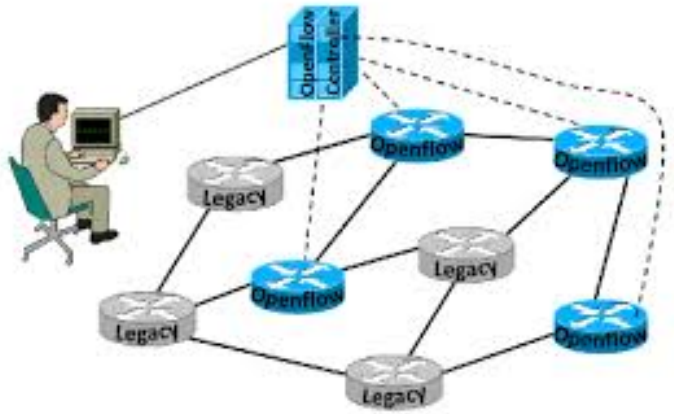
Software-Defined Networking (SDN)



CS 558: Advanced Computer Security

Dong (Kevin) Jin

Some Slides borrowed from

Prof. Jennifer Rexford and Prof. Guofei Gu

Course administrivia

- Paper presentation schedule released
  - https://sites.google.com/site/cs558spring2019/schedule?authuser=0
- Sample project ideas released
  - https://sites.google.com/site/cs558spring2019/sample-projects?authuser=0
- Make-up class needed for 2/20

The Internet: A Remarkable Story

- Tremendous success
  - From research experiment to global infrastructure
- Brilliance of under-specifying
  - Network: best-effort packet delivery
  - Programmable hosts: arbitrary applications
- Enables innovation
  - Apps: Web, P2P, VoIP, social networks, …
  - Links: Ethernet, fiber optics, WiFi, cellular, …
- But, changes are 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

Networks are complex

89% of operators never sure that config changes are bug-free

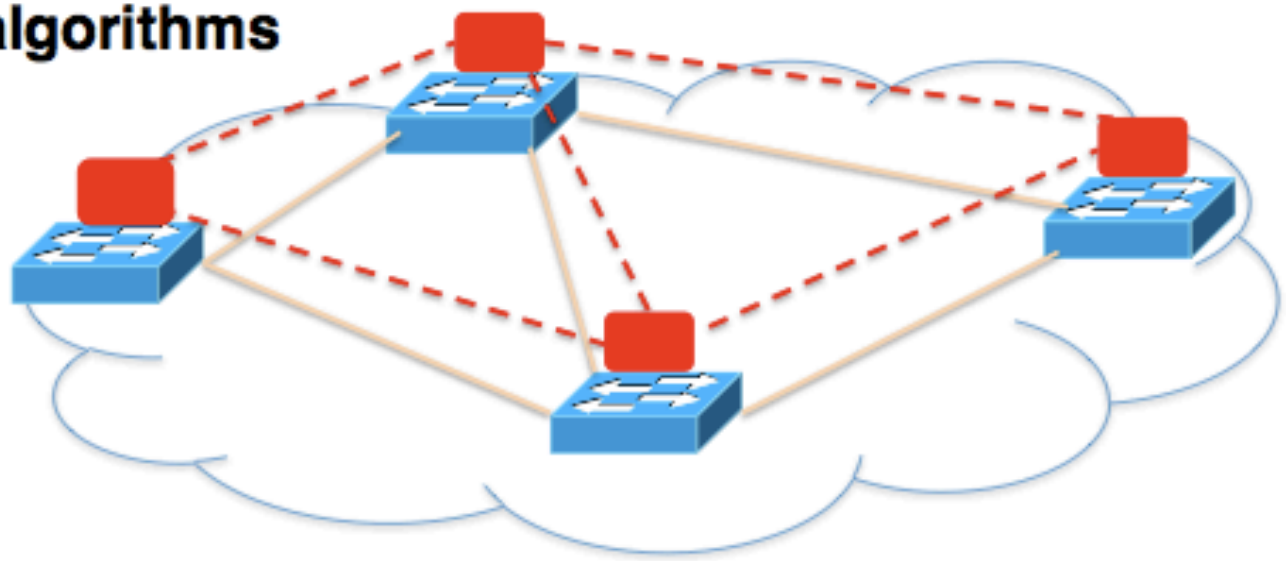
82% concerned that changes would cause problems with existing functionality

Survey of network operators: [Kim, Reich, Gupta, Shahbaz, Feamster, Clark, USENIX NSDI 2015]

Rethinking the "Division of Labor"

Traditional Computer Networks

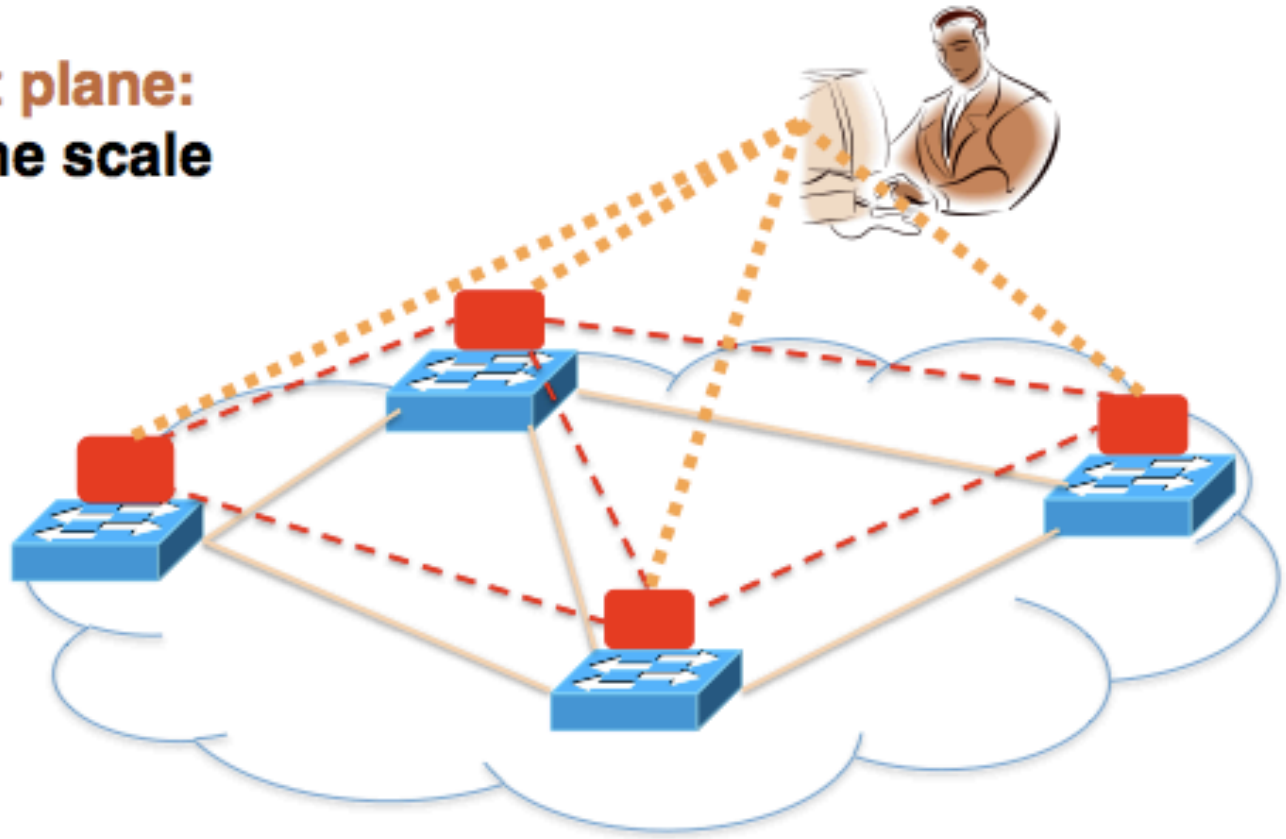
Control plane:
Distributed algorithms



Track topology changes, compute routes, install forwarding rules

Traditional Computer Networks

Management plane:
Human time scale



Collect measurements and configure the equipment

Timescales

	Data	Control	Management
Time-scale	Packet (nsec)		
Tasks	Forwarding, buffering, filtering, scheduling		
Location	Line-card hardware		

Timescales

	Data	Control	Management
Time-scale	Packet (nsec)	Event (10 msec to sec)	
Tasks	Forwarding, buffering, filtering, scheduling	Routing, circuit set-up	
Location	Line-card hardware	Router software	

Timescales

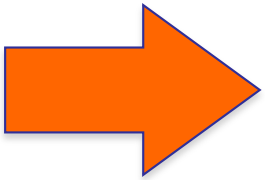
	Data	Control	Management
Time-scale	Packet (nsec)	Event (10 msec to sec)	Human (min to hours)
Tasks	Forwarding, buffering, filtering, scheduling	Routing, circuit set-up	Analysis, configuration
Location	Line-card hardware	Router software	Humans or scripts

