# TP 1 — Ethernet

Le TP est à réaliser en binôme sur l'image MAGEIA4. Un compte-rendu par binôme sera ramassé.

## 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 ainsi que d'autres informations (p.ex., le masque de réseau) attribuées à un client par un serveur DHCP pour une certaine durée. 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.

**1** Relier les interfaces eth1 du serveur et du client.

**2** Sur le serveur : modifier le fichier de configuration de l'interface eth1 afin de lui attribuer statiquement l'adresse IP 10.0.0.254/24. On rappelle que ce fichier est `/etc/sysconfig/network-scripts/ifcfg-eth1`.

**3** Sur le serveur : activer eth1.

**4** Sur le serveur : éditer le fichier de configuration DHCP (`/etc/dhcpd.conf`) afin qu'il ait la structure suivante :

```
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).

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

**Q. 1.1** Donner le contenu du fichier `/etc/dhcpd.conf` du serveur.

**6** Sur une des deux machines : ouvrir `wireshark` puis démarrer la capture de trames sur eth1.

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

**8** Utiliser le filtre `bootp` pour que `wireshark` n'affiche que les messages DHCP. (DHCP est une amélioration de BOOTP.)

**Q. 1.2** Quels sont les PCIs qui apparaissent dans les messages DHCP ?

**Q. 1.3** Tracer sur un chronogramme les échanges effectués entre le client et le serveur en faisant apparaître les types de messages DCHP échangés que l'on voit dans la colonne Info.

**Q. 1.4** 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 ?
2. le protocole de transport utilisé par DHCP (UDP ou TCP)
3. les numéros de port utilisés par `dhclient` et par le serveur DHCP
4. la liste des informations demandées par le client (voir l'option *Parameter Request List*)
5. l'adresse IP attribuée par le serveur
6. la durée du bail (lease) accordé par le serveur
7. la liste des informations fournies par le serveur

**9** Répéter les instructions 6 à 8 mais en utilisant maintenant la commande `dhclient -r eth1` pour révoquer le bail.

**Q. 1.5** Quel est le type du message DHCP envoyé lors de la révocation ?

**Q. 1.6** Pourquoi est-il utile pour le serveur d'être informé de la révocation ?

**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 10.0.0.0/24).

**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.

**2** Sur le serveur : lancer `iperf` en mode serveur.

**3** Sur le client : rendre le script `mesure.sh` exécutable et le lancer.

**Q. 2.1** Donner le contenu du script `mesure.sh`.

**Q. 2.2** Comment peut-on expliquer les différences dans les temps de transfert observés ?

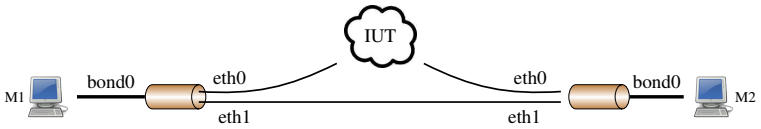
**4** Sur le client : repasser le MTU d'eth1 à 1500 octets.

## 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 dipose 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 transite sur le réseau de l'IUT ; soit vers eth1, auquel le cas il est envoyé directement à M2.

**1** Noter l'adresse IP du routeur sur la route par défaut (cette adresse sera utilisée dans la suite de l'exercice).

**2** Les instructions suivantes doivent être suivies sur les deux machines.

**2.1** Activer le module bonding avec la commande `modprobe`. Les modules sont des extensions au système qui permettent de gérer les périphériques (cartes réseaux, cartes son, carte vidéo, ...).

**2.2** Avec la commande `lsmod`, vérifier que le module a bien été chargé.

**2.3** 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 au démarrage de la machine et qu'elle ait l'adresse IP et le masque associés actuellement à eth0.

**2.4** 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`.

**2.5** Désactiver eth0 et eth1 puis activer bond0.

**Q. 3.1** Donner les contenus des fichiers suivants d'une des deux machines :
- `/etc/sysconfig/network-scripts/ifcfg-bond0`
- `/etc/sysconfig/network-scripts/ifcfg-eth0`

**Q. 3.2** Mesurer avec `iperf` le temps de transfert de 1 Go entre M1 et M2. Qu'observe-t-on ?

**Q. 3.3** En utilisant `ifconfig` que remarque-t-on sur les adresses IP et MAC des deux interfaces physiques ?

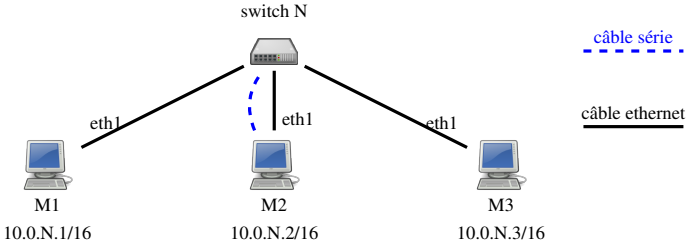
**Q. 3.4** Que pourrait-il se passer avec ARP si les adresses MAC utilisées par eth0 et eth1 étaient différentes ?

**Q. 3.5** Lancer la capture de trames puis envoyer des messages `ping` en continu vers le routeur sur la route par défaut. Qu'observe-t-on ? Pourquoi ?

# TP 2 — Commutation

Le TP est à faire par groupe de 2 ou 3 (selon le nombre de switchs disponibles) sur 3 machines. Un compte-rendu par groupe sera ramassé.

Les 3 machines du groupe seront câblées au switch selon la figure ci-dessous.




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 dont nous aurons besoin dans ce TP sont décrites dans les annexes.

## Exercice 1 — Configuration du switch

Sur M2, configurer le switch en suivant les instructions de l'annexe 1.

## Exercice 2 — Configuration des machines

**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 $10.0.N.M/16$ avec
- N = numéro de groupe
- M = numéro de la machine dans le groupe (entre 1 et 3)

**2** Connecter les interfaces eth1 des trois machines au switch.

**3** Activer les interfaces eth1 des trois machines puis vérifier la connexion avec la commande `ping`.

## 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).

**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

La syntaxe de la commande `sysctl` utilisée pour modifier un paramètre est la suivante :

