Configuration d'un commutateur ou d'un routeur CISCO

Ce document n'est pas un tutoriel complet de configuration d'un commutateur ou d'un routeur Cisco, mais plutôt un recueil de commandes utilisées fréquemment pour commencer à configurer un commutateur ou un routeur. La validité des commandes présentées varient en fonction du type de commutateur et de la version de l'IOS du matériel.

Connexion en série

Pour la première configuration d'un commutateur, la seule façon de le configurer est par l'intermédiaire d'une liaison Serie RS232 connectée au port console du commutateur.

La plupart des commutateurs ont une connexion série RS232 configurée par défaut en :

- 9600 bauds,
- 8 bits de données,
- pas de parité,
- 1 bit de stop,
- pas de contrôle de flux.

Sous windows, on peut utiliser l'hyper terminal, sous linux on peut utiliser un émulateur comme kermit. Voici quelques commandes relatives à kermit :

```
set modem type none     ; on utilise une connexion directe (sans modem)
set line /dev/ttyS0     ; Specify device name
set carrier-watch off   ; If DTR CD are not cross-connected
set escape-character ^A ; changer le caractere d'echapement
set flow none           ; If you can't use RTS/CTS
set speed 57600         ; Or other desired speed
set parity none         ; (or "mark" or "space", if necessary)
set stop-bits 1         ; (rarely necessary)
show escape
connect                 ; Enter Connect (terminal) state
```

Exemple pour se connecter avec le minimum de commandes :

```
$ kermit
C-Kermit 8.0.209, 17 Mar 2003, for Red Hat Linux 8.0
 Copyright (C) 1985, 2003,
  Trustees of Columbia University in the City of New York.
Type ? or HELP for help.
C-Kermit>set modem type none
C-Kermit>set line /dev/ttyS0
C-Kermit>set carrier-watch off
C-Kermit>set escape-character ^A
C-Kermit>connect
```

Dans l'exemple ci-dessus, la sortie de la connexion de kermit se fera par la combinaison de touches :

```
Ctrl-A
q
```

Tant que le commutateur n'est pas configuré, le seul moyen de le configurer est par l'intermédiaire du port console (RS232). Au démarrage, il est possible de rentrer dans le setup. À l'aide de plusieurs questions, le setup permet de rentrer les paramètres de base et les adresses IP. Si on ne rentre pas dans le setup, il est alors possible de le configurer en ligne de commande.

Identification

2 modes d'identification :

- Mode utilisateur (**user**) : le mot de passe est demandé lors de la connexion via telnet/ssh ou à la console. Dans ce mode, nous avons accès à un sous ensemble des commandes : uniquement certaines commandes de visualisation d'informations
- Mode administrateur (**privileged** ou **enable**) : C'est dans ce mode que l'on pourra configurer le commutateur.

Visualisation de la configuration en mode utilisateur :

Liste des commandes disponibles. Attention elle est différente suivant le mode d'identification. Le ? Est aussi utilisable pour connaître les arguments d'une commande :

```
> ?
```

On peut saisir le début de la commande, elle peut être complétée (complétion) comme le bash en utilisant la touche 'TAB'.

Visualisation du modèle de commutateur, de la version de l'IOS et des numéros de série :

```
> sh version
```

Visualisation des switches de la stack (si plusieurs switches sont stackés) :

```
> sh switch
```

Visualisation de l'allocation de la mémoire aux objets de l'IOS :

```
> sh memory
```

Pour surveiller la charge du routeur (table des processus de l'IOS)

```
> sh processes
```

Visualisation de la mémoire utilisée et disponible :

```
> sh processes memory
```

Visualisation d'un résumé des statistiques des queues de toutes les interfaces :

```
> sh interfaces summary
```

Visualisation d'un résumé de l'état de toutes les interfaces (connecté/non connecté, vlan, vitesse, type) :

```
> sh interfaces status
```

Visualisation du détail d'une interface (ici la GigabitEthernet 1/0/1) :

```
> sh interface Gigabitethernet 1/0/1
```

Passer en mode privilégié afin de modifier la configuration :

```
> enable
```

Un mot de passe « enable » est éventuellement demandé. Le prompt '>' devient '#' pour indiquer que l'on est en mode privilégié.