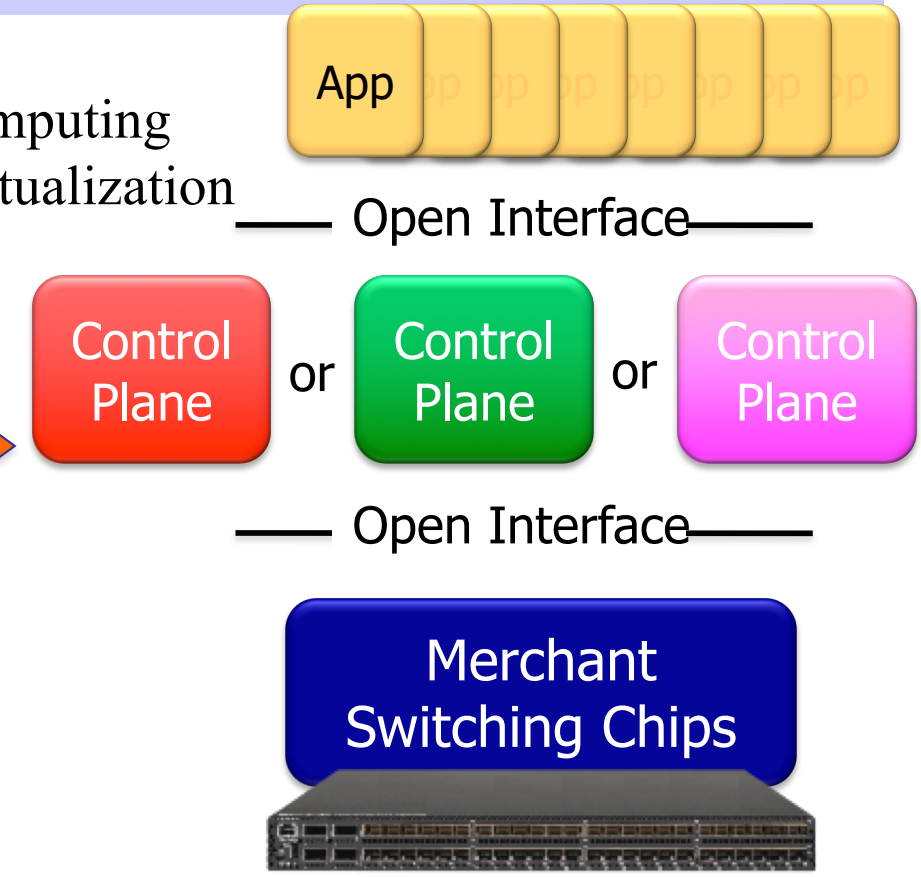
An Ideal Control Plane

- 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

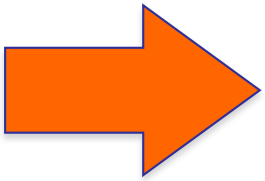
Motivation of SDN

Mobility
Cloud Computing
Server Virtualization
Big Data
…





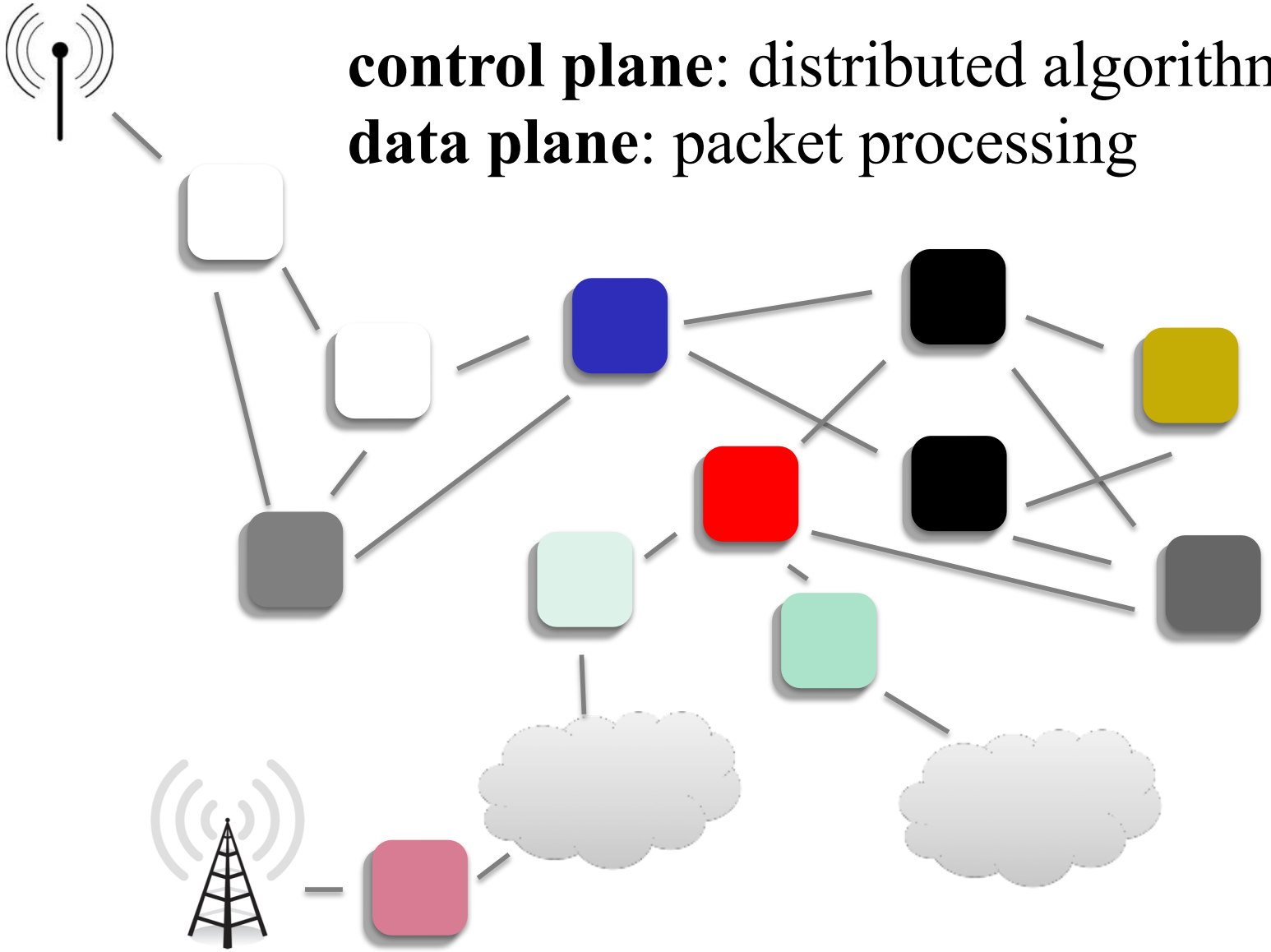
Vertically integrated
Closed, proprietary
Slow innovation



Horizontal
Open interfaces
Rapid innovation

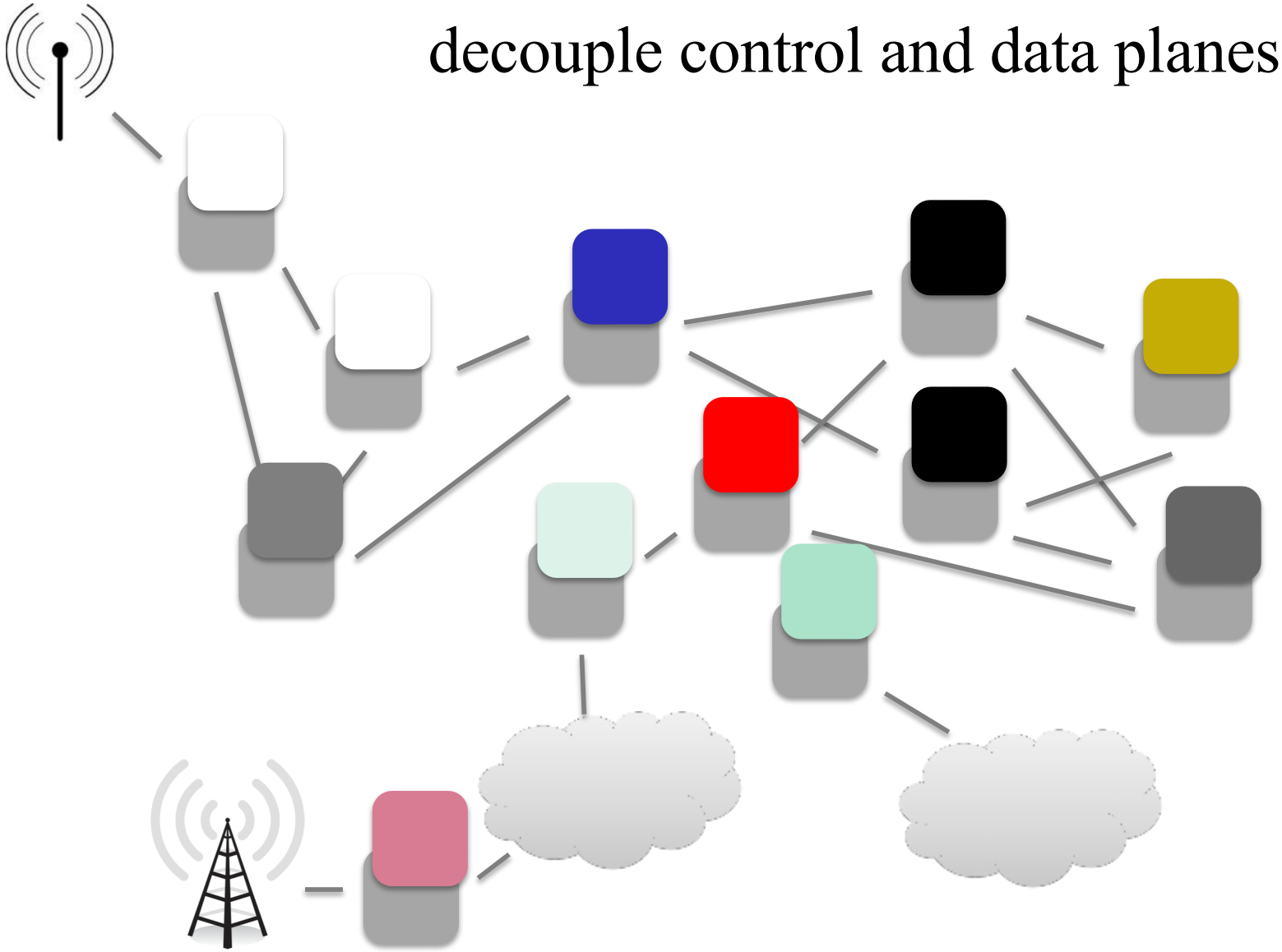
Source: Nick McKeown, Open Networking Summit 2012

