Software Defined Networking and OpenFlow

Saurav Das

With contributions Nick McKeown, Guru Parulkar, Scott Shenker, Brandon Heller, Rob Sherwood, Guido Appenzeller, Martin Casado and many, many others…

OFC Short Course SC401
9th March, 2014

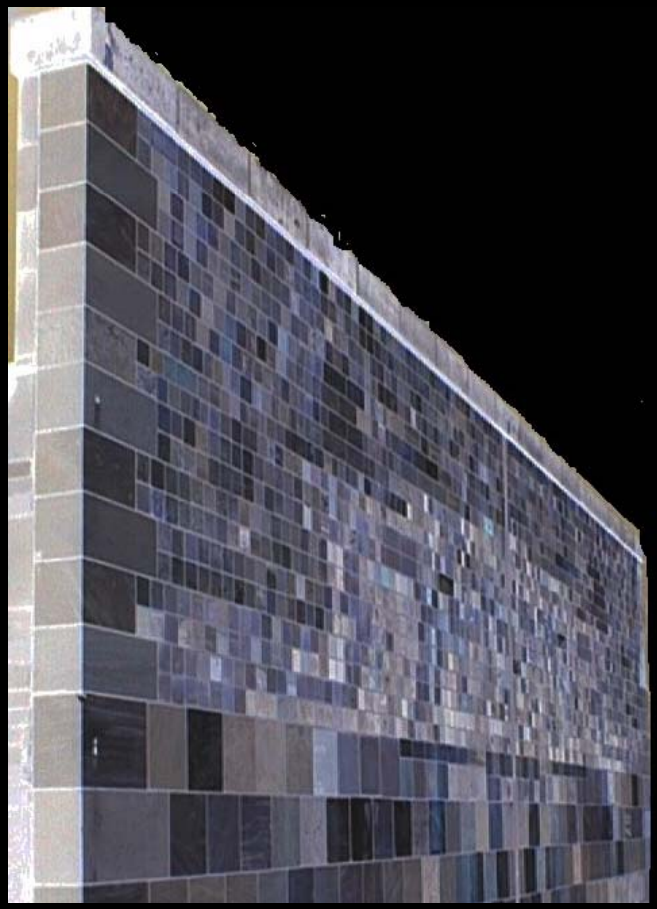
Outline

- Module 1: SDN/OpenFlow Introduction
- Module 2: OpenFlow/SDN Deep Dive
- Module 3: SDN Architecture
- Module 4: Operational Choices & Performance
- Module 5: Benefits & Use Cases
  - ❖ Demo: Network Virtualization
- Module 6: Demystifying SDN
- Q & A

Module 1

SDN/OpenFlow Introduction

Early Motivation



RESEARCH

Many great ideas

PRODUCTION

Next to none transferred to production networks

Networking Industry

Routing, management, mobility management, access control, VPNs, …

App App App

Operating System

Specialized Packet Forwarding Hardware

Million of lines of source code

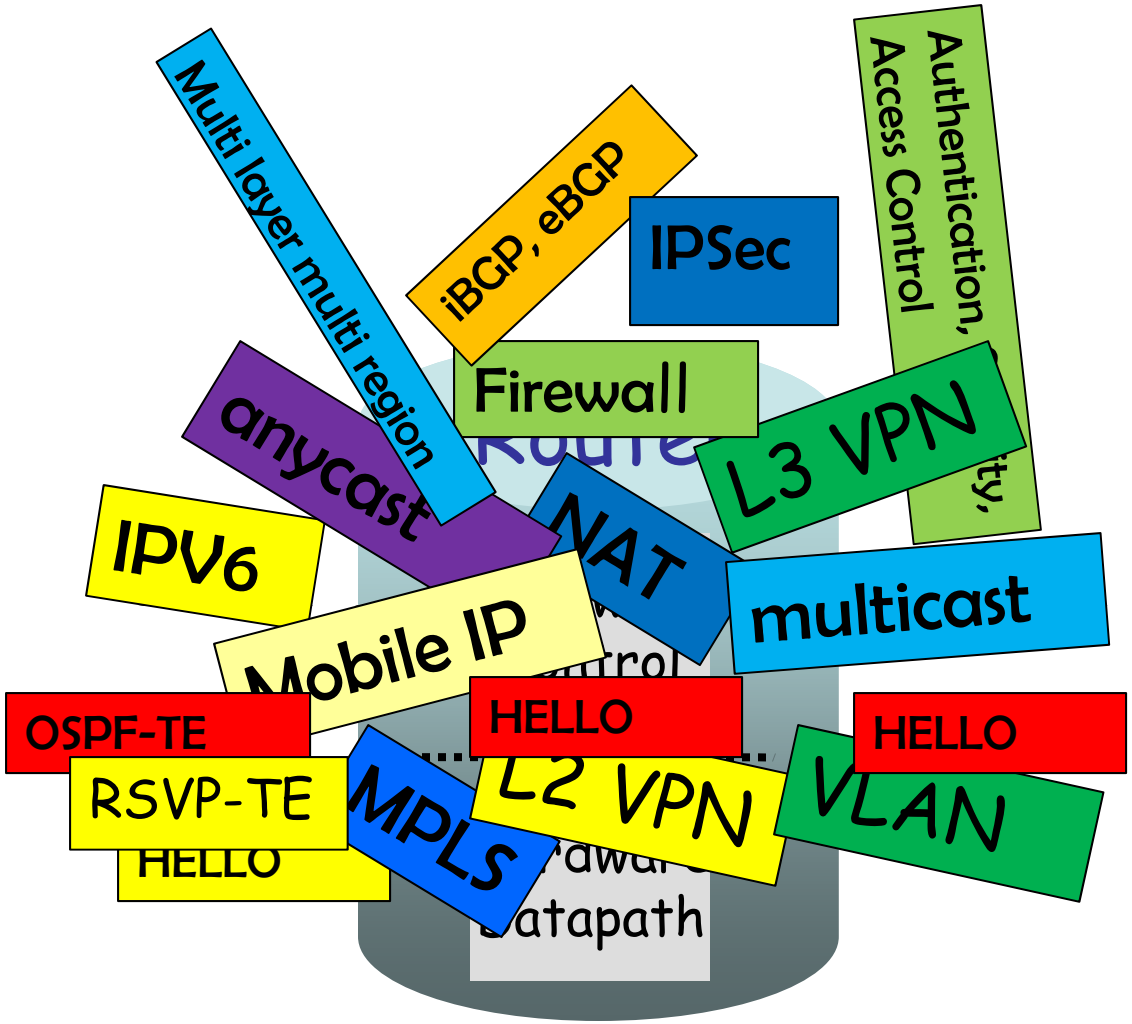
5400 RFCs

Barrier to entry

500M gates
10Gbytes RAM

Bloated

Power Hungry

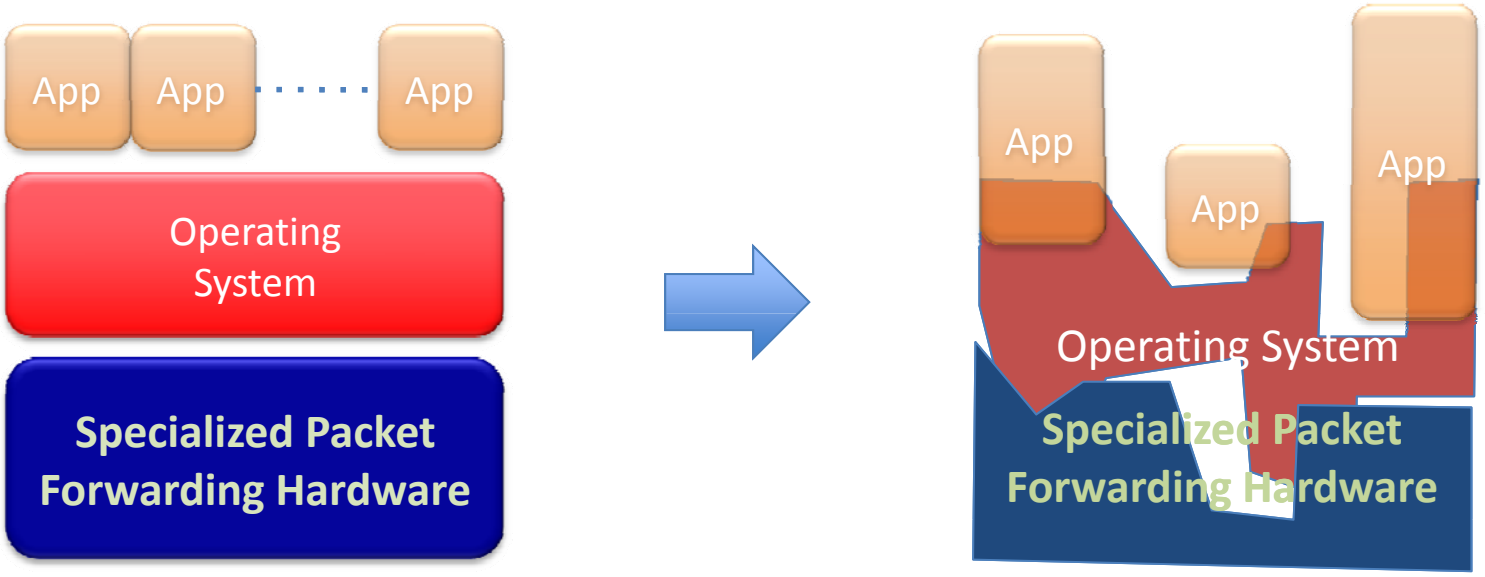


Many complex functions baked into the infrastructure

OSPF, BGP, multicast, differentiated services,
Traffic Engineering, NAT, firewalls, MPLS, redundant layers, …

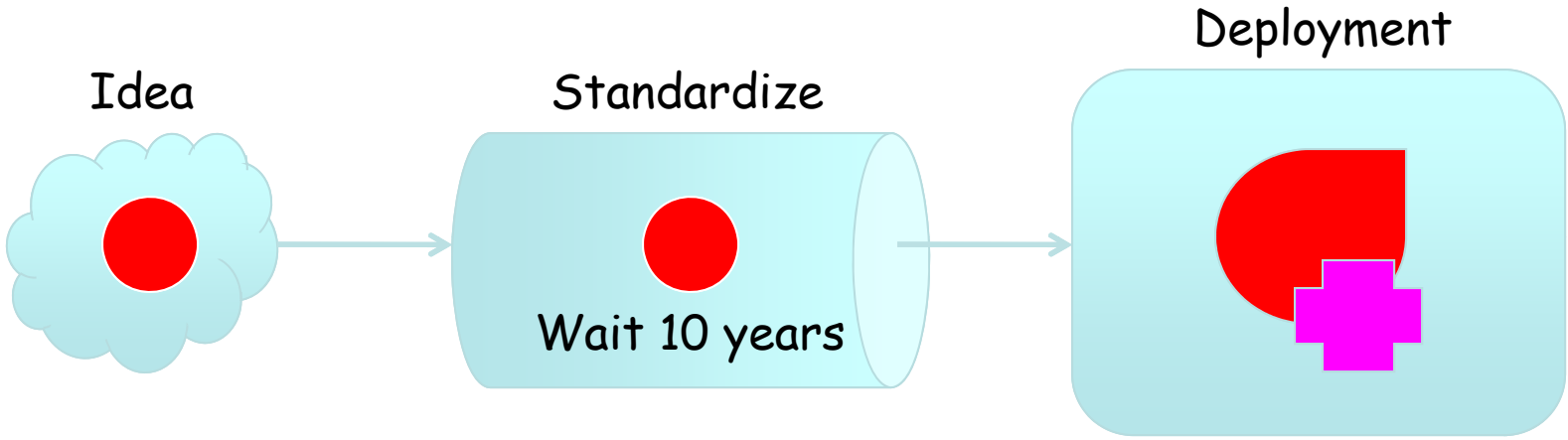
An industry with a “mainframe-mentality”

Reality



- Lack of competition means glacial innovation
- Closed architecture means blurry, closed interfaces

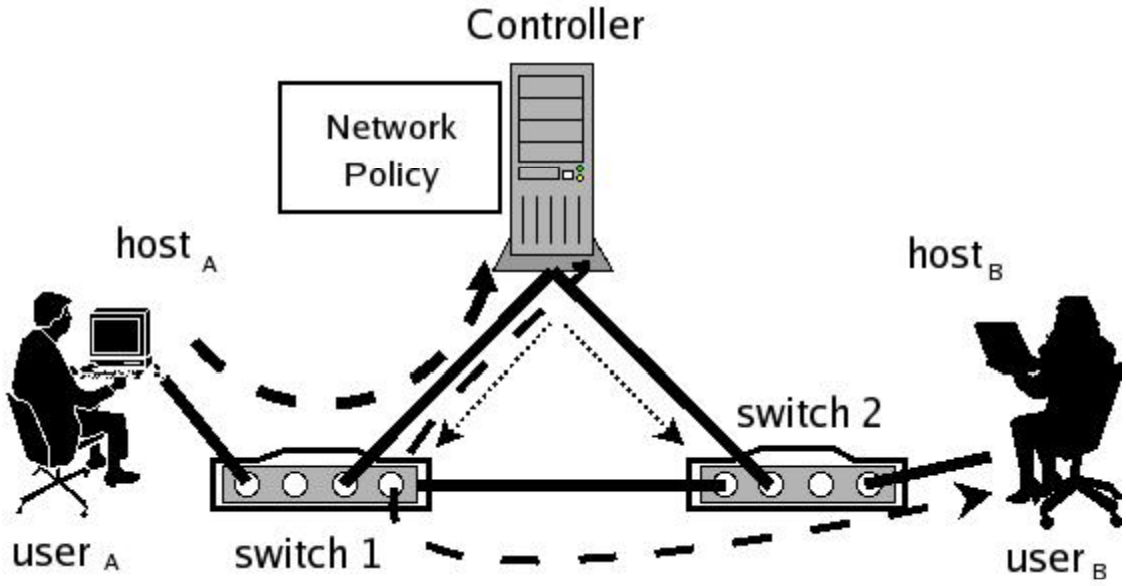
Glacial process of innovation made worse by captive standards process



- Driven by vendors
- Consumers largely locked out
- Glacial innovation

Ethane - precursor to OpenFlow

Martin Casado, Nick McKeown, Scott Shenker (2007)



Reactive, per flow enterprise network control

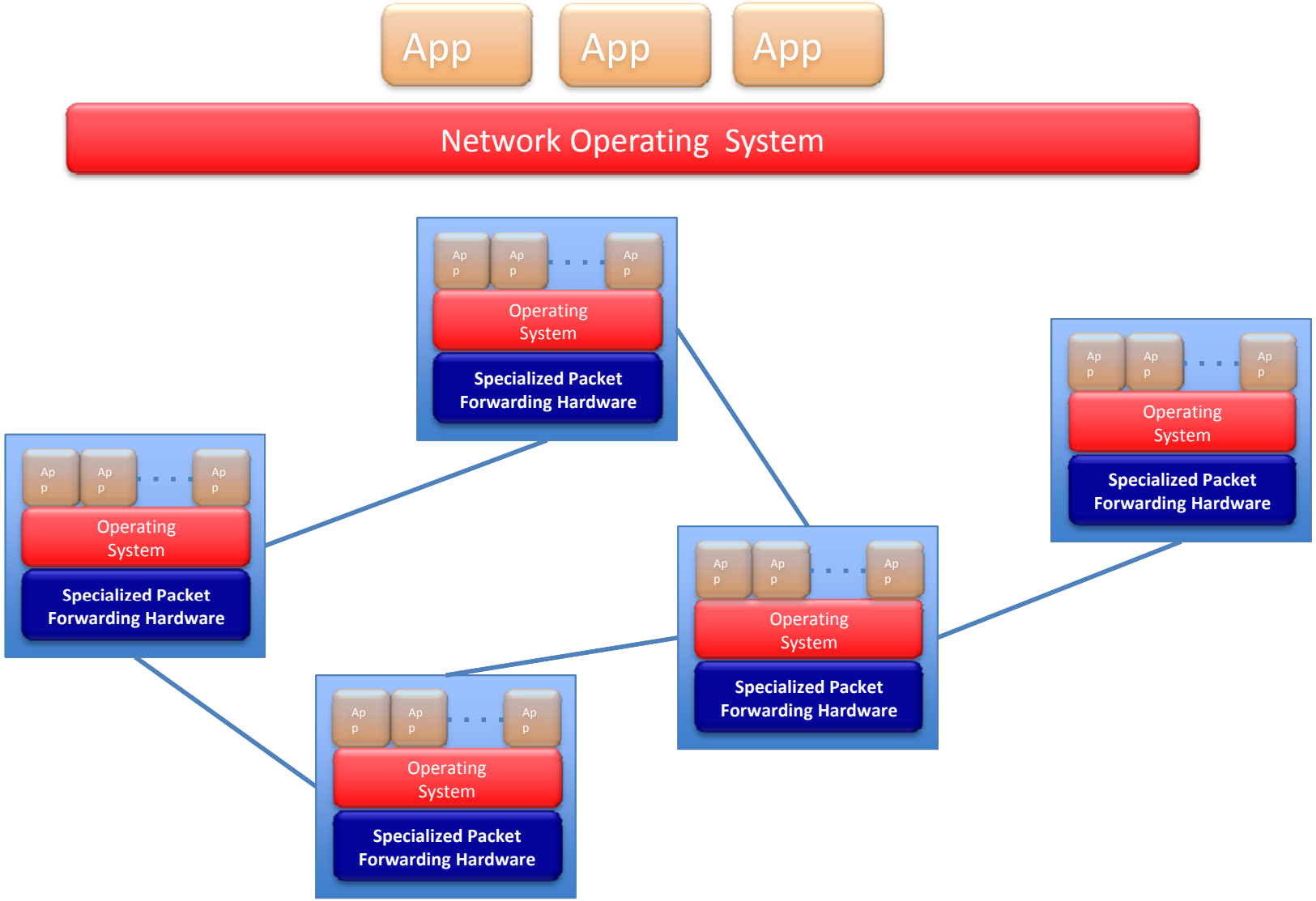
The network should be governed by policies declared over high- level names.

Policy should determine the path that packets follow.

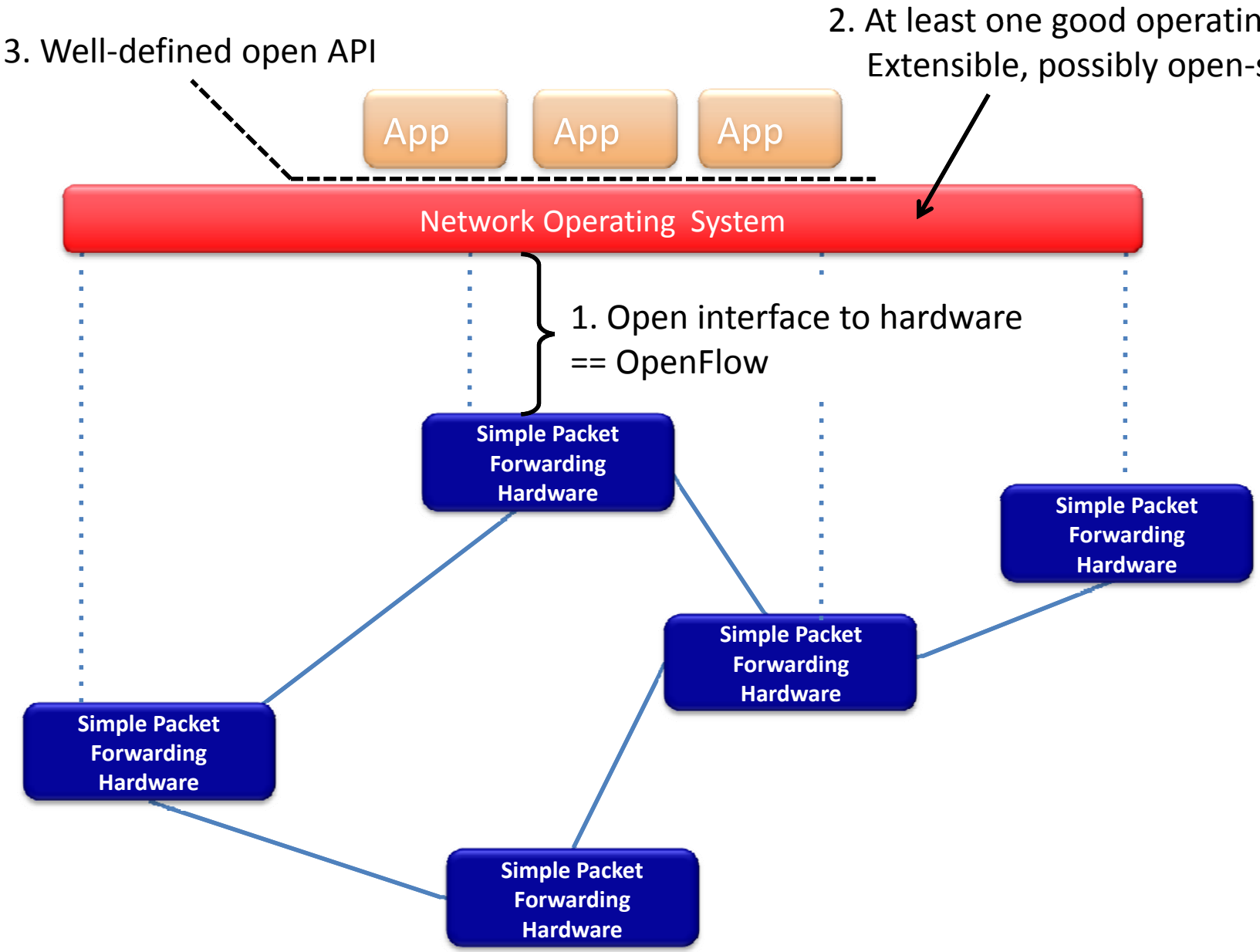
The network should enforce a strong binding between a packet and its origin.

Ethane's dumb, simple switches might be more broadly useful…

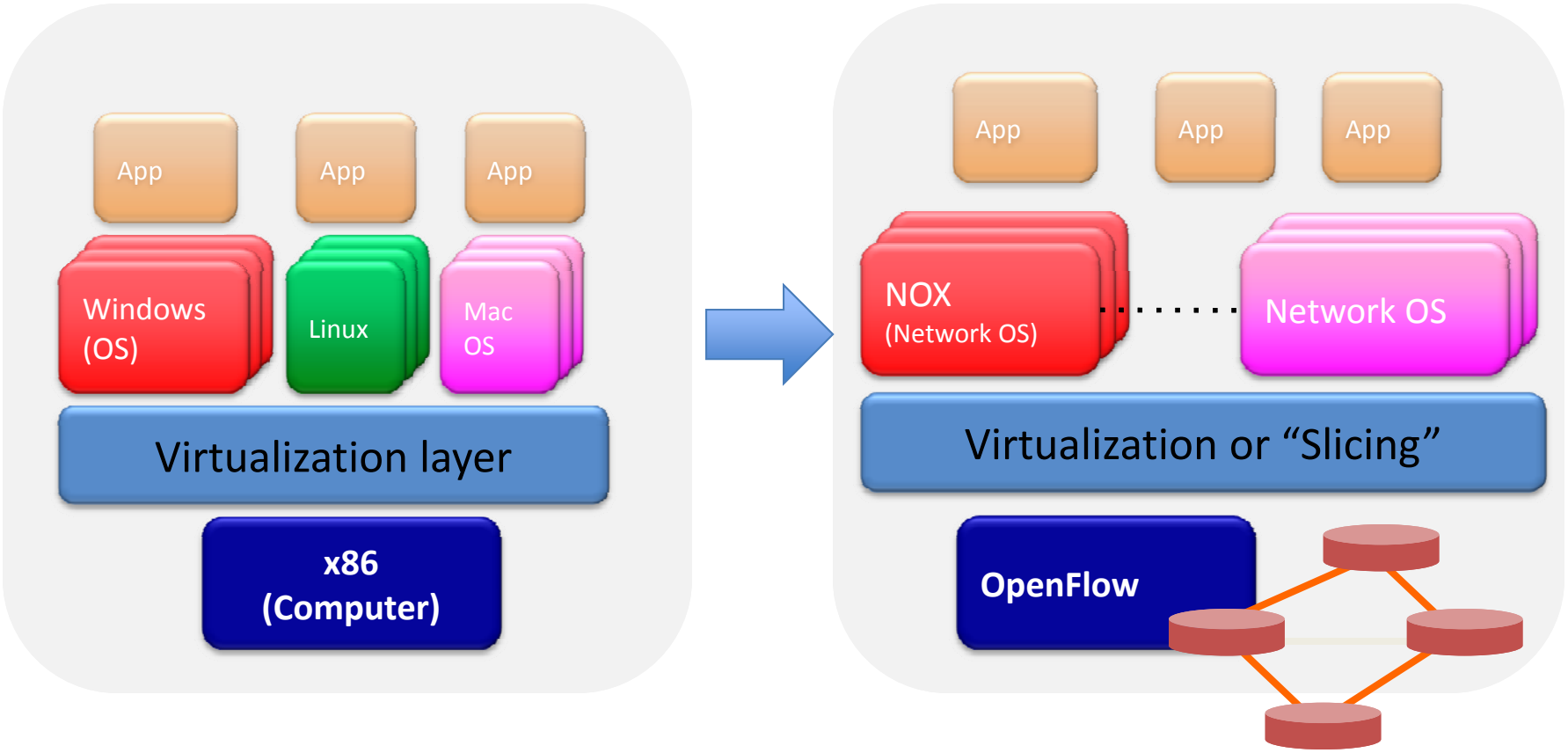
Change happening in non-traditional markets



The “Software Defined Network”



Trend



Computer Industry

Network Industry

Simple common stable hardware substrate below+ programmability + strong isolation model + competition above = Result : faster innovation

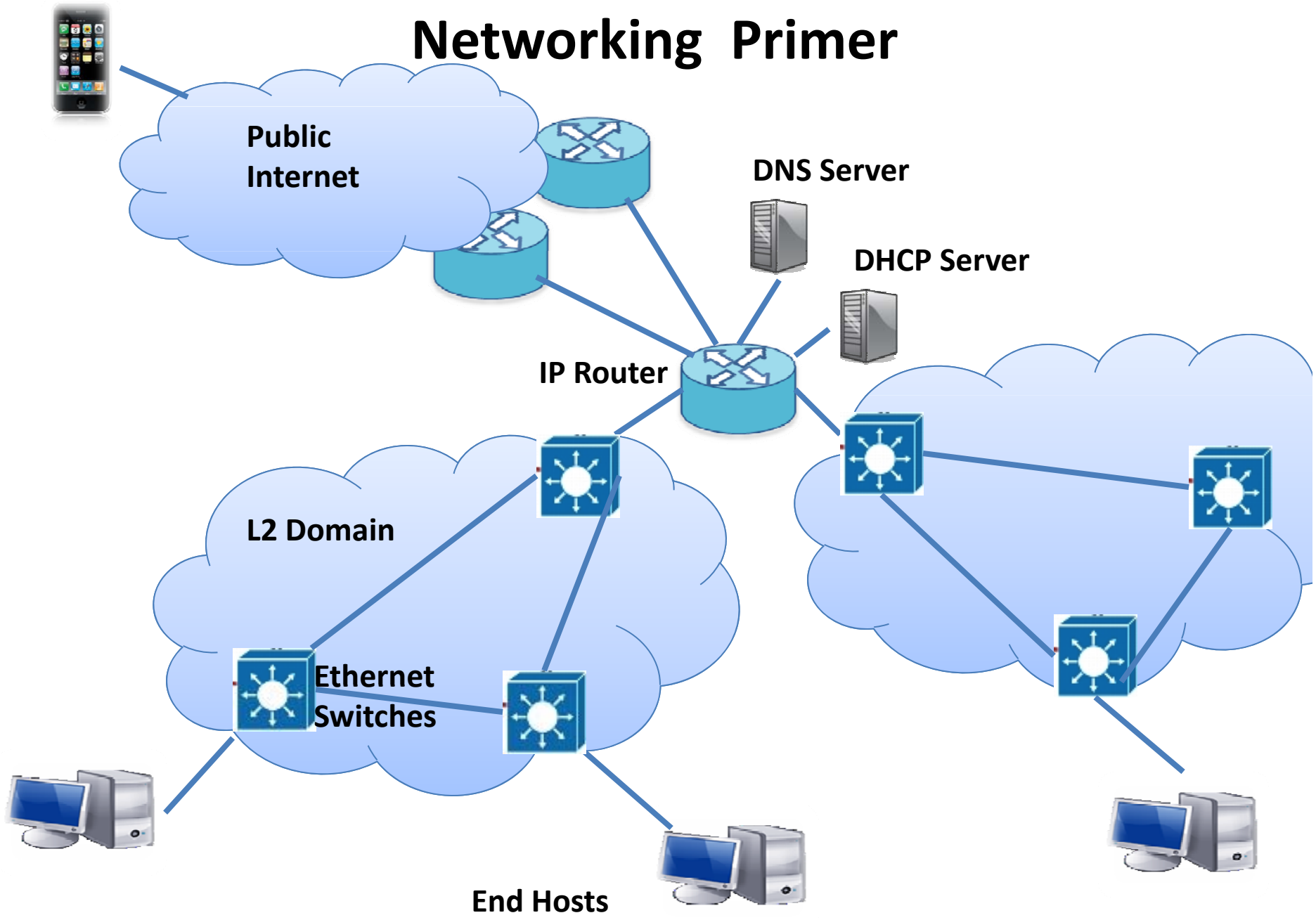
Outline

- Module 1: SDN/OpenFlow Introduction
- Module 2: OpenFlow/SDN Deep Dive
- Module 3: SDN Architecture
- Module 4: Operational Choices & Performance
- Module 5: Benefits & Use Cases
  - ❖ Demo: Network Virtualization
- Module 6: Demystifying SDN
- Q & A

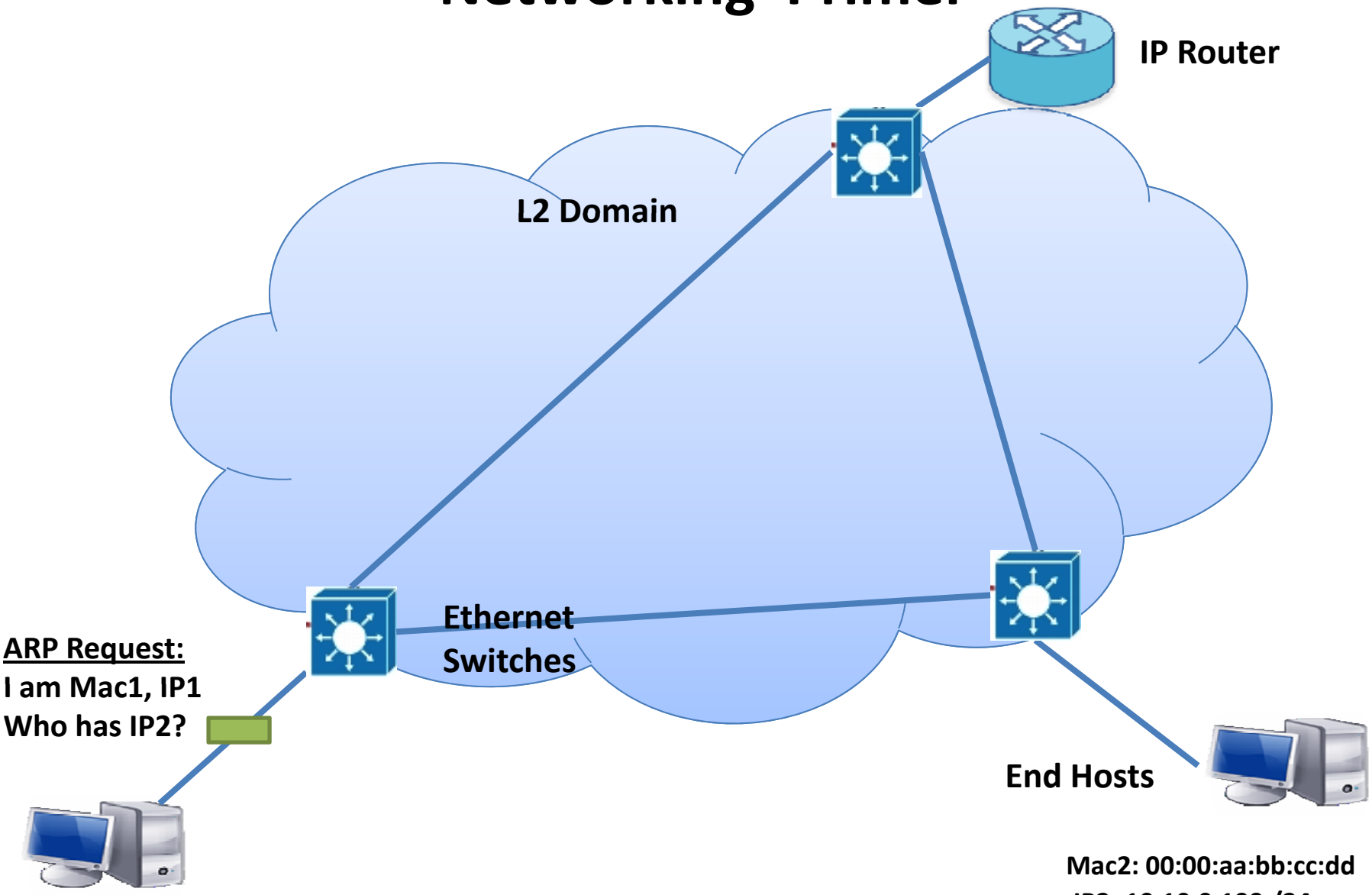
Module 2

OpenFlow/SDN
Deep Dive

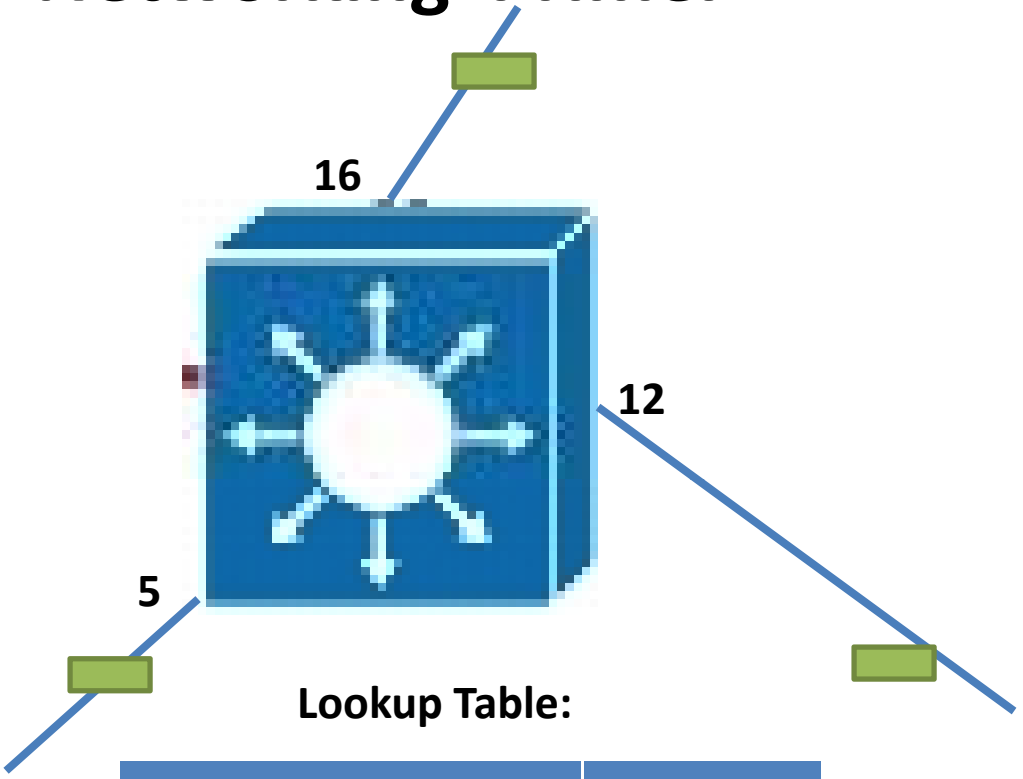
Networking Primer



Networking Primer



Networking Primer



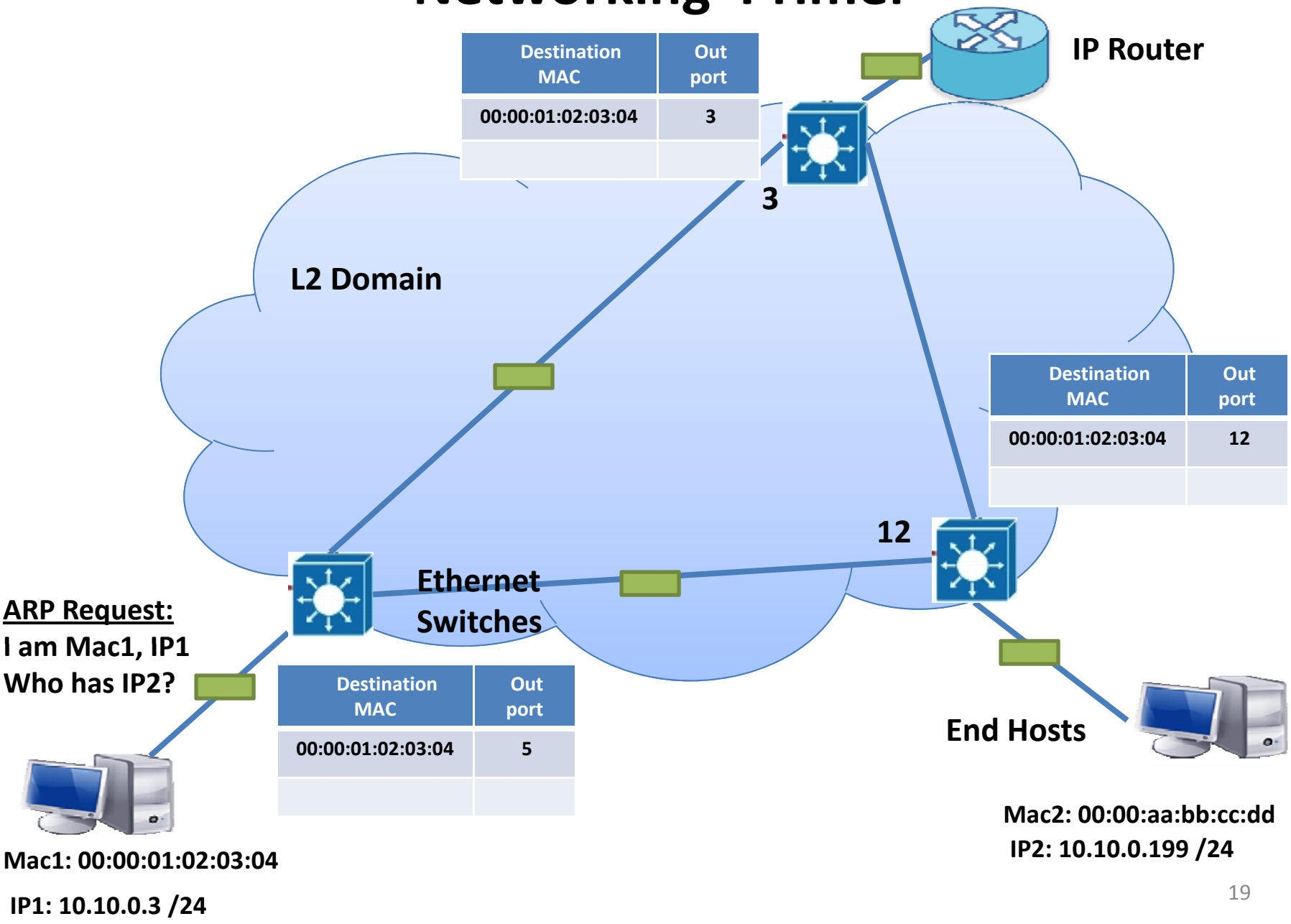
ARP Request:
I am Mac1, IP1
Who has IP2?

ARP Request:
Dst mac: ff:ff:ff:ff:ff:ff
Src mac: 00:00:01:02:03:04
Eth type: 0806
Payload:
Opcode: Arp Request (1)
Sender mac : 00:00:01:02:03:04
Sender IP : 10.10.0.3
Target mac : 00:00:00:00:00:00
Target IP : 10.10.0.199

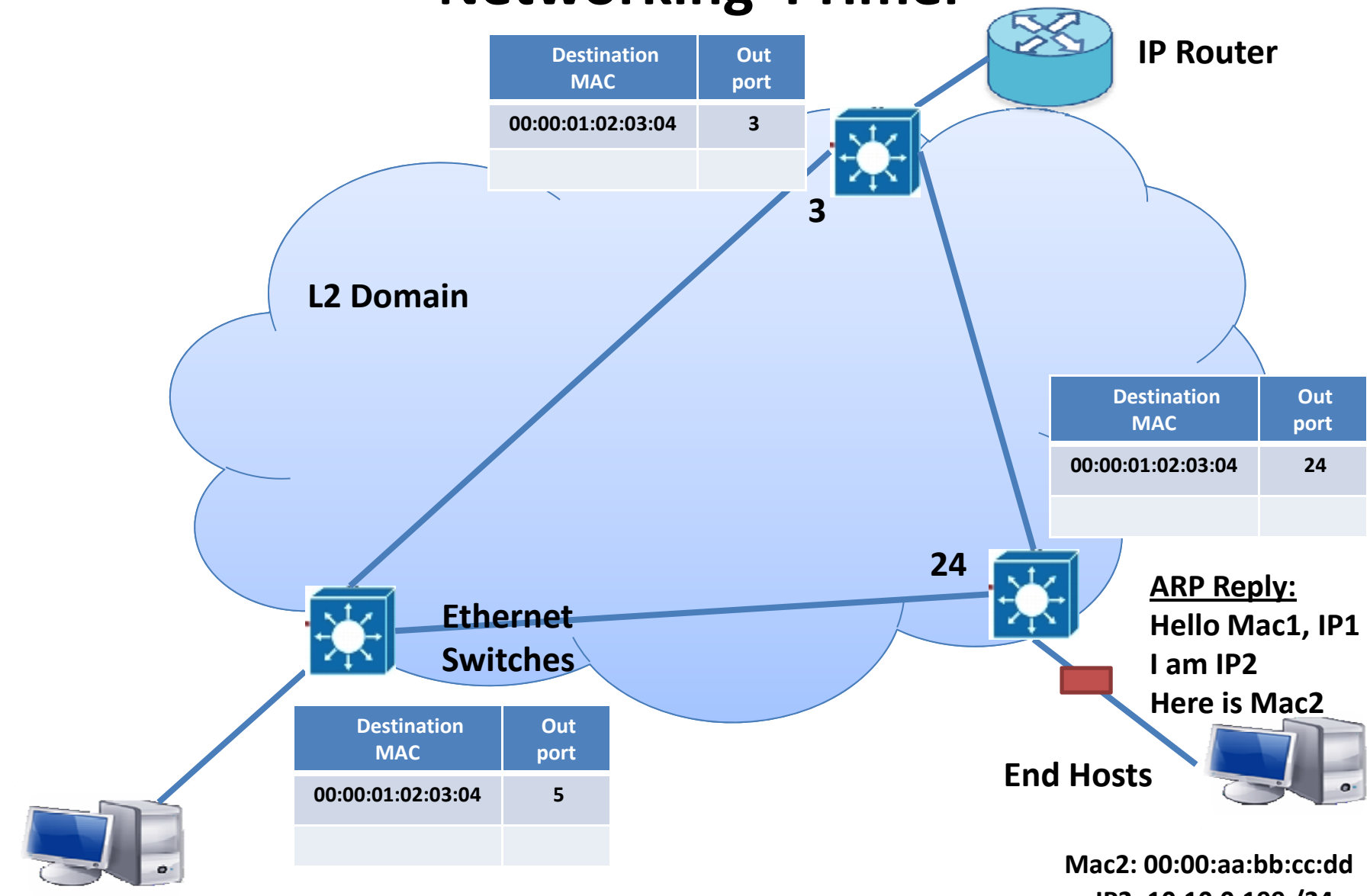
Lookup Table:

Destination MAC	Out port
00:00:01:02:03:04	5

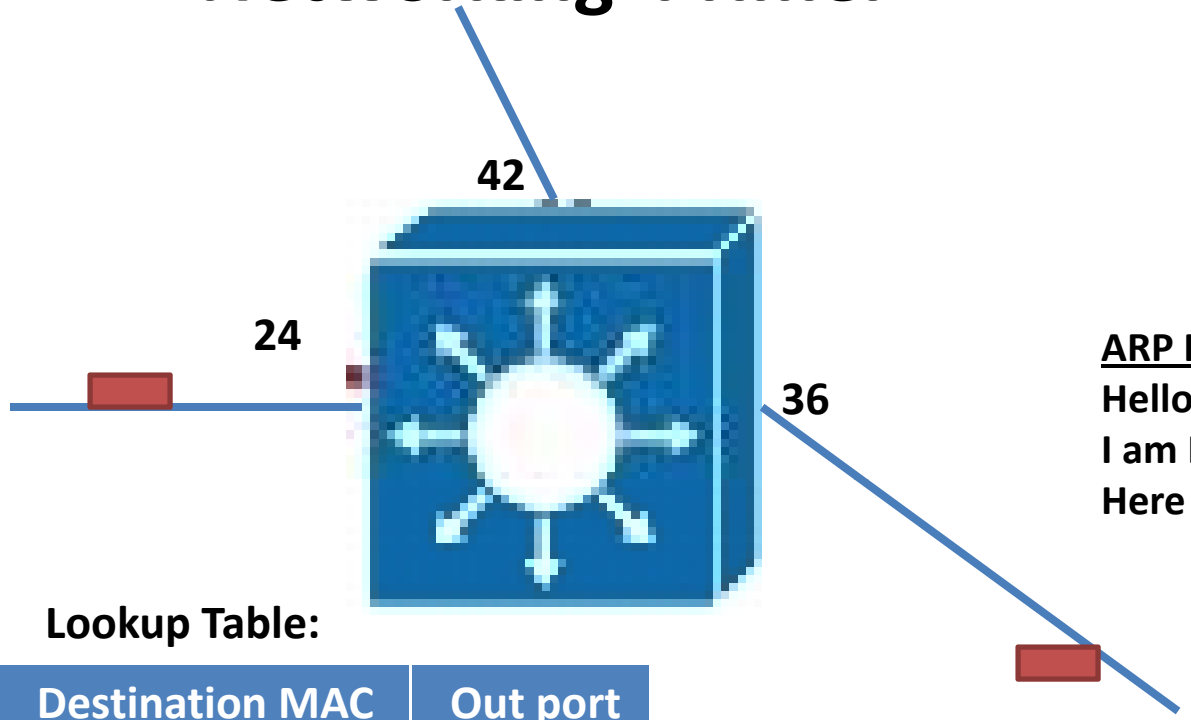
Networking Primer



Networking Primer



Networking Primer



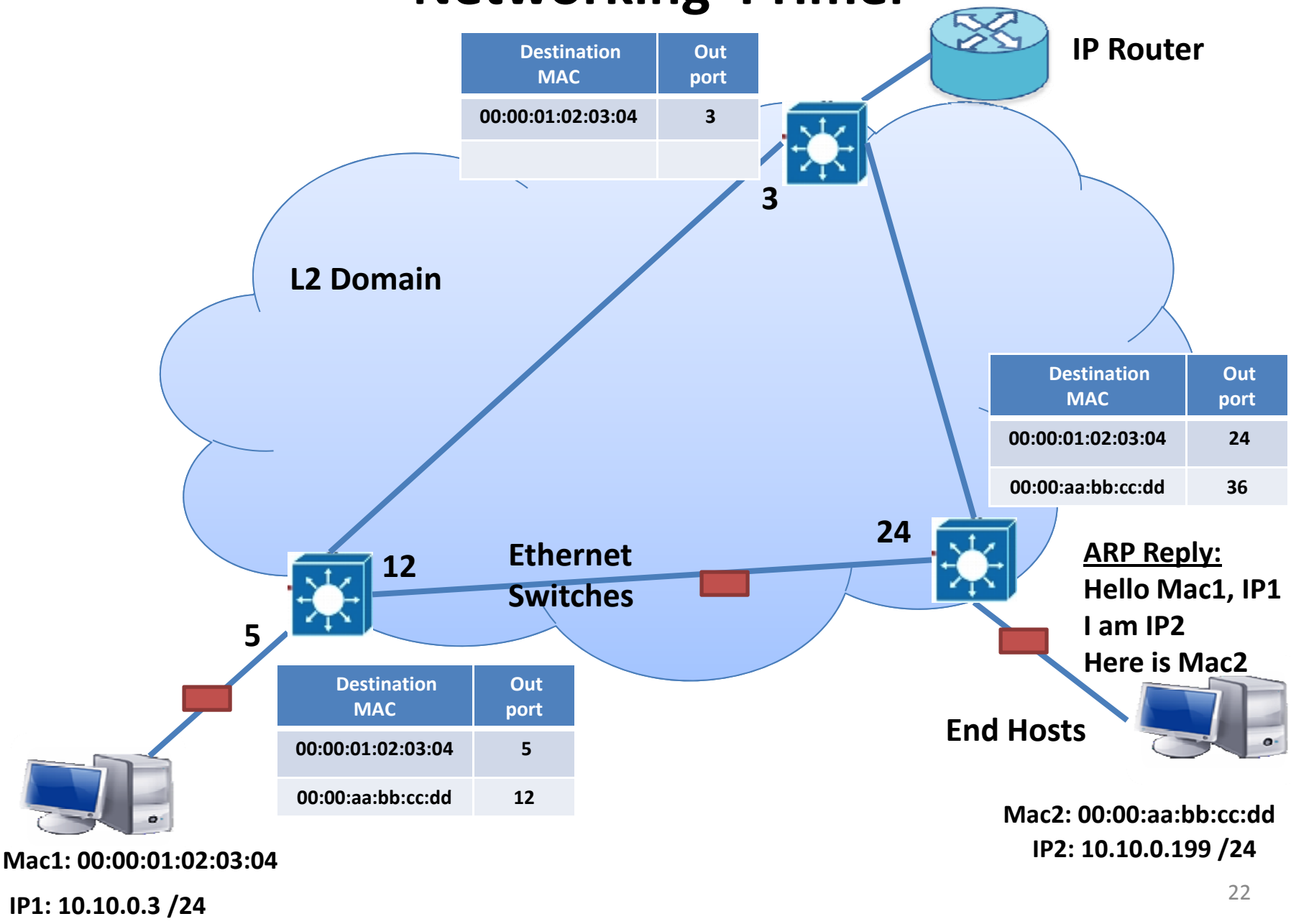
ARP Reply:
Hello Mac1, IP1
I am IP2
Here is Mac2

