Varying bandwidth and latency by using Dockers +Mininet

**1. Install Docker.**

Ref: <https://docs.docker.com/install/>

**2. Install Mininet**

Ref: http://mininet.org/download

**3. Install docker-package for python**

$ sudo pip install docker

**4. Configure docker-Mininet**

4.1 Replace “~/mininet/mininet/util.py” by our own “util.py”.

For “util.py”, we just modified the API-function “makeIntfPair”.

4.2 Run “cd ~/mininet” to change into mininet directory.

4.3 Then run “sudo make install” to make the modification effective.

4.4 Copy our docker-mininet API file “dockernode.py” into the work directory. Then enjoy simulations by using python script. An example code is attached in the Appendix, it is also used in this experiment.

**5. Experiments**

**5.1** Experiment setup



As shown in Figure 1 where there are two dockers connected by a virtual switch. The virtual switch is named Open vSwitch (OVS). The left docker send data flow to the right docker. The data flow is generated by Iperf, a network testing tool. We can vary link delays and bandwidth by using simple API functions as shown:



**5.2** Experiment result

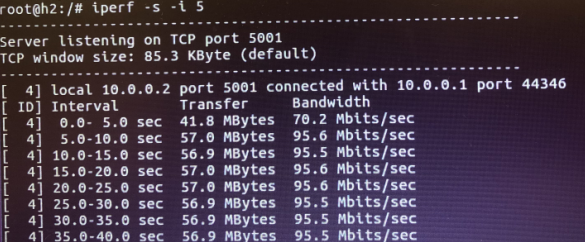


Figure 1. Limit bandwidth to 100Mbps

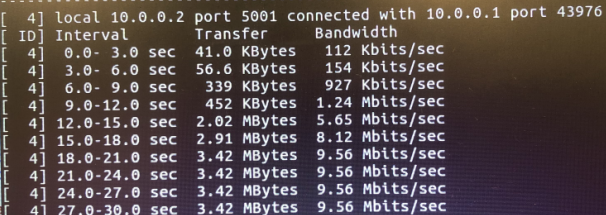


Figure 2. Limit bandwidth to 10Mbps

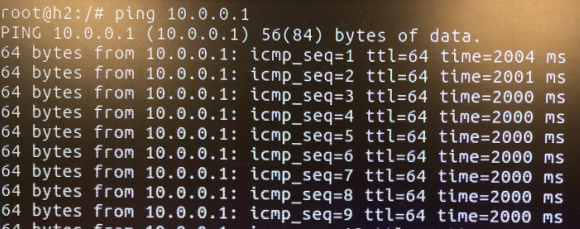


Figure 3. Limit latency of each link to 500ms, then the RTT is 2000ms

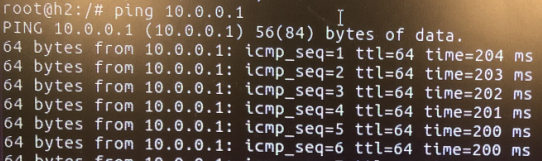


Figure 4. Limit latency of each link to 50ms, then the RTT is 200ms

**5.3** Discussion

By using docker + Mininet, we can connect dockers by Open vSwitch and can limit the bandwidth and delays for each link. If we would like to launch custom number of dockers, we can implement it by writing some automatic script.

In addition, we may have several considerations:

1. As Dumitrel’s paper shows, the maximum bandwidth of Mininet is determined by the hardware performance. So if the overload of simulation is heavy, the maximum bandwidth at each link can achieve is unknown, leading to an inaccurate simulation result.
2. How about the next plan of the project? Maybe we should discuss about it.

Appendix

The code used to test

