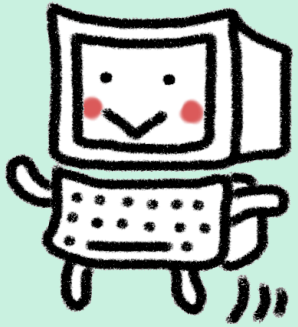


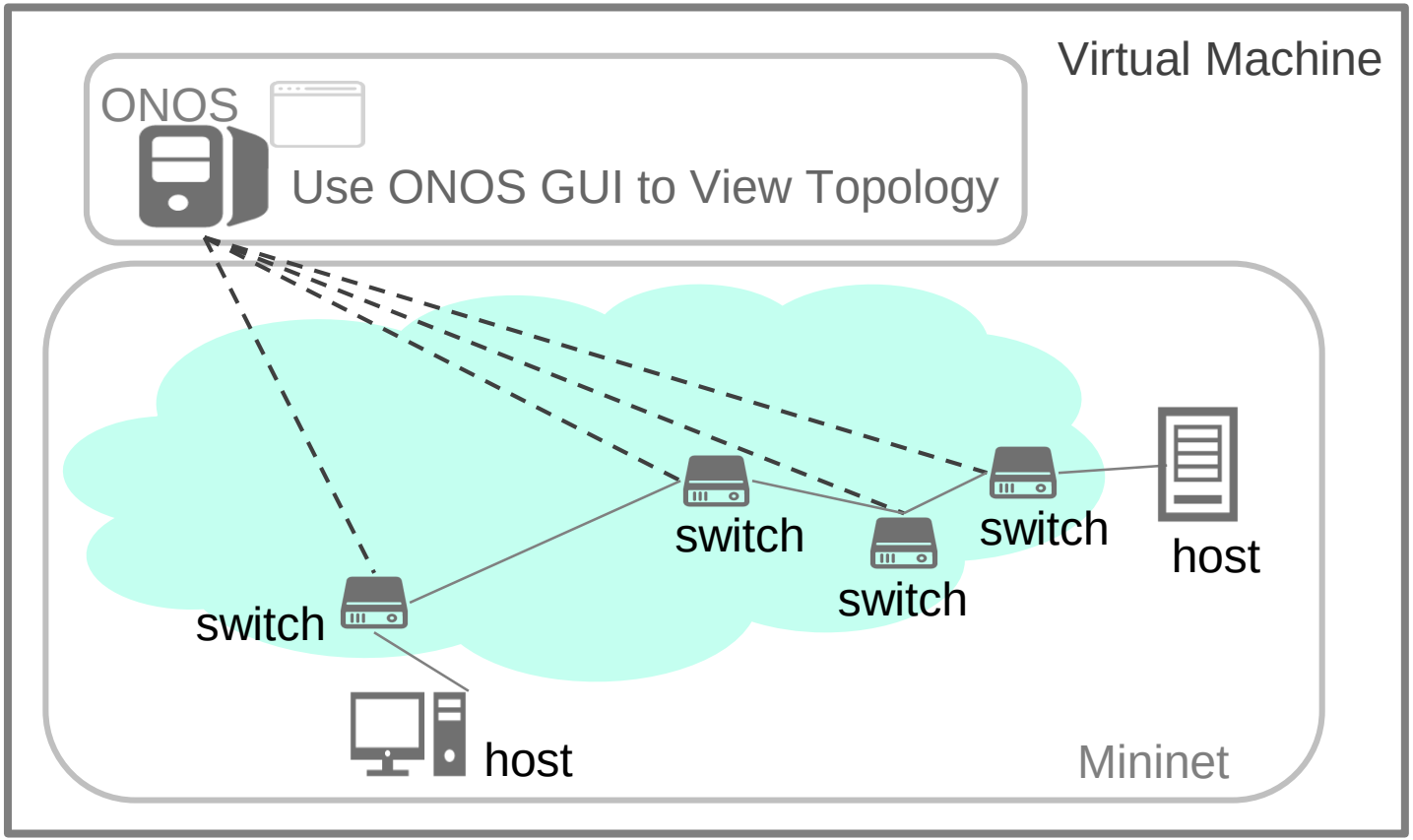
Lab 1
Environment Setup and Basic Operation

