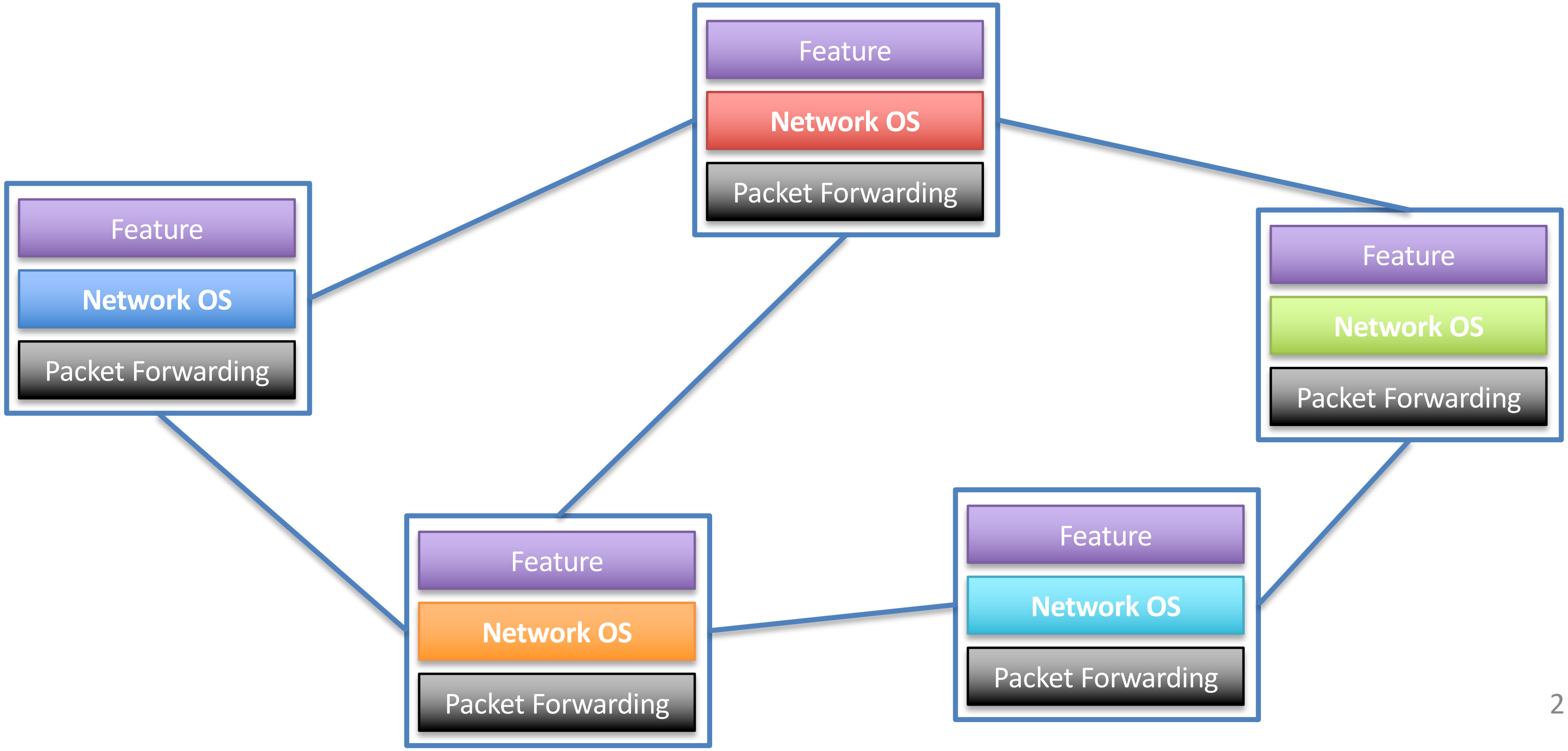
Basics of Software-Defined Networking (SDN)