Lookup Table:

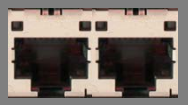
Destination MAC	Out port
00:00:01:02:03:04	24
00:00:aa:bb:cc:dd	36

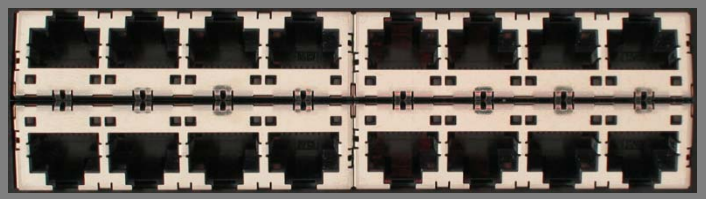
ARP Reply:
Dst mac: 00:00:01:02:03:04
Src mac: 00:00:aa:bb:cc:dd
Eth type: 0806
Payload:
Opcode : Arp Reply (2)
Sender mac : 00:00:aa:bb:cc:dd
Sender IP : 10.10.0.199
Target mac : 00:00:01:02:03:04
Target IP : 10.10.0.3

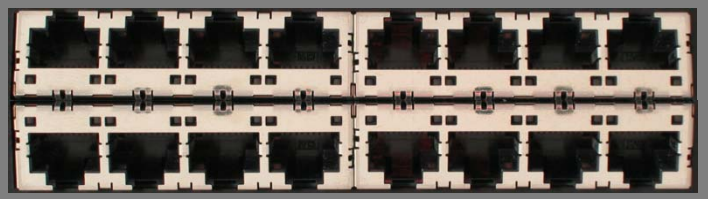
Networking Primer

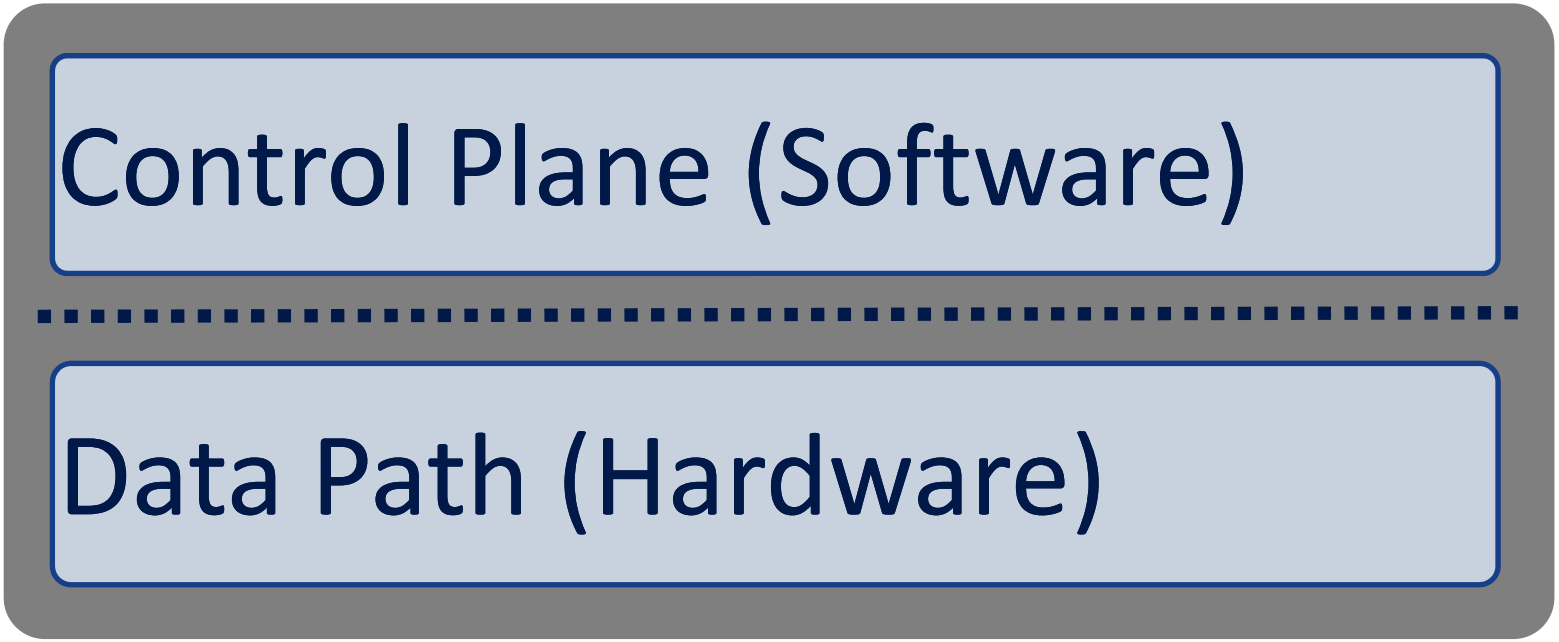


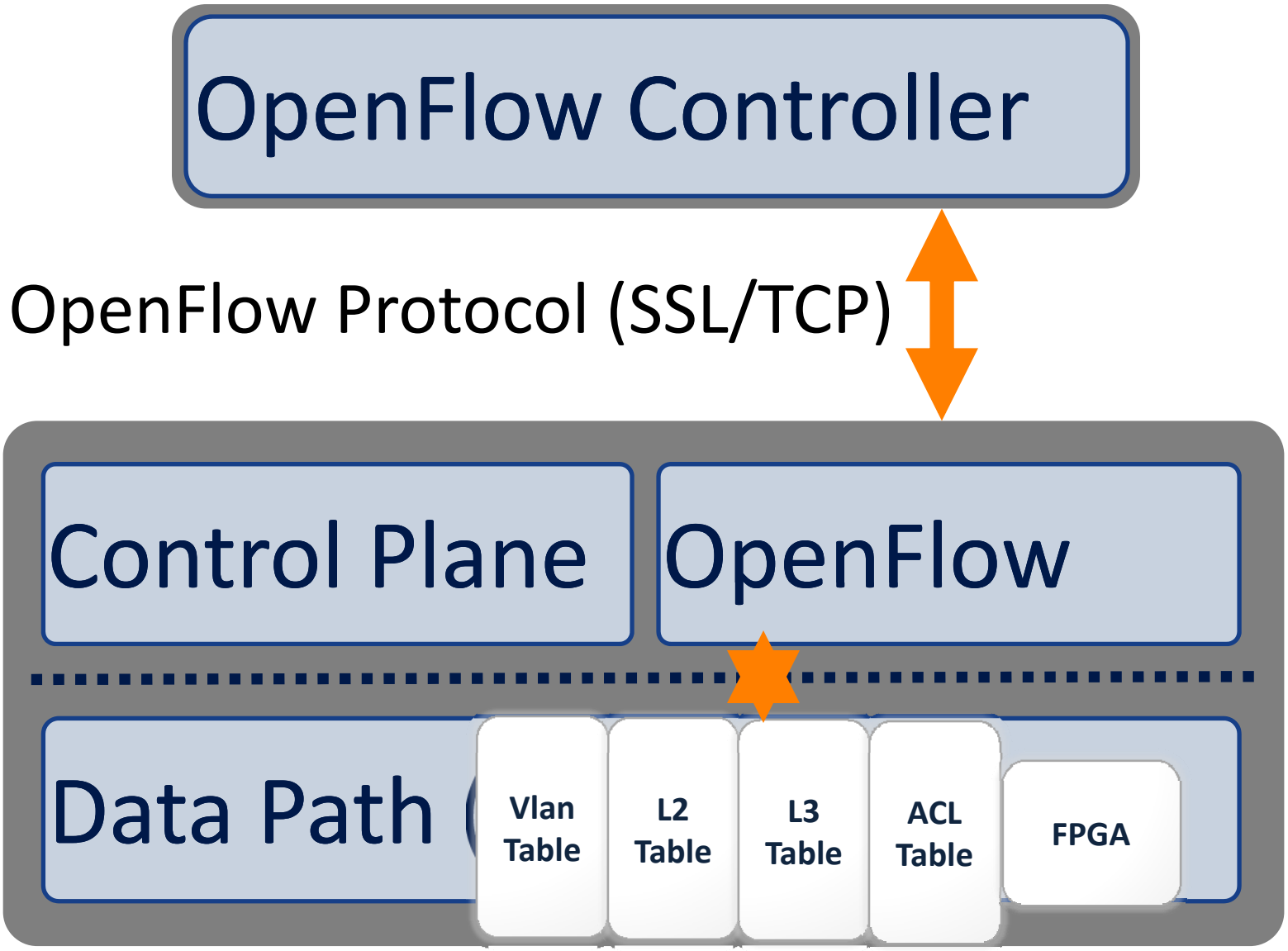
Switch/Router



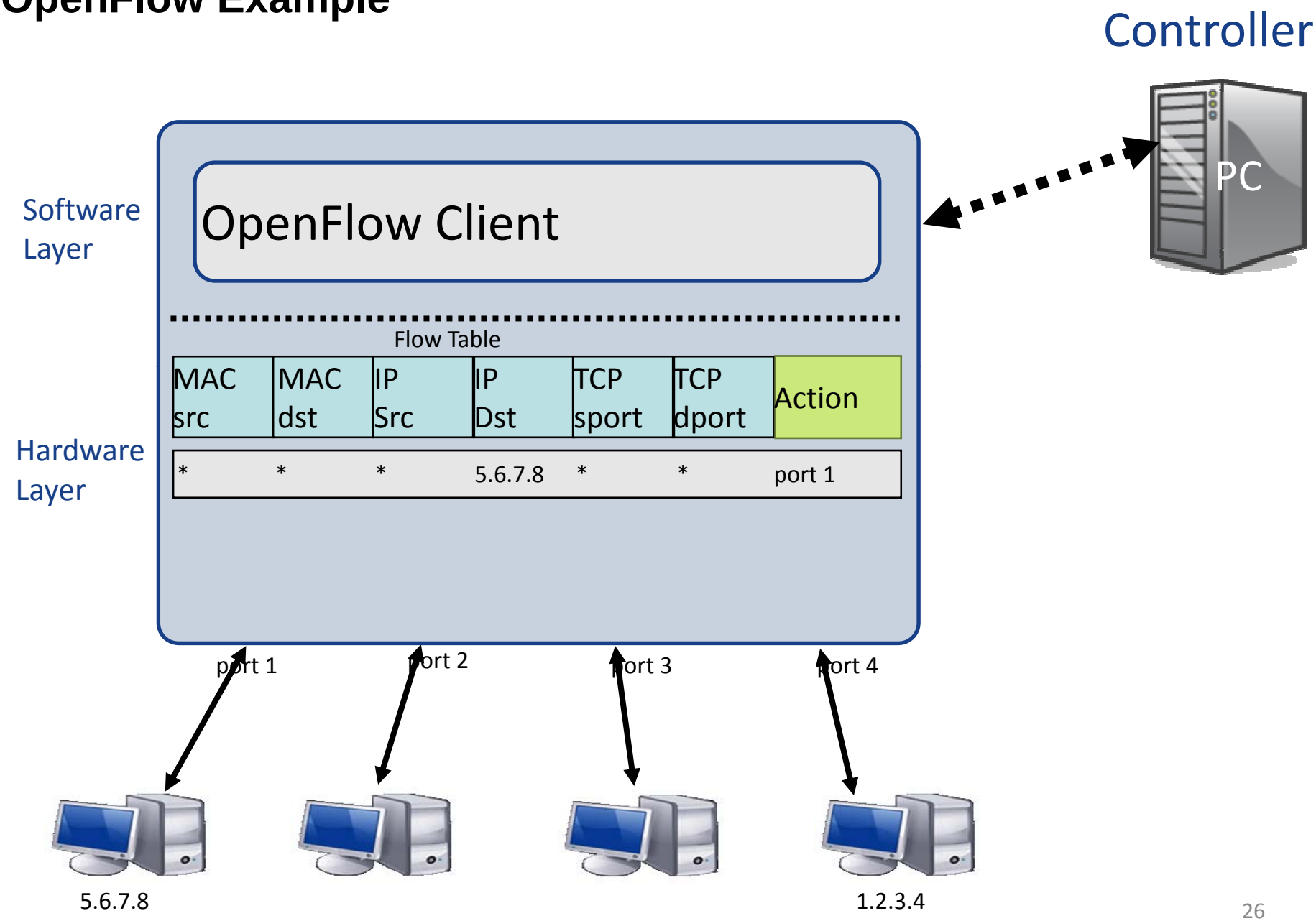





Control Plane (Software)
Data Path (Hardware)

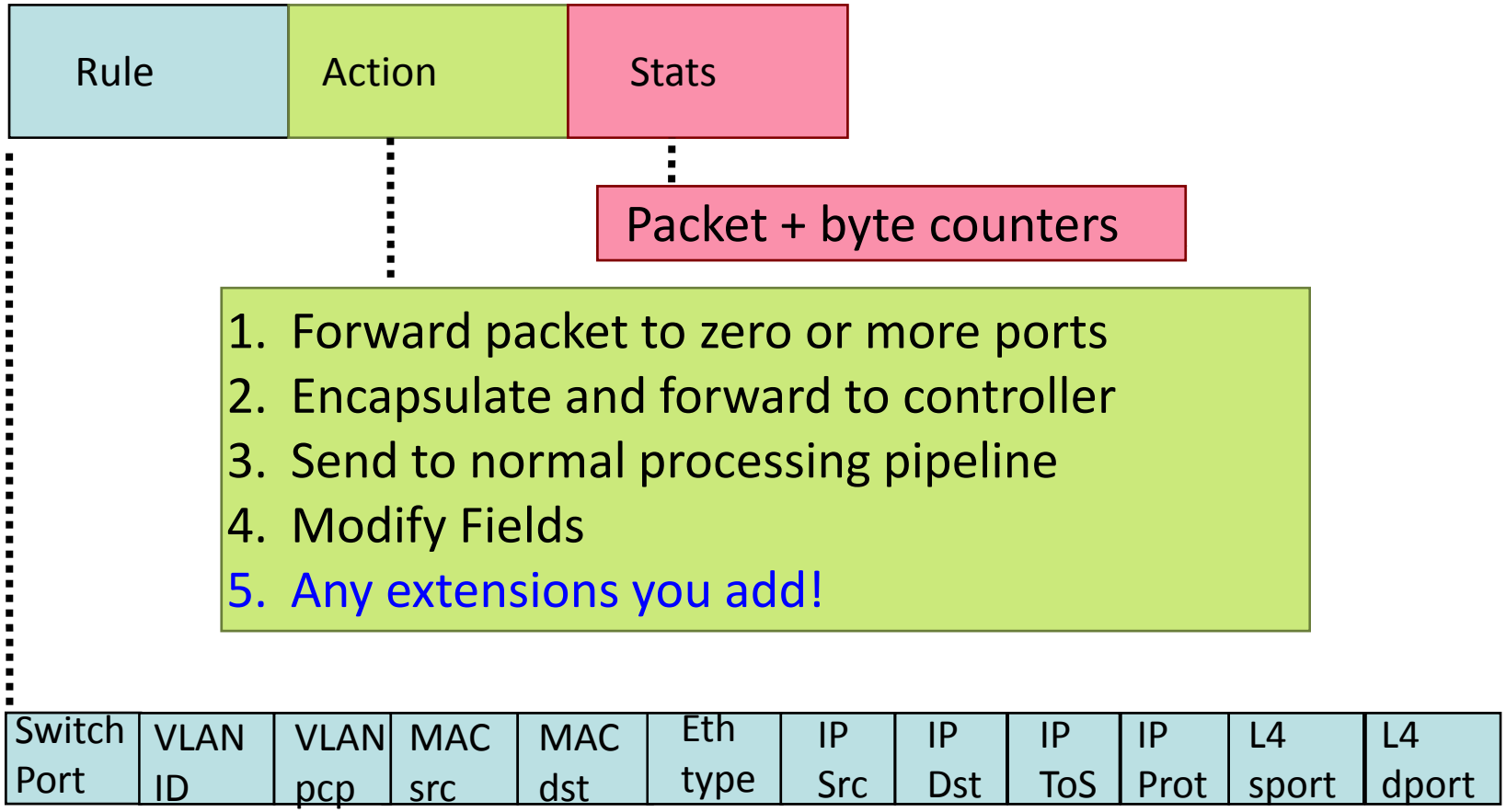

OpenFlow Controller
OpenFlow Protocol (SSL/TCP)
Control Plane
OpenFlow
Data Path
Vlan Table
L2 Table
L3 Table
ACL Table
FPGA

