Configuration d'équipements CISCO

Résumé

Ce document décrit 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 réseaux. Ce ne sont pas des commandes linux. Elles doivent être exécutées dans le terminal d'administration du switch/routeur ouvert avec minicom (voir Section 1).

Table des matières

1 Interfaçage avec le switch

Pour pouvoir interfacer le poste avec le switch afin, ensuite, d'exécuter des commandes CISCO, il faut connecter le poste au switch avec un câble série puis lancer un terminal minicom. Voici les instructions détaillées.

1. Connectez le poste au switch avec un câble série (bleu), *sur le port console du switch* (pas sur un port Ethernet).
2. Installez minicom avec les droits root :

```
# apt update
# apt install minicom
```

3. Lancez minicom en mode *setup* et avec les droits root :

```
# minicom -s
```

4. Sélectionnez *Configuration du port série* dans le menu.
5. Modifiez 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.
6. Revenez au menu de minicom (touche *Entrée*).
7. Sortez du menu (*Sortir*).

Le message *Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales* devrait s'afficher dans le terminal. Vous devriez normalement voir le message d'invite du switch, par exemple, **Switch>**, dans le terminal minicom. Tapez sinon sur *Entrée* et il devrait s'afficher. Vous pouvez maintenant exécuter, dans ce terminal, des commandes de l'IOS.

2 Remarques générales

2.1 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 ;
— le mode *configuration* qui donne accès aux commandes de configuration.

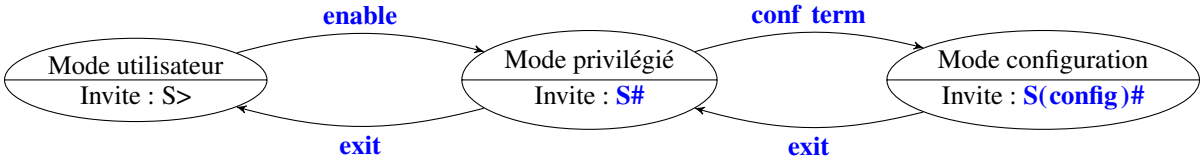
Le message d'invite affiché permet de distinguer le niveau actuel. En mode utilisateur ce message est **S>** (S étant le nom du switch attribué durant la configuration initiale du switch). En mode privilégié, ce message devient **S#**. Enfin, en mode configuration le message est **S(config)#**.

Le mode de configuration a différents "sous-modes". Par exemple, lors de la configuration d'une interface (c'est-à-dire d'un port) du switch, on rentre dans le sous-mode de configuration d'une interface. Le message d'invite devient alors **S(config-if)#** (if pour interface).

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

— mode utilisateur → mode privilégié : **enable**. Si un mot de passe est demandé, il faut entrer celui choisi durant la configuration initiale du switch.
— mode privilégié → mode configuration : **conf term**
— pour revenir au mode précédent : **exit**

La figure ci-dessous résume le fonctionnement des modes.



2.2 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 ↓.

2.3 Nommage des ports

Le switch attribue aux ports de la forme **gi1/0/P** (gi pour gigabit ethernet) où **P** est le numéro du port (par exemple, de 1 à 24 sur les switchs de la salle P203). Pour le routeur, ce nom est de la forme **gi0/0/P** avec **P** $\in \{0, 1, 2, \ldots\}$ selon l'interface.

3 Commandes de l'IOS

3.1 Configurations d'exécution (running-config) et de démarrage (startup-config)

Avec les équipements CISCO, on désigne par *running-config*, la configuration actuelle du switch ; et par *startup-config*, la configuration de démarrage du switch.

— Pour sauvegarder la *running-config* pour le prochain redémarrage :

```
S# copy running-config startup-config
```

Pour remettre le switch à zéro, il suffit d'effacer sa configuration de démarrage et de le rédémarrer.

— Effacer la configuration de démarrage :

```
S# erase startup-config
```

— Redémarrer :

```
S# reload
```

(Après le redémarrage, répondez par `no` aux questions posées pour ne pas rentrer dans le dialogue de configuration initiale du switch.)

3.2 Table de commutation

La table de commutation du switch associe adresses MAC et numéros de port. 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 comme l'adresse de diffusion) et *dynamiques* (ajoutées par le switch à la réception d'une trame). 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
```

— Fixer à **N** secondes la durée de vie des entrées dynamiques de la table :

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

3.3 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 source 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 **P** :

```
S# show port-security int gi1/0/P
```

— Activer la sécurisation sur le port **P** :

```
S(config)# int gi1/0/P
S(config-if)# switchport mode access
S(config-if)# switchport port-security
```

— Autoriser une adresse MAC **AA:BB:CC:DD:EE:FF** sur un port **P** :

```
S(config)# int gi0/0/P
S(config-if)# switchport port-security mac-address AA:BB:CC:DD:EE:FF
```

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

