

EL6363 Lab 2

Mininet Tutorial

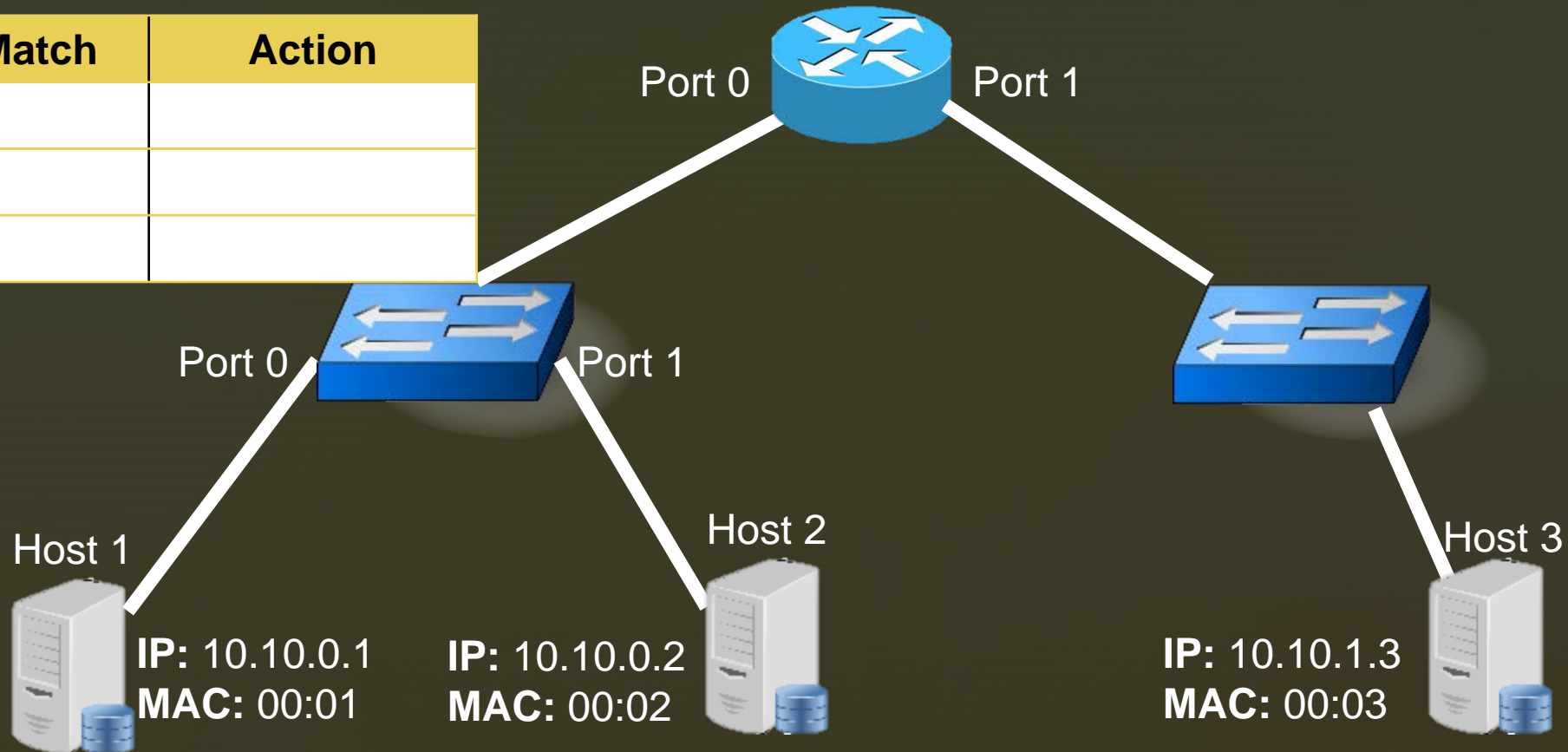


Lab2 Objectives

- Introduction to **Software Defined Network**
- Master the simulation tool: **Mininet**
- Understand the operation of **Openflow** and observe its messages
- Interfacing with **Open Virtual Switch (OVS)**
- Implement a simple flow-based routing algorithm