Kotaro Kataoka

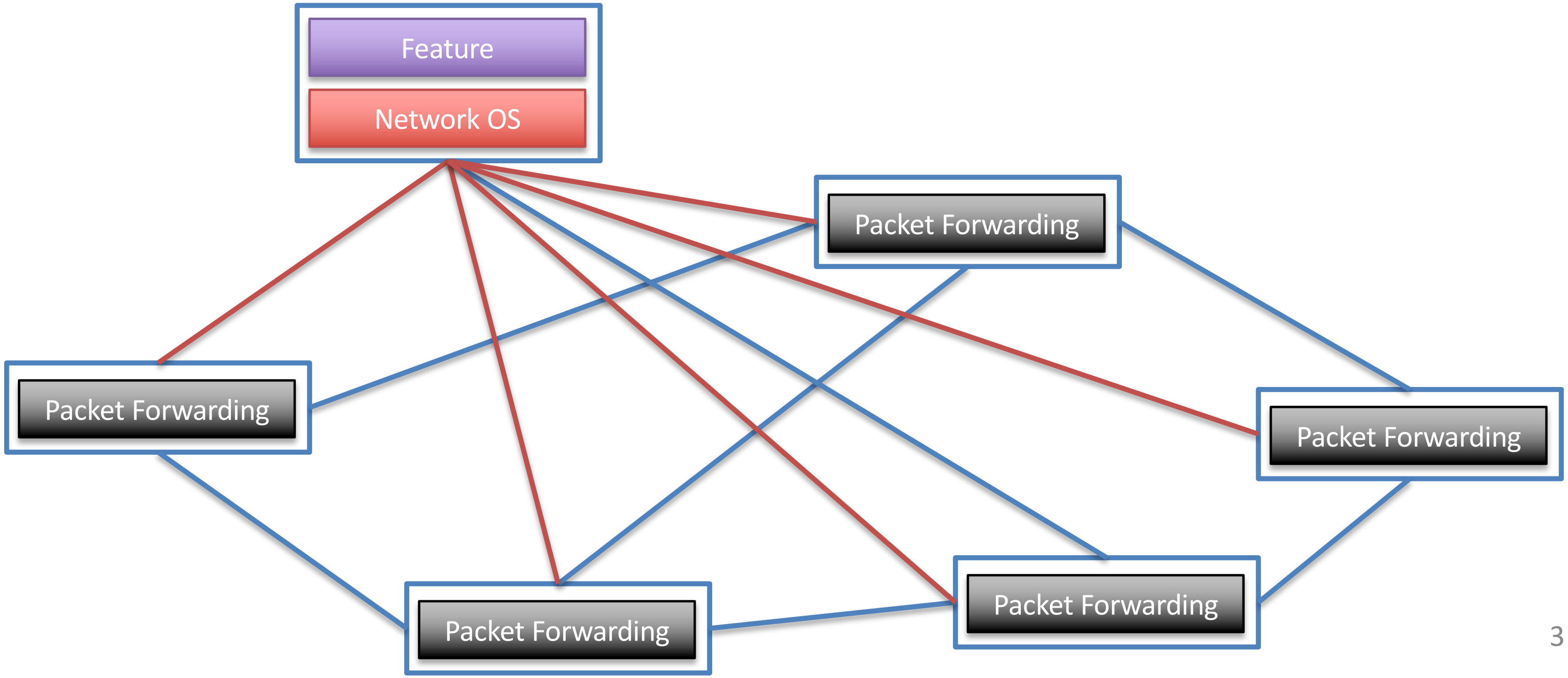
Traditional Network

- Feature, Network OS, Packet Forwarding in one-box
  - Configure every switch to extend VLAN across multiple switches
  - Scripting CLI commands? Yes, we do a lot. Interpreter between different network OSs? Yes, we can make it. But very costly.
  - Any additional feature, that is not implemented by OEM is not available



Software Defined Networks (SDN)

- Separated Feature, Network OS and Packet Forwarding
  - SDN Controller and SDN Switch
  - Making networks programmable
  - OEM independent networks



By the way, is SDN a brand new idea?

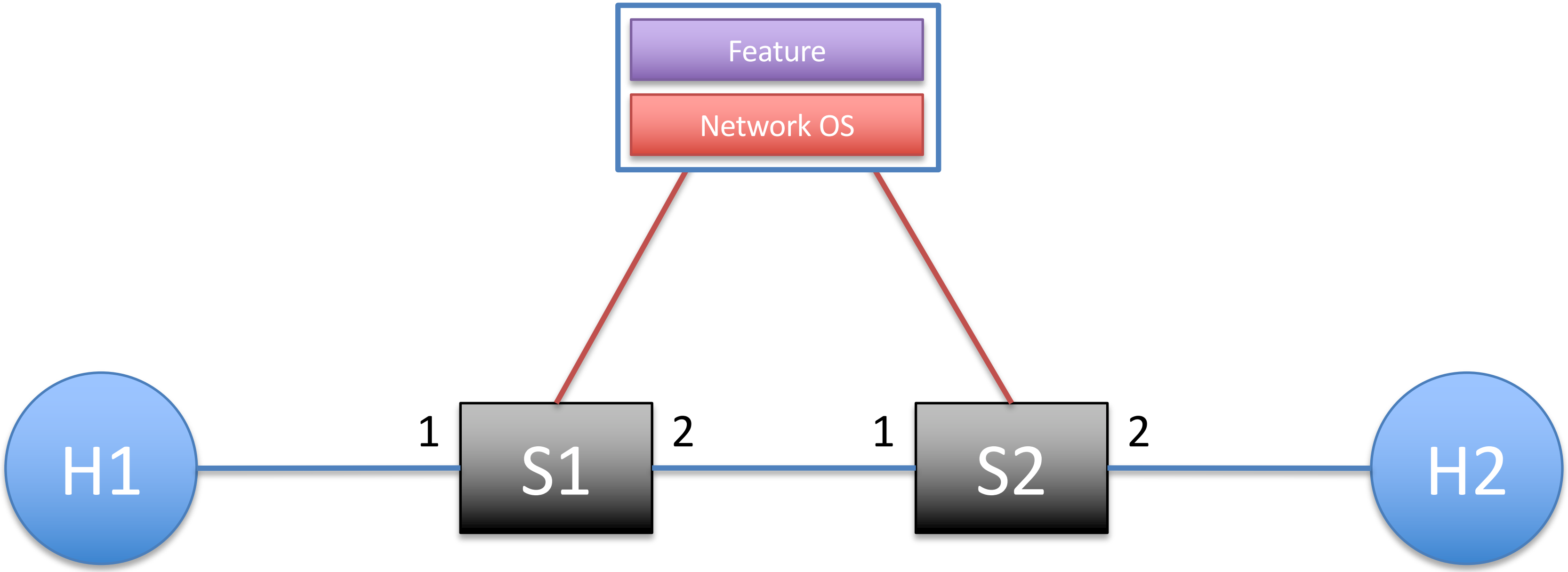
- No.

- Example: Let’s change the router configuration when some event is triggered on a monitoring server!

- Scripting/automating of router/switch configuration
  - Scripting SNMP Trap and Set
  - Configuration command as a part of script (expect and etc.)

