Paramétrage Cisco

Pierre Jaquet

Février 2015

Table des matières

1. Installer un routeur **5**

Le système d'exploitation IOS 5
La connexion au LAN 6
La connexion au WAN 7
La configuration initiale 7
Les outils de configuration 9
Casser le mot de passe 9
Les fichiers de configuration 10
L'aide en ligne 11
Le setup mode 12
Le user mode 13
Le privileged mode 14
Le mode de configuration globale 16
Le mode de configuration d'interface 18
Le mode de configuration de sous-interface 19
Le mode de configuration de ligne 19
Le mode de configuration du routage 20
Les modes en résumé 21
Questions 24

2. Configurer un routeur **25**

Rappel de quelques points importants 26
ROM, flash, NVRAM et RAM 26
Les trois modes de fonctionnement 28
Le registre de configuration 28
Mettre à jour le système d'exploitation 29
IFS 30
Questions 31

3. Configurer un switch **35**

Les switches non configurables 35
La configuration initiale 35
Le VLAN 1 et l'adresse IP 38
Le serveur DHCP 38
La configuration des modules 39

Questions 41

4. Le routage 43

L'acheminement des paquets 44
Exemple 45
Le routage statique 47
Le routage par défaut 48
Le routage dynamique 48
Les protocoles à vecteur de distance 49
Questions 54

5. Le fonctionnement des IGP 57

Critères de comparaison 58
RIP 58
IGRP 58
EIGRP 59
OSPF 59
IS-IS 59
Questions 61

6. Les ACL 63

Le paramétrage 63
Exemple 64
Questions 66

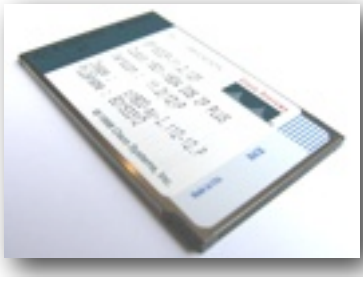
7. Les VLAN 67

Le trunking 68
Le protocole VTP 68
Questions 70

1. Installer un routeur

Lorsqu'on reçoit un routeur, le carton d'emballage contient en général quatre ou cinq éléments : le routeur, l'alimentation, le câble de console (*console cable*), la documentation et éventuellement un adaptateur.

Le **système d'exploitation** se trouve sur une carte préinstallée dans le routeur. Ci-dessous, voici à gauche la Flash PC Card du Cisco 1601 insérée dans son emplacement. Au milieu et à droite, on voit les deux faces de la carte.



Il manque à ces quatre ou éléments un câble Ethernet pour relier le routeur à un switch ainsi que le câble du WAN. D'autres éléments sont optionnels, comme une carte WAN supplémentaire.

Remarque : ce chapitre porte sur le cas du Cisco 1601, mais il est largement généralisable aux autres routeurs, aux switches et aux pare-feu de la marque. Il est souvent aussi valable pour d'autres équipements parce que le langage utilisé pour paramétrer les équipements Cisco a été copié par d'autres fabricants.

Le système d'exploitation IOS

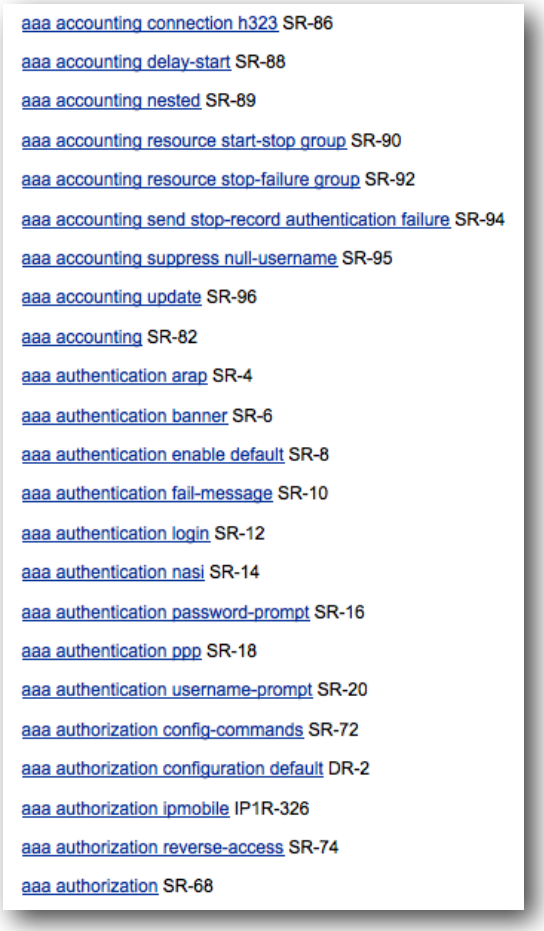
Les routeurs, les pare-feu et la majorité des switches Cisco contiennent un système d'exploitation appelé **IOS** (*Internetwork Operating System*).

Les routeurs à haute disponibilité contiennent, eux, un autre système appelé **IOS XR**. En interne, il est complètement différent de la lignée IOS tout court, mais son interface avec l'utilisateur est la même.

Le noyau d'IOS XR est QNX (http://www.qnx.com), un système d'exploitation Unix qui fonctionne en **temps réel** ou RT (*real time*), ce qui veut dire qu'il sait travailler avec des délais impératifs, ce qui est très important pour des équipements de réseaux.

Techniquement, c'est un système à micro-noyau qui introduit notamment un mécanisme de protection des zones de mémoire utilisées par chaque processus et un ordonnancement à réquisition (*premptive scheduling*). Il gère le ***multithreading*** (le traitement par processus et sous-processus).

En pratique, IOS XR n'est toutefois qu'une version particulière d'IOS ; ci-desssous, j'emploie donc « IOS » pour désigner IOS et IOS XR.


aaa accounting connection h323 SR-86
aaa accounting delay-start SR-88
aaa accounting nested SR-89
aaa accounting resource start-stop group SR-90
aaa accounting resource stop-failure group SR-92
aaa accounting send stop-record authentication failure SR-94
aaa accounting suppress null-username SR-95
aaa accounting update SR-96
aaa accounting SR-82
aaa authentication arap SR-4
aaa authentication banner SR-6
aaa authentication enable default SR-8
aaa authentication fail-message SR-10
aaa authentication login SR-12
aaa authentication nasi SR-14
aaa authentication password-prompt SR-16
aaa authentication ppp SR-18
aaa authentication username-prompt SR-20
aaa authorization config-commands SR-72
aaa authorization configuration default DR-2
aaa authorization ipmobile IP1R-326
aaa authorization reverse-access SR-74
aaa authorization SR-68

Le système regroupe des fonctions de plusieurs types :

- télécommunications,
- routage (détermination des itinéraires),
- connexions interréseaux,
- commutation (*switching* ou *forwarding*),
- qualité de service ou QoS (*Quality of Service*),
- disponibilité (*availability*),
- sécurité (ACLs, VPN, AAA, pare-feu, etc.).

Il comprend un langage de commandes très complet. L'image ci-contre ne montre que celles qui commencent par *aaa_a*[1].

Seize niveaux de privilèges sont associés aux commandes, numérotés de 0 à 15.

Physiquement, IOS est stocké en **mémoire flash**.

On le met à jour depuis le site de Cisco. Ce service est payant ; les entreprises y souscrivent, mais les micro-entreprises et les particuliers s'en passent le plus souvent.

L'interface d'IOS avec l'utilisateur est un ***shell*** ou **CLI** (*command language interface*), c'est à dire une interface par commandes. On peut y accéder de trois façons :

- par la console (c'est le seul moyen d'accès pour la configuration initiale) ;
- par telnet, via le réseau (que ce soit le LAN ou le WAN) ;
- par modem, via une liaison RAS ou VPN.

La connexion au LAN

La première étape consiste en deux choses :

1° brancher le câble d'alimentation du secteur (*power cord*) au routeur, cela sans mettre le routeur sous tension ;

2° brancher le câble Ethernet qui relie le routeur au switch. Sur le routeur, c'est la prise appelée Ethernet 0 (voir l'image ci-contre), qu'on abrège **E 0** (c'est un zéro, pas une lettre O).

Cette abréviation est importante. On va voir qu'on peut l'utiliser dans les commandes.

[1] http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_command_summary09186a008020b3d8.html).

La connexion au WAN

La deuxième étape consiste à relier le routeur au WAN, ce qui est logique puisque le rôle d'un routeur consiste précisément à servir d'interface entre un LAN et un WAN.

Dans le cas du Cisco 1601, on passe par une prise sérielle de type DB-60. La liaison s'appelle Serial 0, abrégée **S 0** (c'est un zéro).

Les protocoles les plus utilisés sont *x*DSL, mais il arrive qu'on emploie d'autres interfaces vers le WAN : ATM, FDDI, ADSL, SDSL, SDH-Sonet, Long Haul Ethernet, etc.

La configuration initiale

Une fois le routeur connecté au LAN et au WAN, la troisième étape consiste à le paramétrer. Un routeur est généralement livré sans aucune configuration. Le fournisseur ou l'administrateur du réseau doit donc le paramétrer à partir de zéro.

Le problème est qu'un routeur qui n'est pas configuré n'a pas d'adresse IP, ce qui veut dire qu'on ne peut pas l'atteindre par le réseau. On est donc obligé de passer par un canal direct, un canal qui n'a pas besoin de connaître le protocole IP.

Installation

1° Connecter le routeur au LAN (prise e 0) ;
2° connecter le routeur au WAN (prise s 0) ;
3° Mettre le routeur sous tension ;
4° Vérifier les connexions (LEDs) ;
5° Configurer le routeur au moyen d'une émulation de terminal.

Sous tous les systèmes d'exploitation (Unix, Linux, Windows, etc.), ce canal existe sous la forme d'une **émulation de terminal**. Il s'agit souvent du terminal ASCII appelé **VT-100** ou de l'émulation **ANSI**. Cette connexion relie directement le routeur à un PC. Elle ne passe pas par le réseau.

Il existe de nombreux programmes d'émulation de terminal, par exemple Minicom pour Linux, iTerm2 pour Mac OS et PuTTY pour Windows.

Il faut un câble spécial appelé ***Cisco console cable***, avec une fiche DB-9 femelle pour le PC et une fiche RJ-45 pour le routeur (image ci-dessous à gauche). On le branche dans la prise ***Console*** du routeur (image ci-dessous au milieu) et la prise sérielle mâle du PC (image ci-dessous à droite).

Dans le logiciel, on crée une nouvelle connexion et on lui donne un nom (par exemple, « Routeur Cisco »). Ensuite, on indique le pays, l'indicatif et le port (par exemple *COM1*).

Cela fait, il faut spécifier les caractéristiques de la liaison :

1° débit : 9'600 bits par seconde,
2° 8 bits de données,
3° pas de parité,
4° 1 bit d'arrêt (*stop bit*),
5° pas de contrôle de flux.

Il faut rester dans le programme d'émulation au moment de mettre le routeur sous tension. Si on en est ressorti, ça ne fonctionne pas.

Lorsque tout cela est fait, on ne ressort pas du programme d'émulation (une diode verte à côté de la prise *Console* montre que la connexion est active) et on met le routeur sous tension.

Il s'initialise, et, si tout se passe normalement, la connexion entre le PC et le routeur s'active toute seule après quelques secondes.

Une suite de lignes s'affiche alors et le routeur affiche la question :

```
Would you like to enter the initial configuration dialog [yes/no] ?
```

On répond *n* et l'invite suivante s'affiche dans le programme d'émulation de terminal :

```
Router>
```

En cas de problème, essayer avec un autre port, par exemple COM2.

Remarque : comme certains fabricants semblent ignorer que les prises sérielles sont le seul moyen de liaison avec la plupart des équipements de réseau, ils n'en dotent plus leurs PC. Il est donc difficile de gérer ces équipements avec certains PC récents. Pour résoudre ce problème, il y a plusieurs solutions :

1° installer dans le PC une carte d'extension munie de prises sérielles DB-9 ;
2° utiliser un câble USB spécial, c'est-à-dire un câble Console muni d'une fiche USB type A mâle à un e extrémité et d'une fiche RJ-45 à l'autre ;
3° utiliser un adaptateur de type *Serial to USB* (mais la correspondance entre les fils n'est pas toujours correcte).

L'invite qui apparaît est celle du ***user mode*** d'IOS. Ce mode ne permet que de voir la configuration, pas de la modifier. Pour cela, il faut passer en ***privileged mode***, qu'on appelle aussi ***enable mode***. On s'y rend en utilisant la commande *enable*, qui peut s'abréger ***en***.

L'invite par défaut devient :

```
Router#
```

Dans IOS, on peut abréger une commande aussi loin que la confusion avec une autre commande n'est pas possible.

Pour revenir en user mode, c'est la commande *disable*. On peut l'abréger *disabl*, *disab* ou ***disa*** (c'est la commande *disconnect* qui fait qu'on ne peut pas abréger au-delà de *disa*).

```
Router> enable
Router# disable
Router> en
Router# disa
```

Ensemble, les deux modes forment l'***EXEC***. C'est l'environnement d'exécution d'IOS, la session qui permet de gérer le routeur.

Remarque : la plupart des commandes servent à opérer des changements dans la configuration, et on fait assez facilement des erreurs qui nous mènent à un point où plus rien ne fonctionne. Il est donc vivement conseillé de garder une copie de sauvegarde de la dernière configuration saine, ce qui ne pose pas de problèmes puisqu'il s'agit d'un fichier de type texte. On peut même imprimer la configuration sur papier et garder dans un classeur un journal des configurations successives que l'on a utilisées.

Les outils de configuration

Il existe trois manières de configurer le routeur.

Premièrement, on peut travailler par commandes (le shell est le même sur l'ensemble des produits IOS). Il existe trois manières d'accéder au langage de commande :

1° si le routeur n'a pas encore d'adresse IP, on utilise un programme d'émulation de terminal et on passe par le câble Console comme on l'a vu plus haut ;

2° s'il est configuré et qu'au moins un de ses ports est doté d'une adresse IP, on peut passer par le réseau, avec deux possibilités :

- travailler en interactif avec **telnet** ou **ssh** (*secure shell*) ;
- travailler en batch avec **tftp** (*trivial file transfer protocol*), qui est un *ftp* simplifié ;

3° si on passe par un modem, on utilise la prise auxiliaire *AUX*.

Éviter telnet si on désire une connexion sécurisée. Tout est transmis en clair, y compris le mot de passe.

Ssh, lui, transmet des données chiffrées. Il offre des fonctions similaires à *rexec* et *rsh*.

Tftp permet de transférer des fichiers via un réseau. Il sert notamment à copier un fichier de configuration d'un ordinateur vers un routeur ou réciproquement.

Deuxièmement, on peut utiliser une interface graphique (généralement HTTP), mais les moyens d'administration offerts sont inférieurs en qualité et en quantité à ceux de l'interface de commande.

Troisièmemement, on peut passer par un logiciel d'administration de réseau comme Cisco Network Assistant (*http://www.cisco.com/c/en/us/products/cloud-systems-management/network-assistant/index.html*), mais ce canal offre également des moyens d'administration inférieurs à ceux de l'interface de commande..

Casser le mot de passe

Si le routeur est sécurisé par un mot de passe, on peut réinitialiser le système en effectuant un reset à partir de l'adresse physique de boot. Pour cela, on relie le routeur à un PC avec un câble Console et on lance une session d'émulation de terminal. Une fois la connexion établie, on met le routeur hors tension, puis on le rallume. On envoie ensuite un Break au routeur dans les soixante secondes (il y a un *timeout*).

Attention, la combinaison de touches Break diffère d'un clavier à l'autre, mais aussi d'un programme et d'un système d'exploitation à l'autre ; c'est par exemple *Ctrl-Break*.

Cette opération met le routeur en mode *rommon* (*ROM Monitor*). Voici ce que le système affiche :

```
monitor : command "boot" aborted due to user interrupt
Rommon 1 >
```

À cette invite, on tape *confreg 0x2142* pour lancer la routine de démarrage du routeur située à l'adresse 2142h de la mémoire. Le moniteur répond par un message demandant de réinitialiser le routeur, et on tape *reset* :

```
Rommon1> confreg 0x2142
You must reset or power cycle for new config to take effect
Rommon2> reset
```

Le système redémarre alors et une suite de messages s'affiche. Si le moniteur pose des questions, on répond par non :

```
[yes/no]: n
```

À la fin, on presse une ou deux fois sur *Enter* et on entre dans le système. Une invite apparaît sous la forme suivante (une autre suite de caractères est possible, mais le dernier est toujours le signe ">") :

```
Router>
```

La forme générale de cette invite est *hostname>*, où hostname est le nom de l'équipement.

Les fichiers de configuration

On sauvegarde la configuration sur une machine distante en employant *tftp* :

```
Router# copy run tftp
Address or name of remote host []? 192.168.1.20
Destination filename [router-confg]? 001-confg
!!
338 bytes copied in 9.542 secs (35 bytes/sec)
```

Dans cet exemple, on sauvegarde la configuration qui est active en ce moment (la ***running configuration***, abrégée *run*), mais on peut aussi copier la dernière configuration enregistrée (la ***startup configuration***) en utilisant la commande *copy startup-config* au lieu de la commande *copy running-config*. Les données entre crochets sont les données par défaut. Par exemple, le système propose d'appeler *hostname-confg* le fichier texte qui contient la configuration.

On affiche la configuration active au moyen de la commande :

```
show running-config
```

qu'on peut abréger :

```
sh run
```

et la configuration enregistrée au moyen de :

```
show startup-config
```

qu'on peut abréger :

```
sh start
```

La *running configuration* est stockée en RAM et la *startup configuration* en NVRAM (*Non-Volatile RAM*). La *running configuration* est donc perdue en cas de panne de courant ou de réinitialisation (*reload*).

