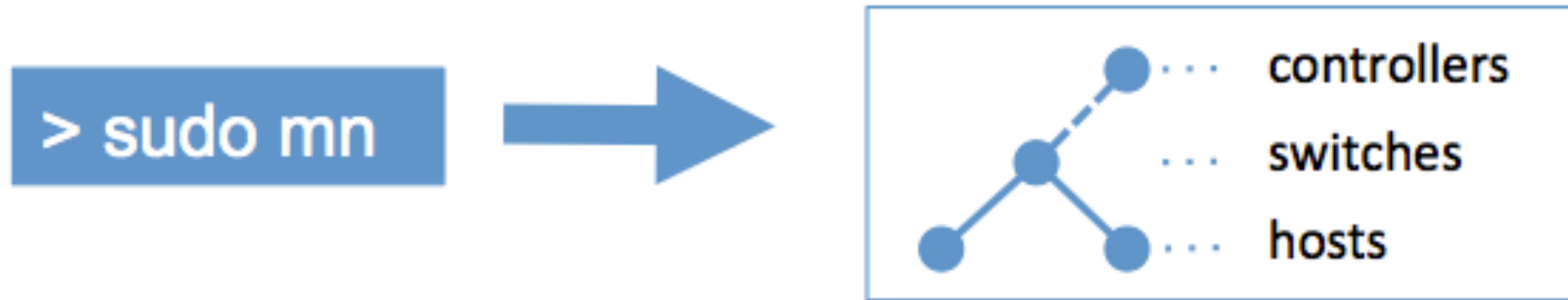


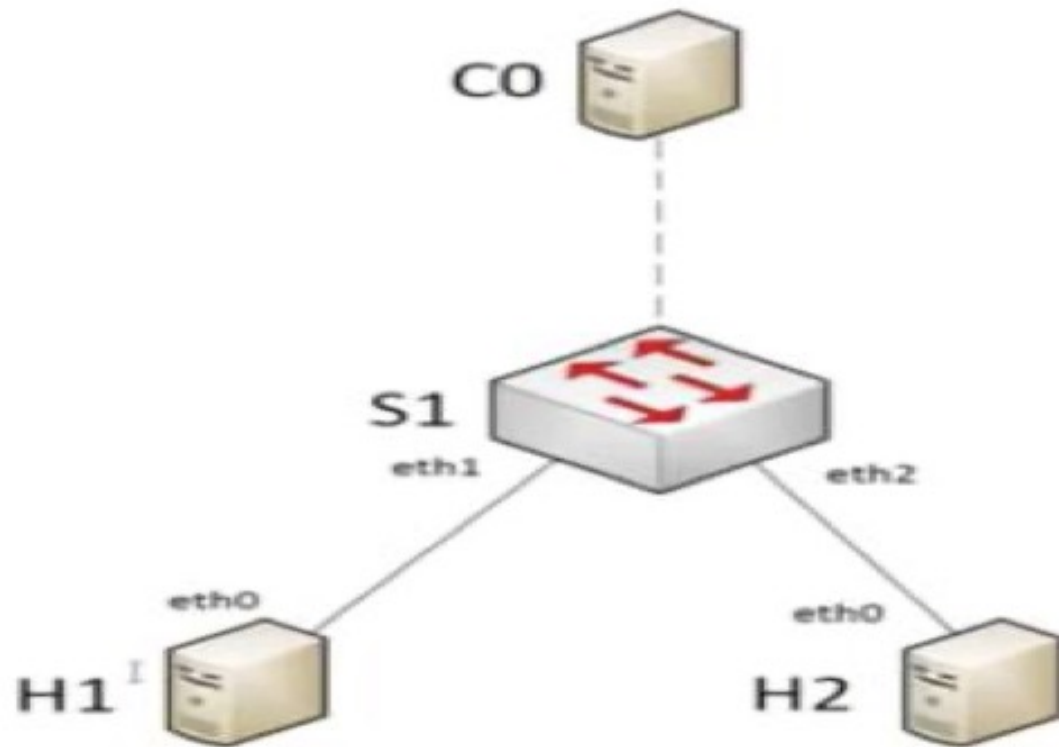
CS452/552  
Assignment 3

Exploring Software-Defined  
Networking

# What's Mininet

- An Instant Virtual Network on your Laptop.
- Mininet creates a realistic virtual network, running real kernel, switch and application code, on a single machine (VM, cloud or native), in seconds, with a single command:





