



OpenFlow: Enabling Technology Transfer to Networking Industry

Cisco Nerd Lunch, July 2009

Nikhil Handigol
nikhilh@cs.stanford.edu



Interesting Problems in Networking Research

- Mobility management
- Network security
- Energy management
- Flow management and measurement
- Packet processing
- ...

Technology Transfer

Academia to Industry

- Accelerates innovation in the field
- Desirable to both academia and industry
 - Academic research can have impact
 - Industry can benefit from academic research, improve products

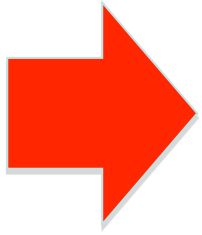
Problem with Networking Research

- Lack of technology transfer from academia to industry
 - No dearth of smart people
 - No lack of ideas
- Lack of ideas tested at scale
 - No way for academia to test ideas at scale
 - No reason for industry to invest in untested ideas

Possible Solutions

- Separate testbed of programmable open source switches and routers
 - Expensive
 - No real traffic
- Make Cisco boxes open source 😊
 - Not practical
- Can we strike a middle ground?

Our Approach

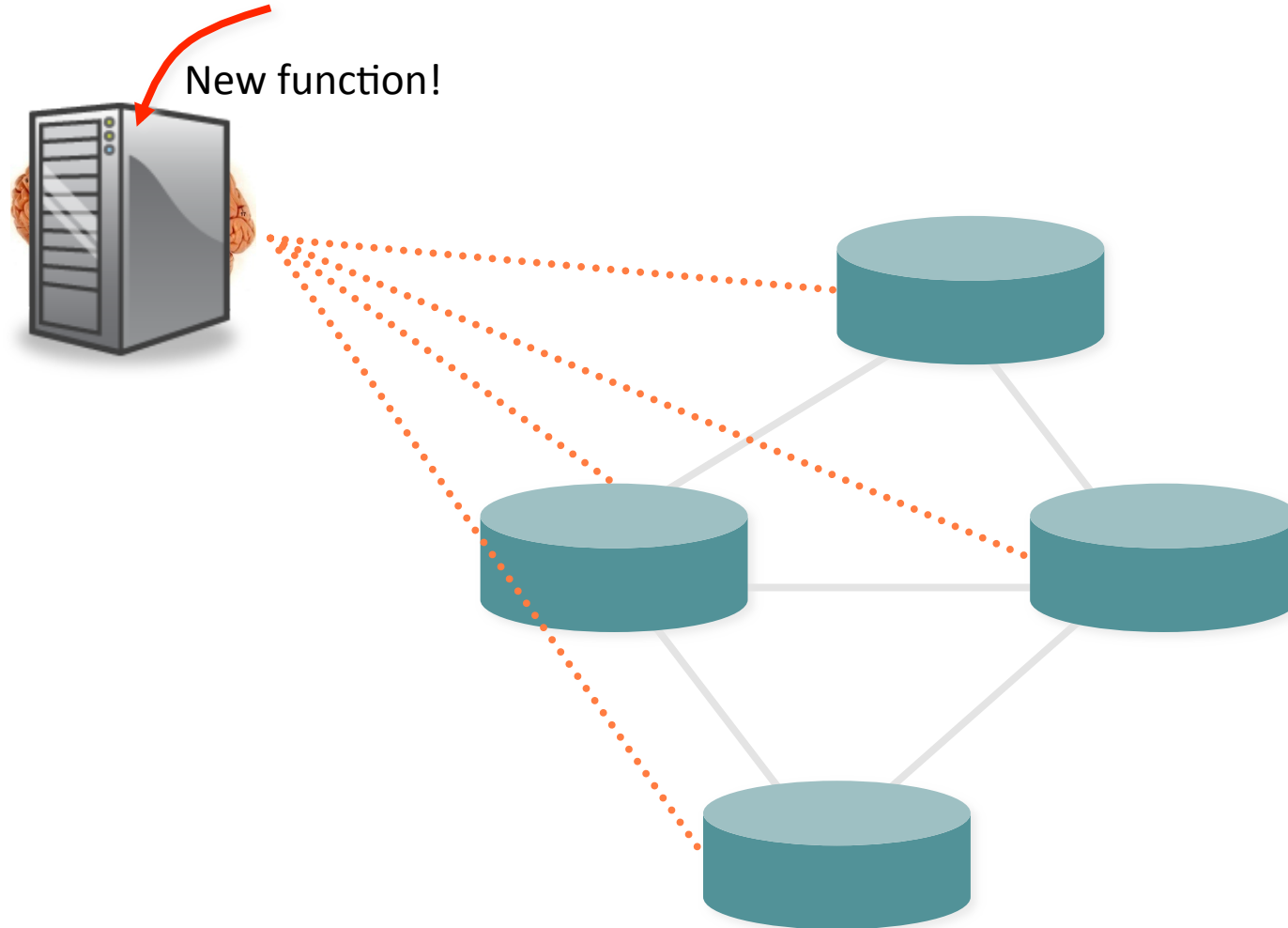


A clean separation between the substrate and an open programming environment

2. A simple hardware substrate that generalizes, subsumes and simplifies the current substrate

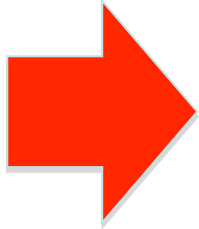
Step 1: Separate intelligence from datapath

Operators, users, 3rd party developers, researchers, ...



