Linnéuniversitetet
Kalmar Växjö

Report

Assignment 4

Network topology and routing

Authour: Mohammed Basel Nasrini
Course: 1DV701
Semester: Spring 2018

Problem 1

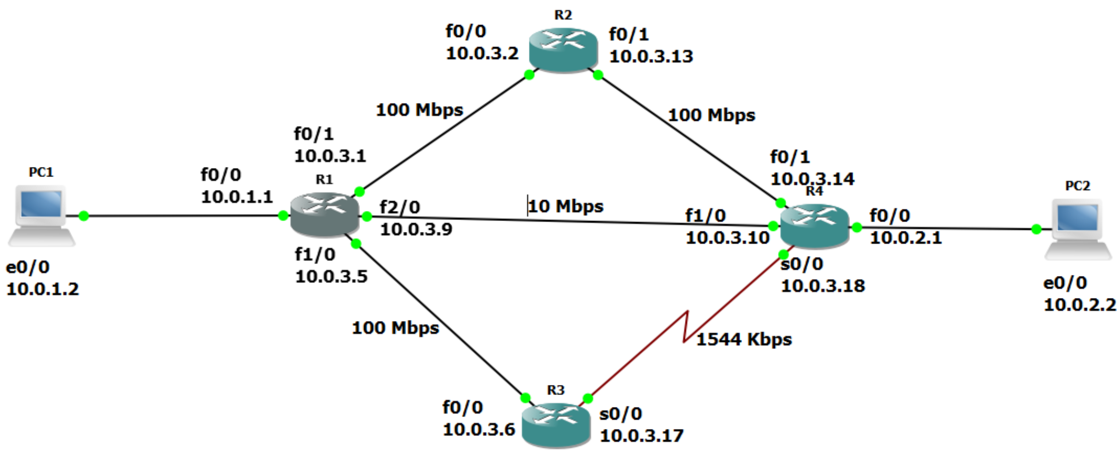
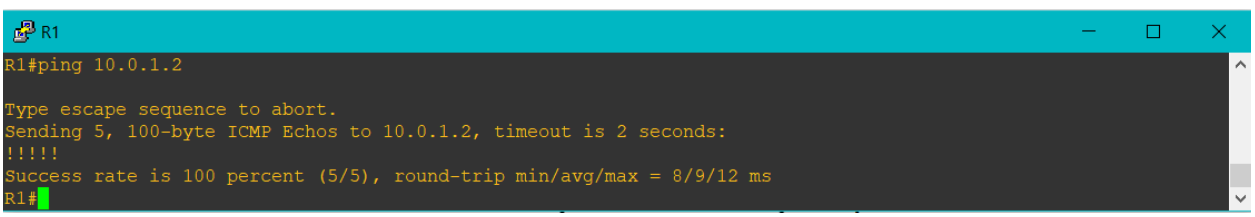


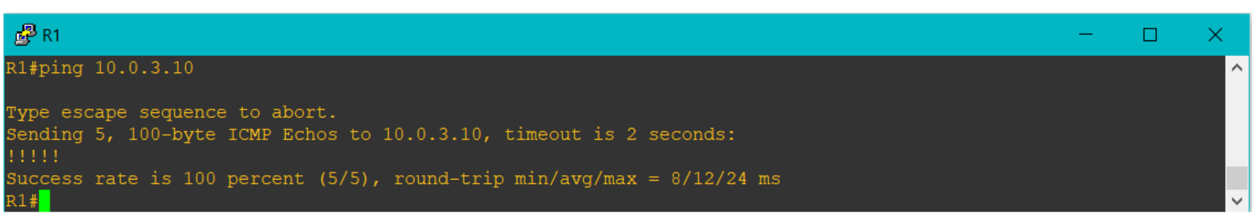
Figure 1 - The topology used in the assignment from GNS3



```
R1
R1#ping 10.0.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
R1#
```

Figure 2 - Ping from R1 to C1



```
R1
R1#ping 10.0.3.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.3.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/24 ms
R1#
```

Figure 3 - Ping from R1 to R4

NM-1FE-TX

It is a 1-port Fast Ethernet network module offers a single auto-sensing 10/100TX connection via an RJ-45 connector. It supports many internetworking features including virtual LAN (VLAN) deployment.

