
M2101

Réseaux locaux et équipements actifs

Travaux pratiques

Sami Evangelista
IUT de Villetaneuse
Département Réseaux et Télécommunications
2018–2019

<http://www.lipn.univ-paris13.fr/~evangelista/cours/M2101>

Table des matières

Introduction aux travaux pratiques	2
TP 1 — Ethernet	3
TP 2 — Commutation	9
TP 3 — STP et VLAN	14
TP 4 — Routage inter-VLAN	19

Introduction aux travaux pratiques

Les TP sont à réaliser sur l'image MAGEIA4 qui sera restaurée sur chaque poste en début de séance. Les TP sont à réaliser en binôme. Chaque binôme réalisera un compte-rendu de TP électronique.

Deux comptes existent par défaut sur l'image MAGEIA4 :

- le compte `root` — mot de passe : `iutparis13`
- le compte `etudiant` — mot de passe : `etudiant`

Dans ce document, toutes les commandes qui sont précédées du caractère `#` doivent être lancées avec le compte `root`. Celles précédées du caractère `$` peuvent être lancées par n'importe quel utilisateur.

Envoi des compte-rendu

Seuls les compte-rendu au format PDF seront acceptés.

Envoyez vos compte-rendu à l'adresse `sami.evangelista@lipn.univ-paris13.fr`.

Copie d'écran

Dans tous les TP, il est demandé d'insérer des copies d'écran dans votre compte-rendu. Vous utiliserez pour cela l'outil `scrot` (SCreenshOT). N'étant pas disponible par défaut, il faut l'installer avec la commande ci-dessous :

```
# urpmi scrot
```

Pour l'utiliser on peut ensuite utiliser la commande ci-dessous :

```
$ scrot -s copie.png
```

Il faut ensuite sélectionner la zone de l'écran que l'on souhaite copier. Cette zone doit être visible à l'écran lorsque la commande `scrot` est exécutée. L'image est alors automatiquement sauvegardée dans le fichier `copie.png`. Une autre possibilité est de cliquer sur la barre de titre d'une fenêtre pour faire une copie d'écran de toute la fenêtre.

Le fichier obtenu peut ensuite être visualisé avec l'éditeur `emacs` :

```
$ emacs copie.png
```

Si le fichier d'image obtenu n'est pas sur la bonne machine (celle sur laquelle le compte-rendu est rédigé), la commande `scp` (Secure CoPy) pourra être utilisée pour copier le fichier sur cette machine. Par exemple, la commande :

```
$ scp copie.png etudiant@192.168.43.1:/tmp
```

copie le fichier `copie.png` dans le répertoire `/tmp` de la machine `192.168.43.1`. `etudiant` est le nom du compte avec lequel la copie est effectuée. Il faut, au préalable, que le service `sshd` ait été démarré sur la machine `192.168.43.1` :

```
# systemctl start sshd.service
```

Essayez, pour tester, de prendre une copie d'écran du résultat d'une commande quelconque dans le terminal (par exemple, `ls`) puis d'envoyer avec `scp` le fichier à votre binôme.

Configuration du proxy

Il est nécessaire de configurer le proxy pour pouvoir accéder à Internet. Pour cela, dans Firefox :

- Aller dans le menu *Preferences*, puis, menu *Advanced*, onglet *Network*, bouton *Settings*.
- Sélectionner *Manual proxy configuration*.
- Entrer le nom et le port du proxy :
 - HTTP proxy : `aquawall`
 - Port : `3128`
- Cocher la case *Use this proxy server for all protocols*.
- Valider.

TP 1 — Ethernet

Le TP est à réaliser en binôme. Notez sur votre compte-rendu le numéro de groupe que vous aura attribué l'enseignant.

Exercice 1 — Analyse du trafic DHCP

L'objectif de cet exercice est d'analyser les messages DHCP échangés lors d'une demande ou d'une révocation de bail. Un bail consiste en une adresse IP attribuée à un client par un serveur DHCP pour une certaine durée. D'autres informations (p.ex., le masque de réseau) sont également transmises par le serveur à l'établissement du bail. La commande `dhclient` est utilisée à l'activation (demande de bail) ou à la désactivation (révocation du bail) d'une interface configurée via DHCP. Une machine du binôme jouera le rôle de serveur et l'autre le rôle de client.

I 1.1 Relier les interfaces `eth1` du serveur et du client.

Correction

Utiliser un câble croisé.

I 1.2 Sur le serveur : modifier le fichier de configuration de `eth1` (`/etc/sysconfig/network-scripts/ifcfg-eth1`) afin de lui attribuer statiquement l'adresse IP `10.18.N.254/24` (avec N = numéro de groupe).

Correction

Le fichier `/etc/sysconfig/network-scripts/ifcfg-eth1` du serveur doit contenir les lignes suivantes :

```
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.18.1.254
NETMASK=255.255.255.0
```

I 1.3 Sur le serveur : activer `eth1`.

Correction

```
$ ifup eth1
```

I 1.4 Sur le serveur : éditer le fichier de configuration DHCP (`/etc/dhcpd.conf`) afin qu'il contienne uniquement les lignes suivantes :

```
subnet <adresse-de-reseau> netmask <masque-de-reseau> {
    option routers <adresse-ip-du-routeur>;
    option domain-name-servers <adresse-ip-du-serveur-dns>;
    range <adresse-ip1> <adresse-ip2>;
}
```

Pour l'adresse IP du routeur on pourra donner l'adresse IP du serveur DHCP lui-même. Le serveur DHCP n'est pas configuré comme un routeur, mais on l'utilisera pour les tests. Pour l'adresse IP du serveur DNS on utilisera l'adresse IP d'un serveur DNS de google (8.8.8.8). Fixer une plage de 100 adresses IP disponibles.

Correction

Exemple de fichier `/etc/dhcpd.conf` valide :

```
subnet 10.18.1.0 netmask 255.255.255.0
{
    option routers 10.18.1.254;
    option domain-name-servers 8.8.8.8;
    range 10.18.1.1 10.18.1.253;
}
```

I 1.5 Sur le serveur : lancer le service `dhcpd` puis vérifier qu'il a été correctement lancé.

Correction

Sur le serveur :

```
$ systemctl start dhcpd.service
$ systemctl status dhcpd.service
```

Q 1.1 Donner une copie d'écran du fichier `/etc/sysconfig/network-scripts/ifcfg-eth1` du serveur.

Q 1.2 Donner une copie d'écran du fichier `/etc/dhcpd.conf` du serveur.

I 1.6 Sur une des deux machines : ouvrir wireshark puis démarrer la capture de trames sur eth1.

I 1.7 Sur le client : demander un nouveau bail pour eth1 (`dhclient eth1`) puis stopper la capture.

I 1.8 Sous wireshark : utiliser le filtre bootp pour n'afficher que les messages DHCP. (DHCP est une amélioration de BOOTP.)

Q 1.3 Quels sont les PCI (en-têtes de protocole) qui apparaissent dans les messages DHCP ?

Correction

Pour DHCP on trouve les PCI suivants : Ethernet - IP - UDP - DHCP. ◆

Q 1.4 Rappeler les quatre types de messages DHCP échangés lors d'une demande de bail.

Correction

DISCOVER, OFFER, REQUEST et ACK. ◆

Q 1.5 Analyser ces trames pour trouver les informations ci-dessous. Donner le cas échéant le type du message DHCP contenant l'information, le PCI dans lequel elle se trouve et le nom que lui donne wireshark.

1 l'adresse MAC et l'adresse IP de destination utilisées dans la demande envoyée par le client. Pourquoi envoie-t-il sa demande à cette adresse ?

Correction

Dans le message DHCP Discover envoyé par le client l'adresse MAC destination est FF:FF:FF:FF:FF:FF et l'adresse IP destination est 255.255.255.255. On trouve ces deux adresses dans le PCI Ethernet et dans le PCI IP.

