SDN architecture and its fundamental abstractions

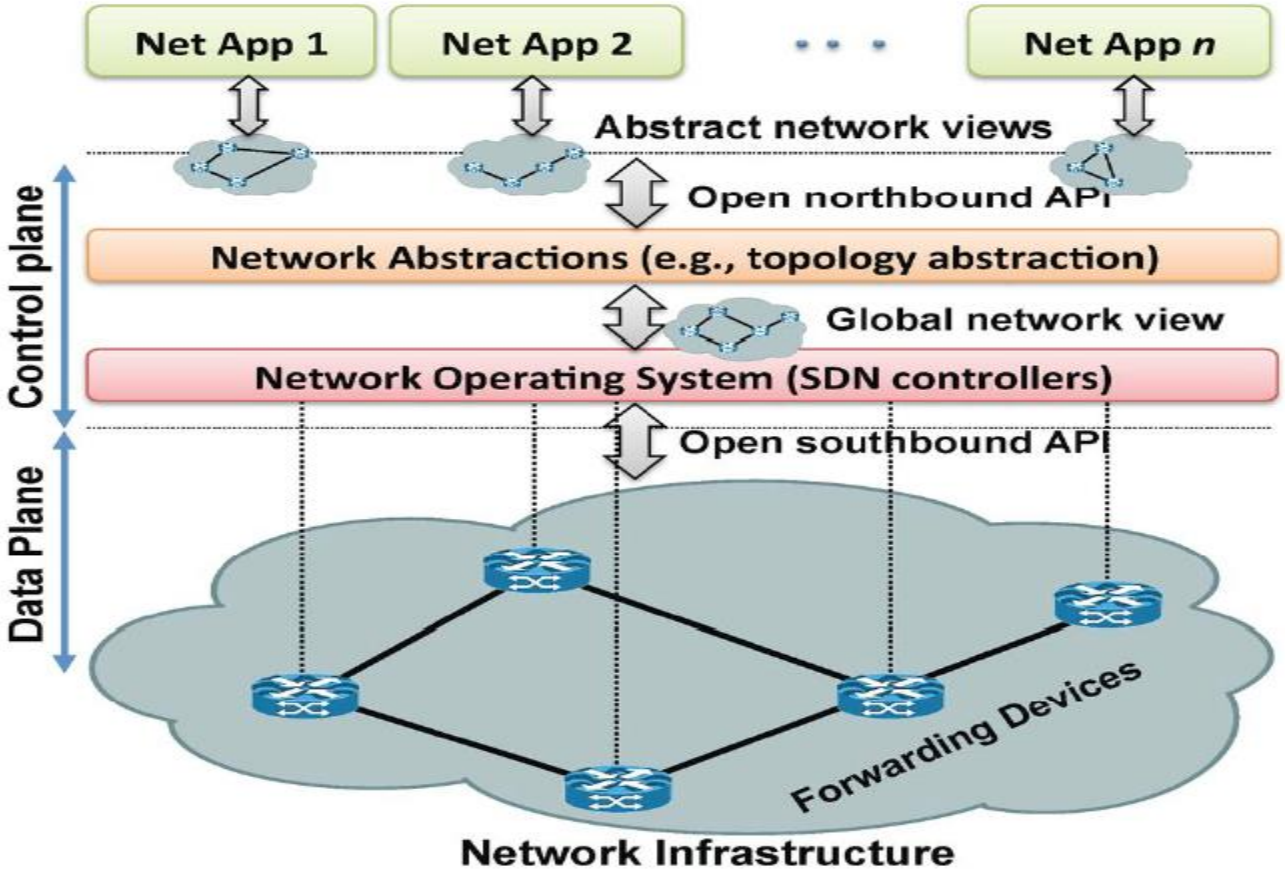
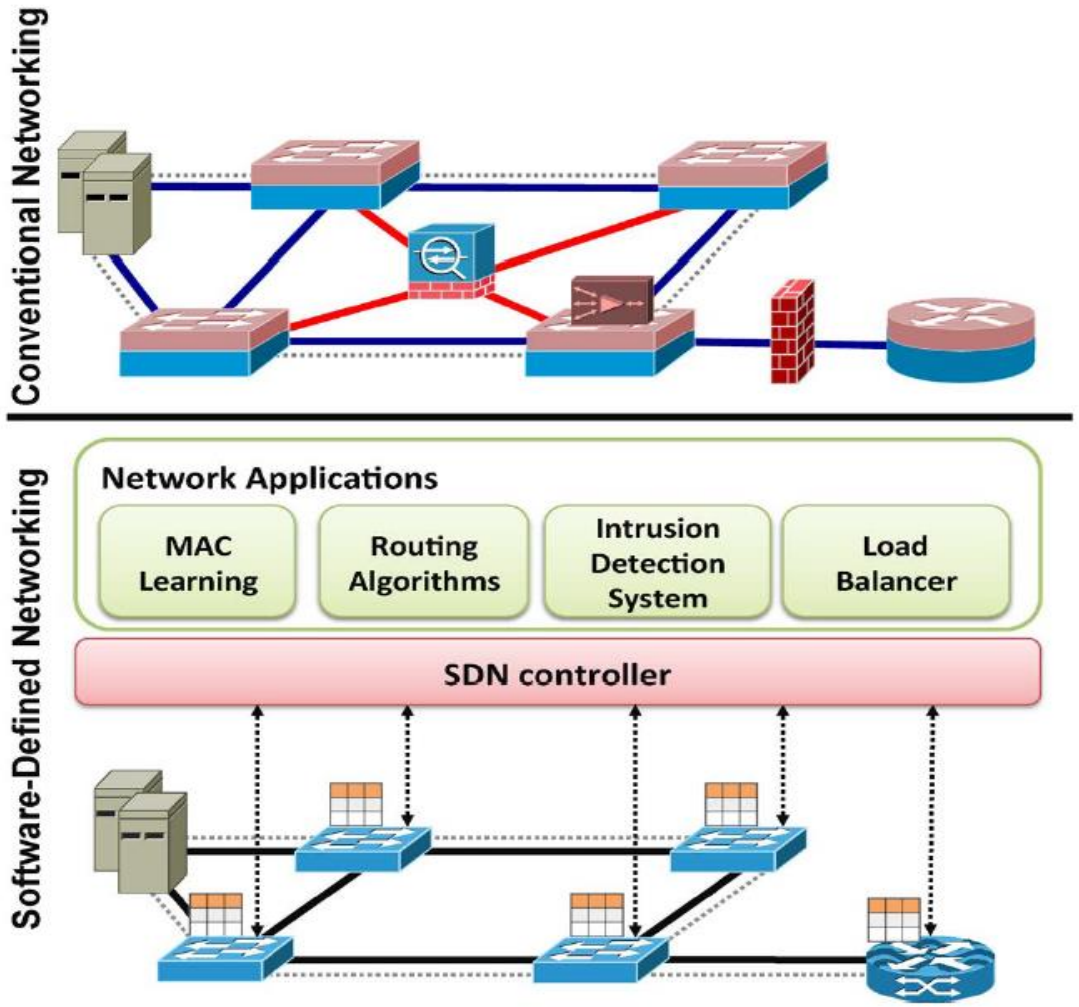


