

Administration de routeurs CISCO

Table des matières

1 Composants des routeurs	2
1.1 Ports d'administration	2
1.2 Interfaces réseau	3
1.3 Emplacements d'extension	5
1.3.1 Cartes WIC	5
1.3.2 Cartes NM	6
1.4 Composants de la carte mère	6
1.5 Processus de démarrage des routeurs	8
2 Logiciels du routeur : IOS et CLI	10
2.1 Les modes principaux de la CLI	10
2.1.1 Mode utilisateur	10
2.1.2 Mode privilégié	10
2.1.3 Mode de configuration globale	10
2.1.4 Mode de configuration d'interface	11
2.1.5 Changements de mode	11
2.2 Annulation d'une commande	12
2.3 Messages d'information	12
2.4 Pagination des messages	13
2.5 L'aide à la saisie	13
2.6 Moyens d'accéder à la CLI	15
3 Configuration d'une interface Ethernet	17
4 Minicom pour accéder à la console	21
4.1 Câblage du port console au port série	21
4.2 Le logiciel minicom	22
4.2.1 Paramétrage de minicom	22
4.2.2 Utilisation	23
5 Sauvegarde/restauration de la configuration avec TFTP	25
5.1 Dépôt d'un fichier du routeur sur le serveur TFTP	26
5.2 Récupération sur le routeur d'un fichier du serveur TFTP	28

1 Composants des routeurs

Les routeurs dont nous disposons pour les TP sont principalement des modèles des séries 1700 et 2600 (figure 1). Ce sont des routeurs "professionnels", destinés aux entreprises. Ce genre de routeurs disposent d'interfaces réseaux intégrées, de différents types. Les routeurs des séries 1700 et 2600 sont modulaires. Ils disposent en plus d'emplacements pour des interfaces réseaux additionnelles. L'ensemble de ces interfaces permettent de router les datagrammes IP sur différents réseaux.

Nous commencerons par décrire les ports d'administration et les interfaces de nos routeurs avant d'aborder leur fonctionnement.



(a) Cisco série 1700



(b) Cisco série 2600

FIGURE 1 – Vue avant de quelques routeurs CISCO

1.1 Ports d'administration

Nos routeurs CISCO, comme de nombreux équipements réseau, **ne possèdent ni clavier ni moniteur**. Ils ne peuvent donc être administrés que via un équipement externe.

Pour cela, ils disposent de deux ports dédiés à son administration, dont une vue (sur un 1721, l'un des modèles de la série 1700) est présentée dans la figure 2 :

- le **port console** à connecteur RJ-45, servant à l'administrer via une liaison série.
- le **port auxiliaire** (AUX) à connecteur RJ-45, destiné en principe à être connecté à un modem pour une administration à distance.

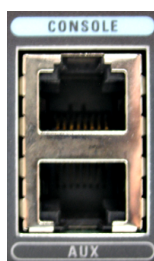


FIGURE 2 – Ports console et AUX d'un 1721. Nous n'utiliserons que le port console (en haut), signalé par le texte CONSOLE, sur fond bleu ciel.

 Nous n'utiliserons pas le port AUX. Seul le port console nous sera utile.

Le port console servira à administrer le routeur via une liaison au port série d'un PC. Comme détaillé en section 4.2, sous Linux le logiciel **minicom** gère cette liaison pour offrir un terminal sur le routeur pour taper des commandes d'administration, obtenir des informations, etc. La figure 3 donne un aperçu d'un dialogue avec le routeur avec **minicom**.

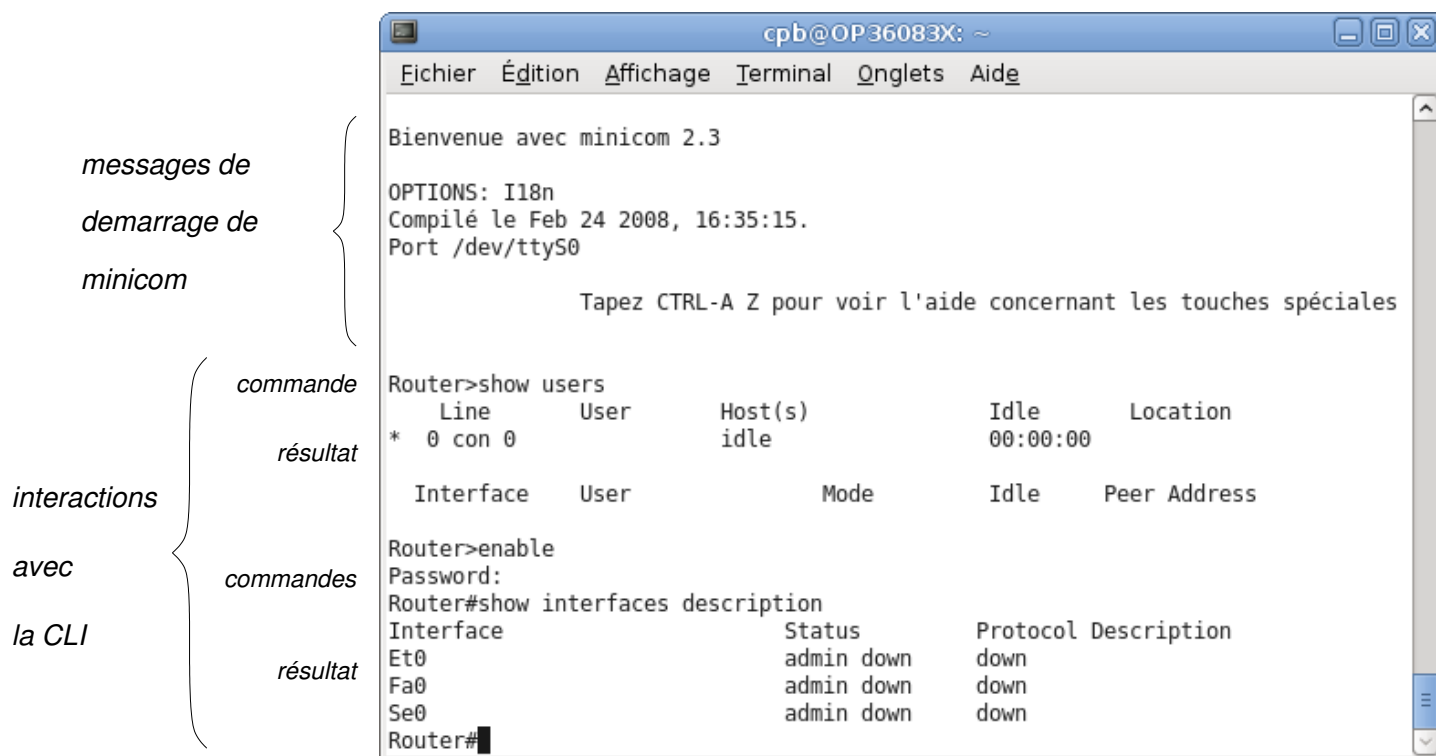
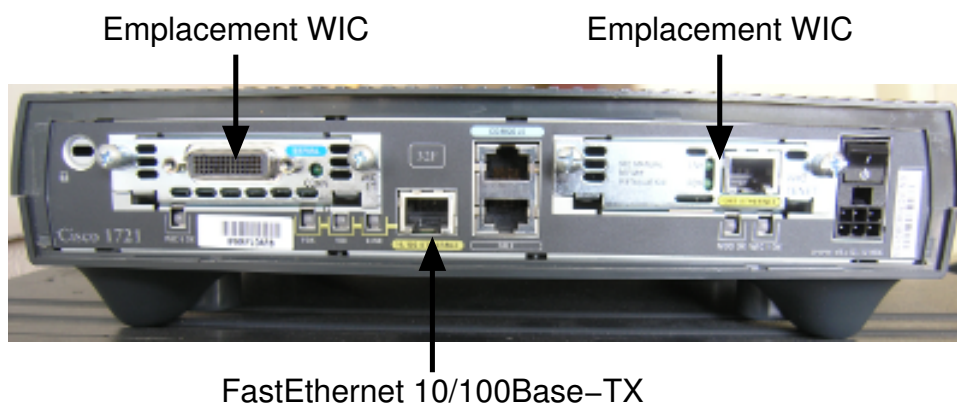


FIGURE 3 – Administration d'un routeur par le port console en utilisant **minicom**