Software Defined Networks



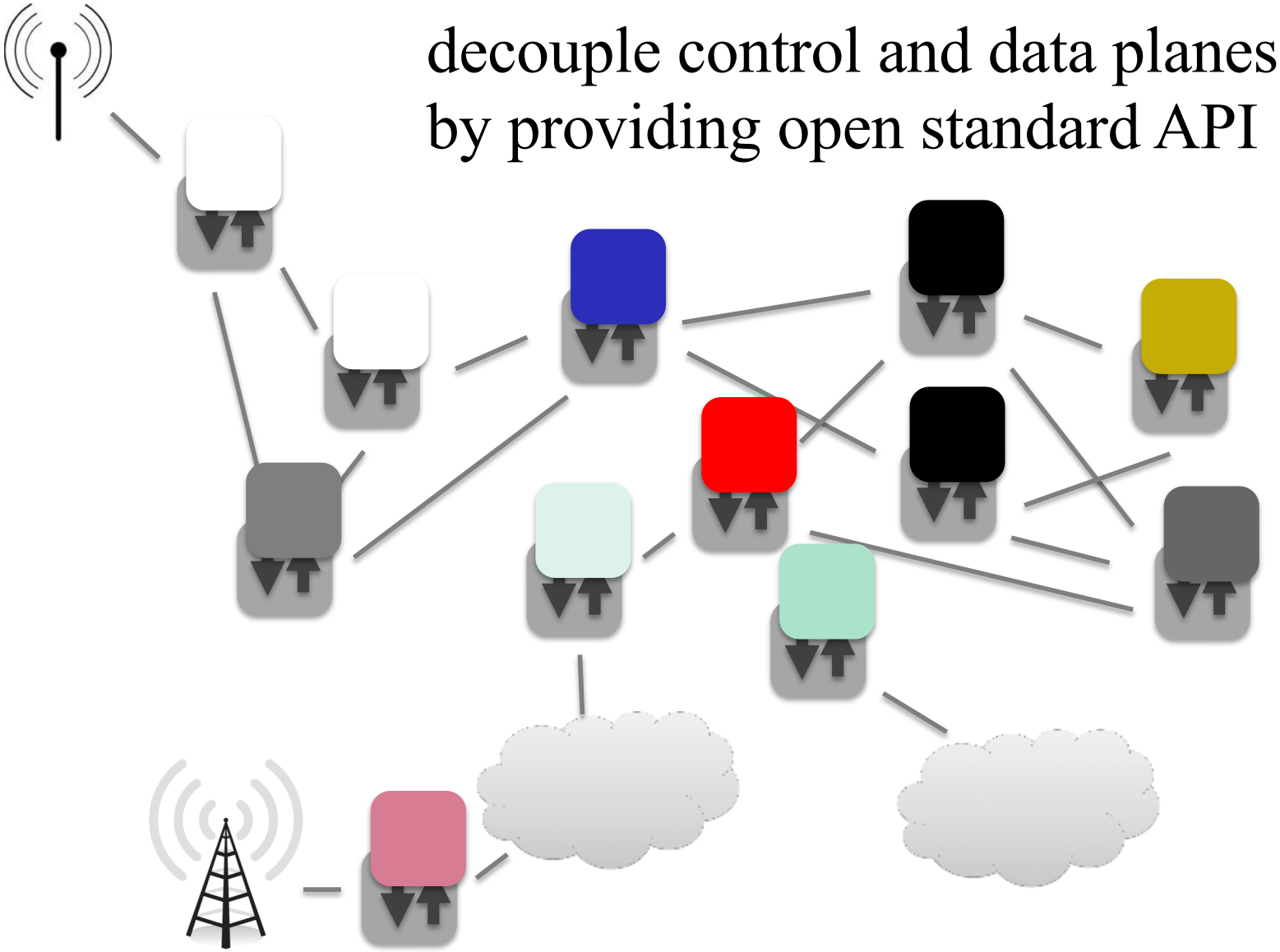
control plane: distributed algorithms
data plane: packet processing

