Testing mptcp, mininet, and floodlight with fdm

1. Configure MPTCP

Our mininet topology requires each ship to launch MPTCP with 3 subflows. Therefore, configure MPTCP following http://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP

Make sure you enable MPTCP, set path manager as fullmesh, and set the parameter as 3 by doing *"sudoedit /sys/module/mptcp\_fullmesh/parameters/num\_subflows"*.

2. Run floodlight controller

You need JAVA8 to build the floodlight controller. Follow https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Installation+Guide

to compile and build floodlight. Basically, there are two options: build with "ant" and then run floodlight with "java -jar target/floodlight.jar"; or compile with Eclipse.

One thing you need to change in floodlight is in floodlight/src/main/resources/floodlightdefault.properties. If you run floodlight on your virtual machine, line 45 should be: *net.floodlightcontroller.core.internal.OFSwitchManager.openFlowAddresses= 0.0.0.0;* if you are running it on another machine, the address should be the IP address of that machine.

3. Run mininet

Once floodlight controller starts to work, wait for a while until you see the console message “**Sending LLDP packets out of all the enabled ports**”. Then create topology with

*sudo mn --custom path\_to\_file/final3\_fullmesh.py --topo=mytopo -- controller=remote,ip=IP\_of\_controller,port=6653 --switch=user*

Here, IP\_of\_controller should be consistent with line 45 of floodlightdefault.properties. If you run floodlight locally, the IP is 127.0.0.1, otherwise it is the IP address of remote controller.

If the connection is established, you should be able to see the topology being updated in floodlight console. Again, wait for floodlight to update the topology, until you see “**Sending LLDP packets out of all the enabled ports**”.

Now load the bandwidth demand and capacity parameters with “path\_to\_file/runconfig.sh” and wait for floodlight to update the parameters. Meantime, you can open xterminals of the ships and destination by typing “xterm ship1 ship2 ship3 dest” in Mininet command line.

You can ping the ships or destination to test the connection. Then use iperf to probe the channel bandwidth.

For Node:dest, do “iperf -s -i 2” to wait for upcoming MPTCP sessions;

For Node:ship1-3, do “iperf -c 10.0.0.1 -t 50 -i 2” to initiate MPTCP session to the dest.

You will see the throughput from each ship to dest printed by iperf. Note that the runconfig.sh file specifies all the requirements as 6Mpbs, and capacities are 5, 10, and 5Mpbs. If FDM works correctly, you should be able to see the throughput is below 6, and the computation of FDM is printed in the console. The current code is easy to break, so occasionally FDM returns “infeasible”, meaning it fails to capture the correct topology or parameters, and you will see the throughput of iperf exceeds 6Mbps. In this case, you could delete the Meters on switches by running ./delete.sh, and then run ./runconfig.sh again, to load the parameters. Also notice that always give floodlight enough time to process the input. After each of the operations above, wait until you see “**Sending LLDP packets out of all the enabled ports**”. To check MPTCP is working properly, you could open one more xterminal e.g. ship1, and do “watch -n -1 “netstat -n” ”, to see if there are 3 subflows running.