On remplace la configuration en cours par la dernière configuration enregistrée au moyen de la commande *copy startup-config running-config* :

```
Router# copy start run
```

On peut aussi effacer purement et simplement la configuration du système au moyen de la commande *erase startup-config*, ce qui équivaut à réactiver la configuration d'origine :

```
Router# erase start
```
⟵ *ne pas utiliser cette commande sur un routeur en production !*

Au boot qui suit cette opération, le routeur se lance automatiquement en ***setup mode***.

L'aide en ligne

IOS comprend une commande d'aide : le point d'interrogation (le choix de cette commande ultra-courte s'explique par le fait que taper « *?* » va plus vite que taper « *help*↵ »).

Il existe deux cas :

1° Aide sur la syntaxe des commandes (*command syntax help*). Il y a deux possibilités :

- on peut taper "?" pour obtenir la liste des commandes disponibles dans le mode dans lequel on se trouve, accompagnées d'une brève explication sur leur fonction ;
- on peut taper le nom d'une commande suivi d'un espace et du signe "?" pour avoir la liste des «sous-commandes» correspondantes. Exemple :

```
Router# configure ?
memory       Configure from NV memory
network      Configure from a TFTP network host
overwrite-network  Overwrite NV memory from TFTP network host
terminal     Configure from the terminal
Router# configure
```
⟵ *l'invite ajoute le nom tapé à la demande d'aide*

2° Aide sur le nom des commandes (*word help*). On peut taper quelques lettres suivies immédiatement (sans espace) de « ? ». On obtient ainsi la liste des commandes qui commencent par ces lettres. Exemple :

```
Router> di?
disable  disconnect
Router> di
```

IOS est l'exemple même d'un système d'exploitation fait pour les informaticiens : comme les administrateurs de systèmes et réseaux sont paresseux, le système leur mâche le travail : si on a demandé de l'aide sur les commandes qui commencent par *di*, c'est sûrement qu'on veut utiliser une commande qui commence par ces lettres. L'invite suivante propose donc déjà *di*.

Une fonction de contrôle de la syntaxe des commandes est comprise dans IOS. L'erreur est signalée par un accent circonflexe en-dessous de l'endroit fautif :

```
Router# config tern
                  ^
% Invalid input detected at '^' marker.
```

Il existe plusieurs *hot keys*, notamment les suivantes :

Tab	termine un nom de commande (par exemple, *en* suivi de *Tab* affiche *enable*) ;
Ctrl-A	déplace le curseur au début de la ligne ;
Ctrl-C	interrompt la commande en cours et rend la main au CLI ;
Ctrl-U	efface la ligne en cours ;
Ctrl-W	efface le mot en cours ;
Ctrl-Z	termine le mode de configuration en cours et revient au mode privilégié.

Remarque : au début, on se trompe souvent dans les commandes, soit parce qu'on se trouve dans le mauvais mode (la majorité des commandes sont spécifiques à un mode), soit parce qu'une commande valide sur un équipement peut être illégale sur un autre.

Le setup mode

Le ***setup mode*** sert à effectuer le paramétrage du système, si on le souhaite. On peut l'utiliser ou pas, au choix : toutes les commandes de ce mode existent aussi dans la CLI.

Quand on lance le *setup mode*, on obtient un dialogue du genre de ce qui suit :

```
--- System Configuration Dialog ---
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
Would you like to enter the initial configuration dialog? [yes]. y
First, would you like to see the current interface summary? [yes]. y
Any interface listed with OK? value 'NO' does not have a valid conf.
Interface    IP-Address    OK?    Method       Status Protocol
Ethernet0    unassigned    NO     unset  up     up
Serial0      unassigned    NO     unset  down   down
Configuring global parameters:
Enter host name [Router] ↵                    ← on accepte le nom par défaut du routeur
The enable secret is a one-way cryptographic secret used
instead of the enable password when it exists.
Enter enable secret : cisco                   ← mot de passe à éviter en production
The enable password is used when there is no enable secret
```

```
and when using older software and some boot images.
```

```
Enter enable password : admin
```
⟵ *à éviter en production*

```
Enter virtual terminal password : admin
```
⟵ *à éviter en production*

Il va de soi que des mots de passe comme *cisco* ou *admin* sont à éviter absolument en production. Par sécurité, on conseille un mélange de caractères et une suite de mots plutôt plutôt qu'un seul.

Pour le nom du routeur, la proposition par défaut, «*Router*», a été gardée. En entreprise, il faut choisir un nom parlant. Par exemple, on peut appeler « *Paris* » le routeur du site de Paris.

Le système demande ensuite si on veut configurer le logiciel d'administration de réseau SNMP (*Simple Network Management Protocol*), l'adressage IP et les interfaces présentes sur le routeur. Dans un premier temps, on peut répondre *non* à toutes les questions.

IOS crée alors le fichier de configuration et l'affiche avant de demander :

```
Use this configuration ? [yes/no]. y
```

Après quelques lignes de messages, le système affiche l'invite du mode privilegié.

Le user mode

Le ***user mode*** donne accès à un sous-ensemble des commandes du mode privilégié (l'ensemble de ces commandes étant les ***Exec commands***).

L'aide en ligne affiche la liste des commandes disponibles :

```
Router> ?

Exec commands:
access-enable    Create a temporary Access-List entry
clear            Reset functions
connect          Open a terminal connection
disable          Turn off privileged commands
disconnect       Disconnect an existing network connection
enable           Turn on privileged commands
exit             Exit from the EXEC
help             Description of the interactive help system
lock             Lock the terminal
login            Log in as a particular user
logout           Exit from the EXEC
mrinfo           Request neighbor and version information from a multicast router
mstat            Show statistics after multiple multicast traceroutes
mtrace           Trace reverse multicast path from destination to source
name-connection  Name an existing network connection
pad              Open a X.29 PAD connection
ping             Send echo messages
ppp              Start IETF Point-to-Point Protocol (PPP)
resume           Resume an active network connection
rlogin           Open an rlogin connection
show             Show running system information
```

```
slip            Start Serial-line IP (SLIP)
systat          Display information about terminal lines
telnet          Open a telnet connection
terminal        Set terminal line parameters
traceroute      Trace route to destination
tunnel          Open a tunnel connection
where           List active connections
x3              Set X.3 parameters on PAD
```

Dans ce mode, on travaille en niveau 1 de privilège, comme on peut le vérifier en tapant la commande *show privilege* :

```
Router> sh pri
Current privilege level is 1
Router>
```

Cela veut dire qu'on peut voir mais pas modifier la configuration. Sachant, en outre, que plusieurs commandes sont en fait des versions appauvries de celles du mode privilégié, le *user mode* ne présente aucun intérêt.

Le privileged mode

On accède au mode privilégié depuis le user mode au moyen de la commande *enable*, abrégée *en*.

Du point de vue du niveau de privilège, la situation s'améliore notablement :

```
Router> en
Password : cisco                    ← en réalité, le mot de passe ne s'affiche pas
Router# sh pri                      ← le signe « # » désigne le mode privilégié
Current privilege level is 15       ← c'est le niveau par défaut ; on peut le modifier
```

Voici les commandes disponibles :

```
Router# ?
Exec commands:
access-enable    Create a temporary Access-List entry
access-template  Create a temporary Access-List entry
bfe              For manual emergency modes setting
clear            Reset functions
clock            Manage the system clock
configure        Enter configuration mode
connect          Open a terminal connection
copy             Copy configuration or image data
debug            Debugging functions (see also 'undebug')
disable          Turn off privileged commands
disconnect       Disconnect an existing network connection
enable           Turn on privileged commands
erase            Erase flash or configuration memory
exit             Exit from the EXEC
```

```
help             Description of the interactive help system
lock             Lock the terminal
login            Log in as a particular user
logout           Exit from the EXEC
mrinfo           Request neighbor and version information from a multicast router
mstat            Show statistics after multiple multicast traceroutes
mtrace           Trace reverse multicast path from destination to source
name-connection  Name an existing network connection
no               Disable debugging functions
pad              Open a X.29 PAD connection
ping             Send echo messages
ppp              Start IETF Point-to-Point Protocol (PPP)
reload           Halt and perform a cold restart
resume           Resume an active network connection
rlogin           Open an rlogin connection
rsh              Execute a remote command
send             Send a message to other tty lines
setup            Run the SETUP command facility
show             Show running system information
slip             Start Serial-line IP (SLIP)
start-chat       Start a chat-script on a line
systat           Display information about terminal lines
telnet           Open a telnet connection
terminal         Set terminal line parameters
test             Test subsystems, memory, and interfaces
traceroute       Trace route to destination
tunnel           Open a tunnel connection
undebug          Disable debugging functions (see also 'debug')
verify           Verify checksum of a Flash file
where            List active connections
x3               Set X.3 parameters on PAD
Router#
```

On emploie beaucoup les commandes ***show startup-config*** et ***show running-config***, qu'on abrège respectivement *sh start* et *sh run*.

Attention, ces deux commandes affichent les mots de passe, à l'exception de ceux qui ont été chiffrés. Elles constituent donc l'outil de base des pirates.

La commande ***show interfaces*** (*sh in*) est aussi très utilisée, car elle permet de voir quelles interfaces sont opérationnelles et lesquels connaissent éventuellement un problème. Quatre états sont possibles :

- statut *up*, protocole *up* : la ligne est opérationnelle ;
- statut *administratively down*, protocole *down* : la ligne est désactivée (*disabled*) ;
- statut *up*, protocole *down* : il y a un problème de connexion ;
- statut *down*, protocole *down* : il y a un problème d'interface.

Les commandes ***show ip protocol*** et ***show ip route*** sont aussi très pratiques.

Dans le *enable mode*, on trouve également d'importantes commandes de communication interplates-formes : *rlogin*, *rsh*, *telnet*, *ping*, *ssh* et *traceroute*.

Comme le *user mode*, ce mode porte surtout sur la gestion des connexions avec le réseau, l'analyse de la configuration et les tests, mais il contient plusieurs commandes qui n'existent pas dans le *user mode* : *access-template*, *bfe*, *clock*, *configure*, *copy*, *debug*, *erase*, *no*, *reload*, *rsh*, *ssh*, *start-chat*, *test*, *undebug* et *verify*. Il s'agit pour la plupart de commandes qui permettent de modifier des paramètres et pas seulement de les consulter.

Le système d'exploitation maintient un journal des dix dernières commandes On peut l'allonger jusqu'à 256 commandes au moyen de la commande *terminal history size* :

```
Router# term histo size 100
```

ou, en abrégeant au maximum :

```
Router# ter hi s 100
```

Comme Unix et Linux, IOS est peu causant. L'absence de réponse de sa part signifie que tout s'est passé normalement.

On enregistre les modifications qu'on a demandées au moyen de la commande *copy running-config start-config* :

```
Router# copy run start
Building configuration...
[OK]
Router#
```

Remarque : la commande *copy running-config start-config* équivaut au bouton *Enregistrer* de Word ou Excel. Pour éviter de perdre tout ce qu'on a fait durant une session de configuration, il faut penser à taper cette commande de temps à autre.

Attention, tous les changements sont immédiats, ce qui veut dire que, si on désactive l'interface par laquelle on est en train de dialoguer avec le routeur, on ne peut plus l'utiliser. Si on veut la réactiver, il faudra passer par une autre interface, par exemple la console (qu'on ne peut heureusement pas désactiver). Le même problème se pose si on configure un filtre : on se retrouve facilement dans la situation où on ne peut plus rien faire parce que le routeur refuse nos paquets. Le cas échéant, on peut réparer les choses en passant par la console.

Le mode de configuration globale

On accède au **mode de configuration globale** depuis le *enable mode* au moyen de la commande *configure terminal*. Cette commande est un raccourci pour *configure from terminal*, c'est-à-dire configurer depuis un terminal.

```
Router> en
Password : cisco
Router# conf t                    ← pour passer du enable mode au mode de configuration globale
```

```
Router(config)# ?                    ← « (config)# » désigne le mode de configuration globale
Exec commands:
aaa                           Authentication, Authorization and Accounting
access-list                   Add an access list entry
alias                         Create command alias
arp                           Set a static ARP entry
async-bootp                   Modify system bootp parameters
banner                        Define a login banner
boot                          Modify system boot parameters
bridge                        Bridge Group.
buffers                       Adjust system buffer pool parameters
busy-message                  Display message when connection to host fails
cdp                           Global CDP configuration subcommands
chat-script                   Define a modem chat script
clock                         Configure time-of-day clock
config-register               Define the configuration register
default                       Set a command to its defaults
default-value                 Default character-bits values
dialer-list                   Create a dialer list entry
dnsix-dmdp                    Provide DMDP service for DNSIX
dnsix-nat                     Provide DNSIX service for audit trails
downward-compatible-config    Generate a configuration compatible with older software
enable                        Modify enable password parameters
end                           Exit from configure mode
exception                     Exception handling
exit                          Exit from configure mode
frame-relay                   Global frame relay configuration commands
help                          Description of the interactive help system
hostname                      Set system's network name
interface                     Select an interface to configure
ip                            Global IP configuration subcommands
key                           Key management
line                          Configure a terminal line
logging                       Modify message logging facilities
login-string                  Define a host-specific login string
map-class                     Configure static map class
map-list                      Configure static map list
menu                          Define a user-interface menu
modemcap                      Modem Capabilities database
multilink                     PPP multilink global configuration
netbios                       NETBIOS access control filtering
no                            Negate a command or set its defaults
ntp                           Configure NTP
partition                     Partition device
priority-list                 Build a priority list
privilege                     Command privilege parameters
prompt                        Set system's prompt
queue-list                    Build a custom queue list
```

```
resume-string               Define a host-specific resume string
rlogin                      Rlogin configuration commands
rmon                        Remote Monitoring
route-map                   Create route-map or enter route-map command mode
router                      Enable a routing process
scheduler                   Scheduler parameters
service                     Modify use of network based services
snmp-server                 Modify SNMP parameters
sntp                        Configure SNTP
stackmaker                  Specify stack name and add its member
state-machine               Define a TCP dispatch state machine
tacacs-server               Modify TACACS query parameters
terminal-queue              Terminal queue commands
tftp-server                 Provide TFTP service for netload requests
username                    Establish User Name Authentication
vpdn                        Virtual Private Dialup Network
x25                         X.25 Level 3
x29                         X29 commands
```

Ce mode donne accès à une vaste palette de commandes de configuration du routeur dans son ensemble (on parle de *system-wide configuration*). Par exemple, c'est dans ce mode qu'on peut donner un nouveau nom au routeur au moyen de la commande ***hostname***, abrégée *host* :

```
Router(config)# host Rochefort
```

```
Rochefort(config)# host RouteurRCH      ⟵ seuls les lettres non accentuées et les chiffres sont acceptés
```

```
RouteurRCH(config)#
```

Attention, on lit souvent que le nom du routeur ne joue pas de rôle dans le système, que c'est juste une dénomination à l'usage des êtres humains. C'est inexact, IOS l'utilise dans ses algorithmes de sécurité.

On ressort du mode de configuration globale au moyen de la combinaison de touches ***Ctrl-Z*** ou de la commande ***exit***.

Le mode de configuration d'interface

On accède au mode de configuration d'interface depuis le mode de configuration globale. Il faut spécifier une interface donnée, par exemple la *serial 0* :

```
Router> enable
```

```
Password : cisco
```

```
Router# conf t
```

```
Router(config)# interface                    ⟵ commande « interface » tout court refusée
```

```
% Incomplete command.
```

```
Router(config)# in s 0                       ⟵ commande « interface serial 0 » tout court acceptée
```

```
Router(config-if)#                           ⟵ « (config-if)# » désigne le mode de configuration d'interface
```

Les interfaces disponibles sont une dizaine :

```
Router(config)# interface ?
BVI               Bridge-Group Virtual Interface
Dialer            Dialer interface
Ethernet          IEEE 802.3
Lex               Lex interface
Loopback          Loopback interface
Null              Null interface
Serial            Serial interface
Tunnel            Tunnel interface
Virtual-Template  Virtual-Template interface
Vlan              Catalyst 5000 Vlan
```

On ressort de ce mode au moyen de la combinaison de touches *Ctrl-Z* si on veut revenir directement au *enable mode* ou de la commande *exit* si on veut revenir au mode de configuration globale (en d'autres termes, un *Ctrl-Z* correspond à deux *exit* successifs).

Voici, par exemple, la configuration de l'adresse IP 192.168.1.100 avec un masque de sous-réseau */24* (ce qui implique l'absence de sous-réseau pour autant que le réseau lui-même soit de type */24* et non, par exemple, */16*) :

```
Router> enable
Password : cisco
Router# conf t                          ← commande « configure from terminal »
Router(config)# in e 0                  ← commande « interface ethernet 0 »
Router(config-if)# ip ad 192.168.1.100 255.255.255.0
Router(config-if)#  Ctrl-Z
Router(config)#  Ctrl-Z
Router#
```

Dans le monde Cisco, ***if*** signifie interface.

Le mode de configuration de sous-interface

On accède au mode de configuration de sous-interface depuis le mode de configuration d'interface. L'invite est *« (config-subif)# »*. Ce mode permet de créer des interfaces logiques, c'est-à-dire des interfaces gérées par IOS à l'intérieur des interfaces réelles. On peut ainsi utiliser un seul canal physique pour faire passer plusieurs canaux virtuels.

Le mode de configuration de ligne

On accède au mode de configuration de ligne depuis le mode de configuration globale :

```
Router(config)# line tty 1
Router(config-line)#
```

Le Cisco 1601 accepte les paramètres suivants :

```
Router(config)# line ?
<0-6>      First Line number
console    Primary terminal line
tty        Terminal controller
vty        Virtual terminal
```

Ce mode sert notamment à configurer les mots de passe d'accès au système.