```
$ sysctl param=valeur
```

Attribuer à ces paramètres une valeur de 1 heure.

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

**3** Effectuer des `pings` entre M1 et M2 pour remplir leurs tables ARP.

**4** Vérifier les contenus des deux tables avec la commande `arp -n`.

**5** Lancer sur les trois machines la capture de trames sur eth1.

**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.*

**7** Exécuter les trois commandes suivantes :
(a) depuis M1 : un seul `ping` (option `-c 1`) vers une machine inexistante du réseau $10.0.0.0/16$
(b) depuis M1 : un seul `ping` vers M2
(c) depuis M2 : un seul `ping` vers M1

**8** Stopper la capture de trames sur les trois machines.

**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 7.
**Q. 3.2** À quelle étape et pourquoi ARP est-il utilisé ?
**Q. 3.3** Quelles sont les trames reçues par M3 ? Pourquoi ?
**Q. 3.4** Quelles adresses MAC apparaissent dans la table de commutation ? Pourquoi ?

## Exercice 4 — Connexion des switchs

**1** Relier le switch du groupe à celui d'un autre groupe.
**2** Envoyer des messages `ping` vers une machine de cet autre groupe.

**Q. 4.1** Quelles nouvelles lignes contient la table de commutation après l'envoi du `ping` ? Quel port y est associé ?

## 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. Deux des machines seront les victimes (M1 et M2) de la troisième machine 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.

**1** Débrancher les câbles entres switchs.
**2** Noter les adresses MAC de M2 et M3.
**3** Envoyer, depuis M1, des `ping`s en continu à une fréquence de 5 secondes (option `-i secondes` de `ping`) vers la machine M2. Ne pas interrompre la commande `ping` pour la suite.
**4** Démarrer la capture de trames sur les trois machines.
**5** Attribuer à M3 l'adresse MAC de M2 en utilisant `ifconfig` avec la syntaxe suivante :

