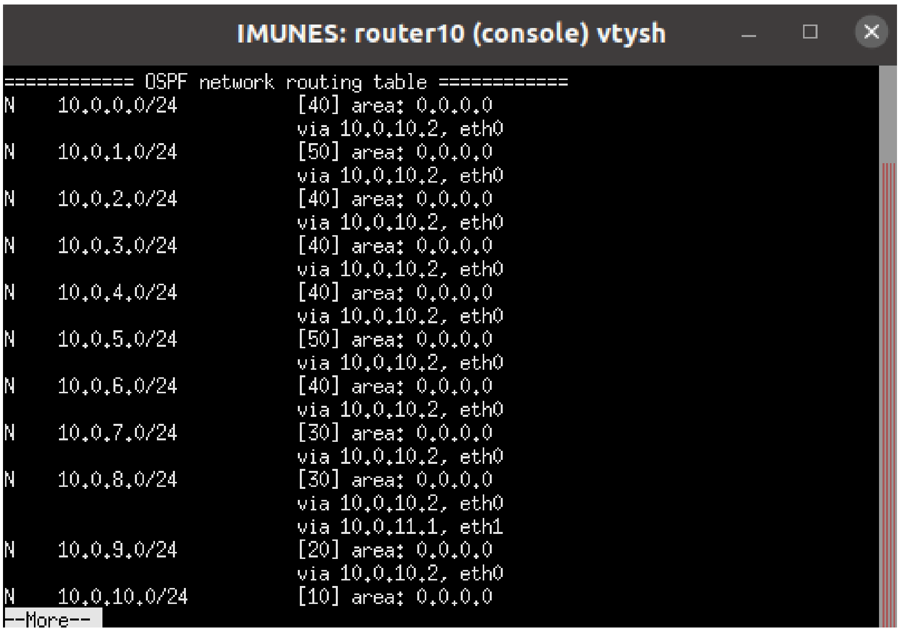
SCC.203 2nd Coursework: OSPF routing
Aleksandar Chalakov, 38038501

Task 1

a.

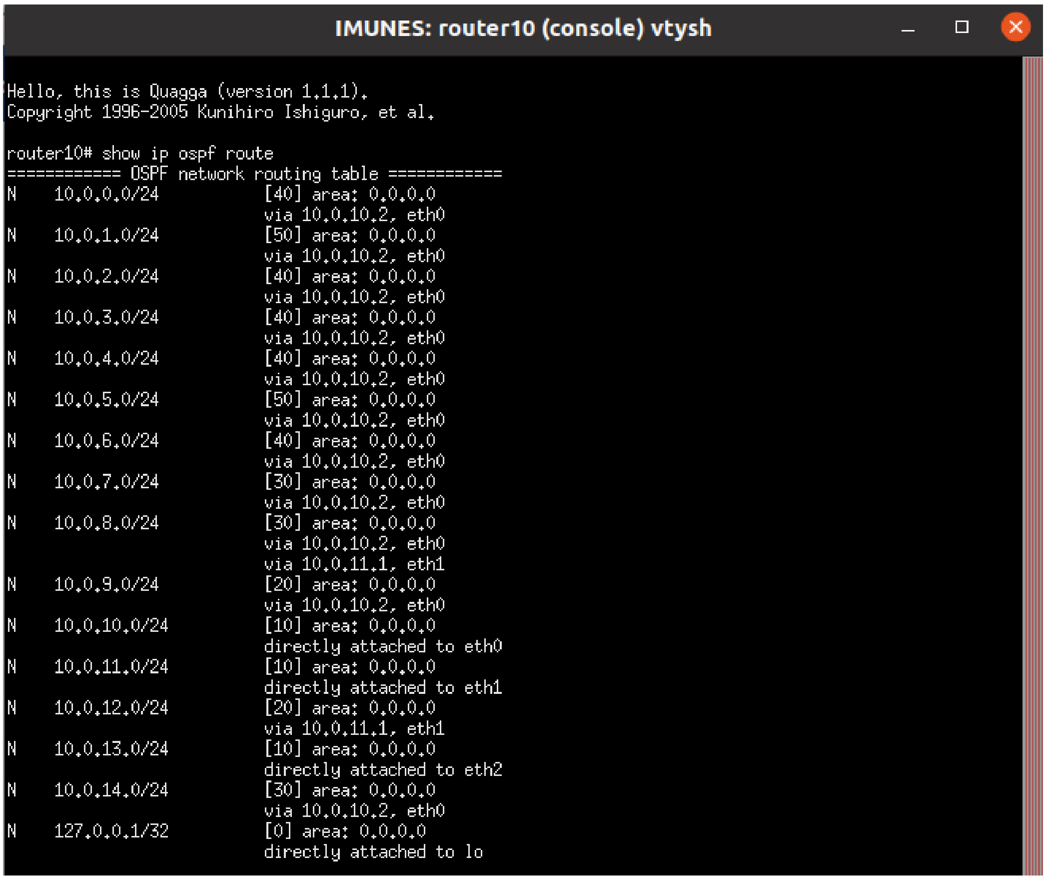
The first screenshot of the command I use to query the OSPF table - show ip ospf route:



