Introduction to Software Defined Networks (SDN)

Irfan Ali

Slides adapted from Prof. Berk Canberk

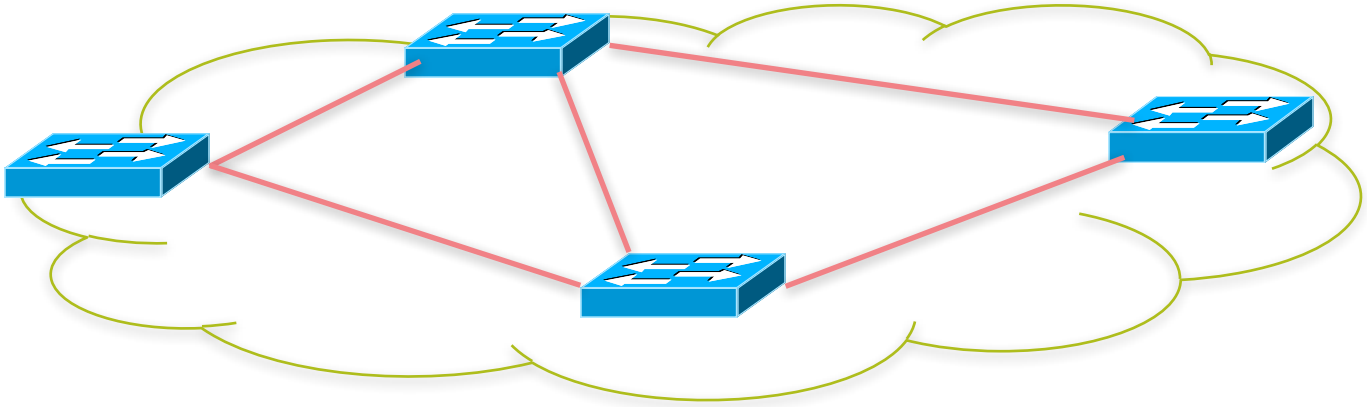
Outline

- Motivation- 1 (This lecture)
  - Centralizing the “routing/forwarding” logic
  - Centralizing “middleboxes”: Firewall and NAT

- Motivation -2 (Next two lectures)
  - Data-center networks
  - Network-virtualization and multi-tenancy support

