Introduction au routage inter-VLAN

Philippe Latu
philippe.latu(at)inetdoc.net

http://www.inetdoc.net

Résumé

Le routage inter-VLAN sur les systèmes GNU/Linux présente de nombreux intérêts tant du point de vue conception que du point de vue exploitation. Avec un système GNU/Linux on peut combiner les fonctions de cloisonnement des domaines de diffusion avec d'autres services tels que le filtrage réseau netfilter/iptables. De plus, avec une infrastructure hétérogène associant plusieurs générations et|ou marques de commutateurs, GNU/Linux permet d'homogénéiser l'exploitation.

Table des matières

1. Copyright et Licence

1.1. Méta-information

Ce document est écrit avec DocBook[1] XML sur un système Debian GNU/Linux[2]. Il est disponible en version imprimable au format PDF : `interco.inter-vlan.qa.pdf`[3].

Les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. C'est la distribution Debian GNU/Linux qui est utilisée pour les tests présentés. Voici une liste des paquets contenant les commandes :

[1] http://www.docbook.org
[2] http://www.debian.org
[3] http://www.inetdoc.net/pdf/interco.inter-vlan.qa.pdf

- ifupdown - High level tools to configure network interfaces
- iproute2 - networking and traffic control tools
- iptables - administration tools for packet filtering and NAT

1.2. Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou prompt spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite `$` ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.
- Toute commande précédée de l'invite `#` nécessite les privilèges du super-utilisateur.

2. Réseaux locaux virtuels et routage

Les définitions importantes sur les réseaux locaux virtuels et le routage associé sont présentées dans l'article Routage Inter-VLAN[4]

On rappelle simplement que la notion de réseau local virtuel ou VLAN permet de constituer des groupes logiques dans les réseaux Ethernet au niveau liaison de la modélisation OSI. Sans l'ajout d'une balise définie dans le standard IEEE 802.1Q, le format des adresses MAC ne permet aucun découpage en sous-ensembles (à l'exception du trafic multicast qui ne nous concerne pas ici). Une fois que l'on peut repérer l'appartenance à un groupe logique sur la base des étiquettes ajoutées aux trames il est possible de distribuer un domaine de diffusion entre plusieurs équipements physiques distincts.

On atteint ainsi un objectif très important. Il est possible de concevoir une topologie logique de réseau totalement indépendante de la topologie physique.

Réseau virtuel ou pas, il ne faut pas oublier les éléments suivants sur la segmentation des réseaux locaux.

- Une interface de commutateur délimite un domaine de collision.
- Une interface de routeur délimite à la fois un domaine de collision et un domaine de diffusion.

3. Etude d'une configuration type

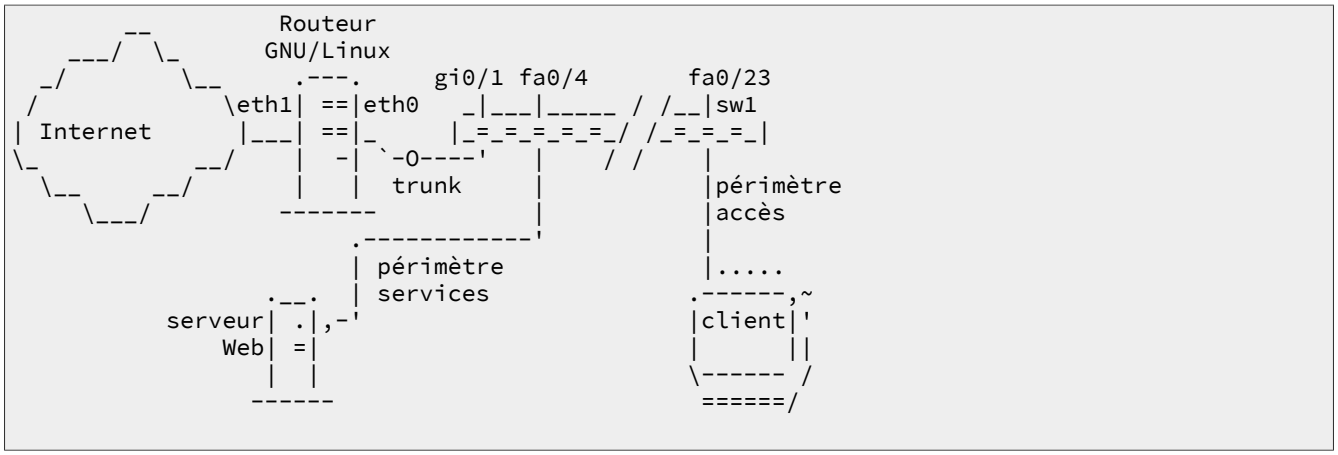
La configuration type étudiée ici est une maquette réduite qui comprend un routeur et un commutateur physique. Pour les besoin des l'illustration, on dissocie l'équipement responsable de la commutation de paquets de l'équipement en charge de la commutation de trames.