OpenFlow Example



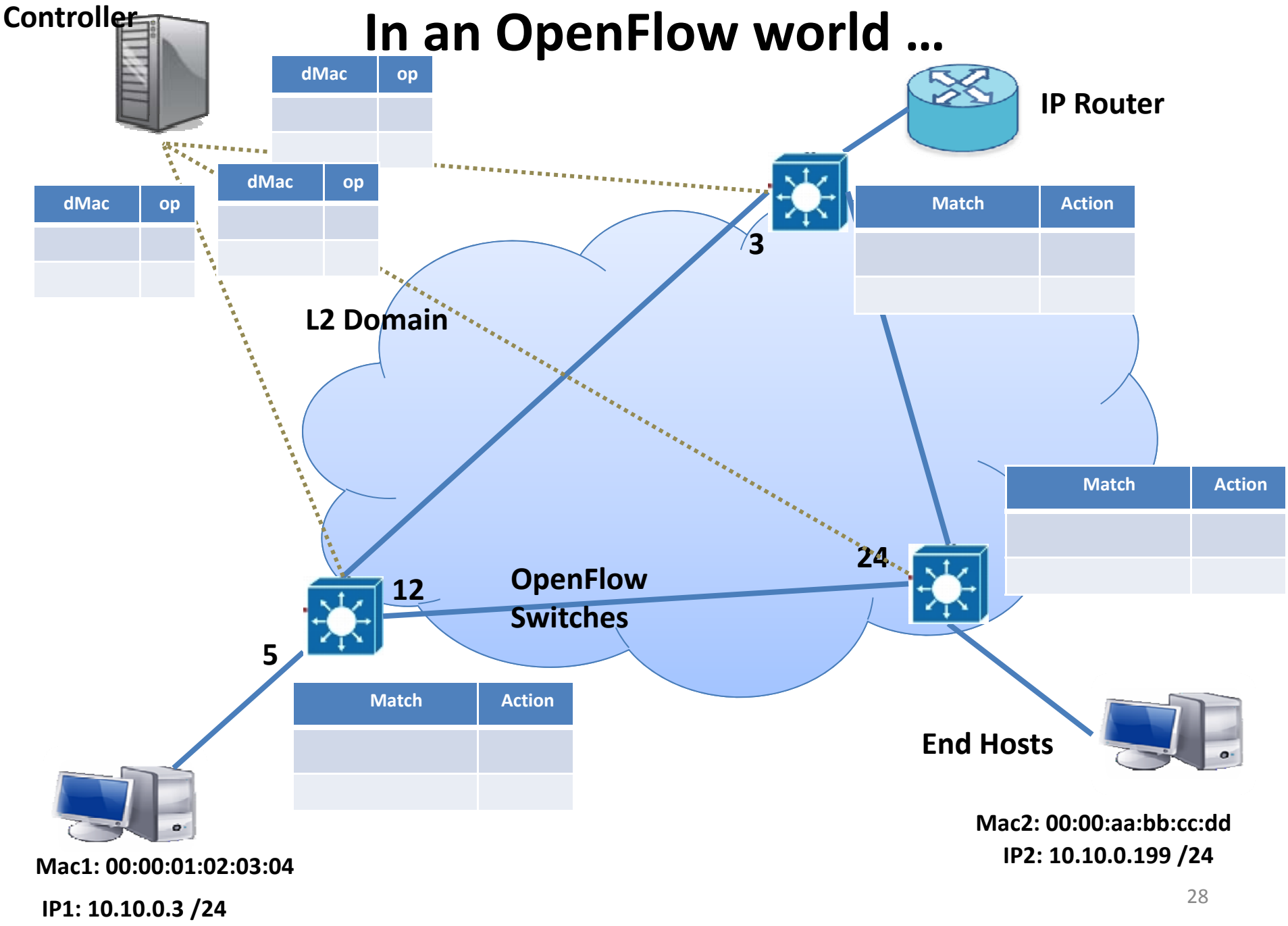
OpenFlow Basics

Flow Table Entries

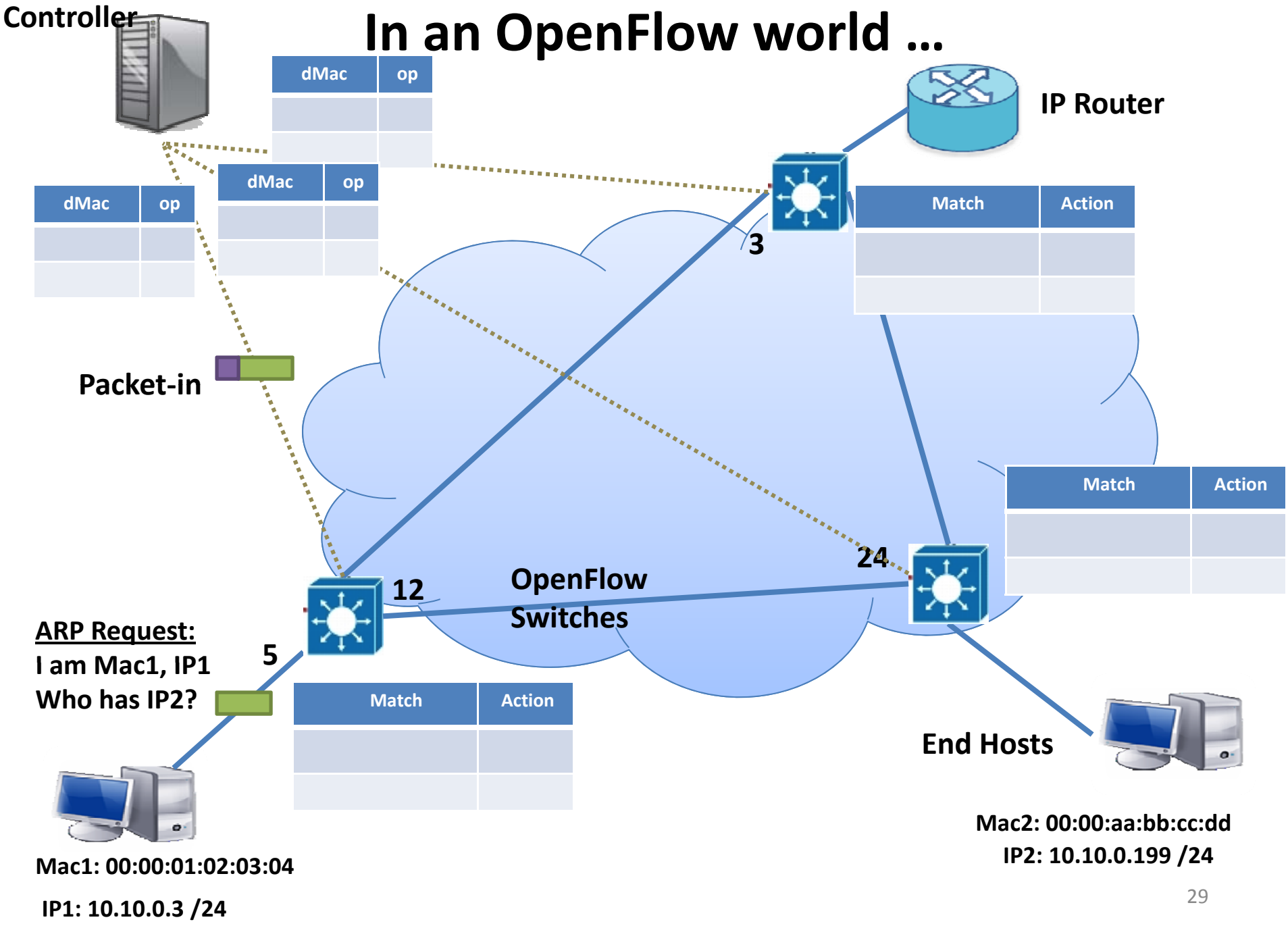


+ mask what fields to match

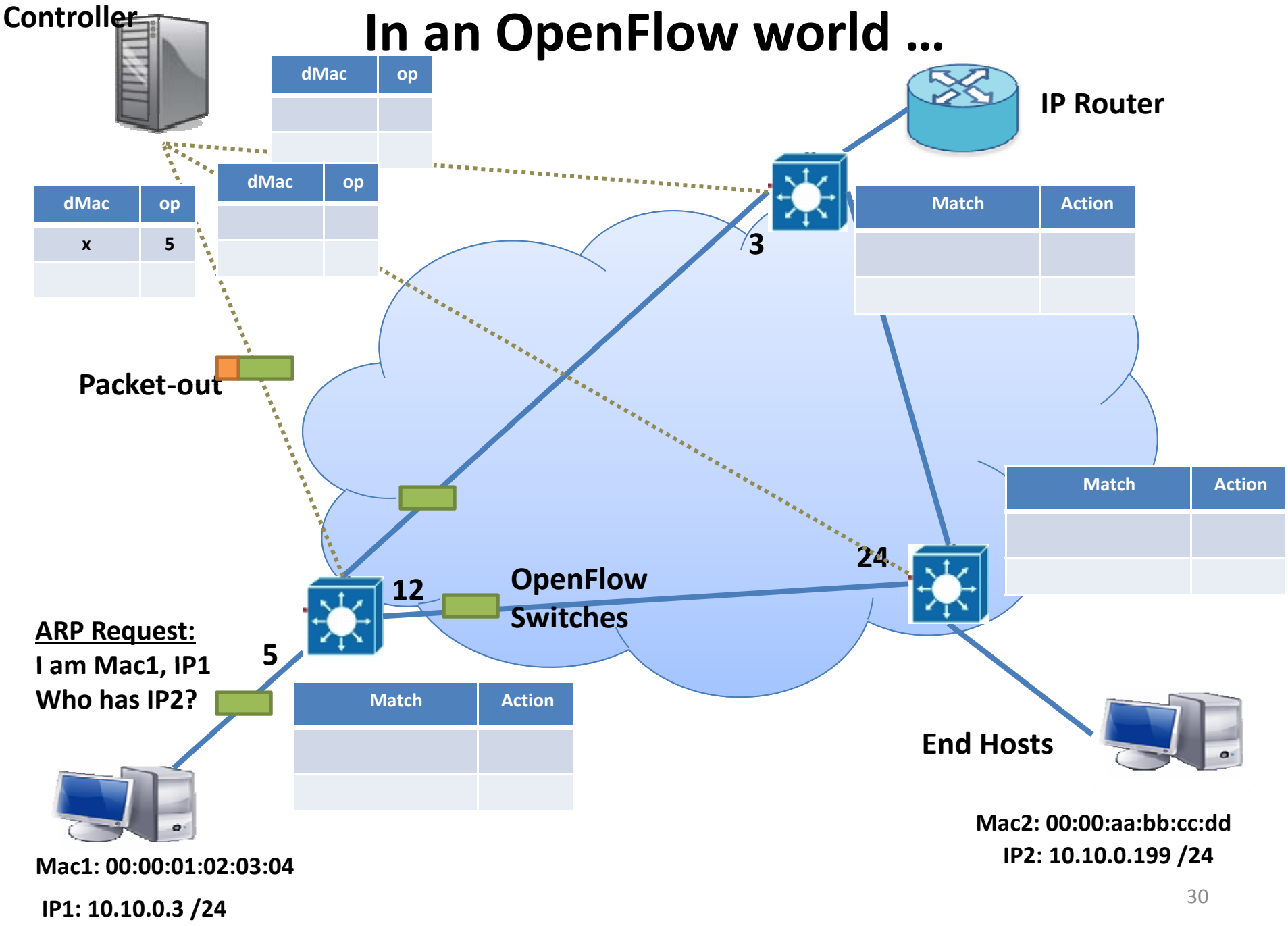
In an OpenFlow world …



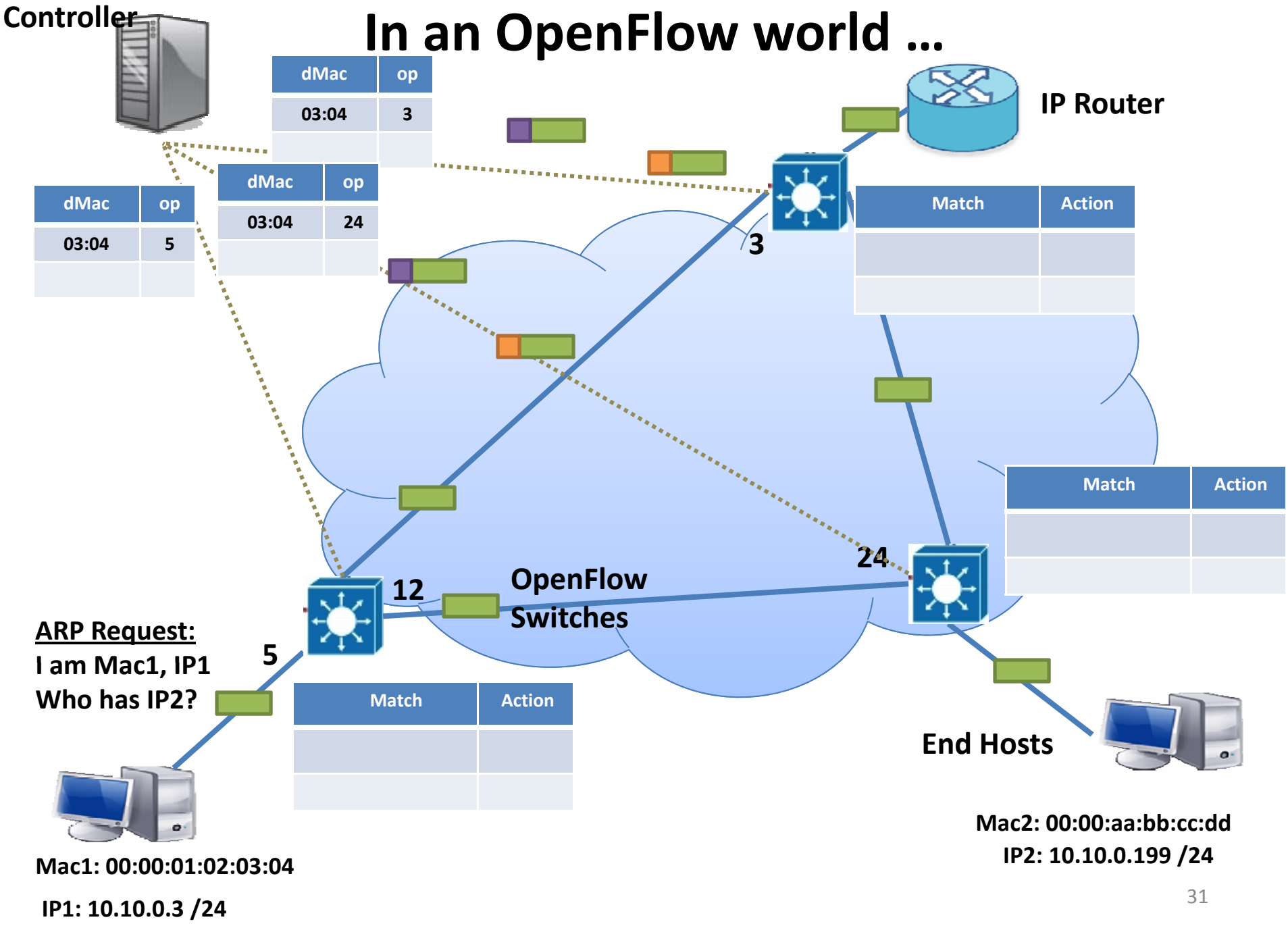
In an OpenFlow world …



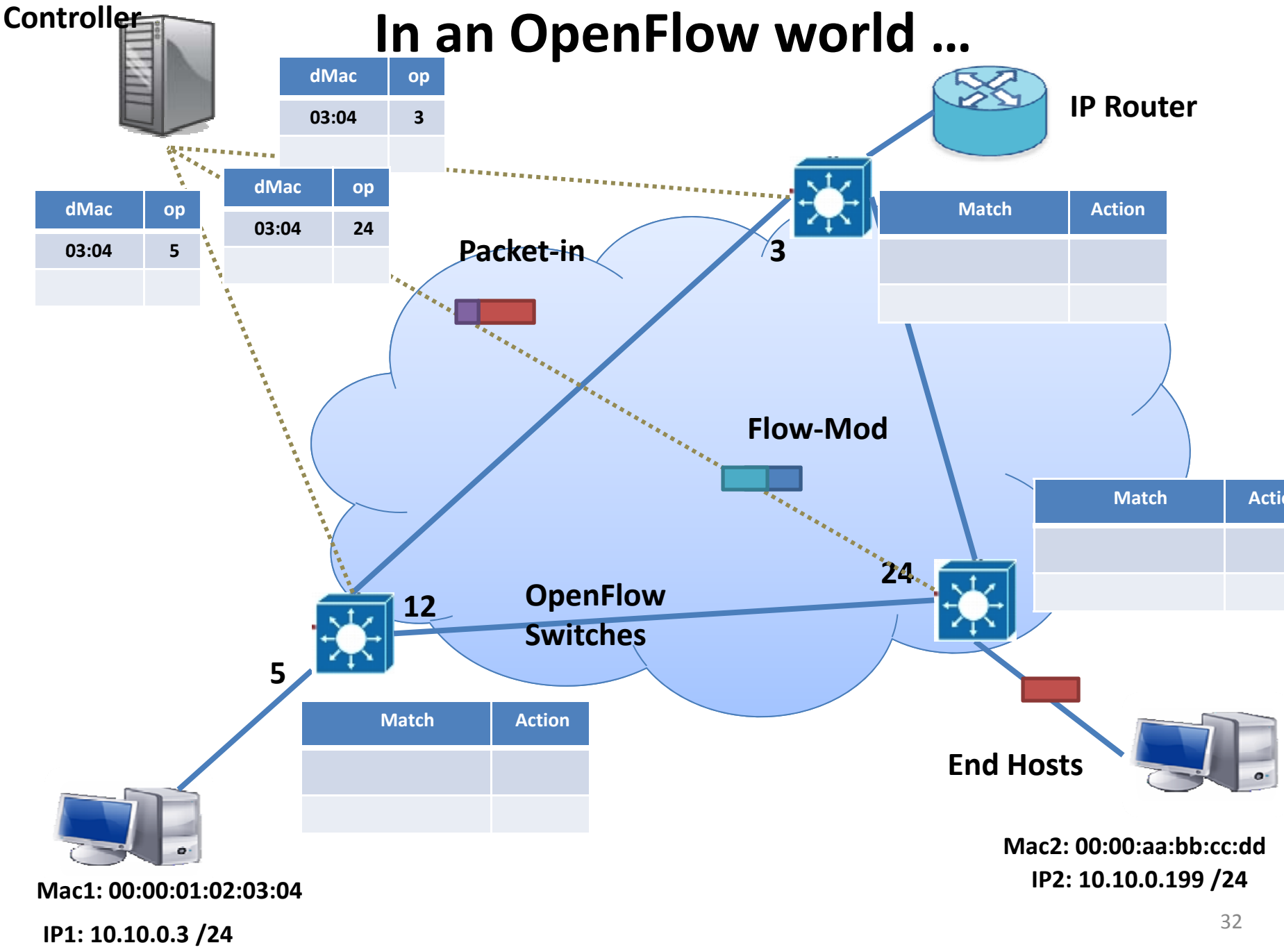
In an OpenFlow world …



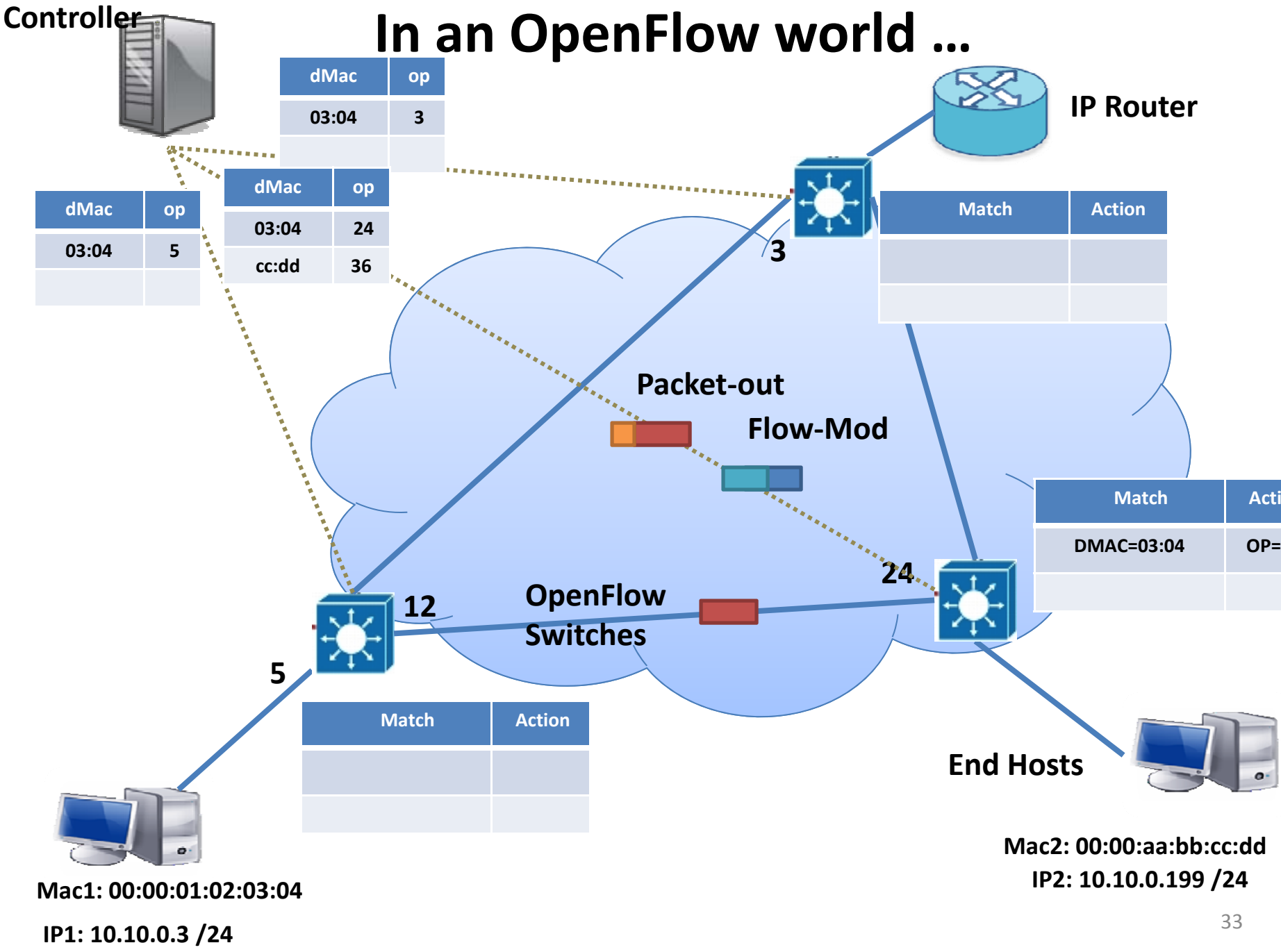
In an OpenFlow world …



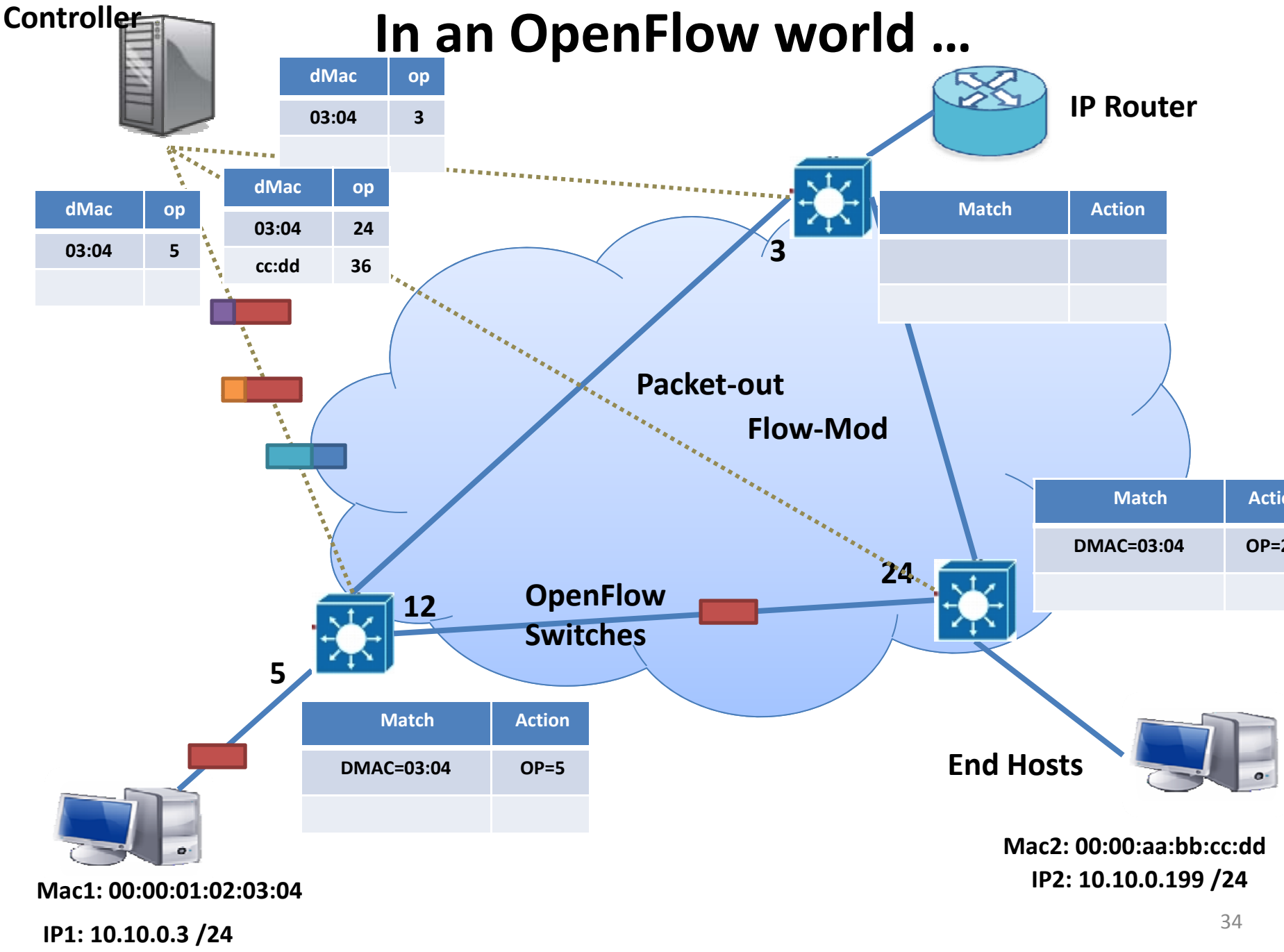
In an OpenFlow world …



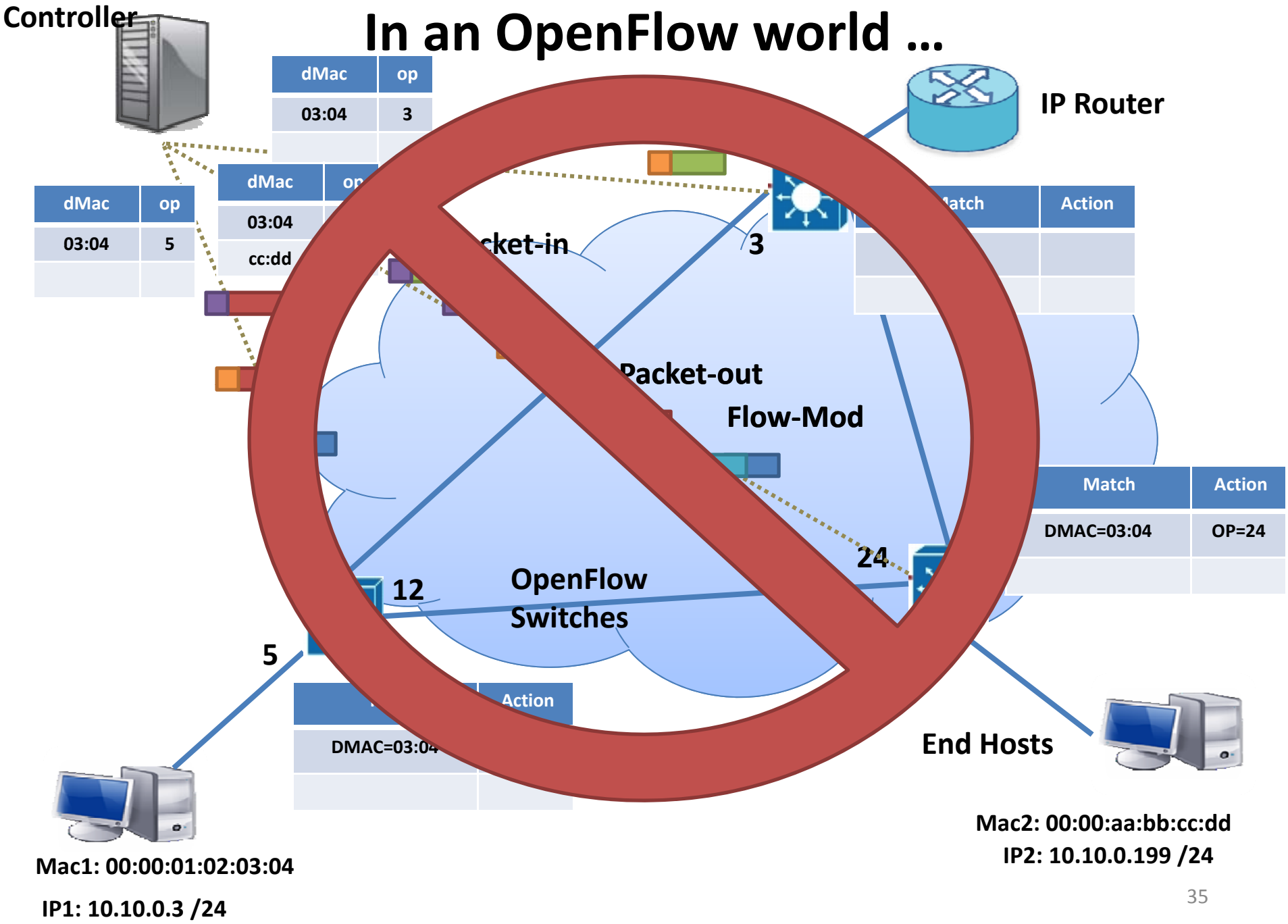
In an OpenFlow world …



In an OpenFlow world …



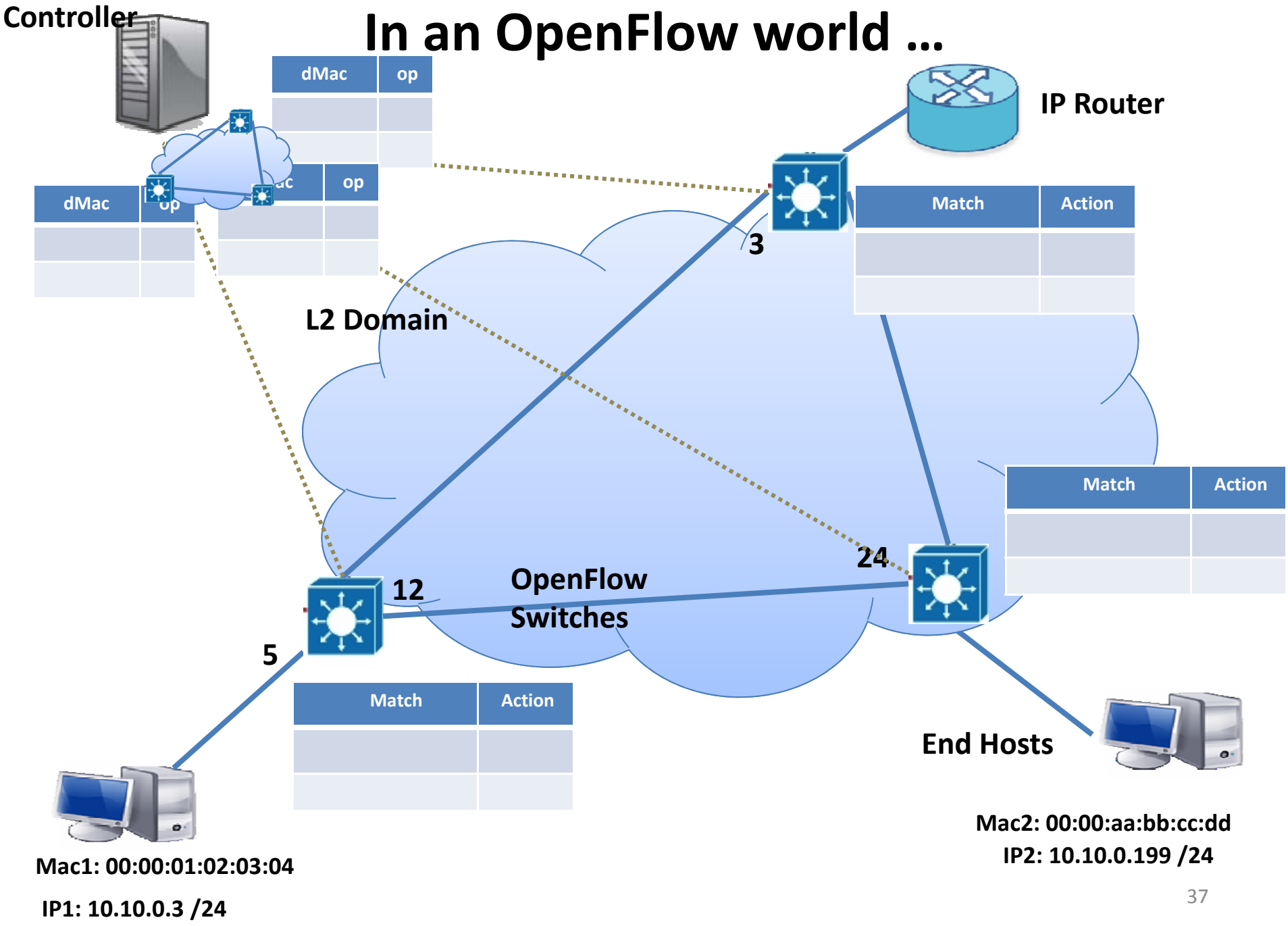
In an OpenFlow world …



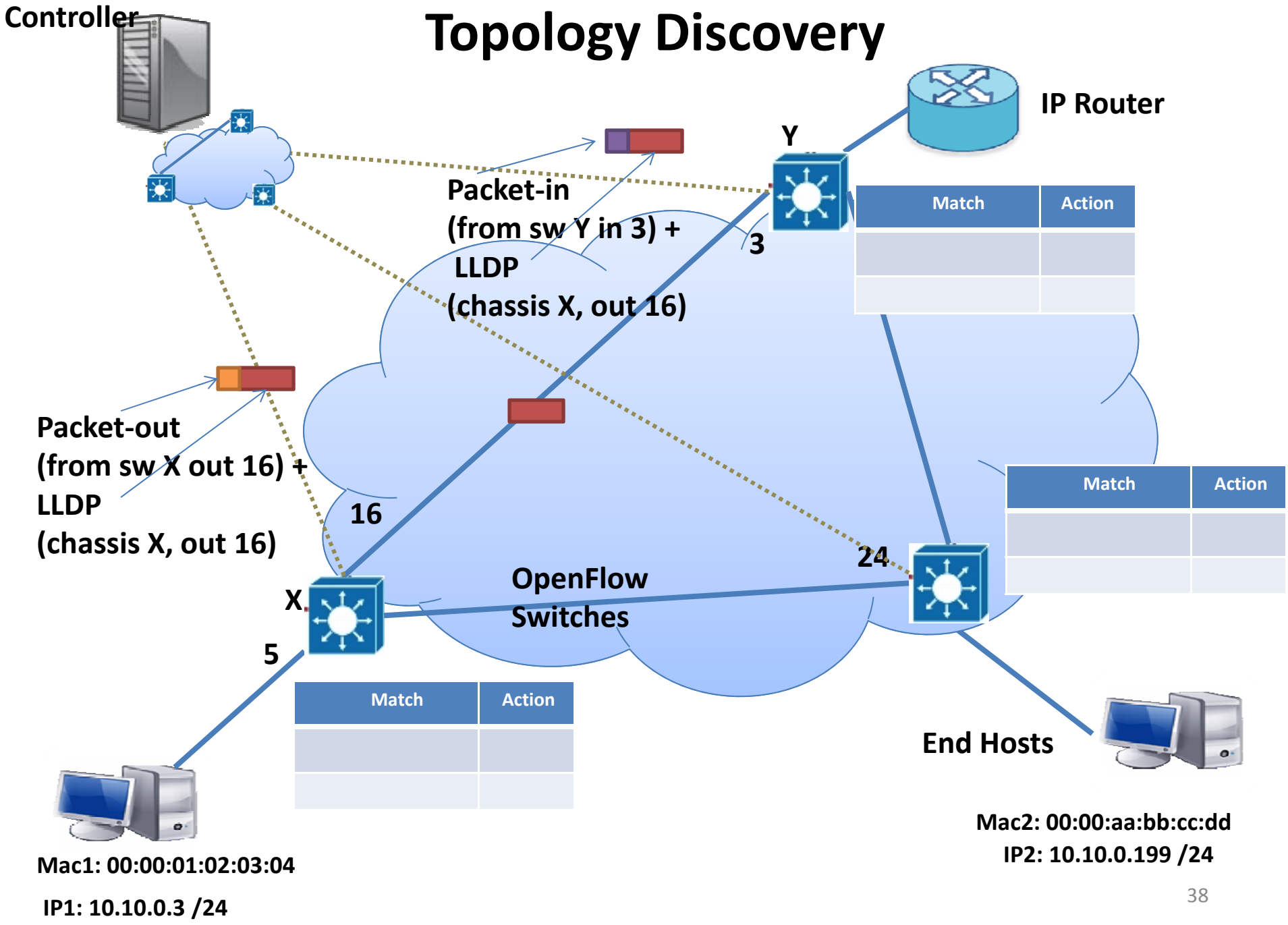
In an OpenFlow world:

- learn NOT on a per switch basis, but on a network basis
- think pathing not switching or routing
- network is programmable
    - think differently
    - flexible match
    - flexible actions

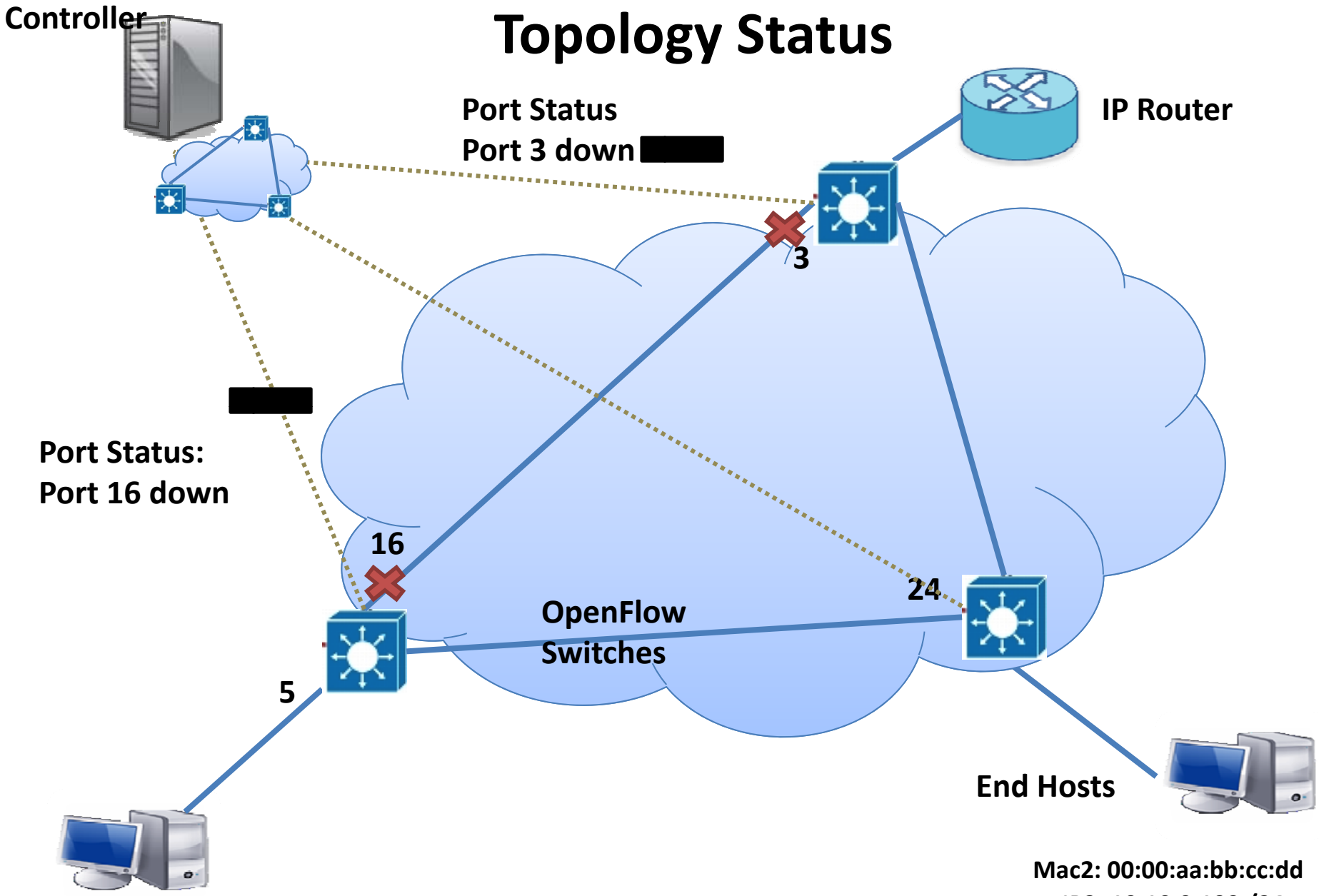
In an OpenFlow world …



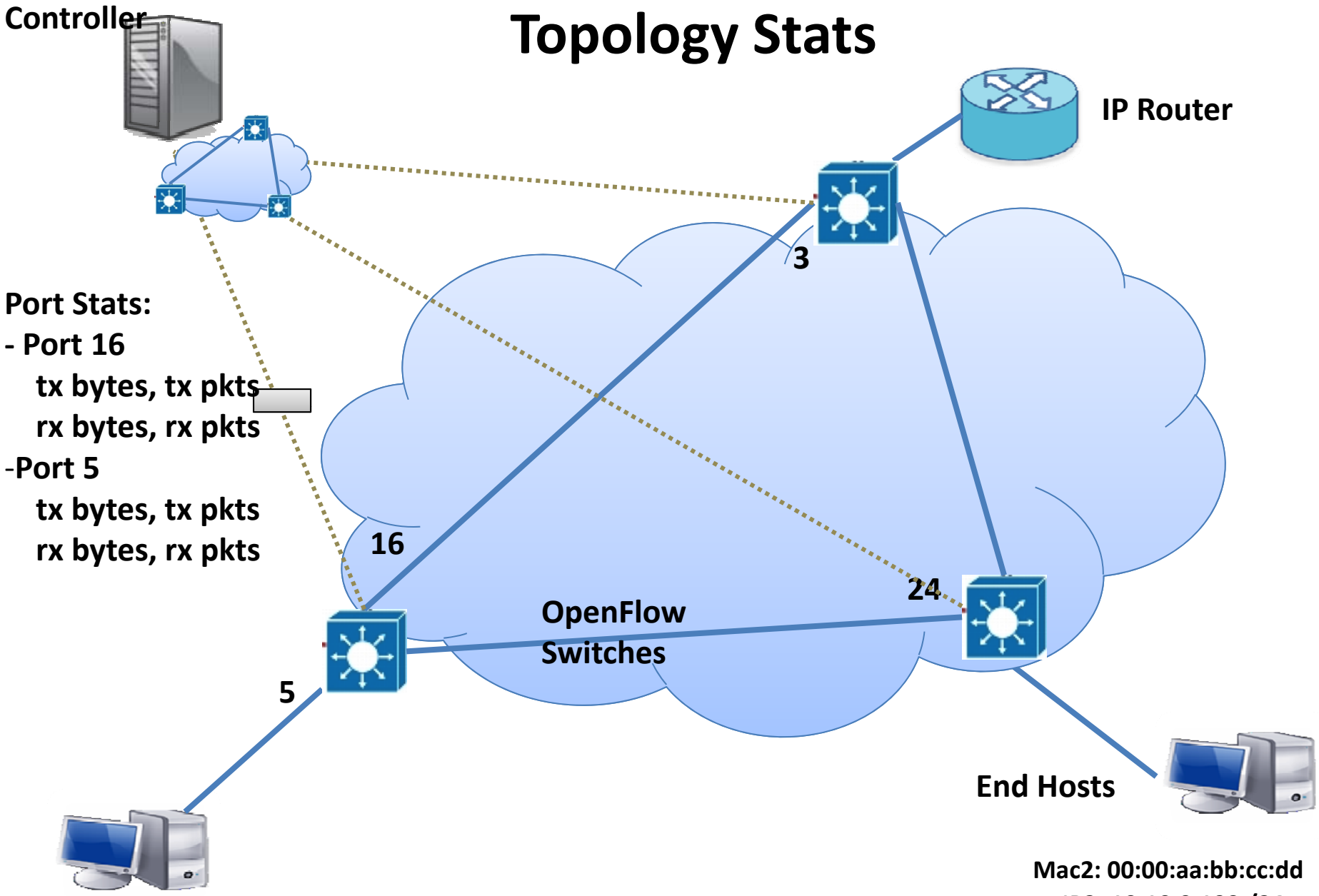
Topology Discovery

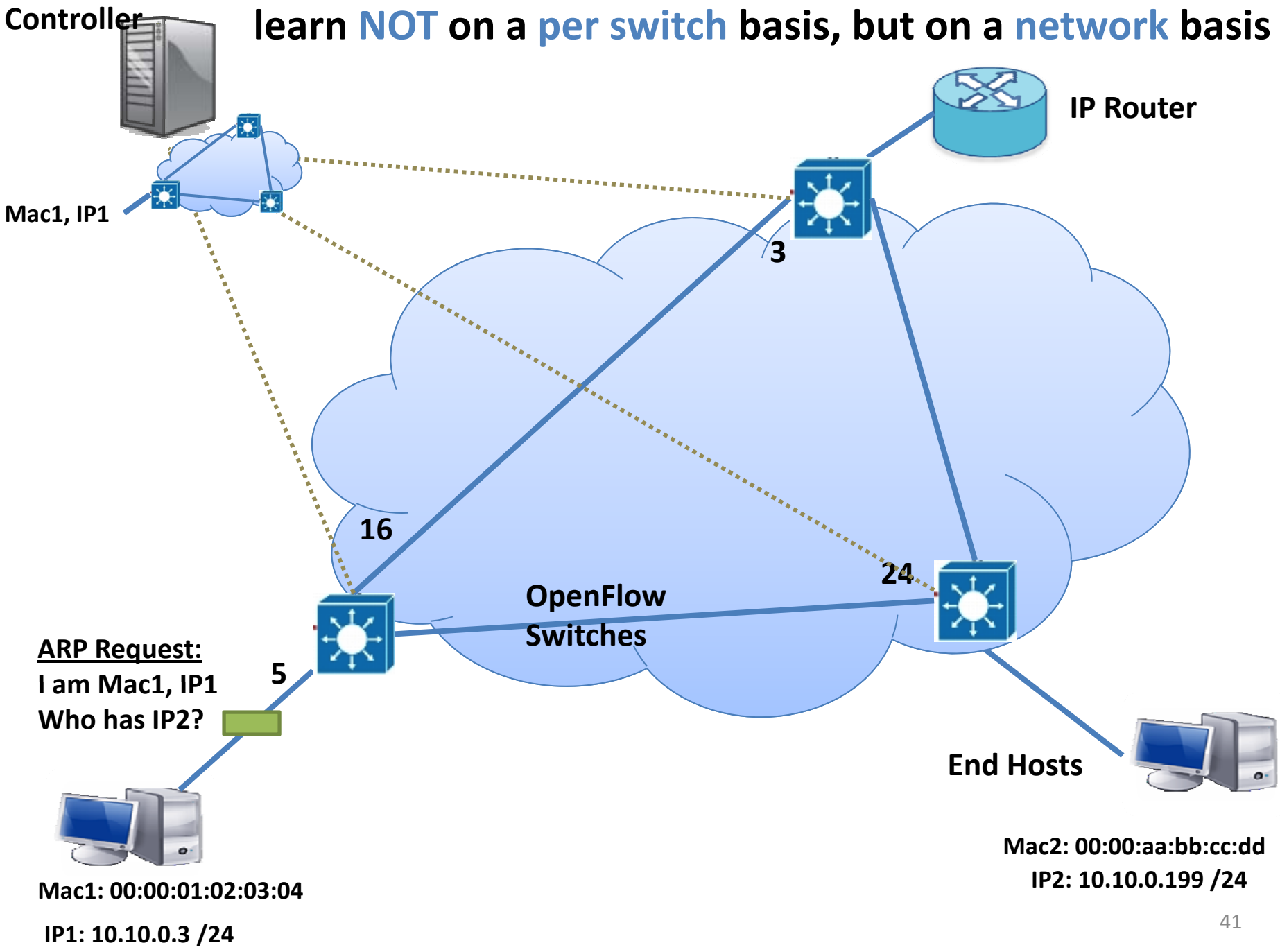


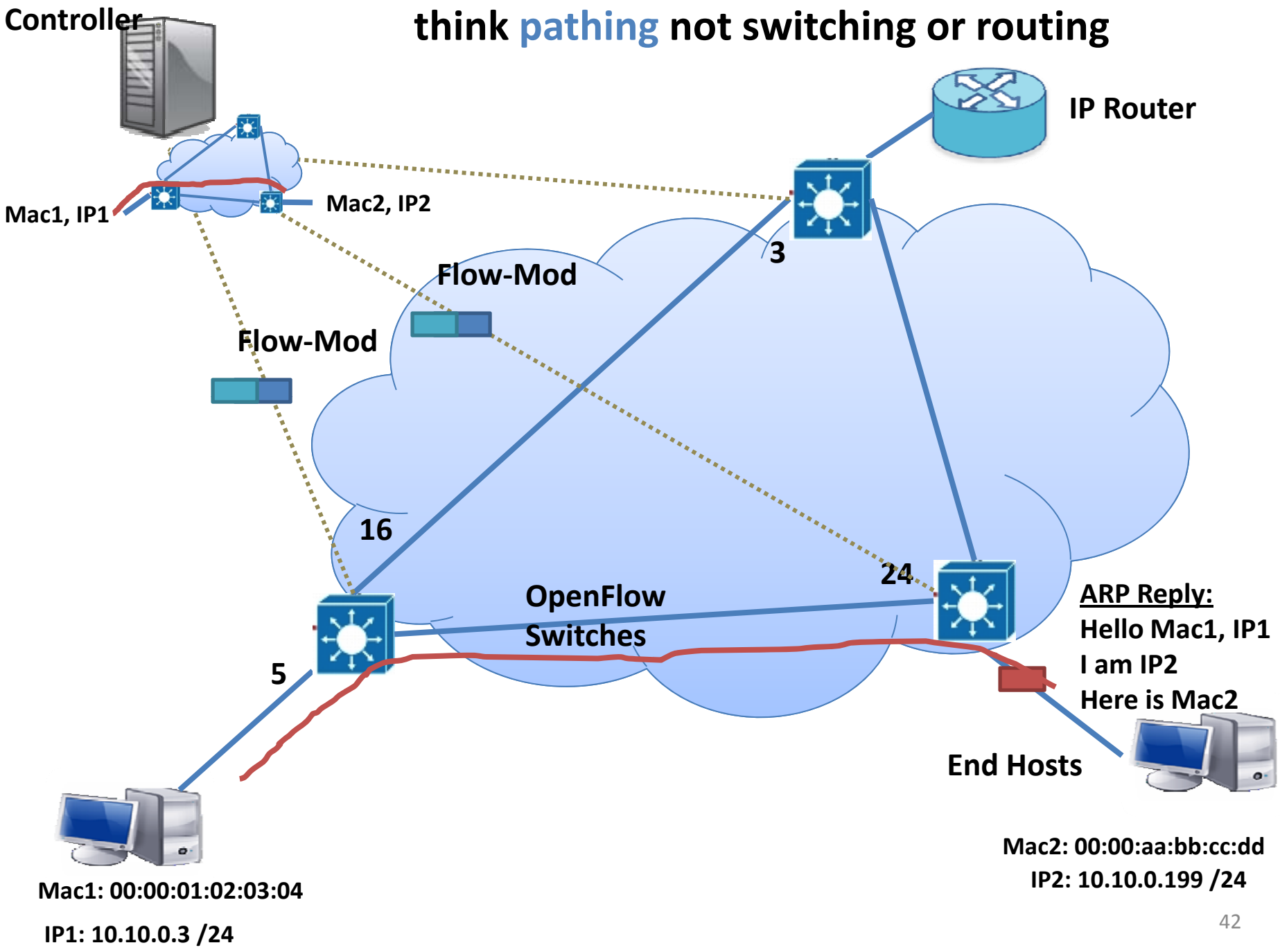
Topology Status

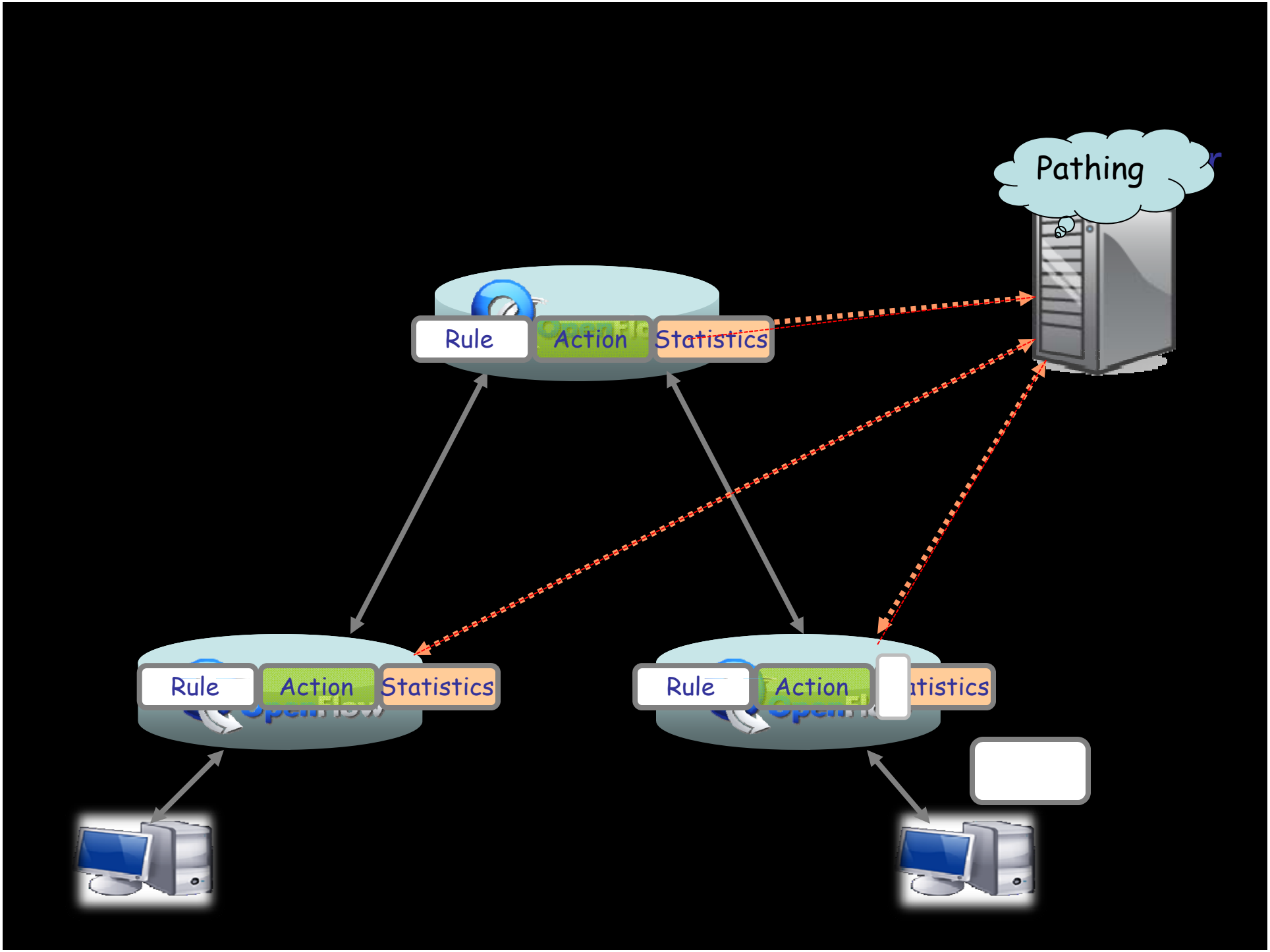


Topology Stats

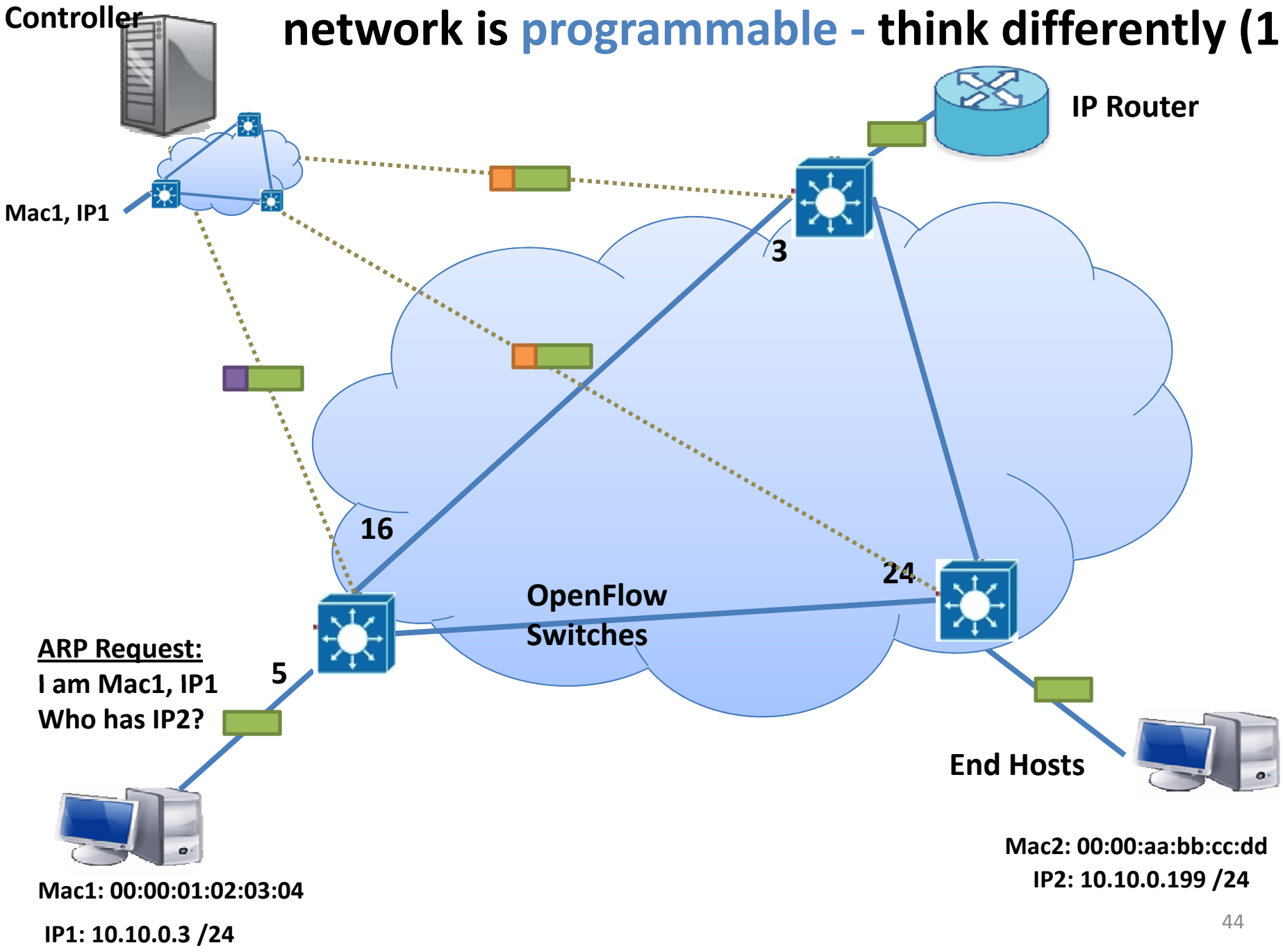



Controller
learn NOT on a per switch basis, but on a network basis
IP Router
Mac1, IP1
3
16
24
OpenFlow
Switches
ARP Request:
I am Mac1, IP1
Who has IP2?
5
End Hosts
Mac1: 00:00:01:02:03:04
IP1: 10.10.0.3 /24
Mac2: 00:00:aa:bb:cc:dd
IP2: 10.10.0.199 /24

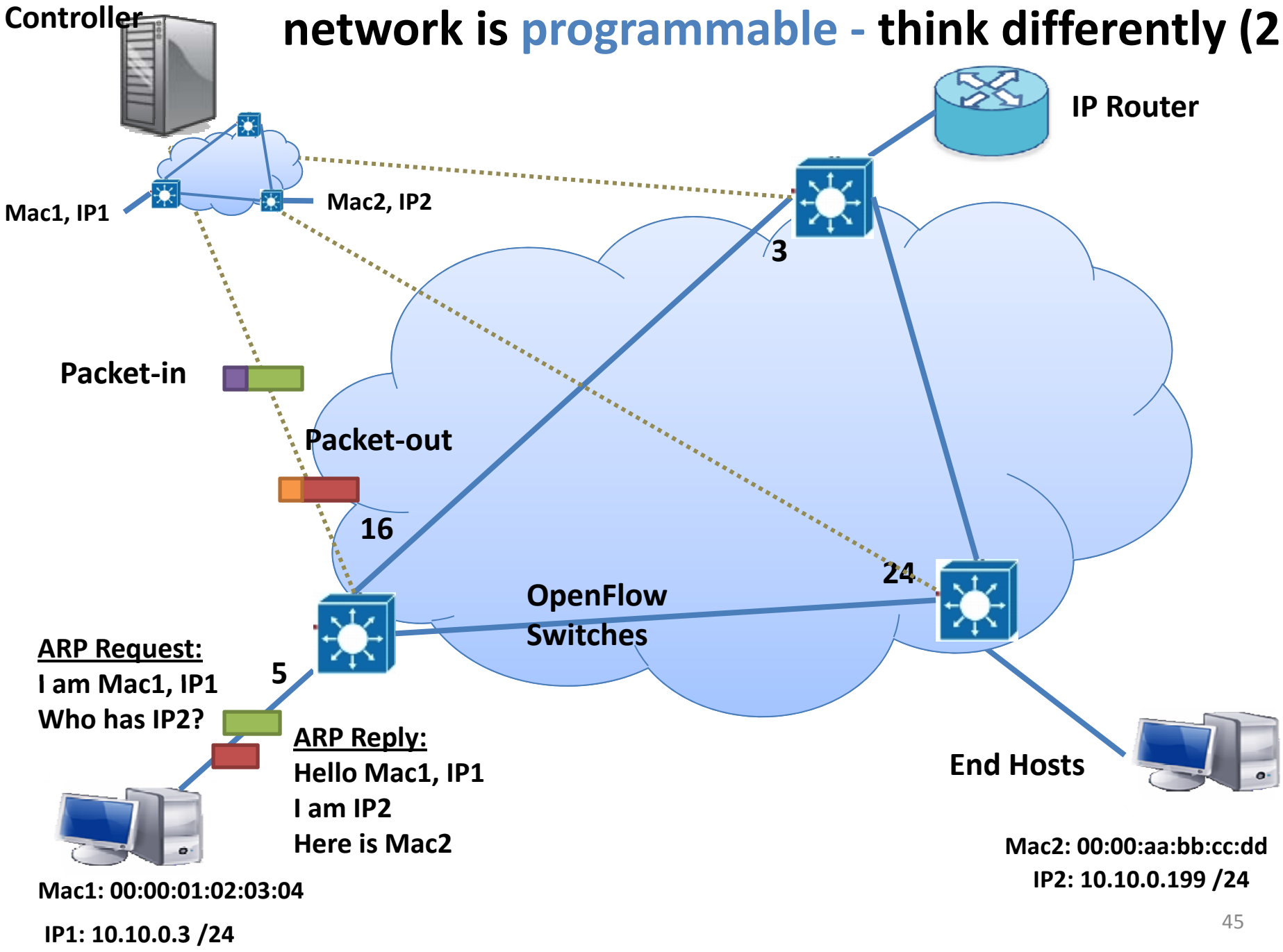

think pathing not switching or routing
Controller
IP Router
Mac1, IP1
Mac2, IP2
3
Flow-Mod
Flow-Mod
16
24
OpenFlow
Switches
5
ARP Reply:
Hello Mac1, IP1
I am IP2
Here is Mac2
End Hosts
Mac1: 00:00:01:02:03:04
IP1: 10.10.0.3 /24
Mac2: 00:00:aa:bb:cc:dd
IP2: 10.10.0.199 /24


Pathing
Rule
Action
Statistics
Rule
Action
Statistics
Rule
Action

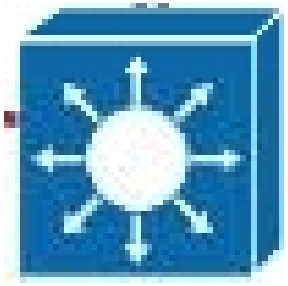
network is programmable - think differently (1)



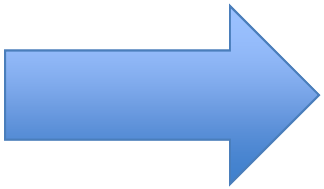
network is programmable - think differently (2)

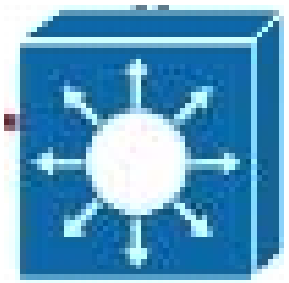


network is programmable – flexible match



Destination MAC	Out port
00:00:01:02:03:04	5
00:00:aa:bb:cc:dd	12





Match Rule	Action
DMAC=03:04	OP=5

Fixed
Behavior
L2 Table

Flexible,
Programmable
Flow Table

OpenFlow Flexible Match

Ethernet Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f	*	*	*	*	*	*	*	port6

IP 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

Application Firewall

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

OpenFlow Flexible Match

Exact Match

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

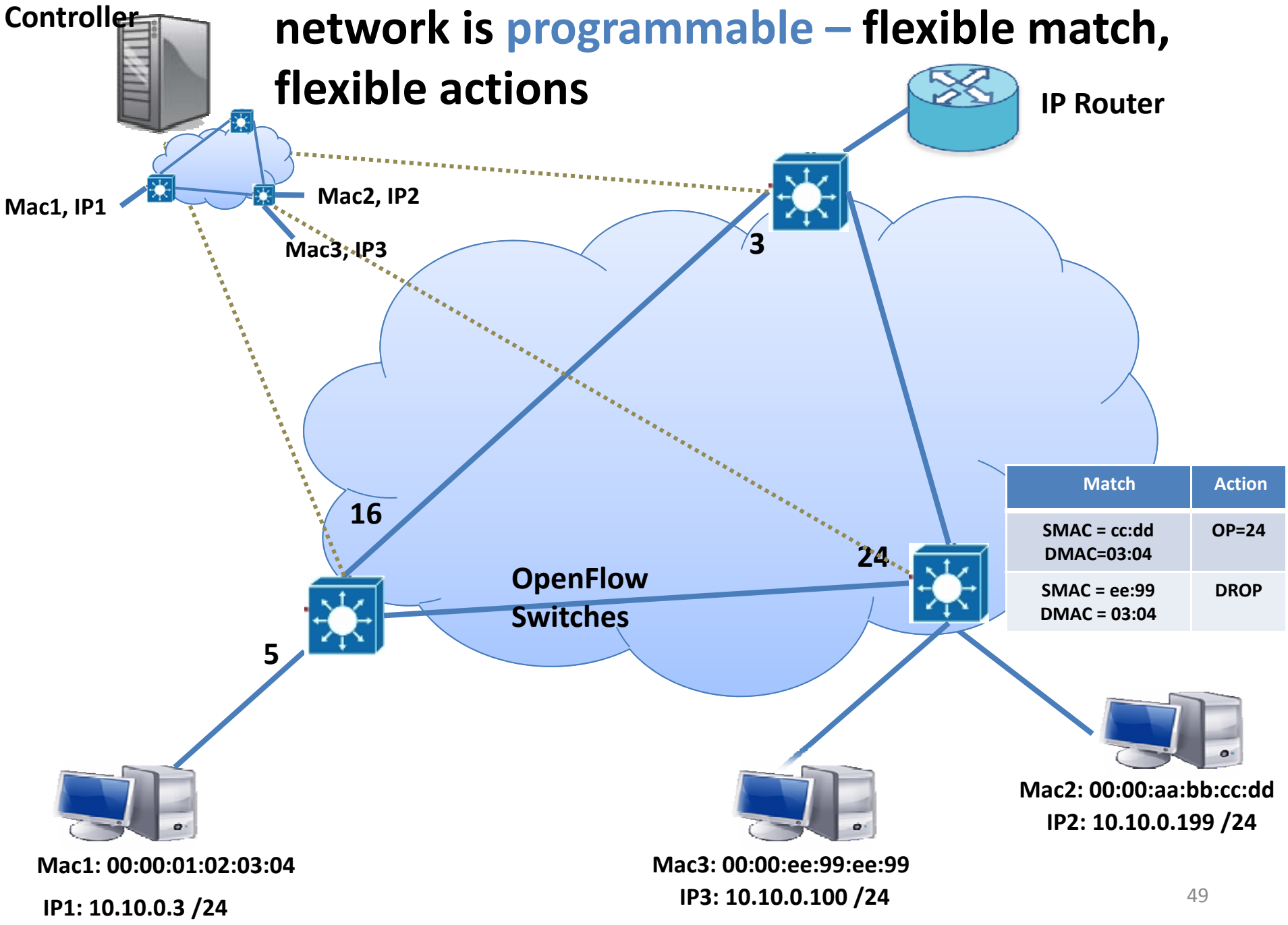
VLAN + App

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	vlan1	*	*	*	*	80	port6, port7

Port + Ethernet + IP

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:2e..	*	0800	*	*	5.6.7.8	4	*	*	port 10

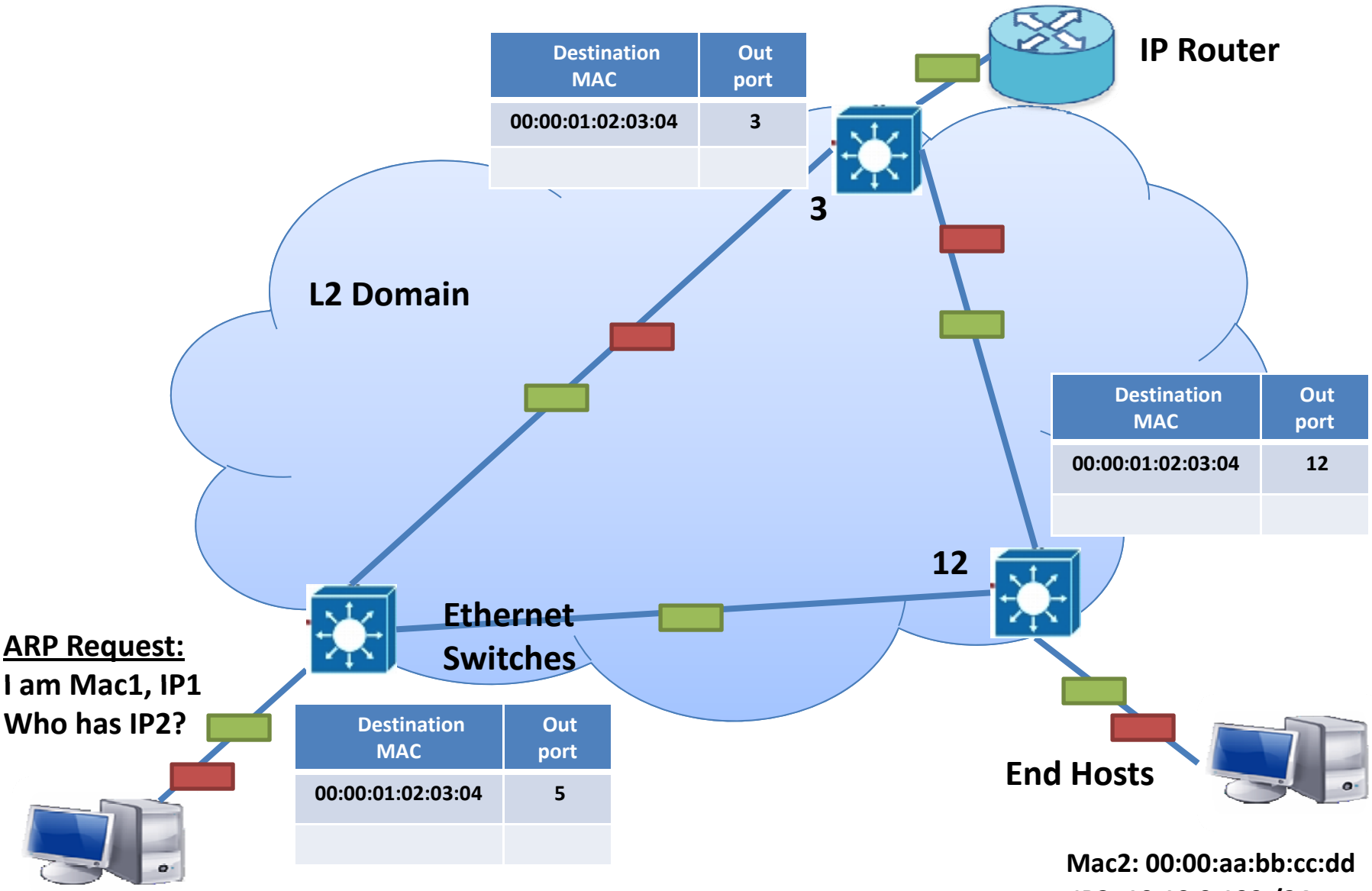
network is programmable – flexible match, flexible actions



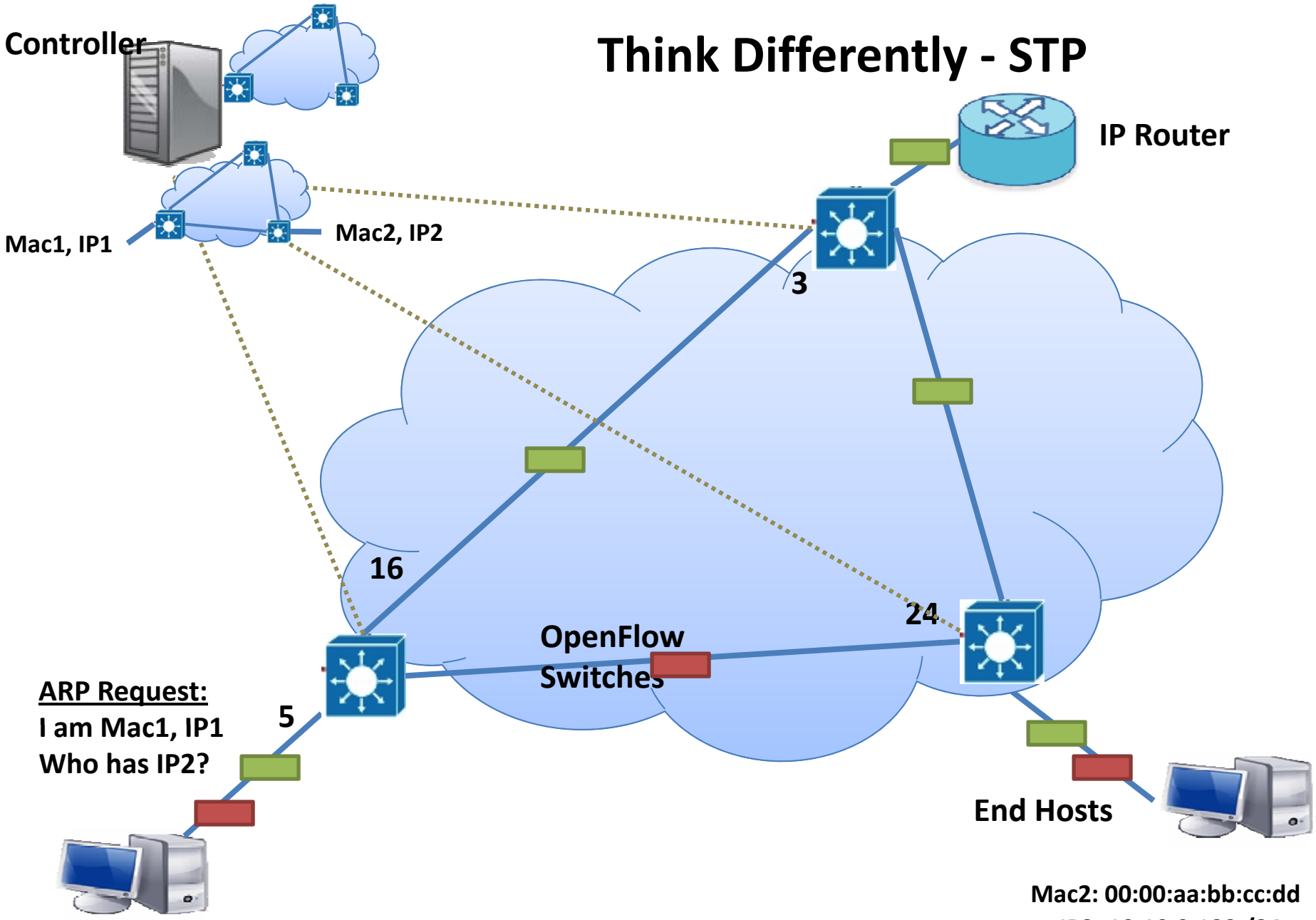
In an OpenFlow world:

- learn NOT on a per switch basis, but on a network basis
- think pathing not switching or routing
- network is programmable
    - think differently
    - flexible match
    - flexible actions

Think Differently - STP

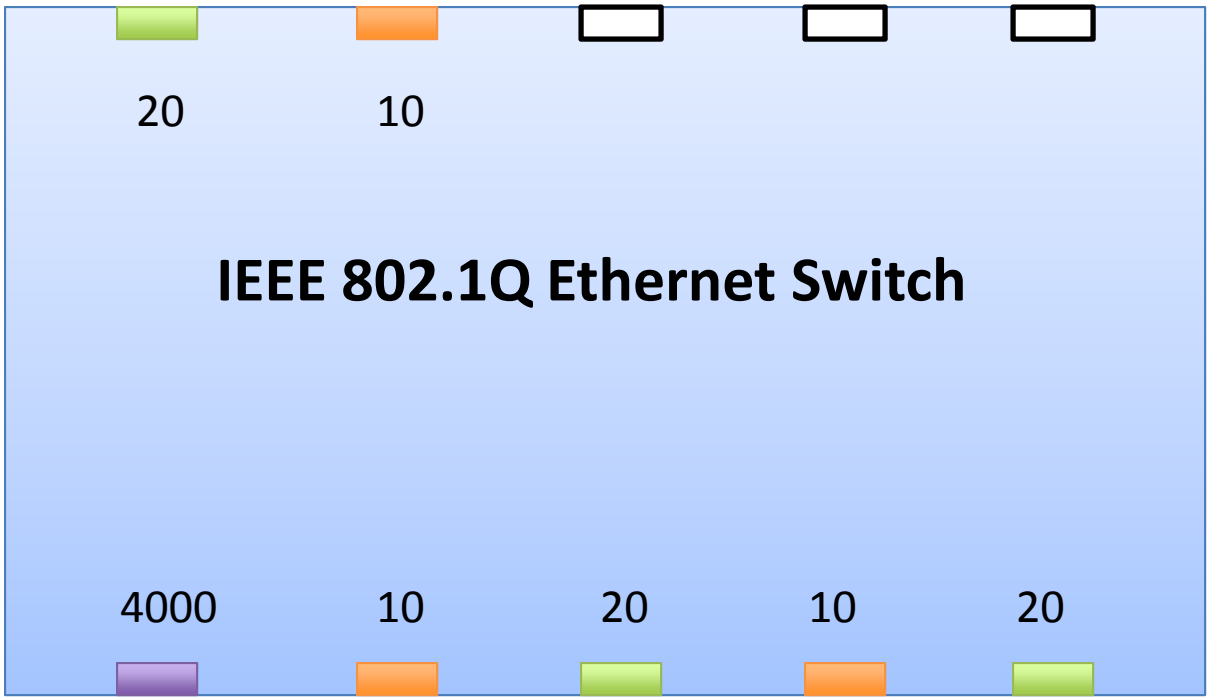


Think Differently - STP



Think Differently - VLANs

Trunk ports – multiple vlans
Connected to other switches



Access ports – single vlan assigned by switch
Connected to end-hosts

OpenFlow Switch != IEEE 802.1Q Ethernet Switch

Think Differently - VLANs

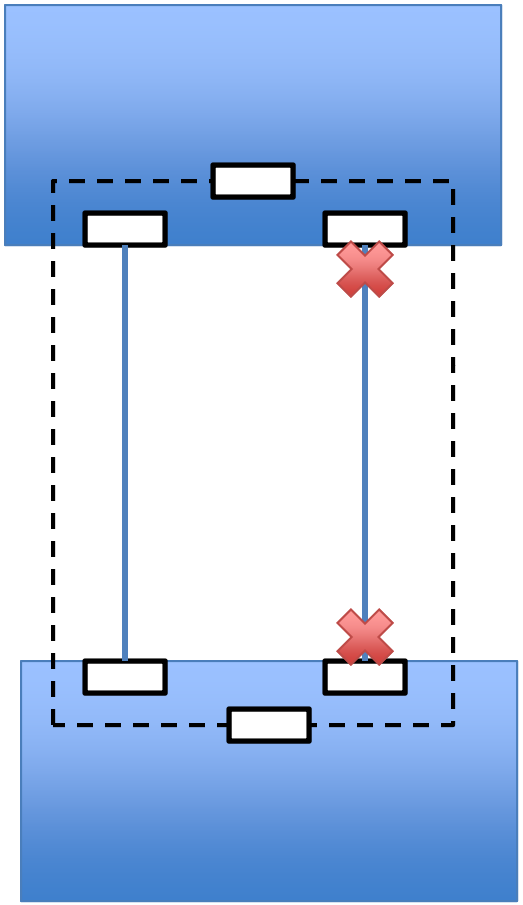
OpenFlow Switch != IEEE 802.1Q Ethernet Switch

OpenFlow treats VLANs as tags or labels – i.e without any switchport configuration

Flexible Match: match on VLAN id
match on VLAN priority
match on the absence of a VLAN
or don't care if there is a VLAN or not

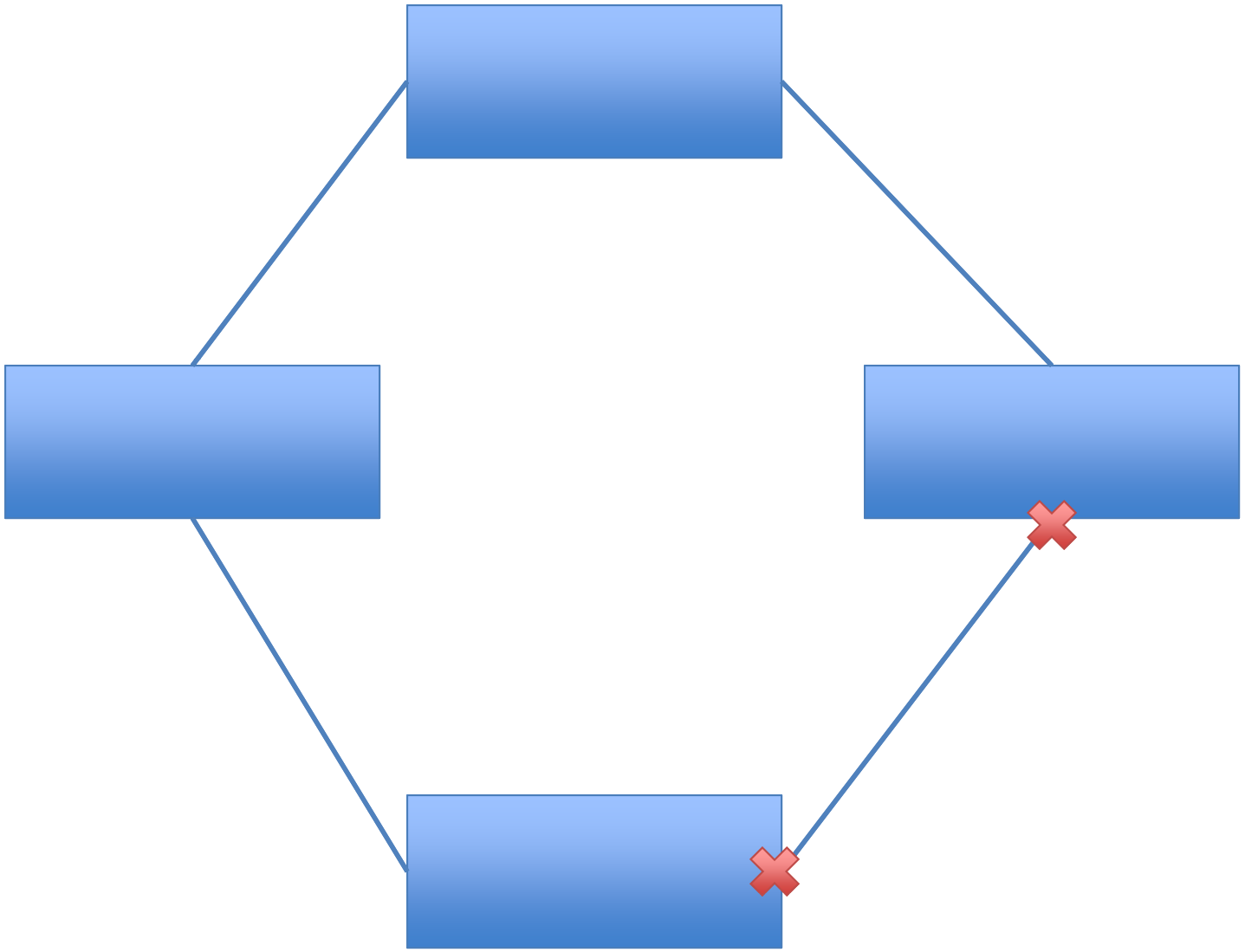
Flexible Action: push a VLAN tag
pop a VLAN tag
set the VLAN id
set the VLAN priority

Think Differently - LAGs



Wasted Bandwidth

Think Differently - MLAGs



“Protocol Soup”

- Current way to handle new functionality in networking is to define a new protocol
- Exponential growth in network protocol standards
- Standards seem to become larger and more complex
- Vendors implement all standards, which increases costs and decreases stability
- Do you need all those standards?

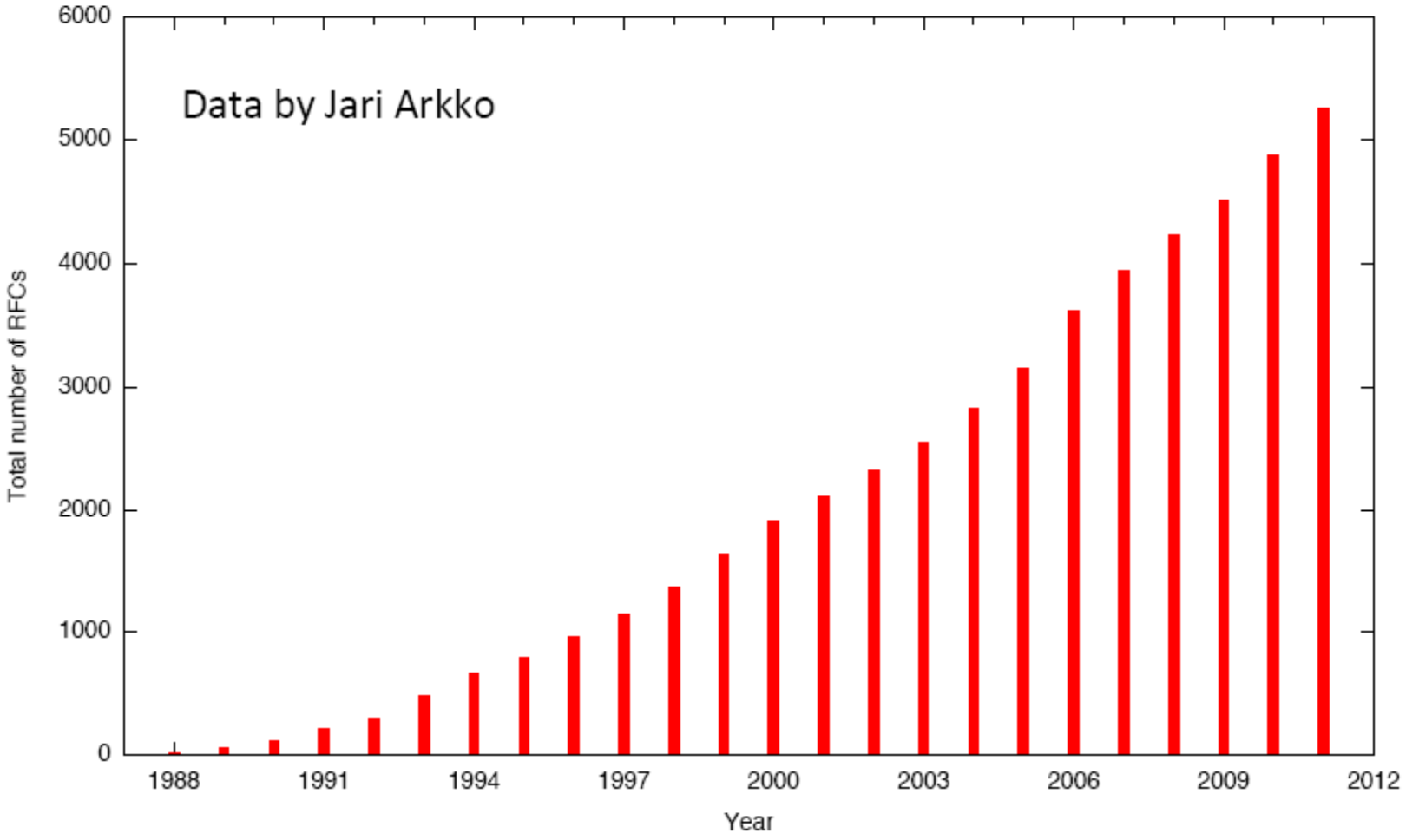
Source: Ronald van der Pol, rvdp@sara.nl

IEEE 802.1Q

- Simple VLAN standard?
- Not really, original version amended by at least 14 additional standards
- 802.1Q-1998 had 211 pages
- 802.1Q-2011 has 1365 pages, and includes:
  - 802.1u, 802.1v, 802.1s (multiple spanning trees), 802.1ad (provider bridging), 802.1ak (MRP, MVRP, MMRP), 802.1ag (CFM), 802.1ah (PBB), 802.1ap (VLAN bridges MIB), 802.1Qaw, 802.1Qay (PBB-TE), 802.1aj, 802.1Qav, 802.1Qau (congestion management), 802.1Qat (SRP)

Source: Ronald van der Pol, rvdp@sara.nl

Total Number of RFCs Published



Source: Ronald van der Pol, rvdp@sara.nl

Number of Supported Protocols in a Modern Ethernet Switch

(random example, but they are all the same)

(SLP and RSLP)
- IEEE 802.1w – 2001 Rapid Reconfiguration for STP, RSTP
- IEEE 802.1Q – 2003 (formerly IEEE 802.1s) Multiple Instances of STP, MSTP
- EMISTP, Extreme Multiple Instances of Spanning Tree Protocol
- PVST+, Per VLAN STP (802.1Q Interoperable)
- Draft-ietf-bridge-rstpmib-03.txt – Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol
- Extreme Standby Router Protocol™ (ESRP)
- IEEE 802.1Q – 1998 Virtual Bridged Local Area Networks
- IEEE 802.3ad Static load sharing configuration and LACP based dynamic configuration
- Software Redundant Ports
- IEEE 802.1AB – LLDP Link Layer Discovery Protocol
- LLDP Media Endpoint Discovery (LLDP-MED), ANSI/TIA-1057, draft 08
- Extreme Discovery Protocol (EDP)
- Extreme Loop Recovery Protocol (ELRP)
- Extreme Link State Monitoring (ELSM)
- IEEE 802.1ag L2 Ping and traceroute, Connectivity Fault Management
- ITU-T Y.1731 Frame delay measurements

Management and Traffic Analysis

- RFC 2030 SNTP, Simple Network Time Protocol v4
- RFC 854 Telnet client and server
- RFC 783 TFTP Protocol (revision 2)
- RFC 951, 1542 BootP
- RFC 2131 BOOTP/DHCP relay agent and DHCP server
- RFC 1591 DNS (client operation)
- RFC 1155 Structure of Management Information (SMIv1)
- RFC 1157 SNMPv1
- RFC 1212, RFC 1213, RFC 1215 MIB-II, Ethernet-Like MIB & TRAPs
- RFC 1573 Evolution of Interface
- RFC 1650 Ethernet-Like MIB (update of RFC 1213 for SNMPv2)
- RFC 1901, 1905 – 1908 SNMPv2c, SMIv2 and Revised MIB-II
- RFC 2576 Coexistence between SNMP Version 1, Version 2 and Version 3
- RFC 2578 – 2580 SMIv2 (update to RFC 1902 – 1903)
- RFC 3410 – 3415 SNMPv3, user based security, encryption and authentication
- RFC 3826 – The Advanced Encryption

- IEEE 802.1ag MIB
- Secure Shell (SSH-2) client and server
- Secure Copy (SCP-2) client and server
- Secure FTP (SFTP) server
- sFlow version 5
- Configuration logging
- Multiple Images, Multiple Configs
- RFC 3164 BSD Syslog Protocol with Multiple Syslog Servers
  - 999 Local Messages (critically stored across reboots)
- Extreme Networks vendor MIBs (includes FDB, PoE, CPU, Memory MIBs)
- XML APIs over Telnet/SSH and HTTP/HTTPS
- Web based device management interface – ExtremeXOS ScreenPlay™
- IP Route Compression

Security, Switch and Network Protection

- Secure Shell (SSH-2), Secure Copy (SCP-2) and SFTP client/server with encryption/ authentication (requires export controlled encryption module)
- SNMPv3 user based security, with encryption/authentication (see above)
- RFC 1492 TACACS+
- RFC 2138 RADIUS Authentication
- RFC 2139 RADIUS Accounting
- RFC 3579 RADIUS EAP support for 802.1x
- RADIUS Per-command Authentication
- Access Profiles on All Routing Protocols
- Access Policies for Telnet/SSH-2/SCP-2
- Network Login – 802.1x, Web and MAC-based mechanisms
- IEEE 802.1x – 2001 Port-Based Network Access Control for Network Login
- Multiple supplicants with multiple VLANs for Network Login (all modes)
- Fallback to local authentication database (MAC and Web-based methods)
- Guest VLAN for 802.1x
- RFC 1866 HTML – Used for Web-based Network Login and ExtremeXOS ScreenPlay
- SSL/TLS transport – used for Web-based Network Login and ExtremeXOS ScreenPlay (requires export controlled encryption module)
- MAC Security – Lockdown and Limit
- IP Security – RFC 3046 DHCP Option 82 with port and VLAN ID
- IP Security – Trusted DHCP Server
- Layer 2/3/4 Access Control Lists (ACLs)
- RFC 2267 Network Ingress Filtering
- RPF (Unicast Reverse Path Forwarding)

- CA-97.28:Teardrop_Land-Teardrop and "LAND" attack
- CA-96.26: ping
- CA-96.21: tcp_syn_flooding
- CA-96.01: UDP_service_denial
- CA-95.01: IP_Spoofing_Attacks_and_Hijacked_Terminal_Connections
- IP Options Attack
- Host Attack Protection
  - Teardrop, bonk, opentear, jolt2, newtear, nestea, syndrop, smurf, fraggle, papasmurf, synk4, raped, winfreeze, ping –f, ping of death, pepsi5, Latierra, Winnuke, Simping, Sping, Ascend, Stream, Land, Octopus

Security, Router Protection

- IP Security – DHCP enforcement via Disable ARP Learning
- IP Security – Gratuitous ARP Protection
- IP Security – DHCP Secured ARP/ARP Validation
- Routing protocol MD5 authentication

Security Detection and Protection

- CLEAR-Flow, threshold based alerts and actions

IPv4 Host Services

- RFC 1122 Host Requirements
- RFC 768 UDP
- RFC 791 IP
- RFC 792 ICMP
- RFC 793 TCP
- RFC 826 ARP
- RFC 894 IP over Ethernet
- RFC 1027 Proxy ARP
- RFC 2068 HTTP server
- IGMP v1/v2/v3 Snooping with Configurable Router Registration Forwarding
- IGMP Filters
- PIM Snooping
- Static IGMP Membership
- Multicast VLAN Registration (MVR)

IPv4 Router Services

- RFC 1812 Requirements for IP Version 4 Routers
- RFC 1519 CIDR
- RFC 1256 IPv4 ICMP Router Discovery (IRDP)
- Static Unicast Routes
- Static Multicast Routes
- RFC 1058 RIP v1
- RFC 2453 RIP v2
- Static ECMP
- RFC 1112 IGMP v1

- RFC 1587 OSPF NSSA Option
- RFC 1765 OSPF Database Overflow
- RFC 2370 OSPF Opaque LSA Option
- RFC 3623 OSPF Graceful Restart
- RFC 1850 OSPFv2 MIB
- RFC 2362 PIM-SM (Edge-mode)
- RFC 2934 PIM MIB
- RFC 3569, draft-ietf-ssm-arch-06.txt PIM-SSM PIM Source Specific Multicast
- draft-ietf-pim-mib-v2-01.txt
- Mtrace, a "traceroute" facility for IP Multicast: draft-ietf-idmr-traceroute-ipm-07
- Mrinfo, the multicast router information tool based on Appendix-B of draft-ietf-idmr-dvmrp-v3-11

IPv6 Host Services

- RFC 3587, Global Unicast Address Format
- Ping over IPv6 transport
- Traceroute over IPv6 transport
- RFC 5095, Internet Protocol, Version 6 (IPv6) Specification
- RFC 4861, Neighbor Discovery for IP Version 6, (IPv6)
- RFC 2463, Internet Control Message Protocol (ICMPv6) for the IPv6 Specification
- RFC 2464, Transmission of IPv6 Packets over Ethernet Networks
- RFC 2465, IPv6 MIB, General Group and Textual Conventions
- RFC 2466, MIB for ICMPv6
- RFC 2462, IPv6 Stateless Address Auto Configuration – Host Requirements
- RFC 1981, Path MTU Discovery for IPv6, August 1996 – Host Requirements
- RFC 3513, Internet Protocol Version 6 (IPv6) Addressing Architecture
- Telnet server over IPv6 transport
- SSH-2 server over IPv6 transport

IPv6 Interworking and Migration

- RFC 2893, Configured Tunnels
- RFC 3056, 6to4

IPv6 Router Services

- RFC 2462, IPv6 Stateless Address Auto Configuration – Router Requirements
- RFC 1981, Path MTU Discovery for IPv6, August 1996 – Router Requirements
- RFC 2710, IPv6 Multicast Listener Discovery v1 (MLDv1) Protocol
- Static Unicast routes for IPv6
- RFC 2080, RIPng

- RFC 1771 Border Gateway Protocol 4
- RFC 1965 Autonomous System Confederations for BGP
- RFC 2796 BGP Route Reflection (supersedes RFC 1966)
- RFC 1997 BGP Communities Attribute
- RFC 1745 BGP4/IDRP for IP-OSPF Interaction
- RFC 2385 TCP MD5 Authentication for BGPv4
- RFC 2439 BGP Route Flap Damping
- RFC 2918 Route Refresh Capability for BGP-4
- RFC 3392 Capabilities Advertisement with BGP-4
- RFC 4360 BGP Extended Communities Attribute
- RFC 4486 Subcodes for BGP Cease Notification message
- draft-ietf-idr-restart-10.txt Graceful Restart Mechanism for BGP
- RFC 4760 Multiprotocol extensions for BGP-4
- RFC 1657 BGP-4 MIB
- RFC 4893 BGP Support for Four-Octet AS Number Space
- Draft-ietf-idr-bgp4-mibv2-02.txt – Enhanced BGP-4 MIB
- RFC 1195 Use of OSI IS-IS for Routing in TCP/IP and Dual Environments (TCP/IP transport only)
- RFC 2763 Dynamic Hostname Exchange Mechanism for IS-IS
- RFC 2966 Domain-wide Prefix Distribution with Two-Level IS-IS
- RFC 2973 IS-IS Mesh Groups
- RFC 3373 Three-way Handshake for IS-IS Point-to-Point Adjacencies
- Draft-ietf-isis-restart-02 Restart Signaling for IS-IS
- Draft-ietf-isis-ipv6-06 Routing IPv6 with IS-IS
- Draft-ietf-isis-wg-multi-topology-11 Multi Topology (MT) Routing in IS-IS

QoS and VLAN Services

Quality of Service and Policies
- IEEE 802.1D – 1998 (802.1p) Packet Priority
- RFC 2474 DiffServ Precedence, including 8 queues/port
- RFC 2598 DiffServ Expedited Forwarding (EF)
- RFC 2597 DiffServ Assured Forwarding (AF)
- RFC 2475 DiffServ Core and Edge Router Functions

Traffic Engineering
- RFC 3784 IS-IS Extens for Traffic Engineering (wide metrics only)

VLAN Services: VLANs, vMANs
- IEEE 802.1Q VLAN Tagging
- IEEE 802.1v: VLAN classification by Protocol and Port

Advanced VLAN Services, MAC-in-MAC
- VLAN Translation in vMAN environments
- vMAN Translation
- IEEE 802.1ah/D1.2 Provider Backbone Bridges (PBB)/MAC-in-MAC

MPLS and VPN Services

Multi-Protocol Label Switching (MPLS)
Requires MPLS Layer 2 Feature Pack License
- RFC 2961 RSVP Refresh Overhead Reduction Extensions
- RFC 3031 Multiprotocol Label Switching Architecture
- RFC 3032 MPLS Label Stack Encoding
- RFC 3036 Label Distribution Protocol (LDP)
- RFC 3209 RSVP-TE: Extensions to RSVP for LSP Tunnels
- RFC 3630 Traffic Engineering Extensions to OSPFv2
- RFC 3784 IS-IS extensions for traffic engineering only (wide metrics only)
- RFC 3811 Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management
- RFC 3812 Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)
- RFC 3813 Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)
- RFC 3815 Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)
- RFC 4090 Fast Re-route Extensions to RSVP-TE for LSP (Detour Paths)
- RFC 4379 Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (LSP Ping)
- draft-ietf-bfd-base-09.txt Bidirectional Forwarding Detection

Layer 2 VPNs
Requires MPLS Layer 2 Feature Pack License
- RFC 4447 Pseudowire Setup and Maintenance using the Label Distribution Protocol (LDP)
- RFC 4448 Encapsulation Methods for Transport of Ethernet over MPLS Networks
- RFC 4762 Virtual Private LAN Services (VPLS) using Label Distribution Protocol (LDP) Signaling
- RFC 5085 Pseudowire Virtual Circuit Connectivity Verification (VCCV)
- RFC 5542 Definitions of Textual Conventions for Pseudowire (PW) Management
- RFC 5601 Pseudowire (PW) Management

Source: Ronald van der Pol, rvdp@sara.nl

What have we learnt so far?

- Networks can be programmable
  - not just configurable
  - or pre-defined
- Need to think differently though..
  - not all we had is useful or necessary anymore
  - paradigm change
- OpenFlow primitives
  - flexible match
  - flexible actions
  - messages: packet-in/out, flow-mod, stats, status

Demo Video

Using all wireless networks around us

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

Outline

- Module 1: SDN/OpenFlow Introduction
- Module 2: OpenFlow/SDN Deep Dive
- Module 3: SDN Architecture
- Module 4: Operational Choices & Performance
- Module 5: Benefits & Use Cases
  - ❖ Demo: Network Virtualization
- Module 6: Demystifying SDN
- Q & A

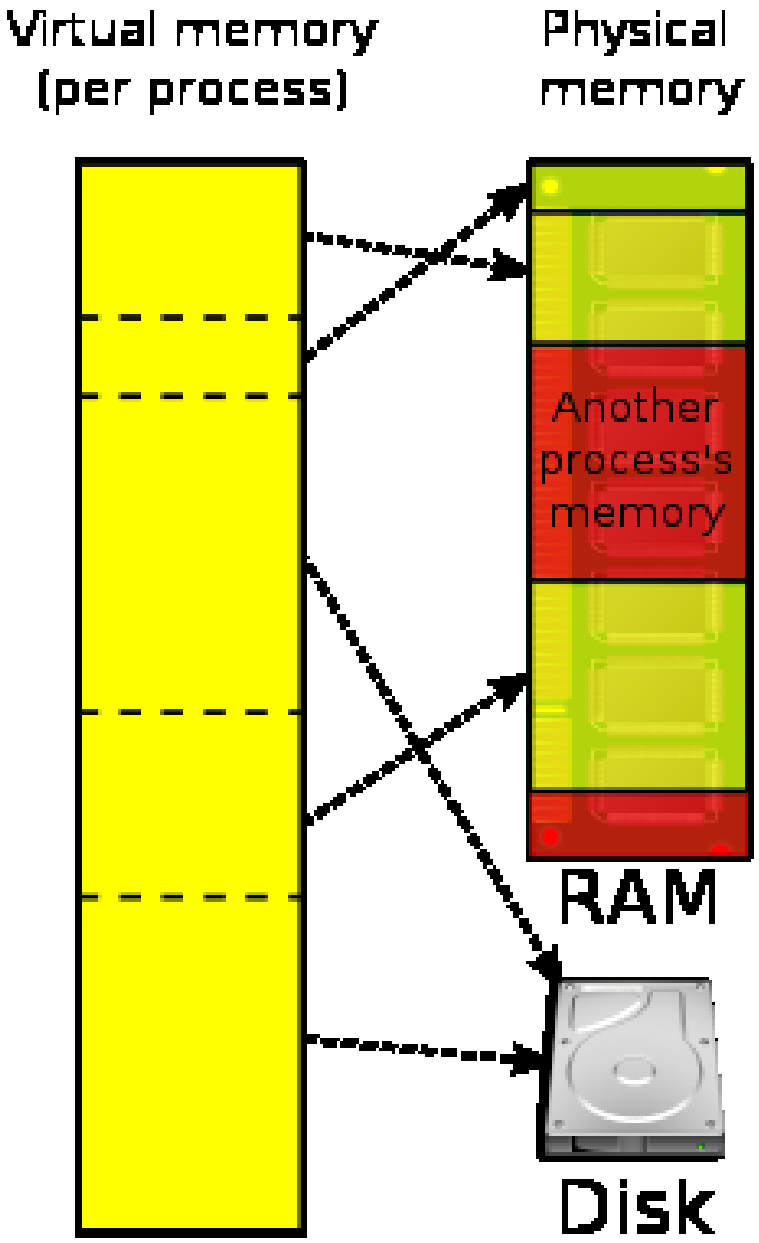
Module 3

SDN Architecture

Flow Abstraction & Switch-API

Map Abstraction & Network-API

What is an Abstraction?



What is an Abstraction?

- Machine Languages: no abstractions
- Higher-level languages, OS + other abstraction
  - files, virtual memory, data structures, …
- Modern languages: even more abstractions
  - objects, garbage collection, threads, locks, …

Abstractions simplify programming: they make it easier to write, maintain, and reason about programs.

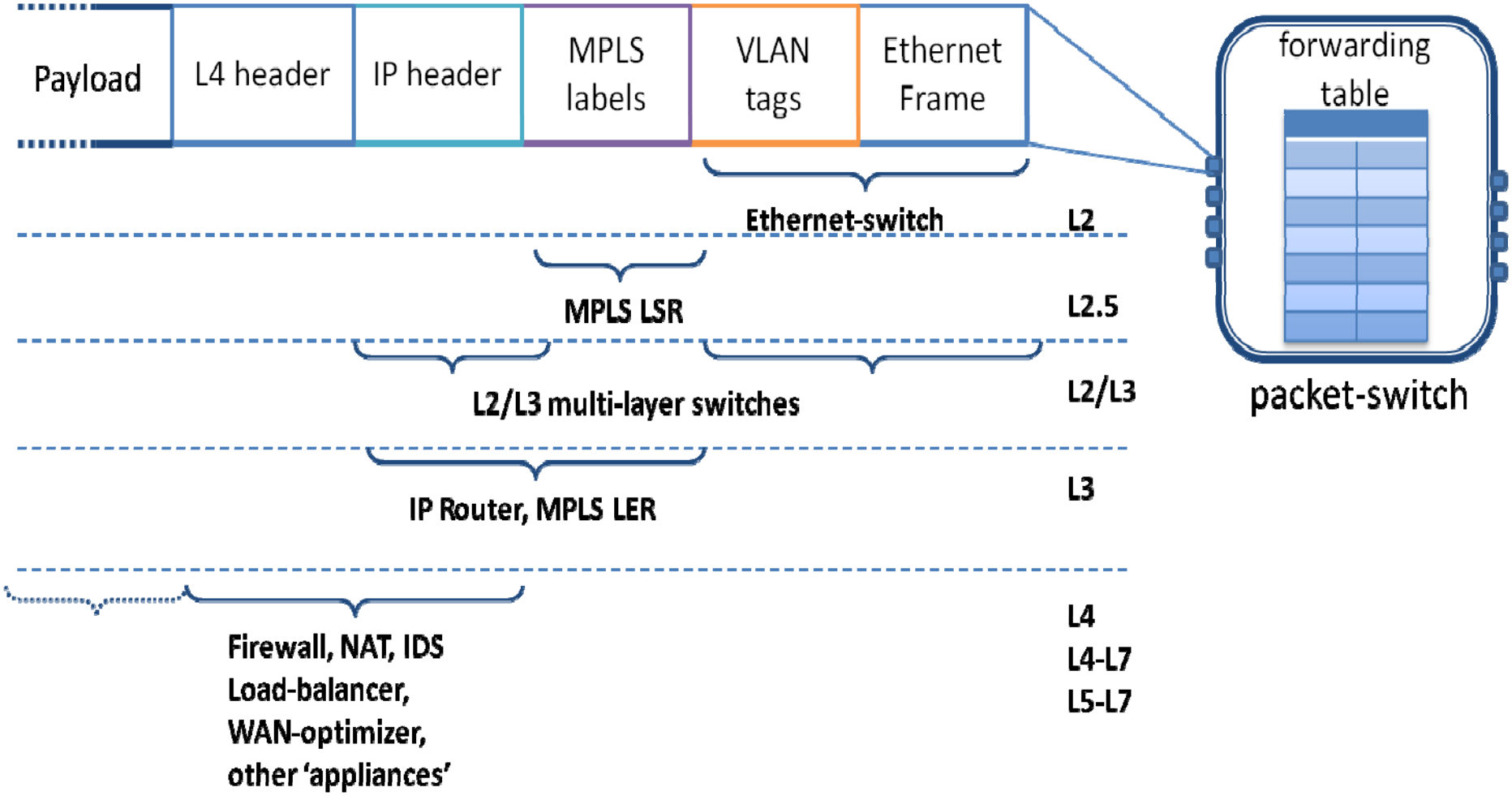
Could networking follow this same path?

Abstraction?

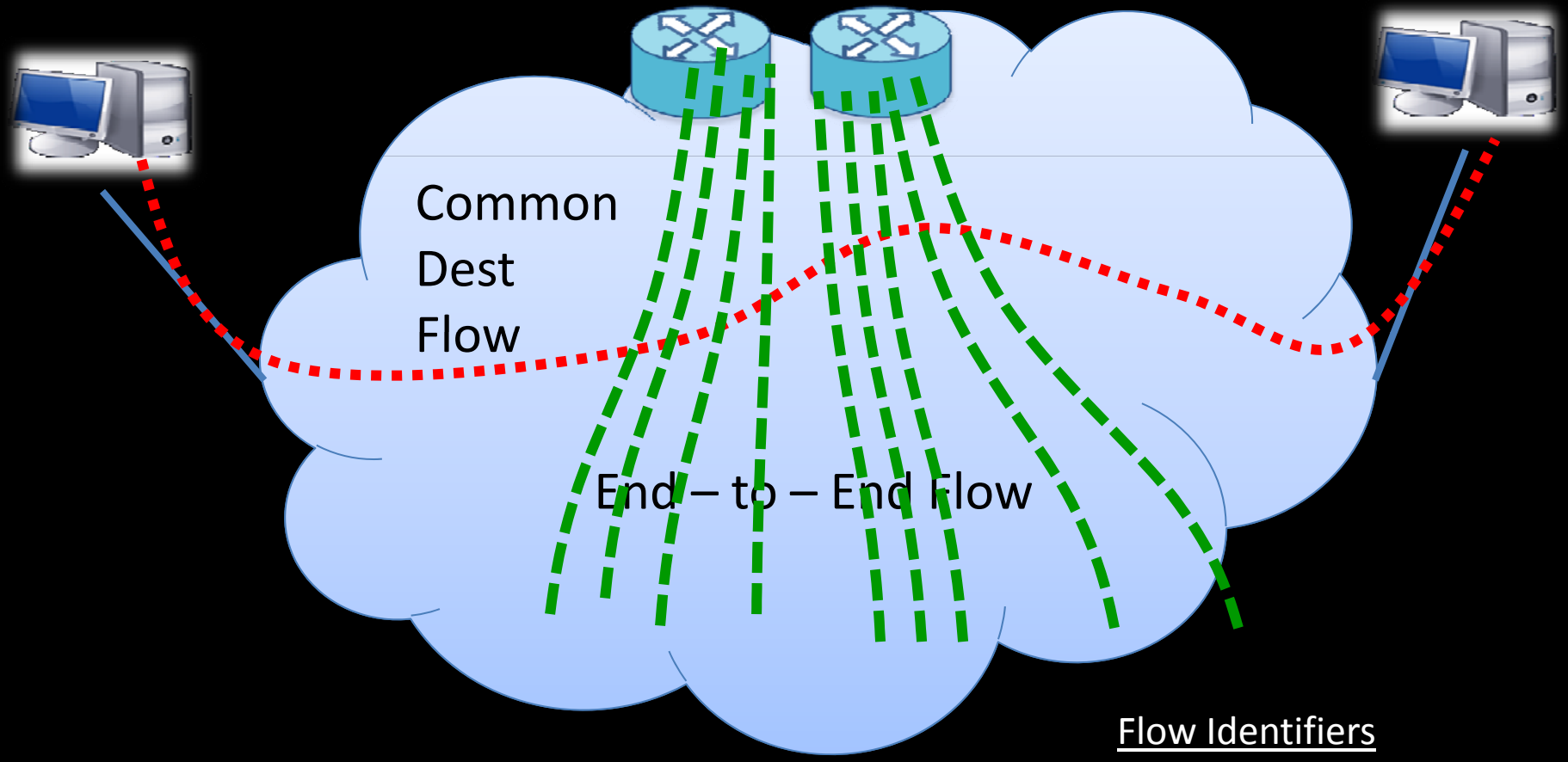
Data – plane abstractions

Where are the control-plane abstractions?

Flow Abstraction and the Switch- API



1. The Flow Abstraction



Flow Identifiers

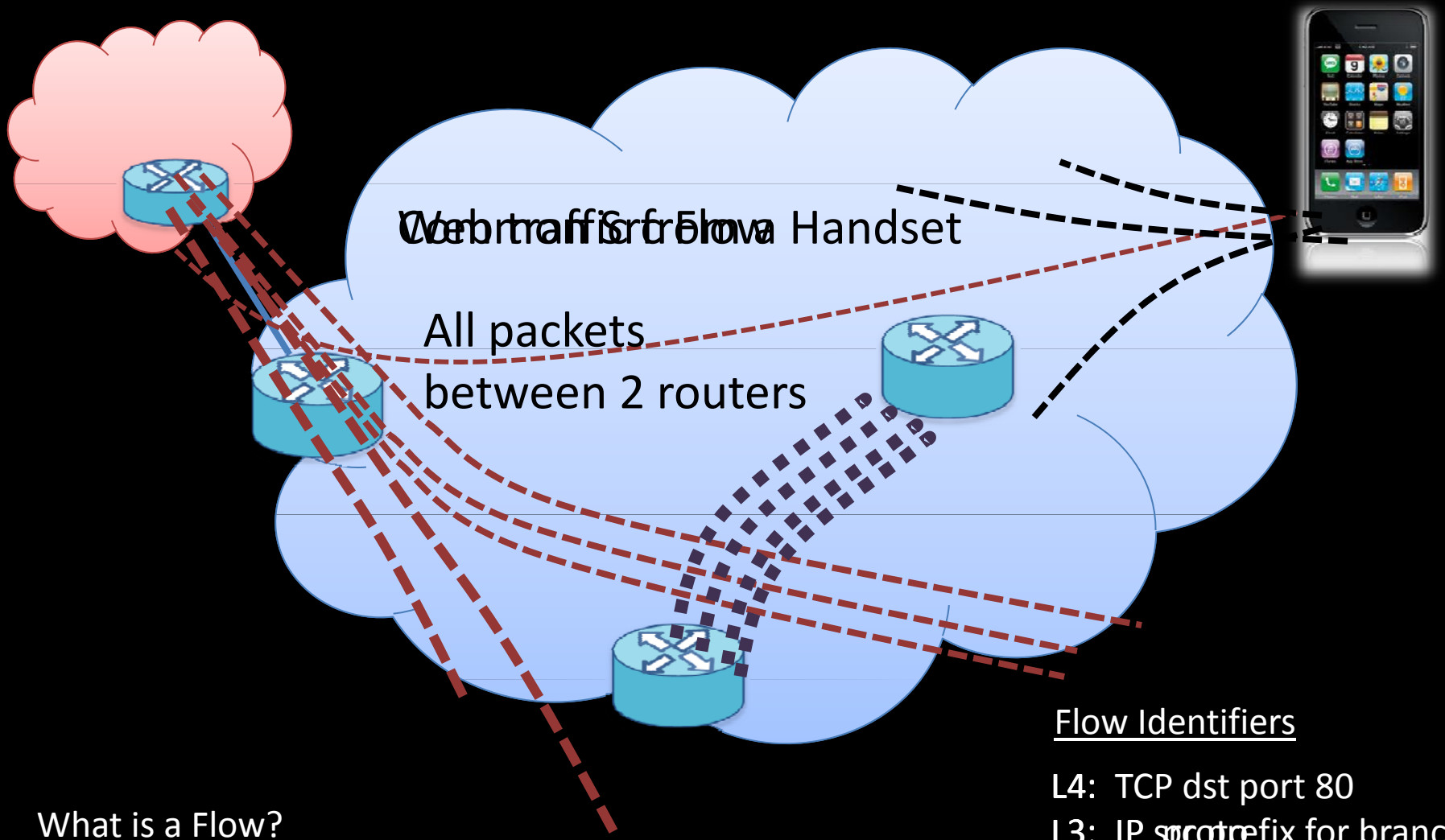
L4: TCP src/dst port

L3: IP src/dst address, IP proto

L2.5:

L2:

1. The Flow Abstraction



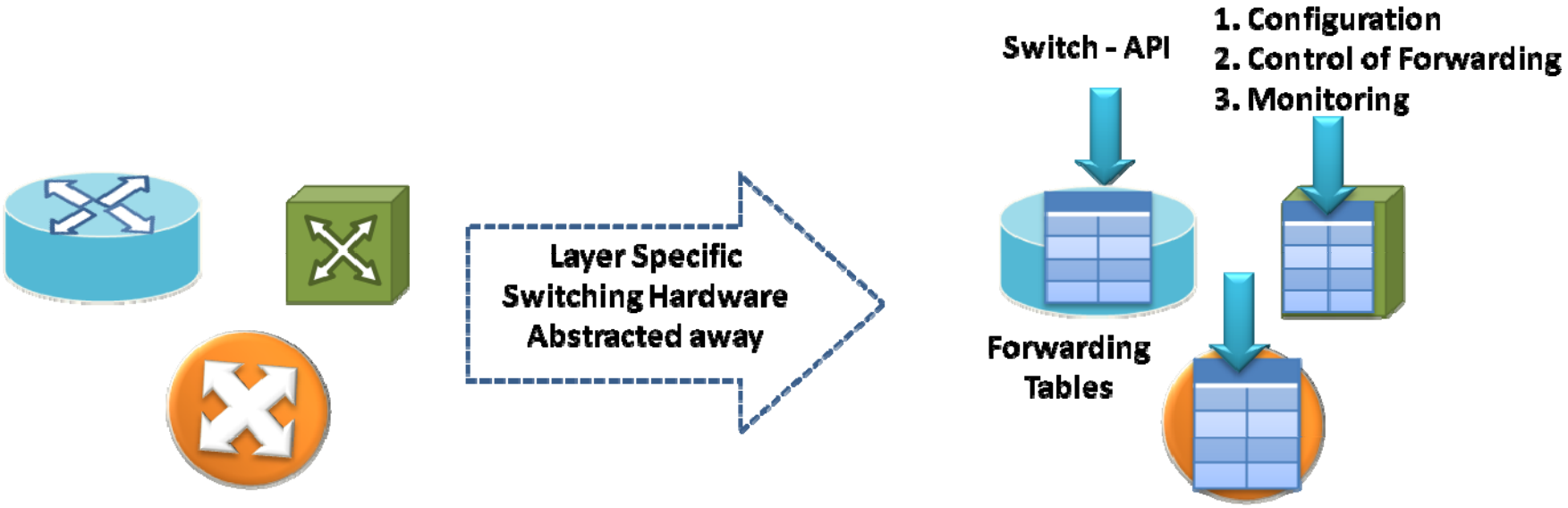
What is a Flow?