Traditional Computer Networks

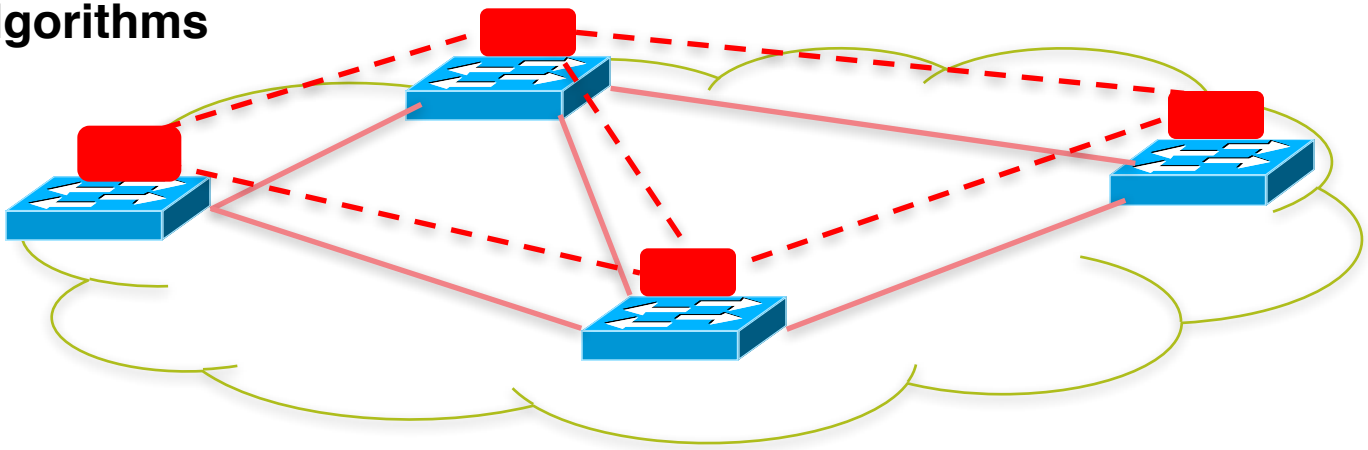
Data plane:
Packet streaming



Forward, filter, buffer, mark,
rate-limit, and measure packets

Traditional Computer Networks

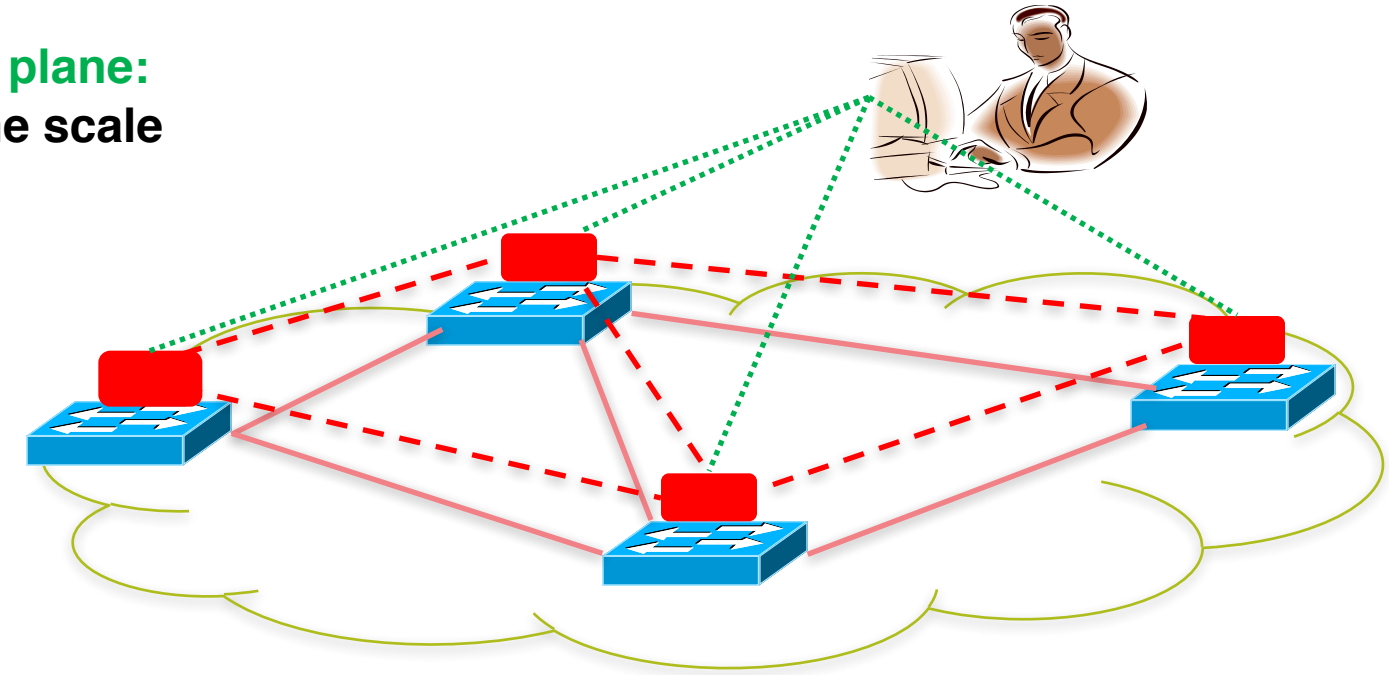
Control plane:
Distributed algorithms



Track topology changes, compute routes, install forwarding rules

Traditional Computer Networks

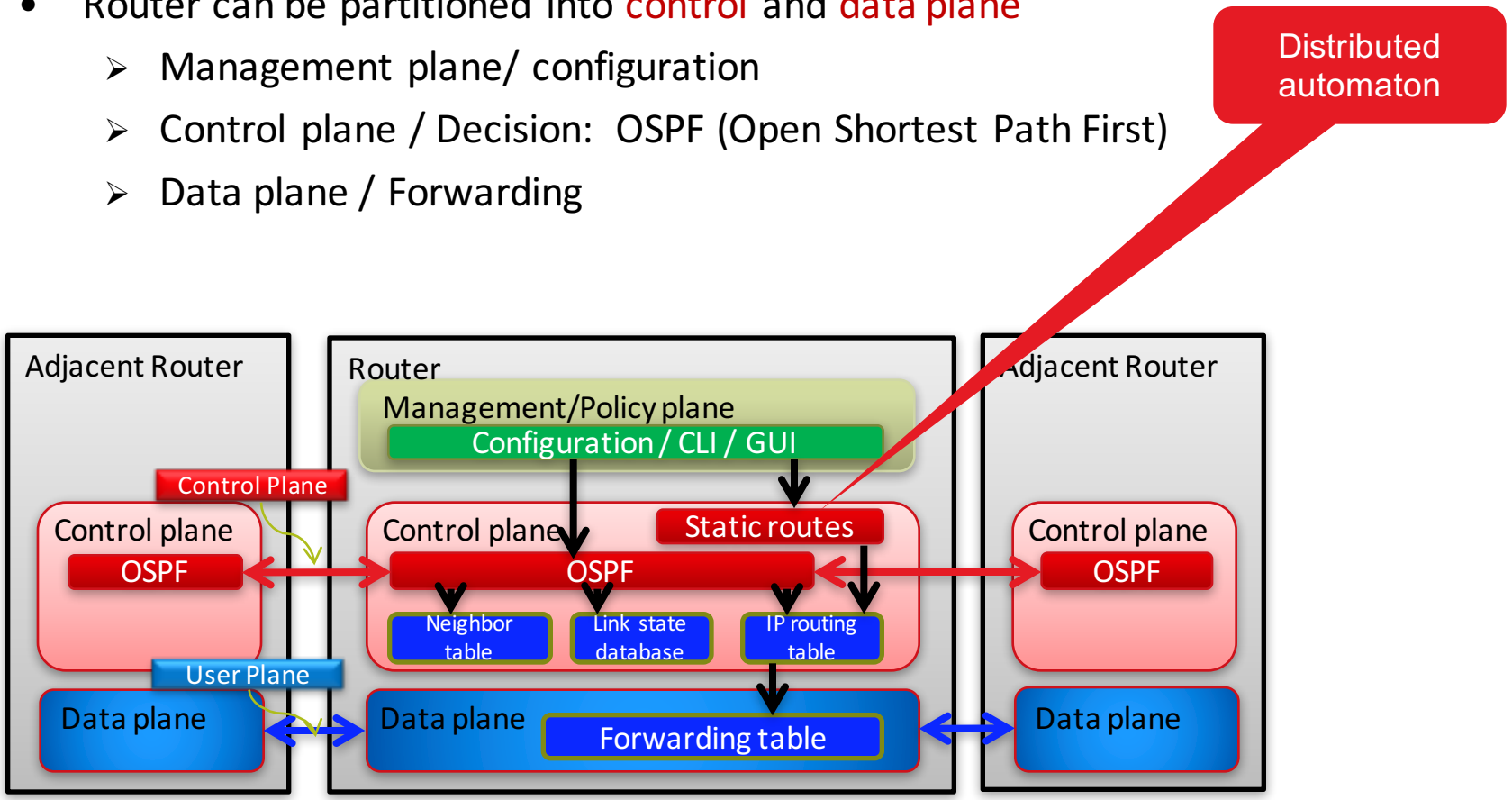
Management plane:
Human time scale



Collect measurements and configure the equipment

Example: Traditional Network Router

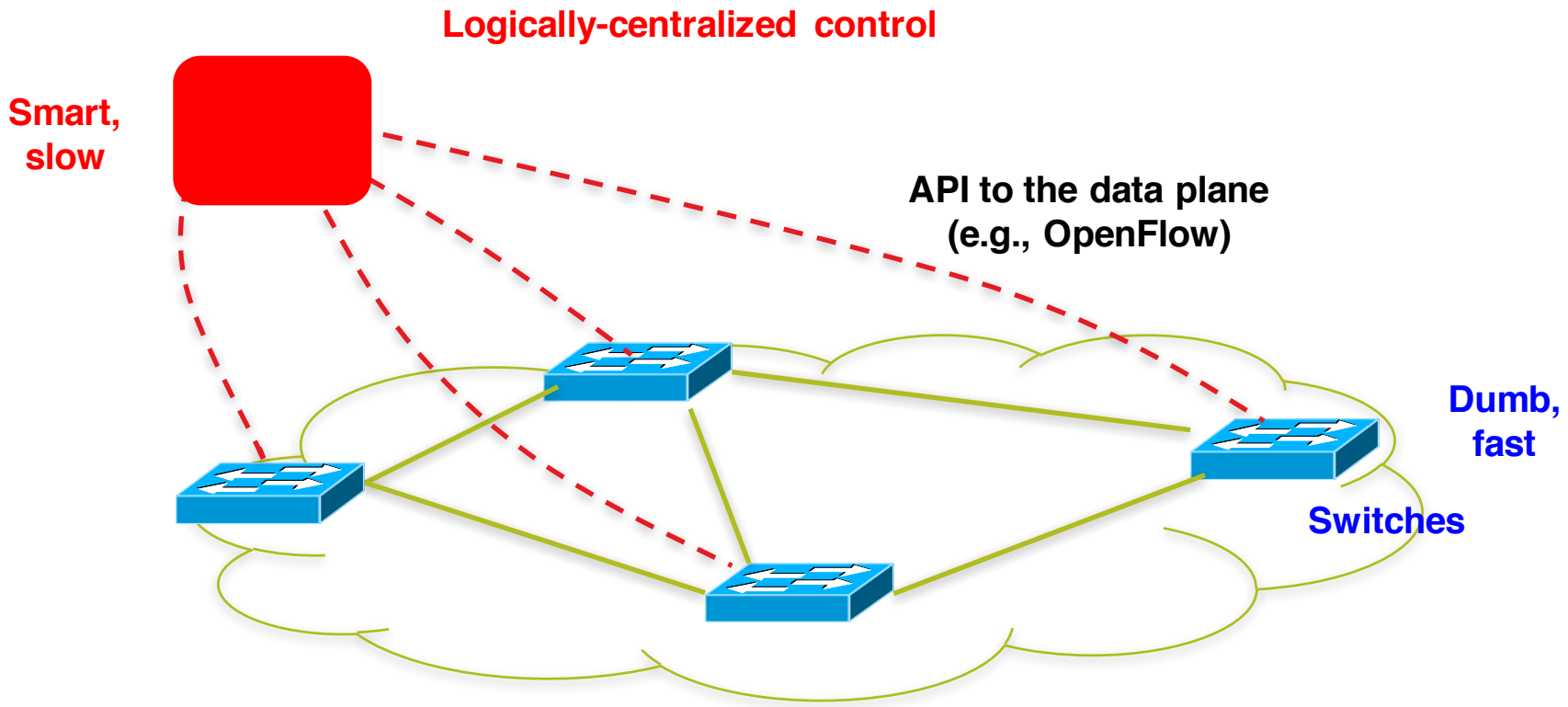
- Router can be partitioned into control and data plane
  - Management plane/ configuration
  - Control plane / Decision: OSPF (Open Shortest Path First)
  - Data plane / Forwarding



Problems with traditional networks

- Trying to “program” a particular behavior is an extremely complicated task.
  - Several hundred lines of configuration on each interface.
  - Several parameters: routing, forwarding, keep-alives.
- Several very tough tradeoffs
  - Fast recovery vs keep-alive traffic vs false positives
- Each router vendor has its own configuration language (command line interface)
- Features between router vendors do not work that easily.
- Every new feature from router vendor costs $$$$.

Software Defined Networking (SDN)

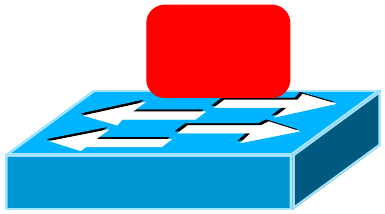


Software Defined Networking = Program the Network, Cheaply

- **Make it easy to "Program the network"**
  - Centralize the control logic of the network
  - Use commodity hardware for user-plane
- **Main drivers for SDN**
  - Reduce cost of data-center networks
  - **Network Multi-tenancy**: Enable ability to share network with multiple tenants each with their own network topology.