# What's Mininet

- Default minimal topology

# Mininet Basic Operations

- `$docker run -it --name=mininet --privileged -v /tmp/.X11-unix:/tmp/.X11-unix -v /lib/modules:/lib/modules iwaseyusuke/mininet`
- `$mn`
- `mininet> help`
- `mininet> h1 ifconfig -a`
- `mininet> s1 ps -a`
- `mininet> h1 ping -c 1 h2`
- `mininet> pingall`
- `mininet> iperf h1 h2`
- `mininet> exit`

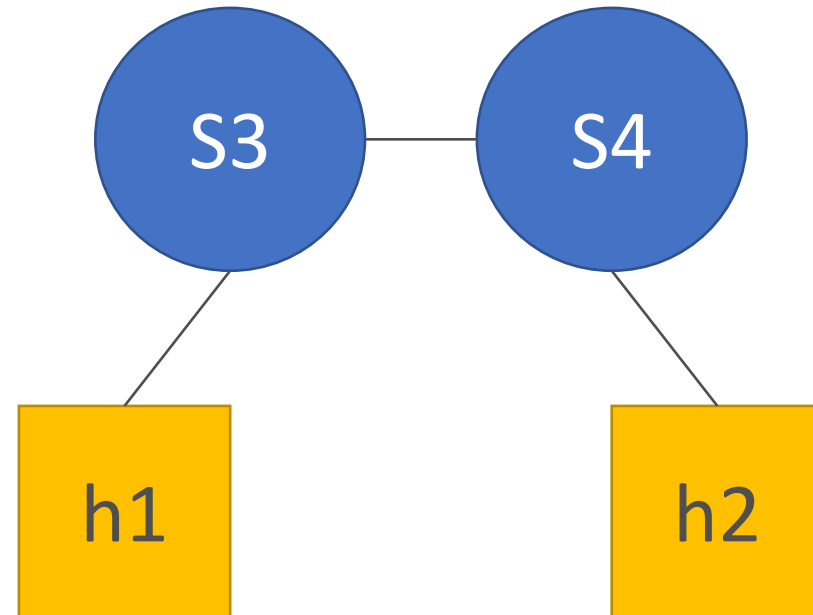
# Mininet



# Virtual Networks in Mininet

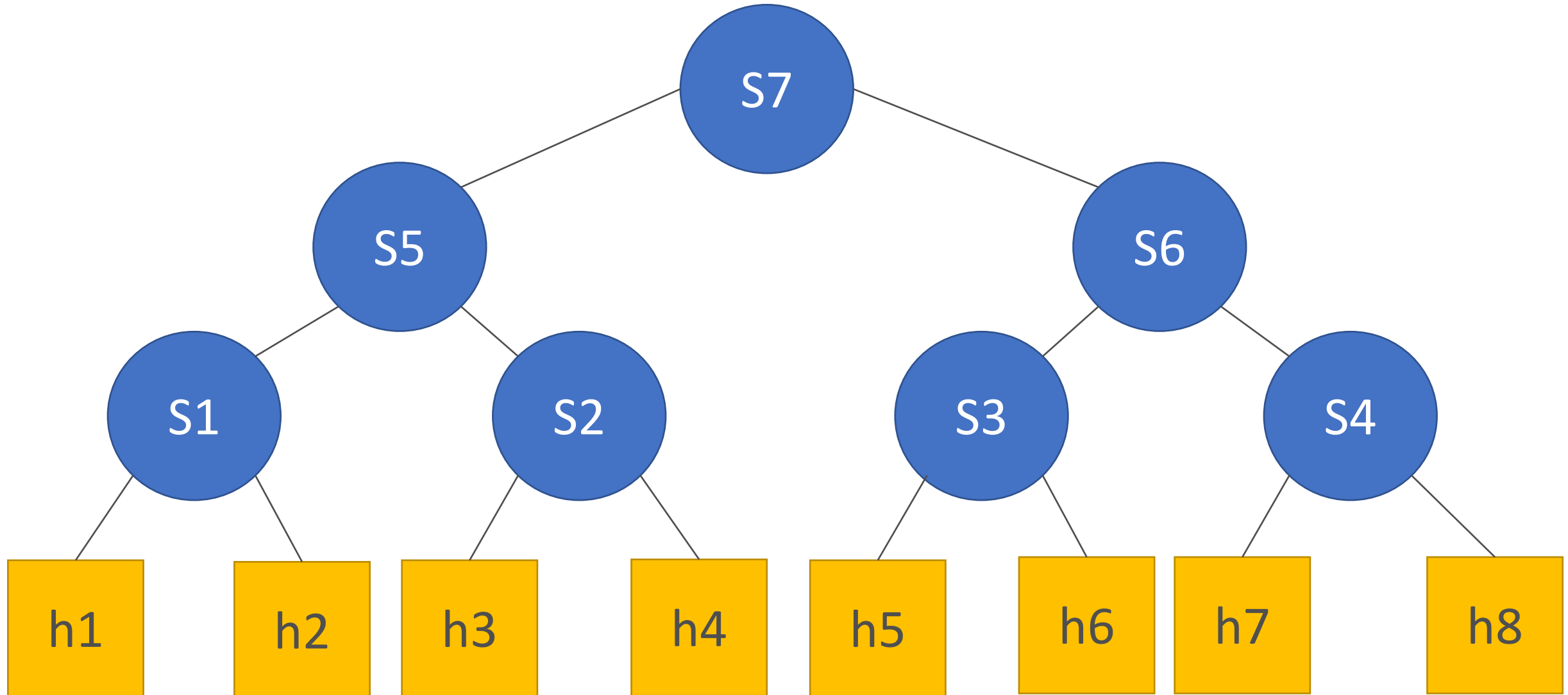
Concrete Example for Task I: `$ sudo mn --custom example.py --topo mytopo`

```
1  """Custom topology example
2
3  Two directly connected switches plus a host for each switch:
4
5  host --- switch --- switch --- host
6
7  Adding the 'topos' dict with a key/value pair to generate our newly defined
8  topology enables one to pass in '--topo=mytopo' from the command line.
9  """
10
11  from mininet.topo import Topo
12
13  class MyTopo( Topo ):
14      "Simple topology example."
15
16      def build( self ):
17          "Create custom topo."
18
19          # Add hosts and switches
20          leftHost = self.addHost( 'h1' )
21          rightHost = self.addHost( 'h2' )
22          leftSwitch = self.addSwitch( 's3' )
23          rightSwitch = self.addSwitch( 's4' )
24
25          # Add links
26          self.addLink( leftHost, leftSwitch )
27          self.addLink( leftSwitch, rightSwitch )
28          self.addLink( rightSwitch, rightHost )
29
30
31  topos = { 'mytopo': ( lambda: MyTopo() ) }
```



