

Problem Sheet #5

Mininet (<http://mininet.org/>) is a network emulator that uses Linux namespaces to emulate a complete network within a single Linux kernel. While you can install and run mininet inside your Linux kernel, it is easier to download a mininet virtual machine image (<http://mininet.org/download/>) and to run that image in VirtualBox (<https://www.virtualbox.org/>).

While mininet has been primarily designed to experiment with new networking technologies such as OpenFlow, we will use mininet here to create some simple network topologies and to do some basic experiments on it. Next to mininet, you will use tools such as `tcpdump`, `wireshark`, `iperf`, and `scapy`.

Problem 5.1: *mininet network emulation*

(4+2+2+2 = 10 points)

- a) Download the mininet virtual machine image and run it in VirtualBox. Configure VirtualBox such that the mininet virtual machine has a NATed network interface to reach the Internet and a host network interface that allows you to login from your computer into the mininet virtual machine using `ssh`.

Within the mininet virtual machine, run mininet by typing the following command:

```
sudo mn
```

Run `iperf` within mininet to determine the throughput between the two nodes `h1` and `h2`. Next, run mininet with different link configurations:

```
sudo mn --link tc,bw=10
sudo mn --link tc,bw=10,delay='10ms'
sudo mn --link tc,bw=10,delay='10ms',loss=1
sudo mn --link tc,bw=10,delay='10ms',loss=5
sudo mn --link tc,bw=10,delay='10ms',loss=10
```

Document the results you measure in a table. For one of the runs, capture the packets in `h1` by running `tcpdump -w /tmp/h1.pcap &` in the background. Load the `h1.pcap` file into `wireshark` in order to study the TCP exchange. Which endpoint is initiating the teardown of the `iperf` TCP connection?

- b) The power of mininet is the Python API provided. Write a Python script to generate a topology consisting of a switch (and an associated controller) and three hosts (`h1`, `h2`, and `h3`) attached to the switch. Use the `addHost`, `addLink`, and `addController` API functions.
- c) Extend the Python script such that it configures two /64 IPv6 networks with the following interfaces:

```
2001:638:709:a::/64  h1-eth0, h2-eth0
2001:638:709:b::/64  h3-eth0, h2-eth0
```

Make sure host `h2` has forwarding enabled. Within the script, configure appropriate static routing table entries such that all hosts can send IPv6 packets to each other. (Run `ping6` to verify your script creates a working setup.)

- d) The `scapy` Python extension provides a powerful interactive packet manipulation program. It is a swiss army knife and as every sharp knife, it can be used for good and bad purposes.

Write a `scapy` Python script to send an IPv6/ICMPv6 echo request packet from `h1` to `h3`. Modify the script to send an IPv6 packet with a topologically incorrect source address. Is the packet delivered to `h3`? If so, does `h3` respond to it?