Computer Networks
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320301
Date: 2016-10-30
Due: 2016-11-07

Problem Sheet #4

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 system, 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`, and `iperf`.

Problem 4.1: *mininet network emulation* (2+3+5 = 10 points)

a) Download the mininet virtual machine image (tested with mininet 2.2.1) 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
```

This will start mininet with a minimal topology consisting of two hosts (`h1` and `h2`), an OpenFlow switch (`s1`), and an OpenFlow controller (`c0`). Create a table of the following format showing the details of the network interfaces of `h1`, `h2`, and `s1`.

system	interface	IPv4 address	description
h1			
h2			
s1			

b) Run `iperf` within mininet to determine the datarate between the two nodes `h1` and `h2`. Conduct a series of measurements with varying link configurations:

```
sudo mn
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 that has the following format:

link configuration	transfer	datarate
bw=10		
bw=10,delay='10ms'		
bw=10,delay='10ms',loss=1		
bw=10,delay='10ms',loss=5		
bw=10,delay='10ms',loss=10		

c) 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. Extend the Python script such that it configures two /64 IPv6 networks with the following prefixes:

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

Host h2 acts as an IPv6 router. Make sure host h2 has IPv6 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.)

Note: Mininet 2.2.1 disables IPv6 by default. To enable IPv6 again, you have to edit the configuration file /etc/default/grub to remove ipv6.disable=1 from the kernel command line. Running

```
sudo update-grub
```

will create a new bootloader configuration. The change will take effect after the next reboot.