- Classification of packets that have a logical association
- Action & Maintaining Flow State
- Flow based Accounting & Resource Management

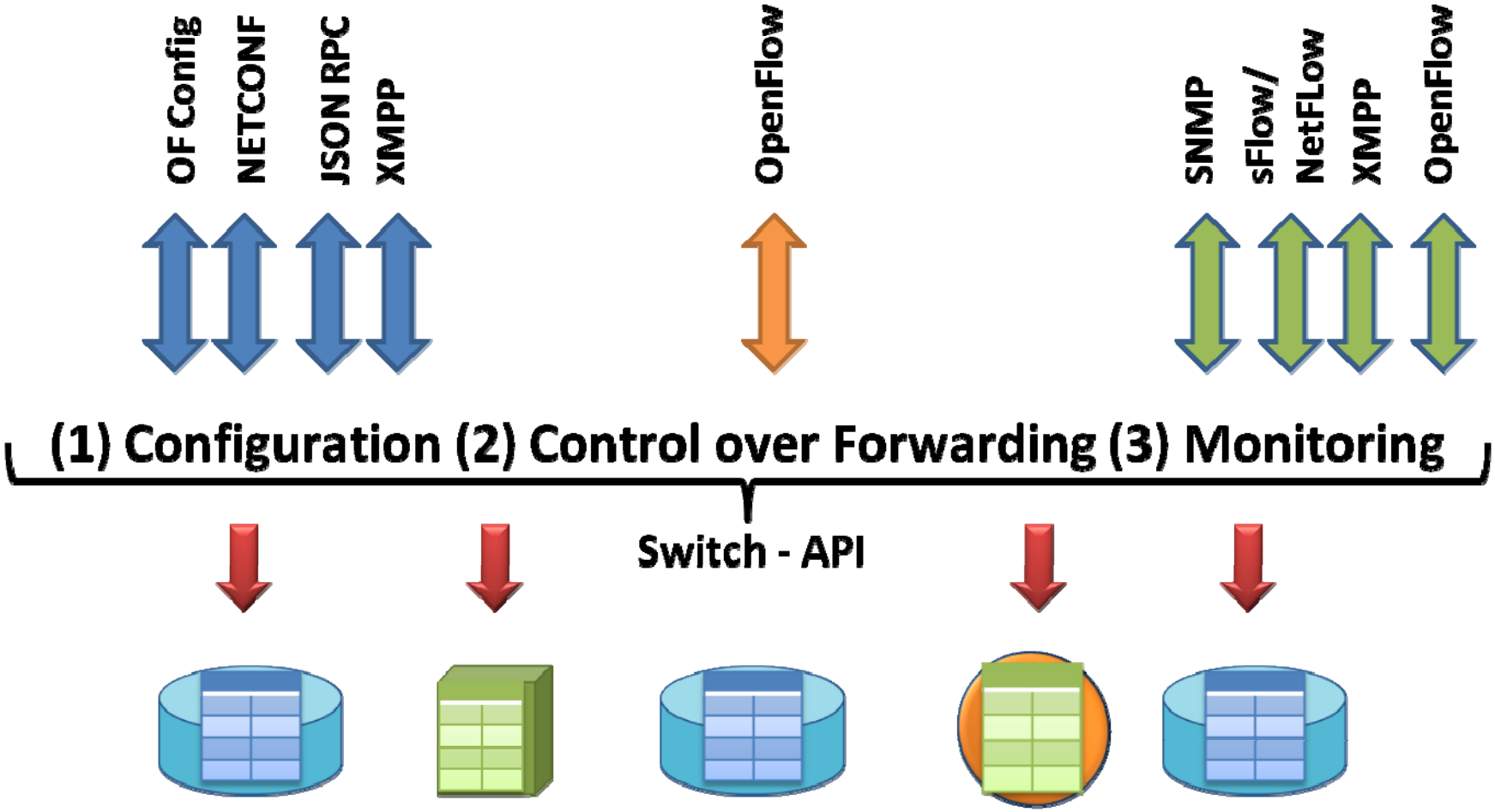
Flow Identifiers

L4: TCP dst port 80
L3: IP src prefix for branch
L2.5: MPLS Label ID
L2: MAC src

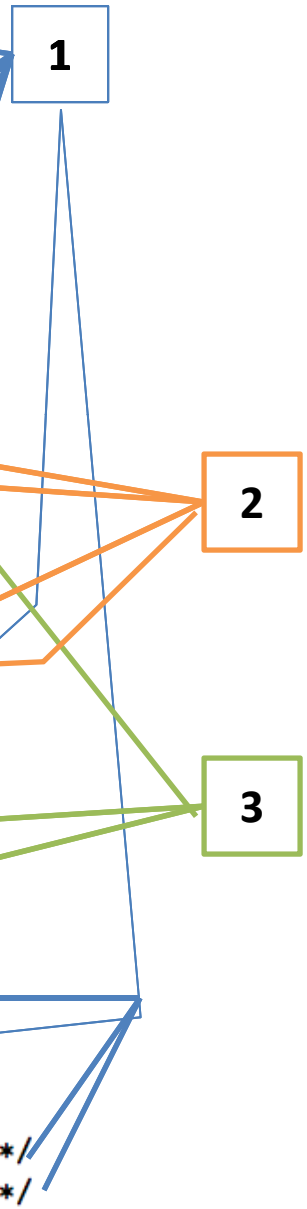
Flow Abstraction and the Switch- API



Flow Abstraction and the Switch- API



```
/* Immutable messages. */
OFPT_HELLO,                 /* Symmetric message */
OFPT_ERROR,                 /* Symmetric message */
OFPT_ECHO_REQUEST,          /* Symmetric message */
OFPT_ECHO_REPLY,            /* Symmetric message */
OFPT_VENDOR,                /* Symmetric message */

/* Switch configuration messages. */
OFPT_FEATURES_REQUEST,      /* Controller/switch message */
OFPT_FEATURES_REPLY,        /* Controller/switch message */
OFPT_GET_CONFIG_REQUEST,    /* Controller/switch message */
OFPT_GET_CONFIG_REPLY,      /* Controller/switch message */
OFPT_SET_CONFIG,            /* Controller/switch message */

/* Asynchronous messages. */
OFPT_PACKET_IN,             /* Async message */
OFPT_FLOW_REMOVED,          /* Async message */
OFPT_PORT_STATUS,           /* Async message */

/* Controller command messages. */
OFPT_PACKET_OUT,            /* Controller/switch message */
OFPT_FLOW_MOD,              /* Controller/switch message */
OFPT_PORT_MOD,              /* Controller/switch message */

/* Statistics messages. */
OFPT_STATS_REQUEST,         /* Controller/switch message */
OFPT_STATS_REPLY,           /* Controller/switch message */

/* Barrier messages. */
OFPT_BARRIER_REQUEST,       /* Controller/switch message */
OFPT_BARRIER_REPLY,         /* Controller/switch message */

/* Queue Configuration messages. */
OFPT_QUEUE_GET_CONFIG_REQUEST,  /* Controller/switch message */
OFPT_QUEUE_GET_CONFIG_REPLY     /* Controller/switch message */
```

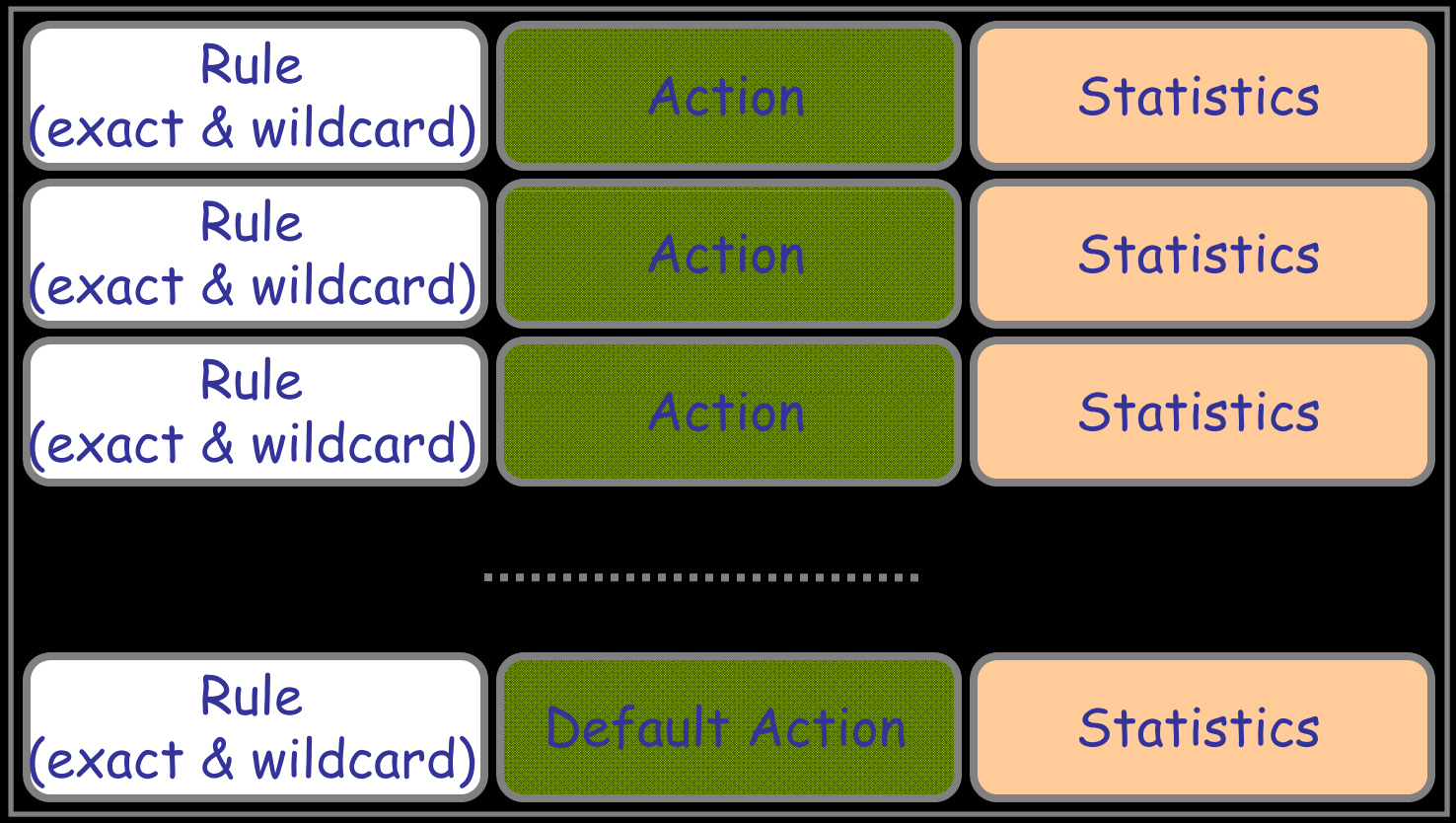


Switch API – a closer look @ OF 1.0

1. **Configuration**
2. **Control over Forwarding**
3. **Monitoring**

Switch API – a closer look @ OF 1.0

Match, Action, Stats



e.g. Port, VLAN ID, L2, L3, L4, …

e.g. unicast, mcast, map-to-queue, drop

Count packets & bytes
Expiration time/count

Switch API – a closer look @ OF 1.0
(2) Control over Forwarding

- Ingress port
- Ethernet source/destination address
- Ethernet type
- VLAN ID
- VLAN priority
- IPv4 source/destination address
- IPv4 protocol number
- IPv4 type of service
- TCP/UDP source/destination port
- ICMP type/code

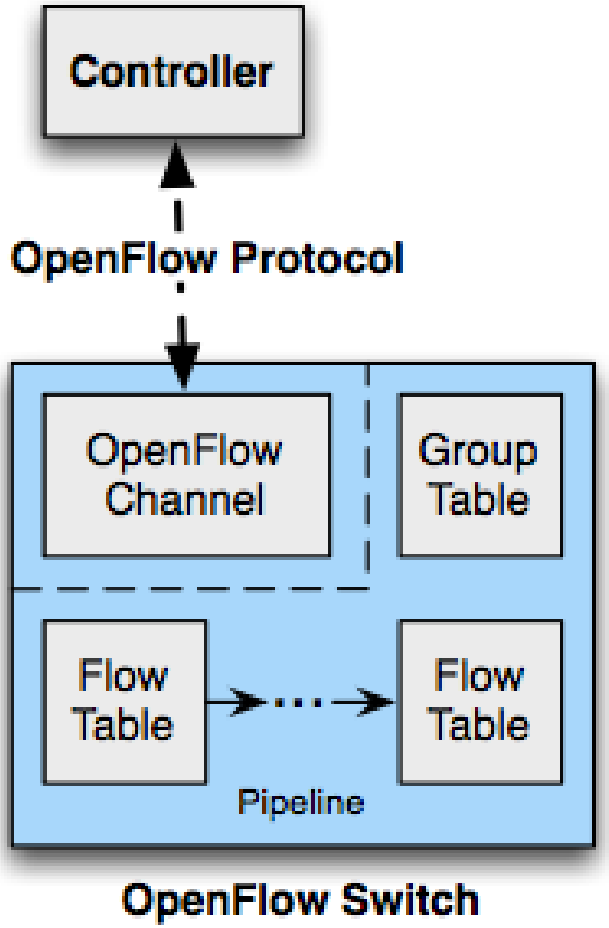
- All
  - Sent packet out on all interfaces, not including incoming interface
- Controller
  - Encapsulate and send the packet to the controller
- Local
  - Send the packet to the switch local network stack
- Table
  - Perform action in flow table (for packet_out)
- IN_PORT
  - Send the packet out to the input port

- Set VLAN ID (or add VLAN tag)
- Set VLAN priority
- Strip VLAN header
- Modify Ethernet source/destination address
- Modify IPv4 source/destination address
- Modify IPv4 type of service bits
- Modify TCP/UDP source/destination port

Switch API – a closer look @ OF 1.0
(3) Monitoring

Counter	Bits
Per Table	
Active Entries	32
Packet Lookups	64
Packet Matches	64
Per Flow	
Received Packets	64
Received Bytes	64
Duration (seconds)	32
Duration (nanoseconds)	32
Per Port	
Received Packets	64
Transmitted Packets	64
Received Bytes	64
Transmitted Bytes	64
Receive Drops	64
Transmit Drops	64
Receive Errors	64
Transmit Errors	64
Receive Frame Alignment Errors	64
Receive Overrun Errors	64
Receive CRC Errors	64
Collisions	64
Per Queue	
Transmit Packets	64
Transmit Bytes	64
Transmit Overrun Errors	64

Switch API – OF 1.1 (a big jump)



- Multiple Tables
- Groups
- MPLS and QinQ support
- Logical ports
- Connection failure modes

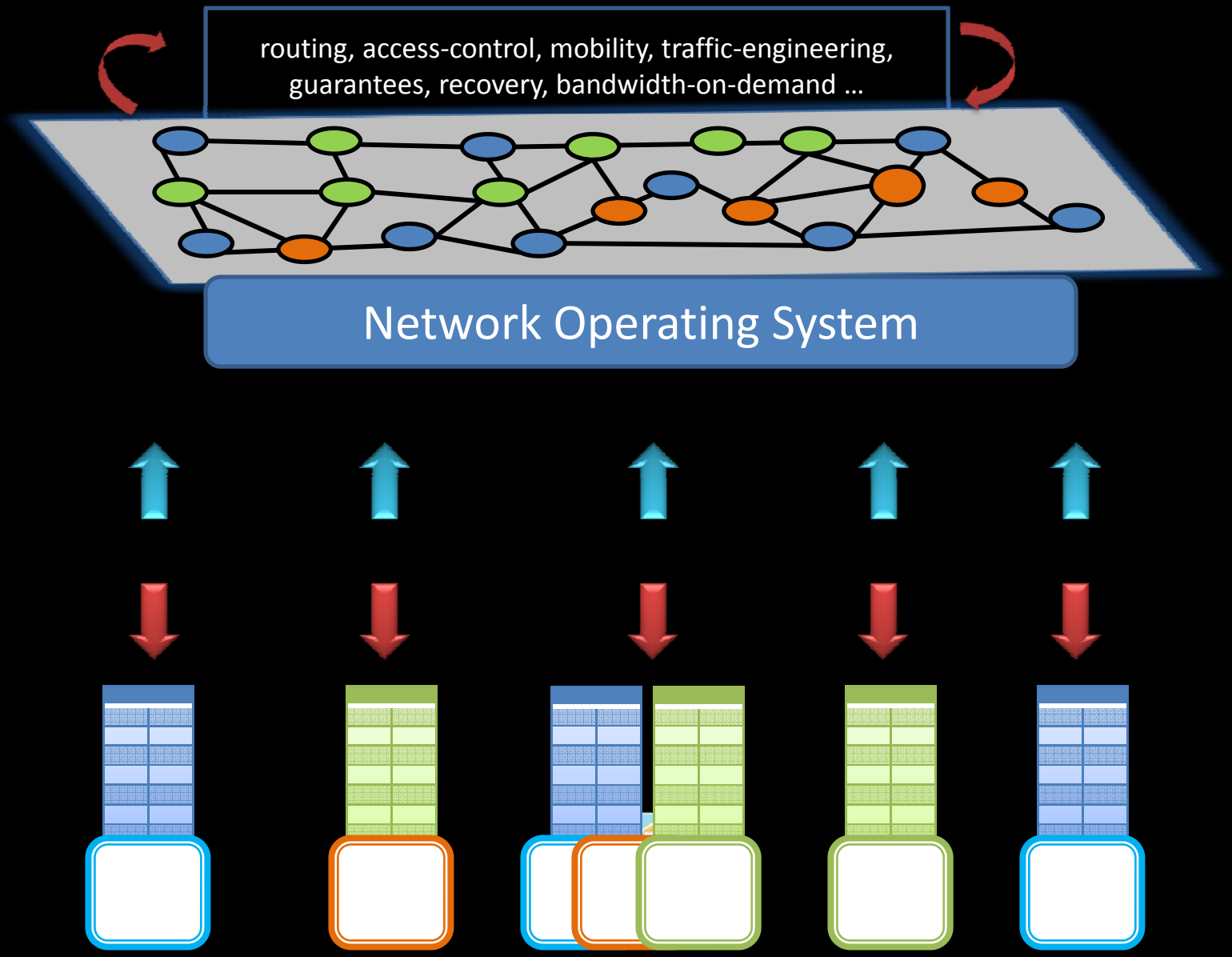
OpenFlow has **evolved** towards production readiness.

	state	behavior	Interface msg
1.0 Q4 '09	flows ports	forward {0, 1, n} match Eth, VLAN, IP, L4	single message queue w/optional barriers
1.1 Q1 '11	+ Group Tables + Multiple Tables/Pipelines:	+ forward 1-in-n (ECMP) + match QinQ, MPLS, SCTP + match virtual ports	
1.2 Q4 '11		+ IPv6 + multiple controllers	+ extensible match + extensible actions
1.3 Q2 '12		+ per-flow metering + tunnel-id	+ multiple channels (auxiliary connections)
1.4 Q4 '13	+ optical ports + synchronized tables		+ bundle messages

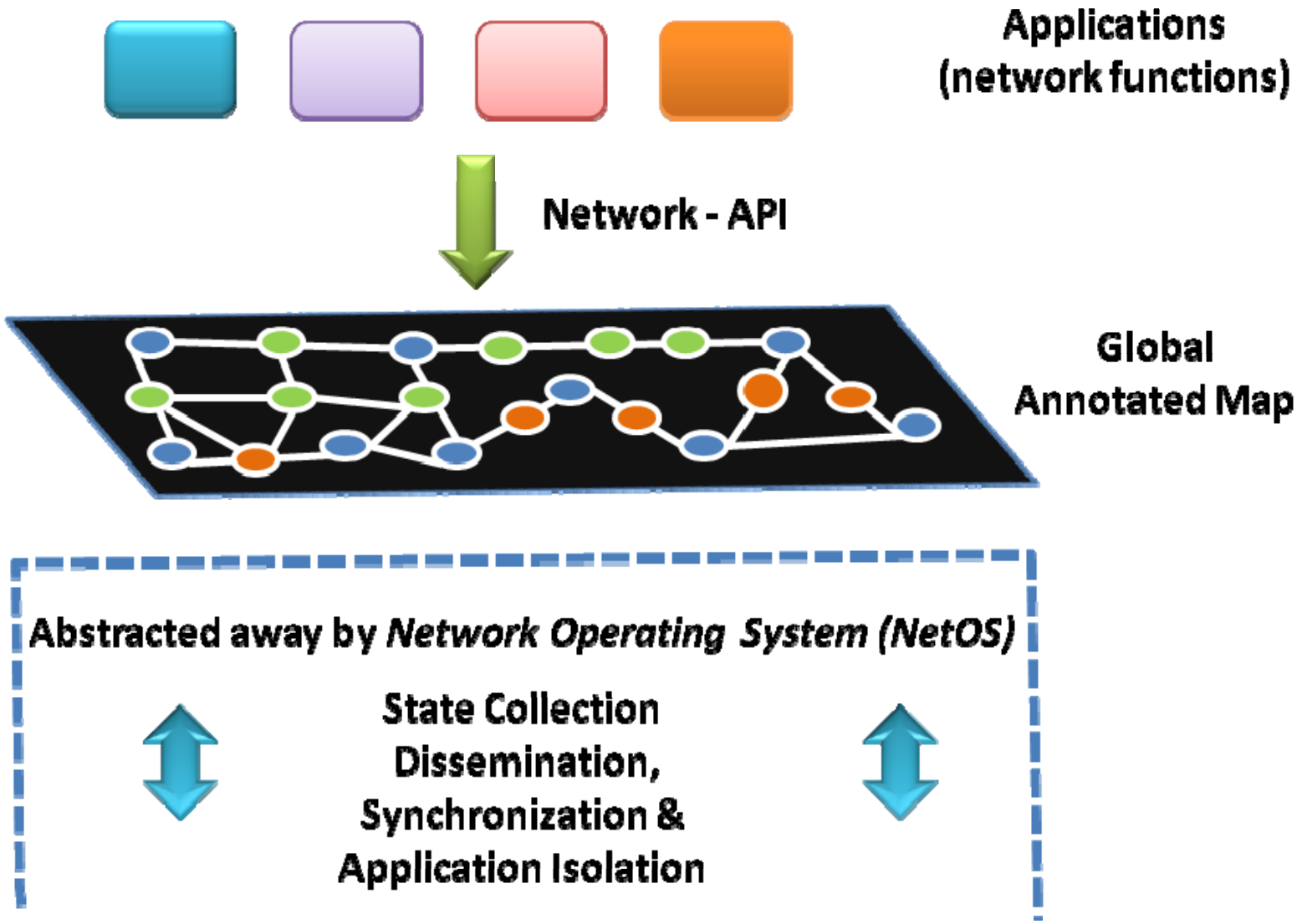
Slide: courtesy of SDN Academy

© 2013 SDN Academy, LLC™. All Rights Reserved.

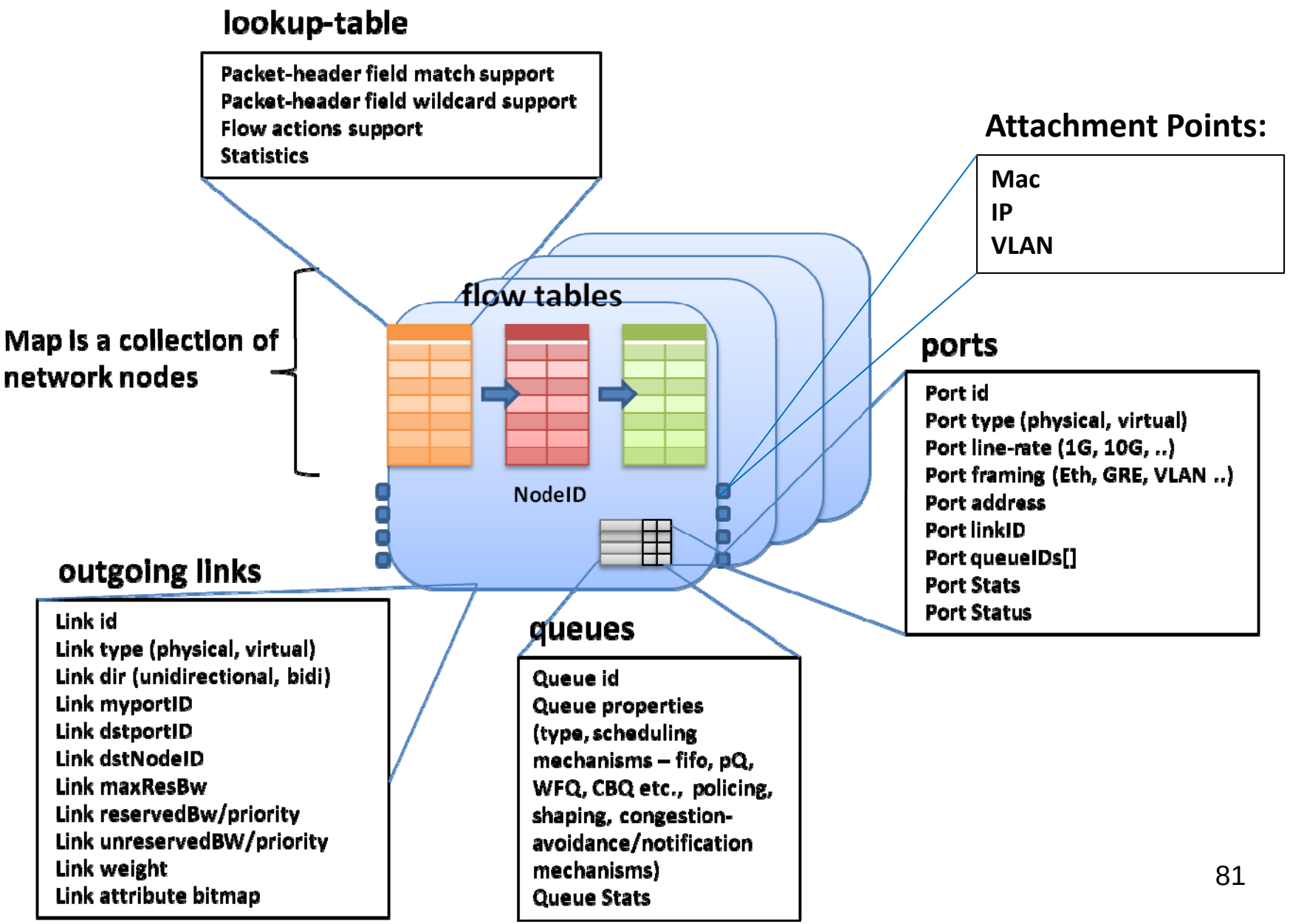
2. The Map Abstraction



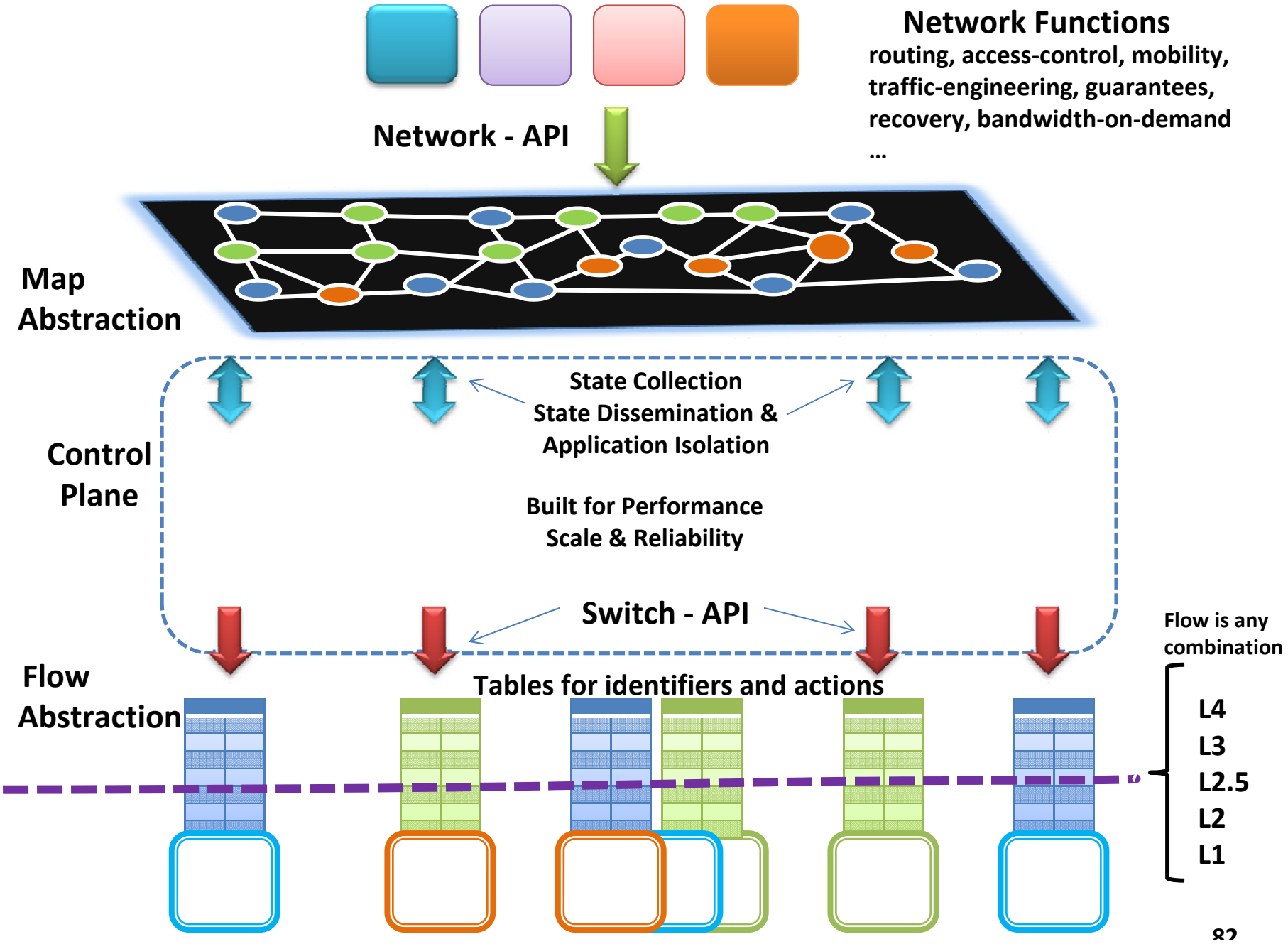
Map Abstraction and the Network- API



Map Abstraction



SDN Control Architecture



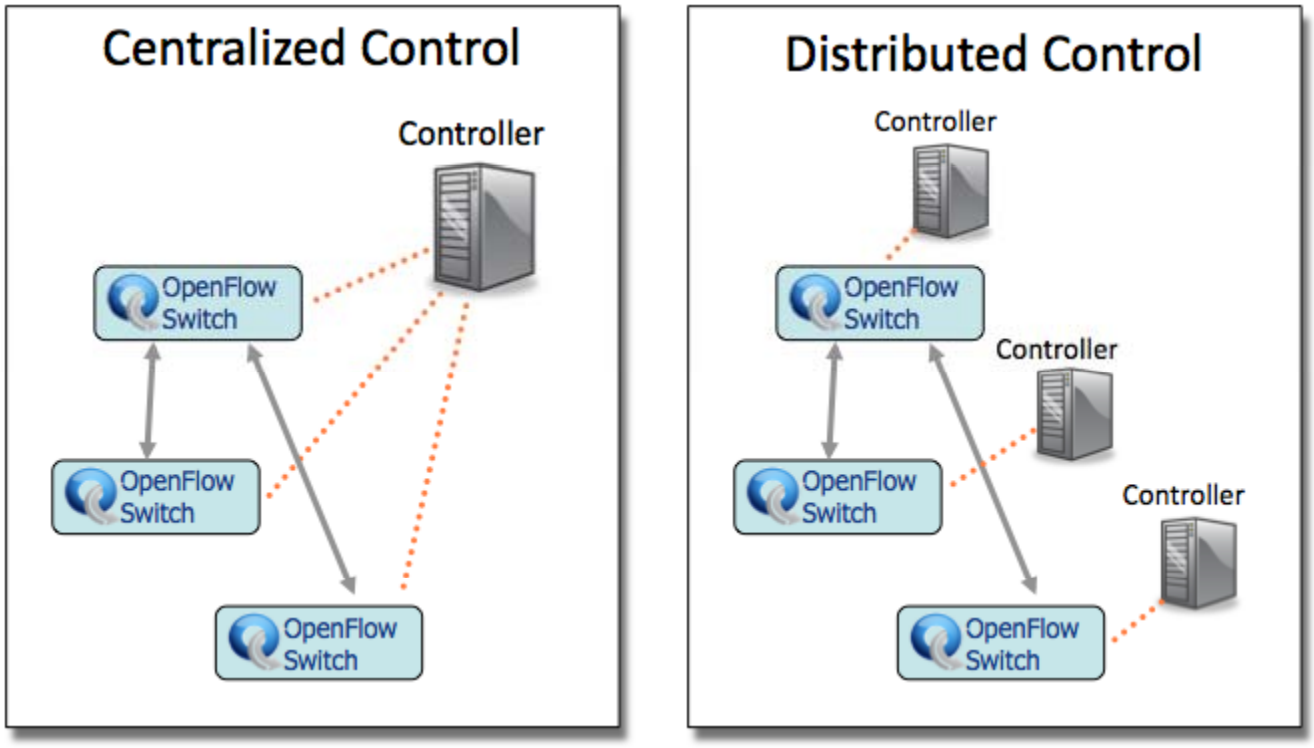
Outline

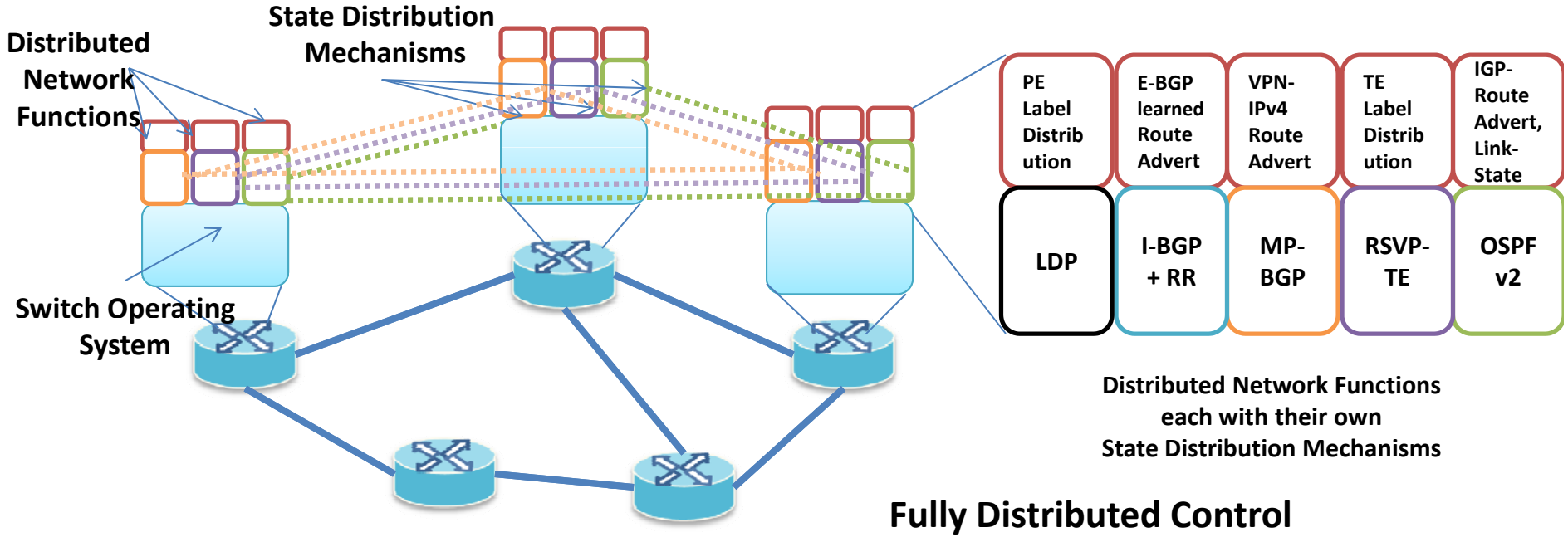
- Module 1: SDN/OpenFlow Introduction
- Module 2: OpenFlow/SDN Deep Dive
- Module 3: SDN Architecture
- Module 4: Operational Choices & Performance
- Module 5: Benefits & Use Cases
  - ❖ Demo: Network Virtualization
- Module 6: Demystifying SDN
- Q & A

Module 4

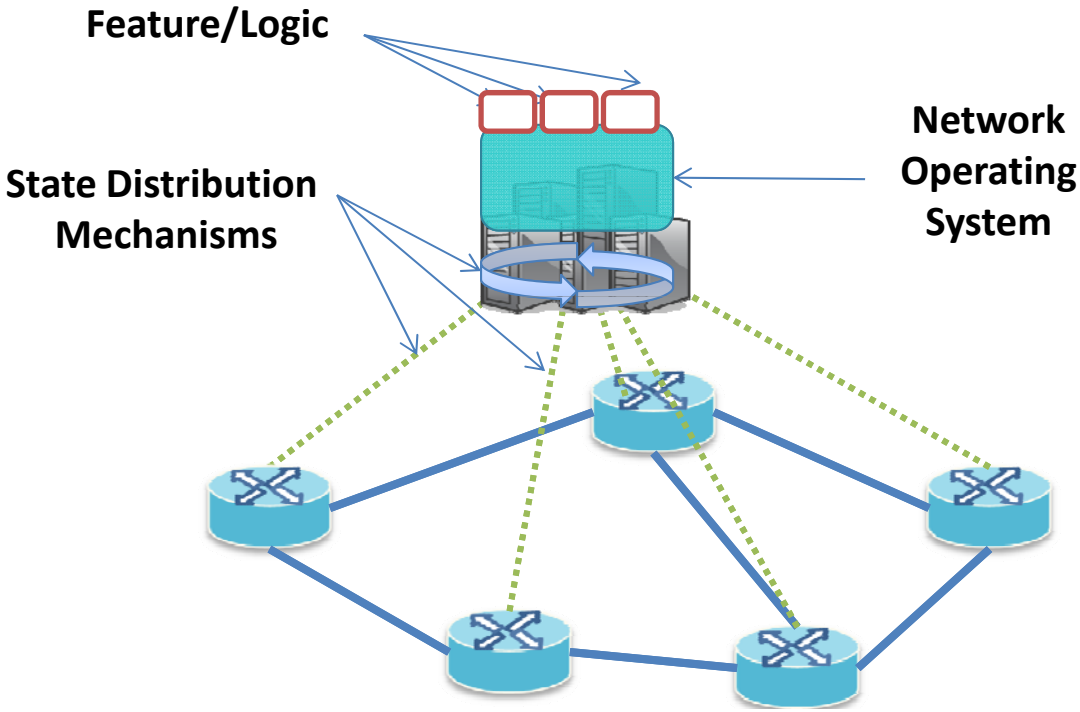
Operational Choices & Performance

1: Centralized vs Distributed Control



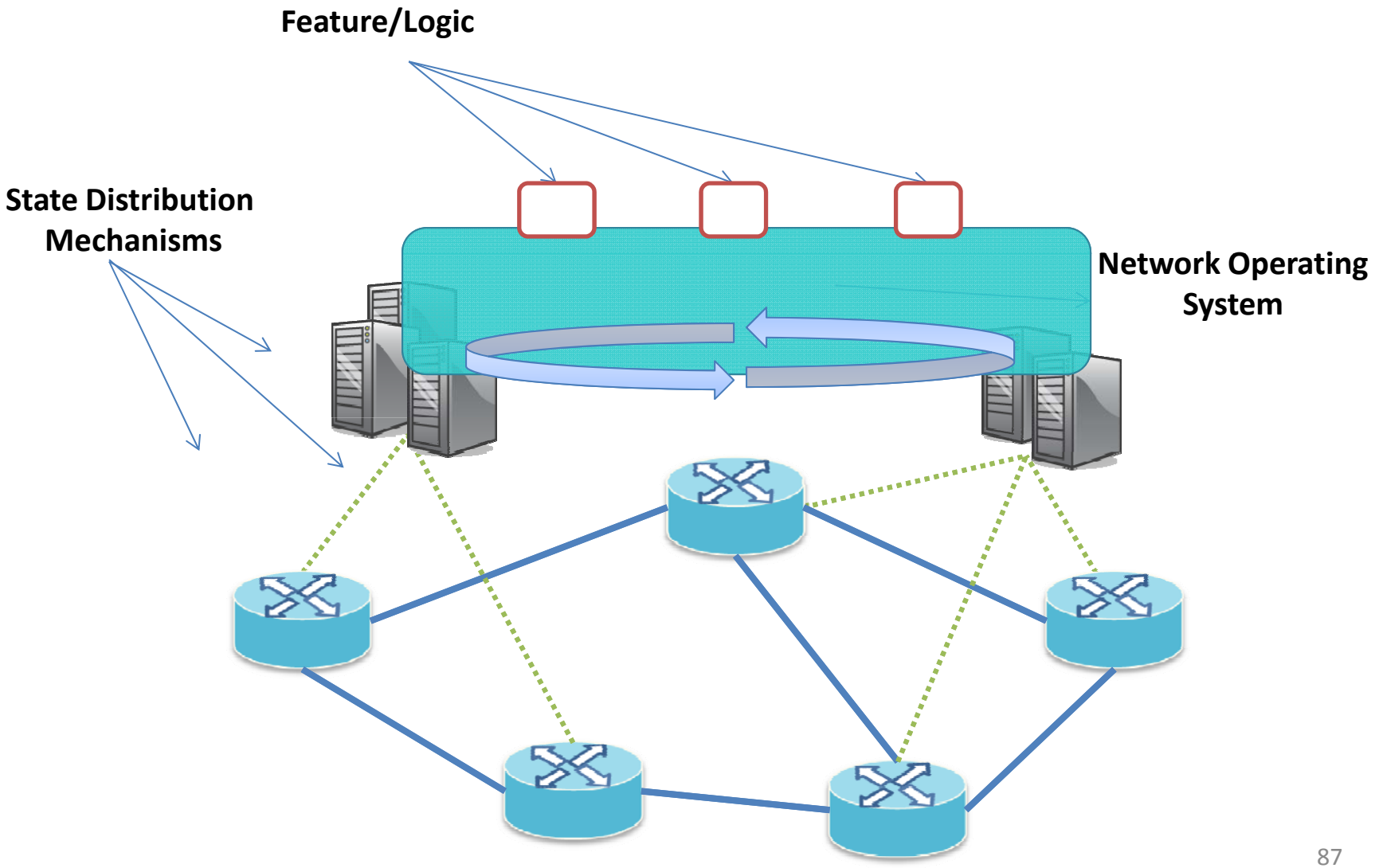


Fully Distributed Control



Fully Centralized Control

Logically Centralized… Physically Distributed



2: Microflow vs. Aggregated

Microflow

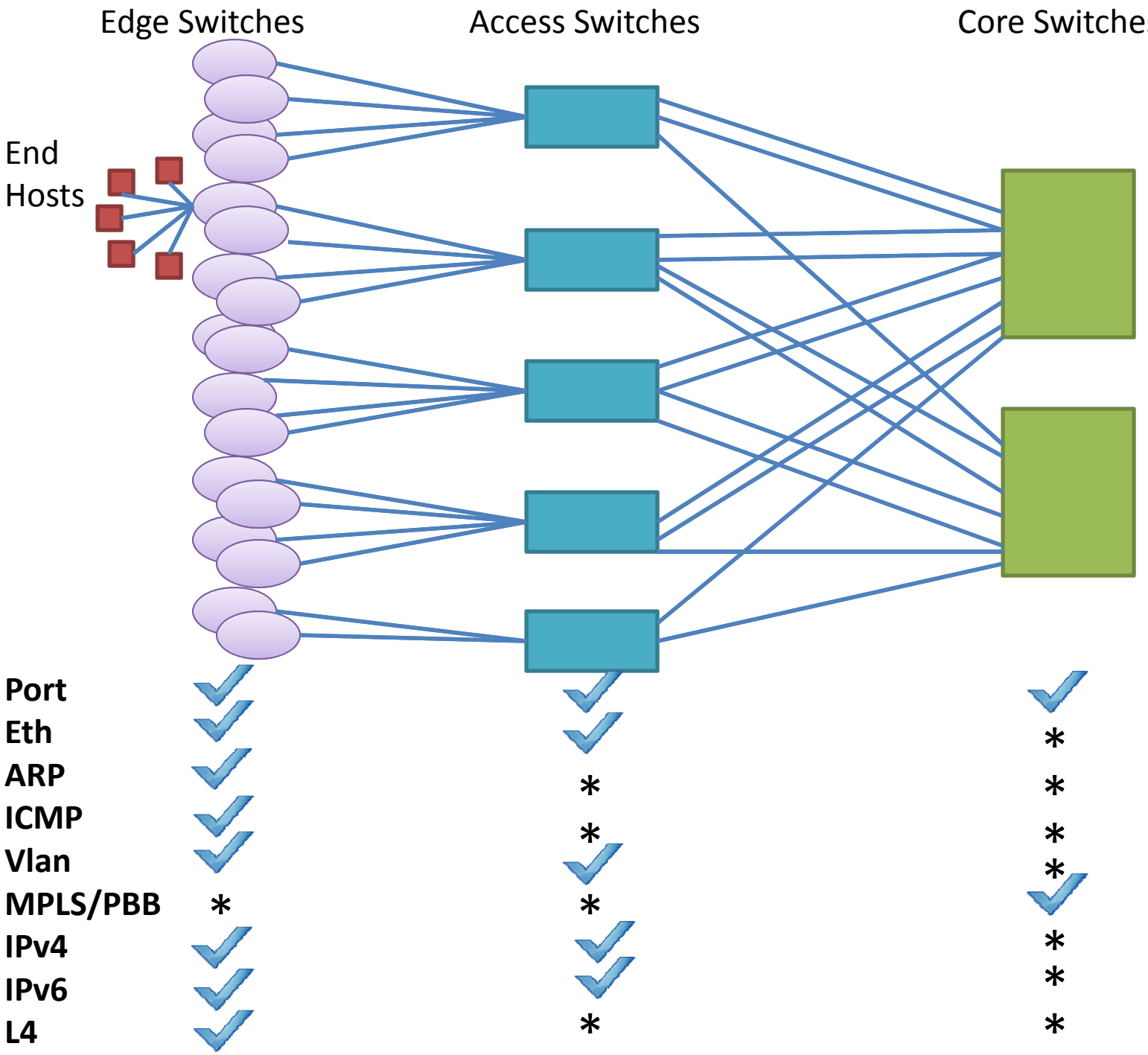
- Every flow is individually set up by controller
- Exact-match flow entries
- Flow table contains one entry per flow

- Good for fine grain control, policy, and monitoring, e.g. campus

Aggregated

- One flow entry covers large groups of flows
- Wildcard flow entries
- Flow table contains one entry per category of flows

- Good for large number of flows, e.g. backbone


Edge Switches
Access Switches
Core Switches
End Hosts
Port
Eth
ARP
ICMP
Vlan
MPLS/PBB
IPv4
IPv6
L4

3: Reactive vs. Proactive (pre-populated)

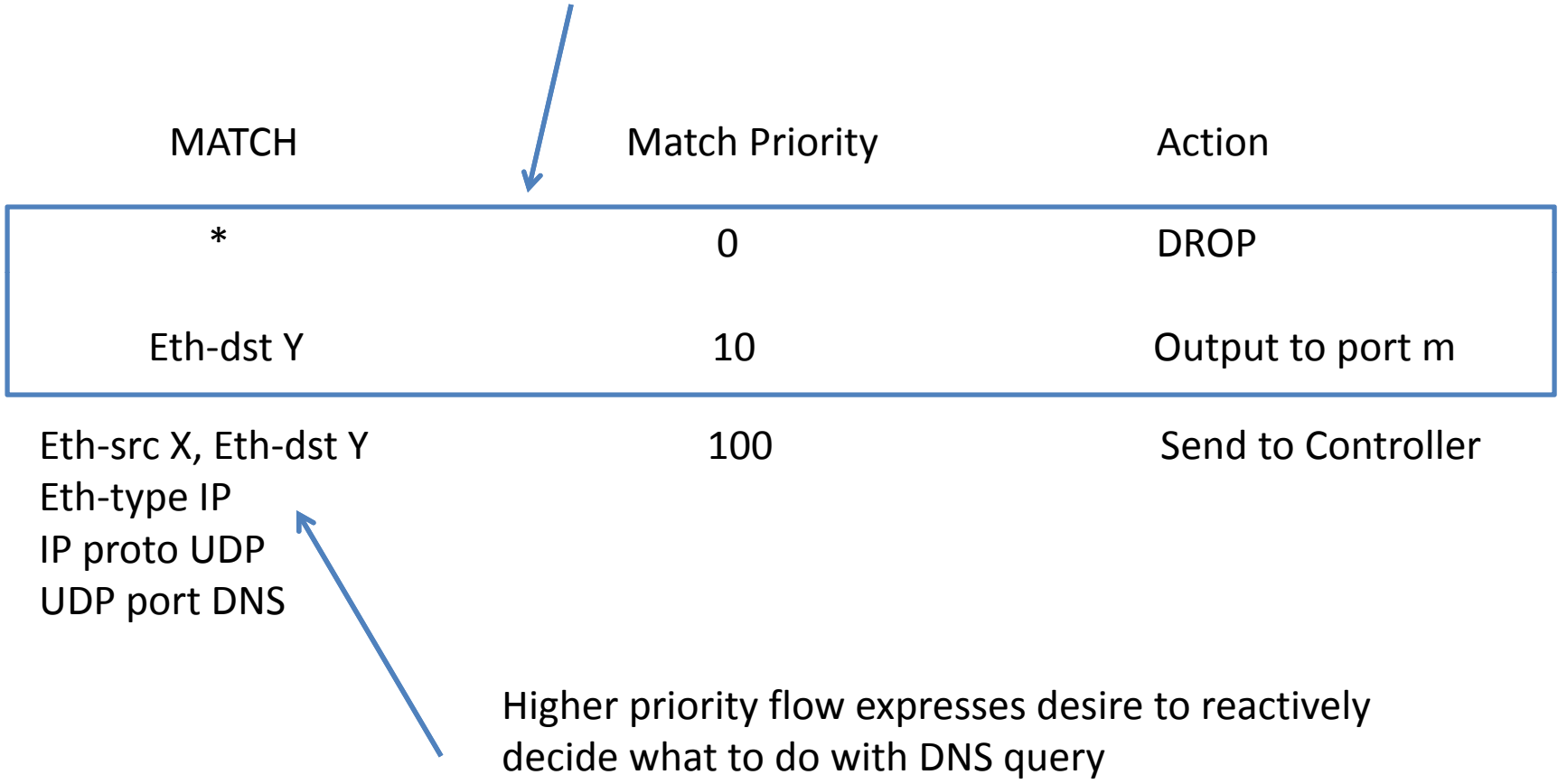
Reactive

- First packet of flow triggers controller to insert flow entries
- Efficient use of flow table
- Every flow incurs small additional flow setup time
- If control connection lost, switch has limited utility
- Extremely simple fault recovery

Proactive

- Controller pre-populates flow table in switch
- Zero additional flow setup time
- Loss of control connection does not disrupt traffic
- Essentially requires aggregated (wildcard) rules

Hybrid: Reactive + Proactive ➔ The use of Match Priorities



4: Virtual vs Physical

Virtual

- Assumes configurable switching within a host: in the OS or hypervisor
- Software! Memory, processing, arbitrary modifications
- Massive flow rates
- Limited to the hardware below

Physical

- **Very high performance in**
  - **Data Plane**
- **But less flexible**
  - **ASIC capabilities**
  - **Table Size limitations**
  - **CPU bus limitations**

5: Fully Consistent vs Eventually Consistent

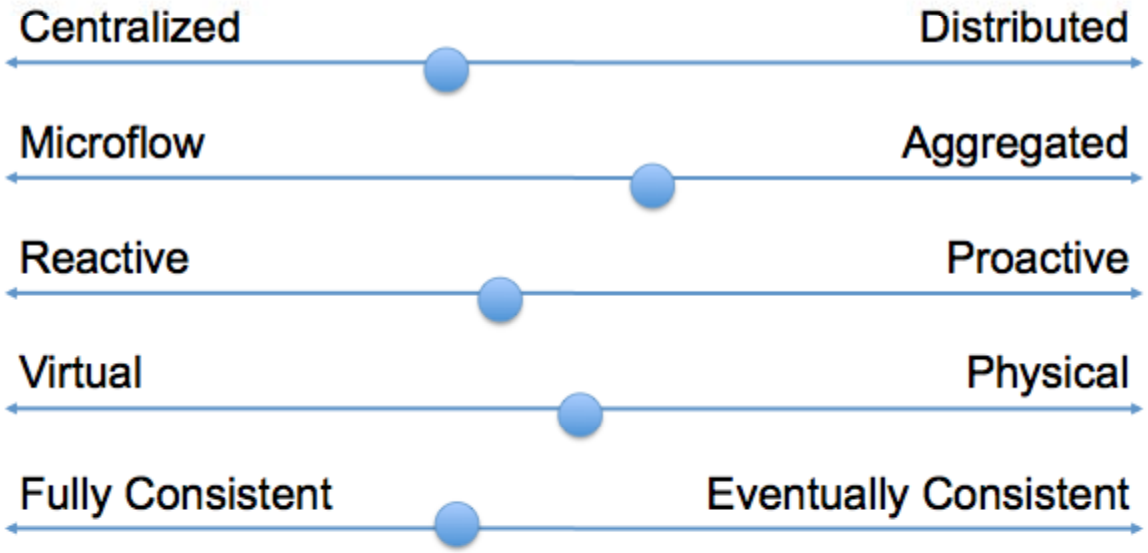
Fully Consistent

- Certainty about state
- Consistent state is harder to scale
- Easier to reason about state and its transitions

Eventually Consistent

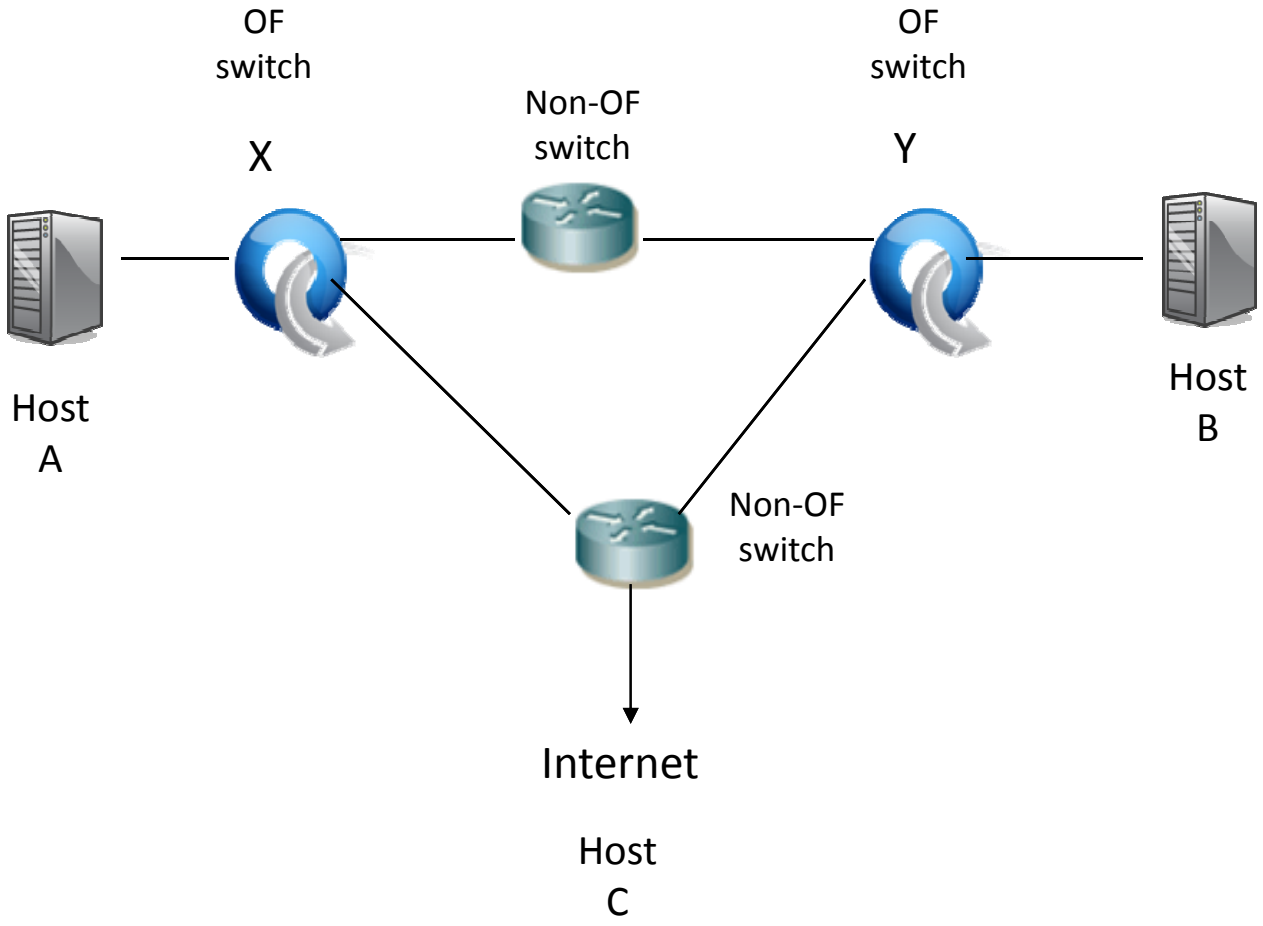
- Uncertainty about state now, but eventually converges
- Probabilistic state is easier to scale
- Introduces the possibility of long-lived route flaps and unstable control systems

Your SDN network or application?

