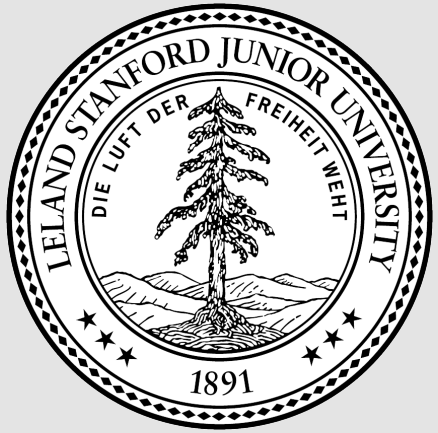


Software Defined Networks and OpenFlow

NANOG 50, October 2010

Nick McKeown
nickm@stanford.edu

With Martin Casado and Scott Shenker
And contributions from many others

Supported by NSF, Stanford Clean Slate Program, Cisco, DoCoMo, DT, Ericsson, Google, NEC, Xilinx

Original question

Q: Can we help students on college campuses to test out new ideas in a real network, at scale?

Problem

- Many good research ideas on college campuses
- No way to test new ideas at scale, on real networks, with real user traffic
- Result: Almost no technology transfer

Example Ideas

- Improvements to BGP, multicast, anycast, Mobile IP, data center networks such as VL2, Portland.
- Access control, energy management, workload/ traffic optimization, VM mobility, …

Build a programmable testbed?

Problems

- Special hardware is expensive or unrealistic
- Buildout at scale is too expensive
- Hard to get users to opt-in

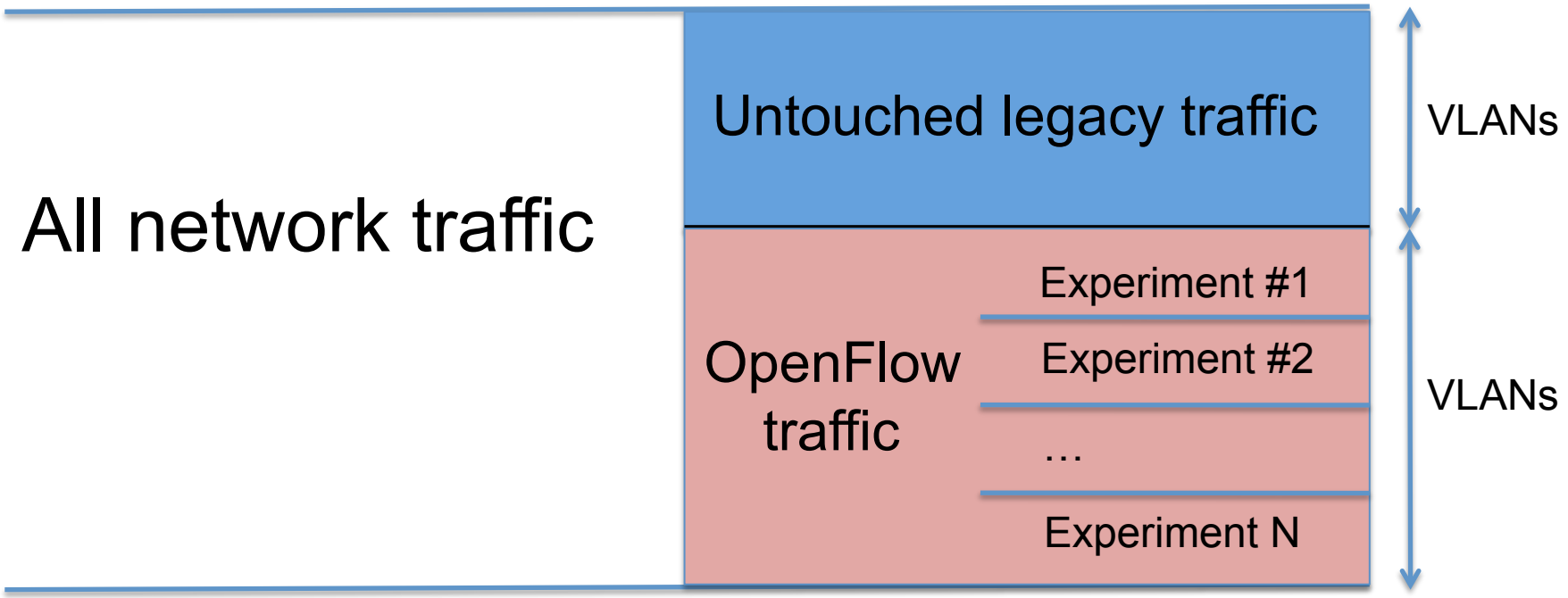
Our approach

- Add the “testbed capability” to existing hardware, then ride on the coat-tails of new deployments

Goals

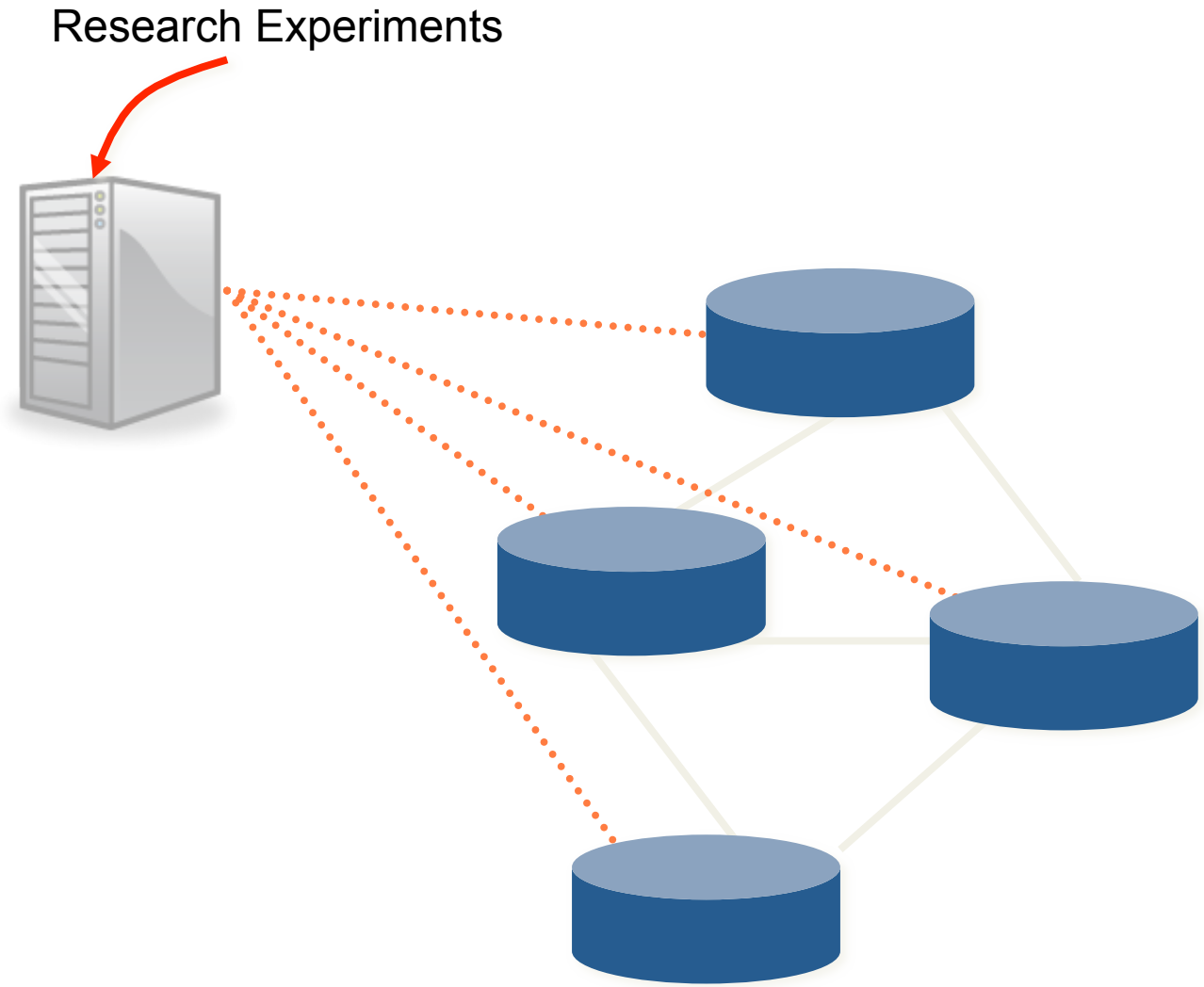
1. Enable deployment of new/experimental network services in a production network. Real traffic, real users, over real topologies at real line-rates.
2. Real network silicon/hardware.
3. Allow users to opt-in to experimental services.