Switch & Networking

MAC Match	Action



Switch & Networking

MAC Match	Action

Port 0

Port 1

Port 0

Port 1

Host 1

IP: 10.10.0.1
MAC: 00:01

IP: 10.10.0.2
MAC: 00:02

Host 2

IP: 10.10.1.3
MAC: 00:03

Host 3

From

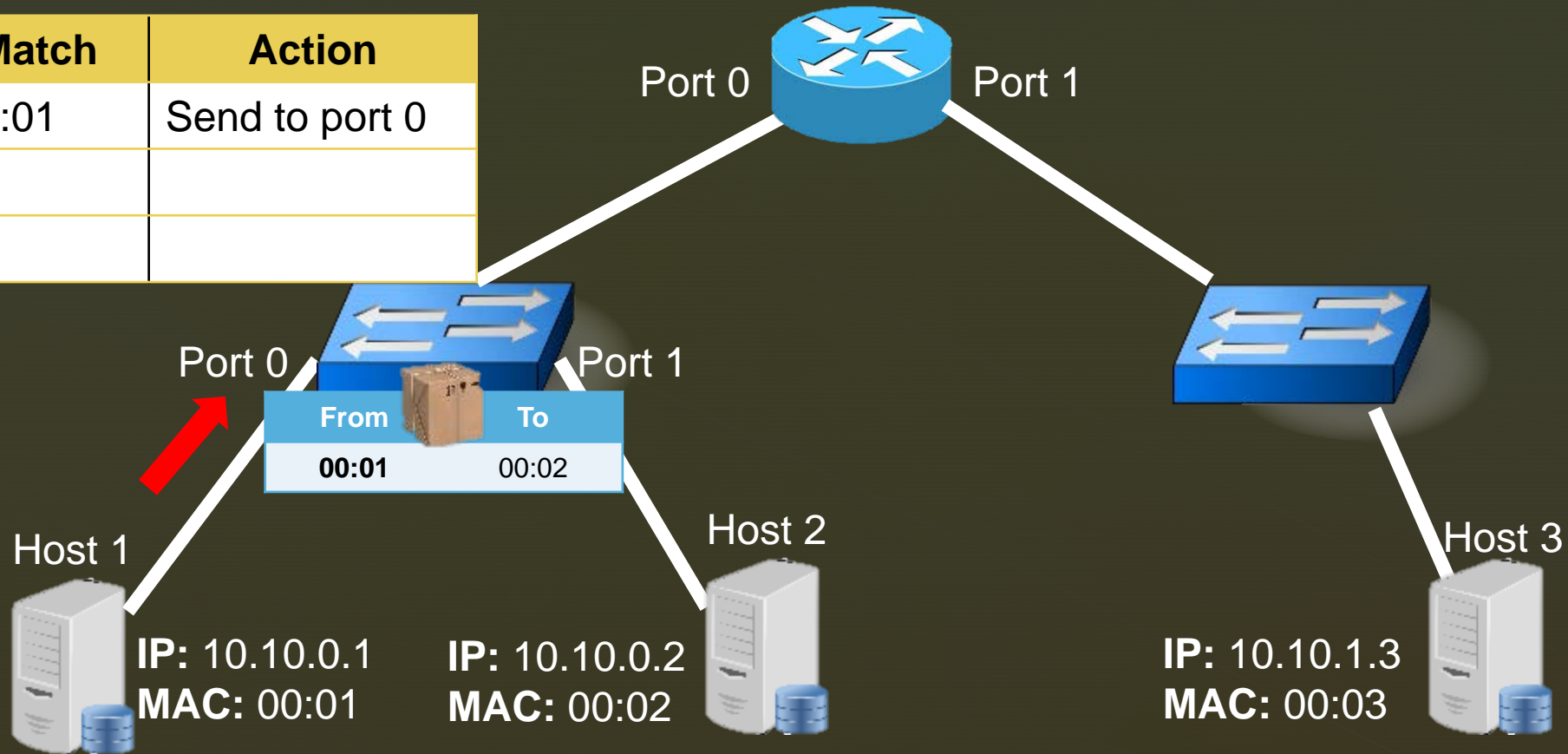
To

00:01

00:02

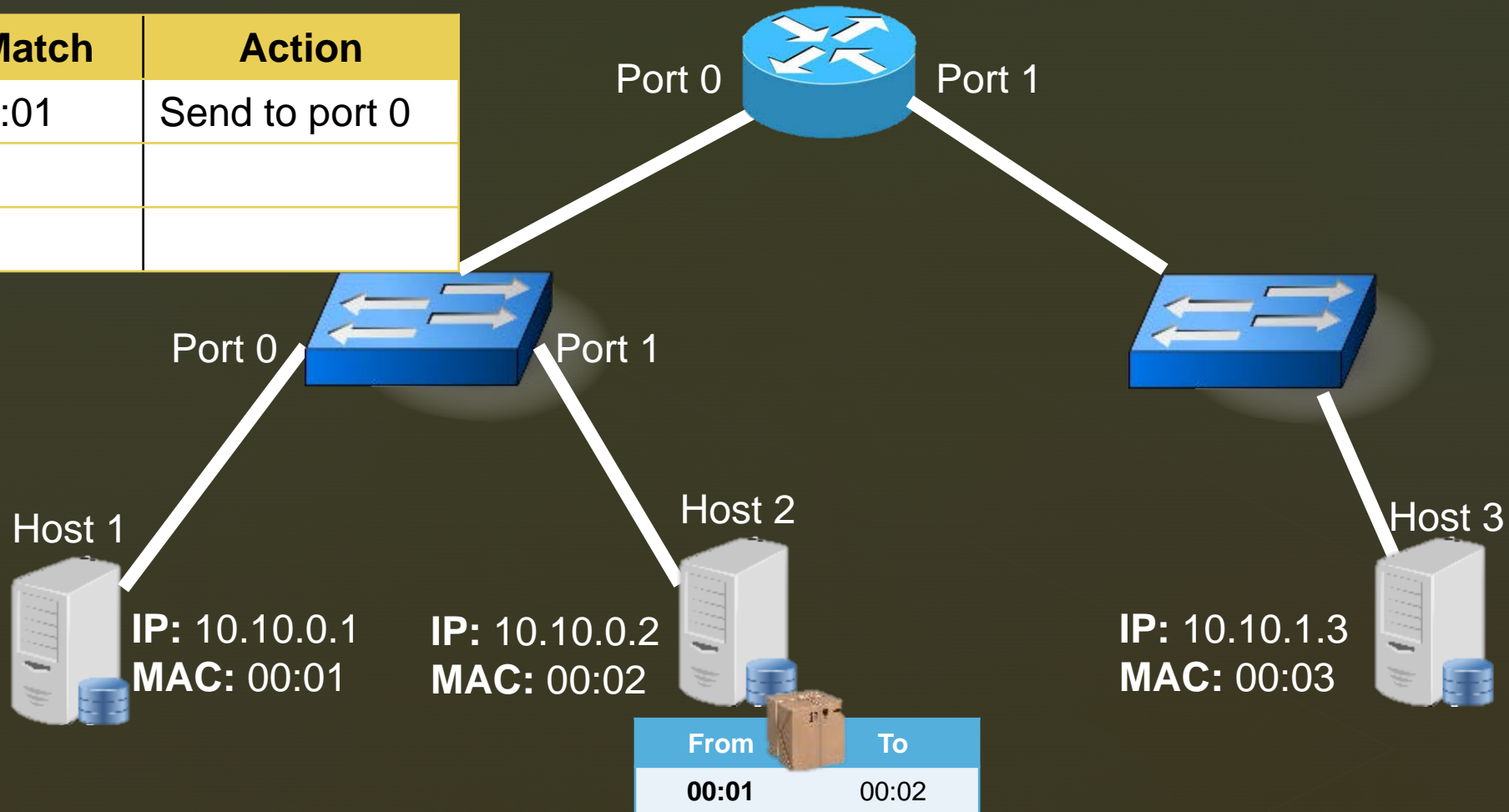
Switch & Networking

MAC Match	Action
MAC: 00:01	Send to port 0



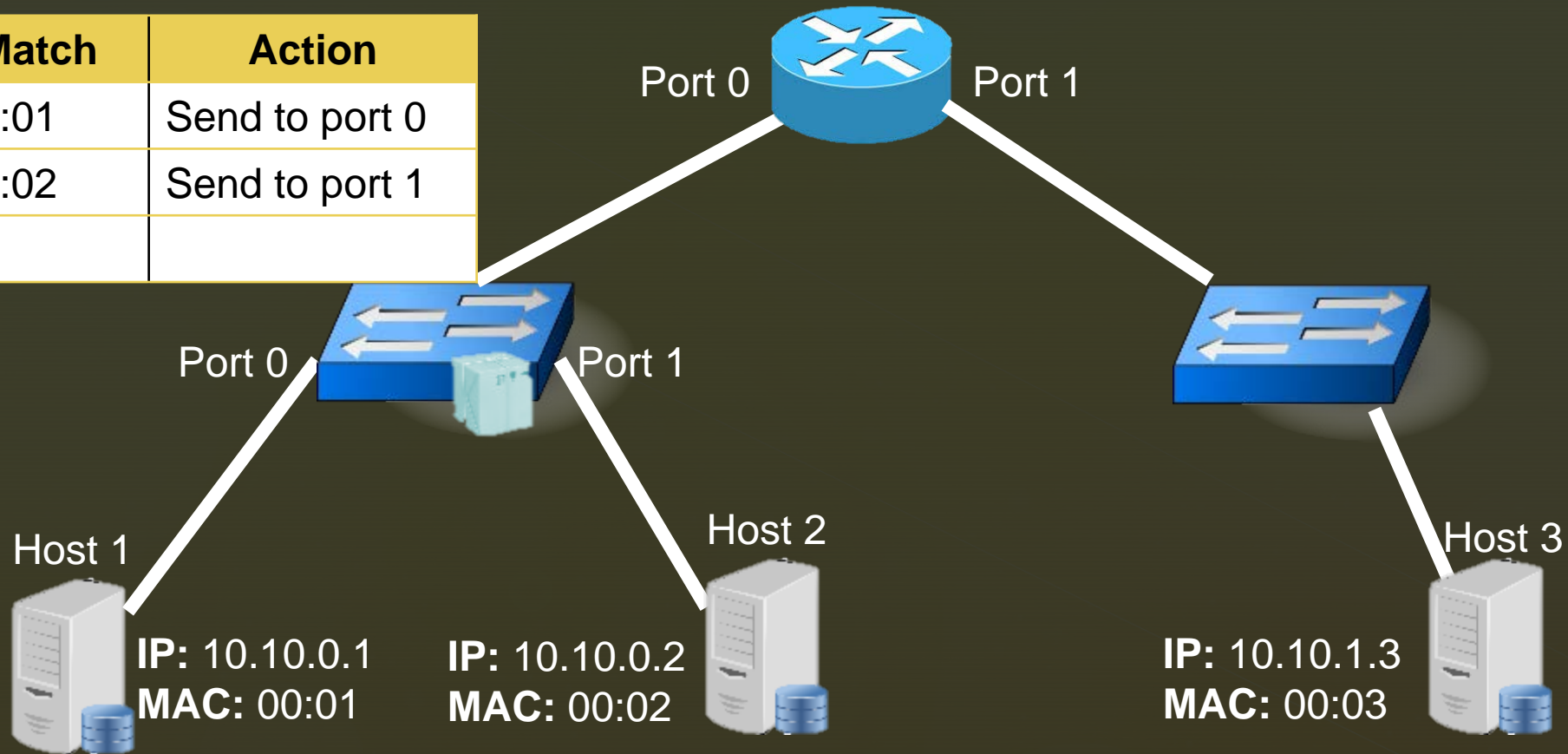
Switch & Networking

MAC Match	Action
MAC: 00:01	Send to port 0



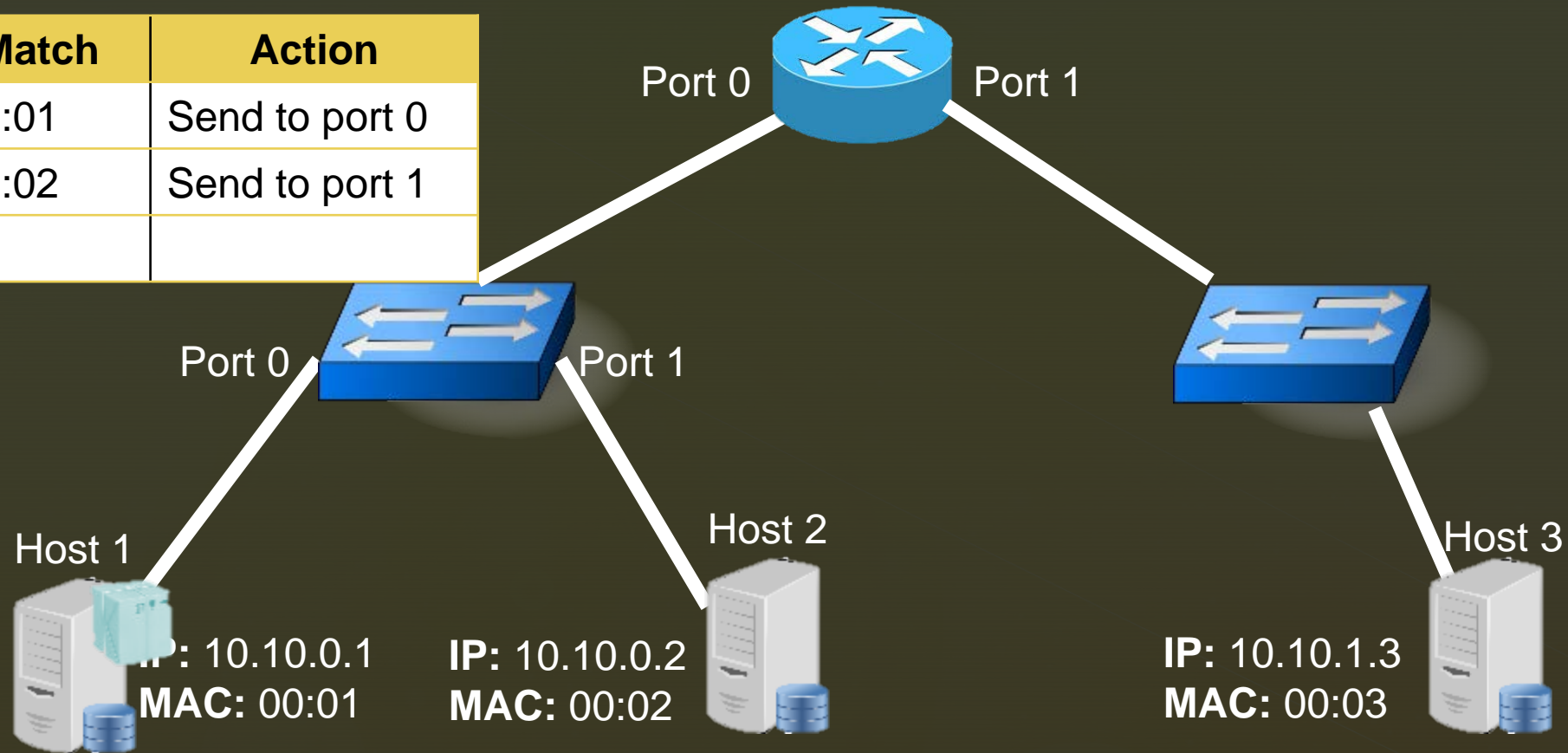
Switch & Networking

MAC Match	Action
MAC: 00:01	Send to port 0
MAC: 00:02	Send to port 1



Switch & Networking

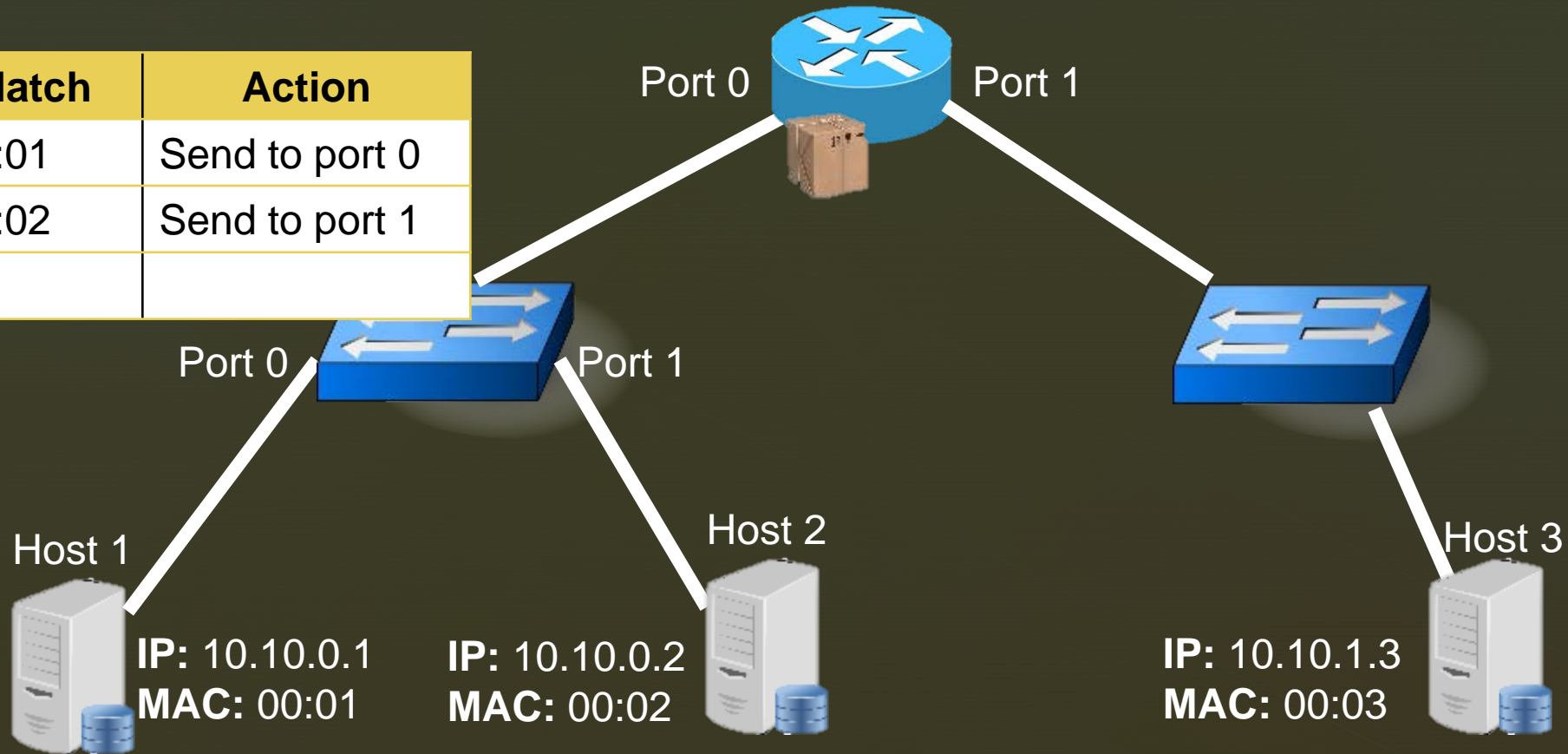
MAC Match	Action
MAC: 00:01	Send to port 0
MAC: 00:02	Send to port 1