Basic SDN Motivation!

- Simpler management
  - No need to "invert" control-plane operations
- Faster pace of innovation
  - Less dependence on vendors and standards
- Easier interoperability
  - Compatibility only in "wire" protocols
- Simpler, cheaper equipment
  - Minimal software



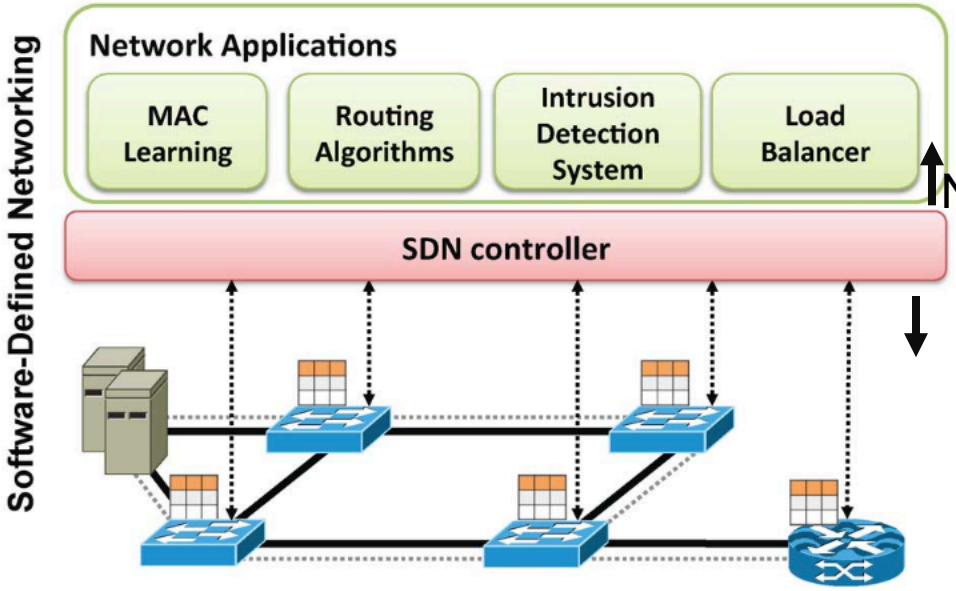
What "logic/box" is being centralized?

- Routing logic/Router
  - Match: longest destination IP prefix
  - Action: forward out a link
- Switching logic/Switch
  - Match: destination MAC address
  - Action: forward or flood
- Firewall logic/Firewall
  - Match: IP addresses and TCP/UDP port numbers
  - Action: permit or deny
- Address-Port translation logic/NAT
  - Match: IP address and port
  - Action: rewrite address and port

