Report

# Assignment 4

Network topology and routing

*Author:* Amata ANANTAPRAYOON
*Student ID:* aa224iu,
*Semester:* VT2019
*Course:* Computer Networks - an introduction
*Course code:* 1DV701

# 1 Problem 1

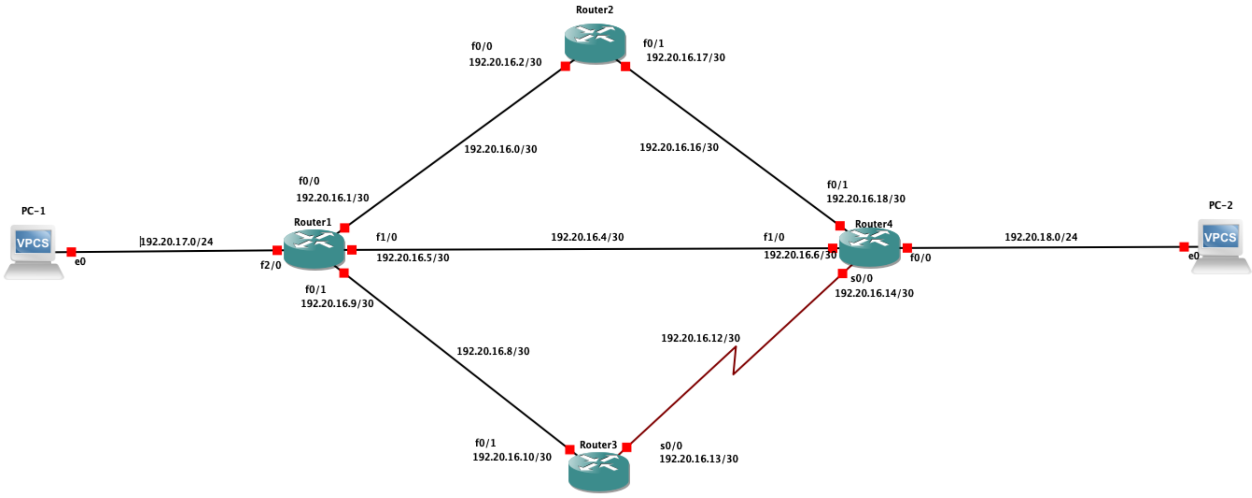



Figure 1: Network topology

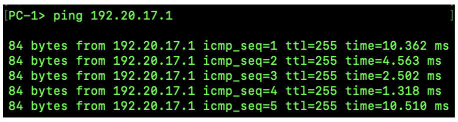



Figure 2: Ping from PC to R1

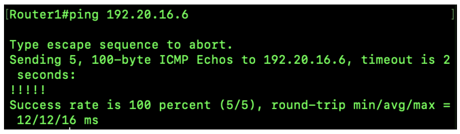



Figure 3: Ping from R1 to R4

- **NM-1FE-TX**:The c3725 has only 2 FastEthernet interfaces on its motherboard. According to the topology, R1 needs to be able to connect 4 networks which mean R1 needs 2 interfaces more. Also, both interfaces need to be able to carry traffic 10-100Mbit/s (fast ethernet). NM-1FE-TX provide1 fast ethernet port.
- **WIC-1T**: The c3725 does not have serial ports. The connection between R3 and R4 are a serial communication which is why we do need to add WIC-1T. We can also use NM-4T instead, it has 4 serial ports. But there is no need for 4 serial ports in this sensorial.

**255.255.255.252 or /30** contain 4 addresses, only 2 addresses can be assigned to hosts. In backbone, each network only has 2 hosts (router to router). That is why it is unnecessarily to use a bigger mask than /30. On the other hand, (255.255.255.0 or/ 24) contain 256 address (254 hosts). Network A and Network B are a LAN and it needs to be flexible. By flexible mean, both networks need to have an address pool that can be given to the hosts when needed. So that new user (PC) can just join the network and use one of the available addresses it directly.

# 2 Problem 2

```
enable
conf t
ip route [ip] [mask] [router_interface] [metric]
end
```

- $[ip]$ = the destination **network** ip address
- $[mask]$ = Netmask of destination network
- $[router\_interface]$ = can be the next hop's interface or ip address of the next hop interface.
- $[metric]$ = optional parameter, Administrative cost = 1 is set by default. The route that have lowest metric wins.

