Introduction to OpenFlow

Raj Jain

Washington University in Saint Louis

Saint Louis, MO 63130

Jain@cse.wustl.edu

These slides and audio/video recordings of this class lecture are at:

http://www.cse.wustl.edu/~jain/cse570-23/

Overview

1. Planes of Networking
2. OpenFlow
3. OpenFlow Operation
4. OpenFlow Switches, including Open vSwitch
5. OpenFlow Evolution
6. Current Limitations and Issues

Note: This is the first module of four modules on OpenFlow, OpenFlow Controllers, SDN, and NFV in this course.

Student Questions

http://www.cse.wustl.edu/~jain/cse570-23/

©2023 Raj Jain

Planes of Networking

❑ **Data Plane**: All activities involving as well as resulting from data packets sent by the end user, e.g.,

- ➢ Forwarding
- ➢ Fragmentation and reassembly
- ➢ Replication for multicasting

❑ **Control Plane**: All activities that are <u>necessary</u> to perform data plane activities but do not involve end-user data packets

- ➢ Making routing tables
- ➢ Setting packet handling policies (e.g., security)
- ➢ Base station beacons announcing the availability of services

Student Questions

Ref: Open Data Center Alliance Usage Model: Software Defined Networking Rev 1.0,"
http://www.opendatacenteralliance.org/docs/Software_Defined_Networking_Master_Usage_Model_Rev1.0.pdf

Planes of Networking (Cont)

❑ **Management Plane**: All activities related to provisioning and monitoring of the networks

- ➢ Fault, Configuration, Accounting, Performance, and Security (**FCAPS**).

- ➢ Instantiate new devices and protocols (Turn devices on/off)

- ➢ <u>Optional</u> ⟹ May be handled manually for small networks.

❑ **Services Plane**: Middlebox services to improve performance or security, e.g.,

- ➢ Load Balancers, Proxy Service, Intrusion Detection, Firewalls, SSL Off-loaders

- ➢ Optional ⟹ Not required for small networks

Student Questions

❑ Are all four planes in layer 2.5?

The planes are not layered. The data plane includes all seven layers. Other planes are parallel to it.
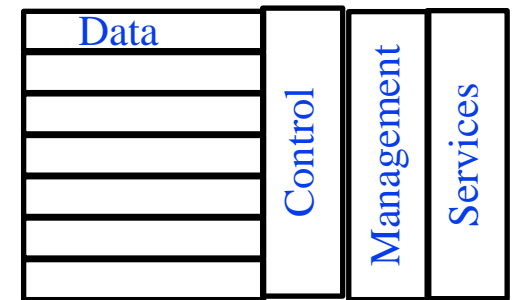
Data vs. Control Logic

❑ Data plane runs at a line rate,
e.g., 100 Gbps for 100 Gbps Ethernet ⇒ Fast Path
⇒ Typically implemented using special hardware,
e.g., Ternary Content Addressable Memories (TCAMs)

❑ Some exceptional data plane activities are handled by the CPU
in the switch ⇒ Slow path
e.g., Broadcast, Unknown, and Multicast (BUM) traffic

❑ All control activities are generally handled by the CPU

http://www.cse.wustl.edu/~jain/cse570-23/
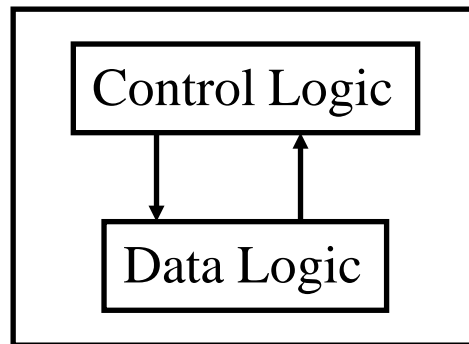
Student Questions

OpenFlow: Key Ideas

1. Separation of control and data planes
2. Centralization of control
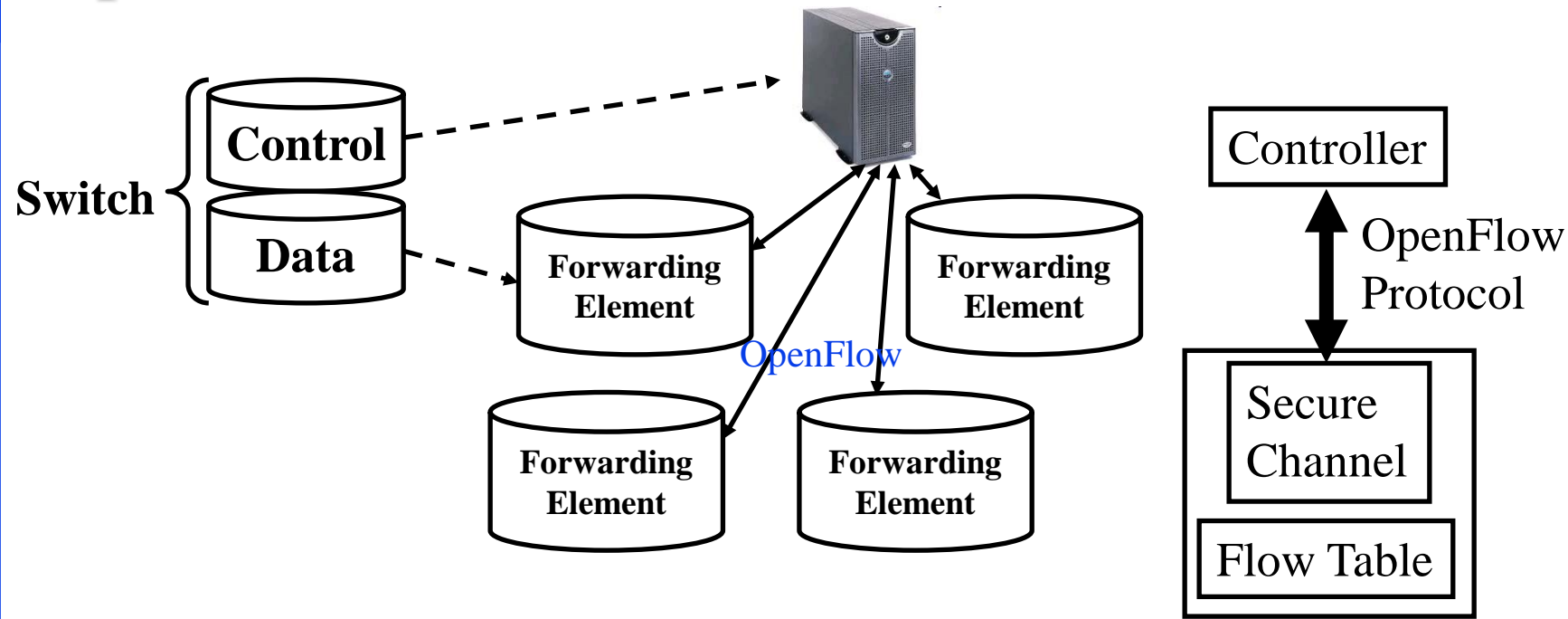3. Flow-based control