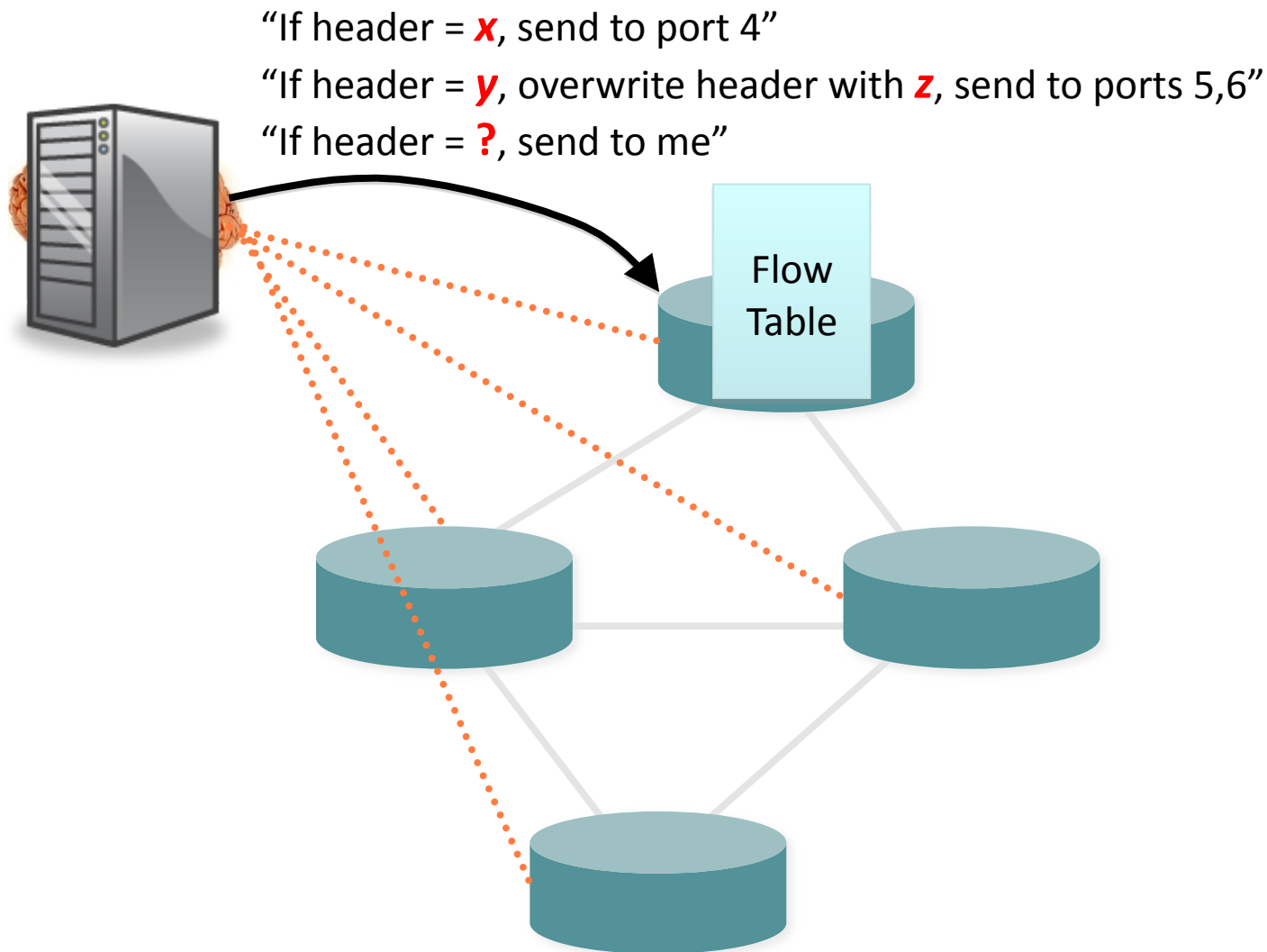
Our Approach

1. A clean separation between the substrate and an open programming environment



A simple hardware substrate that generalizes, subsumes and simplifies the current substrate

Step 2: Cache decisions in minimal flow-based datapath

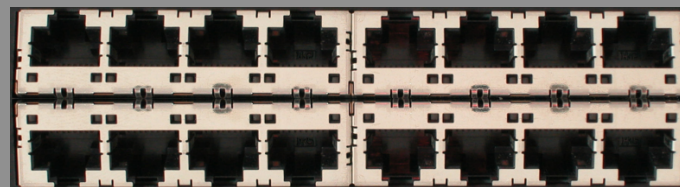
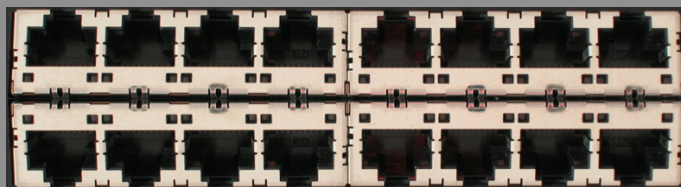


Our Solution: OpenFlow

- OpenFlow is an open external API to a flow-table
- Allows separation of control and data path via a simple, well defined interface
- Defined to be easy to add to existing hardware switches, routers, APs, ...

