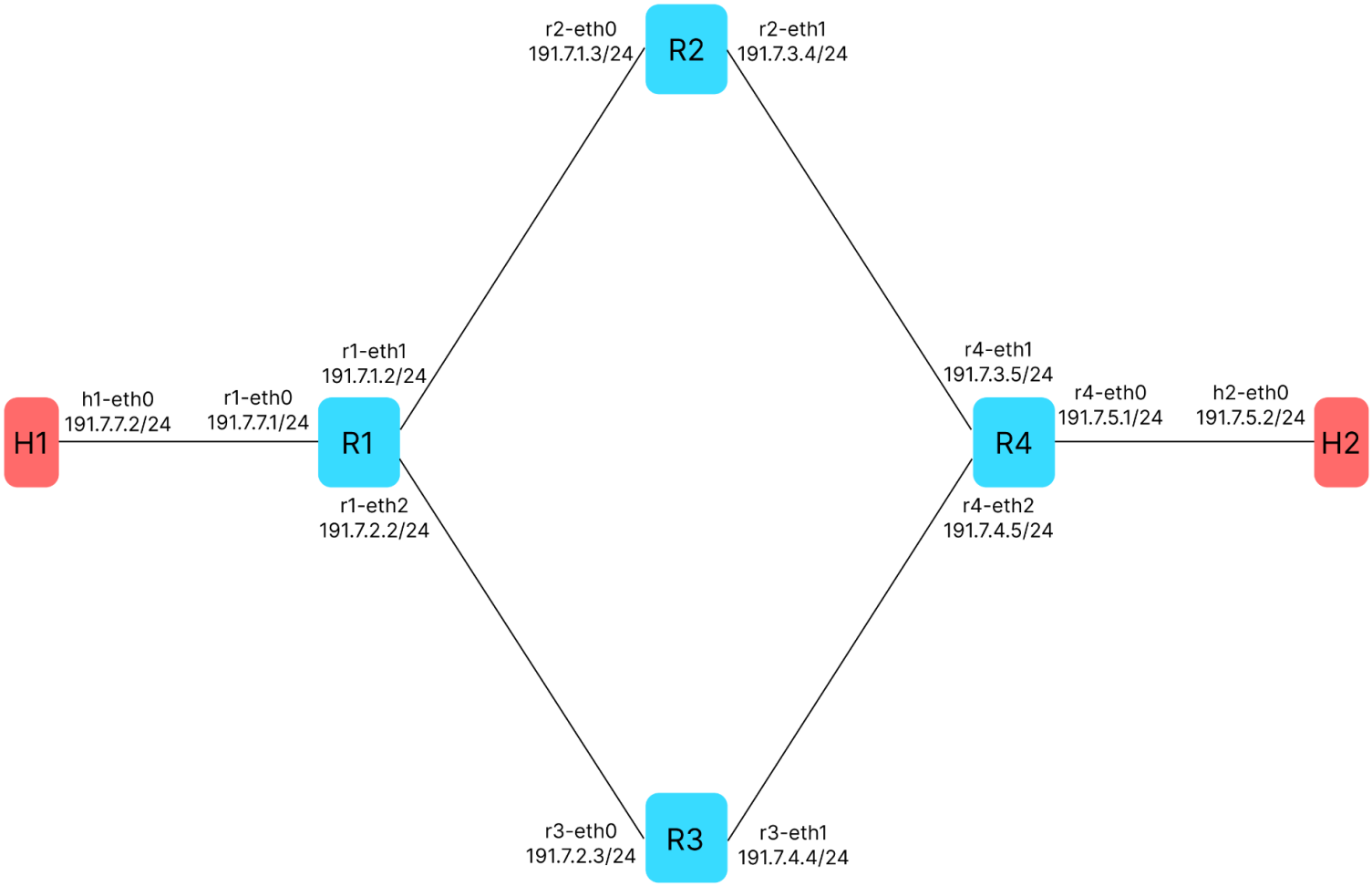
Part A

A1. Create your own topology as below

a. MyTopo.py added

b. Network topology figure



A2. Create static routes

a. Routing tables at each node

```
mininet> r1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
191.7.1.0       0.0.0.0         255.255.255.0   U     0      0        0 r1-eth1
191.7.2.0       0.0.0.0         255.255.255.0   U     0      0        0 r1-eth2
191.7.3.0       191.7.1.3       255.255.255.0   UG    0      0        0 r1-eth1
191.7.4.0       191.7.2.3       255.255.255.0   UG    0      0        0 r1-eth2
191.7.5.0       191.7.1.3       255.255.255.0   UG    0      0        0 r1-eth1
191.7.7.0       0.0.0.0         255.255.255.0   U     0      0        0 r1-eth0
mininet> r2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
191.7.1.0       0.0.0.0         255.255.255.0   U     0      0        0 r2-eth0
191.7.2.0       191.7.1.2       255.255.255.0   UG    0      0        0 r2-eth0
191.7.3.0       0.0.0.0         255.255.255.0   U     0      0        0 r2-eth1
191.7.4.0       191.7.3.5       255.255.255.0   UG    0      0        0 r2-eth1
191.7.5.0       191.7.3.5       255.255.255.0   UG    0      0        0 r2-eth1
191.7.7.0       191.7.1.2       255.255.255.0   UG    0      0        0 r2-eth0
mininet> r3 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
191.7.1.0       191.7.2.2       255.255.255.0   UG    0      0        0 r3-eth0
191.7.2.0       0.0.0.0         255.255.255.0   U     0      0        0 r3-eth0
191.7.3.0       191.7.4.5       255.255.255.0   UG    0      0        0 r3-eth1
191.7.4.0       0.0.0.0         255.255.255.0   U     0      0        0 r3-eth1
191.7.5.0       191.7.4.5       255.255.255.0   UG    0      0        0 r3-eth1
191.7.7.0       191.7.2.2       255.255.255.0   UG    0      0        0 r3-eth0
mininet> r4 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
191.7.1.0       191.7.3.4       255.255.255.0   UG    0      0        0 r4-eth1
191.7.2.0       191.7.4.4       255.255.255.0   UG    0      0        0 r4-eth2
191.7.3.0       0.0.0.0         255.255.255.0   U     0      0        0 r4-eth1
191.7.4.0       0.0.0.0         255.255.255.0   U     0      0        0 r4-eth2
191.7.5.0       0.0.0.0         255.255.255.0   U     0      0        0 r4-eth0
191.7.7.0       191.7.3.4       255.255.255.0   UG    0      0        0 r4-eth1
```

```
mininet> h1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         191.7.7.1       0.0.0.0         UG    0      0        0 h1-eth0
191.7.7.0       0.0.0.0         255.255.255.0   U     0      0        0 h1-eth0
mininet> h2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         191.7.5.1       0.0.0.0         UG    0      0        0 h2-eth0
191.7.5.0       0.0.0.0         255.255.255.0   U     0      0        0 h2-eth0
```

We used the 'ip route add' command to add routes statically, for those not connected directly by the *addLink* function.