Le client ne connaît ni l'adresse IP ni l'adresse MAC du serveur DHCP du réseau. Il utilise donc les adresses de diffusion (MAC = FF:FF:FF:FF:FF:FF et IP = 255.255.255.255) pour être sûr que le serveur DHCP recevra sa requête. L'utilisation de l'adresse MAC de diffusion garantit que la requête sera reçue et traitée par l'interface réseau du serveur DHCP. L'utilisation de l'adresse IP de diffusion garantit que la requête sera traitée par la couche IP (le système d'exploitation) du serveur DHCP. ◆

2 les numéros de port utilisé par `dhclient` et par le serveur DHCP

Correction

Le numéro de port du serveur DHCP se trouve dans les PCI UDP de tous les messages DHCP. C'est 68 pour le client et 67 pour le serveur. ◆

3 la liste des informations demandées par le client (chercher l'option *Parameter Request List*)

Correction

En regardant dans le message DHCP Discover on voit la liste des informations demandées par le client. C'est l'option 55 : Parameter Request List. En voici quelques exemples :

- Subnet Mask = masque de réseau
 - Domain Name Server = adresse IP du serveur DNS à interroger pour faire de la résolution de noms
 - Router = adresse IP du routeur du réseau
- ◆

4 l'adresse IP attribuée par le serveur

Correction

L'adresse IP attribuée par le serveur se trouve dans le message DHCP Offer : Your (client) IP address. ◆

5 la durée du bail (lease) accordé par le serveur

Correction

La durée du bail accordé par le serveur se trouve dans le message DHCP Offer : Option (51) IP address lease time. ◆

6 la liste des informations fournies par le serveur

Correction

La liste des informations fournies par le serveur se trouvent dans le message DHCP Offer. On retrouve les informations se trouvant dans le fichier de configuration du serveur comme : le masque de réseau (Subnet Mask), l'adresse IP du serveur DNS (Domain Name Server) et l'adresse IP du routeur (Router). ◆

I 1.9 Répéter les instructions I 1.6 à I 1.8 mais en utilisant maintenant la commande `dhclient -r eth1` pour révoquer le bail.

Q 1.6 Quel est le type du message DHCP envoyé lors de la révocation ?

Correction

C'est un message DHCP Release.

Q 1.7 Pourquoi est-il utile pour le serveur d'être informé de la révocation ?

Correction

Il sait de cette manière que l'adresse IP utilisée par le client n'est plus utilisée. Il peut alors réattribuer cette adresse IP à d'autres clients.

I 1.10 Sur le client : renouveler le bail de `eth1` pour les exercices suivants.

Exercice 2 — Influence du MTU sur le débit

Une trame Ethernet ne peut pas contenir plus de 1 500 octets. On peut modifier cette taille appelée MTU (*Maximum Transfer Unit*) avec `ifconfig` :

```
# ifconfig <nom-de-l-interface> mtu <taille-en-octets>
```

Dans cet exercice, nous allons étudier l'influence du MTU sur le débit. Pour mesurer ce débit on utilisera la commande `iperf`, vue dans le module M1101 (voir le TP 1). Une des deux machines jouera le rôle de client et l'autre de serveur. On utilisera les adresses IP des interfaces `eth1` dans cet exercice (celles sur le réseau 10.18.N.0/24).

I 2.1 Sur le client : écrire un script bash `mesure.sh` permettant de calculer le temps de transfert de 100 Mo entre les deux machines du binôme (via leurs interfaces `eth0`) avec les tailles de MTU suivantes : 1500, 800, 400, 200 et 100. Ce script devra utiliser une boucle `for`.

I 2.2 Sur le serveur : lancer `iperf` en mode serveur.

Correction

```
$ iperf -s
```

I 2.3 Sur le client : rendre le script `mesure.sh` exécutable et le lancer. On devrait voir le débit diminuer.

Correction

```
$ chmod +x mesure.sh
$ ./mesure.sh
```

Q 2.1 Donner une copie d'écran du script `mesure.sh`.

Correction

```
#!/bin/bash

for m in 1500 800 400 200 100
do
    ifconfig eth0 mtu $m
    echo "avec_un_mtu_de_$m octets"
    iperf -c 10.18.1.254 -n 100000000 | grep sec
done
```

Q 2.2 Donner une copie d'écran du résultat de l'exécution de `mesure.sh` dans le terminal du client.

Correction

En exécutant `mesure.sh` on obtient le résultat suivant :

```
$ ./mesure.sh
avec un mtu de 1500 octets
[ 5] 0.0- 0.9 sec 95.4 MBytes 940 Mbits/sec
avec un mtu de 800 octets
[ 5] 0.0- 1.2 sec 95.4 MBytes 683 Mbits/sec
avec un mtu de 400 octets
[ 5] 0.0- 2.0 sec 95.4 MBytes 394 Mbits/sec
avec un mtu de 200 octets
[ 5] 0.0- 4.0 sec 95.4 MBytes 200 Mbits/sec
avec un mtu de 100 octets
[ 5] 0.0-10.1 sec 95.4 MBytes 79.4 Mbits/sec
```

Q 2.3 Comment peut-on expliquer la diminution du débit lorsque le MTU diminue ?

Correction

Le débit affiché par iperf est le débit utile, c'est-à-dire le débit observé en prenant uniquement en compte les 100 Mo de données transférés (et pas les différents PCI se trouvant dans les trames). On observe que le temps de transfert augmente quand le MTU diminue.

Quand le MTU diminue on doit nécessairement envoyer plus de trames pour transférer le même volume de données. Deux raisons expliquent alors l'augmentation du temps de transfert :

- Entre la transmission de deux trames la carte Ethernet attend toujours pendant $9,6\mu\text{s}$ (l'IFG). En transmettant plus de trames le temps total d'attente augmente.
- Sur chaque trame on va retrouver différents PCI (Ethernet, IP et TCP dans le cas du test iperf) en plus des octets de données à envoyer. Par conséquent, en transmettant plus de trames le nombre total d'octets transmis augmente. ♦

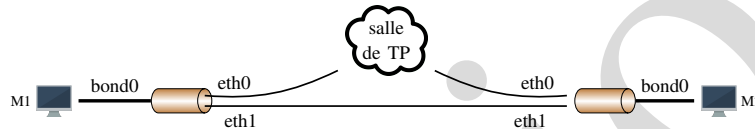
I 2.4 Sur le client : repasser le MTU d'eth1 à 1500 octets pour l'exercice suivant.

Exercice 3 — L'agrégation de liens

La technique de l'agrégation de liens ou *channel bonding* consiste à masquer plusieurs interfaces physiques (cartes Ethernet, Wifi, ...) derrière une interface virtuelle (nommées `bond0`, `bond1`, ...). Les interfaces physiques ne sont plus directement utilisables. Seule l'interface virtuelle l'est. On peut voir l'interface virtuelle comme un alias permettant de désigner n'importe laquelle des interfaces physiques qui lui sont associées.

De même que pour les interfaces physiques, l'interface virtuelle dispose d'une adresse IP. Pour que l'agrégation fonctionne il faut que les interfaces physiques et l'interface virtuelle aient la même adresse IP. Ainsi, lorsque la machine devra envoyer un paquet en utilisant cette adresse IP elle pourra utiliser indifféremment une des deux interfaces physiques.

Dans cet exercice nous allons réaliser l'architecture de la figure ci-dessous.



Pour envoyer un paquet à M2, M1 peut utiliser son interface `bond0`. Le système redirige alors le paquet soit vers `eth0`, auquel cas le paquet passe par le réseau de la salle de TP ; soit vers `eth1`, auquel cas il est envoyé directement à M2.

I 3.1 Noter l'adresse IP du routeur de la salle de TP (cette adresse sera utilisée dans la suite de l'exercice).

Correction

```
$ route -n
Table de routage IP du noyau
Destination  Passerelle      Genmask          Indic Metric Ref    Use Iface
0.0.0.0      192.168.43.254  0.0.0.0          UG     0      0      0 eth0
192.168.43.0 0.0.0.0          255.255.255.0    U      0      0      0 eth0
```

L'adresse IP du routeur sur la route par défaut est 192.168.43.254. ◆

I 3.2 Les instructions suivantes doivent être suivies sur les deux machines avec le compte root.

I 3.2.1 Activer le module bonding :

```
# modprobe bonding
```

(Les modules sont des extensions au système qui permettent de gérer les périphériques (cartes réseaux, cartes son, carte vidéo, ...).)

I 3.2.2 Une interface virtuelle a un fichier de configuration associé, comme les interfaces physiques. Créer le fichier de configuration de `bond0` (`/etc/sysconfig/network-scripts/ifcfg-bond0`) avec les paramètres adéquats afin qu'elle soit activée statiquement et qu'elle ait l'adresse IP et le masque associés actuellement à `eth0`.

