

Les VXLANs

Objectifs

Configurer, observer, analyser et comprendre la configuration de base des VXLANs.

1 Utilisation de l'outil de simulation du réseau

Ce réseau utilisé dans cette séance est "simulé" à l'aide d'un outil d'isolation fourni par le noyau Linux (les *Namespaces* réseau). Une description plus complète est disponible à l'adresse suivante

<http://chaput.perso.enseeiht.fr/teaching/ressources/tp-reseaux-virtualises>

Si vous avez déjà utilisé ces outils, vous pouvez passer à la section suivante.

1.1 Installation des fichiers

La page web citée plus haut décrit les différentes façons d'obtenir les fichiers vous permettant de démarrer la séance.

Grâce à l'aide de cette page et/ou de votre enseignant-e, vous pouvez donc maintenant démarrer un *shell* dans le répertoire contenant les fichiers de la séance voulue et dans lequel vous prendrez l'identité de l'administrateur (nécessaire pour la suite des opérations) :

```
$ cd le-dossier-de-mon-tp
$ sudo su
#
```

vous pouvez alors démarrer la séance.

1.2 Démarrage et arrêt du simulateur

Le lancement du simulateur se fait de la façon suivante

```
# ./creerReseau
```

Un terminal est alors ouvert sur les machines principales du réseau.

Une liste d'options utilisables pour wireshark vous est également fournie. Vous pourrez l'utiliser pour observer le trafic sur les interfaces des différentes machines.

Pour cela, vous lancerez, par exemple, dans le même terminal

```
# wireshark -i /tmp/nssi/host1/v0
```

Ce qui vous permettra d'observer le trafic sur l'interface v0 de la machine host1.

Notez que vous pouvez bien sûr lancer la commande `tcpdump` directement dans le terminal de la machine correspondante!

Vous pourrez arrêter la simulation ainsi :

```
# ./destruireReseau
```

Attention, lorsque vous arrêtez le réseau, toutes les manipulations faites sur les machines sont définitivement perdues!

1.3 Lancement d'une commande ou d'un terminal dans une machine

Si vous avez malencontreusement fermé le terminal d'une machine, vous pouvez le relancer de la façon suivante

```
# ./creerReseau -r machine
```

où *machine* est le nom de la machine.

Vous pouvez également lancer une commande sur une machine :

```
# ./creerReseau -r machine "commande et options"
```

Les guillemets sont nécessaires, par exemple

```
# ./creerReseau -r m1 "ip link show"
```

2 Le réseau de notre expérimentation

Le but de cette séance est de configurer et d'observer le fonctionnement de VXLANs sur un réseau IP.

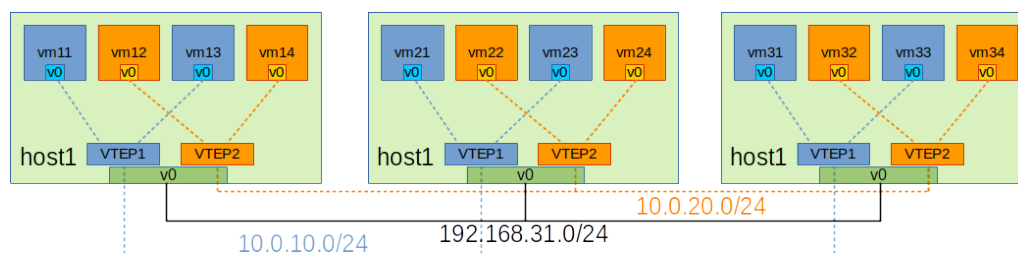


FIGURE 1 – Le réseau de cet exercice

Ici, trois machines hôtes sont créées (sur un réseau 192.168.31.0/24) hébergeant chacune quatre machines virtuelles vmh1 à vmh4.

Les douze machines virtuelles sont réparties dans 2 sous-réseaux : le 10.0.10.0/24 et le 10.0.20.0/24 comme illustré. L'adressage de toutes les machines est fait (l'octet de poids faible se déduit du nom de la machine).

Sur l'hôte *h*, l'interface *vmhv-v0* est reliée à l'interface *v0* de la machine virtuelle *v*.

3 Configuration des VXLANs

Nous allons maintenant procéder à la mise en place effective des VXLANs. Elle se décompose en trois étapes simples

- la création des VTEPs sur les machines *host1*, *host2* et *host3*;
- l'insertion des machines virtuelles dans les VXLANs;
- l'interconnexion des VTEPs entre eux au travers du réseaux entre les *host*.

Un choix important doit être fait concernant l'interconnexion des VTEPs. En effet la question se pose de l'acheminement des trames BUM (*Broadcast, Unknown, Multicast*).

Cette interconnexion peut se fonder sur une communication *unicast* ou sur une communication *multicast*. Dans le cas d'une communication *unicast*, l'apprentissage des adresses IEEE peut être réalisée par *inondation*, ou par une configuration statique des adresses IEEE.

Ce choix a une influence sur la façon dont sont créés les VTEPs, comme nous allons l'observer.

3.1 Création d'un VTEP

Intéressons-nous dans un premier temps à la création des VTEPs.

3.1.1 Pour une communication en *unicast*

Pour une telle communication, un VTEP est créé de la façon suivante

```
# ip link add vxlan42 type vxlan id 42 dstport 4789 local 192.168.10.2
```

Ici le VXLAN utilise le port UDP 4789, son VNI est 42 et son nom vxlan42 et il utilisera l'adresse IP 192.168.10.2 pour dialoguer avec ses homologues.