Packet Forwarding in SDN (1/6)

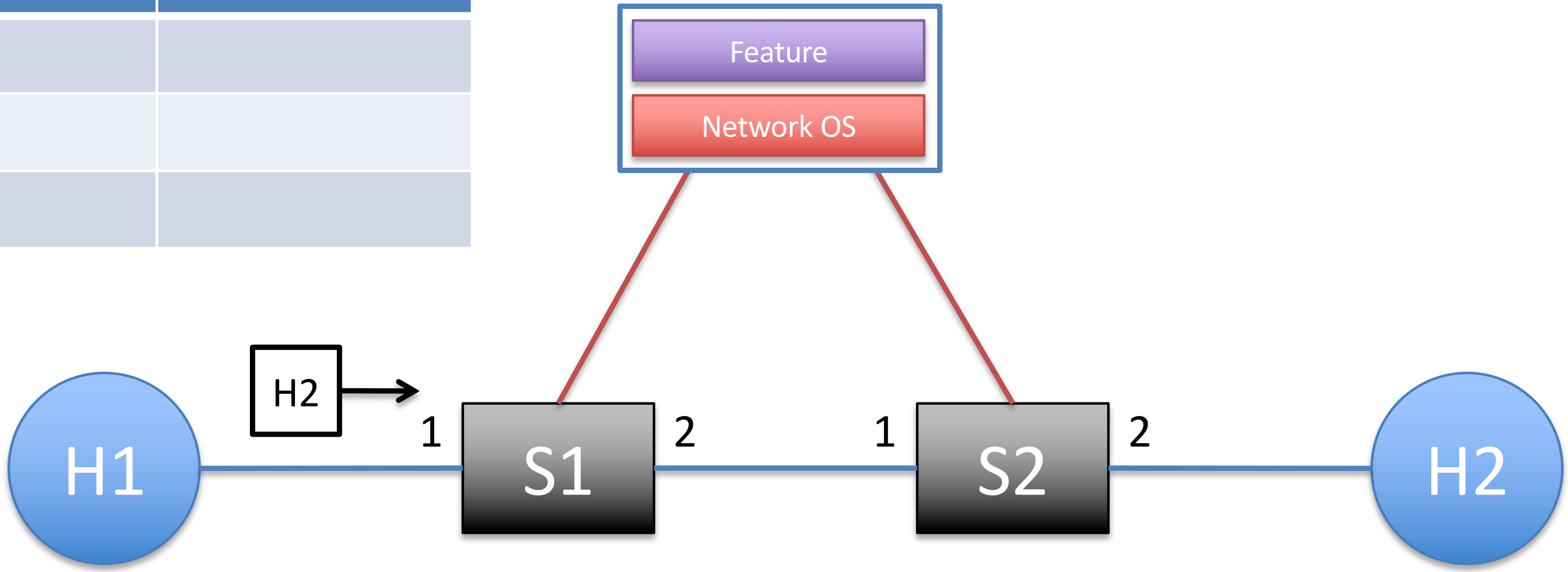
- Controller knows something
- S1, S2, H1, H2 are forming the network
- How can a packet from H1 reach H2?



Packet Forwarding in SDN (2/6)

- A packet is coming from H1 to S1
- S1 doesn’t know anything (no flow rule on S1)

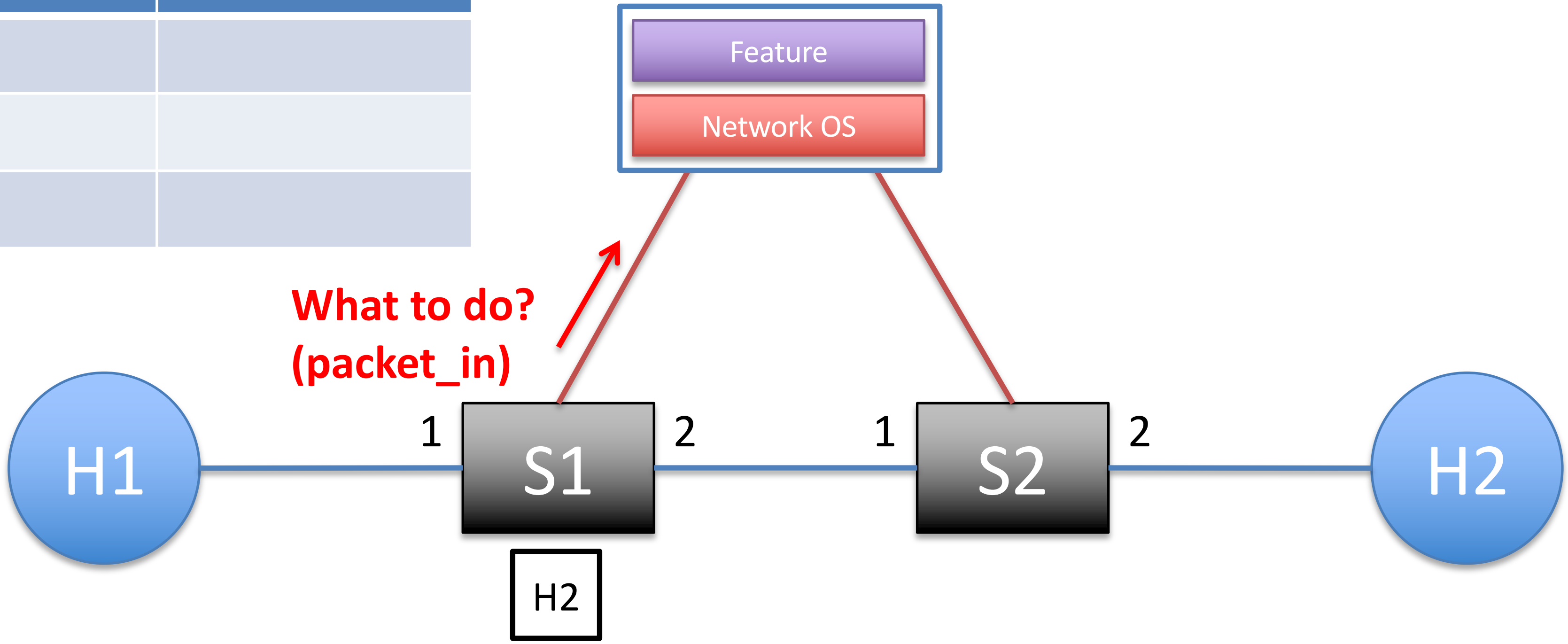
Match	Action



Packet Forwarding in SDN (3/6)