Correction

Le fichier `/etc/sysconfig/network-scripts/ifcfg-bond0` doit contenir les lignes suivantes :

```
DEVICE=bond0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.43.24
NETMASK=255.255.255.0
```

I 3.2.3 La dernière étape est d'attacher les interfaces physiques à l'interface virtuelle. Cela consiste à indiquer dans les fichiers de configuration de `eth0` et `eth1` qu'elles sont *esclaves* (ou *slave*) de `bond0` qui est l'interface *maître* (ou *master*). Trois paramètres sont à ajouter ou modifier : **MASTER=bond0**, **SLAVE=yes** et **BOOTPROTO=none**.

Correction

Ce sont les fichiers `/etc/sysconfig/network-scripts/ifcfg-eth0` et `/etc/sysconfig/network-scripts/ifcfg-eth1` qu'il faut modifier. ◆

I 3.3 Sur M1 et M2 : désactiver `eth0` et `eth1` puis activer `bond0`.

Correction

```
$ ifdown eth0
$ ifdown eth1
$ ifup bond0
```

I 3.4 Sur M1 et M2 : afficher avec `ifconfig` les adresses IP et MAC de toutes les interfaces. On devrait observer qu'`eth0` et `eth1` n'ont plus d'adresses IP et que seule `bond0` en a une. Les trois interfaces ont normalement la même adresse MAC.

I 3.5 Mesurer avec `iperf` le temps de transfert de 1 Go de M1 (client) vers M2 (serveur).

Q 3.1 Donner une copie d'écran du résultat de l'exécution d'`ifconfig` (sur une des deux machines seulement).

Q 3.2 Donner une copie d'écran du résultat de l'exécution d'`iperf` (sur M1).

Q 3.3 Qu'observe-t-on sur le débit par rapport à celui observé dans l'exercice précédent (avec un MTU de 1 500 o.) ?

Correction

On observe que le débit est presque doublé. ◆

Q 3.4 Avec le bonding, pourquoi `eth0` et `eth1` doivent-elles nécessairement partager la même adresse MAC (penser à ce qui pourrait se passer avec ARP si les adresses étaient différentes) ?

Correction

La table ARP de M2 pourrait par exemple contenir la ligne (adr. IP de `bond0` de M1, adr. MAC de `eth0` de M1). Par conséquent, une trame envoyée sur l'interface `eth1` de M2 serait destinée à l'interface `eth0` de M1 mais reçue par l'interface `eth1` de M1. La trame ne lui étant pas destinée, elle serait donc ignorée. ◆

I 3.6 Sur une des deux machines : lancer la capture de trames puis envoyer 4 messages ping vers le routeur de la salle de TP. Utiliser le filtre `icmp` pour ne voir que les messages ping.

Q 3.5 Donner une copie d'écran de la capture wireshark.

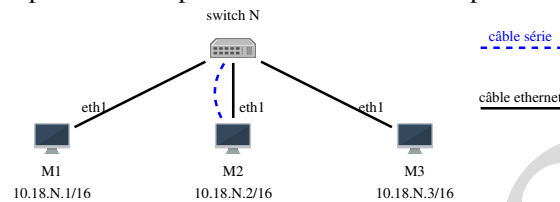
Q 3.6 Qu'observe-t-on sur cette capture ? Pourquoi ?

Correction

On faisant un ping vers 192.168.43.254, on voit que le routeur ne répond qu'à un message sur deux. Comme ping envoie les paquets sur `bond0` donc successivement sur `eth0` puis sur `eth1` les paquets envoyés sur `eth1` sont envoyés directement à M2 et le routeur ne le reçoit donc pas. ◆

TP 2 — Commutation

Le TP est à réaliser sur 3 machines. Notez sur votre compte-rendu le numéro de groupe que vous aura attribué l'enseignant. Chaque groupe travaillera sur un switch Cisco 2960-C. Les 3 machines seront câblées au switch selon la figure ci-dessous. N est le numéro du groupe. Ne câblez pas les machines au switch pour l'instant. Nous le ferons à l'exercice 2.



Seule la machine M2 sera câblée au switch via un câble série (bleu fin). C'est uniquement sur cette machine que sera lancé le programme minicom qui permet d'exécuter des commandes d'administration du switch.

Les commandes CISCO d'administration du switch sont décrites dans l'Annexe 2 (page 24).

Exercice 1 — Configuration du switch

Sur M2, configurer le switch en suivant les instructions de l'Annexe 1 (page 23).

Exercice 2 — Configuration des machines

- I 2.1** En modifiant les fichiers de configuration des interfaces, attribuer à l'interface eth1 de chacune des machines du groupe une adresse IP de la forme $10.18.N.M/16$ avec
- N = numéro de groupe
 - M = numéro de la machine dans le groupe (entre 1 et 3)

Correction

Fichier `/etc/sysconfig/network-scripts/ifcfg-eth1` de la machine M1 du groupe 1 :

```
DEVICE=eth1
BOOTPROTO=static
IPADDR=10.18.1.1
NETMASK=255.255.0.0
ONBOOT=yes
```

- I 2.2** Connecter les interfaces eth1 des trois machines au switch.

- I 2.3** Activer les interfaces eth1 des trois machines puis vérifier la connexion avec la commande ping.

Correction

```
$ ifup eth1
$ ping 10.18.1.2
```

Exercice 3 — La table de commutation

L'objectif de cet exercice est d'observer le fonctionnement de la commutation et les mécanismes de mise à jour de la table de commutation du switch en utilisant les protocoles ARP et ICMP (utilisé par ping).

- I 3.1** Pour ne pas perturber les expériences que nous ferons, nous allons d'abord modifier, sur M1 et M2, les deux paramètres système ci-dessous (exprimés en secondes) dont la valeur par défaut est trop faible.
- `net.ipv4.neigh.eth1.gc_stale_time` = durée de vie des lignes de la table ARP
 - `net.ipv4.neigh.eth1.delay_first_probe_time` = temps au bout duquel une machine vérifie par une requête ARP qu'une ligne de la table ARP est toujours valide

Un paramètre système peut-être modifié comme ceci :

```
# sysctl param=valeur
```

Attribuer à ces paramètres une valeur de 1 heure.

Correction

Sur M1 et M2 :

```
# sysctl net.ipv4.neigh.eth1.gc_stale_time=3600
# sysctl net.ipv4.neigh.eth1.delay_first_probe_time=3600
```

I 3.2 De même, fixer à 1 heure la durée de vie des entrées de la table de commutation du switch.

Correction

```
S(config)# mac address-table aging-time 3600
```

I 3.3 Effectuer des pings entre M1 et M2 pour remplir leurs tables ARP.

I 3.4 Vérifier les contenus des tables ARP de M1 et M2 avec la commande `arp -n`. L'adresse MAC de M1 doit apparaître dans la table de M2 et vice versa.

I 3.5 Lancer sur les trois machines la capture de trames sur eth1.

I 3.6 Vider la table de commutation du switch. *Après avoir vidé la table aucune commande ne doit être exécutée sur les trois machines, ceci afin de ne pas modifier le contenu de la table de commutation observé par la suite.*

Correction

```
S# clear mac address-table dynamic
```

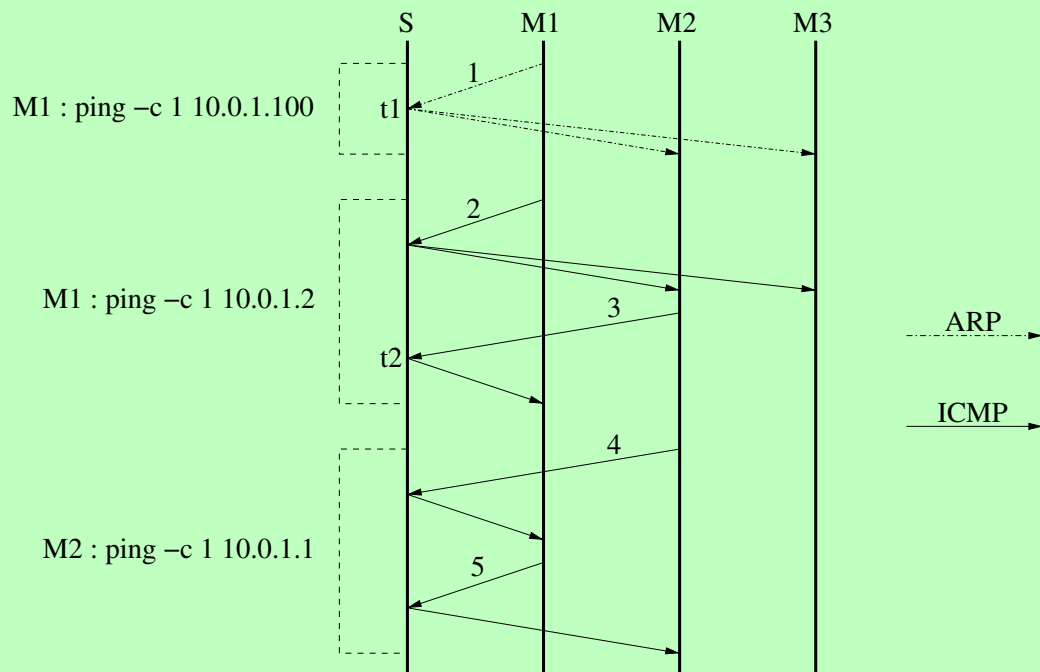
I 3.7 Exécuter les trois commandes suivantes :

