IRELE - MA1 Electrical Engineering

ELEC-H417

Report Lab 2
Tunnels and Encapsulation

Authors :

Amaury ARICO

Alexis BOLLENGIER

Emmeran COLOT

Sefa GÖNEN

Professor :

Jean-Michel DRICOT

Assistant :

Navid LADNER

Academic year :

2024 - 2025

Contents

1 Mission 0 - No Tunnel Configuration **1**
1.1 Bonus question - Lost packet 1

2 Mission 1 - Site-to-Site Tunnel **3**
2.1 Question - Initial route 3
2.2 Question - A new route 3

3 Mission 2 - Deploy IPv6 **7**
3.1 Question - IPv6 compatibility 7
3.2 Question - IPv6 format 8

4 Mission 3 - IPv6-in-IPv4 Tunnel **9**
4.1 Question - Different routes 9
4.2 Question - Modern tunnels 10

1

Mission 0 - No Tunnel Configuration

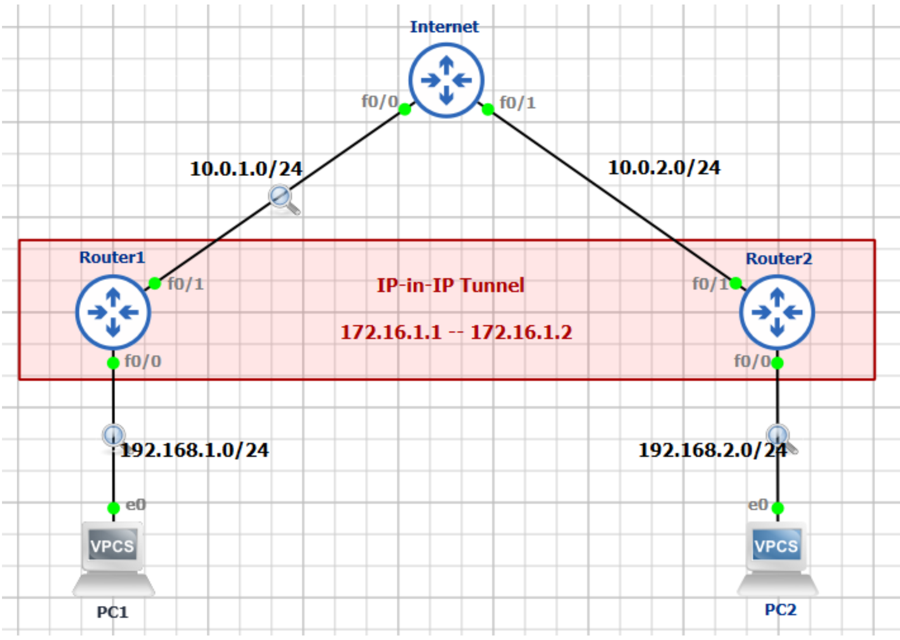

Figure 1.1: Initial topology

1.1 Bonus question - Lost packet

Lost packet

Do you observe something special with the first (or two firsts) packet of the first ping command ? If yes, what is it and why ? (Wireshark could help)

The figure 1.2 shows the result of the ping command from **PC1** to **PC2** :

```
PC1> ping 192.168.2.2

192.168.2.2 icmp_seq=1 timeout
192.168.2.2 icmp_seq=2 timeout
84 bytes from 192.168.2.2 icmp_seq=3 ttl=61 time=63.682 ms
84 bytes from 192.168.2.2 icmp_seq=4 ttl=61 time=60.021 ms
84 bytes from 192.168.2.2 icmp_seq=5 ttl=61 time=56.525 ms
```

Figure 1.2: Ping command from PC1 (192.168.1.1) to PC2 (192.168.2.2)

We can observe that PC1 doesn't get a response (timeout) for the first 2 pings. The reason is that **PC1** and **R1** didn't link their respective MAC addresses yet. It is also the case for **PC2** and **R2**.

Indeed, by analysing the packet traffic on Wireshark, shown in the figures 1.3 and 1.4, we can notice that the first ping request arrives at the last router R2 but this router doesn't have the MAC address of PC2 so that's why we get a timeout response for PC1 and an ARP request from R2 to PC2.

Concerning the second ping, the ping request arrives at PC2 but the ping reply doesn't go further than R1 because again, R1 didn't link its MAC address with PC1 (that's why we get a timeout response for the second ping and an ARP request from R1 to PC1).