SDN High Level Architecture

Control Plane

Data Plane



North-bound Interface

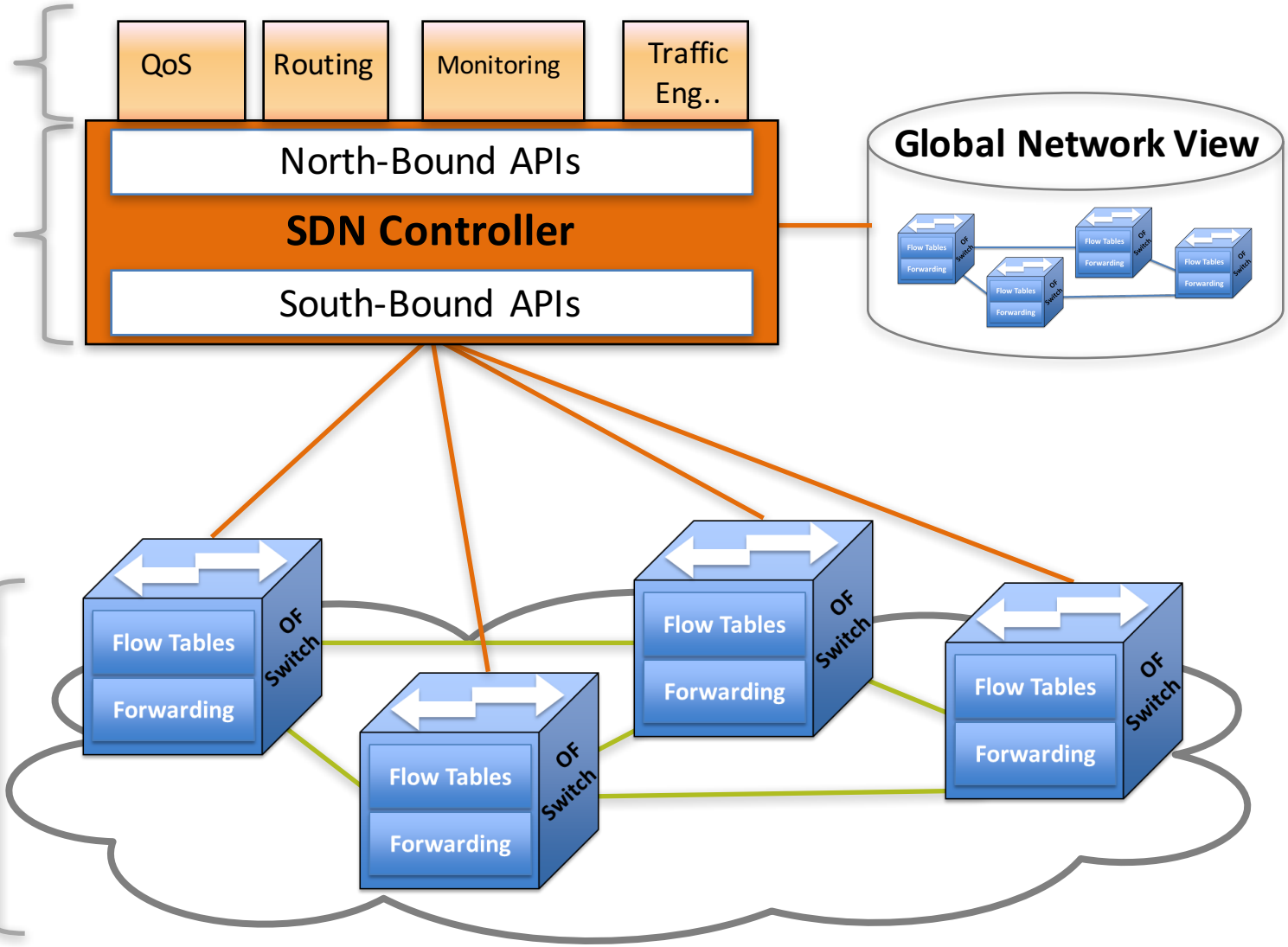
South-bound Interface

Control and Data Planes Isolation

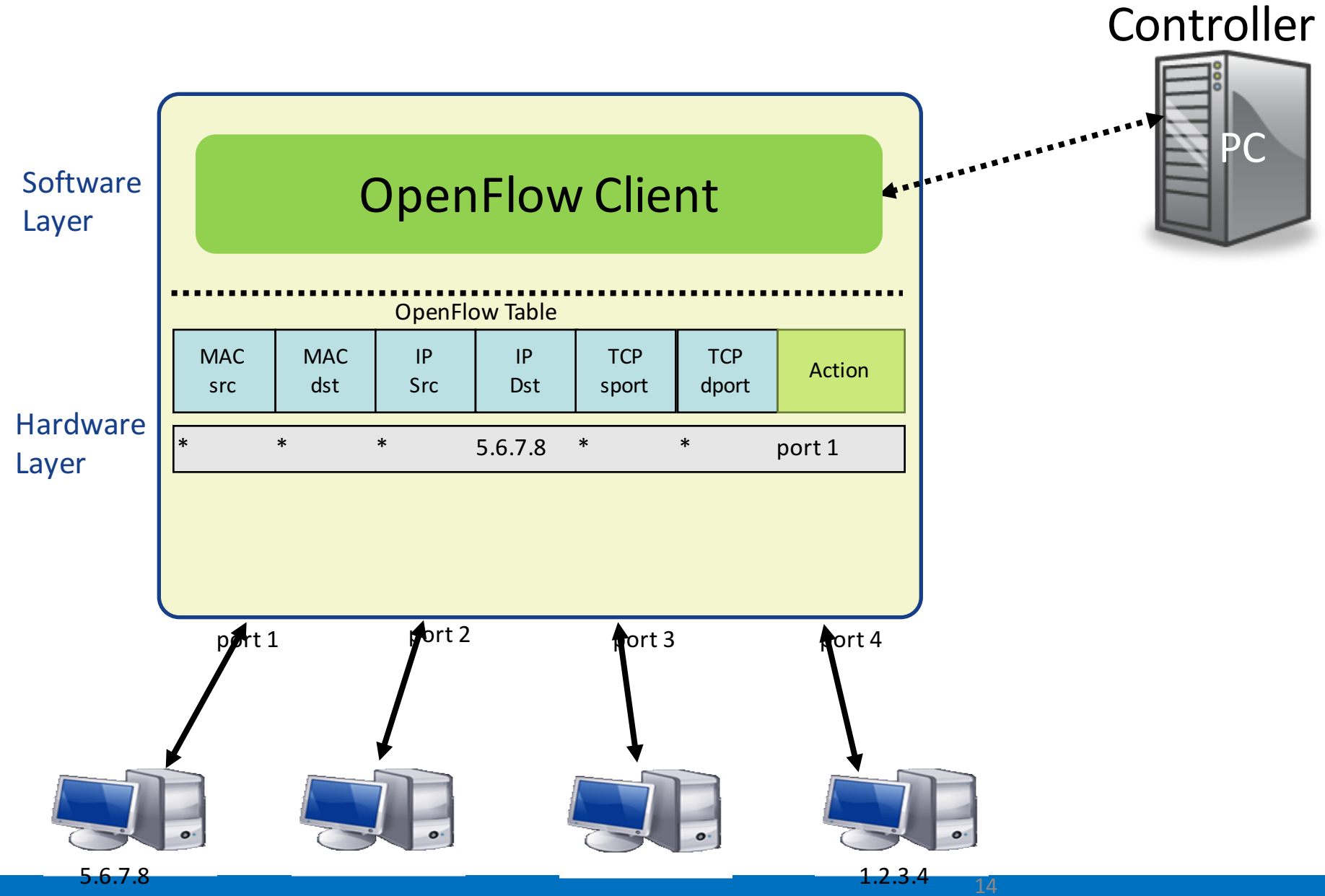
Applications Layer:
- SDN applications
- Business applications

Control-Plane Layer:
- SDN Controller(s)
- North-bound APIs
- South-bound APIs (e.g., OpenFlow protocol)

Data-Plane Layer:
- OpenFlow(OF) Switches
- Routers
- Other Infrastructure elements