Le routeur unique correspond bien à la réalité des réseaux modernes. Du réseau d'agence d'une centaine d'hôtes au réseau de campus de plusieurs milliers d'hôtes, seule la capacité de traitement de l'équipement varie.

Le commutateur unique correspond beaucoup moins à la réalité. Même dans un réseau d'agence, on dépasse très vite le cap des 48 ports connectés. On utilise alors un équipement avec une bonne capacité de commutation qui assure la distribution vers des commutateurs dédiés aux accès des hôtes. Tous ces commutateurs sont reliés entre eux à l'aide de trunks qui véhiculent les flux marqués des réseaux virtuels.

Dans l'illustration présentée ici, les deux couches distribution et accès sont «synthétisées» sur un seul équipement. Un trunk sur un lien gigabit relie le routeur au commutateur. En véhiculant les flux marqués entre le routeur et le commutateur il assure la liaison entre routage et commutation de trames. Les hôtes directement connectés au commutateur n'ont aucune connaissance des balises IEEE802.1Q. Ils ne nécessitent donc aucune configuration particulière.

[4] http://www.inetdoc.net/articles/inter-vlan-routing/



Cette infrastructure type comprend 2 périmètres reliés au réseau public Internet. Un premier périmètre de services utilisé pour l'hébergement des services accessibles depuis le réseau public : DNS, Web, courrier électronique, etc. Un second périmètre pour les postes de travail qui ne doivent pas être accessibles depuis le réseau public.

On ajoute aux deux périmètres classiques, un réseau particulier dédié à la gestion de l'infrastructure : configuration des équipements, métrologie, journalisation, etc.

Tableau 1. Plan d'adressage des périmètres

Nom	n° VLAN	Adresse IP
Management	2	`192.168.2.0/24`
Services	100	`192.168.100.0/24`
Accès	200	`192.168.200.0/24`

Le tableau ci-dessus établit la correspondance entre les périmètres, les réseaux virtuels et les réseaux IP à interconnecter.

3.1. Configuration du *trunk*

Communications réseau dans le périmètre *Management*

Du point de vue configuration, ce réseau est très particulier. Il véhicule les trames sans balises IEEE802.1Q entre le routeur et le commutateur. On associe à ce périmètre le VLAN natif du trunk.

Côté routeur GNU/Linux, on configure l'interface de façon classique puisqu'il s'agit de traiter des trames Ethernet standard.

```
# ip addr add 192.168.2.2/24 brd + dev eth0
```

Côté commutateur, on utilise la notion de VLAN «natif» pour configurer l'interface en mode trunk.

```
!
interface GigabitEthernet0/1
 switchport trunk native vlan 2
 switchport mode trunk
 no cdp enable

<snipped/>
!
interface Vlan2
 ip address 192.168.2.1 255.255.255.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no ip route-cache
```

La configuration du trunk est la suivante :

```
#sh int gi0/1 trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/1       on           802.1q         trunking      2

Port      Vlans allowed on trunk
Gi0/1       1-4094

Port        Vlans allowed and active in management domain
Gi0/1       1-2,100,200

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/1       1-2,100,200
```

Les règles d'utilisation des trames sans balises IEEE802.1Q sont les suivantes :

- Toute trame appartenant au VLAN natif est émise sans balise IEEE802.1Q sur un port en mode trunk par le commutateur.
- Toute trame reçue sans balise IEEE802.1Q sur un port en mode trunk du commutateur appartient au VLAN natif.

On complétera la configuration du commutateur de façon à ce que toutes les opérations de gestion de l'équipement passent par ce VLAN natif.

À ce niveau, les tests de communication réseau sont très simples.

- Côté routeur :

```
RouterA:~$ ping -c 2 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=255 time=19.4 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=255 time=1.22 ms

--- 192.168.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.226/10.355/19.484/9.129 ms
```

- Côté commutateur :

```
Switch#ping 192.168.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

3.2. Configuration IEEE 802.1Q sur le Routeur GNU/Linux

Communications réseau dans les périmètres *Services* et *Accès*

Cette fois-ci, il est indispensable de traiter les flux marqués avec les balises IEEE802.1Q. Aujourd'hui, tous les noyaux fournis avec les distributions Linux comme Debian GNU/Linux disposent d'un module appelé `8021q`.

```
$ find /lib/modules/`uname -r` -name 8021q
/lib/modules/4.2.0-1-amd64/kernel/net/8021q
```

Le chargement de ce module se fait automatiquement dès qu'une opération relative aux étiquettes IEEE802.1Q est effectuée. Il suffit alors de consulter la liste des modules pour vérifier sa présence. Il est toujours possible de charger manuellement ce module. Voici un exemple.