ONOS and Mininet Installation

Deadline: 2021/10/06 (Wed) 23:59

TA: 陳志祥

Overview


Virtual Machine
ONOS
Use ONOS GUI to View Topology
switch
switch
switch
host
switch
host
Mininet

Outline

- **Environment Setup**
  - VirtualBox Installation
  - Bazel, ONOS, Mininet and OVS Installation
- **Basic Operation**
  - Start ONOS
  - Activate basic ONOS APPs
  - Create a topology controlled by ONOS
- **Lab Requirements**
  - Part 1: Answer Questions
  - Part 2: Write a Custom Topology
  - Part 3: Statically Assign Hosts IP Address In Mininet

VirtualBox & ONOS & Mininet

Environment Setup

Outline

- Environment Setup
  - VirtualBox Installation
  - Bazel, ONOS, Mininet and OVS Installation
- Basic Operation
- Lab Requirements

Virtualbox Installation

- Oracle VM VirtualBox:
  - ✔ a free and open-source hosted hypervisor
  - ✔ developed by Oracle Corporation
- Environment
  - **Ubuntu Desktop 16.04**
  - **Min** Hardware settings
    - **2 Cores**
    - **8GB RAM**
    - **20GB HDD**
- For more installation detail, please refer to:
  - **SDN_Environment_Setup.pdf**

Outline

- Environment Setup
  - VirtualBox Installation
  - Bazel, ONOS, Mininet and OVS Installation
- Basic Operation
- Lab Requirements

Bazel, ONOS, Mininet and OVS Installation

- **Bazel**: free SW tool for “automation of building and testing of SW”.
- **Mininet**: An instant virtual network on your computer
- **O**pen **vS**witch **(OVS)**: a multilayer software switch licensed under the open source Apache 2 license.
- **O**pen **N**etwork **O**perating **S**ystem **(ONOS)**: open source network controller for SDN

- Installation:
  - Use TA-provided ***env_setup.sh***

Bazel (Fast, Correct) - Choose two

OvS Open vSwitch

onos Open Network Operating System

Outline

- Environment Setup
- Basic Operation
  - Start ONOS
    - ONOS CLI
    - ONOS Web GUI
  - Activate basic ONOS APPs
    - Method 1: Via ONOS CLI
    - Method 2: Via ONOS GUI
  - Create a topology controlled by ONOS
    - Method 1: Built-in Topology
    - Method 2: Custom Topology
- Lab Requirements

Start ONOS

- Start ONOS in localhost

```
demo@SDN-NFV:~$ cd $ONOS_ROOT
demo@SDN-NFV:~/onos$ bazel run onos-local -- clean debug
      # option 'clean' to delete all previous running status
      # option 'debug' to enable remote debugging (port 5005)
```

```
demo@SDN-NFV:~/onos$ bazel run onos-local -- clean debug
INFO: Analyzed target //:onos-local (0 packages loaded, 0 targets configured).
INFO: Found 1 target...
Target //:onos-local_current-jdk up-to-date:
  bazel-bin/onos-runner_current-jdk
INFO: Elapsed time: 0.486s, Critical Path: 0.00s
INFO: 0 processes.
INFO: Build completed successfully, 1 total action
INFO: Build completed successfully, 1 total action
Killing ONOS server...
Using JDK in /tmp/onos-2.2.0-jdk...
Running clean installation...
Host [localhost]:8101 not found in /home/demo/.ssh/known_hosts
Creating local cluster configs for IP 127.0.0.1...
Waiting for karaf.log
Mar 01, 2020 5:21:31 PM org.apache.karaf.main.Main launch
```

```
ConfigurationEvent: pid=org.onosproject.net.intent.impl.IntentCleanup) | OpenFlowRuleProvider          | 203 - org.onosproject.onos-provid

ConfigurationEvent: pid=org.onosproject.net.intent.impl.IntentCleanup) | OpenFlowRuleProvider          | 203 - org.onosproject.onos-provid
se
tomixClusterStore              | 192 - org.onosproject.onos-core-primitives - 2.2.0 | Updated node 127.0.0.1 state to READY
```