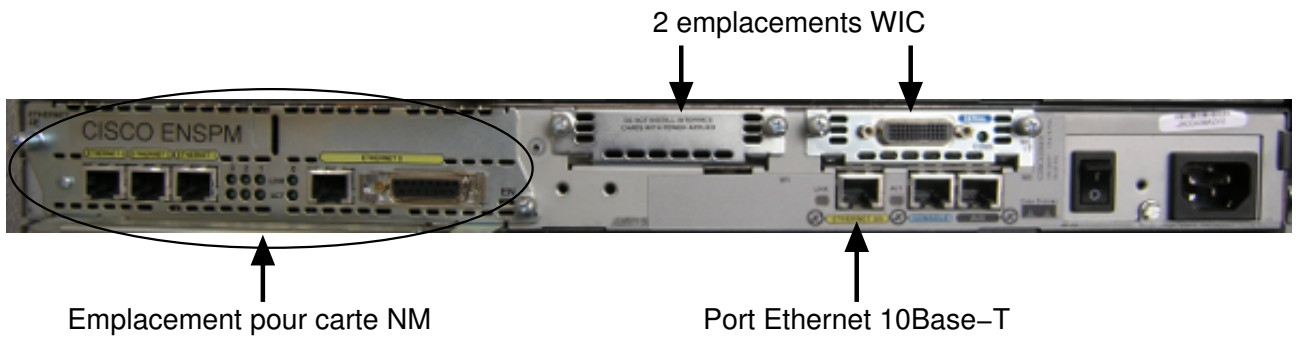
1.2 Interfaces réseau

Les interfaces réseau et les emplacements d'extension (voir section suivante) possibles dépendent du modèle :

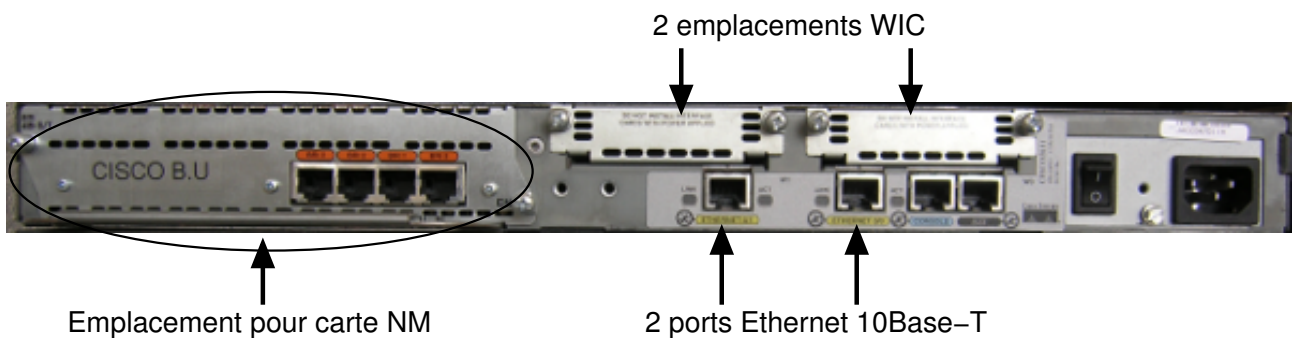
- modèles 1720 et 1721 : ils disposent d'un port **FastEthernet** (10/100Base-TX) et de deux emplacements (*slots*) pour cartes d'extension WIC (*Wan Interface Card*) :



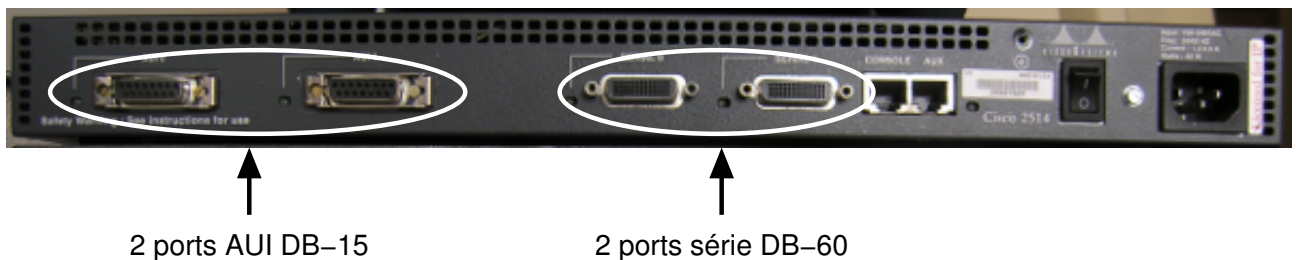
- modèle 2610 : il dispose d'un port **Ethernet 10Base-T**, de deux slots pour carte WIC, et d'un slot pour module réseau NM :



- modèle 2611 : il dispose de deux ports **Ethernet 10Base-T**, de deux slots pour carte WIC, et d'un slot pour module réseau NM :



- modèle 2514 (matériel obsolète cité pour information) : il dispose de 2 ports AUI (Attachment Unit Interface) Ethernet à connecteurs DB-15 et de 2 ports série à connecteurs DB-60 :



En branchant sur un port AUI un transceiver 10Base-T, on obtient une interface Ethernet 10 Mbps Half-Duplex à connecteur RJ-45 classique. Voici un exemple de ce transceiver :



❗ Selon les arrivages ; -), des routeurs d'autres types pourraient être disponibles.


1.3 Emplacements d'extension


Les emplacements d'extension de nos routeurs sont de 2 types : pour carte WIC et pour carte NM. Les 1700 n'ont que des slots WIC, alors que les 2600 ont aussi un slot NM. Les 2500 n'ont aucun emplacement d'extension.

1.3.1 Cartes WIC

Les cartes WIC (*WAN Interface Card*) s'enchâssent dans les slots WIC du routeur et lui ajoutent une ou plusieurs interfaces d'un certain type. Voici quelques cartes WIC possibles :

- **WIC-1T** : 1 port série, asynchrone et synchrone (T1/E1)
- **WIC-2T** : 2 ports série, asynchrones et synchrones (T1/E1)
- **WIC-1B-S/T** : 1 port BRI S/T (Basic Rate Interface) pour Réseau Numérique à Intégration de Service
- **WIC-1ADSL** : 1 port ADSL
- **WIC-1SHDSL** : 1 port SHDSL (Symmetrical High-Bit Rate DSL)
- **WIC-4ESW** : Switch configurable 4 ports Ethernet (10/100BASE-TX)
- **WIC-1ENET** : carte présentée dans la figure 4 qui fournit 1 port Ethernet (10BASE-T) pouvant accepter le full-duplex ;
- ...et il y en a beaucoup d'autres

 Nos routeurs disposent de types variés de cartes mais on se servira uniquement des cartes WIC-1ENET et/ou WIC-1T et/ou WIC-2T. Certains peuvent disposer d'une carte WIC-4ESW qui fournit un switch 4 ports Ethernet.

 Généralement, les cartes WIC des 1700 sont aussi compatibles avec les routeurs des séries 2600, 3600 et 3700. Mais il y a des exceptions. Par exemple, les cartes WIC-1ENET ne sont pas supportées par les 2600.

De façon générale, avant d'investir dans une carte d'extension (WIC, NM ou autre), l'administrateur doit vérifier qu'elle est supportée par son matériel, ainsi que par la version d'IOS dont il dispose.



FIGURE 4 – Carte WIC-1ENET fournissant un port Ethernet 10 Base-T pouvant accepter le Full-Duplex.

1.3.2 Cartes NM

Les cartes NM (*Network Module*) sont des modules réseau plus grands que les WIC et peuvent doter le routeur de nombreuses interfaces (par exemple, Switch 16 ports Ethernet avec le NM-ESW16). Comme les WIC, il existe de nombreuses cartes NM. Nous utiliserons les cartes suivantes :

- **NM-1E** (*One-port 10BASE-T Ethernet interface*) : ce module a la particularité d'offrir le choix entre 2 ports pour une seule interface : 1 port RJ-45 et 1 port AUI. On peut utiliser n'importe lequel, mais pas en même temps.



- **NM-4E** (*4-port 10BASE-T Ethernet interface*) : ce module étend le module précédent en lui ajoutant 3 autres interfaces Ethernet 10Base-T, chacune avec un port RJ45.



❗ Certains 2600 sont équipés d'autres cartes NM (par exemple, NM BRI 4B-S/T et NM ESW-16) mais elles ne nous seront pas utiles ¹.

