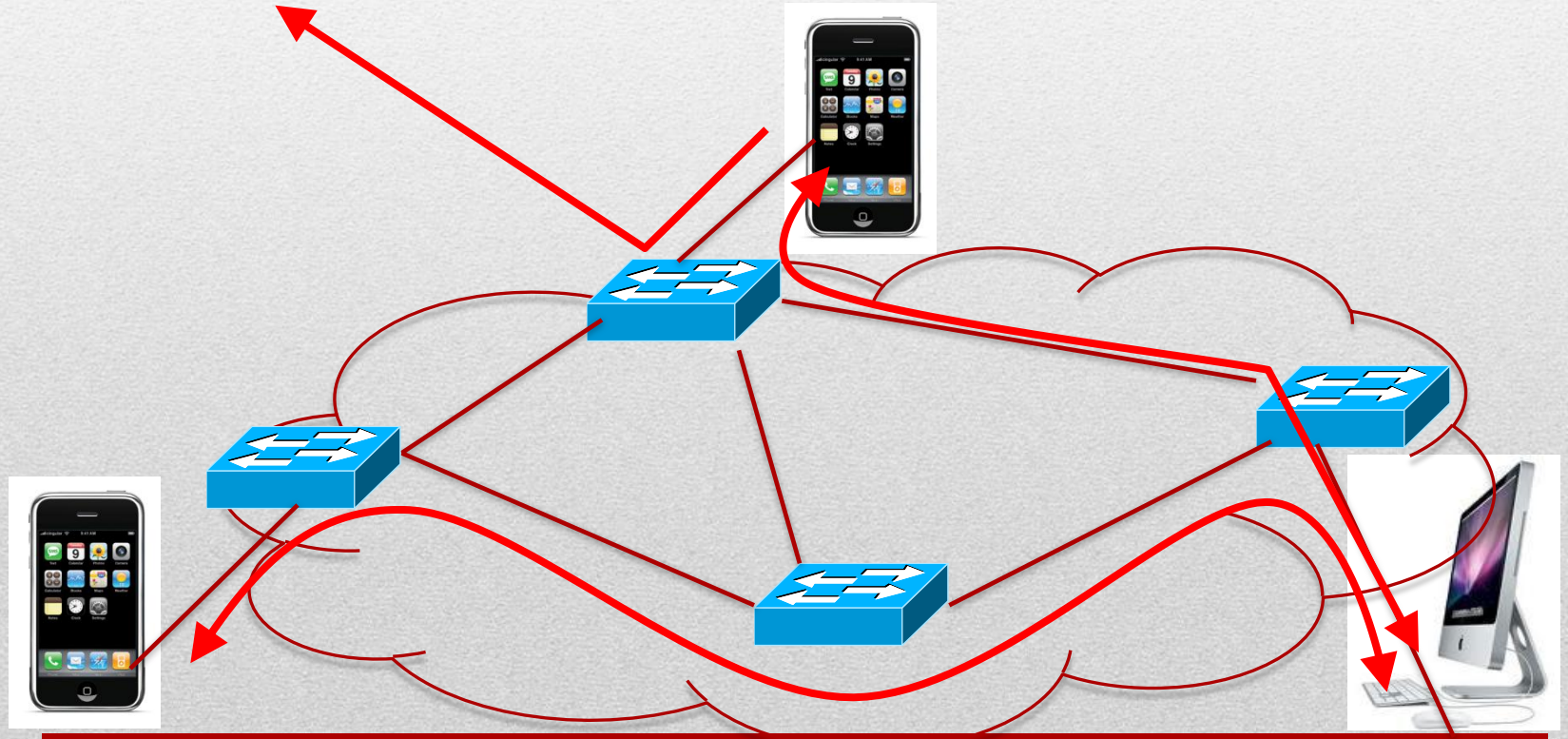
Dynamic Access Control

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



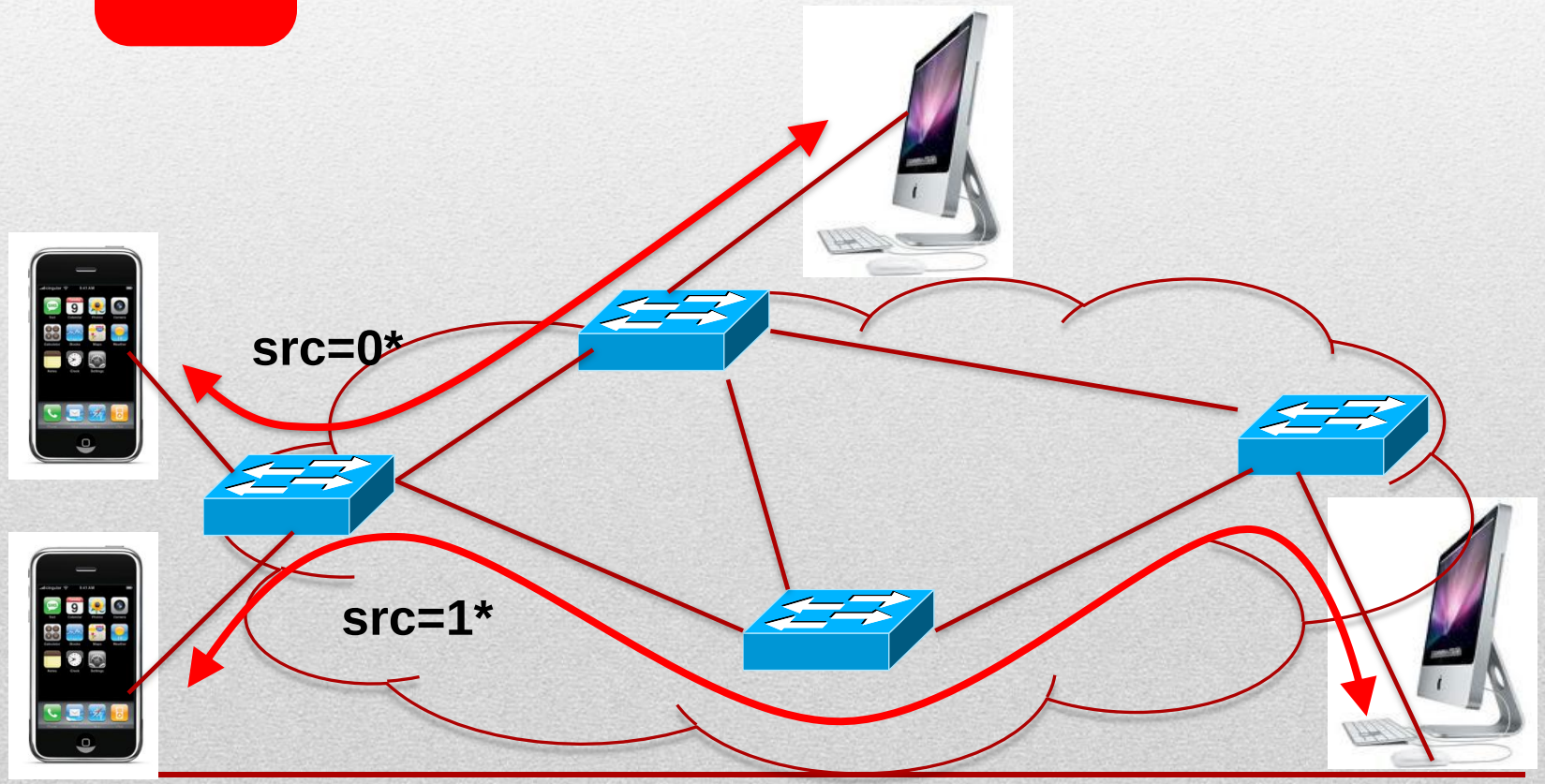
Seamless Mobility Migration

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



Server Load Balancing