OpenFlow Switching



OpenFlow Switching

Switching Use Case

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

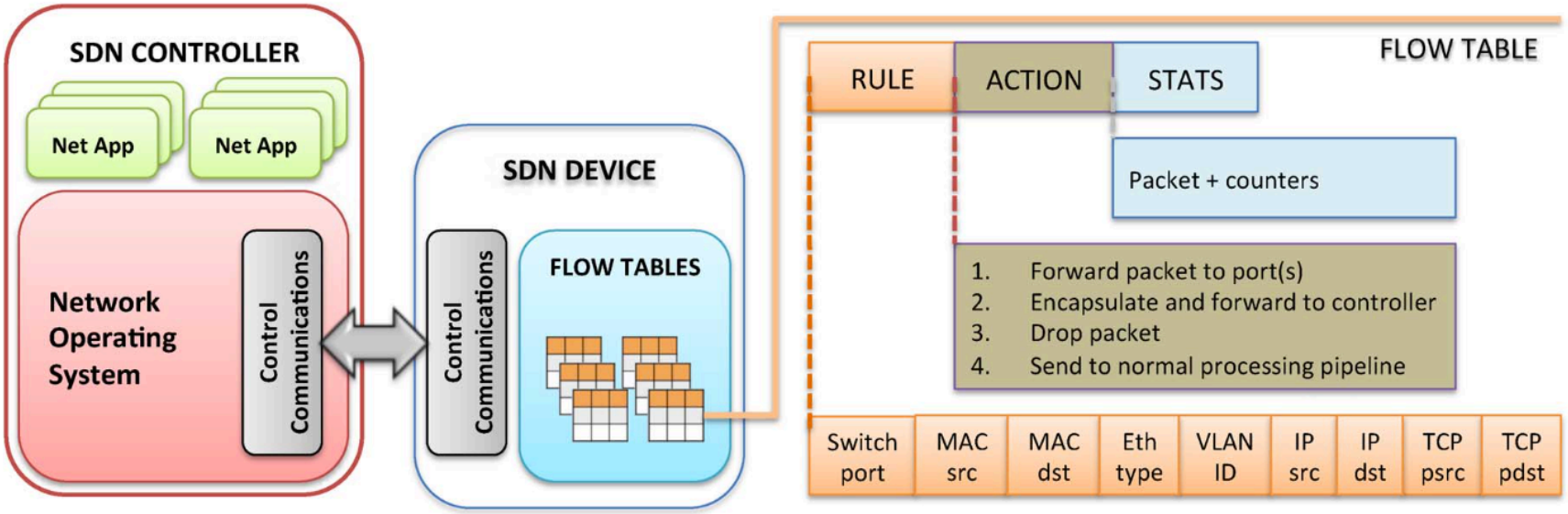
Routing Use Case

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

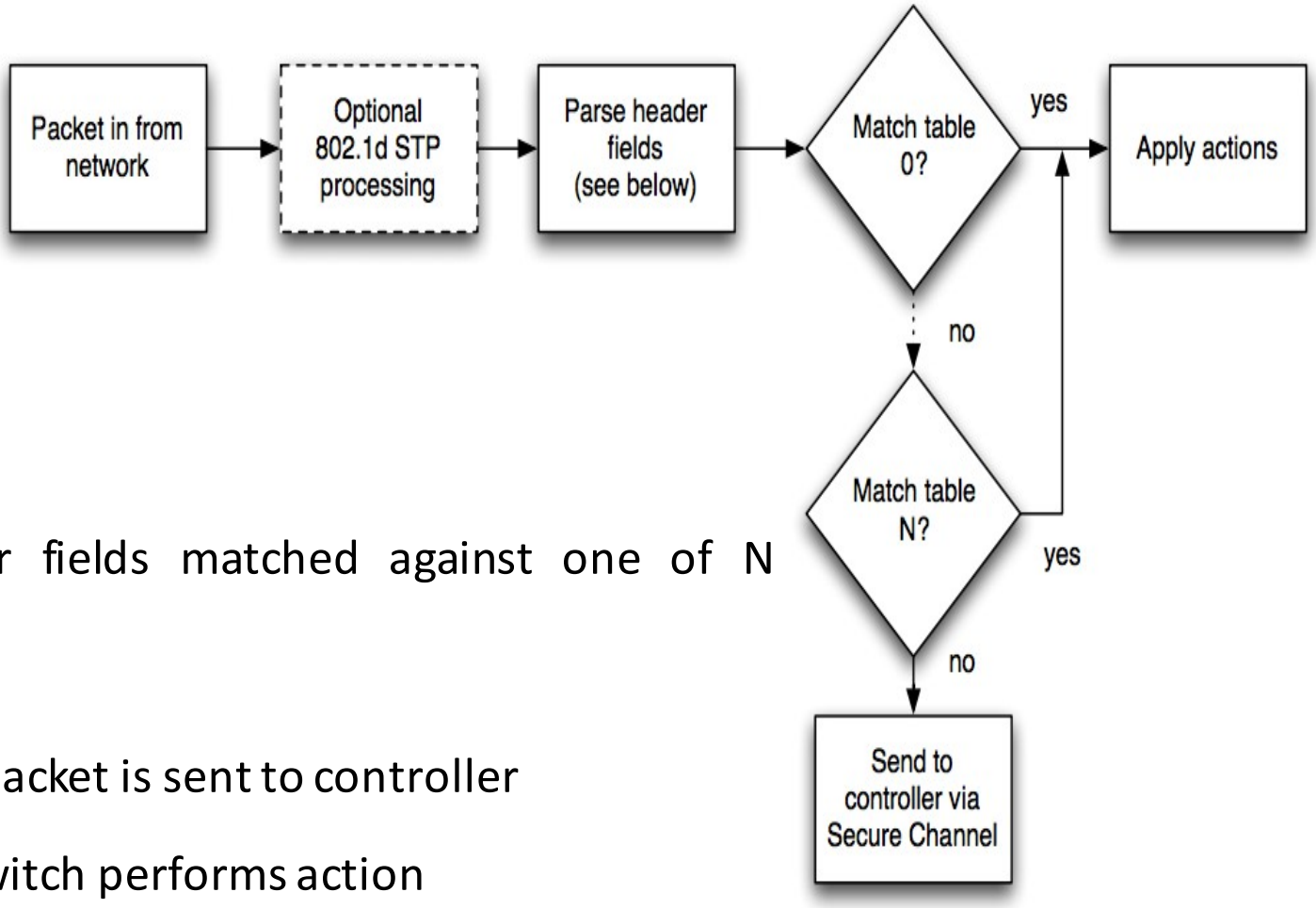
Firewall Use Case

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

Openflow Table



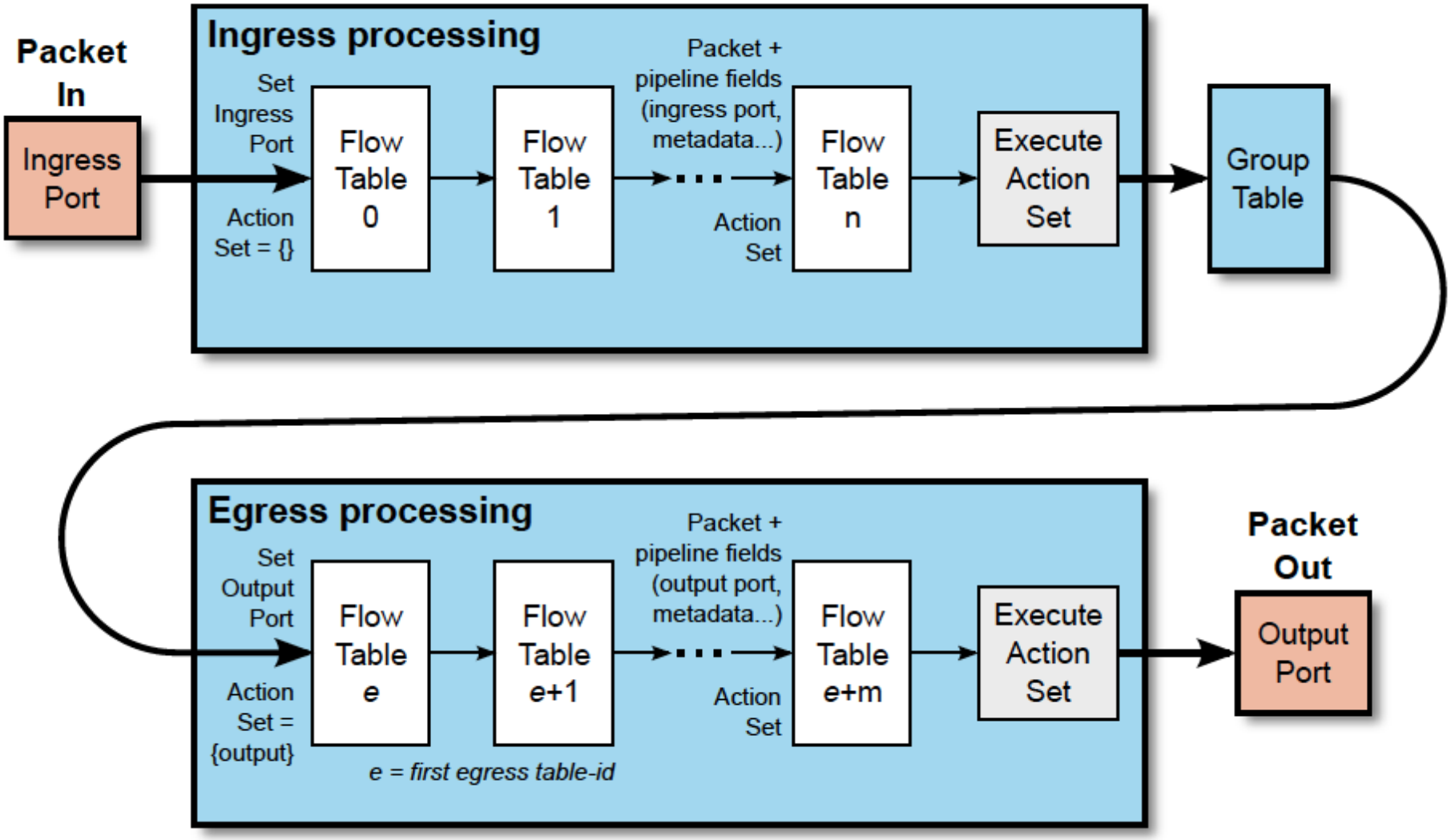
Matching (OpenFlow v.1.0)



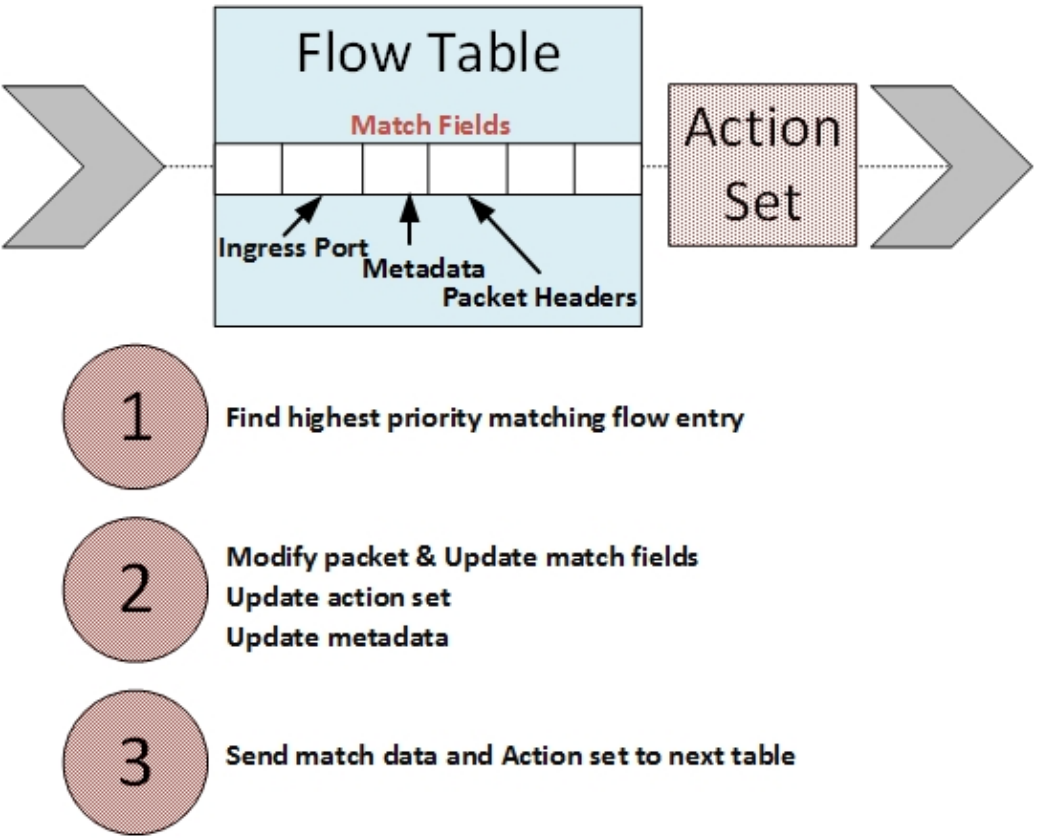
- Packet header fields matched against one of N tables
- If no match, packet is sent to controller
- Otherwise, switch performs action