On en ressort au moyen de la combinaison de touches *Ctrl-Z*.

Le mode de configuration du routage

On accède au mode de configuration du routage depuis le mode de configuration globale :

```
Router(config)# router nom_protocole
Router(config-router)#
```

Attention, le terme de *router* est trompeur. Il ne désigne pas le routeur, mais le protocole de routage. Avec le Cisco 1601, les paramètres possibles sont les suivants :

```
Router(config)# router ?
bgp        Border Gateway Protocol (BGP)
egp        Exterior Gateway Protocol (EGP)
eigrp      Enhanced Interior Gateway Routing Protocol (EIGRP)
igrp       Interior Gateway Routing Protocol (EIGRP)
isis       ISO IS-IS
iso-igrp   IGRP for OSI networks
mobile     Mobile routes
odr        On demand Stub routes
ospf       Open Shortest Path First (OSPF)
rip        Routing Information Protocol (RIP)
static     Static routes
```

Tous les protocoles répandus sont gérés par le routeur, aussi bien en interne (RIP, IGRP, EIGRP et OSPF) qu'en externe (BGP).

Voici un exemple :

```
Router> enable
Password : cisco
Router# config terminal
Router(config)# router rip
Router(config-router)# network 1ère_adresse_IP
Router(config-router)# network 2ème_adresse_IP
```

etc. : on indique les adresses de tous les routeurs et switches qui doivent gérer RIP

```
Router(config-router)#  Ctrl-Z
Router(config)#  Ctrl-Z
```

```
Router#
```

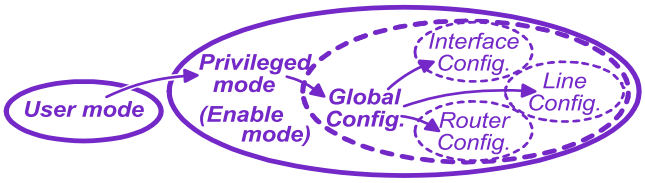
Remarque : les commandes de configuration ont généralement une forme positive et une forme négative. Par exemple, la commande *ip routing* active le routage IP et la commande *no ip routing* le désactive.

Les modes en résumé

Voici à quoi sert chaque mode :

Mode	Utilisation
Setup mode	Créer la configuration initiale (ce mode est optionnel)
User mode	Voir la configuration sans risque de la modifier par erreur
Privileged ou Enable mode	Administrer le système (le routeur, le switch ou le pare-feu)
Global configuration mode	Configurer le système
Line configuration mode	Configurer les lignes de terminal
Router configuration mode	Configurer les protocoles de routage
Interface configuration mode	Configurer les interfaces (réelles) du système
Subinterface configuration mode	Configurer les sous-interfaces (logicielles) du système

L'imbrication des modes est assez bizarre au premier abord. Il faut surtout retenir que le mode privilégié englobe cinq sous-modes (*Global configuration*, *Interface configuration*, *Line configuration* et *Router configuration*) et que, en même temps, il joue le rôle de mode intermédiaire entre le user mode et le mode de configuration globale. Graphiquement, cela donne ceci :



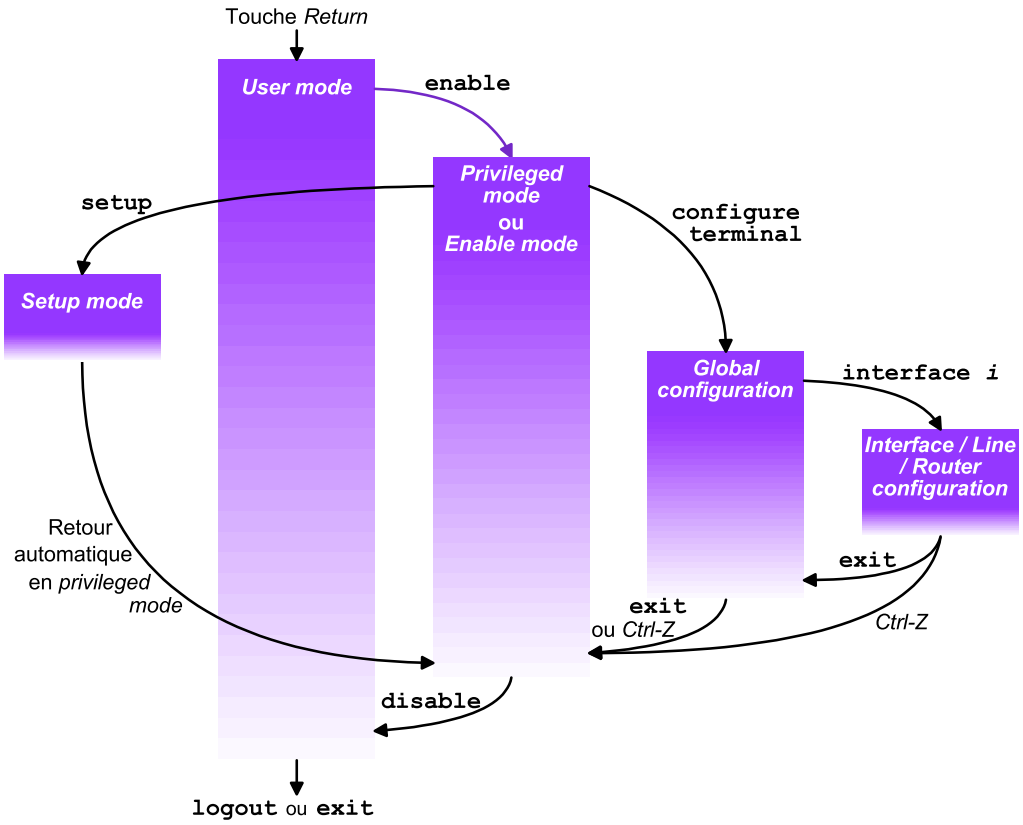
On distingue trois types de commandes :

1° les **commandes globale**s (*global commands*), c'est-à-dire celles qui se trouvent dans le enable mode ou le global configuration mode et qui ne ne mènent pas à un sous-mode ;

2° les **commandes majeures** (*major commands*), comme *configure*, qui mènent à un sous-mode ;

3° les **sous-commandes** (*subcommands*).

Voici le schéma des modes et des commandes de changement de mode :



Voici un tableau récapitulatif de l'entrée dans les huit modes et de leur invite :

Mode	Entrée dans le mode	Invite par défaut
Setup mode	Commande *setup* en mode privilégié	*Pas d'invite*
User mode	Automatique (c'est le mode de base)	Router>
Enable mode	Commande *enable* en user mode	Router#
Global configuration	Commande *configure terminal* en mode privilégié	Router(config)#
Line configuration	Commande *line nom_ligne* en mode global configuration	Router(config-line)#
Router configuration	Commande *router nom_protocole* en mode global configuration	Router(config-router)#
Interface config.	Commande *interface nom_interface* en mode global configuration	Router(config-if)#
Subinterface config.	Commande *interface nom_interface.numéro_sous-interface* en mode global configuration	Router(config-subif)#

Ce document ne donne qu'un petit aperçu du sujet : l'ensemble de la documentation Cisco sur IOS comprend plus de 100'000 pages.

Dans le cadre d'un labo, on peut faire librement des essais. À tout instant, on peut réinitialiser le routeur au moyen des deux commandes suivantes :

```
Router# erase startup-config                ⟵ pour réinitialiser le routeur
[OK]                                        (commande à éviter absolument en production...)
Routeur# reload                             ⟵ pour lancer le redémarrage à froid du routeur
```

Questions

1. Dessinez l'organigramme des étapes de l'installation physique d'un routeur Cisco 1601.
2. La prise Console des routeurs Cisco est de type RJ-45. Que se passe-t-il si vous essayez de configurer votre routeur depuis un swich, au moyen d'un câble Ethernet normal, plutôt que de passer par une liaison directe ?
3. Un câble RJ-45 est-il par définition un câble Ethernet ?
4. Dans le monde MacIntosh, il existe l'équivalent de PuTTY : iTerm.app. Sachant cela, pensez-vous que vous pouvez vous connecter à un routeur Cisco depuis un Mac ?
5. Comment peut-on accéder à un routeur Cisco qu'on vient de recevoir ?
6. Le setup mode est-il nécessaire pour configurer un routeur ?
7. Jusqu'où peut-on abréger la commande *show version* en user mode ?
8. Quels sont les caractères acceptés par la commande *hostname* ?
9. Si vous tapez *router ?* à l'invite de configuration globale, le système vous répond en vous affichant les possibilités (de *bgp* à *static*), puis il affiche l'invite suivie du mot *router*. Pourquoi ajoute-t-il ce mot ?
10. À quoi sert la commande *tftp-server* ?
11. À quoi sert la commande ci-dessous ?

    ```
    Router# sh ip ssh
    ```

12. À quoi sert la commande ci-dessous ?

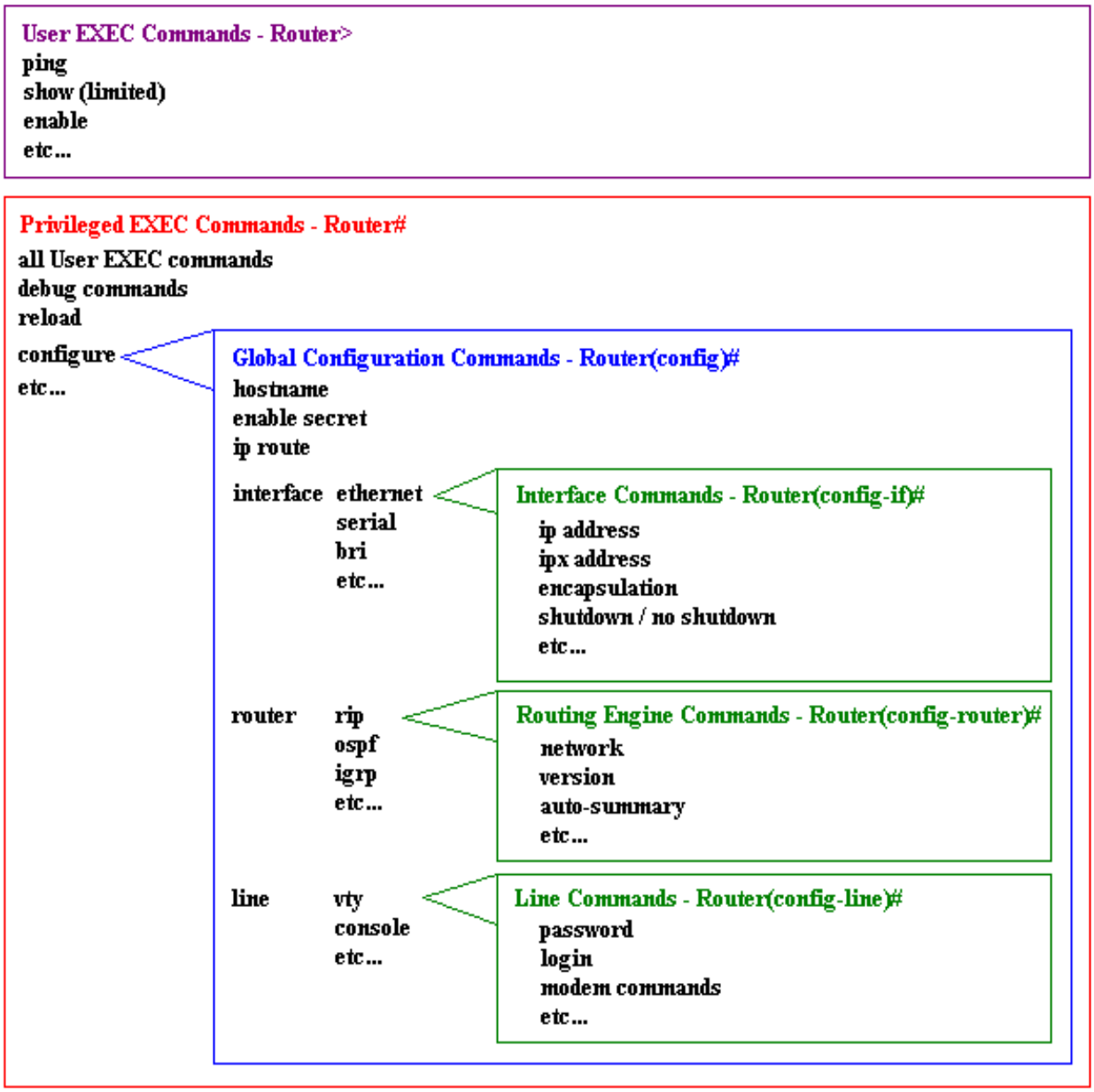
    ```
    Router(config)# crypto key generate rsa
    ```

13. Vos routeurs, vos switches et vos firewalls sont situés dans la salle des serveurs du site central de votre entreprise. Vous êtes sur un poste de travail d'une filiale et vous désirez accéder à la configuration d'un des routeurs. Par quel moyen allez-vous le faire ? Expliquez votre réponse.

2. Configurer un routeur

La configuration d'un routeur Cisco peut paraître difficile à cause de l'organisation de la CLI du système d'exploitation IOS en modes.

Voici comment Cisco présente cette imbrication (*http://www.cisco.com/warp/cpropub/45/tutorial.htm*) :



Rappel de quelques points importants

On utilise :

- le *privileged* ou *enable mode* pour la configuration de l'environnement du routeur ;
- le sous-mode *global configuration* pour la configuration du routeur en tant que tout ;
- les « sous-sous-modes » pour le paramétrage d'un élément particulier du routeur.

Chaque mode ou sous-mode a :

- son **invite** propre ;
- son **jeu de commandes** propre.

Un routeur Cisco s'achète **non configuré** (contrairement à un *switch*).

Il contient un système d'exploitation appelé **IOS** (*Internetwork Operating System*).

L'interface avec l'utilisateur est un **shell** ou **CLI** (*command language interface*) divisée en plusieurs **modes**.

À l'achat, un équipement Cisco se configure au moyen du câble de la **console** (en émulation de terminal).

Ensuite, il existe plusieurs possibilités pour le paramétrer :

- la **console** (liaison directe ne nécessitant pas une adresse IP configurée),
- **telnet** (liaison **vty** non sécurisée via le réseau ; nécessite une adresse IP configurée),
- **ssh** (liaison sécurisée *via* le réseau ; doit être configurée),
- **tftp** (liaison non sécurisée *via* le réseau ; nécessite une adresse IP configurée),
- un logiciel tiers de gestion de réseau utilisant **SNMP** (nécessite une adresse IP configurée),
- une interface web **HTTP** (pour cela, un *plug-in* java est souvent nécessaire).

Ne pas oublier de garder une **copie de sauvegarde** de la dernière configuration saine (des commandes comme *debug* ou *config-register* peuvent être mortelles).

ROM, flash, NVRAM et RAM

Un équipement Cisco comprend normalement quatre mémoires :

1° La **ROM** est non volatile, non modifiable. Elle contient le POST (Power-On Self Test), le bootstrap et le ROM Monitor (le bootstrap est le code qui pilote le démarrage du système).

2° La **mémoire flash** est non volatile, modifiable. Elle contient l'image ou les images d'IOS. Chaque image se trouve dans un fichier binaire unique autoextractible. Elle peut être mise à jour.

3° La **NVRAM** (*non-volatile RAM*) est non volatile, modifiable. Elle contient la configuration de démarrage dans un fichier binaire unique.

4° La **RAM** est volatile. Elle contient les processus d'IOS, la configuration courante, les tables de routage, les tables ARP, les buffers et les files d'attente.

La ROM joue le même rôle que le BIOS des PC. Comme le BIOS, on peut la mettre à jour en changeant la puce, mais, dans la plupart des cas, on ne la touche jamais. En cas de défaillance grave, le ROM Monitor qu'elle contient permet de restaurer le système d'exploitation.

Le fait que la mémoire flash peut contenir plusieurs images d'IOS est particulièrement utile pour les tests d'une nouvelle version du système d'exploitation. Bien entendu, ces tests se conduisent sur un routeur qui n'est pas en production.

Comme c'est la RAM qui abrite les tables de routage, sa taille doit être d'autant plus grande que le réseau est complexe. Plus la mémoire flash et la RAM sont grandes, plus on peut activer de fonctions sur le routeur.

On provoque un crash si on tente de faire fonctionner IOS dans un espace de mémoire insuffisant.

Par défaut, le système tente de lancer la procédure de démarrage depuis la première image contenue sur la mémoire flash. En cas d'insuccès, il se rabat sur la PC Card.

L'ordre de lancement peut être modifié au moyen de la commande *boot*. Exemple :

```
Router> enable                    ← pour aller dans le « enable mode » (désigné par l'invite #)
Router# config t                  ← pour aller dans le « global configuration mode »
Router# boot system slot0         ← « slot0 » désigne la première (ou la seule) PC Card
```

La mémoire flash se trouve sur une barrette de mémoire de taille fixe située sur la carte-mère.

Sur certains routeurs de haut de gamme, le bootstrap se trouve en mémoire flash et non en ROM.

