Network — IPv4 Tunnel over TCP/IPv6

SAAOUDI Mohamed

January 19, 2026

Contents

1 Organization, Demonstration

1.1 Recommended Directory Structure

- `src/` : libraries (`iftun`, `extremite`, `traitement`) + `tunnel46d`
- `script/` : network configuration + demonstration scenarios
- `captures/` : screenshots + `.pcap` files (tshark/Wireshark)
- `fig/` : report figures
- `rapport/` : `rapport.tex` + compiled PDF

2 Network Configuration (Part 1)

2.1 Topology and Addressing Plan

The addressing plan used is the one from the instructions: LAN1..LAN4 in IPv4 (/28) and two IPv6 LANs `fc00:1234:1::/64` and `fc00:1234:2::/64`.[1]

Typical addresses (excerpt):

- VM1 : `172.16.2.151/28` (LAN3)
- VM3 : `172.16.2.183/28` (LAN4)
- VM1-6 : `172.16.2.156/28` (LAN3) and IPv6 `fc00:1234:1::16`
- VM3-6 : `172.16.2.186/28` (LAN4) and IPv6 `fc00:1234:2::36`

Figure 1 summarizes the topology and the key addresses used.

2.2 VM2 Failure and Impact

Since VM2 is stopped, VM1 can no longer reach VM3 directly. The IPv4-over-TCP/IPv6 tunnel via VM1-6 and VM3-6 restores connectivity.[2]

The initial ping failure (VM2 down) is visible in Figure 2.

3 TUN Virtual Interface (Part 2)

3.1 Interface Creation (2.1)

The TUN interface is created via a program (library `iftun`) based on the `tunalloc.c` code (non-persistent interface: it disappears when the creating process stops).[3]

The creation of the `tun0` interface is visible in Figure 3.

3.2 Configuration of tun0 (2.2)

3.2.1 Script `configure-tun.sh` (2.2.1)

The instructions require configuring `tun0` with the address `172.16.2.1` and an appropriate mask, via a script.[4]

Example script (to be adapted to your VM):

[1] Addressing table "1.1 Topology and Addressing".

[2] Context "1.2 A great misfortune!".

[3] Instructions "2.1 Creation of the interface".

[4] Point "2.2.1".

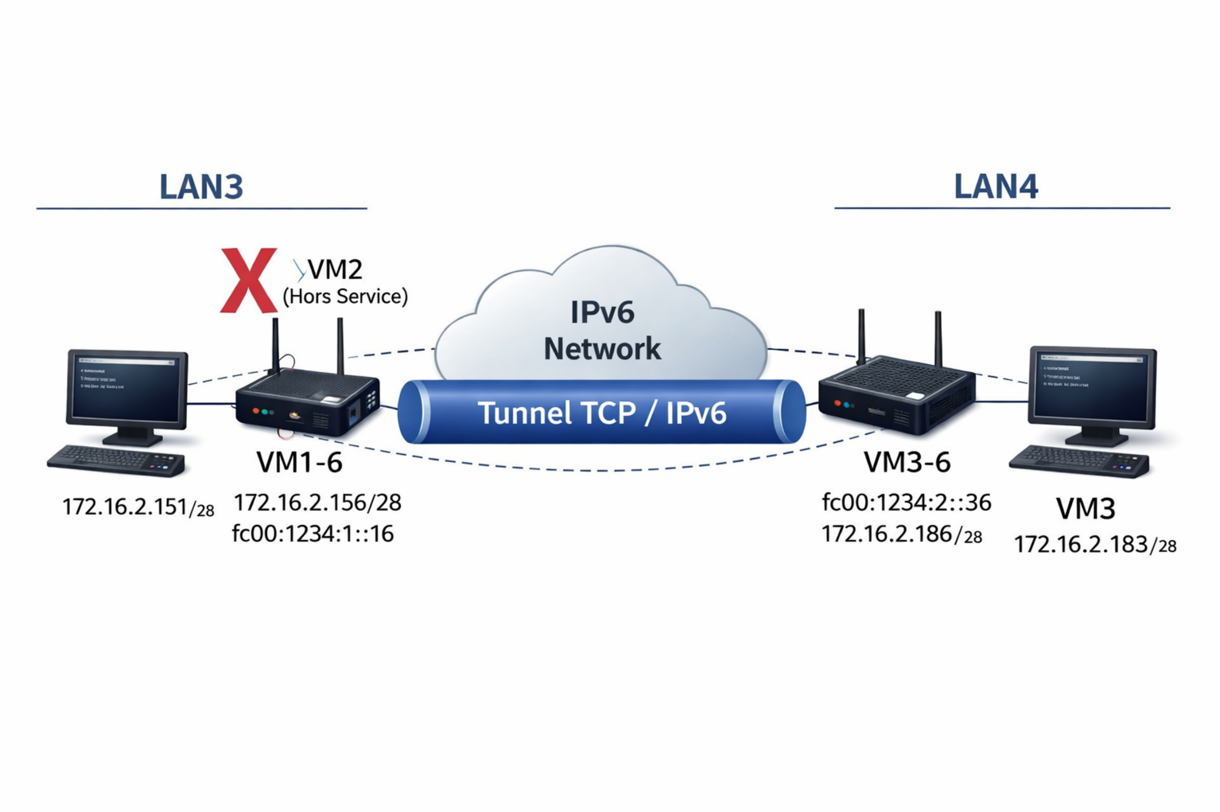
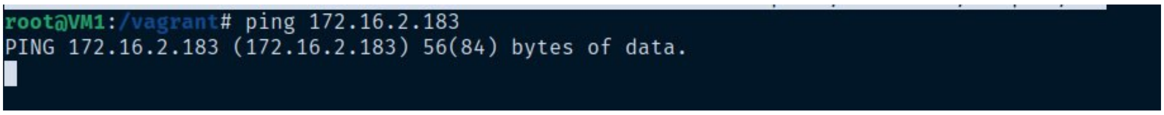
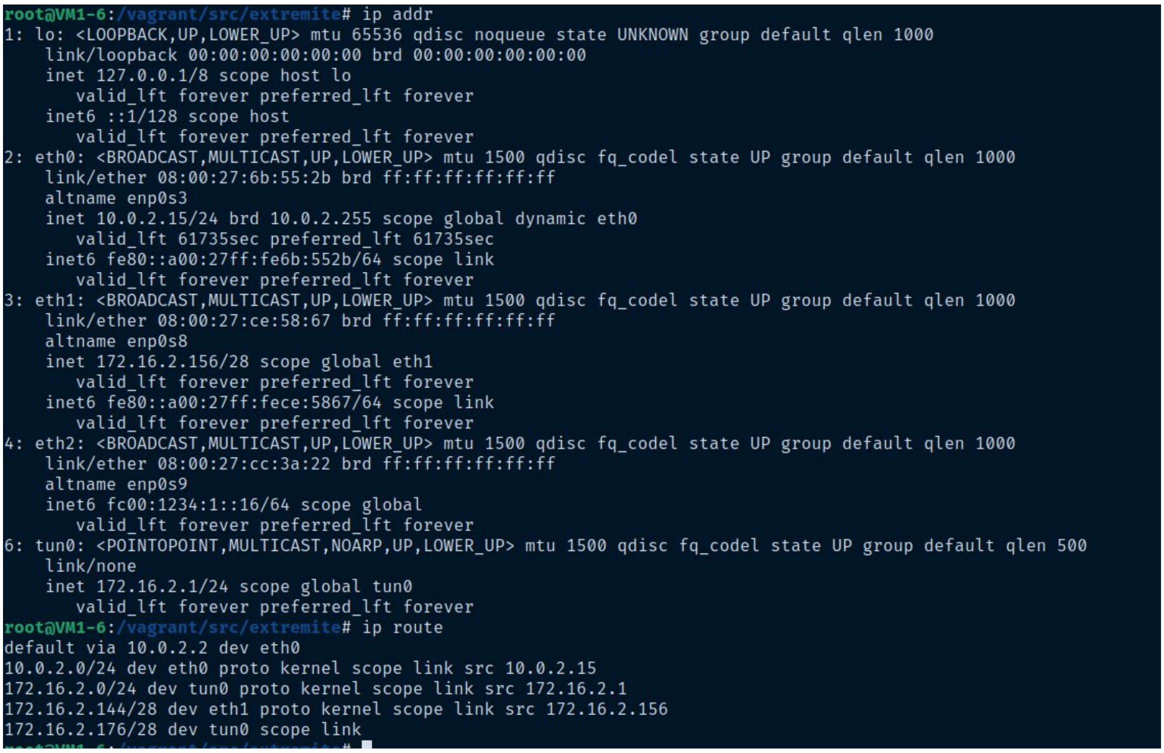


