Assignment 3: Network emulation
CSE 534, Fall 2022
Instructor: Aruna Balasubramanian
Due date: 11/22/2022, 9.00pm

In this homework you will acquire hands on experience in network emulation and implementation. You will need to do your homework using Mininet. If you are not familiar with Mininet, you are strongly advised to start early. First, set up Mininet

Setup (0 points):

For windows:

1. Install a Virtual Machine Software(recommend Oracle VM virtual box: https://www.virtualbox.org/wiki/Downloads)
2. Downloads the mininet VM image: https://github.com/mininet/mininet/releases/download/2.3.0/mininet-2.3.0-210211-ubuntu-18.04.5-server-amd64-ovf.zip
3. Import the VM image into virtual box(You can double-click to open in virtual box)
4. Default setup will be good and you can increase the RAM to 4GB if you want
5. Open the mininet VM.
6. username and password are both mininet

*For VirtualBox Shared Folders, install guest utilities with this command: sudo apt-get install virtualbox-guest-utils

a. Add the folder using Devices → Shared Folders and make the drive auto-mount and permanent. Restart the VM, and your folder will be under /media
b. To access the folder, you will need to add the mininet user to the group vboxsf: sudo adduser mininet vboxsf
c. Reboot

For Linux Ubuntu :

1. You can get source code git clone git://github.com/mininet/mininet
2. Then you can use: mininet/util/install.sh -a
3. This is for test sudo mn --switch ovsbr --test pingall

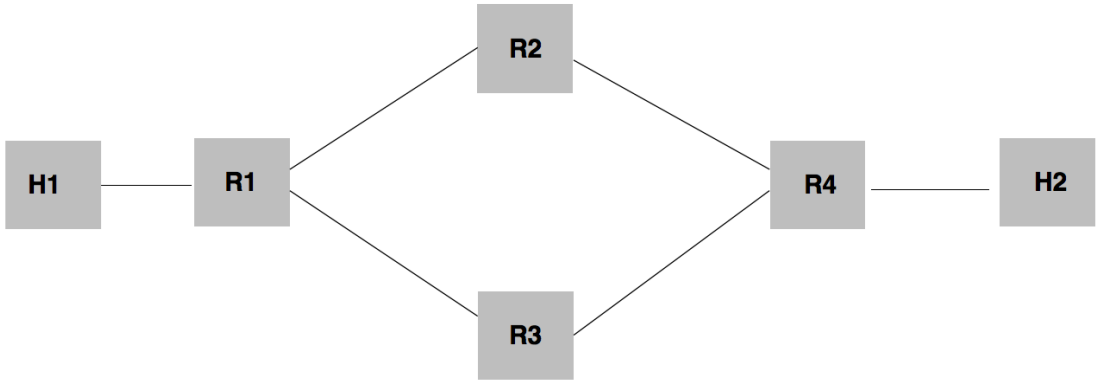
All this above you can find in http://mininet.org/download/

For Mac M1:

1. If you are an M1 user, you cannot install virtualbox. There are two workarounds: the cloud or using VMWare Fusion
2. If you are using the cloud, you can get free versions (e.g., amazon EC2 gives you free tiers for Linux VMs, Alibaba cloud platform also gives you free tiers), or you can get paid versions from linode, EC2, Alibaba and many others. Note that we will not be able to try on all of the cloud providers to see if it works. So far we have tested on Alibaba (paid version) and know it works. But it should likely work on other cloud providers.
3. Another possible workaround is to use VMWare Fusion (the department has a subscription https://it.stonybrook.edu/software/title/vmware) and UTM as a replacement for VirtualBox. However, we have not tried this.

PART A (30 points):

A1. Create your own topology as below.



To do this, write a program called MyTopo.py and create nodes H1, R1, R2, R3, R4, and H2. Hint: Create the two hosts and the router using the right set of commands, and then add a link to each node to create the topology above. You should use the LinuxRouter class included in Mininet for your routers. Remember that to create this topology you will have to choose a suitable set of IP addresses and configure them. Please read up on how to configure IP addresses and how to do this from within your python code. You can find some sample Python files about custom topologies under mininet/example
Check that the nodes are created. Check the routing table at each node.

Submit: (a) the MyTopo.py file you created, (b) The network topology figure including the IP addresses (with subnet information).

A2. Create static routes

Configure each node so that H1 can ping H2.

You will then need to create a static routing table at each node.

Submit: (a) the routing tables at all nodes (as a screen capture) and explain what you did to configure them correctly. (b) provide the trace route output that gives the path between nodes H1 & H2.