```
# modprobe -v 8021q
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/llc/llc.ko
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/802/stp.ko
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/802/mrp.ko
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/802/garp.ko
insmod /lib/modules/4.2.0-1-686-pae/kernel/net/8021q/8021q.ko

# grep 8021q /var/log/kern.log
kernel: [  439.345617] 8021q: 802.1Q VLAN Support v1.8
kernel: [  439.345723] 8021q: adding VLAN 0 to HW filter on device eth0
```

Une fois la partie kernelspace traitée, on passe logiquement à la partie userspace. La commande **ip** du paquet iproute2 dispose de toutes les options utiles pour créer les sous-interfaces associées aux étiquettes IEEE802.1Q.

Dans notre exemple, la syntaxe pour les deux sous-interfaces des deux périmètres définis est la suivante :

```
# ip link add link eth0 name eth0.100 type vlan id 100
# ip link add link eth0 name eth0.200 type vlan id 200
```

On visualise aussi le résultat avec la commande **ip** :

```
$ ip addr ls
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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 pfifo_fast state UP group default qlen 1000
    link/ether ba:ad:00:ca:fe:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.2/24 brd 192.168.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::b8ad:ff:feca:fe00/64 scope link
       valid_lft forever preferred_lft forever
3: eth0.100@eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
    link/ether ba:ad:00:ca:fe:00 brd ff:ff:ff:ff:ff:ff
4: eth0.200@eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
    link/ether ba:ad:00:ca:fe:00 brd ff:ff:ff:ff:ff:ff
```

Les deux nouvelles sous-interfaces se configurent manuellement de façon classique.

```
# ip addr add 192.168.100.1/24 brd + dev eth0.100
# ip addr add 192.168.200.1/24 brd + dev eth0.200
```

Sur un système de la famille Debian GNU/Linux, il est possible de rendre cette configuration permanente en éditant le fichier `/etc/network/interfaces` comme suit :

```
<snipped/>
auto eth0
iface eth0 inet static
        address 192.168.2.2/24

auto eth0.100
iface eth0.100 inet static
        address 192.168.100.1/24

auto eth0.200
iface eth0.200 inet static
        address 192.168.200.1/24
```

Une fois la configuration des interfaces en place, on obtient la table de routage suivante :

```
# ip route ls
default via aaa.bbb.ccc.1 dev eth1❶
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.2❷
192.168.100.0/24 dev eth0.100  proto kernel  scope link  src 192.168.100.1❸
192.168.200.0/24 dev eth0.200  proto kernel  scope link  src 192.168.200.1❹
aaa.bbb.ccc.0/24 dev eth1  proto kernel  scope link  src aaa.bbb.ccc.7❺
```

❶ L'interface `eth1` a la possibilité d'acheminer le trafic issu des deux périmètres vers l'Internet via la passerelle par défaut.

❷ L'interface physique `eth0` sert de trunk entre le routeur et le commutateur. Sa configuration réseau correspond au périmètre Management. Le réseau auquel appartient l'interface utilise des trames sans balises IEEE802.1Q. Dans le jargon, ce VLAN est qualifié de natif.

❸ La sous-interface `eth0.100` est associée au VLAN numéro 100. Sa configuration réseau correspond au périmètre Services. Les trames de ce réseau qui circulent sur le trunk sont complétées avec une balise IEEE802.1Q qui comprend l'identificateur de VLAN 100.

❹ La sous-interface `eth0.200` est associée au VLAN numéro 200. Sa configuration réseau correspond au périmètre Accès. Les trames de ce réseau qui circulent sur le trunk sont complétées avec une balise IEEE802.1Q qui comprend l'identificateur de VLAN 200.

❺ L'interface `eth1` est directement connectée au réseau «public». Elle n'a aucune connaissance du trafic issu des différents périmètres sans configuration spécifique.

Côté commutateur, il faut que la base de données des VLANs connus contienne les mêmes identificateurs que ceux affectés sur le Routeur GNU/Linux.

Le fichier de configuration du commutateur doit contenir les informations suivantes si le protocole VTP[5] a préalablement été configuré en mode transparent :

```
vlan 2
 name management
!
vlan 100
 name services
!
vlan 200
 name access
```

Ensuite, on affecte les ports du commutateurs aux différents VLANs ou périmètres.

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int range fastEthernet 0/1 - 12
Switch(config-if-range)#switchport access vlan 100
Switch(config-if-range)#exit
Switch(config)#int range fastEthernet 0/13 - 48
Switch(config-if-range)#switchport access vlan 200
Switch(config-if)#^Z
Switch#
07:10:45: %SYS-5-CONFIG_I: Configured from console by console
```

On visualise le résultat des affectations de ports en mode accès de la façon suivante.

```
Switch#sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active
2    management                       active
100  services                         active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
200  access                           active    Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Fa0/25, Fa0/26, Fa0/27, Fa0/28
                                                Fa0/29, Fa0/30, Fa0/31, Fa0/32
                                                Fa0/33, Fa0/34, Fa0/35, Fa0/36
                                                Fa0/37, Fa0/38, Fa0/39, Fa0/40
                                                Fa0/41, Fa0/42, Fa0/43, Fa0/44
                                                Fa0/45, Fa0/46, Fa0/47, Fa0/48
```

3.3. Activation de la fonction routage

Avec la configuration actuelle, le Routeur GNU/Linux ne remplit pas sa fonction. Par exemple, les hôtes du périmètre Accès ne peuvent pas communiquer avec les serveurs du périmètre Services. Il est nécessaire d'activer la fonction routage au niveau du noyau Linux pour que les paquets IP puissent être transmis (ou routés) entre des réseaux différents.