1.4 Composants de la carte mère

En interne, nos routeurs comportent les composants principaux suivants, dont une partie est visible sur la vue d'une carte mère d'un 1720 de la figure 5 :

- **CPU** (*Central Processing Unit*) : le microprocesseur est responsable de l'exécution du système d'exploitation du routeur. Sur les routeurs CISCO, ce système est l'**IOS** (*Internetwork Operating System*). L'IOS prend en charge les protocoles et fournit l'interface de commandes **CLI** (*Command Line Interface*) accessible via une liaison série ou une session terminal Telnet ou SSH. De la puissance du CPU dépend la capacité de traitement du routeur. Les processeurs des routeurs utilisés sont des processeurs RISC (*Reduced Instruction Set Computing*) :
 - ◇ pour les séries 1700 et 2600 : Motorola MPC860 cadencés à au plus 80 MHz
 - ◇ pour la série 2500 : Motorola 68030 cadencé à au plus 50 Mhz
- **Mémoire ROM** : mémoire permanente contenant le code pour réaliser les diagnostics de démarrage du matériel (POST, *Power On Self Test*) ainsi que le code pour le chargement (*System Bootstrap*) de l'image de l'IOS vers la RAM et la mise en route du système. Elle contient aussi un système minimal (ROMMON) et une interface de commandes simplifiée permettant de réaliser des opérations de secours : modification de la phase de boot, téléchargement par TFTP d'une image suite à une corruption de la flash, etc.

1. La NM ESW-16 est difficilement supportée par les 2600 : elle nécessite une importante capacité flash et RAM pour exécuter un IOS qui la supporte.

- **Mémoire Flash** (EPROM) : sous forme d'une barrette (*Flash SIMM*) ou d'une carte Flash (*Flash Card*). Elle se présente dans l'IOS comme l'unité de stockage **flash:**. Elle est utilisée pour stocker un (ou plusieurs) fichier contenant l'image de l'IOS, souvent compressée. On peut aussi y ajouter d'autres fichiers. La mémoire flash est en quelque sorte une EPROM (*Erasable Programmable Read-Only Memory*) —une ROM effaçable et programmable. Pour nos modèles, la taille de la flash n'excède pas quelques dizaines de méga-octets.
- **NVRAM** : c'est une mémoire non volatile, de petite taille (souvent 32 Ko). Elle aussi se présente dans l'IOS comme une unité de stockage **nvr:**. Elle contient la configuration, en particulier le fichier de configuration de démarrage **startup-config** et le **registre de configuration** (*configuration register*). Elle peut aussi contenir des sauvegardes de la configuration.
- **RAM** : mémoire volatile dans laquelle sont chargées toutes les informations permettant au matériel de fonctionner (tampons, tables de routage, table ARP, table de VLANs, ...). L'IOS est chargé en RAM au démarrage du matériel. Il réalise ensuite une copie du fichier startup-config (à partir de la NVRAM) dans la RAM. Cette copie est nommée **running-config** : la configuration courante du routeur. La modification de la configuration du routeur est immédiatement appliquée mais n'affecte que la running-config. Elle n'est que temporaire car la RAM est effacée chaque fois que le routeur (re)boote et *a fortiori* lorsque l'alimentation est coupée. Pour rendre une modification permanente, il faut copier la running-config (en RAM) dans la startup-config (en NVRAM).

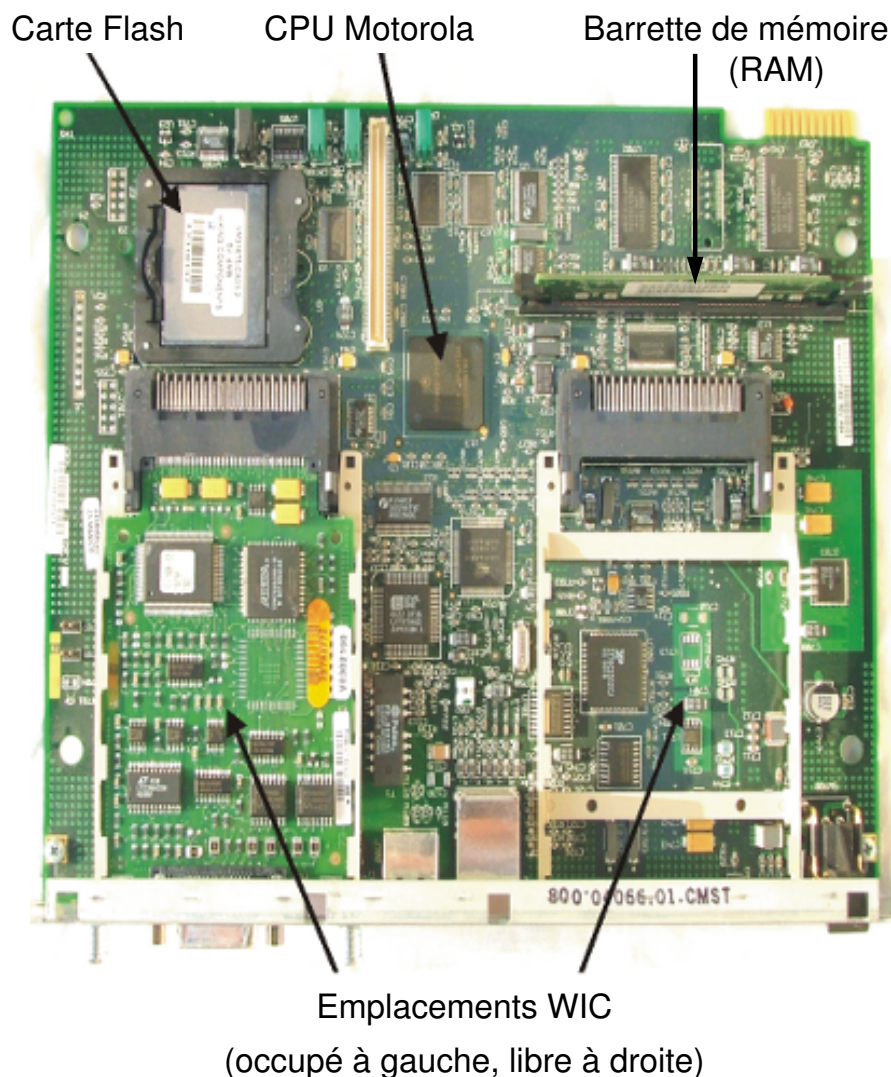



FIGURE 5 – Vue d'une carte mère d'un 1720 (série 1700)

1.5 Processus de démarrage des routeurs

Ce processus est en partie schématisé par la figure 6. Au démarrage, le programme de la ROM (BootROM) d'un routeur CISCO va initialiser le matériel et exécuter le POST (*Power On Self Test*), effectuant les diagnostics de démarrage.

Après le POST, le **registre de configuration** est examiné pour :

- déterminer comment le routeur doit démarrer : démarrage normal ou dans un autre mode tel que ROMMON (mode monitoring de la ROM), RxBoot (IOS simplifié stocké en ROM sur d'anciens modèles, tels que les 2500) ou NetBoot (démarrage par téléchargement du système par TFTP²)
- définir les options à utiliser pendant le démarrage. Par exemple, ignorer la configuration de démarrage (*startup-config*), désactiver les messages de démarrage,...
- configurer la vitesse de la console (*baud rate*)

 Ce registre de configuration (*configuration register*) est stocké dans la NVRAM et contient une valeur sur 16 bits. Les différents bits du registre indiquent au routeur les paramètres de démarrage. Il est modifiable aussi bien en mode ROMMON qu'à travers la CLI de l'IOS. Nous serons amenés à le modifier en TP afin de prendre le contrôle du routeur et réinitialiser sa configuration.

