

Step 1: Install and Run Mininet

Installing Mininet

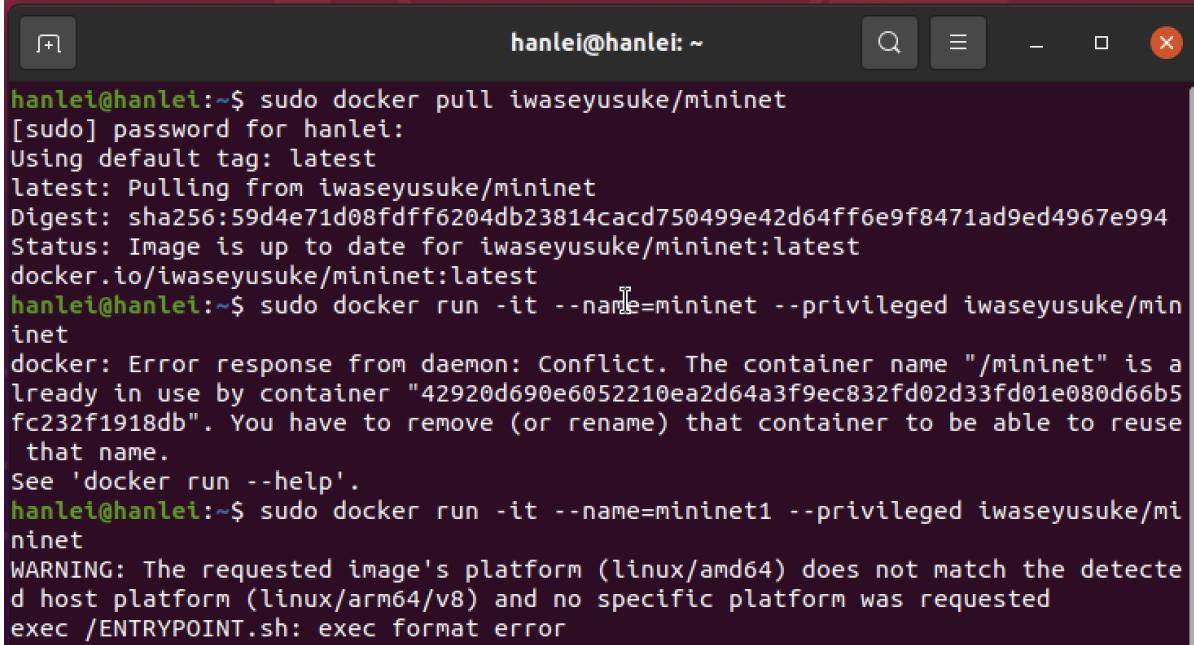
```
$ sudo docker pull iwaseyusuke/mininet
```

```
hanlei@hanlei:~$ sudo docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.dock
er/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

```
hanlei@hanlei:~$ sudo docker pull iwaseyusuke/mininet
[sudo] password for hanlei:
Using default tag: latest
latest: Pulling from iwaseyusuke/mininet
Digest: sha256:59d4e71d08fdff6204db23814cacd750499e42d64ff6e9f8471ad9ed4967e994
Status: Image is up to date for iwaseyusuke/mininet:latest
docker.io/iwaseyusuke/mininet:latest
```

```
$ sudo docker run -it --name=mininet --privileged iwaseyusuke/mininet
```



```
hanlei@hanlei:~$ sudo docker pull iwaseyusuke/mininet
[sudo] password for hanlei:
Using default tag: latest
latest: Pulling from iwaseyusuke/mininet
Digest: sha256:59d4e71d08fdff6204db23814cacd750499e42d64ff6e9f8471ad9ed4967e994
Status: Image is up to date for iwaseyusuke/mininet:latest
docker.io/iwaseyusuke/mininet:latest
hanlei@hanlei:~$ sudo docker run -it --name=mininet --privileged iwaseyusuke/mininet
docker: Error response from daemon: Conflict. The container name "/mininet" is al
ready in use by container "42920d690e6052210ea2d64a3f9ec832fd02d33fd01e080d66b5
fc232f1918db". You have to remove (or rename) that container to be able to reuse
that name.
See 'docker run --help'.
hanlei@hanlei:~$ sudo docker run -it --name=mininet1 --privileged iwaseyusuke/mi
ninet
WARNING: The requested image's platform (linux/amd64) does not match the detected
host platform (linux/arm64/v8) and no specific platform was requested
exec /ENTRYPOINT.sh: exec format error
```

```
$ sudo apt update
```

```
hanlei@hanlei:~$ sudo apt update
Hit:1 http://ports.ubuntu.com/ubuntu-ports focal InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports focal-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports focal-backports InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
hanlei@hanlei:~$
```

```
$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder  
bridge-utils
```

```
hanlei@hanlei:~$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder br  
idge-utils  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package libvirt-bin is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source  
  
E: Package 'libvirt-bin' has no installation candidate  
E: Unable to locate package ubuntu-vm-builder
```

Because the above doesn't work, so i used:

```
$ sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils
```

```
hanlei@hanlei:~$ sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clie  
nts bridge-utils  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
bridge-utils is already the newest version (1.6-2ubuntu1).  
The following additional packages will be installed:  
    ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms libcacard0  
    libfdt1 libibverbs1 libiscsi7 libnss-mymachines libnss-systemd  
    libpam-systemd libpmem1 librados2 librbd1 librdmacm1 libslirp0  
    libspice-server1 libsystemd0 libusbredirparser1 libvirglrenderer1  
    libvirt-daemon libvirt-daemon-driver-qemu libvirt-daemon-storage-rbd  
    libvirt-daemon-system-systemd libvirt0 libxml2-utils qemu-block-extra  
    qemu-efi-aarch64 qemu-efi-arm qemu-system-arm qemu-system-common  
    qemu-system-data qemu-system-gui qemu-utils sharutils systemd  
    systemd-container systemd-sysv systemd-timesyncd  
Suggested packages:  
    gstreamer1.0-plugins-ugly libvirt-daemon-driver-lxc  
    libvirt-daemon-driver-vbox libvirt-daemon-driver-xen  
    libvirt-daemon-storage-gluster libvirt-daemon-storage-zfs
```

```
$ sudo apt install openvswitch-switch
```

```
hanlei@hanlei:~$ sudo apt install openvswitch-switch  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
    libunbound8 openvswitch-common python3-openvswitch python3-sortedcontainers  
Suggested packages:  
    openvswitch-doc python-sortedcontainers-doc  
The following NEW packages will be installed:  
    libunbound8 openvswitch-common openvswitch-switch python3-openvswitch  
    python3-sortedcontainers  
0 upgraded, 5 newly installed, 0 to remove and 19 not upgraded.  
Need to get 2,985 kB of archives.  
After this operation, 14.0 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libunbound8  
arm64 1.9.4-2ubuntu1.3 [317 kB]  
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 python3-sortedcontai  
ners all 2.1.0-2 [27.3 kB]
```

Task 1: Defining custom topologies

```
$ sudo mn --custom binary_tree.py --topo binary_tree
```

```
mininet> h1 ping h8
```

```
binary_tree.py file:
```

The screenshot shows a code editor window titled "binary_tree.py". The code defines a class "BinaryTreeTopo" that inherits from "Topo". It adds 8 hosts (h1-h8) and 7 switches (s1-s7). The hosts are connected to switches s1, s5, and s3 respectively. The switches are interconnected in a binary tree structure: s1 connects to s2 and s5; s2 connects to s3 and s4; s5 connects to s6 and s7; and s3 connects to h1.

```
1 from mininet.topo import Topo
2
3 class BinaryTreeTopo( Topo ):
4     "Binary Tree Topology Class."
5
6     def __init__( self ):
7         "Create the binary tree topology."
8
9         # Initialize topology
10        Topo.__init__( self )
11
12        # Add hosts
13        h1 = self.addHost('h1')
14        h2 = self.addHost('h2')
15        h3 = self.addHost('h3')
16        h4 = self.addHost('h4')
17        h5 = self.addHost('h5')
18        h6 = self.addHost('h6')
19        h7 = self.addHost('h7')
20        h8 = self.addHost('h8')
21
22        # Add switches
23        s1 = self.addSwitch('s1')
24        s2 = self.addSwitch('s2')
25        s3 = self.addSwitch('s3')
26        s4 = self.addSwitch('s4')
27        s5 = self.addSwitch('s5')
28        s6 = self.addSwitch('s6')
29        s7 = self.addSwitch('s7')
30
31        # Add links
32        self.addLink(s1, s2)
33        self.addLink(s1, s5)
34        self.addLink(s2, s3)
35        self.addLink(s2, s4)
36        self.addLink(s5, s6)
37        self.addLink(s5, s7)
38
39        self.addLink(s3, h1)
```

Python ▾ Tab Width: 8 ▾ Ln 3, Col 30 ▾ INS

```
hanlei@hanlei:~$ sudo mn --custom binary_tree.py --topo binary_tree
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:

*** Adding switches:

*** Adding links:

*** Configuring hosts

*** Starting controller

*** Starting 0 switches

*** Starting CLI:
mininet> h1 ping h8
```

```
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
mininet> h1 ping h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=3.36 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.192 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.125 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=0.359 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.105 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.229 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=0.203 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=0.048 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=0.117 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=0.071 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=0.076 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=0.332 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=0.236 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=0.273 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=0.344 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=0.311 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=0.210 ms
64 bytes from 10.0.0.8: icmp_seq=18 ttl=64 time=0.413 ms
64 bytes from 10.0.0.8: icmp_seq=19 ttl=64 time=0.349 ms
64 bytes from 10.0.0.8: icmp_seq=20 ttl=64 time=0.084 ms
64 bytes from 10.0.0.8: icmp_seq=21 ttl=64 time=0.064 ms
```

Questions

1.What is the output of “nodes” and “net”

nodes:

```
mininet> nodes
available nodes are:
h1 h2 h3 h4 h5 h6 h7 h8 s1 s2 s3 s4 s5 s6 s7
```

net:

```
mininet> net
h1 h1-eth0:s3-eth2
h2 h2-eth0:s3-eth3
h3 h3-eth0:s4-eth2
h4 h4-eth0:s4-eth3
h5 h5-eth0:s6-eth2
h6 h6-eth0:s6-eth3
h7 h7-eth0:s7-eth2
h8 h8-eth0:s7-eth3
s1 lo: s1-eth1:s2-eth1 s1-eth2:s5-eth1
s2 lo: s2-eth1:s1-eth1 s2-eth2:s3-eth1 s2-eth3:s4-eth1
s3 lo: s3-eth1:s2-eth2 s3-eth2:h1-eth0 s3-eth3:h2-eth0
s4 lo: s4-eth1:s2-eth3 s4-eth2:h3-eth0 s4-eth3:h4-eth0
s5 lo: s5-eth1:s1-eth2 s5-eth2:s6-eth1 s5-eth3:s7-eth1
s6 lo: s6-eth1:s5-eth2 s6-eth2:h5-eth0 s6-eth3:h6-eth0
s7 lo: s7-eth1:s5-eth3 s7-eth2:h7-eth0 s7-eth3:h8-eth0
c0
```

2.What is the output of “h7 ifconfig”

```
mininet> h7 ifconfig
h7-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.0.7 netmask 255.0.0.0 broadcast 10.255.255.255
        inet6 fe80::4462:2ff:fefb:8423 prefixlen 64 scopeid 0x20<link>
              ether 46:62:02:fb:84:23 txqueuelen 1000 (Ethernet)
                    RX packets 228 bytes 25001 (25.0 KB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 10 bytes 796 (796.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
              loop txqueuelen 1000 (Local Loopback)
                    RX packets 0 bytes 0 (0.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 0 bytes 0 (0.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Step 2: Install POX

Installing POX

```
$ sudo docker exec -it mininet /bin/bash
```

```
hanlei@hanlei:~$ docker run -it ubuntu:latest /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
0509fae36eb0: Pull complete
Digest: sha256:4b1d0c4a2d2aaaf63b3711f34eb9fa89fa1bf53dd6e4ca954d47caebca4005c2
Status: Downloaded newer image for ubuntu:latest
root@ee94e39a2799:/#
root@ee94e39a2799:/#
root@ee94e39a2799:/#
root@ee94e39a2799:/# sudo docker exec -it mininet1 /bin/bash
bash: sudo: command not found
root@ee94e39a2799:/# ^C
root@ee94e39a2799:/# sudo docker exec -it mininet /bin/bash
bash: sudo: command not found
root@ee94e39a2799:/# apt-get update
Get:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease [270 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [114 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease [99.8 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy/main arm64 Packages [1758 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 Packages [17.2 M
B]
Get:7 http://ports.ubuntu.com/ubuntu-ports jammy/multiverse arm64 Packages [224
```

Then download POX by running the following commands:

```
$ apt-get update
```

```
root@ee94e39a2799:/# apt-get update
Get:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease [270 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [114 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease [99.8 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy/main arm64 Packages [1758 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 Packages [17.2 M
B]
Get:7 http://ports.ubuntu.com/ubuntu-ports jammy/multiverse arm64 Packages [224
kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports jammy/restricted arm64 Packages [24.2
kB]
Get:9 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 Packages [84
4 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports jammy-updates/restricted arm64 Packa
ges [222 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports jammy-updates/multiverse arm64 Packa
ges [2308 B]
```

```
$ apt-get install git
```

```
root@ee94e39a2799:/# apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates git-man less libbrotli1 libbsd0 libcbor0.8 libcurl3-gnutls
  libedit2 liberror-perl libexpat1 libfido2-1 libgdbm-compat4 libgdbm6
  libldap-2.5-0 libldap-common libmd0 libnghpp2-14 libperl5.34 libpsl5
  librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libssh-4 libx11-6
  libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxmuu1 netbase
  openssh-client openssl patch perl perl-modules-5.34 publicsuffix xauth
Suggested packages:
  gettext-base git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui
  gitk gitweb git-cvs git-mediawiki git-svn gdbm-l10n
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal
  libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql keychain
  libpam-ssh monkeysphere ssh-askpass ed diffutils-doc perl-doc
  libterm-readline-gnu-perl | libterm-readline-perl-perl make
  libtap-harness-archive-perl
The following NEW packages will be installed:
```

\$ apt-get install python3

```
root@ee94e39a2799:/# apt-get install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libmpdec3 libpython3-stdlib libpython3.10-minimal libpython3.10-stdlib
  libreadline8 libsqlite3-0 media-types python3-minimal python3.10
  python3.10-minimal readline-common
Suggested packages:
  python3-doc python3-tk python3-venv python3.10-venv python3.10-doc binutils
  binfmt-support readline-doc
The following NEW packages will be installed:
  libmpdec3 libpython3-stdlib libpython3.10-minimal libpython3.10-stdlib
  libreadline8 libsqlite3-0 media-types python3 python3-minimal python3.10
  python3.10-minimal readline-common
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 6394 kB of archives.
After this operation, 22.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

\$ env GIT_SSL_NO_VERIFY=true git clone https://github.com/noxrepo/pox

```
root@ee94e39a2799:/# env GIT_SSL_NO_VERIFY=true git clone https://github.com/nox
repo/pox
Cloning into 'pox'...
remote: Enumerating objects: 13036, done.
remote: Counting objects: 100% (261/261), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 13036 (delta 161), reused 230 (delta 140), pack-reused 12775
Receiving objects: 100% (13036/13036), 9.01 MiB | 5.96 MiB/s, done.
Resolving deltas: 100% (8410/8410), done.
```

\$./pox.py log.level --DEBUG misc.of_tutorial