ONOS CLI

- Bring up another new terminal and enter ONOS CLI

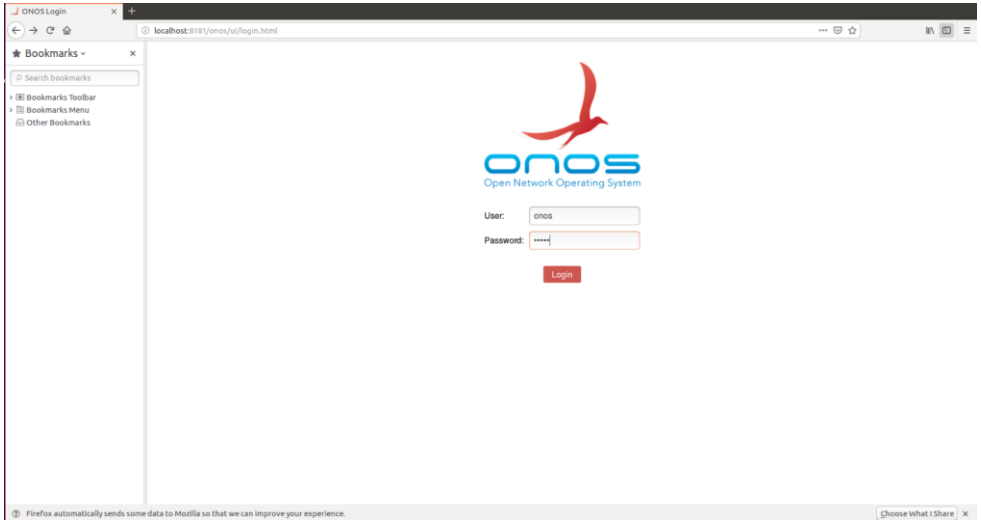
```
demo@SDN-NFV:~/onos$ onos localhost
```

```
demo@SDN-NFV:~/onos$ tools/test/bin/onos localhost
Welcome to Open Network Operating System (ONOS)!
    ____  _  ______  ____
   / __ \/ |/ / __ \/ __/
  / /_/ /    / /_/ /\ \
  \____/_/|_/\____/___/

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org

Come help out! Find out how at: contribute.onosproject.org

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'logout' to exit ONOS session.

demo@root >
```

```
demo@root > logout        # exit onos cli
```

ONOS Web GUI

- Open web browser (e.g. Firefox)
  - visit http://localhost:**8181**/onos/ui
    - User/Password: onos/rocks



- ONOS GUI tutorial
  - https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial#BasicONOSTutorial-ONOSGraphicalUserInterface

Outline

- Environment Setup
- Basic Operation
  - Start ONOS
    - ONOS CLI
    - ONOS Web GUI
  - Activate basic ONOS APPs
    - Method 1: Via ONOS CLI
    - Method 2: Via ONOS GUI
  - Create a topology controlled by ONOS
    - Method 1: Built-in Topology
    - Method 2: Custom Topology
- Lab Requirements

Activate basic ONOS APPs (Via ONOS CLI)

- Via ONOS CLI

```
onos> apps -a -s                    # Show activated apps only
```

```
demo@root > apps -a -s
*   9 org.onosproject.optical-model       2.2.0    Optical Network Model
*  10 org.onosproject.drivers             2.2.0    Default Drivers
*  90 org.onosproject.hostprovider        2.2.0    Host Location Provider
*  91 org.onosproject.lldpprovider        2.2.0    LLDP Link Provider
*  92 org.onosproject.openflow-base       2.2.0    OpenFlow Base Provider
*  93 org.onosproject.openflow            2.2.0    OpenFlow Provider Suite
* 103 org.onosproject.gui2                2.2.0    ONOS GUI2
```

Initially activated Apps

```
onos> app activate <name>           # activate onos app
onos> app deactivate <name>         # deactivate onos app
```

