

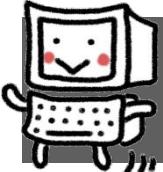
# Project 1

## ONOS and Mininet Installation

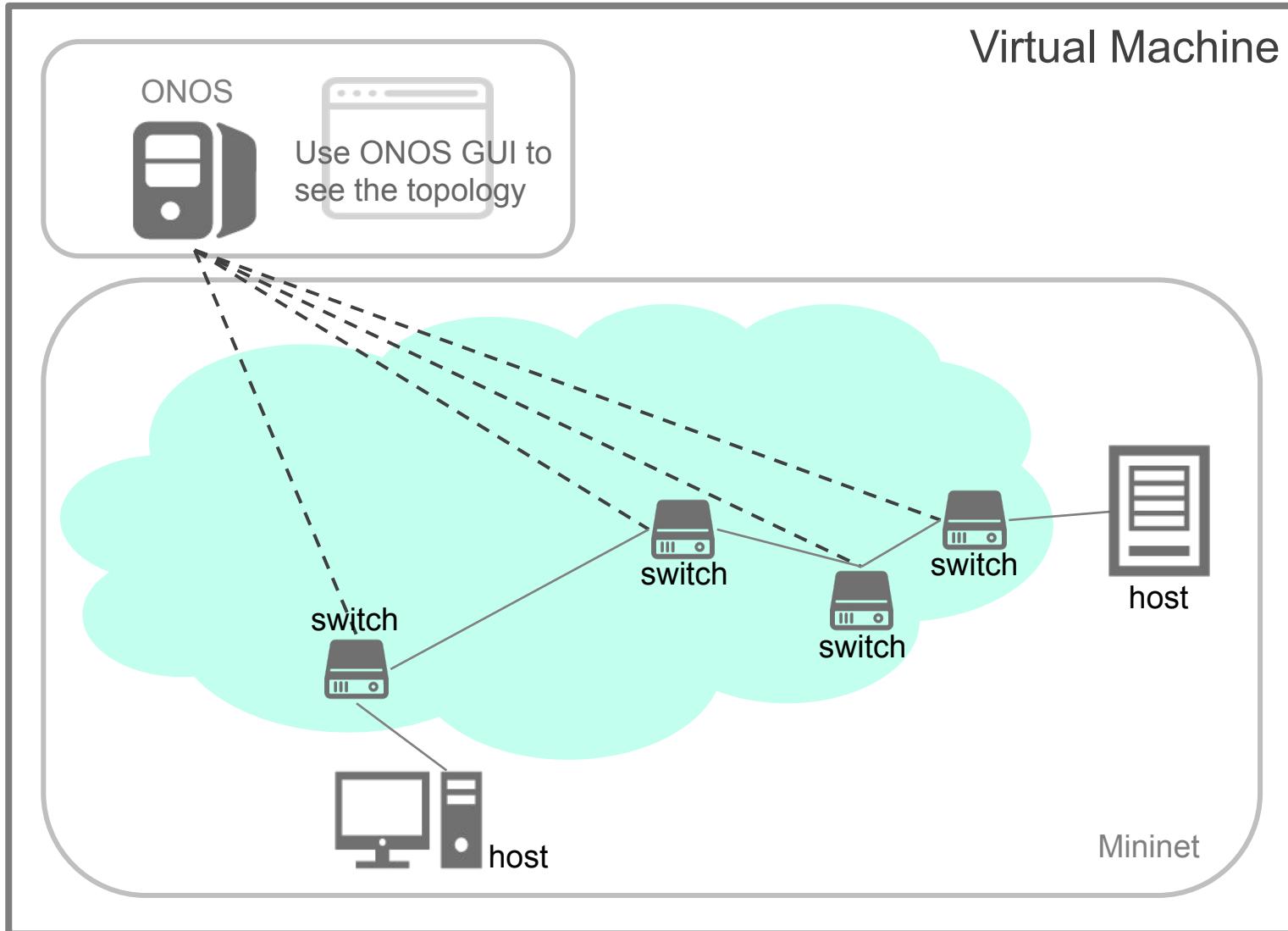
*Environment Setup &  
Basic Operation*

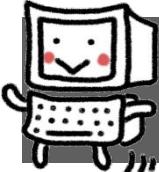
Date : 2019/03/02 (Mon.) 18:30

Deadline: 2019/03/15 (Sun.) 23:59



# Overview





# Outline

## □ Environment Setup

- VirtualBox Installation
- Bazel, ONOS, Mininet and OVS Installation

## □ 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] Via Mininet CLI
  - [Method 2] Custom Topology

## □ Project Requirements

- Part 1: Answer Questions
- Part 2: Write a Custom Topology
- Bonus: Specify Hosts IP Address In Mininet