```
$ ifconfig ethX hw ether XX:XX:XX:XX:XX:XX
```

**6** Envoyer, depuis M3, un seul message `ping` à M1.
**7** Stopper la capture de trames.

**Q. 5.1** Que constate-t-on sur la capture de trames ?
**Q. 5.2** Expliquer comment cette attaque fonctionne.
**Q. 5.3** Quelles sont les limites de cette attaque ?

## 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.

**1** Réaffecter son adresse MAC à l'interface eth1 de M3.
**2** Sécuriser le port du switch auquel est branchée M3 afin que seule l'adresse MAC de eth1 de M3 soit autorisée.
**3** Vérifier que l'adresse MAC est bien dans la table de commutation du switch.
**4** Depuis M3, essayer d'envoyer un `ping` aux autres machines du groupe.
**5** Retenter l'attaque de l'exercice précédent (sans capturer les trames).

**Q. 6.1** Donner les commandes CISCO utilisées pour sécuriser le port.
**Q. 6.2** Qu'observe-t-on sur le switch après avoir retenté l'attaque ?

# TP 3 — STP et VLANs

Le TP est à faire par groupe de 2 ou 3 (selon le nombre de switchs disponibles) sur 3 machines. Un compte-rendu par groupe sera ramassé.

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 du TP 2 pour configurer le switch et les interfaces eth1 des trois machines reliées au switch.

Les commandes CISCO d'administration du switch dont nous aurons besoin dans ce TP sont décrites dans l'annexe.

## Exercice 1 — Analyse des BPDUs

**1** Lancer la capture de trames sur l'interface eth1.
**2** Attendre que quelques trames STP (les BPDUs) arrivent puis stopper la capture.

**Q. 1.1** Trouver dans une des BPDUs capturées les 4 champs vus en cours (racine, distance, pont émetteur, port d'émission). Comment `wireshark` les nomme-t-il ?
**Q. 1.2** À qui appartient l'adresse MAC de la racine dans les BPDUs et que vaut la distance ? Pourquoi ?
**Q. 1.3** Quelle est l'adresse MAC de destination utilisée dans les BPDUs ?

## Exercice 2 — Spanning Tree Protocol

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

**1** Lancer la capture de trames sur eth1.
**2** Relier trois switchs en chaîne, sans créer de boucle : S1 branché à S2 et S2 branché à S3.
**3** Stopper la capture.

**Q. 2.1** Quels sont les changements observés dans les BPDUs capturées ?
**Q. 2.2** Dessiner la topologie du réseau en montrant les configurations des ponts (racine, distance, port racine), et les états des ports (désigné, racine ou bloqué).
**Q. 2.3** 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.

**4** Relier S1 à S3.
**5** Vérifier votre hypothèse en affichant le(s) port(s) bloqué(s) par STP.

**Q. 2.4** Dessiner la nouvelle topologie du réseau.

**6** Lancer des `ping`s en continu entre deux machines branchées à des switchs différents.
**7** Débrancher (sans stopper les `ping`s) 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. 2.5** Que s'est-il passé à la déconnexion du câble qui a permis ensuite au réseau de récupérer ?
**Q. 2.6** A-t-on maintenant des ports bloqués dans le réseau des switchs ? Pourquoi ?

**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 switchs 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. 2.7** Comment faut-il ordonner les priorités pour arriver à cette configuration ? Justifier.

**9** Attribuer sa priorité à votre switch.

**10** Vérifier votre hypothèse en affichant le(s) port(s) bloqué(s) par STP.

## Exercice 3 — Création des VLANs

**1** Débrancher tous les câbles entre switchs.
**2** Créer deux VLANs d'identifiants 100 et 200 nommés respectivement rt et info.
**3** Affecter
— les ports du switch auxquels sont reliées M1 et M2 au VLAN 100 ;
— et le port du switch auquel est relié M3 au VLAN 200.
**4** Vérifier que les VLANs ont bien été créés et que les associations port↔VLAN sont effectives.
**5** Vérifier que M2 peut bien envoyer des pings à M1.
**6** Avec `wireshark` (lancé sur les trois machines) capturer les trames échangées lors de l'envoi d'un ping de M1 vers M3.

**Q. 3.1** Expliquer, en vous basant sur l'analyse des trames capturées, pourquoi la commande `ping` ne fonctionne pas.
**Q. 3.2** Donner les commandes utilisées pour créer le VLAN 100.
**Q. 3.3** Donner les commandes utilisées pour affecter le port de connexion de M3 au VLAN 200.

## Exercice 4 — Réalisation d'un trunk entre switchs

Cet exercice doit être réalisé simultanément sur deux switchs.