Fig. 4. *SDN architecture and its fundamental abstractions.*

Traditional networking versus SDN



Distributed controllers: east/westbound APIs

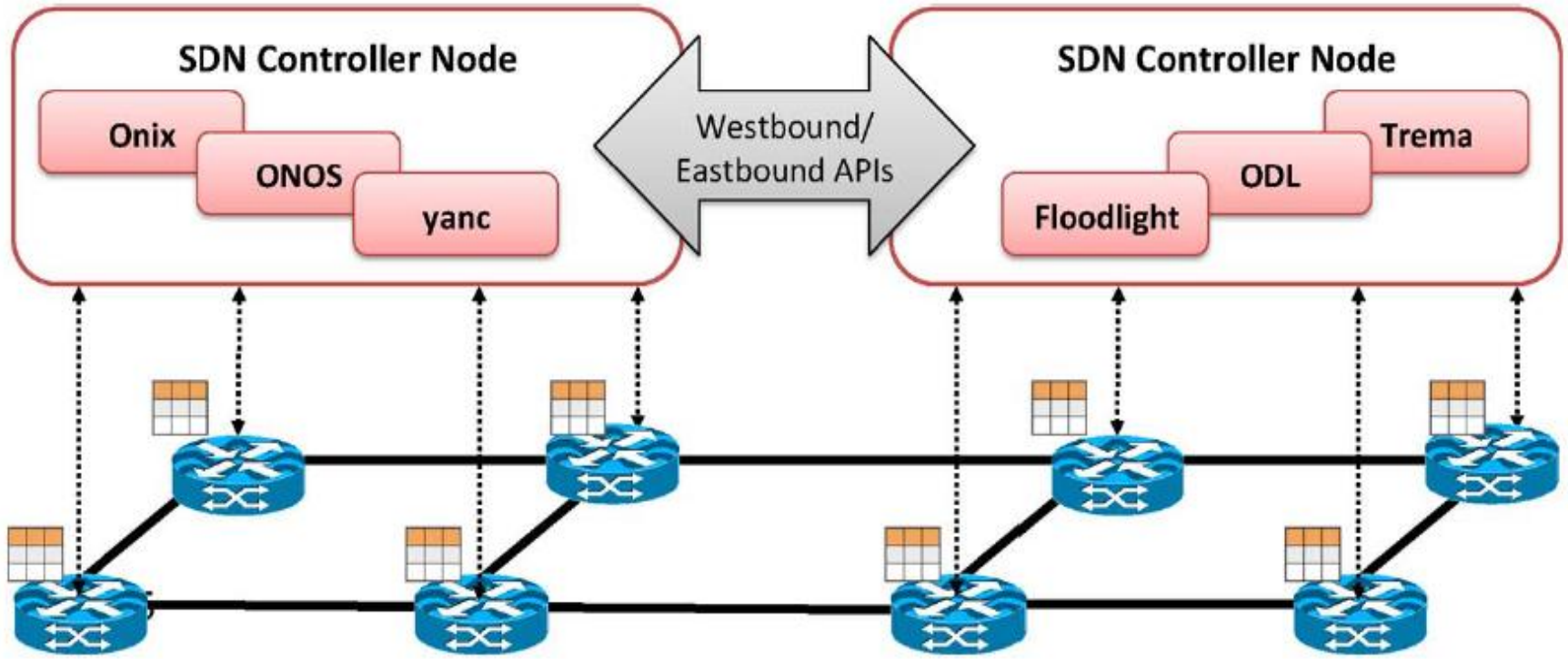


Fig. 9. ***Distributed controllers: east/westbound APIs.***

Standard Bodies

- Open Networking Foundation
  - http://www.openflow.org/
  - https://www.opennetworking.org/

- IETF
  - http://tools.ietf.org/html/draft-nadeau-sdn-problem-statement-00
  - http://tools.ietf.org/html/draft-nadeau-sdn-framework-0

Benefits of SDN

- Ease Innovation in Network
- Experiment and research using non-expensive equipment
- More accessibility since software can be easily developed
- Quick deployment – No hardware fabrication cycles
- More Flexibility with programmability
- Ease of customization and integration with other software applications
- Fast upgrades
- Program a network Vs Configure a network

Software Defined Networking

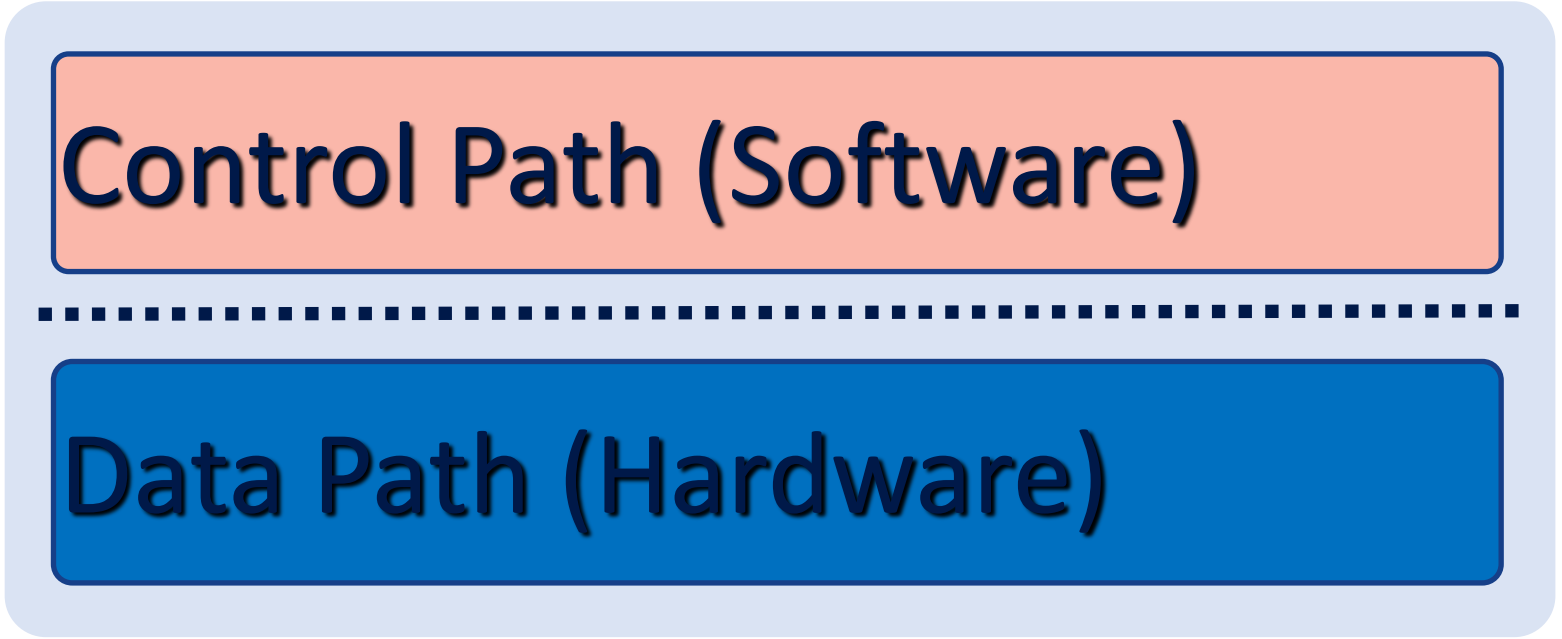
- **Questions:**

  - How to obtain global information?
  - What are the configurations?
  - How to implement?
  - How is the scalability?
  - How does it really work?

