Section 3: Introduction to Mininet

Simulate a simple network using Mininet

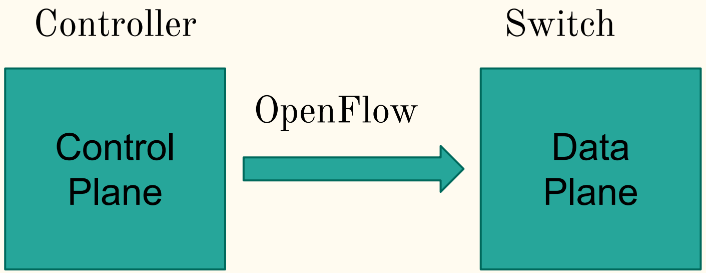
Before we start...

1. Download Vagrant: https://www.vagrantup.com/downloads
2. Download VirtualBox: https://www.virtualbox.org/wiki/Downloads
   (Most recent version should be fine)

Software Defined Network (SDN)

Wikipedia:

(SDN) technology is an approach to network management that enables dynamic, programmatically efficient network configuration in order to improve network performance and monitoring making it more like cloud computing than traditional network management.



Make Decisions:

1. How to forward data.
2. Order to send data.

Perform Actions:

1. Forward
2. Route

What is Mininet & Pox?

- **Mininet**: A software that creates a virtual network. We can set up switches, connect components, and ping, etc.
- **Pox**: A Python-based SDN controller platform geared towards research and education. You will be using it to set up rules on the Mininet switches in parts 2-4.
- **Helpful Links**: See the last slide.

Mininet: Download & Install

Full instructions: https://github.com/kaichengyan/mininet-vagrant

1. Download & Install Vagrant: https://www.vagrantup.com/downloads
2. Download & Install VirtualBox: https://www.virtualbox.org/wiki/Downloads
3. Clone repo at the above link, `cd` into the repository
4. Run `vagrant up`
5. Run `vagrant ssh` to connect to the VM
6. Run `sudo mn --test pingall` to confirm you have installed the VM correctly

Talk to us during OH if you run into problems with Mininet setup!

- Github link also has instructions for installing Mininet directly on a VM
- Also instructions for VSCode and Wireshark setup

Project 2 Setup

1. Preparation: Download project2.zip and install unzip

```
$ cd                          # go back to your home dir
$ sudo apt-get update         # update apt packages
$ sudo apt-get install --reinstall ca-certificates wget tar  # Fix wget
$ sudo apt-get install unzip  # install unzip
$ wget https://courses.cs.washington.edu/courses/cse461/20sp/assignments/project2/project2.zip
$ unzip project2.zip          # unzip, you should see 461_mininet in ~
```