OF Tables

Packet Flow through the processing pipeline



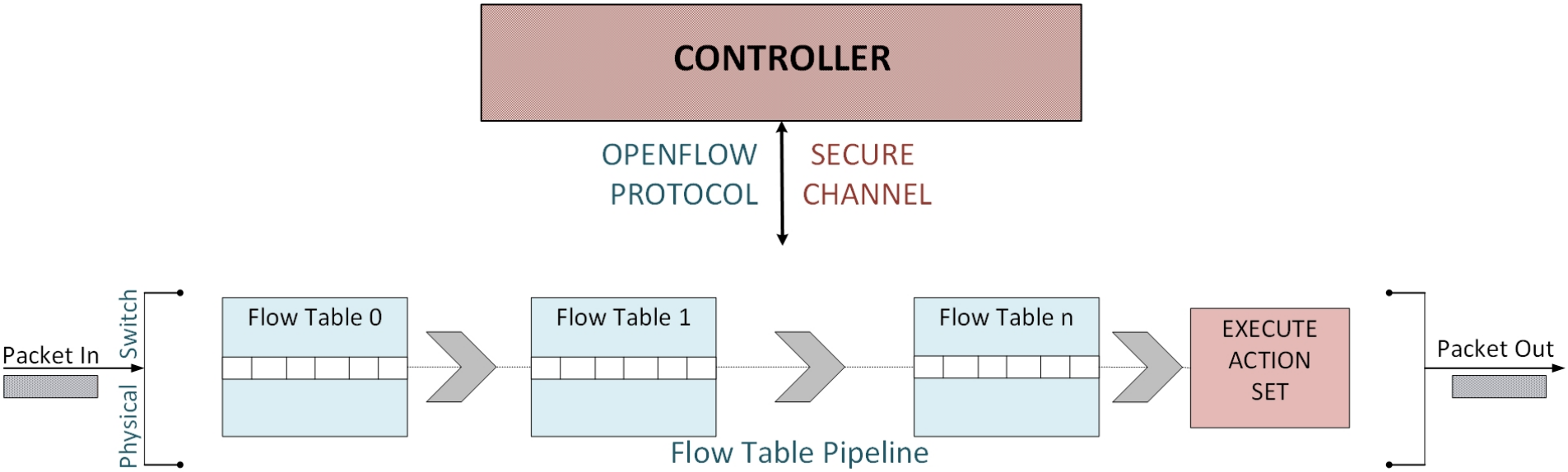
Operation and Flow Management



- All incoming flow are received by the OF switches in Data Plane
- In switch, the destination addresses of each flows are checked from Flow Table Pipeline
- **If it is matched, the actions in figure are performed**
- **If it is not matched, that flow is called as "newcomer" flow and first packet is sent to Controller**
- Controller decides suitable solution, updates the flow entry and assigns a forwarding rule for this newcomer flow

-G. Secinti, B. Canberk, T. Duong, L. Shu., 'Software Defined Architecture for VANET: A Testbed Implementation with Wireless Access Management', IEEE Communications Magazine, 2017

Operation and Flow Management



-G. Secinti, B. Canberk, T. Duong, L. Shu., 'Software Defined Architecture for VANET: A Testbed Implementation with Wireless Access Management', IEEE Communications Magazine, 2017.

The Actual Flow Table Looks Like

Port	Src MAC	Dst MAC	VLAN ID	Priority	EtherType	Src IP	Dst IP	Protocol IP Proto	QoS 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

OpenFlow Table: Basic Actions

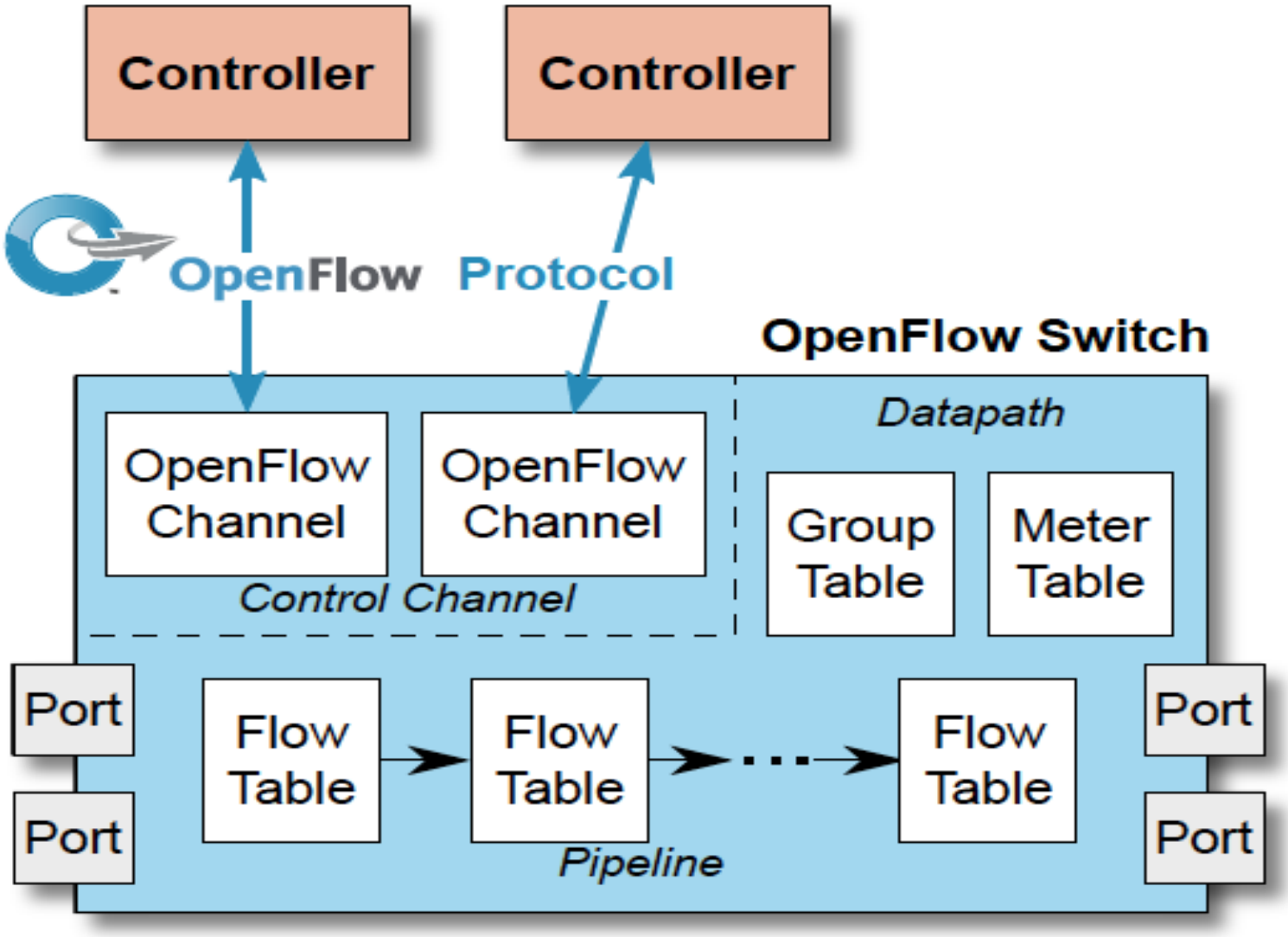
- All — To all interfaces except incoming interface.
- Controller — Encapsulate and send to controller.
- Local — send to its local networking stack.
- Table — Perform actions in the next flow table (table chaining or multiple table instructions).
- In_port — Send back to input port.
- Normal — Forward using traditional Ethernet.
- Flood — Send along minimum spanning tree except the incoming interface.

OpenFlow Table: Basic Stats

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	

- Provide counter for incoming flows or packets.
- Information on counter can be retrieved to control plane.
- Can be used to monitor network traffic.

Main Components of an OF Switch



Main Components of an OF Switch

- **Control Channel:** The aggregation of components of an OpenFlow logical switch that manages communication with controllers. The control channel includes one OpenFlow channel per OpenFlow controller.
  - OpenFlow Channel: **Interface** between an OpenFlow switch and an OpenFlow controller, used by the controller to manage the switch.

- **Datapath:** The aggregation of components of an OpenFlow logical switch that are directly involved **in traffic processing and forwarding**. The datapath includes the pipeline of flow tables, the group table and the ports.
  - Group: **List of action buckets** and some means of choosing one or more of those buckets to apply on a per-packet basis.
  - Meter: Switch element that can **measure and control the rate of pack**ets. The meter triggers a meter band if the packet rate or byte rate passing through the meter exceeds a predefined threshold. If the meter band drops the packet, it is called a Rate Limiter.