Visualiser l'état des interfaces :

```
# show int
```

visualiser la table des routes IP :

```
# sh ip route
```

Visualisation de la table arp :

```
# sh ip arp
```

Compte les trames à destination du routeur (et non toutes celles qui passent) :

```
# sh ip traffic
```

Interrogation des logs des ACL :

```
# show ip accounting access-violations
```

Réinitialisation des compteurs de l'accounting :

```
# clear ip accounting
```

Affichage de la table des adresses ethernet (dynamic, static, vlan...) :

```
#sh mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0024.98ec.6c04    DYNAMIC     Fa0/20
  10    0024.98ec.6c04    DYNAMIC     Fa0/20
  12    001b.63b6.6ece    DYNAMIC     Fa0/20
Total Mac Addresses for this criterion: 3
```

Effacer la table d'adresses mac :

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

Affichage de la table ARP :

```
#sh arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.10.4           -   0011.bb49.6080  ARPA   Vlan10
```

Effacer la table ARP :

```
#clear arp-cache
```

Modification de la configuration

Pour entrer dans l'éditeur de configuration et la modifier (en mode privilégié) :

```
# configure  terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

pour sortir de l'éditeur de configuration : utiliser la combinaison de touches CTRL-Z :

```
switch(config)# ^Z
switch#
```

Assigne un mot de passe encrypté à enable :

```
# enable secret <password>
```

pour visualiser la configuration en mémoire non volatile (celle en mémoire RAM, pas celle sur la mémoire flash) :

```
# write terminal
# show running-config
```

pour visualiser la configuration de démarrage (celle stockée sur la mémoire flash) :

```
# show startup-config
```

Modifier le nom de l'équipement réseau :

```
# hostname <hostname>
```

Pour sauvegarder la nouvelle configuration en mémoire non volatile (FLASH) :

```
# write memory
# copy running-config startup-config
```

Toutes les commandes peuvent être rentrées sous forme complète ou sous forme abrégée :

```
switch# write memory
switch# wr mem
switch# conf t
switch(config)# int fa 0/1
switch(config)# interface fastethernet 0/1
switch(config-if)# ^Z
```

Configuration des interfaces

Le mode de chaque interface physique :

`switchport mode access` : dans ce mode, on indique au commutateur que le port n'est pas un uplink et que l'on connectera une ou plusieurs machines dans le même VLAN

`switchport access vlan N` : dans ce mode, on indique au commutateur que le port n'est pas un uplink et que l'on connectera une ou plusieurs machines dans le VLAN numéro N

`switchport mode trunk` : dans ce mode, on indique au commutateur que le port va transporter tous les VLAN (sauf prunning)

`switchport trunk encap dot1q` : dans ce mode, on indique au commutateur que le protocole de trunking va utiliser le protocole normalisé 802.1q

`switchport trunk encap isl` : dans ce mode, on indique au commutateur que le

protocole de trunking va utiliser le protocole ISL (propriétaire Cisco) : non conseillé

Exemple de configuration de l'interface GigabitEthernet 0/23 dans le vlan 13 et GigabitEthernet 0/24 dans en mode trunk :

```
switch# conf t
switch(config)#interface GigabitEthernet0/23
switch(config-if)# switchport access vlan 13
switch(config-if)# switchport mode access
switch(config-if)# spanning-tree portfast
switch(config-if)#interface GigabitEthernet0/24
switch(config-if)# switchport trunk encapsulation dot1q
switch(config-if)# switchport mode trunk
switch(config-if)# ^Z
```

Attribuer une adresse IP à une interface : **ip address <address> <mask> :**

```
switch# conf t
switch(config)# int vlan1
switch(config-if)# ip address 192.168.1.20 255.255.255.0
switch(config-if)# ^Z
```

Active ou désactive une interface (ou un VLAN) : **shutdown** ou **no shutdown :**

```
switch# conf t
switch(config)# int fastethernet 0/1
switch(config-if)# no shutdown
switch(config-if)#^Z
```

Active ou désactive l'auto-croisement d'une interface : **mdix**

```
switch# conf t
switch(config)# int fastethernet 0/1
switch(config-if)# mdix auto
switch(config-if)# ^Z
```

Commandes par interfaces (sous-commandes) : Elles s'adressent à une partie du commutateur. À 1 ou plusieurs interfaces :

```
#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
routeur(config)#interface fastethernet 1/0/1
routeur(config-if)# ip address x.y.z.y 255.255.255.0
routeur(config)#interface fastethernet 1/0/2
routeur(config-if)# ip address x.y.z.z 255.255.255.0
routeur(config)#interface range fastethernet 1/0/2 - 24
routeur(config-if)# switchport mode acess
routeur(config-if)# switchport acess vlan 3
routeur(config-if)# ^Z
```

Accès au système de fichier de la mémoire FLASH

Affichage du contenu de la mémoire flash :

```
sw4#dir flash:
Directory of flash:/
    2  -rwx         270   Jan 1 1970 00:01:36 +00:00  env_vars
    3  -rwx     3036020   Mar 1 1993 00:03:25 +00:00  c2950-i6q4l2-mz.121-
20.EA1a.bin
    4  -rwx        1368   Mar 1 1993 00:14:18 +00:00  config.text
    5  -rwx        1996   Mar 1 1993 00:31:25 +00:00  vlan.dat
```