- S1 asks the controller what to do (packet_in)

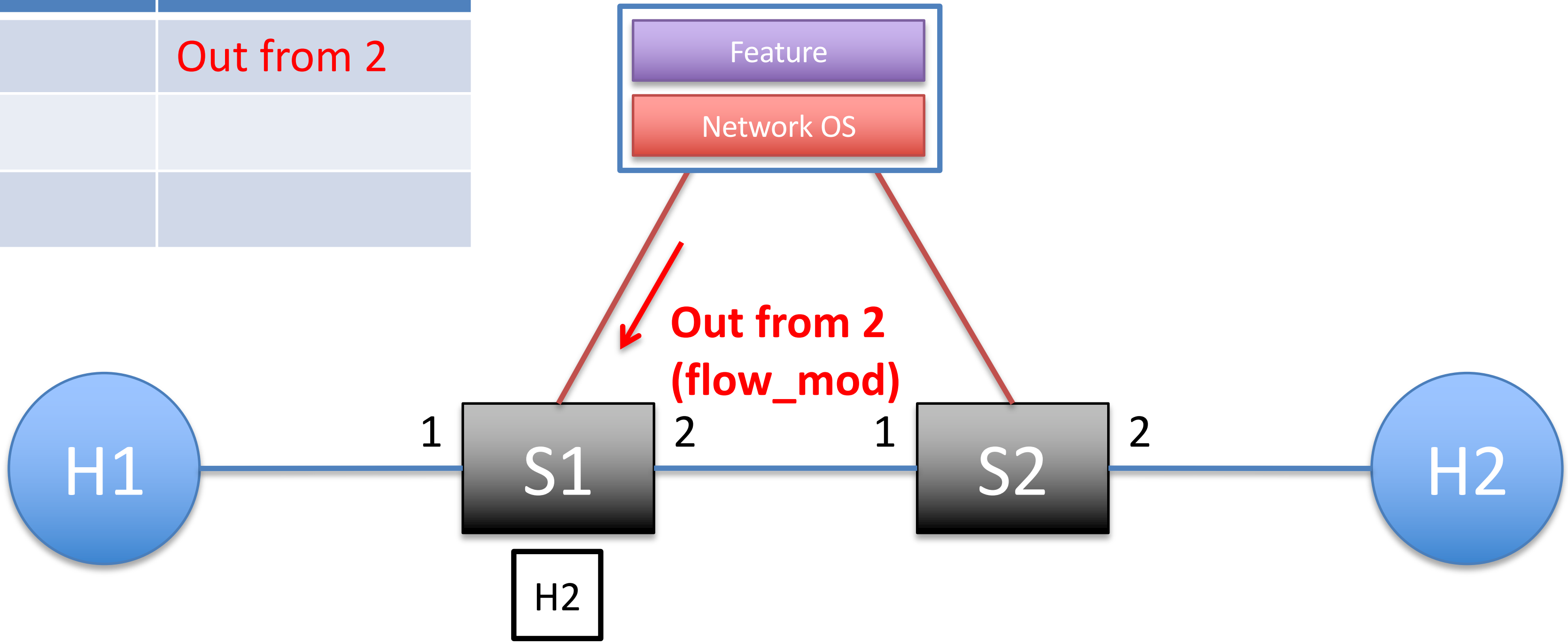
Match	Action



Packet Forwarding in SDN (4/6)

- Controller tells S1 that the packet is to be sent out from I/F 2 (flow_mod)
- S1 installs Flow Rules to Flow Table