La présentation des fonctions réseau d'une interface pilotée par le noyau Linux sort du cadre de ce document. Il faut consulter le support Configuration d'une interface de réseau local pour obtenir les informations nécessaires.

Voici une copie du fichier `/etc/sysctl.conf` comprenant l'ensemble des réglages appliqués au noyau Linux du Routeur de la configuration type. Pour appliquer ces paramètres, il suffit d'utiliser la commande **`sysctl -p`** et de valider la valeur de la «clé» `ip_forward`. Si cette valeur est à `1`, le routage est actif au niveau du noyau Linux.

[5] https://en.wikipedia.org/wiki/VLAN_Trunking_Protocol

```
# /etc/sysctl.conf - Configuration file for setting system variables
# See sysctl.conf (5) for information.
#
# Refuser la prise en charge des requêtes ARP pour d'autres hôtes
net.ipv4.conf.all.proxy_arp=0

# Ignorer les mauvais messages d'erreurs ICMP
net.ipv4.icmp_ignore_bogus_error_responses=1

# Ignorer les messages de diffusion ICMP
net.ipv4.icmp_echo_ignore_broadcasts=1

# Journaliser les adresses sources falsifiées ou non routables
net.ipv4.conf.all.log_martians=1

# Refuser les adresses sources falsifiées ou non routables
net.ipv4.conf.all.rp_filter=1

# Refuser les messages ICMP redirect
net.ipv4.conf.all.accept_redirects=0

net.ipv4.conf.all.send_redirects=0

# Refuser le routage source
net.ipv4.conf.all.accept_source_route=0

# Activer le routage
net.ipv4.ip_forward=1
```

Les tests de communication entre les réseaux des différents périmètres peuvent être effectués depuis le commutateur.

```
Switch#ping 192.168.2.2❶

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Switch#ping 192.168.100.1❷

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1000 ms
Switch#ping 192.168.200.1❸

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.200.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Switch#ping aaa.bbb.ccc.7❹

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to aaa.bbb.ccc.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1004 ms
Switch#ping aaa.bbb.ccc.1❺

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to aaa.bbb.ccc.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

❶ Test de communication ICMP sur le périmètre Management. Ce test n'utilise pas la fonction routage puisqu'il est effectué entre les deux extrémités du trunk.

❷ Test de communication ICMP sur le périmètre Services. Ce test utilise la fonction routage entre le réseau `192.168.2.0/24` et le réseau `192.168.100.0/24`.

❸ Test de communication ICMP sur le périmètre Accès. Ce test utilise la fonction routage entre le réseau `192.168.2.0/24` et le réseau `192.168.200.0/24`.

❹ Test de communication ICMP vers le réseau public. Ce test utilise la fonction routage entre le réseau `192.168.2.0/24` et le réseau `aaa.bbb.ccc.0/24`.

❺ Test de communication ICMP vers l'Internet. Ce test échoue puisque le Routeur GNU/Linux n'échange pas sa table de routage avec les autres routeurs de l'Internet.

Ces tests montrent qu'il faut compléter la configuration pour que les échanges réseau entre les périmètres et l'Internet soient possibles. Comme ces échanges réseau entre l'Internet et les périmètres ne peuvent pas se faire dans n'importe quelles conditions, il est nécessaire d'introduire la fonction de filtrage pour obtenir une interconnexion satisfaisante.

4. Interconnexion et filtrage réseau

L'étude du filtrage réseau avec le noyau Linux sort du cadre de ce document. Il faut consulter les versions françaises du Guide Pratique du Filtrage de Paquets sous Linux 2.4 et du Guide Pratique du NAT sous Linux 2.4 pour obtenir les informations nécessaires.

D'un point de vue général, on dispose de deux solutions distinctes pour interconnecter les périmètres réseau administrés avec l'Internet.

- Partager la table de routage des périmètres administrés avec les routeurs de l'Internet via un protocole de routage tel qu'OSPF. Consulter le guide Initiation au routage, 3ème partie pour obtenir des exemples complets d'exploitation du protocole de routage OSPF avec les services du logiciel GNU/Linux Quagga.

- Camoufler les périmètres administrés derrière une adresse IP publique accessible depuis l'Internet. Cette opération est réalisée avec les fonctions de filtrage réseau du noyau Linux : netfilter pour la partie kernelspace et iptables pour la partie userspace.

C'est la seconde proposition qui offre le plus de facilités de contrôle immédiat sur les flux réseau. L'outil de camouflage (masquerading) généralement utilisé est appelé traduction d'adresses (Native Address Translation ou NAT).

4.1. Fonctionnement minimal

Après avoir activé le routage au niveau noyau (voir Section 3.3, « Activation de la fonction routage »), la fontion de camouflage est simple à mettre en oeuvre :

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Cette règle réalise une traduction d'adresse source. Tout paquet IP sortant par l'interface `eth1` voit son adresse IP source réécrite avec l'adresse IP de l'interface.

L'exécution de la règle entraîne le chargement des modules de gestion de la traduction d'adresses et du suivi dynamique de communication (stateful inspection).