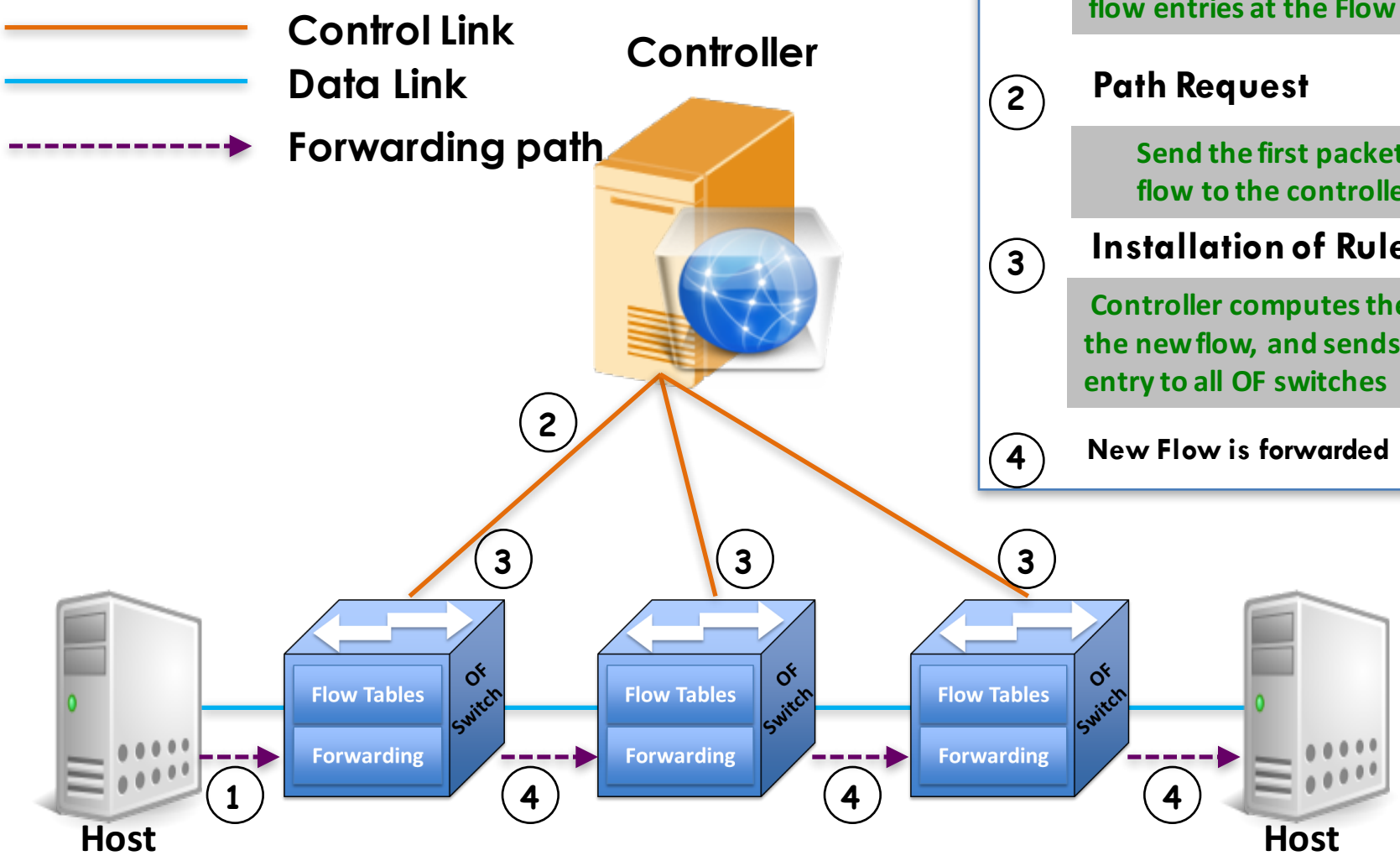
- **Pipeline:** the set of linked **flow tables that provide matching, forwarding, and packet modification** in an OpenFlow switch.
  - Flow Entry: an element in a flow table used to match and process packets. It contains a set of match fields for matching packets, a priority for matching precedence, a set of counters to track packets, and a set of instructions to apply
  - Flow Table: a stage of the pipeline. It contains flow entries.

Flow Tables and Entries

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

- **match fields:** to match against packets. These consist of the ingress port and packet headers, and optionally other pipeline fields such as metadata specified by a previous table.
- **priority:** matching precedence of the flow entry.
- **counters:** updated when packets are matched.
- **instructions:** to modify the action set or pipeline processing.
- **timeouts:** maximum amount of time or idle time before flow is expired by the switch.
- **cookie:** opaque data value chosen by the controller. May be used by the controller to filter flow entries affected by flow statistics, flow modification and flow deletion requests. Not used when processing packets.
- **flags:** flags alter the way flow entries are managed, for example the flag OFPFF_SEND_FLOW_REM triggers flow removed messages for that flow entry.

Operation and Flow Management



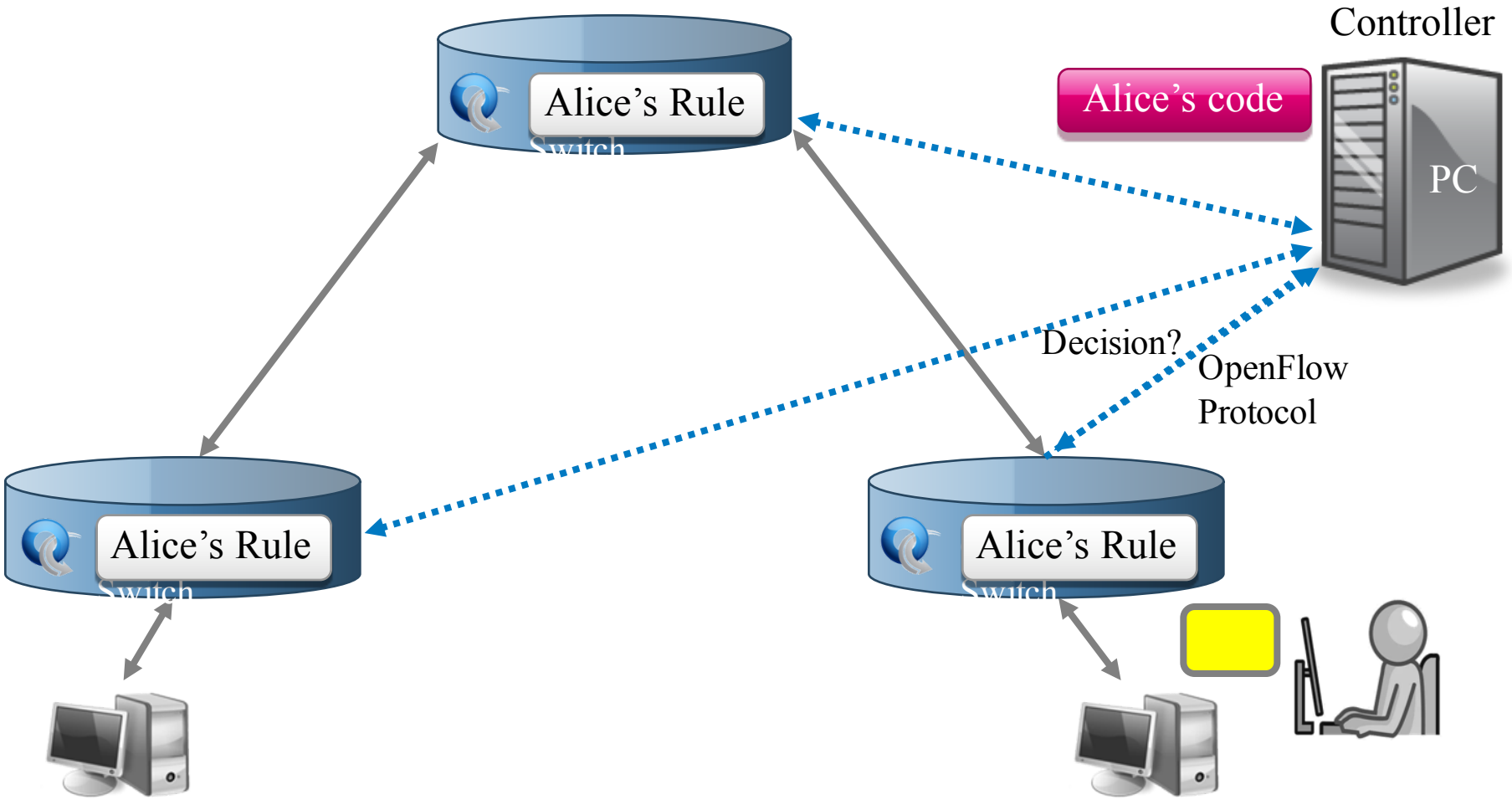
Example of Flow Management

1. **New Flow Arrival**
   If the new flow does not match the flow entries at the Flow Tables
2. **Path Request**
   Send the first packet of the new flow to the controller
3. **Installation of Rules**
   Controller computes the path for the new flow, and sends the flow entry to all OF switches
4. **New Flow is forwarded to destination**

-M. Erel, Z. Arslan, Y. Ozcevik, B. Canberk, 'Grade of Service based Adaptive Flow Management for Software Defined Heterogeneous Networks', Computer Networks (Elsevier), 2015.
-M. Erel, Z. Arslan, Y. Ozcevik, B. Canberk, 'Software-Defined Wireless Networking', Elsevier, 2014.