By studying the network topology, there is some important fact to be considered which are the bandwidth and hops per route, before configuring static routing. There are 3 difference paths:

- The upper route (R1 $\rightarrow$ R2 $\rightarrow$ R4): This route is the fastest route with 100 mbps in bandwidth.
- The middle route (R1 $\rightarrow$ R4) : This route has the lowest hops count (2 hop) with 10 mbps in bandwidth.
- The lower route (R1 $\rightarrow$ R3 $\rightarrow$ R4): 3 hops count as the upper route but it is the slowest route since the bandwidth between R3 and R4 is 1544 kbps.

I decided to assign the metric = 1 for the upper route since it is the fastest route. Follow by middle route with metric = 2. The lower route is the slowest route with 2 hop count which it why it should have the highest metric.

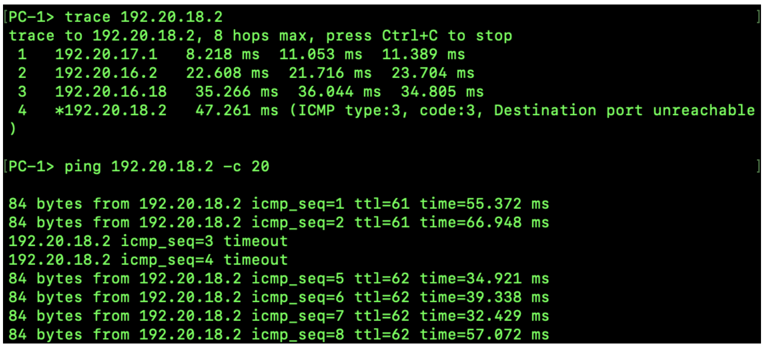



Figure 4: Traceroute + ping from PC 1 to PC 2

The figure above shows the route taken by the packet. As you can see the packet took a upper route, PC1 $\rightarrow$ R1 $\rightarrow$ R2 $\rightarrow$ R4 $\rightarrow$ PC2. The next figure, PC1 continuous ping PC2. I shutdown R1 interface f0/1 and R4 interface f0/1 after a while. The result is the route has been changed to middle route (PC1 $\rightarrow$ R1 $\rightarrow$ R4 $\rightarrow$ PC2), note that 2 packets are lost before the routers find a new route. One of the profs is TTL has increased by 1 (from 61 to 62). To be sure, I traceroute the packet again. Next figure shows traceroute after the path has changed.

```
PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   3.417 ms  11.220 ms  9.049 ms
 2   192.20.16.6   36.109 ms  35.018 ms  36.117 ms
 3   *192.20.18.2   45.171 ms (ICMP type:3, code:3, Destination port unreachable
)

PC-1> ping 192.20.18.2 -c 20

84 bytes from 192.20.18.2 icmp_seq=1 ttl=62 time=38.106 ms
84 bytes from 192.20.18.2 icmp_seq=2 ttl=62 time=35.614 ms
84 bytes from 192.20.18.2 icmp_seq=3 ttl=62 time=48.313 ms
192.20.18.2 icmp_seq=4 timeout
192.20.18.2 icmp_seq=5 timeout
84 bytes from 192.20.18.2 icmp_seq=6 ttl=61 time=34.032 ms
84 bytes from 192.20.18.2 icmp_seq=7 ttl=61 time=48.503 ms
84 bytes from 192.20.18.2 icmp_seq=8 ttl=61 time=38.163 ms
```

Figure 5: Traceroute + ping from PC 1 to PC 2

The figure above shows traceroute after I closed the connection for the upper route, followed by continuous ping from PC1 and PC2. I then close the connection for the middle route by close R1 interface f1/0 and R4 interface f1/0. Note that TTL has decreased from 62 to 61because the route have been changed to the lower route. The figure below shows traceroute PC1 to PC2.

```
PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   12.418 ms  11.301 ms  11.310 ms
 2   192.20.16.10   22.490 ms  23.696 ms  21.782 ms
 3   192.20.16.14   23.936 ms  23.738 ms  22.699 ms
 4   *192.20.18.2   36.062 ms (ICMP type:3, code:3, Destination port unreachable
)
```