Student Questions

Ref: N. McKeown, et al., ``OpenFlow: Enabling Innovation in Campus Networks,'' ACM SIGCOMM CCR, Vol. 38, No. 2, April 2008, pp. 69-74.

Separation of Control and Data Plane

Switch { Control Data }

Forwarding Element

Forwarding Element

OpenFlow

Forwarding Element

Forwarding Element

Controller

OpenFlow Protocol

Secure Channel

Flow Table

- Control logic is moved to a controller
- Switches only have forwarding elements
- One expensive controller with a lot of cheap switches
- OpenFlow is the protocol to send/receive forwarding rules from the controller to switches

OpenFlow V1.0

❑ On packet arrival, match the header fields with flow entries in a table; if any entry matches, update the counters indicated in that entry and perform indicated actions

Flow Table:

Header Fields	Counters	Actions
Header Fields	Counters	Actions
…	…	…
Header Fields	Counters	Actions

Ingress Port	Ether Source	Ether Dest	VLAN ID	VLAN Priority	IP Src	IP Dst	IP Proto	IP ToS	Src L4 Port	Dst L4 Port

Student Questions

14.8

Flow Table Example

Port	Src MAC	Dst MAC	VLAN ID	Priority	EtherType	Src IP	Dst IP	IP Proto	IP ToS	Src L4 Port / ICMP Type	Dst L4 Port / ICMP Code	Action	Counter
*	*	0A:C8:*	*	*	*	*	*	*	*	*	*	Port 1	102
*	*	*	*	*	*	*	192.168.*.*	*	*	*	*	Port 2	202
*	*	*	*	*	*	*	*	*	*	21	21	Drop	420
*	*	*	*	*	*	*	*	0x806	*	*	*	Local	444
*	*	*	*	*	*	*	*	0x1*	*	*	*	Controller	1

❑ Idle timeout: Remove entry if no packets are received for this time

❑ Hard timeout: Remove entry after this time

❑ If both are set, the entry is removed if either expires.

http://www.cse.wustl.edu/~jain/cse570-23/

Student Questions

❑ Can you please explain the difference between idle and hard timeout? *Sure*

❑ Since each control node and forwarding node needs to maintain many information flow tables, will there be some robustness problems?

Yes, things can be out of sync for a little while.

Matching

Set Input Port
Ether Src
Ether Dst
Ether Type
Set all others to zero

EtherType =0x8100? — Y →
Set VLAN ID
Set VLAN Priority
Use EtherType in VLAN tag
for next EtherType Check

N

EtherType =0x0806? — Y →
Set IP Src, IP Dst
IP Proto, IP ToS
from within ARP

N

EtherType =0x0800? — Y →
Set IP Src, IP Dst
IP Proto, IP ToS

N

Match Table 0? — Y → Apply Actions

N

Match Table n? — Y → Apply Actions

N → Send to Controller

Not IP Fragment? — Y → IP Proto =6 or 7 — Y →
Set Src Port, Dst Port for L4 fields

N

N

IP Proto =1? — Y →
Use ICMP Type and code for L4 Fields

N

Packet lookup using assigned header fields

http://www.cse.wustl.edu/~jain/cse570-23/

©2023 Raj Jain

Student Questions

❑ Could you please explain this flow chart in detail again?

Sure.

❑ What's the difference between Ethertype 0x0806 and 0x0800?

x0806 is ARP, and x0800 is IP payload in an Ethernet frame.

Counters

Per Table	Per Flow	Per Port	Per Queue
Active Entries	Received Packets	Received Packets	Transmit Packets
Packet Lookups	Received Bytes	Transmitted Packets	Transmit Bytes
Packet Matches	Duration (Secs)	Received Bytes	Transmit overrun errors
	Duration (nanosecs)	Transmitted Bytes	
		Receive Drops	
		Transmit Drops	
		Receive Errors	
		Transmit Errors	
		Receive Frame Alignment Errors	
		Receive Overrun erorrs	
		Receive CRC Errors	
		Collisions	

Student Questions

http://www.cse.wustl.edu/~jain/cse570-23/ ©2023 Raj Jain

Actions

❑ Forward to Physical Port *i* or to *Virtual Port*:

➢ **All**: to all interfaces <u>except</u> the incoming interface

➢ **Controller**: encapsulate and send to controller

➢ **Local**: send to its local networking stack

➢ **Table**: Perform actions in the flow table

➢ **In_port**: Send back to input port

➢ **Normal**: Forward using traditional Ethernet

➢ **Flood**: Send along minimum spanning tree <u>except</u> for the incoming interface

❑ Enqueue: To a particular queue in the port $\Rightarrow$ QoS

❑ Drop

❑ Modify Field: E.g., add/remove VLAN tags, ToS bits, Change TTL

Actions (Cont)

- Masking allows matching only selected fields, e.g., Dest. IP, Dest. MAC, etc.