Le cas normal (valeur standard du registre de configuration) correspond à la partie centrale inférieure de la figure 6, où :

1. le BootROM charge l'image IOS stockée dans la carte flash vers la RAM (en la décompressant si nécessaire);
2. le BootROM cède sa place à l'IOS (le système d'exploitation effectif) qui démarre
3. Une fois initialisé, l'IOS copie la configuration stockée dans la NVRAM (*startup-config*) pour l'utiliser en RAM comme configuration courante (*running-config*). Une autre possibilité (très pratique en cas de mauvaise manipulation) est de récupérer cette configuration auprès d'un serveur TFTP, si l'administrateur prudent l'y a sauvegardé...
4. le routeur est alors opérationnel et actif

Les démarrages en ROMMON ou en RxBoot (obsolète) chargent un système minimal placé en ROM, permettant l'exécution de quelques commandes via une interface de commandes basique. Ces modes sont utilisés pour des opérations de secours, telles que :

- formatage/gestion de la carte flash dont le système de fichier est endommagé (suite à un incident) provoquant l'impossibilité de démarrer normalement
- téléchargement d'une image de l'IOS dans la flash (le plus souvent par TFTP)
- modification du registre de configuration pour régler de nouveaux paramètres de démarrage

2. *Trivial File Transfer Protocol* (TFTP) est un protocole très simple de téléchargement de fichier, ne nécessitant aucune identification. Il est souvent utilisé dans l'administration des matériels réseau. Notamment en ROMMON, RxBoot ou NetBoot car ces modes ne disposent que du système minimal contenu dans la ROM, qui a une faible capacité et peut difficilement contenir le code de protocoles plus évolués.



FIGURE 6 – Processus de démarrage d'un routeur CISCO

i Si vous avez compris le rôle des fichiers startup-config et running-config, vous devriez comprendre aussi pourquoi il est conseillé de sauvegarder régulièrement la configuration actuelle qui se trouve dans la RAM, idéalement après chaque étape de configuration du routeur. Il suffit simplement de faire une copie de running-config vers startup-config pour ne pas perdre la configuration stockée dans la RAM.

Il est encore plus prudent (et recommandé) de sauvegarder la configuration en lieu sûr, en copiant la startup-config (ou la running-config) via un réseau TCP/IP sur un serveur TFTP, ce qui est très simple à réaliser sur ces routeurs. En cas de problème, par exemple panne de routeur nécessitant son remplacement, la configuration pourra être téléchargée sur le routeur.

2 Logiciels du routeur : IOS et CLI

La CLI (*Command Line Interface*) est l'interface entre l'utilisateur ou l'administrateur et l'IOS (*Internetwork Operating System*). Par analogie aux systèmes Unix, la CLI est un shell (assez basique), alors que l'IOS est le système d'exploitation. La CLI lit des commandes utilisateur et demande à l'IOS de réaliser les actions correspondantes. Si le routeur est vierge (sans configuration), le seul moyen d'accéder à la CLI pour l'administrer est de se connecter à son port console via une liaison série, gérée par exemple avec **minicom** sur un PC sous Linux (voir figure 3 et section 4.2).

2.1 Les modes principaux de la CLI

2.1.1 Mode utilisateur

Lorsque le routeur (ou le switch) CISCO démarre normalement et a chargé l'IOS, il affiche un prompt de la forme :

```
nom-du-routeur>
```

où *nom-du-routeur* est le nom d'hôte (*hostname*) du routeur et est configurable. Si le nom d'hôte n'a jamais été modifié, le prompt est :

```
Router>
```

Le symbole > indique qu'on se trouve en **mode utilisateur**, où l'on a accès à un nombre limité de commandes, *a priori* inoffensives, permettant simplement d'observer l'état du routeur. Bien que ce soit rarement le cas, il est possible de protéger l'accès à ce mode par un mot de passe.

2.1.2 Mode privilégié

Pour effectuer l'administration et la configuration du routeur il faut d'abord passer en **mode privilégié** (auss appelé mode EXEC). En mode utilisateur, la commande **enable** fait passer au mode privilégié. L'inverse se fait avec la commande **disable**. On sait qu'on se trouve en mode privilégié grâce au prompt : le > est remplacé par le symbole #. En mode privilégié le prompt aura donc la forme :

```
nom-du-routeur#
```

Le mode privilégié permet notamment de copier des fichiers (manipulations de la flash, de la NVRAM et de la RAM) et d'obtenir certaines informations qui doivent rester inaccessibles à toute tierce personne, comme par exemple la configuration (startup-config ou running-config) qui contient les mots de passe (cryptés ou non).

2.1.3 Mode de configuration globale

Pour modifier des paramètres globaux réseaux du routeur (par exemple, activation de certains protocoles), il faut entrer en **mode de configuration globale**, accessible à partir du mode privilégié en tapant :

```
nom-du-routeur#configure terminal
```

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

L'accès à ce mode modifie le prompt qui devient :

```
nom-du-routeur(config)#
```

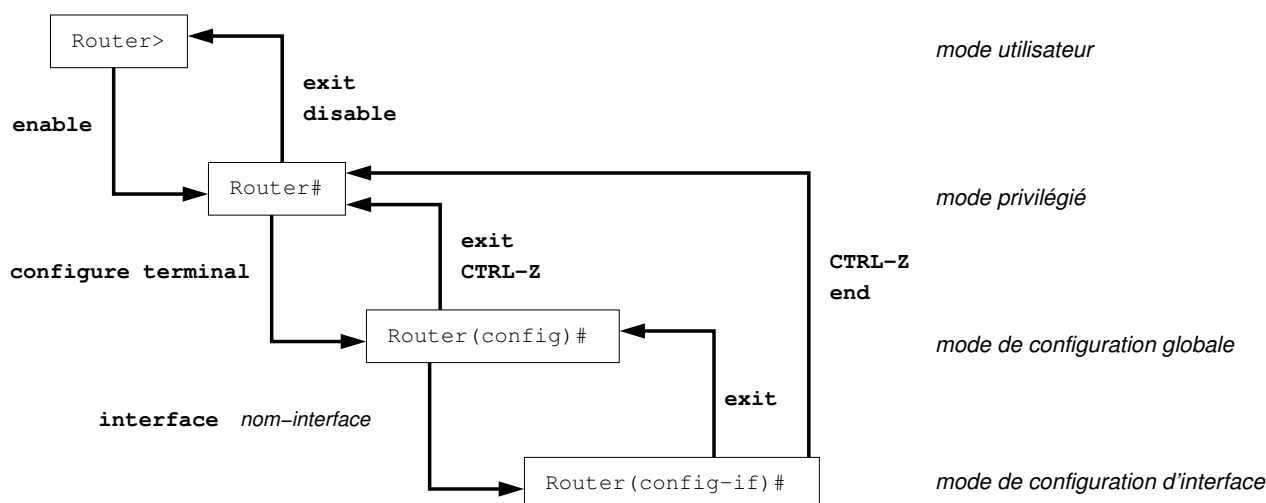


FIGURE 7 – Les principaux modes de la CLI

2.1.4 Mode de configuration d'interface

Pour modifier la configuration d'une des interfaces réseau du routeur, il faut passer en **mode de configuration d'interface**, accessible à partir du mode de configuration globale en tapant :

```
nom-du-routeur(config)#interface nom-interface
```

où *nom-interface* est le nom qu'a donné l'IOS à l'interface qu'on veut configurer. Ce peut être **Ethernet0**, **FastEthernet0**, **Serial0**, **Ethernet0/0**, etc.

- ❗ Le nom IOS des interfaces du routeur pourra être obtenu en utilisant, en mode utilisateur ou privilégié, la commande :

```
Router>show interfaces
```

En mode de configuration d'interface, le prompt change à nouveau et l'on a accès aux commandes permettant de la configurer :

```
nom-du-routeur(config-if)#
```

2.1.5 Changements de mode

L'accès aux modes précédents se fait dans l'ordre : utilisateur puis privilégié puis configuration globale puis configuration de l'interface.

- ✍ Il existe d'autres modes plus spécifiques, comme par exemple le mode de configuration des adresses gérées par le serveur DHCP du routeur, qu'on utilisera aussi au cours du TP.

Pour revenir d'un mode au mode de niveau inférieur, on utilise la commande **exit** (en mode privilégié, elle a le même effet que **disable**). Pour revenir au mode privilégié à partir de n'importe quel mode auquel il mène, on peut utiliser le raccourci **CTRL-Z** ou la commande **end**.

La figure 7 résume les différents modes de la CLI de CISCO et indique à chaque fois les commandes utilisées pour passer d'un mode à un autre.

2.2 Annulation d'une commande

Lorsque dans un mode, on a tapé et validé une commande, on pourra le plus souvent l'annuler en la faisant précéder du mot clé **no**.

Exemple 1

Dans le mode de configuration du terminal, la commande

```
Router(config)#ip http server
```

active un serveur Web sur le routeur, alors que

```
Router(config)#no ip http server
```

le désactive.