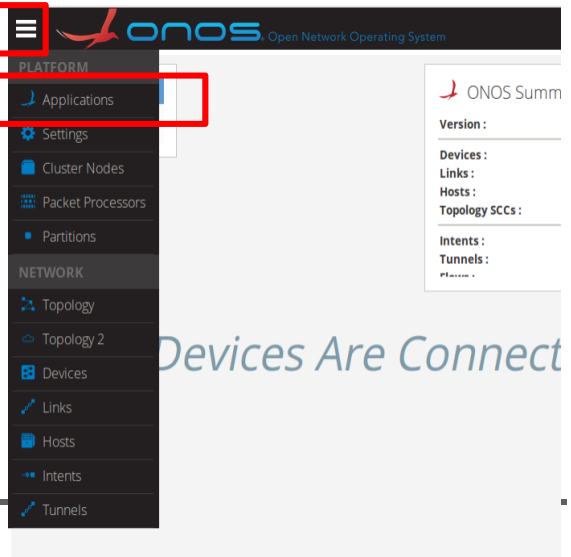
```
demo@root > app activate org.onosproject.openflow
Activated org.onosproject.openflow
demo@root > app activate org.onosproject.fwd
Activated org.onosproject.fwd
```

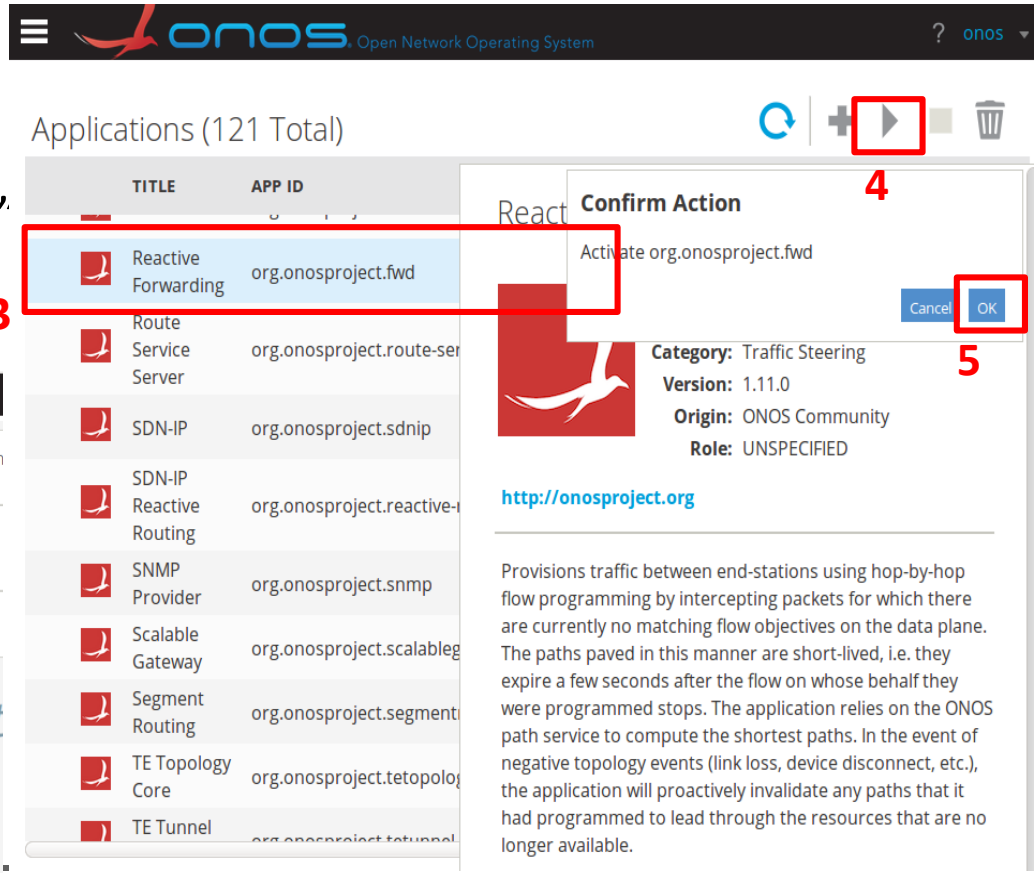
- Note: Use the following command to get more usage information

```
onos> app --help                    # display command help message
```