OpenFlow Basics

Ethernet Switch



Control Path (Software)

Data Path (Hardware)

OpenFlow Controller

OpenFlow Protocol (SSL)



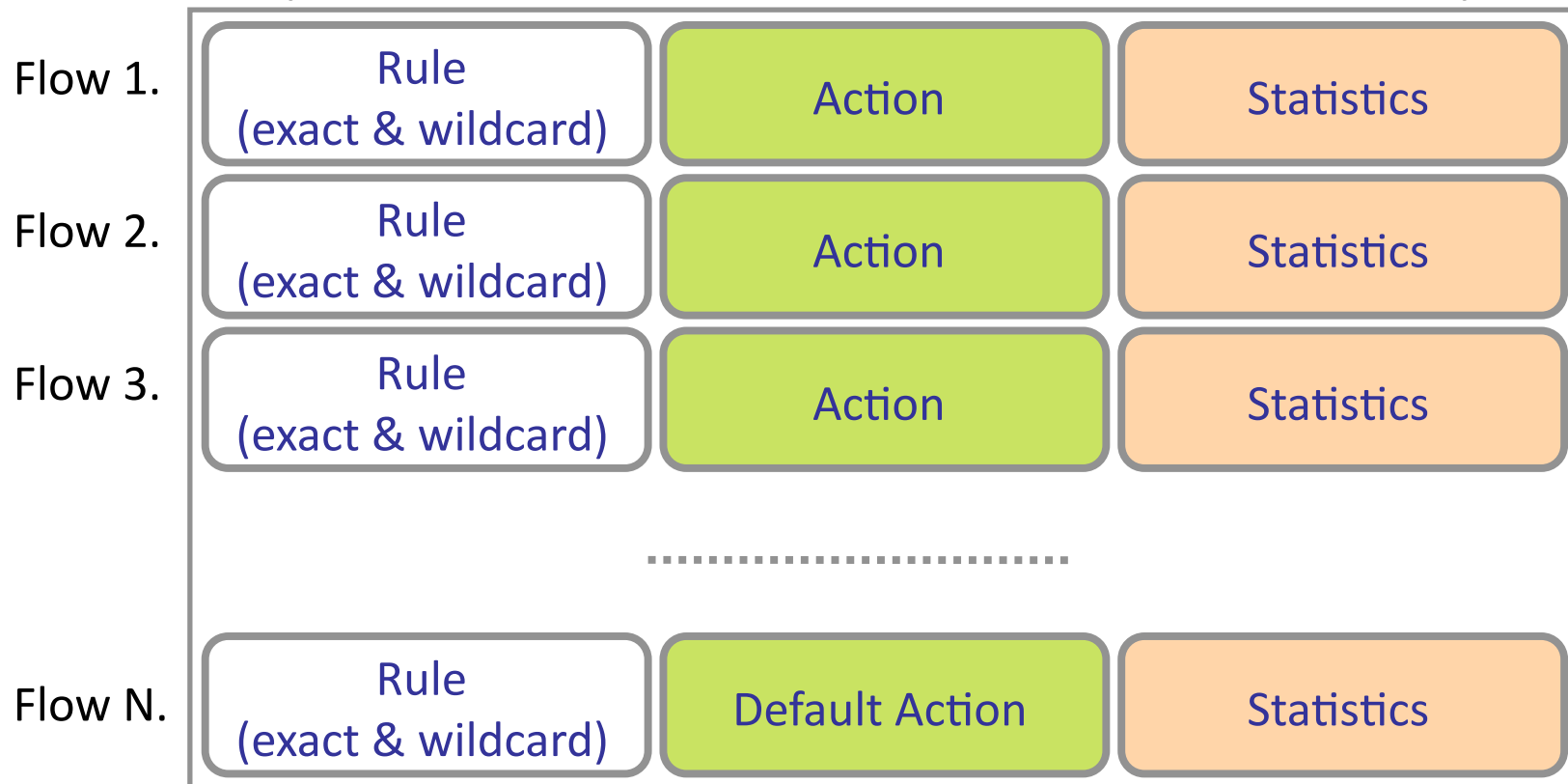
Control Path

OpenFlow

Data Path (Hardware)