En mode de configuration d'interfaces, la commande


```
Router(config-if)#shutdown
```

désactive une interface réseau, alors que

```
Router(config-if)#no shutdown
```

l'active.



 Il sera souvent bien pratique dans ce cas d'utiliser le rappel de l'historique (voir section 2.5), surtout si la commande à annuler est un peu longue ou complexe.

2.3 Messages d'information

Lorsqu'on administre le routeur par son port console, il arrive que le routeur affiche sur la CLI des messages d'information, par exemple lorsqu'il vient de détecter qu'une interface réseau est connectée. Ces messages commencent par % (et sont éventuellement précédés d'une date) comme :

```
%LINEPROTO-5-UPDOWN : Line Protocol on Interface FastEthernet0/0, changed state to up.
```

ou encore :

```
*Mar 1 00:00:04.259: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
```

Ces messages masquent parfois le prompt de la CLI. Pour le faire réafficher, taper simplement sur **CTRL-L** (ou **Entrée** si la ligne de commande est vierge). La combinaison **CTRL-C** peut être aussi utilisée pour réafficher le prompt sans tenir compte de ce qu'on vient de taper, notamment lorsqu'on se trompe en saisissant une commande.



Lorsque vous revenez au mode privilégié, attendez quelques secondes avant de taper de nouvelles commandes car le routeur va afficher des messages d'information. Si l'on est en train de taper une commande, ces messages sont très gênants. Mais il suffit de taper **CTRL-L pour faire réafficher le prompt et ce qu'on a déjà tapé.**


2.4 Pagination des messages

Lorsque la CLI doit afficher plus de 25 lignes d'un coup, elle pagine cet affichage³. Dans ce cas, la dernière ligne affiche quelque chose comme :

```
--More--
```

ce qui signifie que la CLI attend pour afficher la suite. On peut alors taper :

- `Espace` pour passer à la page suivante
- `Entrée` pour voir la ligne suivante
- `q` ou `CTRL-C` pour arrêter l'affichage

 On remarquera qu'il n'est pas possible de revenir en arrière sur le texte affiché...

2.5 L'aide à la saisie

La CLI dispose essentiellement de trois fonctionnalités qui facilitent la saisie de commandes : le rappel de l'historique, une aide contextuelle et le remplissage (*completion*) automatique.

Rappel de l'historique

Le **rappel de l'historique** est la possibilité de rappeler les dernières saisies en les faisant défiler en utilisant les flèches du haut (vers les plus anciennes) et du bas (vers les plus récentes). Chaque mode dispose de son propre historique : dans un mode donné, on ne rappelle que les commandes tapées dans ce mode. En principe, l'historique est activée par défaut. Sinon, en mode privilégié, il faut utiliser la commande :

```
Router#terminal history [nombre-de-lignes]
```

où *nombre-de-lignes* est optionnel et spécifie le nombre de lignes à retenir (par défaut 10).

 Le rappel de l'historique est très utile pour annuler une commande complexe.

Aide contextuelle

L'**aide contextuelle** nous permet de savoir quelles sont les commandes disponibles et, pour chacune, quels sont ses paramètres possibles. Elle intervient lorsque l'on tape le caractère ? :

- `?` seul fait afficher la liste des commandes disponibles dans le mode courant, avec une courte description ;
- si l'on a déjà commencé à taper un mot, `?` fait afficher la liste des mots attendus, commençant par ce qu'on a tapé ;
- lorsqu'on a tapé une commande, à chaque début de mot, on peut taper `?` pour faire afficher les paramètres possibles à cette position ainsi qu'une courte description. Certains paramètres seront à remplacer par une valeur, notamment :
 - ◇ **A.B.C.D** devra être remplacé par une adresse IP ;
 - ◇ **WORD** devra être remplacé par un mot (suite de caractères) dont la signification dépend du contexte ;
 - ◇ **<cr>** représente le retour à la ligne, ce qui veut dire qu'on peut terminer la commande sans avoir à taper autre chose.

3. Ceci parce que le routeur doit pouvoir être administré par un terminal texte 80 colonnes, 25 lignes.

Exemple 2

En mode privilégié, si l'on tape :

```
Router#?  
Exec commands:  
  access-enable    Create a temporary Access-List entry  
  access-profile   Apply user-profile to interface  
  access-template  Create a temporary Access-List entry  
  ...
```

⇒ alors la liste des commandes du mode privilégié est affichée. Comme tous les affichages, elle est paginée si elle excède 25 lignes

```
Router#pi?  
ping
```

⇒ une seule commande commence par pi, c'est **ping**

```
Router#ping ?  
WORD  Ping destination address or hostname  
ip     IP echo  
ipv6   IPv6 echo  
tag    Tag encapsulated IP echo  
<cr>
```

⇒ <cr> indique que les paramètres sont optionnels, **WORD** doit être remplacé par une adresse ou un nom d'hôte, alors que les autres possibilités sont à taper

```
Router#ping 139.124.187.4 ?  
<cr>
```

⇒ <cr> indique qu'il n'y a plus rien à entrer



Remplissage automatique

Le **remplissage automatique** permet de se contenter de ne taper que les débuts des mots lorsqu'il n'y a pas d'ambiguïté. Par exemple, en mode privilégié taper :

```
Router#conf t
```

est synonyme de **configure terminal** car il n'y a qu'une seule commande commençant par **conf** et uniquement **terminal** comme paramètre de cette commande qui commence par **t**. De même, lorsqu'on tape un début de mot, on peut le faire compléter par la CLI s'il n'y a pas d'ambiguïté en tapant sur Tabulation. Dans l'exemple précédent, taper **conf**Tabulation fait compléter par **configure**.

2.6 Moyens d'accéder à la CLI

Ne possédant ni clavier ni de moniteur, nos routeurs CISCO ne peuvent être administrés que via un équipement externe.

i Ces routeurs, comme de nombreux routeurs et switch de type "grand public" ou SOHO (*Small Office Home Office*), peuvent activer un serveur HTTP permettant leur administration via une interface Web. Le point faible de ce type d'interface graphique est qu'elle n'est conçue que pour les cas d'utilisation les plus fréquents et qu'elle ne permet pas un réglage fin de la configuration.

Si la configuration souhaitée n'est pas prévue dans l'interface⁴, il devient nécessaire de configurer le routeur à travers son interface de commandes en ligne (la CLI sur l'IOS CISCO). Cette interface n'est généralement pas disponible sur le matériel destiné au "grand public", alors que les administrateurs des routeurs "professionnels" privilégient son utilisation.


Pour accéder à l'interface en ligne de commandes CLI, l'administrateur a plusieurs possibilités :


- se connecter par SSH : cela suppose que le routeur est opérationnel au niveau TCP/IP, que la version de l'IOS prenne en charge SSH et que le routeur ait été configuré pour permettre cet accès ;
- se connecter par TELNET : cela suppose que le routeur est opérationnel au niveau TCP/IP et ait été configuré pour permettre cet accès ;
- se connecter à distance par l'intermédiaire d'un équipement (généralement un modem) relié au port AUX : cela suppose que l'administrateur ait mis en place cette liaison et configuré le routeur pour permettre cet accès ;
- **se connecter au port "console" du routeur** : ce port est une interface série qui, comme son nom l'indique, est destinée à l'administration du routeur. C'est l'interface de secours, toujours opérationnelle (sauf en cas de défaillance grave matérielle). En contrepartie, il faut avoir accès physiquement au routeur, alors que les possibilités précédentes permettent un accès à distance.

Le port console est indispensable dans de nombreuses situations :

- le matériel est dans son état "sortie d'usine", c-à-d. sans configuration réseau. La console est donc le seul moyen d'accéder à l'interface d'administration du routeur. L'administrateur pourra ensuite le configurer et permettre son administration à distance ;
- le matériel a été configuré et intégré dans un réseau, mais suite à une erreur de configuration, le dispositif n'est plus joignable via le réseau. Il n'est pas rare en effet qu'en modifiant la configuration, l'administrateur finisse par s'interdire l'accès distant au routeur (par exemple en activant un filtre TCP/IP qui bloque le protocole utilisé pour l'administration à distance) ;
- le mot de passe pour accéder à l'interface d'administration du matériel a été oublié ou le matériel acheté est d'occasion avec un mot de passe inconnu. L'interface console permettra de ré-initialiser le mot de passe.