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



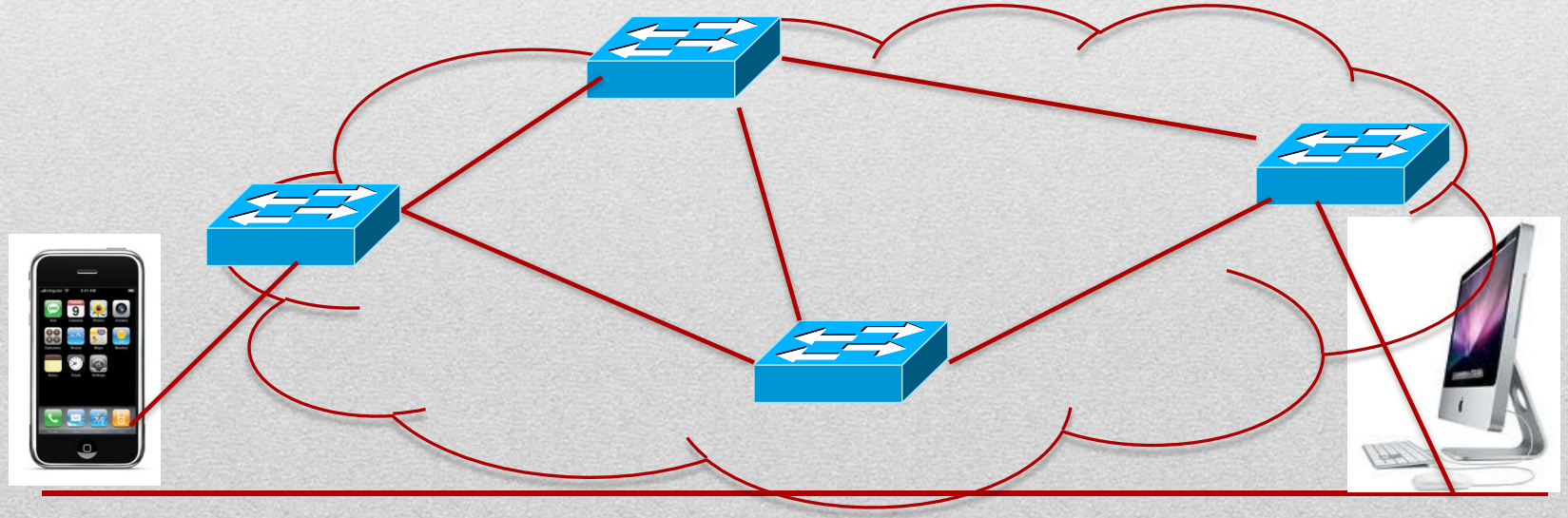
Network Virtualization

Controller #1

Controller #2

Controller #3

Partition the space of packet headers



OpenFlow in the World

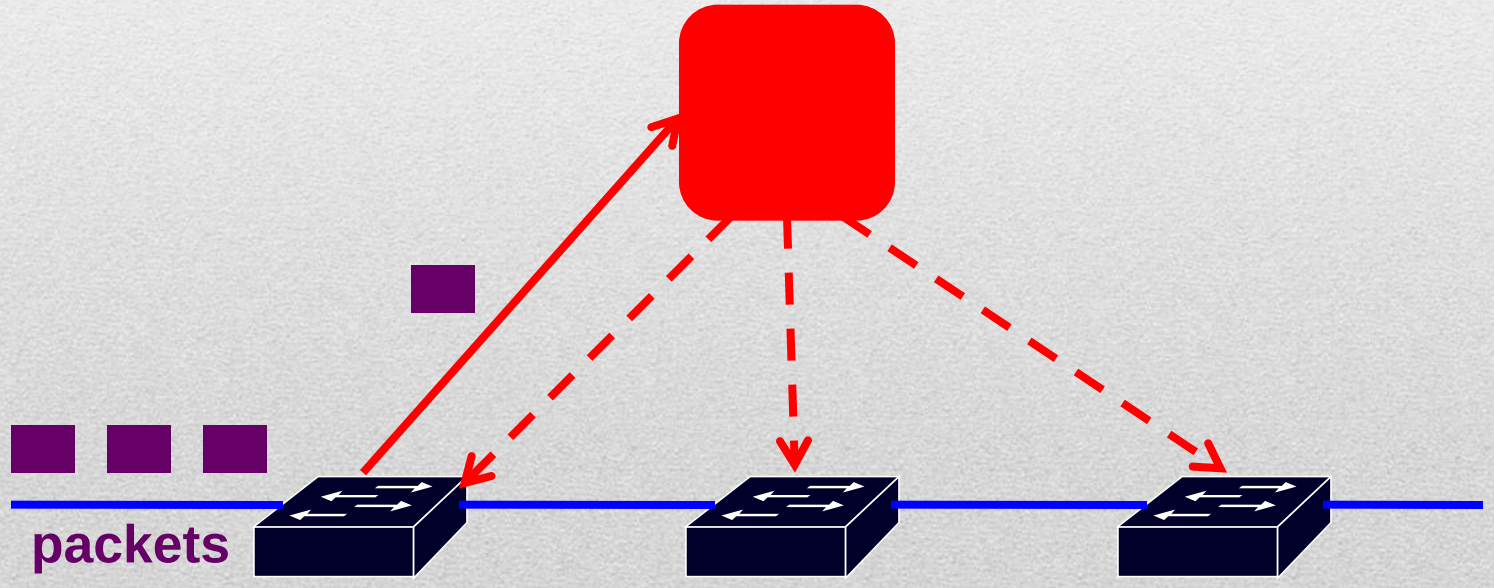
- Open Networking Foundation
  - Google, Facebook, Microsoft, Yahoo, Verizon, Deutsche Telekom, and many other companies
- Commercial OpenFlow switches
  - HP, NEC, Quanta, Dell, IBM, Juniper, …
- Network operating systems
  - NOX, Beacon, Floodlight, Nettle, ONIX, POX, Frenetic
- Network deployments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