- If a header matches an entry, corresponding actions are performed, and counters are updated

- If no header matches, the packet is queued, and the header is sent to the controller, which sends a new rule. Subsequent packets of the flow are handled by this rule.

- Secure Channel: Between the controller and the switch using TLS

- Modern switches already implement flow tables, typically using Ternary Content Addressable Memories (TCAMs)

- A controller can change the forwarding rules if a client moves $\Rightarrow$ Packets for mobile clients are forwarded correctly

- A controller can send flow table entries beforehand (**Proactive**) or Send them on demand (**Reactive**). OpenFlow allows both models.

Student Questions

- When a packet doesn't match rules, we usually send it to the controller. So, we must wait some time to receive the corresponding rule. In the meantime, what do we do with the flow? Particularly, if the flow is an elephant flow, do we queue it somewhere?

Yes. All packets of that flow have to wait.

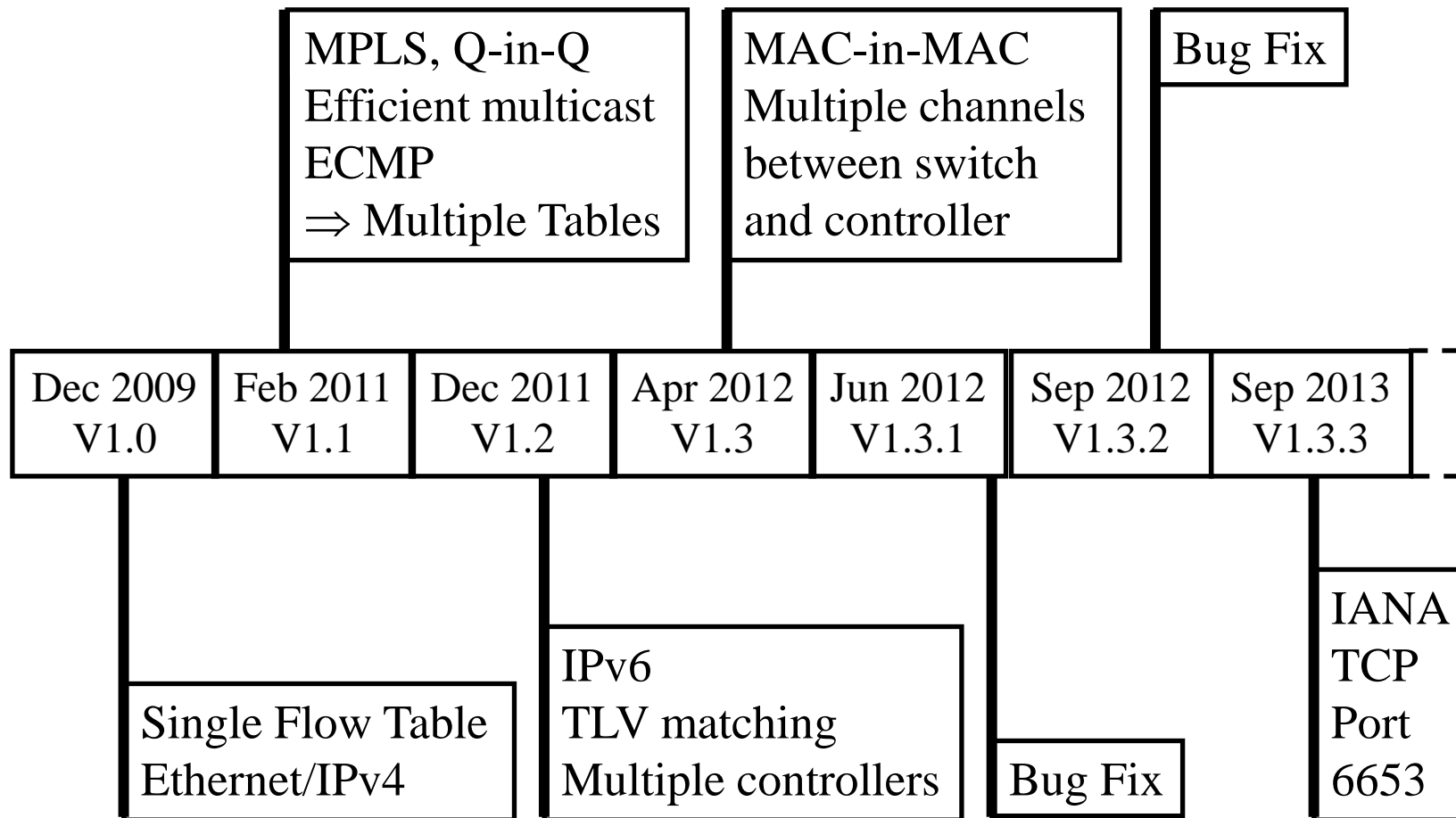- For a controller, is it dedicated hardware or software that runs inside a switch?

Although it can be specialized hardware, a controller is generally a standard computer with controller software.

- Can you explain proactive and reactive in detail again?