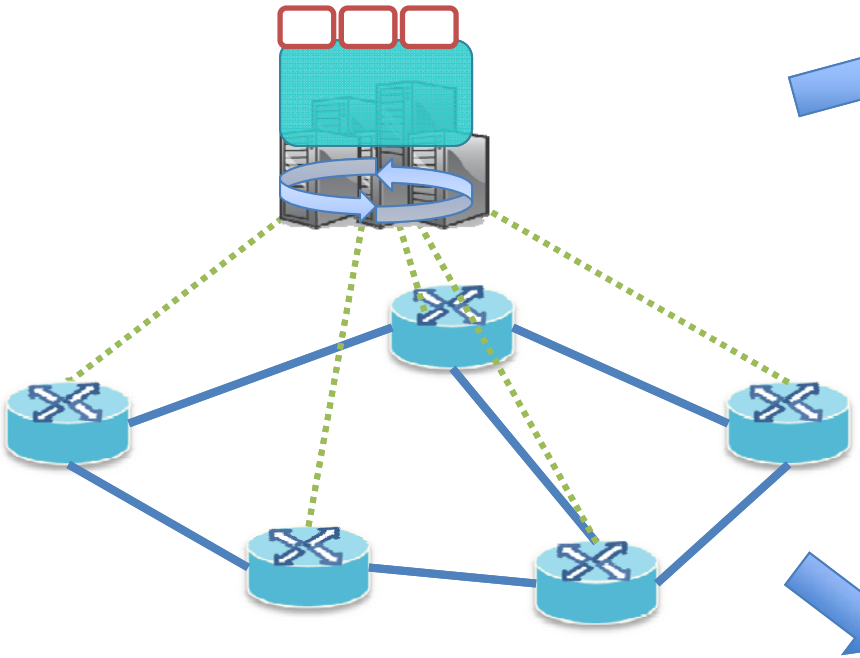


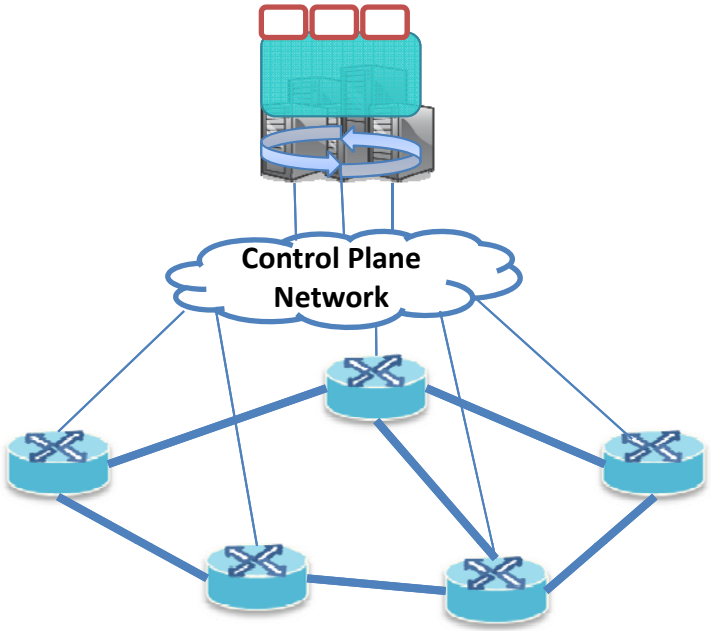
Hybrid approaches often have desirable properties; can use these now.

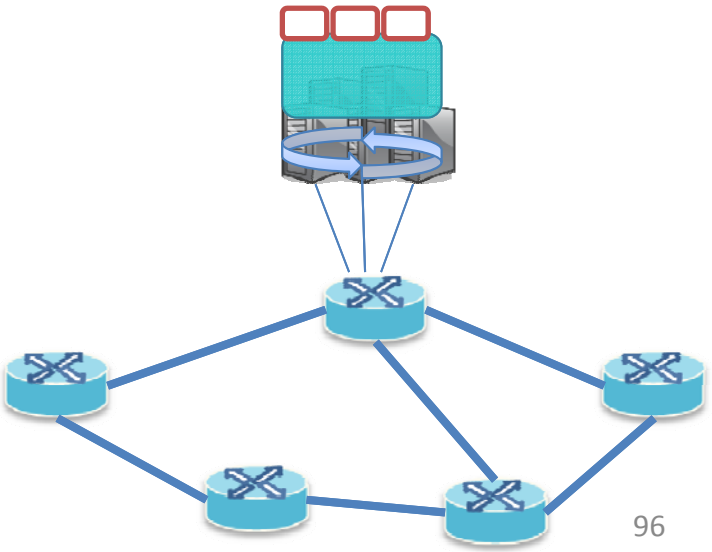
6: All OpenFlow vs. OF + NOF



7: In Band vs. OOB







8: What Does Controller Performance Mean?

Latency: Total time to process a single event at the controller

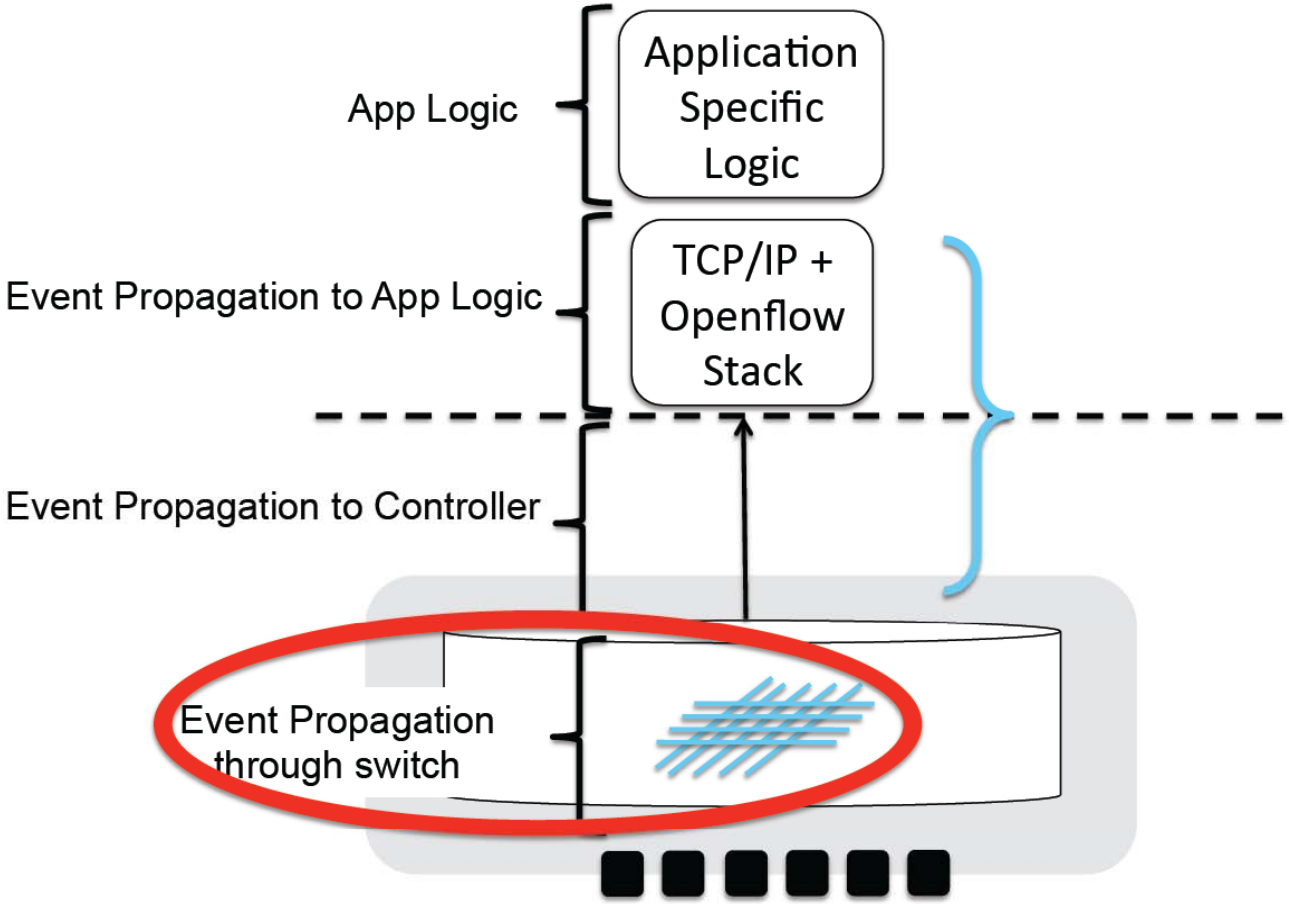
- Indication of responsiveness
- Important in reactive settings

Throughput: Total number of flows/s handled by the controller

- Doesn't really matter
- Indication of scalability
- Important property is how scales w/ number of controllers
- Generally application dependent

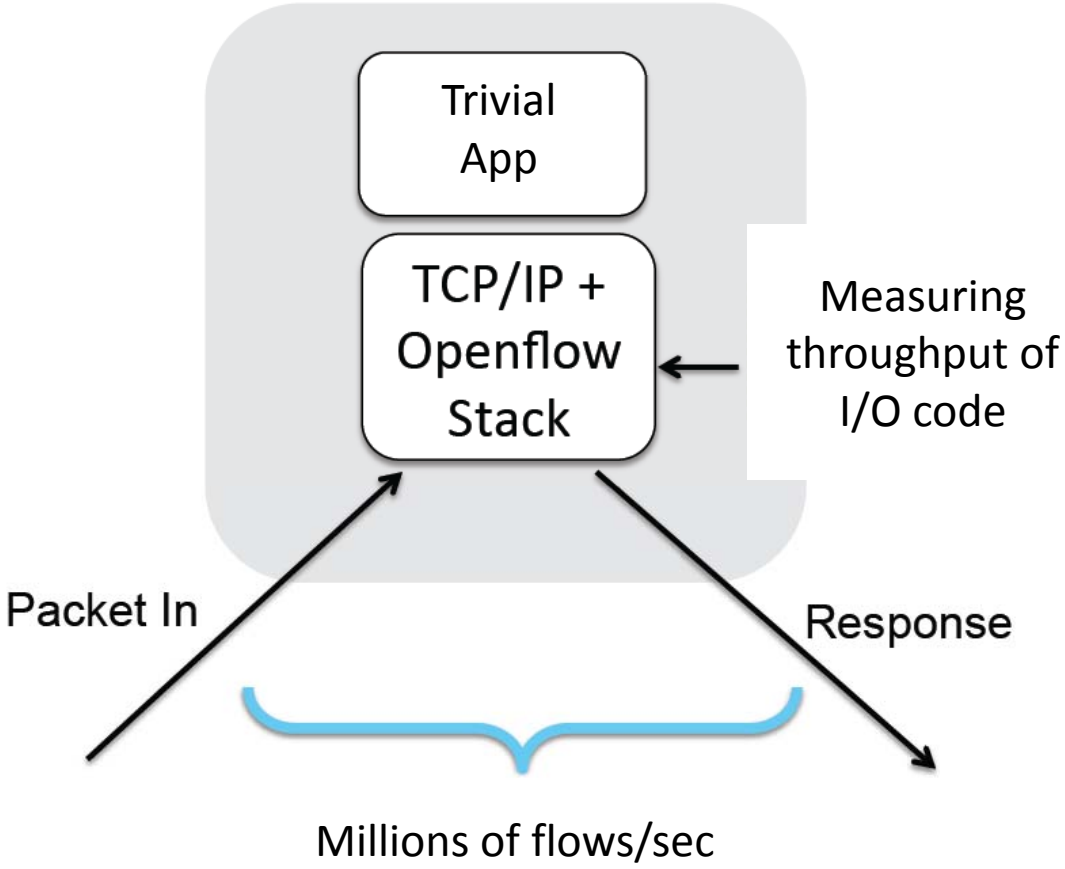
Source: Martin Casado

Throughput



Source: Martin Casado

Throughput



Application code will drop throughput → need more controllers → throughput matters less → what matters is application scale

What is Controller Scalability?

Maximum size of network controllable via SDN

Largely dependent on controller application logic

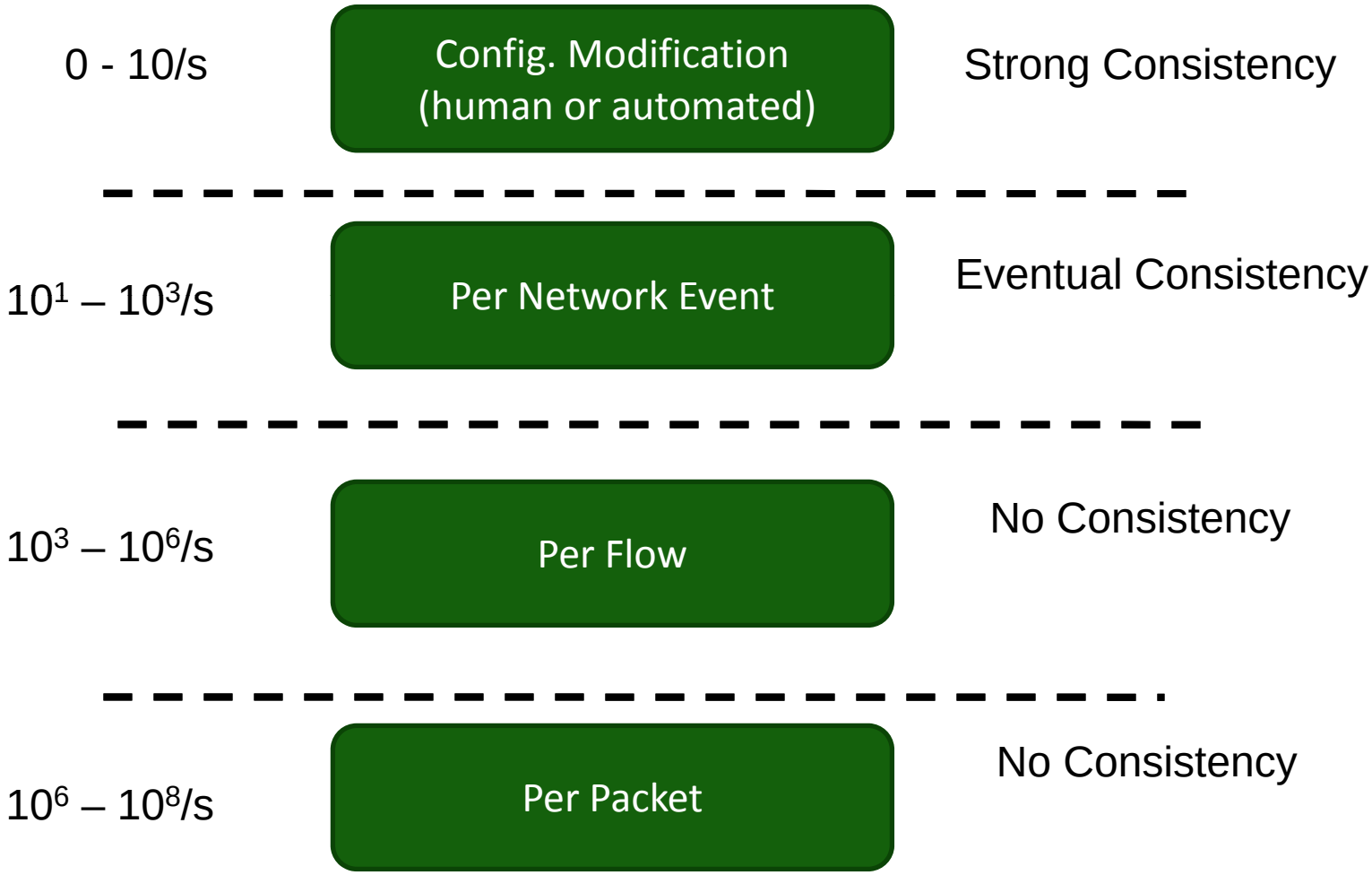
Limiting resources

- **Memory**
- **Compute**
- **Consistency overhead**

The only fundamental bottleneck

Source: Martin Casado

Why Does This Approach Scale?



Source: Scott Shenker, Berkeley

Outline

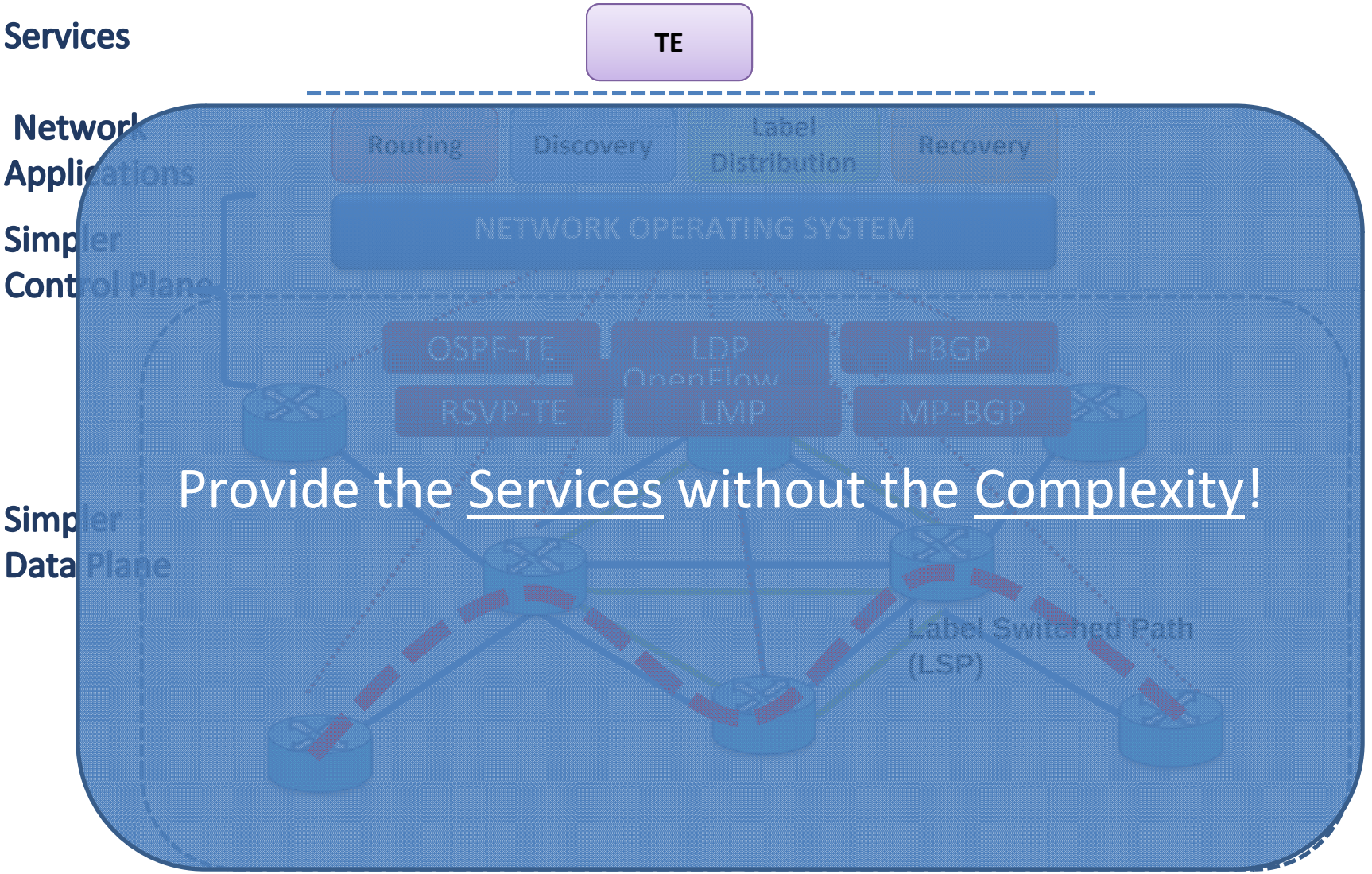
- Module 1: SDN/OpenFlow Introduction
- Module 2: OpenFlow/SDN Deep Dive
- Module 3: SDN Architecture
- Module 4: Operational Choices & Performance
- Module 5: Benefits & Use Cases
  - ❖ Demo: Network Virtualization
- Module 6: Demystifying SDN
- Q & A

Module 5

Benefits & Use Cases

Software Defined Networking

.. provide the services without the complexity



The biggest Use Case for OF/SDN is…

SIMPLICITY

Adding a service

What would it take in today's networks?

Carrier need/idea

DEPLOYMENT

Limited Field Trials

Carrier Lab Trials

Extensions…

3-5 year process

..if it gets off the ground

Ask vendors to implement solution

B C J

Long time later non-interoperable pre-standard solutions

Standard

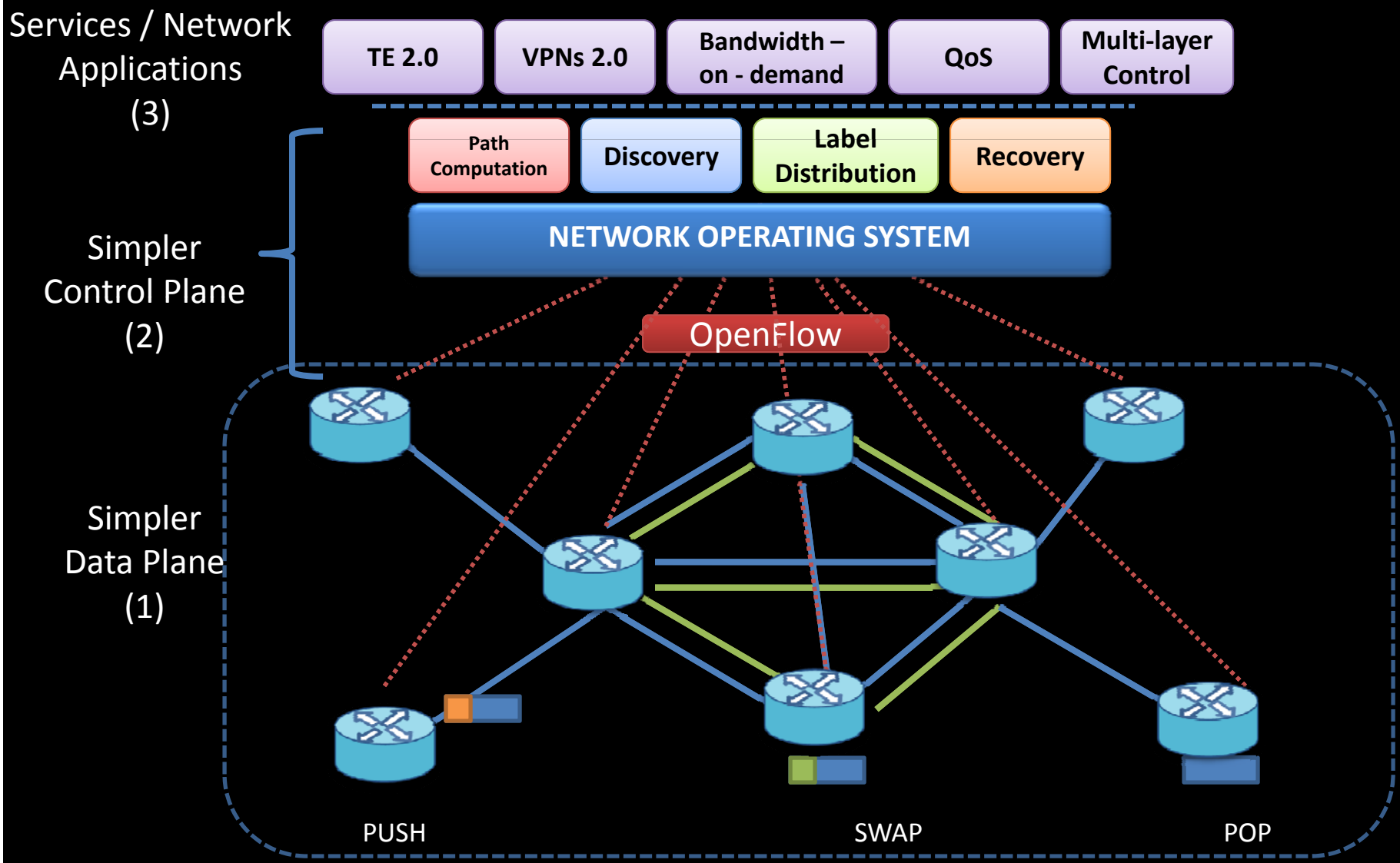
XJBC

B

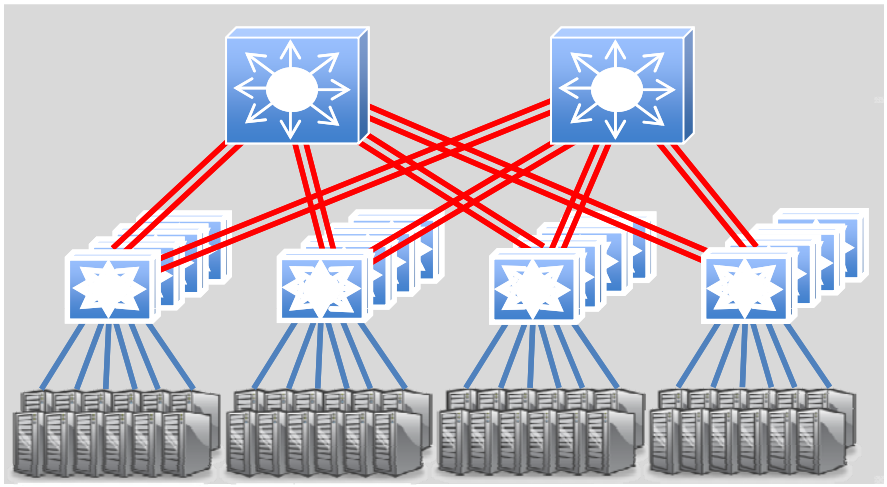
C

J

Adding a Service in OF/SDN



TCO Example: New Data Center



Cost

200,000 servers
Fanout of 20 ⇨ 10,000 switches
$5k commercial switch ⇨ $50M
$1k custom-built switch ⇨ $10M

Savings in 10 data centers = $400M

Control

1. Optimize for features needed
2. Customize for services & apps
3. Quickly improve and innovate

Large data center operators are moving towards defining their own network in software.

Benefits of SDN

1. Simpler Control with Greater Flexibility

- Networks work because we can master complexity, but what we should be doing is extracting simplicity, with the right abstractions

2. Programmability: Speed of Innovation, Ease of Service Insertion & Faster Time to Market

- Does not involve changing/creating a fully distributed protocol

3. Lower Total Cost of Ownership (TCO)

- Lower Opex – easier to manage, troubleshoot, emulate, automate, optimize
- Lower Capex – replacing proprietary hardware, pay for what you need and no more.

Slide: Courtesy of SDN Academy

© 2013 SDN Academy, LLC™. All Rights Reserved.

Use Case # 1: Traffic Engineering

Steering traffic to where the bandwidth is…

- good for the traffic - less congestion
- good for the network - better resource utilization

Deterministic Behavior

Efficient Resource Utilization

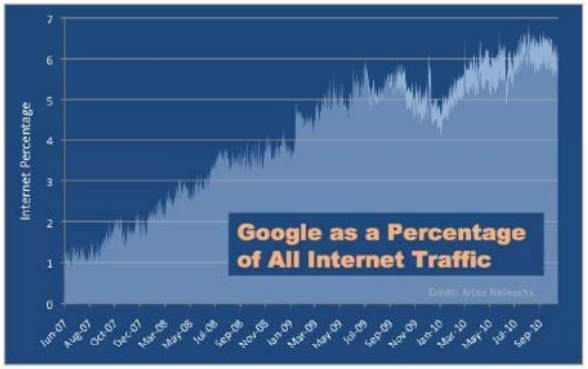
Older ways not used

Google's OpenFlow WAN

Internet Backbone Scale

"If Google were an ISP, as of this month it would rank as the second largest carrier on the planet."

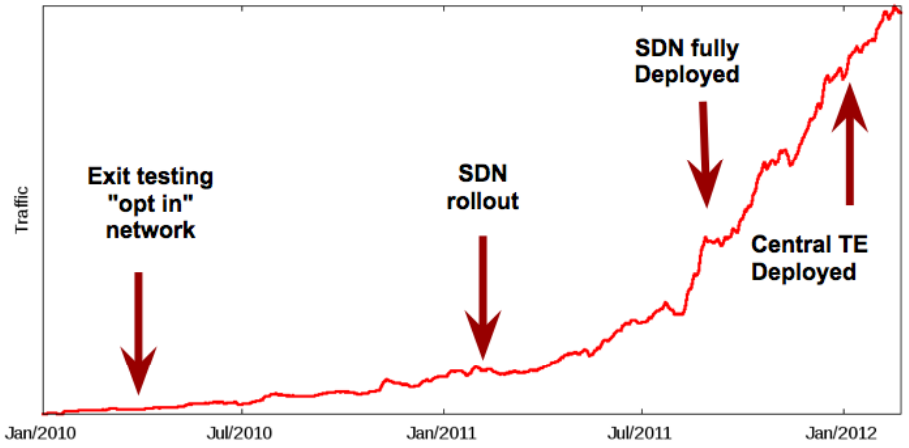
[ATLAS 2010 Traffic Report, Arbor Networks]



G-Scale Network Hardware

- Built from merchant silicon
  - 100s of ports of nonblocking 10GE
- OpenFlow support
- Open source routing stacks for BGP, ISIS
- Does not have all features
  - No support for AppleTalk...
- Multiple chassis per site
  - Fault tolerance
  - Scale to multiple Tbps

G-Scale WAN Usage



Source: Google

Why Software Defined WAN

Google

- Separate hardware from software
  - Choose hardware based on necessary features
  - Choose software based on TE requirements (*not* protocol requirements)
- Logically centralized network control
  - More deterministic
  - More efficient
- Separate monitoring, management, and operation from individual boxes
- *Flexibility and Innovation Velocity*

Source: Google

Advantages of Centralized TE Google

- Better efficiency with global visibility
- Converges faster to *target optimum* on failure
- Higher Efficiency
  - allows for explicit definition of cost functions
  - allows for in-house development of optimization algorithms
- Deterministic behavior
  - simplifies planning vs. over-provisioning for worst case variability
  - Can directly mirror production event streams for testing
- Supports innovation and more robust SW development
- Controller uses modern server hardware
  - significantly higher performance

Source: Google

Google SDN Experiences

Google

- Much faster iteration time: deployed production-grade centralized traffic engineering in two months
  - fewer devices to update
  - much better testing ahead of rollout
- Simplified, high fidelity test environment
  - Can emulate entire backbone in software
- Hitless SW upgrades and new features
  - Almost no packet loss and no capacity degradation
  - Most feature releases do not touch the switch
    - most state does not have to carried by network protocols

Source: Google

Learn more:

Google:

- **Urs Hoelze's talk at Open Networking Summit 2012**
  - **http://www.youtube.com/watch?v=VLHJUfgxEO4**
- **Slides:**
  http://opennetsummit.org/archives/apr12/hoelzle-tue-openflow.pdf
- **Whitepaper:**
  - **http://www.sdncentral.com/sdn-use-cases/**

Stanford:

- **http://www.youtube.com/watch?v=EpttFVKUrzs**
- **Chapter 5 http://www.openflow.org/wk/index.php/PACC_Thesis**

Use Case # 2: DC Network Virtualization

Where Do All the Bytes Come From?

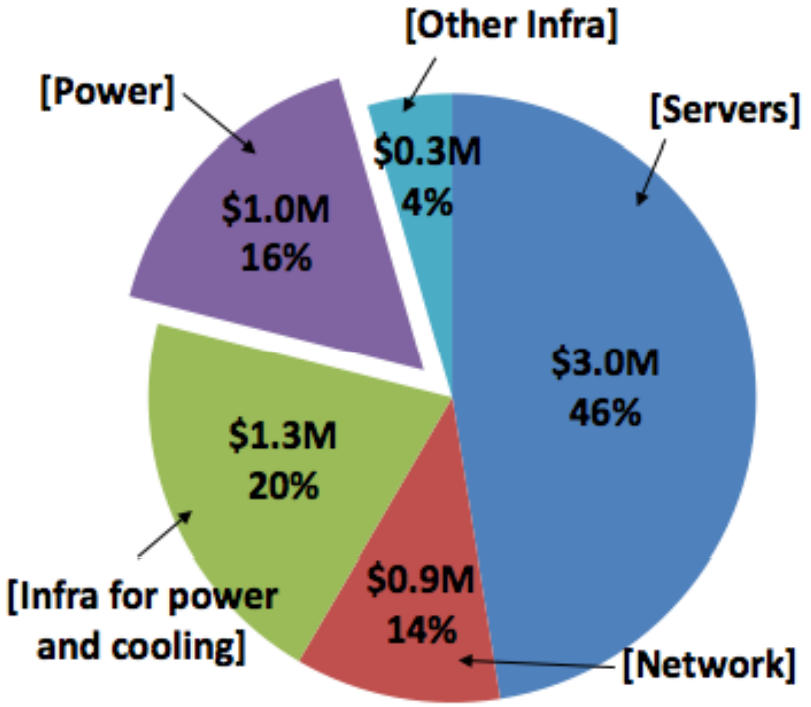
- The vast majority of bytes we consume through the Internet today originates from DCs (~14EB/yr in U.S., estimated by HMI, UCSD)

- Most large enterprises and organizations have their own DCs
- Typical data-center size: 50,000 to 200,000 servers today

Source: Changhoon Kim, Microsoft Research

Key Tenet of Data Centers: Agility at Scale

- DCs are digital-era factories, requiring huge up-front investment
- Golden rule: ***Maximize the amount of useful work per dollar spent***
- Best operating principles: **Multi-tenancy** & **Dynamic resizing**



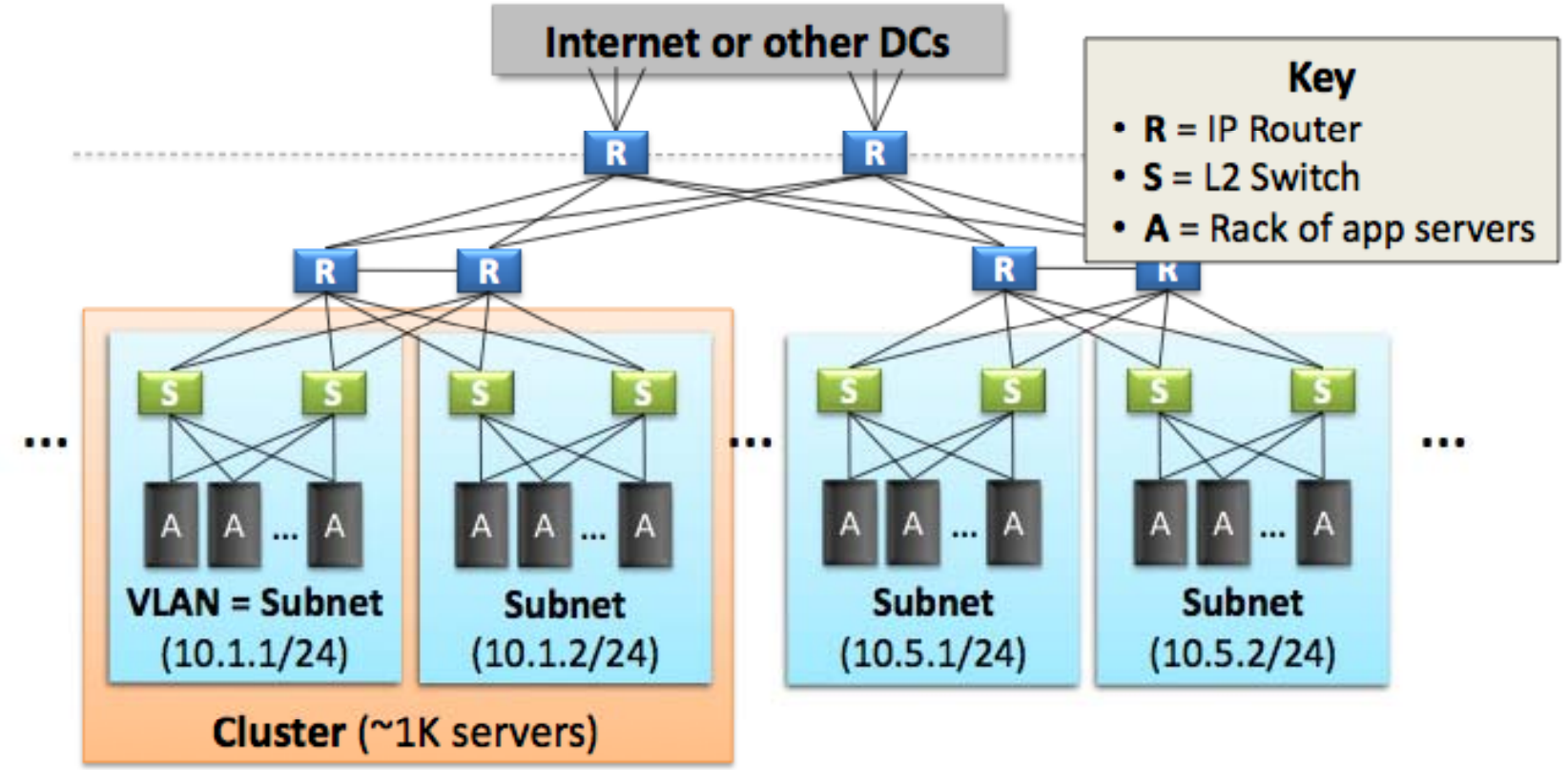
Monthly bill for a 50,000-server DC

Agility **– Capability to assign any servers to any tenants any time**

5

Source: Changhoon Kim, Microsoft Research

Status Quo: Conventional DC Network



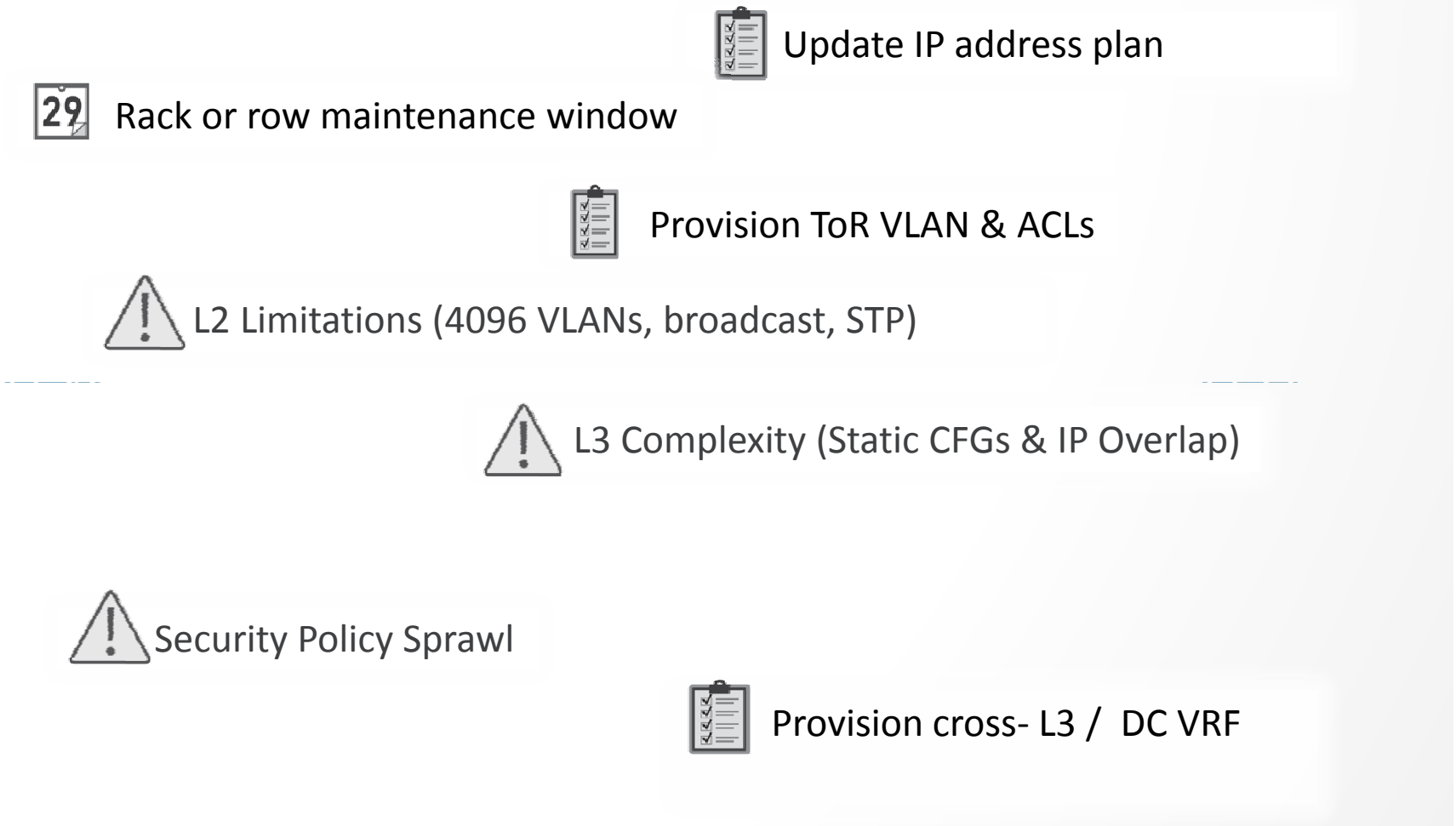
Reference – "Data Center: Load-balancing Data Center Services", Cisco 2004

