Projet Réseaux

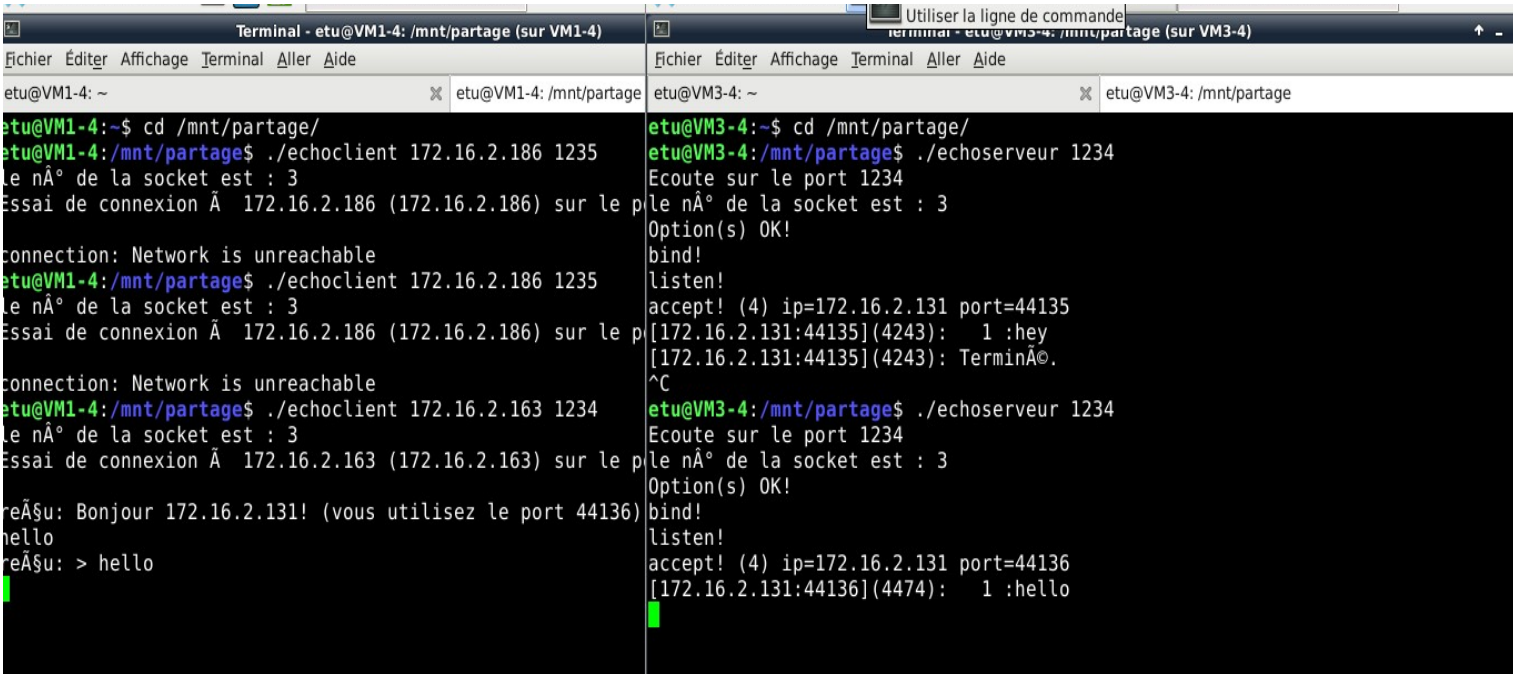
1. Configuration Réseau

1.1. Topologie et Adressage

Mettre en place les 6 Vms

1. Installer un serveur echo sur VM3 et sur VM3-6.

2. Depuis VM1, se connecter avec un client echo sur le serveur de VM3 et VM3-6



3. Peut-on faire que le serveur n'écoute qu'en IPv6 sur VM3-6 ?

> Notre client en VM1 ne pouvait pas se connecter au serveur en VM3-6.