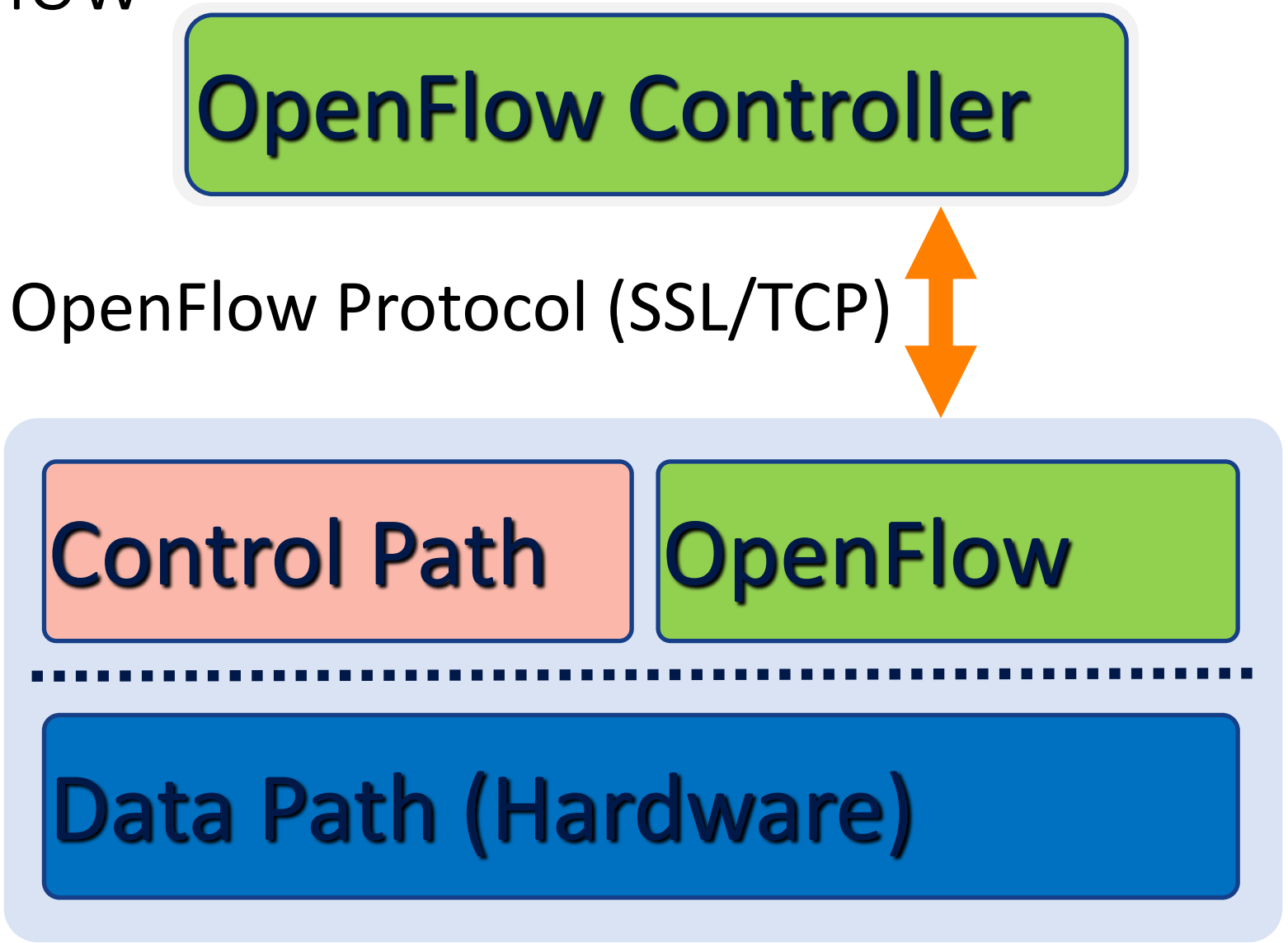
OpenFlow

- **Like hardware drivers**
  - interface between switches and Network OS

OpenFlow

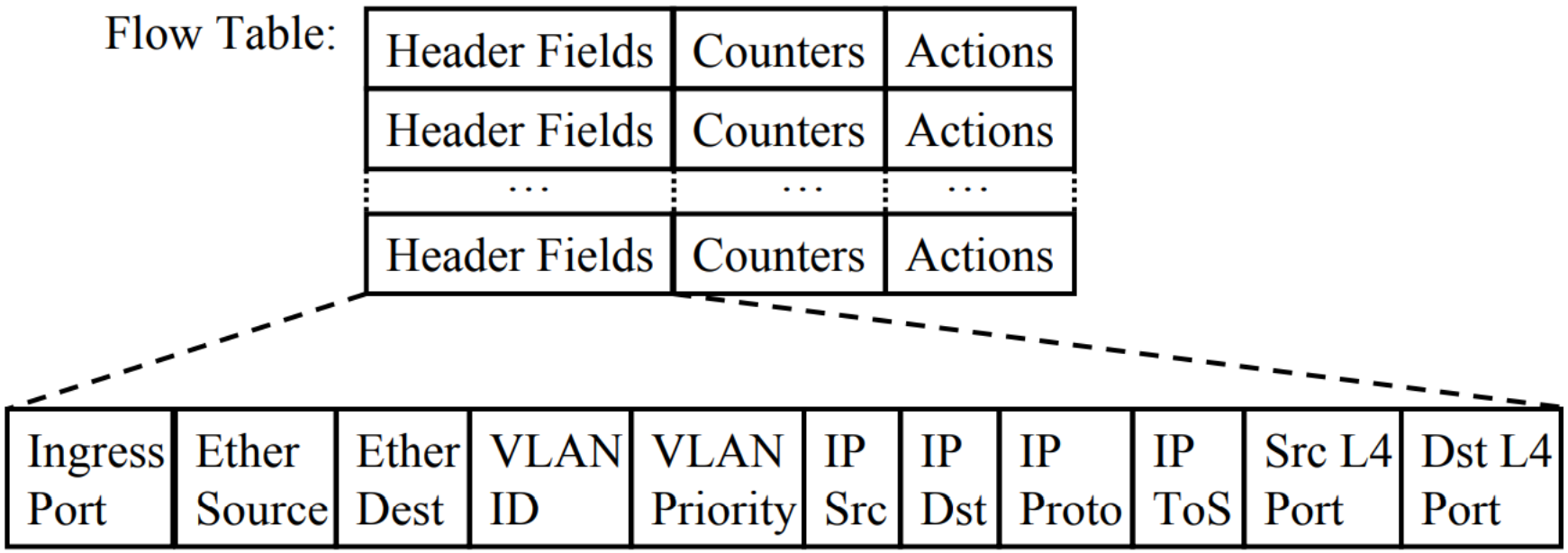


OpenFlow


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

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.



Source: Prof. Raj Jain's OpenFlow slides

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 received for this time
- Hard timeout: Remove entry after this time
- If both are set, the entry is removed if either one expires.