Slicing traffic



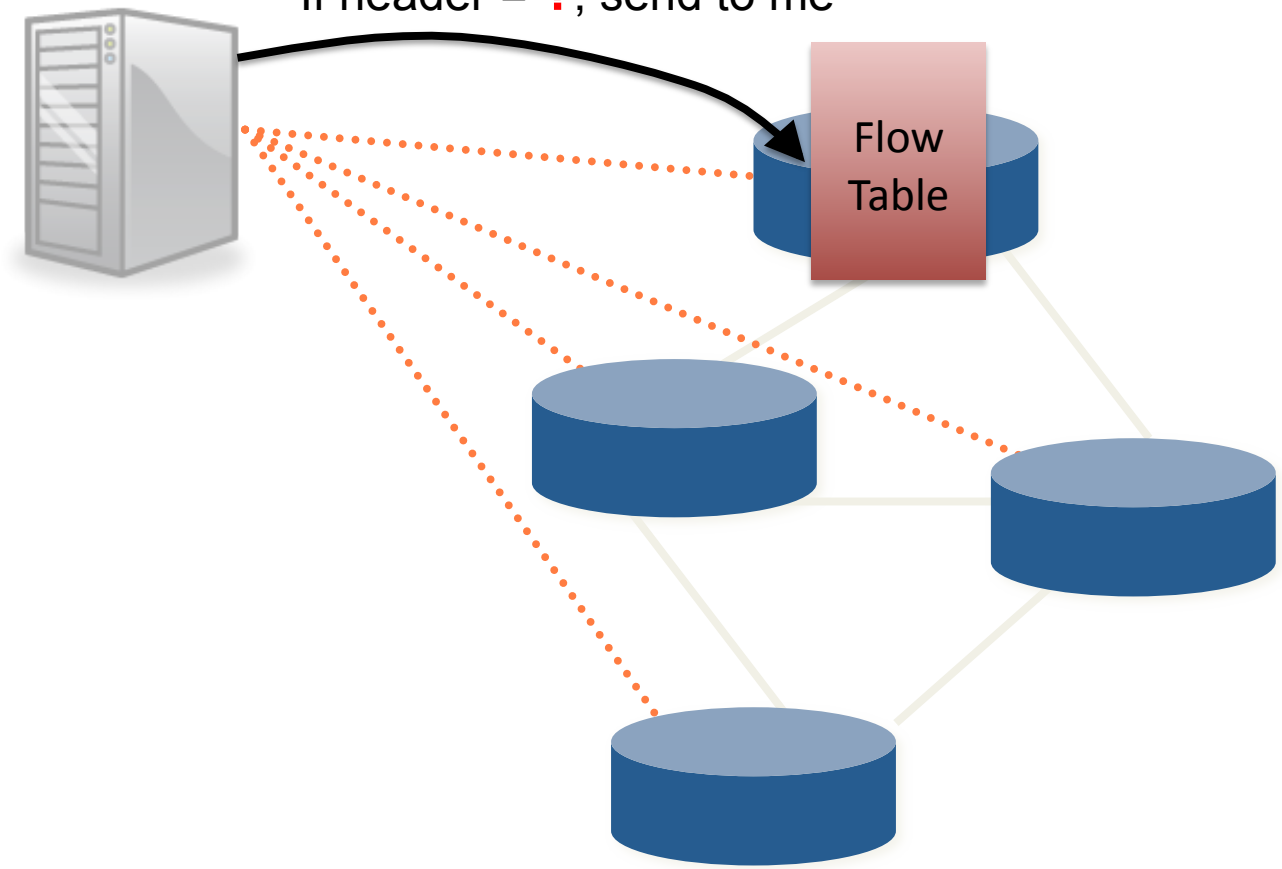
OpenFlow Basics

Step 1:
Separate Control from Datapath



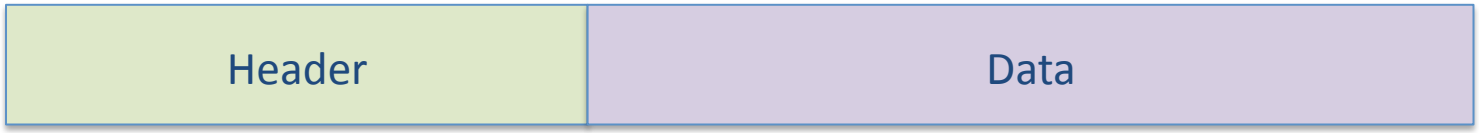
Step 2:
Cache flow decisions in datapath

"If header = ***x***, send to port 4"
"If header = ***y***, overwrite header with ***z***, send to ports 5,6"
"If header = **?**, send to me"



Plumbing Primitives

1. Match arbitrary bits in headers:

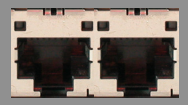


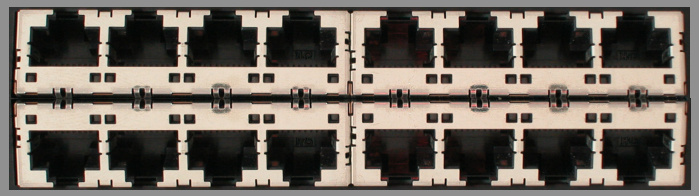
Match: 1000x01xx0101001x

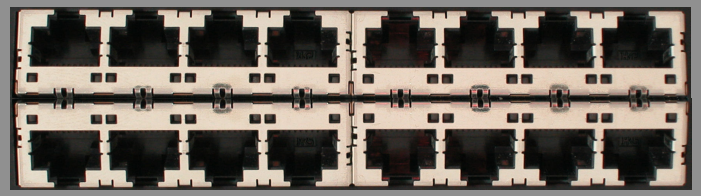
   - Match on any header, or new header
   - Allows any flow granularity

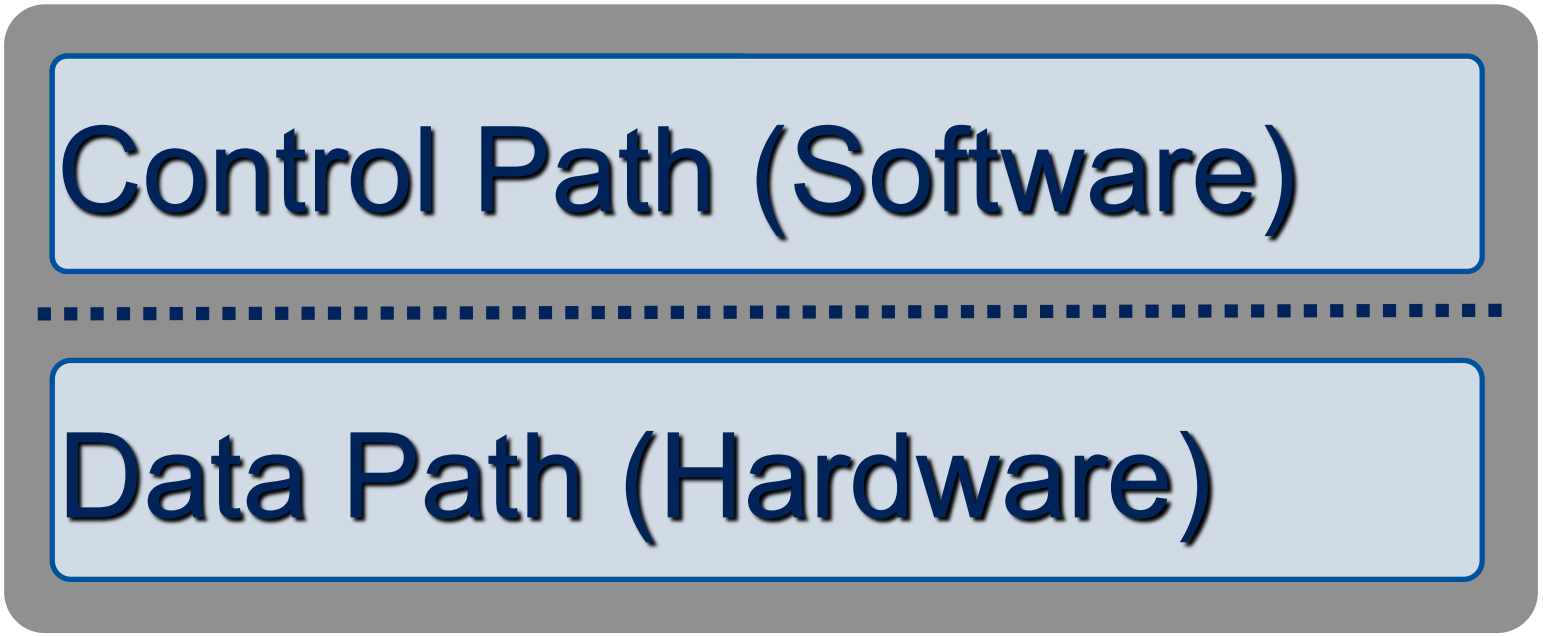
2. Actions:
   - Forward to port(s), drop, send to controller
   - Overwrite header with mask, push or pop
   - Forward at specific bit-rate