```
# dmesg |grep ip_
ip_tables: (C) 2000-2002 Netfilter core team
ip_conntrack version 2.1 (4095 buckets, 32760 max) - 248 bytes per conntrack
```

```
# lsmod |grep ip
iptable_filter          3200  0
ipt_MASQUERADE          3712  1
iptable_nat            23516  2 ipt_MASQUERADE
ip_conntrack           44728  2 ipt_MASQUERADE,iptable_nat
ip_tables              22528  3 iptable_filter,ipt_MASQUERADE,iptable_nat
ipv6                  255936  12
```

Le suivi dynamique de communication consiste à conserver une empreinte de paquet sortant de façon à identifier les paquets retour relatifs à cette «demande». Les empreintes sont stockées dans la table `/proc/net/ip_conntrack` du système de fichiers virtuel du noyau Linux.

```
# cat /proc/net/ip_conntrack
tcp❶      6 431999 ESTABLISHED❷ src=192.168.200.2❸ dst=192.168.200.1❹ \
          sport=33450 dport=22❺ packets=417 bytes=31133 \
          src=192.168.200.1 dst=192.168.200.2 \
          sport=22 dport=33450 packets=306 bytes=111969 [ASSURED] mark=0 use=1
tcp       6 431999 ESTABLISHED src=192.168.200.2 dst=64.236.34.4 \
          sport=33449 dport=80❻ packets=7075 bytes=368009 \
          src=64.236.34.4 dst=aaa.bbb.ccc.7❼ \
          sport=80 dport=33449 packets=9219 bytes=12839148 [ASSURED] mark=0 use=1
```

❶ Protocole de transport utilisé.
❷ État de la connexion TCP.
❸ Adresse IP source. Cette adresse correspond à un poste client appartenant au périmètre Accès.
❹ Adresses IP destination. Dans le premier cas, la communication est interne au réseau du périmètre Accès. Dans le second cas, il s'agit d'une adresse sur l'Internet.
❺ Le numéro de port destination du paquet sortant identifie service Internet utilisé : SSH.
❻ Le numéro de port destination du paquet sortant identifie le service Internet utilisé : HTTP. Plus loin sur la même ligne, on retrouve les adresses IP source et destination attendues.
❼ L'adresse IP destination attendue pour un paquet retour est l'adresse publique du Routeur GNU/Linux. Cette ligne montre bien que le routeur à la connaissance des réseaux internes et du réseau public. C'est à partir de ces correspondances d'adresses IP que les décision d'acheminement sont prises. Dans le cas de la traduction d'adresses par camouflage, l'adresse IP retour est réécrite avec l'adresse IP de l'hôte du périmètre Accès.

Si cette configuration a le mérite d'illustrer le fonctionnement du routage inter-VLAN de façon simple, elle ne correspond pas à un niveau de contrôle d'accès suffisant. L'objet de la section suivante est justement de chercher à augmenter ce niveau de contrôle.

4.2. Meilleur contrôle d'accès

Dans un premier temps, il faut garantir que tous les paquets IP non autorisés sont bloqués ; ce qui revient à appliquer la règle «tout ce qui n'est pas autorisé est interdit».

La traduction de cette règle en termes de configuration revient à jeter tous les nouveaux paquets par défaut sur les «chaînes» d'entrée et de traversée des interfaces réseau

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

En toute rigueur, il faudrait faire de même avec la chaîne de sortie `OUTPUT`. Cette présentation ayant pour but premier d'illustrer les concepts, ajouter les traitements de la chaîne `OUTPUT` ne ferait qu'alourdir les scripts sans apporter d'élément nouveau.

Dans un deuxième temps, il faut affiner la configuration du suivi de communication dynamique. La règle d'or du filtrage avec la fonction stateful inspection, c'est la description la plus fine possible du premier paquet qu'on autorise à passer.

La traduction de cette règle en termes de configuration contient 2 parties :

- Un bloc de règles qui organise le suivi de communication pour chaque chaîne sur laquelle on appliqué la politique par défaut `DROP`.