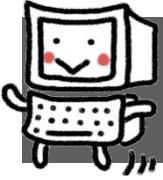
# Outline

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



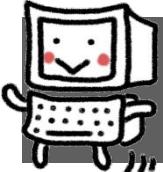
**VirtualBox & ONOS & Mininet**

# **Environment Setup**



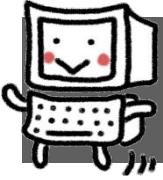
# Outline

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



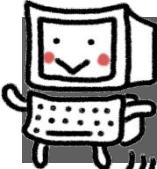
# Virtualbox Installation

- ❑ Open-source & cross-platform virtualization application
  - *Ubuntu Desktop 16.04 (or 18.04) recommended*
  - **Minimum Hardware settings**
    - 2 Cores
    - 8GB RAM
    - 20GB HDD
- ❑ For more installation detail, please refers to:
  - **SDN\_Environment\_Setup.pdf**



# Outline

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

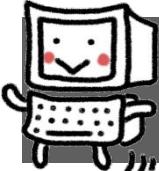


# Bazel, ONOS, Mininet and OVS Installation

- **Bazel**: A free SW tool used to automatically build and test SWs
  - <https://bazel.build/>
- **Mininet**: An instant virtual network on your computer
  - <http://mininet.org/>
- **OVS (Open vSwitch)**: A virtual switch can be used to cut network domains, monitor QoS traffic while supporting OpenFlow protocols
  - <https://www.openvswitch.org/>
- **ONOS (Open Network Operating System)**: The controller of SDN network
  - <https://www.opennetworking.org/onos/>



- Installation:
  - Use TA-provided *env\_setup.sh*



# Outline

## □ Environment Setup

## □ Basic Operation

### ■ Start ONOS

- ONOS CLI
- ONOS Web GUI

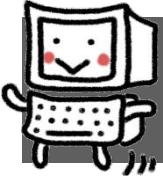
### ■ Activate basic ONOS APPs

- [Method 1] Via ONOS CLI
- [Method 2] Via ONOS GUI

### ■ Create a topology controlled by ONOS

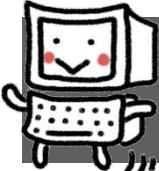
- [Method 1] Via Mininet CLI
- [Method 2] Custom Topology

## □ Project Requirements



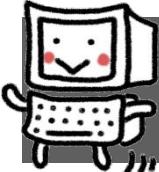
**ONOS APPs & Custom Topo**

# **Basic Operation**



# 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] Via Mininet CLI
    - [Method 2] Custom Topology
- ❑ Project Requirements



# Start ONOS

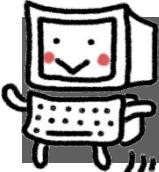
## □ Start ONOS server in localhost

```
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
```

## □ The INFO “Updated node 127.0.0.1 state to READY” means that ONOS is successfully initiated

```
[query@... ~] r (Fire ConfigurationEvent: pid=org.onosproject.net.intent.impl.IntentCleanup) | OpenFlowRuleProvider | 203 - org.onosproject.onos-providers
=5
r (Fire ConfigurationEvent: pid=org.onosproject.net.intent.impl.IntentCleanup) | OpenFlowRuleProvider | 203 - org.onosproject.onos-providers
ing=false
nts | AtomixClusterStore | 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:~$ cd $ONOS_ROOT
demo@SDN-NFV:~/onos$ tools/test/bin/onos localhost
```

```
demo@SDN-NFV:~/onos$ tools/test/bin/onos localhost
Welcome to Open Network Operating System (ONOS)!


$$\begin{array}{c} \text{O} \text{Y} \text{V} \text{I} \text{N} \text{O} \text{X} \text{P} \\ \text{O} \text{Y} \text{V} \text{I} \text{N} \text{O} \text{X} \text{P} \\ \text{O} \text{Y} \text{V} \text{I} \text{N} \text{O} \text{X} \text{P} \end{array}$$


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

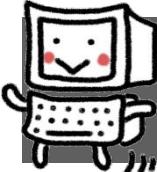
Come help out! Find out how at: contribute.onosproject.org

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 > 
```