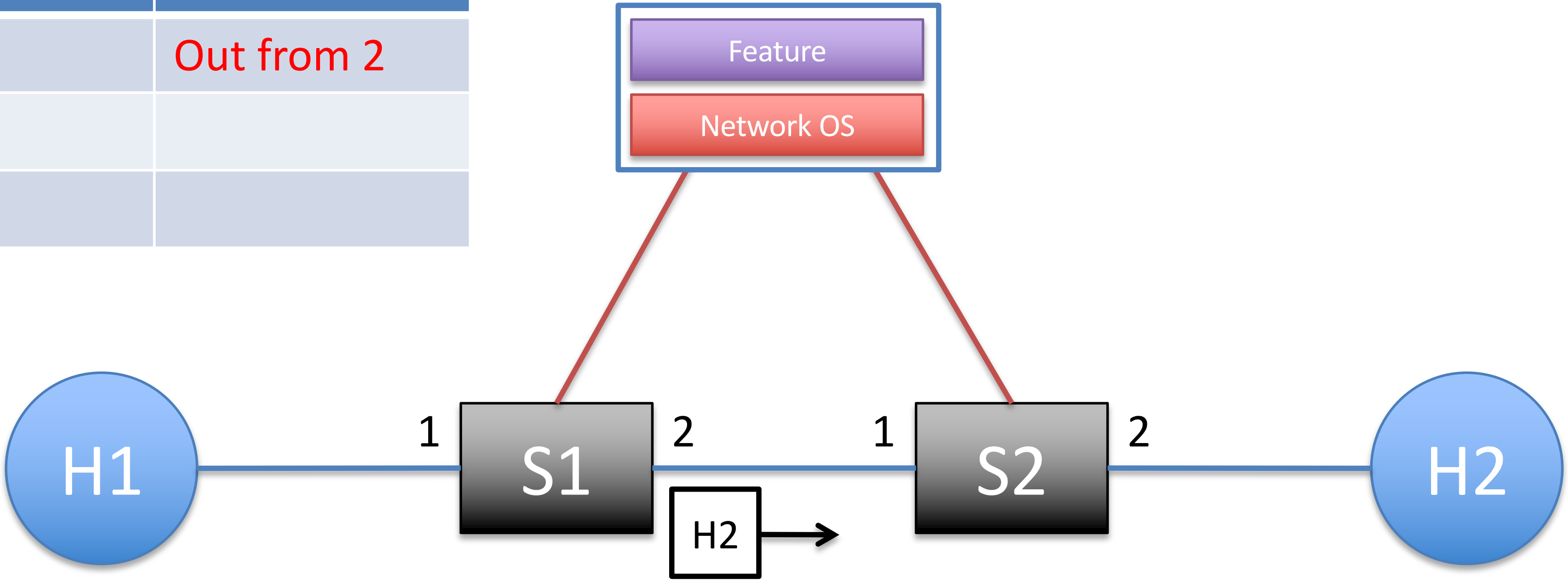
Match	Action
To H2	Out from 2



Packet Forwarding in SDN (5/6)

- S1 sends out the packet to H1 from I/F 2

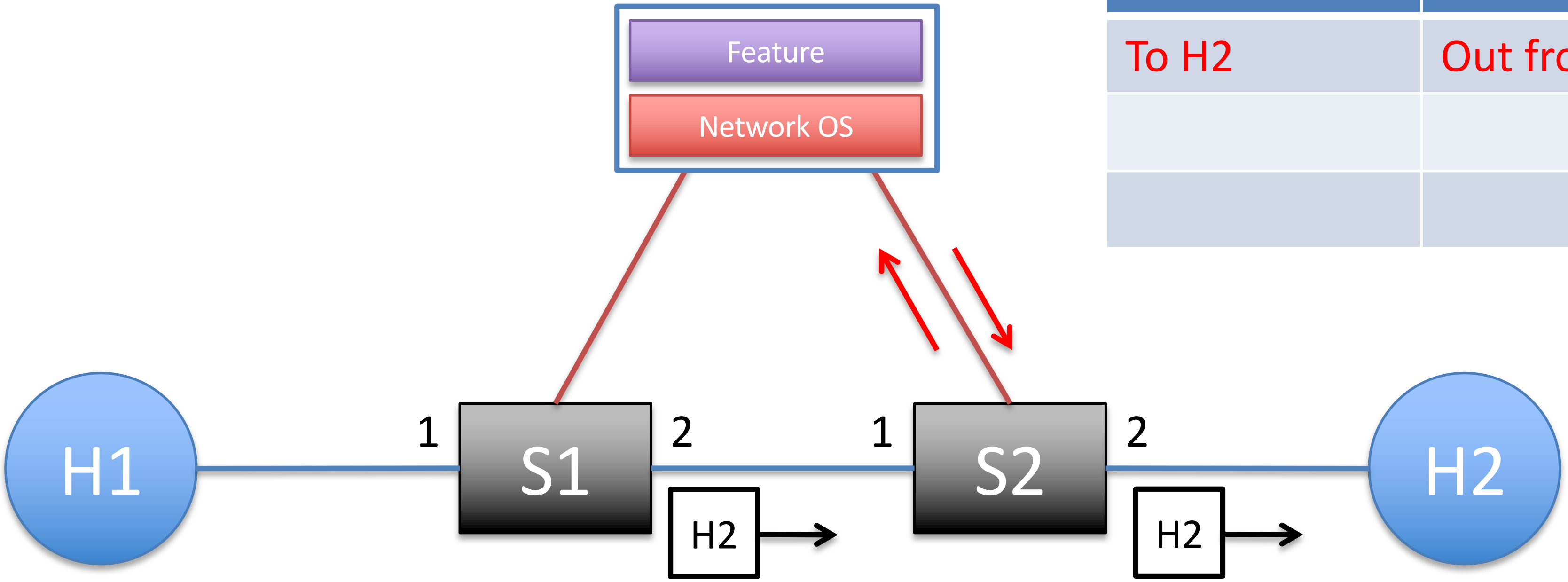
Match	Action
To H2	Out from 2



Packet Forwarding in SDN (6/6)