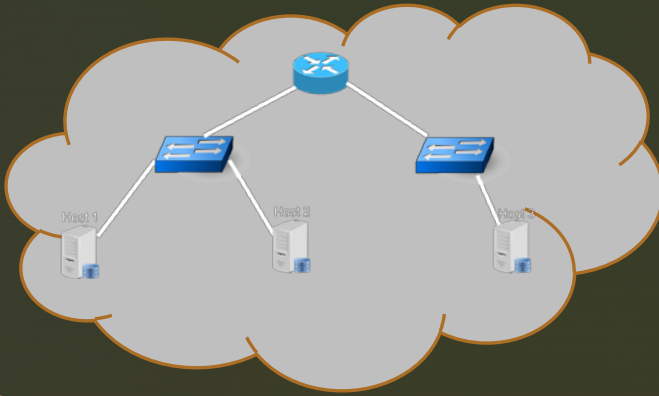
Switch & Networking

IP Match	Action
IP: 10.10.0.0/24	Send to port 0
IP: 10.10.1.0/24	Send to port 1

MAC Match	Action
MAC: 00:01	Send to port 0
MAC: 00:02	Send to port 1



Controller



Software Defined Network

Let the controller tells SDN switches what to do

Match	Action

Port 0

Port 1

Match	Action

Port 1

Host 1



IP: 10.10.0.1
MAC: 00:01

Host 2



IP: 10.10.0.2
MAC: 00:02

Host 3



IP: 10.10.1.3
MAC: 00:03

Controller

Software Defined Network

Match	Action

Switches becomes SDN switches, accepting all kinds of matching fields

Match	Action
IP=	
MAC=	
TCP	

Host 1

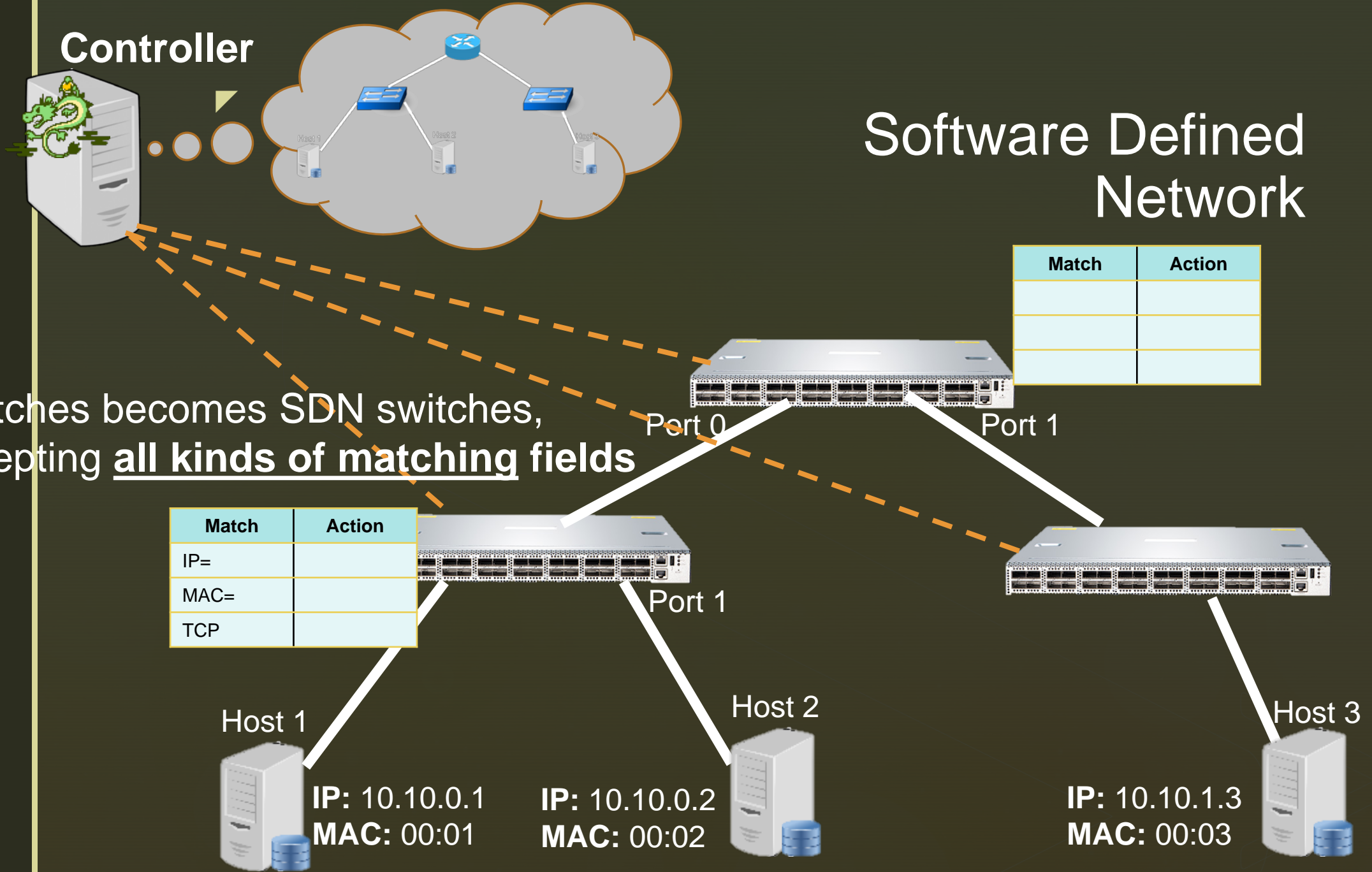
IP: 10.10.0.1
MAC: 00:01

Host 2

IP: 10.10.0.2
MAC: 00:02

Host 3

IP: 10.10.1.3
MAC: 00:03



Controller

Software Defined Network

Control Plane

Data Plane

Match	Action
IP=	
MAC=	
TCP	

Host 1