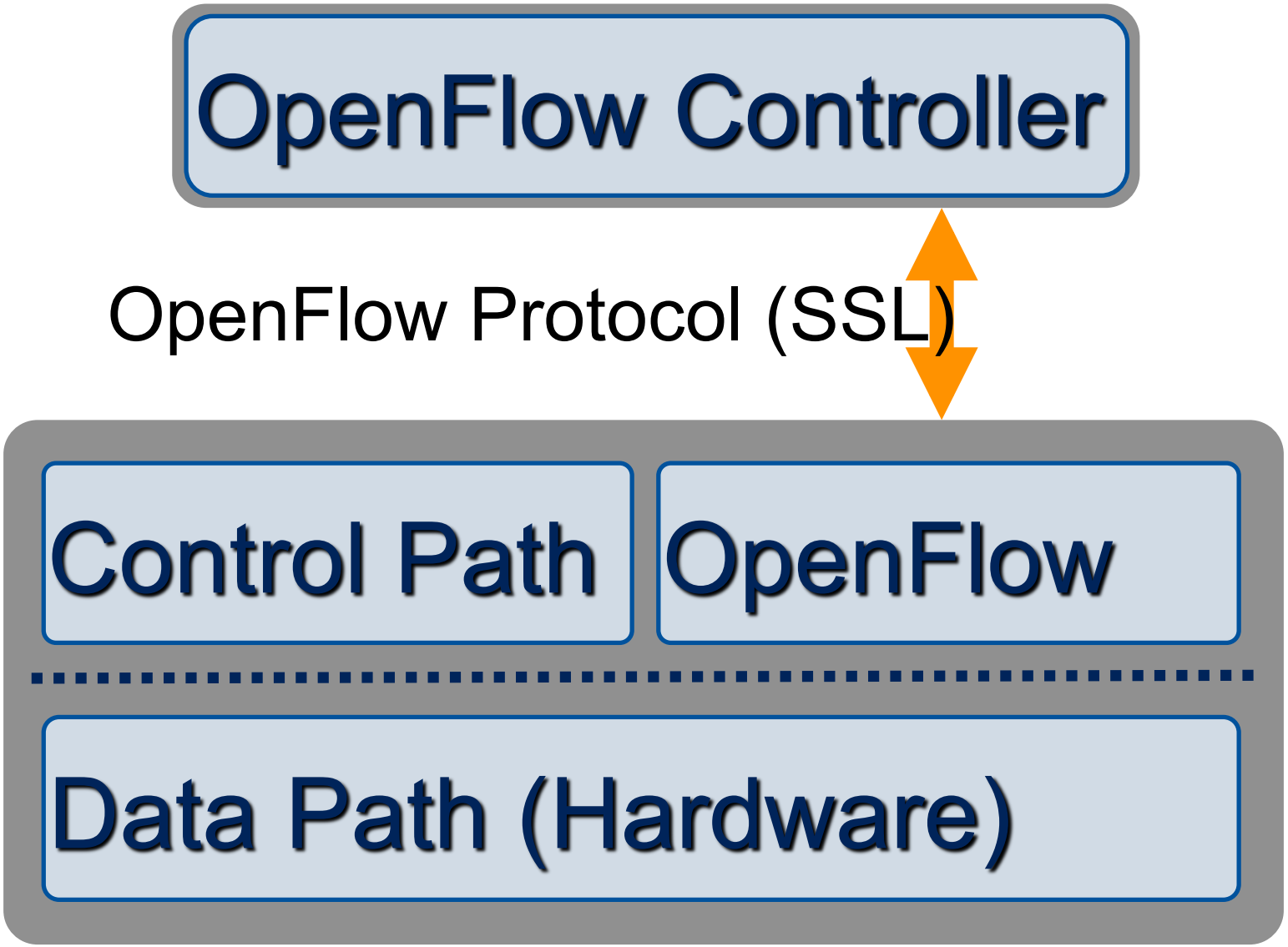
Ethernet Switch/Router








Control Path (Software)
Data Path (Hardware)


OpenFlow Controller
OpenFlow Protocol (SSL)
Control Path
OpenFlow
Data Path (Hardware)

OpenFlow Spec process

http://openflow.org

Current

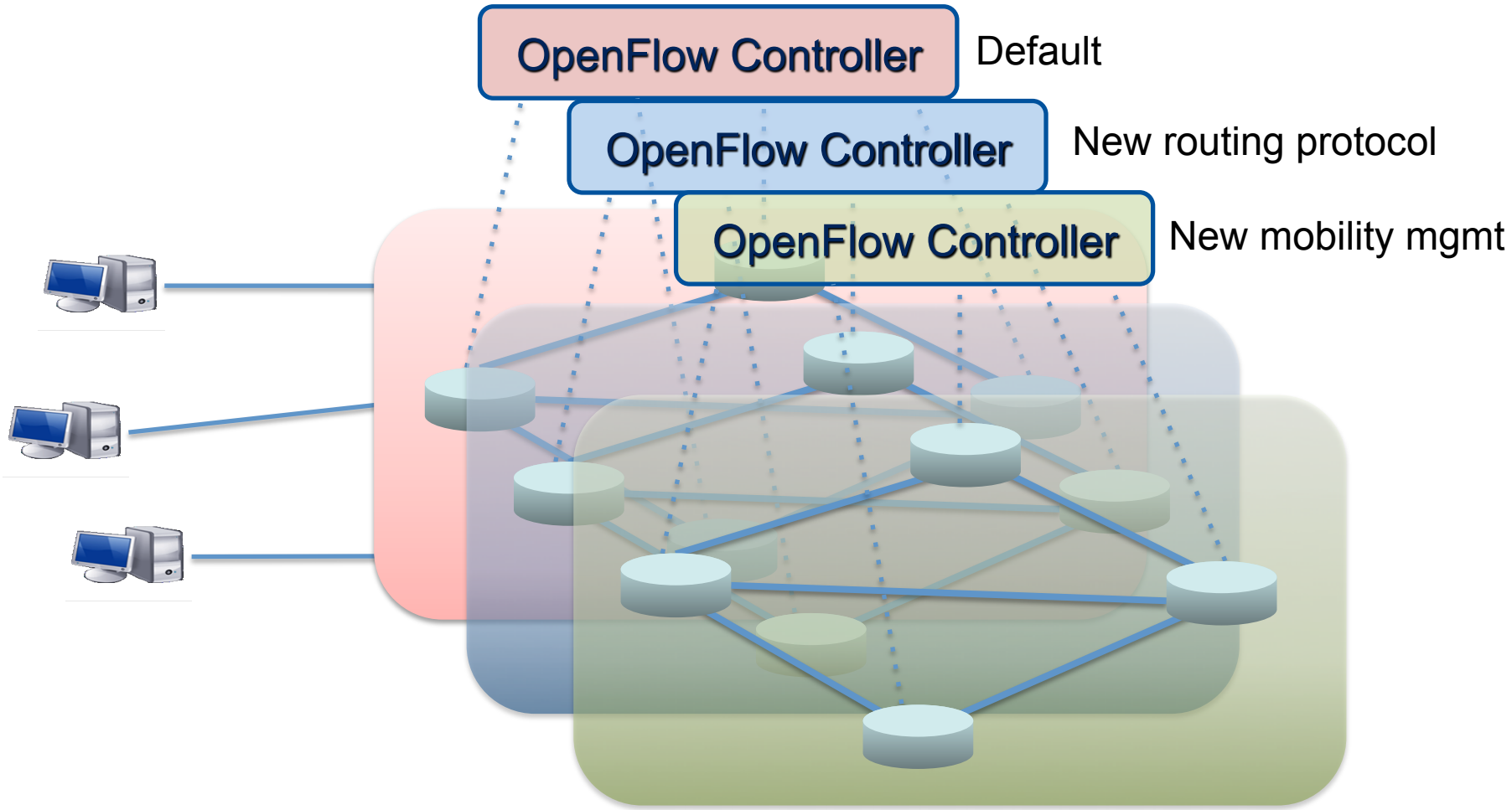
- V1.0: December 2009
- V1.1: Expected November 2010
- Open but ad-hoc process among 10-15 companies

Future

Planning a more “standard” process from 2011

Slicing an OpenFlow Network

Slicing


OpenFlow Controller
Default
OpenFlow Controller
New routing protocol
OpenFlow Controller
New mobility mgmt