```

root@ee94e39a2799:/# cd pox
root@ee94e39a2799:/pox# ./pox.py log.level --DEBUG misc.of_tutorial
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
DEBUG:core:POX 0.7.0 (gar) going up...
DEBUG:core:Running on CPython (3.10.6/Nov 2 2022 18:53:38)
DEBUG:core:Platform is Linux-5.4.0-132-generic-aarch64-with-glibc2.35
WARNING:version:POX requires one of the following versions of Python: 3.6 3.7 3.
8 3.9
WARNING:version:You're running Python 3.10.
WARNING:version:If you run into problems, try using a supported version.
INFO:core:POX 0.7.0 (gar) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633

```

```
$ sudo mn --custom binary_tree.py --controller remote --topo
binary_tree
```

The screenshot shows two terminal windows side-by-side. The left window is titled 'root@ee94e39a2799: /pox' and displays the output of the command above. The right window is titled 'hanlei@hanlei: ~' and shows the usage information for the 'mn' command.

```

hanlei@hanlei:~$ sudo mn --custom binary_tree.py --controller remote --topo bin
ary tree
Usage: mn [options]
(type mn -h for details)

The mn utility creates Mininet network from the command line. It can create
parametrized topologies, invoke the Mininet CLI, and run tests.

Options:
-h, --help          show this help message and exit
--switch=SWITCH    default|ivs|lxbr|ovs|ovsbr|ovsk|user[,param=value...]
                  ovs=OVSSwitch default=OVSSwitch ovsk=OVSSwitch
                  lxbr=LinuxBridge user=UserSwitch ivs=IVSSwitch
                  ovsbr=OVSBridge      []
                  cfs|proc|rt[,param=value...]
                  rt=CPULimitedHost{'sched': 'rt'} proc=Host
                  cfs=CPULimitedHost{'sched': 'cfs'}
--host=HOST         default|none|nox|ovsc|ref|remote|ryu[,param=value...]
                  ovsc=OVSCController none=NullController
                  remote=RemoteController default=DefaultController
                  nox=NOX ryu=Ryu ref=Controller
--controller=CONTROLLER
                  default|none|nox|ovsc|ref|remote|ryu[,param=value...]
                  ovsc=OVSCController none=NullController
                  remote=RemoteController default=DefaultController
                  nox=NOX ryu=Ryu ref=Controller
--link=LINK         default|ovs|tc|tcu[,param=value...] default=Link
                  ovs=OVSLink tcu=TCULink tc=TCLink

```

```

--link=LINK           default|ovs|tc|tcu[,param=value...] default=Link
--topo=TOPO          ovs=OVSLink tcu=TCULink tc=TCLink
                     linear|minimal|reversed|single|torus|tree[,param=value]
                     ...] linear=LinearTopo torus=TorusTopo tree=TreeTopo
                     single=SingleSwitchTopo
                     reversed=SingleSwitchReversedTopo minimal=MinimalTopo
                     clean and exit
--clean              read custom classes or params from .py file(s)
--custom=CUSTOM      cli|build|pingall|pingpair|iperf|all|iperfudp|none|pin
                     gpair|iperfudp|pingall|iperfUDP
--test=TESTS         spawn xterms for each node
-x, --xterms        base IP address for hosts
-i IPBASE, --ipbase=IPBASE
                     automatically set host MACs
--mac               set all-pairs ARP entries
--arp               set all-pairs ARP entries
-v VERBOSITY, --verbosity=VERBOSITY
                     info|warning|critical|error|debug|output
--innamespace       sw and ctrl in namespace?
--listenport=LISTENPORT
                     base port for passive switch listening
--nolistenport      don't use passive listening port
--pre=PRE            CLI script to run before tests
--post=POST          CLI script to run after tests
--pin               pin hosts to CPU cores (requires --host cfs or --host
                   rt)
--nat               [option=val...] adds a NAT to the topology that
                     connects Mininet hosts to the physical network.
                     Warning: This may route any traffic on the machine
                     that uses Mininet's IP subnet into the Mininet
                     network. If you need to change Mininet's IP subnet,
                     see the --ipbase option.
--version           prints the version and exits
--cluster=server1,server2...

```

```

--nat               [option=val...] adds a NAT to the topology that
                     connects Mininet hosts to the physical network.
                     Warning: This may route any traffic on the machine
                     that uses Mininet's IP subnet into the Mininet
                     network. If you need to change Mininet's IP subnet,
                     see the --ipbase option.
--version           prints the version and exits
--cluster=server1,server2...
                     run on multiple servers (experimental!)
--placement=block|random
                     node placement for --cluster (experimental!)

```

Task 2: Analyze the “of_tutorial” controller Questions

1. Draw the function call graph of this controller. For example, once a packet comes to the controller, which function is the first to be called, which one is the second, and so forth?
2. Have h1 ping h2, and h1 ping h8 for 100 times (e.g., h1 ping -c100 p2).
h1 ping h2:

```
hanlei@hanlei:~$ sudo mn --custom binary_tree.py --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping -c100 p2
ping: p2: Name or service not known
mininet> h1 ping -c100 h2
```

