Linnéuniversitetet
Kalmar Växjö

# Network Topology & Routing

# Assignment 4

*1DV701*

*Author: Rashed Qazizada*
*Semester:* Spring 2021
*Email*
rq222ah@student.lnu.se

# Table of Contents

# 1 Problem

Network Topology

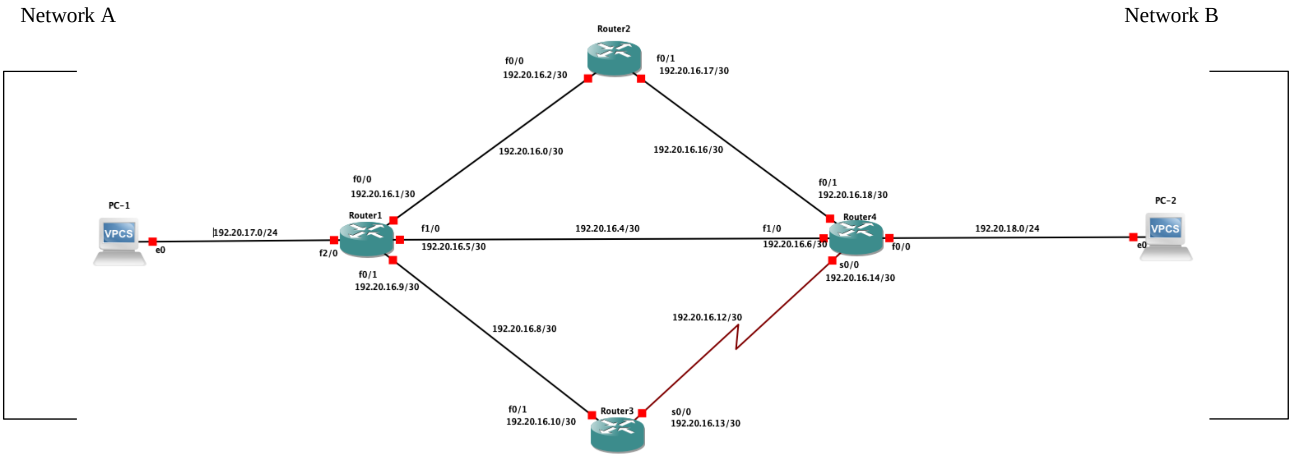
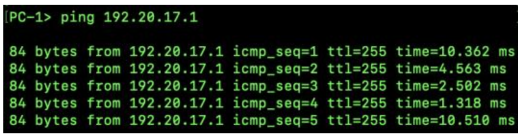


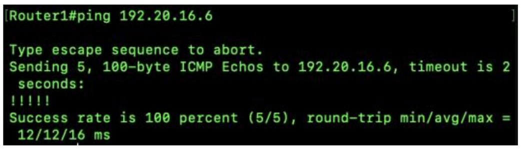

*Figure 1 Network Topology*



```
PC-1> ping 192.20.17.1

84 bytes from 192.20.17.1 icmp_seq=1 ttl=255 time=10.362 ms
84 bytes from 192.20.17.1 icmp_seq=2 ttl=255 time=4.563 ms
84 bytes from 192.20.17.1 icmp_seq=3 ttl=255 time=2.502 ms
84 bytes from 192.20.17.1 icmp_seq=4 ttl=255 time=1.318 ms
84 bytes from 192.20.17.1 icmp_seq=5 ttl=255 time=10.510 ms
```

*Figure 2 R1 receives a ping from the PC*



```
Router1#ping 192.20.16.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.20.16.6, timeout is 2
 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
 12/12/16 ms
```

*Figure 3 R4 receives a ping from the R1.*

## 1.1 NM-1FE-TX

On the motherboard of the c3725, there are only two FastEthernet interfaces. Based on the topology, Router 1 (R1) must be qualified to link four networks, which means R1 would need two additional interfaces. Furthermore, all interfaces must be capable of carrying traffic at speeds of 10 to 100 megabits per second (fast ethernet). One FastEthernet port is provided by the NM-1FE-TX.

## 1.2 WIC-1T

There are no serial ports on the c3725. Serial ports are not available on the c3725. Since the link among R3 & R4 is serial, we'll need to use

WIC-1T. We may also use the NM-4T, which has four serial ports. However, within that sensorial, four serial ports are unnecessary.

Just two of the four addresses used in 255.255.255.252 (/30) may be allocated to hosts. Every network has only two hosts in the backend (Router (R) to Router (R)). As a result, using a mask larger than /30 is unnecessary. However (255.255.255.0 (/ 24)) contains 256 addresses (254 hosts). A LAN is made up of Network A & B, and it must be adaptable. Via versatile means, all networks must have a pool of addresses that could be assigned to hosts as appropriate. As a result, a new user will simply connect to a network. In addition to that the user uses one of the free addresses.

# 2 Problem

Static Routing

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

Ip: the ip address of the target network.
Mask: The target network's netmask.
Router_Interface: could be the interface of another hop or the IP address with the next hops interface.
Metric: 1 is equal to administrator cost and it is the default value for this optional parameter. The path with the lowest metric is the winner.

Until setting up static routing, it's essential to look at the network topology and remember some key facts including bandwidth (connectivity) and hops / network (path). There are three different directions to take:

- The topmost (upper) route (R1=>R2=>R4): is the shortest and of course the fastest, with a bandwidth of 100 megabits per second.
- The middle path (route)(R1=>R4) has the fewest and lowest hops (2 hops) and the most bandwidth (10 mbps).
- The lower path (route) (R1=>R3=>R4) counts as the topmost (upper) route since it has three hops.