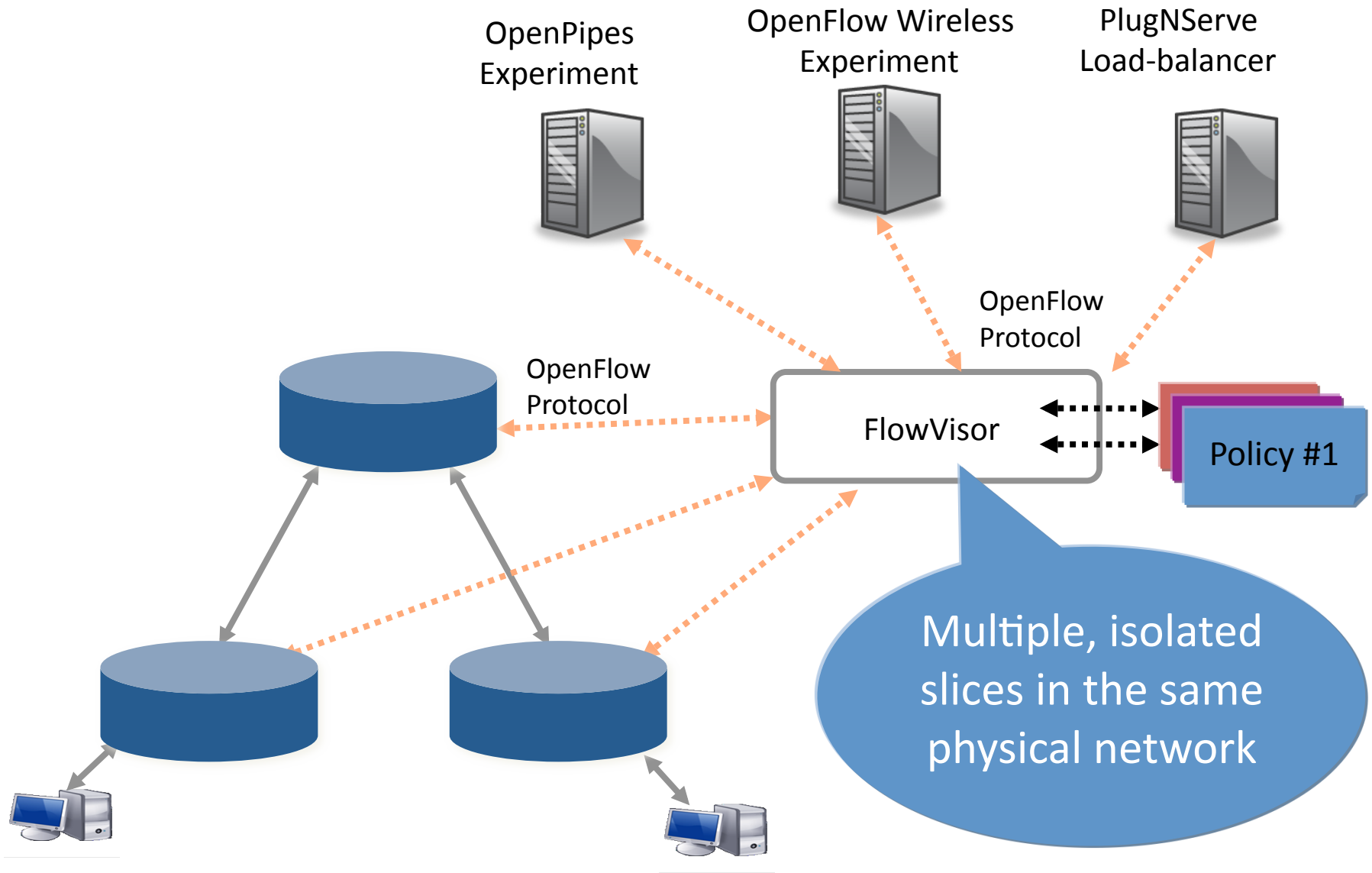
Ways to use slicing

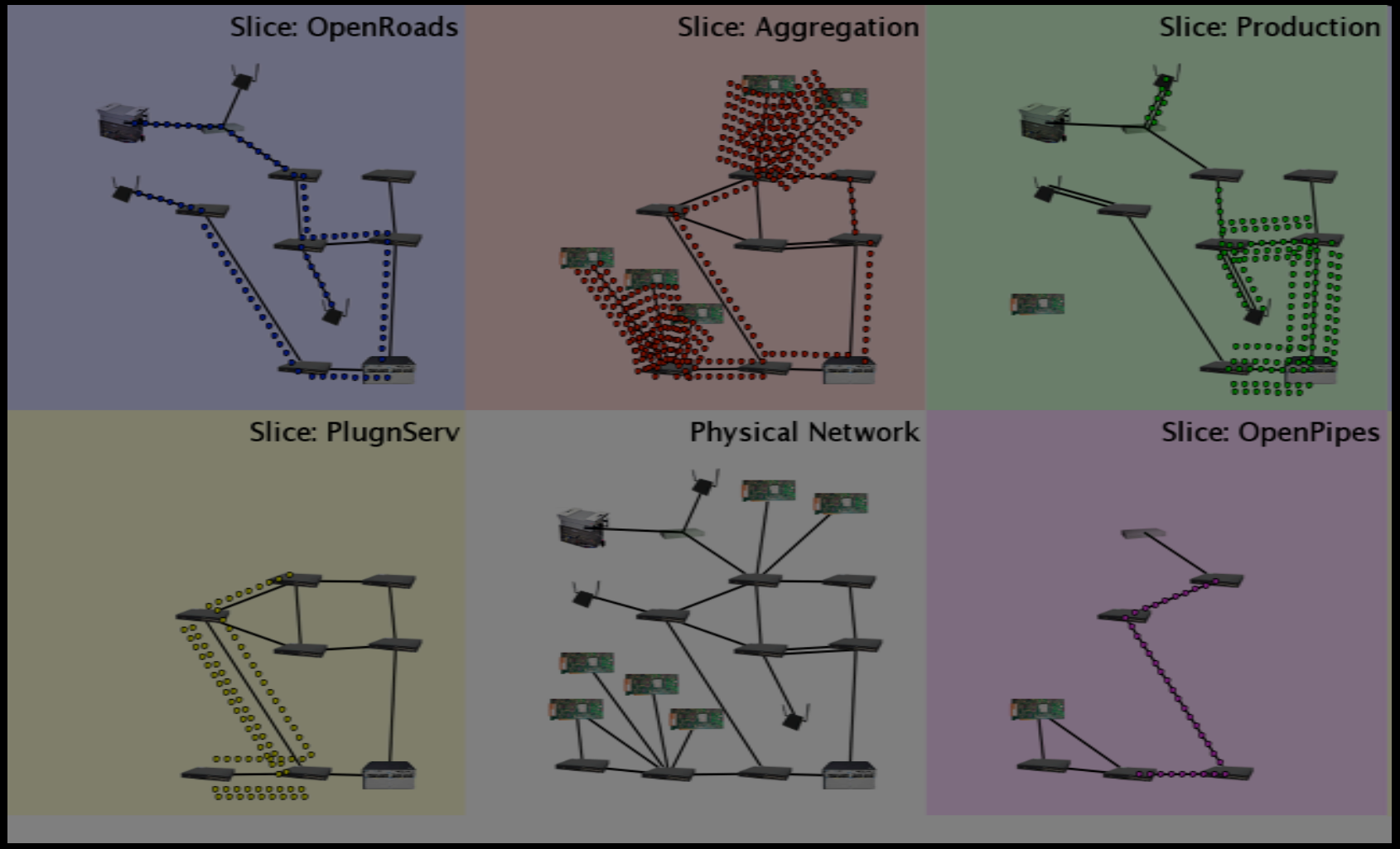
- Slice by feature
- Slice by user

- Home-grown protocols and services
- Download and try new feature
- Versioning

Some research examples

FlowVisor slices an OpenFlow network

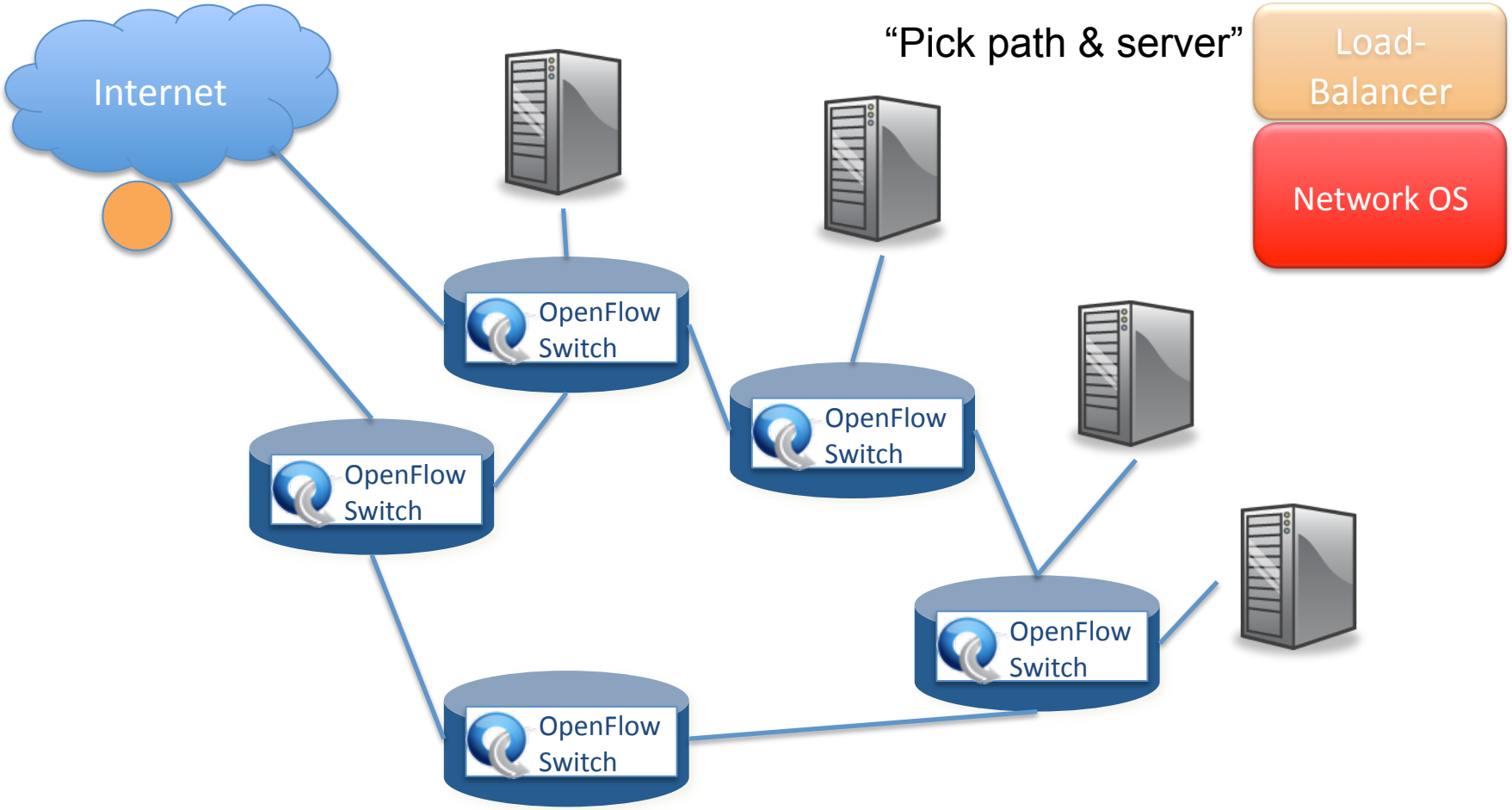



Slice: OpenRoads
Slice: Aggregation
Slice: Production
Slice: PlugnServ
Physical Network