Figure 1: Topology diagram (VM2 down, TCP/IPv6 tunnel between VM1-6 and VM3-6).



```
root@VM1:/vagrant# ping 172.16.2.183
PING 172.16.2.183 (172.16.2.183) 56(84) bytes of data.
```

Figure 2: Ping VM1 → VM3 launched without response while VM2 is down.

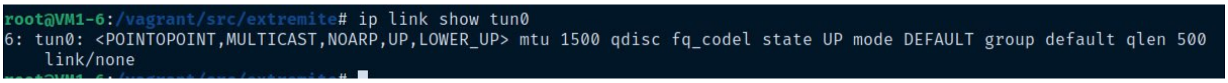


```
root@VM1-6:/vagrant/src/extremite# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6b:55:2b brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 61735sec preferred_lft 61735sec
    inet6 fe80::a00:27ff:fe6b:552b/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ce:58:67 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.156/28 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fece:5867/64 scope link
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cc:3a:22 brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet6 fc00:1234:1::16/64 scope global
       valid_lft forever preferred_lft forever
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 500
    link/none
    inet 172.16.2.1/24 scope global tun0
       valid_lft forever preferred_lft forever
root@VM1-6:/vagrant/src/extremite# ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
172.16.2.0/24 dev tun0 proto kernel scope link src 172.16.2.1
172.16.2.144/28 dev eth1 proto kernel scope link src 172.16.2.156
172.16.2.176/28 dev tun0 scope link
root@VM1-6:/vagrant/src/extremite#
```

Figure 3: Creation of the `tun0` interface.

```
#!/usr/bin/env bash
set -euo pipefail

sudo -s <<'EOS'
modprobe tun 2>/dev/null || true
ip tuntap add dev tun0 mode tun 2>/dev/null || true
ip link set tun0 up

# Address requested by the instructions (example of local config)
ip addr flush dev tun0 || true
ip addr add 172.16.2.1/24 dev tun0
EOS
```

Figure 4 shows the `tun0` interface created and active after executing the script.



```
root@VM1-6:/vagrant/src/extremite# ip link show tun0
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 500
    link/none
```

Figure 4: `tun0` interface created and active (`ip link show tun0` output).

3.2.2 Routing: VM1 or VM1-6? (2.2.2)

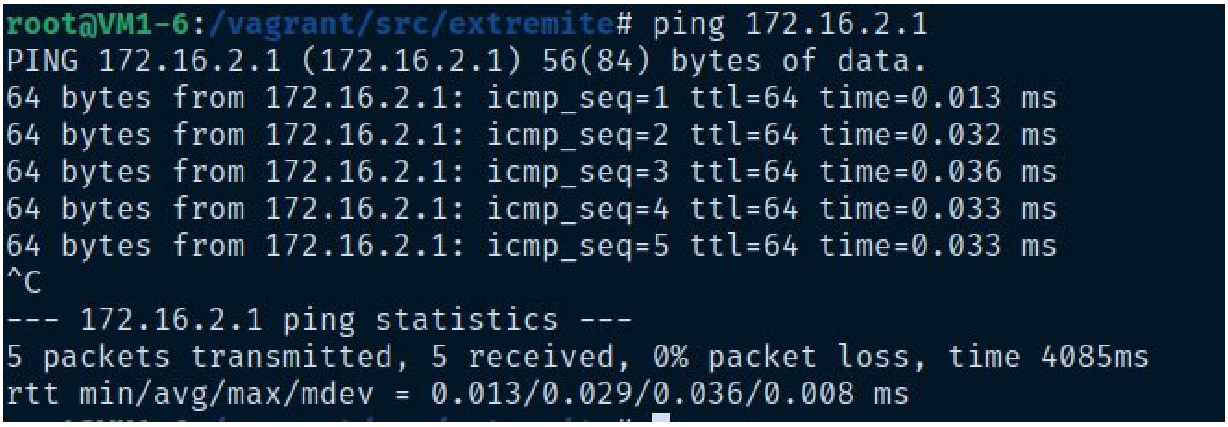
After the VM2 failure, we must **modify the routing on VM1 and on VM1-6**:

- **VM1**: route to VM3's network (LAN4) via VM1-6 (LAN3).
- **VM1-6**: route from LAN4 to `tun0` in order to inject traffic into the tunnel.

This ensures that traffic leaving VM1 is redirected to the endpoint (VM1-6) and then injected into `tun0`.[5]

3.2.3 Local ping tun0 (2.2.3)

Local pings to `tun0` addresses are visible in Figures 5 and 6.



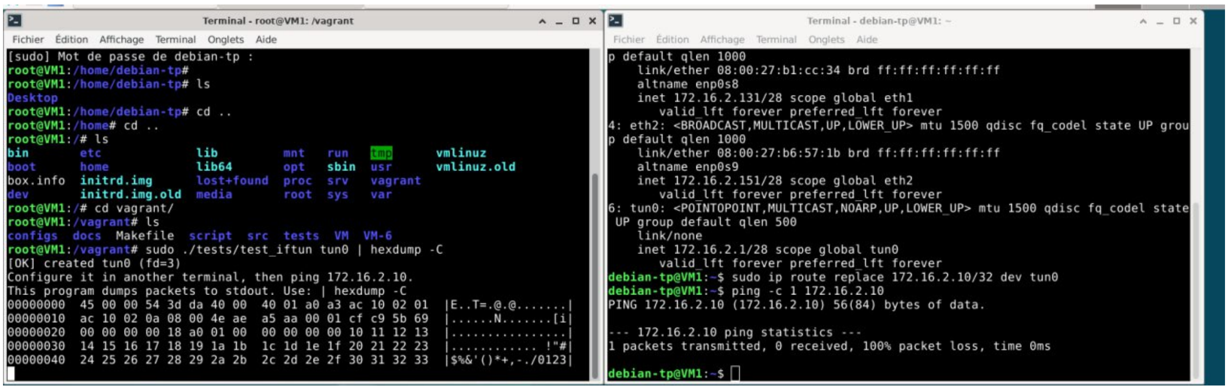
```
root@VM1-6:/vagrant/src/extremite# ping 172.16.2.1
PING 172.16.2.1 (172.16.2.1) 56(84) bytes of data.
64 bytes from 172.16.2.1: icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from 172.16.2.1: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 172.16.2.1: icmp_seq=3 ttl=64 time=0.036 ms
64 bytes from 172.16.2.1: icmp_seq=4 ttl=64 time=0.033 ms
64 bytes from 172.16.2.1: icmp_seq=5 ttl=64 time=0.033 ms
^C
--- 172.16.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.013/0.029/0.036/0.008 ms
```

Figure 5: VM1-6: local ping to `172.16.2.1` (`tun0` address).