2. L'interface virtuelle TUN

2.2. Configuration de l'interface

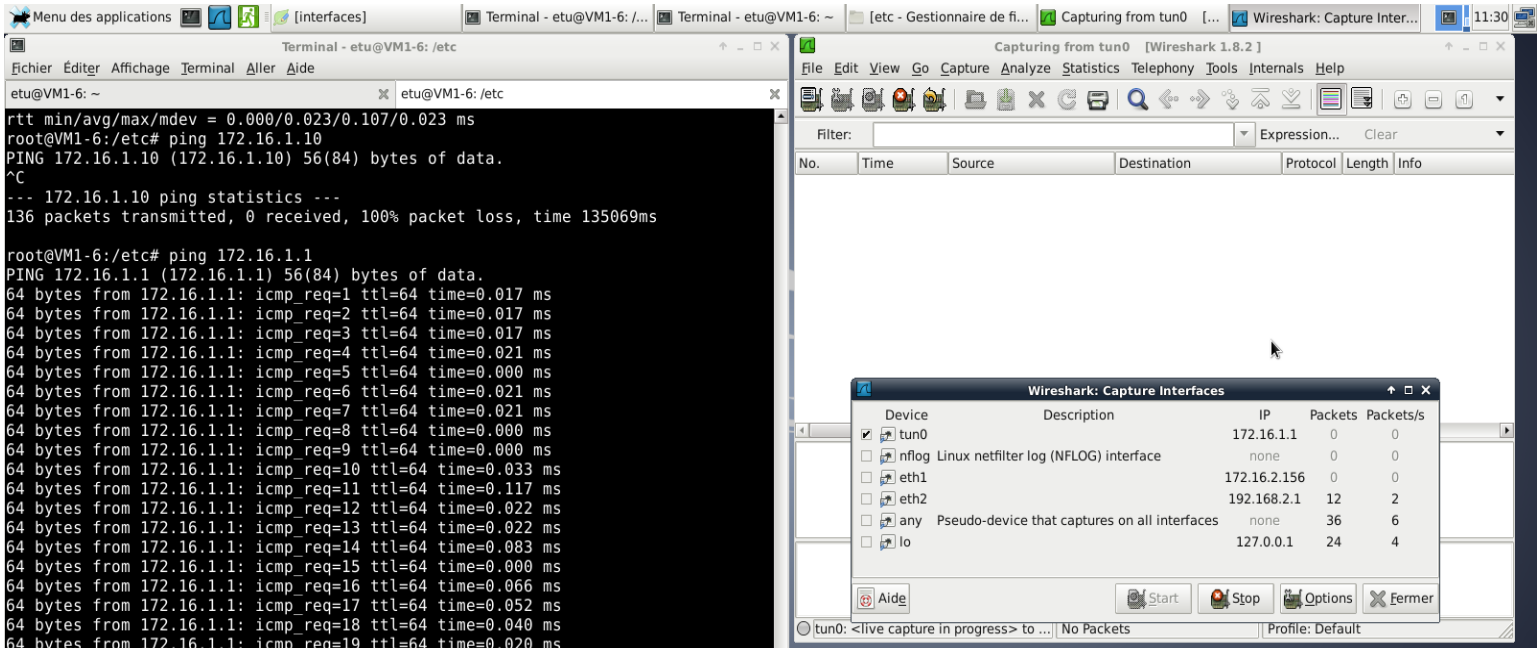
1. Configurer l'interface tun0 avec l'adresse 172.16.1.1, mettre un masque adéquat. Ecrire un script configure-tun.sh reprenant vos commandes de configuration.

2. **Routage :** Suite à la **disparition tragique de VM2**, faut-il modifier les informations de routage sur VM1 ? ou sur VM1-6 ?

Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

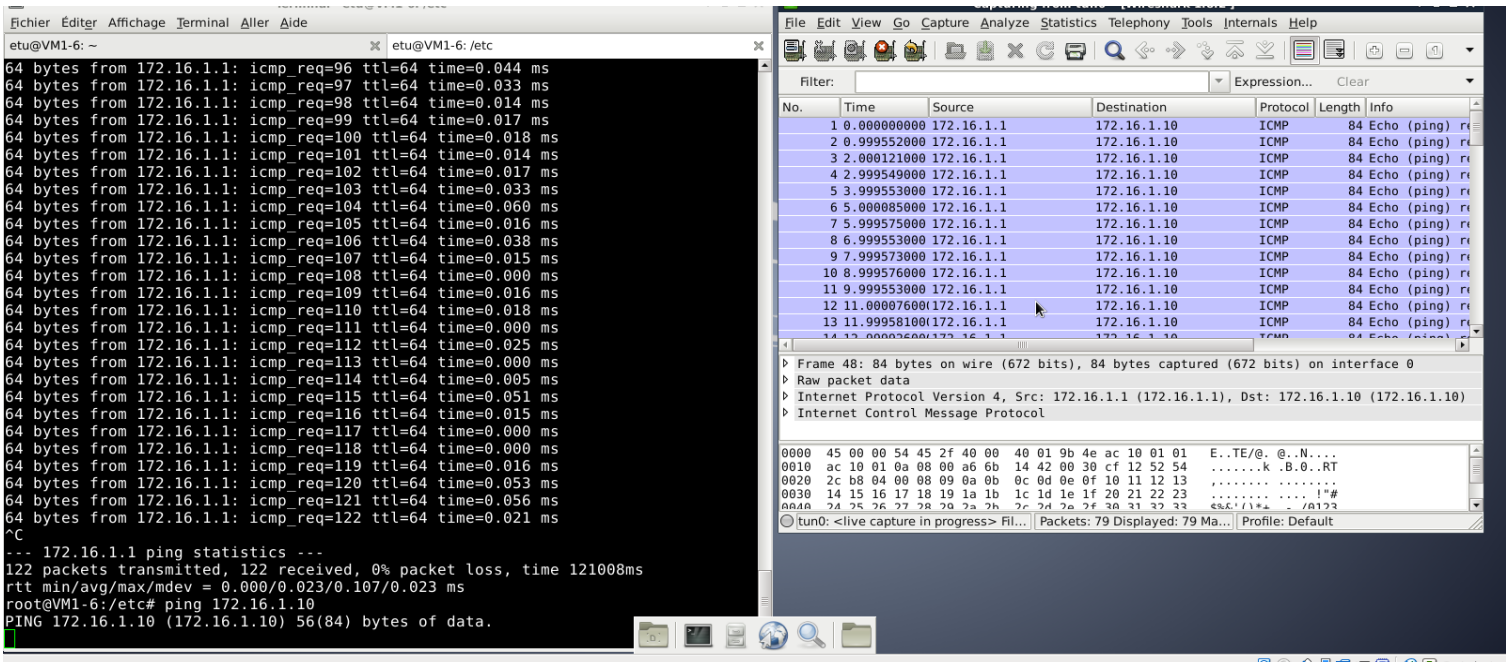
Il n'y a pas de changements à faire si on supprime directement le dossier de la VM, mais on peut toujours enlever les passerelles et LAN qui vont vers VM2