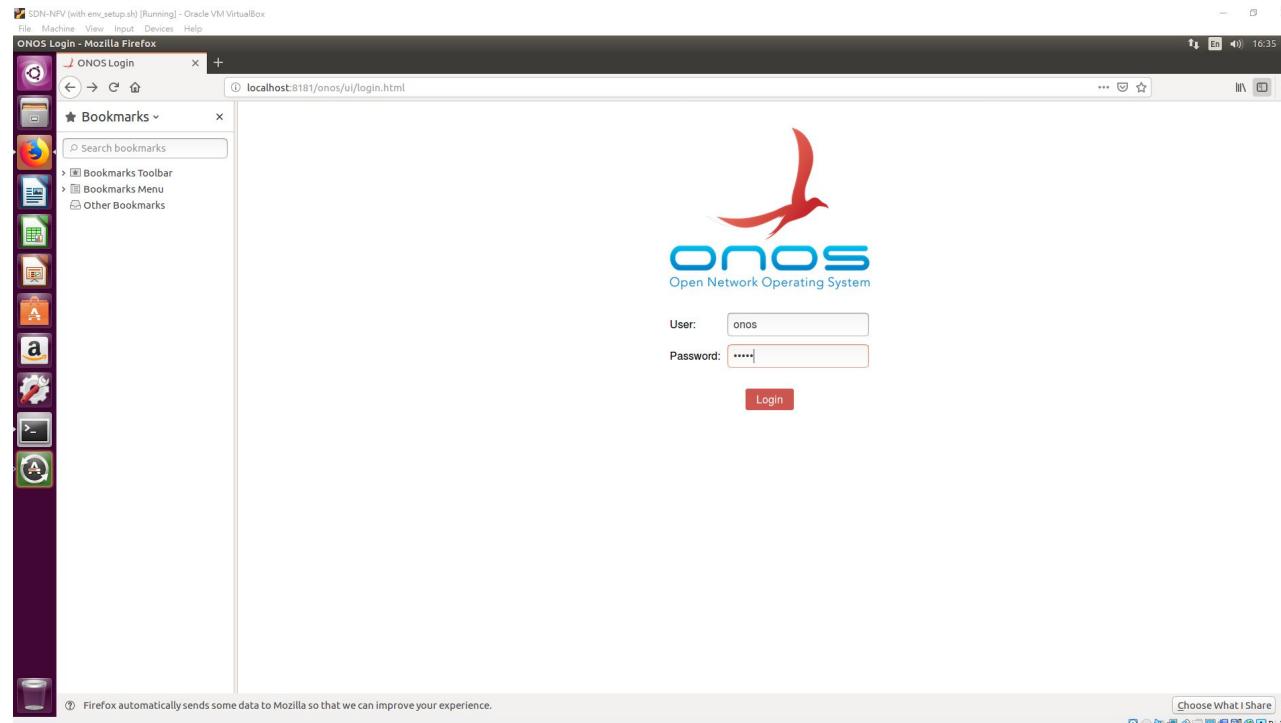
- ONOS command “logout” to exit ONOS CLI

```
demo@root > logout
```

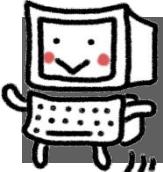


# ONOS Web GUI

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

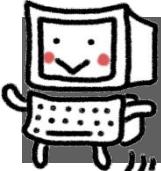


- ONOS GUI (Graphical User Interface) tutorial
  - <https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial#BasicONOSTutorial-ONOSGraphicalUserInterface>



# 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] Via Mininet CLI
    - [Method 2] Custom Topology
- ❑ Project Requirements



