Projet Réseau

Auteurs : PELLETIER Sébastien
BOUDERMINE Antoine

2.2.2)

Il faut modifier le routage de VM1 pour la cible LAN4 afin d'envoyer les trames sur LAN3 via VM1-6.
Pour VM1-6, on redirige les trames en direction de LAN4 vers tun0.

2.2.3-4-5)

Le ping sur l'ip de tun0 fonctionne bien, cependant on ne peut pas capturer le flux de ce ping avec wireshark car il passe directement par le noyau et n'est donc pas envoyé sur l'interface.

Quand on fait un ping sur 172.16.2.10 le noyau envoi les données sur tun0 on a donc les requêtes de ping qui apparaissent sur wireshark (sans réponse).

2.3.3)

Une fois la redirection fait, notre programme "agit" comme wireshark en écoute sur tun0, on obtiens les même données.

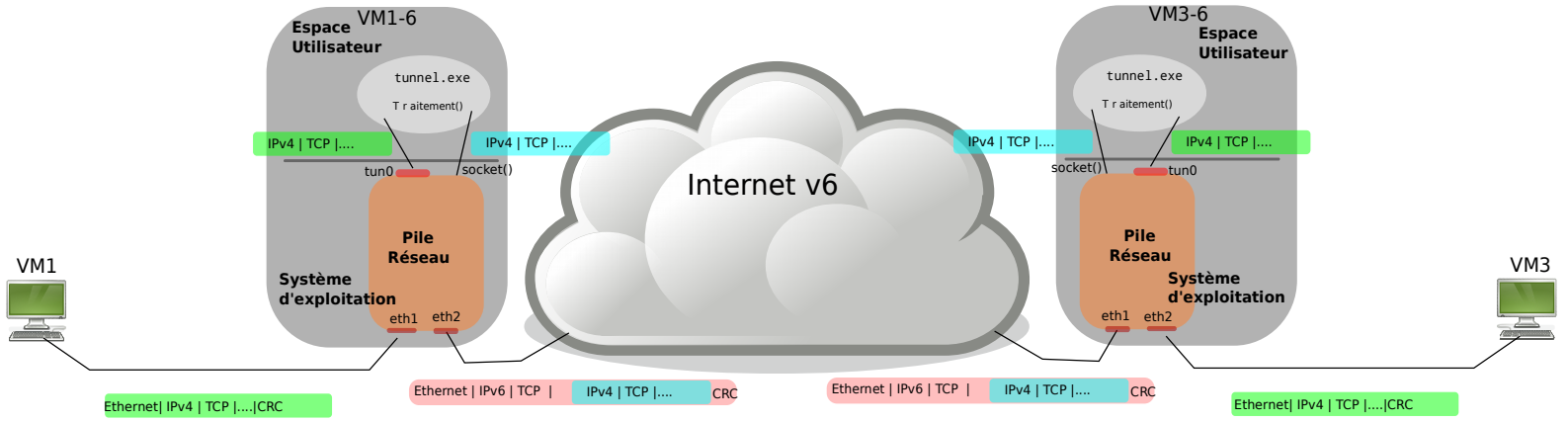
2.3.4)

IFF_NO_PI : Permet d'obtenir seulement les paquets IP tels quelles sans aucune information ajouté par le noyau.

3.2.2)

Théoriquement, les trames envoyé dans tun0 devraient aboutir à rien car elle n’ont aucune entête Ethernet (couche 2) cependant du fait du fonctionnement de tun0, le noyau ajoute la partie ethernet.

Le tunnel



Voici le schéma complété. Ici, les deux machines effectuant la jonction sont VM1-6 et VM3-6.

Pour que VM1 puisse (par exemple) faire un ping sur VM3, il faut que les paquets en direction du réseau de VM3 soient routés vers VM1-6 et plus vers VM2, ainsi le tunnel sur VM1-6 vas pouvoir encapsuler le paquet IPv4 dans une trame TCP.

Pour ce faire, il faut créer une route sur VM1-6 qui redirige toutes les trames en direction de VM3 vers tun0, en faisant cela, notre programme va pouvoir récupérer toutes ces trames pour les envoyer à l'autre extrémité du tunnel.

Inversement, quand une extrémité reçoit un paquet TCP, il envoi directement sont contenu (une trame IPv4) vers tun0, ainsi grâce au routage, le noyau va pouvoir rediriger le paquet vers la bonne interface en ajoutant les entêtes Ethernet adéquats.

Parcour d'un paquet VM1 → VM2 via tunnel.

Le paquet IPv4 et envoyé à VM1-6, VM1-6 redirige la trame sur tun0. Ici, notre tunnel lit le paquet sur tun0 et envoi ces données par TCP à l'autre bou du tunnel (VM3-6). VM3-6 effectu la même chose mais dans l'autre sens : il envoi les données reçu vers tun0 puis, le noyau va pouvoir rediriger le paquet vers la bonne interface (grace à la table de routage) et ajouter les entêtes Ethernet.

Utilisation

Pour initialiser les fichiers de configuration de VM (executez le deux fois desuite pour être sur): ./init-virtual-network.bash

Apres avoir fait la modification du ssh sur chaque VM. Il faut executer les ansible config.vm*.yml (et non config.vm*-tun.yml) sur chaque VM en commencent par VM1-6 et VM3-6.
(la configuration initial sans la coupure au niveau de VM2)

Configuration particulier pour VM1 et VM3 : executer les ansible config.vm1-tunnel.yml pour VM1 et config.vm3-tun.yml pour VM3.

Compilation des tunnels : /mnt/partage/src/ : make

Execution des tunnels :

- VM1-6 : /mnt/partage/src/ext1/tunnel.exe config.conf /vagrant/config.vm1-6-tunnel.yml
- VM3-6 : /mnt/partage/src/ext2/tunnel.exe config.conf /vagrant/config.vm3-6-tun.yml

Ici, le 2ème paramètre est le fichier ansible que le serveur va executer pour configurer tun0 ansi que les nouvelles routes.

Format du fichier de configuration :

```
##### Serveur Distant #####
ip_serveur_distant   = fc00:1234:2::36
port_serveur_distant = 123

##### Serveur local #####
port_serveur_local   = 123

#### Interface ####
interface_name = tun0
```