Le lancement du système s'effectue de la manière suivante :

1° le POST contrôle que les conditions de départ d'un bon fonctionnement sont remplies ;

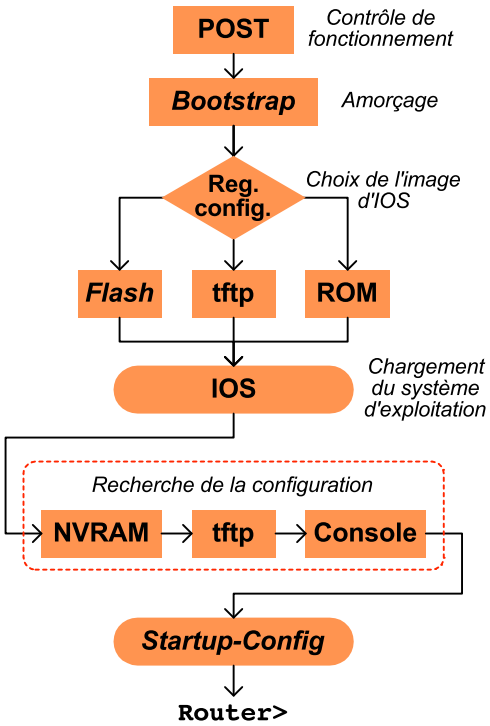
2° le code du bootstrap se charge depuis la ROM ;

3° le registre de configuration (*conf-reg*) est lu ;

4° la *startup-config* est lue ;

5° les directives de chargement de la *startup-config* sont suivies si elle en contient ;

6° si elle n'en contient pas, le *ROM Monitor* se charge et le *setup mode* se lance.



Le lancement normal d'IOS suit la séquence *flash* → *tftp* → *ROM* : le routeur recherche d'abord l'image du système d'exploitation en flash, puis, s'il ne la trouve pas à cet endroit, il la demande sur tftp, puis, à défaut, il la charge depuis la ROM.

Le fonctionnement du système est géré par un ou plusieurs processeurs centraux. Certains routeurs de haut de gamme comprennent des emplacements pour enficher des processeurs centraux supplémentaires. Ces modèles sont très configurables. On peut les équiper de

modules et de cartes d'extension qui leur permettent de gérer toutes sortes de paquets : Ethernet, ATM, E1, E2, E3, SDH-Sonet, MPLS, Ethernet Long Haul, etc.

Les trois modes de fonctionnement

On peut distinguer trois modes de fonctionnement du système :

- le **ROM Monitor**, auquel on accède par la console, avec les fonctions suivantes :
  - piloter l'amorçage du système,
  - fournir les fonctions fondamentales du système (noyau),
  - fournir des diagnostics,
  - permettre la restauration du système en cas de défaillance grave ;
- le **Boot ROM**, qui fournit les fonctions de base du système ; comme le moniteur, il est stocké sur une puce et ne peut donc être mis à jour, sauf en remplaçant la puce ;
- **IOS**, qui constitue le système d'exploitation du routeur, du switch ou du firewall.

Le registre de configuration

Un élément essentiel pour le lancement du système est le registre de configuration (***configuration-register***, abrégé ***conf-reg***). C'est un registre d'une taille de 16 bits qui contient notamment l'adresse de l'image d'IOS à utiliser (la *boot image*), l'adresse de la routine de configuration du système, le bit *oui/non* du mode de diagnostic du matériel (*enable/disable hardware diagnostic mode*) et le débit de la liaison avec la console en bits par seconde.

Le registre de configuration se trouve en NVRAM. On affiche son contenu en mode privilégié au moyen des commandes show version ou show hardware.

On peut modifier le registre en redémarrant le système et en envoyant un Break. On utilise ensuite la commande :

`confreg` ***`valeurhexadécimale`***

pour dire au routeur comment il doit se lancer. La valeur hexadécimale peut être :

1° 0x2002 : lancement normal, mais le Break reste actif après la procédure de boot. On utilise ce paramètre pour le debugging. Il est dangereux sur un routeur en production.

2° 0x2010 : le routeur recherche la configuration en NVRAM puis il se lance en mode rommon (ROM Monitor). On utilise aussi ce paramètre pour le debugging.

3° 0x2101 : lancement à partir de l'image d'IOS en ROM, ou, à défaut, à partir de la flash. Ce paramètre n'est pas disponible sur les routeurs d'accès (séries 800 et 1800).

4° 0x2102 : lancement normal, à partir des valeurs par défaut du registre de configuration.

5° 0x2141 : lancement à partir de l'image en ROM, en ignorant la startup-configuration qui se trouve en NVRAM. Ce paramètre permet de restaurer IOS s'il est corrompu et de casser les mots de passe.

6° 0x2142 : lancement à partir de l'image en flash, en ignorant la startup-configuration qui se trouve en NVRAM

7° 0x8000 : lance le routeur en mode rommon pour effectuer des diagnostics au moyen de la commande priv. Ce mode est dangereux et n'est pas disponible sur les routeurs d'accès.

Mettre à jour le système d'exploitation

Le système d'exploitation IOS pilote le fonctionnement de tous les routeurs et les *firewalls* Cisco, ainsi que la grande majorité des *switches*. Il regroupe des fonctions de télécommunications, de routage, de connexions interréseaux et de commutation. Actuellement, les principales versions d'IOS utilisées en production sont la 10, la 11, la 12 et IOS-XR.

Jusqu'ici, les versions d'IOS se sont suivies tous les trois ans. Grosso modo, une version vit sa première année en LD (*Limited Deployment*), puis elle passe en GD (*General Deployment*), puis il y a les ED (*Early Deployment*), l'équivalent des *service packs* de Microsoft.

La mise à jour d'IOS s'effectue depuis le site de Cisco (*http://www.cisco.com/*). Elle nécessite d'abord de contrôler quelle est la bonne image à installer au moyen du planificateur de mise à jour (*upgrade planner*) qui se trouve sur le site, car l'installation d'une mauvaise image peut aboutir à un *crash* du système.

Le chargement de la nouvelle version du système d'exploitation s'effectue au moyen de la commande *copy*. Une quinzaine d'options se présente :

```
Router# copy ?
/erase Erase destination file system.
flash:          Copy from flash: file system
ftp:            Copy from ftp: file system
null:           Copy from null: file system
nvram:          Copy from nvram: file system
pram:           Copy from pram: file system
rcp:            Copy from rcp: file system
running-config  Copy from current system configuration
slot0:          Copy from slot0: file system
slot1:          Copy from slot1: file system
startup-config  Copy from startup configuration
system:         Copy from system: file system
tftp:           Copy from tftp: file system   ←  l'option la plus utilisée
xmodem:         Copy from xmodem: file system
ymodem:         Copy from ymodem: file system
```

On emploie le plus souvent *copy tftp* ou *copy ftp*. La procédure est la suivante :

1° on place le fichier image sur le serveur TFTP ou FTP ;

2° on tape *copy tftp* ou *copy ftp* ;

3° on répond aux questions sur l'adresse IP du serveur et sur le nom du fichier source et du fichier de destination ;

4° on emploie la commande *boot system* pour indiquer l'image à charger.

Il reste à redémarrer le système au moyen de la commande *reload* du mode privilégié.

Si la NVRAM est vide au moment où le routeur démarre, le système demande si on souhaite entrer en mode *setup*.

IFS

Le système de fichiers d'IOS est **IFS** (*Internetworking File System*), avec un jeu de commandes de gestion de fichiers : *dir* pour l'affichage du contenu de la mémoire spécifiée, *cd* pour le déplacement, *pwd* pour l'affichage de l'endroit où l'on se trouve, *more* pour l'affichage du contenu d'un fichier, *rename* pour le changement de nom d'un fichier, *delete* pour l'effacement (logique) d'un fichier, *undelete* pour la récupération d'un fichier, *squeeze* pour la suppression (physique) d'un fichier et *format* pour l'effacement d'une mémoire.

Les mémoires sont désignées par le nom suivi de deux points. Exemple :

```
flash:
```

Questions

1. Où un routeur va-t-il chercher les instructions de *boot* du système ? [2]

   - ☐ ROM
   - ☐ RAM
   - ☐ NVRAM
   - ☐ Flash

2. Quelle est la séquence par défaut pour le chargement du système d'exploitation ?

   - ☐ NVRAM ⟶ flash ⟶ ROM
   - ☐ Flash ⟶ tftp ⟶ Console
   - ☐ NVRAM ⟶ tftp ⟶ Console
   - ☐ Flash ⟶ ROM ⟶ tftp
   - ☐ Flash ⟶ tftp ⟶ ROM

3. Quelle est la séquence par défaut pour le chargement du fichier de configuration ?

   - ☐ NVRAM ⟶ tftp ⟶ flash
   - ☐ NVRAM ⟶ tftp ⟶ Console
   - ☐ Flash ⟶ ROM ⟶ tftp
   - ☐ Tftp ⟶ flash ⟶ ROM
   - ☐ Flash ⟶ tftp ⟶ ROM

4. Laquelle des commandes suivantes s'utilise pour modifier l'ordre dans lequel le routeur recherche les informations sur le bootstrap ?

   - ☐ config-image
   - ☐ config-system
   - ☐ config-register
   - ☐ config-bootfield
   - ☐ config system bootrstrap

5. Parmi les sources suivantes, deux peuvent être configurées comme sources d'une image d'IOS. Lesquelles ?

   - ☐ Serveur TFTP
   - ☐ Serveur HTTP
   - ☐ Serveur telnet
   - ☐ Mémoire flash
   - ☐ Mémoire NVRAM

[2] Les questions à choix multiples sont extraites d'examens de certification Cisco.

6. Parmi les éléments suivants, deux sont nécessaires pour le fonctionnement de base d'un routeur. Lesquels ?
   - ☐ Serveur TFTP
   - ☐ Serveur telnet
   - ☐ Fichier de configuration
   - ☐ Ensemble des registres de configuration
   - ☐ Fichier du système d'exploitation
   - ☐ Table ARP
   - ☐ Table DNS
7. Parmi les propositions suivantes, laquelle décrit la mémoire flash ?
   - ☐ Elle fournit un espace de travail
   - ☐ Elle stocke une image d'IOS entièrement fonctionnelle
   - ☐ Elle stocke le fichier de configuration startup-config
   - ☐ Elle lance le code utilisé pour le boot du routeur
8. Quelle est la partie du registre de configuration qui indique où se trouve IOS ?
   - ☐ Le bootstrap
   - ☐ Le boot field
   - ☐ L'IOS locator
   - ☐ L'IOS pointer
   - ☐ Le system image locator
9. Quelle est la commande du moniteur en ROM qui lance l'image d'IOS qui se trouve sur la mémoire flash ?
   - ☐ config-register 0x2102
   - ☐ boot flash:filename
   - ☐ xmodem:filename
   - ☐ boot system flash:filename
   - ☐ reload system flash:
10. Dans le registre de configuration, que veut dire 0x2102 ?
    - ☐ Il dit au routeur de charger IOS depuis la NVRAM
    - ☐ il dit au routeur de charger IOS depuis la mémoire flash
    - ☐ Il dit au routeur de « sauter » la configuration en NVRAM
    - ☐ Il dit au routeur de suivre les instructions de la startup-config
11. Laquelle des commandes suivantes déplace les fichiers de configuration entre la RAM, la NVRAM et un serveur TFTP ?
    - ☐ copy
    - ☐ move
    - ☐ bootp

- ☐ mv

12. Laquelle des commandes suivantes copie une image IOS depuis un serveur TFTP vers un routeur ?
    - ☐ Router# copy tftp flash
    - ☐ Router# copy flash tftp
    - ☐ Router(config)# copy tftp flash
    - ☐ Router(config)# copy flash tftp
13. Si le registre de configuration dit au routeur de chercher les informations de configuration en NVRAM mais qu'il n'en existe pas à cet endroit, d'où le routeur va-t-il chercher à lancer le système d'exploitation ?
    - ☐ RAM
    - ☐ ROM
    - ☐ Flash
    - ☐ EPROM
    - ☐ ROMMON
14. Une image IOS peut être copiée depuis un serveur TFTP sur la mémoire *flash*. Trois des propositions suivantes sont vraies concernant ce processus. Lesquelles ?
    - ☐ Des points d'exclamation sont affichés pendant que l'image se charge en flash
    - ☐ La commande *copy tftp flash:filename ip_address* est utilisée
    - ☐ Une fois l'image copiée, elle est vérifiée
    - ☐ Des lettres *E* sont affichées pendant que l'image est effacée de la flash
    - ☐ La flash doit être vidée avec la commande *erase* avant de commencer le processus
15. Si des commandes de *boot* ont été configurées, qu'est-ce qui pourrait arriver si un routeur ne parvient pas à trouver l'image d'IOS dans les sources principales ? Laquelle des propositions suivantes est juste ?
    - ☐ Le routeur charge un sous-ensemble d'IOS depuis la ROM
    - ☐ Le routeur effectue deux tentatives de chargement avant de renoncer
    - ☐ Le routeur charge la dernière version valide d'IOS depuis la NVRAM
    - ☐ Le routeur demande à l'administrateur de charger une image valide
16. Quelle est la suite de commandes pour donner le nom ESIG au routeur ?
17. Quelle est la suite de commandes pour configurer l'horloge du routeur ?
18. Pourquoi est-il important de configurer l'horloge du routeur ?
19. Quelle est la suite de commandes pour que le message bannière (*banner*) « Bonjour » s'affiche à chaque accès au routeur ?
20. Quelle est la suite de commandes pour donner l'adresse 192.168.1.1 /24 au routeur ?
21. Quelle est la suite de commandes pour donner le mot de passe *esig* à l'accès à la console ?
22. Quelle est la suite de commandes pour donner le mot de passe *admin* à l'accès au *enable mode* ?

23. Que faut-il faire pour vérifier l'adresse IP du routeur au moyen d'un ping ?
24. Qu'est-ce qu'on obtient si on tape *show startup-config* en *enable mode* ?
25. Si on tape *show running-config* en enable mode, comment est-ce que les mots de passe s'affichent ? Qu'est-ce que vous en concluez ?
26. Quelle est la suite de commandes qui permet d'effacer la configuration stockée en NVRAM ?
27. Si le port Ethernet = d'un routeur a l'adresse 172.38.130.1 /20, combien de nœuds peut-on placer sur ce sous-réseau ?
28. Quelle est la différence entre une interface qui est *administratively down* et une qui est *down* tout court ?
29. Quelle est la suite de commandes qui permet de se connecter à un routeur *B* depuis un routeur *A* au moyen d'une session telnet ?

3. Configurer un switch

À l'exception des switches de niveau 2 d'entrée de gamme, les switches sont configurables. La configuration d'un switch dépend de son type : niveaux 1 et 2, niveaux 1, 2 et 3 ou MLS (multiniveau).

La tâche essentielle des switches est de transmettre les **trames** (*frames*), c'est-à-dire les données qu'il reçoit au **niveau 2** OSI.

La liste des nœuds se trouve dans la **table de commutation** (*switching table*) du switch. Elle contient la liste des adresses MAC et des ports. Cette table se remplit automatiquement durant les premières minutes de fonctionnement du switch : chaque fois que le switch reçoit une trame, il note le port par lequel elle est arrivée ainsi que l'adresse MAC indiquée dans l'en-tête de la trame.

Pour gagner en vitesse, le switch peut commencer à transmettre une trame avant qu'elle lui soit entièrement parvenue. Pour cela, il regarde l'adresse MAC du destinataire et lance immédiatement l'envoi. Ce mécanisme s'appelle ***cut-through*** ou ***fast-forward***.

Une autre solution est le mode ***store-and-forward***, dans lequel le switch attend d'avoir reçu toute la trame avant de la transmettre. C'est plus lent mais plus sûr que le cut-through.

Un switch peut être configuré pour choisir le cut-through par défaut mais pour retomber en store-and-forward si le taux d'erreur est trop élevé. Ce mode s'appelle l'***adaptive-switching***.

La correspondance entre les adresses MAC et les adresses IP version 4 est assurée automatiquement par **ARP** (*Address Resolution Protocol*).

Si on utilise la version 6 d'IP, c'est le protocole **NDP** (*Neighbour Discovery Protocol*) qui s'utilise à la place d'ARP.

Les switches non configurables

Un switch de groupe de travail est en général de niveau 2 OSI et il est souvent non configurable. On y connecte les câbles des postes de travail, puis le câble qui le relie au switch départemental, et on le met sous tension. Il apprend tout seul les adresses auxquelles il est connecté.

Les LED qui se trouvent sur la face avant du switch permettent de contrôler que tout va bien :

- une LED verte reste allumée pour indiquer que la connexion est bonne ;
- une autre LED clignote quand des données sont en cours de transfert.

La configuration initiale

Les switches disposent en général d'un mode de configuration initiale (*setup mode*) qui permet à l'administrateur d'installer le système rapidement.

Pour cela, on a besoin d'un ordinateur et d'un câble de liaison entre le PC et le switch.

Sous IOS, le mode de configuration initiale s'appelle ***Express Setup***. La procédure est la suivante :

1° Si des câbles sont connectés au switch, les retirer. À part l'ordinateur utilisé pour la liaison Console, aucun appareil ne doit être connecté au switch parce qu'il agit comme serveur DHCP pendant la configuration initiale.

2° Si l'ordinateur utilisé a une adresse IP fixe, le mettre provisoirement en adressage DHCP. Sinon, le switch ne pourra pas dialoguer avec lui.