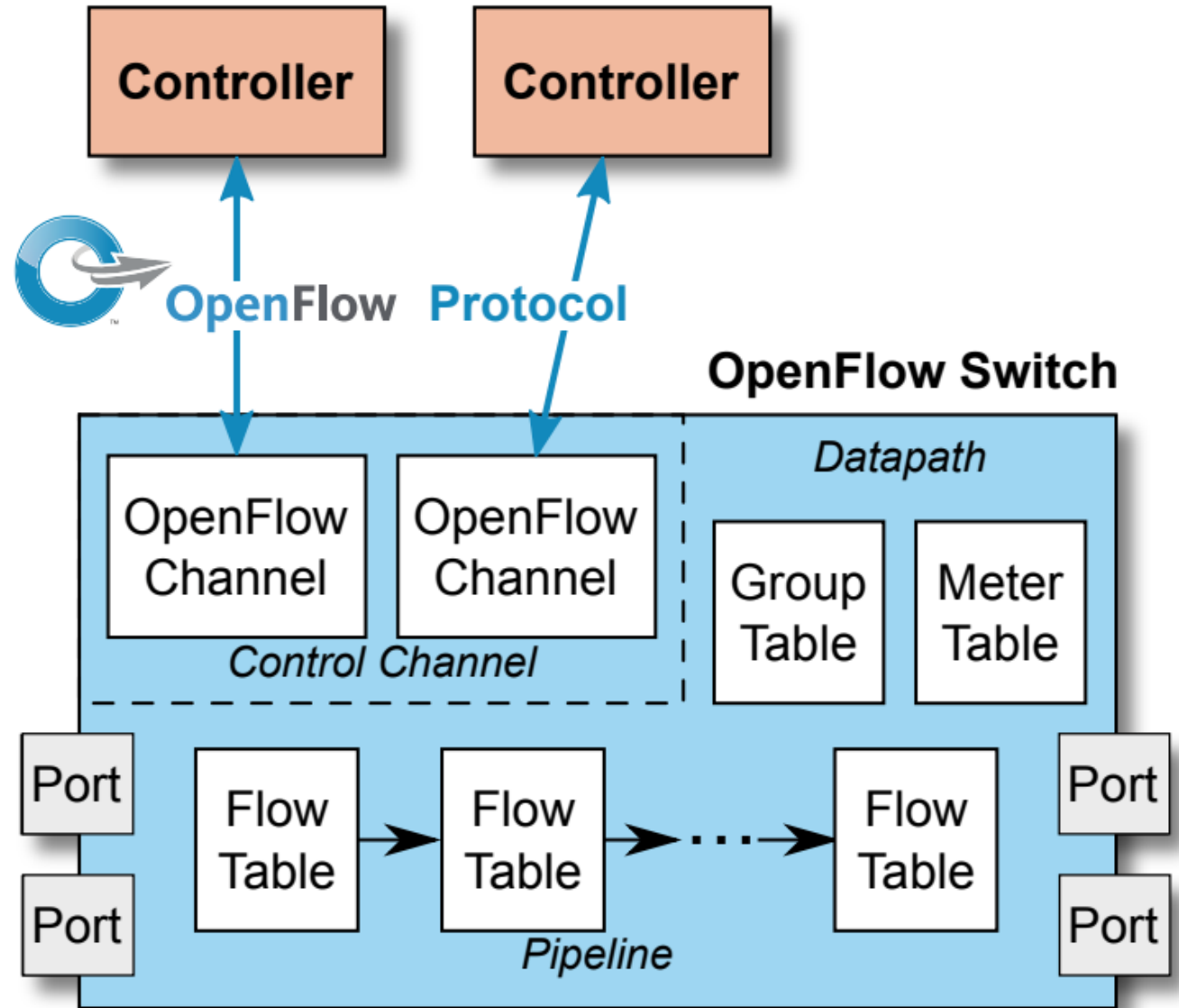
# Concrete Example for Task I

\$ sudo mn --custom binary\_tree.py --topo mytopo



# OpenFlow Switches

- Is an OpenFlow-enabled data center switch realizing network communication through a central controller.



# What's POX

POX is an open source development platform for Python-based software-defined networking (SDN) control application.

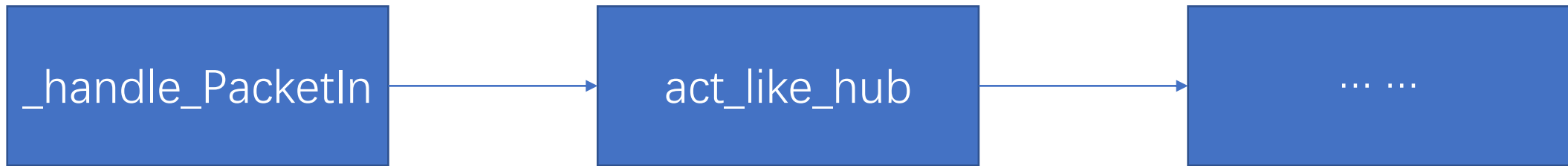
- Download and install it from [github](#).
- `$ cd pox`
- `$ python3 pox.py log.level --DEBUG misc.of_tutorial`
- `~/pox/pox/misc/of_tutorial.py`