Slice: OpenPipes

Application-specific Load-balancing

Goal: Minimize *http* response time over campus network

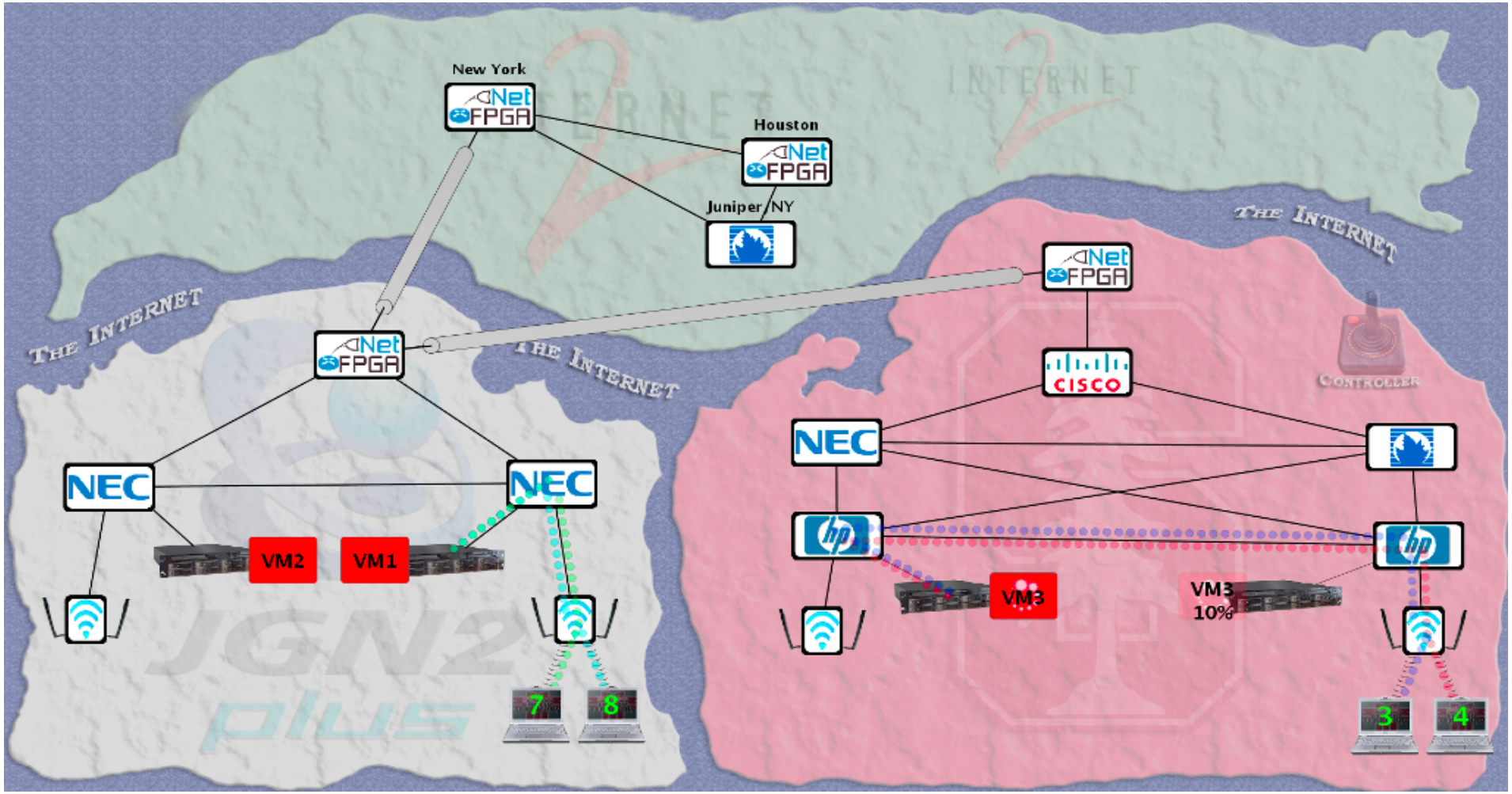
Approach: Route over path to jointly minimize <path latency, server latency>



Intercontinental VM Migration

Moved a VM from Stanford to Japan without changing its IP.

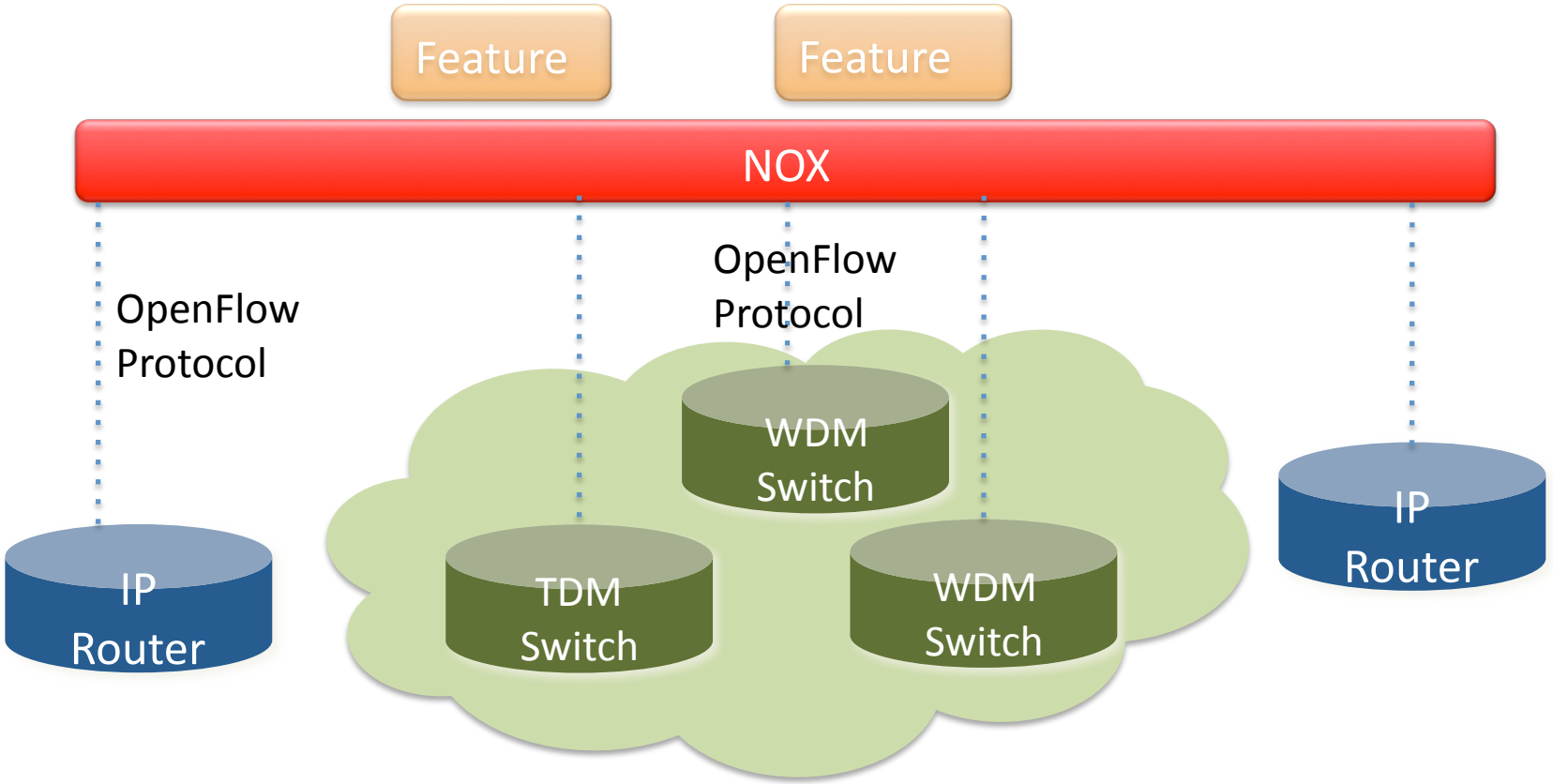
VM hosted a video game server with active network connections.