```
-A <CHAINE> -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A <CHAINE> -p tcp -m conntrack --ctstate ESTABLISHED -m tcp ! --syn -j ACCEPT
-A <CHAINE> -p tcp -m conntrack --ctstate RELATED -m tcp --syn -j ACCEPT
-A <CHAINE> -p icmp -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

- Des règles spécifiques à chaque flux autorisé. C'est à la rédaction de ces règles qui correspondent au premier paquet autorisé qu'il faut apporter le plus grand soin. Un exemple pour les paquets IP émis depuis le périmètre Accès sur la chaîne `FORWARD` :

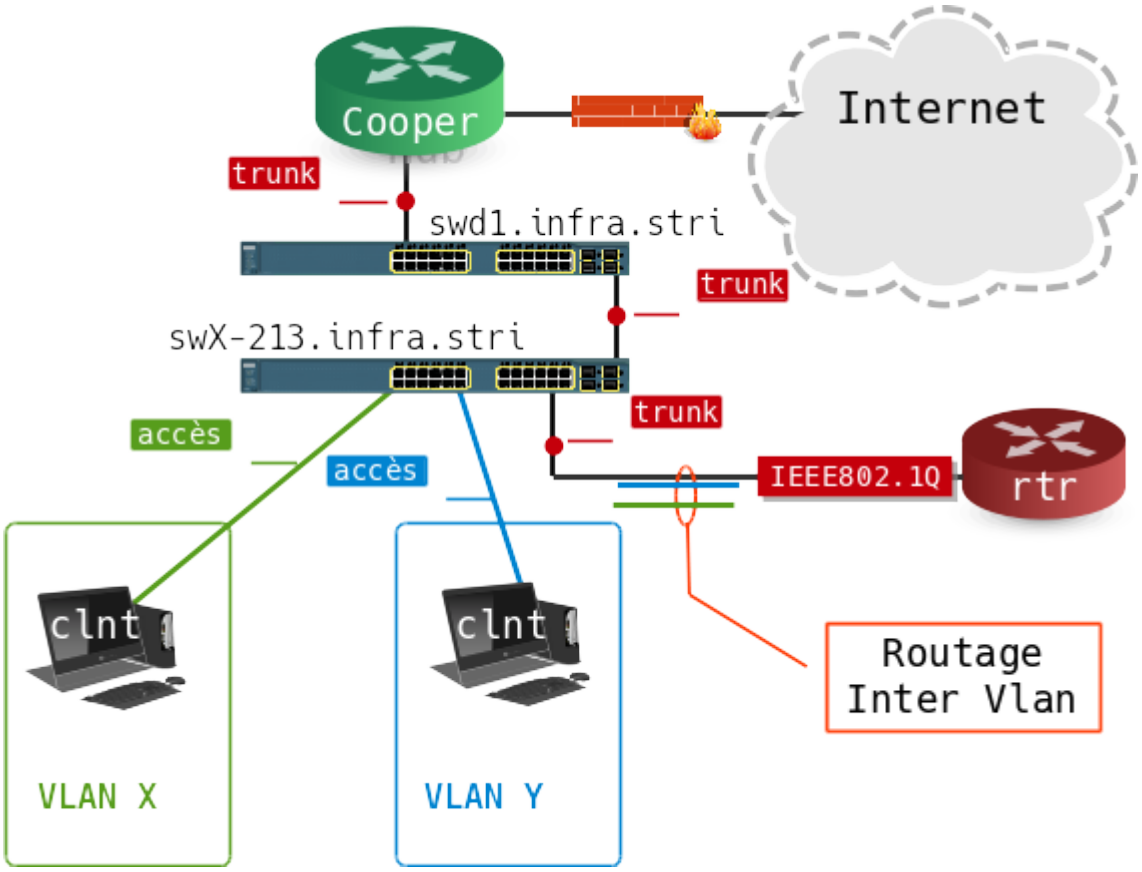
```
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p tcp -m tcp --syn --sport 1024: -m conntrack --ctstate N
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p udp -m udp --sport 1024: -m conntrack --ctstate NEW -j
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p icmp --icmp-type echo-request -m conntrack --ctstate NE
```

Voici une version intermédiaire de script de configuration du filtrage pour le périmètre Accès. En supposant que le fichier des règles est stocké dans le répertoire `/etc/iptables/`, on active les règles avec une commande du type **`iptables-restore </etc/iptables/rules.v4`**.

```
# Filtrage réseau du périmètre Accès
#
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Tables de traduction d'adresses
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth1 -p tcp --syn -m tcpmss --mss 1400:1536 -j TCPMSS --clamp-mss-to-pmtu
-A POSTROUTING -o eth1 -j MASQUERADE
COMMIT
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Tables de filtrage
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
#
# -> Chaîne INPUT
# . suivi de communication
-A INPUT -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m conntrack --ctstate ESTABLISHED -m tcp ! --syn -j ACCEPT
-A INPUT -p tcp -m conntrack --ctstate RELATED -m tcp --syn -j ACCEPT
-A INPUT -p icmp -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
# . toutes les communications internes sont autorisées
-A INPUT -i lo -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i eth0.200 -m conntrack --ctstate NEW -j ACCEPT
# . administration du Routeur GNU/Linux avec SSH
-A INPUT -i eth1 -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
# . services de gestion du commutateur vers le Routeur GNU/Linux
-A INPUT -i eth0 -s 192.168.2.1 -p udp -m multiport --dports 69,123,162,514 -m conntrack --ctstate NEW
# . poubelle propre
-A INPUT -m conntrack --ctstate INVALID -j DROP
-A INPUT -p tcp -j REJECT --reject-with tcp-reset
-A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable
#
# -> Chaîne FORWARD
# . suivi de communication
-A FORWARD -p udp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p tcp -m conntrack --ctstate ESTABLISHED -m tcp ! --syn -j ACCEPT
-A FORWARD -p tcp -m conntrack --ctstate RELATED -m tcp --syn -j ACCEPT
-A FORWARD -p icmp -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A FORWARD -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
-A FORWARD -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
# . communications des hôtes du périmètre Accès
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p tcp --syn --sport 1024: -m conntrack --ctstate NEW -j ACC
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p udp --sport 1024: -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -i eth0.200 -s 192.168.200.0/24 -p icmp --icmp-type echo-request -m conntrack --ctstate NEW
# . poubelle propre
-A FORWARD -m conntrack --ctstate INVALID -j DROP
-A FORWARD -p tcp -j REJECT --reject-with tcp-reset
-A FORWARD -p udp -j REJECT --reject-with icmp-port-unreachable
COMMIT
```