3.2.4 Non-local ping via tun0 (2.2.4) and explanation (2.2.5)

The ping to `172.16.2.10` routed to `tun0` without a user-space process is illustrated in Figure 7.

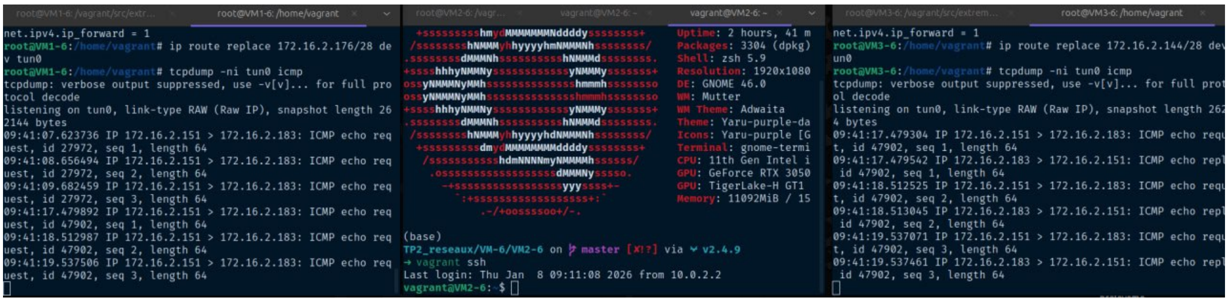
[5] Question "2.2.2".



```
[sudo] Mot de passe de debian-tp :
root@VM1:/home/debian-tp#
root@VM1:/home/debian-tp# ls
Desktop
root@VM1:/home/debian-tp# cd ..
root@VM1:/home# cd ..
root@VM1:/# ls
bin       etc             lib         mnt   run   tmp      vmlinuz
boot      home            lib64       opt   sbin  usr      vmlinuz.old
box.info  initrd.img      lost+found  proc  srv   vagrant
dev       initrd.img.old  media       root  sys   var
root@VM1:/# cd vagrant/
root@VM1:/vagrant# ls
configs  docs  Makefile  script  src  tests  VM  VM-6
root@VM1:/vagrant# sudo ./tests/test_iftun tun0 | hexdump -C
[OK] created tun0 (fd=3)
Configure it in another terminal, then ping 172.16.2.10.
This program dumps packets to stdout. Use: | hexdump -C
00000000  45 00 00 54 3d da 40 00  40 01 a0 a3 ac 10 02 01  |E..T=.@.@.......|
00000010  ac 10 02 0a 08 00 4e ae  a5 aa 00 01 cf c9 5b 69  |......N.......[i|
00000020  00 00 00 00 18 a0 01 00  00 00 00 00 10 11 12 13  |................|
00000030  14 15 16 17 18 19 1a 1b  1c 1d 1e 1f 20 21 22 23  |............ !"#|
00000040  24 25 26 27 28 29 2a 2b  2c 2d 2e 2f 30 31 32 33  |$%&'()*+,-./0123|
```

```
p default qlen 1000
    link/ether 08:00:27:b1:cc:34 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.131/28 scope global eth1
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 08:00:27:b6:57:1b brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet 172.16.2.151/28 scope global eth2
       valid_lft forever preferred_lft forever
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
 UP group default qlen 500
    link/none
    inet 172.16.2.1/28 scope global tun0
       valid_lft forever preferred_lft forever
debian-tp@VM1:~$ sudo ip route replace 172.16.2.10/32 dev tun0
debian-tp@VM1:~$ ping -c 1 172.16.2.10
PING 172.16.2.10 (172.16.2.10) 56(84) bytes of data.

--- 172.16.2.10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

debian-tp@VM1:~$
```

Figure 6: VM3-6: local ping to `172.16.2.10` (`tun0` address).



```
net.ipv4.ip_forward = 1
root@VM1-6:/home/vagrant# ip route replace 172.16.2.176/28 de
v tun0
root@VM1-6:/home/vagrant# tcpdump -ni tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full pro
tocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 26
2144 bytes
09:41:07.623736 IP 172.16.2.151 > 172.16.2.183: ICMP echo req
uest, id 27972, seq 1, length 64
09:41:08.656494 IP 172.16.2.151 > 172.16.2.183: ICMP echo req
uest, id 27972, seq 2, length 64
09:41:09.682459 IP 172.16.2.151 > 172.16.2.183: ICMP echo req
uest, id 27972, seq 3, length 64
09:41:17.479892 IP 172.16.2.151 > 172.16.2.183: ICMP echo req
uest, id 47902, seq 1, length 64
09:41:18.512987 IP 172.16.2.151 > 172.16.2.183: ICMP echo req
uest, id 47902, seq 2, length 64
09:41:19.537506 IP 172.16.2.151 > 172.16.2.183: ICMP echo req
uest, id 47902, seq 3, length 64
```

```
  +sssssssssshmydMMMMMMMNddddyssssssss+    Uptime: 2 hours, 41 m
 /sssssssshNMMMyhhyyyyhmNMMMNhssssssss/    Packages: 3304 (dpkg)
.ssssssssdMMMNhsssssssssshNMMMdssssssss.   Shell: zsh 5.9
+sssshhhyNMMNyssssssssssssyNMMMysssssss+   Resolution: 1920x1080
ossyNMMMNyMMhsssssssssssssshmmmhssssssso   DE: GNOME 46.0
ossyNMMMNyMMhsssssssssssssshmmmhssssssso   WM: Mutter
+sssshhhyNMMNyssssssssssssyNMMMysssssss+   WM Theme: Adwaita
.ssssssssdMMMNhsssssssssshNMMMdssssssss.   Theme: Yaru-purple-da
 /sssssssshNMMMyhhyyyyhdNMMMNhssssssss/    Icons: Yaru-purple [G
  +sssssssssdmydMMMMMMMMddddyssssssss+     Terminal: gnome-termi
   /ssssssssssshdmNNNNmyNMMMMhssssss/      CPU: 11th Gen Intel i
    .ossssssssssssssssssdMMMNysssso.       GPU: GeForce RTX 3050
      -+sssssssssssssssssyyyssss+-         GPU: TigerLake-H GT1
        `:+ssssssssssssssssss+:`           Memory: 11092MiB / 15
            .-/+oossssoo+/-.

(base)
TP2_reseaux/VM-6/VM2-6 on master [X!?] via v2.4.9
→ vagrant ssh
Last login: Thu Jan  8 09:11:08 2026 from 10.0.2.2
vagrant@VM2-6:~$
```

```
net.ipv4.ip_forward = 1
root@VM3-6:/home/vagrant# ip route replace 172.16.2.144/28 dev
un0
root@VM3-6:/home/vagrant# tcpdump -ni tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full prot
ol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262
4 bytes
09:41:17.479304 IP 172.16.2.151 > 172.16.2.183: ICMP echo requ
t, id 47902, seq 1, length 64
09:41:17.479542 IP 172.16.2.183 > 172.16.2.151: ICMP echo repl
 id 47902, seq 1, length 64
09:41:18.512525 IP 172.16.2.151 > 172.16.2.183: ICMP echo requ
t, id 47902, seq 2, length 64
09:41:18.513045 IP 172.16.2.183 > 172.16.2.151: ICMP echo repl
 id 47902, seq 2, length 64
09:41:19.537071 IP 172.16.2.151 > 172.16.2.183: ICMP echo requ
t, id 47902, seq 3, length 64
09:41:19.537461 IP 172.16.2.183 > 172.16.2.151: ICMP echo repl
 id 47902, seq 3, length 64
```

Figure 7: Left: `test_iftun tun0 | hexdump -C`. Right: ping `172.16.2.10` routed to `tun0` with 100% loss (no response).