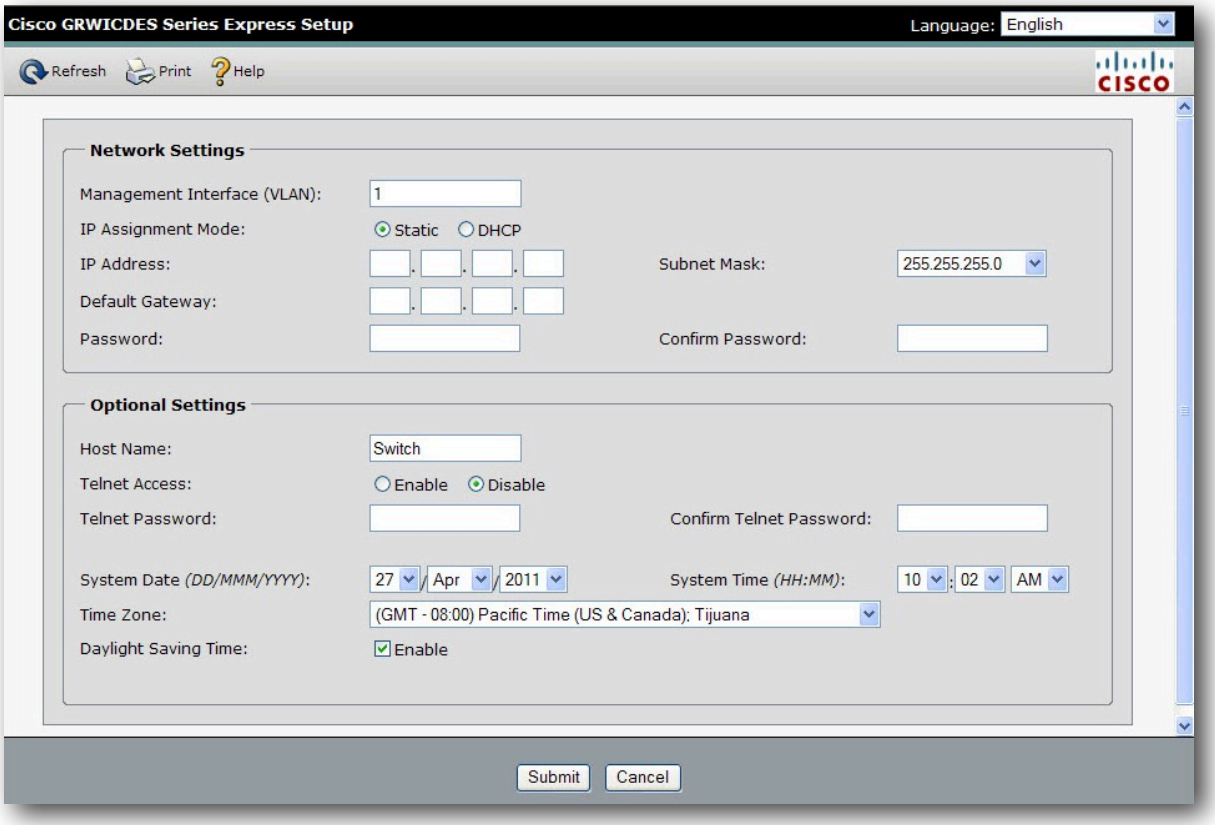
3° Mettre le switch sous tension et attendre qu'il ait effectué le POST. Cela prend plusieurs minutes. Le système indique qu'il a terminé quand les LED du système s'allument en vert.

4° Presser le bouton *Mode* pendant plusieurs secondes. Attendre que les LED qui se trouvent au-dessus deviennent vertes pour relâcher le bouton. Si les LED clignotent, cela signifie que le switch a déjà été configuré. Si on veut reprendre la configuration à zéro, faire un reset en maintenant le bouton *Mode* pressé pendant une vingtaine de secondes.

5° Relier l'ordinateur au switch au moyen d'un câble Ethernet. Utiliser n'importe quel port standard du switch. La LED du port doit s'allumer en vert.

6° Attendre une trentaine de secondes pour laisser au switch le temps de trouver l'ordinateur et d'initier la communication entre les deux appareils.

7° Lancer le navigateur de l'ordinateur et entrer l'adresse *10.0.0.1*. La page du Express Setup apparaît. Elle se présente par exemple ainsi :



Le champ appelé *Default Gateway* désigne l'adresse IP du routeur.

Il est aussi possible de lancer le mode de configuration initiale en mode de texte. Pour cela, utiliser non pas une connexion Ethernet mais la prise Console et le câble RJ-45–DB-9 bleu livré avec le switch. La procédure est la même que celle qu'on a vue dans le chapitre précédent : on relie la prise Console du switch à la prise sérielle de l'ordinateur, on lance un programme d'émulation de terminal sur l'ordinateur (9'600 bps, 8 data bits, pas de parité, 1 stop bit, pas de flow control), et on termine en mettant le switch sous tension.

Le dialogue ressemble à ceci (les messages en caractères normaux sont ceux que le système affiche, les réponses en gras sont entrées par l'utilisateur) :

```
Would you like to enter the initial configuration dialog? [yes/no]: yes
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
Basic management setup configuration only enough connectivity for management of the
system, extended setup will ask you to configure each interface on the system.
Would you like to enter basic management setup? [yes/no]: yes
Enter host name [Switch]: switch1
Enter enable secret: xxx
Enter enable password: yyy
Enter virtual terminal password: zzz
Configure SNMP Network Management? [no]
Enter interface name used to connect to the
management network from the above interface summary: vlan1
Configuring interface vlan1:
Configure IP on this interface? [yes] yes
IP address for this interface: 10.10.20.10
Subnet mask for this interface [255.0.0.0] 255.255.0.0
Would you like to enable as a cluster command switch [yes/no]: no
The following configuration command script was created:
        ...suite de lignes affichant la configuration...
end
[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.
If you want to save the configuration and use it the next time the switch reboots, save
it in nonvolatile RAM (NVRAM) by selecting option 2.
Enter your selection [2]: 2
```

L'interface web n'est pas aussi riche en fonctions que l'interface de commandes. Certaines opérations de configuration ne sont possibles qu'avec les commandes.

Ce dernier choix enregistre la configuration initiale.

Terminer les opérations par la commande :

```
switch1# write memory
```

Le VLAN 1 et l'adresse IP

Sous IOS, tous les switches ont un VLAN activé, le **VLAN 1**. Il n'y a pas de switch « sans VLAN ». Pour afficher les informations sur ce VLAN par défaut, on emploie la commande **show vlan** :

```
switch1# sh run int vlan1

VLAN        Name        Status      Ports
-------     -------     -------     ----------------------------------------
1           default     active      Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7,
                                    Fa0/8, Fa0/9, Fa0/10, Fa0/11, Fa0/12, Gi0/1, Gi0/2
```

Si on n'a pas défini d'adresse IP pour ce switch, on ne pourra le gérer qu'au moyen de la liaison Console. C'est pourquoi on a choisi l'adresse 10.10.20.10 à la page précédente.

Pour connaître cette adresse, c'est la commande *show running-config interface vlan1* :

```
switch1# sh run int vlan1
Building configuration...
Current configuration: 80 bytes
interface Vlan1
   ip address 10.10.20.10 255.255.0.0
   no ip route=cache
end
```

On a besoin de l'adresse IP du switch pour le gérer par le réseau, depuis un hôte du même sous-réseau, mais cela ne suffit pas si on veut pouvoir le gérer depuis un autre sous-réseau ou un autre réseau. Dans ce cas, il faut définir l'adresse du routeur :

```
switch1# conf t
switch1(config)# ip default-gateway 10.10.20.1
```

Le serveur DHCP

Dans la plupart des entreprises, le serveur DHCP est activé sur un serveur Un*x ou Windows, mais on peut aussi l'activer sur un routeur ou un switch. Cela permet d'éviter de donner une tâche supplémentaire inutile au serveur.

Voici une suite de commandes typique pour la mise en œuvre d'un serveur DHCP sur un équipement IOS (routeur ou switch) :

```
system# conf t
system(config)# ip dhcp pool nom-du-pool
system(dhcp-config)# network 10.10.20.0 255.255.255.0
system(dhcp-config)# domain-name nom-du-domaine
system(dhcp-config)# dns-server 10.10.20.253 10.10.20.254
```

```
system(dhcp-config)# default-router 10.10.20.1
system(dhcp-config)# lease 14
system(dhcp-config)# exit
system(config)# ip dhcp excluded-address 10.10.20.0 10.10.20.15
system(config)# ip dhcp excluded-address 10.10.20.253
system(config)# ip dhcp excluded-address 10.10.20.254
system(config)# service dhcp
system(config)# exit
system# show ip dhcp binding
```

La première commande donne le nom du pool d'adresses qu'on va définir pour le service DHCP.

La commande *network* donne l'adresse du réseau et le masque à utiliser.

La commande *domain-name* donne le nom du domaine de notre réseau.

La commande *dns-server* donne l'adresse du serveur DNS primaire et du serveur DNS de secours. La commande *default-router* fait de même pour le routeur du réseau.

La commande *lease* donne la durée de vie de chaque adresse IP en jours. Pour la rendre indéfinie, on emploie la commande *lease infinite*.

La commande *ip dhcp excluded-address* permet d'exclure une plage d'adresse du service DHCP. Ici, les adresses 10.10.20.0 à 15 sont exclues ainsi que les adresses 10.10.20.253 et 254. Ces adresses sont celles des serveurs et des imprimantes. Tous ces équipements doivent avoir une adresse fixe qui se trouve dans le même sous-réseau que les autres (ici, 10.10.20.0 à 255 puisque le masque est 255.255.255.0).

La commande *service dhcp* active le service. La commande *no service dhcp* le désactive.

La commande *show ip dhcp binding* permet de contrôler que le serveur fonctionne bien.

IOS contient un grand nombre de commandes pour administrer un serveur DHCP. Par exemple, on peut mettre en place un serveur DHCP de secours qui s'active en cas de défaillance du serveur primaire.

La configuration des modules

Un switch simple ne comprend qu'un seul module appelé le **module Ethernet**. Comme son nom l'indique, il gère les connexions Ethernet. C'est lui qui s'occupe des fonctions de la couche 2 OSI.

Les switches de niveau 3 et les MLS peuvent, eux, être équipés de divers modules.

Par exemple, on peut installer un module pare-feu. Dans le cas de Cisco, il s'appelle **FWSM** (*firewall services module*) Par rapport aux pare-feu traditionnels, il offre l'avantage d'être plus rapide.

Si un FWSM se trouve dans le slot n°7 du switch qui s'appelle *switch1* et qu'il est géré par le processeur 1, la commande pour y accéder est :

```
switch1# session slot 7 proc 1
```

À partir de là, on utilise les commandes propres aux pare-feu.

Il existe aussi des modules de détection d'intrusion. Cisco les appelle **IDSM** (*intrusion detection system modules*). Ce sont en réalité des serveurs Linux spécialisés. Comme les FWSM, ils sont très rapides. On peut les administrer au moyen d'un logiciel spécialisé.

Un module appelé FlexWAN permet d'intégrer une liaison WAN à un switch.

Des **CSM** (*content switching modules*) permettent de gérer les types de trafic de manière différenciée.

Questions

1. À quelle couche OSI travaille principalement un switch de niveau 3 ? Expliquez votre réponse.
2. Comment faites-vous pour connaître la configuration d'un switch ?
3. Quel est le résultat de ce qui suit ?

   Switch# **config t**
   Switch(config)# **int fa0/22**
   Switch(config-if)# **shutdown**

4. [Suite] Le changement effectué est-il enregistré (= survit à une chute de tension du réseau électrique) ? Expliquez votre réponse.
5. [Suite] Le changement effectué est-il sauvegardé (= survit à une panne du switch) ? Expliquez votre réponse.
6. Avec quel type de switch peut-on activer le mode adaptive-switching ?
7. Parmi les méthodes suivantes, laquelle offre la vitesse de commutation la plus rapide ?

   - ☐ Cut through
   - ☐ Store and forward
   - ☐ Fragment tree
   - ☐ Adaptive switching

8. Comment s'appellent les données transmises par un switch ?
9. Un switch vient à l'instant d'être installé par l'administrateur du réseau. Que fait-il ? Il y a zéro, une ou plusieurs réponses à cocher.

   - ☐ Le switch commence par remplir sa table de commutation en dialoguant avec les nœuds. Quand elle est remplie, il commence à transmettre les trames qu'il reçoit.
   - ☐ Le switch transmet en broadcast les trames dont il ne connaît pas le destinataire.
   - ☐ L'administrateur a rempli la table de commutation avant de mettre le switch en ligne.
   - ☐ L'administrateur a copié dans le nouveau switch la table de commutation d'un switch qui était déjà en ligne sur le réseau.

10. La suite de commandes ci-dessous a pour effet de désactiver le trunking sur le port n°23 du switch (le trunking est le mode qui permet de faire passer le trafic de plusieurs VLAN sur une liaison). En supposant qu'on a mis en place des VLANs sur le réseau, cette suite de commandes est-elle absurde ou a-t-elle un sens dans certains cas ? Expliquez votre réponse.

    ```
    Switch# config t
    Switch(config)# int fa0/23
    Switch(config-if)# switchport mode access
    ```

11. Si la commande ci-dessous ne fonctionne pas, quelle peut en être la raison ?

    ```
    Switch# copy running tftp
    ```

4. Le routage

Le **routage** (*routing*) se définit comme la fonction des routeurs et des switches de niveau 3 d'**acheminer** (*route*) des **paquet** dotés d'une **adresse logique** (en pratique, une adresse IP) depuis le sous-réseau local en direction de leur destination finale, qui peut se trouver dans le même sous-réseau ou de l'autre côté du monde [3].

Le terme de **paquet** (*packet*) désigne un ensemble de bits formatés selon les besoins de la couche 3 du modèle OSI de l'ISO pour être transmis sur un **réseau par paquets** (*packet switching network*), c'est-à-dire sur un réseau capable de scinder les messages en fragments (les paquets), de transmettre ces fragments un à un sans qu'ils aient à passer par le même chemin et, à l'arrivée, de les recombiner en un seul morceau (le message).

Bien entendu, la recompilation finale des paquets s'effectue dans le bon ordre, histoire d'éviter des erreurs dues à une arrivée dans le désordre.

On peut suivre l'acheminement d'un paquet de routeur en routeur au moyen de la commande *traceroute* (*tracert* dans le monde Microsoft).

Le mécanisme de relais implique toute une conversation entre les deux routeurs concernés. Ce sont des dizaines de paquets de service qui s'échangent pour transmettre un seul paquet de données d'utilisateurs (*user data*) [4].

On emploie deux types de protocoles de routing : les protocoles d'acheminement et les protocoles acheminés.

- Les ***routed protocols*** (protocoles acheminés) permettent la transmission des paquets sur le réseau ou l'interréseau. Il n'en subsiste pratiquement plus qu'un : **IP**.
- Les ***routing protocols*** (protocoles d'acheminement) comme **RIP**, **IS-IS**, **OSPF** ou **BGP4** sont utilisés par les routeurs pour :
  - connaître leur environnement, c'est-à-dire découvrir les autres réseaux ;
  - choisir le meilleur chemin ;
  - synchroniser les tables d'acheminement des autres routeurs du réseau avec la leur.

Pour être capable d'acheminer les paquets, un routeur doit connaître :

1° une méthode qui lui permet de gérer les informations d'acheminement ;

2° l'adresse de destination, c'est-à-dire, en général, une adresse IP ;

3° l'adresse des routeurs qui sont ses voisins directs ;

---

[3] Le mot *routeur* désigne dans ce chapitre les routeurs et les switches de niveau 3, et le mot *réseau* désigne les réseaux et les sous-réseaux.

[4] Un paquet de service est un paquet qui ne sert qu'à permettre le mécanisme (ici, le routing) de se dérouler correctement. Un paquet de *user data* contient les données à transmettre.

4° les différents chemins possibles vers tous les autres réseaux (ces informations peuvent être collectées automatiquement ou remplies par l'administrateur du réseau) ;

5° le meilleur itinéraire (***best path***) vers tous les autres réseaux (cela peut aussi être déterminé automatiquement ou manuellement).

L'acheminement des paquets

Le routeur contient une ***routing table*** (table d'acheminement) qui contient les itinéraires possibles vers les autres réseaux.

Par défaut, un routeur ne connaît que les réseaux auxquels il est directement connecté.

La table d'acheminement se remplit de manière statique ou dynamique. Dans le ***static routing*** (acheminement statique), l'administrateur du réseau remplit manuellement la table de routing. Cette solution présente des avantages :

- elle permet à l'administrateur de choisir manuellement les meilleurs chemins ;
- elle diminue le trafic de service entre les routeurs puisqu'ils n'ont pas besoin d'échanger les informations nécessaires pour tenir à jour leurs tables de routage ;
- elle diminue le travail du routeur ;
- elle améliore potentiellement la sécurité puisque l'administrateur peut activer le routage vers certains réseaux et pas d'autres.

Dans le ***dynamic routing*** (acheminement dynamique), les routeurs dialoguent ensemble pour déterminer la topographie du réseau. Cette solution a aussi ses avantages :

- elle évite à l'administrateur de perdre du temps à faire le boulot lui-même ; en particulier, l'ajout d'un sous-réseau dans le réseau peut impliquer la mise à jour de toutes les tables d'acheminement ;
- elle simplifie le travail de l'administrateur ; le besoin de compétences en interréseaux est moindre ;
- elle s'adapte rapidement à tout changement dans la topographie du réseau (défaillance, surcharge d'une ligne, etc.) ;
- si le réseau est grand, c'est la seule solution praticable.

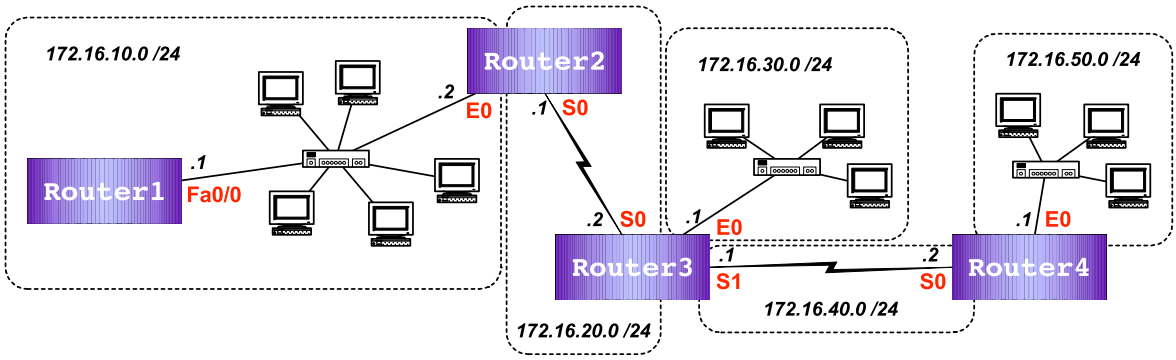
Dans un petit réseau, l'administrateur du réseau n'est pas toujours disponible (vacances, maladie, grossesse ou armée). Il vaut donc souvent mieux utiliser l'acheminement dynamique. Dans un grand réseau, on recourt généralement à une combinaison des deux types d'acheminement.