Source: Prof. Raj Jain's OpenFlow slides

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	

Source: Prof. Raj Jain's OpenFlow slides

Actions

- Forward to Physical Port *i* or to *Virtual Port*:
  - **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 flow table
  - **In_port**: Send back to input port
  - **Normal**: Forward using traditional Ethernet
  - **Flood**: Send along minimum spanning tree except the incoming interface
- Enqueue: To a particular queue in the port ⇒ QoS
- Drop
- Modify Field: E.g., add/remove VLAN tags, ToS bits, Change TTL

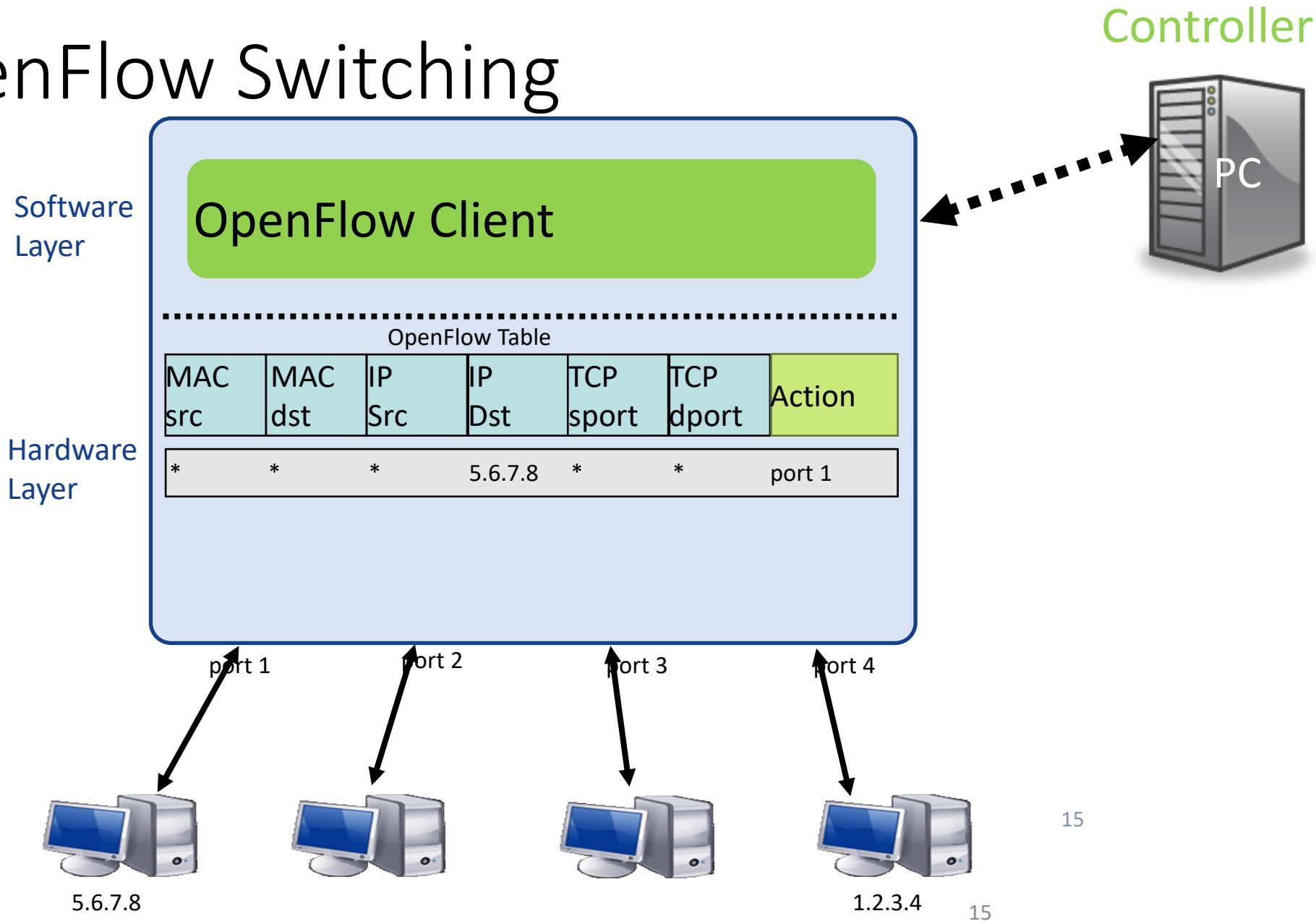
Source: Prof. Raj Jain's OpenFlow slides

Actions Cont..

- Masking allows matching only selected fields,
  - e.g., Dest. IP, Dest. MAC, etc.
- If header matches an entry, corresponding actions are performed and counters are updated.
- If no header match, 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 controller and the switch using TLS
- Modern switches already implement flow tables, typically using Ternary Content Addressable Memories (TCAMs)
- Controller can send flow table entries beforehand (Proactive) or Send on demand (Reactive). OpenFlow allows both models.

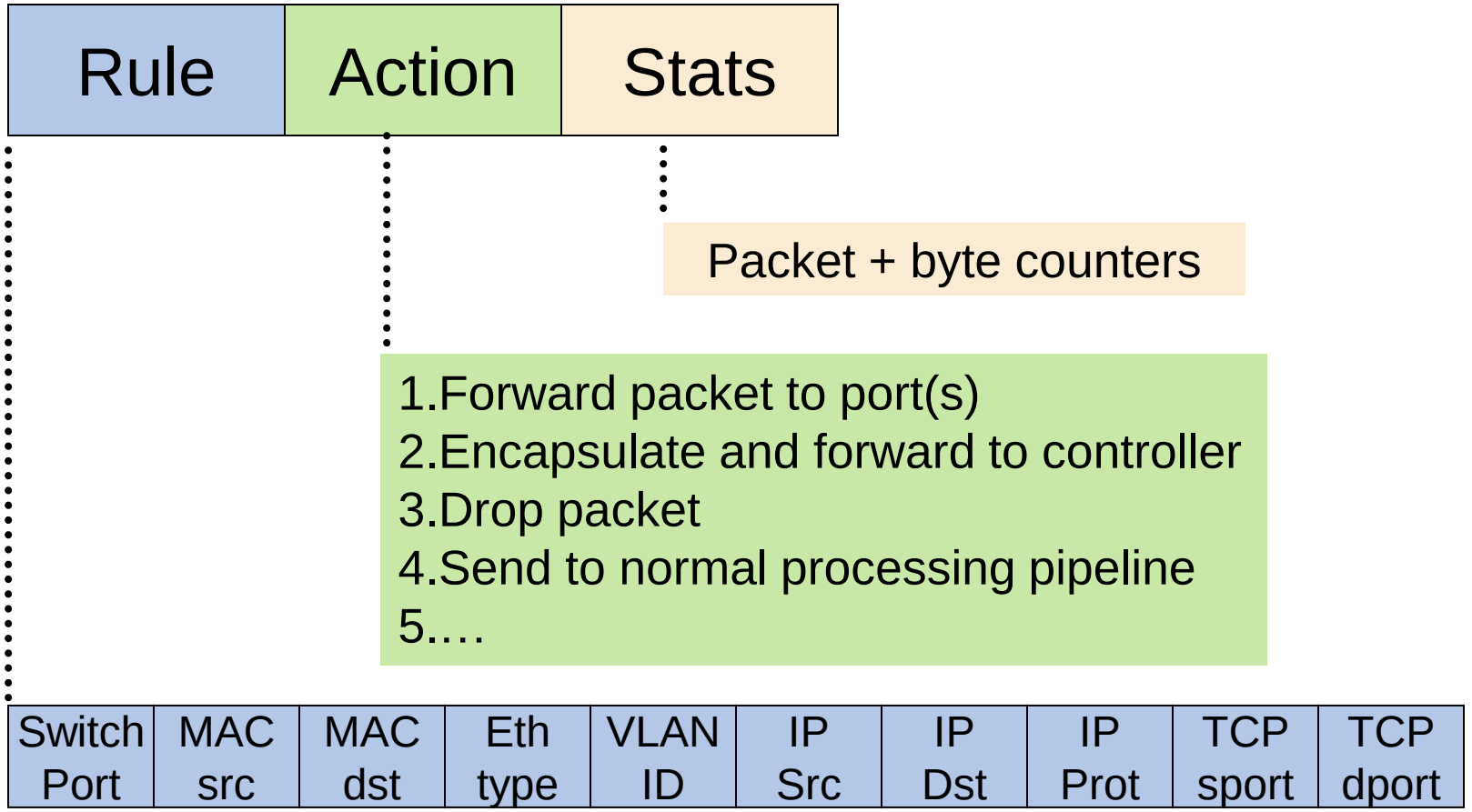
Source: Prof. Raj Jain's OpenFlow slides