5. Travaux pratiques

5.1. Topologie type de travaux pratiques



Topologie type TP[6]

Pour les besoins de ces travaux pratiques, les configurations des 4 commutateurs : `sw5.infra.stri`, `sw6.infra.stri`, `sw7.infra.stri` et `sw8.infra.stri` sont effacées ainsi que leurs bases de données de VLANs.

Comme indiqué dans la topologie type ci-dessus, trois postes de travaux pratiques sont associés à un commutateur. Un poste joue le rôle de routeur inter-VLAN et les deux autres sont des postes clients appartenant chacun à un VLAN ou réseau IP différent.

Le seul point de configuration imposé est le raccordement au réseau d'interconnexion avec le routeur principal de la salle de travaux pratiques. Ce raccordement utilise le port `fa0/24` de chaque commutateur qui doit être configuré en mode trunk en utilisant le VLAN natif numéro 1. Le réseau IP correspondant au VLAN numéro 1 a l'adresse : `172.16.0.0/20`

Point important, la lecture de la section «Plan d'adressage» du document Architecture réseau des travaux pratiques[7] donne les adresses des deux routeurs ayant accès au réseau de Campus.

- Routeur `casper.infra.stri` : `172.16.0.2/20`
- Routeur `cooper.infra.stri` : `172.16.0.4/20`

5.2. Affectation des postes de travail

Les affectations données dans la table ci-dessous ne sont pas figées pour la durée des travaux pratiques. Une fois la configuration validée sur un groupe de trois postes, il est vivement conseillé de permuter les rôles de façon à mieux maîtriser les étapes de configuration.

[6] images/inter-vlan-routing-topology.png
[7] http://www.inetdoc.net/travaux_pratiques/infra.tp/

Tableau 2. Affectation des rôles, des numéros de VLANs et des adresses IP

Groupe	Commutateur	Poste	Rôle	VLAN	Réseau
1	sw5.infra.stri	alderaan	client	250	192.168.1.2/25
		bespin	client	251	192.168.1.130/25
		centares	routeur	1 (natif)	172.16.0.30/20
				250	192.168.1.1/25
				251	192.168.1.129/25
2	sw6.infra.stri	coruscant	client	260	192.168.2.2/25
		dagobah	client	261	192.168.2.130/25
		endor	routeur	1 (natif)	172.16.0.32/20
				260	192.168.2.1/25
				261	192.168.2.129/25
3	sw7.infra.stri	felucia	client	270	192.168.3.2/25
		geonosis	client	271	192.168.3.130/25
		hoth	routeur	1 (natif)	172.16.0.34/20
				270	192.168.3.1/25
				271	192.168.3.129/25
4	sw8.infra.stri	mustafar	client	280	192.168.4.2/25
		naboo	client	281	192.168.4.130/25
		tatooine	routeur	1 (natif)	172.16.0.36/20
				280	192.168.4.1/25
				281	192.168.4.129/25

Le positionnement des 4 commutateurs est référencé dans le support Architecture réseau des travaux pratiques.

5.3. Configuration des postes de travaux pratiques

Q1. Dans un groupe de trois postes tel qu'il a été défini ci-dessus, quel(s) poste(s) nécessite(nt) une configuration spécifique pour l'utilisation des réseaux locaux virtuels ?

Q2. Toujours dans un groupe de trois postes, comment doivent être programmés les ports de commutateur sur lesquels les postes clients sont raccordés ?

Q3. Encore dans un groupe de trois postes, comment doivent être programmés les ports de commutateur sur lesquels les routeurs sont raccordés ?

Q4. Dans la configuration d'un trunk, qu'est-ce qui distingue un VLAN natif ?

Q5. À partir du tableau des affectations ci-dessus, pourquoi les trois postes d'un groupe ne peuvent-ils pas appartenir au même réseau IP ?