- (a) sur M1 : un seul ping (option `-c 1`) vers une machine inexistante du réseau 10.18.0.0/16
- (b) sur M1 : un seul ping vers M2
- (c) sur M2 : un seul ping vers M1

I 3.8 Stopper la capture de trames sur les trois machines.

I 3.9 Vérifier les captures avec l'enseignant.

Q 3.1 Détailler à l'aide d'un chronogramme les échanges de trames entre le switch, M1, M2 et M3 lors de l'exécution des trois commandes ping du point I 3.7. Indiquer sur le chronogramme la nature de chaque trame.

Correction

t1 = ajout de M1 dans la table de commutation
t2 = ajout de M2 dans la table de commutation

Détail des trames :

- (a) requête ARP (qui a l'adresse MAC correspondant à 10.18.1.100 ?) reçue par M2 et M3 car envoyée en diffusion

- (b) demande d'écho ICMP envoyée à 10.18.1.2 et reçue par M2 et M3 car S n'a pas l'adresse MAC de M2 dans sa table de commutation
- (c) réponse ICMP envoyée à 10.18.1.1 et reçue par M1 uniquement car le switch a maintenant M1 dans sa table de commutation (depuis t1)
- (d) demande d'écho ICMP envoyée à 10.18.1.1
- (e) réponse ICMP envoyée à 10.18.1.2 et reçue par M2 uniquement car le switch a maintenant M2 dans sa table de commutation (depuis t2)

Q 3.2 À quelle étape et pourquoi ARP est-il utilisé ?

Correction

ARP est utilisé uniquement par le premier ping pour connaître l'adresse MAC correspondant à 10.18.1.100. Pour les deux autres pings, les adresses IP de destination sont déjà dans les table ARP des machines.

Q 3.3 Quelles sont les trames reçues par M3 ? Expliquer pourquoi M3 a reçu ces trames.

Correction

La première requête ARP parce qu'elle est envoyée en diffusion et le premier ping de M1 vers M2 car le switch n'avait pas encore l'adresse MAC de M2 dans sa table de commutation et a donc retransmis le ping sur tous ses ports.

Q 3.4 Donner une copie d'écran de la commande CISCO affichant les entrées dynamiques de la table de commutation.

Q 3.5 Les adresses MAC de M1 et M2 apparaissent-elles dans la table de commutation ? Pourquoi ?

Correction

Oui. Le switch a reçu des trames ayant M1 et M2 pour source. Il a donc mémorisé leurs adresses MAC dans sa table de commutation.

Q 3.6 L'adresse MAC de M3 apparaît-elle dans la table de commutation ? Pourquoi ?

Correction

Non. Le switch n'a reçu aucune trame de M3. Or seules les adresses sources des trames émises sont mémorisées par le switch.

Exercice 4 — Connexion des switches

I 4.1 Relier le switch du groupe à celui d'un autre groupe.

I 4.2 Envoyer des messages pings vers une machine de cet autre groupe.

Q 4.1 Donner une copie d'écran de la commande CISCO affichant les entrées dynamiques de la table de commutation.

Q 4.2 Quelles nouvelles lignes contient la table de commutation après les envois de ping ? Quel port y est associé ?

Correction

L'adresse MAC du switch de l'autre groupe et l'adresse MAC de la machine pinguée. Les deux devraient être associées au port auquel est relié le switch de l'autre groupe.

I 4.3 Débrancher les câbles entre switches pour les exercices suivants.

Exercice 5 — Détournement de trames

L'objectif de cet exercice est de réaliser une attaque réseau très simple basée sur l'usurpation d'adresse MAC. M1 et M2 seront les victimes de M3 qui sera l'attaquant. M3 va se faire passer pour M2 en usurpant son adresse MAC afin de détourner le trafic entre M1 et M2.

I 5.1 Noter les adresses MAC des interfaces eth1 de M2 et M3.

Correction

On les récupère avec ifconfig (hwaddr ou ether).

I 5.2 Sur M1 : envoyer des pings en continu à une fréquence de 5 secondes (option -i secondes de ping) vers M2.

Ne pas interrompre la commande ping pour la suite.

Correction

Depuis M1 :

```
$ ping -i 5 10.18.1.2
```

I 5.3 Démarrer la capture de trames sur les trois machines.

I 5.4 Sur M3 : attribuer à eth1 l'adresse MAC de M2 en utilisant ifconfig avec la syntaxe suivante :

```
# ifconfig eth1 hw ether XX:XX:XX:XX:XX:XX
```

I 5.5 Sur M3 : envoyer *un seul ping* à M1.

I 5.6 Stopper la capture de trames.

I 5.7 Vérifier la capture avec l'enseignant.

Q 5.1 Donner une copie d'écran de la capture de trames chez M3.

Q 5.2 Que constate-t-on sur la capture de trames chez M2 et M3 ?

Correction

C'est maintenant M3 qui reçoit les demandes d'écho ICMP envoyées par M1. M2 ne reçoit plus rien.

Q 5.3 Expliquer le fonctionnement de cette attaque en vous basant sur le contenu de la table de commutation du switch lors de l'attaque.

Correction

Lorsque M3 envoie un ping vers M1, le switch supprime la ligne de la table de commutation associée à l'adresse MAC de M2 (maintenant aussi utilisée par M3) et insère une nouvelle ligne avec cette adresse mais avec le port auquel est branchée M3. C'est pourquoi le switch retransmet ensuite vers M3 les demandes d'écho envoyées par M1. (Mais M3 ne répond pas à la demande d'écho car l'adresse IP demandée n'est pas la sienne mais celle de M2.)

Q 5.4 Que doit faire M2 pour contrer cette attaque ?

Correction

Il suffit que M2 retransmette une trame pour que le switch rechange le port associé à l'adresse MAC de M2.

I 5.8 Réaffecter son adresse MAC à l'interface eth1 de M3 pour l'exercice suivant.

Exercice 6 — Sécurisation de port

Pour empêcher le type d'attaque de l'exercice précédent une possibilité que nous allons mettre en œuvre est de sécuriser les ports du switch.

I 6.1 Sécuriser le port du switch auquel est branchée M3 afin que *seule* l'adresse MAC de eth1 de M3 soit autorisée.

Correction

Pour sécuriser le port :

```
S(config)# int fa 0/1
S(config-if)# switchport mode access
S(config-if)# switchport port-security
S(config-if)# switchport port-security mac-address AA:BB:CC:DD:EE:FF
S(config-if)# switchport port-security maximum 1
```

Q 6.1 Donner une copie d'écran des commandes CISCO utilisées pour sécuriser le port.

I 6.2 Vérifier que l'adresse MAC est bien dans la table de commutation du switch.

Correction

```
S> show mac address-table
```

- I 6.3 Sur M3 : vérifier qu'il est possible d'envoyer un ping aux autres machines du groupe.
- I 6.4 Retenter l'attaque de l'exercice précédent.
- I 6.5 Vérifier que la diode du port de connexion de M3 au switch est bien éteinte.

Q 6.2 Donner une copie d'écran des messages affichés dans le terminal minicom lors de l'attaque.

TP 3 — STP et VLAN

Le TP est à réaliser sur 3 machines. Notez sur votre compte-rendu le numéro de groupe que vous aura attribué l'enseignant. Chaque groupe travaillera sur un switch Cisco 2960-C. Il faut en début de séance réaliser l'architecture du TP 2 et reprendre les exercices 1 et 2 de ce TP pour configurer le switch et les interfaces eth1 des trois machines reliées au switch.