3. Faire un ping 172.16.1.1. Donner la capture sur tun0 (avec wireshark). Que constatez-vous ?



On constate que des paquets sont reçus mais Wireshark lui ne voit rien.

4. Faire ping 172.16.1.10. Que constatez-vous ?



Dans ce cas, la console n'affiche rien mais Wireshark voit des paquets émis.

Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

Expliquez.

Dans le premier cas, l'émetteur est aussi le destinataire, donc rien ne traverse puisque les paquets arrivent avant. Dans le second cas, on ping une IP non affectée, les paquets sortent dans sa direction et Wireshark le voit mais ils ne reviennent jamais car rien n'est là pour répondre.

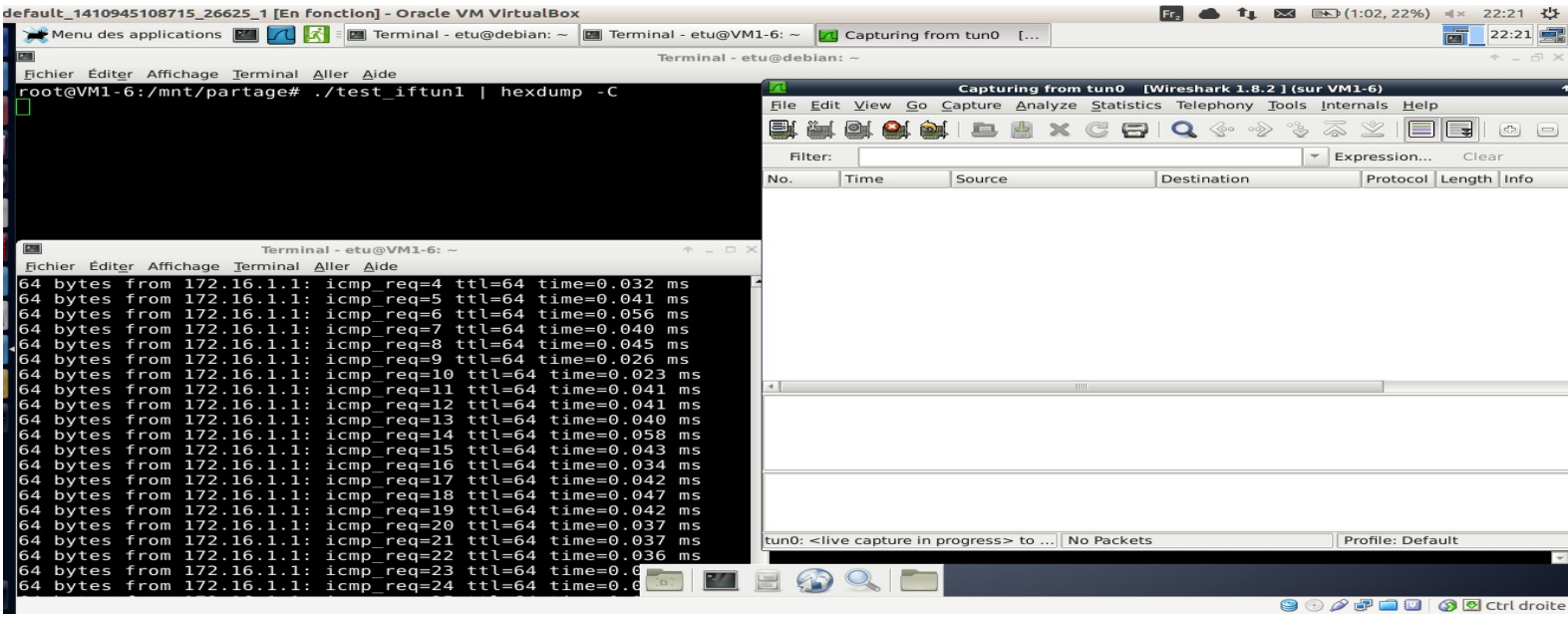
2.3. Récupération des paquets

1. Compléter la bibliothèque iftun avec une fonction avec deux descripteurs de fichiers en paramètres src et dst, qui, étant donné un descripteur srccorrespondant à une interface TUN, recopie perpétuellement toutes les données lisibles sur src dans le fichier décrit par dst.