Proactive = Find a solution before the problem happens

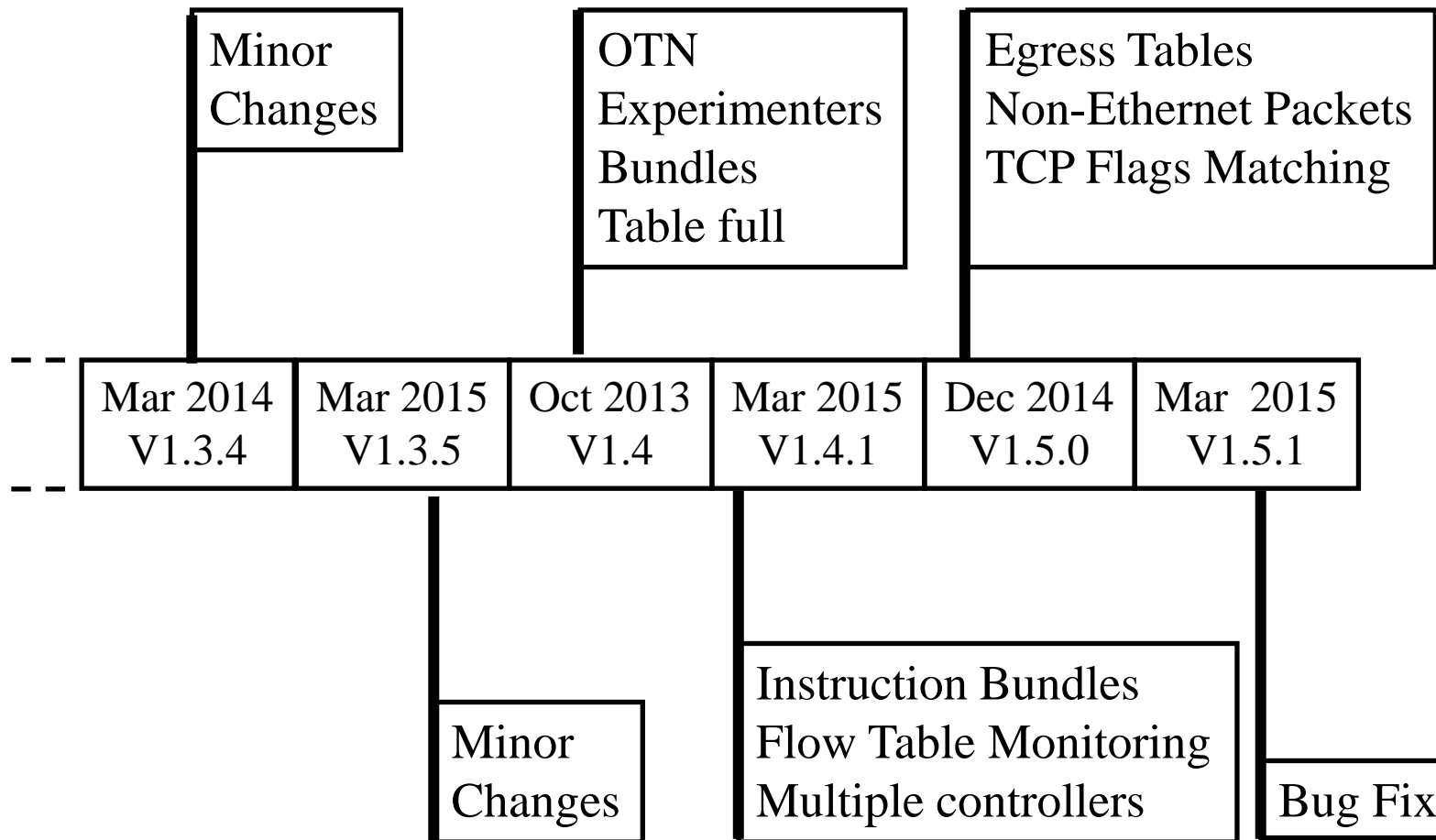*Reactive = Find a solution after the problem happens*

OpenFlow Evolution Summary

| | MPLS, Q-in-Q
Efficient multicast
ECMP
$\Rightarrow$ Multiple Tables | MAC-in-MAC
Multiple channels
between switch
and controller | Bug Fix | |

| Dec 2009
V1.0 | Feb 2011
V1.1 | Dec 2011
V1.2 | Apr 2012
V1.3 | Jun 2012
V1.3.1 | Sep 2012
V1.3.2 | Sep 2013
V1.3.3 |

Single Flow Table
Ethernet/IPv4

IPv6
TLV matching
Multiple controllers

Bug Fix

IANA
TCP
Port
6653

OpenFlow Evolution Summary (Cont)

Minor Changes		OTN Experimenters Bundles Table full		Egress Tables Non-Ethernet Packets TCP Flags Matching	

Mar 2014 V1.3.4	Mar 2015 V1.3.5	Oct 2013 V1.4	Mar 2015 V1.4.1	Dec 2014 V1.5.0	Mar 2015 V1.5.1

Minor Changes

Instruction Bundles
Flow Table Monitoring
Multiple controllers

Bug Fix

Student Questions

Bootstrapping

- Switches require initial configuration: Switch IP address, Controller IP address, Default gateway

- Switches connect to the controller

- The switch provides configuration information about ports

- The controller installs a rule to forward LLDP (Link Layer Discovery Protocol) packets to the controller and then sends, one by one, LLDP packets to be sent out to port I (i=1, 2, …, n), which are forwarded to respective neighbors. The neighbors send the packets back to the controller.

- The controller determines the topology from LLDP packets

- LLDP is a one-way protocol to advertise the capabilities at fixed intervals.

Controller
LLDP 1   LLDP 2
All rules
LLDP 2
LLDP 1
Rule 1
LLDP 1   LLDP 2
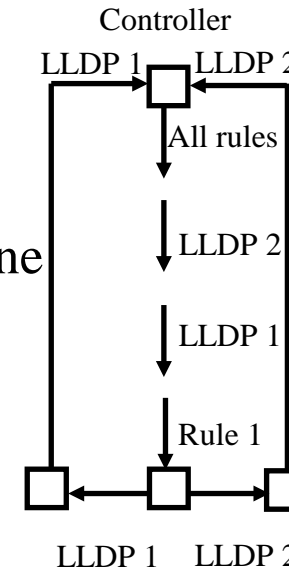
Student Questions

- Please explain bootstrapping more. Is it a control or data plane activity?