Converging Packet and Circuit Networks

Goal: Common control plane for "Layer 3" and "Layer 1" networks

Approach: Add OpenFlow to all switches; use common network OS



[Supercomputing 2009 Demo]

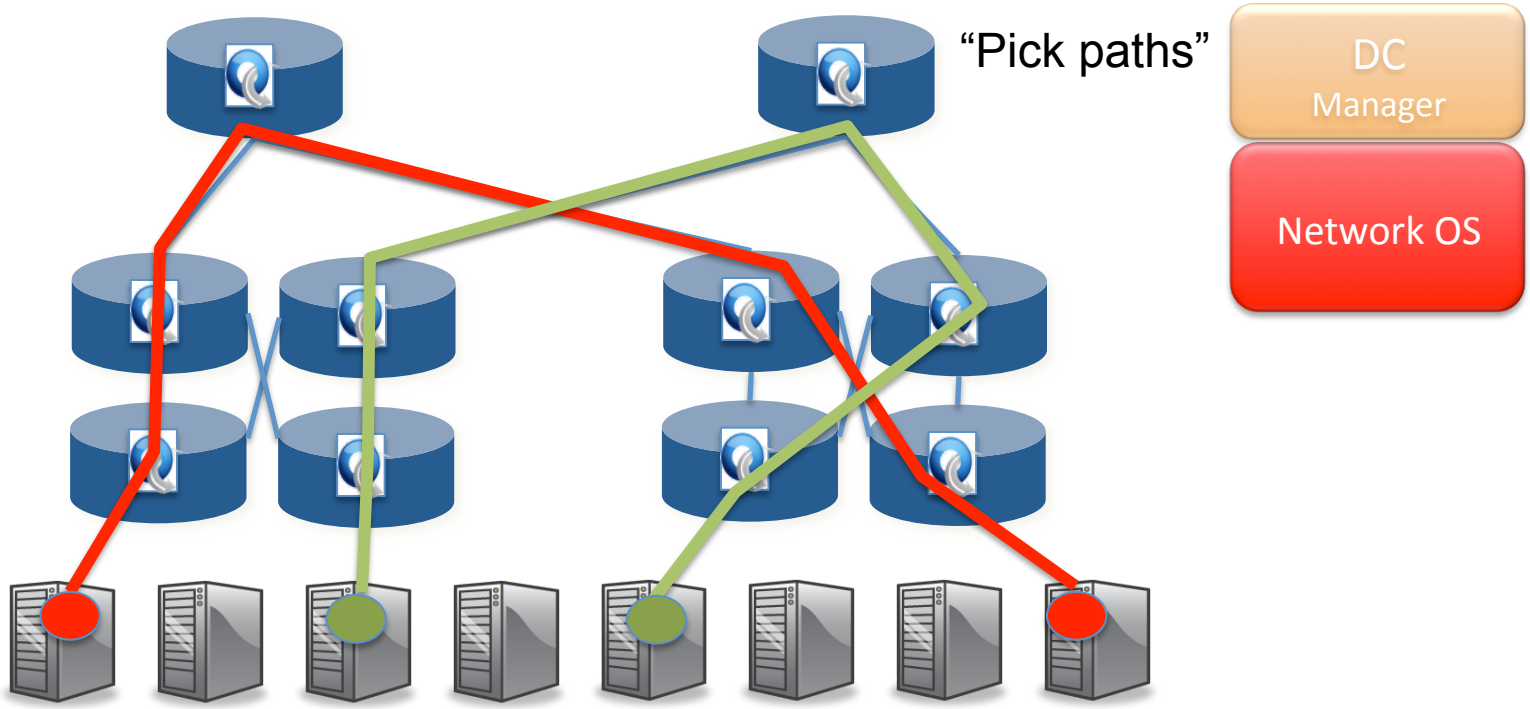
[OFC 2010]

ElasticTree

Goal: Reduce energy usage in data center networks

Approach:

1. Reroute traffic
2. Shut off links and switches to reduce power



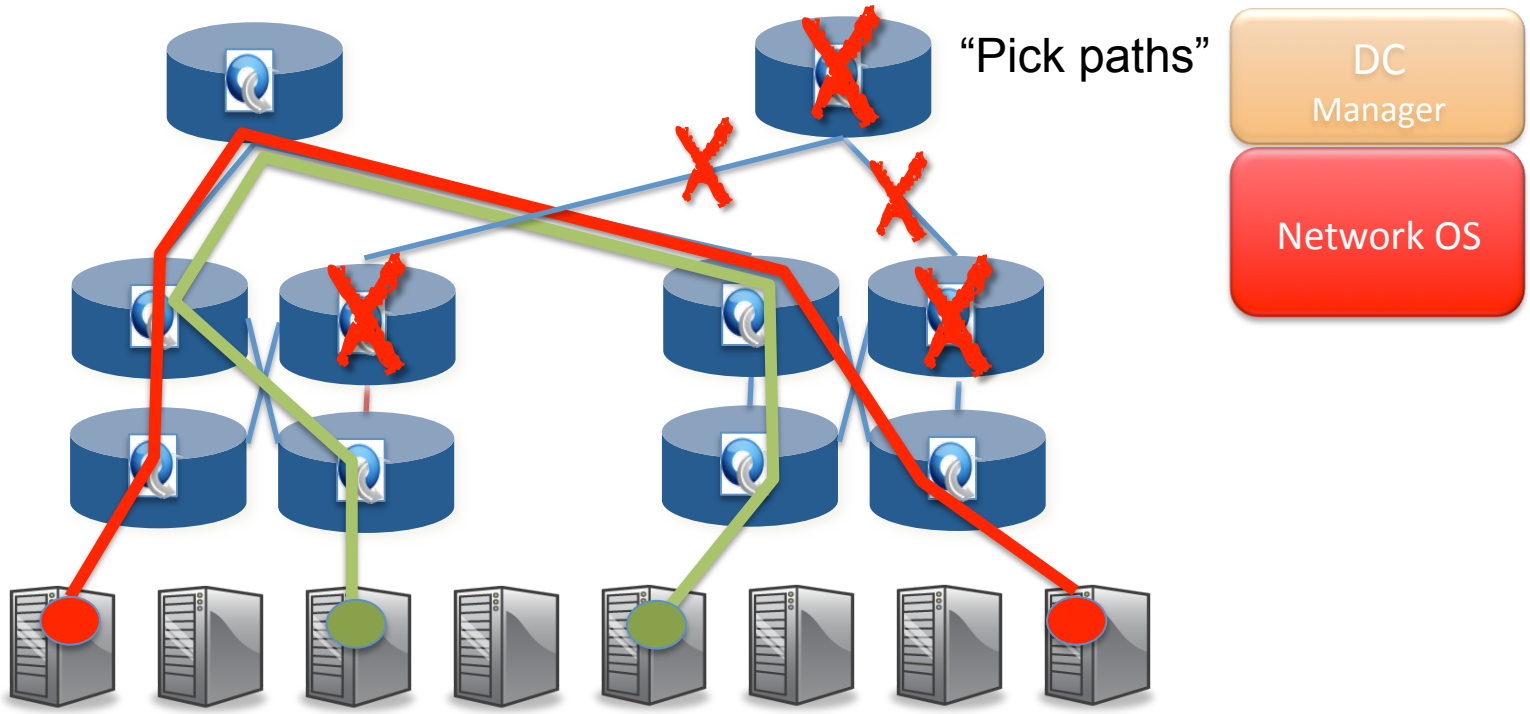
[NSDI 2010]

ElasticTree

Goal: Reduce energy usage in data center networks

Approach:

1. Reroute traffic
2. Shut off links and switches to reduce power



[NSDI 2010]