2. Tester avec dst=1 (sortie standard). Comme ce qui est recopié est du binaire, on filtrera la sortira du programme de test avec hexdump.

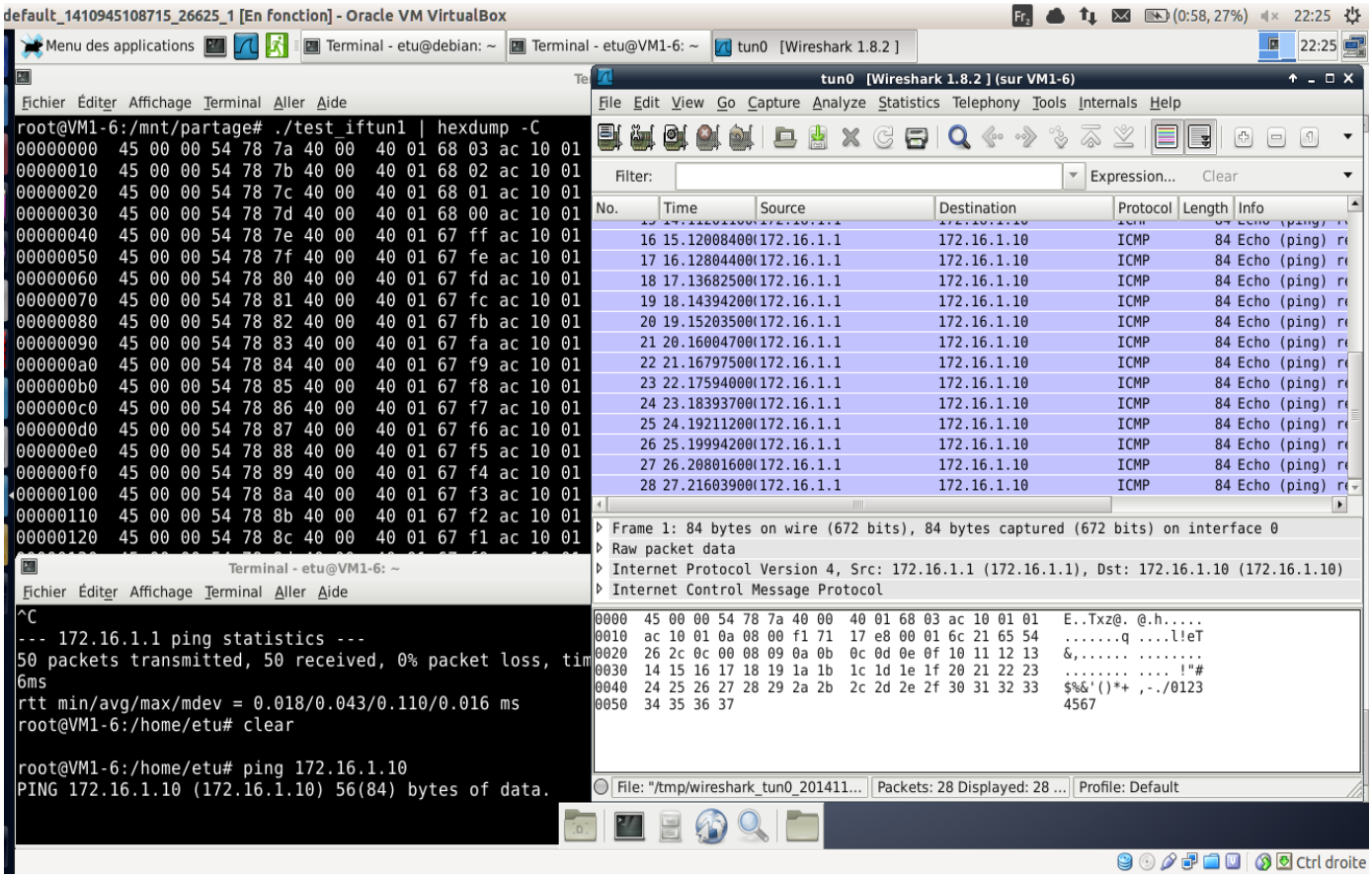
```
00000000  01 00 08 00 45 00 00 54  78 7a 40 00 40 01 68 03  |....E..Txz@.@.h.|
00000010  01 00 08 00 45 00 00 54  78 7b 40 00 40 01 68 02  |....E..Tx{@.@.h.|
00000020  01 00 08 00 45 00 00 54  78 7c 40 00 40 01 68 01  |....E..Tx|@.@.h.|
00000030  01 00 08 00 45 00 00 54  78 7d 40 00 40 01 68 00  |....E..Tx}@.@.h.|
00000040  01 00 08 00 45 00 00 54  78 7e 40 00 40 01 67 ff  |....E..Tx~@.@.g.|
```