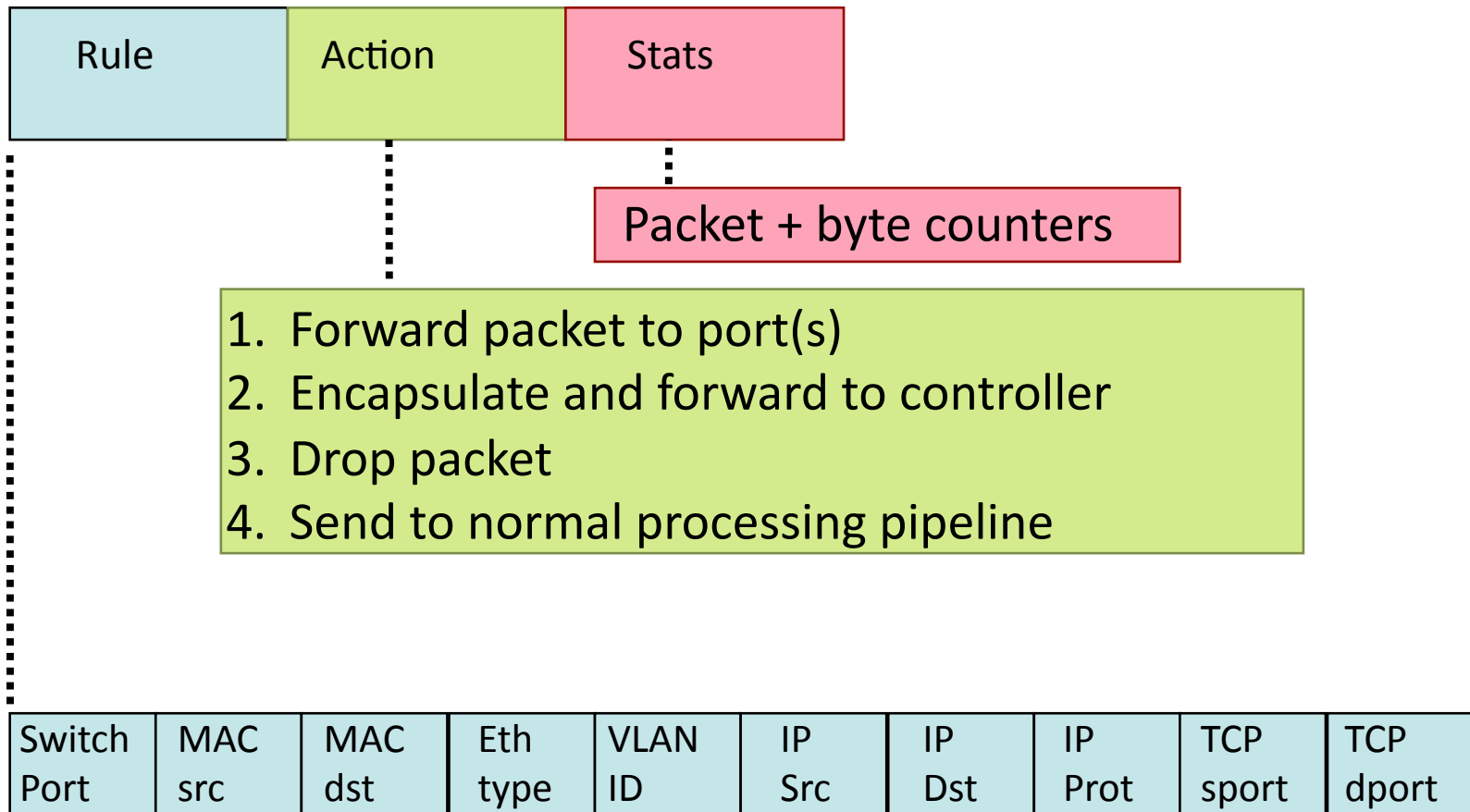
OpenFlow Basics

- Exploit the flow table in switches, routers, and chipsets



Flow Table Entry

OpenFlow Protocol Version 1.0



+ mask what fields to match

Examples

Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

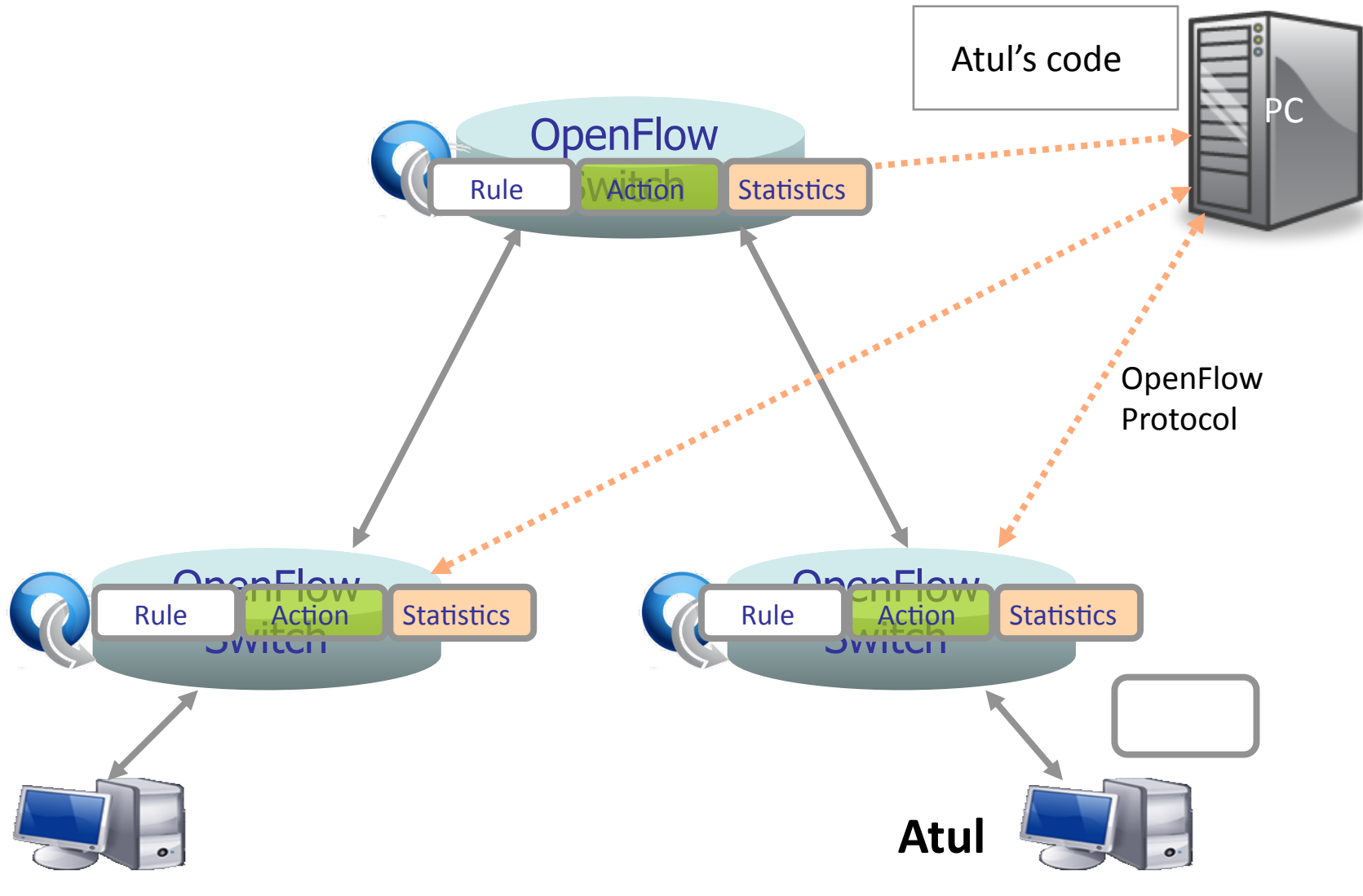
VLAN

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	vlan1	*	*	*	*	*	port6, port7, port9