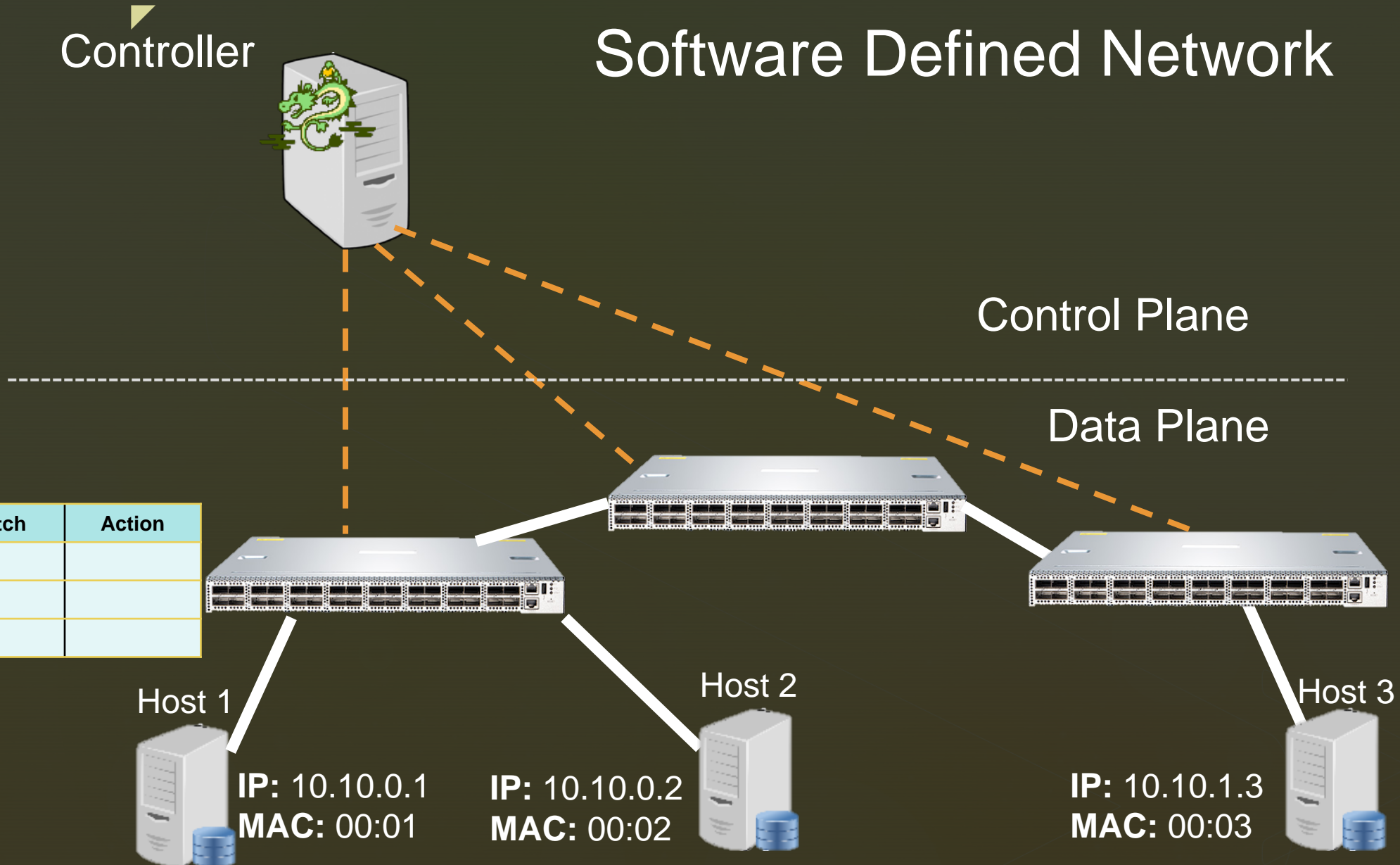
IP: 10.10.0.1
MAC: 00:01

Host 2

IP: 10.10.0.2
MAC: 00:02

Host 3

IP: 10.10.1.3
MAC: 00:03



Controller

Software Defined Network

Control the flows & routes

Control Plane

Data Plane

Forward packets

Match	Action
IP=	
MAC=	
TCP	

Host 1



IP: 10.10.0.1
MAC: 00:01

Host 2

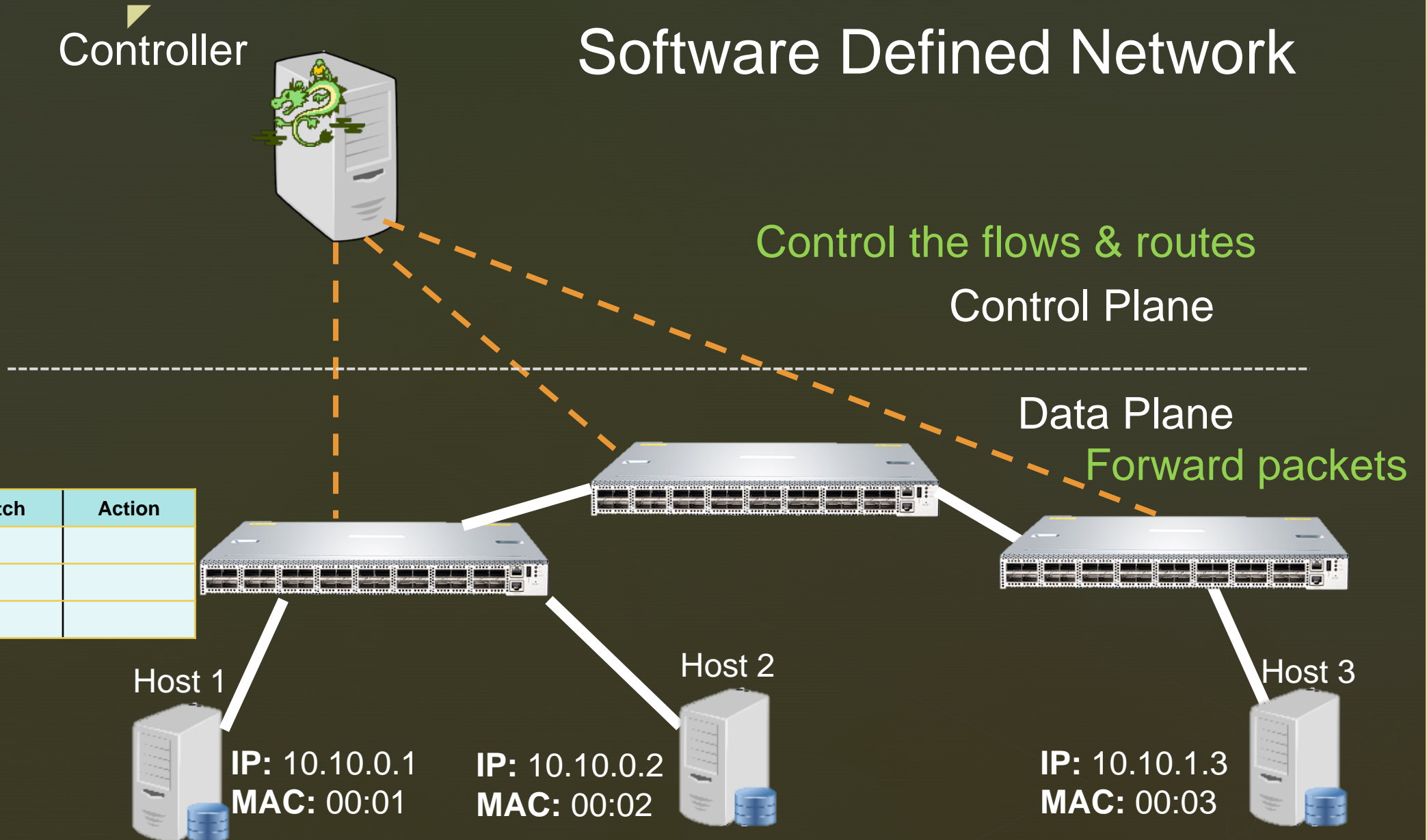


IP: 10.10.0.2
MAC: 00:02

Host 3



IP: 10.10.1.3
MAC: 00:03

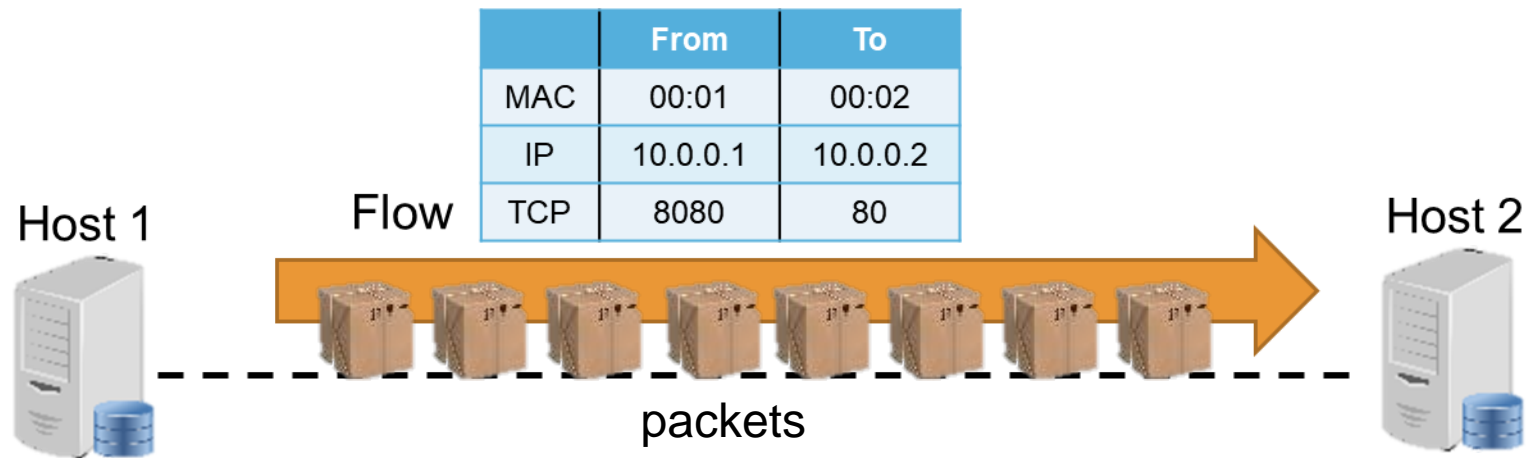


Controller



Software Defined Network

Flow & Packets



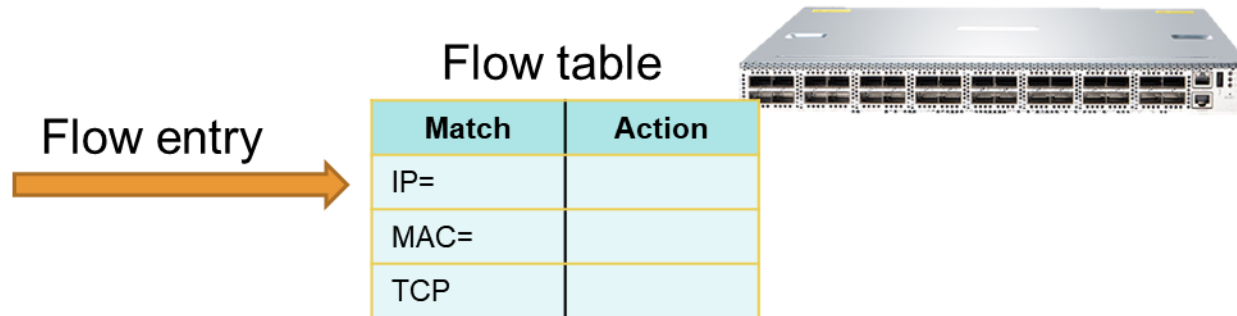
ts

Controller



Software Defined Network

Flow table & Flow entry



Controller

Software Defined Network

Control the flows & routes

Control Plane

Data Plane

Forward packets

Match	Action
IP=	
MAC=	
TCP	

Host 1



IP: 10.10.0.1
MAC: 00:01

Host 2

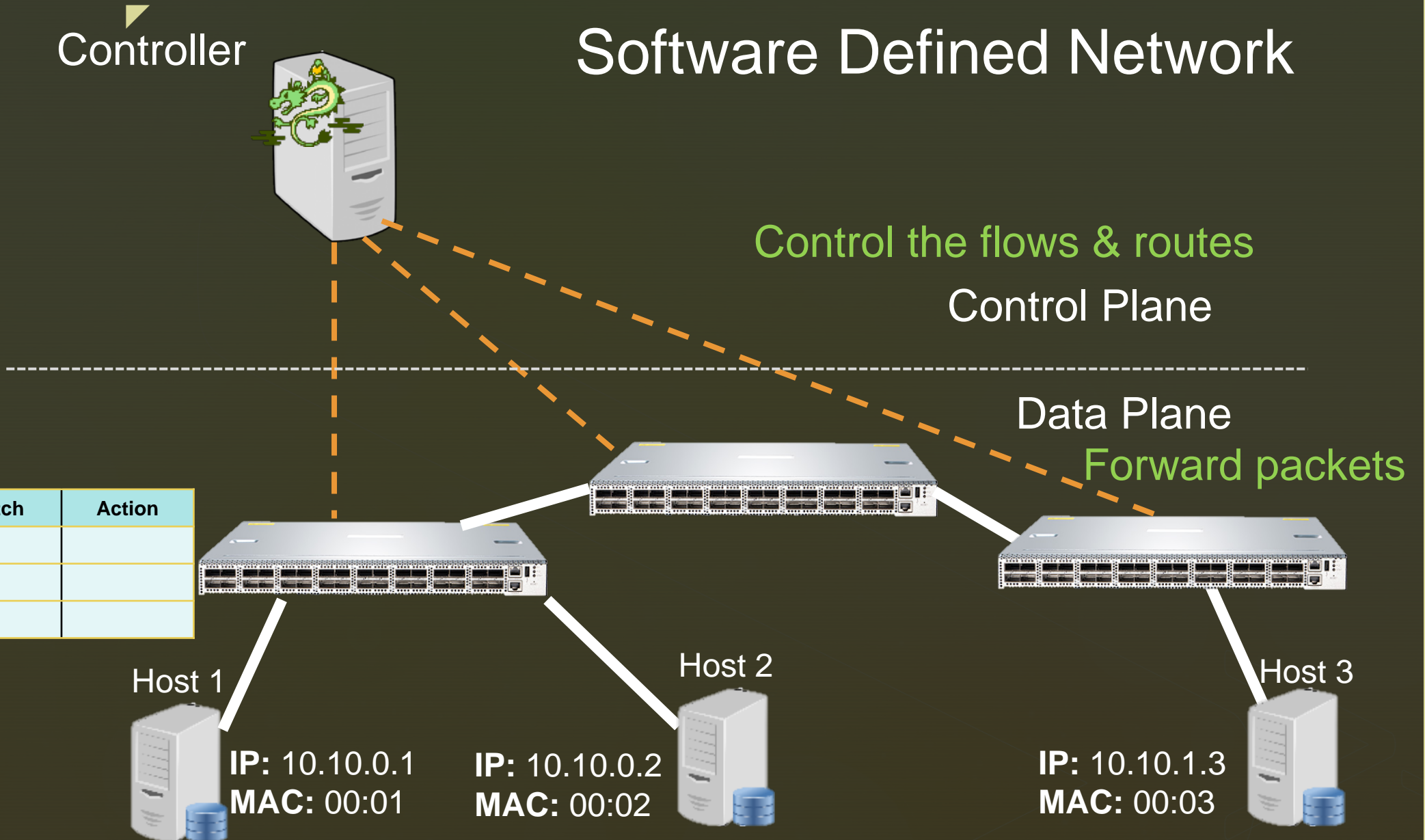


IP: 10.10.0.2
MAC: 00:02

Host 3

