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

List of Figures

1 Interior Routing with OSPF

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

```
lab.conf
LAB_DESCRIPTION="lab.conf for Assignment-5 Task-1"
LAB_VERSION=1.0
LAB_AUTHOR="Abdullah Al Mosabbir"
LAB_EMAIL=abdullah-al.mosabbir@stud.uni-bamberg.de

bifur[0]="J"
bifur[image]="unibaktr/alpine:whoami"
bifur[exec]="ifconfig eth0 11.8.0.10 netmask 255.248.0.0 up"
bifur[exec]="ip route add default via 11.8.0.11"

balin[0]="K"
balin[image]="unibaktr/alpine:whoami"
balin[exec]="ifconfig eth0 12.0.8.20 netmask 255.255.248.0 up"
balin[exec]="ip route add default via 12.0.8.25"

kili[0]="I"
kili[image]="unibaktr/alpine:latest"
kili[exec]="ifconfig eth0 13.0.0.30 netmask 255.255.240.0 up"
kili[exec]="ip route add default via 13.0.0.3"

core1[0]="Z"
core1[1]="X"
core1[2]="A"
core1[3]="B"
core1[image]="unibaktr/alpine:frr"
core1[exec]="ifconfig eth0 10.1.1.9 netmask 255.255.255.252 up"
core1[exec]="ifconfig eth1 10.1.1.1 netmask 255.255.255.252 up"
core1[exec]="ifconfig eth2 11.0.0.1 netmask 255.254.0.0 up"
core1[exec]="ifconfig eth3 11.4.0.1 netmask 255.255.192.0 up"
core1[exec]="frrinit.sh start"
```

Figure 1: lab config file (a)

```
core2[0]="Z"
core2[1]="Y"
core2[2]="E"
core2[3]="F"
core2[image]="unibaktr/alpine:frr"
core2[exec]="ifconfig eth0 10.1.1.10 netmask 255.255.255.252 up"
core2[exec]="ifconfig eth1 10.1.1.5 netmask 255.255.255.252 up"
core2[exec]="ifconfig eth2 12.0.0.2 netmask 255.255.254.0 up"
core2[exec]="ifconfig eth3 12.0.4.2 netmask 255.255.255.128 up"
core2[exec]="frrinit.sh start"

core3[0]="X"
core3[1]="Y"
core3[2]="I"
core3[image]="unibaktr/alpine:frr"
core3[exec]="ifconfig eth0 10.1.1.2 netmask 255.255.255.252 up"
core3[exec]="ifconfig eth1 10.1.1.6 netmask 255.255.255.252 up"
core3[exec]="ifconfig eth2 13.0.0.3 netmask 255.255.240.0 up"
core3[exec]="frrinit.sh start"

ospf1[0]="C"
ospf1[1]="D"
ospf1[2]="J"
ospf1[image]="unibaktr/alpine:frr"
ospf1[exec]="ifconfig eth0 11.4.64.11 netmask 255.255.255.240 up"
ospf1[exec]="ifconfig eth1 11.2.0.11 netmask 255.255.0.0 up"
ospf1[exec]="ifconfig eth2 11.8.0.11 netmask 255.248.0.0 up"
ospf1[exec]="frrinit.sh start"
```

Figure 2: lab config file (b)

```
ospf2[0]="A"
ospf2[1]="C"
ospf2[image]="unibaktr/alpine:frr"
ospf2[exec]="ifconfig eth0 11.0.0.12 netmask 255.254.0.0 up"
ospf2[exec]="ifconfig eth1 11.4.64.12 netmask 255.255.240.0 up"
ospf2[exec]="frrinit.sh start"

ospf3[0]="B"
ospf3[1]="D"
ospf3[image]="unibaktr/alpine:frr"
ospf3[exec]="ifconfig eth0 11.4.0.13 netmask 255.255.192.0 up"
ospf3[exec]="ifconfig eth1 11.2.0.13 netmask 255.255.0.0 up"
ospf3[exec]="frrinit.sh start"

ospf4[0]="E"
ospf4[1]="G"
ospf4[image]="unibaktr/alpine:frr"
ospf4[exec]="ifconfig eth0 12.0.0.24 netmask 255.255.254.0 up"
ospf4[exec]="ifconfig eth1 12.0.5.24 netmask 255.255.255.0 up"
ospf4[exec]="frrinit.sh start"

ospf5[0]="G"
ospf5[1]="H"
ospf5[2]="K"
ospf5[image]="unibaktr/alpine:frr"
ospf5[exec]="ifconfig eth0 12.0.5.25 netmask 255.255.255.0 up"
ospf5[exec]="ifconfig eth1 12.0.2.25 netmask 255.255.254.0 up"
ospf5[exec]="ifconfig eth2 12.0.8.25 netmask 255.255.248.0 up"
ospf5[exec]="frrinit.sh start"
```

Figure 3: lab config file (c)

```
ospf6[0]="F"
ospf6[1]="H"
ospf6[image]="unibaktr/alpine:frr"
ospf6[exec]="ifconfig eth0 12.0.4.26 netmask 255.255.255.128 up"
ospf6[exec]="ifconfig eth1 12.0.2.26 netmask 255.255.254.0 up"
ospf6[exec]="frrinit.sh start"
```

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 ×
core1 > etc > frr > daemons
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
```

Figure 5: deamons file forall routers

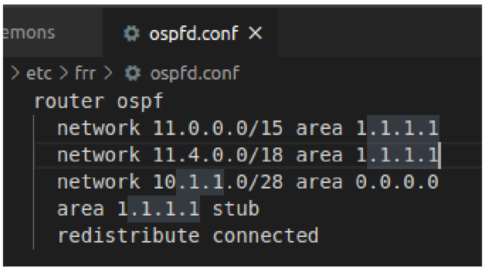


Figure 6: ospfd.conf file for core1

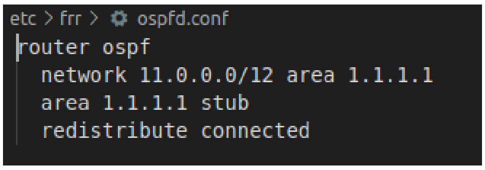


Figure 7: ospfd.conf file for ospf1

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.***

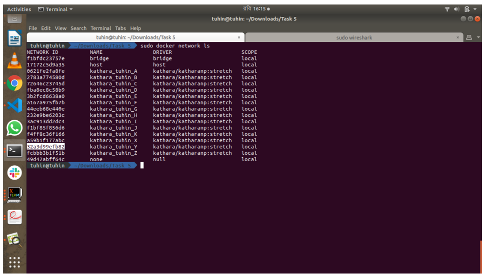


Figure 8: wiresark CD Y

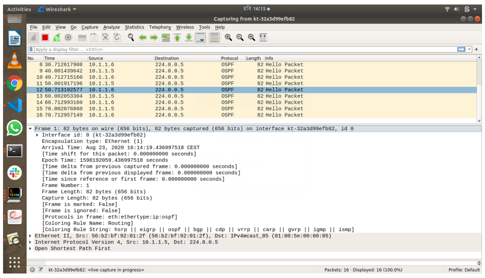


Figure 9: wireshark capture CD Y

4. ***From bifur, run a traceroute to balin. Determine wether 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.

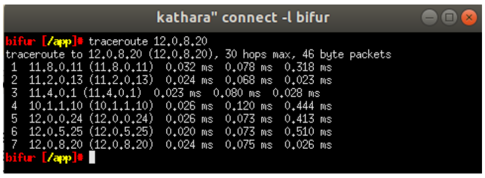


Figure 10: traceroute from bifur to balin

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

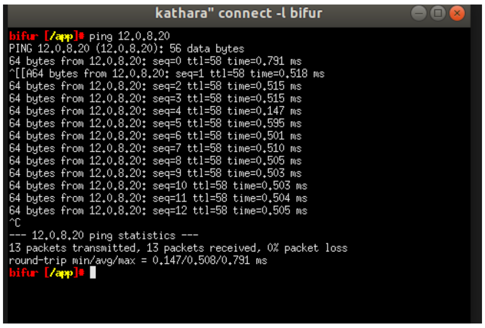


Figure 11: ping from bifur to balin

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 PSPF 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.***

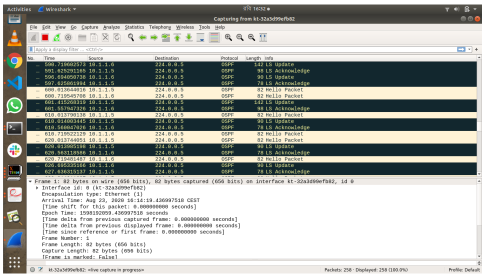


Figure 12: Captures on CD Y with Wireshark

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:

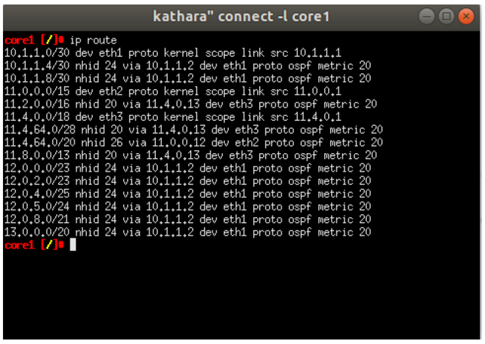


Figure 13: routing table of core1

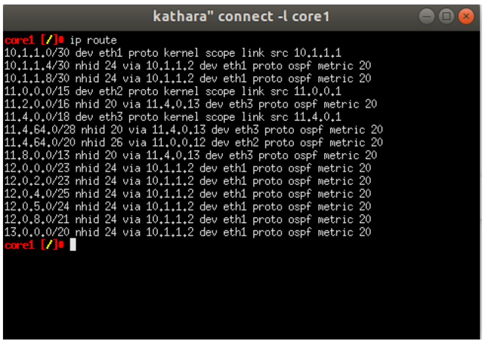


Figure 14: routing table of core1

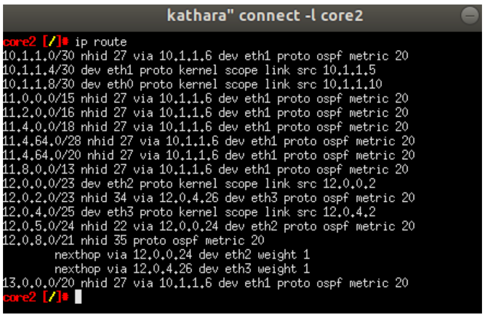


Figure 15: routing table of core2

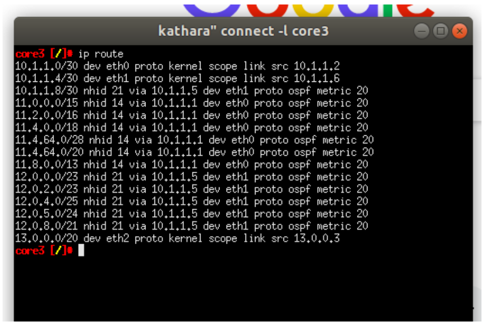


Figure 16: routing table of core3

2 Dynamic Routing with RIP

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

```
LAB_DESCRIPTION="lab.conf for Assignment-5 Task-2"
LAB_VERSION=1.0
LAB_AUTHOR="Abdullah Al Mosabbir"
LAB_EMAIL=abdullah-al.mosabbir@stud.uni-bamberg.de

bofur[0]="R"
bofur[image]="unibaktr/alpine:whoami"
bofur[exec]="ifconfigeth050.10.0.10netmask255.255.0.0up"
bofur[exec]="iprouteadddefaultvia50.10.0.5"

bombur[0]="S"
bombur[image]="unibaktr/alpine:whoami"
bombur[exec]="ifconfigeth050.48.0.20netmask255.240.0.0up"
bombur[exec]="iprouteadddefaultvia50.48.0.3"

dori[0]="T"
dori[image]="unibaktr/alpine:whoami"
dori[exec]="ifconfigeth050.64.0.30netmask255.192.0.0up"
dori[exec]="iprouteadddefaultvia50.64.0.2"

kili[0]="I"
kili[image]="unibaktr/alpine:latest"
kili[exec]="ifconfigeth013.0.0.30netmask255.255.240.0up"
kili[exec]="iprouteadddefaultvia13.0.0.4"

rip1[0]="N"
rip1[1]="P"
rip1[image]="unibaktr/alpine:frr"
rip1[exec]="ifconfigeth050.1.0.1netmask255.255.255.0up"
rip1[exec]="ifconfigeth150.2.2.1netmask255.255.255.0up"
rip1[exec]="frrinit.shstart"
```

Figure 17: lab config file (a)

```

rip2[0]="P"
rip2[1]="Q"
rip2[2]="T"
rip2[image]="unibaktr/alpine:frr"
rip2[exec]="ifconfigeth050.2.2.2netmask255.255.255.0up"
rip2[exec]="ifconfigeth150.3.0.2netmask255.255.192.0up"
rip2[exec]="ifconfigeth250.64.0.2netmask255.192.0.0up"
rip2[exec]="frrinit.shstart"

rip3[0]="L"
rip3[1]="O"
rip3[2]="Q"
rip3[3]="S"
rip3[image]="unibaktr/alpine:frr"
rip3[exec]="ifconfigeth050.0.0.3netmask255.255.240.0up"
rip3[exec]="ifconfigeth150.2.0.3netmask255.255.254.0up"
rip3[exec]="ifconfigeth250.3.0.3netmask255.255.192.0up"
rip3[exec]="ifconfigeth350.48.0.3netmask255.240.0.0up"
rip3[exec]="frrinit.shstart"

rip4[0]="I"
rip4[1]="M"
rip4[2]="L"
rip4[image]="unibaktr/alpine:frr"
rip4[exec]="ifconfigeth013.0.0.4netmask255.255.240.0up"
rip4[exec]="ifconfigeth150.0.16.4netmask255.255.252.0up"
rip4[exec]="ifconfigeth250.0.0.4netmask255.248.240.0up"
rip4[exec]="frrinit.shstart"
```

Figure 18: lab config file (b)

```

rip5[0]="M"
rip5[1]="N"
rip5[2]="O"
rip5[3]="R"
rip5[image]="unibaktr/alpine:frr"
rip5[exec]="ifconfigeth050.0.16.5netmask255.255.252.0up"
rip5[exec]="ifconfigeth150.1.0.5netmask255.255.255.0up"
rip5[exec]="ifconfigeth250.2.0.5netmask255.255.254.0up"
rip5[exec]="ifconfigeth350.10.0.5netmask255.255.0.0up"
rip5[exec]="ifconfigeth2down"
rip5[exec]="frrinit.shstart"
```

Figure 19: lab config file (c)

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

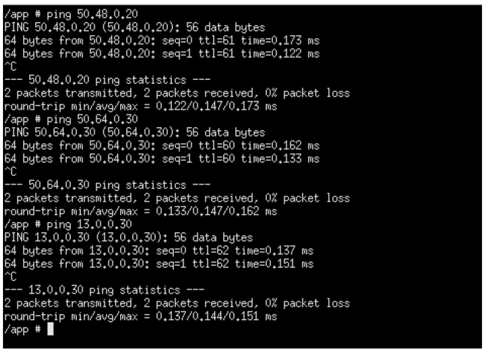


Figure 20: ping bombur, kili, and dori from bofur

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

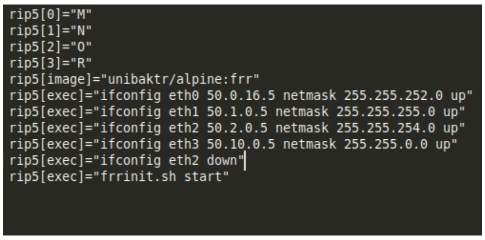


Figure 21: Disable the eth2 interface of rip5 initially

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

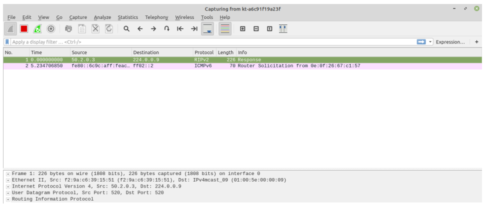
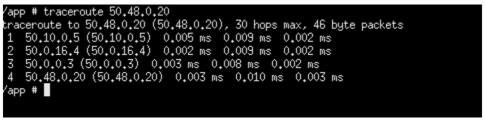


Figure 22: Wireshark capture on CD O

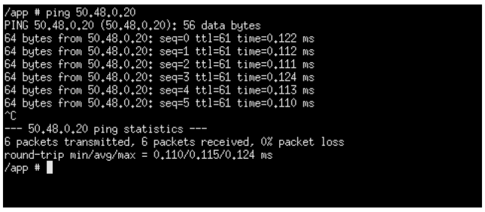
5. ***Use traceroute to explore the route path from bofur to bombur.***



```
/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  0.009 ms  0.002 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)  0.003 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 #
```

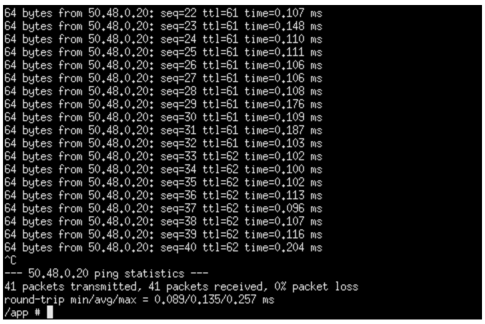
Figure 24: Ping to bombur

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)

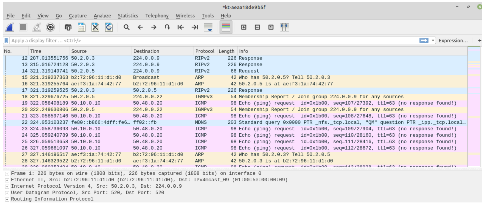


Figure 27: Captured wireshark

9. 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)

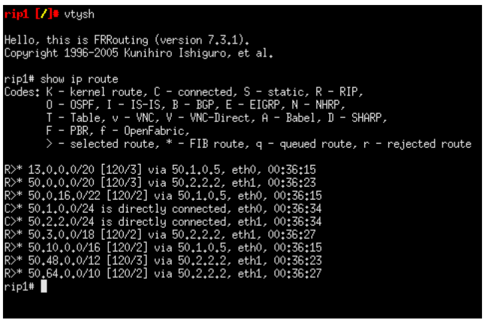


Figure 30: Link between rip5 and rip3 again by disabling the particular inter-face(c)

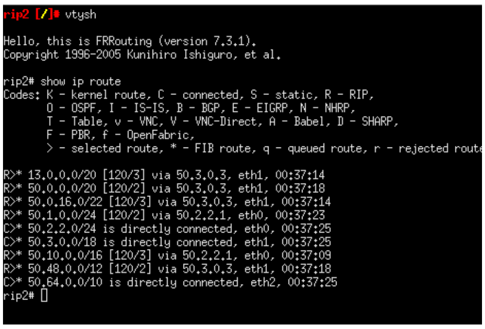


Figure 31: Link between rip5 and rip3 again by disabling the particular inter-face(d)

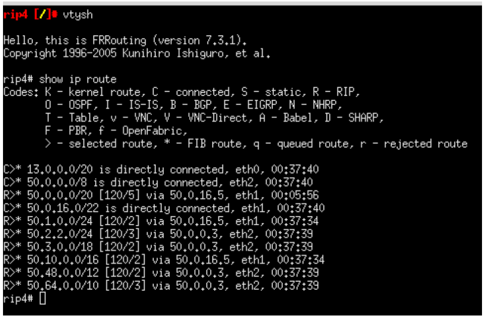


Figure 32: Link between rip5 and rip3 again by disabling the particular inter-face(e)

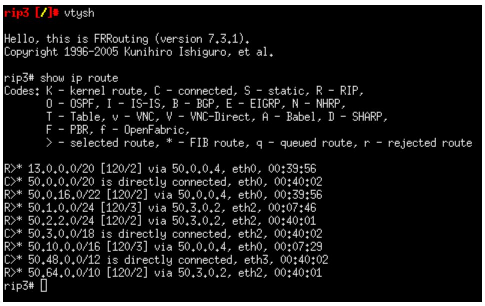


Figure 33: Link between rip5 and rip3 again by disabling the particular inter-face(f)

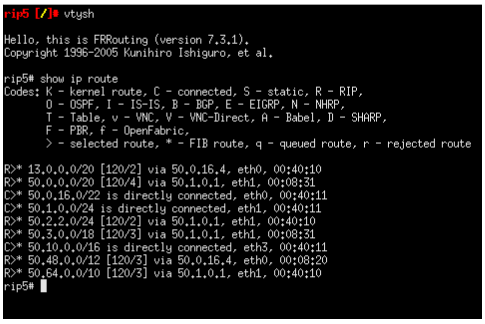


Figure 34: Link between rip5 and rip3 again by disabling the particular inter-face(g)

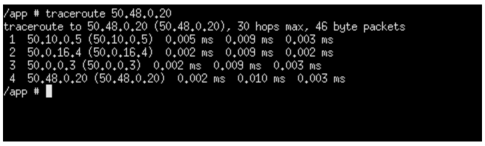


Figure 35: Link between rip5 and rip3 again by disabling the particular inter-face(h)

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.