1. (Optional) Set up Visual Studio Code for remote development

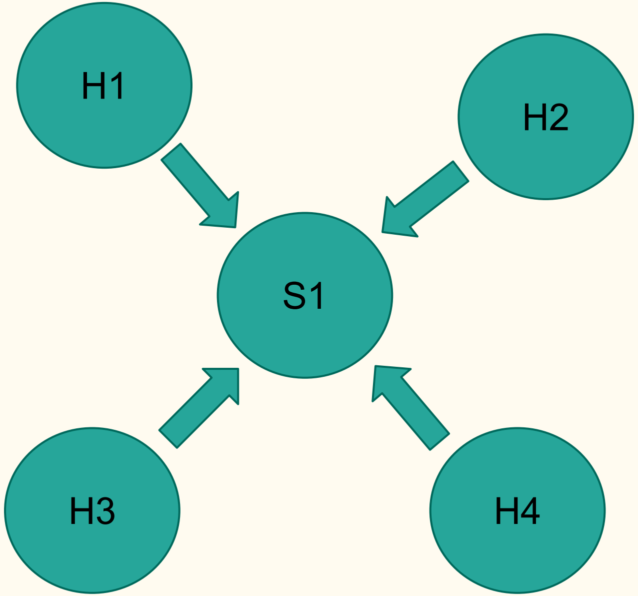
   See next slide

2. Try running part1 code: `$ sudo python 461_mininet/topos/part1.py`

(Optional) Visual Studio Code setup

1. Install the “Remote Development” extension in VS Code
2. Click on the “><” button in the very bottom left of the VS Code window
3. In the pop-up menu, select Remote-SSH: Connect to Host
4. If you have the VM in your SSH config, select your host and skip to step 7
5. If not, Select “+ Add New SSH Host” at the bottom, type in the SSH command:
   ```
   ssh mininet@localhost -p 2222
   ```
6. Optionally, open the SSH config file and rename the host; retry step 4
7. Once you are connected to your VM, open the 461_mininet folder you extracted
   - If the VSCode SSH connection fails, try uninstalling wget on mininet VM
   - sudo apt remove wget
   - Go to VSCode preferences and enable the option: `Remote.SSH: Allow Local Server Download`
   - Then retry the VSCode connection - it should work now
   - Install curl in order to open directories on the VM: sudo apt-get install curl
8. Optionally, for auto completion, install the “Python” extension on the remote VM, and configure python.autoComplete.extraPaths to include “/home/mininet/mininet” and “/home/mininet/pox”
9. Start developing!

Part 1: Build A Simple Network



Goal of part 1:

Modify file part1.py to create the following topology.

Run part 1:

```
$ sudo python 461_mininet/topos/part1.py
```

Some useful Mininet commands:

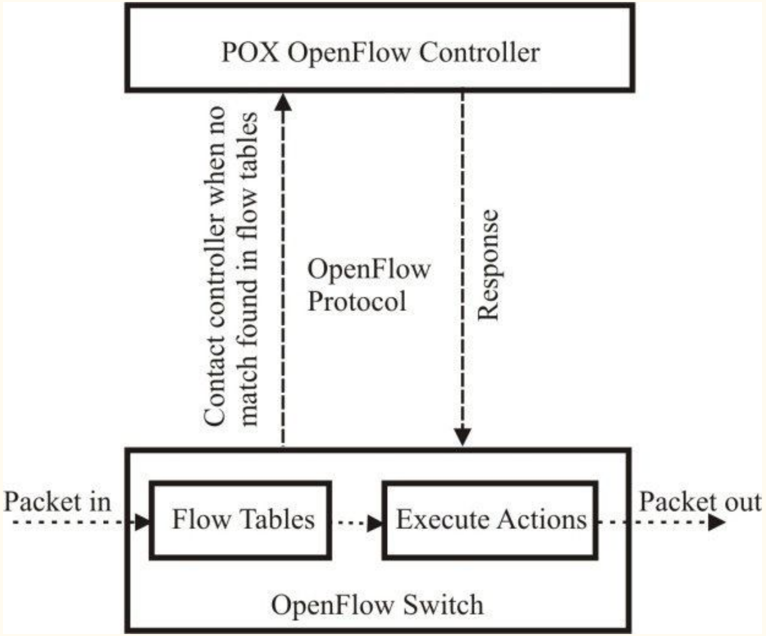
1. dump (dump info about all nodes)
2. pingall (Literally, ping all the connections)
3. ping (ping the connection h1 and h2)
4. iperf (Test the bandwidth between two hosts)

Walk Through Part 1

Watch how the TA creates the network and try it yourself!

Mininet + Pox Controller

In parts 2-4 you using a Pox Controller to manage Mininet switches.



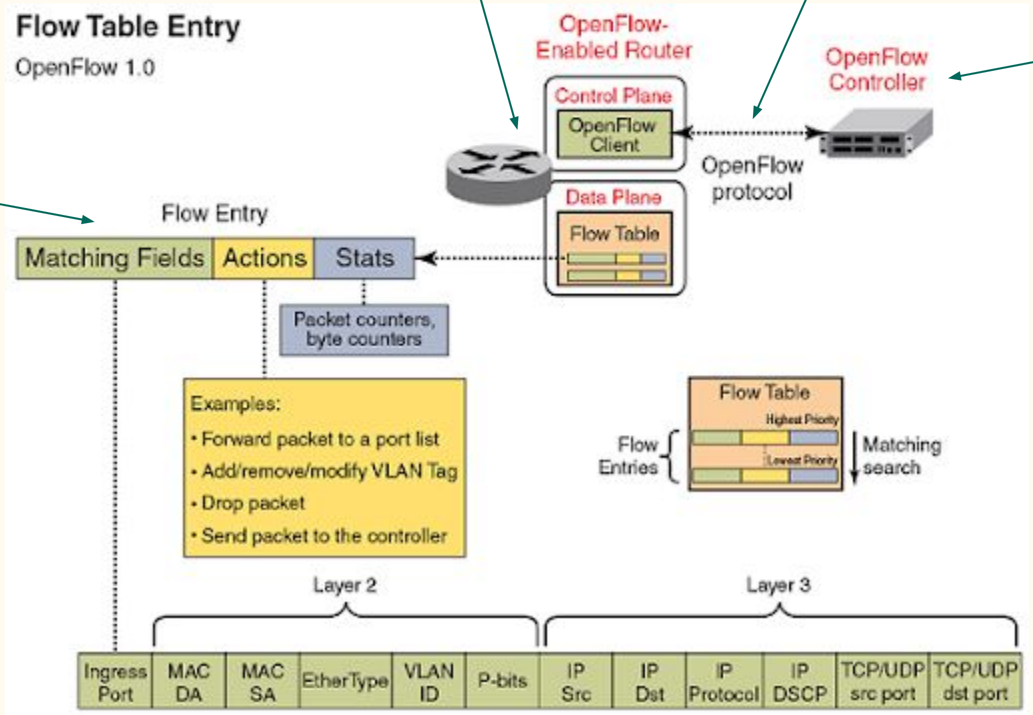
Mininet + Pox Controller

self.connection

Mininet Switch(es)

Pox Controller

of.ofp_flow_mod()



Part 2: Firewall

- Install Firewall rules to accept/drop packets
- Launch controller with `$ sudo ~/pox/pox.py misc.part2controller`
- Run network topo with `$ sudo python ~/461_mininet/topos/part2.py`
- Make sure that controller and topo are running at the same time!
- Look for constant definitions in ~/pox/pox/lib/packet

```
[h1@10.0.1.2/24][h2@10.0.0.2/24][h3@10.0.0.3/24][h4@10.0.1.3/24]
\                \               /               /
 \                \             /               /
  \                \----{s1}---/               /
   \-------------------/ |  \-----------------/
                         |
                    (controller)
```

src ip	**dst ip**	**protocol**	**action**
any ipv4	any ipv4	icmp	accept
any	any	arp	accept
any ipv4	any ipv4	-	drop

Part 3: L2 Switch

- Install forwarding rules to forward packets to its destination
  - No flooding for the central switch!
  - Other switches can continue to flood packets
- Launch controller with `$ sudo ~/pox/pox.py misc.part3controller`
- Run network topo with `$ sudo python ~/461_mininet/topos/part3.py`
- Run network topo with `$ sudo python ~/461_mininet/topos/part3.py`

```
[h10@10.0.1.10/24]--{s1}--\
[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]
[h30@10.0.3.30/24]--{s3}--/    |
                               |
                    [hnotrust1@172.16.10.100/24]
```

Part 4: Learning Router

- Router has to “learn” which switch is at which port
    - No hardcoding!
- By looking at ARP messages!
- Launch controller with `$ sudo ~/pox/pox.py misc.part4controller`
- Run network topo with `$ sudo python ~/461_mininet/topos/part4.py`

```
[h10@10.0.1.10/24]--{s1}--\
[h20@10.0.2.20/24]--{s2}--{cores21}--{dcs31}--[serv1@10.0.4.10/24]
[h30@10.0.3.30/24]--{s3}--/     |
                                |
                    [hnotrust1@172.16.10.100/24]
```

Start early!

It is not hard, but you will probably spend a lot time looking for and reading documentation...

Resources About Mininet & Pox

Mininet:

- https://github.com/mininet/mininet/wiki/Documentation
- https://github.com/mininet/mininet/wiki/Introduction-to-Mininet#creating

Pox Wiki and API docs:

- https://noxrepo.github.io/pox-doc/html/#id97

Pox OpenFlow Tutorials:

- https://github.com/mininet/openflow-tutorial/wiki/Create-a-Learning-Switch#Controller_Choice_POX_Python
- https://haryachyy.wordpress.com/2014/06/14/learning-pox-openflow-controller-proactive-approach/