PART B (40 points):

First remove the static routes.

In this part you configure R1, R2, R3 and R4 as a RIP router. You need to BIRD 2.0.8

You can find the download site and instructions
here: https://bird.network.cz/download/bird-2.0.8.tar.gz
or https://gitlab.labs.nic.cz/labs/bird/

BIRD is a program running in your router, it uses a port to communicate with another router also running BIRD. You will need to run BIRD on each node by using command line utility such as bird - l or bird -c
Note: BIRD has no interface with Python, but you can call the command from within Python.

Here is an example BIRD configuration file:

```
protocol kernel {
        ipv4 {
                import all;
                export all;      #default is export none
        };
        persist;                 # Don't remove routes on BIRD shutdown
}

protocol device {

}

protocol direct {
        ipv4;
        interface "-arc*", "*" #Exclude the ARCnets
}
protocol rip {
       # write your code here
}
```

Here is a page you may want to read for detail:
https://bird.network.cz/?get_doc&v=20&f=bird-4.html

B1: Write a python script myRIP.py that will run the RIP daemon on each router and host.

Submit: (a) the python script myRIP.py, (b) the routing tables at each node, (c) the traceroute output that gives the path between nodes H1 & H2, (d) the BIRD config file

B2:
Bring down **R1-R2 or R1-R3 (**whichever is on your current path from H1 to H2) and wait for the RIP daemon to find the new route.

Submit: (a) how you got the link to go down, (b) provide the traceroute output that gives the new path between nodes H1 & H2.

Part C (30 points)

We will perform an IPerf test of H1 as a host and H2 is the server to perform a TCP bandwidth test. See useful tools to see how to install IPerf.

Write a Python script MyIperf.py to configure each router with the following buffer sizes: 10Kb, 5Mb, 25Mb. The bandwidth is 100Mbps and the delay at each router is 30ms. You can use the "tc" command to configure this. The name of the tc tbf parameter that sets the buffer size is "limit." More TC information can be found on the man page (https://man7.org/linux/man-pages/man8/tc.8.html). Some examples of TC can also be

found at (https://wiki.archlinux.org/title/advanced_traffic_control)

Now run IPerf (using TCP) for 10 seconds with H1 as the host and H2 as the server. Use xterm to open two windows to make it easy. If you do not use xterm, then you can run them from the same machine and redirect your output to a log. Ipfer example can be found here (https://linuxhint.com/iperf_command_usage/)

For each buffer setting, submit the screenshot (or log) at the client and server. Explain the retransmission and the bandwidth in each case by calculating the BDP and comparing it with the buffer size.

Submit: (a) MyIperf.py, (b) screenshots for the three scenarios, (c) your explanation.

Submission instruction

Submit each part in a separate folder A, B, C. Each part needs to be stand-alone and should run on their own.

If there are any special instructions on running the code, please include those instructions in each section

Useful tools (exact command may vary depending on your development environment, these are just hints)

sudo mn –c: for cleaning the previous mininet configurations

sudo apt-get install: traceroute to install traceroute

sudo apt-get install iperf3 to get the iperf3

sudo apt-get update

Xterm usage:

If you are using windows you can use xterm (node name) to open a command line window for a node. To enable this, you need to install a Xming on windows and enable X11 forwarding (It is under SSH options in Putty). You can use Xming in the following link: https://drive.google.com/file/d/1FUhTV2aH7GO-iCcnp6cK9qbMQQZAtSPg/view?usp=sharing
For Mac, you can download XQuartz. If you are using SSH to connect to your Mininet instance, you will need to use the -X option to enable X11 forwarding.
For linux, there may be a similar setting.

Xterm is not available in the built-in terminal provided by VirtualBox, so you will need to SSH into your Mininet instance from another SSH client, such as PuTTY. Use the ifconfig command to obtain the Mininet instance's IP address, then use the username and password ``Mininet" to SSH.

You may need change your network setting for VM to Bridged Adapter to view your IP address.

Some notes on installing BIRD

1. For bird installation, make sure you have flex, bison, libncurses-dev, libreadline-dev installed before installing bird

2. If you're using VM and a shared folder, copy entire folder to some different location within the vm. I figured out that running bird from within the shared folder was creating the issue.

3. When you're entering the individual node directory and running the bird -l command and then exiting, make sure you're running the cd command from within the node (hint: use mininet for this). If you do something like os.chdir('r1') or 'cd ./r1' it won't work.

4. Run the command as a root user to give bird all the privileges.