OpenFlow Switching



15

OpenFlow Table Entry



OpenFlow Examples

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

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

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-enabled SDN devices

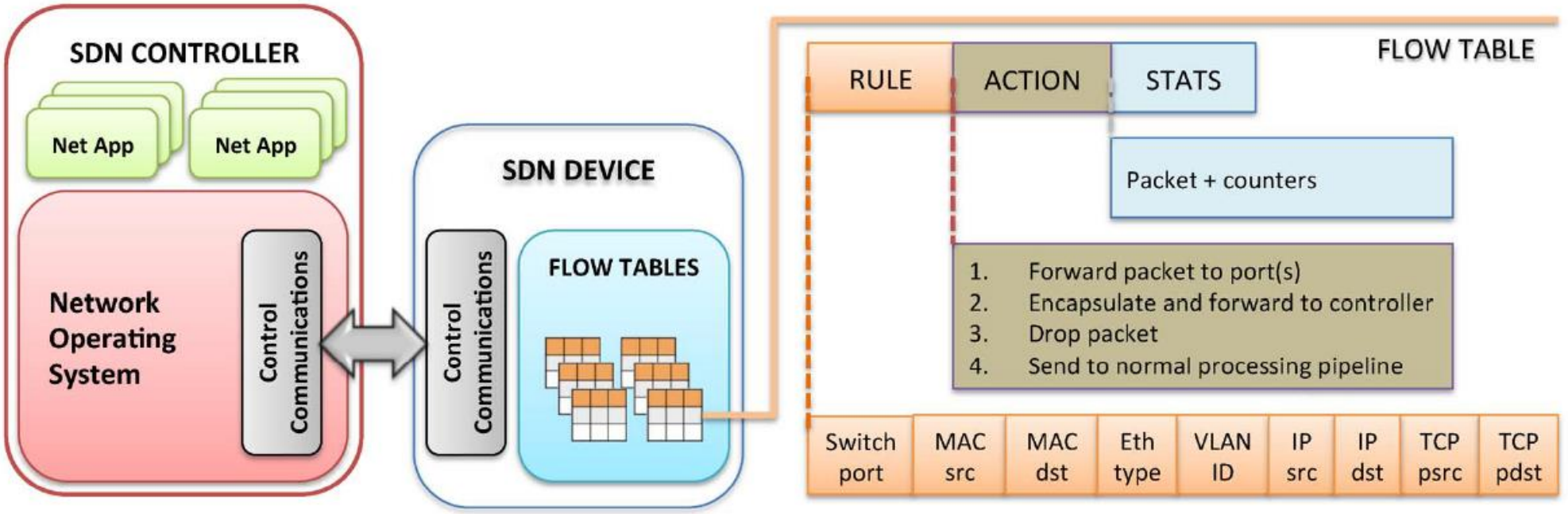
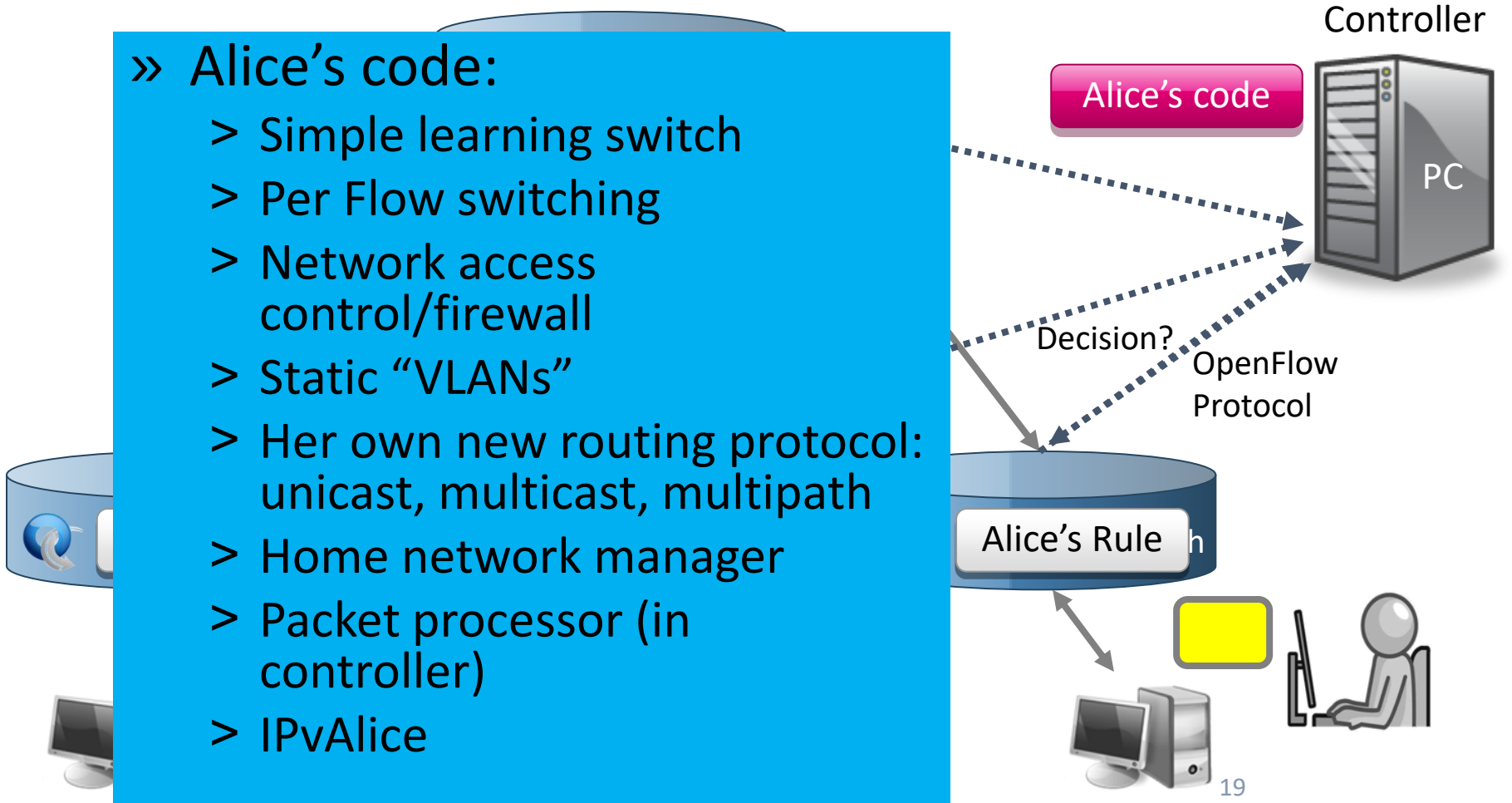