IMUNES: router10 (console) vtysh

```
============ OSPF network routing table ============
N    10.0.0.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.1.0/24           [50] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.2.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.3.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.4.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.5.0/24           [50] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.6.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.7.0/24           [30] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.8.0/24           [30] area: 0.0.0.0
                           via 10.0.10.2, eth0
                           via 10.0.11.1, eth1
N    10.0.9.0/24           [20] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.10.0/24          [10] area: 0.0.0.0
--More--
```

This is the full table by getting all of the options:

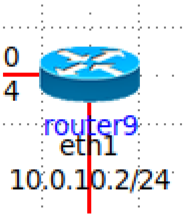


IMUNES: router10 (console) vtysh

```
Hello, this is Quagga (version 1.1.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router10# show ip ospf route
============ OSPF network routing table ============
N    10.0.0.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.1.0/24           [50] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.2.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.3.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.4.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.5.0/24           [50] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.6.0/24           [40] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.7.0/24           [30] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.8.0/24           [30] area: 0.0.0.0
                           via 10.0.10.2, eth0
                           via 10.0.11.1, eth1
N    10.0.9.0/24           [20] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    10.0.10.0/24          [10] area: 0.0.0.0
                           directly attached to eth0
N    10.0.11.0/24          [10] area: 0.0.0.0
                           directly attached to eth1
N    10.0.12.0/24          [20] area: 0.0.0.0
                           via 10.0.11.1, eth1
N    10.0.13.0/24          [10] area: 0.0.0.0
                           directly attached to eth2
N    10.0.14.0/24          [30] area: 0.0.0.0
                           via 10.0.10.2, eth0
N    127.0.0.1/32          [0] area: 0.0.0.0
                           directly attached to lo
```

b. Looking at the OSPF Query table, we find router 4 and its connection of 10.0.14.0, and see that the next hop of the IP address is going to be via 10.0.10.2, which is router 9:

```
N    10.0.14.0/24          [30] area: 0.0.0.0
                           via 10.0.10.2, eth0
```



c. Looking at the OSPF Query Table again, we can see from the screenshot below, there is a "[30]", which is the cost of the path:

```
N    10.0.14.0/24          [30] area: 0.0.0.0
                           via 10.0.10.2, eth0
```

d. In order to get the path installation times, we need to do *show ip route* in the vtysh shell window instead, because only it shows us what we need and the output can be seen in the screenshot below:

```
IMUNES: router10 (console) vtysh

Hello, this is Quagga (version 1.1.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router10# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O>* 10.0.0.0/24 [110/40] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.1.0/24 [110/50] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.2.0/24 [110/40] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.3.0/24 [110/40] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.4.0/24 [110/40] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.5.0/24 [110/50] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.6.0/24 [110/40] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.7.0/24 [110/30] via 10.0.10.2, eth0, 00:02:03
O>* 10.0.8.0/24 [110/30] via 10.0.10.2, eth0, 00:02:02
  *                      via 10.0.11.1, eth1, 00:02:02
O>* 10.0.9.0/24 [110/20] via 10.0.10.2, eth0, 00:02:12
O   10.0.10.0/24 [110/10] is directly connected, eth0, 00:02:56
C>* 10.0.10.0/24 is directly connected, eth0
O   10.0.11.0/24 [110/10] is directly connected, eth1, 00:02:56
C>* 10.0.11.0/24 is directly connected, eth1
O>* 10.0.12.0/24 [110/20] via 10.0.11.1, eth1, 00:02:02
O   10.0.13.0/24 [110/10] is directly connected, eth2, 00:02:56
C>* 10.0.13.0/24 is directly connected, eth2
O>* 10.0.14.0/24 [110/30] via 10.0.10.2, eth0, 00:02:03
C>* 127.0.0.0/8 is directly connected, lo
O>* 127.0.0.1/32 [110/0] is directly connected, lo, 00:02:56
```

As we can see, the path to the IP of router5 (eth0) is 00:02:03, on the connection of 10.0.1.0, while the path to the IP of router9 (eth0) is 00:02:56, which means that the path to router5 was installed before the one to router9, faster with 00:00:53 ms. Even if we want the eth0 for router 9, for which the time would be 00:02:12, it is still 00:00:09 ms faster. This is because the path first goes to the farside destination routers, establishes the connection with them, and only when it's coming back to the source, it goes back through router9 and establishes the path back to it.