Q6. Quel type de poste reçoit les trames complétées par des balises IEEE 802.1Q ?

Une fois le plan d'adressage IP défini, reprendre la Section 3, « Etude d'une configuration type » pour le groupe de postes de travaux pratiques.

Q7. Quel est le paquet qui contient les outils de configuration des interfaces réseau correspondant à chaque VLAN à router ?

Q8. Une fois les interfaces de chaque VLAN configurées sur le poste routeur, quelles sont les opérations à effectuer pour que le transfert des paquets IP d'un réseau local à l'autre soit effectif ?

Q9. Pourquoi doit-on utiliser la traduction d'adresses pour les flux réseau sortants du poste routeur vers l'Internet ? Que deviennent les paquets IP de ces flux sans traduction d'adresses ? Si la traduction d'adresses n'était pas disponible, quelle autre technique faudrait-il employer ?

Q10. Donner la séquence des tests ICMP à effectuer pour valider la connectivité entre :

- les postes clients et le poste routeur,
- les postes clients et l'ensemble des autres interfaces du routeur,
- les postes clients entre eux,
- les postes clients et l'Internet.

Q11. À l'aide de l'analyseur Wireshark, capturer des flux réseau mettant en évidence le marquage des trames avec les balises IEEE 802.1Q. Relever les numéros d'identification des VLANs vus par les interfaces du routeur. Quelle interface faut-il utiliser pour la capture de façon à visualiser l'ensemble du trafic ?

Q12. Pourquoi les flux réseau capturés contiennent-ils autant de trames STP (Spanning Tree Protocol) ?

Q13. Pourquoi la majorité des trames STP capturées sont-elles considérées comme ayant le type Ethernet II ? Quel aurait du être le type d'une trame STP si les balises IEEE 802.1Q n'étaient pas utilisées ?

6. Documents de référence

IEEE 802.1Q Standard
IEEE 802.1Q Standard[8]

How LAN Switches Work, Document ID: 10607
Documentation Cisco™ : How LAN Switches Work[9]

Standards d'encapsulation dans les trunks
Documentation Cisco™ : InterSwitch Link and IEEE 802.1Q Frame Format[10]

Configuring InterVLAN Routing and ISL/802.1Q Trunking, Document ID: 14976
Documentation Cisco™ décrivant une configuration simple sur le routage inter-VLAN : Configuring InterVLAN Routing and ISL/802.1Q Trunking[11].

La segmentation des réseaux locaux
Segmentation des réseaux locaux[12] : argumentation sur les fonctions de commutation et de routage.

[8] http://standards.ieee.org/getieee802/download/802.1Q-1998.pdf
[9] http://www.cisco.com/en/US/tech/tk389/tk689/technologies_tech_note09186a00800a7af3.shtml
[10] http://www.cisco.com/en/US/tech/tk389/tk689/technologies_tech_note09186a0080094665.shtml
[11] http://www.cisco.com/en/US/tech/tk389/tk815/technologies_configuration_example09186a00800949fd.shtml
[12] http://www.inetdoc.net/articles/lan-segmentation/

Configuration d'une interface de réseau local
: Configuration d'une interface de réseau local[13] : présentation complète sur la configuration des interfaces réseau avec un système GNU/Linux. La section sur les Fonctions réseau d'un interface traite des réglages possibles au niveau du noyau Linux. C'est à ce niveau que l'on retrouve l'activation du routage. Voir Section 3.3, « Activation de la fonction routage ».

Guide Pratique du Filtrage de Paquets sous Linux 2.4
: Guide Pratique du Filtrage de Paquets[14] : présentation des concepts du filtrage réseau avec le noyau Linux.

Guide Pratique du NAT sous Linux 2.4
: Guide Pratique du NAT[15] : présentation des concepts de la fonction de traduction d'adresses IP avec le noyau Linux.

Initiation au routage, 3ème partie
: Initiation au routage, 3ème partie[16] : guide complet sur l'utilisation du logiciel Quagga pour transformer un système GNU/Linux en routeur OSPF.

Architecture réseau des travaux pratiques
: Le support Architecture réseau des travaux pratiques[17] présente la topologie physique de la salle de travaux pratiques avec la Disposition des équipements dans l'armoire de brassage[18] ainsi que les configurations par défaut des équipements. On y trouve aussi le plan d'adressage IP utilisé avec les autres supports de travaux pratiques, le plan de numérotations des VLANs et les affectations des groupes de ports des commutateurs.

[13] http://www.inetdoc.net/travaux_pratiques/config.interface.lan/
[14] http://www.netfilter.org/documentation/HOWTO/fr/packet-filtering-HOWTO.html
[15] http://www.netfilter.org/documentation/HOWTO/fr/NAT-HOWTO.html
[16] http://www.inetdoc.net/guides/zebra.ospf/
[17] http://www.inetdoc.net/travaux_pratiques/infra.tp/
[18] http://www.inetdoc.net/travaux_pratiques/infra.tp/infra.tp.interco.html#infra.tp.interco.cabling