4. Par exemple, la configuration d'un accès DSL via IPoATM (*IP over ATM*) pour "Free dégroupé" sur un routeur CISCO 827H ou 1720

 Le port console est souvent présent sur du matériel réseau qualifié de "professionnel" (souvent cher et seulement disponible chez les fournisseurs de matériel réseau pour les entreprises), alors qu'il ne l'est pas sur du matériel destiné au "grand public" (souvent bon marché et assez bas de gamme). Notamment, le matériel Linksys (filiale de CISCO pour le grand public), n'en possède généralement pas.

 **Il existe une procédure permettant de prendre le contrôle d'un routeur dont le mot de passe est inconnu. Quiconque peut l'appliquer s'il a accès au port console du routeur. C'est pour cela qu'il est extrêmement important que les routeurs aient également une sécurité physique pour empêcher tout accès non autorisé, par exemple en les installant dans des locaux fermés et sécurisés.**

3 Configuration d'une interface Ethernet

Une interface réseau peut se configurer aux niveaux (couches OSI) 1-2 (pour la partie Liaison/trame) et 3 (partie IP). Il existe plusieurs types d'interfaces physiques, mais nous nous intéresserons ici principalement à Ethernet.

Aux niveaux 1 et 2, nous devons prendre en considération le matériel que nous utilisons, sachant :

- qu'il y a différents types de hub/switch (10 Mbit/s en half-duplex, FastEthernet en full-duplex) ;
- que les routeurs possèdent différents types d'interfaces des routeurs (FastEthernet 100Base-TX full-duplex ou à auto-négociation, Ethernet 10Base-T half-duplex ou full-duplex) ;

L'idéal est que les interfaces du routeur fonctionnent au mieux de leurs capacités : à leur plus haut débit et en full-duplex si possible. Alors que les cartes Ethernet modernes détectent les caractéristiques de la liaison Ethernet et s'adaptent à cette liaison, ce n'est pas le cas de la plupart des cartes de nos routeurs. Il est nécessaire d'adapter les caractéristiques physiques des interfaces Ethernet à l'équipement utilisé.

Pour cela, nous disposons de commandes pour la configuration de niveaux 1 et 2 dans le mode configuration de l'interface, précisées dans ce qui suit.

i Selon l'interface (et l'IOS), certaines de ces commandes peuvent ne pas être disponibles. Dans ce cas, les caractéristiques correspondantes ne sont pas modifiables.

Configurations de niveaux 1 et 2 pour tout type d'interface (pas uniquement Ethernet) :

- Router(config-if)#**full-duplex** *(ou selon le cas, **duplex full**)*

pour placer la liaison en mode *full-duplex*

- Router(config-if)#**half-duplex** *(ou selon le cas, **duplex half**)*

pour plaer la liaison en mode *half-duplex*

- Router(config-if)#**no shutdown**

pour activer la liaison. Cela a pour effet de rendre l'interface administrativement active (**up**) et d'essayer de rendre la liaison opérationnelle (**up**).

i Notons que selon le cas, il est possible de laisser le routeur configurer automatiquement le mode duplex, par négociation. Cela se fait en configurant l'interface avec :

```
Router(config-if)#duplex auto
```

mais comme nous ne ferons aucune configuration automatique, nous ne l'utiliserons pas...

Configuration de niveaux 1 et 2 spécifique à Ethernet :

- Router(config-if)#**speed** *débit*

pour fixer le débit de la liaison. Dans notre cas, *débit* devrait valoir 10 ou 100. Utiliser l'aide de la CLI pour voir les paramètres disponibles.



Il est inutile d'essayer de faire fonctionner une carte Ethernet 10BaseT en 100 Mbit/s!!



Notons à nouveau que dans certains cas, **speed** admet comme argument **auto**, signifiant qu'on laisse le routeur négocier la vitesse de la liaison. Nous n'utiliserons pas non plus cette facilité.

Configurations de niveau 3 (IP) pour tout type d'interfaces :

Concernant la configuration de niveau 3 (IP), nous devons affecter à l'interface une adresse IP et un masque de sous-réseau, puis spécifier l'adresse *broadcast* sur le réseau correspondant. Pour cela nous disposons des commandes (toujours en mode de configuration de l'interface) :

- Router(config-if)#**ip address** *adresse masque*

où *adresse* est l'adresse IP de l'interface et *masque* est son masque de sous-réseau en notation décimale pointée ;

- Router(config-if)#**ip broadcast-address** *adresse-broadcast*

où *adresse-broadcast* est l'adresse IP de broadcast sur le réseau de l'interface. Par défaut, c'est l'adresse de diffusion limitée **255.255.255.255** et il ne devrait pas être nécessaire de la modifier.

Vérification de la configuration :

Une fois une interface configurée et active, on peut vérifier que tout fonctionne correctement en utilisant la commande **ping** (échange de messages ICMP de type ECHO) dans le mode privilégié ou utilisateur en spécifiant l'adresse IP (ou éventuellement le nom) d'un hôte (ou routeur) connecté à ce réseau (et correctement configuré) :

```
Router#ping adresse-cible
```

où *adresse-cible* est l'adresse (ou le nom) de l'hôte cible. En retour on obtient un taux de réponse pour l'envoi de 5 messages ICMP ECHO REQUEST. Si ce taux est nul, il y a un problème quelque part.



N'oubliez pas d'utiliser l'aide contextuelle de la CLI pour voir à chaque fois les paramètres possibles.

Exemple 3

(Configuration d'une interface Ethernet) Supposons que l'on connecte une interface FastEthernet 10/100Base-TX du routeur à un Hub/Switch 10Base-T (ce qui n'est pas forcément votre cas). Supposons que ce réseau de raccordement a pour bloc d'adresses 139.124.187.0/24. Dans un premier temps, on va configurer l'interface aux niveaux 1 et 2 :

```
Router#show interfaces FastEthernet0
FastEthernet0 is administratively down, line protocol is down
Hardware is PQUICC_FEC, address is 0011.93ca.ffb5 (bia 0011.93ca.ffb5)
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Auto-duplex, Auto Speed, 100BaseTX/FX
...
```

- ☞ on voit que l'interface est inactive et doit pouvoir négocier la vitesse et le mode duplex, ce que nous ne laisserons pas faire dans le TP.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0
Router(config-if)#no shutdown
Router(config-if)#exit
*Apr 17 09:46:51.895: %LINK-3-UPDOWN: Interface FastEthernet0, changed state to up
Router(config)#
*Apr 17 09:46:59.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet
Router(config)#exit
Router#
*Apr 17 09:47:09.739: 05-configurations_reseauYS-5-CONFIG_I: Configured from console by console
Router#show interfaces FastEthernet0
FastEthernet0 is up, line protocol is up
...
Half-duplex, 10Mb/s, 100BaseTX/FX
...
```

- ☞ on a juste activé l'interface par le **no shutdown** et laissé la négociation se faire. L'interface s'est configurée. Cependant, il arrive que la négociation se fasse à la baisse alors qu'on aurait pu obtenir des caractéristiques plus importantes. Dans notre cas, le full-duplex est possible et on va l'activer ! Puisque nous ne voulons pas nous plus que la vitesse soit négociée, nous la spécifions aussi dans ce qui suit.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0
Router(config-if)#speed 10
Router(config-if)#full-duplex
Router(config-if)#CTRL+Z
Router#sh int fa0
FastEthernet0 is up, line protocol is up
...
Full-duplex, 10Mb/s, 100BaseTX/FX
...
```

Dans le mode de configuration de l'interface, on configure aussi la couche IP avec les paramètres voulus (on aurait pu tout faire à la fois) :

```
Router#conf ter
Router(config)#int f0
Router(config-if)#ip address 139.124.187.50 255.255.255.0
Router(config-if)#CTRL+Z
Router#sh int fa0
FastEthernet0 is up, line protocol is up
...
Internet address is 139.124.187.50/24
...
Router#show ip interface fa0
FastEthernet0 is up, line protocol is up
Internet address is 139.124.187.50/24
Broadcast address is 255.255.255.255
...
```


En supposant que ce réseau comporte une station (ou un routeur) opérationnel d'adresse 139.124.187.25, on peut vérifier la configuration avec un **ping** :

```
Router#ping 139.124.187.25

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 139.124.187.25, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms
```

➡ le taux de réussite est acceptable : tout est ok.