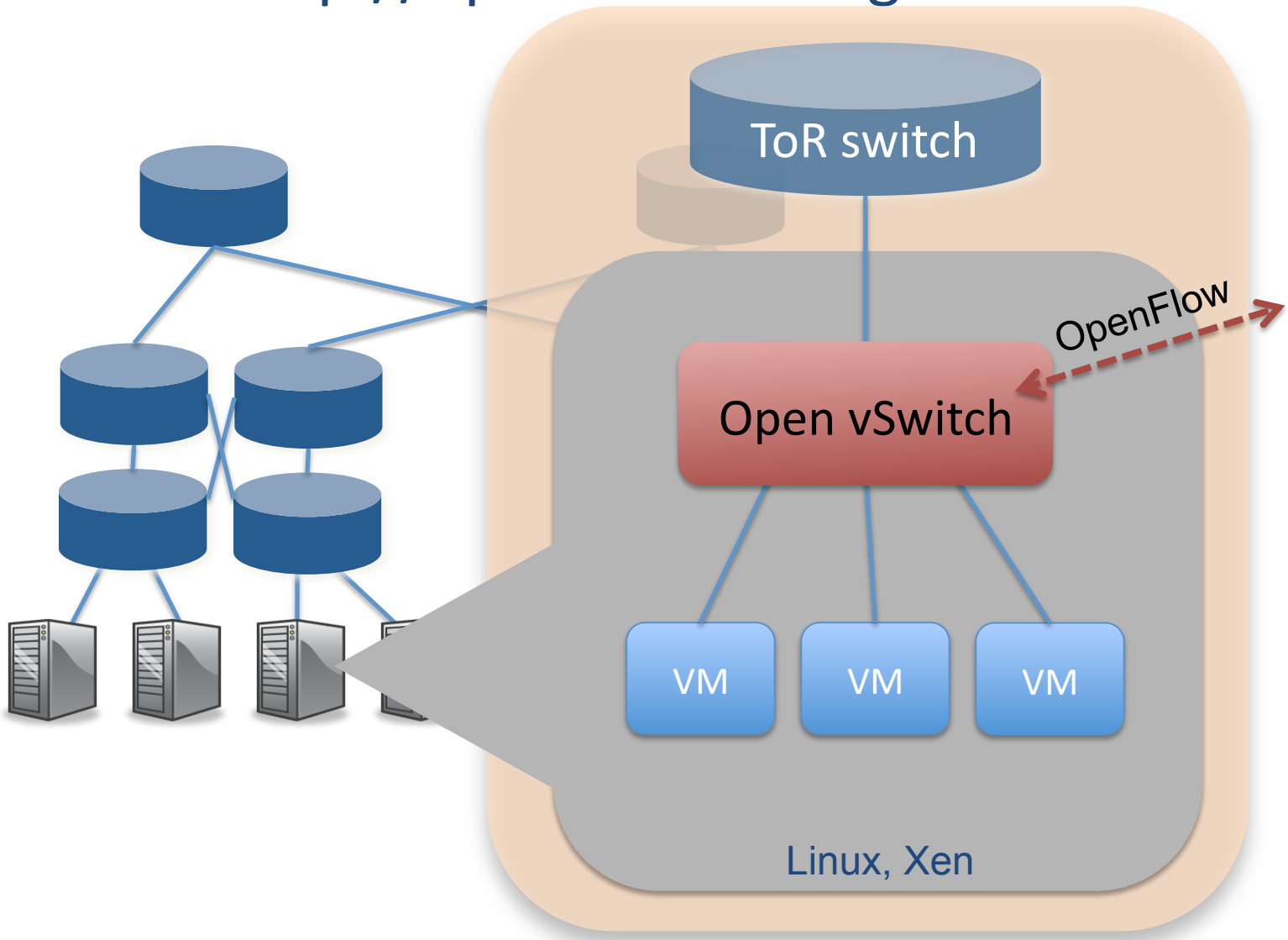
OpenFlow has been prototyped on….

- Ethernet switches
  - HP, Cisco, NEC, Quanta, + more underway
- IP routers
  - Cisco, Juniper, NEC
- Switching chips
  - Broadcom, Marvell
- Transport switches
  - Ciena, Fujitsu
- WiFi APs and WiMAX Basestations

Most (all?) hardware switches now based on Open vSwitch…

Open vSwitch

http://openvswitch.org



Network OS

Several commercial Network OS in development

- Commercial deployments 2010/2011

Research

- Research community mostly uses NOX
- Open-source available at: http://noxrepo.org

Part 2: Where does this lead?

What’s the problem?

Cellular industry

- Recently made transition to IP
- Billions of mobile users
- Need to securely extract payments and hold users accountable

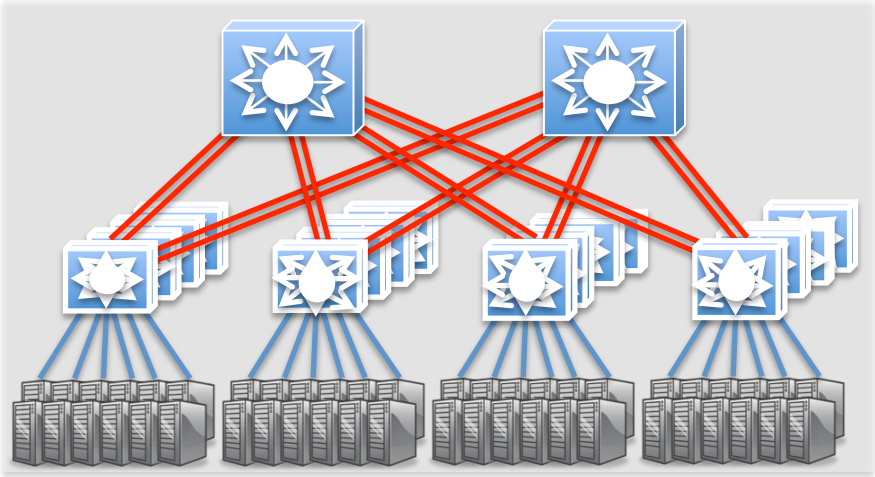
- IP sucks at both, yet hard to change

Telco Operators

- Global IP traffic growing 40-50% per year
- End-customer monthly bill remains unchanged
- Therefore, CAPEX and OPEX need to reduce 40-50% per Gb/s per year
- But in practice, reduces by ~20% per year

How can they differentiate their service offering?