Bootstrapping is the "booting" process. It is a management plane activity.

Ref: S. Sharma, et al., "Automatic Bootstrapping of OpenFlow Networks," 19th IEEE Workshop on LANMAN, 2013, pp. 1-6, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6528283 (Available to subscribers only)

OpenFlow Configuration Protocol (OF-Config)

❑ **OpenFlow Configuration Point**: Entity that configures OpenFlow switches

❑ **OF-Config**: Protocol used for configuration and management of OpenFlow Switches.
Assignment of OF controllers so that switches can initiate connections to them:

➤ The IP address of the controller

➤ The port number at the controller

➤ Transport protocol:
TLS or TCP

➤ Configuration of queues (min/max rates) and ports

➤ Enable/disable receive/forward speed, media on ports
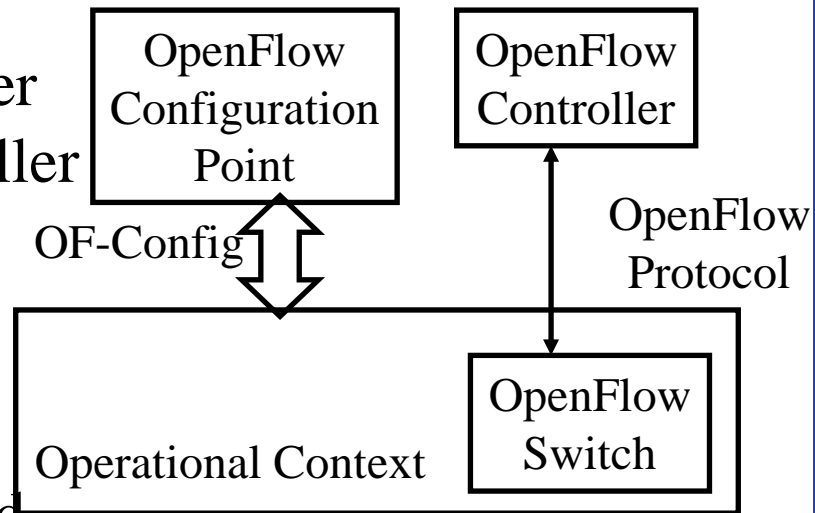
Student Questions

❑ What makes the OF Config point necessary in addition to the separation of function? It sounds like an OF controller can perform its function if it speaks OF-config protocol, e.g., sends a particular config packet that sets the transport protocol, config of queues, etc.

OF-Config is necessary if you have a large number of switches.

OF-Config (Cont)

❑ A physical switch = one or more logical switches each controlled by an OF Controller

❑ OF-Config allows the configuration of logical switches.
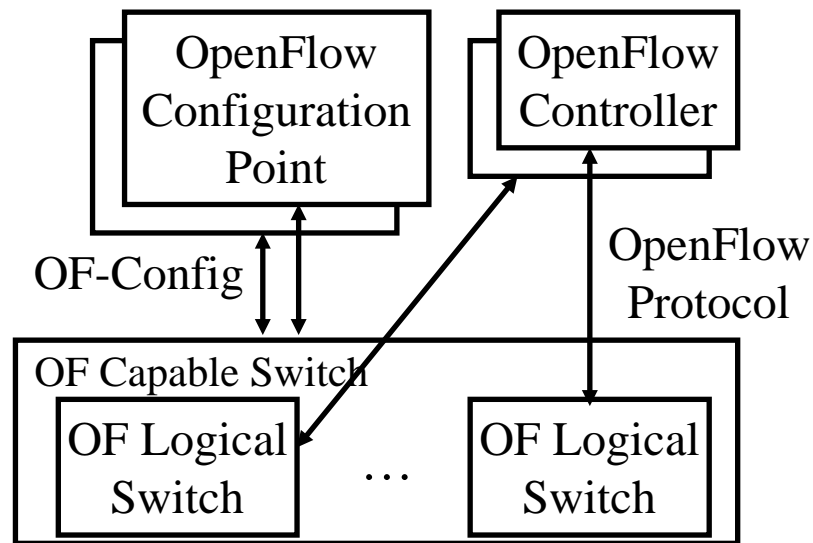
Student Questions

❑ Can one logical switch connect to multiple controllers?

No.

OpenFlow Notification Framework

- **Notification**: Event-triggered messages, e.g., link down
- **Publish/subscribe model**: Switch = publisher. OpenFlow controller or OpenFlow config points, and others can subscribe. They will be notified about the events they subscribe to.
- Use **ITU-T M.3702** Notifications: Attribute value change, Communication alarm, Environmental alarm, Equipment alarm, QoS alarm, Processing error alarm, Security alarm, State change, Object creation, and deletion
- **Pre-existing Notifications**: Do not fit in the framework but will be recognized.
  - OpenFlow: Packet-in, Flow removed, Port Status, Error, Hello, Echo request, Echo reply, Experimenter
  - OpenFlow Config: OpenFlow logical switch instantiation, OpenFlow capability switch capability change, Successful OpenFLow session establishment, Failed OpenFlow session establishment, Port failure or recovery