Task 2

a. In order to measure the IP-levelpath, we need to traceroute pc2 from the terminal in pc1. Since the IP of pc2 is 10.0.5.20, we just do this:

```
traceroute to 10.0.5.20 (10.0.5.20), 30 hops max, 60 byte packets
 1  10.0.13.1 (10.0.13.1)  0.656 ms  0.570 ms  0.552 ms
 2  10.0.10.2 (10.0.10.2)  0.530 ms  0.486 ms  0.466 ms
 3  10.0.9.2 (10.0.9.2)  0.443 ms  0.392 ms  0.368 ms
 4  10.0.14.2 (10.0.14.2)  0.340 ms  0.291 ms  0.257 ms
 5  10.0.0.1 (10.0.0.1)  0.232 ms  0.188 ms  0.159 ms
 6  10.0.5.20 (10.0.5.20)  0.128 ms  0.145 ms  0.043 ms
root@pc1:/# traceroute 10.0.5.20
traceroute to 10.0.5.20 (10.0.5.20), 30 hops max, 60 byte packets
 1  10.0.13.1 (10.0.13.1)  0.439 ms  0.383 ms  0.370 ms
 2  10.0.10.2 (10.0.10.2)  0.357 ms  0.323 ms  0.309 ms
 3  10.0.9.2 (10.0.9.2)  0.293 ms  0.264 ms  0.246 ms
 4  10.0.14.2 (10.0.14.2)  0.228 ms  0.198 ms  0.178 ms
 5  10.0.0.1 (10.0.0.1)  0.157 ms  0.117 ms  0.097 ms
 6  10.0.5.20 (10.0.5.20)  0.074 ms  0.080 ms  0.034 ms
```

b. To measure the bandwidth between pc1 and pc2, we first initialize the iperf server in pc2 - (iperf -s 10.0.5.20), with the following screenshot:

IMUNES: pc2 (console) bash

```
root@pc2:/# iperf -s 10.0.5.20
iperf: ignoring extra argument -- 10.0.5.20
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 10.0.5.20 port 5001 connected with 10.0.13.20 port 55328
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  21.5 GBytes  18.5 Gbits/sec
```

And then execute the client mode of iperf - (iperf -c 10.0.5.20), connecting to the IP address of pc2:

IMUNES: pc1 (console) bash

```
root@pc1:/# iperf -c 10.0.5.20
------------------------------------------------------------
Client connecting to 10.0.5.20, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.0.13.20 port 55328 connected with 10.0.5.20 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  21.5 GBytes  18.5 Gbits/sec
root@pc1:/# 
```

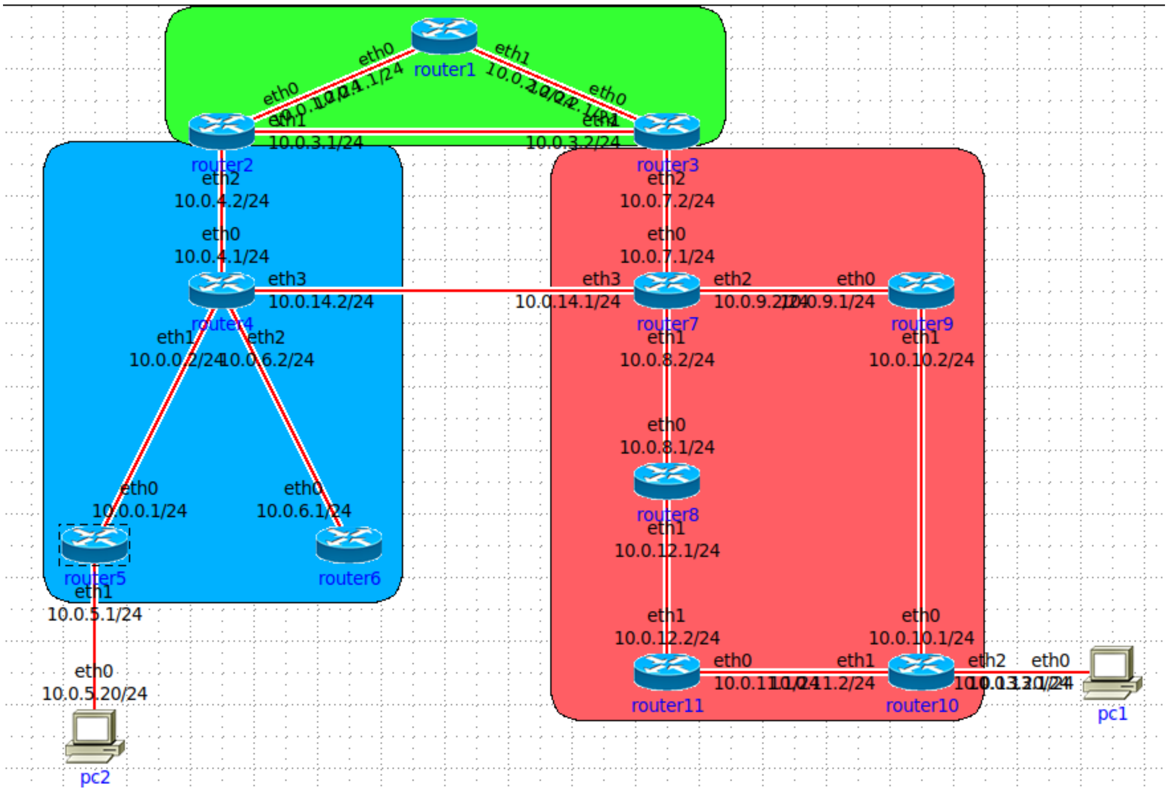
c. The sequence of routers is the following:
    1. 10.0.13.1 - Router 10
    2. 10.0.10.2 - Router 9
    3. 10.0.9.2 - Router 7
    4. 10.0.14.2 - Router 4
    5. 10.0.0.1 - Router 5
    6. being the pc2 - 10.0.5.20