The routing table is configured in the code as follows:

```
    info( net['r1'].cmd("ip route add 191.7.3.0/24 via 191.7.1.3 dev r1-eth1"))
    info( net['r1'].cmd("ip route add 191.7.4.0/24 via 191.7.2.3 dev r1-eth2"))
    info( net['r1'].cmd("ip route add 191.7.5.0/24 via 191.7.1.3 dev r1-eth1"))
    info( net['r1'].cmd("ip route add 191.7.5.0/24 via 191.7.2.3 dev r1-eth2"))
```

For router R1:

1. Direct traffic from subnet between R2 and R4 to the next hop 191.7.1.3 at r1-eth1
2. Direct traffic from subnet between R3 and R4 to the next hop 191.7.2.3 at r1-eth2
3. Direct traffic from subnet between R4 and H2 to the next hop 191.7.1.3 at r1-eth1
4. Direct traffic from subnet between R3 and R4 to the next hop 191.7.2.3 at r1-eth2

```
    info( net['r2'].cmd("ip route add 191.7.5.0/24 via 191.7.3.5 dev r2-eth1"))
    info( net['r2'].cmd("ip route add 191.7.7.0/24 via 191.7.1.2 dev r2-eth0"))
    info( net['r2'].cmd("ip route add 191.7.4.0/24 via 191.7.3.5 dev r2-eth1"))
    info( net['r2'].cmd("ip route add 191.7.2.0/24 via 191.7.1.2 dev r2-eth0"))
```

For router R2:

5. Direct traffic from subnet between R4 and H2 to the next hop 191.7.3.5 at r2-eth1
6. Direct traffic from subnet between R1 and H1 to the next hop 191.7.1.2 at r2-eth0
7. Direct traffic from subnet between R3 and R4 to the next hop 191.7.3.5 at r2-eth1
8. Direct traffic from subnet between R1 and R3 to the next hop 191.7.1.2 at r2-eth0

```
      info( net['r3'].cmd("ip route add 191.7.5.0/24 via 191.7.4.5 dev r3-eth1"))
      info( net['r3'].cmd("ip route add 191.7.7.0/24 via 191.7.2.2 dev r3-eth0"))
      info( net['r3'].cmd("ip route add 191.7.3.0/24 via 191.7.4.5 dev r3-eth1"))
      info( net['r3'].cmd("ip route add 191.7.1.0/24 via 191.7.2.2 dev r3-eth0"))
```

For router R3:

9. Direct traffic from subnet between R4 and H2 to the next hop 191.7.4.5 at r3-eth1
10. Direct traffic from subnet between R1 and H1 to the next hop 191.7.2.2 at r3-eth0
11. Direct traffic from subnet between R2 and R4 to the next hop 191.7.4.5 at r3-eth1
12. Direct traffic from subnet between R1 and R2 to the next hop 191.7.2.2 at r3-eth0

```
    info( net['r4'].cmd("ip route add 191.7.7.0/24 via 191.7.3.4 dev r4-eth1"))
    info( net['r4'].cmd("ip route add 191.7.1.0/24 via 191.7.3.4 dev r4-eth1"))
    info( net['r4'].cmd("ip route add 191.7.2.0/24 via 191.7.4.4 dev r4-eth2"))
    info( net['r4'].cmd("ip route add 191.7.7.0/24 via 191.7.4.4 dev r4-eth2"))
```

For router R4:

13. Direct traffic from subnet between H1 and R1 to the next hop 191.7.3.4 at r4-eth1
14. Direct traffic from subnet between R1 and R2 to the next hop 191.7.2.2 at r4-eth1
15. Direct traffic from subnet between R1 and R3 to the next hop 191.7.4.5 at r4-eth2
16. Direct traffic from subnet between H1 and R1 to the next hop 191.7.2.2 at r4-eth2

b. Trace route output - h1 traceroute h2