Fig. 7. ***OpenFlow-enabled SDN devices.***

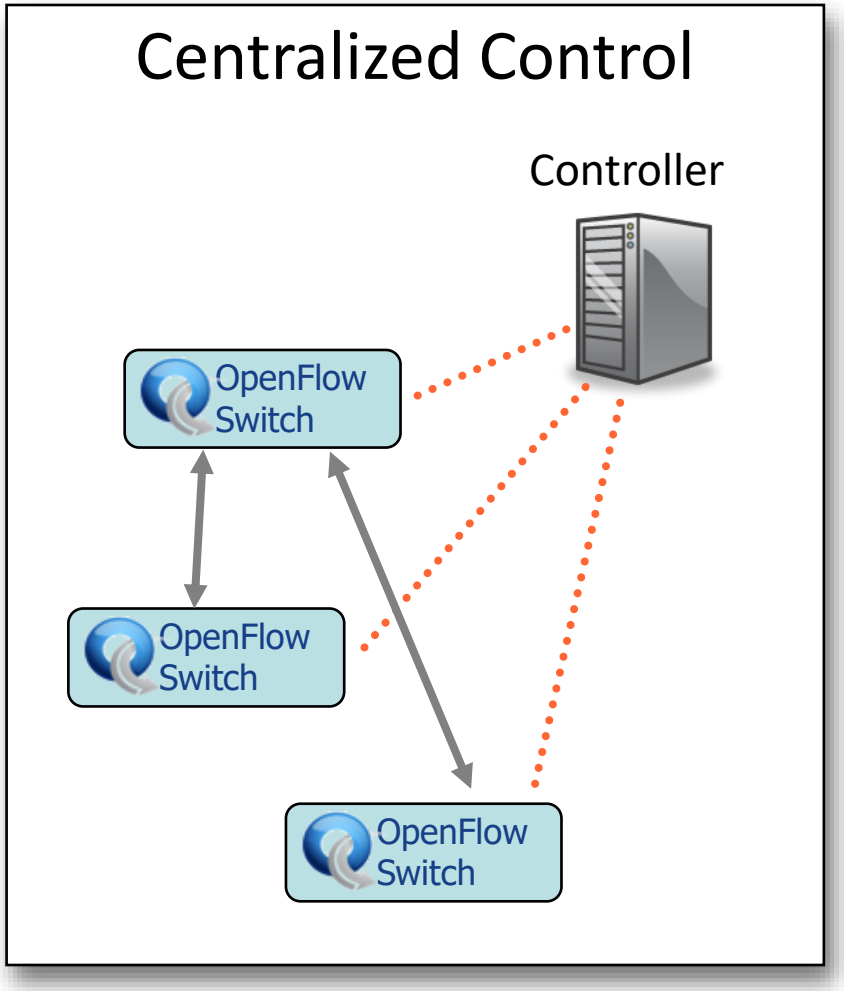
OpenFlow Usage

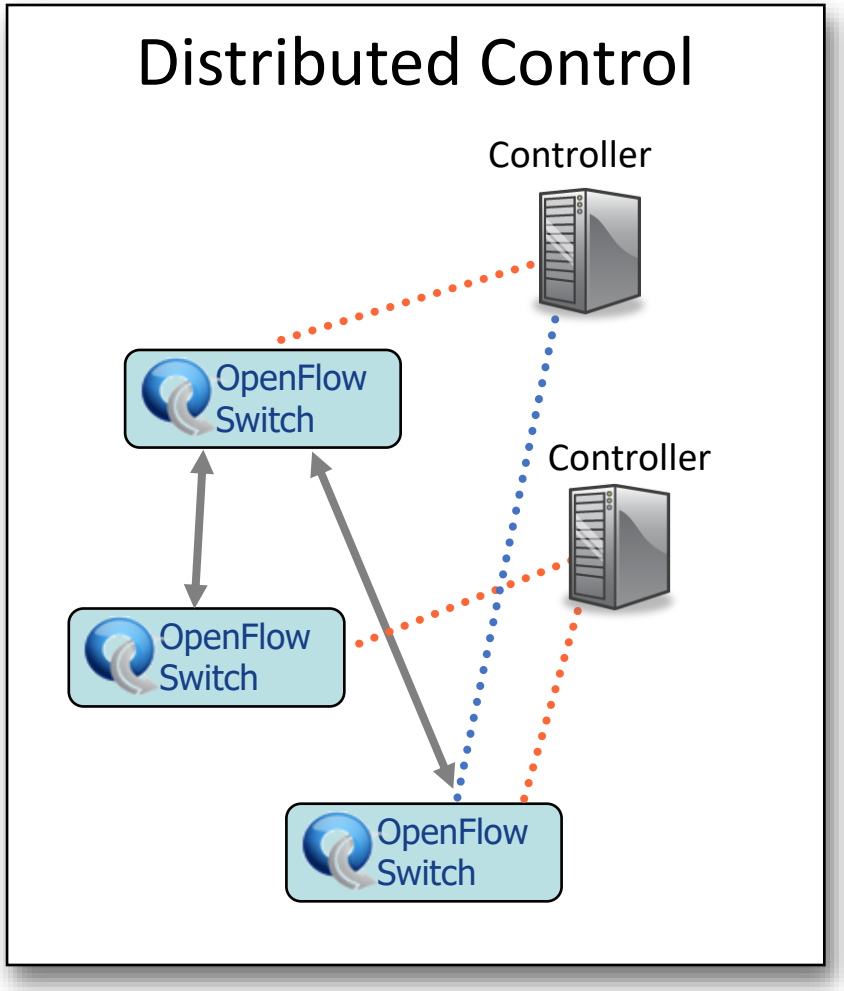
- Alice's code:
  - Simple learning switch
  - Per Flow switching
  - Network access control/firewall
  - Static "VLANs"
  - Her own new routing protocol: unicast, multicast, multipath
  - Home network manager
  - Packet processor (in controller)
  - IPvAlice