**1** Relier les deux switchs.
**2** Configurer le port de connexion à l'autre switch afin de le passer en mode trunk.
**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.
**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.
**5** Analyser les messages ICMP reçus par le moniteur M2 lors de `ping`s entre machines du même VLAN mais connectées à des switchs différents.

**Q. 4.1** Les trames capturées circulant sur la liaison trunk n'ont pas la même structure que les trames standard. Quelle(s) information(s) supplémentaire(s) y trouve-t-on ?
**Q. 4.2** Donner les commandes utilisées pour passer le port en mode trunk.
**Q. 4.3** Donner les commandes utilisées pour mettre en place le monitoring de port.

# TP 4 — Routage inter-VLANs

Le TP est à faire par groupe de 2 ou 3 (selon le nombre de switchs disponibles) sur 3 machines. Un compte-rendu par groupe sera ramassé.

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

| VLAN | Réseau IP | Machines | |
|---|---|---|---|
| 20 | 10.20.0.0/16 | M1 | 10.20.*N*.1/16 |
| | | M2 | 10.20.*N*.2/16 |
| 30 | 10.30.0.0/16 | M1 | 10.30.*N*.1/16 |
| | | M3 | 10.30.*N*.3/16 |

N = numéro du switch

Deux architectures de routage vont être étudiées.

## Exercice 1 — Configuration des machines M2 et M3 et des VLANs

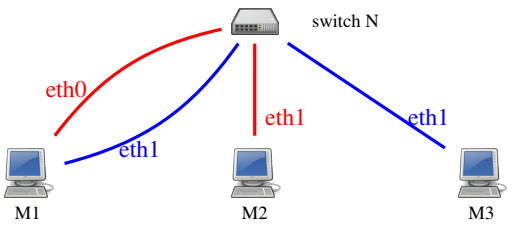
Dans un premier temps nous allons configurer M2 et M3 uniquement. La configuration de M1 dépendra de la solution de routage choisie.

1. Reprendre la procédure de configuration du switch (voir l'annexe en ligne). Il faut exécuter `del flash:vlan.dat` après avoir exécuté `del flash:config.text`, ceci afin de supprimer les informations existantes sur les VLANs.
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.
3. Connecter les interfaces eth1 de M2 et M3 au switch.
4. Activer les interfaces eth1 de M2 et M3.
5. Créer les VLANs 20 et 30 sur le switch et leur attribuer des noms.
6. Associer les ports auxquels sont reliées M2 et M3 aux VLANs 20 et 30 respectivement.
7. Vérifier que les VLANs ont bien été créés et que les associations port↔VLAN sont effectives.
8. Sur M1, installer, avec la commande `urpmi vlan-utils`, le package vlan-utils (utilisé par la suite).

**Q. 1.1** Donner les commandes utilisées pour créer les VLANS et leur associer les ports.

## 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 interface eth0, soit son interface eth1.




| Machine | Interface | VLAN | Adresse IP |
|---|---|---|---|
| M1 | eth0 | 20 | 10.20.*N*.1/16 |
| M1 | eth1 | 30 | 10.30.*N*.1/16 |
| M2 | eth1 | 20 | 10.20.*N*.2/16 |
| M3 | eth1 | 30 | 10.30.*N*.3/16 |

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

**1** Réaliser le câblage de la Figure 1.
**2** Réaliser, pour M1, les associations interface↔adresse IP de la Figure 1.
**3** Réaliser sur le switch l'association port↔VLAN de la Figure 1.
**4** Configurer M1 pour qu'elle accepte de router les paquets en modifiant le paramètre système `net.ipv4.ip_forward`.
**5** Modifier les tables de routage de M2 et M3 :
— sur M2 : créer une route vers le réseau 10.30.0.0/16 passant par M1
— sur M3 : créer une route vers le réseau 10.20.0.0/16 passant par M1
L'ajout d'une route vers un réseau en remise indirecte se fait par la commande ci-dessous :

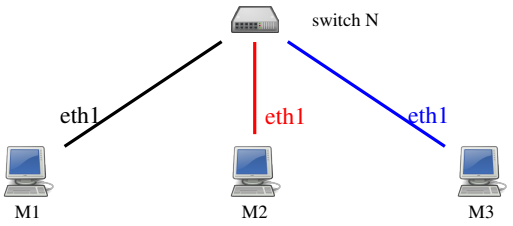
```
$ route add -net <adresse-du-reseau> netmask <masque-du-reseau> gw <adresse-du-routeur>
```

**6** Vérifier que M2 est bien accessible depuis M3 et vice versa.
**7** Sur M2 : utiliser la commande `tracepath <ip-de-M3>`.

**Q. 2.1** Donner les commandes utilisées pour modifier les tables de routage de M2 et M3 (point 5).
**Q. 2.2** Qu'affiche la commande `tracepath` ?

## 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.




| Machine | Interface | VLAN | Adresse IP |
|---|---|---|---|
| M1 | eth1 | trunk | — |
| M1 | eth1.20 | 20 | 10.20.*N*.1/16 |
| M1 | eth1.30 | 30 | 10.30.*N*.1/16 |
| M2 | eth1 | 20 | 10.20.*N*.2/16 |
| M3 | eth1 | 30 | 10.30.*N*.3/16 |

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

En présence de VLANs 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.

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`**.

**1** Désactiver les interfaces eth0 et eth1 de M1.
**2** Réaliser le câblage de la Figure 2.
**3** Passer la liaison du switch à M1 en mode trunk.
**4** *Configuration des VLANs sur M1*
**4.1** Avec la commande `modprobe`, charger le module 8021q qui permet à Linux de comprendre les trames étiquetées.
**4.2** Créer les VLANs 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>
```

