

Section 4: Introduction to Mininet

Simulate a simple network using Mininet

Before we start...

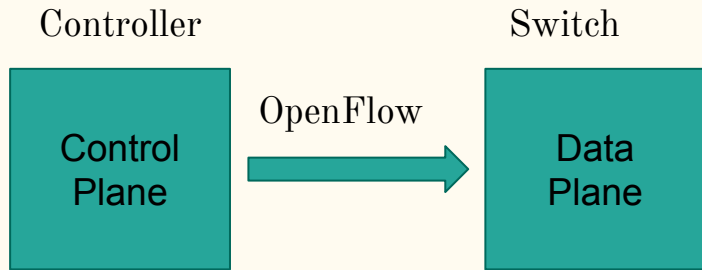
1. Download VirtualBox: <https://www.virtualbox.org/wiki/Downloads>
2. Download Vagrant: <https://www.vagrantup.com/downloads>

(We expect that you have already downloaded and installed VirtualBox and Vagrant!)

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

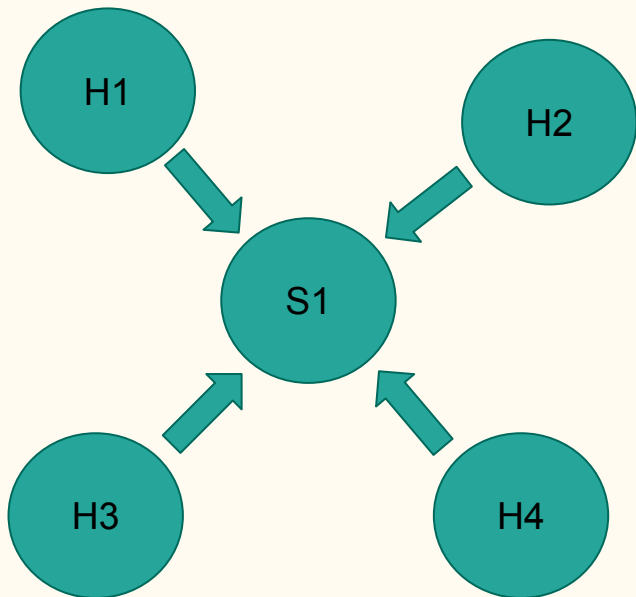
1. Download VirtualBox: <https://www.virtualbox.org/wiki/Downloads>
2. Download Vagrant: <https://www.vagrantup.com/downloads>
3. Clone/download: <https://github.com/kaichengyan/mininet-vagrant>
4. Follow the following commands to complete mininet setup:

```
$ cd mininet-vagrant           # go to the mininet-vagrant dir
$ vagrant up                   # setup the vm (takes a few minutes)
$ vagrant ssh                  # ssh into the vm
$ sudo mn --test pingall       # test mininet has been correctly installed
```

(Optional) Visual Studio Code setup

1. Run `vagrant ssh-config` from your `mininet-vagrant` directory
2. Add the output from running above command to your `~/.ssh/config` file. You can rename the host as you like (change the first line to `Host mininet`)
3. Install the “Remote Development” extension in VS Code
4. Click on the “><” button in the very bottom left of the VS Code window
5. In the pop-up menu, select **Remote-SSH: Connect to Host**
6. Here, you should be able to connect to the `mininet` host.
7. Optionally, for auto completion, install the Python and Pylance extensions on the remote VM, and configure `python.autoComplete.extraPaths` to include “`/home/vagrant/mininet`” “`/home/vagrant/pox`” and “`/home/vagrant/openflow`”
8. Start developing!

Part 1: Build A Simple Network



Goal of part 1:

Modify file part1.py to create the following topology.

Run part 1:

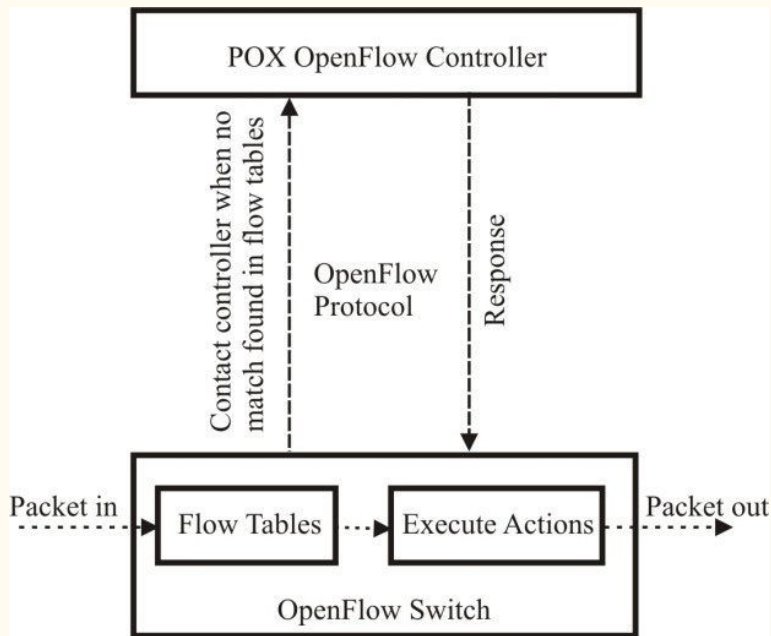
```
$ sudo python 461_mininet/topos/part1.py
```