```
mininet> h1 ping -c100 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.80 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.195 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.149 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.108 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.127 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.323 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.321 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.206 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.166 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.151 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.691 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.131 ms
```

```
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.131 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.217 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.180 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=0.263 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.238 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.510 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.278 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.148 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=1.70 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.336 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.160 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.138 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.121 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.623 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=2.02 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=0.148 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.159 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=51 ttl=64 time=0.287 ms
64 bytes from 10.0.0.2: icmp_seq=52 ttl=64 time=0.228 ms
64 bytes from 10.0.0.2: icmp_seq=53 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=54 ttl=64 time=0.114 ms
64 bytes from 10.0.0.2: icmp_seq=55 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=56 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=57 ttl=64 time=0.063 ms
```

```
64 bytes from 10.0.0.2: icmp_seq=57 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=58 ttl=64 time=0.379 ms
64 bytes from 10.0.0.2: icmp_seq=59 ttl=64 time=0.204 ms
64 bytes from 10.0.0.2: icmp_seq=60 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=61 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=62 ttl=64 time=0.112 ms
64 bytes from 10.0.0.2: icmp_seq=63 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=65 ttl=64 time=0.150 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=0.357 ms
64 bytes from 10.0.0.2: icmp_seq=67 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=68 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=0.335 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=0.204 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=0.248 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=0.275 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.140 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.360 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.545 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.268 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.286 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.295 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.091 ms
```

```

64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.545 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.268 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.286 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.295 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.138 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.166 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.111 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.187 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.307 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=2.06 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.315 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.202 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.176 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 101129ms
rtt min/avg/max/mdev = 0.032/0.288/4.800/0.568 ms
mininet>

```

h1 ping h8:

```

s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
mininet> h1 ping -c100 h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=2.59 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.136 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.248 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=0.453 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.130 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.142 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=0.412 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=2.32 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=0.122 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=0.347 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=0.175 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=0.197 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=0.281 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=0.127 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=0.270 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=0.065 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=0.129 ms
64 bytes from 10.0.0.8: icmp_seq=18 ttl=64 time=0.342 ms
64 bytes from 10.0.0.8: icmp_seq=19 ttl=64 time=0.252 ms
64 bytes from 10.0.0.8: icmp_seq=20 ttl=64 time=0.349 ms
64 bytes from 10.0.0.8: icmp_seq=21 ttl=64 time=0.175 ms
64 bytes from 10.0.0.8: icmp_seq=22 ttl=64 time=0.208 ms
64 bytes from 10.0.0.8: icmp_seq=23 ttl=64 time=0.145 ms
64 bytes from 10.0.0.8: icmp_seq=24 ttl=64 time=0.501 ms
64 bytes from 10.0.0.8: icmp_seq=25 ttl=64 time=0.155 ms
64 bytes from 10.0.0.8: icmp_seq=26 ttl=64 time=0.170 ms
64 bytes from 10.0.0.8: icmp_seq=27 ttl=64 time=0.121 ms
64 bytes from 10.0.0.8: icmp_seq=28 ttl=64 time=0.170 ms
64 bytes from 10.0.0.8: icmp_seq=29 ttl=64 time=0.406 ms
64 bytes from 10.0.0.8: icmp_seq=30 ttl=64 time=0.048 ms
64 bytes from 10.0.0.8: icmp_seq=31 ttl=64 time=1.04 ms
64 bytes from 10.0.0.8: icmp_seq=32 ttl=64 time=0.230 ms
64 bytes from 10.0.0.8: icmp_seq=33 ttl=64 time=0.140 ms

```