Centralized/Distributed Control

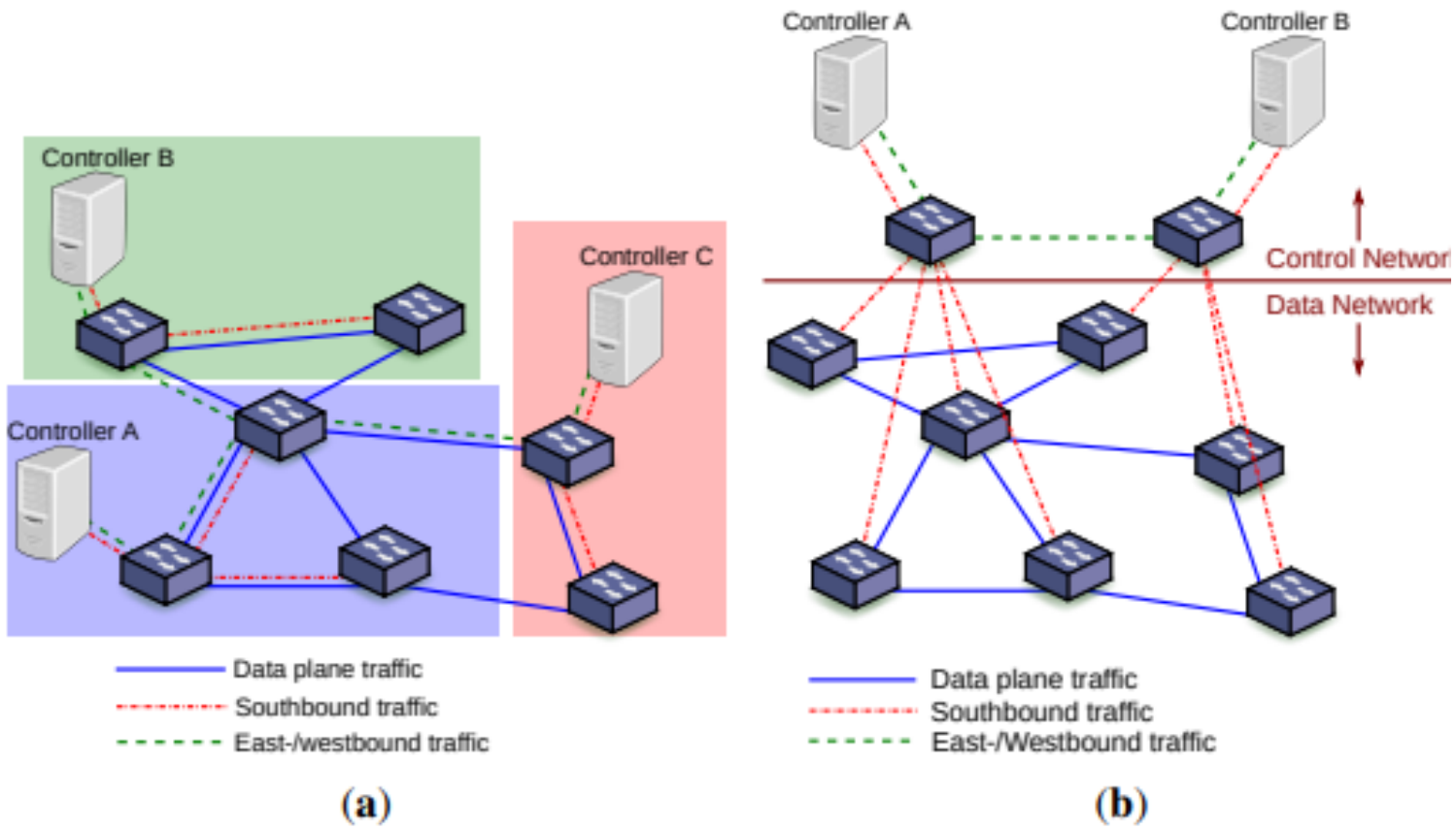
- **"Onix: A Distributed Control Platform for Large-scale Production Networks"**





Control plane signaling

Figure 9. (a) In-band signaling; (b) Out-of-band signaling.



Source: Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices, Future Internet

OpenFlow Cont..

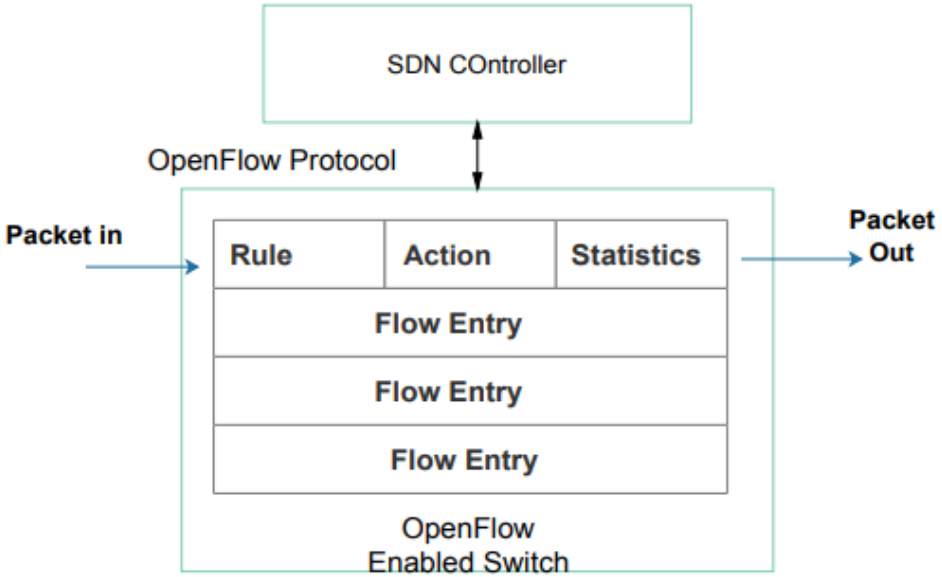


Figure 4: An abstract model of an OpenFlow switch.

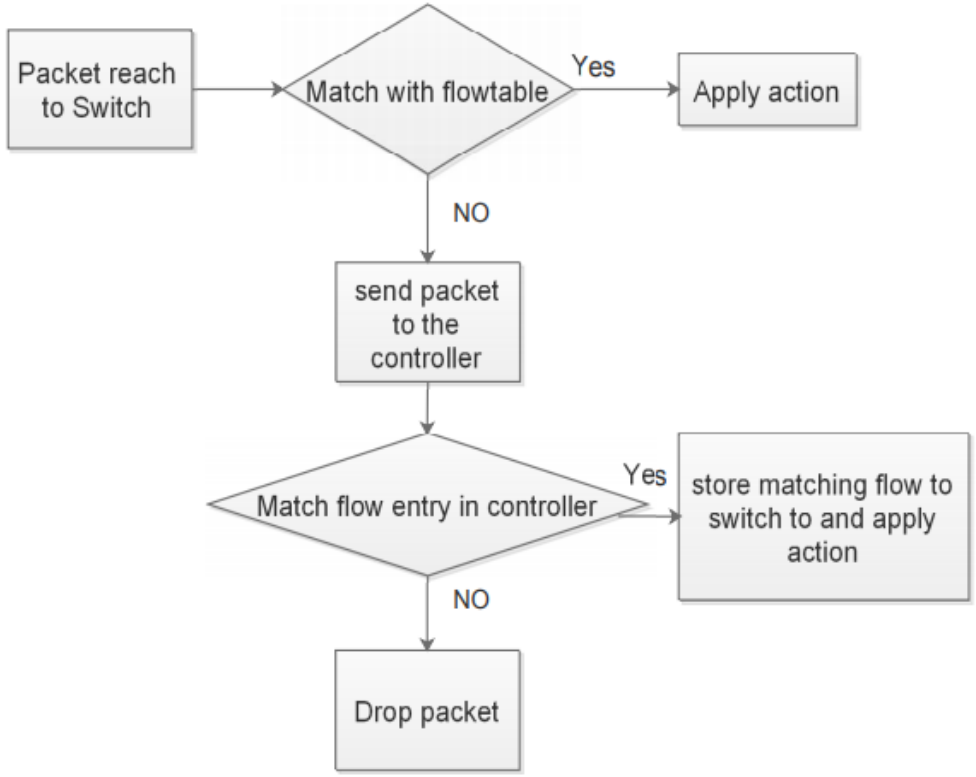


Figure 5: Flowchart of execution sequence of a packet in SDN.