3. Refaire ping 172.16.1.1 puis ping 172.16.1.10. Comparer et expliquer. Quel type de trafic voyez-vous? Refaire une capture avec wireshark dans le second cas et comparer avec ce qui est obtenu par votre programme test_iftun.



Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

Un ping 172.16.1.1 ne change rien aux résultats précédents.



Le principe est le même que lors des tests précédents pour un ping 172.16 .1.10 (on ping vers l'inconnu), mais on remarque que les 5 premiers paquets envoyés et affichés dans la console sont identiques à ceux que Wireshark a pu lire de 172.16.1.1 vers 172.16.1.10.

4. A quoi sert l'option IFF_NO_PI ? Que ce passe-t-il si vous ajoutez cette option lors de la création de l'interface ?

Elle permet de ne pas fournir les informations sur le paquets, elle enlève les 4 premiers octets.

Sans IFF_NO_PI :

```
00000000  01 00 08 00 45 00 00 54  78 7a 40 00 40 01 68 03  |....E..Txz@.@.h.|
00000010  01 00 08 00 45 00 00 54  78 7b 40 00 40 01 68 02  |....E..Tx{@.@.h.|
00000020  01 00 08 00 45 00 00 54  78 7c 40 00 40 01 68 01  |....E..Tx|@.@.h.|
00000030  01 00 08 00 45 00 00 54  78 7d 40 00 40 01 68 00  |....E..Tx}@.@.h.|
```

Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

Avec IFF_NO_PI :

```
00000000  45 00 00 54 78 7a 40 00  40 01 68 03 ac 10 01 01  |E..Txz@.@.h.....|
00000010  45 00 00 54 78 7b 40 00  40 01 68 02 ac 10 01 01  |E..Tx{@.@.h.....|
00000020  45 00 00 54 78 7c 40 00  40 01 68 01 ac 10 01 01  |E..Tx|@.@.h.....|
00000030  45 00 00 54 78 7d 40 00  40 01 68 00 ac 10 01 01  |E..Tx}@.@.h.....|
```

3. Un tunnel simple pour IPv4

Le but de cette partie est de créer un tunnel simple encapsulant un trafic IPv4 dans des paquets TCP/IPv6.

3.1. Redirection du trafic entrant

On utilisera par défaut le port 123.

Dans cette partie, on créera une bibliothèque extremite qui gérera le trafic entre extrémités du tunnel.

1. Ecrire une fonction ext-out qui crée un serveur écoutant sur le port 123, et redirige les données reçues sur la sortie standard.

2. Ecrire une fonction ext-in qui ouvre une connexion TCP avec l'autre extrémité du tunnel, puis lit le trafic provenant de tun0 et le retransmet dans la socket.

3. Quelle est l'adresse IPv6, noté IPOUT, de VM3-6 ? Comment pourrait-on l'obtenir simplement ? Automatiquement ?

Nous avons configuré l'IPOUT manuellement, fc00:1234:2::36. On peut lui en donner une automatique avec radvd comme on l'a vu en tp.

4. La commande nc (netcat) permet de transférer des données sur un port réseau. Pour le support IPv6 sur vos VMs, il faudra installer éventuellement netcat6. L'option -u permet d'envoyer en UDP (puisque votre tunnel est unidirectionnel pour l'instant.

1. Mettre en place une extrémité ext-in et une extrémité ext-out.

2. Tester avec un pair ping 172.16.1.10 pour injecter du trafic comme dans la partie précédente.

Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

Ecoute sur le port 123

le nÂ° de la socket est : 3

Option(s) OK!

bind!

listen!

accept! (4) ip=fc00:1234:1::16 port=34032

```
00000000  45 00 00 54 c4 e3 40 00  3f 01 19 57 ac 10 02 97  |E..T..@.?..W....|
00000010  ac 10 02 b7 08 00 af 84  11 d3 00 02 43 6c 6a 54  |............CljT|
00000020  9a e2 03 00 08 09 0a 0b  0c 0d 0e 0f 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|
00000050  34 35 36 37 45 00 00 54  c4 e5 40 00 3f 01 19 55  |4567E..T..@.?..U|
00000060  ac 10 02 97 ac 10 02 b7  08 00 be 83 11 d3 00 04  |................|
```

