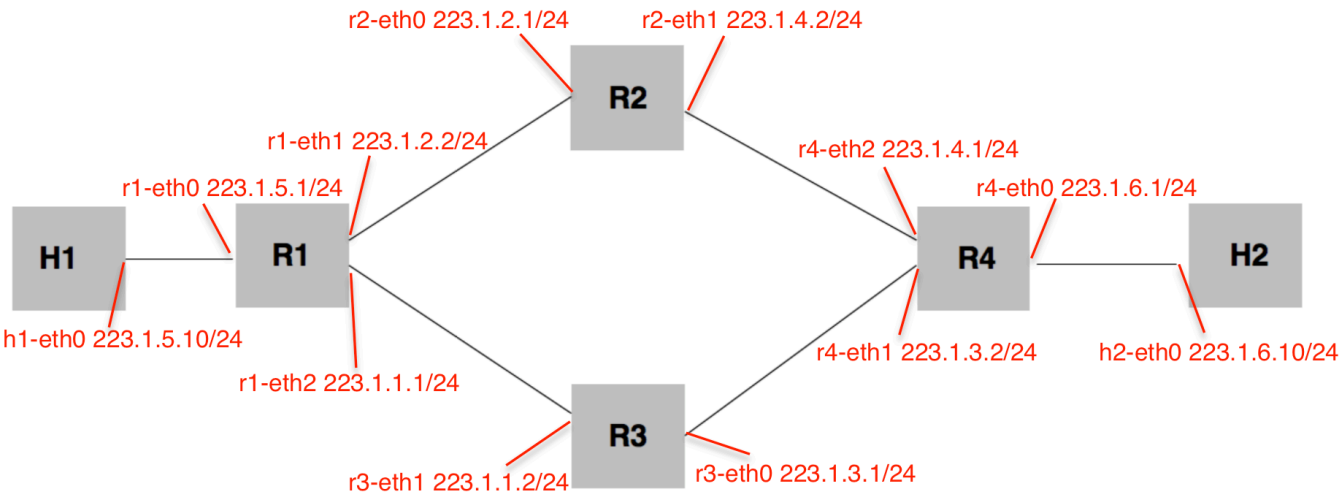
A1. Create your own topology as below.
(a) topo.py
---Not in this PDF file---

(b) the network topology figure.



A2. Create static routes.
(a) The routing table at all nodes (as a screen capture), explain what you did.

Here are the steps I did.

1. **Create hose and add link in topo.py**
2. **Set every host's interface an appropriate IP address in topo.py**
3. **Set configure file configs/daemons: zebra=yes**
4. **Activate the ip forwarding variable to 1 in start.py**
5. **Set configure file configs/zebra.conf: set routing table with command "ip route dest gateway"**

Here are the screen captures.

```
mininext> h1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         223.1.5.1       0.0.0.0         UG    0      0        0 h1-eth0
223.1.5.0       *               255.255.255.0   U     0      0        0 h1-eth0
```

```
mininext> h2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         223.1.6.1       0.0.0.0         UG    0      0        0 h2-eth0
223.1.6.0       *               255.255.255.0   U     0      0        0 h2-eth0
```

```
mininext> r1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
223.1.1.0       *               255.255.255.0   U     0      0        0 r1-eth2
223.1.2.0       *               255.255.255.0   U     0      0        0 r1-eth1
223.1.3.0       223.1.1.2       255.255.255.0   UG    0      0        0 r1-eth2
223.1.4.0       223.1.2.1       255.255.255.0   UG    0      0        0 r1-eth1
223.1.5.0       *               255.255.255.0   U     0      0        0 r1-eth0
223.1.6.0       223.1.1.2       255.255.255.0   UG    0      0        0 r1-eth2
mininext> r2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
223.1.1.0       223.1.2.2       255.255.255.0   UG    0      0        0 r2-eth0
223.1.2.0       *               255.255.255.0   U     0      0        0 r2-eth0
223.1.3.0       223.1.4.1       255.255.255.0   UG    0      0        0 r2-eth1
223.1.4.0       *               255.255.255.0   U     0      0        0 r2-eth1
223.1.5.0       223.1.2.2       255.255.255.0   UG    0      0        0 r2-eth0
223.1.6.0       223.1.4.1       255.255.255.0   UG    0      0        0 r2-eth1
```

```
mininext> r3 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
223.1.1.0       *               255.255.255.0   U     0      0        0 r3-eth1
223.1.2.0       223.1.1.1       255.255.255.0   UG    0      0        0 r3-eth1
223.1.3.0       *               255.255.255.0   U     0      0        0 r3-eth0
223.1.4.0       223.1.3.2       255.255.255.0   UG    0      0        0 r3-eth0
223.1.5.0       223.1.1.1       255.255.255.0   UG    0      0        0 r3-eth1
223.1.6.0       223.1.3.2       255.255.255.0   UG    0      0        0 r3-eth0
```

```
mininext> r4 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
223.1.1.0       223.1.3.1       255.255.255.0   UG    0      0        0 r4-eth1
223.1.2.0       223.1.4.2       255.255.255.0   UG    0      0        0 r4-eth2
223.1.3.0       *               255.255.255.0   U     0      0        0 r4-eth1
223.1.4.0       *               255.255.255.0   U     0      0        0 r4-eth2
223.1.5.0       223.1.3.1       255.255.255.0   UG    0      0        0 r4-eth1
223.1.6.0       *               255.255.255.0   U     0      0        0 r4-eth0
```

(b) provide the trace route output that gives the path between nodes h1 and h2

```
mininext> h1 traceroute h2
traceroute to 223.1.6.10 (223.1.6.10), 30 hops max, 60 byte packets
 1  223.1.5.1 (223.1.5.1)  0.019 ms  0.004 ms  0.003 ms
 2  223.1.1.2 (223.1.1.2)  0.011 ms  0.005 ms  0.004 ms
 3  223.1.3.2 (223.1.3.2)  0.011 ms  0.007 ms  0.006 ms
 4  223.1.6.10 (223.1.6.10)  0.012 ms  0.007 ms  0.006 ms
```