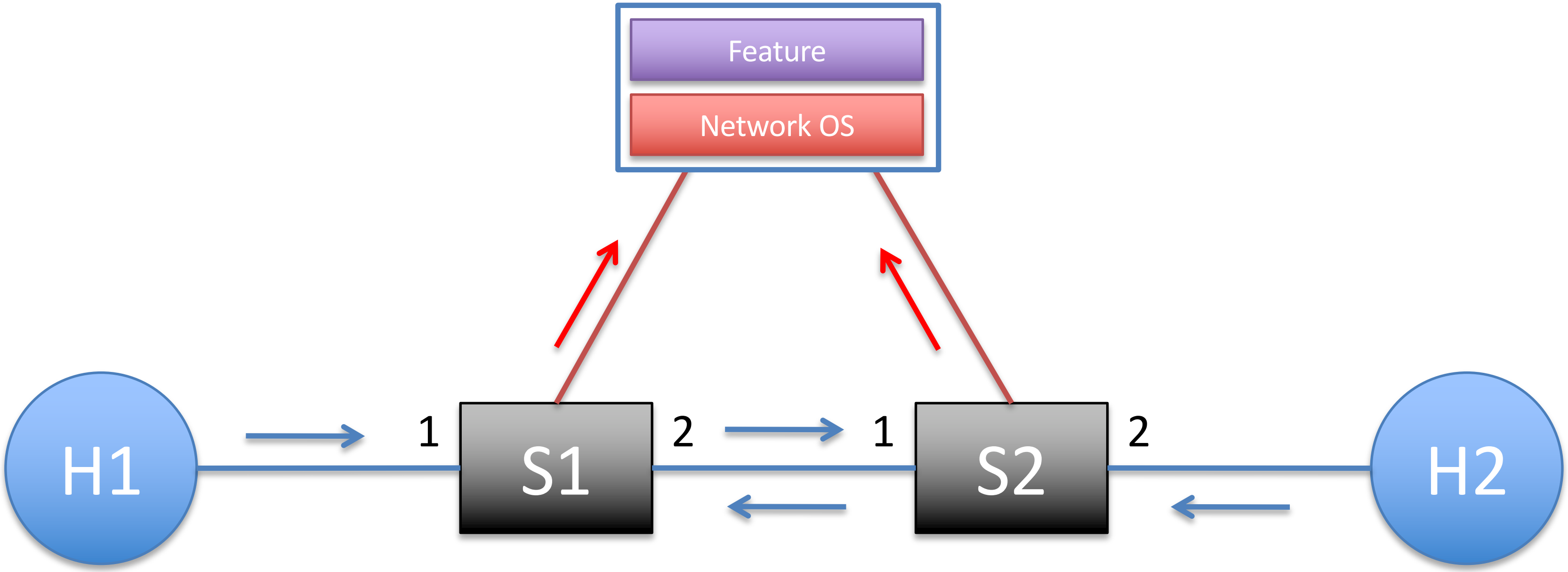
- S2 doesn’t know what t do again with the packet to H2
- S2 repeats the same process with S1

Match	Action
To H2	Out from 2



Global View of Network at Controller

- Topology Update
  - Which {switch / host} is connecting to which switch using which port?
  - SDN controller knows the network topology

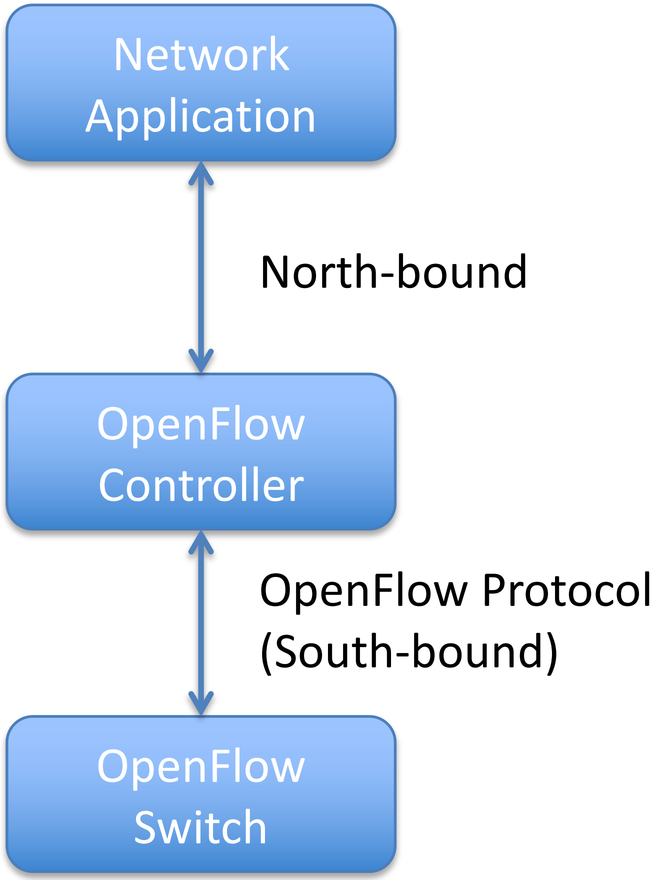


SDN Mechanism

- Control Plane
  - Communication between SDN Controller and Switch
  - Switch will forward a packet, which it doesn't know how to process, to Controller
  - Switch also gives statistics to Controller
  - Controller gives Flow Rule to tell Switch how to handle the packet
  - OpenFlow is popular protocol for Control Plane
  - Many OSS as options for SDN Controller
- Data Plane
  - Packet forwarding based on Flow Rule

OpenFlow: SDN Implementation

OpenFlow Overview



- Data Plane
  - Hardware abstracted as Flow Table
  - Handling packet forwarding
- Control Plane
  - Software that controls the data plane
  - Network applications (features) and networking OS
  - Determines the actions on flows

OpenFlow Overview

- Actions in OpenFlow
  - Forward packet to possible output(s)
    - Port #, Input Port, FLOOD/ALL, Controller using Packet-In, NORMAL using Switch Functions, LOCAL
  - Enqueue packet
  - Drop packet
  - Modify fields
- Flow Rules: Dataset defining a flow

H/W	**Layer 2 (Ethernet)**					**Layer 3 (IP)**				**L4 (TCP/UDP)**	
Input Port	Src MAC	Dest MAC	Ether Type	VLAN ID	VLAN Priority	Src Addr	Dest Addr	Proto	ToS	Src Port	Dest Port

Flow Table(1/2)