Designed mainly for pre-cloud web-hosting services

Source: Changhoon Kim, Microsoft Research

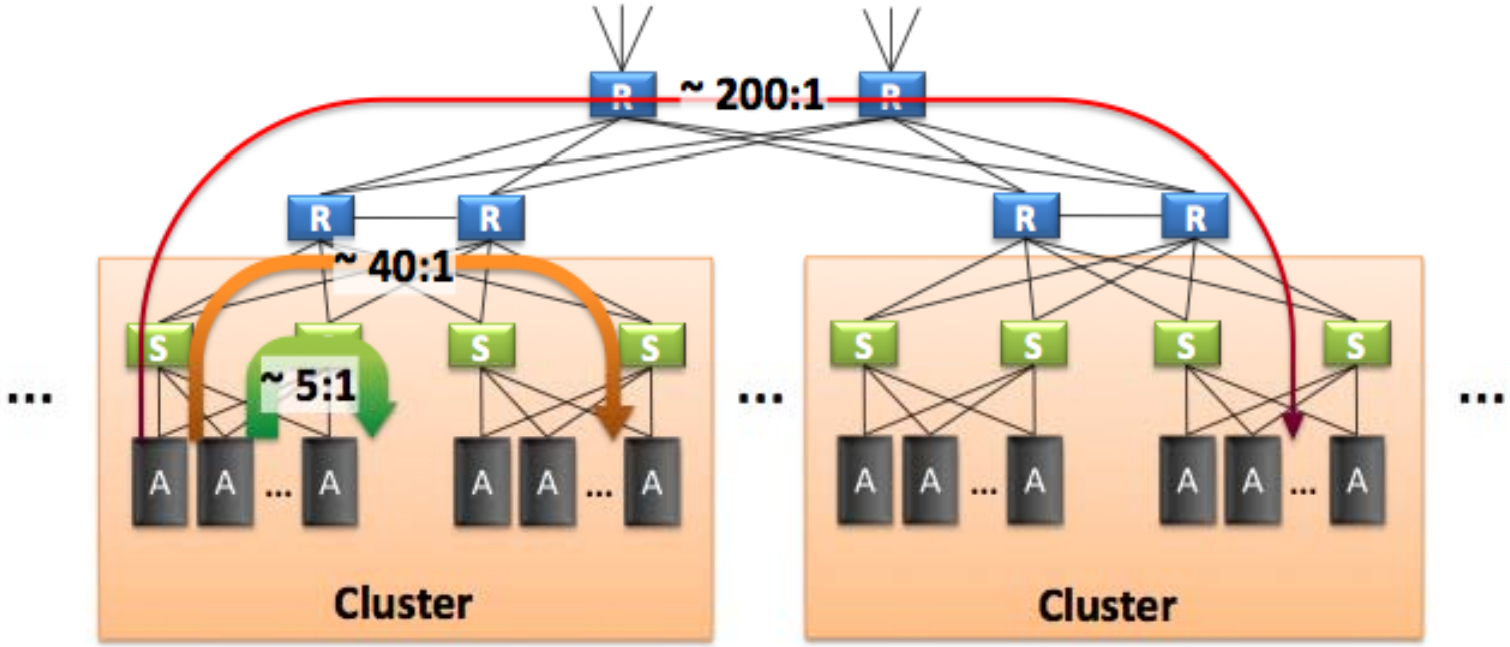
Cloud Data Center Network Challenges

Projects: DC Consolidation, x-DC Mobility and Multi-Tenant IaaS



Source: big switch networks

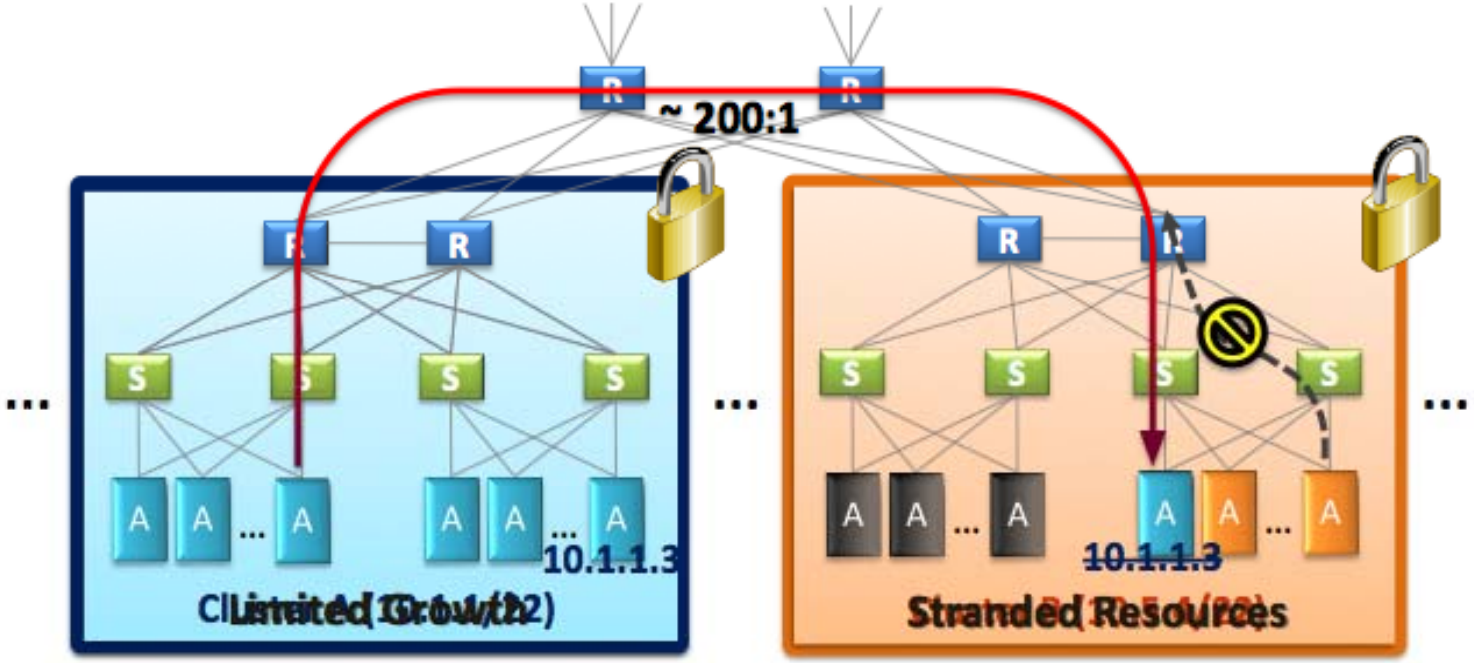
Today's Workload On Yesterday's Network



- Depends on high-cost mainframe-style network devices
- Extremely limited server-to-server capacity
- Highly-distributed apps suffer

Source: Changhoon Kim, Microsoft Research

Agility Is Very Hard To Achieve



- Cause waste of resources, lowering DC utilization

Source: Changhoon Kim, Microsoft Research

What We Desire Precisely

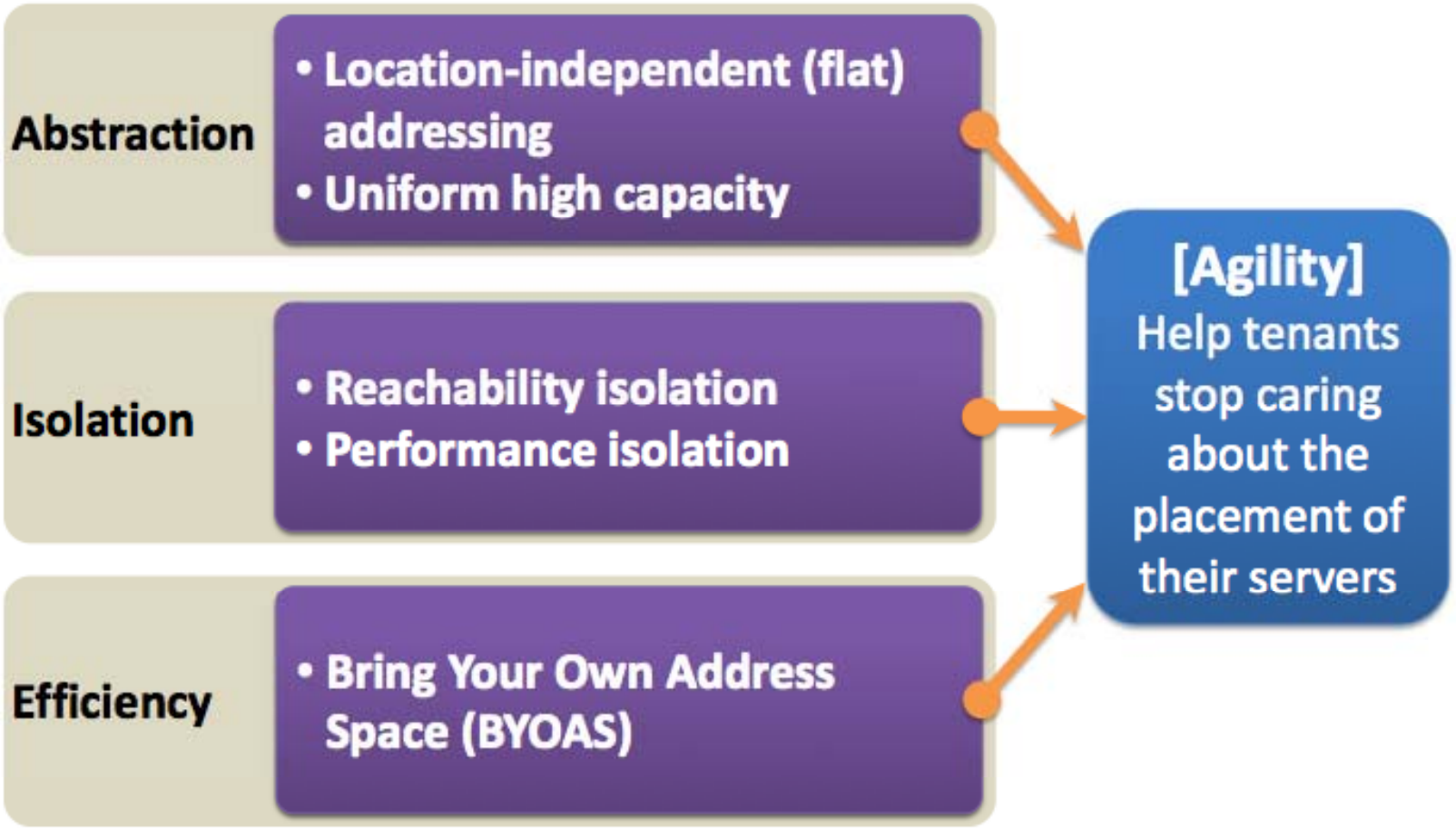
- To network, ***"Support for* Agility"** means

Help tenants stop caring about the placement of their servers

- ✓ **Capability to assign any IP addresses to any servers**
- ✓ **Consistently high networking performance between any servers**
- ✓ **Mechanisms to protect tenants from one another**

Source: Changhoon Kim, Microsoft Research

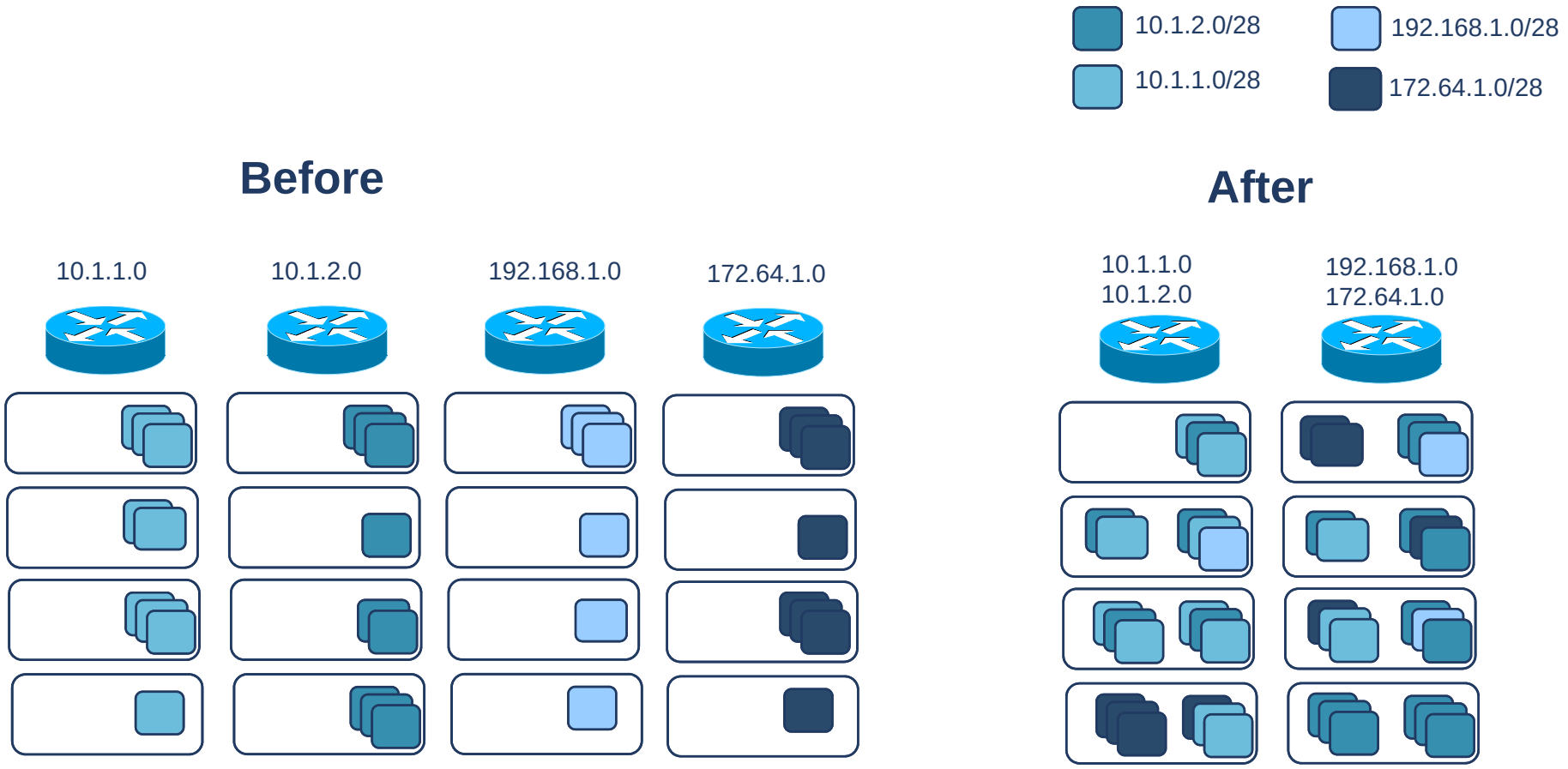
What Is Network Virtualization? And, How Does It Enable Agility?



Source: Changhoon Kim, Microsoft Research

Network Virtualization: Before vs After

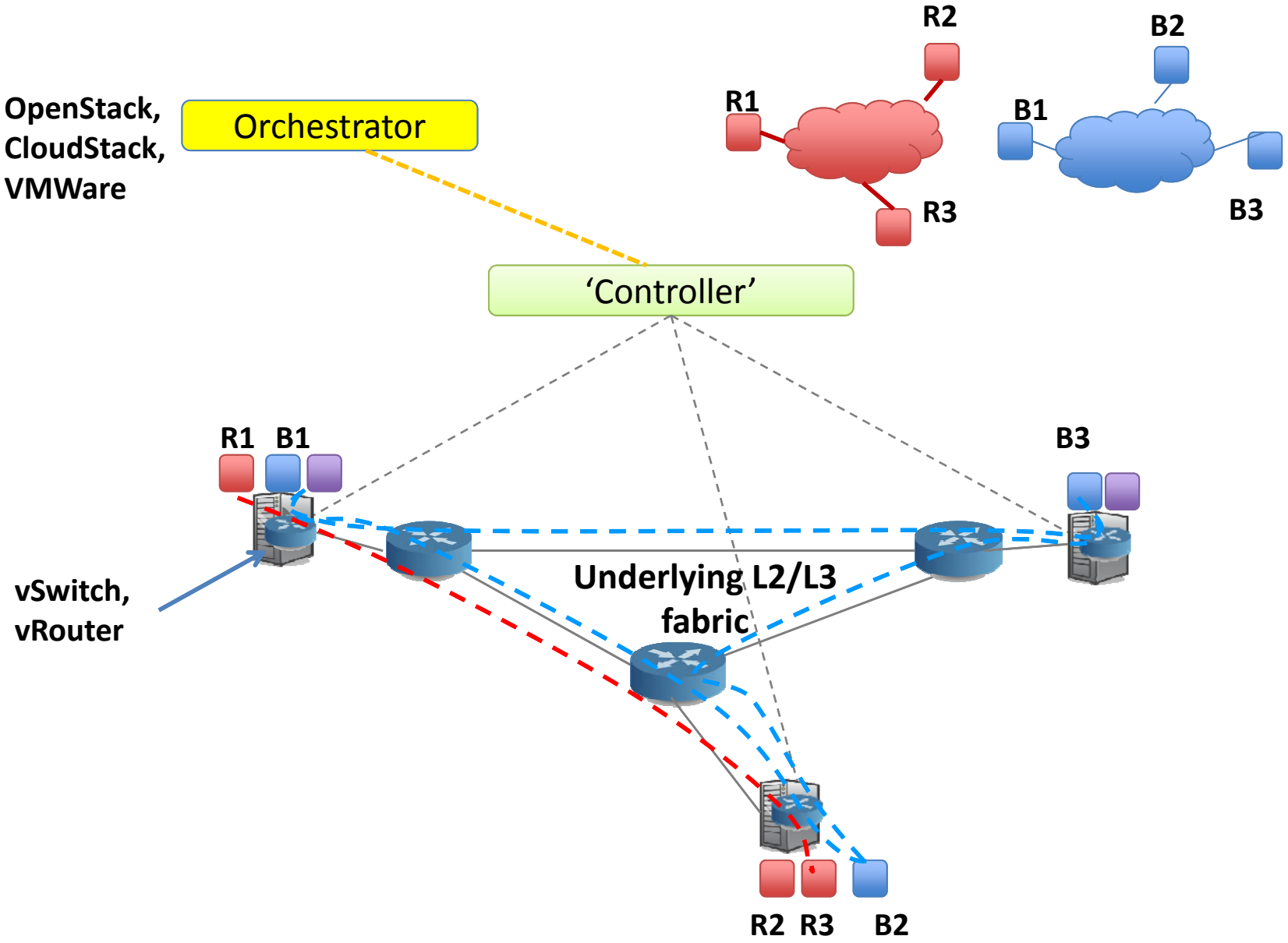
VM density goes up when the network is out of the way



Goal: Maintain a workload's IP addresses and L2-L7 design regardless of where it is physically located in or across datacenters

Source: Big Switch Networks

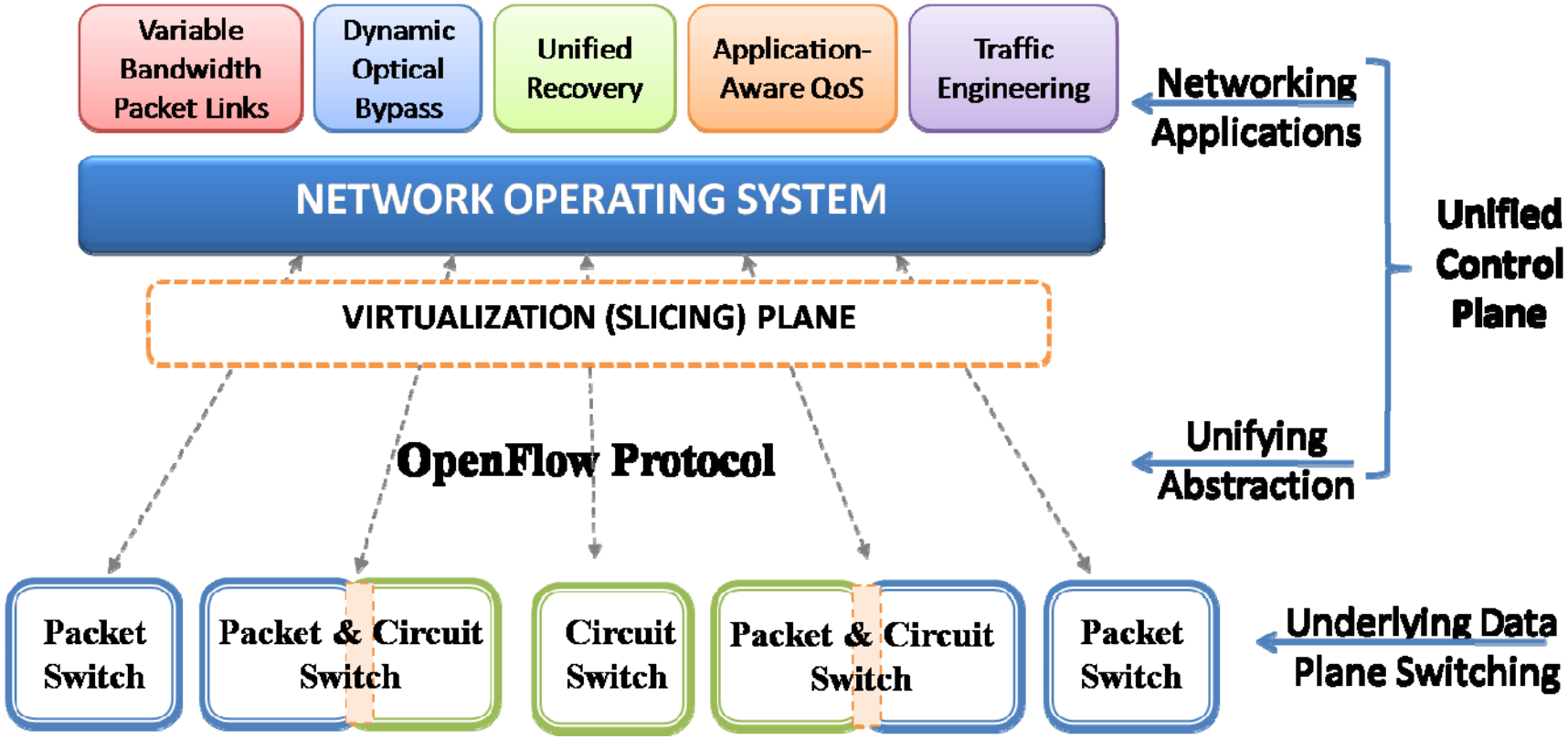
Network Virtualization using Overlays



Demonstration

Network Virtualization

Use Case # 3: IP & Optical Network Convergence



What is the Optical Network good at?

Guarantees:

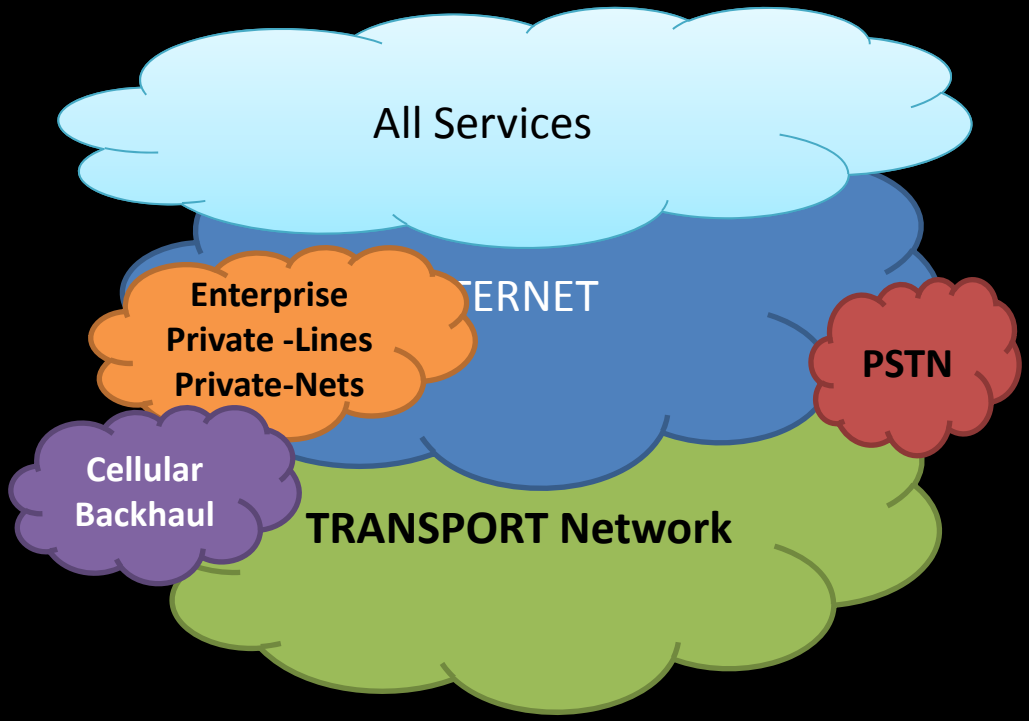
- Bandwidth
- Latency
- Jitter
- Path

Bandwidth on Demand

- Scheduled
- On- Demand

Recovery

The Future?



As much as 60% of AT&T's Transport Network directly or indirectly supports the Internet

A. Gerber, R. Doverspike. "Traffic Types and Growth in Backbone Networks". OFC/NFOEC 2011

What is the Transport Network good at?

Guarantees:

- Bandwidth
- Latency
- Jitter
- Path

Bandwidth on Demand

- Scheduled
- On- Demand

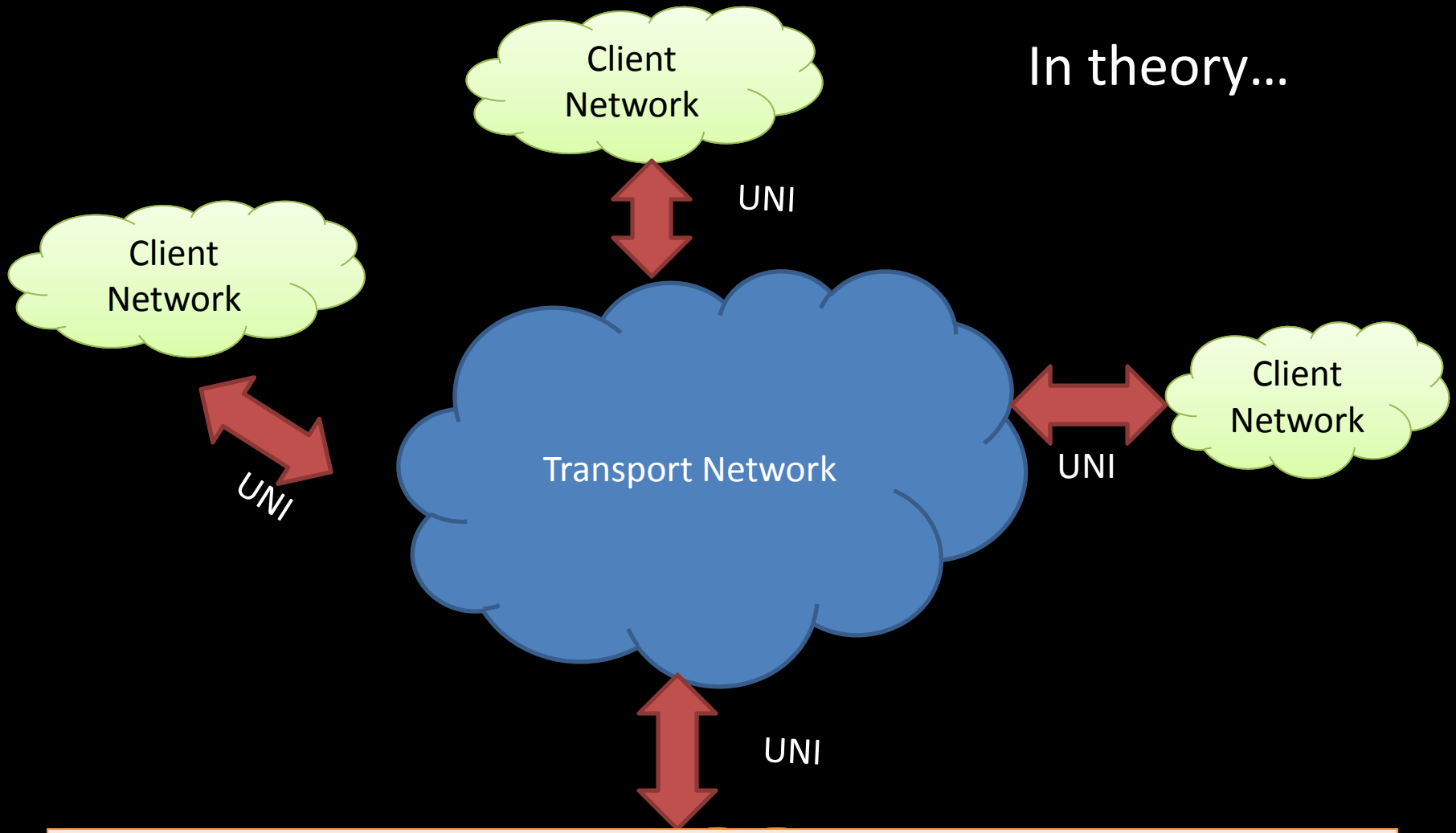
Recovery

What do the IP network vendors want?

-- Give me a Big Fat Dumb Pipe

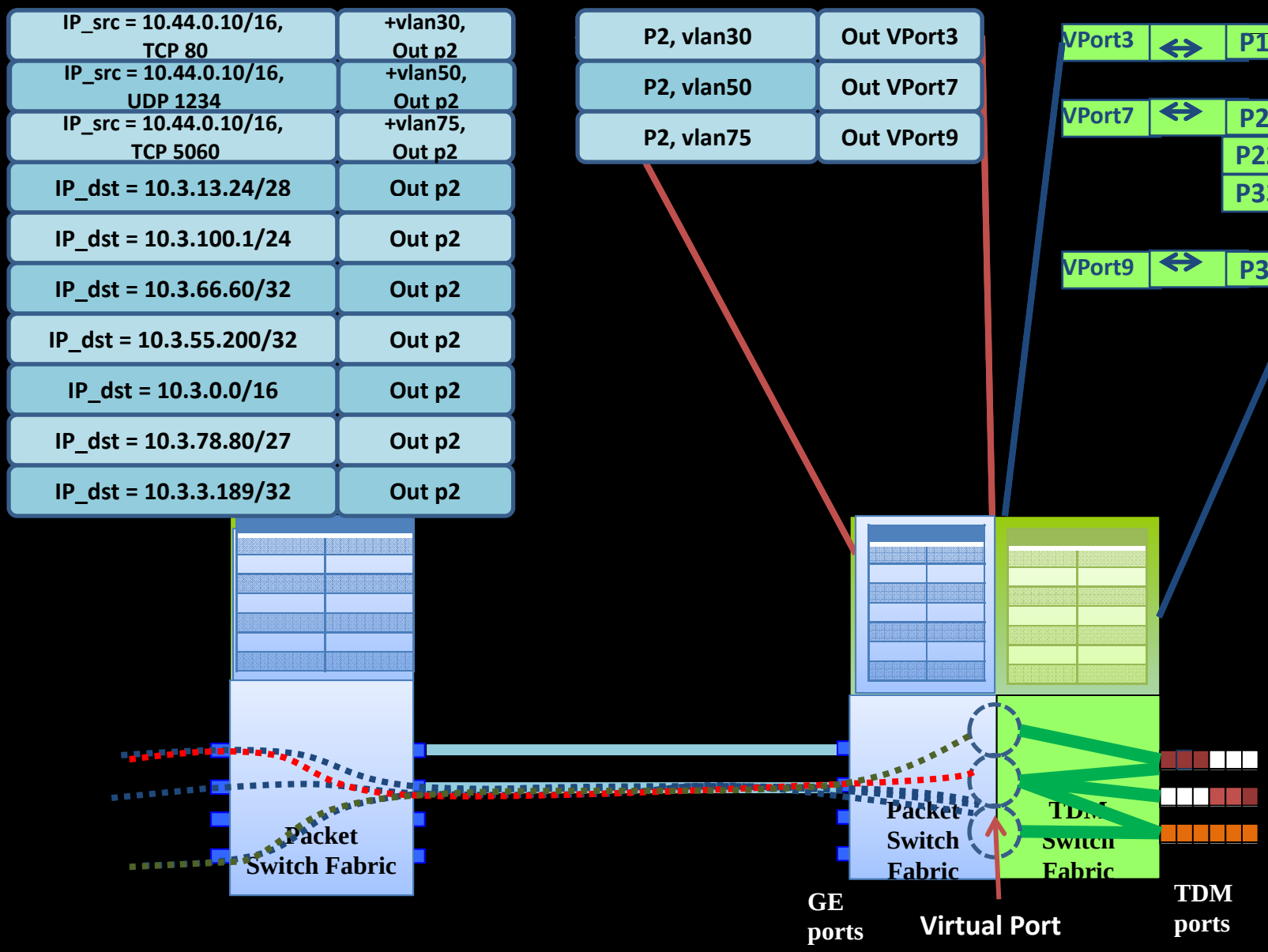
It will cost you!

Carriers want converged operation …



In practice: There is <u>no</u> commercial deployment of an IP network in the world that dynamically interacts with a transport network using UNI/GMPLS – many reasons, but complexity is No. 1

Programming Flow Tables is Simpler

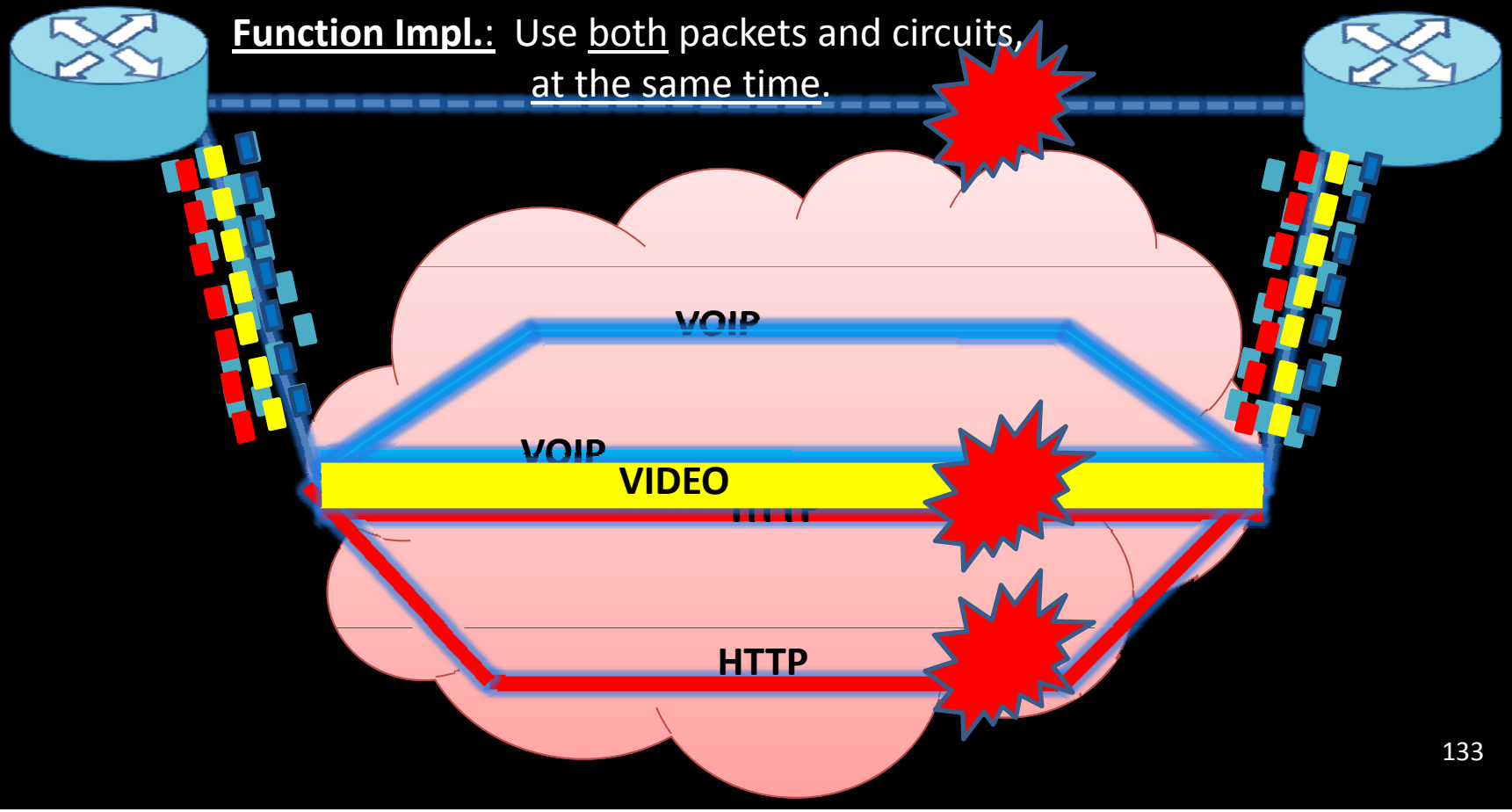


Example Network Application

Control Function: Treat different kinds of traffic differently

Traffic-type	Delay/Jitter	Bandwidth	Recovery
VoIP	Lowest Delay	Low	Medium
Video	Zero Jitter	High	Highest
Web	Best-effort	Medium	Lowest

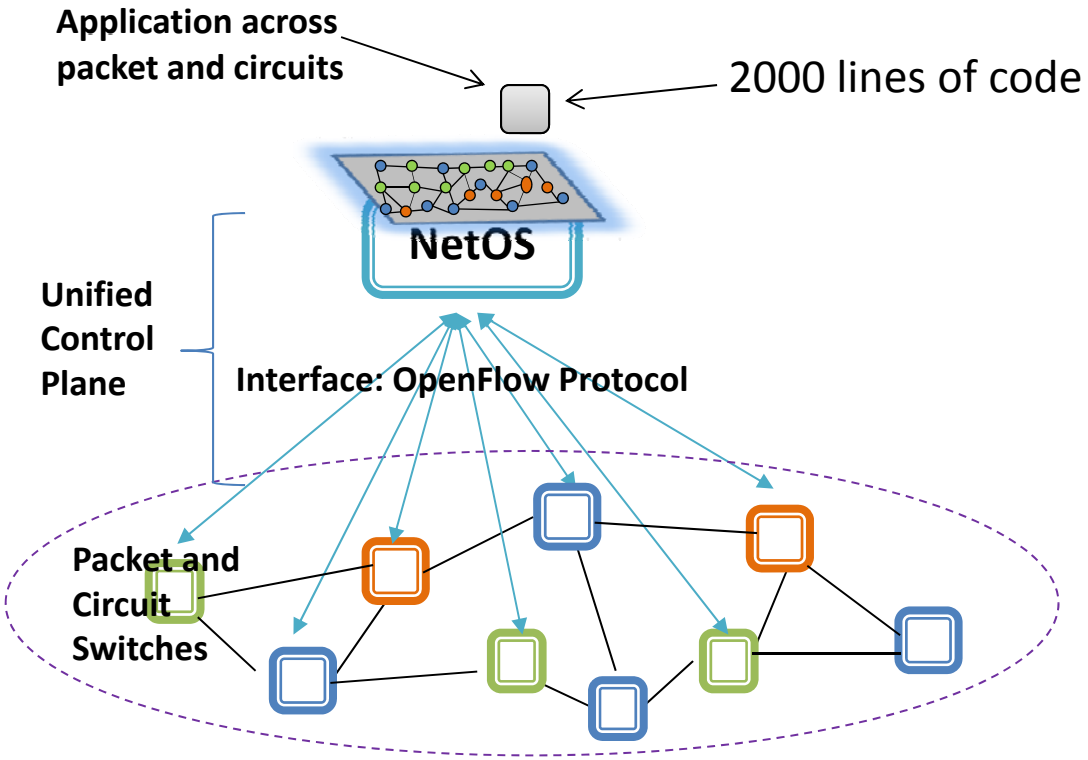
Function Impl.: Use both packets and circuits, at the same time.



Why is it Simpler?

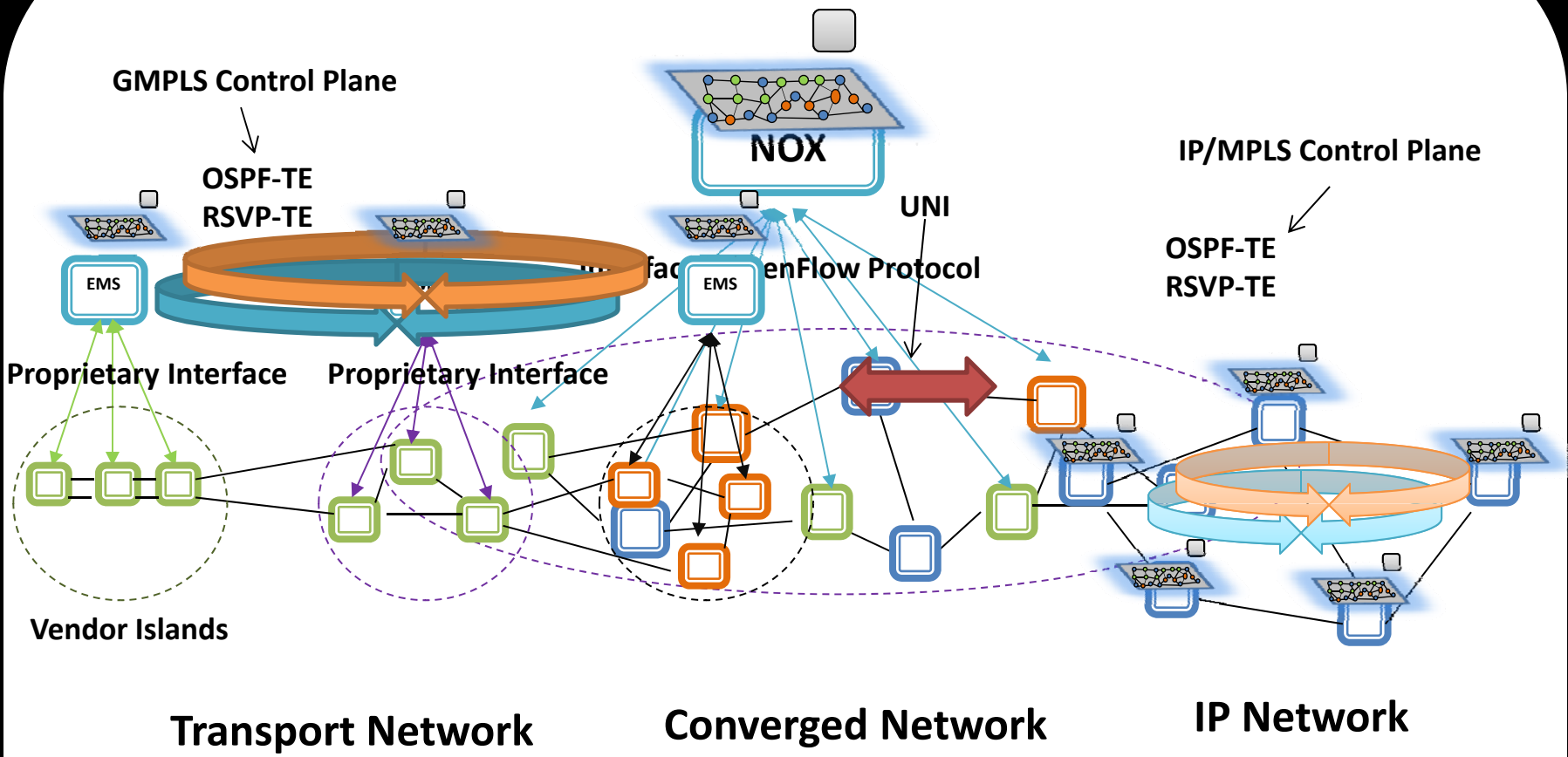
2. **Common Map Abstraction**

1. **Common Flow Abstraction**

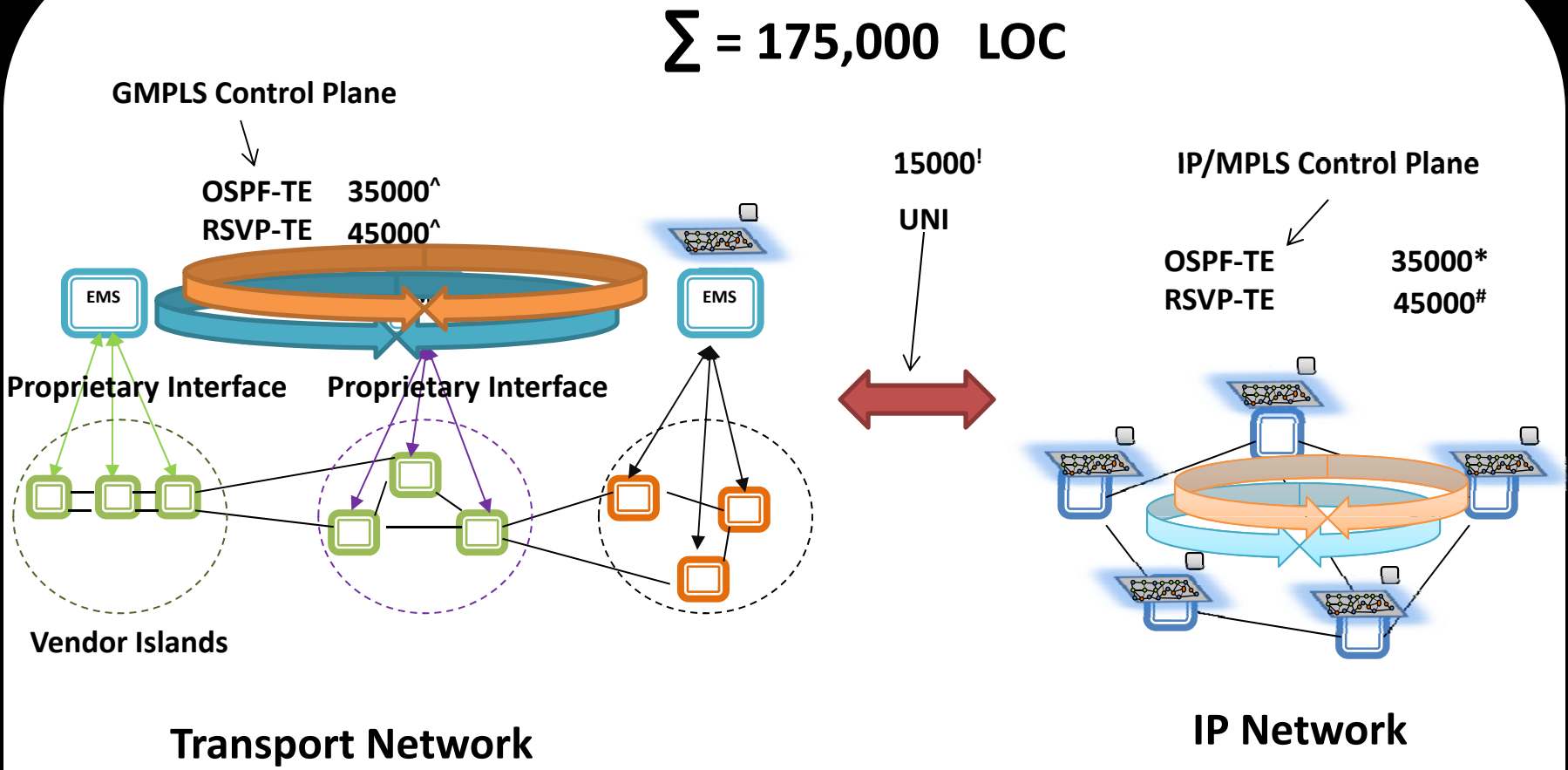


Converged Network

Why is it Simpler?



Why is it Simpler?



Sources: * Quagga # Tequila ! MUPBED ^ DRAGON

Learn more:

See this demo: http://www.youtube.com/watch?v=Yq3yLPtOxpc

Stanford:

• Saurav Das, Guru Parulkar, Nick McKeown, Why OpenFlow/SDN Can Succeed Where GMPLS Failed, European Conference on Optical Communications (ECOC), September 2012
http://archive.openflow.org/wk/images/4/46/OFvsGMPLS_ECOC.pdf

• Saurav Das, Guru Parulkar, Nick McKeown, Rethinking IP Core Networks, Journal of Optical Communications and Networking, December 2013
http://yuba.stanford.edu/~nickm/papers/191340-2a.pdf

• Thesis: http://www.openflow.org/wk/index.php/PACC_Thesis

• Get Involved:
- **• ONF Optical Working Group**

Many Use Cases Emerging…

Talked about these today …

- Traffic Engineering (in production)
- Network Virtualization (in production)
- Packet-Optical Convergence

More Use Cases gaining traction …

- Network Monitoring (in production)
- LTE EPC Scale-out
- Mobile Network
  - traffic management
  - radio management
  - backhaul management
- Enterprise network
  - Policy Management/Load Balancing/Security
- Access networks & Managed Home Networks

Outline

- Module 1: SDN/OpenFlow Introduction
- Module 2: OpenFlow/SDN Deep Dive
- Module 3: SDN Architecture
- Module 4: Operational Choices & Performance
- Module 5: Benefits & Use Cases
  - ❖ Demo: Network Virtualization
- Module 6: Demystifying SDN
- Q & A

Module 6

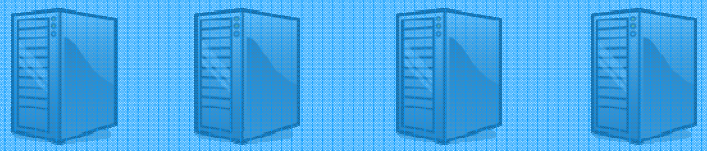
Demystifying SDN

The following 14 slides are courtesy of

SDN ACADEMY

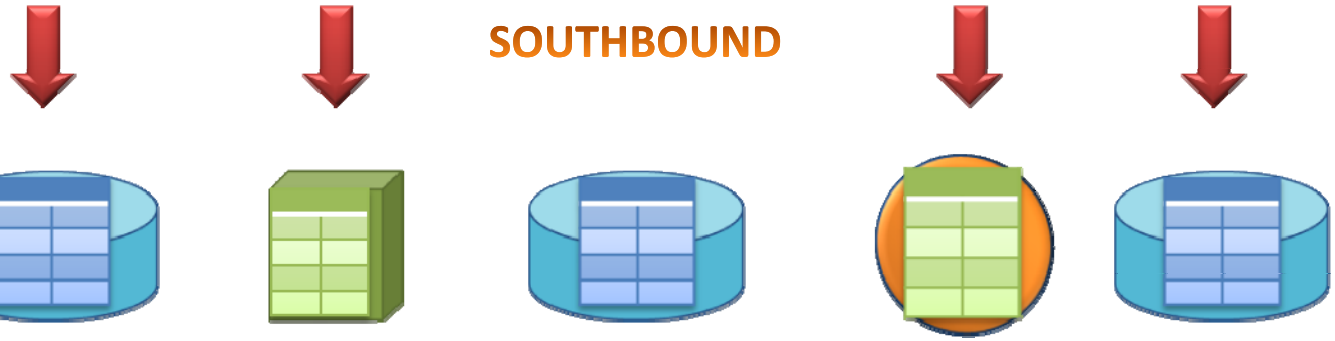
Quick Recap: What is SDN?