d. To get the RTT between pc1 and pc2, we ping pc2 from pc1 and get that it is based on how many pings are processed, for 8 packets transmitted, I got that the **rtt min/avg/max/mdev = 0.093/0.117/0.176/0.028 ms**. Tested it out for 20 packets as well, and got the **rtt min/avg/max/mdev - 0.086/0.101/0.132/0.014 ms.**

```
root@pc1:/# ping 10.0.5.20
PING 10.0.5.20 (10.0.5.20) 56(84) bytes of data.
64 bytes from 10.0.5.20: icmp_seq=1 ttl=59 time=0.101 ms
64 bytes from 10.0.5.20: icmp_seq=2 ttl=59 time=0.093 ms
64 bytes from 10.0.5.20: icmp_seq=3 ttl=59 time=0.100 ms
64 bytes from 10.0.5.20: icmp_seq=4 ttl=59 time=0.115 ms
64 bytes from 10.0.5.20: icmp_seq=5 ttl=59 time=0.176 ms
64 bytes from 10.0.5.20: icmp_seq=6 ttl=59 time=0.135 ms
64 bytes from 10.0.5.20: icmp_seq=7 ttl=59 time=0.112 ms
64 bytes from 10.0.5.20: icmp_seq=8 ttl=59 time=0.110 ms
^C
--- 10.0.5.20 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7150ms
rtt min/avg/max/mdev = 0.093/0.117/0.176/0.028 ms
root@pc1:/# 
```

