

Assignment 2 - Mininet Topology

Goal

The goal of this assignment is to familiarize yourself with Mininet. You will modify a Mininet topology using Python and then observe the throughput and latency measurements to understand how the simulator works.

In the directions below, you might not be able to cut and paste into your virtualbox terminal window, but you should be able to if you use `ssh mininet@ip_address` from (Terminal on Mac OSX, Putty on Windows and xterm on Linux) as you did at the end of the first assignment.

Directions

1. Clone the project starter code from Github into the Mininet virtual machine `git clone https://github.com/udacity/gt-cs6250.git`
2. Install two libraries necessary to run the measurement code `sudo apt-get install bwm-ng python-matplotlib`.
3. You can now run the example topology provided to simulate a host communicating with another host. Change into the assignment directory `cd gt-cs6250/assignment-2`. Then run the topology: `sudo ./topology.sh`. The script produces a time-stamped results folder as well as a couple graphs, one of the TCP congestion window and another of the bandwidth in megabits per second. To view the graphs, start a simple web server and navigate to the IP address of the virtual machine. Find the IP address which should be associated with the 'eth0' interface: `sudo ifconfig`. Then start the server `python -m SimpleHTTPServer`. On the host machine in a browser navigate to `http://ip_address:8000`. You can then browse to the results folder and view the graphs.



4. Now you will modify the Mininet topology to add another switch. The current topology is setup as shown below. You will modify the topology to the second topology shown below. To modify the topology, you should edit the Mininet topology file `mntopo.py`. The file is imported by the `measure.py` script which adds logging code to the topology. You should add a new switch and two new links. Refer to the [Mininet documentation](#) to piece apart the topology file. When you have modified the topology, re-run the topology test script `sudo ./topology.sh`. The graphs should be similar to the graphs produced in your earlier test run. The similarity should come as no surprise because the new switch and links in the topology are adding a slight amount of total latency but have the same bandwidth properties as the other links.



5. The next two steps involve tweaking topology parameters and observing their outputs. First we will modify the latency of the topology. Before we modify the latency, we will test the current latency using the `ping` command. Run the following from the assignment folder `sudo python ./ping.py`. To modify the latency we will adjust the delay on the links in the `mntopo.py` file. Adjust the `delay` in the `LinkConfig` dictionary to 10ms. Then run ping script again `sudo python ./ping.py`. The output from the first run should take approximately 6 ms, the time for one packet to traverse three links to the receiver and the acknowledgement three links back to the sender. The packet latency is the fourth column of the ping output and note you should discard the first result as it is an outlier from the command starting up. Once you've increased the latency you should see the latency increase to approximately 60 -


- 4/6/2018
- Assignment 2 - Mininet Topology - Udacity
- 65 ms as expected adding the individual latencies of the links for one round trip. The beauty of Mininet is these configuration parameters allow us to emulate real network events without modifying common network tools like `ping`.
6. Now we will modify the bandwidth and observe the change in the topology. Adjust the `bw` in the `LinkConfig` dictionary to 50 which will adjust the bandwidth along each link to 50Mbits per second. To confirm Mininet performs this correctly run `sudo ./topology.sh`. And view the `rate.png` output graph using the Python web server as in step 3.
7. To complete this assignment, submit your modified topology and the `bwm.txt` file generated in step 6 to the [code editor for Assignment 2](#).

Rubric

This rubric is here to help you understand the expectations for the assignment. It is the same rubric that the person evaluating your project will use. We recommend you look at the rubric before you begin working on your project and again before you decide to submit it.

Criteria	Does Not Meet Expectations	Meets Expectations	Exceeds Expectations
Topology code			
Topology is as described above.	Topology is not as described.	Topology is as described with two switches and two hosts.	There is no "Exceeds Expectations" option for this criteria.
Bandwidth output			
Output looks reasonable.	Output does not pass the sanity check when it is submitted.	Output does pass the sanity check and the rate log matches the 50Mbits per second specified above.	There is no "Exceeds Expectations" option for this criteria.

This revision was added on 2014/04/25 08:14:50.



POPULAR NANODEGREE PROGRAMS

STUDENT RESOURCES

UDACITY

INQUIRIES

+

+

+

+

Nanodegree is a trademark of Udacity
© 2011–2018 Udacity, Inc.