Software Defined Networks

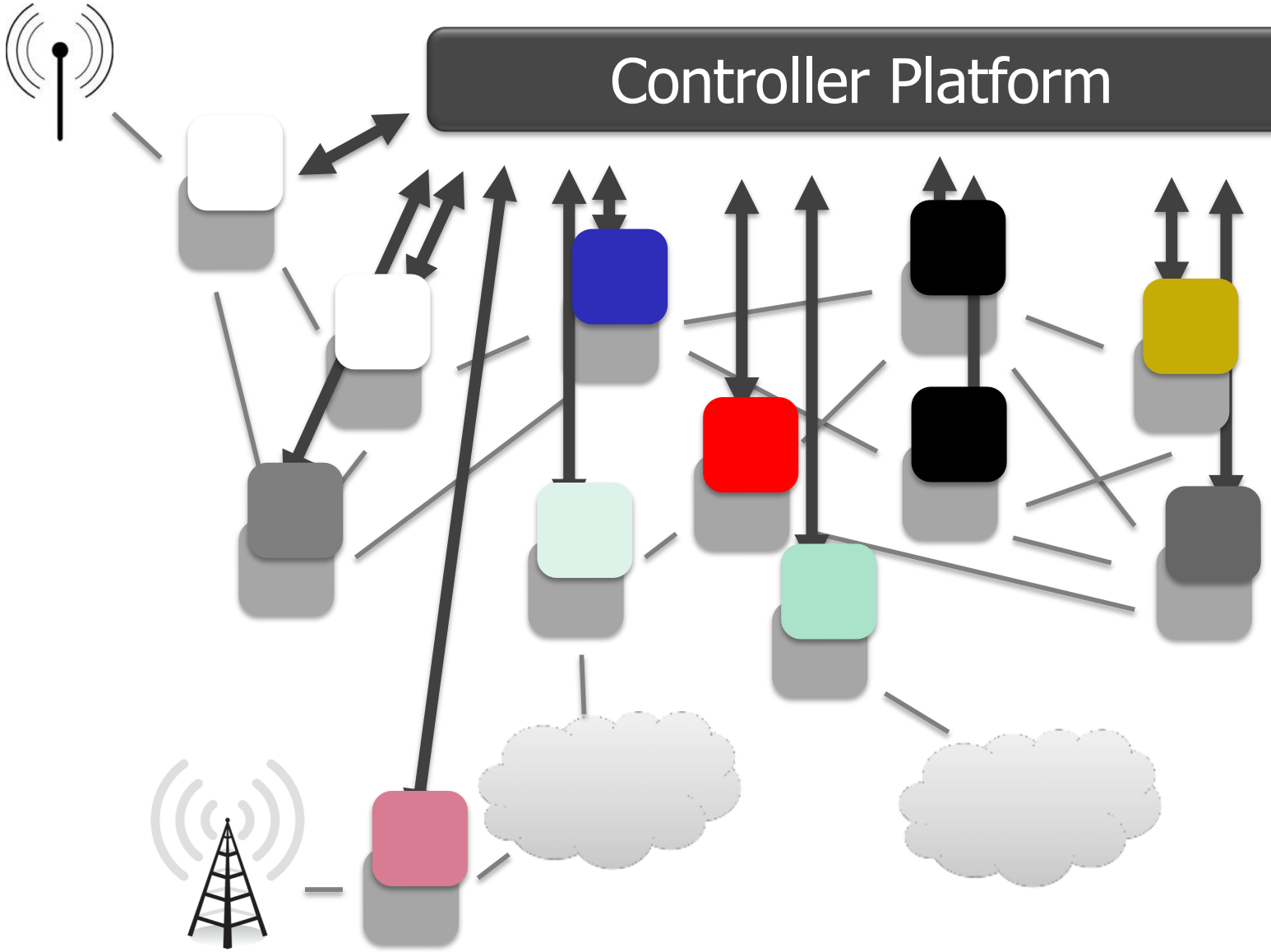


Software Defined Networks

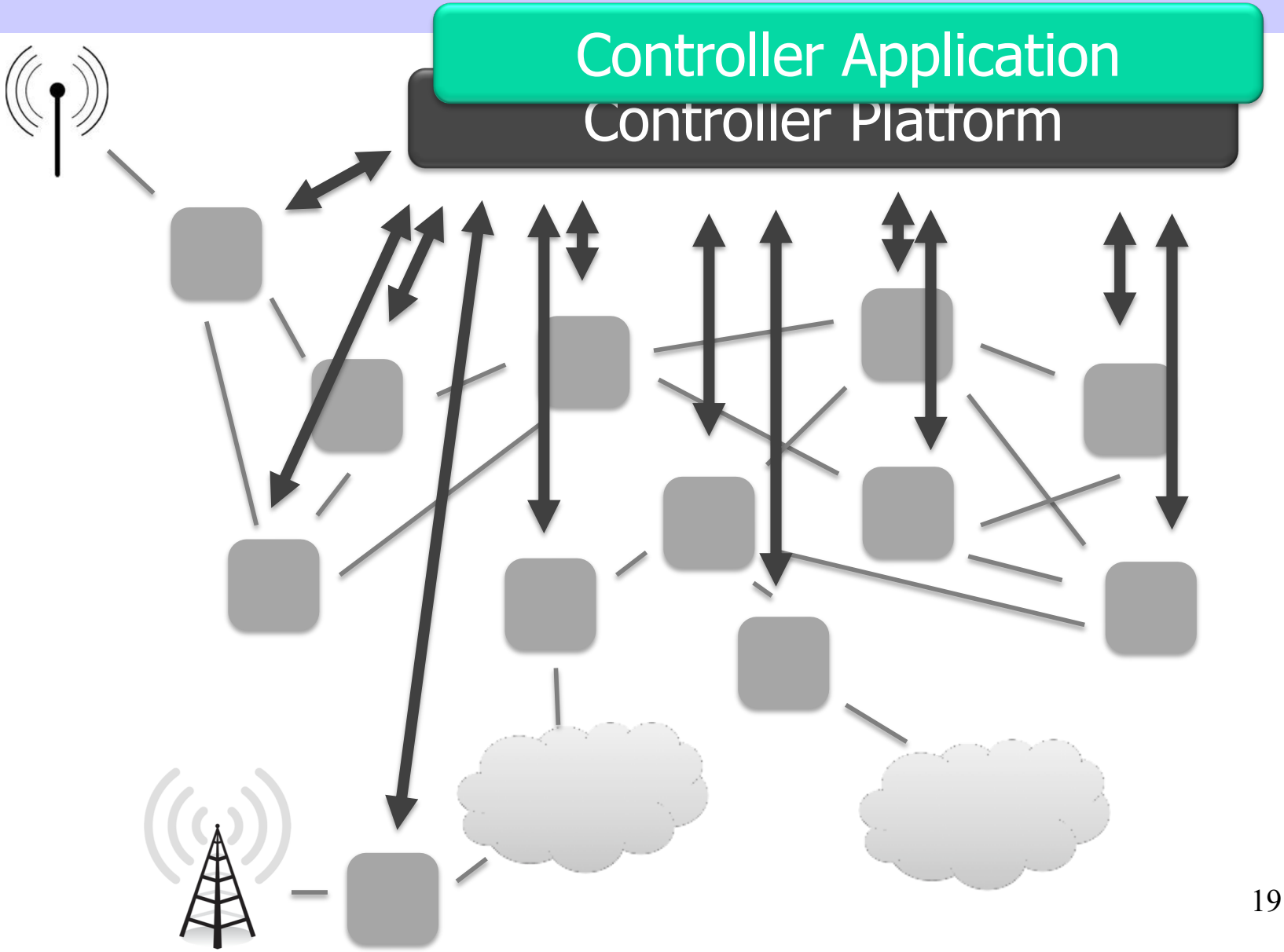
decouple control and data planes
by providing open standard API