Activate basic ONOS APPs (Via ONOS GUI)

- Via ONOS GUI
  1. Click
  2. Choose “Applications”
  3. Choose “Reactive Forwarding”
  4. Click
  5. Click “OK”



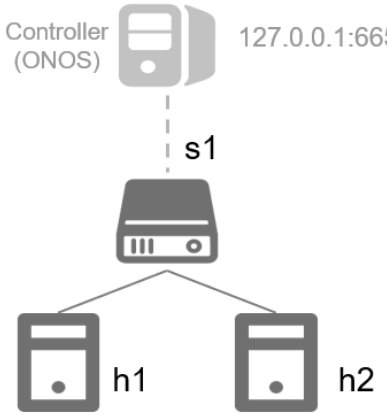


Outline

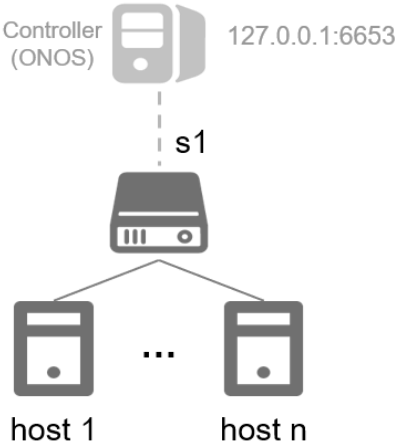
- Environment Setup
  - VirtualBox Installation
  - Bazel, ONOS, Mininet and OVS Installation
- Basic Operation
  - Start ONOS
  - Activate basic ONOS APPs
  - Create a topology in Mininet controlled by ONOS
    - Method 1: Built-in Topology
    - Method 2: Custom Topology
- Lab requirements

Built-in topologies in Mininet

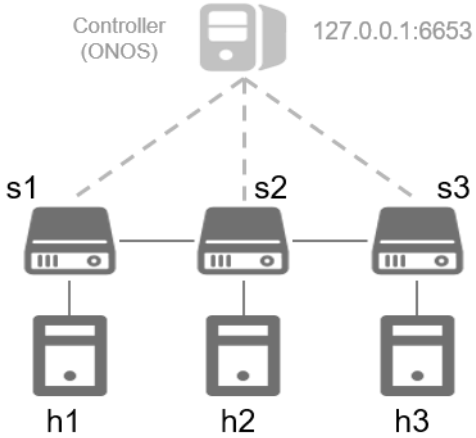
- Five Built-in topologies:
  - Minimal
    - Also called “Default”
  - Single
  - Linear
  - Torus
  - Tree



▲ Minimal



▲ Single



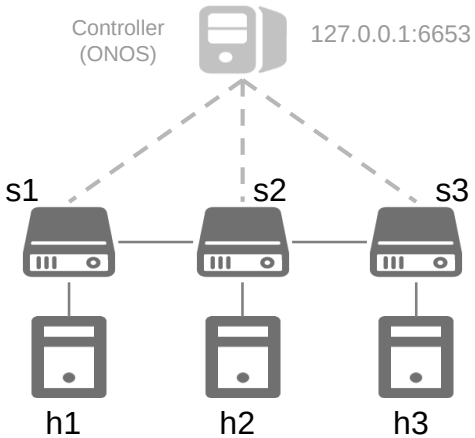
▲ Linear

Create Built-in Linear Topology

- Create a linear topology

```
$ sudo mn --topo=linear,3 --controller=remote,127.0.0.1:6653
```

  - "--controller" adds remote controller
  - "--topo" specifies the topology




```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s2, s1) (s3, s2)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet>
```

- Exit mininet

```
mininet> exit      # exit mininet cli
```

Clear Your Experiment Environment

- Note:

  Make sure to clean up the environment of Mininet after every time you exit Mininet CLI

```
$ sudo mn -c          #clean and exit
```

  - A "cleanup" command to get rid of junk (interfaces, processes, files in /tmp, etc.) which might be left around by Mininet or Linux