OpenFlow Usage

Dedicated OpenFlow Network

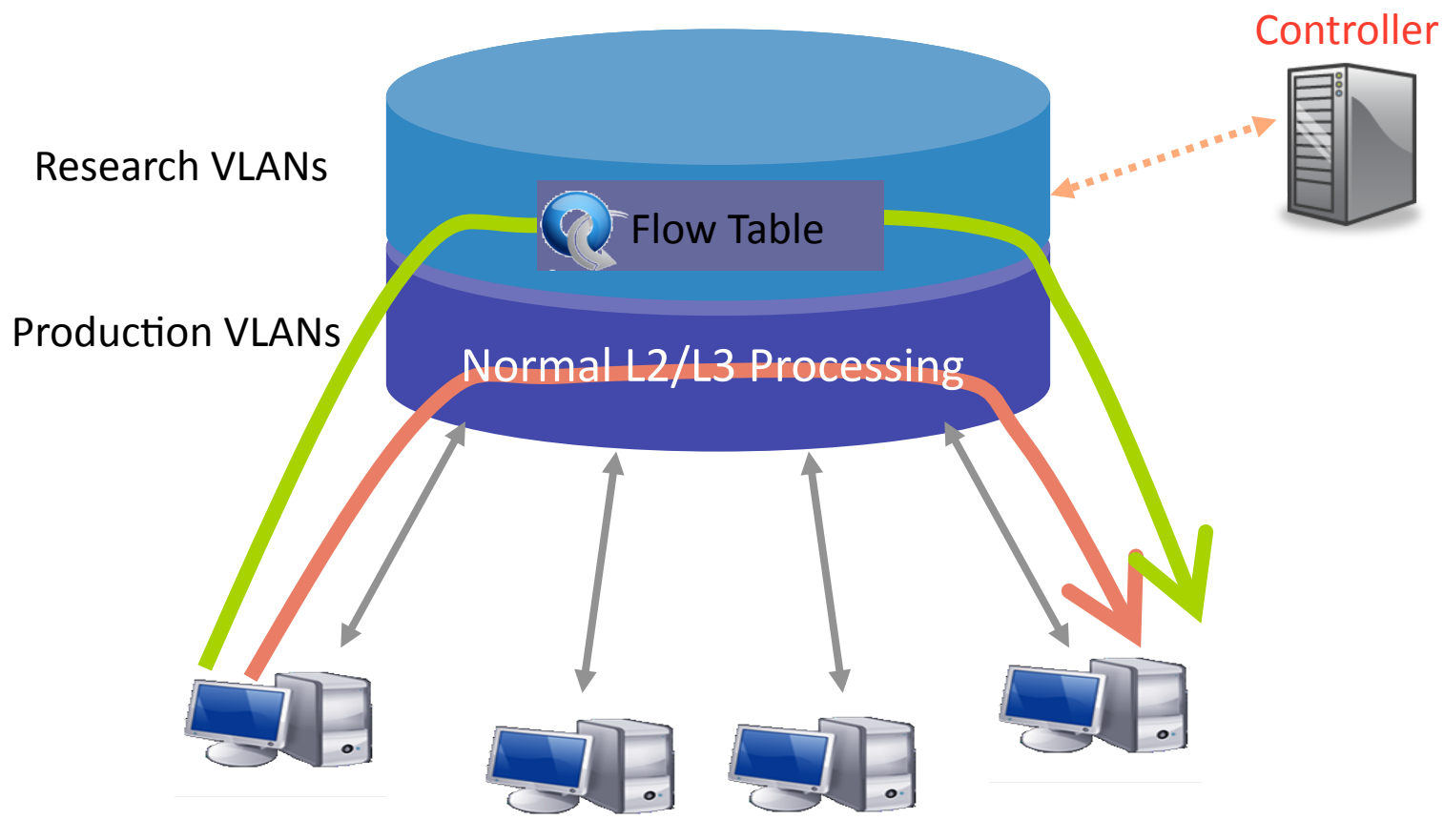
Controller



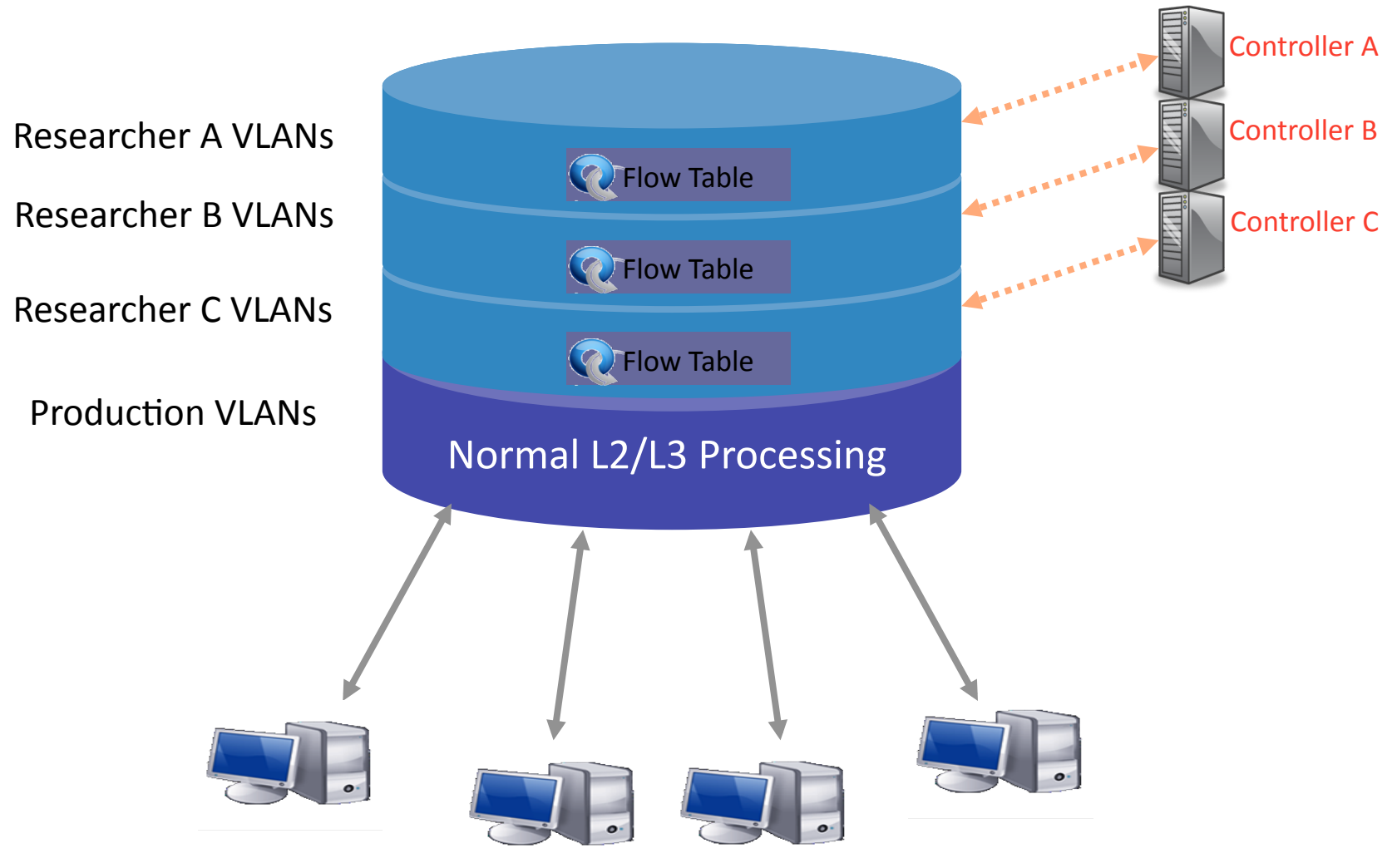
Usage examples

- Atul's code:
 - Static “VLANs”
 - His own new routing protocol: unicast, multicast, multipath, load-balancing
 - Network access control
 - Home network manager
 - Mobility manager
 - Energy manager
 - Packet processor (in controller)
 - IPvAtul
 - Network measurement and visualization
 - ...