# Task II: “of\_tutorial” Controller

Q1. Draw the function call graph

- Where's the code?
- Which function first been called?
  - Example of the graph: Once a package comes to the controller



`_handle_PacketIn -> act_like_hub -> ... ..`

# Task III: MAC Learning Controller

Code:

```
def act_like_switch (self, packet, packet_in):
    # Learn the port for the source MAC
    # print "Src: ",str(packet.src),":", packet_in.in_port,"Dst:", str(packet.dst)
    if packet.src not in self.mac_to_port:
        print "Learning that " + str(packet.src) + " is attached at port " + str(packet_in.in_port)
        self.mac_to_port[packet.src] = packet_in.in_port

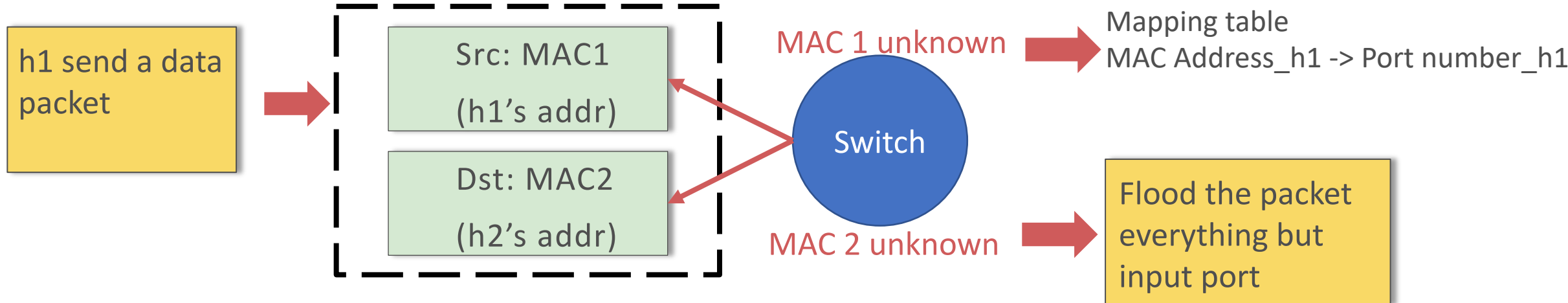
    # if the port associated with the destination MAC of the packet is known:
    if packet.dst in self.mac_to_port:
        # Send packet out the associated port
        print str(packet.dst) + " destination known. only send message to it"
        self.resend_acket(packet_in, self.mac_to_port[packet.dst])
    else:
        # Flood the packet out everything but the input port
        # This part looks familiar, right?
        print str(packet.dst) + " not known, resend to everybody"
        self.resend_packet(packet_in, of.OFPP_ALL)
```

# Discussion

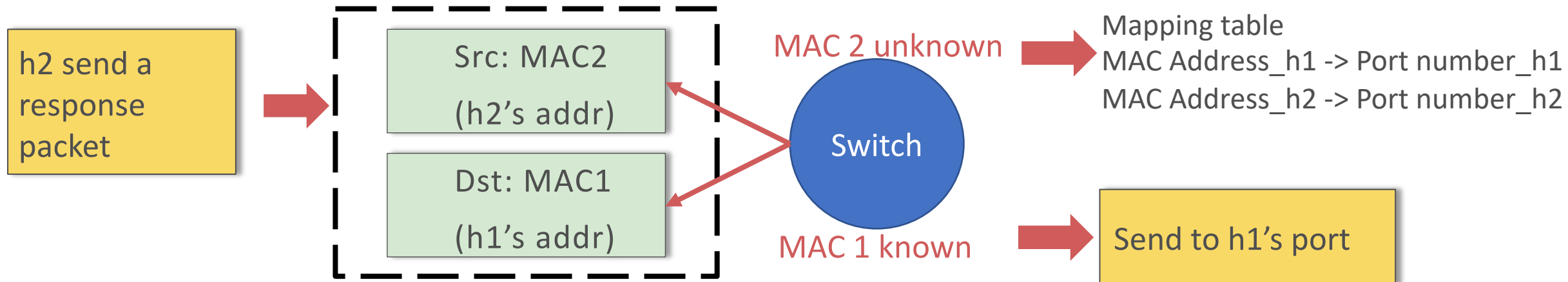
- Difference between Task II and Task III
  - Which performance is better?
    - ping
    - iperf
- act\_like\_hub      **vs.**      act\_like\_switch