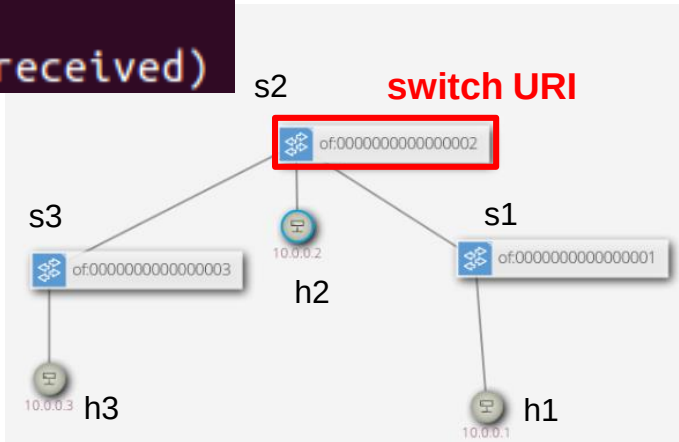
Make hosts appear in ONOS GUI

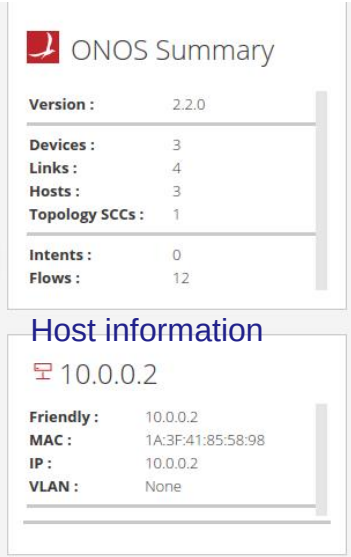
1. First, use “pingall” in the Mininet CLI

```
mininet> pingall    # ping between all hosts
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
```

Note: Remember to activate org.onosproject.fwd

2. Hotkeys on GUI
   - “h” to show hosts
   - “l” to show switch URI





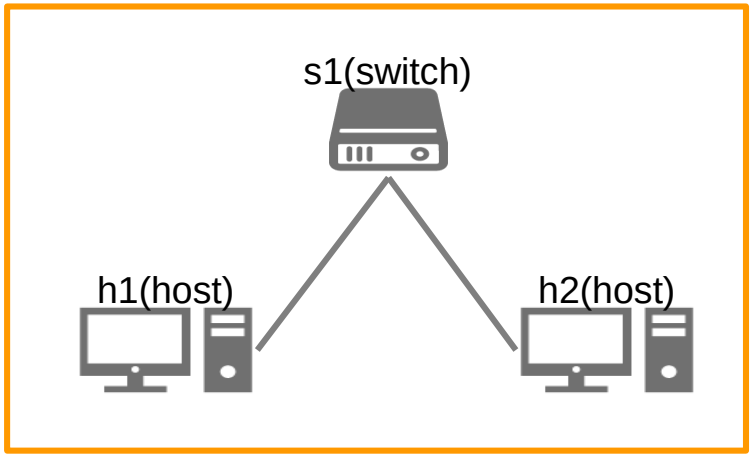
Outline

- Environment Setup
  - VirtualBox Installation
  - Bazel, ONOS, Mininet and OVS Installation
- Basic Operation
  - Activate basic ONOS APPs
  - Create a topology in Mininet controlled by ONOS
    - Method 1: Built-in Topology
    - Method 2: Custom Topology
- Lab requirements

Create a custom topology

1. Specify topology with Python script

E.g., Single switch with two hosts



sample.py

```
from mininet.topo import Topo

class MyTopo( Topo ):
    def __init__( self ):
        Topo.__init__( self )

        # Add hosts
        h1 = self.addHost( 'h1' )
        h2 = self.addHost( 'h2' )

        # Add switches
        s1 = self.addSwitch( 's1' )

        # Add links
        self.addLink( h1, s1 )
        self.addLink( h2, s1 )

topos = { 'mytopo': MyTopo }
```

2. Run Mininet with options “custom” and “topo”

```
$ sudo mn --custom=sample.py --topo=mytopo \
  --controller=remote,ip=127.0.0.1,port=6653
```