```
S(config)# int gi0/0/P
S(config-if)# switchport port-security maximum N
```

— Réactiver un port **P** désactivé en cas de violation de sécurité :

```
S(config)# int gi0/0/P
S(config-if)# shutdown
S(config-if)# no shutdown
```

3.4 STP

STP est le protocole 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
```

— Fixer à **P** la priorité du switch sur le VLAN **V** :

```
S(config)# spanning-tree vlan V priority P
```

3.5 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 **V** et lui donner un **NOM** :

```
S(config)# vlan V
S(config-vlan)# name NOM
```

Sur certaines versions de l'IOS, il est nécessaire d'exécuter la commande suivante avant de pouvoir créer un VLAN :

```
S(config)# vtp mode server
```

— Supprimer un VLAN **V** :

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

— Affecter un port **P** à un VLAN **V** :

```
S(config)# int gi0/0/P
S(config-if)# switchport mode access
S(config-if)# switchport access vlan V
```

— Passer un port **P** en mode trunk :

```
S(config)# int gi0/0/P
S(config-if)# switchport mode trunk
```

— Consulter l'état des VLAN

```
S> show vlan
```

— Afficher les associations port↔VLAN

```
S# show vlan brief
```

3.6 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. Il est à noter que la machine monitrice ne peut plus émettre de trames. Elle sert uniquement à observer le trafic Plusieurs sessions de monitoring peuvent être lancées simultanément mais nous n'utiliserons que la session 1.

— Préciser un port source **S** (un port que l'on veut observer) :

```
S(config)# monitor session 1 source int gi0/0/S
```

— Préciser le port de destination **D** (celui auquel est branchée la machine monitrice) :

```
S(config)# monitor session 1 destination int gi0/0/D encapsulation replicate
```

("encapsulation replicate" ⇔ la trame est recopiée sans modification vers le port destination. Cette option n'est pas reconnue par toutes les versions de l'IOS. Ne pas l'indiquer en cas d'erreur.)

— Afficher les informations de monitoring :

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

— Supprimer les informations de monitoring :

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

3.7 Débit

— Pour passer à **D** Mbit/s le débit sur le port **P** :

```
S(config)# int gi0/0/P
S(config-if)# speed D
```

— Pour afficher le débit sur le port **P** :

```
S> show int gi0/0/P
```

3.8 Qualité de service

Sur certaines versions de l'IOS il est d'abord nécessaire d'activer la qualité de service (QoS) :

— Pour activer la qualité de service :

```
S(config)# mls qos
```

(mls signifie *Multi Layer Switching*.)

Si l'on veut gérer la qualité de service (QoS), chaque port doit être configuré comme un port de confiance ou *trusted port*, c'est-à-dire un port pour lequel le marquage de priorité des trames entrantes est fiable, ou comme un *untrusted port*. Dans ce dernier cas, la priorité (aussi appelée *class of service* ou cos) associée aux trames sera la plus faible, c'est-à-dire 0. Par défaut, tous les ports sont des untrusted ports.

— Indiquer que le port **P** est un port de confiance :

```
S(config)# int gi0/0/P
S(config-if)# mls qos trust cos
```

— Préciser la priorité **I** ($\in [0,7]$) associée d'un port de confiance **P** :

```
S(config)# int gi0/0/P
S(config-if)# mls qos cos I
```

— Pour afficher les informations de QoS d'un port **P** :

```
S> show mls qos interface gi0/0/P
```

3.9 IP

3.9.1 Adressage (switch)

On peut attribuer une (ou des) adresse(s) IP au switch. C'est indispensable si l'on souhaite administrer le switch à distance (p.ex., via telnet) ou le surveiller grâce à SNMP. En présence de VLAN, il faut lui attribuer une IP sur chaque VLAN. Le VLAN 1 est le VLAN par défaut. Il faut donc attribuer au switch une adresse IP sur ce VLAN si l'on souhaite qu'il puisse communiquer avec les machines qui n'ont été rattachées à aucun VLAN.

— Attribuer au switch une IP **A.B.C.D** avec le masque **W.X.Y.Z** sur un VLAN **V** :

```
S(config)# interface vlan V
S(config-if)# ip address A.B.C.D W.X.Y.Z
S(config-if)# no shutdown
```

L'instruction **no shutdown** permet d'activer l'interface.

— Vérifier les informations IP associées à un VLAN **V** :

```
S> show ip interface vlan V
```

— Envoyer des messages ping :

```
S> ping A.B.C.D
```

— Changer le nom du switch :

```
S(config)# hostname NOUVEAU-NOM
```

3.9.2 Adressage (routeur)

Pour un routeur la syntaxe est la même que dans la section précédente, à ceci près qu'on précise une interface ethernet plutôt qu'un VLAN.