Why NM-1FE-TX?
- NM-16ESW is a switching module with 16 fast ethernet ports and we do not need here.
- NM-4T supports sync mode only.
- NM-NAM does not have some features we need, like the external console port on the NM-NAM means that the initial boot configuration is possible only through the router.

WIC-1T
The 1-port serial WAN interface card (WIC-1T) provides serial connections to remote sites or legacy serial network device

Why WIC-1T?
WIC-2T has to 2 ports and a sync max speed of 8 Mbps per port that we do not need.

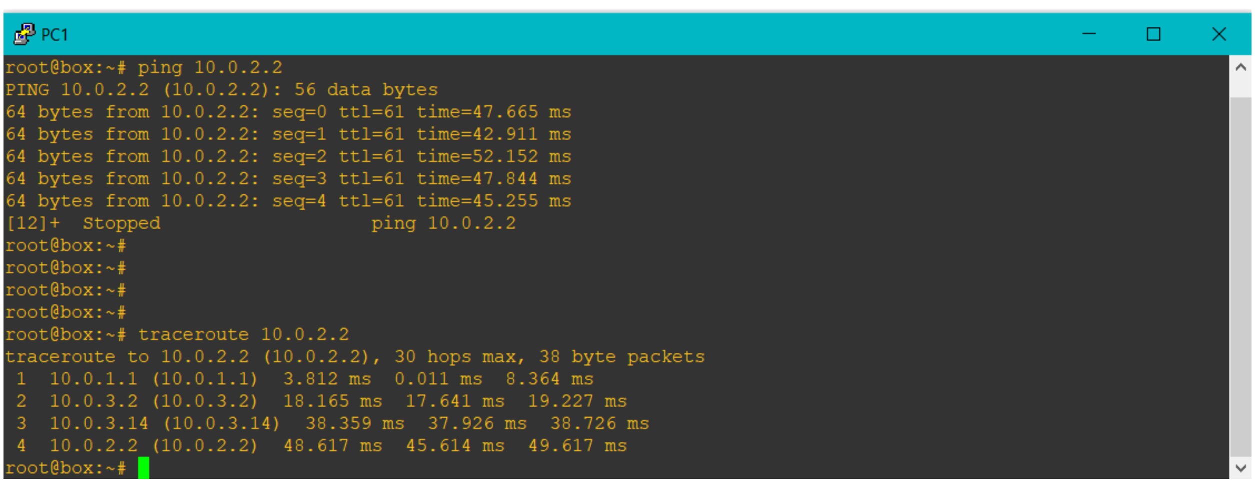
Practical difference between /24 and /30 subnet
/30 subnet does not waste any IP address when it comes to a point of point network.

Problem 2

The explanation of *ip route* command:

```
ip route [ip] [mask] [router_interface] [metric]
```

`[ip]:` The ip address of the next connected hop
`[mask]:` The subnet mask of the ip
`[router_interface]:` The output interface of the router
`[metric]:` An integer that decides the priority of the route



```
PC1
root@box:~# ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: seq=0 ttl=61 time=47.665 ms
64 bytes from 10.0.2.2: seq=1 ttl=61 time=42.911 ms
64 bytes from 10.0.2.2: seq=2 ttl=61 time=52.152 ms
64 bytes from 10.0.2.2: seq=3 ttl=61 time=47.844 ms
64 bytes from 10.0.2.2: seq=4 ttl=61 time=45.255 ms
[12]+  Stopped                    ping 10.0.2.2
root@box:~#
root@box:~#
root@box:~#
root@box:~#
root@box:~# traceroute 10.0.2.2
traceroute to 10.0.2.2 (10.0.2.2), 30 hops max, 38 byte packets
 1  10.0.1.1 (10.0.1.1)  3.812 ms  0.011 ms  8.364 ms
 2  10.0.3.2 (10.0.3.2)  18.165 ms  17.641 ms  19.227 ms
 3  10.0.3.14 (10.0.3.14)  38.359 ms  37.926 ms  38.726 ms
 4  10.0.2.2 (10.0.2.2)  48.617 ms  45.614 ms  49.617 ms
root@box:~#
```

Figure 4 - Ping from C1 to C2 and the path that token

The path I chose is C1 > R1 > R2 > R4 > C2 because of the speed of this path in compare to the speed of the other paths which are slower.