# Activate basic ONOS APPs (Via ONOS CLI)

## □ [Method 1]

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

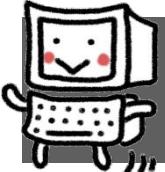
```
demo@root > apps -a -s
* 9 org.onosproject.optical-model      2.2.0    Optical Network Model
* 10 org.onosproject.drivers          2.2.0    Default Drivers
* 90 org.onosproject.hostprovider    2.2.0    Host Location Provider
* 91 org.onosproject.lldpprovider    2.2.0    LLDP Link Provider
* 92 org.onosproject.openflow-base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

```
onos> app activate <name>
```

```
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)

## ❑ [Method 2]

1. Click
2. Choose “Applications”
3. Choose “Reactive Forwarding”
4. Click
5. Click “OK”

1 2 3 4 5

Devices Are Connected

ONOS Summary	Version : 1.11
Devices : 0	
Links : 0	
Hosts : 0	
Topology SCCs : 0	
Intents : 0	
Tunnels : 0	

1 2 3 4 5

Applications (121 Total)

TITLE	APP ID
Reactive Forwarding	org.onosproject.fwd
Route Service Server	org.onosproject.route-server
SDN-IP	org.onosproject.sdnip
SDN-IP Reactive Routing	org.onosproject.reactive-routing
SNMP Provider	org.onosproject.snmp
Scalable Gateway	org.onosproject.scalable-gateway
Segment Routing	org.onosproject.segment-routing
TE Topology Core	org.onosproject.tetopology
TE Tunnel	org.onosproject.tetunnel

React

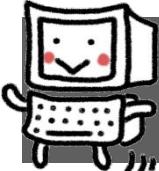
Confirm Action

Activate org.onosproject.fwd

Category: Traffic Steering  
Version: 1.11.0  
Origin: ONOS Community  
Role: UNSPECIFIED

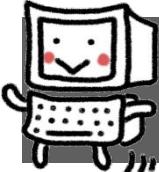
<http://onosproject.org>

Provisions traffic between end-stations using hop-by-hop flow programming by intercepting packets for which there are currently no matching flow objectives on the data plane. The paths paved in this manner are short-lived, i.e. they expire a few seconds after the flow on whose behalf they were programmed stops. The application relies on the ONOS path service to compute the shortest paths. In the event of negative topology events (link loss, device disconnect, etc.), the application will proactively invalidate any paths that it had programmed to lead through the resources that are no longer available.



# 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**
    - [Method 1] Via Mininet CLI
      - Create a topology in Mininet controlled by ONOS
      - Make hosts appear in ONOS GUI
    - [Method 2] Custom Topology
- ❑ Project1 requirements

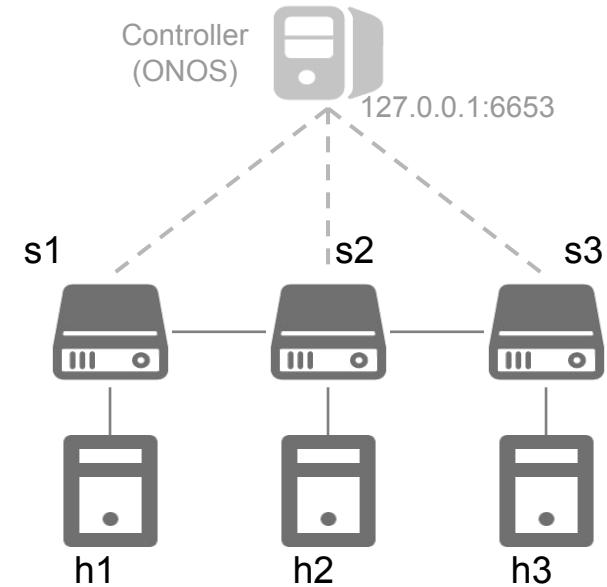


# Create a topology in Mininet controlled by ONOS

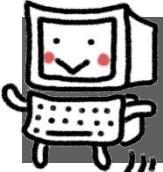
- Run Mininet with “--topo” & “--controller” options

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

```
*** 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> █
```



- In this built-in topology, each switch would have one host
  - “--controller” option adds remote controller
- 
- Use command “exit” in Mininet CLI to exit Mininet CLI