3. Tester avec un pair nc6 pour injecter du trafic application.

Avec le mot « salut» depuis VM1-6

```
00000260  45 00 00 22 74 cb 40 00  40 11 6b d4 ac 10 01 01  |E.."t.@.@.k.....|
00000270  ac 10 01 0a 93 22 00 7b  00 0e be 27 73 61 6c 75  |.....".{...'salu|
```

4. Tester avec un pair nc6 sur des VMs différentes.

Avec le mot « hey » depuis VM1-4

```
00000240  45 00 00 20 c4 a6 40 00  3f 11 19 b8 ac 10 02 97  |E.. ..@.?.......|
00000250  ac 10 02 b7 d8 c5 00 7b  00 0c e7 b6 68 65 79 0a  |.......{....hey.|
```

3.2. Redirection du trafic sortant

1. Compléter la fonction ext-out de la bibliothèque extrémité pour créer une extrémité qui lit le trafic provenant de la socket TCP et le retransmet dans le tun0 local.

Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

2. Que se passe-t-il lorsque vous écrivez dans le descripteur correspondant à tun0 ?

Le paquet passe ensuite dans VM3-4 si c'est l'IP qu'on a spécifiée.

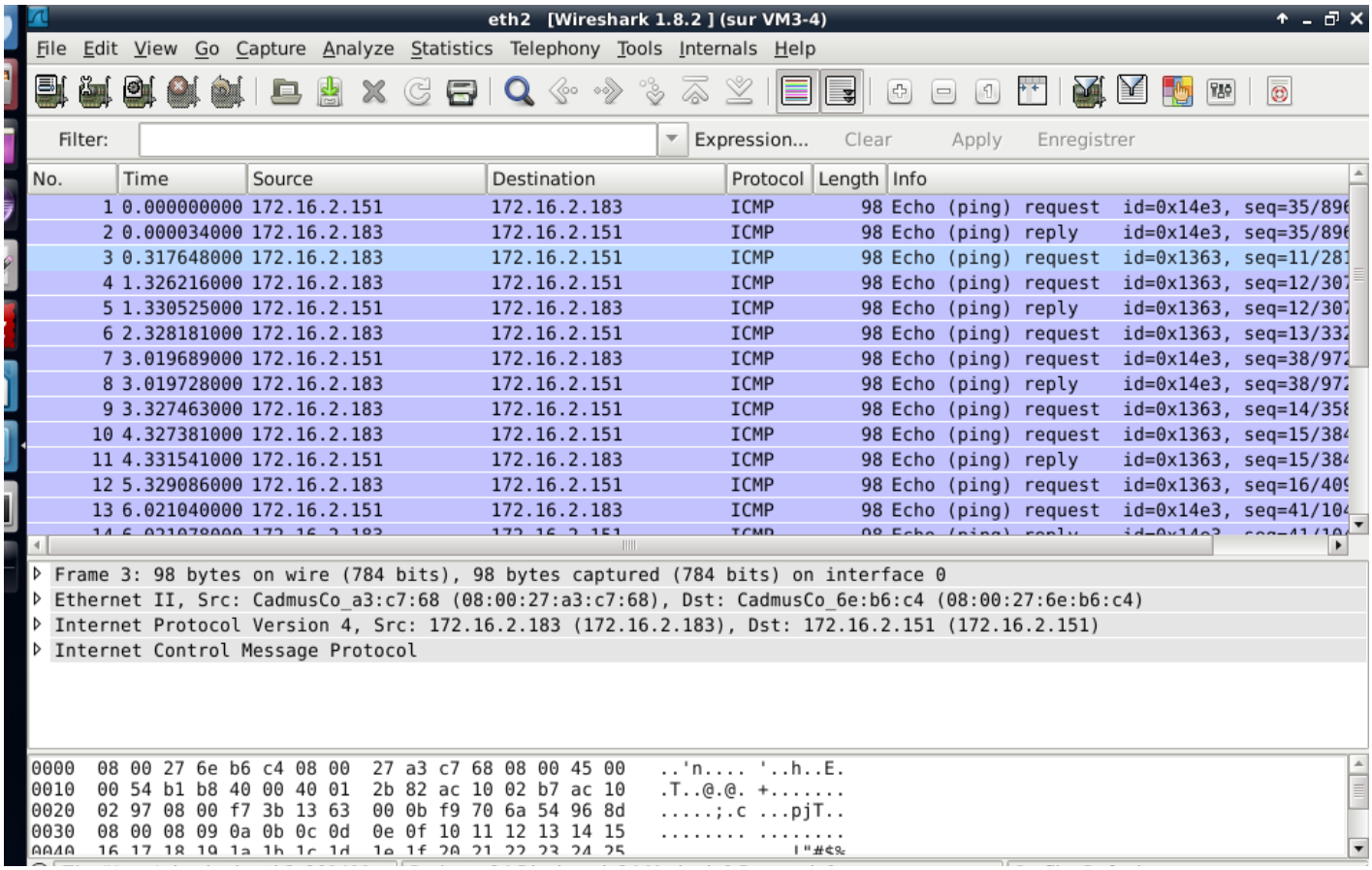
3. Vérifier avec l'un de vos paquets capturés dans le cas précédent. Pensez à le modifier éventuellement (avec un éditeur hexadécimal) pour que cela corresponde à tous **vos** paramètres réseau.