3.3 Packet Retrieval (2.3)

3.3.1 Option IFF_NO_PI (2.3.4)

Role: `IFF_NO_PI` asks the kernel not to prefix packets read on `tun0` with a PI header. **Without IFF_NO_PI:** readings begin with extra PI bytes; a naive IPv4 parser would see its data shifted.[6]

4 Simple IPv4 Tunnel over TCP/IPv6 (Part 3)

4.1 Incoming Traffic Redirection (3.1)

The test of the `ext-out` server with TCP/IPv6 reception redirected to stdout is visible in Figure 8.

4.2 Final Integration (3.3)

The setup of the bidirectional tunnel is visible in Figure 9.

4.3 VM1-6 <-> VM3-6 Tunnel: Diagrams (3.4)

The complete diagram is provided in Figure 10.

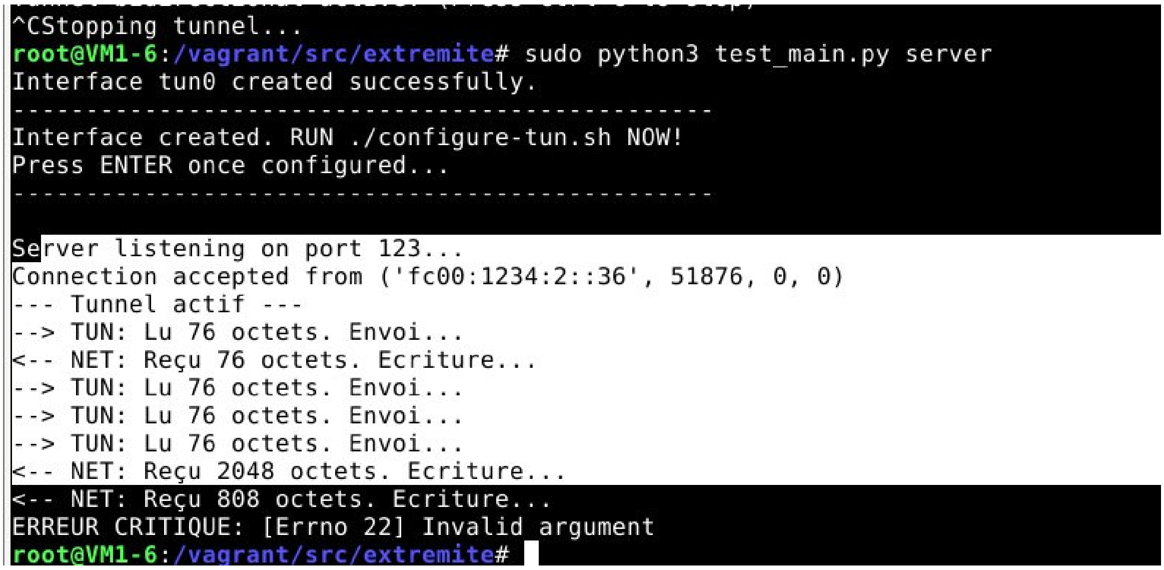
5 Functional Validation (Part 4)

5.1 Configuration (4.1)

On VM1-6, VM1, VM2-6, VM3-6, and VM3: display `ip addr` and `ip route`.[7]

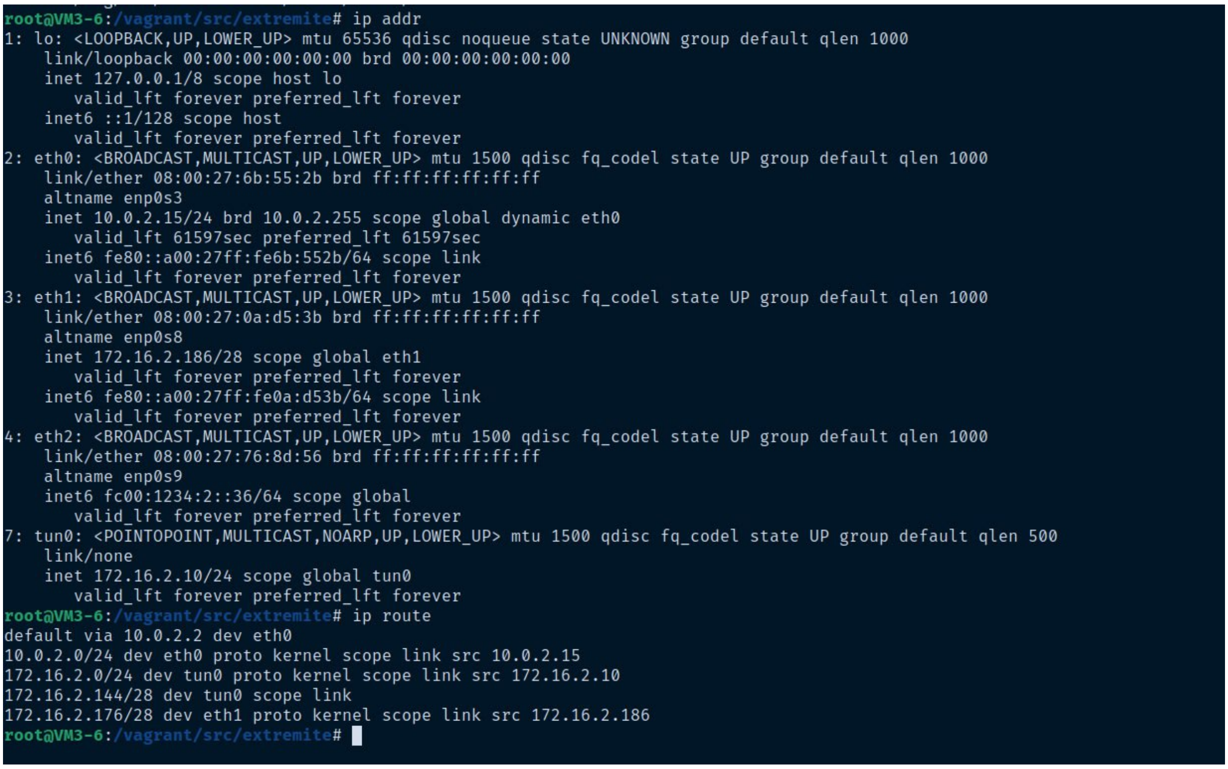
[6] Question "2.3.4".