Exercice 1 — Analyse des BPDU

I 1.1 Lancer la capture de trames sur l'interface eth1.

I 1.2 Attendre que quelques trames STP (les BPDU) arrivent puis stopper la capture.

Q 1.1 Trouver dans une des BPDU capturées les 4 champs vus en cours (racine, distance, pont émetteur, port d'émission). Comment wireshark les nomme-t-il ?

Correction

On trouve ces informations dans le PCI Spanning Tree Protocol :

- racine = *Root Identifier*
- distance = *Root Path Cost*
- pont émetteur = *Bridge Identifier*
- port d'émission = *Port Identifier*

Q 1.2 À qui appartient l'adresse MAC de la racine dans les BPDU et que vaut la distance ? Pourquoi ?

Correction

Comme le switch n'est pour l'instant relié à aucun autre switch, il est la racine. On observe donc que la racine est l'adresse MAC du switch et que la distance est de 0.

Q 1.3 L'adresse MAC de destination utilisée dans les BPDU est une adresse réservée. Quelle est-elle ? Pourquoi utilise-t-on une adresse spéciale ? (Penser à ce que ferait le switch de la BPDU si c'était une adresse quelconque.)

Correction

L'adresse MAC de destination dans une BPDU est toujours 01:80:C2:00:00:00. On utilise une adresse réservée pour que le switch sache que c'est une BPDU qui lui est destinée. Si c'était une adresse quelconque le switch la retransmettrait comme n'importe quelle autre trame.

Exercice 2 — Spanning Tree Protocol

Cet exercice doit être réalisé simultanément sur trois switchs que nous nommerons S1, S2 et S3.

I 2.1 Lancer la capture de trames sur eth1.

I 2.2 Relier trois switchs en chaîne, sans créer de boucle : S1 branché à S2 et S2 branché à S3.

I 2.3 Stopper la capture.

Q 2.1 Noter les adresses MAC de S1, S2 et S3 et le numéro de votre switch (S1, S2 ou S3).

Q 2.2 Quel est le switch racine parmi S1, S2 et S3 ? Pourquoi ?

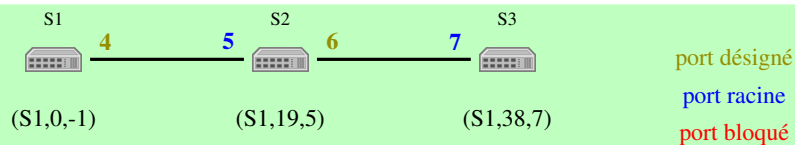
Correction

Celui qui a la plus petite adresse MAC.

Q 2.3 Dessiner la topologie du réseau en montrant les configurations des ponts (racine, distance, port racine), et les états des ports (désigné, racine ou bloqué).

Correction

On suppose dans l'exercice que les adresses MAC sont ordonnées ainsi : S1 < S2 < S3.



Q 2.4 Supposons que l'on crée une boucle en branchant le switch S1 au switch S3. Quel(s) serai(en)t alors le(s) port(s) bloqué(s) par STP? Justifier.

Correction

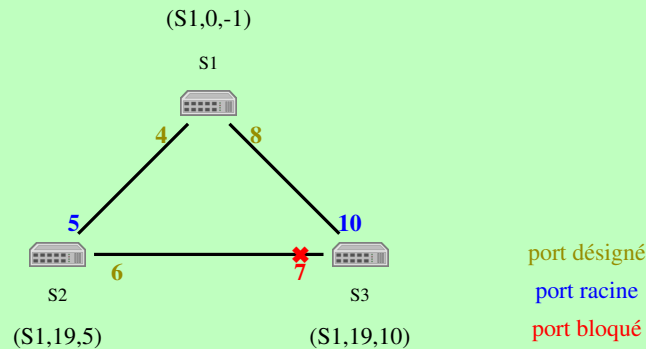
On se retrouverait alors dans la situation où deux switches non racine et à égale distance de la racine (S2 et S3) sont reliés entre eux. C'est alors le switch avec la plus grande adresse MAC qui bloque son port qui le mène à l'autre. Dans notre cas, ça serait donc S3 qui bloquerait son port qui le relie à S2 (on a fait l'hypothèse que $S1 < S2 < S3$).

I 2.4 Relier S1 à S3.

I 2.5 Vérifier votre hypothèse en affichant le(s) port(s) bloqué(s) par STP.

Q 2.5 Dessiner la nouvelle topologie du réseau.

Correction



I 2.6 Lancer des pings en continu entre deux machines branchées à des switches différents.

I 2.7 Débrancher (sans stopper les pings) un des câbles par lesquels passent les messages ICMP envoyés par le ping. Pendant quelques dizaines de secondes la communication est coupée mais le réseau devrait ensuite récupérer et les messages ICMP passer de nouveau.

Q 2.6 Que s'est-il passé à la déconnexion du câble qui a permis ensuite au réseau de récupérer ?

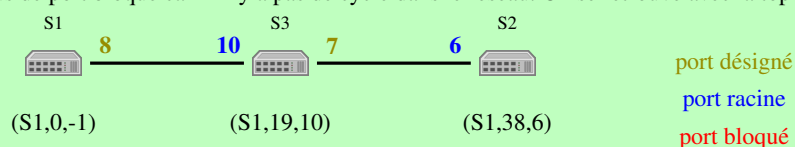
Correction

Supposons qu'on lance, depuis une machine reliée à S2, un ping vers une machine reliée à S3. Avec la topologie que l'on a les messages ICMP passent par S2, S1 et S3. Si on débranche le câble entre S1 et S2 pendant le ping, le problème sera immédiatement détecté par S1 et S2 qui vont déclencher le protocole STP, ce qui va avoir pour conséquence pour S3 de débloquent son port 7. Au bout d'une trentaine de secondes (temps qui dépend en fait de la configuration de STP sur les switches), le protocole se termine et les pings fonctionnent de nouveau.

Q 2.7 A-t-on maintenant des ports bloqués dans le réseau des switches ? Pourquoi ?

Correction

Non, il n'y a plus de port bloqué car il n'y a pas de cycle dans le réseau. On se retrouve avec la topologie ci-dessous.



I 2.8 Rebrancher le câble débranché précédemment pour revenir à la configuration initiale.

Nous avons vu que STP utilise les adresses MAC pour déterminer le pont racine et les ports bloqués. Le problème est que ces adresses étant fixes, l'administrateur du réseau n'a pas la possibilité de déterminer quel switch sera la racine et/ou quels ports seront bloqués. La configuration obtenue ne sera donc pas forcément la configuration souhaitée par l'administrateur (voir TD 2). Un système de priorités existe afin de résoudre ce problème. L'administrateur peut associer une priorité (un entier) à chaque switch. Les switches sont d'abord ordonnés sur leurs priorités et, à priorité égale, sur leurs adresses MAC. C'est donc le switch qui aura la plus petite priorité qui deviendra la racine. La priorité est toujours un multiple de 4 096. Il y a 16 niveaux allant de 0 à 61 440. Par défaut la priorité d'un switch est de 32 768.

En utilisant le système de priorités on souhaite arriver à la configuration suivante : le switch ayant un port bloqué deviendra la racine, et celui qui est la racine aura un port bloqué.

Q 2.8 Comment faut-il ordonner les priorités pour arriver à cette configuration ? Justifier.

Correction

Si S3 avait un port bloqué il faut lui associer la plus petite priorité pour qu'il devienne la racine. La liaison entre S1 et S2 sera ensuite coupée en raison d'un port bloqué, soit par S1, soit par S2. Si on souhaite que S1 bloque son port il faut qu'il ait une priorité plus élevée que S2. Si on note P_1 , P_2 et P_3 les priorités de S_1 , S_2 et S_3 on doit donc avoir : $P_1 > P_2 > P_3$. ♦

I 2.9 Attribuer sa priorité à votre switch.

I 2.10 Vérifier votre hypothèse en affichant le(s) port(s) bloqué(s) par STP.

Q 2.9 Donner une copie d'écran des commandes CISCO utilisées aux points I 2.9 et I 2.10.

I 2.11 Débrancher tous les câbles entre switches pour les exercices suivants.

Exercice 3 — Création des VLAN

I 3.1 Créer deux VLAN d'identifiants 100 et 200 nommés respectivement rt et info.