No.	Time	Source	Destination	Protocol	Length	Time to Live	Info
5	27.801663	192.168.1.1	192.168.2.2	ICMP	98	64	Echo (ping) request id=0x0935, seq=1/256, ttl=64 (no response found!)
6	29.803323	192.168.1.1	192.168.2.2	ICMP	98	64	Echo (ping) request id=0x0b35, seq=2/512, ttl=64 (no response found!)
7	29.836882	c4:01:2b:63:00:00	Broadcast	ARP	60		Who has 192.168.1.1? Tell 192.168.1.101
8	29.837336	Private_66:68:00	c4:01:2b:63:00:00	ARP	60		192.168.1.1 is at 00:50:79:66:68:00
9	31.803941	192.168.1.1	192.168.2.2	ICMP	98	64	Echo (ping) request id=0x0d35, seq=3/768, ttl=64 (reply in 10)
10	31.866946	192.168.2.2	192.168.1.1	ICMP	98	61	Echo (ping) reply id=0x0d35, seq=3/768, ttl=61 (request in 9)
11	32.869272	192.168.1.1	192.168.2.2	ICMP	98	64	Echo (ping) request id=0x0e35, seq=4/1024, ttl=64 (reply in 12)
12	32.928881	192.168.2.2	192.168.1.1	ICMP	98	61	Echo (ping) reply id=0x0e35, seq=4/1024, ttl=61 (request in 11)
13	33.930530	192.168.1.1	192.168.2.2	ICMP	98	64	Echo (ping) request id=0x0f35, seq=5/1280, ttl=64 (reply in 14)
14	33.986476	192.168.2.2	192.168.1.1	ICMP	98	61	Echo (ping) reply id=0x0f35, seq=5/1280, ttl=61 (request in 13)

Figure 1.3: Wireshark - Ping command from PC1 (192.168.1.1) to PC2 (192.168.2.2) - Listening to link between PC1 and R1

No.	Time	Source	Destination	Protocol	Length	Time to Live	Info
2	13.057910	c4:03:2b:81:00:00	Broadcast	ARP	60		Who has 192.168.2.2? Tell 192.168.2.202
3	13.058041	Private_66:68:01	c4:03:2b:81:00:00	ARP	60		192.168.2.2 is at 00:50:79:66:68:01
4	15.031031	192.168.1.1	192.168.2.2	ICMP	98	61	Echo (ping) request id=0x0b35, seq=2/512, ttl=61 (reply in 5)
5	15.031458	192.168.2.2	192.168.1.1	ICMP	98	64	Echo (ping) reply id=0x0b35, seq=2/512, ttl=64 (request in 4)
6	17.050822	192.168.1.1	192.168.2.2	ICMP	98	61	Echo (ping) request id=0x0d35, seq=3/768, ttl=61 (reply in 7)
7	17.051290	192.168.2.2	192.168.1.1	ICMP	98	64	Echo (ping) reply id=0x0d35, seq=3/768, ttl=64 (request in 6)
8	18.112079	192.168.1.1	192.168.2.2	ICMP	98	61	Echo (ping) request id=0x0e35, seq=4/1024, ttl=61 (reply in 9)
9	18.112479	192.168.2.2	192.168.1.1	ICMP	98	64	Echo (ping) reply id=0x0e35, seq=4/1024, ttl=64 (request in 8)
10	19.170951	192.168.1.1	192.168.2.2	ICMP	98	61	Echo (ping) request id=0x0f35, seq=5/1280, ttl=61 (reply in 11)
11	19.171228	192.168.2.2	192.168.1.1	ICMP	98	64	Echo (ping) reply id=0x0f35, seq=5/1280, ttl=64 (request in 10)

Figure 1.4: Wireshark - Ping command from PC1 (192.168.1.1) to PC2 (192.168.2.2) - Listening to link between PC2 and R2

2 Mission 1 - Site-to-Site Tunnel

2.1 Question - Initial route

Initial route

Make a traceroute (before doing this mission). What is the route ?

