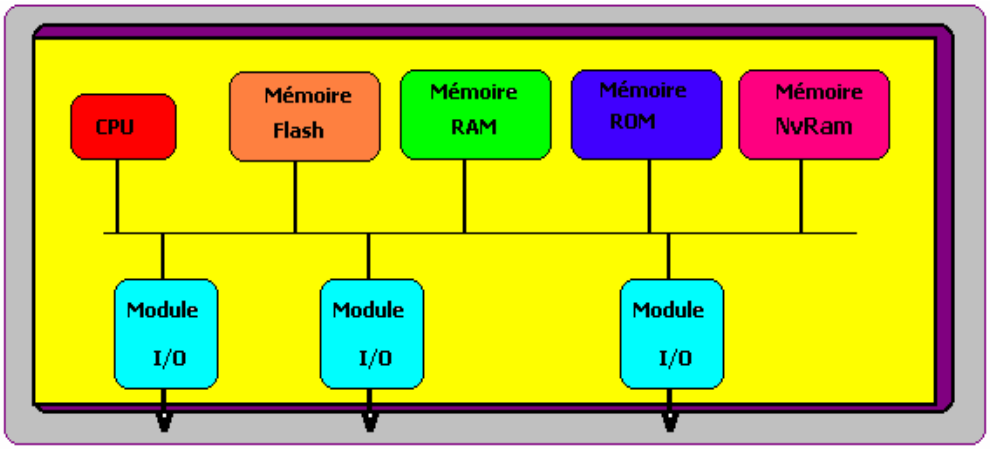
Initiation à la configuration des Routeurs Cisco

Architecture interne

Tous Les routeurs Cisco ont une architecture interne qui peut être représenté par :



Ils contiennent tous :

- ✓ Une mémoire ***NvRam*** pour Ram non Volatile et sur laquelle l'administrateur va stocker la configuration qu'il aura mise dans le routeur. Elle contient également la configuration de l'IOS.
- ✓ Une carte mère qui est en général intégrée au châssis. Par contre les interfaces sont des cartes additionnelles. Il existe des interfaces Ethernet, Token-Ring, série V35-V24, RNIS (ISDN), ATM, etc…
- ✓ Une ***CPU*** qui est un microprocesseur Motorola avec un *BIOS* spécial nommé " *I.O.S.* " pour *Internetwork Operating System*.
- ✓ Une mémoire ***RAM*** principale contenant le logiciel IOS, c'est dans laquelle tout sera exécuté un peu à la manière d'un simple ordinateur. C'est là où l'IOS est exécuté ainsi que le bootstrap, etc. On y retrouve également toutes les tables créées pendant l'utilisation du routeur (tables de routages, ARP, etc.), mais aussi tous les buffers utilisés par les cartes d'entrée sorties.
- ✓ Une mémoire ***FLASH*** , également une mémoire non volatile sur laquelle on stocke la version courante de l'IOS du routeur. Cette mémoire peut contenir plusieurs mégaoctets de mémoire, on y stocke parfois plusieurs versions du logiciel.
- ✓ Une mémoire ***ROM*** non volatile et qui, quant à elle, contient les instructions de démarrage (bootstrap) et est utilisée pour des opérations de maintenance difficiles.

Commandes de base

Modes d'accès

On peut configurer un routeur, en utilisant :

- ❖ Soit le mode console,
- ❖ Soit une plate forme d'administration (exemple CiscoWorks qui utilise des requêtes SNMP).

Pour atteindre le menu de configuration d'un routeur, on utilise :

- Soit le port console du routeur,
- Soit des terminaux virtuels en émulation telnet.

Les routeurs *Cisco* fonctionnent dans trois modes différents. Le mode ligne de commande **'exec'** permet d'exécuter quelques commandes de base, mais sans modifier la configuration du routeur. Le mode administrateur **'exec privilégié'** permet de modifier certains paramètres du routeur et d'accéder à des commandes complémentaires. Le mode **'global configuration'** permet lui de modifier complètement la configuration du routeur.

Lorsque l'on se connecte à un routeur **Cisco** (via console ou via telnet) on se retrouve dans le mode ligne de commande. Le prompt est alors **>** précédé du nom du routeur :

Routeur1 >

L'utilisateur est alors capable de passer différents types de commandes (ping, show, etc…) mais il ne peut modifier la configuration du routeur.