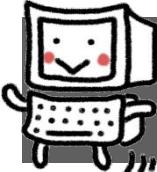
# Create a topology in Mininet controlled by ONOS



Note:

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

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



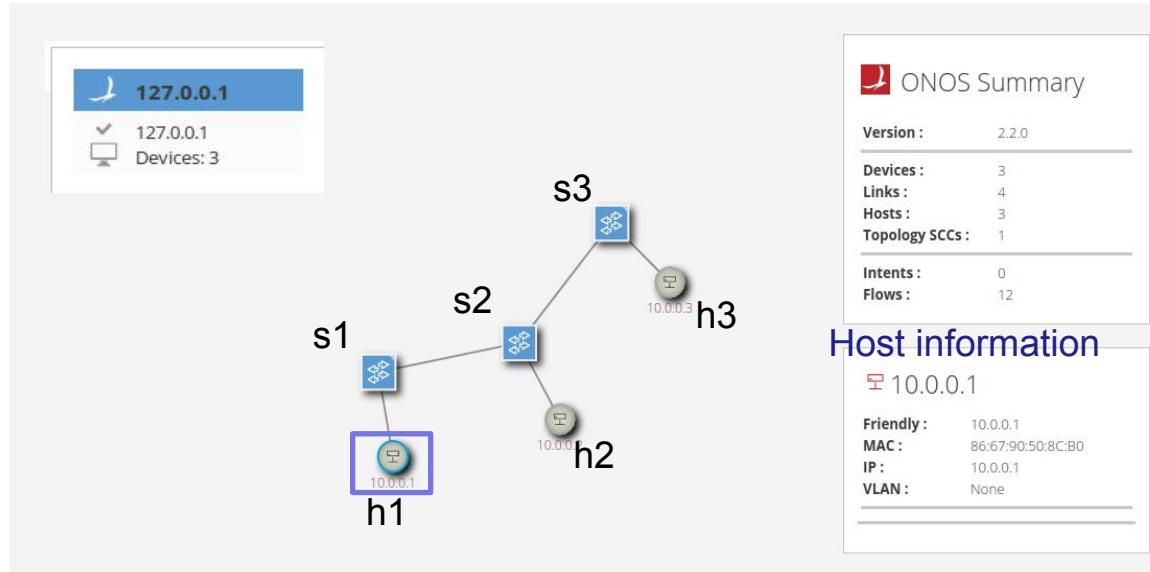
# Make hosts appear in ONOS GUI

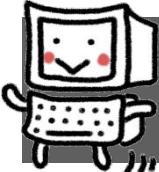
- First, use “pingall” in the Mininet CLI

```
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)
```

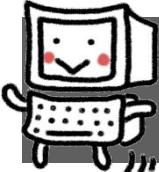
- Press “h” in ONOS GUI to see hosts





# Outline

- ❑ Environment Setup
  - VirtualBox Installation
  - Bazel, ONOS, Mininet and OVS Installation
- ❑ Basic Operation
  - Activate basic ONOS APPs
  - **Create a topology controlled by ONOS**
    - [Method 1] Via Mininet CLI
    - **[Method 2] Custom Topology**
      - Create a custom topology in Mininet
      - Illustration of topology dictionary
- ❑ Project1 requirements