```
64 bytes from 10.0.0.8: icmp_seq=33 ttl=64 time=0.140 ms
64 bytes from 10.0.0.8: icmp_seq=34 ttl=64 time=0.378 ms
64 bytes from 10.0.0.8: icmp_seq=35 ttl=64 time=0.399 ms
64 bytes from 10.0.0.8: icmp_seq=36 ttl=64 time=0.333 ms
64 bytes from 10.0.0.8: icmp_seq=37 ttl=64 time=0.670 ms
64 bytes from 10.0.0.8: icmp_seq=38 ttl=64 time=0.452 ms
64 bytes from 10.0.0.8: icmp_seq=39 ttl=64 time=0.160 ms
64 bytes from 10.0.0.8: icmp_seq=40 ttl=64 time=0.412 ms
64 bytes from 10.0.0.8: icmp_seq=41 ttl=64 time=0.260 ms
64 bytes from 10.0.0.8: icmp_seq=42 ttl=64 time=0.140 ms
64 bytes from 10.0.0.8: icmp_seq=43 ttl=64 time=0.210 ms
64 bytes from 10.0.0.8: icmp_seq=44 ttl=64 time=0.077 ms
64 bytes from 10.0.0.8: icmp_seq=45 ttl=64 time=0.579 ms
64 bytes from 10.0.0.8: icmp_seq=46 ttl=64 time=0.081 ms
64 bytes from 10.0.0.8: icmp_seq=47 ttl=64 time=0.491 ms
64 bytes from 10.0.0.8: icmp_seq=48 ttl=64 time=0.187 ms
64 bytes from 10.0.0.8: icmp_seq=49 ttl=64 time=0.193 ms
64 bytes from 10.0.0.8: icmp_seq=50 ttl=64 time=0.195 ms
64 bytes from 10.0.0.8: icmp_seq=51 ttl=64 time=0.113 ms
64 bytes from 10.0.0.8: icmp_seq=52 ttl=64 time=0.070 ms
64 bytes from 10.0.0.8: icmp_seq=53 ttl=64 time=0.128 ms
64 bytes from 10.0.0.8: icmp_seq=54 ttl=64 time=0.419 ms
64 bytes from 10.0.0.8: icmp_seq=55 ttl=64 time=0.373 ms
64 bytes from 10.0.0.8: icmp_seq=56 ttl=64 time=0.230 ms
64 bytes from 10.0.0.8: icmp_seq=57 ttl=64 time=0.212 ms
64 bytes from 10.0.0.8: icmp_seq=58 ttl=64 time=0.165 ms
64 bytes from 10.0.0.8: icmp_seq=59 ttl=64 time=0.142 ms
64 bytes from 10.0.0.8: icmp_seq=60 ttl=64 time=0.130 ms
64 bytes from 10.0.0.8: icmp_seq=61 ttl=64 time=0.691 ms
64 bytes from 10.0.0.8: icmp_seq=62 ttl=64 time=0.397 ms
64 bytes from 10.0.0.8: icmp_seq=63 ttl=64 time=0.292 ms
64 bytes from 10.0.0.8: icmp_seq=64 ttl=64 time=0.147 ms
64 bytes from 10.0.0.8: icmp_seq=65 ttl=64 time=0.163 ms
64 bytes from 10.0.0.8: icmp_seq=66 ttl=64 time=0.351 ms
64 bytes from 10.0.0.8: icmp_seq=67 ttl=64 time=0.341 ms
64 bytes from 10.0.0.8: icmp_seq=68 ttl=64 time=0.157 ms
64 bytes from 10.0.0.8: icmp_seq=69 ttl=64 time=0.196 ms
```

```

64 bytes from 10.0.0.8: icmp_seq=69 ttl=64 time=0.196 ms
64 bytes from 10.0.0.8: icmp_seq=70 ttl=64 time=0.410 ms
64 bytes from 10.0.0.8: icmp_seq=71 ttl=64 time=0.648 ms
64 bytes from 10.0.0.8: icmp_seq=72 ttl=64 time=0.385 ms
64 bytes from 10.0.0.8: icmp_seq=73 ttl=64 time=0.171 ms
64 bytes from 10.0.0.8: icmp_seq=74 ttl=64 time=0.299 ms
64 bytes from 10.0.0.8: icmp_seq=75 ttl=64 time=0.130 ms
64 bytes from 10.0.0.8: icmp_seq=76 ttl=64 time=0.248 ms
64 bytes from 10.0.0.8: icmp_seq=77 ttl=64 time=0.253 ms
64 bytes from 10.0.0.8: icmp_seq=78 ttl=64 time=0.705 ms
64 bytes from 10.0.0.8: icmp_seq=79 ttl=64 time=0.063 ms
64 bytes from 10.0.0.8: icmp_seq=80 ttl=64 time=0.059 ms
64 bytes from 10.0.0.8: icmp_seq=81 ttl=64 time=0.115 ms
64 bytes from 10.0.0.8: icmp_seq=82 ttl=64 time=0.735 ms
64 bytes from 10.0.0.8: icmp_seq=83 ttl=64 time=0.145 ms
64 bytes from 10.0.0.8: icmp_seq=84 ttl=64 time=0.191 ms
64 bytes from 10.0.0.8: icmp_seq=85 ttl=64 time=0.142 ms
64 bytes from 10.0.0.8: icmp_seq=86 ttl=64 time=0.375 ms
64 bytes from 10.0.0.8: icmp_seq=87 ttl=64 time=0.746 ms
64 bytes from 10.0.0.8: icmp_seq=88 ttl=64 time=0.450 ms
64 bytes from 10.0.0.8: icmp_seq=89 ttl=64 time=0.136 ms
64 bytes from 10.0.0.8: icmp_seq=90 ttl=64 time=0.297 ms
64 bytes from 10.0.0.8: icmp_seq=91 ttl=64 time=0.141 ms
64 bytes from 10.0.0.8: icmp_seq=92 ttl=64 time=0.216 ms
64 bytes from 10.0.0.8: icmp_seq=93 ttl=64 time=0.184 ms
64 bytes from 10.0.0.8: icmp_seq=94 ttl=64 time=0.155 ms
64 bytes from 10.0.0.8: icmp_seq=95 ttl=64 time=0.372 ms
64 bytes from 10.0.0.8: icmp_seq=96 ttl=64 time=0.350 ms
64 bytes from 10.0.0.8: icmp_seq=97 ttl=64 time=0.219 ms
64 bytes from 10.0.0.8: icmp_seq=98 ttl=64 time=0.220 ms
64 bytes from 10.0.0.8: icmp_seq=99 ttl=64 time=0.426 ms
64 bytes from 10.0.0.8: icmp_seq=100 ttl=64 time=0.064 ms