Student Questions

Implementation Issues

- ❑ 40+ matching fields in a flow
- ❑ Multiple tables, each with a large number of flow entries
- ❑ Instructions and actions for each table
- ❑ Need VXLAN, NVGRE, etc. support
- ❑ For an extensive network, flow-level programming can take a long time
- ❑ Can't keep adding header fields ⇒ Move to P4 programming
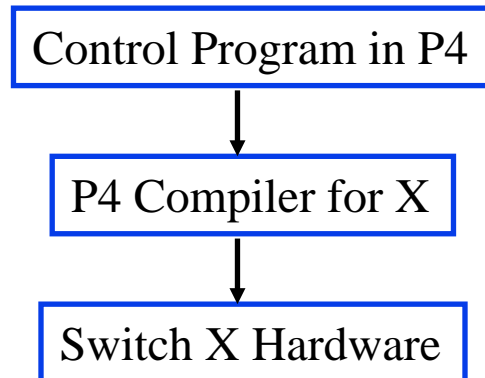
Student Questions
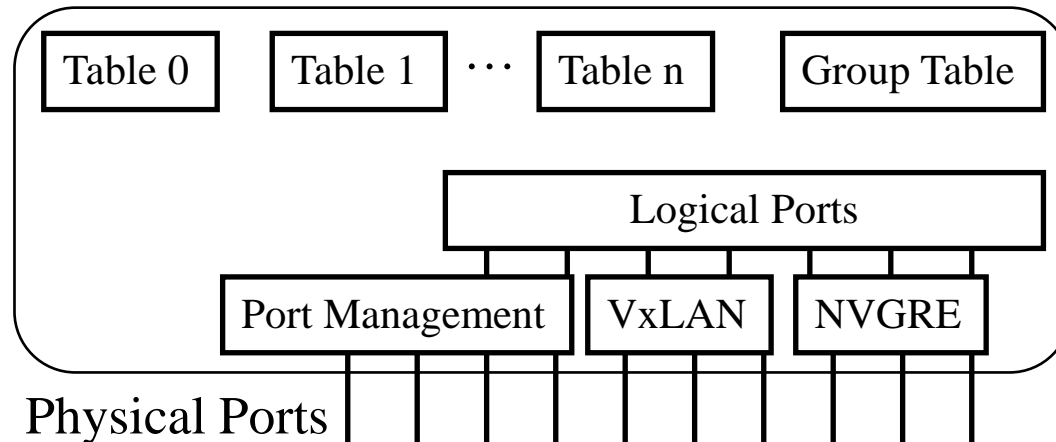
Ref: R. Oshana and S. Addepalli, "Networking Trends- Software Defined Networking, Network Virtualization and Cloud Orchestration," Asia Power Arch. Conf, Oct 2012, https://www.power.org/wp-content/uploads/2012/10/13.-FSL-SDN-Openflow-and-Cloud-computing-UPD_Rob-Oshana.pdf

P4 Language

- **P**rogramming **P**rotocol-Independent **P**acket **P**rocessors
- **Protocol-Independent**: Produces forwarding instructions for any L2-L5 protocol
- **Target Independent**: Programmers write one program that can run on many different hardware ⟹ A hw specific compiler translates the program to instructions for that hardware
  - ➢ Similar to how C program can run on any hardware
- **Reconfigurability**: Can change the program w/o changing the hardware

```
Control Program in P4
        |
        v
  P4 Compiler for X
        |
        v
  Switch X Hardware
```
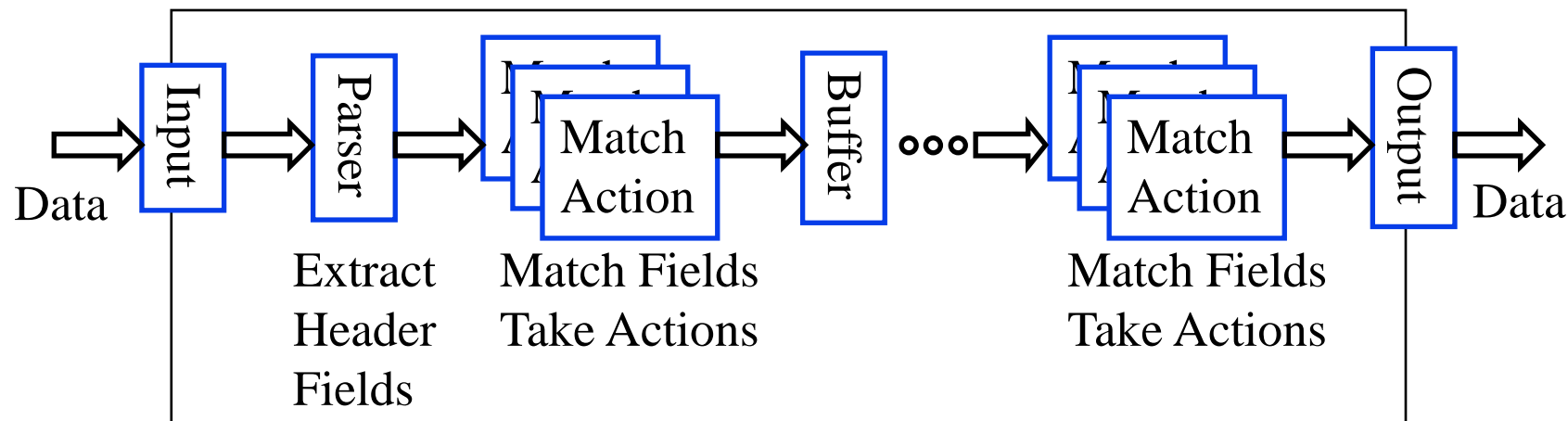
Ref: P. Bosshart, et al., "P4: Programming Protocol-Independent Packet Processors," Computer Communication Review, ACM SIGCOMM, July 2014, 8 pp., http://www.sigcomm.org/node/3503

http://www.cse.wustl.edu/~jain/cse570-23/
©2023 Raj Jain

P4 Switch Design

❑ Packet parser: How to extract header fields, their names, and types

  ➢ OpenFlow assumed a fixed parser and needed a new version for each new field

❑ Tables: Match and Action (similar to OpenFlow)

❑ Actions: Decrement Hop Count
             Forward packets to port three queue 4