(B) State-Distribution Abstraction: Global View



Network OS running on distributed controllers

Configuration; Control over Forwarding; Monitoring



(A) Forwarding Abstraction: Match-Action Tables

© 2013 SDN Academy, LLC™. All Rights Reserved.

What is SDN? – A Simpler View

SDN is the separation of the control plane from the forwarding plane..
Everything else is a benefit that derives from that separation
-- Nick McKeown, OFC 2013 Plenary

Software-Defined Networking (SDN) Definition WHAT IS SDN?
The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.
-- Open Networking Foundation website

"The Future of Networking, and the Past of Protocols",
-- Scott Shenker, ONS2011

© 2013 SDN Academy, LLC™. All Rights Reserved.

Benefits of SDN

1. Simpler Control with Greater Flexibility

- Networks work because we can master complexity, but what we should be doing is extracting simplicity, with the right abstractions

2. Programmability: Speed of Innovation, Ease of Service Insertion & Faster Time to Market

- Does not involve changing/creating a fully distributed protocol

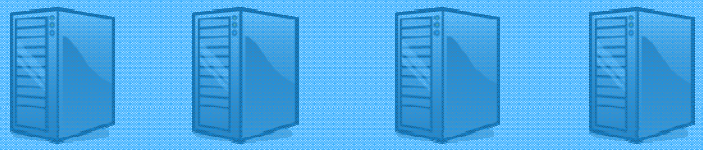
3. Lower Total Cost of Ownership (TCO)

- Lower Opex – easier to manage, troubleshoot, emulate, automate, optimize
- Lower Capex – replacing proprietary hardware, pay for what you need and no more.

© 2013 SDN Academy, LLC™. All Rights Reserved.

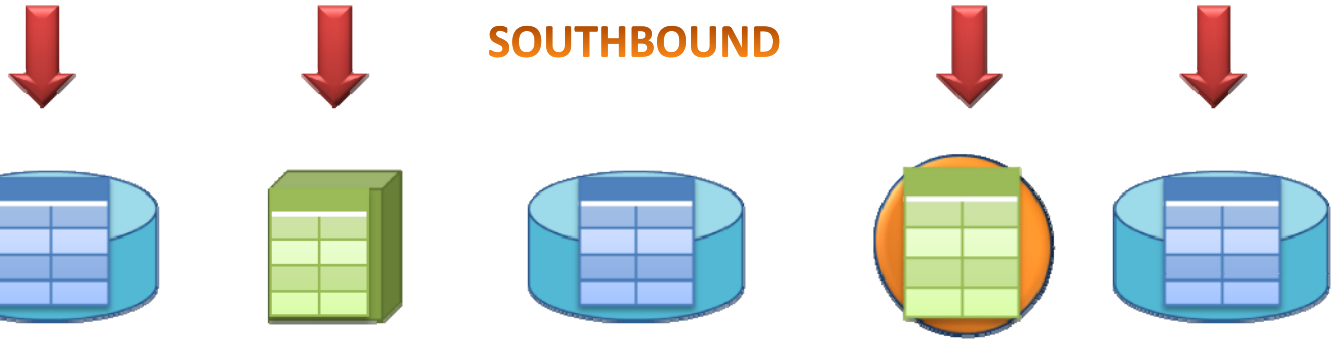
[1] ‘Pure’ SDN / ‘True’ SDN?

(B) State-Distribution Abstraction: Global View



Network OS running on distributed controllers

Configuration; Control over Forwarding; Monitoring



(A) Forwarding Abstraction: Match-Action Tables

Separation
D <---> C

Simplicity

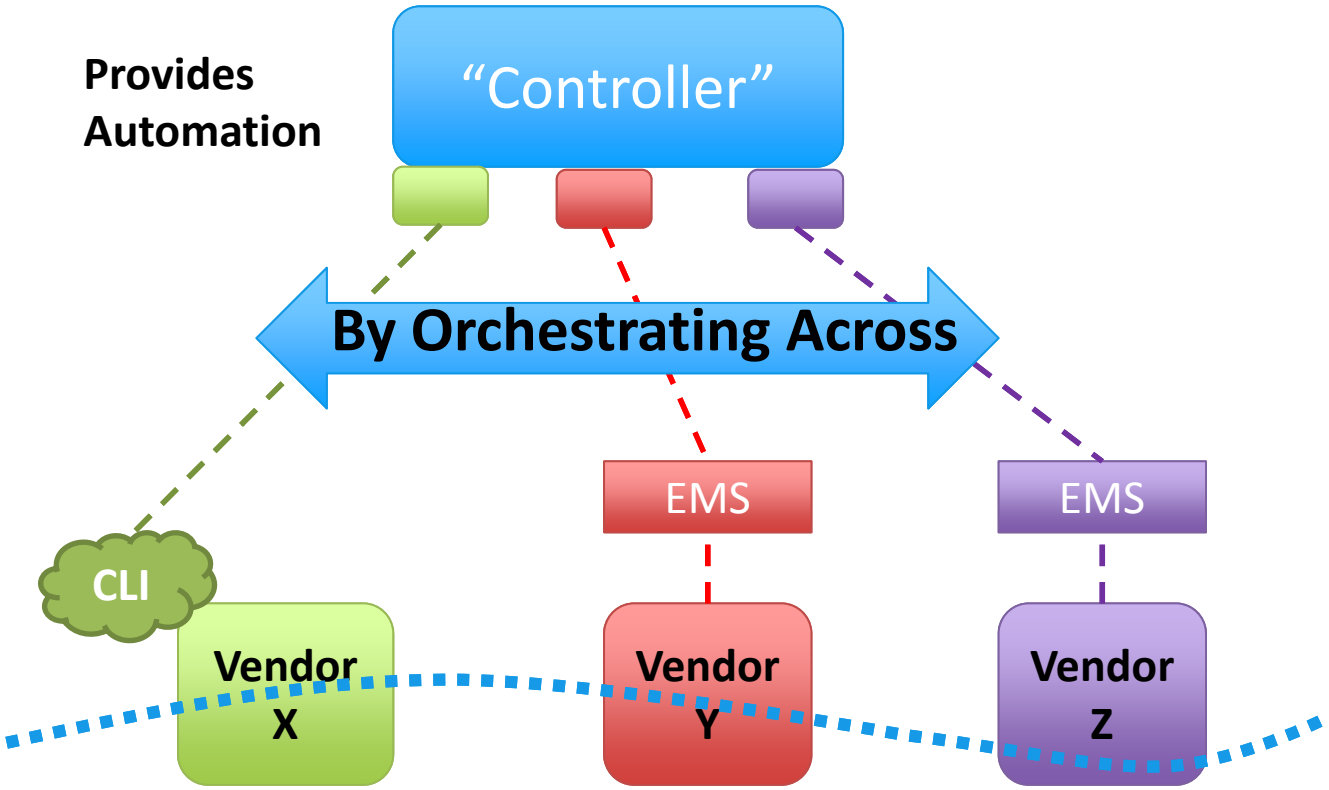
Programmability

Lower Capex

Lower Opex

© 2013 SDN Academy, LLC™. All Rights Reserved.

[2] Orchestration/Automation/Gluework



Automation ≠ **Programmability**

Modified NMS, OpenStack, cli-scripts → automate Configuration & provisioning

Changing Behavior (if-then-else)

Separation D <---> C

Simplicity

Programmability

Lower Capex

Lower Opex

© 2013 SDN Academy, LLC™. All Rights Reserved.

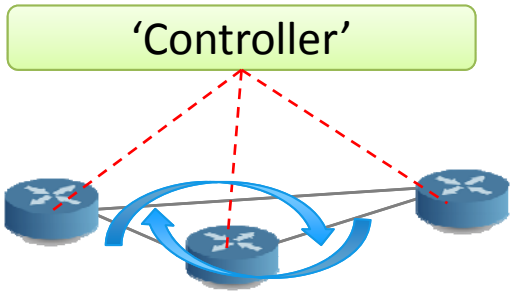
[3] Traditional Networking + 'SDN Hook'

Legacy Preserving with Claimed Advantages →

A. Customer realizes benefits of SDN on same infrastructure

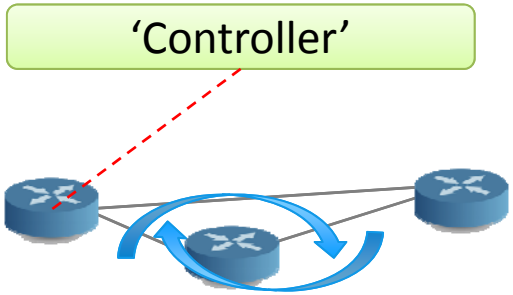
B. Allows vendor to build SDN on existing products

CLI Replacement



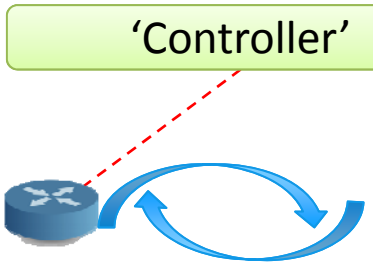
Instead of CLI expect scripts, write python programs

Point Solutions



Exposing ACL tables to override default behavior

Hybrid Solutions / 'Ships-in-the-night'



Some VLANs have SDN control

Separation D <---> C

Simplicity

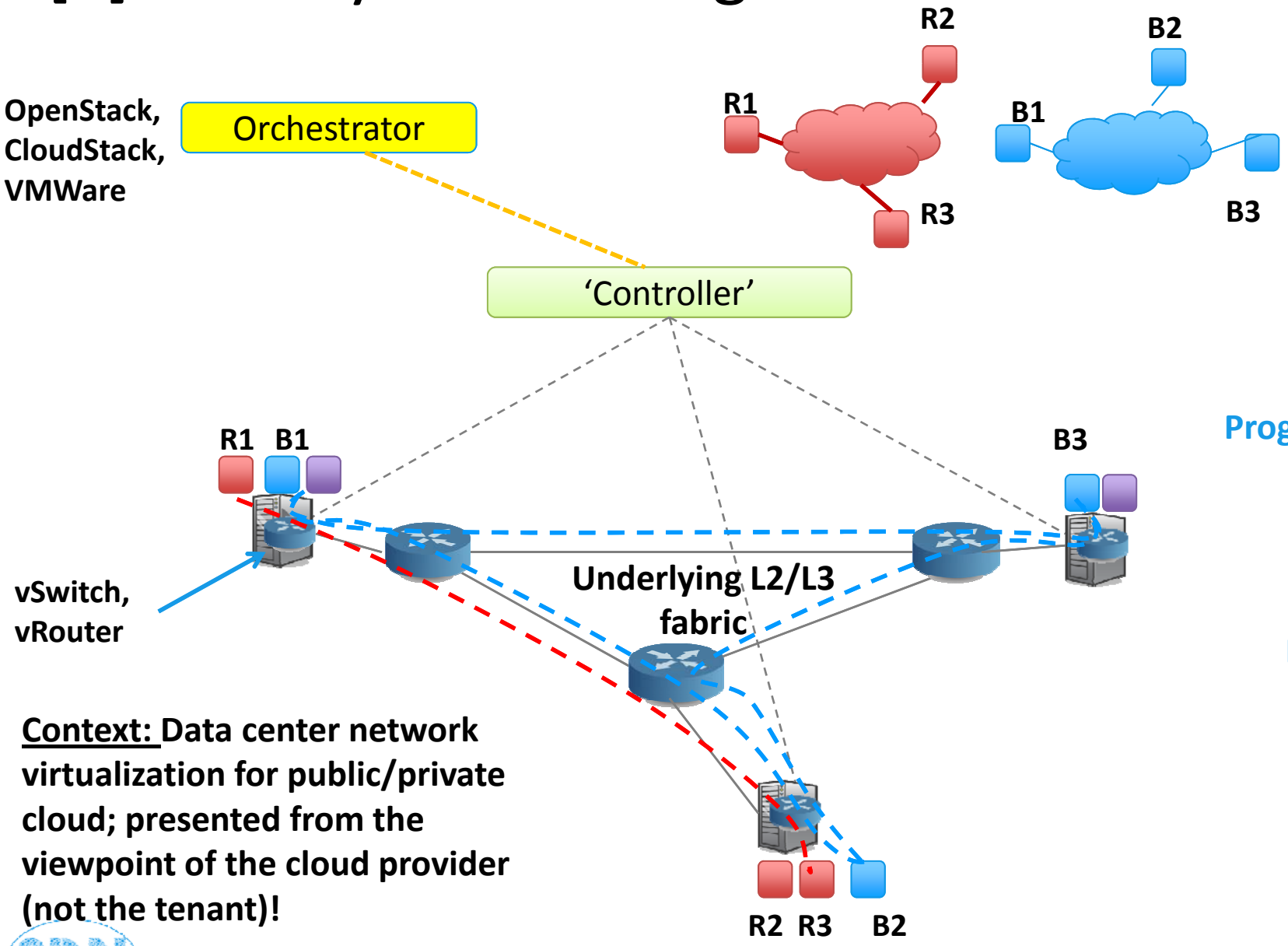
Programmability

Lower Capex

Lower Opex

© 2013 SDN Academy, LLC™. All Rights Reserved.

[4] Overlay Networking



Separation
D <---> C

Simplicity

Programmability

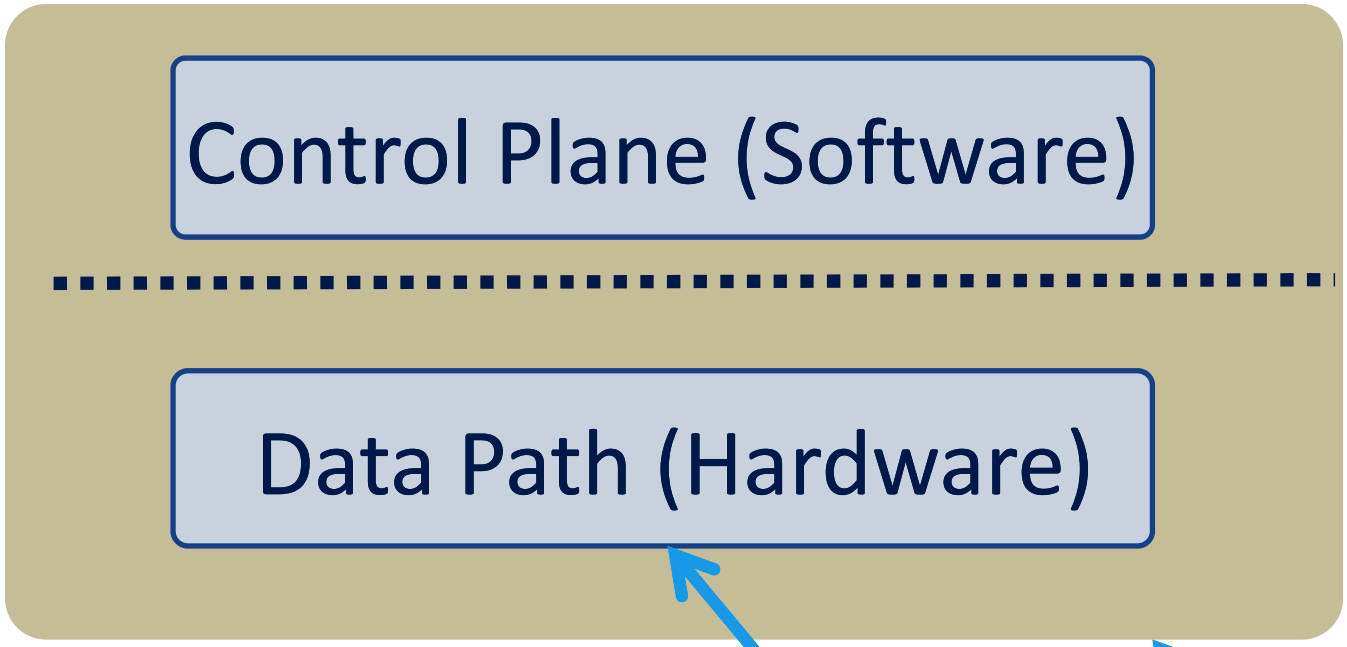
Lower Capex

Lower Opex

Context: Data center network virtualization for public/private cloud; presented from the viewpoint of the cloud provider (not the tenant)!

© 2013 SDN Academy, LLC™. All Rights Reserved.

[5] Traditional Networking + White Box



Separation
D <---> C

Simplicity

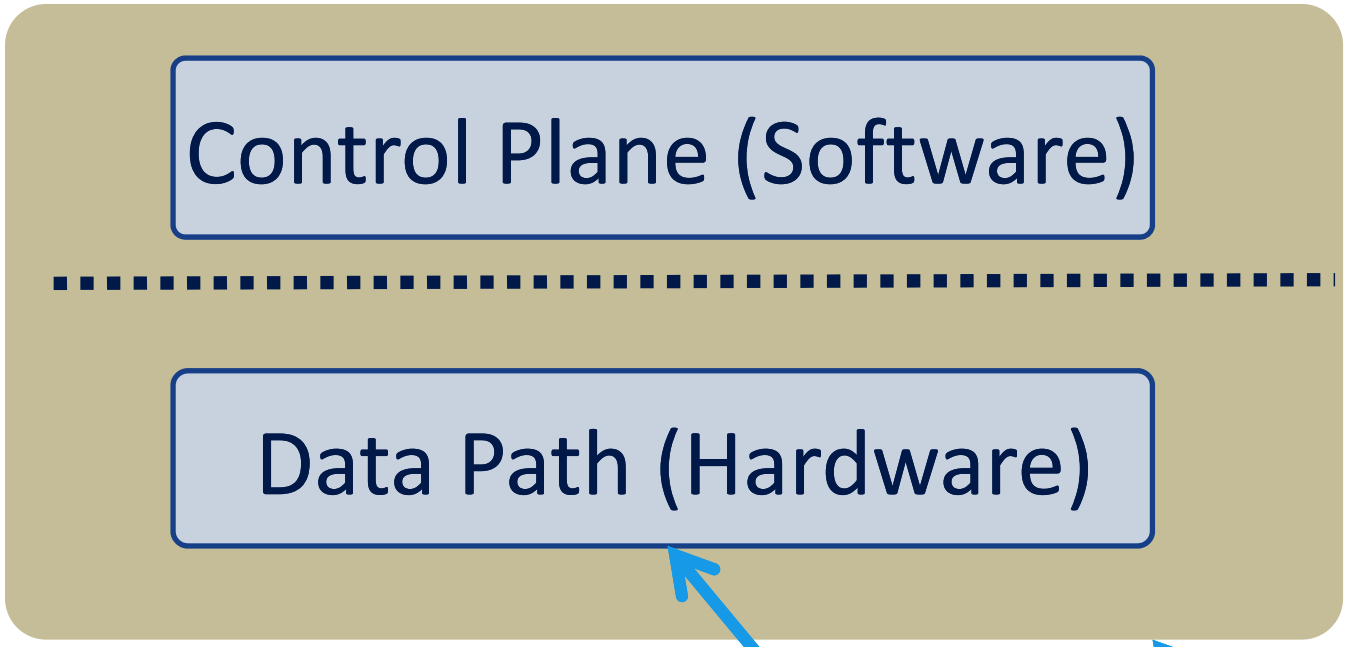
Programmability

Lower Capex

Lower Opex

© 2013 SDN Academy, LLC™. All Rights Reserved.

[5.5] Traditional Networking + White Box + Orchestration



Separation
D <---> C

Simplicity

Programmability

Lower Capex

Lower Opex

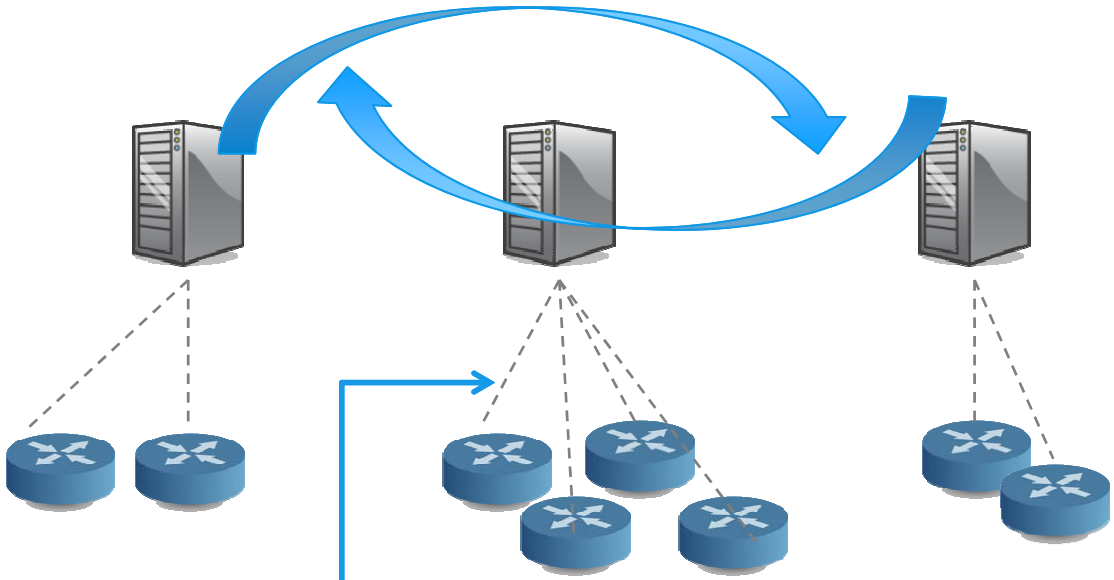
© 2013 SDN Academy, LLC™. All Rights Reserved.

[6] Decoupled Traditional Networking

Separation
D <---> C

Simplicity

Traditional networking control plane



1. Software stacks

2. Merchant Silicon

Usually proprietary, could be OF, See IETF ForCES

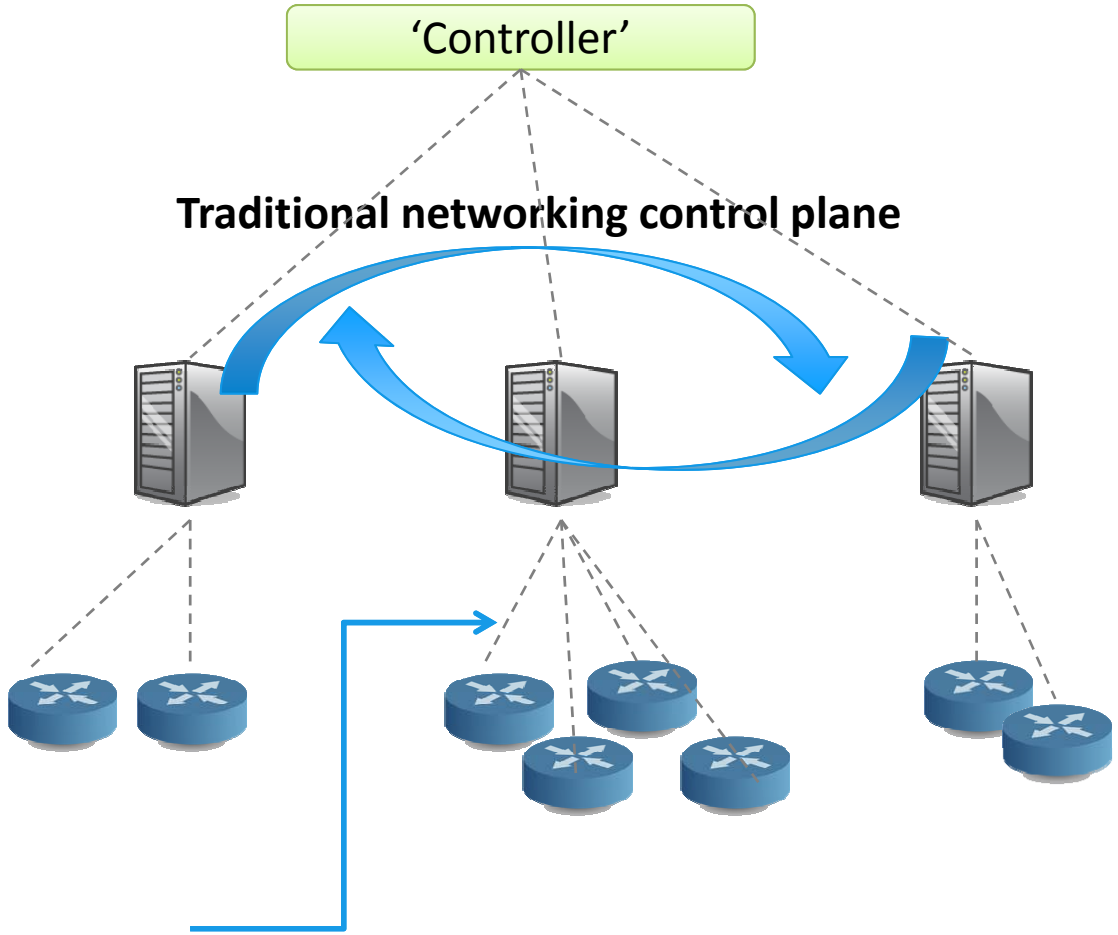
Programmability

Lower Capex

Lower Opex

© 2013 SDN Academy, LLC™. All Rights Reserved.

[6.5] Decoupled Traditional Networking + Global View



1. **Software stacks**
2. **Merchant Silicon**

Usually proprietary, could be OF, See IETF ForCES

Separation D <---> C

Simplicity

Programmability

Lower Capex

Lower Opex

© 2013 SDN Academy, LLC™. All Rights Reserved.

[7] Open Source Networking

Has Open Source ever worked in networking?

What are the best know open source networking projects…

- **Quagga**
- **OVS**

… not used in production without modification

What about controllers?:

- **No open source controller used in production**
- **No open source networking project today is complete solution**
- **Open source is part of the solution**

Separation
D <---> C

Simplicity

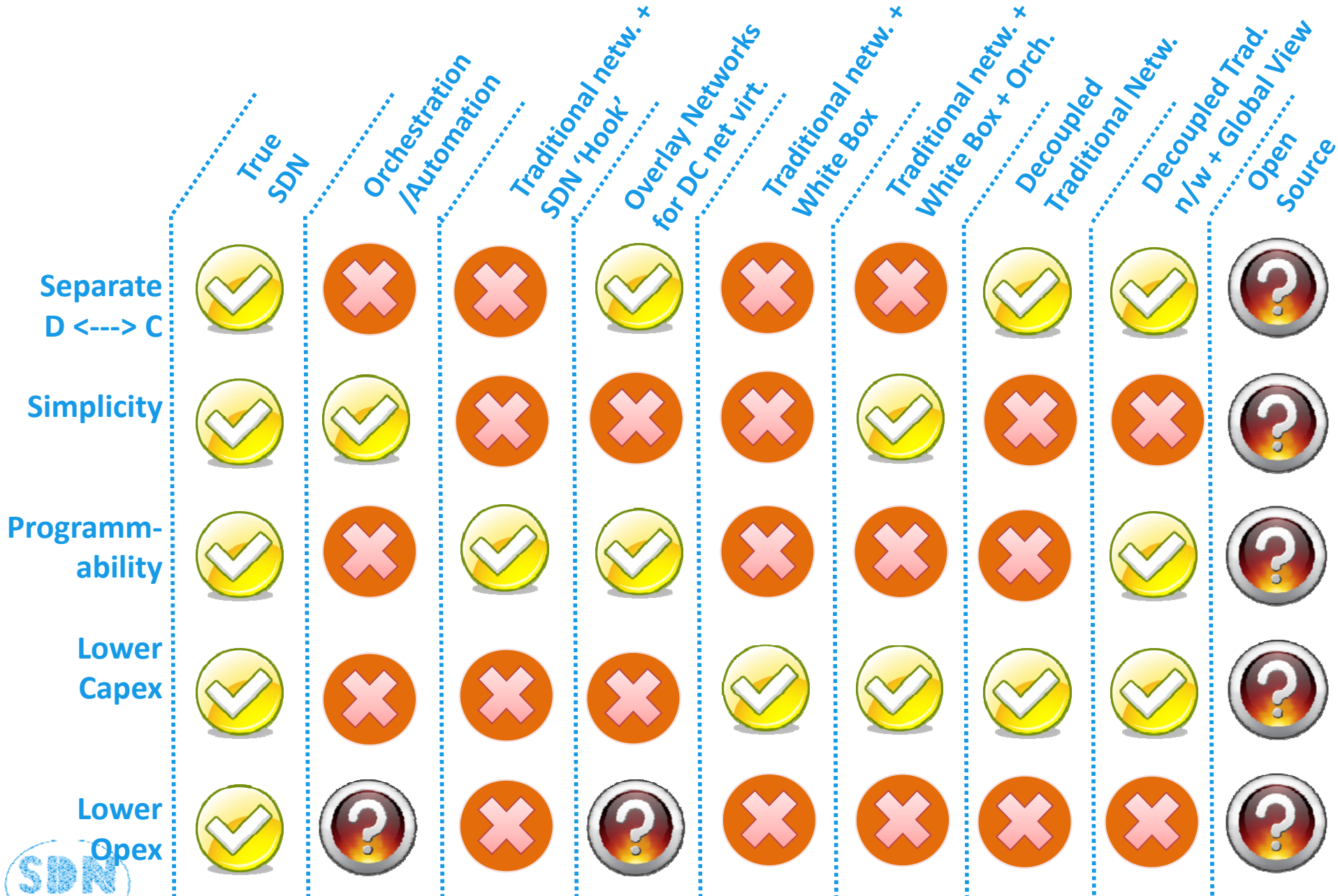
Programmability

Lower Capex

Lower Opex

© 2013 SDN Academy, LLC™. All Rights Reserved.

Shades of SDN



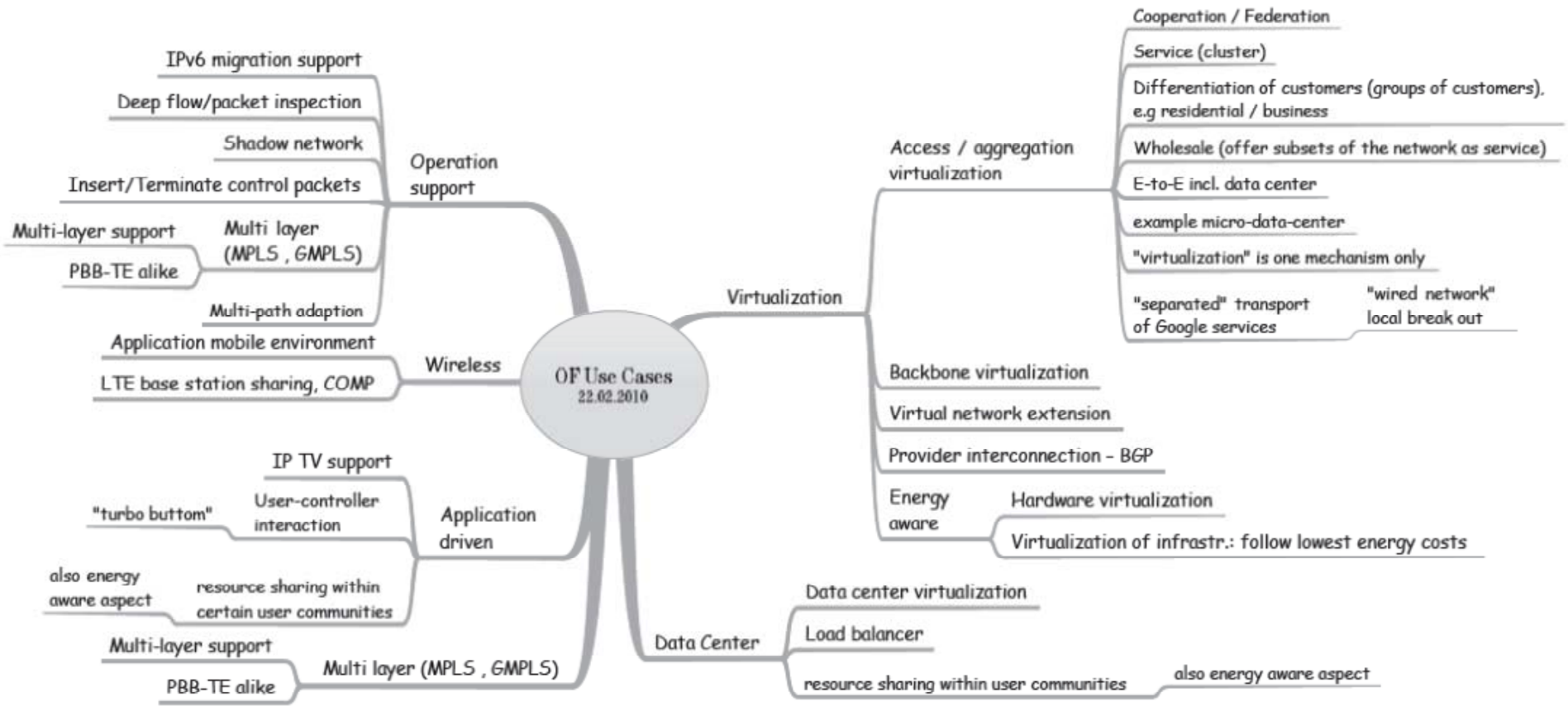
© 2013 SDN Acacemy, LLC™. All Rights Reserved.

Other Misconceptions

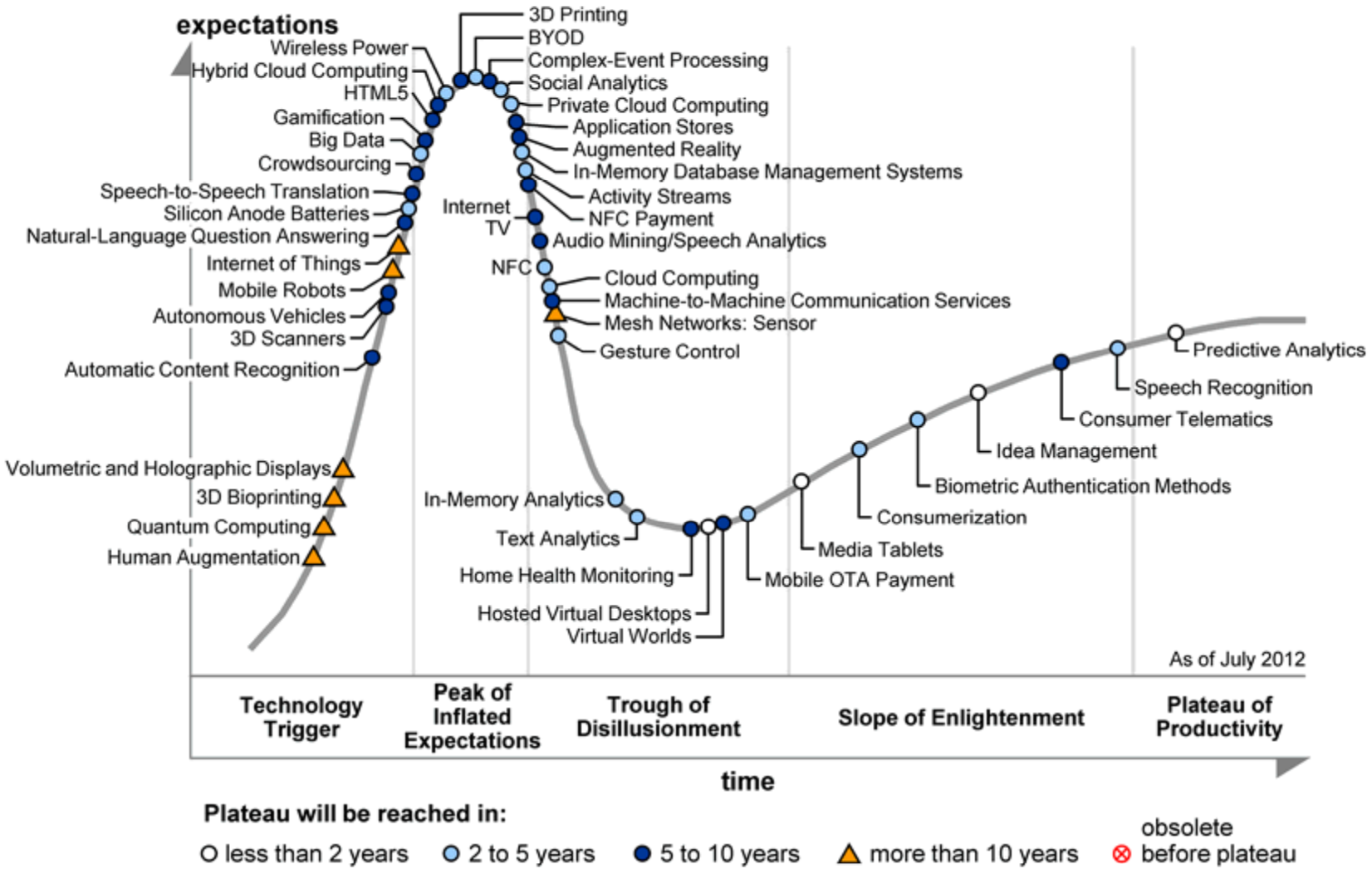
- SDN is Network Virtualization
- SDN == NFV
- SDN is about centralized control
- OpenFlow is not mature
- OpenFlow does not scale
- Do we really need another protocol?

© 2013 SDN Academy, LLC™. All Rights Reserved.

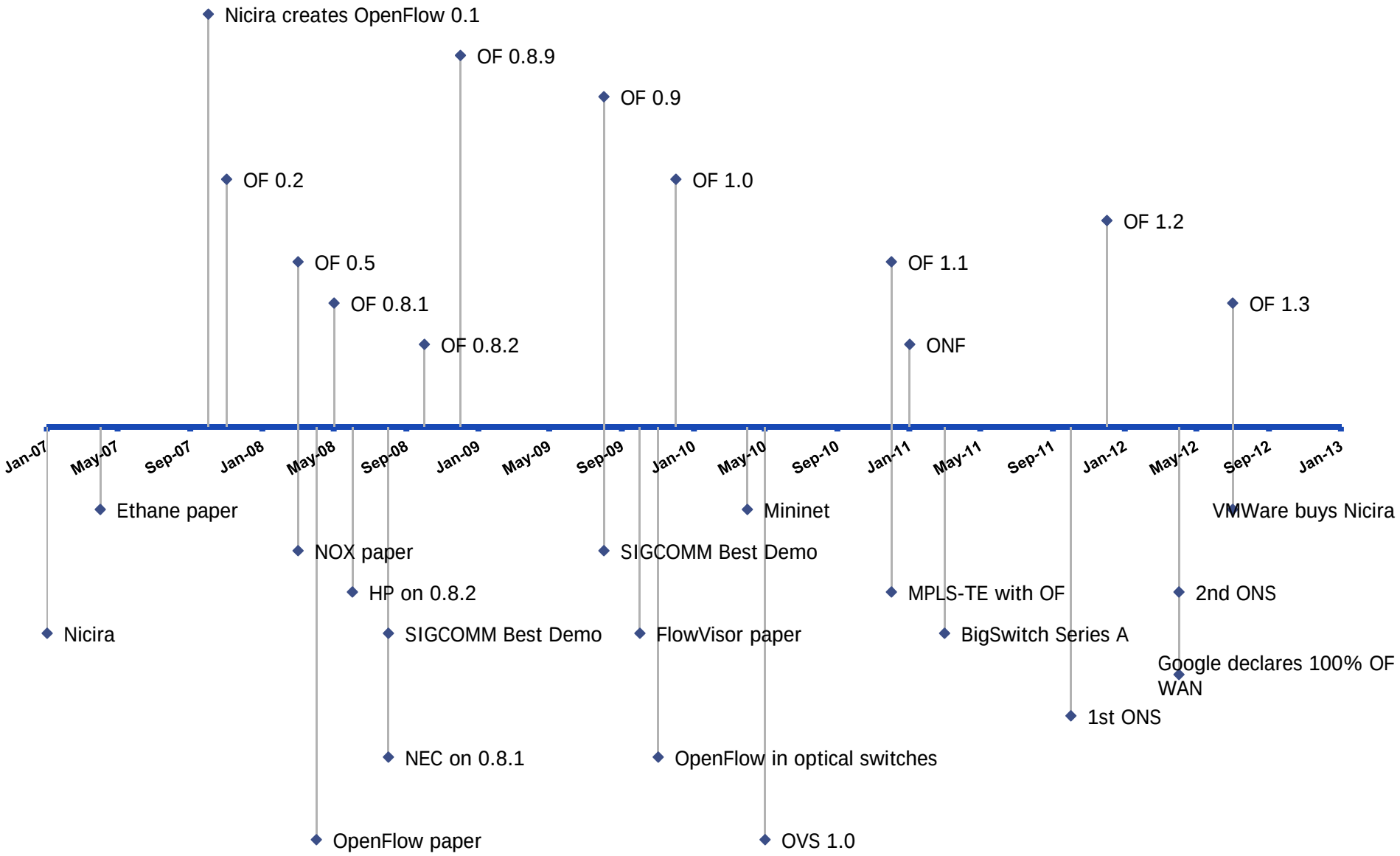
SDN is a hammer for what nail?



Gartner Hype Cycle



OpenFlow/SDN @ Stanford



SDN had been Around for Quite a While

Google

Ipsilon GSMP	1996
Cambridge's The Tempest	1998
IETF FORCES	2000
IETF PCE	2004
Princeton's Routing Control Platform	2004
4d Initiative	2005
Ethane	2007
Openflow	2008

Source: Ed Crabbe, Google

SDN Opportunities

Google

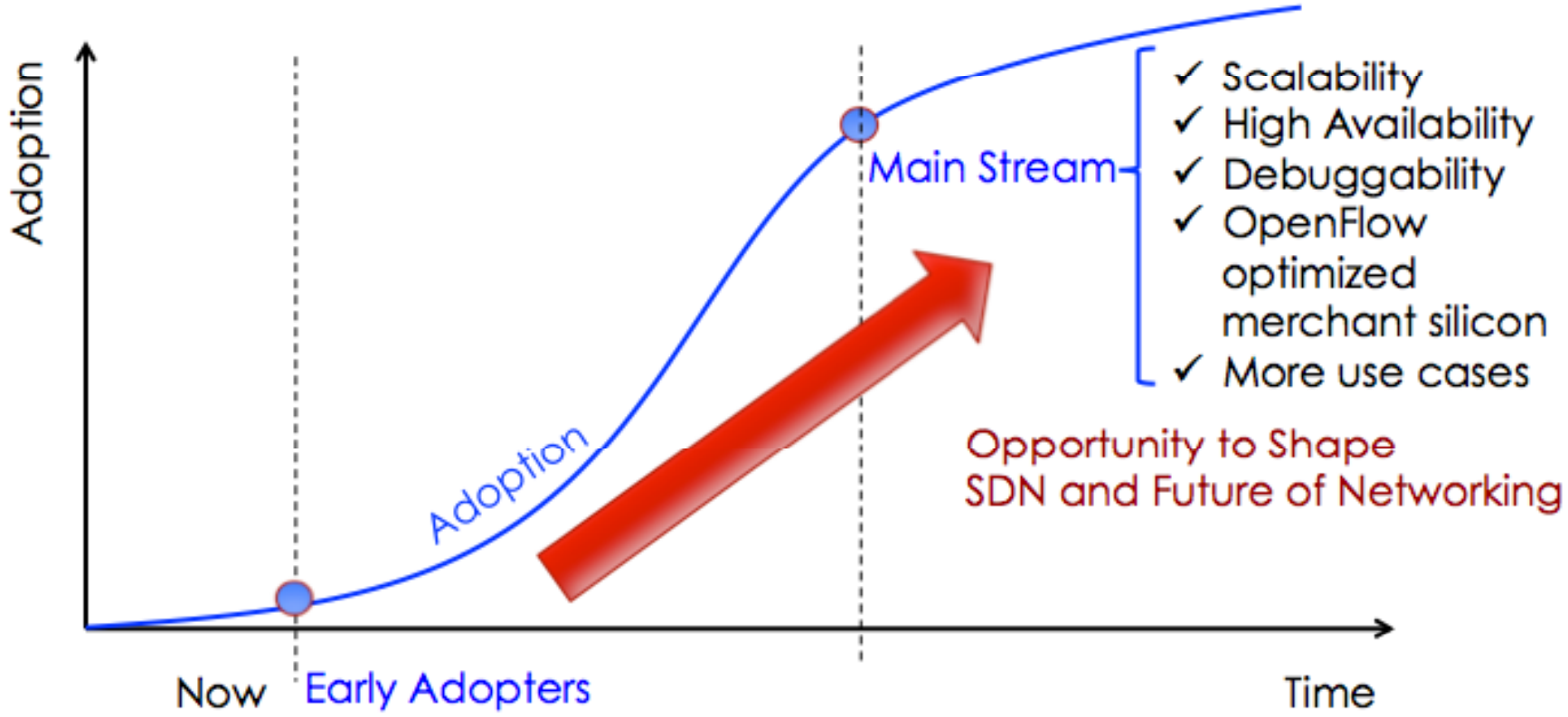
And yet all of SDN is in it's infancy:

1. Controller ↔ Switch abstractions
   south-bound
2. Controller ↔ Application abstractions
   north-bound
3. Controller ↔ Controller abstractions
   east-west
4. Applications

Source: Ed Crabbe, Google

Where is SDN right now?

- SDN definitely represents a new paradigm of networking
- Market Size: $25-30B in 2018, 30-40% of total spend in networking
- SDN still requires many key capabilities to become main stream



Source: Guru Parulkar

Scaling of SDN Innovation

ONRC
ON.LAB

Build strong intellectual foundation
Bring open source SDN tools/platforms to community

ONF

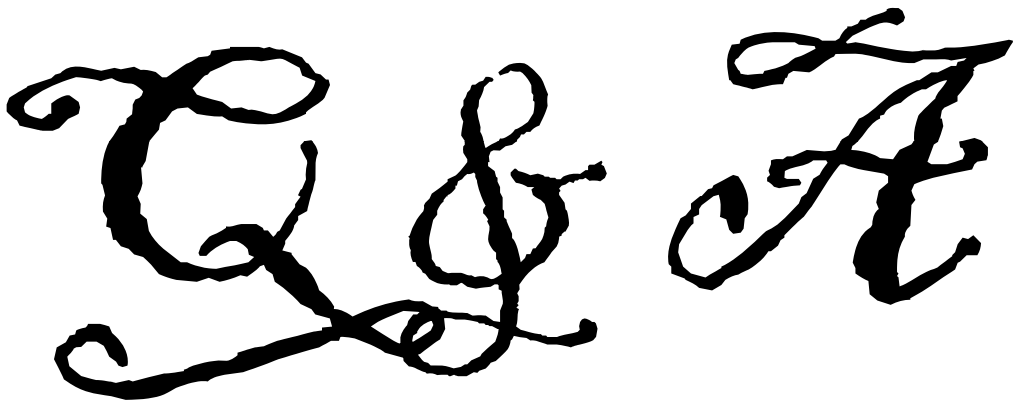
Standardize OpenFlow and promote SDN
100+ Members from all parts of the industry

ONS
OPEN NETWORKING SUMMIT

Bring best SDN content; facilitate high quality dialogue
3 successive sold out events; participation of ecosys

SDN ACADEMY

Bring best SDN training to companies
to accelerate SDN development and adoption

Source: Guru Parulkar

Thanks!