Topology Dictionary

- Recall: create a custom topology specified in sameple.py

```
$ sudo mn --custom=sample.py --topo=mytopo \
  --controller=remote,ip=127.0.0.1,port=6653
```

- Need to define a topology dictionary in the sample.py

```
from mininet.topo import Topo

class MyTopo( Topo ):
    def __init__( self ):
        Topo.__init__( self )

        # Add hosts
        h1 = self.addHost( 'h1' )
        h2 = self.addHost( 'h2' )

        # Add switches
        s1 = self.addSwitch( 's1' )

        # Add links
        self.addLink( h1, s1 )
        self.addLink( h2, s1 )

topos = { 'mytopo': MyTopo }
```

```
topos = { 'mytopo': MyTopo }
```

(Key) Topo Name	(Value) Topo Constructor
mytopo	MyTopo

Option	Topo Name
--topo	mytopo

- Topo Name can be passed to option “--topo”
- Topo Constructor may be subclasses, constructors or functions

Outline

- Environment Setup
- Basic Operation
- Project Requirements
  - Part 1: Answer Questions (40%)
  - Part 2: Create a Custom Topology (50%)
  - Part 3: Statically Assign Hosts IP Address In Mininet (10%)

Part 1: Answer Questions

Activate ONOS APPs

1. When ONOS activates **"org.onosproject.openflow,"** what are the APPs which it also activates?
2. As topology in p.22, can H1 ping H2 successfully? Why or why not?

Hint: Please refer to the reference "Basic ONOS Tutorial" attached at the end of slide

Observe listening port

3. Which TCP port the controller listens for the OpenFlow connection request from the switch?
4. In question 3, which APP enables the controller to listen on the TCP port?

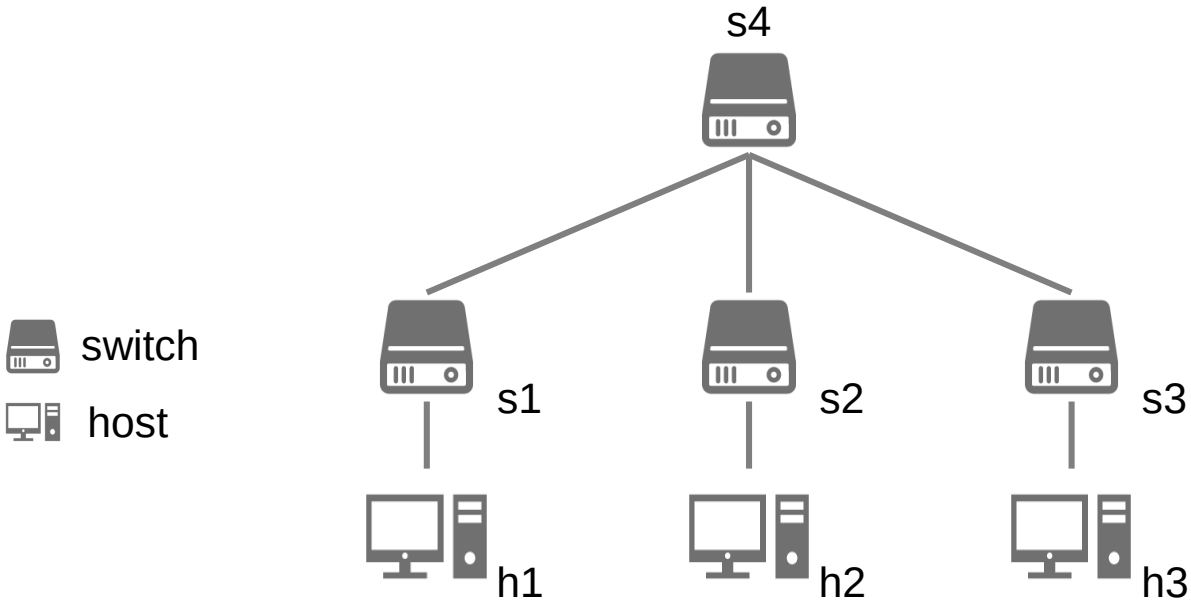
Hint: Observation of network connection

1. *bring up and enter a new terminal*
2. *deactivate/activate apps and use "netstat" in the new terminal to observe network connections*

```
$ netstat -nlpt                    #show only listening TCP sockets
```