**4.3** Modifier les fichiers de configuration des interfaces virtuelles de M1 pour leur attribuer les adresses de la Figure 2.
**4.4** Activer les interfaces eth1.20 et eth1.30 de M1.
**5** Vérifier que M2 est bien accessible depuis M3 et vice-versa.
**6** Capturer, sur M1, les trames échangées lors de l'envoi d'un seul message `ping` de M2 vers M3.

**Q. 3.1** Que remarque-t-on sur les deux trames capturées transportant le message ICMP ?
**Q. 3.2** Donner le contenu du fichier de configuration de l'interface eth1.20 de M1.

# 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.

**1** *Démarrage du switch*

**1.1** Lancer la commande `minicom -s` (`s` pour setup) pour démarrer minicom avec le menu de configuration du switch.

**1.2** Sélectionner *Configuration du port série* dans le menu.

**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).

**1.4** Revenir au menu de minicom (touche *Entrée*).

**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.

**1.6** Brancher maintenant le 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.

**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.

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

**2.1** Would you like to enter the initial configuration dialog ? [yes/no] → yes

**2.2** Would you like to enter basic management setup ? [yes/no] : → yes

**2.3** Enter host name [Switch] : → SN (avec N le numéro du groupe)

**2.4** Enter enable secret : → titi

**2.5** Enter enable password : → toto

**2.6** Enter virtual terminal password : → tutu

**2.7** Configure SNMP Network Management ? [no] : → no

**2.8** Enter interface name used to connect to the management network from the above interface summary : → Vlan1

**2.9** Configure IP on this interface ? [no] : → no

**2.10** Would you like to enable as a cluster command switch ? [yes/no] : → no

**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. Vous êtes maintenant sur un terminal de l'IOS et avez accès aux commandes de configuration du switch.

# Annexe 2 — Langage de commande des switchs

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 TPs 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'exercice 1 du TP 2). 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 switchs 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 la table de commutation :

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

— 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 appliquables 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 switchs, 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 VLANs

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
```

## 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 simultanèment 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
```