(Logically) Centralized Controller

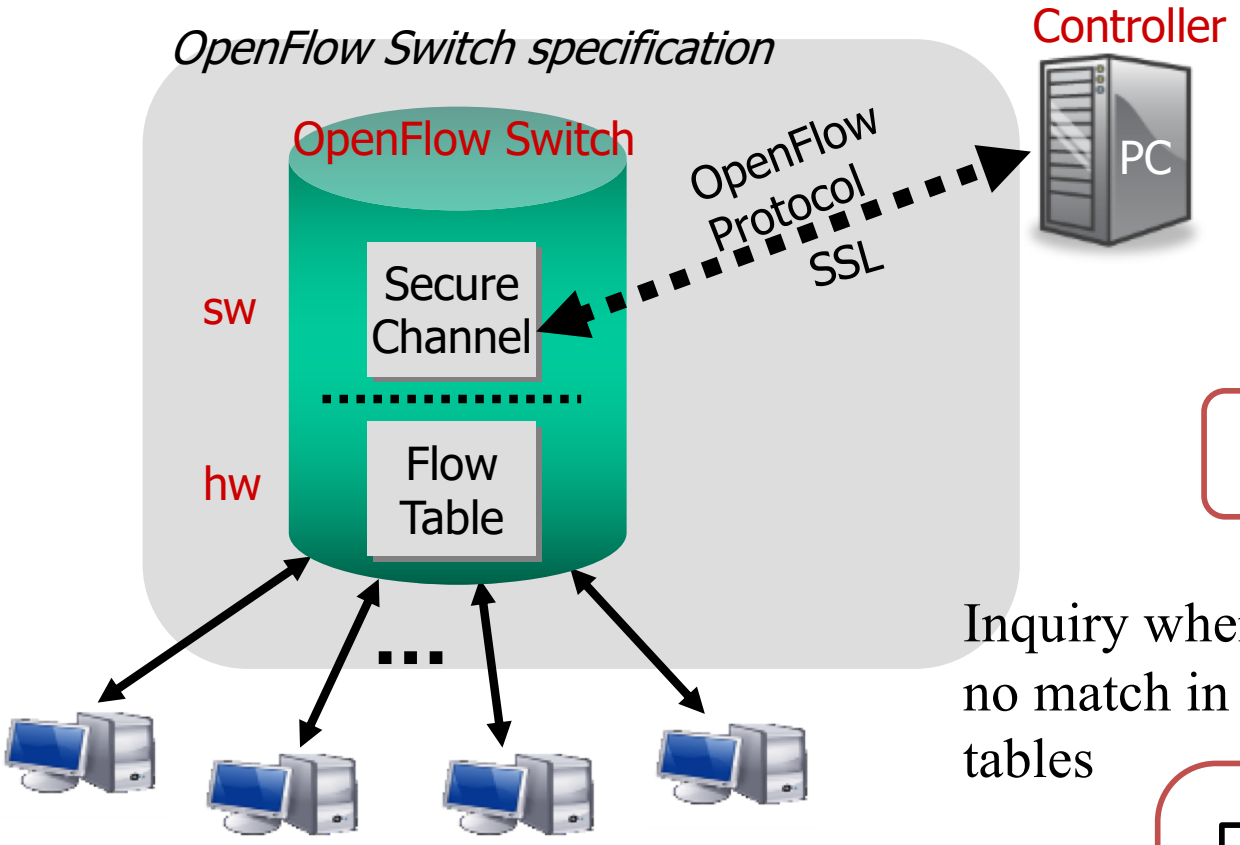


Protocols → Applications

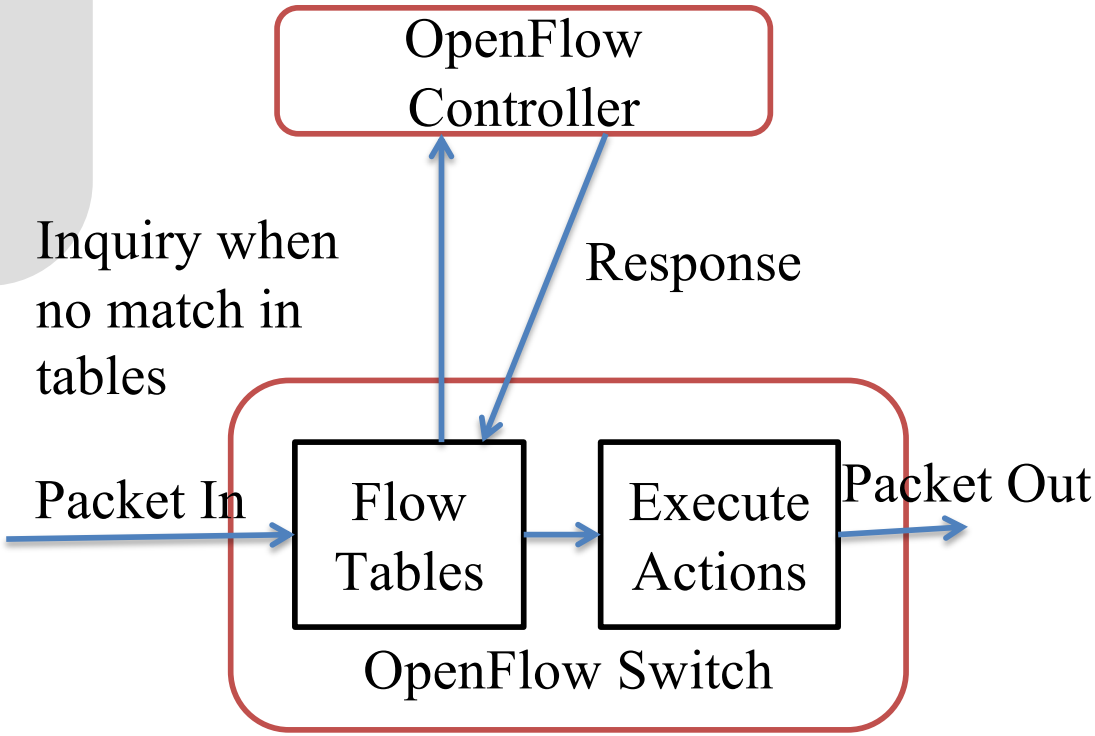


OpenFlow Networks

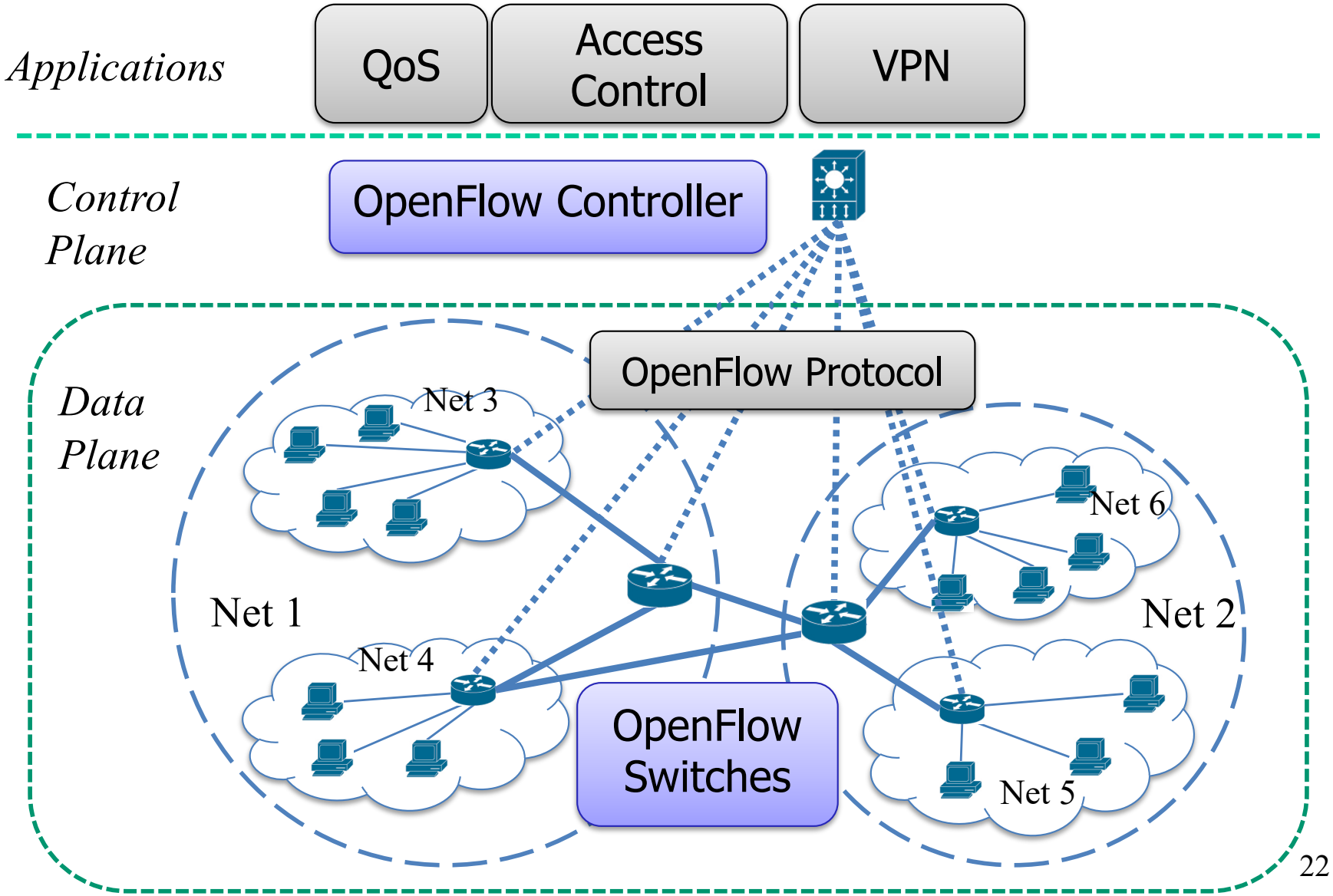
OpenFlow Switching



Picture source: The Stanford Clean Slate Program, http://cleanslate.stanford.edu



OpenFlow-based SDN Architecture

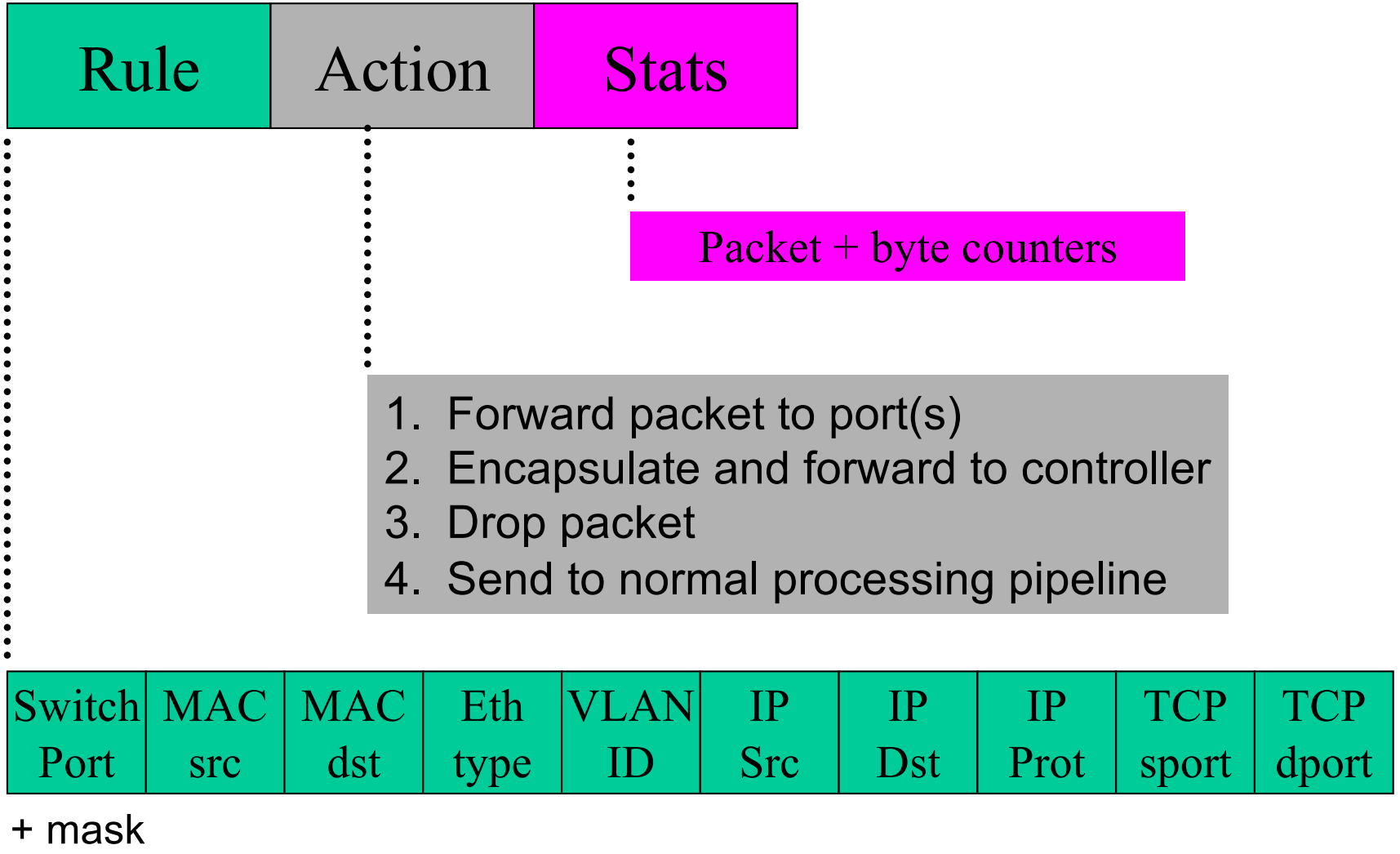


Simple, Open Data-Plane API

- Prioritized list of 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

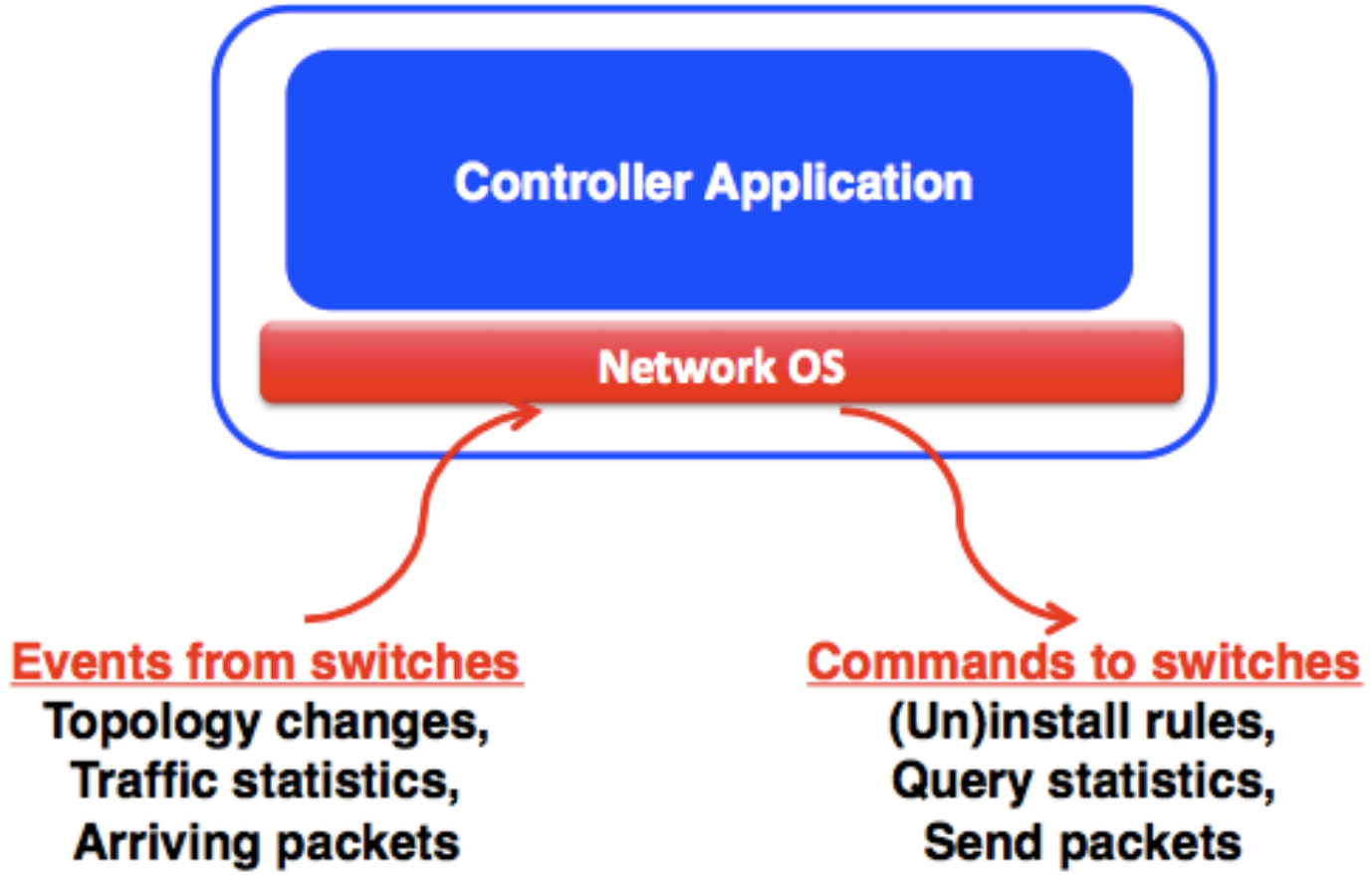
Flow Table Entry
"Type 0" OpenFlow Switch