The trace command results (see figure 2.1 and 2.2) gives us the following route :

PC1 ↔ R1 ↔ Internet ↔ R2 ↔ PC2

```
PC1> trace 192.168.2.2
trace to 192.168.2.2, 8 hops max, press Ctrl+C to stop
 1   192.168.1.101   10.077 ms  8.779 ms  8.963 ms
 2   10.0.1.31   33.720 ms  28.625 ms  30.917 ms
 3   10.0.2.23   51.020 ms  41.646 ms  52.313 ms
 4   *192.168.2.2   75.195 ms (ICMP type:3, code:3, Destination port unreachable
)
```

Figure 2.1: Trace command result from PC1 (192.168.1.1) to PC2 (192.168.2.2)

```
PC2> trace 192.168.1.1
trace to 192.168.1.1, 8 hops max, press Ctrl+C to stop
 1   192.168.2.202   9.865 ms  10.548 ms  9.915 ms
 2   10.0.2.32   19.277 ms  21.417 ms  21.869 ms
 3   10.0.1.13   30.454 ms  51.297 ms  41.334 ms
 4   *192.168.1.1   72.572 ms (ICMP type:3, code:3, Destination port unreachable
)
```

Figure 2.2: Trace command result from PC2 (192.168.2.2) to PC1 (192.168.1.1)

2.2 Question - A new route

A new route

Make a traceroute (after doing this mission). What is the route ? Using wireshark, Analyse the ping between Router1 and Router2. What do you observe ? What is physically the route used by the packet ? What is the route seen by the packet's point of view ? Explain precisely what is happening. Put a screenshot and highlight the encapsulation.

The trace command results are shown in the figures 2.3 and 2.4. For **PC1 → PC2**, we have the route :

PC1 (192.168.1.1) → **R1** (192.168.1.101) → **R2** (172.16.1.2) → **PC2** (192.168.2.2)

```
PC1> trace 192.168.2.2
trace to 192.168.2.2, 8 hops max, press Ctrl+C to stop
 1   192.168.1.101   10.938 ms  9.015 ms  10.476 ms
 2   172.16.1.2   51.120 ms  41.290 ms  51.195 ms
 3   *192.168.2.2   76.971 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 2.3: Trace command result from PC1 to PC2 – bi-directional tunnel

```
PC2> trace 192.168.1.1
trace to 192.168.1.1, 8 hops max, press Ctrl+C to stop
 1   192.168.2.202   12.435 ms  9.155 ms  8.174 ms
 2   172.16.1.1   51.033 ms  52.729 ms  52.643 ms
 3   *192.168.1.1   59.898 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 2.4: Trace command result from PC2 to PC1 – bi-directional tunnel

By using Wireshark and analysing the traffic for a trace command from PC1 to PC2 (see figures 2.5, 2.6, 2.7), we can observe the following :

- The **physical** route used by the packets is the same as before (before setting up the tunnel), see the figure 2.1 : **PC1 ↔ R1 ↔ Internet ↔ R2 ↔ PC2**.
- The route seen by the packets' point of view is the route given by the trace command (figure 2.3) : from their point of view, they don't go through the router Internet but rather **through the tunnel** we've configured.
- When packets from PC1 arrive at R1 (the same applies from PC2 to R2), a new IPv4 header is added on top of the original one. This process is called **encapsulation** and it is due to the tunnel we've set up in the router.
- When PC1 gets an ICMP reply from router R2, the IP source of this reply is 172.16.1.2 and not 10.0.2.23 (the IP destination of the new IPv4 header added in the UDP packet). Also we don't get an ICMP reply from the Internet router. That means the TTL in the UDP packet isn't decreased by this router but directly by R2.

From these observations, we can explain what is happening when we ping PC2 from PC1[1] :

1. PC1 creates UDP packets with an IPv4 header in which the source IP address is PC1's IP address (192.168.1.1) and destination IP address is PC2's IP address (192.168.2.2)
2. The packet arrives at R1. At this stage, the router encapsulates the packet (see figure 2.5) : the previous IPv4 header and payload become **the new payload** and a **new IPv4 header** is created with the source IP address being the tunnel source IP address (10.0.1.13) and the destination IP address being the tunnel destination IP address (10.0.2.23)
3. The packet travels from R1 to R2 by going through the router Internet physically but from the packet points of view, it goes through the tunnel. That's why the TTL isn't decreased by the router Internet.
4. The packet arrives at R2. This time, the router decapsulates the packet (see figure 2.7) : the new IPv4 header is removed and the previous IPv4 header in the payload is reestablished as the IP header.
5. Packet arrives at PC2.