Attention : si le choix d'apprentissage des adresses qui est fait est fondé sur une configuration statique, on ajoutera l'option `nolearning` à la commande IP ci dessus.

On n'oubliera pas d'activer cette interface :

```
# ip link set up dev vxlan42
```

3.1.2 Pour une communication en *multicast*

Dans ce cas, les VTEPs doivent être créés de la façon suivante

```
# ip link add vxlan42 type vxlan id 42 dstport 4789 group 239.0.0.42 \
dev eth0 ttl 8
```

Ici le VXLAN est lié à l'adresse multicast IPv4 239.0.0.42, il utilise le port UDP 4789, son VNI est 42 et son nom vxlan42.

L'interface, ici aussi, devra être activée.

▷ Exercice 1 : Configuration de vos VTEPs

Sur chacune de vos machines hôte, créez des VTEPs pour différents VXLANs. On utilisera, dans la mesure du possible, les deux techniques évoquées ci dessus. ■

3.2 Insertion des machines virtuelles

Pour insérer plusieurs machines virtuelles dans le même VLAN (et en particulier dans le même VXLAN, nous allons créer un pont et y insérer les interfaces liées à celles des VM :

```
# brctl addbr brvx42
# brctl addif brvx42 vxlan42
# brctl addif brvx42 vm1-v0
# brctl addif brvx42 vm2-v0
```

On active ensuite le pont en ayant pris soin de désactiver le *spanning tree*

```
# brctl stp brvx42 off
# ip link set up dev brvx42
```

▷ Exercice 2 : Insertion de vos machines dans les VXLANs

Répartissez vos différentes machines virtuelles sur les VXLANs que vous avez créés.

Configurez les adresses IP de ces machines et vérifiez que la communication est possible au sein d'un VXLAN.

Quelle encapsulation observez-vous? ■

3.3 Interconnexion des VTEPs

Lorsque plusieurs VTEPs appartenant au même VXLAN ont été créés, il est possible de les faire dialoguer (et d'observer les échanges). Pour cela, chaque VTEP doit connaître ses homologues, ainsi que les adresses IEEE des machines qui y sont reliées.

3.3.1 Interconnexion en *unicast* avec découverte par inondation

Dans une telle configuration, chaque VTEP doit être informé de l'existence des autres VTEP participant au même VXLAN.

On utilisera pour cela la commande suivante

```
# bridge fdb append 00:00:00:00:00:00 dev vxlan42 dst 192.168.10.2
```

Ici, on informe le VTEP local du vxlan42 de l'existence d'un homologue à l'adresse 192.168.10.2.

▷ Exercice 3 : Interconnexion des VTEPs

Interconnectez ainsi certains des VTEPs de vos VXLANs configurés en unicast.

Vérifiez que les machines virtuelles distantes communiquent correctement.

Quelle encapsulation observez-vous ?

Quel est le comportement des VTEPs vis à vis des trames BUM ? ■

3.3.2 Interconnexion en *unicast* avec configuration statique des adresses IEEE

Dans cette configuration, chaque VTEP doit connaître les autres VTEPs du même VXLAN, ainsi que les adresses IEEE des machines virtuelles desservies.

On utilisera pour cela la commande suivante

```
# bridge fdb append de:ad:be:ef:c0:ff:ee dev vxlan42 dst 192.168.10.2
```

Ici, on informe le VTEP vxlan42 de l'existence d'un homologue à l'adresse 192.168.10.2 et de la présence de l'adresse IEEE sur ce VTEP. Il faudra bien sûr une commande équivalente pour chaque adresse IEEE.

Les trames utilisant l'adresse IEEE 00:00:00:00:00:00 devront cependant toujours être transmises vers chacun des VTEPs du VXLAN, si bien que la configuration décrite ci-dessus est toujours nécessaire.

▷ Exercice 4 : Interconnexion des VTEPs

Interconnectez ainsi certains des VTEPs de vos VXLANs configurés en unicast.

Vérifiez que les machines virtuelles distantes communiquent correctement.

Quelle encapsulation observez-vous ?

Quel est le comportement des VTEPs vis à vis des trames BUM ? ■

Il est possible d'éliminer toute duplication inutile en utilisant chaque VTEP comme un proxy ARP pour les machines virtuelles qu'il dessert. On utilisera pour cela la configuration suivante

```
# ip neigh add 192.168.42.10 lladdr c0:ff:ee:c0::ff:ee dev vxlan42
```

Ici, on stipule à la machine locale qu'elle doit se comporter comme un *proxy* ARP pour les adresses passées en paramètres sur le VTEP vxlan42.

3.3.3 Interconnexion en *multicast*

La solution la plus élégante pour interconnecter des VTEPs appartenant à un même VXLAN est d'utiliser du MULTICAST.

Pour qu'une telle configuration soit opérationnelle, il faut, naturellement, que le routage *multicast* soit effectif entre les machines hébergeant les VTEP. La mise en place de ce routage est largement en dehors du sujet de cette séance, nous nous contenterons donc d'utiliser des machines directement reliées entre elles.

▷ **Exercice 5 : Interconnexion des VTEPs**

Interconnectez ainsi les VTEPs de vos VXLANs configurés en multicast.

Vérifiez que les machines virtuelles distantes communiquent correctement.

Quelle encapsulation observez-vous ?

Quel est le comportement des VTEPs vis à vis des trames BUM ? ■