Match Fields	Actions	Counter

Table 1: Main Components of a flow entry in a flow table

Match Fields

- 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

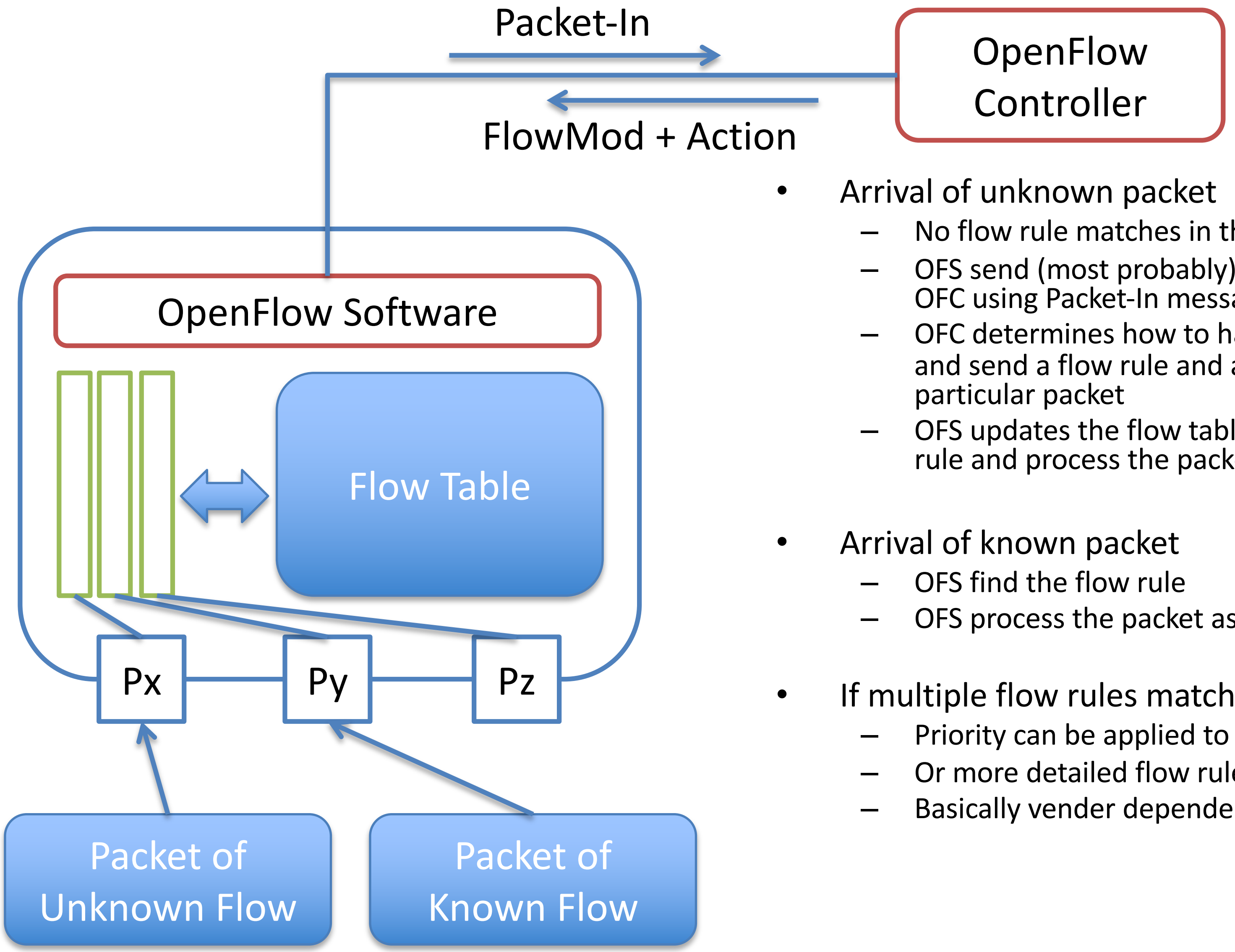
Actions

- Forward
- Enqueue
- Drop
- Modify Field

Flow Table(2/2)

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- **instructions**: to modify the action set or pipeline processing.
- **priority**: matching precedence of the flow entry.
- **counters**: to update for matching packets.
- **timeouts**: maximum amount of time or idle time before the flow is expired by the switch.
- **cookie**: used to add, modify and aggregate flows.

OpenFlow Switch Behavior



- Arrival of unknown packet
  - No flow rule matches in the flow table
  - OFS send (most probably) a part of packet to OFC using Packet-In message
  - OFC determines how to handle the packet and send a flow rule and action to the particular packet
  - OFS updates the flow table with the new flow rule and process the packet
- Arrival of known packet
  - OFS find the flow rule
  - OFS process the packet as flow rule defines
- If multiple flow rules match to the packet?
  - Priority can be applied to flow rules (16 bits)
  - Or more detailed flow rule will be applied
  - Basically vender dependent

Some Other Basic Terminologies

- Datapath ID (DPID)
  - A unique ID for OpenFlow Switch
- Secure Channel
  - Connection between Controller and Switch using TCP SSL/TLS
- Statistics (per flow rule)
  - Input Bytes, Input Packets
  - Age of a flow rule

SDN Development Tools

- OpenFlow Switch
  - Vendor Switches
  - Open vSwitch
  - ofsoftswitch13
  - Lagopas
  - (Emulations) Mininet
- P4

- OpenFlow Controller
  - Trema
  - Floodlight
  - OpenDaylight
  - Pox
  - Ryu
- Integrations
  - ONOS: Open Network Operating System
  - CORD: Central Office Re-architected as a Datacenter

What does it mean to implement an application using SDN? (using Ryu)

- Ability to **listen to asynchronous events** (e.g., **PACKET_IN, FLOW_REMOVED**) and to **observe events** using **ryu.controller.handler.set_ev_cls** decorator
- Ability to **parse incoming packets** (e.g., ARP, ICMP, TCP) and **fabricate packets to send out into the network**
- Ability to **create and send an OpenFlow/SDN message** (e.g., **PACKET_OUT, FLOW_MOD, STATS_REQUEST**) to the programmable dataplane.