 Pour annuler une commande, il faut la faire précéder du mot-clé **no** dans le même mode. Par exemple, on peut annuler l'affectation de l'adresse précédente en tapant :

```
Router(config-if)#no ip address 139.124.187.50 255.255.255.0
```

ou encore par :


```
Router(config-if)#no ip address
```

car il n'y a pas d'ambiguïté.

4 Minicom pour accéder à la console

Le routeur n'étant pas configuré dans un premier temps, il nous faudra en prendre le contrôle et l'administrer. Dans cette situation, la seule possibilité est d'utiliser le port d'administration du routeur : son **port console**.

Ce port console permet d'y connecter la **console** du routeur : un terminal informatique constitué d'un clavier et d'un écran textuel. Les terminaux en mode texte présentent les sorties uniquement sous forme textuelle, et disposent simplement d'un clavier pour les entrées. Un exemple de terminal texte qui fut répandu en France est le Minitel, relié aux serveurs par l'intermédiaire de la ligne téléphonique. Dans le monde informatique, le VT100 de DEC est devenu un standard de fait, avec un affichage de 25 lignes sur 80 colonnes.

 La console est le seul terminal permettant d'administrer le routeur dans ses différents modes de démarrage (ROMMON, RxBoot, NetBoot ou Normal). Pour prendre le contrôle du routeur, il faudra le démarrer en mode ROMMON.


La console (le terminal informatique) sera émulée par le logiciel **minicom** exécuté sur un PC sous Linux. Pour cela, le port console du routeur doit être relié au port série du PC en utilisant un câble console DB-9/RJ-45. Sur la fenêtre exécutant **minicom**, nous pourrions interagir avec le routeur dans ses différents modes de démarrage, en particulier ROMMON et normal. Ce dernier nous donnera accès à la CLI (voir figure 3, page 3).

4.1 Câblage du port console au port série

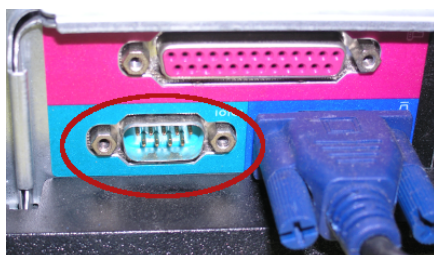
Le port (ou interface) série d'un PC est de type RS-232 avec un connecteur mâle DB-9 (9 broches). RS-232 (aussi appelé EIA RS-232C ou V.24) est une norme standardisant un port de communication de type série.

Sous Linux, ces ports sont généralement désignés par le fichiers spéciaux **/dev/ttyS0**, **/dev/ttyS1**, etc. Sous MS-DOS et Windows, ils sont désignés par les noms **COM1**, **COM2**, etc. (ce qui leur a valu le surnom de "ports COM").

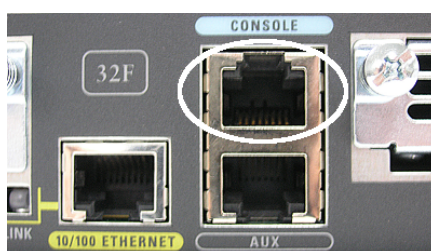
Le port RS-232 est fréquemment utilisé dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure, carte mères, PDA, etc.).

 Le port série disparaît progressivement des PC au profit des ports USB. Il existe néanmoins des adaptateurs USB/Série.

Sur le routeur, le port console est aussi un port série mais avec un connecteur RJ-45 (8 broches). Nous les connecterons par l'intermédiaire du **câble console DB-9/RJ-45** (de couleur bleu clair pour les câbles de marque CISCO).



Port série DB-9 du PC

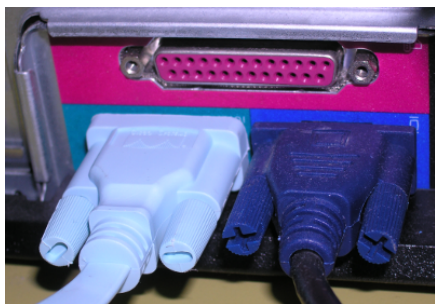


Port console RJ-45 d'un 1700



Câble console

Le câblage consiste simplement à enficher le connecteur DB-9 femelle du câble console dans le port série du PC, et son connecteur RJ-45 dans le port console :



Port série



Port console



Câblage final

4.2 Le logiciel minicom

minicom est logiciel de communication "terminal" (terminal en mode texte) qui permet de se connecter à un équipement distant en utilisant un modem ou une interface série. Dans notre cas, il nous permettra d'administrer le routeur via la liaison série à son port console. Sur les systèmes Windows, **HyperTerminal** mais aussi **Putty** peuvent jouer ce rôle.

Plus précisément, **minicom** est un émulateur de terminal, aussi appelé console virtuelle (ou terminal virtuel) qui émule le fonctionnement d'un terminal informatique.

i **minicom** a bien d'autres fonctions mais qui ne seront pas utiles dans notre contexte.

4.2.1 Paramétrage de minicom

Le port console du routeur est par défaut configuré pour fonctionner à 9600 baud, en transmettant 8 bits de données sans bit de parité mais avec un bit stop.

i Les paramètres de communication du port console peuvent être modifiés en modifiant le registre de configuration du routeur.

Il est nécessaire de configurer **minicom** pour utiliser les paramètres de communication suivants :

- Port série à utiliser : **/dev/ttyS0**
- Vitesse de modulation : **9600 baud**
- Nombre de bits de données : **8**
- Nombre de bits de parité : **aucun**
- Nombre de bits stop : **1**
- contrôle de flux matériel : **oui** (inutile)
- contrôle de flux logiciel : **aucun**

Le paramétrage de **minicom** se fait interactivement en tapant sur un terminal Linux la commande :

```
$ minicom -s
```

qui affiche le menu de configuration :

```
+-----[configuration]-----+
| Noms de fichiers et chemins  |
| Protocoles de transfert      |
| Configuration du port série  |
| Modem et appel               |
| Ecran et clavier             |
| Enregistrer config. sous dfl |
| Enregistrer la configuration sous... |
| Sortir                       |
| Sortir de Minicom            |
+-----+
```

❗ En général, si l'on n'est pas root, on peut modifier mais pas sauver la configuration de **minicom**.

4.2.2 Utilisation

Une fois configuré, en choisissant "*Sortir*", on se retrouve directement sur le terminal virtuel :

```
Bienvenue avec minicom 2.3

OPTIONS: I18n
Compilé le Feb 24 2008, 16:35:15.
Port /dev/ttyS0

Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales
```

❗ Si root a modifié et sauvé la configuration, alors on peut lancer le terminal virtuel en exécutant **mini-com** sans option :

```
$ minicom
```

Si on allume le routeur, on voit apparaître à l'écran les premiers messages du démarrage du routeur :

```
-
System Bootstrap, Version 12.2(7r)XM2, RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 2003 by cisco Systems, Inc.
```

Ce que le routeur envoie est affiché et ce que l'on tape est transmis au routeur (voir figure 3, page 3).

Dans certains cas, nous aurons besoin d'agir sur **minicom**. Par exemple, pour quitter **minicom**, il faut taper **CTRL-A** puis **X** (la casse n'a pas d'importance) et confirmer la sortie.

La combinaison **CTRL-A** est la **séquence d'échappement**, permettant d'entrer des commandes **minicom**. Outre la sortie, voici quelques commandes d'intérêt :

- **CTRL-A** puis **Z** afficher un menu présentant les commandes disponibles (voir figure 8);
- **CTRL-A** puis **F** envoyer un "**break**" sur la liaison série

 L'émission du *break* est indispensable pour prendre le contrôle du routeur.

- **CTRL-A** puis **O** afficher le menu de configuration.

Résumé des commandes de Minicom		
Les commandes peuvent être appelées par CTRL-A <touche>		
Fonctions principales		Autres fonctions
Répertoire.....D	Exécuter un script.G	Effacer l'écran....C
Envoyer des fichiersS	Recevoir des fichiersR	Configurer Minicom.O
Paramètres de comm.P	Ajouter LF.....A	Suspendre minicom..J
Capture act/désact.L	Raccrocher.....H	Sortir et razer....X
Envoyer « break »..F	Initialiser modem..M	Quitter sans razer.Q
Réglages du term...T	Exécuter Kermit....K	Touches du curseur.I
Coupage des lignes.W	local Echo on/off..E	Help screen.....Z
Paste file.....Y		revenir en arrière.B
Sélectionnez une commande ou tapez Entrée pour revenir.□		
Ecrit par Miquel van Smoorenburg 1991-1995		
Quelques additions de Jukka Lahtinen 1997-2000		
i18n par Arnaldo Carvalho de Melo 1998		

FIGURE 8 – Menu des commandes **minicom**

5 Sauvegarde/restauration de la configuration avec TFTP

TFTP (*Trivial File Transfer Protocol*) est un protocole de transfert de fichiers, très simple, reposant sur UDP. Il est défini dans la [RFC 1350](#). Ses fonctionnalités sont très limitées, notamment comparé à FTP. Il se limite au transfert de fichier uniquement, sans authentification ni cryptage. TFTP suit le modèle client-serveur. Le port UDP réservé aux serveurs TFTP est le port 69.