```
    6  -rwx           5   Mar 1 1993 00:14:18 +00:00  private-config.text
    7  -rwx         110   Mar 1 1993 00:01:45 +00:00  info
    8  drwx        2688   Mar 1 1993 00:07:05 +00:00  html
   90  -rwx         110   Mar 1 1993 00:07:50 +00:00  info.ver

7741440 bytes total (1606144 bytes free)
sw4#
```

Suppression de la base des VLAN (ATTENTION : efface la liste des VLAN reçus ou enregistrés) :

```
#delete flash:vlan.dat
Delete filename [vlan.dat]?
Delete flash:vlan.dat? [confirm]
```

Affichage d'un fichier contenu sur la flash :

```
#more flash:info.ver
image_name: c2950-i6q4l2-mz.121-20.EA1a.bin
image_file_size: 3041280
image_min_dram: 16
tar_file_size_k: 3133
```

Suppression de la configuration

Pour effacer la configuration de mémoire non volatile (FLASH) : **ATTENTION : Cela efface la configuration qui est chargée au démarrage, si on re-démarre le commutateur après cette commande, il perd toute sa configuration.**

```
# erase startup-config
```

Supprimer le fichier (*vlan.dat*) contenant la liste des VLAN enregistrés localement sur le switch :

```
# delete flash:vlan.dat
```

Rédémarrage du routeur (reboot)

```
# reload
```

Une autre méthode consiste à ré-initialiser le switch à sa configuration par défaut d'usine, pour cela :

- couper l'alimentation du switch
- tout en maintenant appuyé le bouton **MODE**, brancher l'alimentation du switch,
- le switch démarre en mode « maintenance »

Taper :

```
Switch: flash_init
```

Le switch donne alors accès au contenu de la flash, il est alors possible de modifier le contenu de la flash, le fichier contenant la configuration courante s'appelle *config.text*, il est possible de le renommer ou de le supprimer :

```
Switch: rename flash:config.text flash:config.ancien
Switch: delete flash:config.text
```

Idem pour supprimer la liste des VLAN enregistrés localement sur le switch (fichier *vlan.dat*) il est possible de le renommer ou de le supprimer :

```
Switch: delete flash:vlan.dat
```

Puis redémarrer le switch

```
Switch: reset
```

Sauvegarde de la configuration du routeur sur un serveur

On peut sauvegarder la configuration du routeur sur un serveur du réseau via TFTP, RCP ou FTP ou HTTP... (en fonction de ce que supporte le commutateur). La commande copy demande 2 paramètres : le premier est la source et le second est la destination.

La référence à un fichier sur un serveur TFTP se fait par la syntaxe suivante (spécifier l'adresse IP du serveur et le nom du fichier) :

```
tftp://serveur/fichier
```

La référence à un fichier sur un serveur FTP se fait par la syntaxe suivante (spécifier l'utilisateur à la place de 'user', le mot de passe de l'utilisateur dans 'password', puis l'adresse IP du serveur et le nom du fichier) :

```
ftp://user:password@serveur/fichier
```

Il faut d'abord mettre en place un serveur TFTP. On suppose que /tftpboot est le répertoire de chargement du serveur tftp :

```
# copy system:/running-config tftp://192.168.1.14/test
Address or name of remote host [192.168.1.14]?
Destination filename [test]?
!!!!!!!!
35761 bytes copied in 1.157 secs (30908 bytes/sec)
#
```

La configuration est sauvegardée dans /tftpboot/test du serveur tftp.

Chargement de la configuration à partir d'un serveur TFTP

De même on peut charger la configuration via un serveur TFTP ou ftp. Cette méthode présente l'avantage de pouvoir écrire tranquillement sa configuration via un éditeur de texte et la charger quand on veut.

```
# copy tftp://192.168.1.14/test system:/running-config
# copy ftp://user:password@192.168.1.14/test system:/running-config
# copy system:/running-config system:/startup-config
# copy ftp://user:password@192.168.1.14/test system:/startup-config
```

On peut donc charger un fichier de configuration et soit le rendre actif immédiatement (destination = system:/running-config) ou au prochain démarrage (destination = system:/startup-config)
Commandes de tests et de visualisation de l'état du routeur :

Les VLAN

Les VLANs peuvent être crées localement sur chaque commutateur (appelé : *mode VTP transparent*) ou alors être propagés à l'aide d'un protocole de propagation des VLAN (appelé : *mode VTP client/serveur*).

Configuration des VLAN en mode transparent :