[1] We won't explain in detail here how the ping command works

Remark : we've configured a bi-directional tunnel

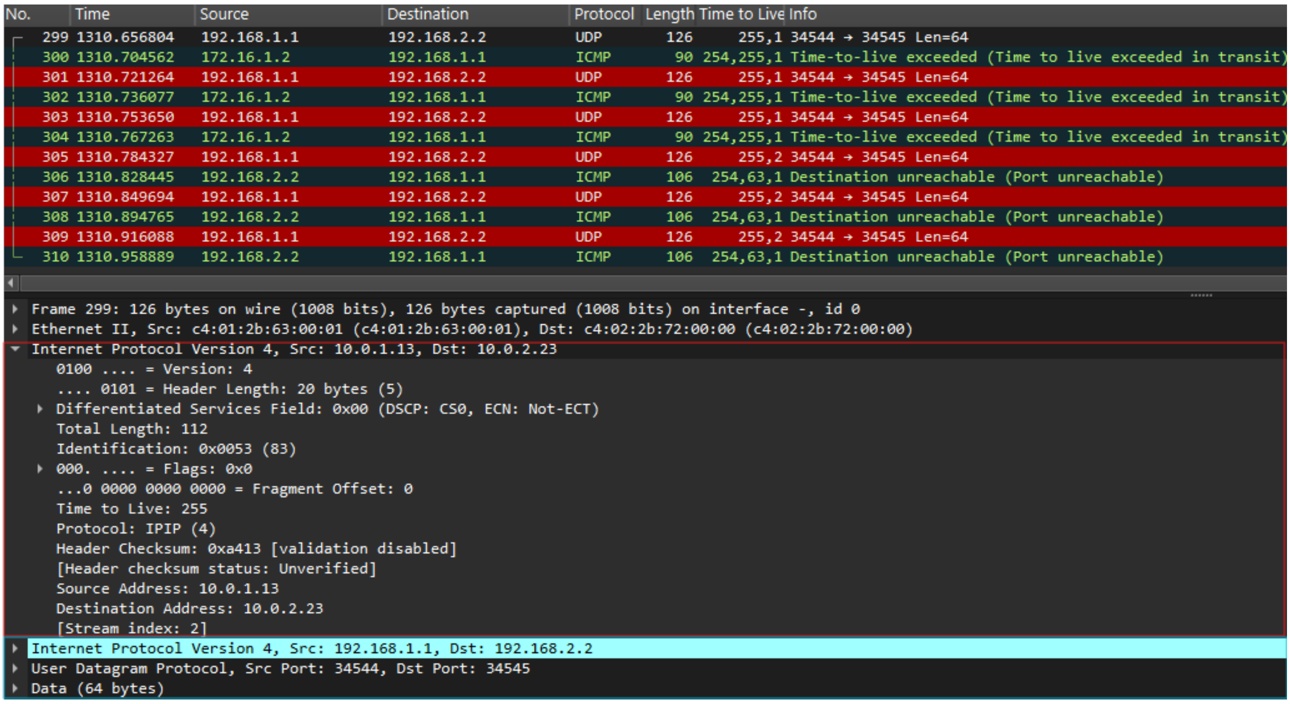

Figure 2.5: Wireshark screenshot - Trace command from PC1 to PC2 - Listening to link between R1 and Internet - in the red frame is the IPv4 header of the encapsulation and in the blue frame the payload of the encapsulation