--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 101116ms
rtt min/avg/max/mdev = 0.048/0.315/2.594/0.354 ms
mininet> 
```

a. How long does it take (on average) to ping for each case?

For h1 ping h2, the average time is 0.288 ms.

For h1 ping h8, the average time is 0.315 ms.

b. What is the minimum and maximum ping you have observed?

For h1 ping h2, the minimum time is 0.032 ms, the maximum time is 4.800 ms.

For h1 ping h8, the minimum time is 0.048 ms, the maximum time is 2.594 ms.

c. What is the difference, and why?

The difference is that h1 ping h8 takes longer time than h1 ping h2.

Because for h1 ping h2, packets only need to go through switch s3,

but for h1 ping h8, packets need to go through switches

s3 -> s2 -> s1 -> s5 -> s7.

3. Run “iperf h1 h2” and “iperf h1 h8”

```

mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['101 Gbits/sec', '101 Gbits/sec']
mininet> 
```

```
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['83.2 Gbits/sec', '83.3 Gbits/sec']
```

a.What is “iperf” used for?

Iperf: testing TCP bandwidth between h1 and h8, h1 and h2 by measuring how much data can be transferred between two nodes within a given amount of time

b.What is the throughput for each case?

For h1 - h2: 101 Mbits/sec

For h2 - h1: 101 Mbits/sec

For h1 - h8: 83.2 Mbits/sec

For h8 - h1: 83.3 Mbits/sec

c.What is the difference, and explain the reasons for the difference.

The difference is that the throughput of h1 ping h8 is lower than h1 ping h2.

Because h1 - h2 only needs to go through switch s3, which takes less time to transfer for one packet and more packets can be transferred within a given amount of time.

But h1 ping h8 needs to go through s3 -> s2 -> s1 -> s5 -> s7, which takes more time to transfer for one packet and less packets can be transferred within a given amount of time.

4.Which of the switches observe traffic? Please describe your way for observing such traffic on switches (e.g., adding some functions in the “of_tutorial” controller).

All the switches observe traffic, because when a packet is sent, it would be sent to all the switches instead of just the ones between the two communicating hosts. If I want to observe this traffic, I can add a print statement in the _handle_PacketIn() function to show when the traffic goes through the switches.

Task 3: MAC Learning Controller

The screenshot shows a code editor window with the file name *off_tutorial.py. The code is a Python script for a MAC Learning Controller. It defines a class Tutorial that handles packet forwarding and learning. The code includes comments explaining the purpose of each section.

```
20 """
21
22 from pox.core import core
23 import pox.openflow.libopenflow_01 as of
24
25 log = core.getLogger()
26
27
28
29 class Tutorial (object):
30     """
31     A Tutorial object is created for each switch that connects.
32     A Connection object for that switch is passed to the
33     __init__ function.
34     """
35     def __init__ (self, connection):
36         # Keep track of the connection to the switch so that we
37         # can
38         # send it messages!
39         self.connection = connection
40
41         # This binds our PacketIn event listener
42         connection.addListeners(self)
43
44         # Use this table to keep track of which ethernet address
45         # is on
46         # which switch port (keys are MACs, values are ports).
47         self.mac_to_port = {}
48
49
50     def resend_packet (self, packet_in, out_port):
51         """
52         Instructs the switch to resend a packet that it had sent
53         to us.
54         "packet_in" is the ofp_packet_in object the switch had
55         sent to the
56         controller due to a table-miss.
57         """
58
59         msg = of.ofp_packet_out()
```

Python ▾ Tab Width: 8 ▾ Ln 127, Col 64 ▾ INS

Questions

1. Describe how the above code works, such as how the "MAC to Port" map is established.

For example when h1 ping h2, in the of_tutorial.py, there is a mac_to_port map that stores mac to port the key-value pairs. When this class is instantiated, the map is empty. When a packet comes

in, the program will check if the source port and source mac are stored in mac_to_port map, if not, it will “learn”, which means to add this source mac-port key-value pair into the mac_to_port map. The program would check if the destination mac has a related port stored in mac_to_port map, if so, it would send to this port, else, send a packet to everyone.

2.(Comment out all prints before doing this experiment) Have h1 ping h2, and h1 ping h8 for 100 times (e.g., h1 ping -c100 p2).

h1 ping -c100 h2:

```
hanlei@hanlei:~$ sudo mn --custom offTutorial.py
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping -c100 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.268 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.269 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.147 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.110 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.264 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.126 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.108 ms
^[$64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.294 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.481 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.127 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.595 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.213 ms
```

The screenshot shows a terminal window with two tabs. The left tab, titled 'root@ee94e39a2799: /pox', displays a continuous stream of ICMP sequence numbers and times. The right tab, titled 'hanlei@hanlei: ~', shows the results of a ping command to 10.0.0.2.