```
#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# vtp mode transparent
switch(config-if)# ^Z
```

Création d'un VLAN localement :

```
#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# vlan 3
switch(config)# name vlan-trois
switch(config)# ^Z
```

Affectation de l'interface fastEthernet1/0/1 au VLAN 3 :

```
#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
routeur(config)#interface fastethernet 1/0/1
routeur(config-if)# switchport mode acess
routeur(config-if)# switchport acess vlan 3
routeur(config-if)# ^Z
```

Activation du VLAN 3 et affectation d'une adresse IP à ce switch dans ce VLAN :

```
#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)#interface Vlan3
switch(config-if)# no shutdown
switch(config-if)# ip address x.y.z.z 255.255.255.0
switch(config-if)# ^Z
```

Cisco dispose d'un protocole d'échange de la liste des VLAN configurés dans un commutateur, c'est VTP (Vlan Trunking Protocol). C'est un mode client serveur. Un commutateur doit être configuré en mode serveur, les autres commutateurs du réseau local doivent être configurés en mode client :

sur le commutateur « serveur VTP » :

```
vlan database
  vtp domain <domaine>
  vtp server
  vlan 2
  name vlan-deux
  exit
```

sur le commutateur « client VTP » :

```
vlan database
  vtp domain <domaine>
  vtp client
  exit
```

On peut visualiser si un client a bien reçu des VLAN par le protocole VTP :

```
# sh vtp status
```

Une fois que le commutateur Client a reçu la liste des VLAN, la commande suivante affiche la liste des VLAN reçus ou configurés localement :

```
# show vlan brief
```

Routage

Commandes pour ajouter une route statique :
Route statique : `ip route reseau masque <passerelle>`
Route par défaut : `ip route 0.0.0.0 0.0.0.0 <passerelle>`

```
routeur#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
routeur(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.13
routeur(config)# ip route 192.168.85.4 255.255.255.252 192.168.84.5
routeur(config)# ip route 192.168.85.16 255.255.255.240 192.168.84.5
routeur(config)# ip route 192.168.85.32 255.255.255.240 192.168.84.4
routeur(config)# ^Z
```

Le commutateur de niveau 2 ne peut pas faire de routage (statique ou dynamique). Pour pouvoir réaliser du routage entre VLAN ou entre réseaux IP différents, le commutateur doit supporter le routage (donc de niveau 3 de la couche ISO).

Les commutateurs de la série 2xxx (2950, 2960...) sont de niveau 2 (sans routage).
Les commutateurs de la série 3xxx (3750...) sont de niveau 3 (avec routage).

En fonction des licences disponibles sur le commutateur, il sera capable (ou non) de supporter les protocoles de routage dynamique interne (RIP, OSPF...) ou externe (BGP...).

Par exemple, cette commande peut uniquement être prise en compte sur les commutateurs de niveau 3 :

```
routeur#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
routeur(config)# ip routing
routeur(config)# ^Z
```

Filtrage, access-lists

Les access-lists cisco sont répertoriées par numéro :

```
routeur(config)#access-list ?
  <1-99>       IP standard access list
  <100-199>    IP extended access list
  <1100-1199>  Extended 48-bit MAC address access list
  <200-299>    Protocol type-code access list
  <700-799>    48-bit MAC address access list
```

- IP standard access list : Ne permet d'utiliser que les adresses source pour identifier les

paquets
- IP extended access list : Permet d'identifier un paquet par les adresses IP, protocoles et ports source et destination
- Protocol type-code access list : Filtrage sur le protocole
- 48-bit MAC address access list : Filtrage en fonction de l'adresse MAC

Filtres sur ICMP

Filtrer icmp est toujours problématique. C'est un protocole à la fois très utile mais aussi potentiellement dangereux. De nombreuse attaques sont basées dessus (« smurf » et autres joyeusetées). Parmis ses différentes fonctions on peut trouver : fragmentation (type 3, message du type "destination unreachable"), PATH MTU Discovery (type 4), drop des paquets, traceroute, ping, controle de flux.

Pour ceux qui veulent filtrer le protocol icmp , voici au moins ce qu'il faut autoriser :

```
access-list 102 permit icmp any any unreachable parameter-problem source-quench
time-exceeded ttl-exceeded packet-too-big administratively-prohibited
access-list 102 deny icmp any any
```

Autrement, pour ceux qui autorisent ping , on peut limiter les excès éventuels en ajoutant ceci sur l'interface d'entrée :

```
rate-limit input access-group 2000 744000 10000 10000 conform-action transmit
exceedaction drop
```

avec :

```
access-list 2000 permit icmp any any echo-reply
access-list 2000 permit icmp any any echo
```