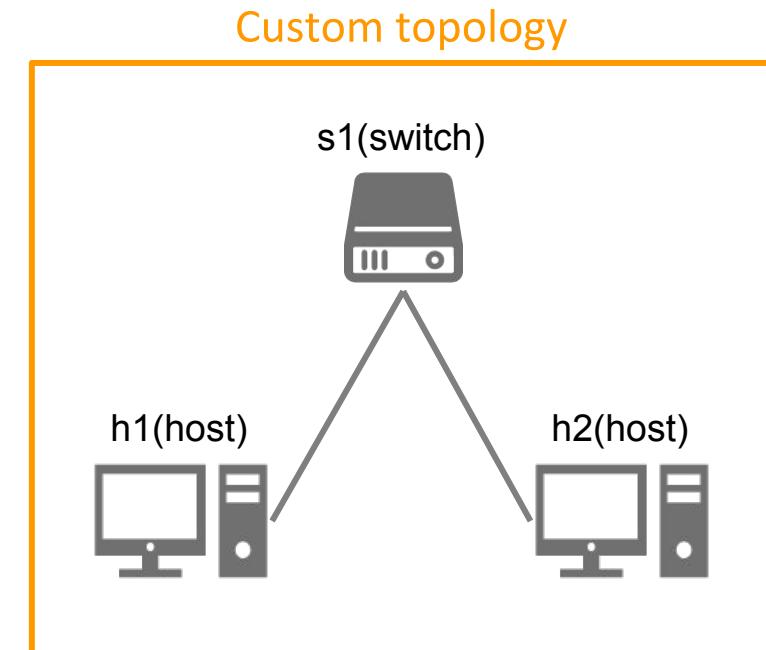
# Create a custom topology in Mininet

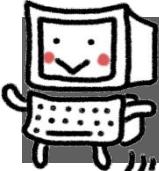
- Specify topology in Python and run the Python script

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

```
1 from mininet.topo import Topo
2
3 class MyTopo(Topo):
4     def __init__(self):
5         Topo.__init__(self)
6
7         # Add hosts
8         h1 = self.addHost("h1")
9         h2 = self.addHost("h2")
10
11        # Add switches
12        s1 = self.addSwitch("s1")
13
14        # Add links
15        self.addLink(h1,s1)
16        self.addLink(h2,s1)
17
18
19 topos = {"mytopo": MyTopo}
```

Sample.py

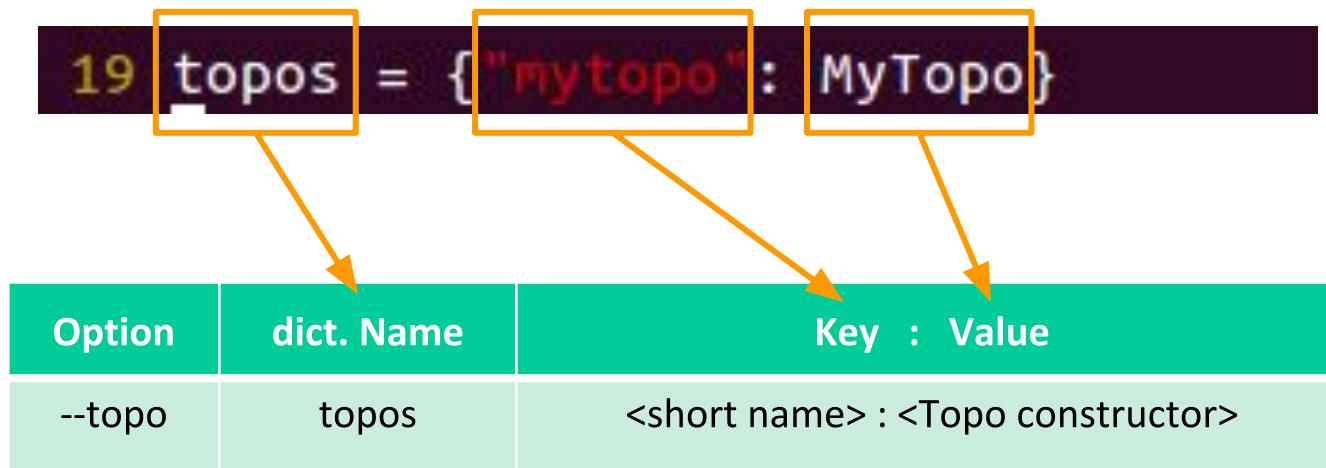




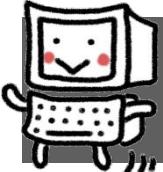
# Illustration of topology dictionary

- ❑ Need to define a dictionary in Python script (i.e. sample.py) of custom topology

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

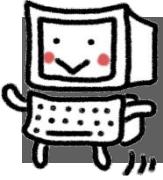


- Key is the short name passed to the appropriate option
- Value may be subclasses, constructors or functions



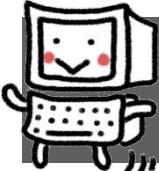
# Outline

- Environment Setup
- Basic Operation
- Project Requirements
  - Part 1: Answer Questions
  - Part 2: Create a Custom Topology
  - Bonus : Specify Hosts IP Address In Mininet



- Part I. Answer Questions (40%)**
- Part II. Create a Custom Topology (60%)**
- Bonus. Specify Hosts IP Address In Mininet (20%)**

# Project Requirement



# Part 1: Answer Questions

## 1. Activate ONOS APPs

- 1-1. When activating “**org.onosproject.openflow**”, what are the APPs which also be activated?
- 1-2. Which **APP** enables hosts to ping each other?

