Logout

My Workspace CS-6250-O01 FALL14 CS-6300-O01 FALL14

ASSIGNMENTS

- Home
- Syllabus
- Announcements
- Resources
- Assignments
- Gradebook
- Email Archive
- Roster
- Site Info
- Section Info
- Piazza
- Help

Assignment 2 - Mininet Topology - Returned

Title	Assignment 2 - Mininet Topology
Student	Telfort, Dominique Gayot
Submitted Date	Sep 2, 2014 9:39 pm
Grade	10.0 (max 10.0)

Instructions

Assignment 2 - Mininet Topology

Goal

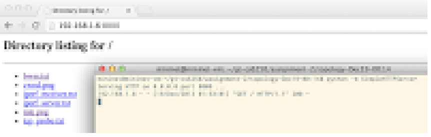
The goal of this assignment is to familiarize yourself with Mininet. You will modify a Mininet topology using Python and then observe the throughput and latency measurements to understand how the simulator works.

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

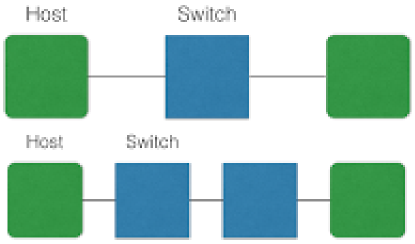
1. Clone the project starter code from Github into the Mininet virtual machine `git clone https://github.com/OMS6250/gt-cs6250.git`
2. Update the repository locations by issuing the following command (be sure to copy and paste *exactly*): `sudo sed -i "s/(security.ubuntu.com\|mirrors.k`
3. From here, issue the following command to get apt-get working again: `sudo apt-get update`
4. Install two libraries necessary to run the measurement code `sudo apt-get install bwm-ng python-matplotlib`.

5. 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.



Directory listing for /

6. 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.



7. 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