Pour passer en mode administrateur, il suffit de rentrer l'instruction **enable**, et de donner s'il existe, le mode de passe pour passer dans ce mode. Le prompt change alors pour devenir **#** :

```
Routeur1> enable
Routeur1 #
```

Les deux lignes précédentes nous ont permis de passer en mode administrateur, nous allons maintenant passer en mode configuration du routeur :

```
Routeur1 # config term
Routeur1(config)#
```

Pour ressortir de ce mode configuration, il suffit de taper la commande **<ctrl> Z** et l'on revient au mode privilégié. Pour remonter les niveaux d'accès ou pour sortir du mode privilégié, on utilise la commande **disable** ou tout simplement la commande **exit**.

Lorsque l'on est dans le mode configuration, il est possible de passer tous les ordres s'appliquant de manière globale au routeur.

Sauvegarde de configuration

Lorsque l'on passe en mode configuration globale à l'aide de la commande **# conf term** on est alors dans la **RAM** et toute commande passée a un effet immédiat.

La commande **# show conf** permet d'afficher la configuration sauvegardée dans la **NvRam**(et pas forcément celle actuellement en fonction dans la RAM). Pour afficher la configuration qui fonctionne actuellement et qui se trouve dans la **RAM** (et qui peut être différente de celle stockée dans la **NvRam**), il suffit de taper la commande :

 # write term.

Et finalement, lorsque l'on souhaite par précaution (en cas de coupure de courant) sauvegarder la configuration actuelle de la **RAM** dans la **NvRam**, on utilise la commande :

 # write memory

Configuration des Interfaces Ethernet

La configuration des interfaces se fait interface par interface. Pour configurer l'interface Ethernet 1/0 d'un routeur, on doit accéder à la configuration de l'interface en question à l'aide de la commande:
Routeur1 (config)# **interface Ethernet 1/0**.

Le résultat de cette commande est le changement de l'invite en :

Routeur1 (config-if)# **.**

Un ensemble de commandes commençant par :
Routeur1 (config-if)# **ip**
sert à la configuration du protocole IP sur l'interface courante. La commande :
Routeur1 (config-if)# **ip ?**
fournit la liste des commandes possibles. Parmi ces commandes celle que l'on va souvent utiliser et qui permet de spécifier l'adresse IP de cette interface ainsi que son masque de sous-réseau :
Routeur1 (config-if)# **ip address *adresse masque***

Pour activer/désactiver une interface on utilise la commande :

Routeur1 (config-if)# **no shutdown/shutdown**

Sur les routeurs Cisco, la commande **no *commande paramètres*** permet de supprimer l'effet d'une commande antérieure.

Pour notre TP, nous avons utilisé toutes ces commandes pour paramétrer les interfaces de nos deux routeurs que nous avons nommé **R1** et **R2.**

Ci-dessous une copie du déroulement de la configuration des interfaces Ethernet de notre routeur **R1**:

```
R1>
R1>en
Password:
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int e0
R1(config-if)#ip address 157.160.0.1 255.255.0.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#int e1
R1(config-if)#ip address 157.161.0.1 255.255.0.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#exit
R1#sh ru
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname R1
!
enable password int
!
```

```
ip subnet-zero
!
!
!
interface Ethernet0
 ip address 157.160.0.1 255.255.0.0
 no ip directed-broadcast
!
interface Ethernet1
 ip address 157.161.0.1 255.255.0.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
ip classless
!
!
line con 0
 transport input none
line vty 0 4
 password int
 login
!
end
R1#write memory
Building configuration...
[OK]
R1#
```

Remarque : *Il est recommander d'activer les interfaces après les avoir configurées.*

La commande **show running-configuration** (sh ru) permet de voir la configuration courante. Les commandes relatives à la configuration du routeur sont accessibles à partir du mode **exec privilégié**. Nous avons déjà signalé que les routeurs Cisco maintiennent leur configuration dans la mémoire NvRam.

Il est également possible de charger sur le routeur via le réseau une configuration en utilisant le protocole **tftp**.

C'est dans le fichier **startup-config** de la mémoire NvRam qu'est stocké la configuration qui va être prise en compte lors du démarrage du routeur. Cette configuration peut être visualisée avec la commande **show startup-config**.