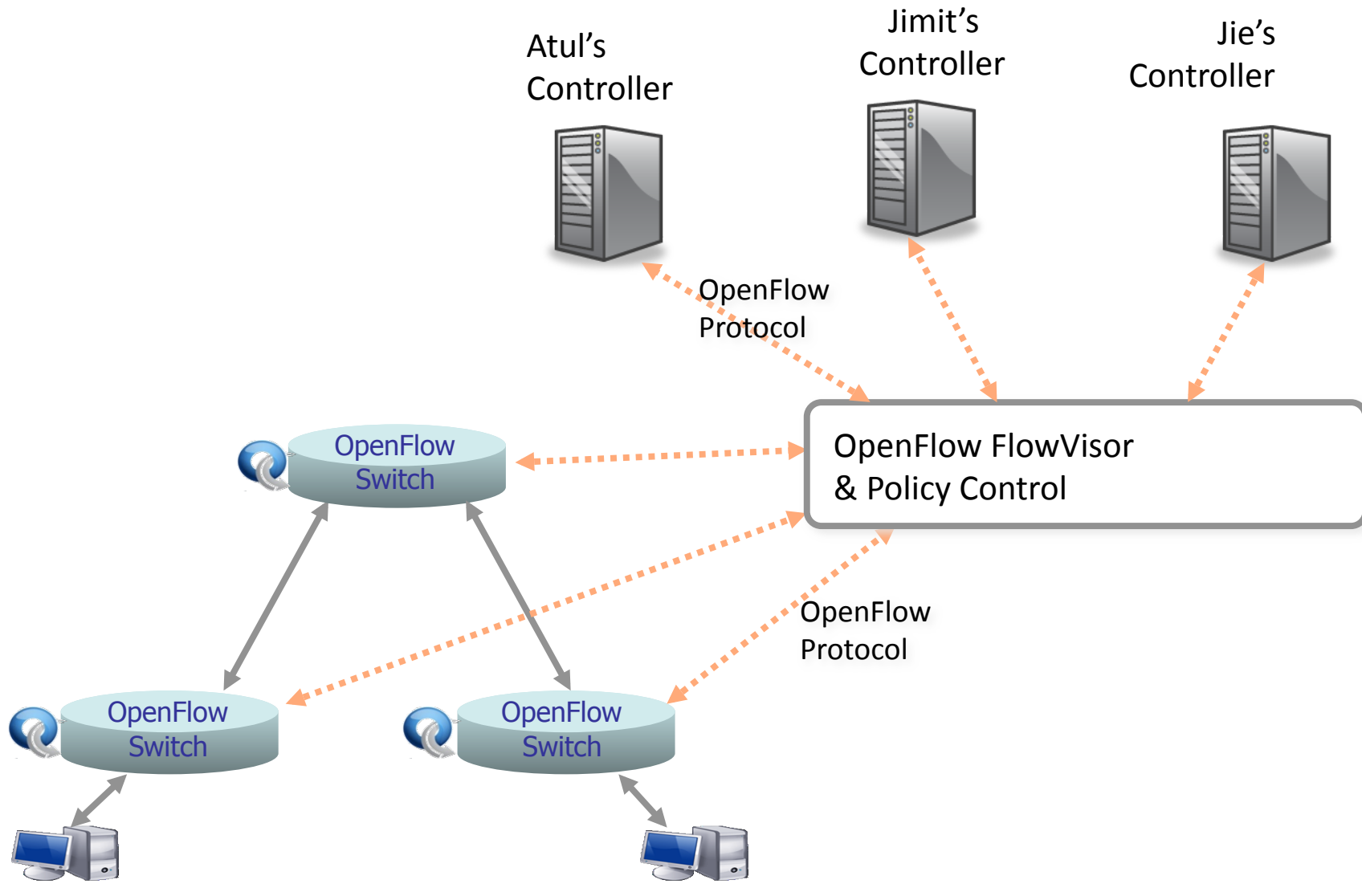
Separate VLANs for Production and Research Traffic