OpenFlow Switching

OF in action!



Openflow Protocol Versions

OpenFlow Version	Match fields	Statistics	# Matches		# Instructions		# Actions		# Ports	
			Req	Opt	Req	Opt	Req	Opt	Req	Opt
v 1.0	Ingress Port	Per table statistics	18	2	1	0	2	11	6	2
	Ethernet: src, dst, type, VLAN	Per flow statistics								
	IPv4: src, dst, proto, ToS	Per port statistics								
	TCP/UDP: src port, dst port	Per queue statistics								
v 1.1	Metadata, SCTP, VLAN tagging	Group statistics	23	2	0	0	3	28	5	3
	MPLS: label, traffic class	Action bucket statistics								
v 1.2	OpenFlow Extensible Match (OXM)		14	18	2	3	2	49	5	3
	IPv6: src, dst, flow label, ICMPv6									
v 1.3	PBB, IPv6 Extension Headers	Per-flow meter	14	26	2	4	2	56	5	3
		Per-flow meter band								
v 1.4	—	—	**14**	**27**	**2**	**4**	**2**	**57**	**5**	**3**
		Optical port properties								

Source: Kreutz et al.: Software-Defined Networking: A Comprehensive Survey, Proceedings of IEEE 2015

Openflow enabled hardware and software

Group	Product	Type	Maker/Developer	Version	Short description
Hardware	8200zl and 5400zl [125]	chassis	Hewlett-Packard	v1.0	Data center class chassis (switch modules).
	Arista 7150 Series [126]	switch	Arista Networks	v1.0	Data centers hybrid Ethernet/OpenFlow switches.
	BlackDiamond X8 [127]	switch	Extreme Networks	v1.0	Cloud-scale hybrid Ethernet/OpenFlow switches.
	CX600 Series [128]	router	Huawei	v1.0	Carrier class MAN routers.
	EX9200 Ethernet [129]	chassis	Juniper	v1.0	Chassis based switches for cloud data centers.
	EZchip NP-4 [130]	chip	EZchip Technologies	v1.1	High performance 100-Gigabit network processors.
	MLX Series [131]	router	Brocade	v1.0	Service providers and enterprise class routers.
	NoviSwitch 1248 [124]	switch	NoviFlow	v1.3	High performance OpenFlow switch.
	NetFPGA [48]	card	NetFPGA	v1.0	1G and 10G OpenFlow implementations.
	RackSwitch G8264 [132]	switch	IBM	v1.0	Data center switches supporting Virtual Fabric and OpenFlow.
	PF5240 and PF5820 [133]	switch	NEC	v1.0	Enterprise class hybrid Ethernet/OpenFlow switches.
	Pica8 3920 [134]	switch	Pica8	v1.0	Hybrid Ethernet/OpenFlow switches.
	Plexxi Switch 1 [135]	switch	Plexxi	v1.0	Optical multiplexing interconnect for data centers.
	V330 Series [136]	switch	Centec Networks	v1.0	Hybrid Ethernet/OpenFlow switches.
	Z-Series [137]	switch	Cyan	v1.0	Family of packet-optical transport platforms.
Software	contrail-vrouter [138]	vrouter	Juniper Networks	v1.0	Data-plane function to interface with a VRF.
	LINC [139], [140]	switch	FlowForwarding	v1.4	Erlang-based soft switch with OF-Config 1.1 support.
	ofsoftswitch13 [141]	switch	Ericsson, CPqD	v1.3	OF 1.3 compatible user-space software switch implementation.
	Open vSwitch [142], [109]	switch	Open Community	v1.0-1.3	Switch platform designed for virtualized server environments.
	OpenFlow Reference [143]	switch	Stanford	v1.0	OF Switching capability to a Linux PC with multiple NICs.
	OpenFlowClick [144]	vrouter	Yogesh Mundada	v1.0	OpenFlow switching element for Click software routers.
	Switch Light [145]	switch	Big Switch	v1.0	Thin switching software platform for physical/virtual switches.
	Pantou/OpenWRT [146]	switch	Stanford	v1.0	Turns a wireless router into an OF-enabled switch.
	XorPlus [46]	switch	Pica8	v1.0	Switching software for high performance ASICs.

Source: Kreutz et al.: Software-Defined Networking: A Comprehensive Survey, Proceedings of IEEE 2015

Open vSwitch

Figure 1. Traditional networking infrastructure

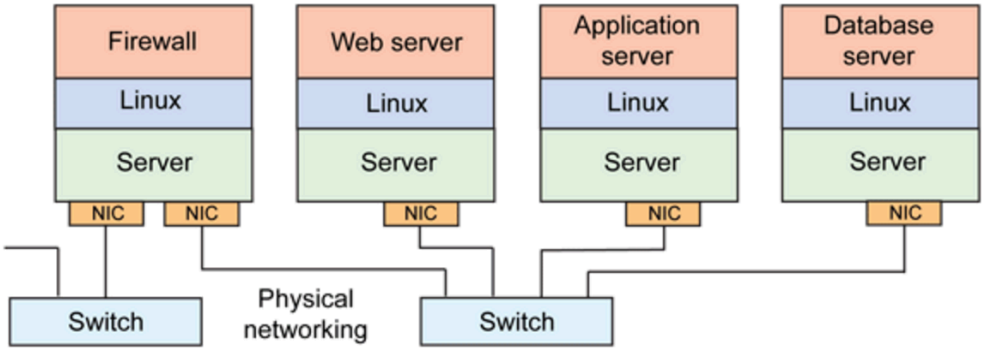
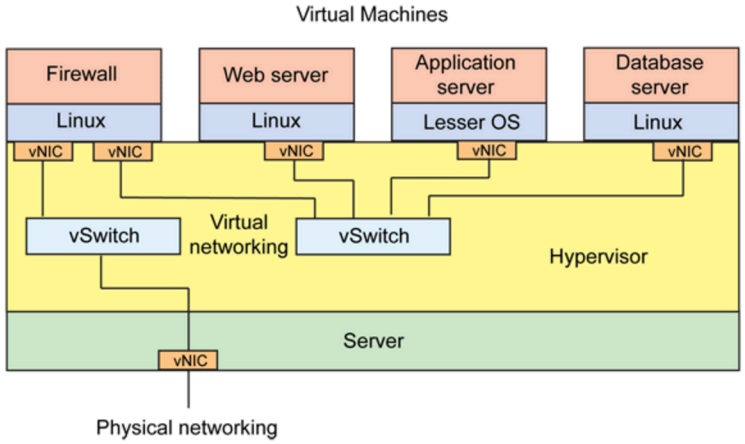


Figure 2. Virtualized networking infrastructure



- The Open vSwitch is a virtual switch that's available as open source under the Apache 2.0 license.
- Open vSwitch supports the leading open source hypervisor solutions, including Kernel-based VM (KVM), VirtualBox, Xen, and XenServer. It's also a drop-in replacement for the current Linux bridge module.
- Open vSwitch consists of a switch daemon and companion kernel module that manages the flow-based switching. A variety of other daemons and utilities also exist for managing the switch (particularly from the perspective of OpenFlow).

Source: https://www.ibm.com/developerworks/linux/library/l-virtual-networking/