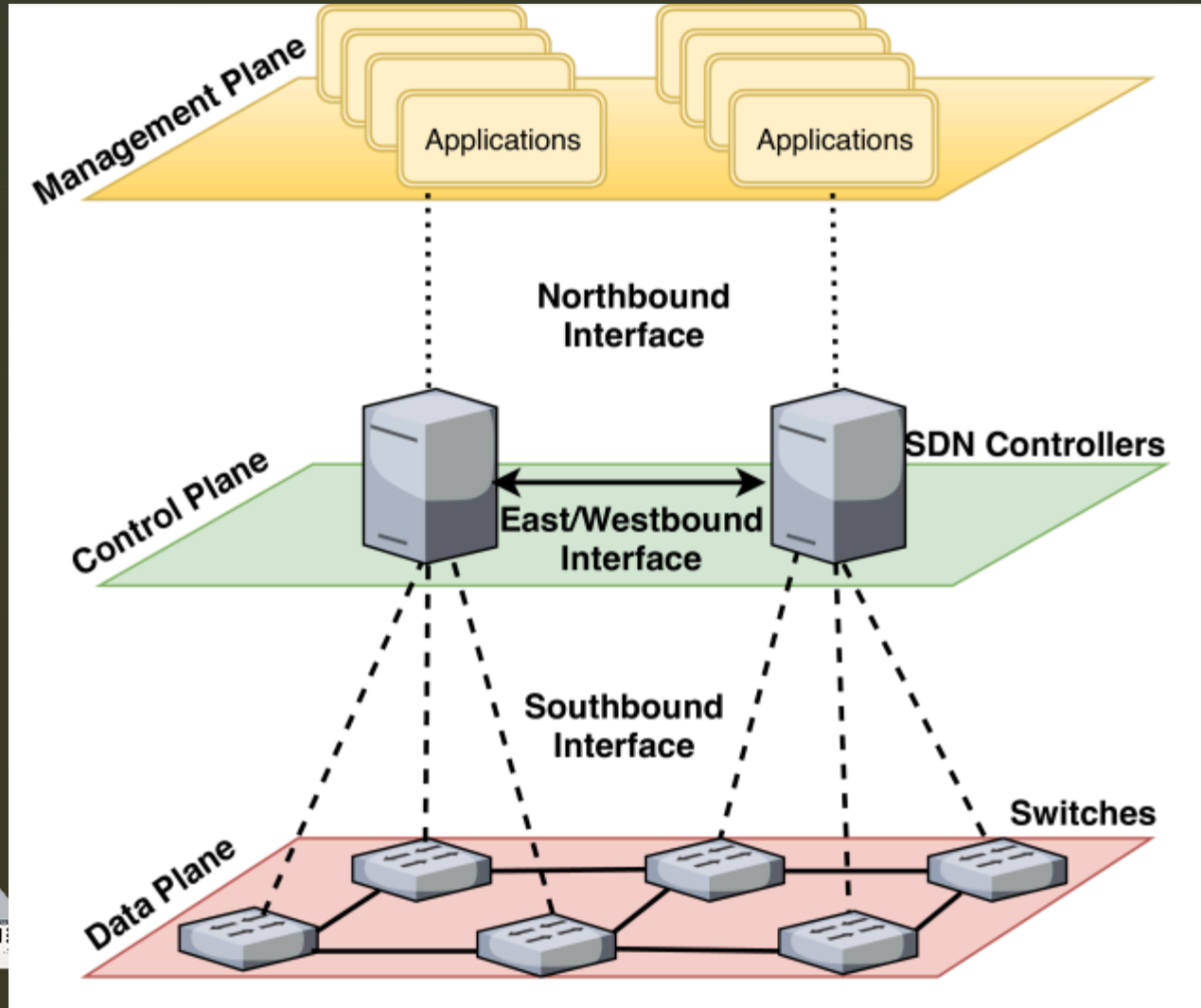


IP: 10.10.1.3
MAC: 00:03

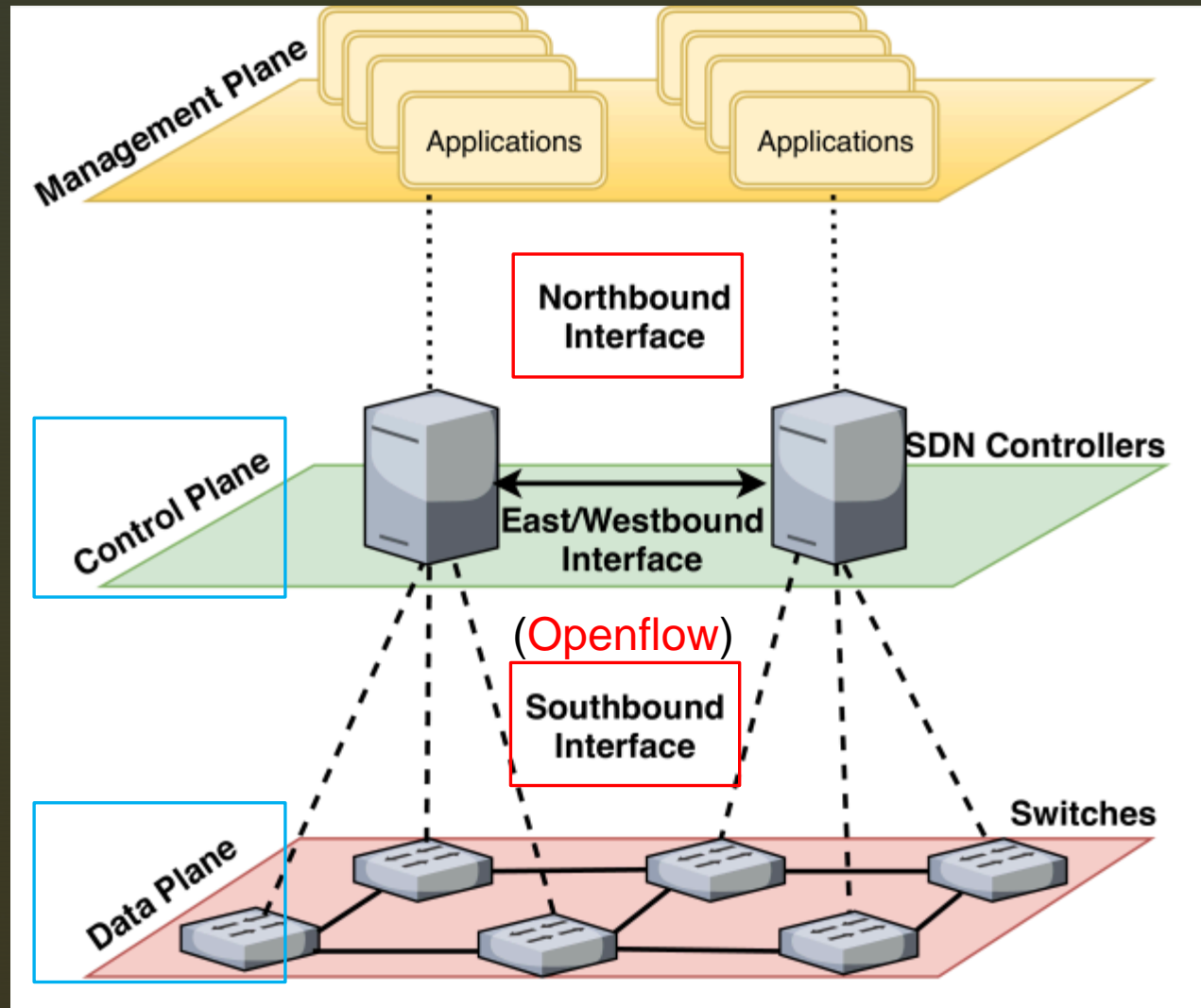


Overview

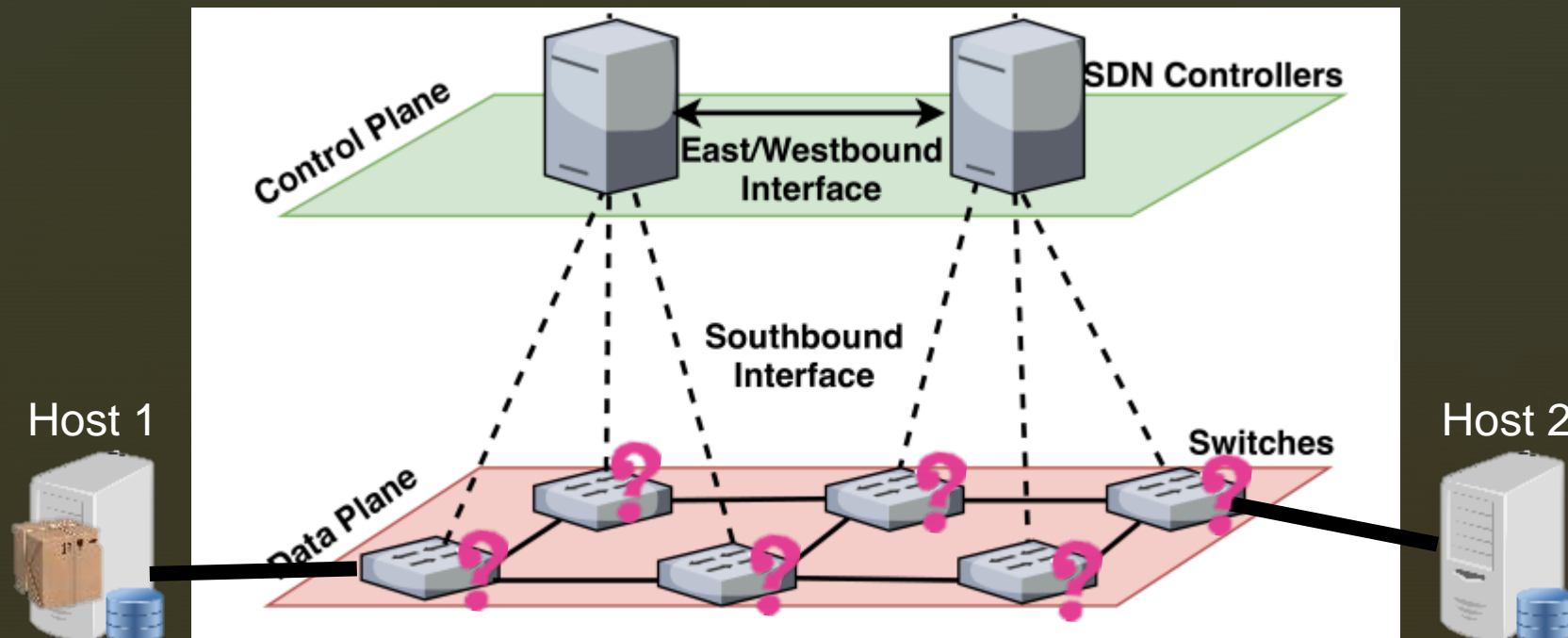
Controller



Overview

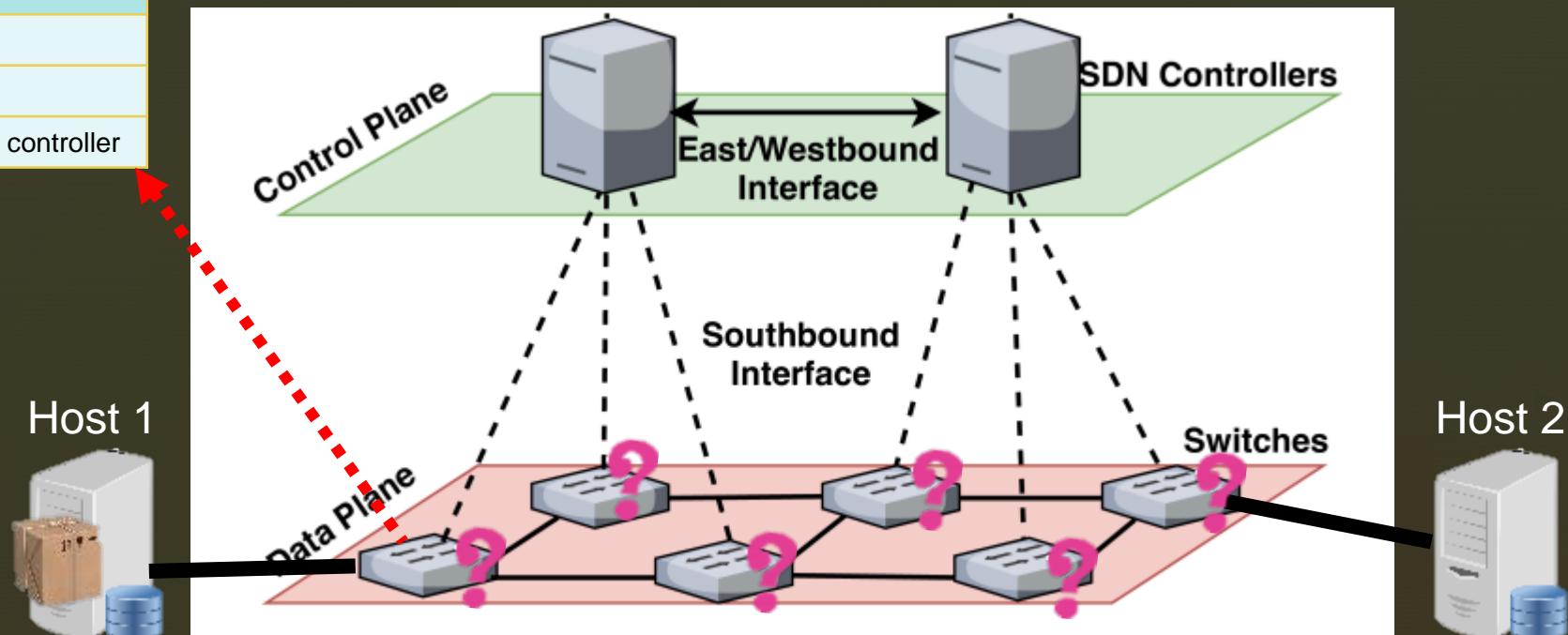


Overview



Overview

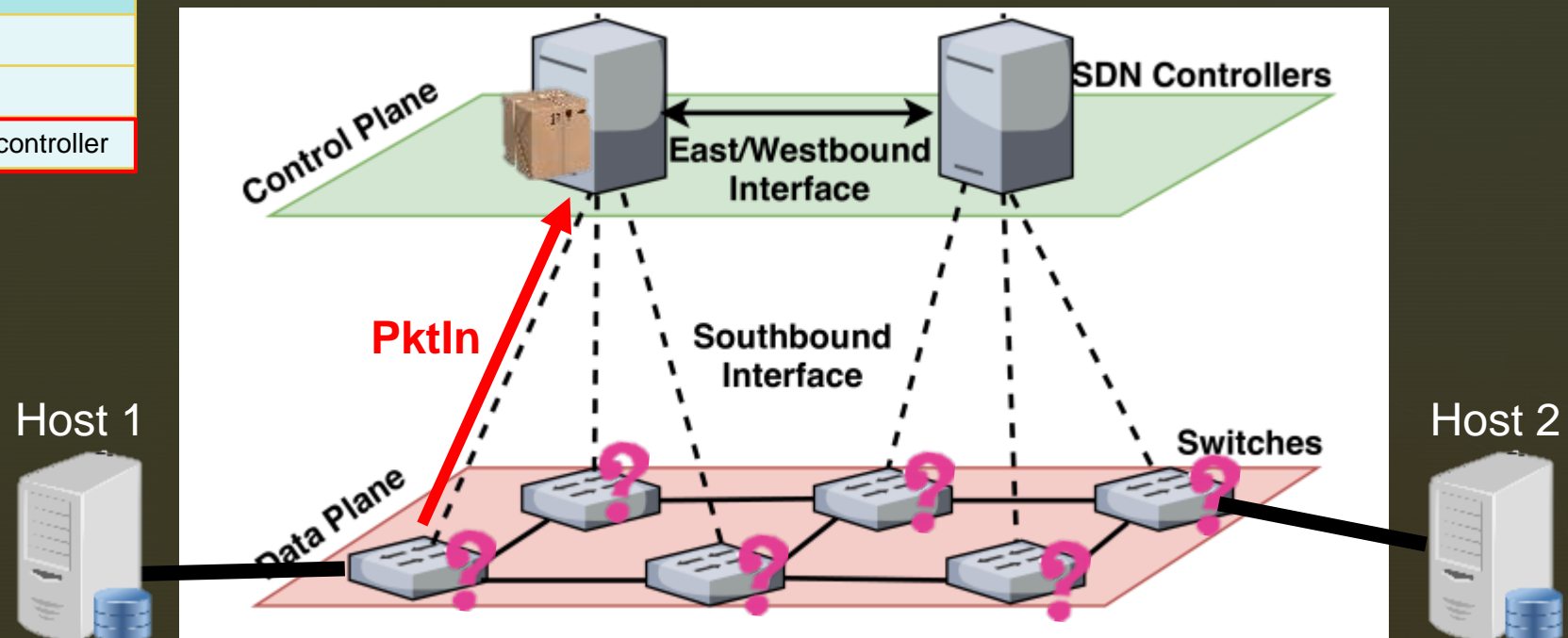
Match	Action
any	To controller



Overview

PktIn: Packet-in

Match	Action
any	To controller



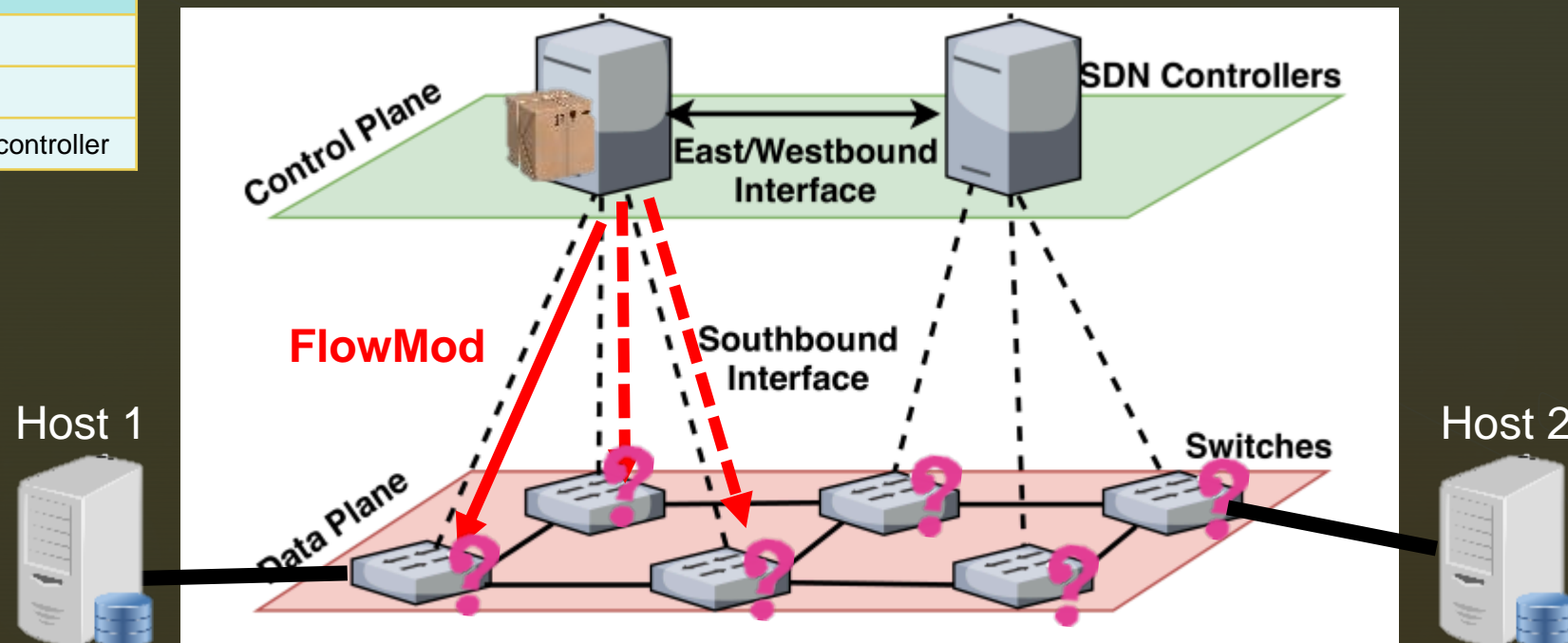
Overview

PktIn: Packet-in

PktOut: Packet-out

FlowMod: flow entry modification

Match	Action
any	To controller



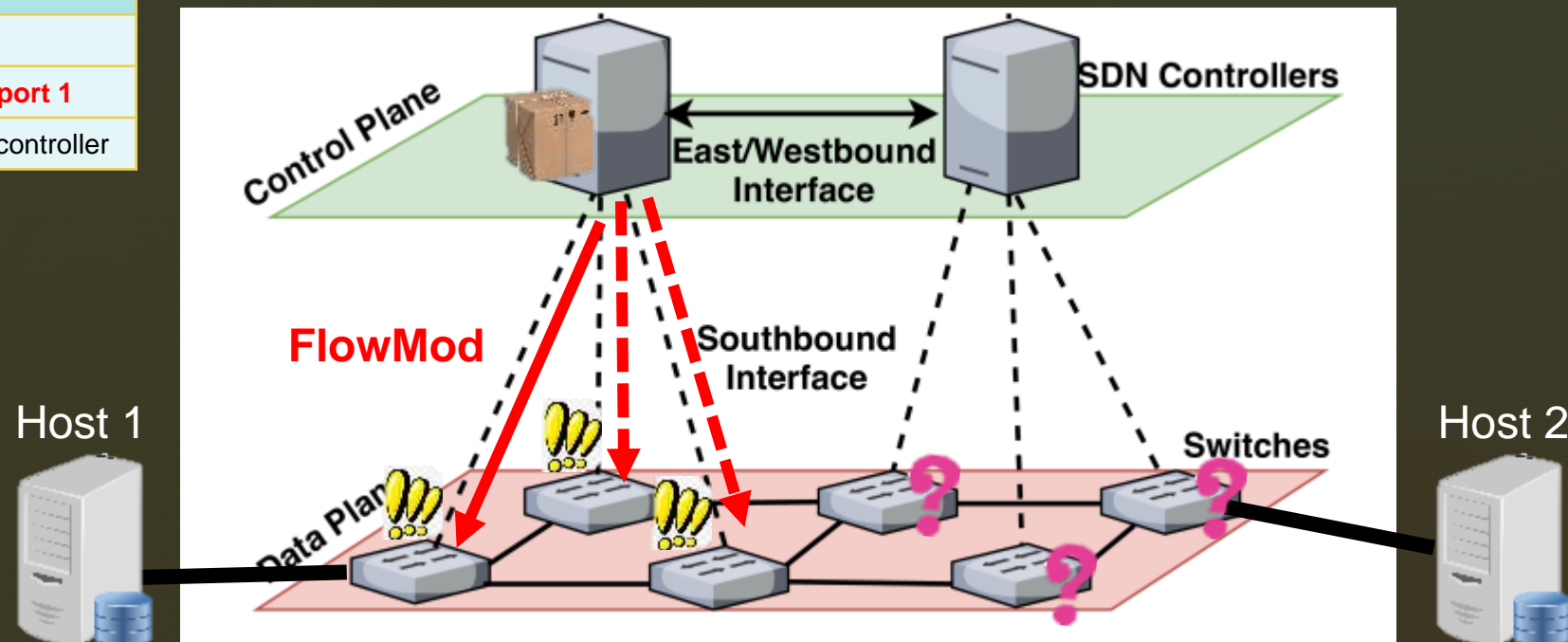
Overview

PktIn: Packet-in

PktOut: Packet-out

FlowMod: flow entry modification

Match	Action
IP= ;TCP=	To port 1
any	To controller

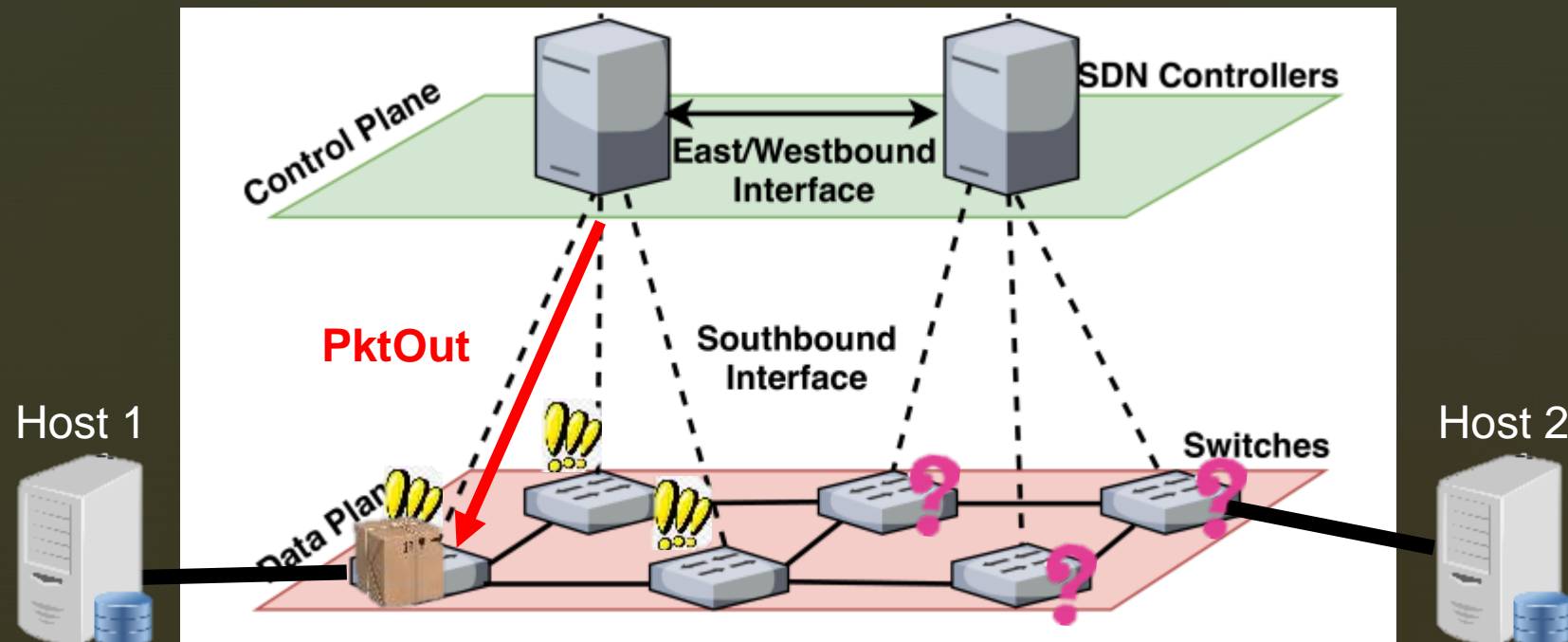


Overview

PktIn: Packet-in

PktOut: Packet-out

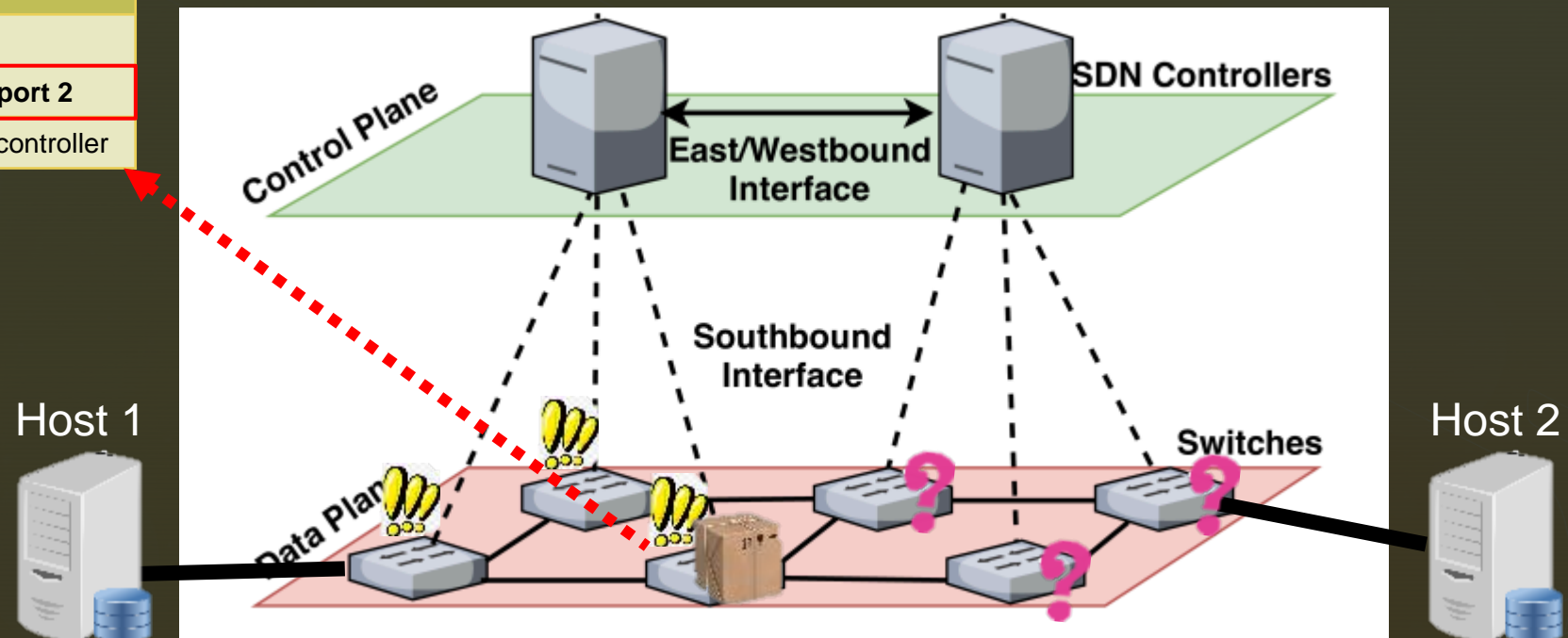
FlowMod: flow entry modification



