CS348 Computer Networks
Assignment 7

Shikhar Jaiswal
1601CS44

We have to write Python code in Mininet to create the topology as shown in the figure, where hosts h1 and h2 are connected to routers r1 and r2 respectively. Each of the router has their specific static routing table. We then ping and trace the route among all the hosts.

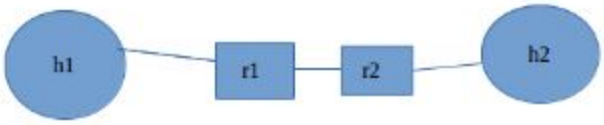

Figure 1: A switching network

We start by importing the necessary modules:

```
from mininet.net import Mininet
from mininet.node import Controller, RemoteController,
OVSKernelSwitch, UserSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.link import Link, TCLink
```

We then define the `topology()` function by initializing the `Mininet` object:

```
def topology():
```

```
    net = Mininet(controller = RemoteController, link = TCLink,
switch = OVSKernelSwitch)
```

Then we define the network topology:

```
    # Add hosts, switches and controller.
    h1 = net.addHost('h1', ip = "10.0.1.10/24", mac =
"00:00:00:00:00:01")
    h2 = net.addHost('h2', ip = "10.0.2.10/24", mac =
"00:00:00:00:00:02")
    r1 = net.addHost('r1')
    r2 = net.addHost('r2')
    net.addLink(r1, h1)
    net.addLink(r2, h2)
    net.addLink(r1, r2)
```

Then we initialize the network and start the controllers, while flushing the routers and setting automatic MAC addresses:

```
    net.build()
    r1.cmd("sysctl net.ipv4.ip_forward=1")
    r2.cmd("sysctl net.ipv4.ip_forward=1")

    r1.cmd("ifconfig r1-eth0 0")
    r2.cmd("ifconfig r2-eth0 0")
    r1.cmd("ifconfig r1-eth1 0")
    r2.cmd("ifconfig r2-eth1 0")

    r1.cmd("ifconfig r1-eth0 hw ether 00:00:00:00:01:01")
    r2.cmd("ifconfig r2-eth0 hw ether 00:00:00:00:01:02")
```

Then we assign IP addresses to specific hardware ports:

```
    r1.cmd("ip addr add 10.0.1.1/24 brd + dev r1-eth0")
    r1.cmd("ip addr add 10.0.3.2/24 brd + dev r1-eth1")
    r1.cmd("ip route add to 10.0.2.0/24 via 10.0.3.1 dev r1-eth1")

    r2.cmd("ip addr add 10.0.2.1/24 brd + dev r2-eth0")
    r2.cmd("ip addr add 10.0.3.1/24 brd + dev r2-eth1")
```

```
    r2.cmd("ip route add to 10.0.1.0/24 via 10.0.3.2 dev r2-eth1")
```

We also add default IPs for forwarding table:

```
    r1.cmd("echo 1 > /proc/sys/net/ipv4/ip_forward")
    h1.cmd("ip route add default via 10.0.1.1")
    h2.cmd("ip route add default via 10.0.2.1")
```

Finally, we add the command-line interface functionality, and stop the network:

```
    print "*** Running CLI"
    CLI(net)
    print "*** Stopping network"
    net.stop()
```

To run the `topology()` function, we do:

```
if __name__ == '__main__':
    setLogLevel('info')
    topology()
```

Then we finally get the result:

```
*** Configuring hosts
h1 h2 r1 r2
*** Running CLI
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X
h2 -> h1 X X
r1 -> h1 h2 X
r2 -> h1 h2 X
*** Results: 50% dropped (6/12 received)
mininet> r1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0
r1-eth0
```

```
10.0.2.0        10.0.3.1        255.255.255.0   UG    0     0     0
r1-eth1
10.0.3.0        0.0.0.0         255.255.255.0   U     0     0     0
r1-eth1
mininet> exit
*** Stopping network
*** Stopping 0 controllers

*** Stopping 3 links
...
*** Stopping 0 switches

*** Stopping 4 hosts
h1 h2 r1 r2
*** Done
```