

Otto-Friedrich-University of Bamberg

Professorship for Computer Science,  
Communication Services, Telecommunication  
Systems and Computer Networks



## Foundation of Internet Communication

### Assignment 5 Intradomain Dynamic Routing Protocols

Submitted by:

**Group X**

Moktahid Al Faisal  
Abdullah Al Mosabbir  
Mohammed Mehedi Hasan  
Kazi Sayef Shawgat  
Sheikh Jumon Ahmed

Supervisor: Prof. Dr. Udo Krieger

Bamberg, July 5, 2020  
Summer Term 2020

## Contents

<b>1</b>	<b>Interior Routing with OSPF</b>	<b>1</b>
<b>2</b>	<b>Dynamic Routing with RIP</b>	<b>7</b>

## List of Figures

1	lab config file (a) . . . . .	1
2	lab config file (b) . . . . .	1
3	lab config file (c) . . . . .	2
4	lab config file (d) . . . . .	2
5	deamons file forall routers . . . . .	2
6	ospfd.conf file for core1 . . . . .	3
7	ospfd.conf file for ospf1 . . . . .	3
8	wiresark CD Y . . . . .	3
9	wireshark capture CD Y . . . . .	4
10	traceroute from bifur to balin . . . . .	4
11	ping from bifur to balin . . . . .	4
12	Captures on CD Y with Wireshark . . . . .	6
13	routing table of core1 . . . . .	6
14	routing table of core1 . . . . .	7
15	routing table of core2 . . . . .	7
16	routing table of core3 . . . . .	7
17	lab config file (a) . . . . .	8
18	lab config file (b) . . . . .	8
19	lab config file (c) . . . . .	8
20	ping bombur, kili, and dori from bofur . . . . .	9
21	Disable the eth2 interface of rip5 initially . . . . .	9
22	Wireshark capture on CD O . . . . .	9
23	Traceroute to explore . . . . .	10
24	Ping to bombur . . . . .	10

25	Enable the eth2 interface . . . . .	10
26	RIP protocol on the basis of the captured packets(a) . . . . .	10
27	Captured wireshark . . . . .	11
28	Link between rip5 and rip3 again by disabling the particular inter-face(a) . . . . .	11
29	Link between rip5 and rip3 again by disabling the particular inter-face(b) . . . . .	11
30	Link between rip5 and rip3 again by disabling the particular inter-face(c) . . . . .	11
31	Link between rip5 and rip3 again by disabling the particular inter-face(d) . . . . .	12
32	Link between rip5 and rip3 again by disabling the particular inter-face(e) . . . . .	12
33	Link between rip5 and rip3 again by disabling the particular inter-face(f) . . . . .	12
34	Link between rip5 and rip3 again by disabling the particular inter-face(g) . . . . .	13
35	Link between rip5 and rip3 again by disabling the particular inter-face(h) . . . . .	13

# 1 Interior Routing with OSPF

1. *Build the topology depicted in Figure 1 with Kathara.*

```
1 LAB_DESCRIPTION="lab.conf for Assignment-5 Task-1"
2 LAB_VERSION=1.0
3 LAB_AUTHOR="Abdullah Al Mosabbir"
4 LAB_EMAIL="abdullah-al.mosabbir@stud.uni-bamberg.de"
5
6 bifur[0]="J"
7 bifur[image]="unibaktr/alpine:whoami"
8 bifur[exec]="ifconfig eth0 11.0.0.10 netmask 255.248.0.0 up"
9 bifur[exec]="ip route add default via 11.0.0.11"
10
11 balin[0]="K"
12 balin[image]="unibaktr/alpine:whoami"
13 balin[exec]="ifconfig eth0 12.0.0.20 netmask 255.255.248.0 up"
14 balin[exec]="ip route add default via 12.0.0.25"
15
16 killi[0]="I"
17 killi[image]="unibaktr/alpine:latest"
18 killi[exec]="ifconfig eth0 13.0.0.30 netmask 255.255.248.0 up"
19 killi[exec]="ip route add default via 13.0.0.3"
20
21 core1[0]="Z"
22 core1[1]="X"
23 core1[2]="A"
24 core1[3]="B"
25 core1[image]="unibaktr/alpine:frr"
26 core1[exec]="ifconfig eth0 10.1.1.9 netmask 255.255.255.252 up"
27 core1[exec]="ifconfig eth1 10.1.1.1 netmask 255.255.255.252 up"
28 core1[exec]="ifconfig eth2 11.0.0.1 netmask 255.254.0.0 up"
29 core1[exec]="ifconfig eth3 11.4.0.1 netmask 255.255.192.0 up"
30 core1[exec]="frrinit.sh start"
```

Figure 1: lab config file (a)

```
32 core2[0]="Z"
33 core2[1]="Y"
34 core2[2]="E"
35 core2[3]="F"
36 core2[image]="unibaktr/alpine:frr"
37 core2[exec]="ifconfig eth0 10.1.1.10 netmask 255.255.255.252 up"
38 core2[exec]="ifconfig eth1 10.1.1.5 netmask 255.255.255.252 up"
39 core2[exec]="ifconfig eth2 12.0.0.2 netmask 255.255.254.0 up"
40 core2[exec]="ifconfig eth3 12.0.4.2 netmask 255.255.255.128 up"
41 core2[exec]="frrinit.sh start"
42
43 core3[0]="X"
44 core3[1]="Y"
45 core3[2]="I"
46 core3[image]="unibaktr/alpine:frr"
47 core3[exec]="ifconfig eth0 10.1.1.2 netmask 255.255.255.252 up"
48 core3[exec]="ifconfig eth1 10.1.1.6 netmask 255.255.255.252 up"
49 core3[exec]="ifconfig eth2 13.0.0.3 netmask 255.255.248.0 up"
50 core3[exec]="frrinit.sh start"
51
52 ospf1[0]="C"
53 ospf1[1]="B"
54 ospf1[2]="J"
55 ospf1[image]="unibaktr/alpine:frr"
56 ospf1[exec]="ifconfig eth0 11.4.64.11 netmask 255.255.255.240 up"
57 ospf1[exec]="ifconfig eth1 11.2.0.11 netmask 255.255.0.0 up"
58 ospf1[exec]="ifconfig eth2 11.0.0.11 netmask 255.248.0.0 up"
59 ospf1[exec]="frrinit.sh start"
```

Figure 2: lab config file (b)

```

60 ospf2[0]="A"
61 ospf2[1]="C"
62 ospf2[image]="unibaktr/alpine:frr"
63 ospf2[exec]="ifconfig eth0 11.0.0.12 netmask 255.254.0.0 up"
64 ospf2[exec]="ifconfig eth1 11.4.0.12 netmask 255.255.240.0 up"
65 ospf2[exec]="frrinit.sh start"
66
67
68 ospf3[0]="B"
69 ospf3[1]="D"
70 ospf3[image]="unibaktr/alpine:frr"
71 ospf3[exec]="ifconfig eth0 11.4.0.13 netmask 255.255.192.0 up"
72 ospf3[exec]="ifconfig eth1 11.2.0.13 netmask 255.255.0.0 up"
73 ospf3[exec]="frrinit.sh start"
74
75 ospf4[0]="E"
76 ospf4[1]="G"
77 ospf4[image]="unibaktr/alpine:frr"
78 ospf4[exec]="ifconfig eth0 12.0.0.24 netmask 255.255.254.0 up"
79 ospf4[exec]="ifconfig eth1 12.0.0.24 netmask 255.255.255.0 up"
80 ospf4[exec]="frrinit.sh start"
81
82 ospf5[0]="D"
83 ospf5[1]="H"
84 ospf5[2]="K"
85 ospf5[image]="unibaktr/alpine:frr"
86 ospf5[exec]="ifconfig eth0 12.0.0.25 netmask 255.255.255.0 up"
87 ospf5[exec]="ifconfig eth1 12.0.0.25 netmask 255.255.254.0 up"
88 ospf5[exec]="ifconfig eth2 12.0.0.25 netmask 255.255.240.0 up"
89 ospf5[exec]="frrinit.sh start"

```

Figure 3: lab config file (c)

```

90
91 ospf6[0]="F"
92 ospf6[1]="H"
93 ospf6[image]="unibaktr/alpine:frr"
94 ospf6[exec]="ifconfig eth0 12.0.0.26 netmask 255.255.255.128 up"
95 ospf6[exec]="ifconfig eth1 12.0.0.26 netmask 255.255.254.0 up"
96 ospf6[exec]="frrinit.sh start"
97

```

Figure 4: lab config file (d)

2. *Enable and configure OSPF on all routers. Let the coreX routers form the back-bone area, place ospf1 to ospf3 in one stub area and ospf4 to ospf6 in another one. Explain the concept of different areas in OSPF. What is the purpose to use different areas?*

A directory `\etc\frr\` is created with two files `daemons` and `ospfd.conf` for each routers. For example:

```

≡ daemons x
core1 > etc > frr > ≡ daemons
1  zebra=yes
2  bgpd=no
3  ospfd=yes
4  ospf6d=no
5  ripd=no
6  ripngd=no

```

Figure 5: deamons file forall routers

```

etc > frr > ospfd.conf
router ospf
  network 11.0.0.0/15 area 1.1.1.1
  network 11.4.0.0/18 area 1.1.1.1
  network 10.1.1.0/28 area 0.0.0.0
  area 1.1.1.1 stub
  redistribute connected

```

Figure 6: ospfd.conf file for core1

```

etc > frr > ospfd.conf
router ospf
  network 11.0.0.0/12 area 1.1.1.1
  area 1.1.1.1 stub
  redistribute connected

```

Figure 7: ospfd.conf file for ospf1

**Areas:** An OSPF network can be divided into sub-domains called areas. An area is a logical collection of OSPF networks, routers, and links that have the same area identification. A router within an area must maintain a topological database for the area to which it belongs. The router does not have detailed information about network topology outside of its area, which thereby reduces the size of its database.

### 3. *With wireshark start to capture traffic on CD Y.*

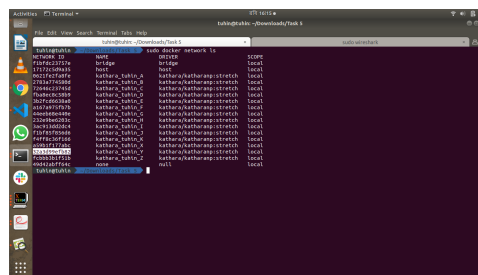


Figure 8: wireshark CD Y

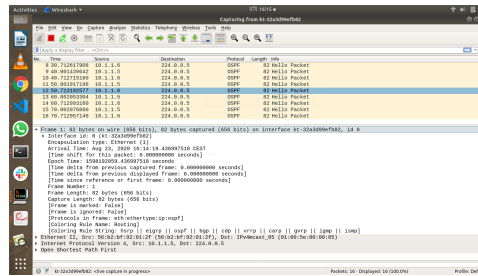


Figure 9: wireshark capture CD Y

4. *From bifur, run a traceroute to balin. Determine whether the path includes ospf2 or ospf3 and ospf4 or ospf6.*

The following figure shows the route from bifur to balin. The route includes router ospf3 and ospf6.

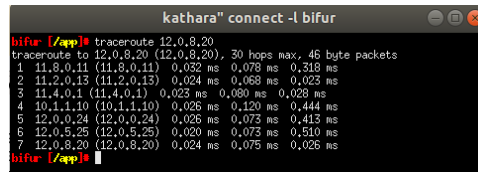


Figure 10: traceroute from bifur to balin

5. *Now, start to continuously ping balin from bifur.*

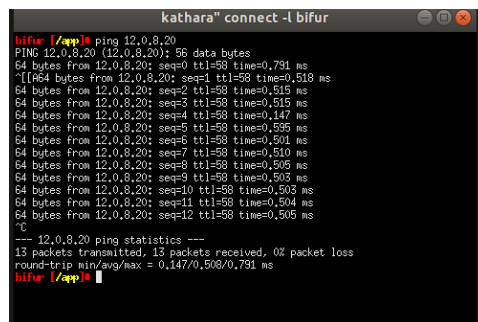


Figure 11: ping from bifur to balin



6. *Disconnect the interface eth0 of the core1*

Executed command in **core1**: *ifconfig eth0 down*

7. *Now, OSPF should update the routing tables. Examine the OSPF messages captured on CD Y to answer the following questions:*

- The OSPF messages are sent almost immediately after the interface was shut down.

There were total 22 OSPF messages were sent. 11 of them are LS UPDATE and 11 are LS ACKNOWLEDGE messages.

- LS UPDATE and LS ACKNOWLEDGE messages are used to flood the link state information.
- 
- The OSPF IPG protocol is used for transport.
- The destination address is 224.0.0.5. It is a multi-cast.

8. *Wait until the ttl (Time to Live) value of ping changed, then, stop the capture and save its output.*

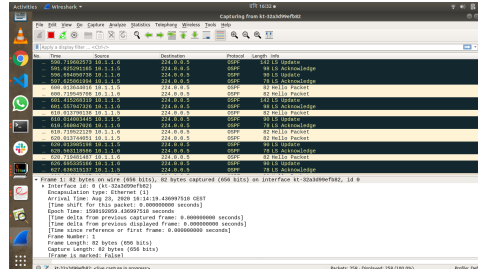


Figure 12: Captures on CD Y with Wireshark

9. *Save and compare the routing tables of the core routers. Are they all identical?*

The routing tables of the core routers are given below:

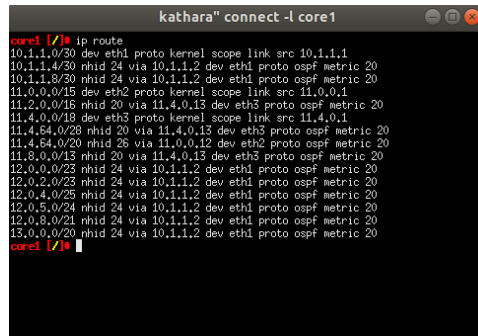


Figure 13: routing table of core1

```

kathara" connect -l core1
core1 [/]# ip route
10.1.1.0/30 dev eth1 proto kernel scope link src 10.1.1.1
10.1.1.4/30 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
10.1.1.8/30 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
11.0.0.0/15 dev eth2 proto kernel scope link src 11.0.0.1
11.2.0.0/18 nhid 20 via 11.4.0.13 dev eth3 proto ospf metric 20
11.4.0.0/18 dev eth3 proto kernel scope link src 11.4.0.1
11.4.64.0/28 nhid 20 via 11.4.0.13 dev eth3 proto ospf metric 20
11.4.64.0/20 nhid 26 via 11.0.0.12 dev eth2 proto ospf metric 20
11.8.0.0/13 nhid 20 via 11.4.0.13 dev eth3 proto ospf metric 20
12.0.0.0/23 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
12.0.2.0/23 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
12.0.4.0/25 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
12.0.5.0/24 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
12.0.8.0/21 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
13.0.0.0/20 nhid 24 via 10.1.1.2 dev eth1 proto ospf metric 20
core1 [/]#

```

Figure 14: routing table of core1

```

kathara" connect -l core2
core2 [/]# ip route
10.1.1.0/30 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
10.1.1.4/30 dev eth1 proto kernel scope link src 10.1.1.5
10.1.1.8/30 dev eth0 proto kernel scope link src 10.1.1.10
11.0.0.0/15 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
11.2.0.0/18 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
11.4.0.0/18 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
11.4.64.0/28 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
11.4.64.0/20 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
11.8.0.0/13 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
12.0.0.0/23 dev eth2 proto kernel scope link src 12.0.0.2
12.0.2.0/23 nhid 34 via 12.0.4.26 dev eth3 proto ospf metric 20
12.0.4.0/25 dev eth3 proto kernel scope link src 12.0.4.2
12.0.5.0/24 nhid 22 via 12.0.0.24 dev eth2 proto ospf metric 20
12.0.8.0/21 nhid 35 proto ospf metric 20
    nexthop via 12.0.0.24 dev eth2 weight 1
    nexthop via 12.0.4.26 dev eth3 weight 1
13.0.0.0/20 nhid 27 via 10.1.1.6 dev eth1 proto ospf metric 20
core2 [/]#

```

Figure 15: routing table of core2

```

kathara" connect -l core3
core3 [/]# ip route
10.1.1.0/30 dev eth0 proto kernel scope link src 10.1.1.2
10.1.1.4/30 dev eth1 proto kernel scope link src 10.1.1.6
10.1.1.8/30 nhid 21 via 10.1.1.5 dev eth1 proto ospf metric 20
11.0.0.0/15 nhid 14 via 10.1.1.1 dev eth0 proto ospf metric 20
11.2.0.0/18 nhid 14 via 10.1.1.1 dev eth0 proto ospf metric 20
11.4.0.0/18 nhid 14 via 10.1.1.1 dev eth0 proto ospf metric 20
11.4.64.0/28 nhid 14 via 10.1.1.1 dev eth0 proto ospf metric 20
11.4.64.0/20 nhid 14 via 10.1.1.1 dev eth0 proto ospf metric 20
11.8.0.0/13 nhid 14 via 10.1.1.1 dev eth0 proto ospf metric 20
12.0.0.0/23 nhid 21 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.2.0/23 nhid 21 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.4.0/25 nhid 21 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.5.0/24 nhid 21 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.8.0/21 nhid 21 via 10.1.1.5 dev eth1 proto ospf metric 20
13.0.0.0/20 dev eth2 proto kernel scope link src 13.0.0.3
core3 [/]#

```

Figure 16: routing table of core3

## 2 Dynamic Routing with RIP

1. *Deploy the topology shown in Figure 2 in Kathará and use RIP with the FRRouting framework..*

```

1 LAB_DESCRIPTION="lab.conf for Assignment-5 Task-2"
2 LAB_VERSION=1.0
3 LAB_AUTHOR="Abdullah Al Mosabbir"
4 LAB_EMAIL="abduallah-al.mosabbir@stud.uni-bamberg.de"
5
6 bofur[0]="R"
7 bofur[image]="unibaktr/alpine:whoami"
8 bofur[exec]="ifconfigeth050.10.0.10netmask255.255.0.0up"
9 bofur[exec]="iprouteadddefaultvia50.10.0.5"
10
11 bombur[0]="S"
12 bombur[image]="unibaktr/alpine:whoami"
13 bombur[exec]="ifconfigeth050.48.0.20netmask255.248.0.0up"
14 bombur[exec]="iprouteadddefaultvia50.48.0.3"
15
16 dori[0]="T"
17 dori[image]="unibaktr/alpine:whoami"
18 dori[exec]="ifconfigeth050.64.0.30netmask255.192.0.0up"
19 dori[exec]="iprouteadddefaultvia50.64.0.2"
20
21 kili[0]="I"
22 kili[image]="unibaktr/alpine:latest"
23 kili[exec]="ifconfigeth013.0.0.30netmask255.255.240.0up"
24 kili[exec]="iprouteadddefaultvia13.0.0.4"
25
26 rip1[0]="N"
27 rip1[1]="P"
28 rip1[image]="unibaktr/alpine:frr"
29 rip1[exec]="ifconfigeth050.1.0.1netmask255.255.255.0up"
30 rip1[exec]="ifconfigeth150.2.2.1netmask255.255.255.0up"
31 rip1[exec]="frrinit.shstart"

```

Figure 17: lab config file (a)

```

32
33 rip2[0]="P"
34 rip2[1]="Q"
35 rip2[2]="T"
36 rip2[image]="unibaktr/alpine:frr"
37 rip2[exec]="ifconfigeth050.2.2.2netmask255.255.255.0up"
38 rip2[exec]="ifconfigeth150.3.0.2netmask255.255.192.0up"
39 rip2[exec]="ifconfigeth250.64.0.2netmask255.192.0.0up"
40 rip2[exec]="frrinit.shstart"
41
42 rip3[0]="L"
43 rip3[1]="Q"
44 rip3[2]="Q"
45 rip3[3]="S"
46 rip3[image]="unibaktr/alpine:frr"
47 rip3[exec]="ifconfigeth050.0.0.3netmask255.255.240.0up"
48 rip3[exec]="ifconfigeth150.2.0.3netmask255.255.254.0up"
49 rip3[exec]="ifconfigeth250.3.0.3netmask255.255.192.0up"
50 rip3[exec]="ifconfigeth350.48.0.3netmask255.240.0.0up"
51 rip3[exec]="frrinit.shstart"
52
53 rip4[0]="T"
54 rip4[1]="M"
55 rip4[2]="L"
56 rip4[image]="unibaktr/alpine:frr"
57 rip4[exec]="ifconfigeth013.0.0.4netmask255.255.240.0up"
58 rip4[exec]="ifconfigeth150.0.16.4netmask255.255.252.0up"
59 rip4[exec]="ifconfigeth250.0.0.4netmask255.248.240.0up"
60 rip4[exec]="frrinit.shstart"

```

Figure 18: lab config file (b)

```

61
62 rip5[0]="M"
63 rip5[1]="N"
64 rip5[2]="Q"
65 rip5[3]="R"
66 rip5[image]="unibaktr/alpine:frr"
67 rip5[exec]="ifconfigeth050.0.16.5netmask255.255.252.0up"
68 rip5[exec]="ifconfigeth150.1.0.5netmask255.255.255.0up"
69 rip5[exec]="ifconfigeth250.2.0.5netmask255.255.254.0up"
70 rip5[exec]="ifconfigeth350.10.0.5netmask255.255.0.0up"
71 rip5[exec]="ifconfigeth2down"
72 rip5[exec]="frrinit.shstart"

```

Figure 19: lab config file (c)

2. *Ensure connectivity between bofur, bombur, kili, and dori.*

```

/app # ping 50.48.0.20
PING 50.48.0.20 (50.48.0.20): 56 data bytes
64 bytes from 50.48.0.20: seq=0 ttl=61 time=0.173 ms
64 bytes from 50.48.0.20: seq=1 ttl=61 time=0.122 ms
^C
--- 50.48.0.20 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.122/0.147/0.173 ms
/app # ping 50.64.0.30
PING 50.64.0.30 (50.64.0.30): 56 data bytes
64 bytes from 50.64.0.30: seq=0 ttl=60 time=0.162 ms
64 bytes from 50.64.0.30: seq=1 ttl=60 time=0.133 ms
^C
--- 50.64.0.30 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.133/0.147/0.162 ms
/app # ping 13.0.0.30
PING 13.0.0.30 (13.0.0.30): 56 data bytes
64 bytes from 13.0.0.30: seq=0 ttl=62 time=0.137 ms
64 bytes from 13.0.0.30: seq=1 ttl=62 time=0.151 ms
^C
--- 13.0.0.30 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.137/0.144/0.151 ms
/app #

```

Figure 20: ping bombur, kili, and dori from bofur

3. *For a short evaluation of RIP, start the lab, but disable the eth2 interface of rip5 initially.*

```

rip5[0]="M"
rip5[1]="N"
rip5[2]="O"
rip5[3]="R"
rip5[image]="unibaktr/alpine:frr"
rip5[exec]="ifconfig eth0 50.0.16.5 netmask 255.255.252.0 up"
rip5[exec]="ifconfig eth1 50.1.0.5 netmask 255.255.255.0 up"
rip5[exec]="ifconfig eth2 50.2.0.5 netmask 255.255.254.0 up"
rip5[exec]="ifconfig eth3 50.10.0.5 netmask 255.255.0.0 up"
rip5[exec]="ifconfig eth2 down"
rip5[exec]="frrinit.sh start"

```

Figure 21: Disable the eth2 interface of rip5 initially

4. *Start a Wireshark capture on CD O.*

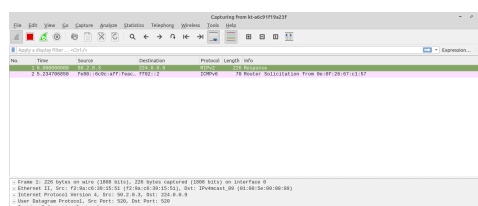


Figure 22: Wireshark capture on CD O

5. *Use traceroute to explore the route path from bofur to bombur.*

```

/app # traceroute 50.48.0.20
traceroute to 50.48.0.20 (50.48.0.20), 30 hops max, 46 byte packets
 1 50.10.0.5 (50.10.0.5) 0.005 ms 0.008 ms 0.002 ms
 2 50.0.16.4 (50.0.16.4) 0.002 ms 0.008 ms 0.002 ms
 3 50.0.0.3 (50.0.0.3) 0.003 ms 0.008 ms 0.002 ms
 4 50.48.0.20 (50.48.0.20) 0.003 ms 0.010 ms 0.003 ms
/app #

```

Figure 23: Traceroute to explore

6. *From bofur start a ping to bombur.*

```

/app # ping 50.48.0.20
PING 50.48.0.20 (50.48.0.20): 56 data bytes
64 bytes from 50.48.0.20: seq=0 ttl=61 time=0.122 ms
64 bytes from 50.48.0.20: seq=1 ttl=61 time=0.112 ms
64 bytes from 50.48.0.20: seq=2 ttl=61 time=0.111 ms
64 bytes from 50.48.0.20: seq=3 ttl=61 time=0.124 ms
64 bytes from 50.48.0.20: seq=4 ttl=61 time=0.113 ms
64 bytes from 50.48.0.20: seq=5 ttl=61 time=0.110 ms
^C
--- 50.48.0.20 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.110/0.115/0.124 ms
/app #

```

Figure 24: Ping to bombur

7. *Enable the eth2 interface of rip5.*

```

rip5 [/]/# ifconfig eth2 up
rip5 [/]/#

```

Figure 25: Enable the eth2 interface

8. *Wait until the ttl value of ping changed and describe the workings of the RIP protocol on the basis of the captured packets*

```

64 bytes from 50.48.0.20: seq=22 ttl=61 time=0.107 ms
64 bytes from 50.48.0.20: seq=23 ttl=61 time=0.148 ms
64 bytes from 50.48.0.20: seq=24 ttl=61 time=0.110 ms
64 bytes from 50.48.0.20: seq=25 ttl=61 time=0.111 ms
64 bytes from 50.48.0.20: seq=26 ttl=61 time=0.106 ms
64 bytes from 50.48.0.20: seq=27 ttl=61 time=0.106 ms
64 bytes from 50.48.0.20: seq=28 ttl=61 time=0.108 ms
64 bytes from 50.48.0.20: seq=29 ttl=61 time=0.176 ms
64 bytes from 50.48.0.20: seq=30 ttl=61 time=0.109 ms
64 bytes from 50.48.0.20: seq=31 ttl=61 time=0.187 ms
64 bytes from 50.48.0.20: seq=32 ttl=61 time=0.103 ms
64 bytes from 50.48.0.20: seq=33 ttl=62 time=0.102 ms
64 bytes from 50.48.0.20: seq=34 ttl=62 time=0.100 ms
64 bytes from 50.48.0.20: seq=35 ttl=62 time=0.102 ms
64 bytes from 50.48.0.20: seq=36 ttl=62 time=0.113 ms
64 bytes from 50.48.0.20: seq=37 ttl=62 time=0.096 ms
64 bytes from 50.48.0.20: seq=38 ttl=62 time=0.107 ms
64 bytes from 50.48.0.20: seq=39 ttl=62 time=0.116 ms
64 bytes from 50.48.0.20: seq=40 ttl=62 time=0.204 ms
^C
--- 50.48.0.20 ping statistics ---
41 packets transmitted, 41 packets received, 0% packet loss
round-trip min/avg/max = 0.089/0.135/0.257 ms
/app #

```

Figure 26: RIP protocol on the basis of the captured packets(a)

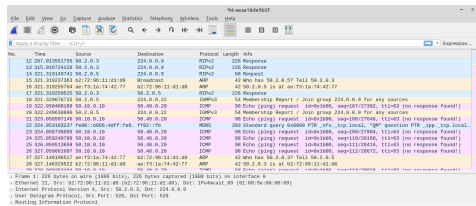


Figure 27: Captured wireshark

- Shutdown the link between rip5 and rip3 again by disabling the particular inter-face. What is the effect on the routing tables of the other routers? Show the routing tables of each router again. Can you still reach bombur from bofur? Is it the same path?



Figure 28: Link between rip5 and rip3 again by disabling the particular inter-face(a)



Figure 29: Link between rip5 and rip3 again by disabling the particular inter-face(b)

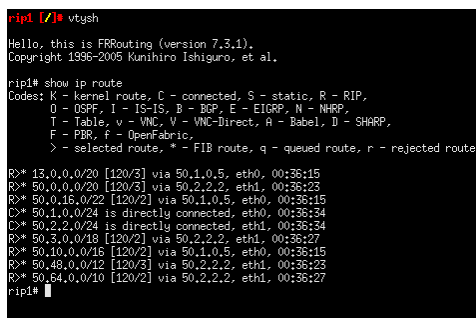


Figure 30: Link between rip5 and rip3 again by disabling the particular inter-face(c)

```

rip2 [Z]# vtysh
Hello, this is FRRouting (version 7.3.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

rip2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

R>* 13.0.0.0/20 [120/3] via 50.3.0.3, eth1, 00:37:14
R>* 50.0.0.0/20 [120/2] via 50.3.0.3, eth1, 00:37:18
R>* 50.0.16.0/22 [120/3] via 50.3.0.3, eth1, 00:37:14
R>* 50.1.0.0/24 [120/2] via 50.2.2.1, eth0, 00:37:23
C>* 50.2.2.0/24 is directly connected, eth0, 00:37:25
C>* 50.3.0.0/18 is directly connected, eth1, 00:37:25
R>* 50.10.0.0/16 [120/3] via 50.2.2.1, eth0, 00:37:09
R>* 50.48.0.0/12 [120/2] via 50.3.0.3, eth1, 00:37:18
C>* 50.64.0.0/10 is directly connected, eth2, 00:37:25
rip2#

```

Figure 31: Link between rip5 and rip3 again by disabling the particular inter-face(d)

```

rip4 [Z]# vtysh
Hello, this is FRRouting (version 7.3.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

rip4# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

C>* 13.0.0.0/20 is directly connected, eth0, 00:37:40
C>* 50.0.0.0/8 is directly connected, eth2, 00:37:40
R>* 50.0.0.0/20 [120/5] via 50.0.16.5, eth1, 00:05:56
C>* 50.0.16.0/22 is directly connected, eth1, 00:37:40
R>* 50.1.0.0/24 [120/2] via 50.0.16.5, eth1, 00:37:34
R>* 50.2.2.0/24 [120/3] via 50.0.0.3, eth2, 00:37:39
R>* 50.3.0.0/18 [120/2] via 50.0.0.3, eth2, 00:37:39
R>* 50.10.0.0/16 [120/2] via 50.0.16.5, eth1, 00:37:34
R>* 50.48.0.0/12 [120/2] via 50.0.0.3, eth2, 00:37:39
R>* 50.64.0.0/10 [120/3] via 50.0.0.3, eth2, 00:37:39
rip4#

```

Figure 32: Link between rip5 and rip3 again by disabling the particular inter-face(e)

```

rip3 [Z]# vtysh
Hello, this is FRRouting (version 7.3.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

rip3# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

R>* 13.0.0.0/20 [120/2] via 50.0.0.4, eth0, 00:39:56
C>* 50.0.0.0/20 is directly connected, eth0, 00:40:02
R>* 50.0.16.0/22 [120/2] via 50.0.0.4, eth0, 00:39:56
R>* 50.1.0.0/24 [120/3] via 50.3.0.2, eth2, 00:07:46
R>* 50.2.2.0/24 [120/2] via 50.3.0.2, eth2, 00:40:01
C>* 50.3.0.0/18 is directly connected, eth2, 00:40:02
R>* 50.10.0.0/16 [120/3] via 50.0.0.4, eth0, 00:07:29
C>* 50.48.0.0/12 is directly connected, eth3, 00:40:02
R>* 50.64.0.0/10 [120/2] via 50.3.0.2, eth2, 00:40:01
rip3#

```

Figure 33: Link between rip5 and rip3 again by disabling the particular inter-face(f)



```

rip5 [/]# vtysh
Hello, this is FRRouting (version 7.3.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

rip5# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, H - Babel, D - SHARP,
       F - FRR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

R>* 13.0.0.0/20 [120/2] via 50.0.16.4, eth0, 00:40:10
R>* 50.0.0.0/20 [120/4] via 50.1.0.1, eth1, 00:08:31
C>* 50.0.16.0/22 is directly connected, eth0, 00:40:11
C>* 50.1.0.0/24 is directly connected, eth1, 00:40:11
R>* 50.2.2.0/24 [120/2] via 50.1.0.1, eth1, 00:40:10
R>* 50.3.0.0/18 [120/3] via 50.1.0.1, eth1, 00:08:31
C>* 50.10.0.0/16 is directly connected, eth3, 00:40:11
R>* 50.48.0.0/12 [120/3] via 50.0.16.4, eth0, 00:08:20
R>* 50.64.0.0/10 [120/5] via 50.1.0.1, eth1, 00:40:10
rip5#

```

Figure 34: Link between rip5 and rip3 again by disabling the particular inter-face(g)

```

/app # traceroute 50.48.0.20
traceroute to 50.48.0.20 (50.48.0.20), 30 hops max, 46 byte packets
 1 50.10.0.5 (50.10.0.5) 0.005 ms 0.009 ms 0.003 ms
 2 50.0.16.4 (50.0.16.4) 0.002 ms 0.009 ms 0.002 ms
 3 50.0.0.3 (50.0.0.3) 0.002 ms 0.009 ms 0.003 ms
 4 50.48.0.20 (50.48.0.20) 0.002 ms 0.010 ms 0.003 ms
/app #

```

Figure 35: Link between rip5 and rip3 again by disabling the particular inter-face(h)

10. *Is the basic RIP a secure protocol? What kind of attacks can be done by the forged manipulation of RIP updates?.*

It is important to note that RIPv1 cannot be authenticated. Further, if RIPv1 is enabled then RIP will reply to REQUEST packets, sending the state of its RIP routing table to any remote routers that ask on demand. RIPv2 allows packets to be authenticated via either an insecure plain text password, included with the packet, or via a more secure MD5 based HMAC, RIPv1 can not be authenticated at all, thus when authentication is configured ripd will discard routing updates received via RIPv1 packets. However, unless RIPv1 reception is disabled entirely, RIP Version Control, RIPv1 REQUEST packets which are received, which query the router for routing information, will still be honoured by ripd, and ripd WILL reply to such packets. This allows ripd to honour such REQUESTs (which sometimes is used by old equipment and very simple devices to bootstrap their default route), while still providing security for route updates which are received.