[7] Point "4.1".



```
^CStopping tunnel...
root@VM1-6:/vagrant/src/extremite# sudo python3 test_main.py server
Interface tun0 created successfully.
-----------------------------------------------
Interface created. RUN ./configure-tun.sh NOW!
Press ENTER once configured...
-----------------------------------------------

Server listening on port 123...
Connection accepted from ('fc00:1234:2::36', 51876, 0, 0)
--- Tunnel actif ---
--> TUN: Lu 76 octets. Envoi...
<-- NET: Reçu 76 octets. Ecriture...
--> TUN: Lu 76 octets. Envoi...
--> TUN: Lu 76 octets. Envoi...
--> TUN: Lu 76 octets. Envoi...
<-- NET: Reçu 2048 octets. Ecriture...
<-- NET: Reçu 808 octets. Ecriture...
ERREUR CRITIQUE: [Errno 22] Invalid argument
root@VM1-6:/vagrant/src/extremite# 
```

Figure 8: `ext-out` test: reception via TCP/IPv6 redirected to stdout.



```
root@VM3-6:/vagrant/src/extremite# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6b:55:2b brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 61597sec preferred_lft 61597sec
    inet6 fe80::a00:27ff:fe6b:552b/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0a:d5:3b brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.186/28 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0a:d53b/64 scope link
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:76:8d:56 brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet6 fc00:1234:2::36/64 scope global
       valid_lft forever preferred_lft forever
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 500
    link/none
    inet 172.16.2.10/24 scope global tun0
       valid_lft forever preferred_lft forever
root@VM3-6:/vagrant/src/extremite# ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
172.16.2.0/24 dev tun0 proto kernel scope link src 172.16.2.10
172.16.2.144/28 dev tun0 scope link
172.16.2.176/28 dev eth1 proto kernel scope link src 172.16.2.186
root@VM3-6:/vagrant/src/extremite# 
```

Figure 9: TCP/IPv6 connection established and bidirectional tunnel active (`test_main.py`).

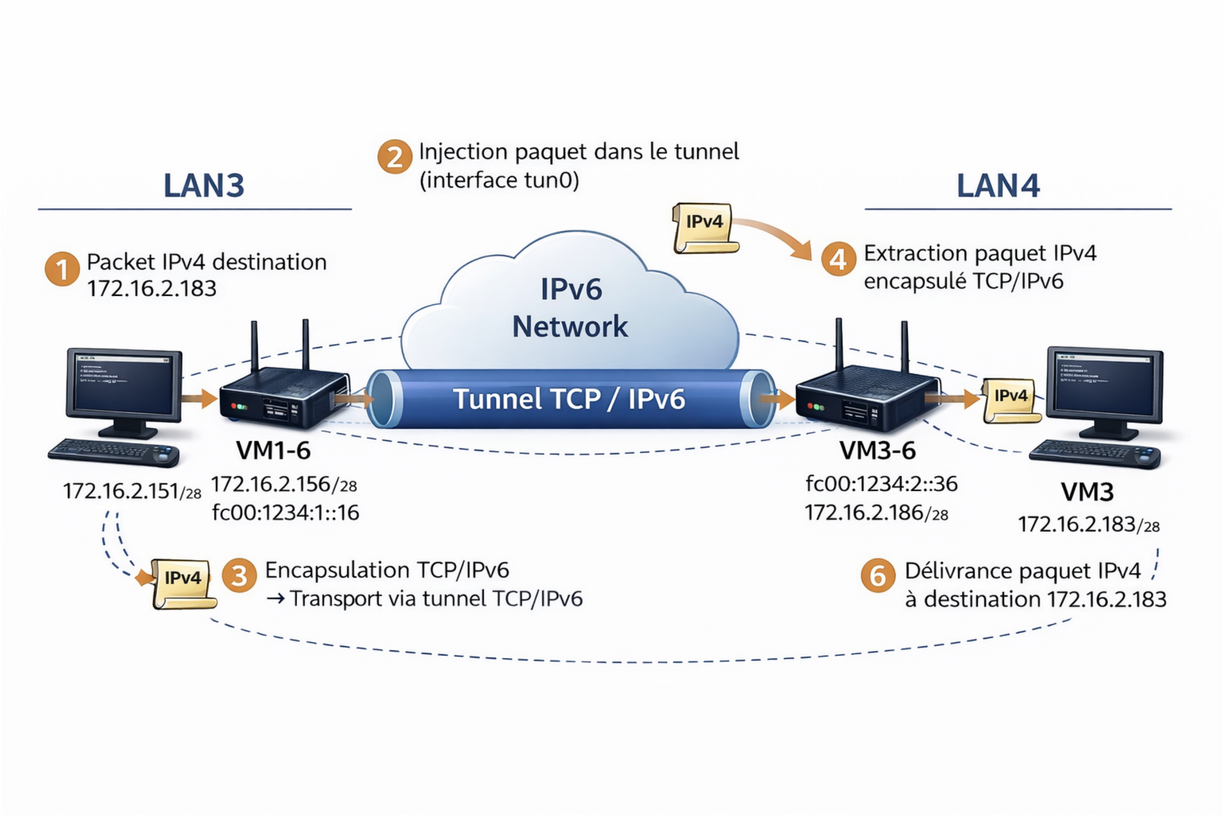
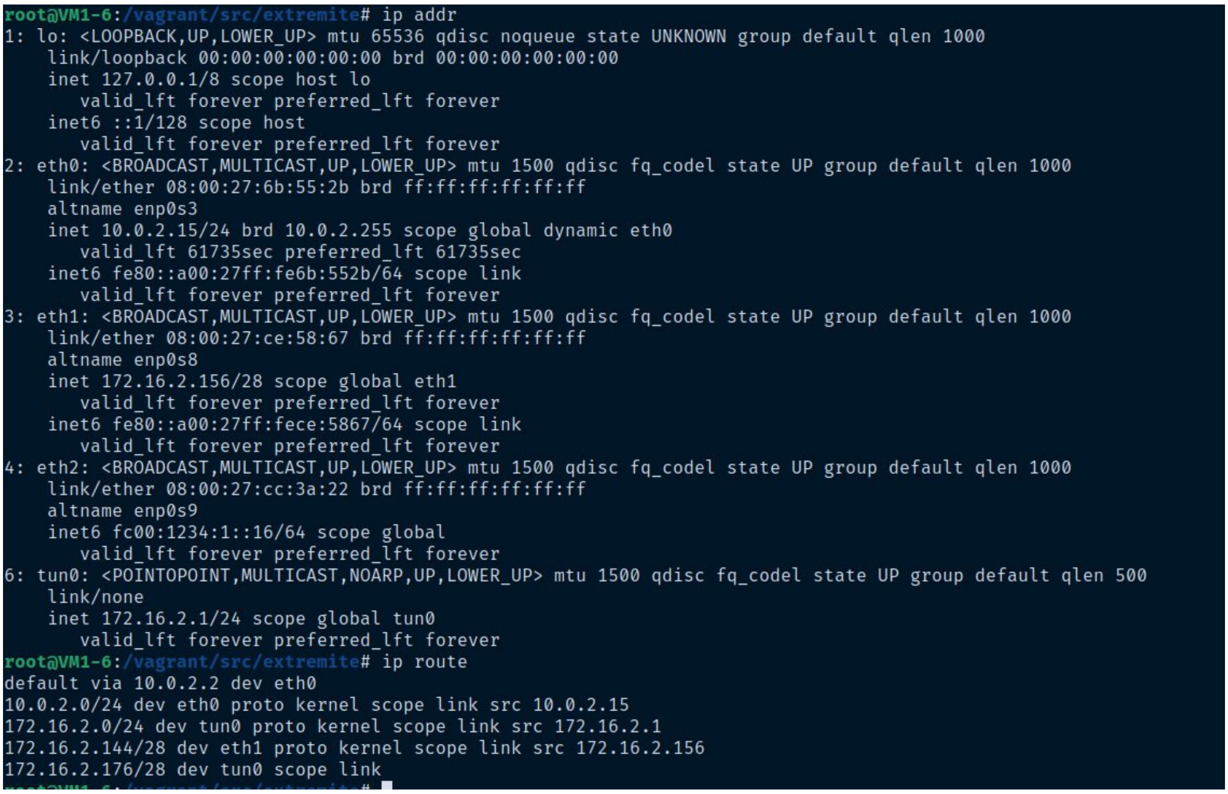
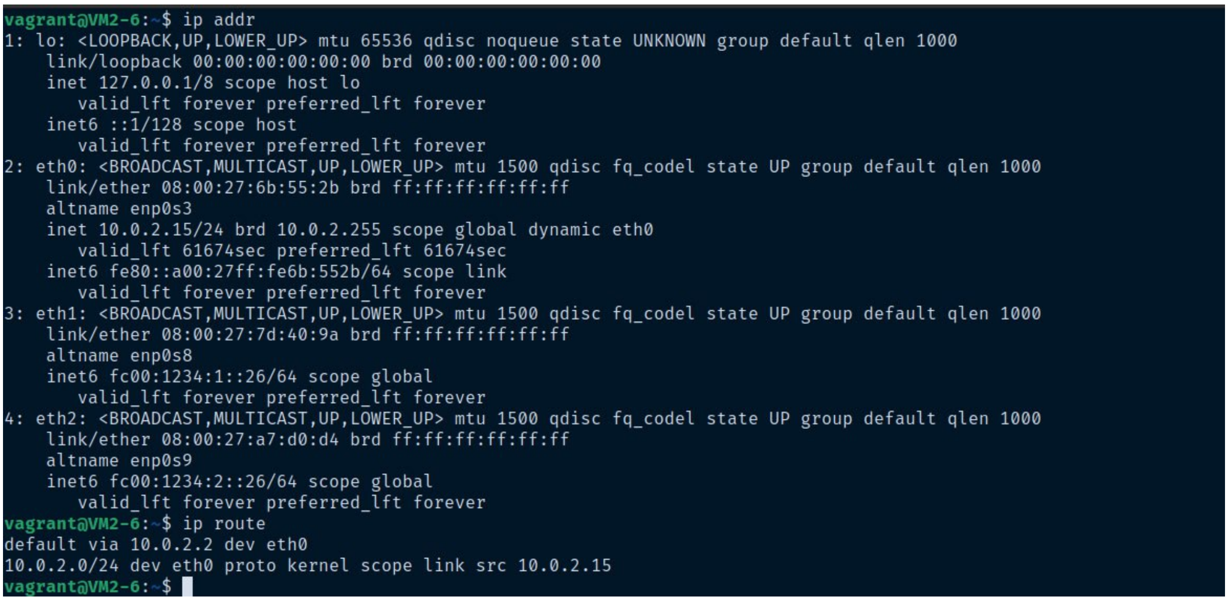