Task 3

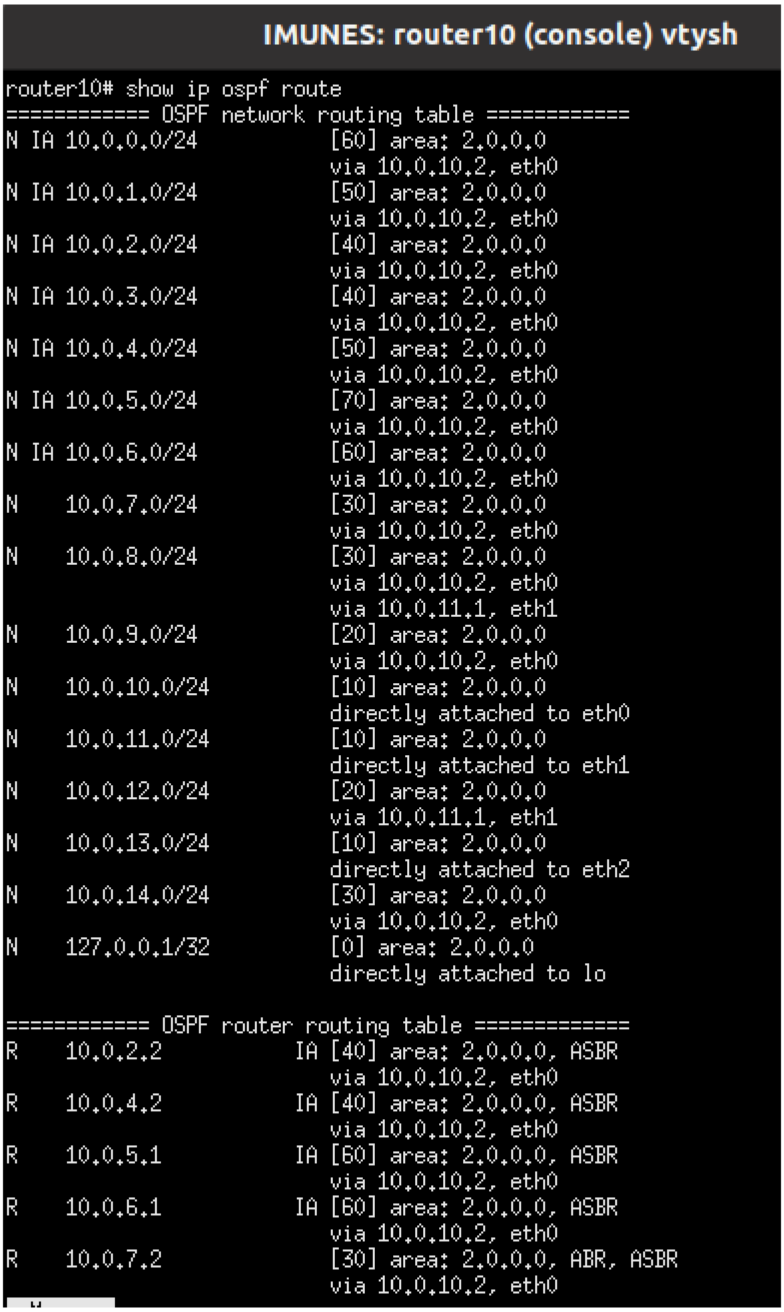
Areas done, backbone routers configured with networks of incoming router IPs so that connection is possible between areas:



Task 4

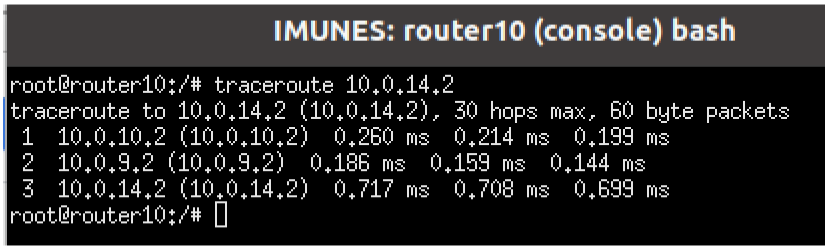
a. I'd say that the backbone area of the topology is the default area 0.0.0.0, with its router2 and router3 being backbone with respectively area 1.0.0.0 and area 2.0.0.0. Router 2 connects both with areas 0 and 1 via its network with router 4 - 10.0.4.2/24, while router 3 connects both with areas 0 and 2 via its network with router 7 - 10.0.7.2/24.

b. After querying again the OSPF routing table of router10 and comparing it to my answers in 1.b and 1.c, I can conclude that none of the values changed, and that is because even of the areas I've done now, the IP is still going to take its only path to router 4 (eth3), which has the same cost:



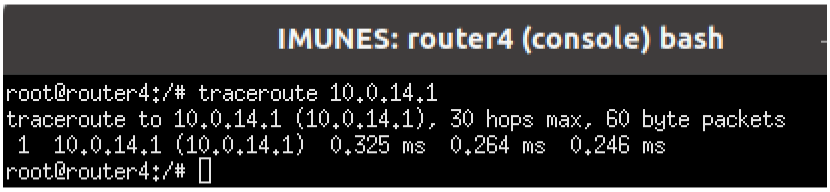
```
IMUNES: router10 (console) vtysh

router10# show ip ospf route
============ OSPF network routing table ============
N IA 10.0.0.0/24           [60] area: 2.0.0.0
                           via 10.0.10.2, eth0
N IA 10.0.1.0/24           [50] area: 2.0.0.0
                           via 10.0.10.2, eth0
N IA 10.0.2.0/24           [40] area: 2.0.0.0
                           via 10.0.10.2, eth0
N IA 10.0.3.0/24           [40] area: 2.0.0.0
                           via 10.0.10.2, eth0
N IA 10.0.4.0/24           [50] area: 2.0.0.0
                           via 10.0.10.2, eth0
N IA 10.0.5.0/24           [70] area: 2.0.0.0
                           via 10.0.10.2, eth0
N IA 10.0.6.0/24           [60] area: 2.0.0.0
                           via 10.0.10.2, eth0
N    10.0.7.0/24           [30] area: 2.0.0.0
                           via 10.0.10.2, eth0
N    10.0.8.0/24           [30] area: 2.0.0.0
                           via 10.0.10.2, eth0
                           via 10.0.11.1, eth1
N    10.0.9.0/24           [20] area: 2.0.0.0
                           via 10.0.10.2, eth0
N    10.0.10.0/24          [10] area: 2.0.0.0
                           directly attached to eth0
N    10.0.11.0/24          [10] area: 2.0.0.0
                           directly attached to eth1
N    10.0.12.0/24          [20] area: 2.0.0.0
                           via 10.0.11.1, eth1
N    10.0.13.0/24          [10] area: 2.0.0.0
                           directly attached to eth2
N    10.0.14.0/24          [30] area: 2.0.0.0
                           via 10.0.10.2, eth0
N    127.0.0.1/32          [0] area: 2.0.0.0
                           directly attached to lo

============ OSPF router routing table =============
R    10.0.2.2           IA [40] area: 2.0.0.0, ASBR
                           via 10.0.10.2, eth0
R    10.0.4.2           IA [40] area: 2.0.0.0, ASBR
                           via 10.0.10.2, eth0
R    10.0.5.1           IA [60] area: 2.0.0.0, ASBR
                           via 10.0.10.2, eth0
R    10.0.6.1           IA [60] area: 2.0.0.0, ASBR
                           via 10.0.10.2, eth0
R    10.0.7.2              [30] area: 2.0.0.0, ABR, ASBR
                           via 10.0.10.2, eth0
```

The cost stays [30]. If we wanted another interface of router 4, for instance, the eth0 being 10.0.4.1, the cost would have changed. However, we want the eth3 of router 4. The route is shown by the IP-levelpaths, when you traceroute, we can see the route it takes, and it goes through the direct link between router 7 and router 4 (eth3):



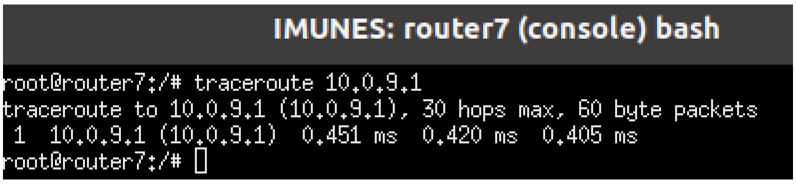
```
IMUNES: router10 (console) bash
root@router10:/# traceroute 10.0.14.2
traceroute to 10.0.14.2 (10.0.14.2), 30 hops max, 60 byte packets
 1  10.0.10.2 (10.0.10.2)  0.260 ms  0.214 ms  0.199 ms
 2  10.0.9.2 (10.0.9.2)  0.186 ms  0.159 ms  0.144 ms
 3  10.0.14.2 (10.0.14.2)  0.717 ms  0.708 ms  0.699 ms
root@router10:/#
```

c. Measuring the IP-levelpaths:
   i.



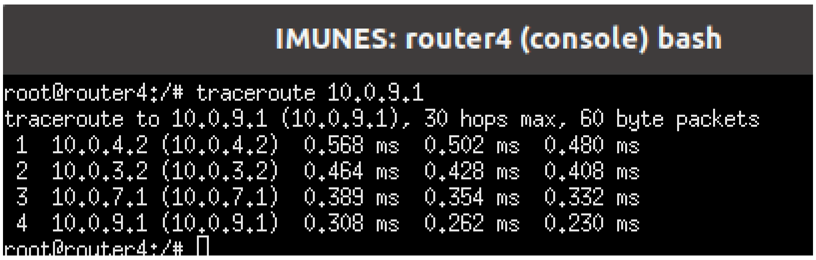
```
IMUNES: router4 (console) bash
root@router4:/# traceroute 10.0.14.1
traceroute to 10.0.14.1 (10.0.14.1), 30 hops max, 60 byte packets
 1  10.0.14.1 (10.0.14.1)  0.325 ms  0.264 ms  0.246 ms
root@router4:/#
```

   ii.



```
IMUNES: router7 (console) bash
root@router7:/# traceroute 10.0.9.1
traceroute to 10.0.9.1 (10.0.9.1), 30 hops max, 60 byte packets
 1  10.0.9.1 (10.0.9.1)  0.451 ms  0.420 ms  0.405 ms
root@router7:/#
```

   iii.