— Attribuer au routeur une IP **A.B.C.D** avec le masque **W.X.Y.Z** à son interface **I** :

```
R(config)# interface gi0/0/I
R(config-if)# ip address A.B.C.D W.X.Y.Z
R(config-if)# no shutdown
```

— Vérifier les informations IP associées à une interface **I** :

```
R> show ip interface gi0/0/I
```

3.9.3 Routage

— Fixer la passerelle par défaut :

```
S(config)# ip default-gateway A.B.C.D
```

— Ajouter une route vers le réseau A.B.C.D/M.N.O.P passant par le routeur R.S.T.U :

```
S(config)# ip route A.B.C.D M.N.O.P R.S.T.U
```

— Supprimer une route vers le réseau A.B.C.D/M.N.O.P passant par le routeur R.S.T.U :

```
S(config)# no ip route A.B.C.D M.N.O.P R.S.T.U
```

— Afficher la table de routage :

```
S> show ip route
```

3.10 ACL

Les ACL (Access Control List) sont principalement utilisées sur les routeurs comme moyen de filtrer les paquets. On les utilise aussi dans d'autres cas, par exemple pour restreindre les membres d'une communauté SNMP. Une ACL basée sur l'adresse IP consiste en une liste de règles indiquant si le paquet doit être accepté ou refusé en fonction de sa source. Une ACL peut donc être vue comme une liste d'adresses IP autorisées. Il y a 99 ACL standards, chacune ayant un numéro compris entre 1 et 99. Par défaut les ACL sont vides : elles ne contiennent aucune adresse.

— Afficher les ACL :

```
S> show access-lists
```

— Dans l'ACL n°**A**, refuser l'adresse (ou les adresses) IP **SOURCE** (voir plus bas comment écrire la **SOURCE**) :

```
S(config)# access-list A deny SOURCE
```

— Dans l'ACL n°**A**, accepter l'adresse (ou les adresses) IP **SOURCE** (voir plus bas comment écrire la **SOURCE**) :

```
S(config)# access-list A permit SOURCE
```

— Vider l'ACL n°**A** :

```
S(config)# no access-list A
```

Dans les deux instructions précédentes, **SOURCE** peut être :

— **any** ⇔ n'importe quelle IP

— **host A.B.C.D** ⇔ l'unique IP A.B.C.D

— **A.B.C.D W.X.Y.Z** ⇔ l'ensemble d'IP A.B.C.D avec un wildcard mask de W.X.Y.Z. Le wildcard mask détermine les bits qui doivent être identiques dans l'IP source et dans l'IP A.B.C.D, comme ceci :

  — Si le i^{ème} bit dans le wildcard mask est à 0 alors les bits à la i^{ème} position dans l'IP source et dans A.B.C.D doivent être égaux.

  — Si le i^{ème} bit dans le wildcard mask est à 1 alors le bit à la i^{ème} position dans l'IP source peut prendre n'importe quelle valeur.

Par exemple, une **SOURCE** fixée à 10.0.0.254 0.255.255.0 permet de désigner toutes les machines dont l'IP est de la forme 10.X.Y.254.

L'ordre d'écriture des règles d'une ACL est important. Pour trouver si une IP fait partie d'une ACL, le switch appliquera les règles une par une (selon l'ordre dans lequel elles ont été écrites) jusqu'à trouver une règle qui s'applique. Dans l'exemple ci-dessous, la deuxième règle est inutile car la première indique qu'aucune IP ne fait partie de l'ACL.

```
S(config)# access-list 1 deny any
S(config)# access-list 1 permit 1.2.3.4
```

3.11 SNMP

Un agent SNMP peut être lancé sur les switchs que nous utilisons. Il suffit de déclarer une communauté SNMP pour l'activer.

— Désactiver l'agent SNMP :

```
S(config)# no snmp-server
```

3.11.1 Communautés SNMP

— Déclarer une communauté **COMM** en lecture seule restreinte à l'ACL n°**A** :

```
S(config)# snmp-server community COMM ro A
```

L'argument **A** est facultatif. Dans ce cas, il n'y a pas de restriction sur l'utilisation de la communauté déclarée.

— Déclarer une communauté **COMM** en lecture/écriture restreinte à l'ACL n°**A** :

```
S(config)# snmp-server community COMM rw A
```

L'argument **A** est facultatif. Dans ce cas, il n'y a pas de restriction sur l'utilisation de la communauté déclarée.

3.11.2 Notifications SNMP

— Activer l'envoi de notifications vers la machine **A.B.C.D**. Les notifications envoyées sont au format **V** (1, 2c ou 3) et utilisent la communauté **COMM**.

```
S(config)# snmp-server host A.B.C.D traps version V COMM
```

— Activer l'envoi de notifications du type **TYPE-NOTIF**.

```
S(config)# snmp-server enable traps TYPE-NOTIF
```

Il y a de nombreux types de notifications. Demander de l'aide (caractère ?) après avoir tappé le mot **traps** de la commande pour visualiser tous les types de notification.