Part 2: Create a Custom Topology

- Write a Python script to build the following topology:



- Hand in the Python script you write in this part

Naming Conventions for part 2

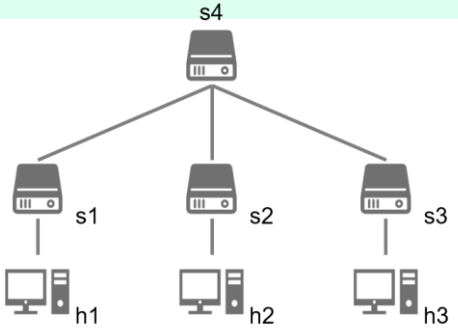
- Naming conventions in your python script
  a. Name of Python script: project1_part2_<studentID>.py
  b. Name of topology class: Project1_Topo_<studentID>
  c. Name of dictionary's key: topo_part2_<studentID>

➢ Command to execute your script:

```
$ sudo mn --custom=project1_part2_<studentID>.py \
  --topo=topo_part2_<studentID> \
  --controller=remote,ip=127.0.0.1:6653
```

Part 3: Statically Assign Hosts IP Address In Mininet (I)

- Reuse the topology in part 2



- Mininet automatically assigns an IP address and a subnet mask to each interface of each host
  - E.g., 10.0.0.1/8, 10.0.0.2/8, 10.0.0.3/8

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=11188>
<Host h2: h2-eth0:10.0.0.2 pid=11190>
```

```
mininet> h1 ifconfig
h1-eth0   Link encap:Ethernet  HWaddr ae:c2:c4:b8:d3:ac
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::acc2:c4ff:feb8:d3ac/64 Scope:Link
```

Part 3: Statically Assign Hosts IP Address In Mininet (II)

- Format for manual assignment of host IP address:
  - **192.168.0.<host_number>**
  - netmask **255.255.255.224**

Host	IP Address
h1	192.168.0.1
h2	192.168.0.2
…	…

- Take screenshots of the result of the Mininet command “dump” and “pingall”

```
mininet> dump                    # dump all the node info
 … (result) …
mininet> pingall                 # ping between all hosts
 … (result) …
```

- Hand in the Python script you’ve edited
- Remember to activate “**org.onosproject.fwd**” before “pingall”

Naming Conventions for part 3

- Naming conventions in your python script
  a. Name of Python script: project1_part3_<studentID>.py
  b. Name of topology class: Project1_Topo_<studentID>
  c. Name of dictionary's key: topo_part3_<studentID>

➢ Command to execute your script:

```
$ sudo mn --custom=project1_part3_<studentID>.py \
  --topo=topo_part3_<studentID> \
  --controller=remote,ip=127.0.0.1:6653
```

Turn-in Reports & Python scripts

Report Submission

Report Submission

- Files
  - Two Python scripts:
    - **project1_part2_<studentID>.py**
    - **project1_part3_<studentID>.py**
  - A report: **project1_<studentID>.pdf**
    1. Part 1: Answers to those four questions
    2. Part 2: Take screenshots and explain what you've done
    3. Part 3: Take screenshots and explain what you've done
    4. What you've learned or solved
- Submission
  - Zip Python scripts and the report into a zip file
    - Named: **project1_<studentID>.zip**
  - Incorrect file name or format will result in 10 points deduction

Q & A

Thank you

Reference

1. Basic ONOS Tutorial
   - https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial
2. Introduction to Mininet
   - https://github.com/mininet/mininet/wiki/Introduction-to-Mininet
3. Mininet Python API
   - http://mininet.org/api/annotated.html
4. Manpage for Linux command
   - netstat
     - http://manpages.ubuntu.com/manpages/trusty/man8/netstat.8.html
   - mn
     - http://manpages.ubuntu.com/manpages/bionic/man1/mn.1.html

Appendix: Network Topology for Mininet Emulation

- Mininet employs lightweight virtualization features in the Linux kernel, including process groups, CPU bandwidth isolation, and network namespaces
- An emulated host in Mininet is a group of user-level processes moved into a network namespace