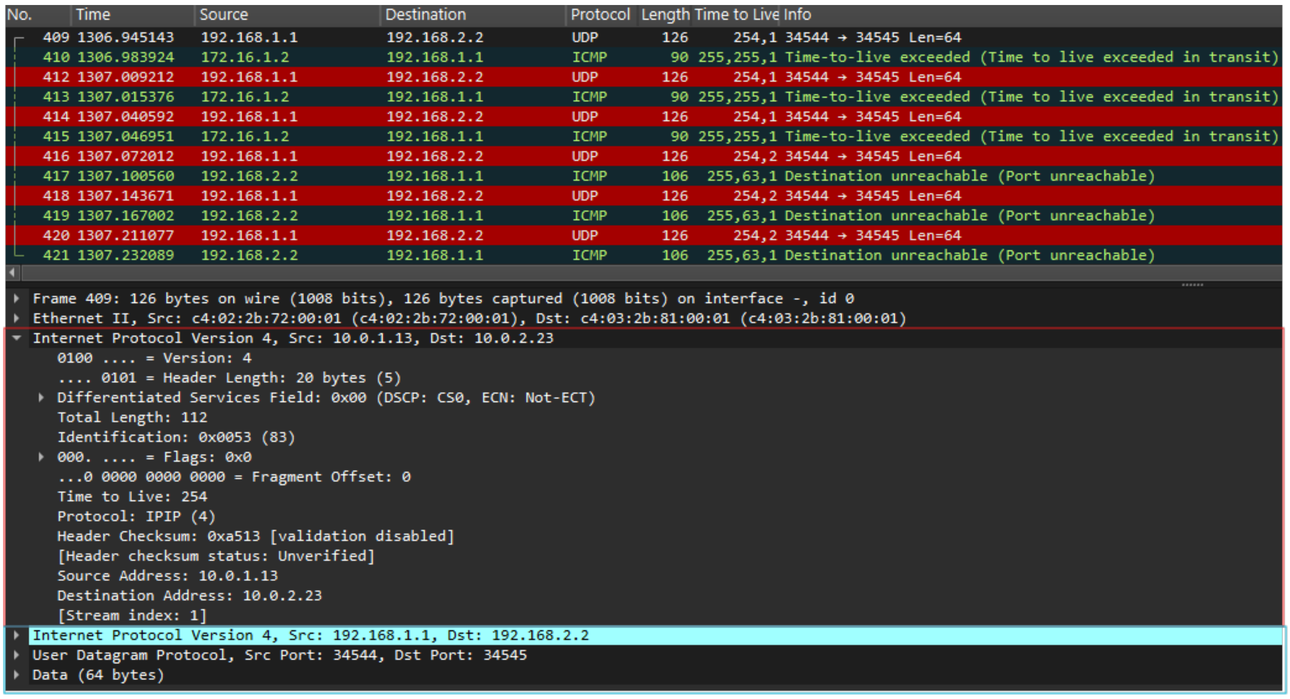

Figure 2.6: Wireshark screenshot - Trace command from PC1 to PC2 - Listening to link between R2 and Internet - in the red frame is the IPv4 header of the encapsulation and in the blue frame the payload of the encapsulation

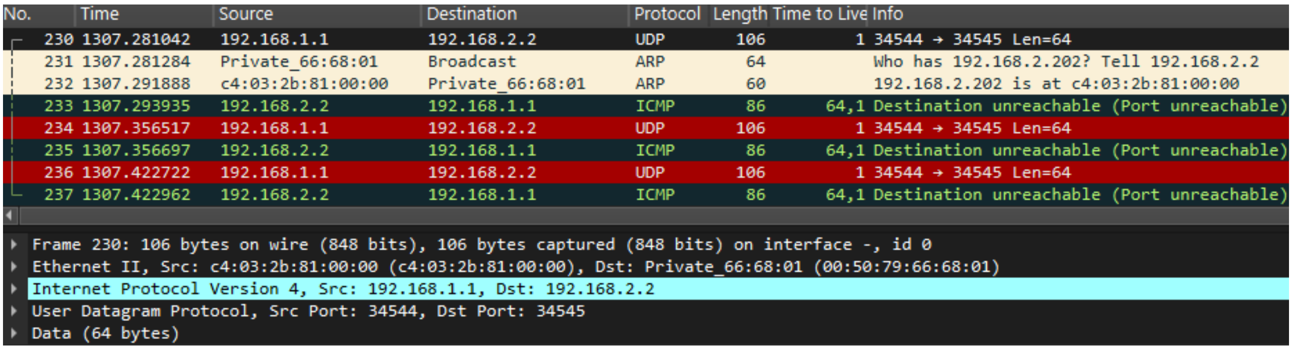

Figure 2.7: Wireshark screenshot - Trace command from PC1 to PC2 - Listening to link between R2 and PC2

3 Mission 2 - Deploy IPv6

3.1 Question - IPv6 compatibility

IPv6 compatibility

From PC1, ping Router1 and PC2 IPv4 and IPv6 addresses. What is working ? Why ?

Pinging the IPv4 addresses of **R1** (192.168.1.101) and of **PC2** (192.168.2.2) works fine (see figure 3.1). This is not the case for IPv6 addresses : only **R1** (FC00:1::101) replies to the ping as shown in the figure 3.2.