❑ Some actions can be done in parallel, others sequential

Data →│ Input │→│ Parser │→│ Match Action │→│ Buffer │ ooo →│ Match Action │→│ Output │→ Data

Parser: Extract Header Fields

Match Action: Match Fields Take Actions

Match Action: Match Fields Take Actions

http://www.cse.wustl.edu/~jain/cse570-23/

Open vSwitch

❑ Open Source Virtual Switch

❑ Nicira Concept

❑ It can Run as a stand-alone hypervisor switch or as a distributed switch across multiple physical servers

❑ Default switch in XenServer 6.0, Xen Cloud Platform and supported in Proxmox VE, VirtualBox, Xen KVM

❑ Integrated into many cloud management systems, including OpenStack, openQRM, OpenNebula, and oVirt

❑ Distributed with Ubuntu, Debian, and Fedora Linux. Also FreeBSD

❑ Intel has an accelerated version of Open vSwitch in its own Data Plane Development Kit (**DPDK**)

Student Questions

❑ What is the difference between this OpenvSwitch and the vSwitch concept we learned in the previous module?
Same.

Ref: http://openvswitch.org/

Open vSwitch Features

❑ Link Aggregation Control Protocol (LACP)

❑ IEEE 802.1Q VLAN

❑ IEEE 802.1ag Connectivity Fault Management (CFM)

❑ Bidirectional Forwarding Detection (BFD) to detect link faults (RFC 5880)

❑ IEEE 802.1D-1998 Spanning Tree Protocol (STP)

❑ Per-VM traffic policing

❑ OpenFlow

❑ Multi-table forwarding pipeline

❑ IPv6

❑ GRE, VXLAN, IPSec tunneling

❑ Kernel and user-space forwarding engine options

Student Questions
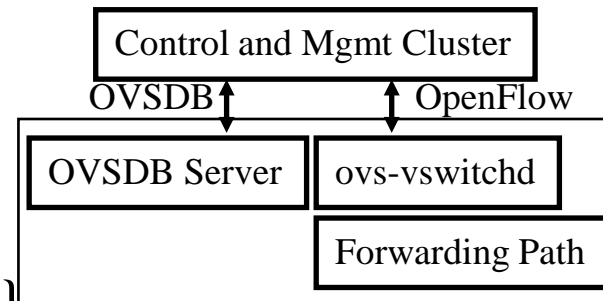
http://www.cse.wustl.edu/~jain/cse570-23/

OVSDB

❑ Open vSwitch Database Management Protocol (OVSDB)

❑ Monitoring capability using publish-subscribe mechanisms

❑ Stores both provisioning and operational state

❑ Java Script Object Notation (JSON) is used for schema format and JSON-RPC over TCP for wire protocol (RFC 4627)

   `<database-schema>`

      "name": `<id>`

      "version": `<version>`

      "tables": {`<id>`: `<table-schema>`,…}

❑ RPC Methods: List databases, Get Schema, Update, Lock, …

❑ Open vSwitch project includes open-source OVSDB client and server implementations

Ref: B. Pfaff and B. Davie, "The Open vSwitch Database Management Protocol," IETF RFC 7047m December 2013, https://tools.ietf.org/pdf/rfc7047

Summary

1. Four planes of Networking: Data, Control, Management, Service

2. OpenFlow separates the control plane and moves it to a central controller $\Rightarrow$ Simplifies the forwarding element

3. Switches match incoming packets with flow entries in a table and handle it as instructed. The controller supplies the flow tables and other instructions.

4. OpenFlow has been extended to IPv4, MPLS, IPv6, and Optical Network. But more work is ahead.

5. Many hardware and software-based switches, including Open vSwitch

Reading List

Any one book:

- ❑ T. Nadeau and K. Gray, "SDN," O'Reilly, 2013, 384 pp, ISBN:978-1-449-34230-2B (Safari Book)

- ❑ Oswald Coker, Siamak Azodolmolky, "Software-Defined Networking with OpenFlow - Second Edition," Packt Publishing, October 2017, 246 pp., ISBN:978-1-78398-429-9 (Safari Book).

- ❑ William Stallings, "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud," Addison-Wesley Professional, October 2015, 544 pp., ISBN:0-13-417539-5 (Safari Book).

- ❑ Kingston Smiler. S, "OpenFlow Cookbook," Packt Publishing, April 2015, 292 pp., ISBN:978-1-78398-795-5 (Safari Book).

- ❑ P. Bosshart et al., "P4: Programming Protocol-Independent Packet Processors," Computer Communication Review, ACM SIGCOMM, July 2014, 8 pp., http://www.sigcomm.org/node/3503

Student Questions

http://www.cse.wustl.edu/~jain/cse570-23/   ©2023 Raj Jain

References

❑ N. McKeown et al., ``OpenFlow: Enabling Innovation in Campus Networks,'' ACM SIGCOMM CCR, Vol. 38, No. 2, April 2008, pp. 69-74.

❑ ONF, "The OpenFlow Timeline," http://openflownetworks.com/of_timeline.php

❑ Open Data Center Alliance Usage Model: Software Defined Networking Rev 1.0," http://www.opendatacenteralliance.org/docs/Software_Defined_Networking _Master_Usage_Model_Rev1.0.pdf

❑ R. Oshana and S. Addepalli, "Networking Trends- Software Defined Networking, Network Virtualization and Cloud Orchestration," Asia Power Arch. Conf, Oct 2012, https://www.power.org/wp-content/uploads/2012/10/13.-FSL-SDN-Openflow-and-Cloud-computing-UPD_Rob-Oshana.pdf

❑ ONF, **Technical Library** (includes all OpenFlow, OF-Config, and other specifications), https://www.opennetworking.org/sdn-resources/technical-library

Student Questions

References (Cont)