Les interfaces Ethernet du routeur R2 ont été configurées de la même façon que celles du routeur R1.

Routage

Les interfaces Ethernet de nos deux routeurs sont maintenant configurées. On va s'attaquer à la configuration du routage. Deux types de routage sont utilisables. Le premier est le routage statique où les tables de routage sont alimentées manuellement sur chaque routeur du réseau. Le second est le routage dynamique où un protocole de routage est utilisé pour distribuer les routes dans l'ensemble du réseau.

Routage statique

Il consiste à indiquer à chaque routeur du réseau, le chemin à prendre (interface du routeur suivant) pour véhiculer un datagramme vers le réseau destination. Concrètement cette configuration se fait manuellement. En routage statique, il est évident qu'une rupture de lien empêche tout routage transitant par ce lien.

Ce type de routage fonctionne bien lorsque le réseau est de petite taille. Il est également utilisable lorsque le réseau ne comporte qu'un seul point de connexion aux autres réseaux ou lorsqu'il n'inclut aucune route redondante (ex: route de backup au cas où la route principale deviendrait invalide).

La déclaration d'une route statique sur le routeur s'effectue en mode configuration globale.

Remarque : *La décision de routage est globale au routeur et s'applique à toutes les interfaces.*

La commande permettant de créer une route statique sur le routeur est la suivante :

Routeur1(config)# **ip route <@IPdest> <masque> <le prochain saut> <métrique>**

Nous avons essayé de configurer nos routes en statique dans notre réseau. On s'est vite rendu compte que la configuration statique est une méthode fastidieuse dès que le nombre de machines du réseau devient important.

Les routeurs ont des tables de routage qui sont structurées en général de 4 colonnes:

- *un réseau de destination*
- *un masque,*
- *l'adresse du prochain saut,*
- *une métrique de saut (un poids ou une priorité).*

Cette table peut être consulter à l'aide de la commande :

Routeur1# show ip route.

Routage dynamique

Le routage dynamique fonctionne à l'aide d'un protocole de communication entre les routeurs. Ce protocole permet aux routeurs l'échange d'informations concernant les réseaux auxquels ils sont connectés.

Les protocoles de routage les plus utilisés sont groupés en deux familles. La première utilise des algorithmes de type vecteurs de distance (*Distance Vector*) comme le protocole **RIP** (*Routing Information Protocol*) ou bien le protocole **BGP** (Border Gateway Protocol). La seconde famille utilise un algorithme à états de lien *(Link State)* comme par exemple le protocole **OSPF** (*Open Shortest Path First).*

Dans le cadre de notre TP nous nous sommes plus intéressé au protocole RIP qu'au protocole OSPF.

Routage par RIP

Le protocole RIP appartient à une autre grande famille de protocoles dits de routage interne. Le processus de RIP affecte aux routes une priorité basée sur la métrique. Un routeur, pour lequel le protocole de routage RIP est activé, diffuse sa table de routage à ses voisins toutes les 30 secondes. Par défaut, un routeur RIP envoie ses vecteurs de distance toutes les trente secondes. Ce délai peut être modifié grâce à la commande **timers basic**.

Les messages RIP sont transmis à l'intérieur de datagrammes UDP en utilisant le port numéro 520. La métrique utilisée par RIP pour mesurer la valeur des différentes routes est une métrique très simple: *le nombre de sauts*. Le nombre de sauts (*hop count*) est le nombre de routeurs intermédiaires que traverse une route.

La déclaration d'un processus de routage dynamique s'effectue dans un mode de configuration particulier. Lorsque l'on est en mode de configuration globale, il faut déclarer un processus de routage.

Avant d'activer le routage RIP sur un routeur, il est préférable, au préalable, d'effacer toutes les routes préexistantes par la commande :

Routeur1(config)# **no ip route** .

L'utilisation des protocoles de routage dynamique se fait en activant le protocole de routage par l'intermédiaire de la commande **router**. Par exemple, la commande :

Routeur1(config)# **router rip**

Routeur1(config-router)#

active le protocole RIP sur le routeur. Il faut également, chaque fois que RIP est activé, spécifier la version de RIP via la commande :

Routeur1(config-router)# **version 2**.