- **PC1 → R1** : *IPv4* and *IPv6* works because PC1 and R1 are directly connected.
- **PC1 → PC2** : *IPv4* and *IPv6* aren't compatible and the routers along the path know how to handle *IPv4* addresses but not *IPv6* addresses. The ping requests for IPv6 addresses don't go any further than R1 (FC00:1::101 replies "No route to destination").

```
PC1> ping 192.168.1.101

84 bytes from 192.168.1.101 icmp_seq=1 ttl=255 time=9.072 ms
84 bytes from 192.168.1.101 icmp_seq=2 ttl=255 time=5.962 ms
84 bytes from 192.168.1.101 icmp_seq=3 ttl=255 time=8.142 ms
84 bytes from 192.168.1.101 icmp_seq=4 ttl=255 time=5.271 ms
84 bytes from 192.168.1.101 icmp_seq=5 ttl=255 time=20.942 ms

PC1> ping 192.168.2.2

84 bytes from 192.168.2.2 icmp_seq=1 ttl=62 time=67.838 ms
84 bytes from 192.168.2.2 icmp_seq=2 ttl=62 time=42.031 ms
84 bytes from 192.168.2.2 icmp_seq=3 ttl=62 time=54.648 ms
84 bytes from 192.168.2.2 icmp_seq=4 ttl=62 time=43.195 ms
84 bytes from 192.168.2.2 icmp_seq=5 ttl=62 time=64.522 ms
```

Figure 3.1: Ping from PC1 (192.168.1.1) to R1 (192.168.1.101) and PC2 (192.168.2.2) IPv4 addresses

```
PC1> ping FC00:1::101

FC00:1::101 icmp6_seq=1 ttl=64 time=10.394 ms
FC00:1::101 icmp6_seq=2 ttl=64 time=3.717 ms
FC00:1::101 icmp6_seq=3 ttl=64 time=6.490 ms
FC00:1::101 icmp6_seq=4 ttl=64 time=10.998 ms
FC00:1::101 icmp6_seq=5 ttl=64 time=2.182 ms

PC1> ping FC00:2::2

*fc00:1::101 icmp6_seq=1 ttl=64 time=2.270 ms (ICMP type:1, code:0, No route to
destination)
*fc00:1::101 icmp6_seq=2 ttl=64 time=5.363 ms (ICMP type:1, code:0, No route to
destination)
*fc00:1::101 icmp6_seq=3 ttl=64 time=3.558 ms (ICMP type:1, code:0, No route to
destination)
*fc00:1::101 icmp6_seq=4 ttl=64 time=6.822 ms (ICMP type:1, code:0, No route to
destination)
*fc00:1::101 icmp6_seq=5 ttl=64 time=6.214 ms (ICMP type:1, code:0, No route to
destination)
```

Figure 3.2: Ping from PC1 (FC00:1::1) to R1 (FC00:1::101) and PC2 (FC00:2::2) IPv6 addresses

3.2 Question - IPv6 format

IPv6 format

FC00:1::1/64 is the "compressed" representation of the IPv6 address. Write the "expanded" representation (in hexadecimal is enough). How many bits are needed to encode this address ? Which part of the address represent the subnet and which part represent the host ? How many differents IPv6 addresses could you have in this subnet ?

- The expanded form of the IPv6 address is FC00:0001:0000:0000:0000:0000:0000:0001/64.
- We need 128 bits to encode this address.
- **Subnet** : *FC00:0001:0000:0000* (the last[1] 64 bits)
- **Host** : *0000:0000:0000:0001* (the first 64 bits)
- We can define 2^{64} IPv6 addresses for this specific subnet.