Correction

```
S(config)# vlan 100
S(config-name)# name rt
S(config-name)# exit
S(config)# vlan 200
S(config-name)# name info
S(config-name)# exit
```

I 3.2 Affecter

- les ports du switch auxquels sont reliés M1 et M2 au VLAN 100;
- et le port du switch auquel est relié M3 au VLAN 200.

Correction

```
S(config)# int fa 0/1
S(config-if)# switchport mode access
S(config-if)# switchport access vlan 100
S(config-if)# exit
S(config)# int fa 0/2
S(config-if)# switchport mode access
S(config-if)# switchport access vlan 100
S(config-if)# exit
S(config)# int fa 0/3
S(config-if)# switchport mode access
S(config-if)# switchport access vlan 200
S(config-if)# exit
```

I 3.3 Vérifier que les VLAN ont bien été créés et que les ports ont bien été affectés aux VLAN.

Correction

```
S# show vlan brief
```

I 3.4 Vérifier que M2 peut bien envoyer des pings à M1.

I 3.5 Avec wireshark (lancé sur les trois machines) capturer les trames échangées lors de l'envoi d'un ping de M3 vers M1. Le ping ne devrait normalement pas passer.

Q 3.1 Expliquer, en vous basant sur l'analyse des trames capturées, pourquoi la commande ping ne fonctionne pas.

Correction

Avant l'envoi du message ICMP, M1 envoie une requête ARP en diffusion pour demander l'adresse MAC correspondant à l'adresse IP de M3. Le switch redirige la requête ARP uniquement vers les ports associés au même VLAN que M1 (le 100). M3 ne reçoit donc pas la requête ARP. M2 la reçoit mais l'ignore. M1 n'envoie donc pas son message ICMP.

Q 3.2 Donner des copies d'écran des commandes CISCO utilisées aux points I 3.1, I 3.2 et I 3.3.

Exercice 4 — Réalisation d'un trunk entre switches

Cet exercice doit être réalisé simultanément sur deux switches.

I 4.1 Relier les deux switches.

I 4.2 Configurer le port de connexion à l'autre switch afin de le passer en mode trunk.

Correction

```
S(config)# int fa 0/4
S(config-if)# switchport mode trunk
```

(4 = numéro du port sur lequel l'autre switch est relié)

I 4.3 Faire les vérifications suivantes :

- Les machines M1 et M2 du premier switch peuvent bien pinguer les machines M1 et M2 du deuxième switch.
- La machine M3 du premier switch peut bien pinguer la machine M3 du deuxième switch.
- Tout autre ping est impossible.

I 4.4 Mettre en place le monitoring de port pour que toutes les trames envoyées ou reçues sur le port relié à l'autre switch soient recopiées sur le port de la machine M2.

Correction

```
S(config)# monitor session 1 source int fa0/4
S(config)# monitor session 1 destination int fa0/2 encapsulation replicate
```

(4 = numéro du port sur lequel l'autre switch est relié)

I 4.5 Sur M2 : capturer un seul message ping entre deux machines du même VLAN mais connectées à des switches différents.

Q 4.1 Donner une copie d'écran de la commande CISCO affichant les ports trunk du switch.

Q 4.2 Donner une copie d'écran des PCI affichés par wireshark dans le panneau du milieu lorsque l'on sélectionne une des trames capturées.

Correction

On devrait normalement voir Ethernet - Étiquette 802.1Q - IP - ICMP.

Q 4.3 Les trames capturées circulant sur la liaison trunk n'ont pas la même structure que les trames standard. Quelles informations supplémentaires y trouve-t-on ?

Correction

On remarque que les trames contenant les messages ICMP et circulant sur la liaison trunk entre les deux switches contiennent l'étiquette de VLAN. Cette étiquette contient l'identifiant du VLAN auquel appartient la machine émettrice de la trame. Quand un switch reçoit une trame en provenance d'une machine et qu'il doit transmettre cette trame sur une liaison trunk, il ajoute cette étiquette de 4 octets pour indiquer au switch à l'autre bout de la liaison l'identifiant du VLAN sur lequel la trame circule. Cette étiquette permet au switch destinataire de savoir vers quel(s) port(s) retransmettre la trame.

Correction

TP 4 — Routage inter-VLAN

Le TP est à réaliser sur 3 machines. Notez sur votre compte-rendu le numéro de groupe que vous aura attribué l'enseignant. Chaque groupe travaillera sur un switch Cisco 2960-C. L'objectif est de réaliser une architecture avec 3 machines M1, M2 et M3 reliées à un switch. Une contrainte de confidentialité impose que les trames échangées entre M1 et M2 ne doivent pas être vues par M3. De même, les trames échangées entre M1 et M3 ne doivent pas être vues par M2. On propose donc la solution suivante :

- Créer deux VLANS d'identifiants 20 et 30.
- Placer M2 et M1 sur le VLAN 20.
- Placer M3 et M1 sur le VLAN 30.

Par contre, il n'y a pas de contrainte sur les données échangées entre M2 et M3. Pour que M2 et M3 puissent s'échanger des données, on propose donc d'associer chaque VLAN à un réseau IP et d'utiliser M1 pour jouer le rôle de routeur entre ces 2 VLAN. Pour envoyer un paquet IP à M3, M2 l'encapsulera d'abord dans une trame destinée à M1 et circulant sur le VLAN 20 ; puis M1 routera ce paquet en l'encapsulant dans une trame destinée à M3 et circulant sur le VLAN 30.

Les deux VLAN seront associés aux deux réseaux IP selon le tableau ci-dessous.

VLAN	Réseau IP	Machines
20	18.20.0.0/16	M1 18.20.N.1/16
		M2 18.20.N.2/16
30	18.30.0.0/16	M1 18.30.N.1/16
		M3 18.30.N.3/16

N = numéro du groupe

Deux architectures de routage vont être étudiées.

Exercice 1 — Configuration des machines M2 et M3 et des VLAN

Dans un premier temps nous allons configurer M2 et M3 uniquement. La configuration de M1 dépendra de la solution de routage choisie.

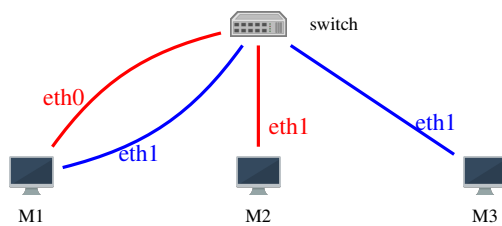
- I 1.1 Reprendre la procédure de configuration du switch (Annexe 1, page 23).
- I 1.2 En modifiant les fichiers de configuration des interfaces, attribuer aux interfaces eth1 de M2 et M3 les adresses indiquées dans le tableau ci-dessus.
- I 1.3 Connecter les interfaces eth1 de M2 et M3 au switch.
- I 1.4 Activer les interfaces eth1 de M2 et M3.
- I 1.5 Créer les VLAN 20 et 30 sur le switch et leur attribuer des noms.
- I 1.6 Associer les ports auxquels sont reliées M2 et M3 aux VLAN 20 et 30 respectivement.
- I 1.7 Vérifier que les VLAN ont bien été créés et que les ports ont bien été affectés aux VLAN.

Q 1.1 Donner une copie d'écran de la commande CISCO utilisée au point I 1.7.

- I 1.8 Sur M1 : installer, avec la commande `urpmi vlan-utils`, le package `vlan-utils` (utilisé par la suite).

Exercice 2 — Routage avec plusieurs interfaces

La première solution de routage consiste à réaliser le réseau de la Figure 1. Les deux interfaces eth0 et eth1 de M1 sont utilisées. Chacune est associée à un VLAN différent. Selon le VLAN sur lequel elle souhaite envoyer une trame, M1 utilisera soit eth0, soit eth1.



Machine	Interface	VLAN	Adresse IP
M1	eth0	20	18.20.N.1/16
M1	eth1	30	18.30.N.1/16
M2	eth1	20	18.20.N.2/16
M3	eth1	30	18.30.N.3/16

(N = num. du groupe)

FIGURE 1 – Une première solution de routage avec une interface par VLAN sur M1

I 2.1 Réaliser le câblage de la Figure 1.

I 2.2 Attribuer aux interfaces de M1 les adresses IP de la Figure 1.

I 2.3 Associer les ports auxquels est reliée M1 aux VLAN 20 et 30 comme indiqué sur la Figure 1.