4. Proposer des tests de connectivité. Tester et vérifier !

3.3. Test du Tunnel

1. Compléter la bibliothèque pour pouvoir avoir un flux bidirectionnel

2. Assurez-vous que les communications sont bien asynchrones.



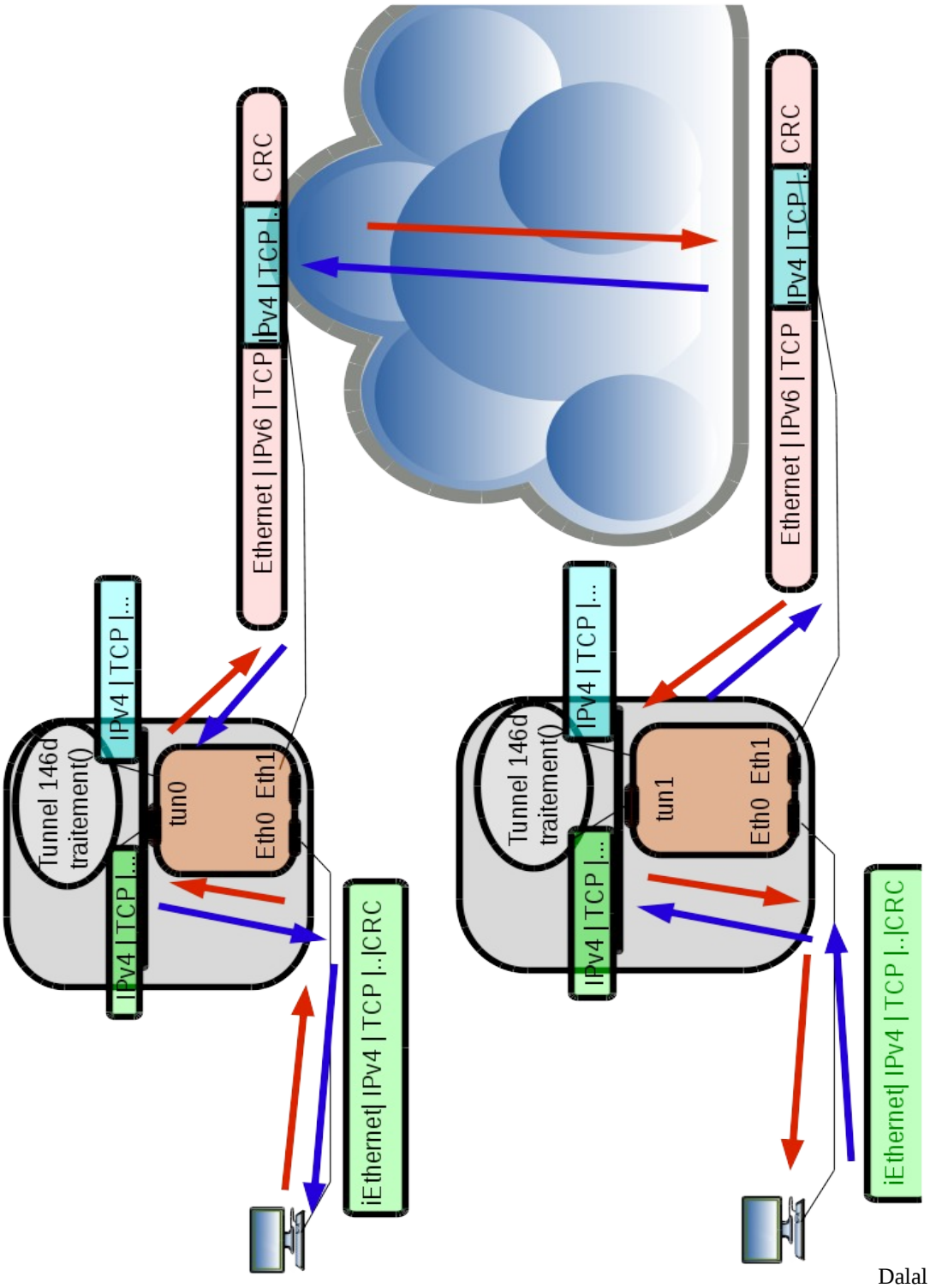
On remarque qu'il n'y a pas vraiment d'ordre, c'est donc bien asynchrone

Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

3.4. Mise en place du tunnel entre VM1-6 et VM3-6 : Schémas

On veut utiliser VM1-6 et VM3-6 pour pouvoir faire un tunnel entre LAN3 et LAN4.