Figure 6: Traceroute from PC1 to PC2

# 3 Problem 3

```
PC-1> ping 192.20.18.2

84 bytes from 192.20.18.2 icmp_seq=1 ttl=62 time=38.065 ms
84 bytes from 192.20.18.2 icmp_seq=2 ttl=62 time=40.403 ms
84 bytes from 192.20.18.2 icmp_seq=3 ttl=62 time=28.240 ms
84 bytes from 192.20.18.2 icmp_seq=4 ttl=62 time=37.843 ms
84 bytes from 192.20.18.2 icmp_seq=5 ttl=62 time=25.147 ms

PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   11.428 ms  11.286 ms  11.150 ms
 2   192.20.16.6   22.672 ms  22.625 ms  23.744 ms
 3   *192.20.18.2   36.153 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 7: Ping PC2 with RIPv2

Figure above shows which path the packet took to reach pc2. As you can see, the packet went through R2→R4 because the optimal paths by using RIP are base on hop count. PC1 → R1 → R4 → PC2 have a lowest hop count with 2 hops another paths have 3 hop count. Next step is run a continuous ping then shutdown R1 interface f0/0 and R3 interface f0/1. The figure below shows the result.

```
PC-1> ping 192.20.18.2 -c 99

192.20.18.2 icmp_seq=1 timeout
84 bytes from 192.20.18.2 icmp_seq=2 ttl=62 time=57.278 ms
84 bytes from 192.20.18.2 icmp_seq=3 ttl=62 time=37.153 ms
84 bytes from 192.20.18.2 icmp_seq=4 ttl=62 time=47.245 ms
84 bytes from 192.20.18.2 icmp_seq=5 ttl=62 time=59.348 ms
84 bytes from 192.20.18.2 icmp_seq=6 ttl=62 time=46.371 ms
84 bytes from 192.20.18.2 icmp_seq=7 ttl=62 time=39.410 ms
84 bytes from 192.20.18.2 icmp_seq=8 ttl=62 time=44.170 ms
192.20.18.2 icmp_seq=9 timeout
192.20.18.2 icmp_seq=10 timeout
84 bytes from 192.20.18.2 icmp_seq=11 ttl=61 time=59.374 ms
84 bytes from 192.20.18.2 icmp_seq=12 ttl=61 time=49.448 ms
84 bytes from 192.20.18.2 icmp_seq=13 ttl=61 time=51.662 ms
84 bytes from 192.20.18.2 icmp_seq=14 ttl=61 time=48.295 ms
^C
PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   9.026 ms  10.162 ms  10.251 ms
 2   192.20.16.2   36.012 ms  23.718 ms  35.737 ms
 3   192.20.16.18   34.686 ms  34.962 ms  33.672 ms
 4   *192.20.18.2   59.361 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 8: Ping from PC1 to PC, using RIPv2

As you see in figure 8, the packet went through the upper route( R1→R→R3) instead because the upper route has the highest bandwidth even though it has same hop count as the lower route. The figure below is a repetition of continuous ping PC2 + closing interface at the current route.

```
PC-1> ping 192.20.18.2 -c 99

192.20.18.2 icmp_seq=1 timeout
84 bytes from 192.20.18.2 icmp_seq=2 ttl=62 time=57.278 ms
84 bytes from 192.20.18.2 icmp_seq=3 ttl=62 time=37.153 ms
84 bytes from 192.20.18.2 icmp_seq=4 ttl=62 time=47.245 ms
84 bytes from 192.20.18.2 icmp_seq=5 ttl=62 time=59.348 ms
84 bytes from 192.20.18.2 icmp_seq=6 ttl=62 time=46.371 ms
84 bytes from 192.20.18.2 icmp_seq=7 ttl=62 time=39.410 ms
84 bytes from 192.20.18.2 icmp_seq=8 ttl=62 time=44.170 ms
192.20.18.2 icmp_seq=9 timeout
192.20.18.2 icmp_seq=10 timeout
84 bytes from 192.20.18.2 icmp_seq=11 ttl=61 time=59.374 ms
84 bytes from 192.20.18.2 icmp_seq=12 ttl=61 time=49.448 ms
84 bytes from 192.20.18.2 icmp_seq=13 ttl=61 time=51.662 ms
84 bytes from 192.20.18.2 icmp_seq=14 ttl=61 time=48.295 ms
^C
PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   9.026 ms  10.162 ms  10.251 ms
 2   192.20.16.2   36.012 ms  23.718 ms  35.737 ms
 3   192.20.16.18   34.686 ms  34.962 ms  33.672 ms
 4   *192.20.18.2   59.361 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 9: Ping PC2, lower route

# 4 Problem 4

I'm going to run the same process as the problem 3 but using OSPF instead. The figure below shows ping from PC1 to PC2 and traceroute. It shows that packet took the upper route which is the fastest route (100Mbit/s).