Figure 10: Complete path of an IPv4 packet from VM1 to VM3 via the TCP/IPv6 tunnel.



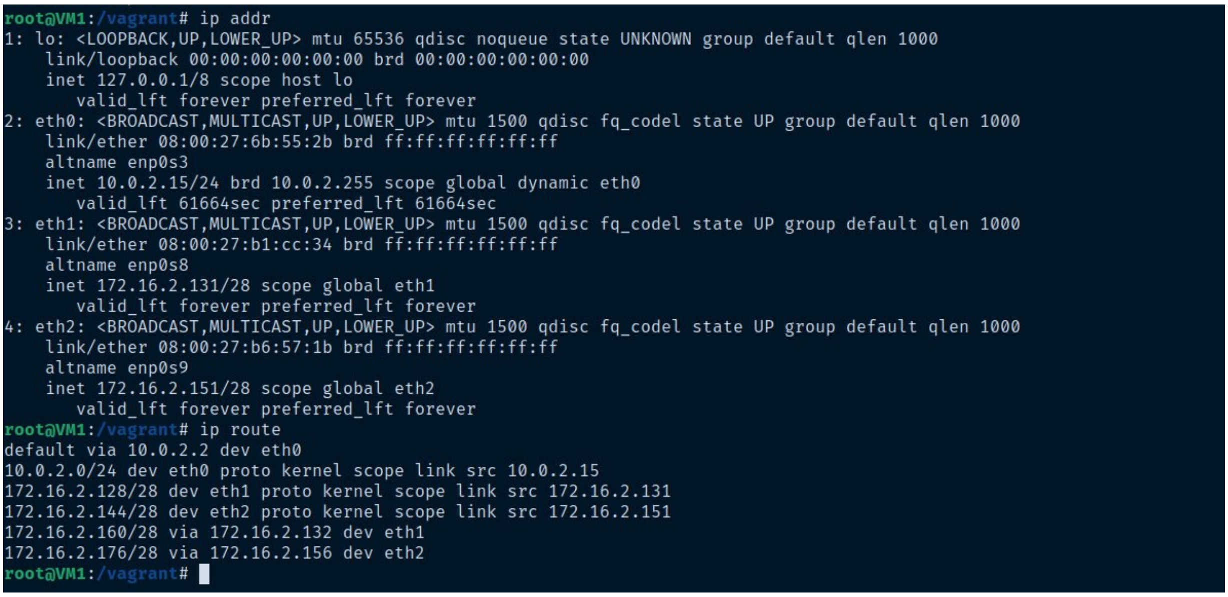
```
root@VM1-6:/vagrant/src/extremite# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6b:55:2b brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 61735sec preferred_lft 61735sec
    inet6 fe80::a00:27ff:fe6b:552b/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ce:58:67 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.156/28 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fece:5867/64 scope link
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cc:3a:22 brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet6 fc00:1234:1::16/64 scope global
       valid_lft forever preferred_lft forever
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 500
    link/none
    inet 172.16.2.1/24 scope global tun0
       valid_lft forever preferred_lft forever
root@VM1-6:/vagrant/src/extremite# ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
172.16.2.0/24 dev tun0 proto kernel scope link src 172.16.2.1
172.16.2.144/28 dev eth1 proto kernel scope link src 172.16.2.156
172.16.2.176/28 dev tun0 scope link
root@VM1-6:/vagrant/src/extremite#
```

Figure 11: VM1-6 `ip addr` and `ip route`.