- Compléter le **schéma simplifié** en expliquant **en détail** le parcours complet d'un paquet.



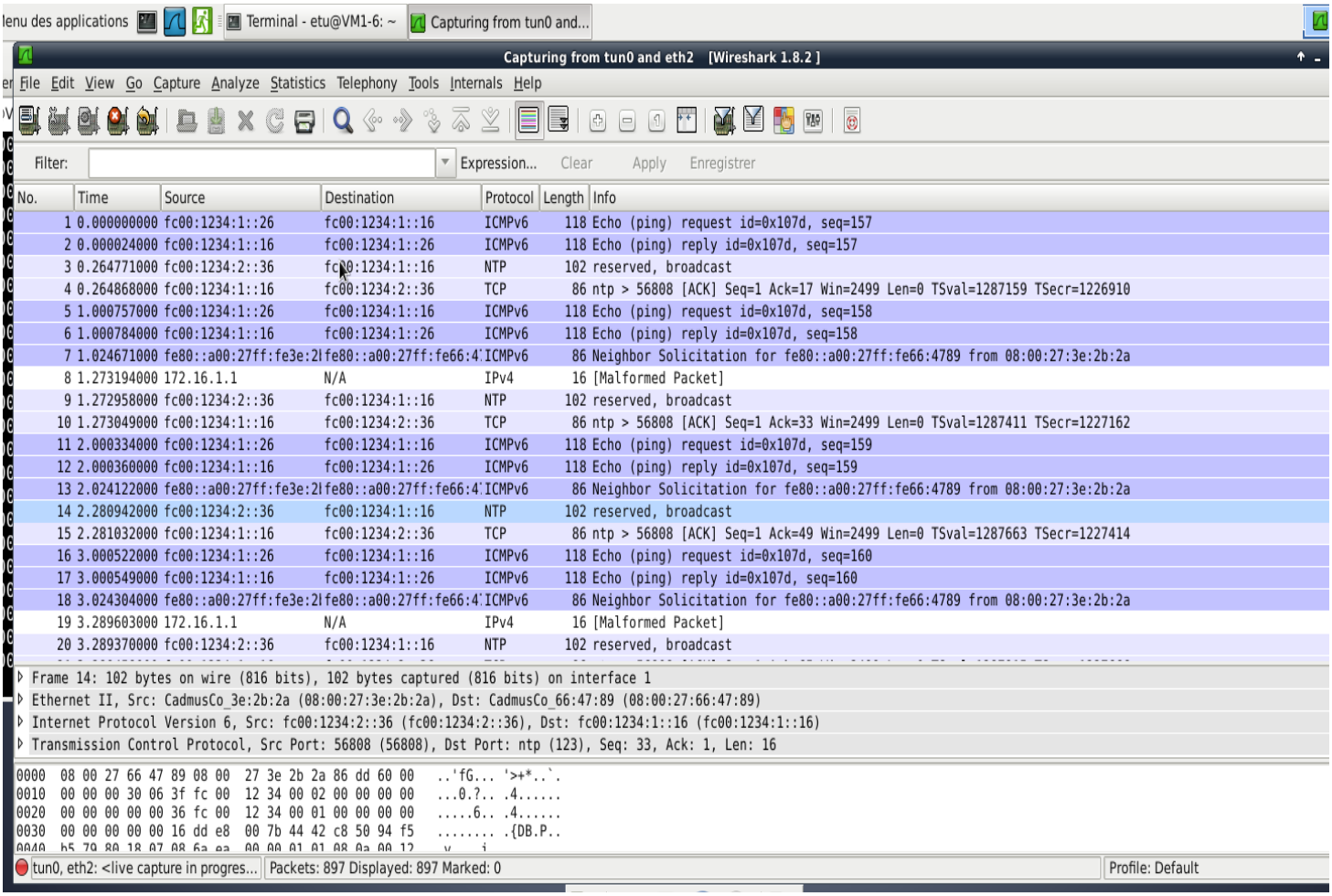
Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC

3.5. Mise en place du tunnel entre VM1-6 et VM3-6 : Système

1. Créer un exécutable tunnel46d qui, en s'appuyant sur la bibliothèque extremite crée un service offrant un tunnel TCP bidirectionnel.

Il pourra appeler le script de configuration réseau avec ces paramètres une fois que l'interface est créée. Il est également possible de faire des appels systèmes directs avec ioctl.

1. Tester en configurant et lançant tunnel46d sur deux VMs différentes.
2. Comparer le trafic direct et celui passant par le tunnel.



Les protocoles NTP et TCP sont les ping direct et l'IPv4 est issu du tunnel.

4. Améliorations

Nous n'avons pas eu le temps, malheureusement !

Dalal ZOUAI
Claire GERARD
Sébastien GOUVERNEUR
Ismaël CUSSAC