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



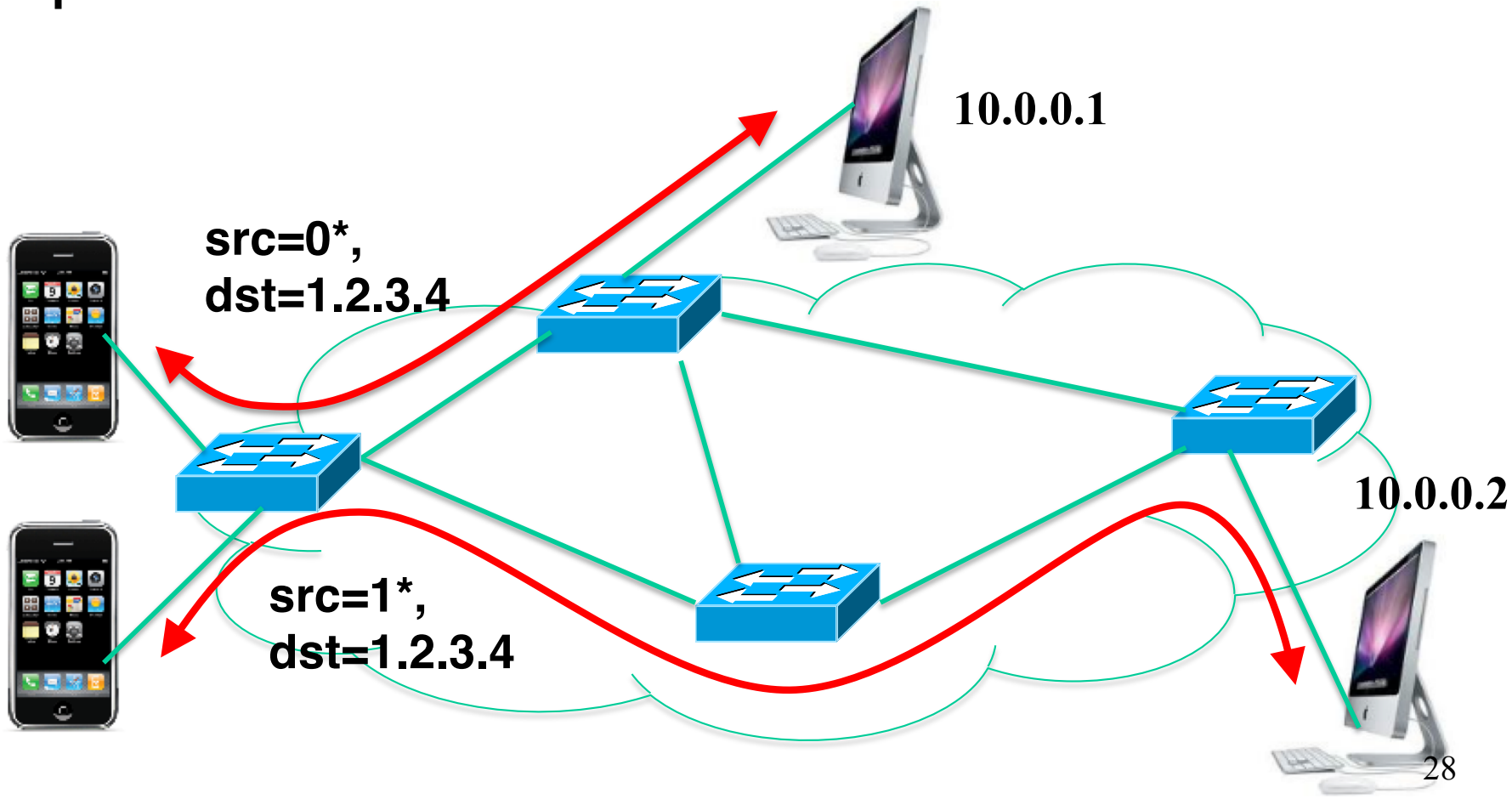
Example SDN Applications

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

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

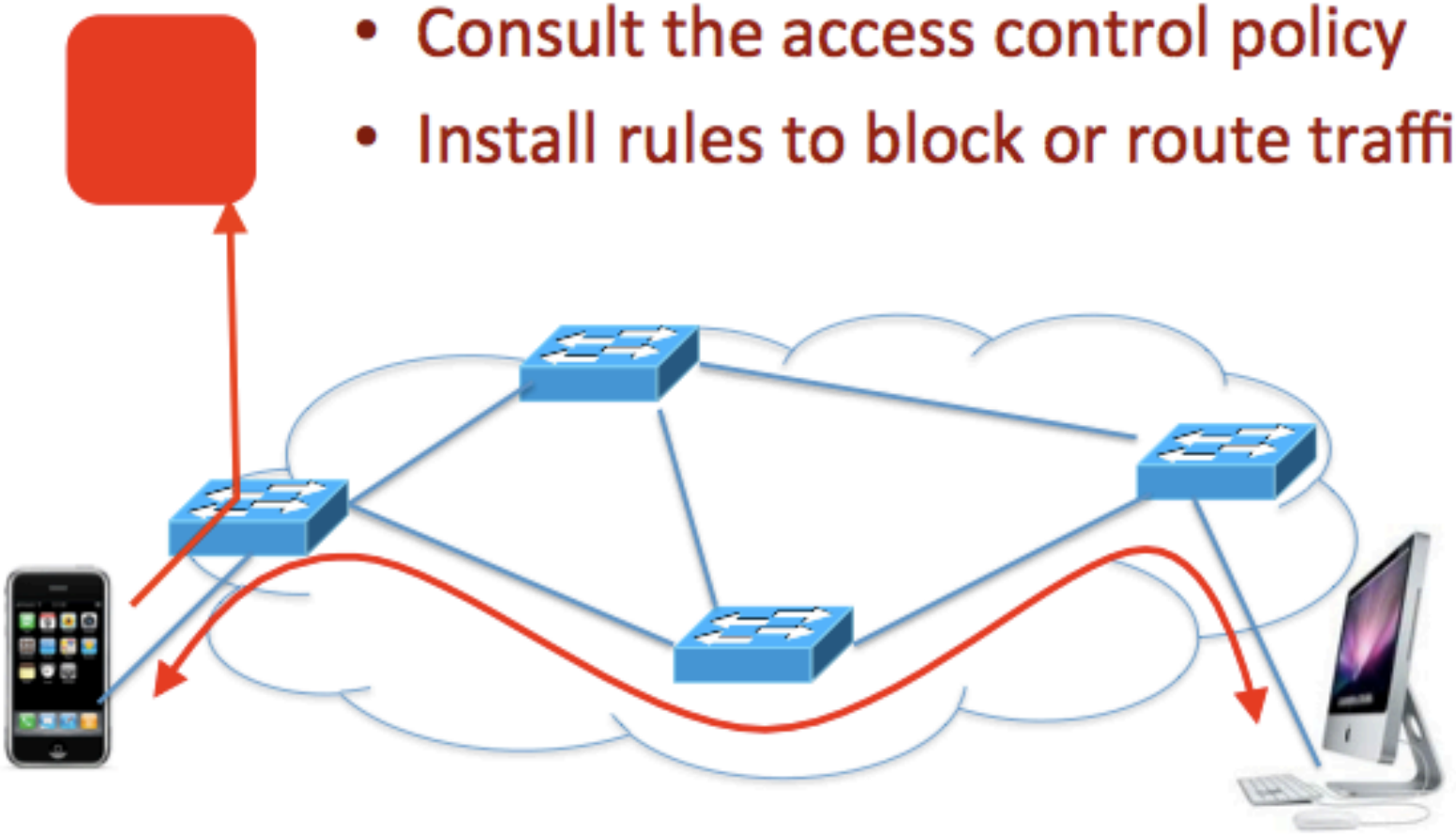
Example: Server Load Balancing

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