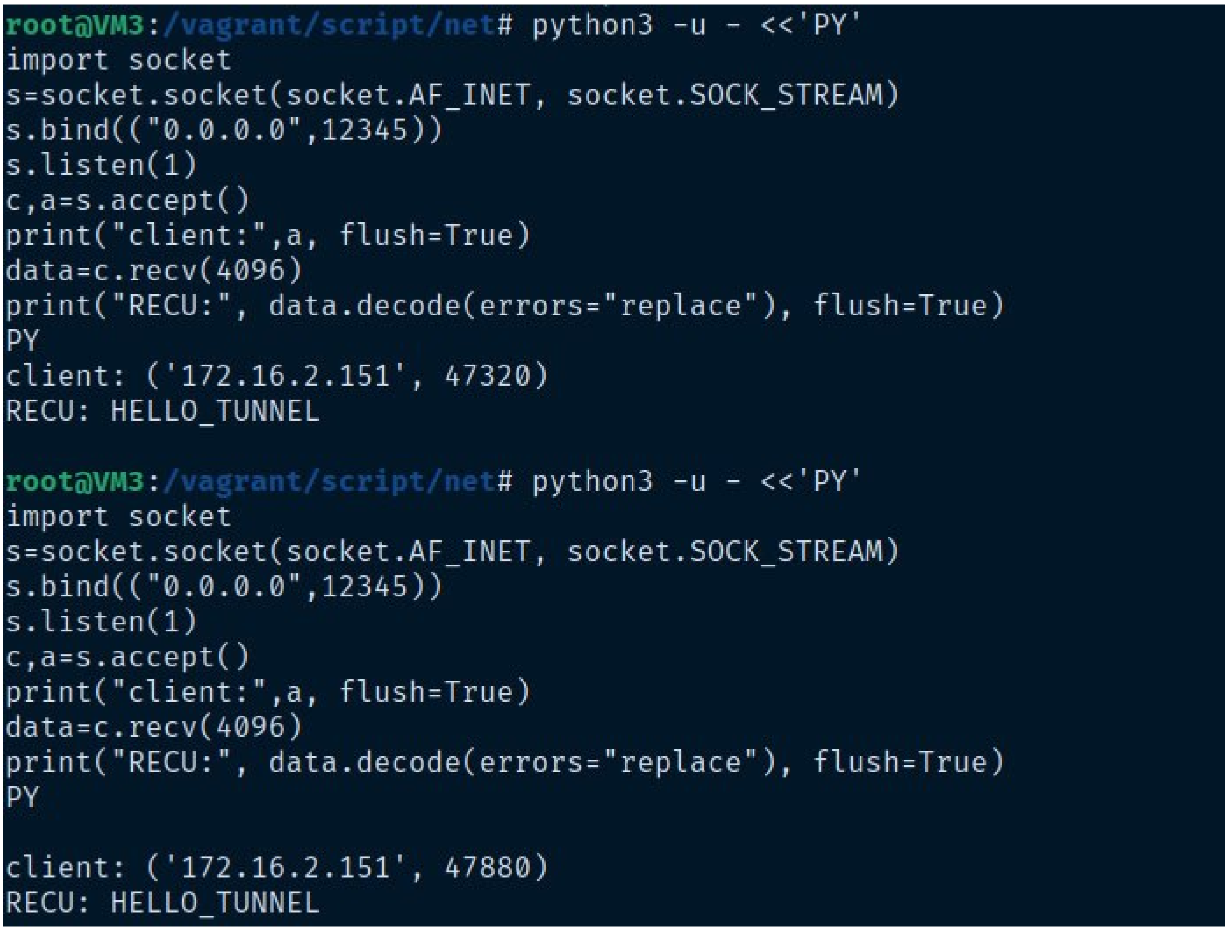
```
vagrant@VM2-6:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6b:55:2b brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 61674sec preferred_lft 61674sec
    inet6 fe80::a00:27ff:fe6b:552b/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:7d:40:9a brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet6 fc00:1234:1::26/64 scope global
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:a7:d0:d4 brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet6 fc00:1234:2::26/64 scope global
       valid_lft forever preferred_lft forever
vagrant@VM2-6:~$ ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
vagrant@VM2-6:~$
```

Figure 12: VM2-6 (IPv4 router) `ip addr` and `ip route`.



```
root@VM1:/vagrant# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6b:55:2b brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 61664sec preferred_lft 61664sec
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:cc:34 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.131/28 scope global eth1
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b6:57:1b brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet 172.16.2.151/28 scope global eth2
       valid_lft forever preferred_lft forever
root@VM1:/vagrant# ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
172.16.2.128/28 dev eth1 proto kernel scope link src 172.16.2.131
172.16.2.144/28 dev eth2 proto kernel scope link src 172.16.2.151
172.16.2.160/28 via 172.16.2.132 dev eth1
172.16.2.176/28 via 172.16.2.156 dev eth2
root@VM1:/vagrant#
```

Figure 13: VM1 `ip addr` and `ip route`.

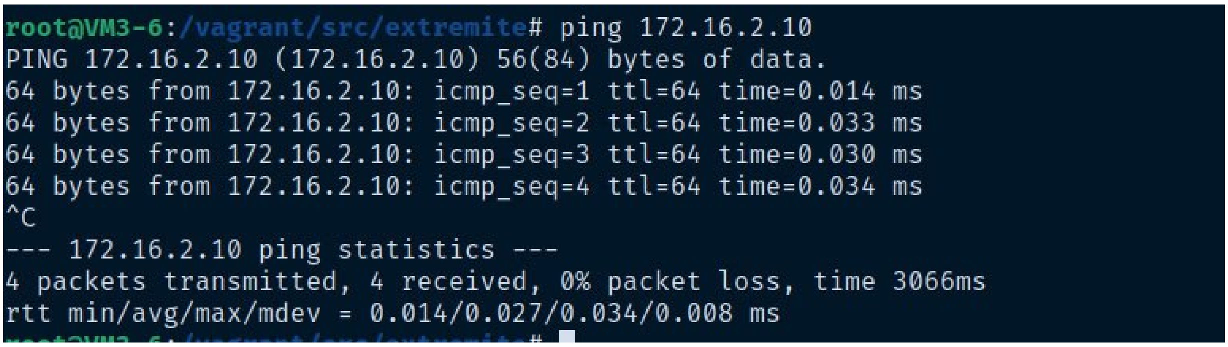


```
root@VM3:/vagrant/script/net# python3 -u - <<'PY'
import socket
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("0.0.0.0",12345))
s.listen(1)
c,a=s.accept()
print("client:",a, flush=True)
data=c.recv(4096)
print("RECU:", data.decode(errors="replace"), flush=True)
PY
client: ('172.16.2.151', 47320)
RECU: HELLO_TUNNEL

root@VM3:/vagrant/script/net# python3 -u - <<'PY'
import socket
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("0.0.0.0",12345))
s.listen(1)
c,a=s.accept()
print("client:",a, flush=True)
data=c.recv(4096)
print("RECU:", data.decode(errors="replace"), flush=True)
PY

client: ('172.16.2.151', 47880)
RECU: HELLO_TUNNEL
```

Figure 14: VM3 `ip addr` and `ip route`.

5.2 Layer 3 (4.2)

From VM1: `ping 172.16.2.183`.[8]

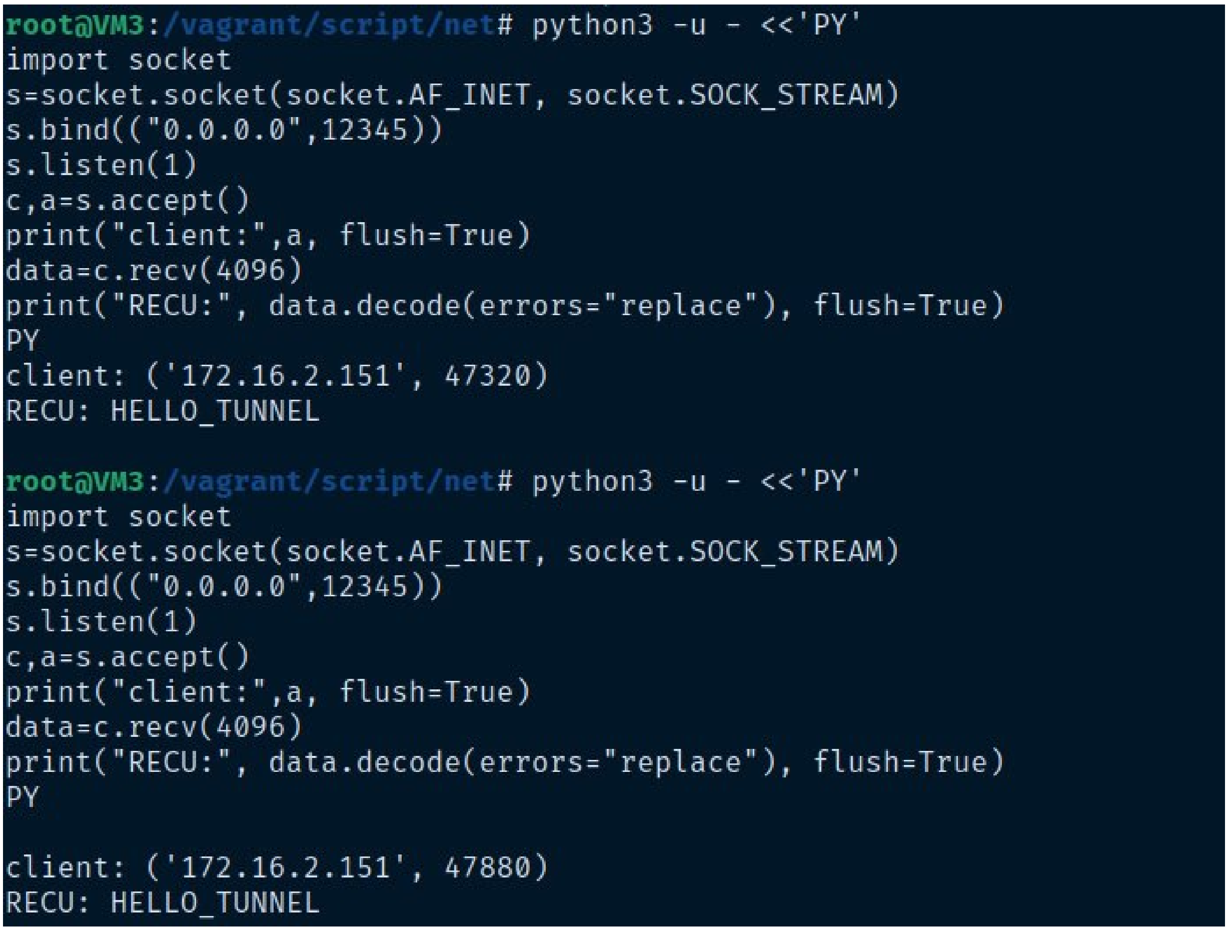


```
root@VM3-6:/vagrant/src/extremite# ping 172.16.2.10
PING 172.16.2.10 (172.16.2.10) 56(84) bytes of data.
64 bytes from 172.16.2.10: icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from 172.16.2.10: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 172.16.2.10: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 172.16.2.10: icmp_seq=4 ttl=64 time=0.034 ms
^C
--- 172.16.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.014/0.027/0.034/0.008 ms
```