Remarque : les routeurs ne s'intéressent qu'aux réseaux. Ils ne voient que les chemins d'un routeur à l'autre, ils ne connaissent pas la notion d'ordinateur. Par exemple, s'il existe deux chemins vers un serveur (beaucoup de serveurs sont dotés de deux interfaces de réseau et donc de deux câbles Ethernet), un routeur ne sait rien faire de cette information. Pour gérer les choix d'itinéraires entre switches et serveurs, on utilise des protocoles spécialisés comme **VRRP** (*Virtual Router Redundancy Protocol*) ou **HSRP** (*Hot Standby Router Protocol*). Pour cela, il faut bien entendu que les switches concernés soient dotés des fonctions VRRP ou HSRP, ce qui est le cas des équipements de haut de gamme.

- Acheminement de routeur à routeur ou de switch à switch ⟶ RIP, OSPF, IS-IS ;
- Acheminement de switch à serveur ⟶ VRRP, HSRP.

Exemple

Voici un exemple de réseau avec quatre routeurs [5] :



Le but est que chaque routeur connaisse les cinq sous-réseaux qui constituent le réseau.

Chacun a un masque de sous-réseau de 24 bits (la notation CIDR est */24*).

La configuration de base est assez simple : il faut attribuer une adresse IP à chaque interface avec la commande *ip address* et enregistrer (rendre permanent) chaque changement avec la commande *no shutdown*. Cela s'effectue avec le câble de console.

Voici ce que cela donne avec le premier routeur :

```
Router> enable                                  ← pour entrer dans le mode privilégié
Router# config terminal                         ← pour entrer dans le mode de configuration globale
Router(config)# hostname Router1                ← pour donner le nom Router1 au routeur
Router1(config)# interface fa0/0                ← interface Fast Ethernet
Router1(config-if)# ip address 172.16.10.1 255.255.255.0
Router1(config-if)# no shutdown                 ← pour enregistrer la configuration
Router1(config-if)# Ctrl-Z                      ← pour retourner au mode privilégié normal
Router1# show ip route                          ← pour vérifier le résultat de ce qu'on a fait
```

À partir de cet instant, le routeur a une interface Ethernet dotée de l'adresse 172.16.10.1. On n'a donc plus besoin du câble de console. On peut configurer le routeur avec *telnet* ou *ssh*, en passant par le réseau.

Le deuxième routeur a deux interfaces, ce qui allonge un peu la configuration :

```
Router> enable
Router# config terminal
Router(config)# hostname Router2
Router2(config)# interface e0
Router2(config-if)# ip address 172.16.10.2 255.255.255.0
```

[5] Tom LAMMLE, *Cisco Certified Network Associate Study Guide*, Sybex, San Francisco, 2002, p. 251.

```
Router2(config-if)# no shutdown
Router2(config-if)# interface s0
Router2(config-if)# ip address 172.16.20.1 255.255.255.0
Router2(config-if)# no shutdown
Router2(config-if)# Ctrl-Z
Router2# show ip route
```

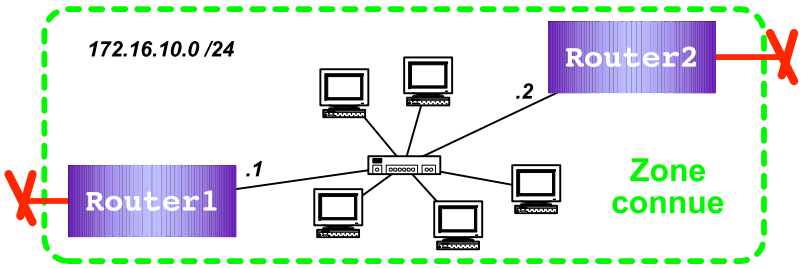
Le troisième routeur a trois interfaces à configurer, et il s'y ajoute la vitesse d'horloge (*clock rate*) à fixer pour les interfaces sérielles :

```
Router> enable
Router# config terminal
Router(config)# hostname Router3
Router3(config)# interface e0
Router3(config-if)# ip address 172.16.30.1 255.255.255.0
Router3(config-if)# no shutdown
Router3(config-if)# interface s0
Router3(config-if)# ip address 172.16.20.2 255.255.255.0
Router3(config-if)# clock rate 64000
Router3(config-if)# no shutdown
Router3(config-if)# interface s1
Router3(config-if)# ip address 172.16.40.1 255.255.255.0
Router3(config-if)# clock rate 64000
Router3(config-if)# no shutdown
Router3(config-if)# Ctrl-Z
Router3# show ip route
```

Le quatrième et dernier routeur a les deux mêmes interfaces *E0* et *S0* que le deuxième, et sa configuration est la même à ceci près, bien entendu, que les adresses sont différentes :

```
Router> enable
Router# config terminal
Router(config)# hostname Router4
Router4(config)# interface e0
Router4(config-if)# ip address 172.16.50.1 255.255.255.0
Router4(config-if)# no shutdown
Router4(config-if)# interface s0
Router4(config-if)# ip address 172.16.40.2 255.255.255.0
Router4(config-if)# no shutdown
Router4(config-if)# Ctrl-Z
Router4# show ip route
```

À ce stade, les routeurs ne voient que leurs voisins immédiats. Pour des raisons de sécurité, tout paquet provenant d'une autre adresse serait purement et simplement rejeté (*discarded*).



Pour que les routeurs se voient, on a trois possibilités : on peut configurer les routeurs en utilisant l'acheminement statique (manuel), l'acheminement par défaut ou l'acheminement dynamique (automatique).

Le routage statique

Si on choisit la méthode de l'acheminement statique, on donne à chaque routeur l'adresse de chaque sous-réseau, le masque de sous-réseau utilisé, et l'adresse du routeur qui se trouve à la frontière entre la partie connue et la partie inconnue du réseau :

```
Router1(config)# ip route 172.16.20.0 255.255.255.0 172.16.10.2
Router1(config)# ip route 172.16.30.0 255.255.255.0 172.16.10.2
Router1(config)# ip route 172.16.40.0 255.255.255.0 172.16.10.2
Router1(config)# ip route 172.16.50.0 255.255.255.0 172.16.10.2
Router1(config)# Ctrl-Z
Router1# show ip route                    ← pour vérifier le résultat
```

On voit que les sous-réseaux sont ajoutés un à un. Le masque reste toujours le même, ainsi que le routeur de la frontière : pour le routeur 1, c'est le routeur 2 qui se trouve à la frontière entre le monde connu et le monde inconnu.

On répète la même opération sur le deuxième routeur :

```
Router2(config)# ip route 172.16.30.0 255.255.255.0 172.16.20.2
Router2(config)# ip route 172.16.40.0 255.255.255.0 172.16.20.2
Router2(config)# ip route 172.16.50.0 255.255.255.0 172.16.20.2
```

Et sur le troisième routeur :

```
Router3(config)# ip route 172.16.10.0 255.255.255.0 172.16.20.1
Router3(config)# ip route 172.16.50.0 255.255.255.0 172.16.40.2
```

Et sur le quatrième :

```
Router4(config)# ip route 172.16.10.0 255.255.255.0 172.16.40.1
Router4(config)# ip route 172.16.20.0 255.255.255.0 172.16.40.1
Router4(config)# ip route 172.16.30.0 255.255.255.0 172.16.40.1
```

On peut vérifier l'interconnectivité des routeurs avec la commande *ping* :

```
Router1# ping 172.16.50.1                    ← contact avec le Router 4 depuis le 1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.50.1, timout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/68/71

Router1#
```

Le routage par défaut

Au lieu de définir des chemins statiques, on peut établir un **acheminement par défaut** (*default routing*), pour autant que le routeur sur lesquel on veut le faire soit la seule porte d'entrée-sortie du réseau (on parle de *stub network*). Si on n'a pas affaire à un *stub network*, on n'a le choix qu'entre l'acheminement statique et l'acheminement dynamique.

Dans notre exemple, c'est le cas du routeur 4. On doit d'abord désactiver les chemins statiques au moyen de la commande *no*, puis activer le chemin par défaut :

```
Router4(config)# no ip route 172.16.10.0 255.255.255.0 172.16.40.1
Router4(config)# no ip route 172.16.20.0 255.255.255.0 172.16.40.1
Router4(config)# no ip route 172.16.30.0 255.255.255.0 172.16.40.1
Router4(config)# ip route 0.0.0.0 0.0.0.0 172.16.40.1
```

De cette manière, le réseau voit tout et accepte toute proposition de dialogue — avec le monde entier si on ne place pas de barrière ailleurs.

Remarque : pour fonctionner correctement, l'acheminement par défaut nécessite l'activation de l'acheminement CIDR au moyen de la commande *ip classless* :

```
RouterX(config)# ip classless
```

Le routage dynamique

Étant automatique, l'acheminement dynamique serait la solution idéale s'il ne constituait pas une charge pour les routeurs (traitements) et le réseau (trafic supplémentaire de paquets de service).

L'acheminement dynamique fait appel à la notion de **distance administrative** (*administrative distance*), ce qui désigne le degré de confiance qu'un routeur accorde aux informations d'acheminement qu'il reçoit. Une « distance » de 0 représente la confiance absolue, une « distance » de 255 l'absence totale de confiance. Un routeur marqué d'un 255 se retrouve en liste noire. Aucun paquet ne lui sera envoyé.

Au démarrage, le système travaille avec des coefficients par défaut :

Interface connectée :	0	OSPF :	110
Itinéraire statique :	1	RIP :	120
EIGRP :	90	EIGRP externe :	170
IGRP :	100	Inconnu :	255

Quand un routeur reçoit un message destiné à la mise à jour de sa table d'acheminement de la part de deux autres routeurs, il utilise les informations données par le routeur en lequel il a le plus confiance et ignore celles de l'autre routeur.

Si les deux routeurs distants ont le même coefficient de confiance (la même «distance» administrative), le routeur utilise le nombre de sauts, le débit des lignes ou d'autres critères pour choisir le meilleur itinéraire.

Un **saut** (*hop*) est un passage à travers un routeur. Le nombre de sauts (*hop count*) est une variable importante du *routing*.

Il existe trois catégories de protocoles d'acheminement dynamique :

1° les protocoles à **vecteur de distance** (*distance-vector* ou *DV protocols*) ;

2° les protocoles à **état de lien** (*link-state* ou *LS protocols*) ;

3° les protocoles hybrides.

Remarque : ces protocoles sont destinés à l'acheminement interne, le mot *interne* étant à prendre dans le sens de « interne à une organisation ». Ils concernent l'acheminement dans le cadre d'un **système autonome** ou **AS** (*autonomous system*), ce qui veut dire que ce sont des **IGP** (*interior gateway protocols*), par opposition aux **EGP** (*exterior gateway protocols*), qui concernent, eux, les liaisons entre systèmes autonomes.

Les protocoles à vecteur de distance

Les protocoles à vecteur de distance se caractérisent par le fait que le routeur choisit le chemin le plus court. Il utilise pour cela un calcul mathématique qui varie selon le protocole.

RIP

RIP (*Routing Interior Protocol*) est un protocole à vecteur de distance. Il utilise un seul critère de choix : pour lui, le chemin le plus court est celui qui comprend le moins de sauts.

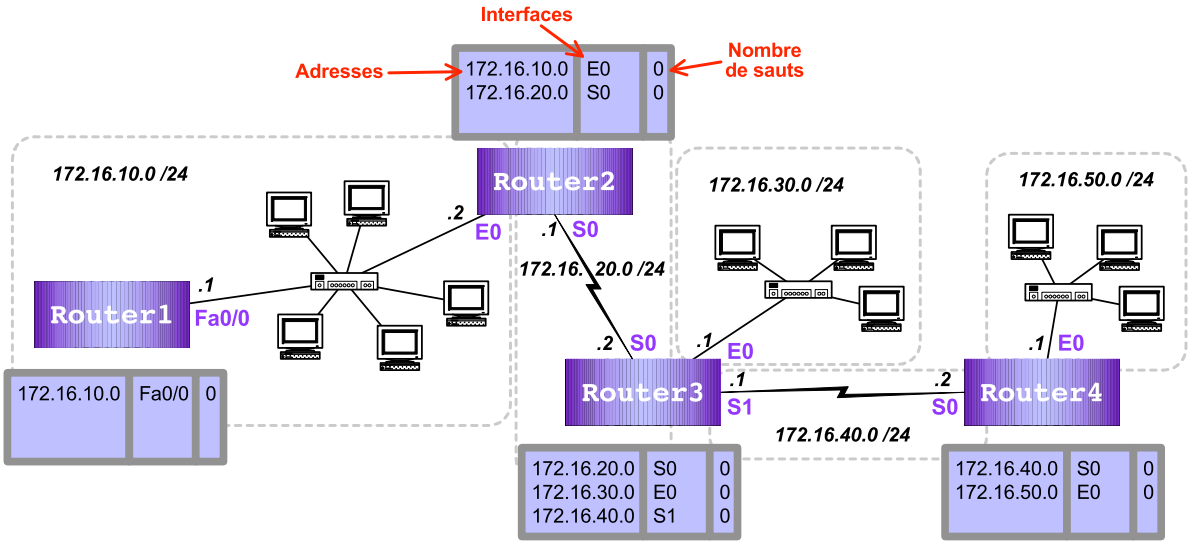
S'il existe plusieurs itinéraires équivalents, RIP les utilise à tour de rôle (on parle de *round-robin*). C'est donc un protocole qui ne nécessite aucun réglage. La tâche de l'administrateur se limite à l'activer ou le désactiver.

Pour que tous les routeurs d'un réseau voient le même réseau, chacun d'eux envoie les informations d'acheminement qu'il connaît à ses voisins immédiats. La propagation se fait ainsi de voisin à voisin, raison pour laquelle on parle de *routing by rumor*. Le temps que prend la propagation pour s'effectuer entièrement s'appelle le temps de **convergence**.

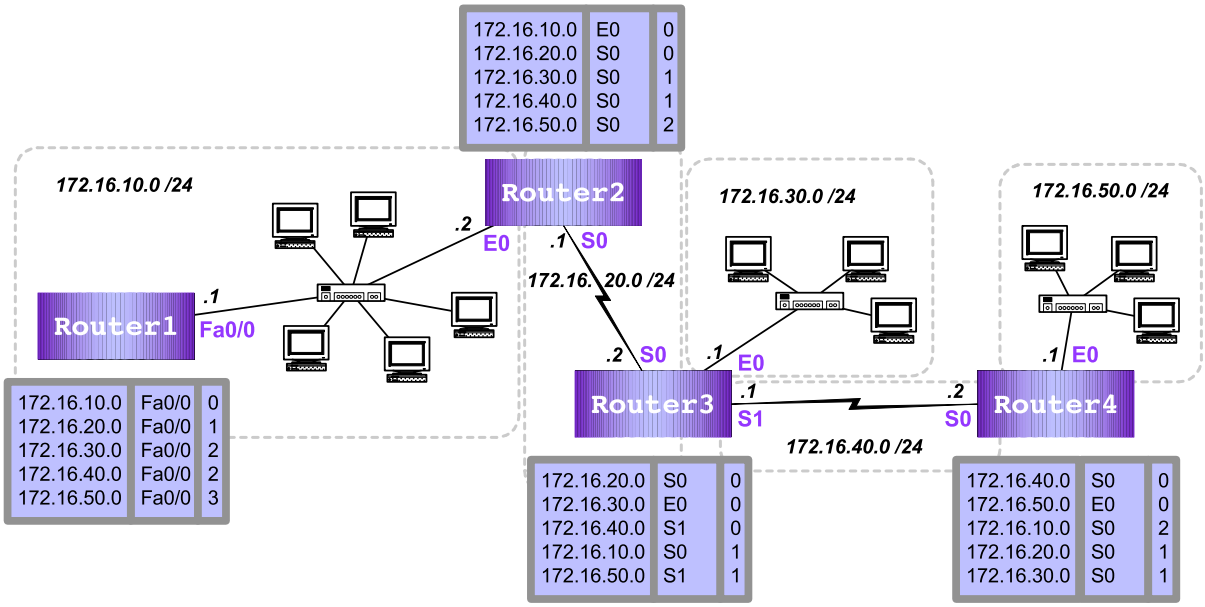
Les tables d'acheminement RIP se mettent à jour toutes les 30 secondes (c'est l'*update timer*). Chaque routeur envoie toute sa table à chacun de ses voisins.

Si un routeur ne reçoit plus d'annonces de mise à jour concernant un chemin pendant 90 secondes, il marque ce chemin comme invalide (c'est le *route invalid timer*). Il s'écoule ensuite 240 secondes jusqu'à ce que le chemin en question soit retiré de la table (c'est le *route flush timer*).

Dans l'exemple qu'on a vu plus haut, les tables d'acheminement se présentent ainsi au moment où on met les routeurs sous tension :



Une fois la convergence terminée, les tables se présentent ainsi :



Pendant la convergence, les routeurs ne transmettent pas de *user data*.

Avec ce système de propagation de voisin à voisin, une boucle infinie pourrait se créer. Pour éviter cela, RIP admet au maximum 15 sauts (cette valeur peut être modifiiée). Au-delà, il considère que l'itinéraire est erroné. Ce mécanisme se nomme *count to infinity* (« comptage à l'infini »).