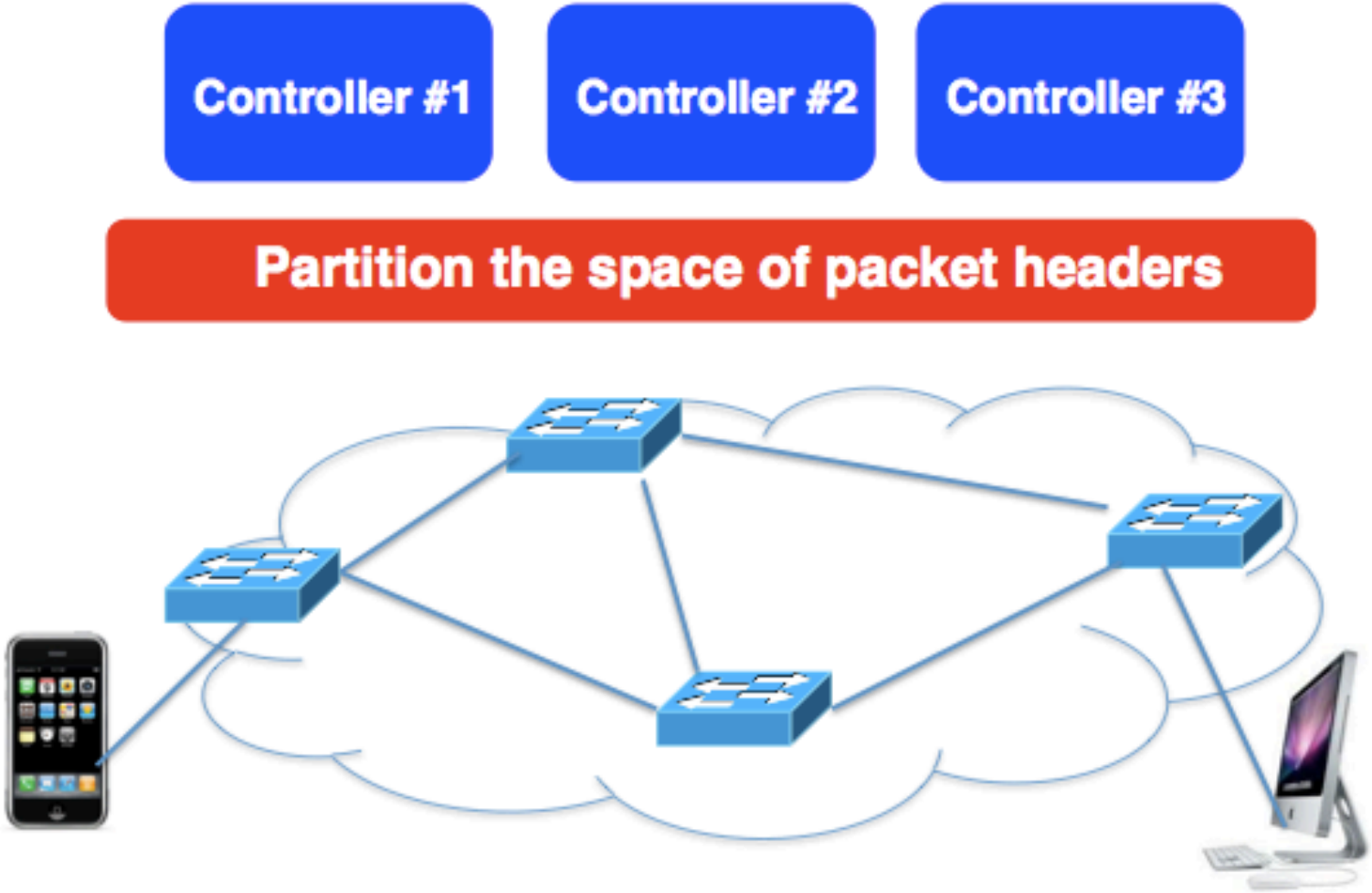


E.g.: Dynamic Access Control

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

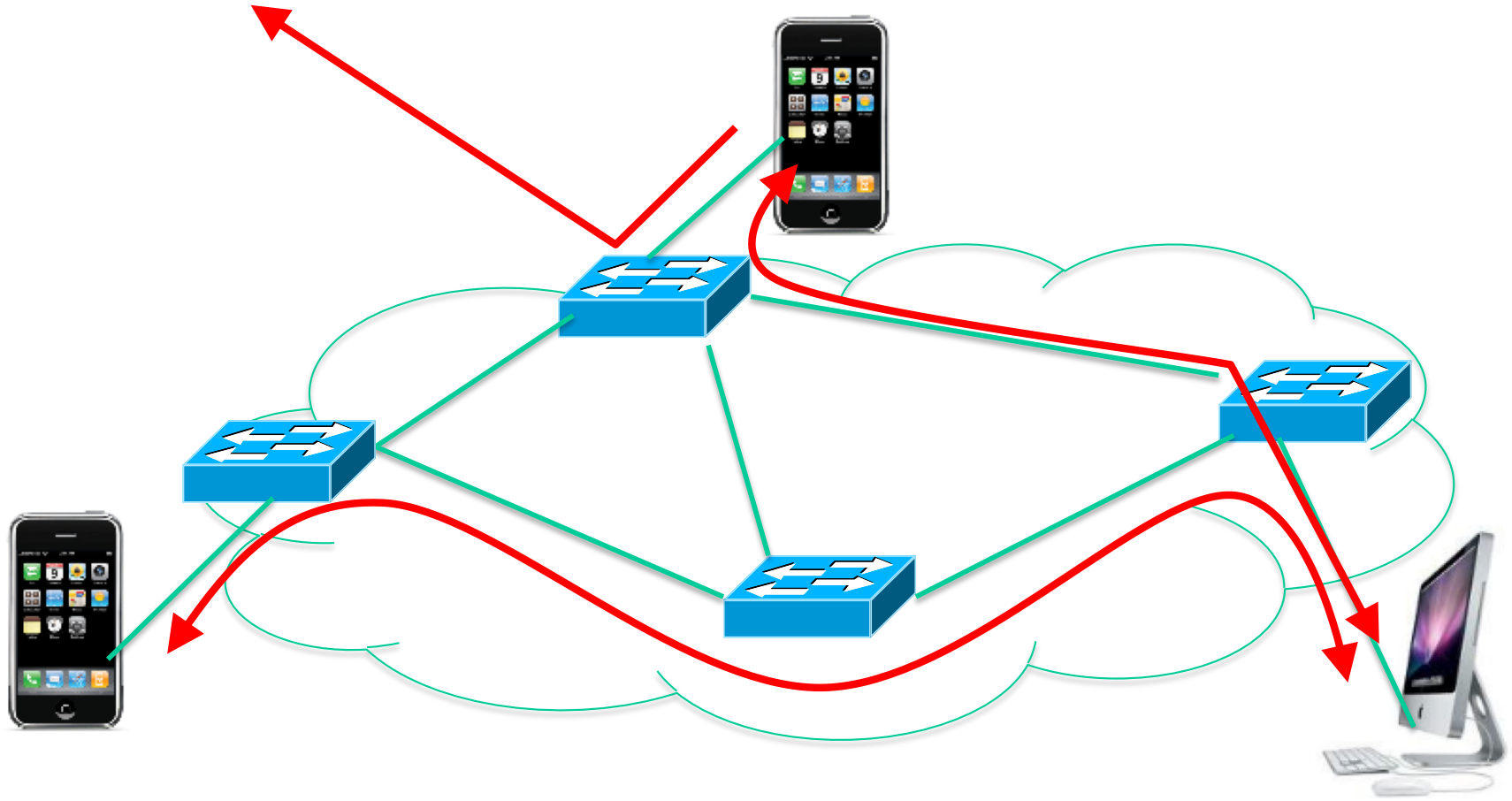


E.g.: Network Virtualization



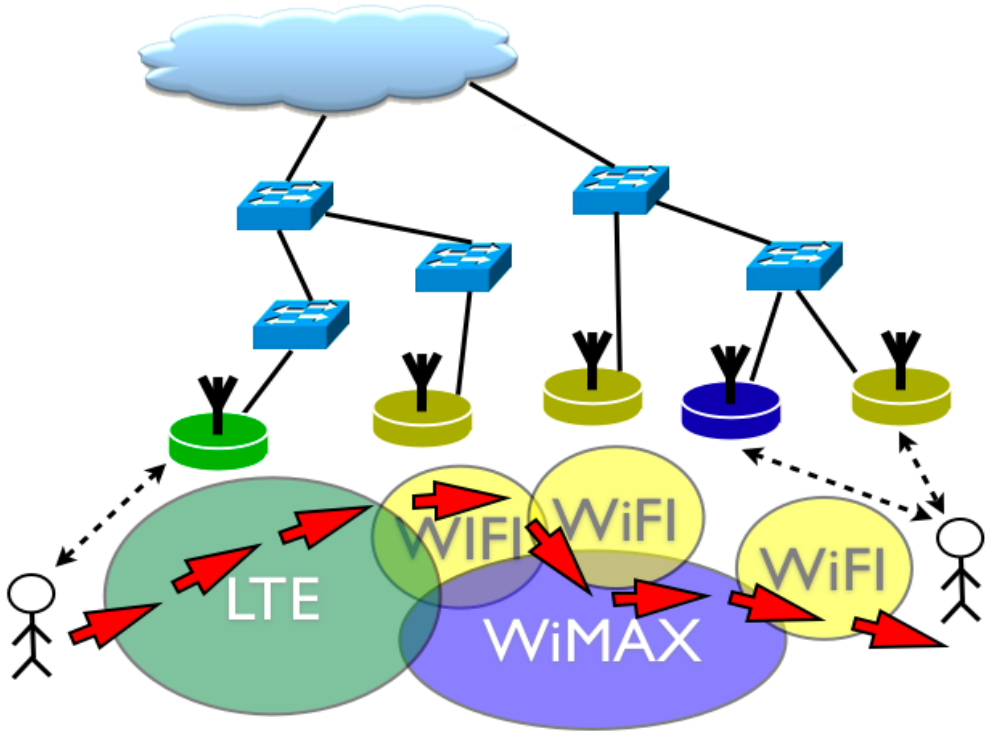
Example: Seamless Mobility

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



More Application?