Figure 15: Ping VM1 → VM3 (172.16.2.183) and ICMP capture on `tun0` side VM1-6/VM3-6.

5.3 Layer 4 (4.3)

From VM1: create `msg.txt` and send it to VM3 (TCP service).[9]



```
root@VM3:/vagrant/script/net# python3 -u - <<'PY'
import socket
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("0.0.0.0",12345))
s.listen(1)
c,a=s.accept()
print("client:",a, flush=True)
data=c.recv(4096)
print("RECU:", data.decode(errors="replace"), flush=True)
PY
client: ('172.16.2.151', 47320)
RECU: HELLO_TUNNEL

root@VM3:/vagrant/script/net# python3 -u - <<'PY'
import socket
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("0.0.0.0",12345))
s.listen(1)
c,a=s.accept()
print("client:",a, flush=True)
data=c.recv(4096)
print("RECU:", data.decode(errors="replace"), flush=True)
PY

client: ('172.16.2.151', 47880)
RECU: HELLO_TUNNEL
```

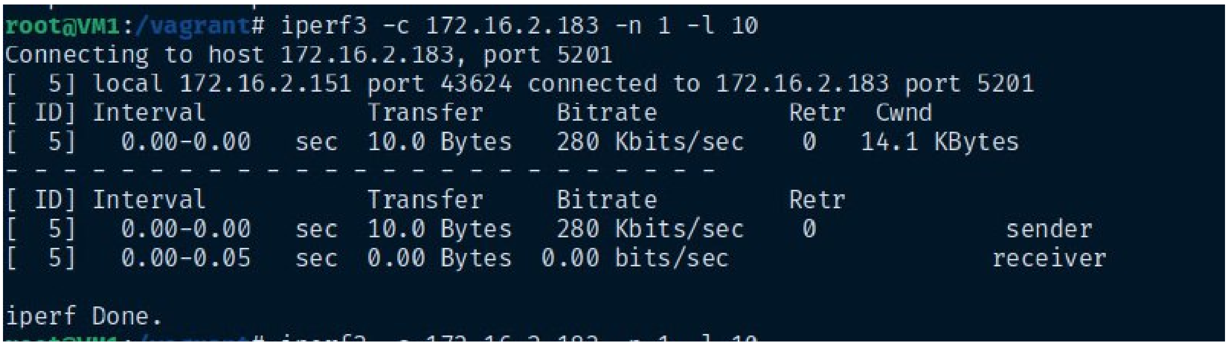
Figure 16: VM1: creation of `msg.txt` and sending to VM3 (TCP client).

[8] Point "4.2".

[9] Point "4.3".

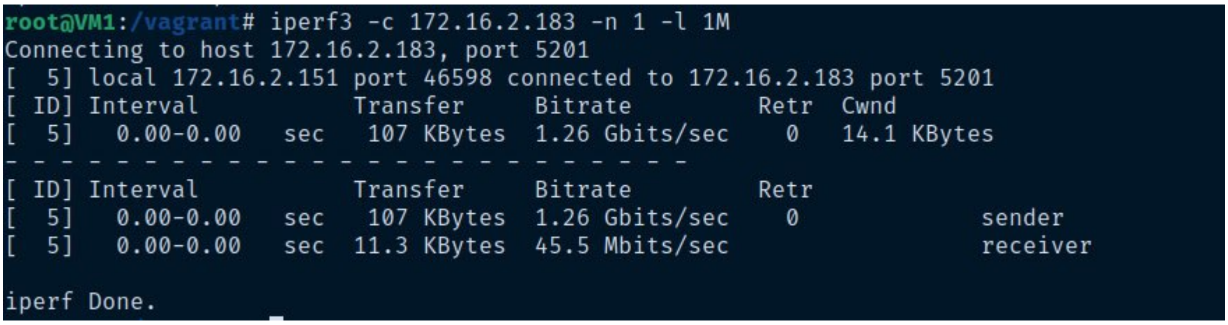
5.4 Layer 4: Bandwidth (4.4)

Iperf3 server on VM3 then client on VM1-6 with multiple buffer sizes: 10, 2K, 128K, 1M.[10]



```
root@VM1:/vagrant# iperf3 -c 172.16.2.183 -n 1 -l 10
Connecting to host 172.16.2.183, port 5201
[  5] local 172.16.2.151 port 43624 connected to 172.16.2.183 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-0.00   sec  10.0 Bytes   280 Kbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-0.00   sec  10.0 Bytes   280 Kbits/sec    0             sender
[  5]   0.00-0.05   sec  0.00 Bytes  0.00 bits/sec                  receiver

iperf Done.
```

Figure 17: Iperf3 results buffers (10): measured throughput.



```
root@VM1:/vagrant# iperf3 -c 172.16.2.183 -n 1 -l 1M
Connecting to host 172.16.2.183, port 5201
[  5] local 172.16.2.151 port 46598 connected to 172.16.2.183 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-0.00   sec   107 KBytes  1.26 Gbits/sec    0   14.1 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-0.00   sec   107 KBytes  1.26 Gbits/sec    0             sender
[  5]   0.00-0.00   sec  11.3 KBytes  45.5 Mbits/sec                  receiver

iperf Done.
```

Figure 18: Iperf3 results buffers (1M): measured throughput.

6 Conclusion

The IPv4 tunnel encapsulated within TCP/IPv6 restores connectivity between LAN3 and LAN4 despite the VM2 failure. The required tests (configuration, ping, application TCP, iperf3) are satisfied and the captures confirm the effective circulation of packets via `tun0` and TCP/IPv6 transport.

[10] Point "4.4".