```
PC-1> ping 192.20.18.2

84 bytes from 192.20.18.2 icmp_seq=1 ttl=61 time=67.146 ms
84 bytes from 192.20.18.2 icmp_seq=2 ttl=61 time=69.809 ms
84 bytes from 192.20.18.2 icmp_seq=3 ttl=61 time=70.163 ms
84 bytes from 192.20.18.2 icmp_seq=4 ttl=61 time=66.143 ms
84 bytes from 192.20.18.2 icmp_seq=5 ttl=61 time=46.147 ms

PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   7.154 ms  11.731 ms  11.581 ms
 2   192.20.16.2   33.673 ms  35.031 ms  36.285 ms
 3   192.20.16.18   47.104 ms  45.155 ms  48.212 ms
 4   *192.20.18.2   60.614 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 10: Ping PC2, upper route

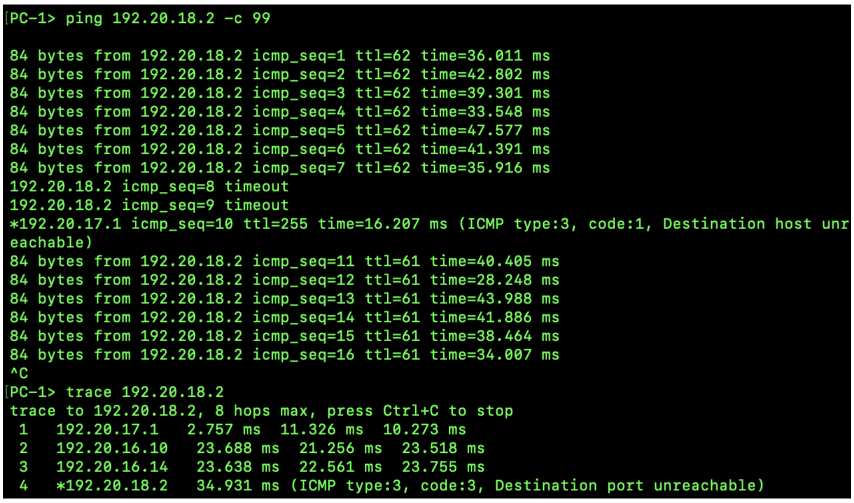
Next step is continuous ping from PC1 to PC2, followed by closing interface F0/0 on R1, interface f0/1 on R4. The figure below shows the result. Packet travel through the middle route because is the next fastest route (10Mbit/sec) with 2 hops count.

```
PC-1> ping 192.20.18.2 -c 99

84 bytes from 192.20.18.2 icmp_seq=1 ttl=61 time=84.433 ms
84 bytes from 192.20.18.2 icmp_seq=2 ttl=61 time=63.748 ms
84 bytes from 192.20.18.2 icmp_seq=3 ttl=61 time=49.136 ms
84 bytes from 192.20.18.2 icmp_seq=4 ttl=61 time=60.726 ms
192.20.18.2 icmp_seq=5 timeout
192.20.18.2 icmp_seq=6 timeout
*192.20.17.1 icmp_seq=7 ttl=255 time=18.124 ms (ICMP type:3, code:1, Destination host unre
achable)
*192.20.17.1 icmp_seq=8 ttl=255 time=4.031 ms (ICMP type:3, code:1, Destination host unrea
chable)
84 bytes from 192.20.18.2 icmp_seq=9 ttl=62 time=39.601 ms
84 bytes from 192.20.18.2 icmp_seq=10 ttl=62 time=26.028 ms
84 bytes from 192.20.18.2 icmp_seq=11 ttl=62 time=45.106 ms
84 bytes from 192.20.18.2 icmp_seq=12 ttl=62 time=27.169 ms
84 bytes from 192.20.18.2 icmp_seq=13 ttl=62 time=24.974 ms
84 bytes from 192.20.18.2 icmp_seq=14 ttl=62 time=23.762 ms
^C
PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   1.234 ms  11.453 ms  11.489 ms
 2   192.20.16.6   36.026 ms  36.040 ms  34.737 ms
 3   *192.20.18.2   48.562 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 11: Ping from PC1 to PC2, middle route is taken

The last step is a repetition of continuous ping PC2 and closing interfaces current route taken by the packet, follow by traceroute PC2. Figure 12 shows that the packet takes the lowest route with is the worse route.