Il existe trois autres mécanismes de récupération sur défaillance :

- le ***split horizon***, qui interdit aux informations d'acheminement d'être renvoyées à leur expéditeur ;

- le ***poison reverse***, qui assigne à tout chemin non disponible une métrique plus élevée que la valeur maximale, ce qui a pour effet de fermer ce chemin ;
- le ***hold-down***, qui oblige les routeurs à ignorer pendant un moment toute annonce au sujet d'une ligne qui passe sans cesse de l'état *OK* à l'état *en panne* (on parle de *flapping*) ; cela empêcherait le réseau de converger et plus rien ne serait transmis.

Pour activer RIP, il faut d'abord désactiver le routing statique, s'il existe, puis activer RIP au moyen des commandes *router rip* et *network adresse_réseau*. Voici ce que cela donne avec le premier routeur de l'exemple :

```
Router1(config)# no ip route 172.16.20.0 255.255.255.0 172.16.10.2
Router1(config)# no ip route 172.16.30.0 255.255.255.0 172.16.10.2
Router1(config)# no ip route 172.16.40.0 255.255.255.0 172.16.10.2
Router1(config)# no ip route 172.16.50.0 255.255.255.0 172.16.10.2
Router1(config)# router rip
Router1(config-router)# network 172.16.0.0
```

L'adresse *172.16.0.0* indique que RIP doit s'activer pour les adresses *172.16.0.0* à *172.16.255.255*.

La même opération s'effectue sur le deuxième routeur :

```
Router2(config)# no ip route 172.16.30.0 255.255.255.0 172.16.20.2
Router2(config)# no ip route 172.16.40.0 255.255.255.0 172.16.20.2
Router2(config)# no ip route 172.16.50.0 255.255.255.0 172.16.20.2
Router2(config)# router rip
Router2(config-router)# network 172.16.0.0
```

Et ainsi de suite pour les autres routeurs. La commande *show ip route* permet de vérifier que RIP s'est correctement activé.

Pour éviter que le protocole se propage au-delà d'une frontière donnée, on utilise la commande *passive-interface* sur le routeur concerné. Par exemple :

```
Router(config-router)# passive-interface s 1
```

Il existe trois version de RIP : RIP (RFC 1388), RIP II (RFC 1723) et RIP ng (*next generation*), aussi appelée RIP v6 (RFC 2080) parce qu'elle est compatible avec la version 6 d'IP. La version 1 ne reconnaît pas les masques de sous-réseaux et n'est plus utilisée. Actuellement, la version par défaut est la 2, ce qui signifie que le sigle *RIP* tout court veut normalement dire **RIP II** (ou RIP 2).

RIP ne tient compte que du nombre de sauts. En particulier, il ignore le débit des lignes impliquées. Par conséquent, si une ligne passe par une liaison Ethernet à 100 Mbps et une autre par une ligne téléphonique de 30 kbps (ce qui veut dire qu'elle est 3'000 fois plus lente), le routeur va considérer la ligne lente comme la meilleure si le nombre de sauts est moins élevé. Cela veut dire que, si on utilise RIP sur un réseau où coexistent des débits différents, un phénomène de goulet d'étranglement va se former. On parle de congestion de chas d'aiguille (*pinhole congestion*).

Par contre, RIP est le plus léger et le plus simple à paramétrer de tous les IGP. Il reste donc très populaire dans les petits réseaux. Si tout le réseau fonctionne sur un seul débit, sa métrique simpliste ne constitue pas un inconvénient.

IGRP

IGRP (*Interior Gateway Routing Protocol*) est également un protocole à vecteur de distance, mais il est plus subtil que RIP car il utilise quatre critères pour déterminer le meilleur itinéraire :

- le débit de la ligne (*bandwith*), de 1200 bps à 10 Gbps ;
- la charge sur la ligne (*load*), de 1 à 255 ;
- la fiabilité de la ligne (*reliability*), de 1 à 255 ;
- le délai de transmission (*delay*), de 2^0 à 2^{24}.

Le poids respectif de ces quatre éléments peut être modifié par l'administrateur, mais les valeurs par défaut conviennent dans la majorité des cas.

Par rapport à RIP, IGRP a toutefois un défaut : il est la propriété de Cisco.

Par défaut, l'*update timer* d'IGRP est de 90 secondes, le *route invalid timer* de trois fois l'*update timer* et le *hold-down timer* de trois fois l'*update timer* plus dix secondes. Le *flush timer* est égal à sept fois l'*update timer*.

Le protocole accepte jusqu'à 255 sauts, un nombre suffisant pour les plus grands réseaux.

IGRP peut gérer un flux de données sur deux lignes. En cas de panne de l'une d'elles, la transmission se poursuit en mode dégradé sur l'autre (*automatic switch-over*).

L'activation d'IGRP est presque aussi simple que celle de RIP. On doit seulement y ajouter un numéro identifiant le système autonome (*AS number*), par exemple 2 :

```
Router1(config)# router igrp 2
Router1(config-router)# network 172.16.0.0
Router2(config)# router igrp 2
Router2(config-router)# network 172.16.0.0
```

Et ainsi de suite. La commande *show ip route* permet de vérifier que le protocole s'est correctement activé.

Un routeur peut faire partie de plusieurs systèmes autonomes. Par exemple :

```
Router5(config)# router igrp 2
Router5(config-router)# network 172.16.0.0
Router5(config-router)# exit
Router5(config)# router igrp 4
Router5(config-router)# network 10.0.0.0
```

Plusieurs commandes sont utiles pour voir si la configuration est correcte :

- *show protocols* affiche les interfaces activées et les adresses IP
- *show ip protocol* affiche le ou les protocoles d'acheminement activés (RIP, IGRP, etc.)
- *debug ip rip* affiche les messages de mise à jour des *routing tables* de RIP au fur et à mesure de leur réception
- *debug ip igrp events* affiche les messages de mise à jour des *routing tables* de EIGRP au fur et à mesure de leur réception et de leur expédition

- *debug ip igrp transactions* affiche les messages de demande de mise à jour de la part des routeurs voisins et les réponses du routeur sur lequel on a tapé la commande.

Questions

1. Quelle est la commande qui affiche les protocoles de routing IP configurés ?
   - ☐ show ip route
   - ☐ show protocole
   - ☐ show ip protocole
   - ☐ debug all
2. Parmi les propositions ci-dessous, laquelle est un chemin par défaut correct ?
   - ☐ route ip 172.0.0.0 255.0.0.0 s0
   - ☐ ip route 0.0.0.0 0.0.0.0 172.16.20.1
   - ☐ ip route 0.0.0.0 255.255.255.255 172.16.20.1
   - ☐ route ip 0.0.0.0 0.0.0.0 172.16.10.256
3. Parmi les propositions ci-dessous, laquelle est une adresse de broadcast ?
   - ☐ 0.0.0.0
   - ☐ 1.1.1.1
   - ☐ 127.1.1.0
   - ☐ 255.255.255.255
4. Quelle est la portée d'un broadcast ?
   - ☐ le sous-réseau
   - ☐ le réseau
   - ☐ l'AS (*Autonomous System*)
   - ☐ l'interrréseau
5. Quelle est la distance administrative par défaut de RIP ?
   - ☐ 0
   - ☐ 1
   - ☐ 100
   - ☐ 120
6. Comment s'appelle l'algorithme d'acheminement utilisé par RIP ?
   - ☐ Routed information
   - ☐ Link together
   - ☐ Link state
   - ☐ Distance vector
7. Quelle est la métrique de routing de RIP ?
   - ☐ Count to infinity
   - ☐ Hop count
   - ☐ TTL

- ☐ Bandwith, delay

8. Quelle est la commande qui affiche la table de routing de RIP ?
   - ☐ show ip config
   - ☐ show ip arp
   - ☐ show ip route
   - ☐ show ip table
9. Quelle est la commande qui affiche l'état actuel du protocole de routing activé ?
   - ☐ show ip route
   - ☐ show ip protocols
   - ☐ show ip broadcast
   - ☐ debug ip
10. Quelle est la commande qui empêche les mises à jour de routing de sortir de l'interface sérielle 0 ?
    - ☐ Router(config-if)# **no routing**
    - ☐ Router(config-if)# **passive interface**
    - ☐ Router(config-router)# **passive-interface s0**
    - ☐ Router(config-router)# **no routing updates**
11. Parmi les propositions ci-dessous, lesquelles sont vraies ?
    - ☐ RIP 2 est un protocole à vecteur de distance
    - ☐ RIP 2 utilise le débit comme métrique de distance
    - ☐ RIP 2 utilise les masques de sous-réseau
    - ☐ L'intervalle de mise à jour (*update timer value*) d'OSPF est de 90 secondes
12. Qu'est-ce qu'une métrique de 16 sauts représente dans un réseau RIP ?
    - ☐ Le nombre de routeurs dans le réseau
    - ☐ Le nombre de sauts
    - ☐ Une distance inaccessible
    - ☐ Le dernier saut possible
13. Parmi les propositions ci-dessous, lesquelles sont des solutions pour réduire les routing loops en mode de vecteur de distance ?
    - ☐ Split horizon
    - ☐ Réseau hiérarchique
    - ☐ Route poison
    - ☐ Link-state routing
    - ☐ Count to infinity
14. À quoi servent les hold-downs ?
    - ☐ À empêcher le protocole d'aller plus loin

- ☐ À empêcher les annonces normales de mise à jour de réactiver un chemin qui n'est plus utilisable
- ☐ À empêcher les annonces normales de mise à jour de restaurer un chemin qui vient de redevenir valide
- ☐ À empêcher des annonces de mise à jour invalides de réactiver un chemin qui n'est plus valide

15. Qu'est-ce que le split horizon ?

- ☐ Lorsqu'un routeur reçoit sur une interface donnée une annonce de chemin (un *route advertisement*), il ne la renvoie pas en retour sur la même interface
- ☐ Le mécanisme qui scinde un trafic en plusieurs flux sur un grand réseau
- ☐ Le mécanisme consistant à retenir les annonces de mise à jour destinées à une ligne devenue invalide
- ☐ Le mécanisme qui empêche des annonces de mise à jour de restaurer un chemin qui n'est plus valide

16. Quelle est la distance administrative par défaut d'OSPF ?

- ☐ 90
- ☐ 100
- ☐ 110

17. Qu'est-ce que le poison reverse ?

- ☐ Le mécanisme qui renvoie les annonces reçues d'un routeur, ce qui stoppe ces annonces
- ☐ Des informations reçues d'un routeur, informations qui ne peuvent pas être renvoyées au routeur qui est à leur origine
- ☐ Le mécanisme qui empêche les annonces valides de mise à jour de restaurer un chemin qui vient de redevenir valide
- ☐ Le mécanisme qui amène un routeur à définir la métrique d'une liaison à invalider.

5. Le fonctionnement des IGP

Un ***routing protocol*** est un ensemble de règles qui définissent une méthode de choix du meilleur chemin entre deux systèmes de niveau 3 OSI [6]. Les chemins sont stockés dans des ***routing tables***.

Pour être acheminés, les paquets de données doivent être **routables**, ce qui veut dire qu'ils doivent se conformer aux règles d'un ***routed protocol***.

Actuellement, le seul protocole routé répandu est **IP**, versions 4 et 6 : **IPv4** et **IPv6**.

L'acheminement des paquests peut s'effectuer soit entre plusieurs organisations, soit à l'intérieur d'une seule organisation sur plusieurs sites, soit sur un seul site.

Dans le premier cas, on utilise un protocole de type **EGP** (*Exterior Gateway Protocol*, où *exterior* veut dire « entre organisations »). Le seul protocole EGP utilisé aujourd'hui est **BGP4** (*Border Gateway Protocol, version 4*). Il utilise toute une palette de méthodes (*policies*) pour choisir le chemin. Chaque organisation est appelée **AS** (*Autonomous System*). Sur l'Internet, chaque AS est identifié de manière unique par un numéro unique appelé **ASN** (*AS Number*).

Dans le second cas, on utilise un protocole de type **IGP** (*Interior Gateway Protocol*). Il en existe plusieurs : **RIP** (*Routing Information Protocol*), IGRP (*Interior Gateway Routing Protocol*), **EIGRP** (*Enhanced IGRP*), **OSPF** (*Open Shortest Path First*) et **IS-IS** (*Intermediate System to Intermediate System*).

Remarque importante : les routeurs n'ont besoin d'aucun protocole pour détecter les sous-réseaux auxquels ils sont reliés *directement*. Ce sont seulement les sous-réseaux masqués derrière les routeurs voisins qui doivent être découverts au moyen d'un protocole de routage.

Les protocoles de routage poursuivent six buts principaux :

1° apprendre dynamiquement les chemins qui mènent vers les autres sous-réseaux du réseau ;

2° choisir le meilleur chemin vers chaque sous-réseau ;

3° remplir les tables de routage avec ces informations ;

4° détecter tout chemin qui n'est plus valide et le retirer des tables de routage ;

5° empêcher les boucles de routage (*routing loops*), c'est-à-dire éviter qu'un paquet de données ne tourne en rond de routeur en routeur ;

6° mettre à jour les tables de routage aussi rapidement que possible.

Le temps nécessaire pour mettre à jour les tables est appelé **temps de convergence** (*convergence time*). Plus il est court, mieux c'est, car l'intégrité des données de routage n'est obtenue qu'une fois la convergence terminée.

---

[6] Ces systèmes sont soit des routeurs soit des commutateurs de niveau 3. Dans ce texte, ils sont appelés «routeurs» pour simplifier la lecture.

Critères de comparaison

Plusieurs critères permettent de comparer les IGPs. Le plus important est la **logique d'acheminement**. Il en existe deux : le vecteur de distance et l'état de la liaison.

Avec le **vecteur de distance** (*distance vector*), les routeurs recherchent le chemin le plus court. Ils n'ont pas une vue globale du réseau. Par rapport aux autres routeurs du réseau, ils sont passifs : ils attendent de recevoir un message de leur part pour les inclure dans leurs tables. Pour choisir le meilleur chemin, ils calculent celui qui leur paraît le plus court ; on parle de **métrique** (*metric*). Certaines métriques sont plus réalistes que d'autres. Le meilleur algorithme est celui qui trouve le meilleur chemin en un minimum de temps.

Avec l'**état de la liaison** (*link-state*), les routeurs du réseau ont une vision globale du réseau. Ils l'explorent, recherchant tous les routeurs qui en font partie.

Le **temps de convergence** constitue un autre critère important. En pratique, il varie de quelques secondes à quelques minutes.

La **mise à jour** des tables est un troisième critère. Elle peut être complète (*full routing update*) ou partielle (*partial routing update*). Dans le second cas, seuls les changements sont publiés, ce qui génère moins de trafic sur le réseau.

Enfin, il y a la compatibilité avec le **CIDR** (*Classless Inter-Domain Routing*). Elle est nécessaire pour la gestion de masques de sous-réseaux différents à l'intérieur d'un seul réseau (*VLSM, Variable-Length Subnet Masking*).

RIP

RIP est un protocole à vecteur de distance. Il n'utilise que le nombre de sauts (*hop count*) comme métrique. Il ne tient donc aucun compte du débit des lignes.

La version 1 de RIP ne s'emploie plus parce qu'elle n'est pas compatible avec le CIDR. La version 2 l'a remplacée.

RIP envoie seulement des mises à jour complètes. Par défaut, il le fait toutes les 30 secondes.

Il converge lentement.

C'est un standard public (RFC 2453) et il est très simple à mettre en place. Il reste donc un bon choix sur les petits réseaux où le débit est le même partout.

IGRP

IGRP appartient à Cisco. Il ressemble à RIP, mais il tient compte du débit et du délai de transmission, ce qui est beaucoup plus réaliste que le nombre de sauts. En outre, il peut être paramétré pour tenir compte d'autres paramètres : la charge sur la ligne, sa fiabilité et la taille des paquets transmis. Par contre, il est incompatible avec VLSM. Il n'existe plus depuis la version 12.3 d'IOS.

EIGRP

EIGRP appartient aussi à Cisco. Il remplace IGRP. Comme lui, il se base sur le débit du tronçon le plus lent, le délai de transmission, la fiabilité de la ligne et la charge pour choisir le meilleur chemin (mais pas la taille des paquets).

Il est très simple à mettre en place, raison pour laquelle il a remplacé IGRP.

C'est un protocole hybride (ni à vecteur de distance, ni à état de la liaison). La logique de routage suit un algorithme appelé DUAL (*Diffusing Update Algorithm*), qui est plus léger que celui d'OSPF parce qu'il ne collecte pas les infomations nécessaires pour avoir une vue globale du réseau.

Contrairement à RIP, il recherche ses voisins avant de leur envoyer les mises à jour (RIP se contente d'attendre d'avoir de leurs nouvelles).

Il envoie des mises à jour partielles de sa table lorsqu'un changement se produit.

Il est compatible avec VLSM.

Il précalcule les chemins alternatifs, si bien que, quand un chemin devient invalide, il peut commuter très rapidement sur un autre chemin. Cela permet un temps de convergence très court (1 ou 2 secondes).

EIGRP est un bon choix pour autant qu'on n'utilise que du matériel Cisco.

OSPF

OSPF est le protocole de routage le plus évolué, mais aussi le plus lourd. C'est un protocole de type *link-state*.

Un routeur OSPF commence par rechercher ses voisins et les placer dans sa table des voisins (*neighbour table*). Il échange ensuite avec eux les informations nécessaires pour établir la géographie du réseau, qu'il introduit dans une table appelée la *topology table*. Il exécute ensuite un algorithme appelé **SPF** (*Shortest Path First*) pour calculer les meilleurs chemins.

Par défaut, le critère pour la métrique est la somme de la bande passante de chaque tronçon du chemin total.

Il est compatible avec VLSM.

Il converge rapidement.

Contrairement à EIGRP, c'est un standard public. C'est un bon choix pour les réseaux de taille moyenne ou grande.