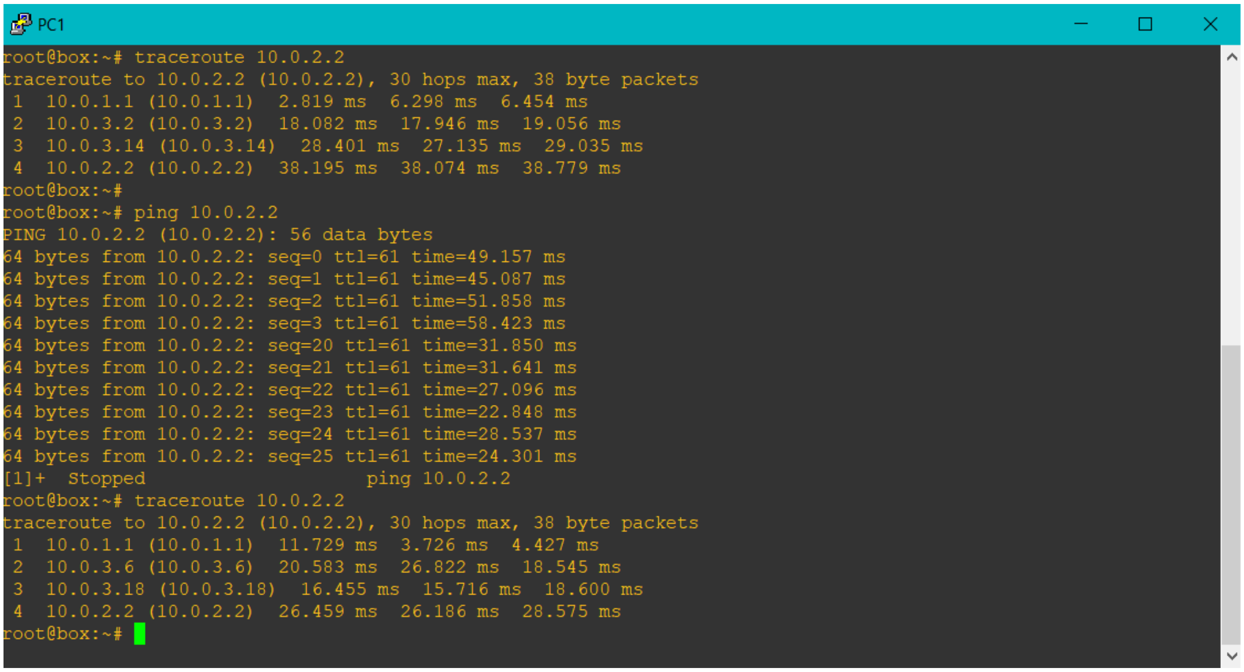


Figure 5 - Ping from C1 to C2 with a shout-off of the path showing the paths before and after the shut-off

We can see that 7 packets are lost. By using `traceroute` command we can see that the route has been changed from C1 > R1 > R2 > R4 > C2 to C1 > R1 > R3 > R4 > C2 and by working the ping again after shutting off the route.

VG-task 1

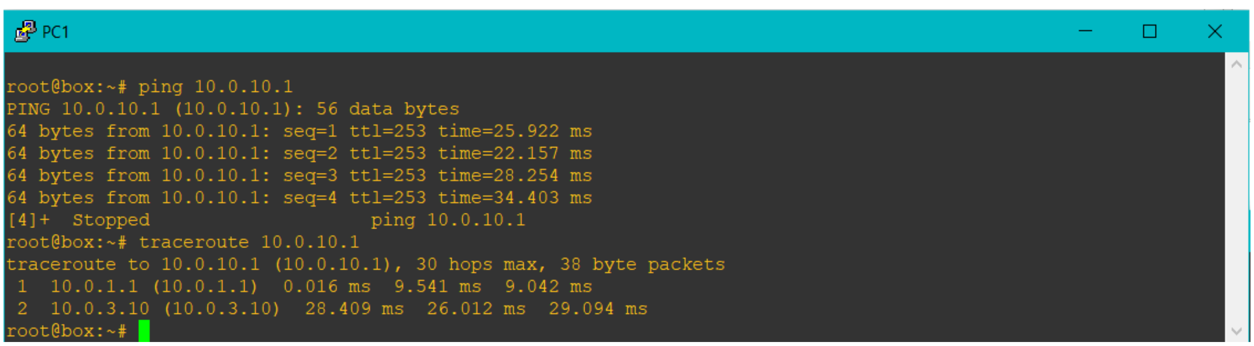


Figure 6 - Ping from C1 to a loopback interface

A Screenshot shows a continuous ping from C1 to the loopback interface added to R4

Problem 3

```
PC1
root@box:~# ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: seq=1 ttl=61 time=42.941 ms
64 bytes from 10.0.2.2: seq=2 ttl=61 time=37.938 ms
64 bytes from 10.0.2.2: seq=3 ttl=61 time=33.382 ms
64 bytes from 10.0.2.2: seq=4 ttl=61 time=48.943 ms
64 bytes from 10.0.2.2: seq=5 ttl=61 time=36.602 ms
[2]+  Stopped                    ping 10.0.2.2
root@box:~# traceroute 10.0.2.2
traceroute to 10.0.2.2 (10.0.2.2), 30 hops max, 38 byte packets
 1  10.0.1.1 (10.0.1.1)  5.058 ms  7.087 ms  8.713 ms
 2  10.0.3.10 (10.0.3.10)  16.162 ms  16.742 ms  19.090 ms
 3  10.0.2.2 (10.0.2.2)  28.011 ms  27.643 ms  29.421 ms
root@box:~#
```

Figure 7 - Ping from C1 to C2 using RIPv2

A ping from C1 to C2 using a dynamic routing protocol RIPv2. It takes the path C1 > R1 > R4 > C2

```
PC1
root@box:~# ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: seq=0 ttl=62 time=25.296 ms
64 bytes from 10.0.2.2: seq=1 ttl=62 time=23.614 ms
64 bytes from 10.0.2.2: seq=2 ttl=62 time=29.548 ms
64 bytes from 10.0.2.2: seq=3 ttl=62 time=25.400 ms
64 bytes from 10.0.2.2: seq=237 ttl=61 time=40.004 ms
64 bytes from 10.0.2.2: seq=239 ttl=61 time=33.465 ms
64 bytes from 10.0.2.2: seq=241 ttl=61 time=51.804 ms
64 bytes from 10.0.2.2: seq=243 ttl=61 time=35.987 ms
64 bytes from 10.0.2.2: seq=245 ttl=61 time=37.998 ms
64 bytes from 10.0.2.2: seq=247 ttl=61 time=39.787 ms
[6]+  Stopped                    ping 10.0.2.2
root@box:~#
root@box:~# traceroute 10.0.2.2
traceroute to 10.0.2.2 (10.0.2.2), 30 hops max, 38 byte packets
 1  10.0.1.1 (10.0.1.1)  5.515 ms  0.017 ms  6.683 ms
 2  10.0.3.6 (10.0.3.6)  28.265 ms  25.627 ms  29.169 ms
 3  10.0.3.18 (10.0.3.18)  28.107 ms  26.204 ms  29.215 ms
 4  10.0.2.2 (10.0.2.2)  28.130 ms  28.052 ms  29.101 ms
root@box:~#
```

Figure 8 - Ping from C1 to C2 using RIPv2 and the route after shutting down an interface

A screenshot showing the ping from C1 to C2 with a shut down of an interface while the ping and `traceroute` command shows the new path which is C1 > R1 > R3 > R4 > C2. We can see that about 230 packets are lost until take a new route path.

The RIPPv2 is **Distance Vector Routing Protocols** that use router hop counts as their metrics. That it takes the shortest path from the source to the destination. Firstly, it took C1 > R1 > R4 > C2 but after shutting of the interface in the R1 it took C1 > R1 > R3 > R4 > C2.

VG-task 1

```
PC1
root@box:~# ping 10.0.10.1
PING 10.0.10.1 (10.0.10.1): 56 data bytes
64 bytes from 10.0.10.1: seq=0 ttl=254 time=24.431 ms
64 bytes from 10.0.10.1: seq=1 ttl=254 time=24.871 ms
64 bytes from 10.0.10.1: seq=2 ttl=254 time=23.690 ms
64 bytes from 10.0.10.1: seq=3 ttl=254 time=24.956 ms
[1]+  Stopped                    ping 10.0.10.1
root@box:~# traceroute 10.0.10.1
traceroute to 10.0.10.1 (10.0.10.1), 30 hops max, 38 byte packets
 1  10.0.1.1 (10.0.1.1)  4.796 ms  2.898 ms  2.799 ms
 2  10.0.3.10 (10.0.3.10)  25.047 ms  23.674 ms  23.755 ms
root@box:~#
```

Figure 9 - Ping from C1 to a loopback interface with RIPv2

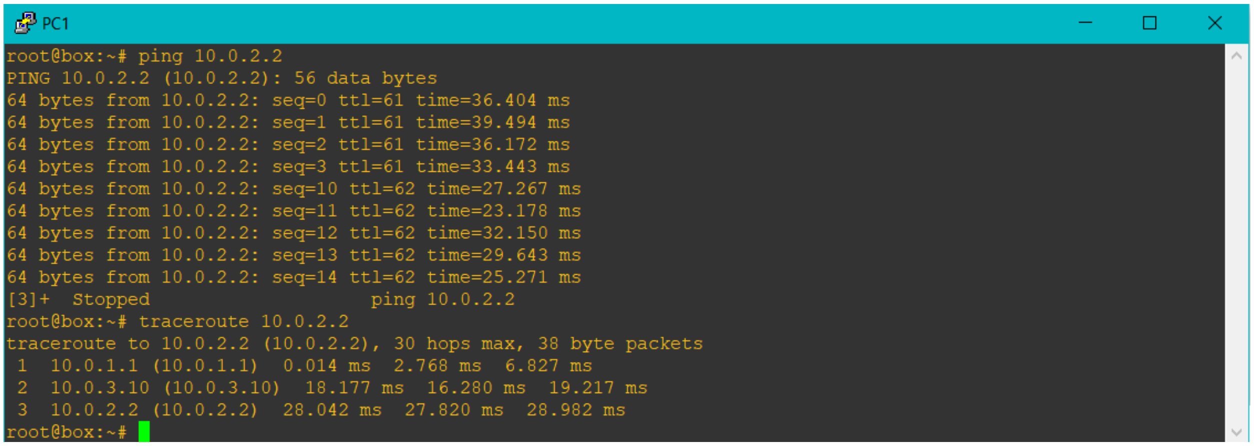
A Screenshot shows a continuous ping from C1 to the loopback interface added to R4 with the use of RIPv2 protocol.

Problem 4

```
PC1
root@box:~# ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: seq=0 ttl=62 time=26.039 ms
64 bytes from 10.0.2.2: seq=1 ttl=62 time=32.227 ms
64 bytes from 10.0.2.2: seq=2 ttl=62 time=27.918 ms
64 bytes from 10.0.2.2: seq=3 ttl=61 time=33.883 ms
64 bytes from 10.0.2.2: seq=4 ttl=61 time=39.103 ms
64 bytes from 10.0.2.2: seq=5 ttl=61 time=36.263 ms
[2]+  Stopped                    ping 10.0.2.2
root@box:~# traceroute 10.0.2.2
traceroute to 10.0.2.2 (10.0.2.2), 30 hops max, 38 byte packets
 1  10.0.1.1 (10.0.1.1)  10.674 ms  2.930 ms  5.156 ms
 2  10.0.3.2 (10.0.3.2)  17.108 ms  16.318 ms  14.422 ms
 3  10.0.3.14 (10.0.3.14)  28.206 ms  27.334 ms  29.084 ms
 4  10.0.2.2 (10.0.2.2)  38.129 ms  37.171 ms  39.260 ms
root@box:~#
```

Figure 10 - Ping from C1 to C2 using OSPF

A ping from C1 to C2 using a dynamic routing protocol OSPF. It takes the path C1 > R1 > R2 > R4 > C2.
The protocol uses Dijkstra algorithm to decide the shortest path to the destination and to calculate the OSPF cost (metric) for each interface. A higher bandwidth indicates a lower cost. So, the protocol takes the fastest path to the destination first.



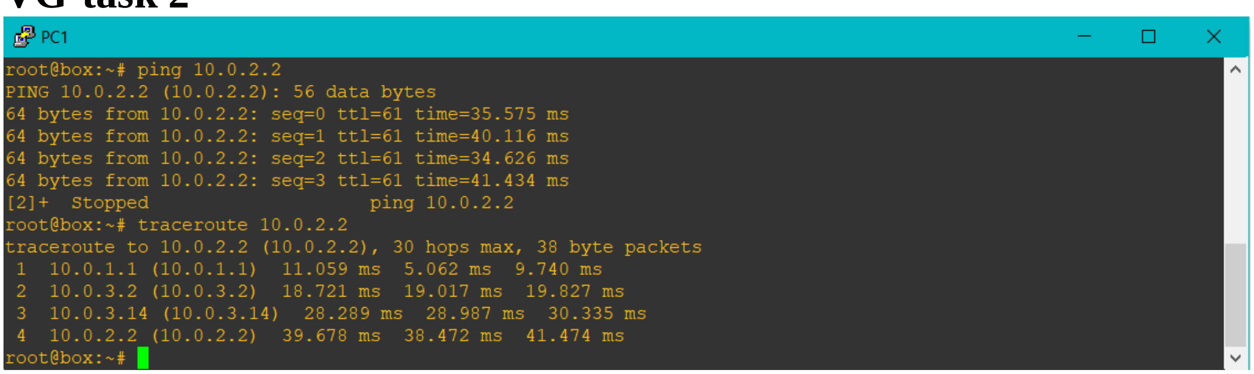
```
PC1
root@box:~# ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: seq=0 ttl=61 time=36.404 ms
64 bytes from 10.0.2.2: seq=1 ttl=61 time=39.494 ms
64 bytes from 10.0.2.2: seq=2 ttl=61 time=36.172 ms
64 bytes from 10.0.2.2: seq=3 ttl=61 time=33.443 ms
64 bytes from 10.0.2.2: seq=10 ttl=62 time=27.267 ms
64 bytes from 10.0.2.2: seq=11 ttl=62 time=23.178 ms
64 bytes from 10.0.2.2: seq=12 ttl=62 time=32.150 ms
64 bytes from 10.0.2.2: seq=13 ttl=62 time=29.643 ms
64 bytes from 10.0.2.2: seq=14 ttl=62 time=25.271 ms
[3]+  Stopped                    ping 10.0.2.2
root@box:~# traceroute 10.0.2.2
traceroute to 10.0.2.2 (10.0.2.2), 30 hops max, 38 byte packets
 1  10.0.1.1 (10.0.1.1)  0.014 ms  2.768 ms  6.827 ms
 2  10.0.3.10 (10.0.3.10)  18.177 ms  16.280 ms  19.217 ms
 3  10.0.2.2 (10.0.2.2)  28.042 ms  27.820 ms  28.982 ms
root@box:~#
```

Figure 11- Figure 8 - Ping from C1 to C2 using OSPF and the route after shutting down an interface

A screenshot showing the ping from C1 to C2 using OSPF with a shut-down of an interface while the ping and `traceroute` command shows the new path which is C1 > R1 > R4 > C2.

Static	RIPv2	OSPF
Based on the user configuration (High chance of human error)	Based on distance vector approach	Use link-state algorithm
Easy to configure	Easiest to configure	Hard to configure
Small networks and with a simple string topology with no redundancy	When there are redundant paths.	When there are redundant paths.
Low number of lost packets in case of failure. (17 packets in my example above)	High number of lost packets in case of failure. (About 230 packets in my example above)	Lowest number of lost packets in case of failure. (7 packets in my example above)