What did happen to Controller?

```
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
                          packet in 1 00:00:00:00:00:01 ff:ff:ff:ff:ff:ff 1
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
                          packet in 1 00:00:00:00:00:02 00:00:00:00:00:01 2
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
                          packet in 1 00:00:00:00:00:01 00:00:00:00:00:02 1
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
                          packet in 1 00:00:00:00:00:01 ff:ff:ff:ff:ff:ff 1
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
                          packet in 1 00:00:00:00:00:03 00:00:00:00:00:01 3
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
                          packet in 1 00:00:00:00:00:01 00:00:00:00:00:03 1
EVENT ofp_event->SimpleSwitch13 EventOFPPacketIn
                          packet in 1 00:00:00:00:00:02 ff:ff:ff:ff:ff:ff 2
```

Let’s take a look of Flow Table

```
root@mininet-vm:~# ovs-ofctl -O OpenFlow13 dump-flows s1
OFPST_FLOW reply (OF1.3) (xid=0x2):
cookie=0x0, duration=331.815s, table=0, n_packets=3, n_bytes=238,
    priority=1,in_port=3,dl_dst=00:00:00:00:00:02 actions=output:2
cookie=0x0, duration=331.834s, table=0, n_packets=373014,  n_bytes=24618948,
    priority=1,in_port=2,dl_dst=00:00:00:00:00:01   actions=output:1
cookie=0x0, duration=331.825s, table=0, n_packets=3, n_bytes=238,
    priority=1,in_port=3,dl_dst=00:00:00:00:00:01 actions=output:1
cookie=0x0, duration=331.814s, table=0, n_packets=2, n_bytes=140,
    priority=1,in_port=2,dl_dst=00:00:00:00:00:03 actions=output:3
cookie=0x0, duration=331.824s, table=0, n_packets=2, n_bytes=140,
    priority=1,in_port=1,dl_dst=00:00:00:00:00:03 actions=output:3
cookie=0x0, duration=331.831s, table=0, n_packets=373036,  n_bytes=16322899304,
    priority=1,in_port=1,dl_dst=00:00:00:00:00:02   actions=output:2
```

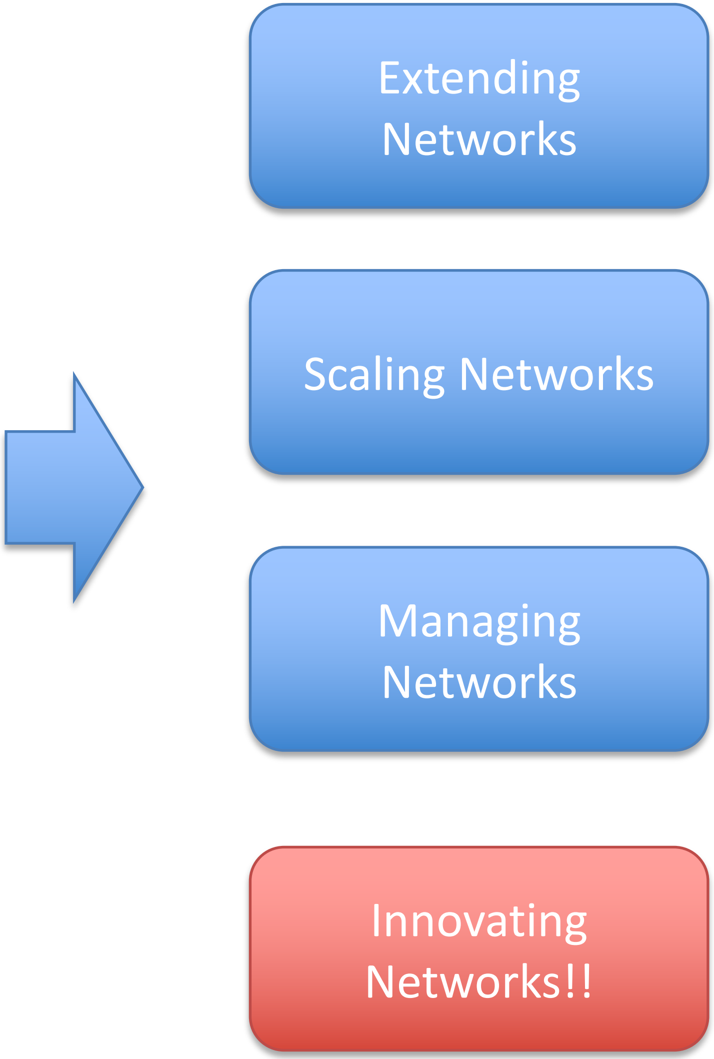
Some points

What does SDN allow us to do?

- It’s up to us about what kind of process do we write in the source code.
- The information used for decision making does not have to come from the network. Users, servers, application-oriented information, etc. Is it cheat?
- Be careful about Overhead and TCAM
  - How many message exchange may happen before installing flow rules on the path?
  - TCAM (Ternary Content-Addressable Memory) is expensive and energy eater. How you may save the consumption of TCAM?

Does SDN turn “Real” or “Myth”?

- Network Monitoring
- Firewall
- Load Balancer
- New Routing Protocol
- VLAN Management
- Networking without VLAN
- Performance Optimization
- Networking for Cloud
- Etc.....



Extending Networks

Scaling Networks

Managing Networks

Innovating Networks!!

Some more things

- What does expect SDN to do something
  - Softwarization of Networks
  - Cloud Computing
  - Network Function Virtualization
  - 5G Networking
  - Internet of Things
  - Data Center Networking

- Let’s explore SDN before we say “yes”, “no” or “xxx” about it.

Known Issues of SDN

- Latency (overhead) of switch-controller communication
- Distributed SDN controller?
- Quality Control of Controller Software
- TCAM (Ternary Content Access Memory) is very expensive and consumes energy (i.e. generates heats)
- Optimizing what?
- Even if we don’t, it comes!!