IS-IS

Comme OSPF, IS-IS est un protocole de type *link-state* et il utilise aussi l'algorithme SPF pour calculer le meilleur chemin.

Il y a beaucoup de petites différences entre OSPF et IS-IS, mais aucune qui permettrait de dire que l'un est clairement supérieur à l'autre.

La configuration d'un réseau OSPF ou IS-IS nécessite une très bonne connaissance du fonctionnement du protocole. Il existe des livres entiers sur la question. Exemple : Jeff DOYLE, *OSPF and IS-IS: Choosing an IGP for Large-Scale Networks*, Addison-Wesley, New York, 2005 (480 pages). Contrairement à ce que le titre fait croire, ce n'est pas un livre sur le choix entre les deux solutions (phase d'analyse préliminaire), mais sur la compréhension des protocoles et leur mise en œuvre (phases de conception et de réalisation). Cisco consacre un volume entier de son ouvrage en quatre tomes sur les commandes IOS liées à IP : *Cisco IOS IP Command Reference, Volume 2 of 4: Routing Protocols*, San Jose, 2003 (690 pages).

Questions

1. Une entreprise comprend plusieurs dizaines de switches de niveau 3. Est-ce qu'ils se voient ?

   - ☐ Oui, ils se voient tous
   - ☐ Oui, mais seulement s'ils emploient un IGP
   - ☐ Oui, mais seulement s'ils emploient un routage à vecteur de distance
   - ☐ Oui, mais seulement s'ils emploient un routage de type link-state
   - ☐ Chacun d'eux ne voit que ses voisins immédiats

6. Les ACL

Une **ACL** (*Access Control List*) est une structure de données qui liste des **utilisateurs**, des **ressources** et les **droits d'accès** que ces utilisateurs ont sur ces ressources. Les *ACLs* existent notamment sous Unix, Linux, Windows et IOS.

Dans le contexte des ACL, le mot « utilisateur » désigne normalement un hôte (un équipement qui a une adresse IP) et non pas une personne humaine.

Chez Cisco, les ACLs sont des listes de conditions, c'est-à-dire des **règles** qui contrôlent les accès entrants et sortants à un segment de réseau.

Remarque importante : la seule règle suivie est la première qui concerne la ressource concernée :

1° IOS lit l'ACL séquentiellement, en commençant par la première ligne ;

2° dès qu'il trouve une ligne qui s'applique au paquet concerné, la lecture se termine.

Il faut donc placer les tests les plus spécifiques en tête de liste et les tests les plus génériques en fin de liste.

Si un paquet ne répond à aucune des règles de la liste, il est rejeté sauf si la dernière ligne de l'ACL est *permit any*.

Une **ACL standard** filtre l'accès en se basant sur l'adresse IP de l'expéditeur du paquet. On la place aussi près que possible du segment de réseau destinataire.

Une **ACL étendue** (*extended ACL*) filtre l'accès en se basant sur plusieurs éléments :

- l'adresse IP de l'expéditeur du paquet ;
- l'adresse IP du destinataire ;
- le protocole de réseau utilisé (couche 3 OSI) ;
- le numéro de *port* (couche 4).

On la place aussi près que possible de la source des paquets.

Une ACL d'entrée (*inbound ACL*) contrôle les paquets entrants. Une ACL de sortie (*outbound ACL*) contrôle les paquets sortants.

Chaque interface d'un routeur (ou d'un *switch* de niveau 3) peut avoir au maximum deux ACLs : une entrante et une sortante.

Le paramétrage

Chaque ACL est identifiée par un numéro. Les ACLs standard doivent porter un numéro entre 1 et 99, et les ACLs étendues entre 100 et 199.

Un mécanisme appelé ***wildcard mask*** analogue à un masque de sous-réseau inversé permet de spécifier une suite d'adresses. Par exemple, le masque 0.0.0.255 appliqué à l'adresse 172.16.0.0 représente les adresses comprises entre 172.16.0.0 et 172.16.0.255.

On peut définir des suites de 4, 8, 16, 32, 64 ou 256 adresses, avec des masques respectifs de 3, 7, 15, 31, 63 et 255. Par exemple, le masque 0.0.0.7 appliqué à l'adresse 172.16.0.8 représente les adresses comprises entre 172.16.0.8 et 172.16.0.15.

Le mot clé ***deny*** ferme un accès, le mot clé ***permit*** l'autorise.

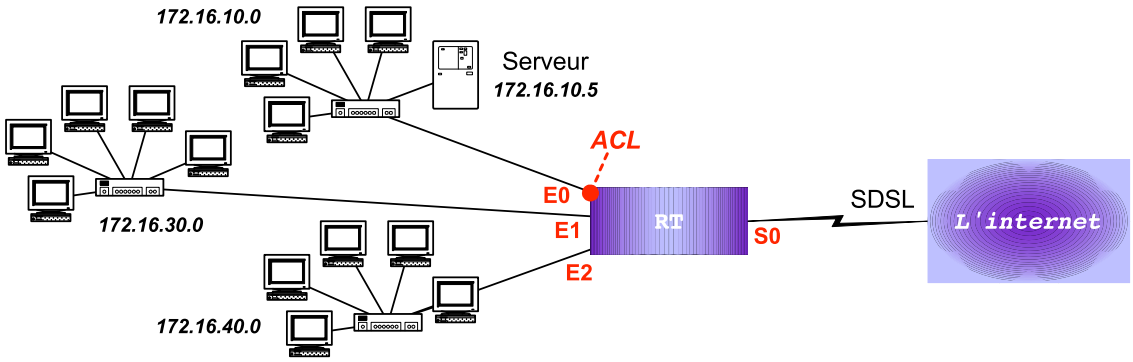
L'expression *permit any* ouvre tous les accès sauf ceux qui ont été fermés précédemment dans l'ACL. Elle équivaut à *permit 0.0.0.0 255.255.255.255*.

La commande ***show access-list*** affiche les ACL configurées sur le routeur mais elle n'indique pas sur quelles interfaces elles se trouvent. La commande *show access-list 10* fait de même pour l'ACL 10. La commande *show running-config*, elle, affiche les ACL et les interfaces.

Les commandes *show ip access-list* et *show ip interface* affichent respectivement les ACL de type IP du routeur et les interfaces dotées de listes d'accès.

Exemple

Voici un réseau avec un routeur, un LAN composé de trois sous-réseaux et un accès SDSL à l'Internet :



Ci-dessous, voici une ACL standard possible. On lui donne l'identifiant 10. C'est une liste sortante (*outbound*) située sur l'interface E0. Elle interdit au sous-réseau 172.16.30.0 d'accéder au sous-réseau 172.16.10.0. Par contre, il a accès aux autres sous-réseaux ainsi qu'à l'Internet.

```
RT> enable
RT# config t
RT(config)# access-list 10 deny 172.16.30.0 0.0.0.255
RT(config)# access-list 10 permit any
RT(config)# interface e0
RT(config-if)# ip access-group 10 out
```

Voici maintenant un exemple d'ACL étendue. Les accès *ftp* et *telnet* au serveur sont interdits, les autres autorisés.

```
RT> enable
RT# config t
RT(config)# access-list 110 deny tcp any host 172.16.30.5 eq ftp
RT(config)# access-list 110 deny tcp any host 172.16.30.5 eq telnet
RT(config)# access-list 110 permit ip any any
RT(config)# interface e0
RT(config-if)# ip access-group 110 out
```

Questions

1. Sur quels éléments les ACLs standard sont-elles basées ?

   - ☐ L'adresse de destination et le wildcard mask
   - ☐ L'adresse de l'expéditeur et le wildcard mask
   - ☐ L'adresse de destination et le subnet mask
   - ☐ L'adresse de l'expéditeur et le subnet mask

1. Quelles sont les affirmations ci-dessous qui sont vraies des ACLs ?

   - ☐ Toute ACL va implicitement refuser tout trafic
   - ☐ Toute ACL doit avoir au moins une règle *permit*
   - ☐ La dernière ligne de toute ACL doit être une règle *permit*
   - ☐ Toutes ces affirmations

7. Les VLAN

Un **VLAN** (*Virtual Local Area Network*) est un **ensemble de nœuds** (des postes de travail, des serveurs, des imprimantes, etc.) connectés à des ports de switches, où les ports sont définis par l'administrateur selon le ou les critères qu'il souhaite.

Pour être compatible avec les VLAN, un switch ou un routeur doit permettre la configuration de topologies logiques par-dessus l'infrastructure physique du réseau.

Par défaut, les switches (niveau 2) séparent un réseau en plusieurs domaines de collision, tandis que les routeurs et les fonctions de niveau 3 des switches le séparent en plusieurs domaines de diffusion. Les VLANs font de même : ils séparent le réseau plusieurs **domaines de diffusion** (*broadcast*).

Des nœuds qui se trouvent dans des VLANs différents ne peuvent pas communiquer au niveau 2 du modèle OSI. Pour communiquer, ils doivent passer par un routeur ou un switch de niveau 3 ou plus.

Un réseau sans VLANs est appelé un **réseau plat** (*flat network*) parce qu'il ne définit qu'un seul domaine de diffusion.

Par rapport au modèle plat, le modèle à VLAN présente plusieurs avantages :

- il réduit le trafic en segmentant les domaines de broadcast ;
- il améliore la sécurité puisque le VLAN *a* est invisible depuis le VLAN *b* et réciproquement ;
- il simplifie la recherche des défaillances (par exemple, une tempête de broadcast ne se propage pas partout, elle reste limitée au VLAN où elle est née) ;
- il simplifie les changements (ajouter, modifier ou supprimer un nœud se fait simplement en configurant un port).

Il a aussi un inconvénient :

- si une trame passe d'un VLAN à l'autre, elle doit traverser un équipement de niveau 3 (routeur ou switch), ce qui prend plus de temps qu'un déplacement à plat.

Pour bien faire, on essaye d'observer la **règle des 80/20** : on construit l'architecture du réseau de telle manière que 80 % au moins du trafic reste dans le VLAN où il est né.

On peut créer autant de VLAN qu'on veut, et cela sur un ou plusieurs switches. Un ensemble de switches sur lequel a été défini un ensemble de VLAN s'appelle un ***switch fabric*** (« tissu de *switches* »). On peut créer un ou plusieurs VLAN sur un ou plusieurs switches. C'est une relation *n* à *n*.

Pour identifier les trames (*frames*), on leur attache un identifiant qu'on appelle *tag*, *VLAN ID* ou *colour*. C'est le ***frame tagging*** (marquage de trame). Il s'ajoute à l'adresse MAC.

Un **VLAN statique** est un VLAN créé manuellement par l'administrateur. Ce travail peut être fait dans IOS ou dans la GUI d'une application tierce de management de réseau (elle génère automatiquement en arrière-fond les commandes IOS).

Un **VLAN dynamique** se modifie automatiquement en suivant les mouvements des utilisateurs. Par exemple, si l'utilisateur dont l'ordinateur qui a l'adresse MAC *x* se déplace dans une autre salle, le ou les VLAN auxquels il appartient se redéfinissent automatiquement sur le nouveau port du nouveau switch auquel il se connecte.

Le critère d'appartenance à un VLAN peut être l'adresse MAC, le protocole ou l'application, mais le cas le plus répandu est celui des adresses MAC. Dans IOS, le serveur VMPS (*VLAN Management Policy Server*) s'emploie pour construire une base de données de correspondance d'adresses MAC et de VLAN — une adresse MAC peut être définie comme appartenant à 1 à *n* VLAN.

La mise en place de VLAN dynamiques demande plus de travail que celle de VLAN statiques. Par contre, les VLAN dynamiques demandent moins de travail de maintenance.

Le trunking

Une **liaison d'accès** (*access link*) relie un nœud à un switch. À un instant donné, elle appartient à un VLAN et un seul. Le débit peut être de 10, 100, 1'000 ou 10'000 Mbps. Un ***trunk link***, lui, est une liaison inter-VLANs (l'expression de *trunk line* signifie liaison interurbaine). Il transporte le trafic de plusieurs VLANs. Il relie soit un switch à un autre switch, soit un switch à un routeur, soit un switch à un serveur.

Grâce au ***trunking***, un serveur peut être défini comme membre de tous les VLAN du réseau, ce qui évite au trafic de devoir passer à travers un routeur ou un switch de niveau 3. Par défaut, un trunk transporte le trafic de tous les VLAN définis. L'administrateur peut en retirer manuellement.

Il existe plusieurs protocoles de trunking :

- LANE (*LAN Emulation*), pour les réseaux ATM (en voie de disparition) ;
- 802.10 pour les réseaux CDDI et FDDI (en voie de disparition) ;
- **ISL** (*Inter-Switch Link*), un protocole qui fonctionne sur 100Base*x* et 1 GE ;
- **802.1q**, un protocole normalisé de l'IEEE qui doit s'utiliser si on utilise plusieurs marques de routeurs.

ISL offre des performances supérieures à celles de 802.1q, mais il est spécifique à Cisco.

Les trames ISL peuvent atteindre une taille de 1522 octets, quatre octets au-dessus de la limite autorisée sur Ethernet, qui est de 1518 octets. Certains nœuds peuvent donc les rejeter parce qu'ils les voient comme des trames trop longues (appelées trames géantes, *giant frames*).

Le protocole VTP

L'administration des VLAN s'effectue au moyen du protocole **VTP** (*VLAN Trunk Protocol*), dont il existe 3 versions. C'est un *messaging protocol*, c'est-à-dire un protocole qui travaille en gérant des échanges de messages. Ces échanges d'informations s'effectuent entre les switches et les routeurs qui se trouvent dans le **domaine VTP** (*VTP domain* ou *VLAN management domain*). Ils concernent l'ajout, la suppression et le changement de nom des VLANs. Un switch ne peut faire partie que d'un seul VLAN.

Un switch peut être configuré pour fonctionner dans l'un des modes VTP suivants :

- **mode serveur secondaire** (*secondary server*, mode par défaut) : le switch met les autres *switches* du domaine VTP au courant de sa configuration (*configuration advertising*) et la synchronise avec les leurs, mais il ne peut pas initier un changement de configuration ;
- **mode serveur primaire** (*primary server*) : le switch met les autres switches du domaine VTP au courant de sa configuration (*configuration advertising*) et la synchronise avec les leurs ; il est capable d'initier un changement de configuration ; on n'a qu'un seul serveur primaire par domaine VTP ;
- **mode client** : le switch met les autres switches du domaine VTP au courant de sa configuration et la synchronise avec les leurs pour autant que cela n'implique pas la création, la suppression ou le changement de nom de VLANs ; contrairement aux serveurs, un client n'enregistre pas la configuration VTP en NVRAM ;
- **mode transparent** : le switch ne participe pas à VTP (mais, sous VTP 2, il transmet les informations qu'il reçoit).

off mode : le switch ne participe pas à VTP.

En mode serveur, le switch peut générer ou modifier une configuration VTP, alors qu'en mode client, il se limite à accepter la configuration qu'il reçoit d'un serveur. Il ne peut ni la générer ni la modifier.

En VTP 3, la configuration de VTP n'est possible que sur un *primary server*.

On peut minimiser le trafic VTP sur le domaine au moyen d'une fonction appelée *VTP pruning* (*to prune* signifie tailler, élaguer). Par défaut, elle est désactivée.

VTP se configure sous IOS en mode privilégié. Par exemple :

```
Console> en                                          ← pour entrer dans le mode privilégié
Console>(enable) set vtp domain DOMVTP               ← pour créer le domaine DOMVTP
Console>(enable) set vtp spantree macreduction enable
Console>(enable) set vtp version 3                   ← pour activer VTP 3
This command will enable VTP version 3 on this switch
Do you want to continue (y/n) [n]? y
VTP3 domain DOMVTP modified
Console>(enable) set vtp mode server                 ← pour activer le mode serveur
Changing VTP mode for all features
VTP3 domain DOMVTP modified
Console>(enable) show vtp domain                     ← pour vérifier la configuration
```

Pour permettre à deux VLANs de communiquer, on doit utiliser un routeur ou un switch de niveau 3, et il faut soit qu'il ait une interface pour chaque VLAN, soit qu'il soit doté des fonctions ISL (*Inter-Switch Link*).

Questions

1. Vous placez un port dans un VLAN qui n'existe pas. Comment réagit le switch ?

   - ☐ Il crée le VLAN en silence
   - ☐ Il demande s'il doit créer le VLAN
   - ☐ Il interroge les autres switches du réseau pour savoir s'ils connaissent le VLAN
   - ☐ Il affiche un message d'erreur.

2. Quels sont trois avantages des VLANs ? Cochez trois propositions.

   - ☐ Les VLANs établissent des domaines de diffusion dans les réseaux commutés
   - ☐ Les VLANs utilisent le filtrage de paquets pour améliorer la sécurité du réseau
   - ☐ Les VLANs fournissent une méthode pour économiser les adresses IP dans les grands réseaux
   - ☐ Par rapport aux réseaux à routeurs, les VLANs fournissent une solution de rechange pour la mise en interréseau à faibles temps d'attente
   - ☐ Les VLANs donnent accès aux services de réseaux en se basant sur les départements et non la localisation physique
   - ☐ Les VLANs simplifient considérablement l'ajout, le déplacement ou le changement d'hôtes sur le réseau.

3. Quels sont trois avantages des VLANs ? Cochez trois propositions.

   - ☐ Ils augmentent la taille des domaines de diffusion
   - ☐ Ils permettent le groupage logique des utilisateurs par fonctions, pas par location ou géographie
   - ☐ Ils peuvent améliorer la sécurité des réseaux
   - ☐ Ils augmentent la taille des domaines de diffusion tout en diminuant le nombre des domaines de collision
   - ☐ Ils augmentent le nombre des domaines de diffusion tout en diminuant la taille des domaines de collisione
   - ☐ Ils simplifient l'administration des switches.