I 2.4 Configurer M1 pour qu'elle accepte de router les paquets en modifiant le paramètre système `net.ipv4.ip_forward`.

Correction

Sur M1 :

```
$ sysctl net.ipv4.ip_forward=1
```

I 2.5 Modifier les tables de routage de M2 et M3 :

— sur M2 : créer une route vers le réseau 18.30.0.0/16 passant par M1

— sur M3 : créer une route vers le réseau 18.20.0.0/16 passant par M1

L'ajout d'une route vers un réseau en remise indirecte se fait par la commande ci-dessous :

```
# route add -net <adresse-du-reseau> netmask <masque-du-reseau> gw <adresse-du-routeur>
```

Correction

Sur M2, la route vers le réseau 18.30.0.0/16 passe par le routeur 18.20.0.1/16 :

```
# route add -net 18.30.0.0 netmask 255.255.0.0 gw 18.20.1.1
```

Symétriquement, sur M3 :

```
# route add -net 18.20.0.0 netmask 255.255.0.0 gw 18.30.1.1
```

I 2.6 Vérifier que M2 est bien accessible depuis M3 et vice versa.

I 2.7 Sur M2 : utiliser la commande `tracpath <adresse-ip-de-M3>`.

Q 2.1 Donner une copie d'écran des tables de routage (commande `route -n`) de M2 et M3.

Q 2.2 Donner une copie d'écran de la commande `tracpath` (point I 2.7).

Q 2.3 Qu'affiche la commande `tracpath` ?

Correction

La liste des routeurs sur le chemin vers la destination.

Exercice 3 — Routage avec une liaison trunk

Une deuxième solution (voir Figure 2) consiste à n'utiliser qu'une interface de M1 (eth1) et à configurer la liaison de M1 au switch comme une liaison trunk afin que M1 puisse recevoir et émettre des trames étiquetées sur eth1.

En présence de VLAN sur une interface X, Linux crée pour chaque VLAN d'identifiant V une interface virtuelle nommée ethX.V. Nous aurons donc deux interfaces virtuelles pour eth1 : eth1.20 et eth1.30 qui joueront respectivement le rôle de eth0 et de eth1 dans la solution précédente.

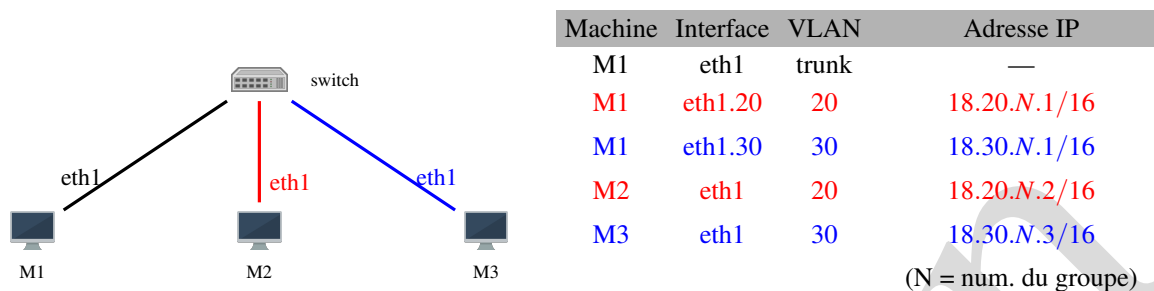


FIGURE 2 – Une deuxième solution de routage avec une liaison trunk sur M1

Comme les interfaces physiques, les interfaces virtuelles ont un fichier de configuration associé. Par exemple, pour l'interface virtuelle eth1.20, c'est /etc/sysconfig/network-scripts/ifcfg-eth1.20. Ces fichiers doivent contenir la ligne **VLAN=yes** pour indiquer que l'interface virtuelle est associée à un VLAN. On doit aussi y trouver les mêmes paramètres que pour les interfaces physiques configurées statiquement : **DEVICE**, **BOOTPROTO**, **IPADDR**, **NETMASK**.

I 3.1 Désactiver les interfaces eth0 et eth1 de M1.

I 3.2 Réaliser le câblage de la Figure 2.

I 3.3 Passer la liaison du switch à M1 en mode trunk.

I 3.4 Configuration des VLAN sur M1

I 3.4.1 Avec la commande modprobe, charger le module 8021q qui permet à Linux de comprendre les trames étiquetées.

I 3.4.2 Créer les VLAN 20 et 30 pour eth1. L'ajout d'un VLAN sur une interface se fait avec la commande vconfig :

```
# vconfig add <interface> <id-de-vlan>
```

I 3.4.3 Modifier les fichiers de configuration des interfaces virtuelles de M1 pour leur attribuer les adresses de la Figure 2.

I 3.4.4 Activer les interfaces eth1.20 et eth1.30 de M1.

I 3.5 Vérifier que M2 est bien accessible depuis M3 et vice versa.

I 3.6 Capturer, sur M1, les trames échangées lors de l'envoi d'un seul message ping de M2 vers M3.

Q 3.1 Donner une copie d'écran du fichier de configuration de l'interface eth1.20 de M1.

Q 3.2 Donner une copie d'écran d'une étiquette de VLAN observée sur Wireshark (pour une trame seulement).

Q 3.3 Qu'est-ce qui change dans l'étiquette de VLAN avant et après le routage par M1 ?

Correction

Dans la trame qui arrive à M1 le VLAN-ID vaut 20 (VLAN de M2). Après retransmission le VLAN-ID vaut 30 (VLAN de M3).

Exercice 4 — Filtrage inter-VLAN

M1 peut aussi être utilisée pour filtrer les paquets entre M2 et M3 puisque ceux-ci passent par M1. L'objectif de cet exercice est de mettre en place avec iptables un filtrage des paquets SSH sur M1 afin qu'il soit impossible sur M2 de se connecter en SSH sur M3 (et inversement).

Nous allons, dans un premier temps, vérifier que la connexion SSH fonctionne.

- I 4.1 Sur M2 : démarrer le service sshd qui permet aux autres machines de se connecter sur celle-ci.
- I 4.2 Sur M3 : ouvrir une connexion SSH sur M2.
- I 4.3 Sur M3 : fermer la connexion SSH.

Q 4.1 Donner une copie d'écran de la commande ssh utilisée au point I 4.2.

Nous allons maintenant mettre en place le filtrage des paquets SSH sur M1.

- I 4.4 Sur M1 : utiliser une commande iptables permettant :
 - (a) de refuser de router les paquets SSH ;
 - (b) et de prévenir la source du paquet que le paquet a été rejeté.
- Rechercher sur Internet les options iptables à utiliser.

Correction

```
# iptables -A FORWARD -p tcp --dport 22 -j REJECT
```

- I 4.5 Retenter l'instruction ssh du point I 4.2 pour vérifier que le filtrage fonctionne.
- I 4.6 Vérifier que les pings passent bien entre M2 et M3.

Q 4.2 Donner une copie d'écran des commandes utilisées aux points I 4.4 à I 4.6.

Annexe 1 — Configuration initiale des switchs

La procédure doit être suivie uniquement sur la machine reliée au switch par le port console. Le switch doit être éteint au début de la procédure et le câble série bleu branché de la machine qui administrera le switch au port console du switch.

I 1 Démarrage du switch

- I 1.1 Lancer la commande `minicom -s` (s pour setup) pour démarrer minicom avec le menu de configuration du switch.
- I 1.2 Sélectionner *Configuration du port série* dans le menu.
- I 1.3 Modifier les paramètres *Port série* et *Débit/Parité/Bits* pour leur donner les valeurs suivantes : `/dev/ttyUSB0` et `9600 8N1`. Le fichier `/dev/ttyUSB0` identifie le port usb connecté au switch et `9600 8N1` correspond à un débit de 9600 bit/s avec 1 bit de parité tous les 8 bits (ce sont les caractéristiques de la liaison série).
- I 1.4 Revenir au menu de minicom (touche *Entrée*).
- I 1.5 Sortir du menu (*Sortir*). Le message *Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales* devrait s'afficher dans le terminal.
- I 1.6 Brancher maintenant le câble d'alimentation du switch tout en maintenant le bouton *MODE* (à l'avant du switch) pressé *jusqu'à ce qu'un message soit affiché dans terminal minicom*. Ce bouton permet de modifier le mot de passe d'administration du switch et autorise la suppression de la configuration existante.
- I 1.7 Entrer les commandes suivantes :

```
switch: flash_init
switch: del flash:config.text
switch: del flash:vlan.dat
switch: boot
```

La première commande initialise la mémoire interne du switch. Les deux suivantes suppriment la configuration précédente. La dernière démarre le système d'exploitation du switch.