Challenges

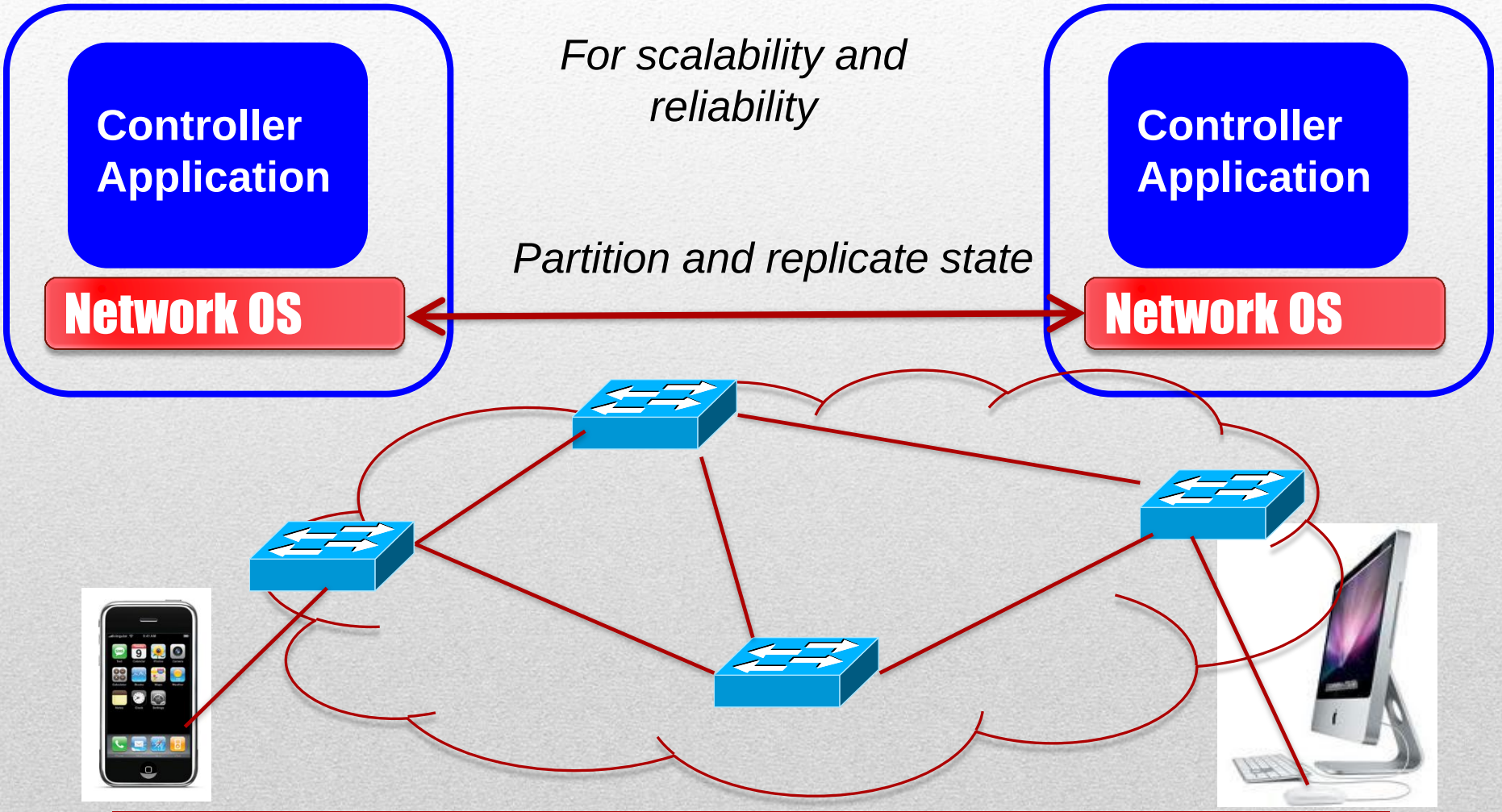
- **Heterogeneous Switches**
  - Number of packet-handling rules
  - Range of matches and actions
  - Multi-stage pipeline of packet processing
  - Offload some control-plane functionality (?)