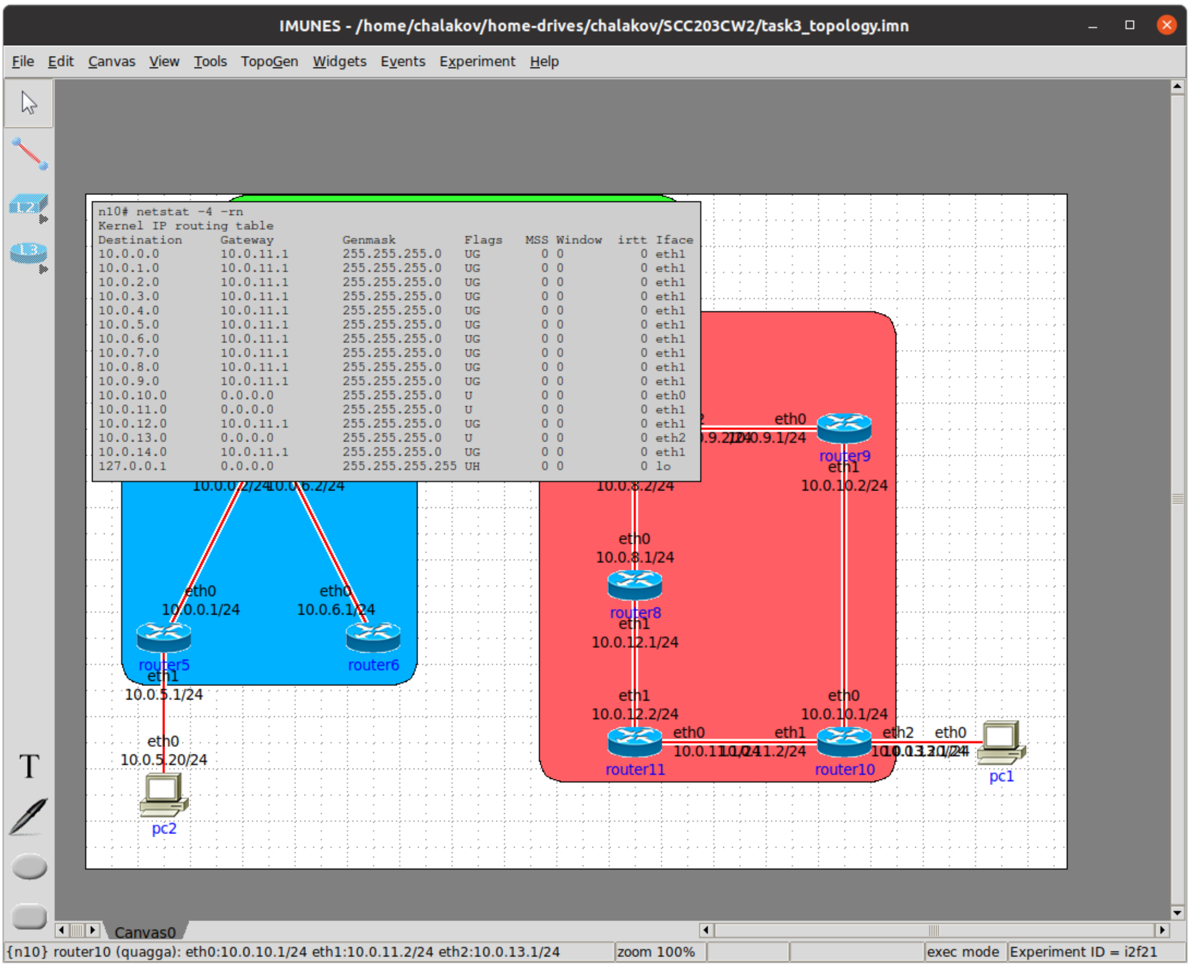


```
IMUNES: router4 (console) bash
root@router4:/# traceroute 10.0.9.1
traceroute to 10.0.9.1 (10.0.9.1), 30 hops max, 60 byte packets
 1  10.0.4.2 (10.0.4.2)  0.568 ms  0.502 ms  0.480 ms
 2  10.0.3.2 (10.0.3.2)  0.464 ms  0.428 ms  0.408 ms
 3  10.0.7.1 (10.0.7.1)  0.389 ms  0.354 ms  0.332 ms
 4  10.0.9.1 (10.0.9.1)  0.308 ms  0.262 ms  0.230 ms
root@router4:/#
```

The path in iii is not the same as the concatenation of path I and path ii for several reasons. First, the reason the path in i is a single line is because of the direct link between router 4 and router 7 (eth3) and should be noted that other routers cannot use this direct link. That's why in iii when we traceroute from router 4 to router 9, we have to go through our backbone area 0.0.0.0 and that's why we get a longer and different path, we go through router 2, router 3, router 7 and then finally router 9.