Overview

PktIn: Packet-in
PktOut: Packet-out
FlowMod: flow entry modification

Match	Action
IP= ;TCP=	To port 2
any	To controller

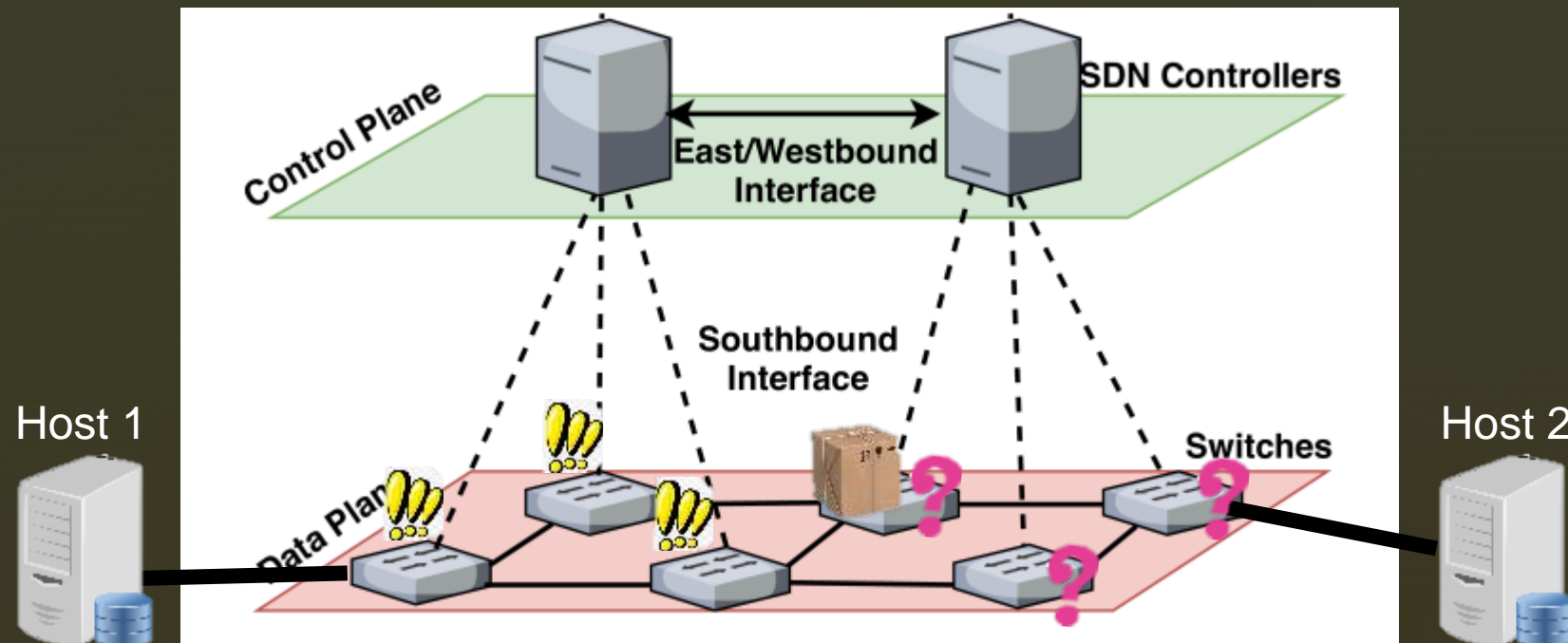


Overview

PktIn: Packet-in

PktOut: Packet-out

FlowMod: flow entry modification

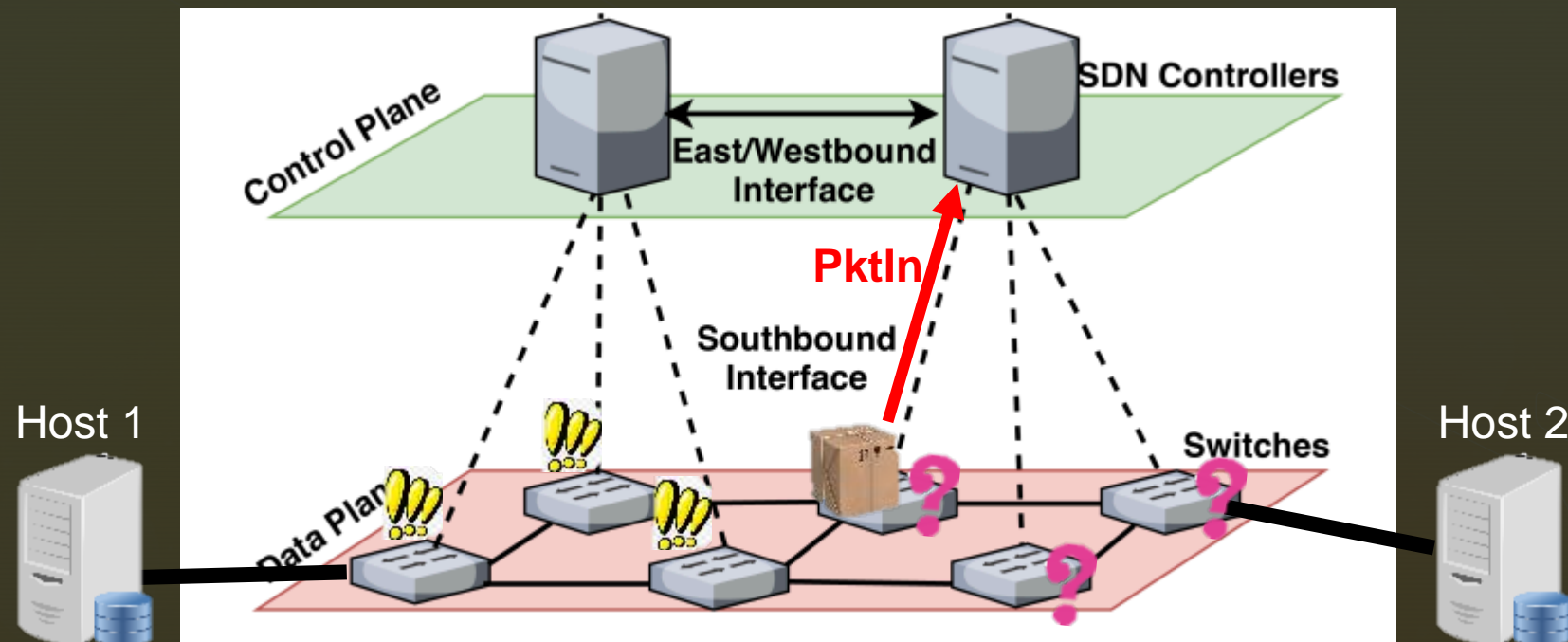


Overview

PktIn: Packet-in

PktOut: Packet-out

FlowMod: flow entry modification

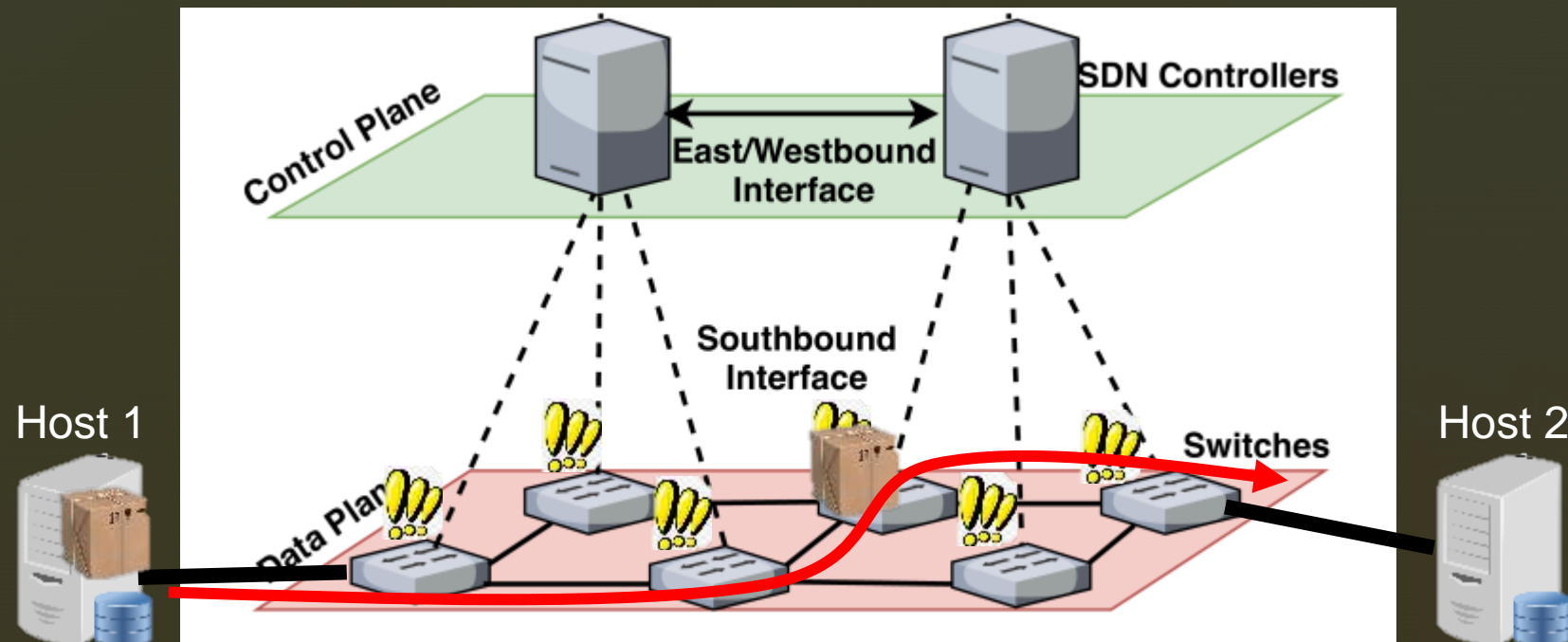


Overview

PktIn: Packet-in

PktOut: Packet-out

FlowMod: flow entry modification

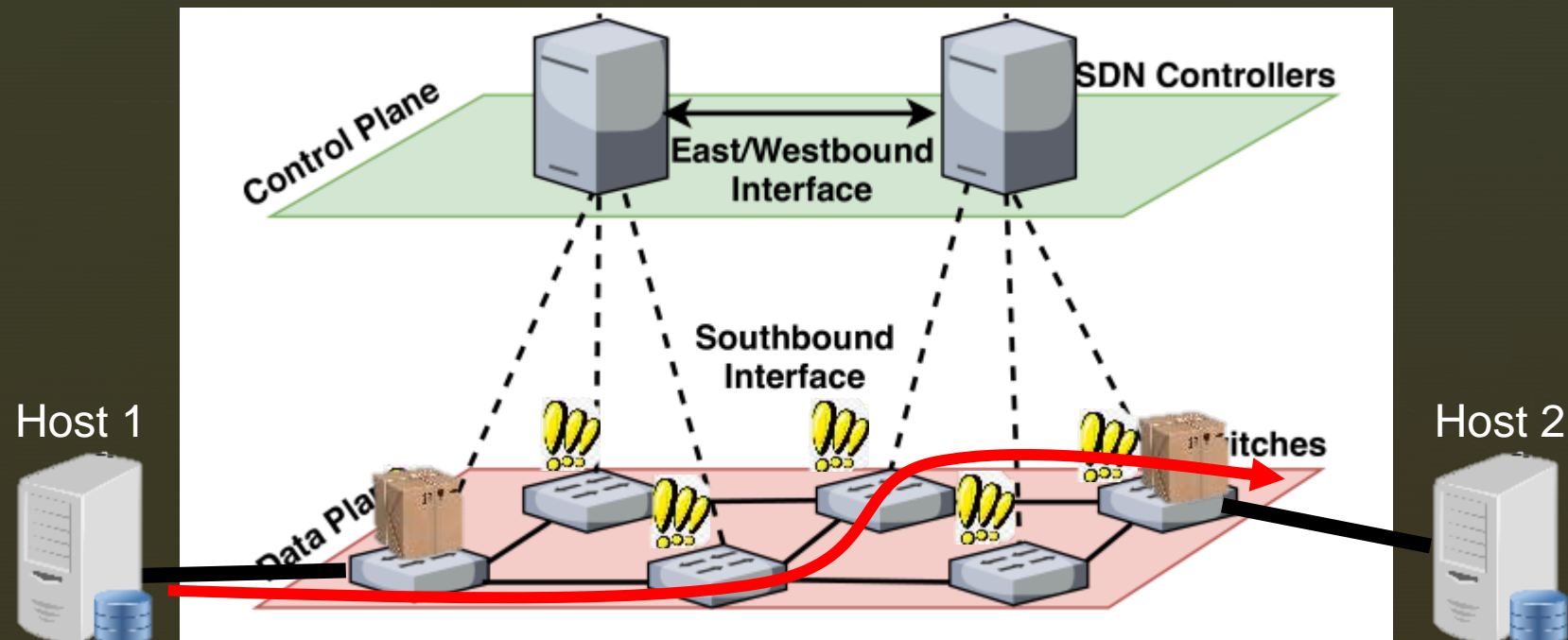


Overview

PktIn: Packet-in

PktOut: Packet-out

FlowMod: flow entry modification

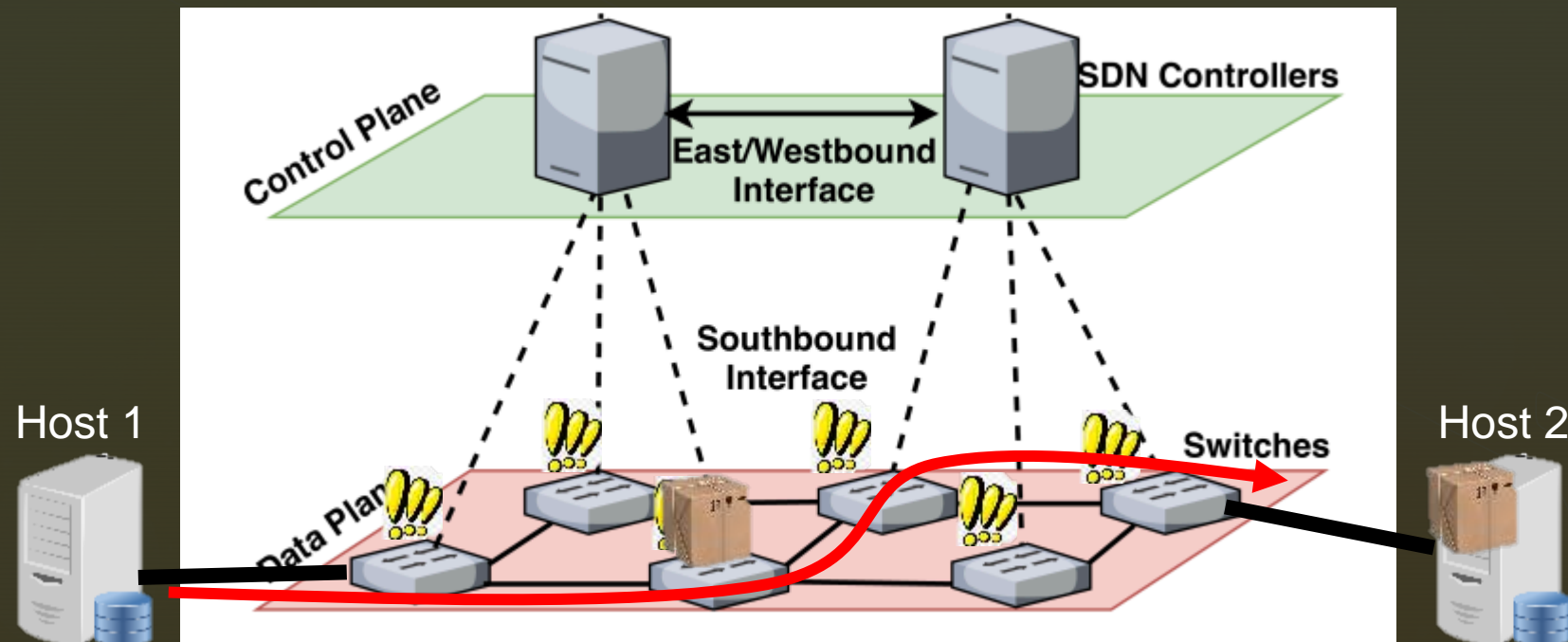


Overview

PktIn: Packet-in

PktOut: Packet-out

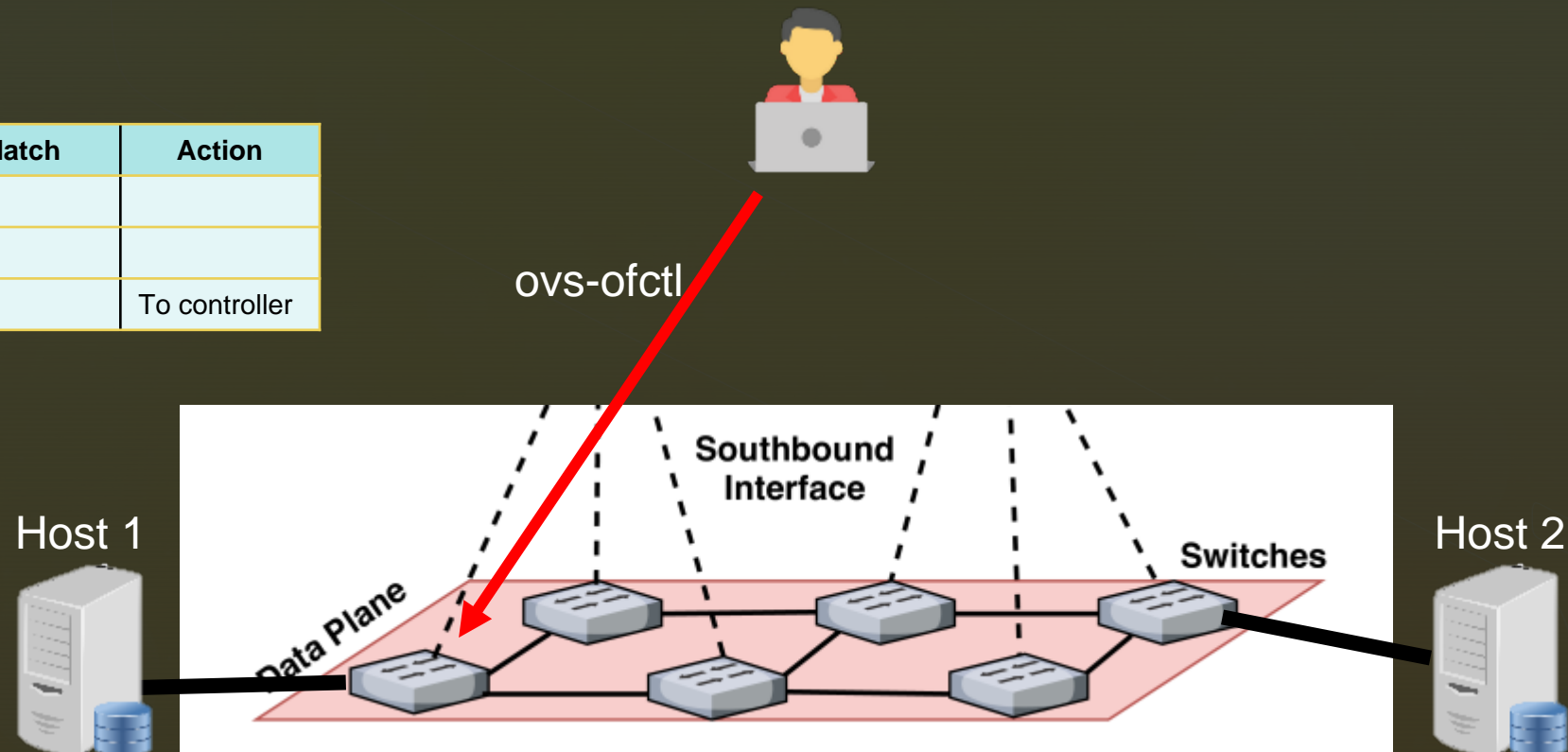
FlowMod: flow entry modification



In Lab 2

We use OpenFlow command to control SDN switches.
(set up flow entries)

Match	Action
any	To controller

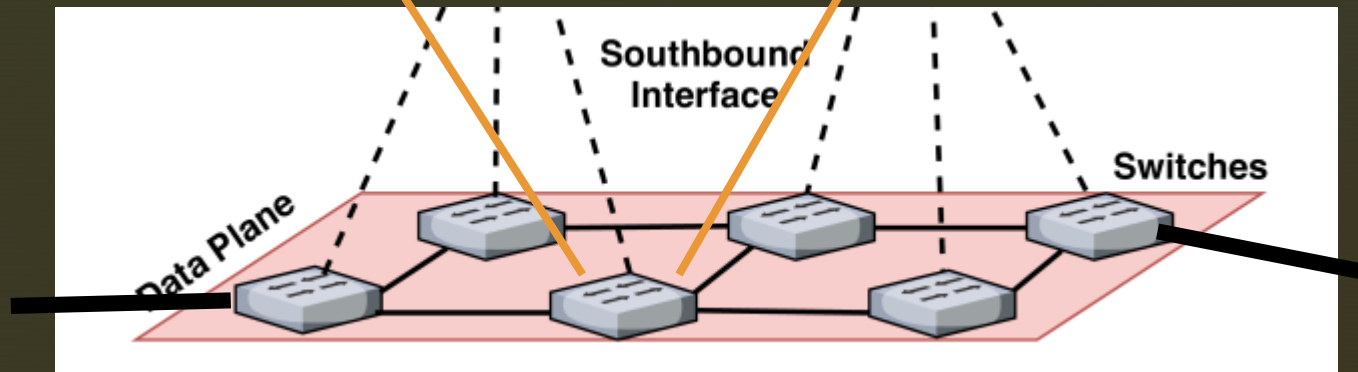


SDN Switch

Software switch



Hardware switch

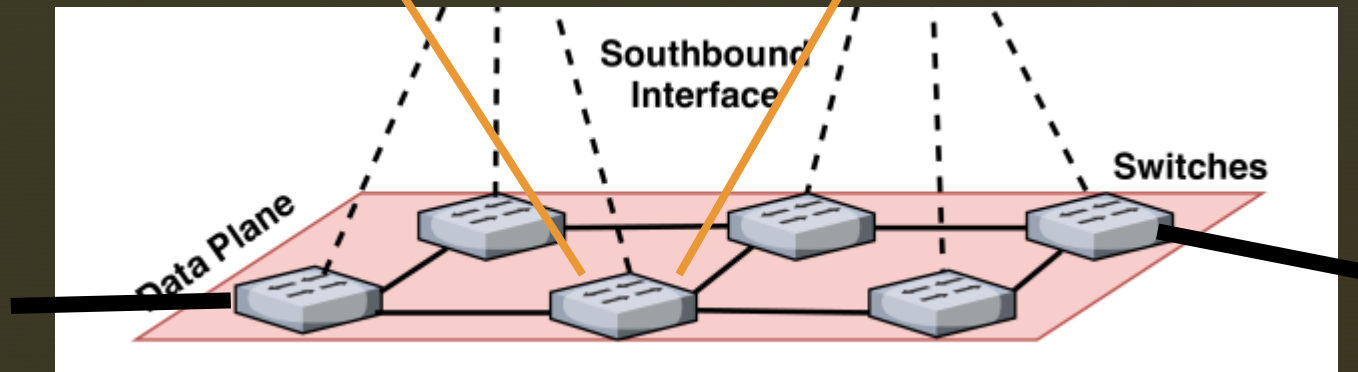


SDN Switch

Software switch



Hardware switch

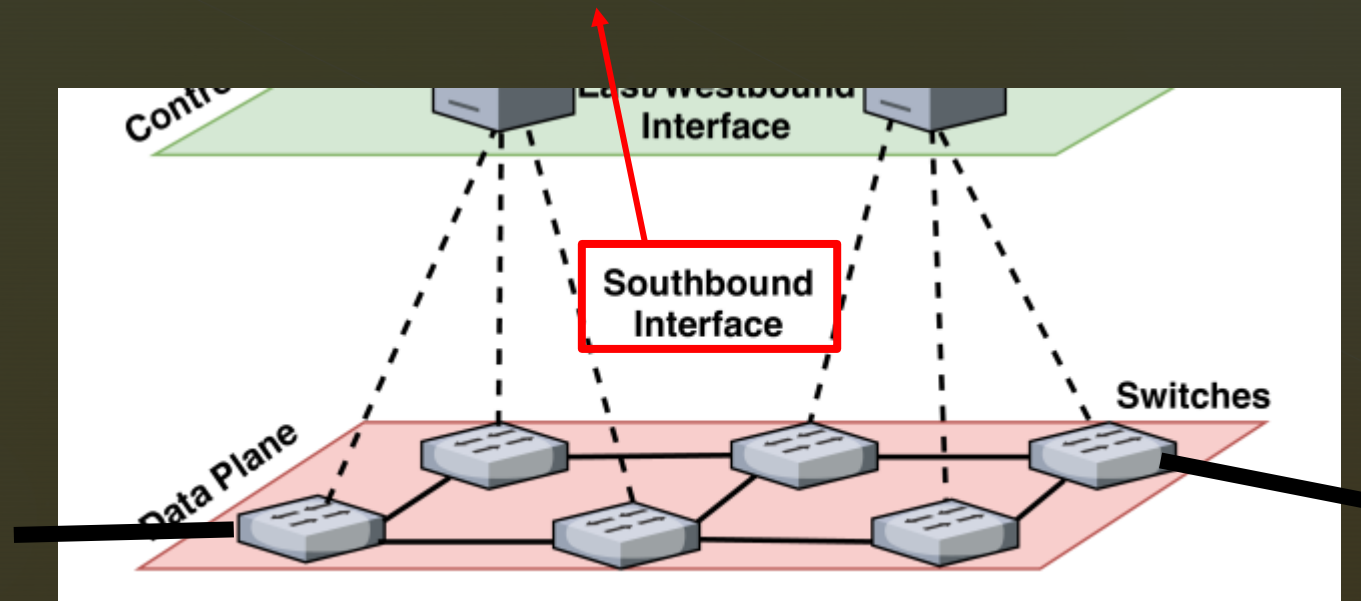


Terms



Open vSwitch: Software switch

OpenFlows: Protocol between control plane & data plane



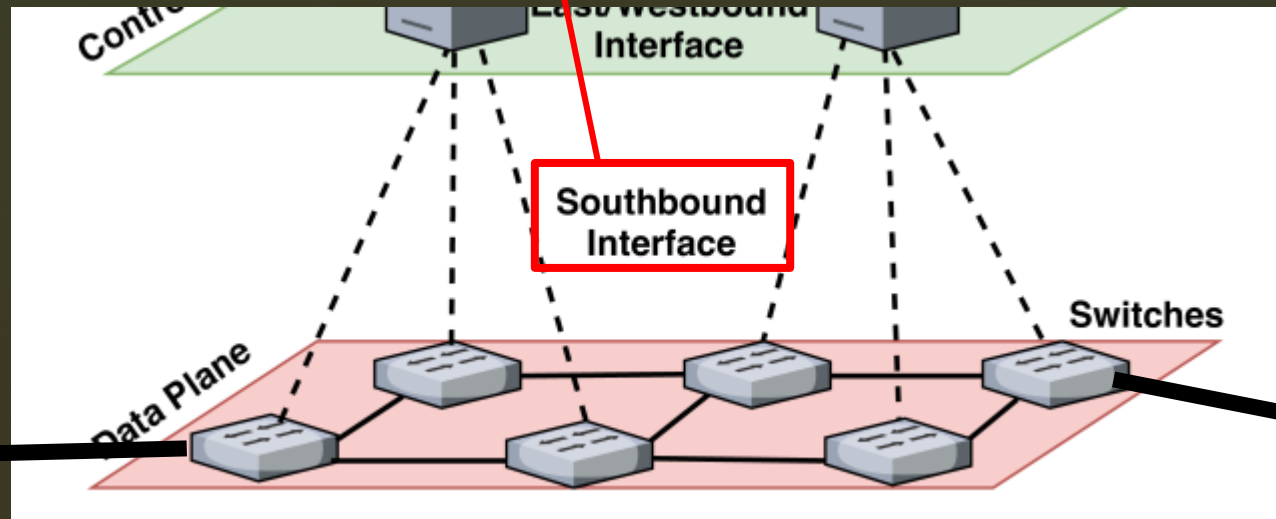
Terms



Open vSwitch: Software switch

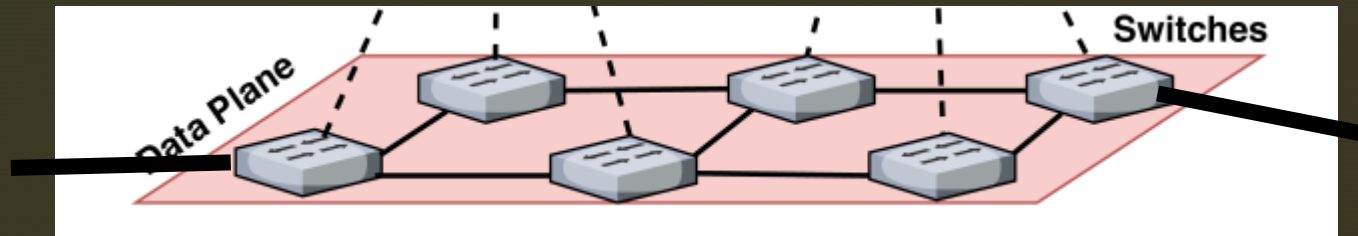
Packet-in
Packet-out
FlowMod

OpenFlows: Protocol between control plane & data plane



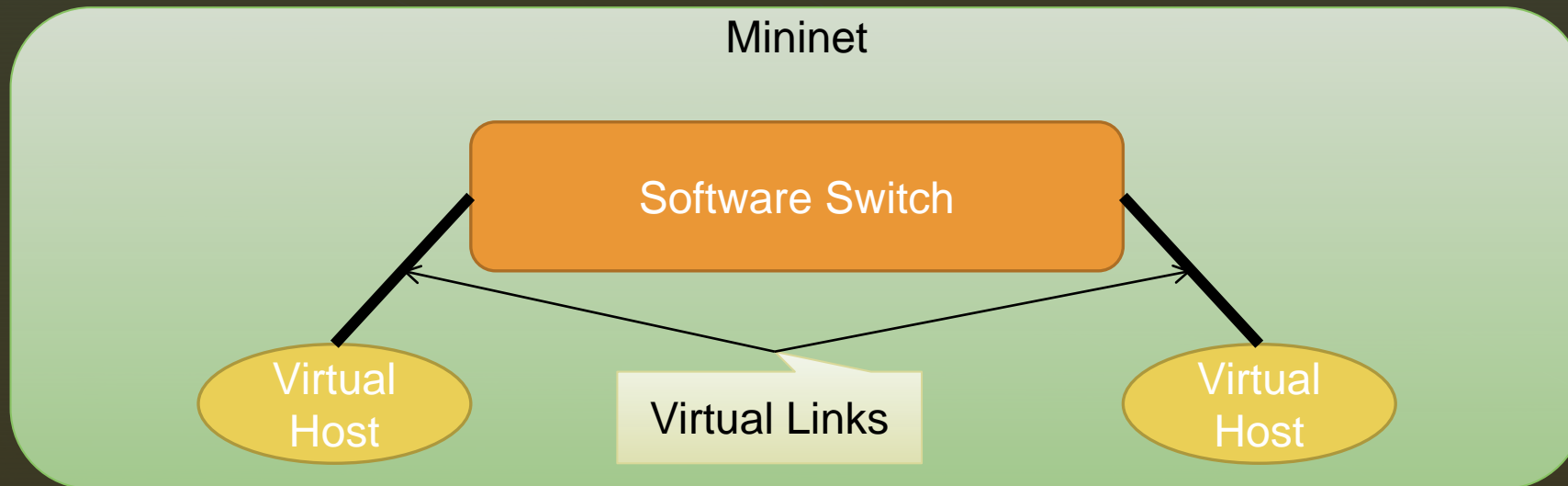
Mininet

- Creates scalable Networks (up to hundreds of nodes)
- Emulator (send real packets within your PC)
- Able to work with any kind of OpenFlow controller
- Easy to program
- Open source project (Free!!)



What does Mininet provide?

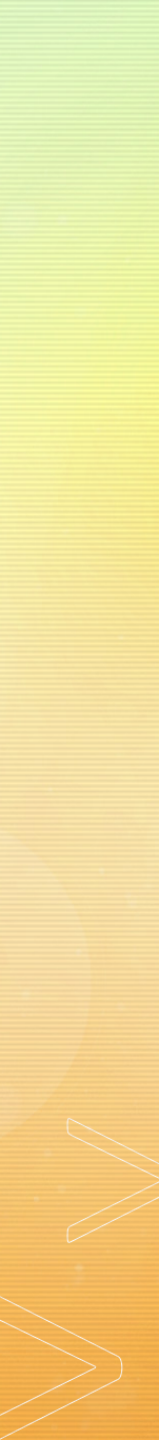
- Mininet emulates software defined networks.
- Create virtual hosts and virtual switches
- Test the connectivity and performance in a virtual environment





Lab2

Tasks & Steps





Lab2 Steps

- Install and run mininet
- Follow mininet walkthrough and learn its commands
- Capture OpenFlow messages passed through mininet hosts
- Create a customized topology
- Install custom forwarding rules on switches

■ Installing mininet (Option 1) - recommended

- Install a Ubuntu VM using either VMWare Player or VirtualBox
- Install mininet through apt-get (remember to update first)
 - `sudo apt-get update`
 - `sudo apt-get install mininet`
- Test installed mininet
 - `sudo mn --test pingall`
- To install default ovs controller
 - `sudo apt-get install openvswitch-testcontroller`
 - `sudo cp /usr/bin/ovs-testcontroller /usr/bin/ovs-controller`

Running mininet



```
frank@frank-VirtualBox: ~  
frank@frank-VirtualBox:~$ sudo mn --test pingall  
[sudo] password for frank:  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Waiting for switches to connect  
s1  
*** Ping: testing ping reachability  
h1 -> h2  
h2 -> h1  
*** Results: 0% dropped (2/2 received)  
*** Stopping 1 controllers  
c0  
*** Stopping 2 links  
..  
*** Stopping 1 switches  
s1  
*** Stopping 2 hosts  
h1 h2  
*** Done  
completed in 3.787 seconds  
frank@frank-VirtualBox:~$
```

- `sudo mn --test pingall`

Mininet Walkthrough

- Follow the official mininet walkthrough
 - <http://mininet.org/walkthrough/>
 - Example:

Display Startup Options

Let's get started with Mininet's startup options.

Type the following command to display a help message describing Mininet's startup options:

```
$ sudo mn -h
```

- Learn
 - How to Run mininet
 - mininet's basic commands

- 1. Observe the OpenFlow Control message

1. Running Mininet with Custom Topology

1. Run the Mininet topology

Open Wireshark in Ubuntu and start!!

- To launch Mininet with **default controller**:
 - `sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo`

No “--controller remote” !



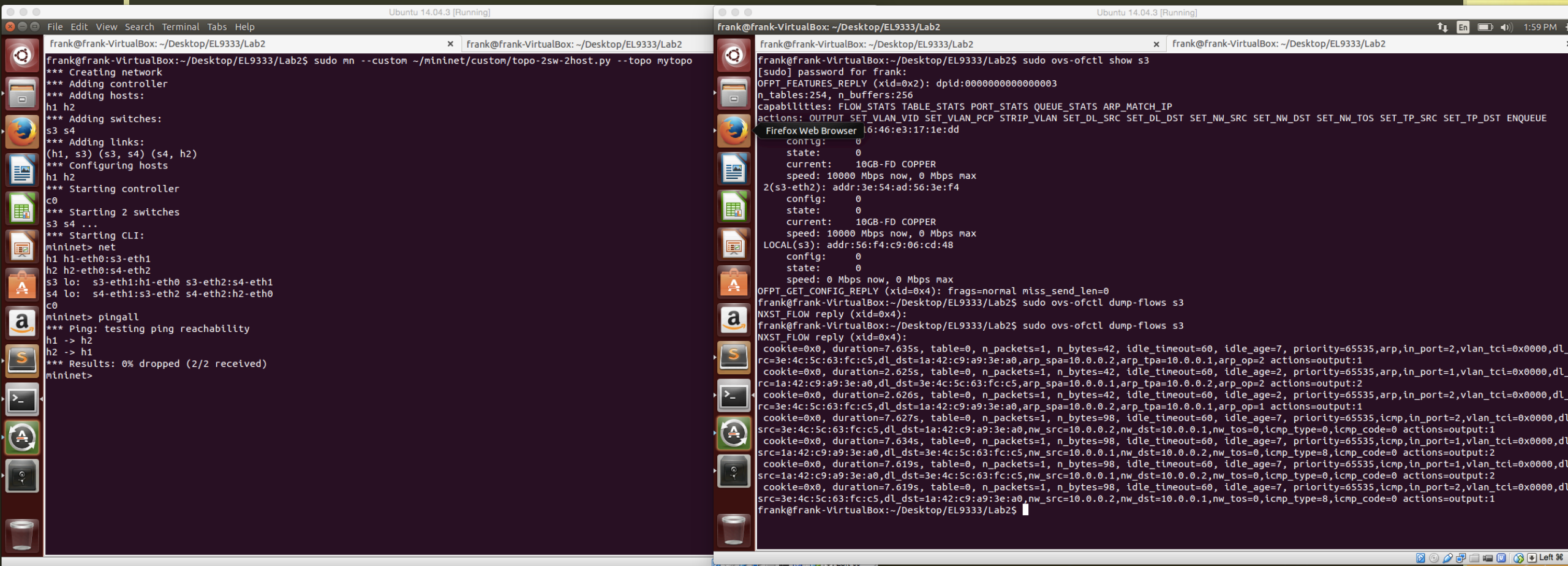
The screenshot shows a terminal window titled 'Ubuntu 14.04.3 [Running]' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help) and a status bar (1:59 PM). The terminal content is as follows:

```
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```


1. Running Mininet with Custom Topology

1. Run the Mininet topology

2. Open another terminal for observation



```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl show s3
[sudo] password for frank:
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
Firefox Web Browser 16:46:e3:17:1e:dd
  config:
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
2(s3-eth2): addr:3e:54:ad:56:3e:f4
  config:
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
LOCAL(s3): addr:56:f4:c9:06:cd:48
  config:
    state: 0
    speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=7.635s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=7, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
  cookie=0x0, duration=2.625s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=output:2
  cookie=0x0, duration=2.626s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
  cookie=0x0, duration=7.627s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
  cookie=0x0, duration=7.634s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:1
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$

```

1. Running Mininet with Custom Topology

1. Run the Mininet topology

2. Open another terminal for observation

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
  
```

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl show s3
[sudo] password for frank:
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
Firefox Web Browser 16:46:e3:17:1e:dd
  config:
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
2(s3-eth2): addr:3e:54:ad:56:3e:f4
  config:
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
LOCAL(s3): addr:56:f4:c9:06:cd:48
  config:
    state: 0
    speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=7.635s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=7, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
  cookie=0x0, duration=2.625s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=output:2
  cookie=0x0, duration=2.626s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
  cookie=0x0, duration=7.627s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
  cookie=0x0, duration=7.634s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:1
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$
  
```

3. Dump flows

before pingall

1. Running Mininet with Custom Topology

1. Run the Mininet topology

2. Open another terminal for observation

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
  
```

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl show s3
[sudo] password for frank:
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
Firefox Web Browser 16:46:e3:17:1e:dd
  contlg: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
2(s3-eth2): addr:3e:54:ad:56:3e:f4
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(s3): addr:56:f4:c9:06:cd:48
  config: 0
  state: 0
  speed: 0 Mbps now, 0 Mbps max
  OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=7.635s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=7, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
  cookie=0x0, duration=2.625s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=output:2
  cookie=0x0, duration=2.626s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
  cookie=0x0, duration=7.627s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
  cookie=0x0, duration=7.634s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:1
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$
  
```

3. Dump flows

before pingall

4. pingall

1. Running Mininet with Custom Topology

1. Run the Mininet topology

2. Open another terminal for observation

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
  
```

```

frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl show s3
[sudo] password for frank:
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
Firefox Web Browser 16:46:e3:17:1e:dd
  config:
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
2(s3-eth2): addr:3e:54:ad:56:3e:f4
  config:
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
LOCAL(s3): addr:56:f4:c9:06:cd:48
  config:
    state: 0
    speed: 0 Mbps now, 0 Mbps max
  OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=7.635s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=7, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl
rc=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
  cookie=0x0, duration=2.625s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl
rc=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=output:2
  cookie=0x0, duration=2.626s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl
rc=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
  cookie=0x0, duration=7.627s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,d
src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
  cookie=0x0, duration=7.634s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,d
src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,d
src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,d
src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:1
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$
  
```

3. Dump flows

5. Dump flows

before pingall

after pingall

4. pingall

Running Mininet with Custom Topology

1. Run the Mininet topology

2. Open another terminal for observation

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
  
```

```

frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl show s3
[sudo] password for frank:
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
Firefox Web Browser 16:46:e3:17:1e:dd
  contlg: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
2(s3-eth2): addr:3e:54:ad:56:3e:f4
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(s3): addr:56:f4:c9:06:cd:48
  config: 0
  state: 0
  speed: 0 Mbps now, 0 Mbps max
  OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=7.635s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=7, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl
rc=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
  cookie=0x0, duration=2.625s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl
rc=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=output:2
  cookie=0x0, duration=2.626s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=2, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl
rc=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
  cookie=0x0, duration=7.627s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,d
src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
  cookie=0x0, duration=7.634s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,d
src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=1,vlan_tci=0x0000,d
src=1a:42:c9:a9:3e:a0,dl_dst=3e:4c:5c:63:fc:c5,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:2
  cookie=0x0, duration=7.619s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=7, priority=65535,icmp,in_port=2,vlan_tci=0x0000,d
src=3e:4c:5c:63:fc:c5,dl_dst=1a:42:c9:a9:3e:a0,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:1
frank@frank-VirtualBox:~/Desktop/EL9333/Lab2$
  
```

3. Dump flows

5. Dump flows

before pingall

after pingall

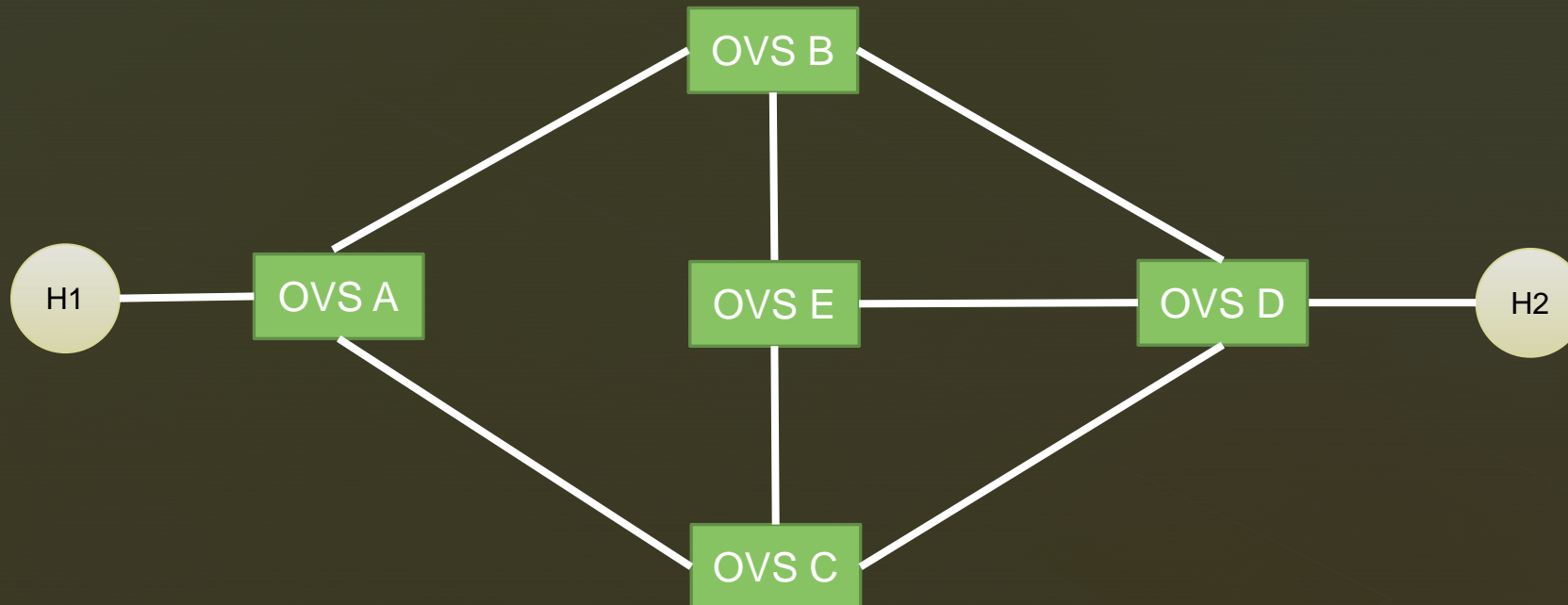
4. pingall

Stop Wireshark and find Openflow control flows!!

- 2. Create custom topology and install rules

2. Create custom topology and install rules

- Simulate a software defined network with 2 hosts and 5 switches



Creating a custom topology

```
# Add hosts and switches
```

```
H1 = self.addHost( 'H1' )
```

```
H2 = self.addHost( 'H2' )
```

```
S1 = self.addSwitch( 'S1' )
```

```
S2 = self.addSwitch( 'S2' )
```

```
# Add links
```

```
self.addLink( H1 , S1 )
```

```
self.addLink( S1, S2 )
```

```
self.addLink( S2, H2 )
```



Refer to: <https://github.com/mininet/mininet/blob/master/custom/topo-2sw-2host.py>

Creating a custom topology

```
# Add hosts and switches
```

```
H1 = self.addHost( 'H1' )
```

```
H2 = self.addHost( 'H2' )
```

```
S1 = self.addSwitch( 'S1' )
```

```
S2 = self.addSwitch( 'S2' )
```

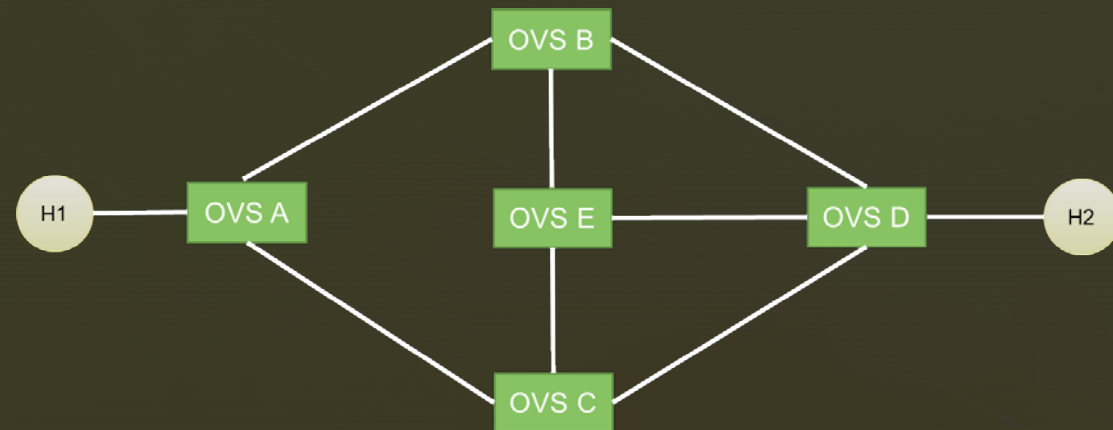
```
# Add links
```