❑ http://www.openvswitch.org/

❑ http://www.projectfloodlight.org/indigo/

❑ http://flowforwarding.github.io/LINC-Switch/

❑ http://github.com/CPqD/openflow-openwrt

❑ http://cpqd.github.io/ofsoftswitch13/

❑ http://sourceforge.net/projects/xorplus

Student Questions

Wikipedia Links

❑ http://en.wikipedia.org/wiki/OpenFlow

❑ http://en.wikipedia.org/wiki/Software-defined_networking

❑ http://en.wikipedia.org/wiki/Network_Functions_Virtualization

❑ http://en.wikipedia.org/wiki/Forwarding_plane

❑ https://en.wikipedia.org/wiki/P4_(programming_language)

Student Questions

Wikipedia Links (Optional)

- http://en.wikipedia.org/wiki/NetFlow
- http://en.wikipedia.org/wiki/IP_Flow_Information_Export
- http://en.wikipedia.org/wiki/SFlow
- http://en.wikipedia.org/wiki/Northbound_interface
- http://en.wikipedia.org/wiki/Big_Switch_Networks
- http://en.wikipedia.org/wiki/Open_Data_Center_Alliance
- http://en.wikipedia.org/wiki/Virtual_Extensible_LAN
- http://en.wikipedia.org/wiki/Optical_Transport_Network
- http://en.wikipedia.org/wiki/Automatically_switched_optical_network
- http://en.wikipedia.org/wiki/Wavelength-division_multiplexing
- http://en.wikipedia.org/wiki/IEEE_802.1ad
- http://en.wikipedia.org/wiki/Transport_Layer_Security
- http://en.wikipedia.org/wiki/OpenStack
- http://en.wikipedia.org/wiki/IPv6_packet
- http://en.wikipedia.org/wiki/ICMPv6
- http://en.wikipedia.org/wiki/Multiprotocol_Label_Switching

Student Questions

Acronyms

- ACL   Access Control List
- API   Application Programming Interface
- ARP   Address Resolution Protocol
- ASICs   Application Specific Integrated Circuit
- BFD   Bidirectional Forwarding Detection
- BUM   Broadcast, Unknown, and Multicast
- CFM   Connectivity Fault Management
- CPU   Central Processing Unit
- DFCA   Dynamic Frequency Channel Allocation
- DSCP   Differentiated Service Control Point
- ECMP   Equal Cost Multipath
- ESP   Encrypted Security Payload
- FCAPS   Fault, Configuration, Accounting, Performance and Security
- GRE   Generic Routing Encapsulation
- ICMP   Internet Control Message Protocol
- ID   Identifier

Student Questions

Acronyms (Cont)

- IDS   Intrusion Detection System
- IEEE   Institution of Electrical and Electronic Engineers
- IETF   Internet Engineering Task Force
- IGMP   Internet Group Multicast Protocol
- IP   Internet Protocol
- IPFIX   IP Flow Information Export Protocol
- IPSec   IP Security
- IPv4   Internet Protocol version 4
- IPv6   Internet Protocol version 6
- JSON   Java Script Object Notation
- KVM   Kernel-based Virtual Machine
- LACP   Link Aggregation Control Protocol
- LLDP   Link Layer Discovery Protocol
- MAC   Media Access Control
- MAN   Metropolitan Area Network
- MPLS   Multiprotocol Label Switching

Student Questions

Acronyms (Cont)

- NFV — Network Function Virtualization
- NVGRE — Network Virtualization using Generic Routing Encapsulation
- OF — OpenFlow
- ONF — Open Networking Foundation
- openQRM — Open Qlusters Resource Manager
- OpenWRT — Open WRT54G (Linksys product name) software
- OSPF — Open Shortest Path First
- OTN — Optical Transport Network
- OVSDB — Open vSwitch Database
- P4 — Programming Protocol-Independent Packet Processors
- PIM-SM — Protocol Independent Multicast - Sparse Mode
- PIM — Protocol Independent Multicast
- QoS — Quality of Service
- RAN — Radio area networks
- RFC — Request for Comments
- RIP — IGMP, IPv6, PIM-SM
- RIP — Routing Information Protocol

Student Questions

Acronyms (Cont)

- RPC   Remote Procedure Call
- RSPAN   Remote Switch Port Analyzer
- SDN   Software Defined Network
- SPAN   Switch Port Analyzer
- SSL   Secure Socket Layer
- STP   Spanning Tree Protocol
- TCAM   Ternary Content Addressable Memory
- TCP   Transmission Control Protocol
- TLS   Transport Level Security
- TLV   Type-Length-Value
- ToS   Type of Service
- TTL   Time to Live
- TTP   Table Typing Patterns
- UDP   User Datagram Protocol
- VLAN   Virtual Local Area Network
- VM   Virtual Machine
- VxLAN   Virtual Extensible Local Area Network
- WG   Working Group

Student Questions

http://www.cse.wustl.edu/~jain/cse570-23/

Scan This to Download These Slides

Raj Jain

http://rajjain.com

Student Questions

http://www.cse.wustl.edu/~jain/cse570-23/m_14ofl.htm

Related Modules

CSE567M: Computer Systems Analysis (Spring 2013),
https://www.youtube.com/playlist?list=PLjGG94etKypJEKjNAa1n_1X0bWWNyZcof

CSE473S: Introduction to Computer Networks (Fall 2011),
https://www.youtube.com/playlist?list=PLjGG94etKypJWOSPMh8Azcgy5e_10TiDw

Wireless and Mobile Networking (Spring 2016),
https://www.youtube.com/playlist?list=PLjGG94etKypKeb0nzyN9tSs_HCd5c4wXF

CSE571S: Network Security (Fall 2011),
https://www.youtube.com/playlist?list=PLjGG94etKypKvzfVtutHcPFJXumyyg93u

Video Podcasts of Prof. Raj Jain's Lectures,
https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw

Student Questions