```
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=0.836 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=0.272 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=0.303 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=0.646 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.267 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.240 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.171 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.252 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.275 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.145 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.218 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.183 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.260 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.320 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.128 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.297 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.703 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.239 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.425 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.294 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.319 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.133 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 101153ms
rtt min/avg/max/mdev = 0.053/0.283/2.224/0.262 ms
mininet>
```

h1 ping -c100 h8:

```
mininet> h1 ping -c100 h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=1.87 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.085 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.836 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=0.111 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.376 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.351 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=0.247 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=0.149 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=0.170 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=0.139 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=0.416 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=0.353 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=0.385 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=0.412 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=0.116 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=0.348 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=0.223 ms
64 bytes from 10.0.0.8: icmp_seq=18 ttl=64 time=0.128 ms
64 bytes from 10.0.0.8: icmp_seq=19 ttl=64 time=0.269 ms
64 bytes from 10.0.0.8: icmp_seq=20 ttl=64 time=0.080 ms
64 bytes from 10.0.0.8: icmp_seq=21 ttl=64 time=0.270 ms
64 bytes from 10.0.0.8: icmp_seq=22 ttl=64 time=0.143 ms
64 bytes from 10.0.0.8: icmp_seq=23 ttl=64 time=0.303 ms
64 bytes from 10.0.0.8: icmp_seq=24 ttl=64 time=0.175 ms
64 bytes from 10.0.0.8: icmp_seq=25 ttl=64 time=0.974 ms
64 bytes from 10.0.0.8: icmp_seq=26 ttl=64 time=0.233 ms
64 bytes from 10.0.0.8: icmp_seq=27 ttl=64 time=0.315 ms
64 bytes from 10.0.0.8: icmp_seq=28 ttl=64 time=0.124 ms
64 bytes from 10.0.0.8: icmp_seq=29 ttl=64 time=0.336 ms
64 bytes from 10.0.0.8: icmp_seq=30 ttl=64 time=0.402 ms
64 bytes from 10.0.0.8: icmp_seq=31 ttl=64 time=0.128 ms
64 bytes from 10.0.0.8: icmp_seq=32 ttl=64 time=0.138 ms
64 bytes from 10.0.0.8: icmp_seq=33 ttl=64 time=0.112 ms
```

```
64 bytes from 10.0.0.8: icmp_seq=79 ttl=64 time=0.141 ms
64 bytes from 10.0.0.8: icmp_seq=80 ttl=64 time=0.397 ms
64 bytes from 10.0.0.8: icmp_seq=81 ttl=64 time=0.341 ms
64 bytes from 10.0.0.8: icmp_seq=82 ttl=64 time=0.207 ms
64 bytes from 10.0.0.8: icmp_seq=83 ttl=64 time=0.230 ms
64 bytes from 10.0.0.8: icmp_seq=84 ttl=64 time=0.151 ms
64 bytes from 10.0.0.8: icmp_seq=85 ttl=64 time=0.159 ms
64 bytes from 10.0.0.8: icmp_seq=86 ttl=64 time=0.230 ms
64 bytes from 10.0.0.8: icmp_seq=87 ttl=64 time=0.217 ms
64 bytes from 10.0.0.8: icmp_seq=88 ttl=64 time=0.303 ms
64 bytes from 10.0.0.8: icmp_seq=89 ttl=64 time=0.468 ms
64 bytes from 10.0.0.8: icmp_seq=90 ttl=64 time=0.189 ms
64 bytes from 10.0.0.8: icmp_seq=91 ttl=64 time=1.00 ms
64 bytes from 10.0.0.8: icmp_seq=92 ttl=64 time=0.267 ms
64 bytes from 10.0.0.8: icmp_seq=93 ttl=64 time=1.50 ms
64 bytes from 10.0.0.8: icmp_seq=94 ttl=64 time=0.319 ms
64 bytes from 10.0.0.8: icmp_seq=95 ttl=64 time=0.413 ms
64 bytes from 10.0.0.8: icmp_seq=96 ttl=64 time=0.197 ms
64 bytes from 10.0.0.8: icmp_seq=97 ttl=64 time=0.368 ms
64 bytes from 10.0.0.8: icmp_seq=98 ttl=64 time=0.317 ms
64 bytes from 10.0.0.8: icmp_seq=99 ttl=64 time=0.204 ms
64 bytes from 10.0.0.8: icmp_seq=100 ttl=64 time=0.115 ms

--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 101200ms
rtt min/avg/max/mdev = 0.057/0.291/1.873/0.278 ms
```

a.How long did it take (on average) to ping for each case?

For h1 ping h2, the average time is 0.283 ms.

For h1 ping h8, the average time is 0.291 ms.

b.What is the minimum and maximum ping you have observed?

For h1 ping h2, the minimum time is 0.032 ms, the maximum time is 2.224 ms.

For h1 ping h8, the minimum time is 0.057 ms, the maximum time is 1.873 ms.

c.Any difference from Task 2 and why do you think there is a change if there is?

Compared to Task2, the average time for h1 ping h2 and h1 ping h8 is relatively longer with the off-tutorial. Because in Task 3 it takes more time to check the mac_to_port map.

3.Run “iperf h1 h2” and “iperf h1 h8”.

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['107 Gbits/sec', '108 Gbits/sec']
```

```
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['84.7 Gbits/sec', '84.9 Gbits/sec']
mininet>
```

a.What is the throughput for each case?

For h1 - h2: 107 Mbits/sec

For h2 - h1: 108 Mbits/sec

For h1 - h8: 84.7 Mbits/sec

For h8 - h1: 84.9 Mbits/sec

b.What is the difference from Task 2 and why do you think there is a change if there is?

For h1-h2, the throughput is larger when the destination port is known by the sender. Thus more data can be transferred within a given amount of time.

For h1-h8, there isn't a large change, it could because the network path still consists of five switches so that the map may not have a big influence on the “iperf” measurement.