```
self.addLink( H1 , S1 )
```

```
self.addLink( S1, S2 )
```

```
self.addLink( S2, H2 )
```

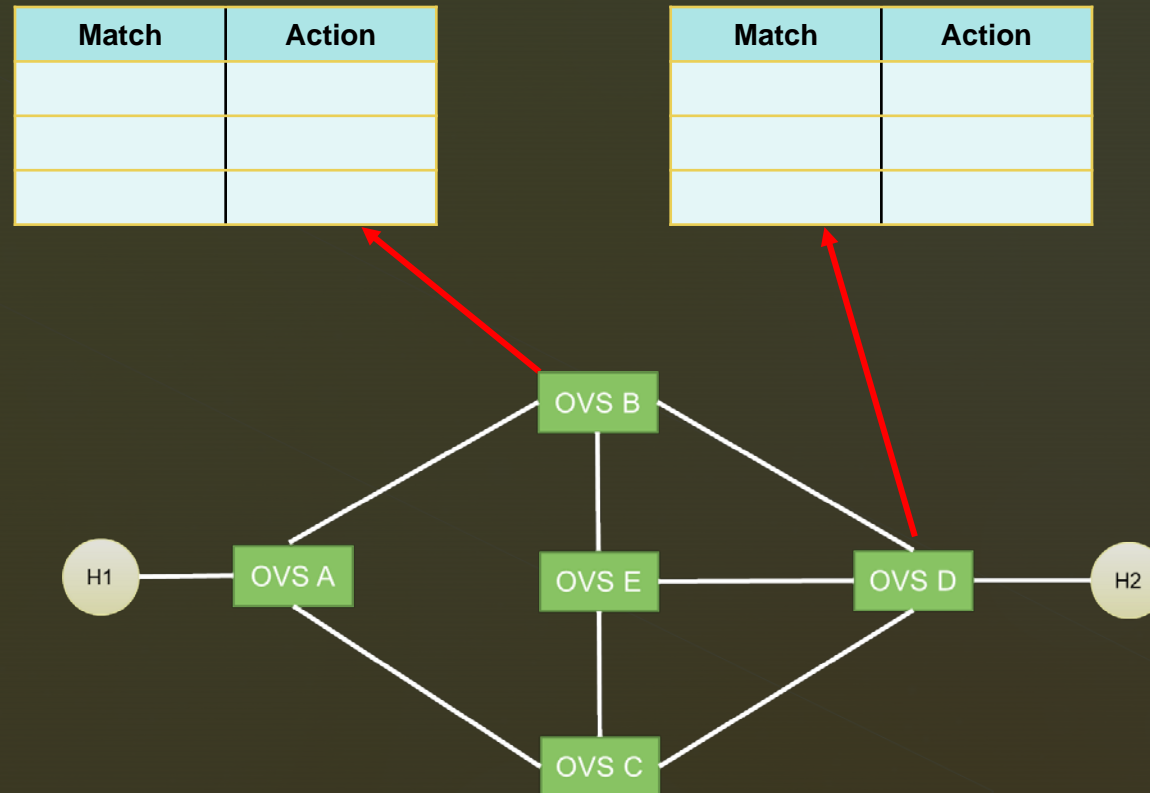
- `sudo mn --custom *.py --topo mytopo --controller remote`



Refer to: <https://github.com/mininet/mininet/blob/master/custom/topo-2sw-2host.py>

Manually installing rules on OVS nodes

Write the rule yourself
“Teach” the switches



(a switch is also called a “bridge”)

Manually installing rules on OVS nodes

- In the lab we need these commands (you may need to sudo):
 - ovs-ofctl show s1
 - ovs-ofctl dump-flows s1
 - ovs-ofctl add-flow s1 in_port=1,actions=output:2
 - ovs-ofctl add-flow s1 priority=500,in_port=1,dl_type=0x0800,nw_proto=6,actions=output:2

Match	Action


- Example:

```
mininet> sh ovs-ofctl dump-flows s2 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=582.062s, table=0, n_packets=1728516, n_bytes=2613493008, in_port=1 actions=output:2
  cookie=0x0, duration=582.036s, table=0, n_packets=52, n_bytes=15918, in_port=2 actions=output:1
mininet> sh ovs-ofctl dump-flows s3 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=584.106s, table=0, n_packets=345719, n_bytes=522722774, in_port=1 actions=output:2
  cookie=0x0, duration=584.082s, table=0, n_packets=0, n_bytes=0, in_port=2 actions=output:1
mininet> sh ovs-ofctl dump-flows s4 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=586.256s, table=0, n_packets=1037149, n_bytes=1568109186, in_port=1 actions=output:2
  cookie=0x0, duration=586.233s, table=0, n_packets=0, n_bytes=0, in_port=2 actions=output:1
mininet>
```

- (Important!!!)** For more details refer to: <https://man7.org/linux/man-pages/man7/ovs-fields.7.html>
<http://manpages.ubuntu.com/manpages/xenial/man8/ovs-ofctl.8.html>

Manually installing rules on OVS nodes

- In the lab we need these commands (you may need to sudo):
 - ovs-ofctl show s1
 - ovs-ofctl dump-flows s1
 - ovs-ofctl add-flow s1 in_port=1,actions=output:2
 - ovs-ofctl add-flow s1 priority=500,in_port=1,dl_type=0x0800,nw_proto=6,actions=output:2



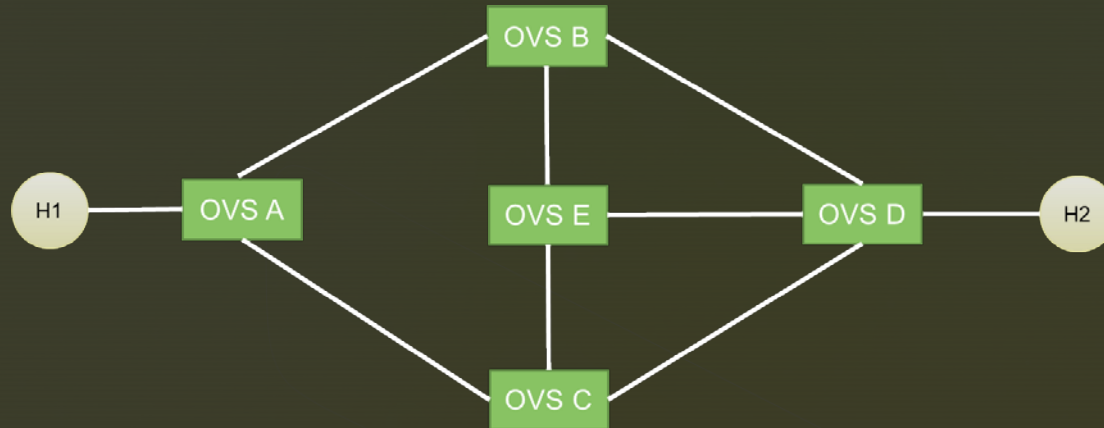
Match	Action
in_port=1	Output: 2

- Example:

```
mininet> sh ovs-ofctl dump-flows s2 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=582.062s, table=0, n_packets=1728516, n_bytes=2613493008, in_port=1 actions=output:2
  cookie=0x0, duration=582.036s, table=0, n_packets=52, n_bytes=15918, in_port=2 actions=output:1
mininet> sh ovs-ofctl dump-flows s3 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=584.106s, table=0, n_packets=345719, n_bytes=522722774, in_port=1 actions=output:2
  cookie=0x0, duration=584.082s, table=0, n_packets=0, n_bytes=0, in_port=2 actions=output:1
mininet> sh ovs-ofctl dump-flows s4 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=586.256s, table=0, n_packets=1037149, n_bytes=1568109186, in_port=1 actions=output:2
  cookie=0x0, duration=586.233s, table=0, n_packets=0, n_bytes=0, in_port=2 actions=output:1
mininet>
```

- (Important!!!)** For more details refer to: <https://man7.org/linux/man-pages/man7/ovs-fields.7.html>
<http://manpages.ubuntu.com/manpages/xenial/man8/ovs-ofctl.8.html>

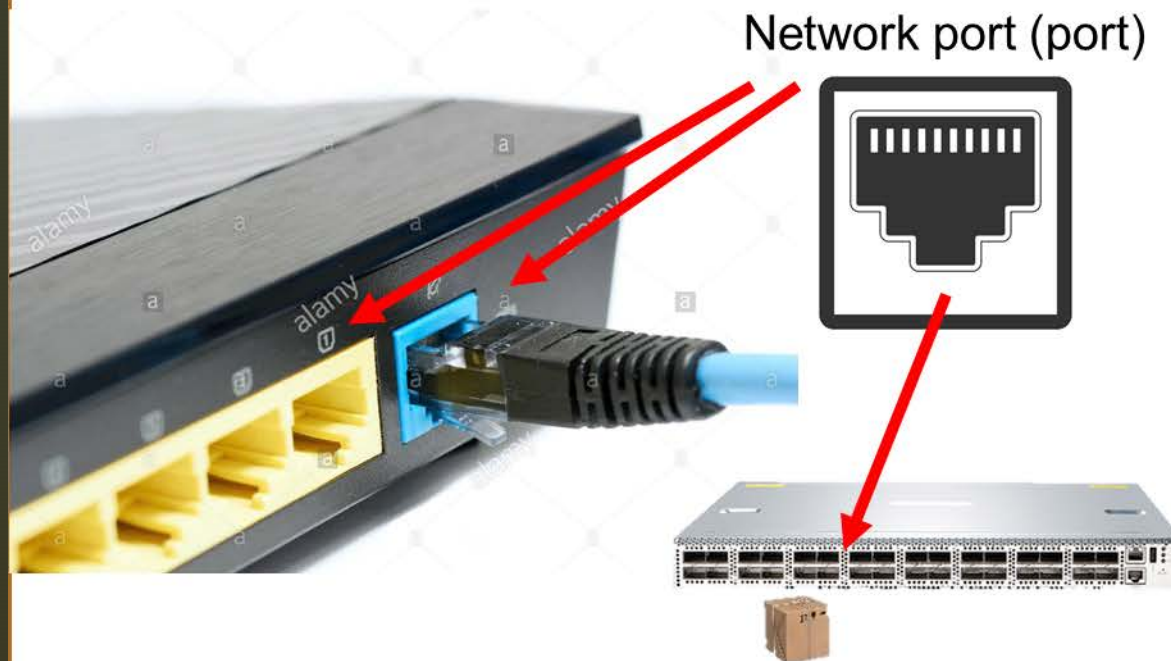
2. Create custom topology and install rules



- Implement flow-based routing instead of destination-based routing
 - Traffic from H1 → H2
 - **HTTP traffic with d_port=80**, follow path: A-B-D
 - other traffic, follow path: A-C-E-D
 - Traffic from H2 → H1
 - HTTP traffic with s_port=80, follow path: D-C-A
 - other traffic, follow path: D-B-E-C-A
- Generate traffic and verify your result with Wireshark

Lab2 Overview (part 1)

Port vs TCP port



TCP port

Labels for application

Port Number	Protocol	Application
20	TCP	FTP data
21	TCP	FTP control
22	TCP	SSH
25	TCP	SMTP
53	UDP, TCP	DNS
80	TCP	HTTP (W/W)
110	TCP	POP3
443	TCP	SSL

- Generate traffic and verify your result with vtireshark

2. Running Mininet with Custom Topology



```
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

- To launch Mininet without default controller:
- `sudo mn --custom *.py --topo mytopo --`
controller remote

2. Running Mininet with Custom Topology

The image displays two terminal windows from a VirtualBox environment running Ubuntu 14.04.3. The left window shows the initial Mininet setup, including network creation, host addition, and starting the controller and switches. A red box highlights the initial ping test results, which show 100% dropped packets. The right window shows the installation of OpenFlow rules on switch s3, with a red box highlighting the successful installation of two rules. The text 'before installing rules' is written in red above the first red box, and 'installing rules' is written in red above the second red box.

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.p
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>

```

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl show s3
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
1(s3-eth1): addr:26:69:da:db:bb:fe
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
2(s3-eth2): addr:4e:d9:99:91:50:7d
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(s3): addr:4e:bb:fb:8b:91:4a
  config: 0
  state: 0
  speed: 0 Mbps now, 0 Mbps max
Ubuntu Software Center (xid=0x4): frags=normal miss_send_len=0
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s3 in_port=1,actions=output:2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s3 in_port=2,actions=output:1
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s4 in_port=1,actions=output:2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s4 in_port=2,actions=output:1
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=28.372s, table=0, n_packets=0, n_bytes=0, idle_age=28, in_port=1 actions=output:2
cookie=0x0, duration=20.517s, table=0, n_packets=25, n_bytes=4964, idle_age=8, in_port=2 actions=output:1
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$

```


2. Running Mininet with Custom Topology

The image displays two terminal windows from a VirtualBox environment running Ubuntu 14.04.3. The left window shows the initial Mininet setup, including network creation, host addition, and a failed ping test. The right window shows the installation of OpenFlow rules on switch s3, followed by a successful ping test.

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.p
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo: s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
  
```

before installing rules

```

frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl show s3
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
1(s3-eth1): addr:26:69:da:db:bb:fe
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
2(s3-eth2): addr:4e:d9:99:91:50:7d
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(s3): addr:4e:bb:fb:8b:91:4a
  config: 0
  state: 0
  speed: 0 Mbps now, 0 Mbps max
Ubuntu Software Center (xid=0x4): frags=normal miss_send_len=0
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s3 in_port=1,actions=output:2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s3 in_port=2,actions=output:1
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s4 in_port=1,actions=output:2
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl add-flow s4 in_port=2,actions=output:1
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$ sudo ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=28.372s, table=0, n_packets=0, n_bytes=0, idle_age=28, in_port=1 actions=output:2
cookie=0x0, duration=20.517s, table=0, n_packets=25, n_bytes=4964, idle_age=8, in_port=2 actions=output:1
frank@frank-VirtualBox: ~/Desktop/EL9333/Lab2$
  
```

installing rules

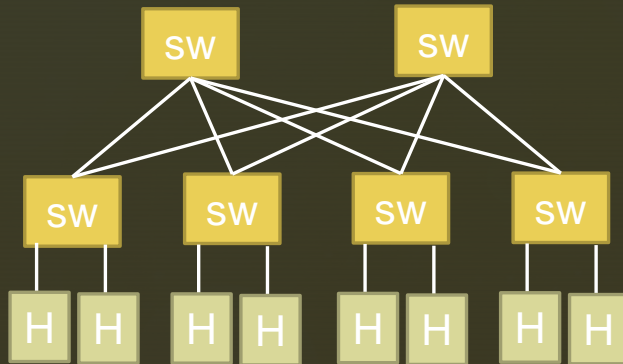
after installing rules

Useful tips

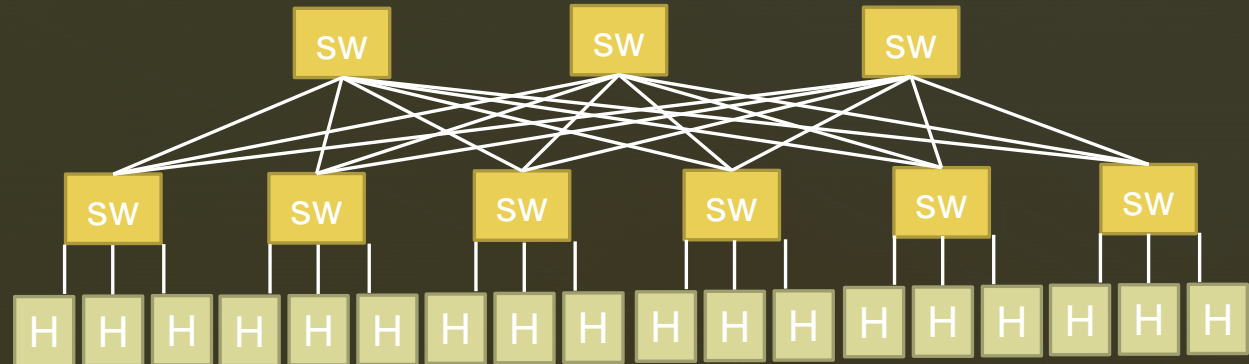
- To clean up the Mininet
 - `$ mn -c`
- Open terminal on one host in Mininet
 - `mininet> xterm h1`
- Run command on host
 - `mininet> h1 <commands>`
- **To generate traffic**
 - **iperf/hping (Refer to lab 1) for HTTP packets**
 - **ping to generate ICMP packets**

Lab2 Overview (part 2)

- Given switches with N ports, create a 2-stage **fat-tree topology** with hosts in Mininet
 - Make N a variable in your topology file/program
- Ex:



N=4



N=6