Example: New Data Center



Cost

200,000 servers

Fanout of 20 ➔ 10,000 switches

$5k vendor switch = $50M

$1k commodity switch = $10M

Savings in 10 data centers = $400M

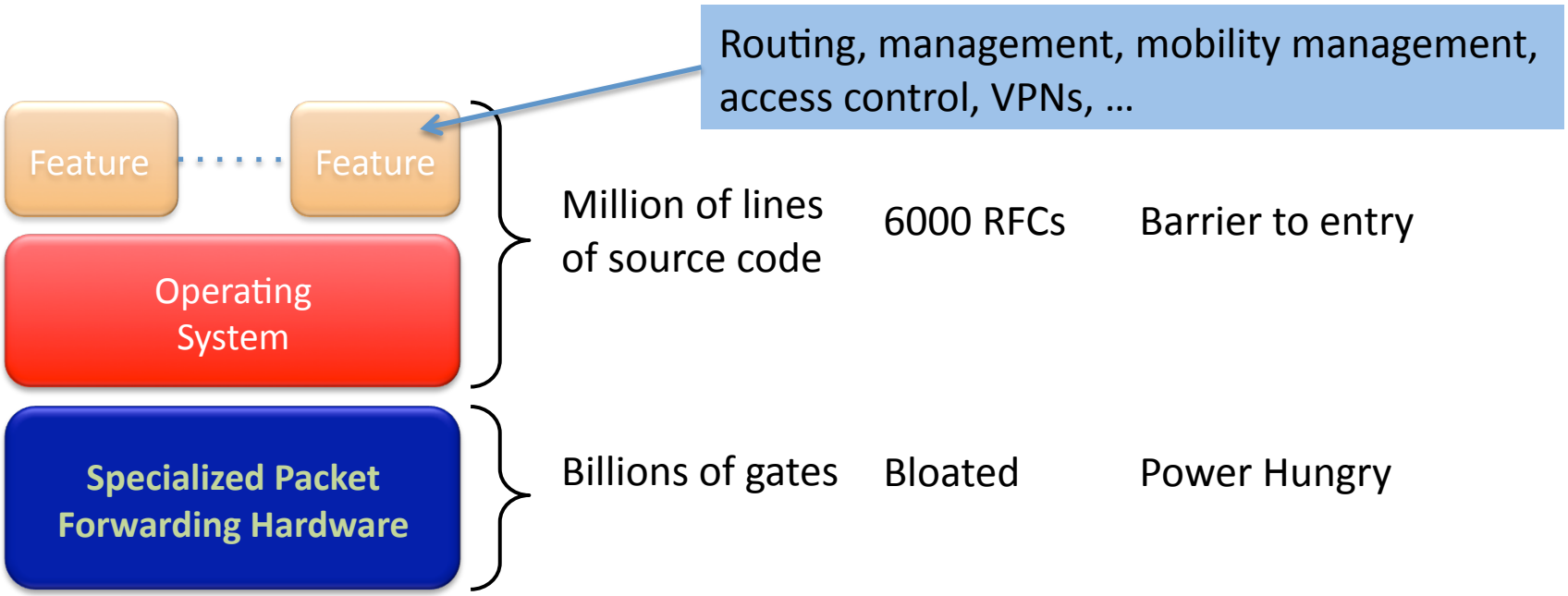
Control

More flexible control

Tailor network for services

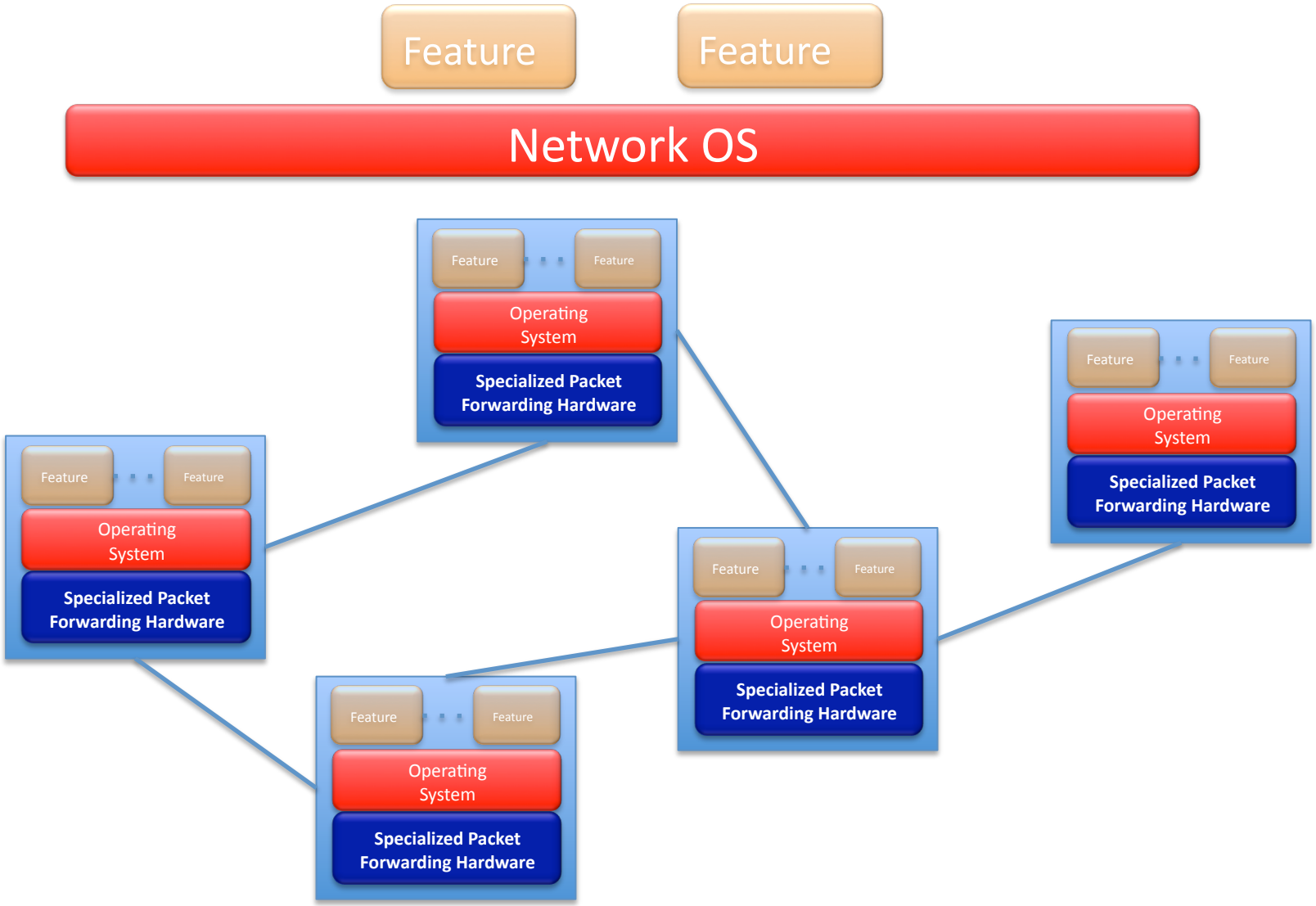
Quickly improve and innovate

A closed and proprietary industry

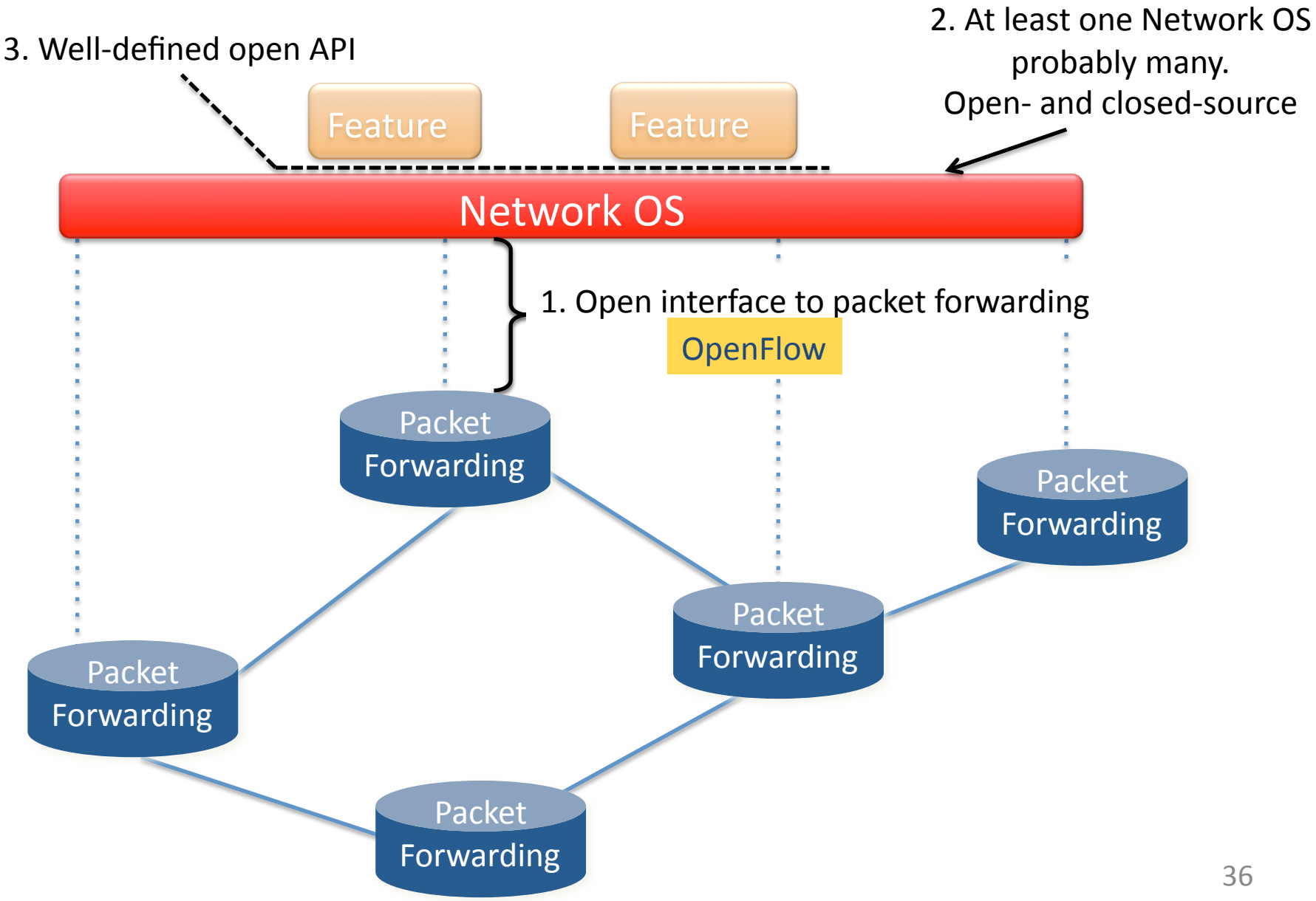


Looks like the mainframe industry in the 1980s

Restructured Network



The "Software-defined Network"



The SDN Approach

Separate control from the datapath

- i.e. separate policy from mechanism

Datapath: Define minimal network instruction set

- A set of "plumbling primitives"
- A vendor-agnostic interface: e.g. OpenFlow

Control: Define a network-wide OS

- An API that others can develop on

Where next?

Expect to see in

- Data centers
- Small WAN trials
- Some Campus production networks

Eventually could move into

- Larger WAN trials
- Enterprises
- Homes

Thank you