# MAC Learning Controller Example:

- h1 ping h2
  - First time h1 ping h2

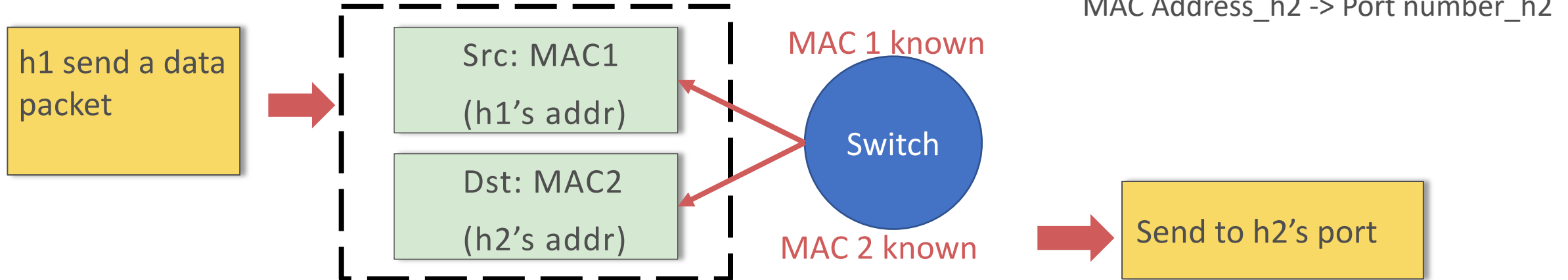


- h2 responds to h1's ping

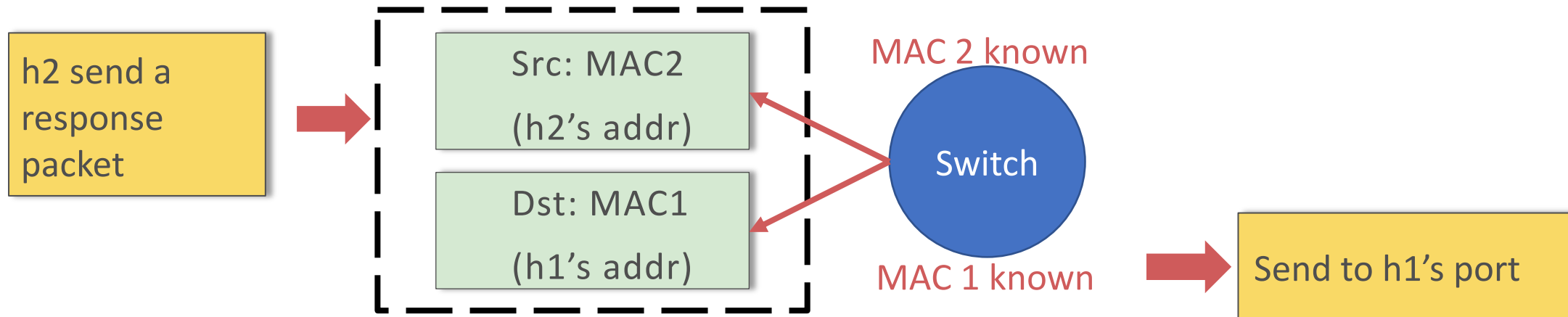


# MAC Learning Controller Example:

- h1 ping h2
  - Second time h1 ping h2



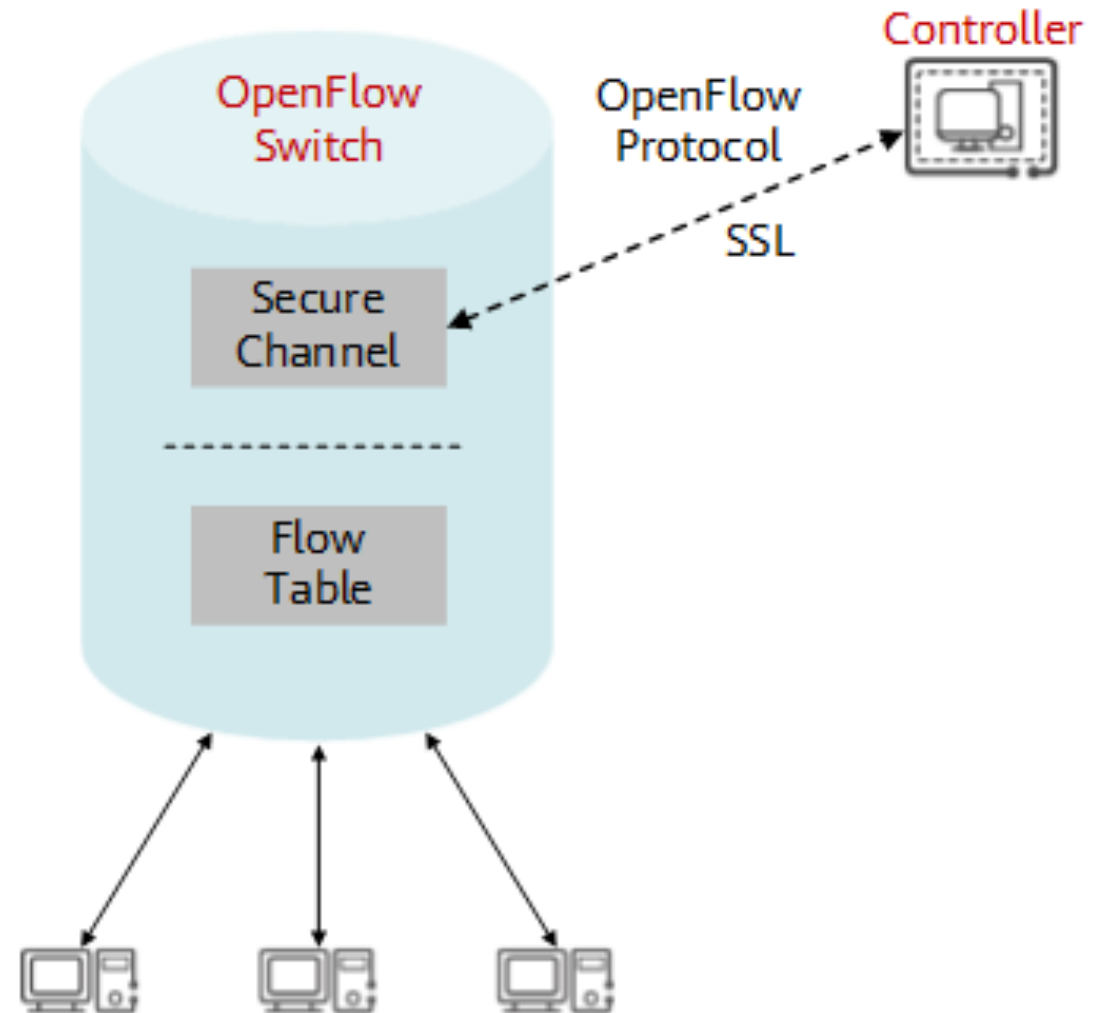
- h2 responds to h1's ping



# Task IV: MAC Learning Controller with OpenFlow Rules

- Benefits

- Difference with Task IV



# Some Questions during Office Hours

- Does task II need to implement the method `act_like_switch`?
- Task II Q4, which of the switches observe traffic?
  - `print()`
  - location of `print()`
- How to do the demo?
  - `of_tutorial3.py`, `of_tutorial4.py`
  - `docker start container_ID`
  - `docker exec -it container_ID /bin/bash`

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
S	NAMES			
ec42c4b4e25c	iwaseyusuke/mininet	"/ENTRYPOINT.sh"	2 days ago	Exited (137) 47 hours ago
mininet				

- Why `$python3 pox.py log.level --DEBUG misc.of_tutorial` doesn't work?

# Summary

- Finish Task I ~ IV. Notice that there should be a pdf to answer the questions.
- Task V is an Optional task, students and can get 10 **bonus** points for it.
- Feel Free to ask questions during office hours or via Email.



**THE END**

**THANK YOU!**