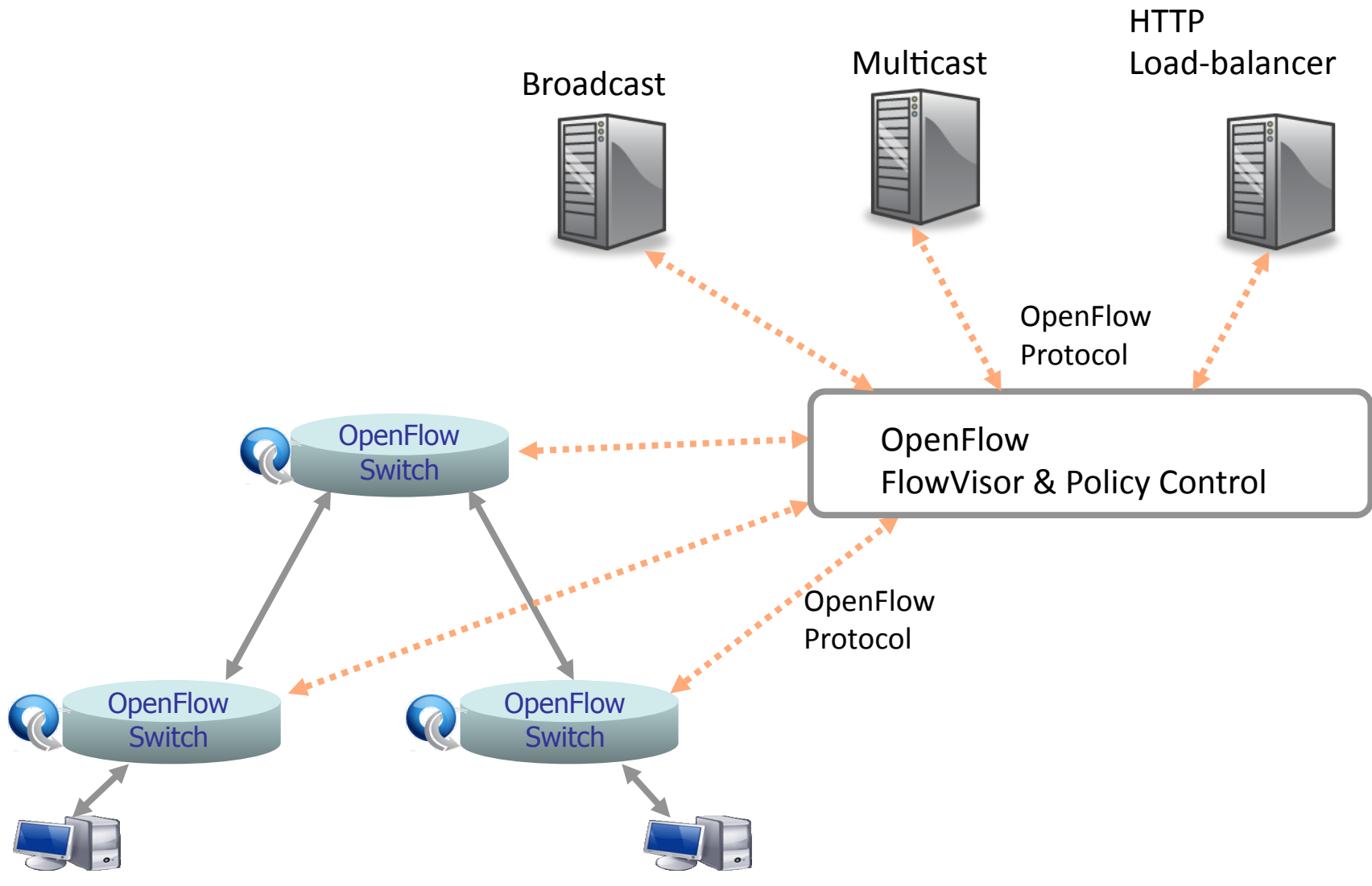
Virtualize OpenFlow Switch



Virtualizing OpenFlow



Virtualizing OpenFlow



OpenFlow Deployment

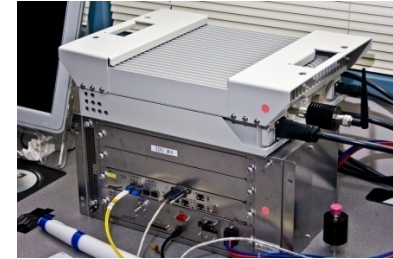
OpenFlow Hardware



Juniper MX-series



NEC IP8800



WiMax (NEC)



HP Procurve 5400



Quanta LB4G



PC Engines

coming soon...



Cisco Catalyst 3K

OpenFlow Deployments

- Stanford Deployments
 - **Wired**: CS Gates building, EE CIS building, EE Packard building
 - **WiFi**: 100 OpenFlow APs across SoE
 - **WiMAX**: OpenFlow service in SoE
- Other deployments
 - Internet2 (NetFPGA switches)
 - JGN2plus, Japan (NEC switches)
 - 10-15 research groups have switches

Summer Plan

Summer Plan

Step-1: Software Implementation

- OpenFlow as an IOS subsystem in the C3750E switch
- Thorough testing and debugging
- Fully functional OpenFlow switch, though not efficient

Summer Plan

Step-2: Hardware Implementation

- Explore feasibility
- Implement as many features in hardware as possible
- Eg. Exploit ACLs
 - Define packet matching rules
 - Define basic actions such as packet dropping and packet forwarding

Thank you!