```
PC-1> ping 192.20.18.2 -c 99

84 bytes from 192.20.18.2 icmp_seq=1 ttl=62 time=36.011 ms
84 bytes from 192.20.18.2 icmp_seq=2 ttl=62 time=42.802 ms
84 bytes from 192.20.18.2 icmp_seq=3 ttl=62 time=39.301 ms
84 bytes from 192.20.18.2 icmp_seq=4 ttl=62 time=33.548 ms
84 bytes from 192.20.18.2 icmp_seq=5 ttl=62 time=47.577 ms
84 bytes from 192.20.18.2 icmp_seq=6 ttl=62 time=41.391 ms
84 bytes from 192.20.18.2 icmp_seq=7 ttl=62 time=35.916 ms
192.20.18.2 icmp_seq=8 timeout
192.20.18.2 icmp_seq=9 timeout
*192.20.17.1 icmp_seq=10 ttl=255 time=16.207 ms (ICMP type:3, code:1, Destination host unr
eachable)
84 bytes from 192.20.18.2 icmp_seq=11 ttl=61 time=40.405 ms
84 bytes from 192.20.18.2 icmp_seq=12 ttl=61 time=28.248 ms
84 bytes from 192.20.18.2 icmp_seq=13 ttl=61 time=43.988 ms
84 bytes from 192.20.18.2 icmp_seq=14 ttl=61 time=41.886 ms
84 bytes from 192.20.18.2 icmp_seq=15 ttl=61 time=38.464 ms
84 bytes from 192.20.18.2 icmp_seq=16 ttl=61 time=34.007 ms
^C
PC-1> trace 192.20.18.2
trace to 192.20.18.2, 8 hops max, press Ctrl+C to stop
 1   192.20.17.1   2.757 ms  11.326 ms  10.273 ms
 2   192.20.16.10   23.688 ms  21.256 ms  23.518 ms
 3   192.20.16.14   23.638 ms  22.561 ms  23.755 ms
 4   *192.20.18.2   34.931 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 12: Ping from PC1 to PC2, lowest route is taken

# 5 Static VS RIP VS OSPF

- **Static routing** is straightforward, the administrator needs to assign it manually. There are some advantage and disadvantage. Since it needs to be configured manually means it is not flexible. Imagine if one of the routers is down, this might cause a black-hole or a loop. Static routing can be used for Route filtering and it can be really helpful to manipulate the network flows. There is some case that we might want the flow to travel through a specific path, static routing can provide this possibility
- **Routing Information Protocol** (RIP) is one of the distance-vector routing protocol. RIP determines the best route for data packets based on hops count. It uses a Bellman-Ford Distance Vector algorithm to calculate the best path. It is slow to fault-discovery and rerouting since it periodic sending updated routing table which can cost a routing loop. Also, it sends out the full routing table every periodic update means more bandwidth needs.
- **Open Shortest Path First** (OSPF) is one of the link-state routing protocols. It using a Dijkstra's algorithm to calculate the shortest path to each destination. Each router keeps track of the link states of its neighbours. Whenever there is a change, the routers update the changes and re-calculate the shortest path. Which is why OSPF is fast to fault-discovery and rerouting. Although it demands higher processing and memory than RIP since each router needs to generate the shortest path