[1] We consider that the MSB is at the left

4 Mission 3 - IPv6-in-IPv4 Tunnel

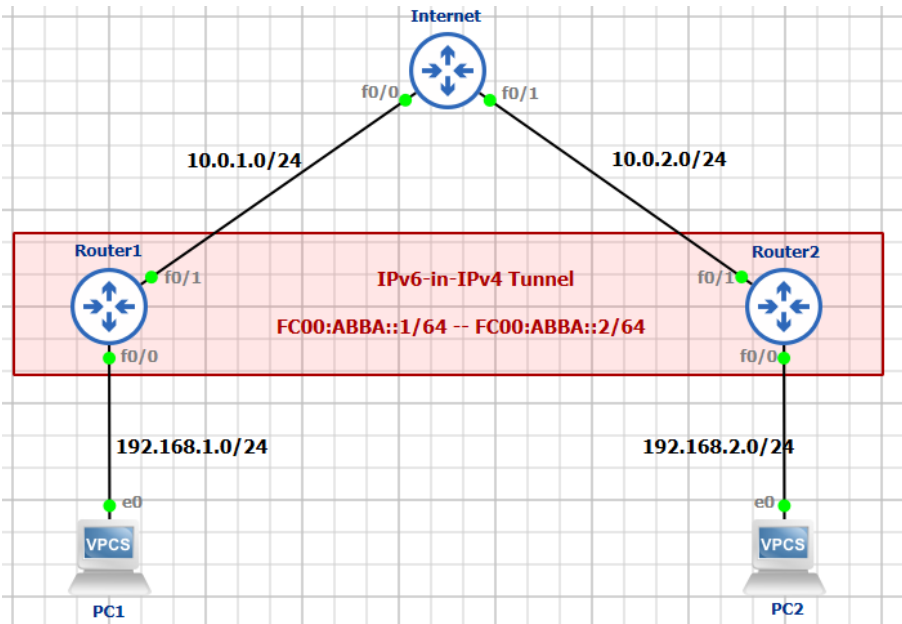

Figure 4.1: New topology - 6in4 tunnel

4.1 Question - Different routes

Different routes

Do a ping and traceroute command between PC1 and PC2 in IPv4 and IPv6. What can you obeserve ? What are the difference ? Explain with you words, what is happening. Put a Wireshark screenshot and highlight important information from the IPv6 ping.

Considering the incompatibility of **IPv4** and **IPv6**, the idea for the last mission is to use the tunnel to encapsulate the **IPv6** packet at the router **R1** by an **IPv4** header and decapsulate the **IPv4** header at the router **R2**, same thing as what we did for the *mission 1*.

By proceeding in this way, we resolve the incompatibility between the networks and we are able to use **IPv6** addresses for the ping command. The encapsulation is shown in a Wireshark screenshot (see figure 4.2).

No.	Time	Source	Destination	Protocol	Length	Time to Live	Info
62	106.168182	fc00:1::1	fc00:2::2	UDP	146	255	21728 → 21729 Len=64
63	106.201654	fc00:abba::2	fc00:1::1	ICMPv6	194	254	Time Exceeded (Hop limit exceeded in transit)
64	106.233726	fc00:1::1	fc00:2::2	UDP	146	255	21728 → 21729 Len=64
65	106.267206	fc00:abba::2	fc00:1::1	ICMPv6	194	254	Time Exceeded (Hop limit exceeded in transit)
66	106.287062	fc00:1::1	fc00:2::2	UDP	146	255	21728 → 21729 Len=64
67	106.298067	fc00:abba::2	fc00:1::1	ICMPv6	194	254	Time Exceeded (Hop limit exceeded in transit)
68	106.319171	fc00:1::1	fc00:2::2	UDP	146	255	21728 → 21729 Len=64
69	106.360820	fc00:2::2	fc00:1::1	ICMPv6	194	254	Destination Unreachable (Port unreachable)[Malformed Packet]
70	106.381987	fc00:1::1	fc00:2::2	UDP	146	255	21728 → 21729 Len=64
71	106.423556	fc00:2::2	fc00:1::1	ICMPv6	194	254	Destination Unreachable (Port unreachable)[Malformed Packet]
72	106.443827	fc00:1::1	fc00:2::2	UDP	146	255	21728 → 21729 Len=64
73	106.485506	fc00:2::2	fc00:1::1	ICMPv6	194	254	Destination Unreachable (Port unreachable)[Malformed Packet]

```
Frame 62: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface -, id 0
Ethernet II, Src: c4:01:2b:63:00:01 (c4:01:2b:63:00:01), Dst: c4:02:2b:72:00:00 (c4:02:2b:72:00:00)
Internet Protocol Version 4, Src: 10.0.1.13, Dst: 10.0.2.23
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 132
    Identification: 0x0135 (309)
  000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 255
    Protocol: IPv6 (41)
    Header Checksum: 0xa2f8 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.0.1.13
    Destination Address: 10.0.2.23
    [Stream index: 0]
Internet Protocol Version 6, Src: fc00:1::1, Dst: fc00:2::2
User Datagram Protocol, Src Port: 21728, Dst Port: 21729
Data (64 bytes)
```

Figure 4.2: Wireshark screenshot - Trace command from PC1 (FC00:1::1) to PC2 (FC00:2::2) - Listening to link between R1 and Internet - in the red frame is the IPv4 header and in the blue frame the IPv6 payload

It should be mentionned that the tunnel mode has been reset from *mode ipip* to *mode ipv6ip*. It implies that, from now on, only the **IPv6** packets are encapsulated by an **IPv4** header at the tunnel.

The **IPv4** packets are no longer encapsulated as it was previously the case in the *mission 1*. This mode change can be noticed by comparing the trace command results for **IPv4** addresses and **IPv6** addresses, as shown in the figure 4.3.