Some useful Mininet commands:

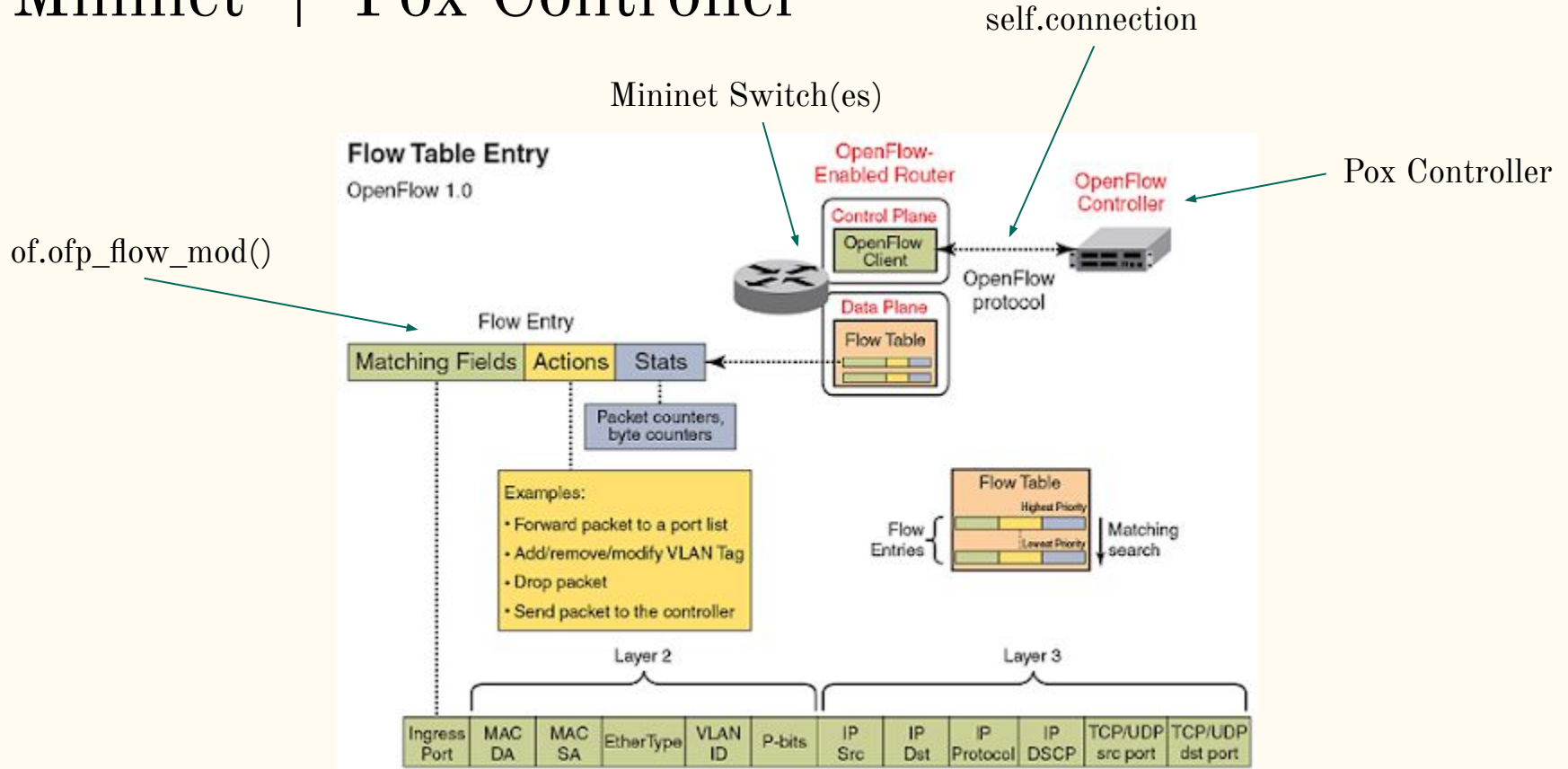
1. dump (dump info about all nodes)
2. pingall (Literally, ping all the connections)
3. ping (ping the connection h1 and h2)
4. iperf (Test the bandwidth between two hosts)

Mininet + Pox Controller

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



Mininet + Pox Controller



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`

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

Walk Through Setup and Part 1

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

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/>