Controller Delay & Overhead

- **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

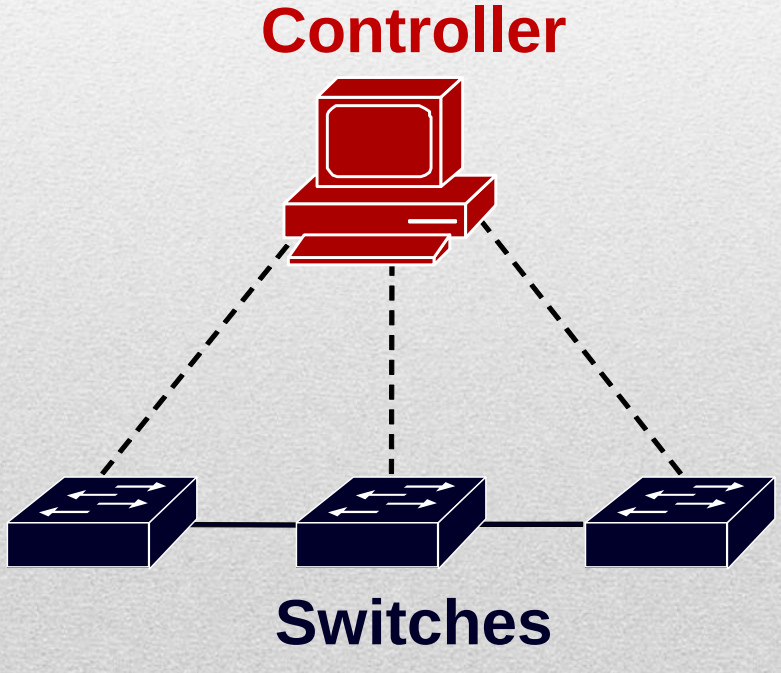


Testing & Debugging

- 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/



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