Connection Est.

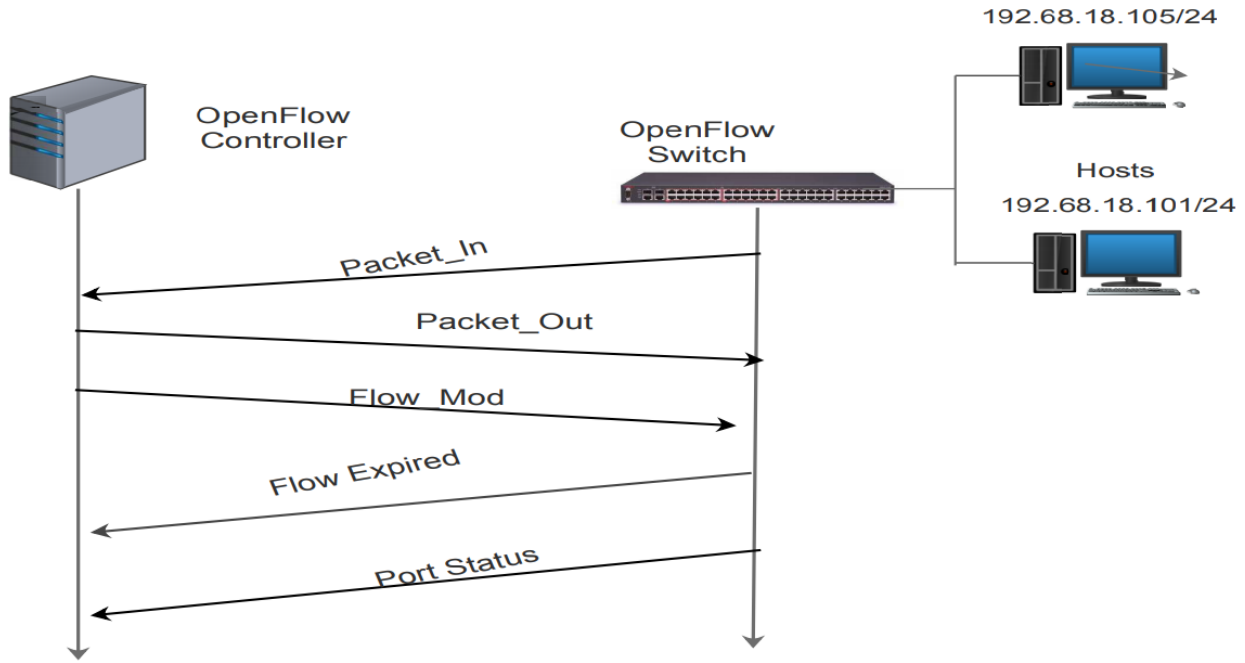
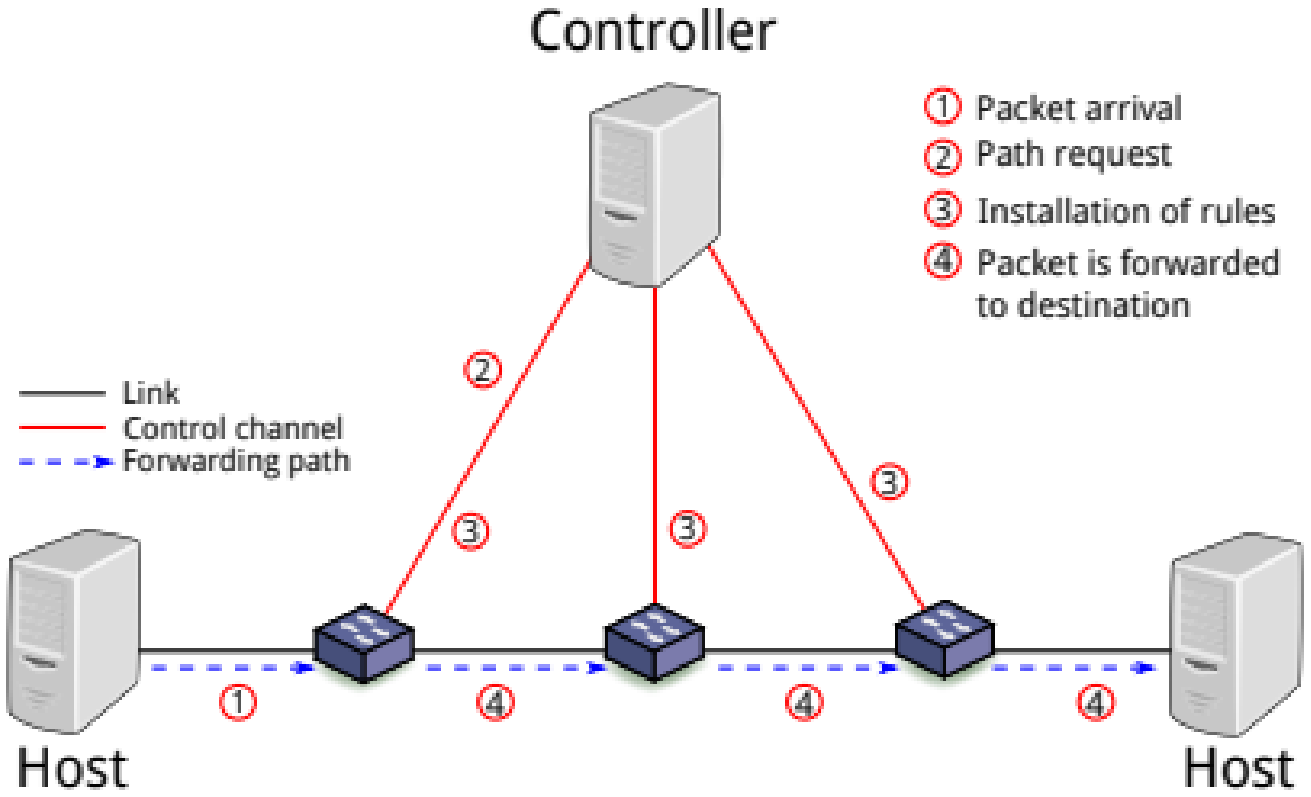


Figure 6: Connection establishment between hosts and the OpenFlow network.

Flow Management

Figure 10. Reactive flow management.



Source: Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices, Future Internet

Very Good Survey Paper on SDN

- Software-Defined Networking: A Comprehensive Survey by Diego Kreutz and others [Added in the google classroom]