In fact, the route for the **IPv4** packet's point of view is the route passing by all the routers : **R1 ↔ Internet ↔ R2**. In the case of the **IPv6** packet perspective, the packet is going through the tunnel as it is encapsulated : **R1 ↔ R2**.

Physically, this is the same case as in the *mission 1*, both **IPv4** and **IPv6** packets are taking the same route : **R1 ↔ Internet ↔ R2**.

```
PC1> trace FC00:2::2

trace to FC00:2::2, 64 hops max
 1 fc00:1::101   10.284 ms  10.265 ms  9.628 ms
 2 fc00:abba::2   50.391 ms  50.925 ms  41.554 ms
 3 fc00:2::2   50.932 ms  51.626 ms  60.680 ms

PC1> trace 192.168.2.2
trace to 192.168.2.2, 8 hops max, press Ctrl+C to stop
 1   192.168.1.101   3.436 ms  10.181 ms  10.024 ms
 2   10.0.1.31   27.287 ms  18.954 ms  20.172 ms
 3   10.0.2.23   50.427 ms  40.762 ms  40.411 ms
 4   *192.168.2.2   51.229 ms (ICMP type:3, code:3, Destination port unreachable)
```

Figure 4.3: Trace route command from PC1 (192.168.1.1 / FC00:1::1) to PC2 (192.168.2.2 / FC00:2::2)

4.2 Question - Modern tunnels

Modern tunnels

Why the concept of tunnel and encapsulation is really important in modern internet (IPv6) ?

Many networks are still using the IPv4 (see figure 4.4) and the problem is that it isn't compatible with IPv6. To be able to use IPv6 within older IPv4 networks, we resort to tunnels and encapsulation process to use IPv6 packets as IPv4 payloads (or the opposite depending on the route taken by the packet).

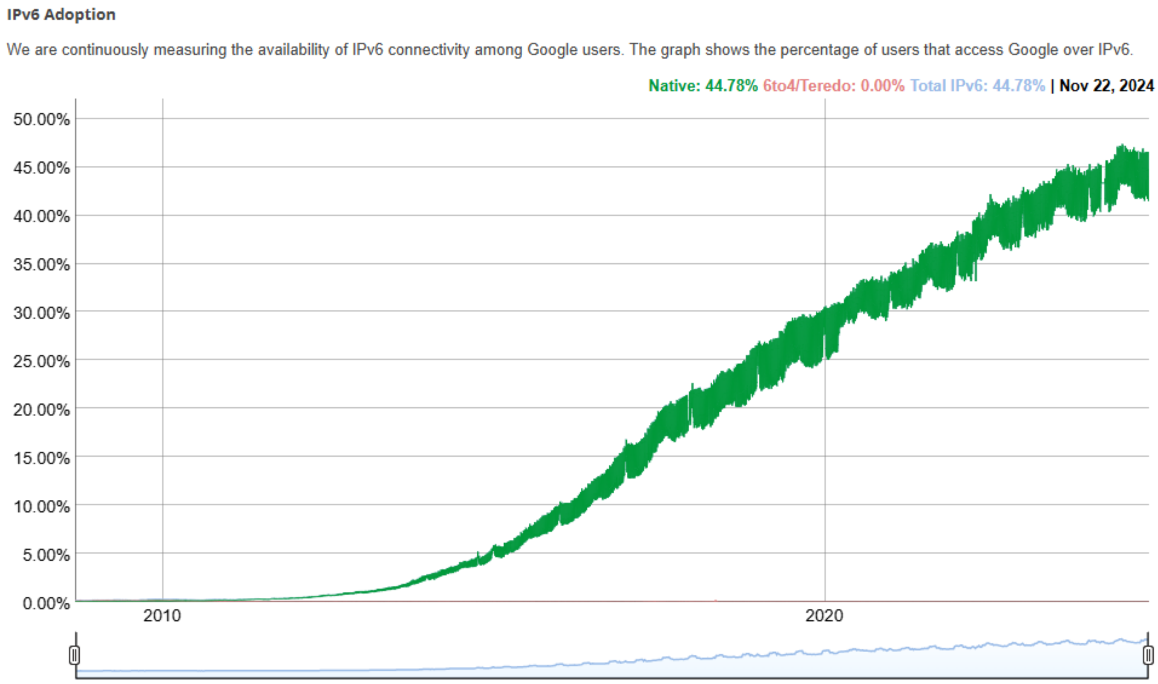

Figure 4.4: IPv6 Adoption - Percentage of users that acces Google over IPv6 over time - source `https://www.google.com/intl/en/ipv6/statistics.html`