Dans les grandes lignes, le transfert de fichier entre un client et un serveur TFTP suit la procédure suivante :

1. Le client demande au serveur (port UDP 69) le transfert d'un fichier dont il indique la référence. TFTP n'offrant pas de possibilité de se déplacer dans des répertoires ni d'obtenir une liste de fichiers disponibles, le client doit connaître la référence du fichier :
 - qu'il veut récupérer s'il s'agit d'une requête de lecture (message RRQ);
 - qu'il veut déposer sur le serveur s'il s'agit d'une requête d'écriture (message WRQ).
2. Si le serveur accepte le transfert, il répond au client en utilisant un autre port qui sera désormais utilisé pour communiquer avec le client (envoyer/recevoir le fichier)
3. TFTP utilisant UDP, la bonne marche du transfert doit être assurée par le client et le serveur. Ils utilisent une succession de messages DATA (contenant une partie du fichier) et ACK (accusant réception d'une partie), selon un protocole de type envoyer et attendre : une partie du fichier n'est envoyée que si la partie précédente a été correctement reçue (acquittée).

On utilisera régulièrement les fonctionnalités de client TFTP du routeur afin de déposer/récupérer des fichiers de configuration sur le serveur TFTP d'une machine virtuelle (VM). Ce serveur TFTP est **in.tftpd** (se trouvant dans `/usr/sbin/`), activé au démarrage de cette VM, ce que l'on peut vérifier sur un terminal en tapant :

```
# netstat -lun
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale      Adresse distante     Etat
...
udp        0      0 0.0.0.0:69          0.0.0.0:*
...
```

où l'on peut remarquer que le port 69 d'UDP est bien utilisé et que ce serveur est joignable en utilisant n'importe laquelle des adresses IP du système invité.

i Comme souvent pour un service assez rarement utilisé comme TFTP, c'est en réalité le **super-serveur** **inetd** qui écoute sur ce port. Dès qu'un datagramme parvient sur ce port, **inetd** exécute **in.tftpd** pour le traiter et tout se passera comme si **in.tftpd** avait été actif dès le départ. C'est une solution économe en processus. On peut toutefois modifier cette configuration et installer un serveur TFTP qui fonctionne en mode **standalone** : sans dépendre de **inetd** et activé dès le départ.

Dans le fichier `/etc/inetd.conf` contenant une partie de la configuration de **inetd**, on voit les paramètres de configuration du serveur TFTP :

```
# cat /etc/inetd.conf
# /etc/inetd.conf: see inetd(8) for further informations.
#
# Internet superserver configuration database
...
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
...
tftp          dgram      udp      wait     nobody /usr/sbin/tcpd /usr/sbin/in.tftpd /srv/tftp
...
```

où :

- **tftp** est le nom du service TFTP dans le fichier `/etc/services`, auquel est associé le port UDP 69
- **dgram** indique qu'il fonctionne en mode datagramme
- **udp** indique qu'il repose sur UDP
- **wait** indique qu'une fois activé, le serveur s'occupera de la totalité des datagrammes parvenant sur le port 69
- **nobody** est le nom d'utilisateur du (processus) serveur
- le reste est la commande exécutant le serveur :
 - ◇ **/usr/sbin/tcpd** est un *wrapper* : un programme permettant de filtrer les accès aux serveurs en interdisant ceux provenant des hôtes spécifiés dans `/etc/hosts.deny` sauf ceux de `/etc/hosts.allow`. Ces fichiers sont vides sur la machine virtuelle ;
 - ◇ **/usr/sbin/in.tftpd** est le code (exécutable) du serveur TFTP lui-même
 - ◇ **/srv/tftp** est le répertoire de travail du serveur TFTP, qui contiendra les fichiers envoyés/reçus. La référence du fichier, indiquée par le client, doit être relative à ce répertoire.

Le transfert d'un fichier par TFTP peut se faire du client vers le serveur (dépôt du fichier) et inversement (récupération du fichier).

5.1 Dépôt d'un fichier du routeur sur le serveur TFTP

Pour pouvoir déposer un fichier dans le répertoire de travail (`/srv/tftp`) du serveur **in.tftpd**, il faut préparer le serveur à accueillir le fichier. En effet, aucune sécurité n'est prévue par TFTP. Pour limiter un peu les risques, **in.tftpd** requiert que le fichier à déposer existe déjà (même vide) et qu'il ait le droit d'écriture dessus. Dans notre configuration, **in.tftpd** a pour identité **nobody**.

La procédure à suivre est la suivante :

1. Créer dans `/srv/tftp` le fichier (même vide) ayant le nom que le fichier à déposer portera sur le serveur ;
2. Modifier les permissions de ce fichier pour que le serveur ait le droit d'écriture dessus. Par la même occasion, on lui accordera aussi le droit de lecture, utile pour un éventuel transfert du serveur vers le routeur (client) ;
3. Le serveur est maintenant prêt, on peut lancer la commande sur le routeur pour déposer le fichier.

Exemple 4

Supposons que le serveur TFTP soit opérationnel sur la station d'adresse 150.151.152.1 et que l'on veuille y déposer la configuration courante (running-config) du routeur en l'appelant config-routeur.txt.

On prépare d'abord le serveur pour accueillir le fichier config-routeur.txt :

```
# touch /srv/tftp/config-routeur.txt
# chmod a+rw /srv/tftp/config-routeur.txt
# ls -l /srv/tftp/config-routeur.txt
-rw-rw-rw- 1 root root 0 mai  3 13:30 /srv/tftp/config-routeur.txt
```

- ➡ création du fichier cible vide, et ajout des droits de lecture/écriture pour tout le monde sur ce fichier. La dernière commande sert juste à vérifier que tout est ok

Sur le routeur et en mode privilégié, on peut exécuter la commande demandant le transfert. Cette commande demande quelques informations :

```
Router#copy running-config tftp:config-routeur.txt
Address or name of remote host []? 150.151.152.1
Destination filename [config-routeur.txt]? 
!!
709 bytes copied in 4.792 secs (148 bytes/sec)
```

- ➡ c'est tout : le dépôt du fichier a réussi

Sur le serveur, on peut maintenant voir que le fichier n'est plus vide :

```
# ls -l /srv/tftp/config-routeur.txt
-rw-rw-rw- 1 root root 709 mai  3 13:35 /srv/tftp/config-routeur.txt
# cat /srv/tftp/config-routeur.txt

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
...
```



5.2 Récupération sur le routeur d'un fichier du serveur TFTP

L'opération est encore plus simple que la précédente car on n'a pas à préparer le serveur s'il a déjà le droit de lecture sur le fichier demandé.

Exemple 5

On poursuit l'exemple précédent, mais cette fois on récupère le fichier `config-routeur.txt` et on le sauve dans la NVRAM sous le nom `running.old` :

```
Router#copy tftp:config-routeur.txt nvram:running.old
Address or name of remote host []? 150.151.152.1
Destination filename [running.old]? Entrée
Accessing tftp://150.151.152.1/config-routeur.txt...
Loading config-routeur.txt from 150.151.152.1 (via FastEthernet0): !
[OK - 709 bytes]

709 bytes copied in 9.396 secs (75 bytes/sec)
```

⇒ le fichier a été récupéré avec succès

```
Router#dir nvram:
Directory of nvram:/

 27  -rw-          596          <no date>  startup-config
 28  ----           5          <no date>  private-config
  1  -rw-           0          <no date>  ifIndex-table
  2  ----          12          <no date>  persistent-data
  3  -rw-          709          <no date>  running.old

29688 bytes total (25963 bytes free)
```

⇒ et figure bien dans la NVRAM.

❗ On aurait pu tout aussi bien le copier directement à la place de la `running-config` ou de la `startup-config`.