Task 5

a. Gateway being 10.0.11.1 shows that we successfully changed the next hop via router 11:



b. Showing the router hops for the two options:

i. from router 10 to router 5:

```
IMUNES: router10 (console) bash

root@router10:/# traceroute 10.0.0.1
traceroute to 10.0.0.1 (10.0.0.1), 30 hops max, 60 byte packets
 1  10.0.11.1 (10.0.11.1)  0.477 ms  0.423 ms  0.411 ms
 2  10.0.12.1 (10.0.12.1)  0.400 ms  0.369 ms  0.356 ms
 3  10.0.8.2 (10.0.8.2)  0.341 ms  0.308 ms  0.285 ms
 4  10.0.7.2 (10.0.7.2)  0.263 ms  0.229 ms  0.209 ms
 5  10.0.3.1 (10.0.3.1)  0.192 ms  0.162 ms  0.141 ms
 6  10.0.4.1 (10.0.4.1)  0.112 ms  0.763 ms  0.696 ms
 7  10.0.0.1 (10.0.0.1)  0.674 ms  0.635 ms  0.610 ms
root@router10:/#
```

1. Router 11
2. Router 8
3. Router 7
4. Router 3
5. Router 2
6. Router 4
7. Router 5

ii. from router 5 to router 10

```
root@router5:/# traceroute 10.0.13.1
traceroute to 10.0.13.1 (10.0.13.1), 30 hops max, 60 byte packets
 1  10.0.0.2 (10.0.0.2)  0.478 ms  0.415 ms  0.398 ms
 2  10.0.4.2 (10.0.4.2)  0.379 ms  0.349 ms  0.333 ms
 3  10.0.3.2 (10.0.3.2)  0.314 ms  0.282 ms  0.264 ms
 4  10.0.7.1 (10.0.7.1)  0.244 ms  0.210 ms  0.190 ms
 5  10.0.9.1 (10.0.9.1)  0.171 ms  0.134 ms  0.113 ms
 6  10.0.13.1 (10.0.13.1)  0.089 ms  0.083 ms  0.027 ms
```

1. Router 4
2. Router 2
3. Router 3
4. Router 7
5. Router 9
6. Router 10

It depends which eth link we traceroute to, however, both to eth 0 and eth2 the route would be shorter, they are not symmetrical or traversing through the same routers, since whenever we traceroute from router 10 via the eth0 connection, the cost would not be efficient, we fixed it manually to a big number so that the connection sees that it's not good to go, so it reroutes through router 11.

c. Yes. You just configure router 11 and router 8 to have additional costs. In my case, I've put ip ospf cost 1 for both of the interfaces on router 11 and ip ospf cost 2 for only one of the interfaces in router 8. This way when I run the program, the connection still goes through router 11, as the task says it should. I'm attaching show ip ospf route for evidence:

```
router10# show ip ospf route
============ OSPF network routing table ============
N IA 10.0.0.0/24          [53] area: 2.0.0.0
                          via 10.0.11.1, eth1
N IA 10.0.1.0/24          [43] area: 2.0.0.0
                          via 10.0.11.1, eth1
N IA 10.0.2.0/24          [33] area: 2.0.0.0
                          via 10.0.11.1, eth1
N IA 10.0.3.0/24          [33] area: 2.0.0.0
                          via 10.0.11.1, eth1
N IA 10.0.4.0/24          [43] area: 2.0.0.0
                          via 10.0.11.1, eth1
N IA 10.0.5.0/24          [63] area: 2.0.0.0
                          via 10.0.11.1, eth1
N IA 10.0.6.0/24          [53] area: 2.0.0.0
                          via 10.0.11.1, eth1
N    10.0.7.0/24          [23] area: 2.0.0.0
                          via 10.0.11.1, eth1
N    10.0.8.0/24          [13] area: 2.0.0.0
                          via 10.0.11.1, eth1
N    10.0.9.0/24          [20] area: 2.0.0.0
                          via 10.0.10.2, eth0
N    10.0.10.0/24         [10] area: 2.0.0.0
                          directly attached to eth0
N    10.0.11.0/24         [10] area: 2.0.0.0
                          directly attached to eth1
N    10.0.12.0/24         [11] area: 2.0.0.0
                          via 10.0.11.1, eth1
N    10.0.13.0/24         [10] area: 2.0.0.0
                          directly attached to eth2
N    10.0.14.0/24         [23] area: 2.0.0.0
                          via 10.0.11.1, eth1
```

Task 6

The task is done and a screenshot of the successful connection is provided below page: