Université Libre de Bruxelles

Communication networks : protocols and architectures
ELEC-H417

Lab 2 - Tunnels & Encapsulation

Group D :
Bienfait Alexandre 513930
Boistel Julien 440915
Humblet Raphaël 514085
Mutkowski Philippe 494470

Teacher :
Dricot Jean-Michel

Teaching Assistant :
Chassagne Aurélien

November 2023

Academic year 2023-2024

Contents

1 Configuration

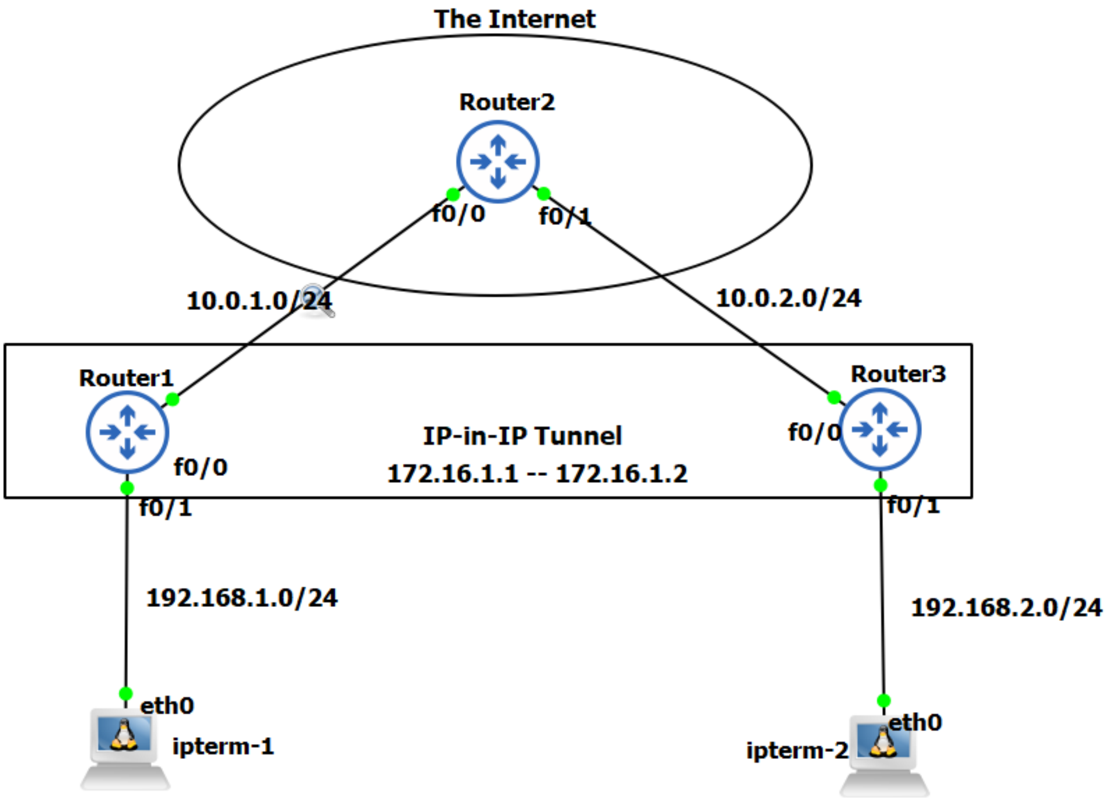

Figure 1: Initial topology

The topology has been reproduced in the GNS3 window, the routers and terminals were correctly setup. RIP is active on the routers, it was verified by displaying the different routing tables.

2 Mission 1: No Tunnel

Here is the traceroute of ipterm1 to ipterm2. It goes through R1, R2 and R3. The length is of 4 hops.

```
root@ipterm-1:~# traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  6.467 ms  17.230 ms  28.067 ms
 2  10.0.1.2 (10.0.1.2)  38.237 ms  48.743 ms  59.221 ms
 3  10.0.2.1 (10.0.2.1)  70.504 ms  81.384 ms  91.528 ms
 4  192.168.2.2 (192.168.2.2)  112.182 ms  124.865 ms  135.606 ms
```

Figure 2: Route from ipterm 1 to ipterm 2

3 Mission 2: Site-to-site Tunnel

The Figure 3 shows the configuration of the tunnel in the Router 1. The same configuration has been applied in the Router 3 for the tunnel to be used bidirectionally.

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#interface tunnel0
Router1(config-if)#ip add
*Mar  1 00:14:41.771: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to down
Router1(config-if)#ip address 172.16.1.1 255.255.255.0
Router1(config-if)#tunnel mode ipip
Router1(config-if)#tunnel source 10.0.1.1
Router1(config-if)#tunnel destination 10.0.2.1
Router1(config-if)#
*Mar  1 00:15:38.283: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
Router1(config-if)#no shutdown
Router1(config-if)#end
Router1#write
Building configuration...

*Mar  1 00:16:20.347: %SYS-5-CONFIG_I: Configured from console by console[OK]
```

Figure 3: Tunnel configuration

Router 1

```
Router1#show ip int br
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            10.0.1.1        YES NVRAM  up                    up
FastEthernet0/1            192.168.1.1     YES NVRAM  up                    up
FastEthernet1/0            unassigned      YES NVRAM  administratively down down
FastEthernet2/0            unassigned      YES NVRAM  administratively down down
Tunnel0                    172.16.1.1      YES manual up                    up
```

Router 3

```
Router3#show ip int br
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            10.0.2.1        YES NVRAM  up                    up
FastEthernet0/1            192.168.2.1     YES NVRAM  up                    up
FastEthernet1/0            unassigned      YES NVRAM  administratively down down
FastEthernet2/0            unassigned      YES NVRAM  administratively down down
Tunnel0                    172.16.1.2      YES manual up                    up
```

Figure 4: Tunnel Setup

When showing the `traceroute` between ipterm 1 and ipterm 2, it can be seen that instead of going through the router 2, it will go directly from router 1 (192.168.1.1) to router 3 (172.16.1.2) through the tunnel. This is good because the route is now in 3 hops instead of 4 before the construction of the tunnel.

```
root@ipterm-1:~# traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  12.195 ms  23.209 ms  33.835 ms
 2  172.16.1.2 (172.16.1.2)  56.265 ms  67.131 ms  77.898 ms
 3  192.168.2.2 (192.168.2.2)  98.514 ms  109.540 ms  131.056 ms
```

Figure 5: Route from ipterm 1 to ipterm 2

The figure 6 shows that the tunnel encapsulates the message. It can be seen that the main message has 192.168.1.2 as source and 192.168.2.2 as destination. This message is contained in another message that goes from Router 1 (10.0.1.1) to Router 3 (10.0.2.1). When this package arrives at the third Router, the packet is analysed and the original message is thus sent to the ipterm 2.

```
ping from ipterm1 to ipterm2
174 385.790560  192.168.1.2        192.168.2.2        ICMP   118 Echo (ping) request  id=0x0049, seq=1/256, ttl=63 (reply in 175)
175 385.836677  192.168.2.2        192.168.1.2        ICMP   118 Echo (ping) reply    id=0x0049, seq=1/256, ttl=63 (request in 174)
176 386.784367  192.168.1.2        192.168.2.2        ICMP   118 Echo (ping) request  id=0x0049, seq=2/512, ttl=63 (reply in 177)
177 386.820611  192.168.2.2        192.168.1.2        ICMP   118 Echo (ping) reply    id=0x0049, seq=2/512, ttl=63 (request in 176)
178 387.794165  192.168.1.2        192.168.2.2        ICMP   118 Echo (ping) request  id=0x0049, seq=3/768, ttl=63 (reply in 179)
179 387.826258  192.168.2.2        192.168.1.2        ICMP   118 Echo (ping) reply    id=0x0049, seq=3/768, ttl=63 (request in 178)
180 387.857268  c4:01:2b:63:00:00  c4:01:2b:63:00:00  LOOP    60 Reply
181 393.699968  172.16.1.2         224.0.0.9          RIPv2  106 Response
182 395.896646  c4:02:2b:72:00:00  c4:02:2b:72:00:00  LOOP    60 Reply
183 397.959249  10.0.1.1           224.0.0.9          RIPv2  126 Response
184 398.642476  c4:01:2b:63:00:00  c4:01:2b:63:00:00  LOOP    60 Reply
185 406.410748  c4:02:2b:72:00:00  c4:02:2b:72:00:00  LOOP    60 Reply
> Frame 174: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface -, id 0
> Ethernet II, Src: c4:01:2b:63:00:00 (c4:01:2b:63:00:00), Dst: c4:02:2b:72:00:00 (c4:02:2b:72:00:00)
> Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.2.1
> Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.2.2

0000  c4 02 2b 72 00 00 c4 01  2b 63 00
0010  00 68 00 4d 00 00 ff 04  a4 43 0a
0020  02 01 45 00 00 54 81 0b  40 00 3f
0030  01 02 c0 a8 02 02 08 00  02 9c 00
```

Figure 6: Message encapsulation in the tunnel

4 Mission 3: Deploy IPv6

Here, the IPv6 interface has been set up on Router 1 and Router 3.

```
Router1#show ipv6 interface brief
FastEthernet0/0            [up/up]
FastEthernet0/1            [up/up]
    FE80::C601:2BFF:FE63:1
    FC00:1::1
FastEthernet1/0            [administratively down/down]
FastEthernet2/0            [administratively down/down]
Tunnel0                    [up/up]
Router1#
```

Figure 7: Configuration Router 1

When the following command is run in the ipterm 1 terminal:

```
>>> ping6 -c3 FC00:2::2
```

The packets are not transmitted (as seen on Figure 8). This is logical because the Router 2 and the tunnel have not been configured for IPv6 communication. Thus the IPv6 message can't go from ipterm 1 to ipterm 2.
But when a ping is send under the IPv4 format via a `ping` call in the console. The packets are transmitted and go through the tunnel.

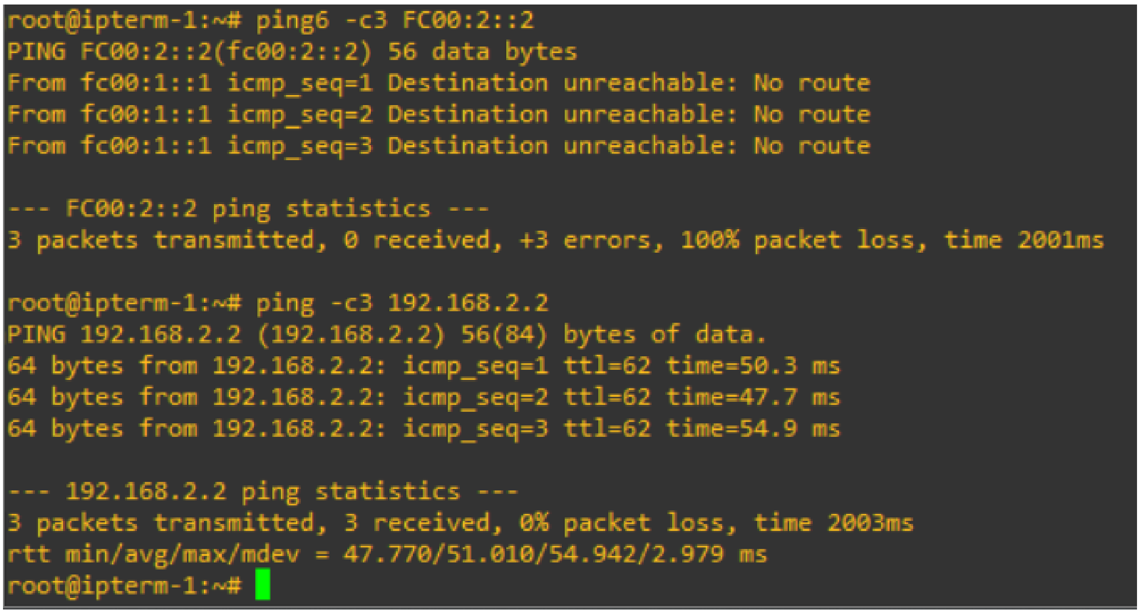

Figure 8: Ping from ipterm 1 to ipterm 2 (ping6 and ping)

5 Mission 4: IPv6-in-IPv4 tunnel

Here, Router 2 (ISP) is not IPv6 ready. To ensure that an IPv6 message can travel from ipterm 1 to ipterm 2, the mode of the tunnel has to be changed from `ipip` to `ipv6ip`.

This has been done by adding an IPv6 network for both router that communicates through the tunnel `FC:00:ABBA::0/64`. The interfaces of the fastEthernet and tunnel of both routers (1 and 3) needed to be enabled for the IPv6 RIP instance.

5.1 IPv6 routes and IPv6 RIP setup

The following figure shows all the IPv6 routes from the Router 1. It can be seen that the tunnel is now accessible for IPv6 protocol messages.

```
Router1#show ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   FC00:1::/64 [0/0]
     via ::, FastEthernet0/1
L   FC00:1::1/128 [0/0]
     via ::, FastEthernet0/1
C   FC00:ABBA::/64 [0/0]
     via ::, Tunnel0
L   FC00:ABBA::1/128 [0/0]
     via ::, Tunnel0
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
```

Figure 9: IPv6 routes after IPv6 RIP setup

5.2 Ping IPv6 from Router 1 to Router 3

```
Router1#ping FC00:ABBA::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FC00:ABBA::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/52/56 ms
```

Figure 10: IPv6 ping from R1 to R3

The ping between Router 1 and Router 3 succeeds even if Router 2 has no IPv6 interface configured. This is because the message has been encapsulated in an IPv4 message to go through the tunnel. When arriving at the end of the tunnel (Router 1), the IPv6 message is de-encapsulated from the IPv4 message and sent to the ipterm 1 through eth0 that has been previously defined to allow IPv6 communication.

```
Router3#ping FC00:ABBA::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FC00:ABBA::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/52/60 ms
Router3#ping FC00:1::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FC00:1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/54/64 ms
Router3#
```

Figure 11: Ping between R3 and R1 followed by a ping between R3 and ipterm 1

5.3 Ping IPv6 from ipterm 1 to ipterm 2

As shown on figure 12, the ping commabd from ipterm 1 to ipterm 2 in IPv6 works and goes through the tunnel. The tunnel encapsulates the IPv6 message into an IPv4 message to "go through router 2". When reaching Router 3, the IPV4 message is de-encapsulated to find back the IPv6 message and send it to the ipterm 2.

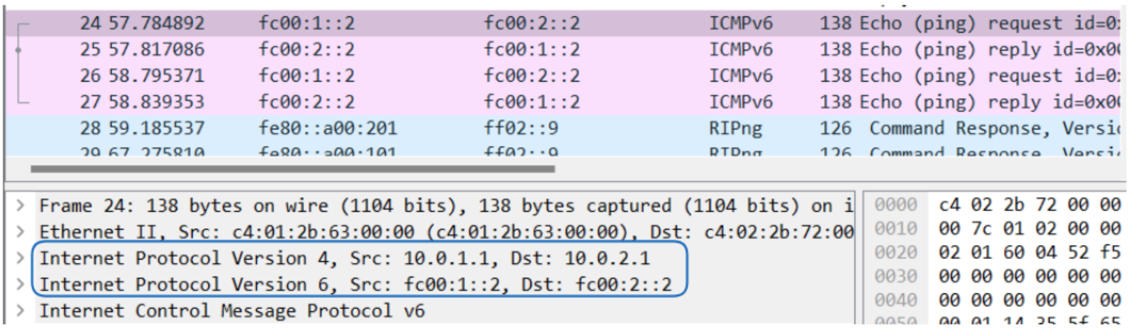

Figure 12: Capture of R1-R2 connection when pinging from ipterm 1 to ipterm 2

5.4 Ping from ipterm 1 to ipterm 2

The route used when trying to ping an IPv4 message from ipterm 1 to ipterm 2 goes through Router 1, Router 2 and Router 3. The message is not encapsulated anymore. That is because the tunnel has been set up to encapsulate IPv6 messages into a IPv4 format from Router 1 to Router 3. This is also shown by the traceroute command that shows that the route has now 4 hops.

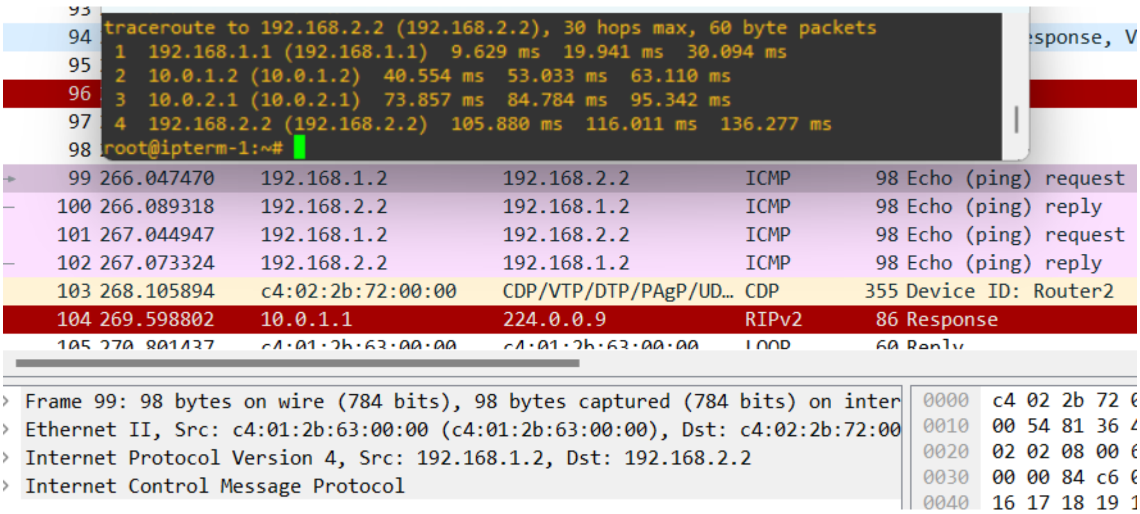

Figure 13: Ping IPv4 from ipterm 1 to iptarm2

6 Conclusion

To conclude, an IPv6-in-IPv4 tunnel allows communication between two hosts in IPv6 even if an ISP on the route (represented here by Router 2) is not set up for

IPv6 communication.

This is really helpful because every part of the ISP is not necessary configured to use IPv6. Some devices are too old to support IPv6 and are therefore not compatible. However, IPv6 is faster and allows for a larger address space (2^{128} for IPv6 compared to 2^{32} for IPv4). Therefore, it is important to have a means of transmitting messages between devices using IPv6 while passing through routers that only accept IPv4.