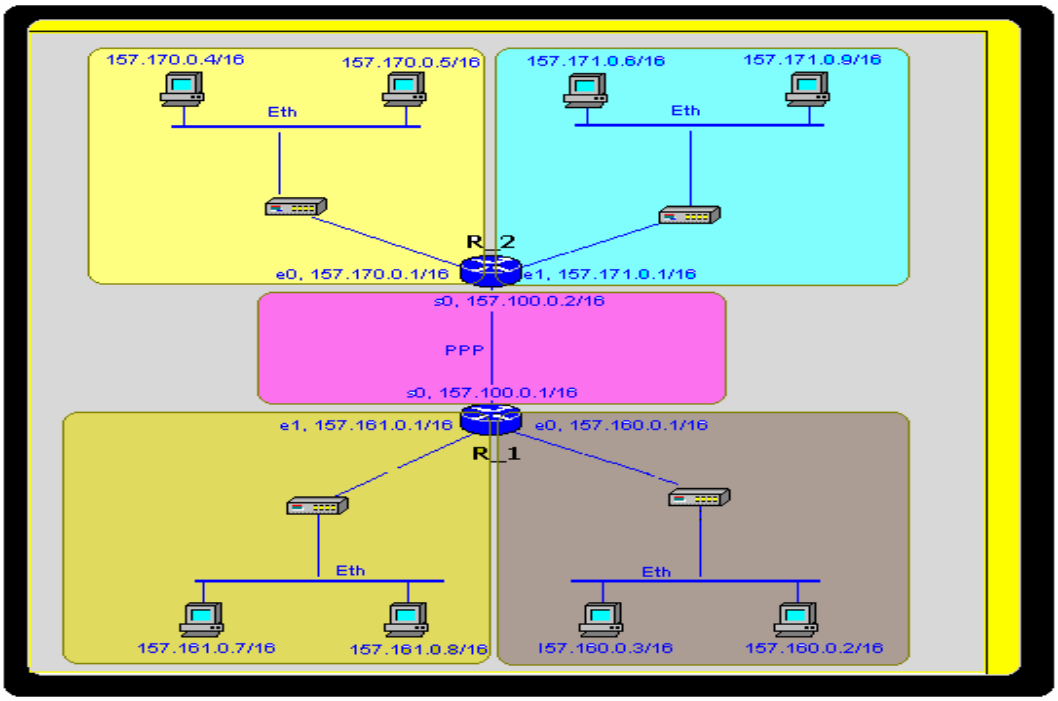
Il est également préférable de désactiver l'option de concaténation des adresses IP avec la commande :

Routeur1(config-router)# **no auto-summary**.

Une fois le protocole de routage activé, le routeur est prêt à accepter les messages RIP venant d'autres routeurs du réseau.

> **Remarque :** *Un routeur (Cisco) n'annoncera des routes avec le protocole RIP que si chaque route à annoncer est explicitement spécifiée. Cela se fait avec la commande* **network**.

Par exemple, les commandes ci-dessous permettent au routeur **R1** d'annoncer à l'ensemble du réseau qu'il parvient à joindre les réseaux : 157.160.0.0, 157.161.0.0 et 157.100.0.0 .



Routeur1(config-router)# **network 157.160.0.0**
Routeur1(config-router)# ***network 157.161.0.0***
Routeur1(config-router)# ***network 157.100.0.0***

Au départ nous n'avions spécifié au routeur **R1** que les deux premiers réseaux. Le résultat était que le RIP ne fonctionnait pas du tout. Par conséquent le contenu de la table de routage est réduit aux réseaux directement connectés. Les routeurs d'autres constructeurs n'ont pas besoin de cette spécification des routes qui est faite par défaut dès que les interfaces sont configurées. Ce qui paraît tout à fait normal mais si on ne veut, pour des raisons de sécurité ou d'autre, ne pas annoncer une route aux voisins, on est obligé de le préciser à ce type de routeurs. Ce qui revient au même.

On a pu vérifier notre configuration actuelle en examinant le fichier **running-config** ou en exécutant la commande **show ip protocol** ou la commande **show ip rip database**.