3.11.3 Vues SNMP

Une *vue* consiste en un sous-arbre de la MIB. En définissant des vues, on peut ensuite donner des droits aux communautés sur certaines parties de la MIB uniquement (p.ex., donner un droit en lecture/écriture sur la branche system uniquement). Une vue a un nom. Par défaut, une vue ne contient aucune branche de la MIB. Il faut donc lui inclure les branches de la MIB qui en font partie puis éventuellement en retrancher certaines.

— Inclure la branche **X.Y.Z** (un OID) dans la vue **V** :

```
S(config)# snmp-server view V X.Y.Z included
```

— Exclure la branche **X.Y.Z** (un OID) de la vue **V** :

```
S(config)# snmp-server view V X.Y.Z excluded
```

— Déclarer une communauté **COMM** en lecture/écriture sur la vue **V** et restreinte à l'ACL n°**A** :

```
S(config)# snmp-server community COMM view V rw A
```

L'argument **A** est facultatif. Dans ce cas, il n'y a pas de restriction sur l'utilisation de la communauté déclarée.

— Déclarer une communauté **COMM** en lecture seule sur la vue **V** et restreinte à l'ACL n°**A** :

```
S(config)# snmp-server community COMM view V ro A
```

L'argument **A** est facultatif. Dans ce cas, il n'y a pas de restriction sur l'utilisation de la communauté déclarée.

3.12 Syslog

Syslog est un service permettant de centraliser des événements de journal sur une station de supervision.

— Activer l'envoi d'événements syslog vers l'hôte **A.B.C.D** ·

```
S(config)# logging A.B.C.D
```

— Fixer le niveau maximal des notifications envoyées à **N** ($\in [0,7]$) :

```
S(config)# logging trap N
```

3.13 NAT

Les routeurs CISCO intégrent la fonction de translation d'adresse (NAT). On procède généralement en suivant les étapes suivantes pour la mettre en place :

1. Spécifier l'interface interne (généralement celle connectée à un réseau privé) et l'interface externe (généralement celle connectée au réseau public).
2. Définir une ACL (voir 3.10) contenant les IP du réseau interne.
3. Activer la translation d'adresses pour les IP de cette ACL afin de masquer les adresses du réseau interne dans les paquets circulant sur le réseau externe.
4. Faire de la redirection de port (ou *port forwarding*), si l'on souhaite que des serveurs du réseau interne puissent être contactés depuis le réseau externe.

Voici le détail des commandes :

— Spécifier que l'interface interne est **I** :

```
R(config)# interface gi 0/0/I
R(config-if)# ip nat inside
```

et que l'interface externe est **E** :

```
R(config)# interface gi 0/0/E
R(config-if)# ip nat outside
```

— Activer la translation d'adresses sur tous les paquets en provenance de l'ACL **A** et routés vers l'interface **E** :

```
R(config)# ip nat inside source list A interface gi 0/0/E overload
```

L'exécution de la commande précédente a deux conséquences :

— quand un paquet ayant pour source une IP de l'ACL **A** est routé vers l'interface **E**, le routeur modifie l'IP source par l'IP associée à l'interface **E** et rajoute une ligne dans sa table de translation NAT ;

— et quand un paquet est routé dans le sens inverse, l'IP destination est modifiée par l'adresse IP de l'ACL **A** trouvée dans la table de translation NAT.

Le mot-clé **overload** permet de faire de la surcharge, c'est-à-dire d'associer plusieurs IP (celles de l'ACL) à une seule (celle associée à l'interface **E**).

— Afficher la table de translation NAT :

```
R> show ip nat translations
```

— Supprimer les entrées dynamiques de la table de translation NAT (i.e., celles rajoutées par le routeur quand un paquet sort du réseau interne) :

```
R# clear ip nat translation *
```

— Rediriger les paquets arrivant sur l'interface **E** et destinés au port **PORTEXT** vers **A.B.C.D:PORTINT** (port forwarding) :

```
R(config)# ip nat inside source static PROT A.B.C.D PORTINT interface gi 0/0/E PORTEXT
```

PROT peut être **udp** ou **tcp**. Cette fonctionnalité peut être utilisée pour rendre un service accessible depuis l'extérieur du réseau interne. Par exemple, si un serveur web s'exécute dans le réseau interne sur 10.1.0.100 et écoute sur le port 80, on pourra exécuter la commande ci-dessous afin de rediriger tous les paquets arrivant sur l'interface 1 et destinés au port TCP/80 vers le serveur web s'exécutant sur 10.1.0.100 :

```
R(config)# ip nat inside source static tcp 10.1.0.100 80 interface gi 0/0/1 80
```