Example : OpenFlow Wireless



- Separate the network service from the underlying physical infrastructure
- Possibly get rid of IP in mobile Internet (fixed architecture)

Picture Source: Blueprint for Introducing Innovation into the Wireless Networks we use every day

Videos of OpenFlow Projects

- OpenRoads
- FlowVisor

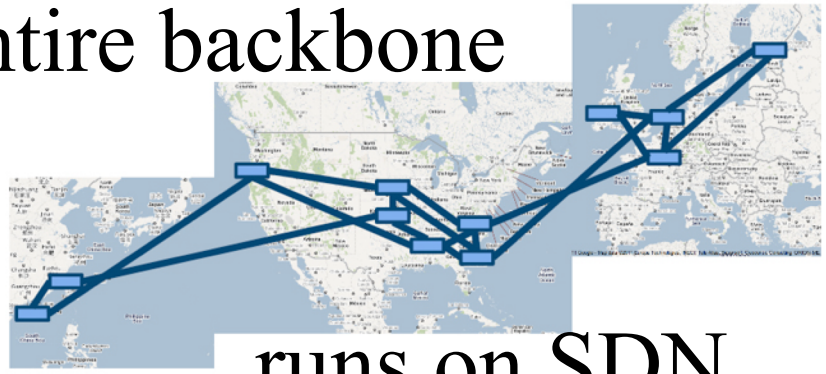
OpenFlow in the Wild

- 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 Major Trend in Networking

Google

Entire backbone



runs on SDN

nicira

Bought for **$1.2 x 10^9**
(mostly cash)

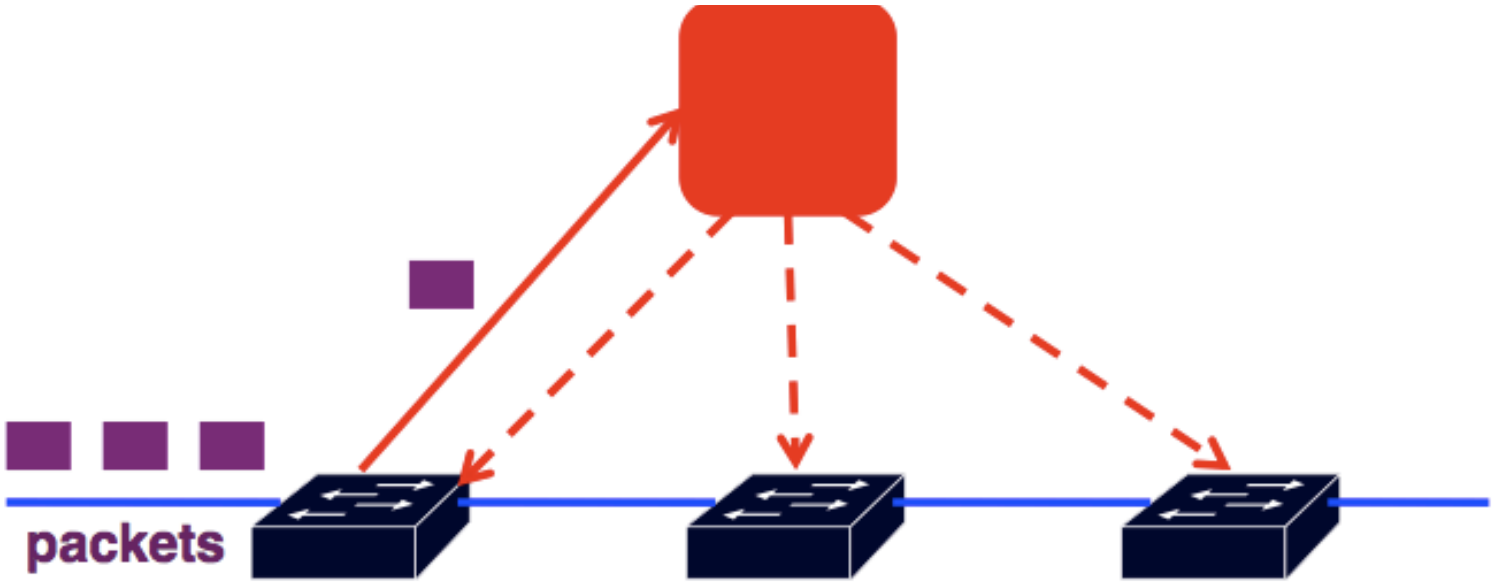
Challenges

Heterogeneous Switches

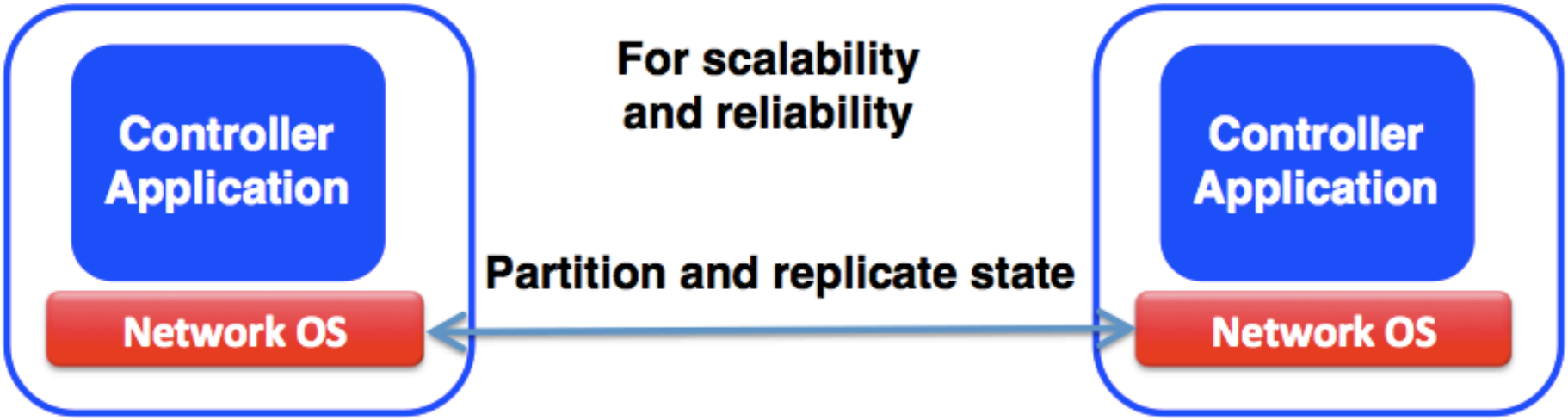
- Number of packet-handling rules
- Range of matches and actions
- Multi-stage pipeline of packet processing
- Offload some control-plane functionality (?)

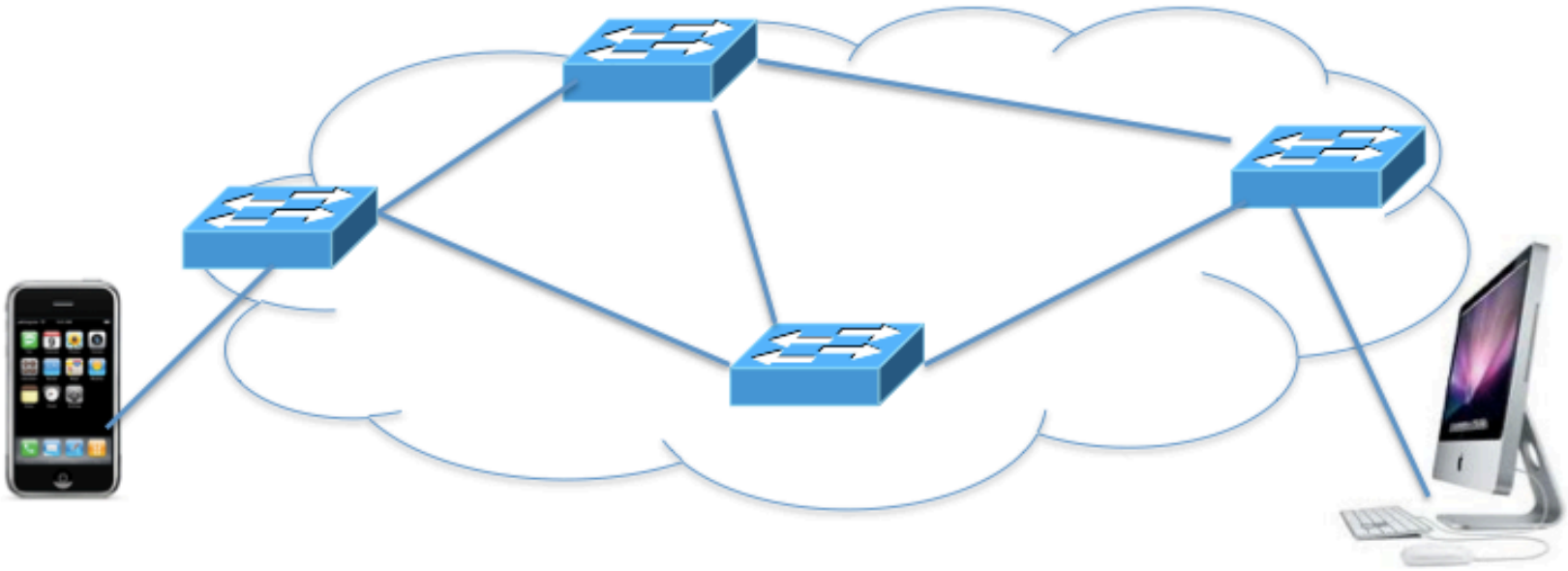
Controller Delay and Overhead

- Controller is much slower than 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

An Opportunity to Rethink

- How should future networks be
  - Designed
  - Managed
  - Programmed
- What are the right abstractions
  - Simple
  - Powerful
  - Reusable

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

Scott Shenker's ONS'11 talk
https://www.youtube.com/watch?v=YHeyuD89n1Y&feature=youtu.be

Discuss Questions

1. Any security issues for OpenFlow switch?

2. How to build a secure and resilient OpenFlow switch?

3. Does centralized solution always apply well in networking?