I 2 Configuration du switch — La configuration précédente ayant été supprimée, un dialogue commence pour reconfigurer le switch. Donner les réponses suivantes aux questions posées.

- I 2.1 Would you like to enter the initial configuration dialog? [yes/no] → yes
- I 2.2 Would you like to enter basic management setup? [yes/no] : → yes
- I 2.3 Enter host name [Switch] : → SN (avec N le numéro du groupe)
- I 2.4 Enter enable secret : → titi
- I 2.5 Enter enable password : → toto
- I 2.6 Enter virtual terminal password : → tutu
- I 2.7 Configure SNMP Network Management? [no] : → no
- I 2.8 Enter interface name used to connect to the management network from the above interface summary : → Vlan1
- I 2.9 Configure IP on this interface? [no] : → no
- I 2.10 Would you like to enable as a cluster command switch? [yes/no] : → no
- I 2.11 Enter your selection [2] : → 2

Seul le premier mot de passe saisi (titi) sera utile par la suite. Il permet de passer en mode administrateur sur le switch. Les deux autres servent sur les versions antérieures de l'IOS ou pour configurer le switch à distance.

Une fois la configuration terminée, le message `SN>` devrait s'afficher dans le terminal, SN étant le nom du switch choisi au point I 2.3. Vous êtes maintenant sur un terminal de l'IOS et avez accès aux commandes de configuration du switch.

Annexe 2 — Langage de commande des switches

Cet annexe décrit brièvement les commandes de l’IOS Cisco (*Internetwork Operating System*, le système d’exploitation installé sur la plupart des équipements Cisco) qui seront utilisées dans les TP de ce module. Ce ne sont pas des commandes linux. Elles doivent être exécutées dans le terminal d’administration du switch ouvert avec minicom. Ce programme doit être lancé uniquement sur la machine reliée au switch par le câble console.

Les différents modes

Il y a trois niveaux de permission pour l’exécution des commandes de l’IOS :

- le mode *utilisateur* qui donne uniquement accès à des commandes de consultation ;
- le mode *privilégié* ou *enable* qui donne accès à certaines commandes de modification (p.ex., vider la table de commutation) mais qui ne permet pas de changer la configuration du switch ;
- le mode *configuration* qui donne accès aux commandes de configuration du switch.

Le message d’invite affiché par le switch permet de distinguer le niveau actuel. En mode utilisateur ce message est `S>` (S étant le nom du switch attribué durant la configuration initiale du switch : voir l’Annexe 1). En mode privilégié, ce message devient `S#`. Enfin, en mode configuration le message est `S(config)#`.

Voici les commandes pour passer d’un mode à l’autre :

- mode utilisateur → mode privilégié : **enable**. Il faut alors entrer le mot de passe choisi durant la configuration initiale du switch.
- mode privilégié → mode configuration : **conf term**
- pour revenir au mode précédent : **exit**

Aide

À tout moment, il est possible de taper le caractère `?` pour que l’IOS propose les différentes commandes possibles dans le mode actuel ou les différentes possibilités pour compléter une commande non terminée.

Comme sous Linux, une commande peut être complétée en utilisant la tabulation. De même, il est possible de naviguer dans l’historique des commandes exécutées avec les flèches `↑` et `↓`.

Nommage des ports du switch

Le switch attribue aux ports à 100 Mbit/s (Fast Ethernet) un nom de la forme `fa 0/X` où X est le numéro du port (de 1 à 8 sur les switches 2960-C que nous utilisons). De même, les noms attribués aux interfaces à 1 Gbit/s sont de la forme `gi 0/X`.

La table de commutation

La table de commutation du switch associe adresses MAC et numéros de ports. Il y a deux types d’entrées dans cette table : les entrées *statiques* (ajoutées par l’administrateur du switch ou identifiant des adresses MAC réservées) et *dynamiques* (ajoutées par le switch). Les entrées dynamiques ont une durée de vie limitée : passé un certain délai elles sont automatiquement supprimées de la table par le switch.

- Afficher les entrées dynamique de la table de commutation :

```
S> show mac address-table dynamic
```

- Vider la table de commutation (les entrées dynamiques uniquement) :

```
S# clear mac address-table dynamic
```

- Afficher la durée de vie des entrées dynamiques de la table :

```
S> show mac address-table aging-time
```

- Changer la durée de vie des entrées de la table à N secondes :

```
S(config)# mac address-table aging-time N
```

Sécurisation de port

La sécurisation d'un port consiste à n'autoriser que certaines adresses MAC à envoyer des trames au switch sur un port donné. Si le switch reçoit sur ce port une trame dont l'adresse MAC n'est pas dans la liste des adresses autorisées (on parle de *violation de sécurité*), la politique par défaut est de fermer complètement le port (politique *shutdown*). D'autres politiques sont applicables comme celle d'ignorer simplement la trame (politique *restrict*).

— Afficher le détail des informations de sécurité d'un port :

```
S# show port-security int fa 0/X
```

— Activer la sécurisation sur un port :

```
S(config)# int fa 0/X
S(config-if)# switchport mode access
S(config-if)# switchport port-security
```

— Autoriser une adresse MAC sur un port :

```
S(config)# int fa 0/X
S(config-if)# switchport port-security mac-address XX:XX:XX:XX:XX:XX
```

— Fixer à N le nombre maximal d'adresses MAC autorisées sur un port :

```
S(config)# int fa 0/X
S(config-if)# switchport port-security maximum N
```

— Réactiver un port désactivé (par exemple, en cas de violation de sécurité) :

```
S(config)# int fa 0/X
S(config-if)# shutdown
S(config-if)# no shutdown
```

Le protocole STP

STP est le protocole vu en cours qui permet de bloquer certains ports pour éviter les boucles dans le réseau. Dans la version de STP utilisée sur ces switches, il existe un arbre couvrant par VLAN. Un système de priorités permet à l'administrateur de choisir le switch qui deviendra la racine : c'est le switch ayant la plus petite priorité (plutôt que la plus petite adresse MAC) qui deviendra la racine.

— Afficher les informations de STP :

```
S# show spanning-tree
```

— Afficher les ports bloqués :

```
S# show spanning-tree blockedports
```

— Modifier la priorité d'un switch sur un VLAN :

```
S(config)# spanning-tree vlan X priority Y
```

Les VLAN

Chaque VLAN a un identifiant allant de 1 à 4094 et un nom utilisé uniquement par l'administrateur. Par défaut, toutes les machines sont sur le VLAN 1. Les ports du switch peuvent être en mode *access* : le port est associé à un unique VLAN ; ou en mode *trunk* : le port n'est associé à aucun VLAN en particulier et les trames transmises ou reçues sur ce port sont étiquetées.

— Créer un VLAN et lui donner un nom :

```
S(config)# vlan X
S(config-vlan)# name X
```

— Supprimer un VLAN :

```
S(config)# no vlan X
```

— Affecter un port à un VLAN :

```
S(config)# int fa 0/X
S(config-if)# switchport mode access
S(config-if)# switchport access vlan X
```

- Passer un port en mode trunk :

```
S(config)# int fa 0/X
S(config-if)# switchport mode trunk
```

- Afficher les associations port↔VLAN

```
S# show vlan brief
```

- Afficher les ports en mode trunk

```
S# show interfaces trunk
```

Monitoring de port

Le monitoring de port consiste à observer les trames transmises ou reçues sur un (ou des) port(s) du switch en les recopiant vers un port auquel est branchée une machine *monitrice*. Cette machine monitrice recevra donc toutes les trames reçues ou transmises sur un port observé. Plusieurs sessions de monitoring peuvent être lancées simultanément mais nous n'utiliserons que la session 1.

- Préciser un port source (un port que l'on veut observer) :

```
S(config)# monitor session 1 source int fa0/X
```

- Préciser le port de destination (celui auquel est branchée la machine monitrice) :

```
S(config)# monitor session 1 destination int fa0/X encapsulation replicate
```

(“encapsulation replicate” ⇔ la trame est recopiée sans modification vers le port destination.)

- Afficher les informations de monitoring :

```
S> show monitor session 1
```

- Supprimer les informations de monitoring :

```
S(config)# no monitor session 1
```