VG-task 2



```
PC1
root@box:~# ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: seq=0 ttl=61 time=35.575 ms
64 bytes from 10.0.2.2: seq=1 ttl=61 time=40.116 ms
64 bytes from 10.0.2.2: seq=2 ttl=61 time=34.626 ms
64 bytes from 10.0.2.2: seq=3 ttl=61 time=41.434 ms
[2]+  Stopped                    ping 10.0.2.2
root@box:~# traceroute 10.0.2.2
traceroute to 10.0.2.2 (10.0.2.2), 30 hops max, 38 byte packets
 1  10.0.1.1 (10.0.1.1)  11.059 ms  5.062 ms  9.740 ms
 2  10.0.3.2 (10.0.3.2)  18.721 ms  19.017 ms  19.827 ms
 3  10.0.3.14 (10.0.3.14)  28.289 ms  28.987 ms  30.335 ms
 4  10.0.2.2 (10.0.2.2)  39.678 ms  38.472 ms  41.474 ms
root@box:~#
```

Figure 12 - Ping from C1 to C2 using EIGRP

A ping from C1 to C2 using a dynamic routing protocol EIGRP. It takes the path C1 > R1 > R2 > R4 > C2.

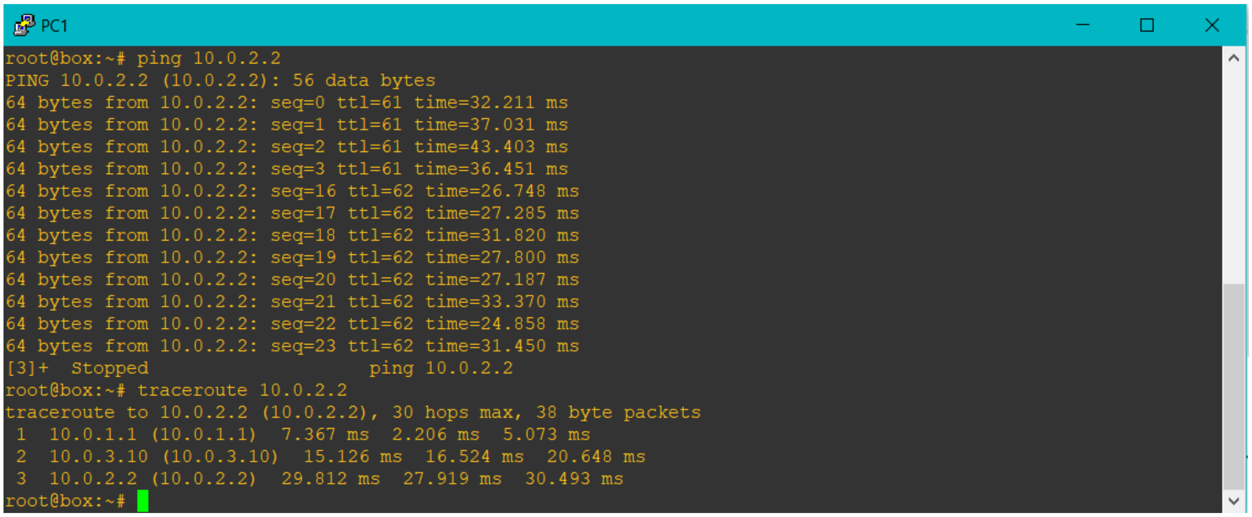


Figure 13 - Ping from C1 to C2 using EIGRP and the route after shutting down an interface

A screenshot showing the ping from C1 to C2 using EIGRP with a shut-down of an interface while the ping and `traceroute` command shows the new path which is C1 > R1 > R4 > C2.

Choosing the path with EIGRP (EIGRP metrics) depends on the minimum bandwidth on the path to a destination network and the total delay to compute routing metrics. The bandwidth and delay metrics are determined from values configured on the interfaces of routers in the path to the destination network.

One can configure other metrics, but it could cause routing loops in the network.

Distance Vector (EIGRP)	Link State (OSPF)
DUAL algorithm	Dijkstra algorithm
Best for simple, flat design, non-hierarchal networks, less scalable	Best for large, hierarchal networks, scalable
Minimum administrator knowledge	Advanced administrator knowledge
Knowledge of the network directly from the neighbors	Routers have a complete overview of the network (the entire topology)
Send periodic updates of entire routing table	Send triggered partial updates
Convergence time is not an issue	Convergence time is crucial
We lose the visibility to decide the path	It allows us to decide the path through the network by being aware of all links in a given logical topology (an OSPF area).