Hint: Please refer to the reference “Basic ONOS Tutorial”[1] attached at the end of slide

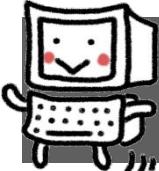
## 2. Observe listening port

- 2-1. Openflow protocol defines the TCP port for connection between controller and switch. What is the **number** of this port?
- 2-2. Regarding to the previous question, which **APP** enables that TCP port be listening?

Hint: Observation of network connection

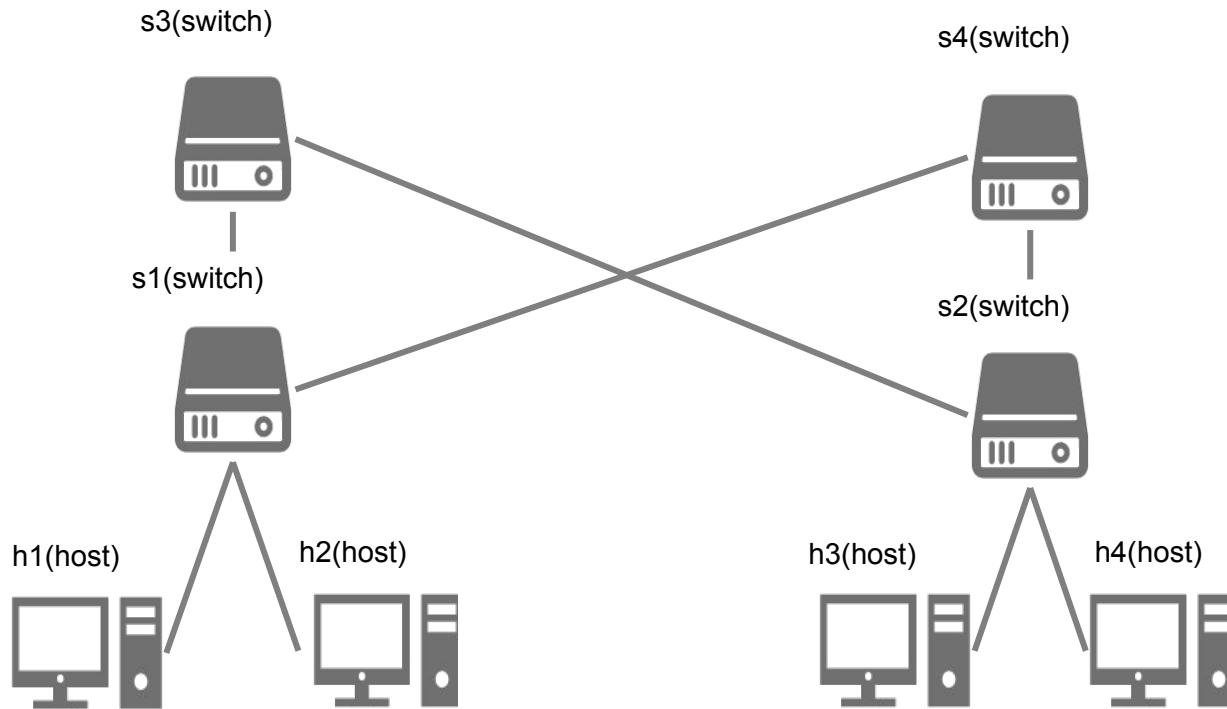
1. bring up a new terminal
2. use the command “netstat” to print network connection

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

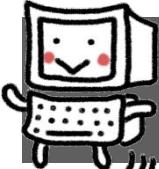


## Part 2: Create a Custom Topology

- Edit a Python script to build the following topology:



- Hand in the Python script you edit in this part

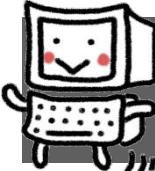


## Part 2: Naming Convention & Command

1. Name of Python script: `project1_<studentID>.py`
2. Name of topology class: `Project1_Topo_<studentID>`
3. Name of dictionary's key: `topo_<studentID>`

- Command to execute your script:

```
$ sudo mn --custom=project1_<studentID>.py \
> --topo=topo_<studentID> \
> --controller=remote,ip=127.0.0.1:6653
```



## Part 2: Example of Naming Convention

- In this case, should use the following command:

```
$ sudo mn --custom=project1_0748787.py \
> --topo=topo_0748787 \
> --controller=remote,ip=127.0.0.1:6653
```

```
from mininet.topo import Topo

class Project1_Topo_0748787( Topo ):
    "Simple topology example.

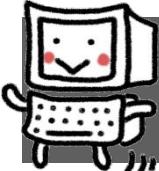
    def __init__( self ):
        "Create custom topo.

        #Initailize topology
        Topo.__init__( self )

        # Add hosts and switches
        # ( To Do )

        # Add links
        # ( To Do )

    topos = { 'topo_0748787': ( lambda: Project1_Topo_0748787() ) }
```



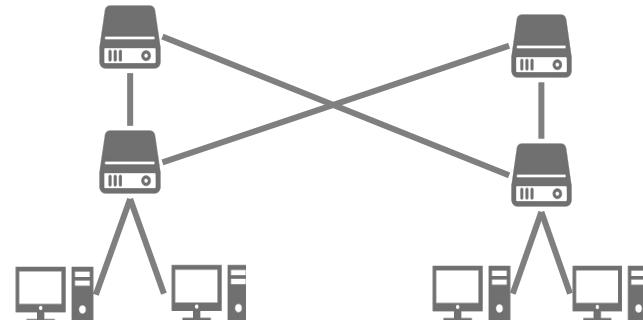
## Bonus: Manually Assign Hosts IP Address In Mininet (I)

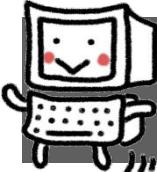
- By default, Mininet will automatically assign an ip address and a subnet mask to each interface of each host  
(i.e. 10.0.0.1/8, 10.0.0.2/8, 10.0.0.3/8)

```
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
```

- Reuse the topology created in part 2





## Bonus: Manually Assign Hosts IP Address In Mininet (II)

- ❑ Manually assign each host's ip address in the following format:
  - **192.168.0.<host\_number>**

Note: host\_number starts from 1

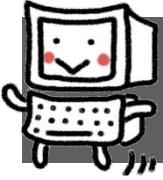
e.g.

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” in Mininet CLI

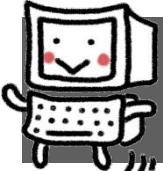
```
mininet> dump          # dump all the node info
... (result) ...
mininet> pingall        # ping between all hosts
... (result) ...
```

- ❑ Hand in the Python script you edit in this part
- ❑ Note: activate “org.onosproject.fwd” in ONOS before “pingall”



**Report & Python script**

# **About Submit**



# Submit to new e3

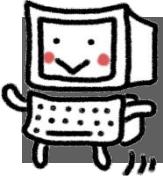
## ❑ Files

- Two Python scripts:
  - **project1\_<studentID>.py** of part 2
  - **bonus\_<studentID>.py** of bonus part
- A report: **project1\_<studentID>.pdf**
  - Part 1: Answers to four questions
  - Part 2: Take screenshots (ONOS GUI and CLI) and explain what you've done
  - Bonus: Take screenshots (ONOS GUI and CLI) and explain what you've done
  - Please also write down what you've learned or solved

## ❑ Submit

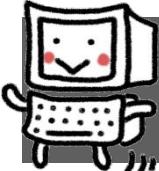
- Zip Python scripts and the report into a zip file
  - Named: **project1\_<studentID>.zip**
- Wrong file name or format would not be scored

## ❑ Feel free to ask questions on the forum at new e3



Q & A

**Thank you**



# Reference

## 1. Basic ONOS Tutorial

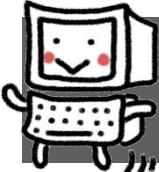
- <https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial>

## 2. Introduction to Mininet

- <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>

## 3. 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