Since the upper route is the shortest, I wanted to set the metric to 1. Then take the middle path, which has a metric of 2. For two hops, the lower path is the slowest, which is why it have the highest metric.

```
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   8.218 ms  11.053 ms  11.389 ms
 2   192.20.16.2   22.608 ms  21.716 ms  23.704 ms
 3   192.20.16.18   35.266 ms  36.044 ms  34.805 ms
 4   *192.20.18.2   47.261 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=61 time=55.372 ms
84 bytes from 192.20.18.2 icmp_seq=2 ttl=61 time=66.948 ms
192.20.18.2 icmp_seq=3 timeout
192.20.18.2 icmp_seq=4 timeout
84 bytes from 192.20.18.2 icmp_seq=5 ttl=62 time=34.921 ms
84 bytes from 192.20.18.2 icmp_seq=6 ttl=62 time=39.338 ms
84 bytes from 192.20.18.2 icmp_seq=7 ttl=62 time=32.429 ms
84 bytes from 192.20.18.2 icmp_seq=8 ttl=62 time=57.072 ms
```

*Figure 4 Ping and Traceroute from PC1 to PC2*

The packet's path is shown in the diagram above. The packet, you see, took a higher direction (the upper route), PC1=>R1 => R2 => R4 =>PC 2. PC 1 continues to ping PC 2 in next figure.
After a bit, I turned off f0/1 R1 interface and R4 f0/1 interface. The route was already modified to a middle route, (PC 1 => R1 =>R4 => PC 2) but two packets have been lost even before routers reach a new route. TTL, is one of several proof that has expanded by one.  fr. 61 – 62

```
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 Ping and Traceroute from PC1 to PC2*

Traceroute after closing (the upper) the topmost route link, followed by persistent ping fr. PC 1 and PC 2. R1 f1/0 interface and R4 f1/0 interface are then closed to close the middle path (route) link. Since the route was moved to the lowest route, TTL is decreased fr. 62 – 61. PC 1 to PC 2 traceroute as seen in the diagram below.

```
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 fr. PC 1 to PC 2*

# 3 Problem

```
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 Using the RIPv2 protocol, send a ping to PC2.*

The route taken by the packet to enter pc 2 is depicted in the diagram above. As you observe, the packet passed through R2 => R4 since RIP's optimum paths are determined by hop count. R1 => R4 PC1 => R1 => R4 PC2 has the fewest hops (2), while the other routes have 3 hops. Conduct a regular ping after that, then shut down R1 f0/0 interface and R3 f0/1 interface. The outcome is shown in the diagram below.

```
[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 Using RIPv2, ping from PC 1 to PC.*

Figure 8 shows that the packet was routed via the upper(topmost) route (R1 => R=> R3) instead of the lower (shortest) route since the upper (topmost) route does have bandwidth while having a certain number of hops. The image below shows a regular ping PC 2 and closing the interface on the current path, the route repeated many times.

```
[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 PC 2, Shortest, the lower route*

# 4 Problem

I'll repeat the procedure as in problem 3, However this time with OSPF. Ping from PC 1 to PC 2 and traceroute are seen in the following diagram. It indicates that the packet has taken the shortest (100Mbit/s) path.

```
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 PC 2, the topmost, the upper route*

The next step could be to ping continuously from PC 1 to PC 2, then close f0/0 interface on R1 and f0/1 interface on R4. Which the outcome is shown in the diagram below. Since it's the next shortest path (10Mbit/sec) with two hops, packets pass along the middle route.

```
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 The middle path is taken while pinging from PC 1 to PC 2.*

The final move is to repeat the repeated ping PC 2 and close the interfaces which the packet is currently using, followed via traceroute PC 2. The packet can take the shortest, "lowest" route, which would be the worse, the bad route, as seen in Figure 12.

```
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 The lowest route is taken while pinging from PC 1 to PC 2.*

# 5 Problem

## The main distinctions between the 3 routing strategies (static/RIPv2/OSPF) are as follows:

- **Static routing** is simple to set up, but it must be assigned manually by the administrator. There are a few benefits and drawbacks. It's not really flexible so it must be programmed manually. Consider what would happen if one of routers went down, resulting in a black hole or a circle. Static routing is useful for the route filtering and manipulating network flows. In certain cases, we can wish for the flows to follow a certain direction; static routing may accommodate this request.
- **The RIP (Routing Info Protocol** is a distance-vector routing protocol. Based on the number of hops, RIP decides the best path, the route for packets of data. It calculates the best direction using the Bellman-Ford Distance Vector algorithm. The algorithm is slow in fault detection and rerouting because it sends modified routing tables on a regular basis, which can trigger a routing cycle or loop. Furthermore, it starts sending out the entire routing table with each periodic update, requiring more bandwidth.
- **The OSPF Open Shortest Path First** is one of several link-state routing protocols is OSPF. It computes the shortest route to every destination using Dijkstra's algorithm. Every router keeps a record of its neighbours’ connection states. The routers refresh the updates and adjust the shortest path if anything changes. As a result, OSPF is faster to detect and reroute faults. Since every router must produce the shortest path, it requires more memory and processing unlike RIP.