

Section 3: Introduction to Mininet

Simulate a simple network using Mininet

Before we start...

1. Download VirtualBox: <https://www.virtualbox.org/wiki/Downloads>
(We expect that you have already downloaded and installed VirtualBox!)
2. Download Mininet: <http://mininet.org/download/>

Or directly from here

<https://github.com/mininet/mininet/releases/download/2.2.2/mininet-2.2.2-170321-ubuntu-14.04.4-server-amd64.zip>

TCP/UDP Review...

- **TCP:** Connections, reliable and in-order delivery, arbitrary length content, flow control, congestion control...
- **UDP:** Datagrams, unreliable, possibly out-of-order delivery, limited message size, no flow/congestion control... doesn't really do much, but still useful

A Quick TCP Exercise Problem

Imagine a link with the following (symmetric) properties:

Rate (bandwidth) = 10Mbps; $D = 25\text{ms}$; 10kb packets

(1kb = 1000b; 1Mbps = 10^6 bps; assume D is one-way delay; ignore transmission delay)

- a) What is the effective data rate with $W=1$?

- b) What is the appropriate window size to fully utilize the link?

A Quick TCP Exercise Problem - Solution

Rate (bandwidth) = 10Mbps; $D = 25\text{ms}$; 10kb packets

a) What is the effective data rate with $W=1$?

$$10\text{kb} / (25\text{ms} * 2) = 200 \text{ kbps}$$

b) What is the appropriate window size to fully utilize the link?

$$W = 2BD = 2 * 10\text{Mbps} * 25\text{ms} = 500\text{kb} = 50 \text{ packets}$$

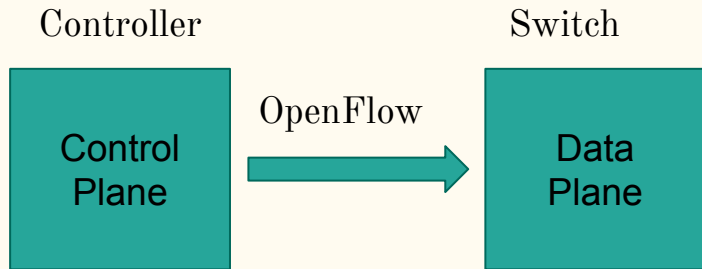
$$\text{or, } (10\text{Mbps}/200\text{kbps}) * 1 = 50$$

Review lecture slides if you didn't get it right! (It was not an easy problem.)

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.
- **Helpful Links:** See the last slide.

Mininet: Download & Install

1. Download VirtualBox: <https://www.virtualbox.org/wiki/Downloads>
(We expect that you have already downloaded and installed VirtualBox!)
2. Download Mininet: <http://mininet.org/download/>
Or directly from here <https://github.com/mininet/mininet/releases/download/2.2.2/mininet-2.2.2-170321-ubuntu-14.04.4-server-amd64.zip>
3. Import Mininet into VirtualBox
4. Set up VirtualBox SSH (Try scp and sshfs as well!):
Network → Network Settings → Advanced → Port Forwarding
Rule name = SSH, Host port = 2222, Guest port = 22
5. Log into Mininet: Username=mininet, Password=mininet
6. Talk to us during OH if you run into problems with VirtualBox setup!
 - Or try using VMware, which tends to work more smoothly... It's free for UW students!

Project 2 Setup

1. Preparation: Download project2.zip and install unzip

```
$ cd # go back to your home dir
$ 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 ~
```

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

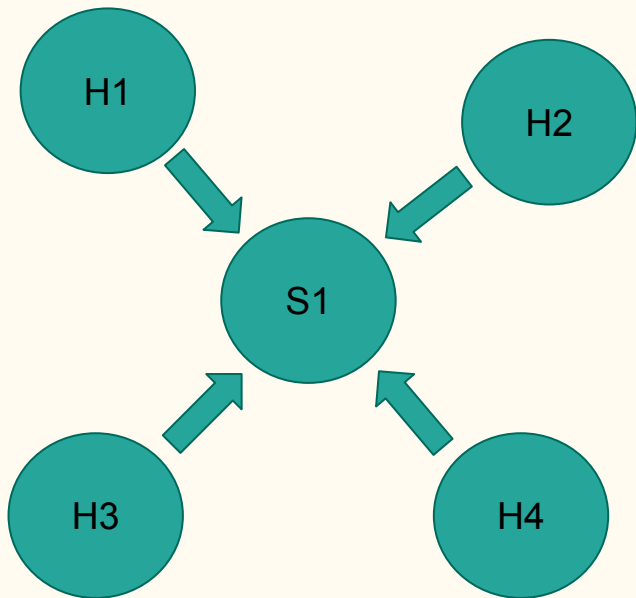
See next slide

3. 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
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!

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`
- 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

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`

```
[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:

- 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/>