Ci dessous une copie du résultat des deux commandes **show running-config** et **show ip route** passées sur le routeur **R2** .

```
R2#sh ru
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname R2
!
enable password int
!
ip subnet-zero
!
!
```

```
!
interface Ethernet0
 ip address 157.170.0.1 255.255.0.0
 no ip directed-broadcast
!
interface Ethernet1
 ip address 157.171.0.1 255.255.0.0
 ip directed-broadcast
!
interface Serial0
 ip address 157.100.0.2 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
!
router rip
 version 2
 network 157.100.0.0
 network 157.170.0.0
 network 157.171.0.0
!
ip classless
!
!
line con 0
 transport input none
line vty 0 4
 password int
 login
!
end
```

R2#sh ip rou

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    157.170.0.0/16 is directly connected, Ethernet0
C    157.171.0.0/16 is directly connected, Ethernet1
R    157.160.0.0/16 [120/1] via 157.100.0.1, 00:00:16, Serial0
R    157.161.0.0/16 [120/1] via 157.100.0.1, 00:00:16, Serial0
     157.100.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       157.100.0.1/32 is directly connected, Serial0
C       157.100.0.0/16 is directly connected, Serial0
```

Nous avons également testé l'effet d'une rupture de lien sur le routage. Nous avons remarqué que la convergence de l'algorithme RIP est très lente par rapport à celle de l'OSPF. Chaque route possède en effet un **timeout** associé. Si un message RIP n'est pas parvenu à son voisin au bout de 3 minutes, ce dernier considère que la liaison n'est plus valide, c'est à dire que la métrique est supérieure à 15. Comme par défaut, un routeur émet un message RIP toutes les trente secondes, il faut six mises à jour de la part du routeur qui a signalé cette route. Si après 240 secondes le routeur n'a toujours pas reçu de vecteurs RIP, la route est supprimée de la table de routage.

Routage par OSPF

OSPF fait partie d'une deuxième génération de protocoles de routage. Il n'utilise pas d'encapsulation TCP ou UDP mais il se place directement au-dessus du protocole IP avec 89 comme numéro de protocole. C'est un protocole de type « Link state » basé sur un algorithme de calcul de la route dérivé de celui de Djikstra. OSPF repose sur un principe fondamental qui est la hiérarchisation du routage. Cette dernière notion a pour but de mieux contrôler la diffusion des informations de routage. A la différence de RIP, OSPF privilégie plutôt l'échange des parties des tables ayant été modifiées.

Un système autonome SA, est un ensemble de routeurs qui partagent la même administration. Un SA peut être subdivisé en plusieurs zones dites « areas ».

Comme pour tout protocole de routage, avant d'activer l'OSPF, il faut au préalable effacer toutes les routes préexistantes **no ip route** , puis demander le mode de configuration globale :

Routeur1(config)# **router ospf 100**

100 étant le numéro de processus. Ensuite, il suffit de spécifier tous les réseaux auxquels le routeur est connecté. Cependant à la différence du mode RIP, il faut préciser le masque du sous-réseau et la zone dont le routeur fait partie. Les interfaces de chaque routeur ont été définies précédemment.

Une fois ce travail achevé, les routeurs commencent à échanger leurs tables de routages. A l'issue d'un temps de convergence assez rapide par rapport à RIP, tous les routeurs possèdent une table de routage actualisée.

Configuration des Interfaces Séries (WAN)

La configuration des interfaces séries (WAN) des routeurs est un peu plus compliqué que celle des interfaces Ethernet. Dans ce travail nous avons essayé de comprendre et configurer les protocoles **PPP** et **Frame Relay** sur des interfaces séries. Nous allons simuler une interconnexion WAN en utilisant les 2 routeurs R1 et R2 connectés *back-to-back*. Nous allons également développer les notions de DCE et DTE.

Encapsulation PPP

Nous reprenons notre architecture du paragraphe précédent. En l'absence du modem qui, en général, joue le rôle de DCE, nous sommes contraints à désigner parmi nos deux routeurs (R1 et R2) celui qui va être DCE.

Nous affectons, tout d'abord, une adresse IP à chacune des interfaces séries des routeurs. Comme pour les interfaces Ethernet, l'assignation d'une adresse IP à une interface série se fait en deux étapes :

- En mode configuration globale en tape la commande :

  Routeur1(config)# **interface s0**, ce qui nous permet d'accéder à la configuration de l'interface série s0 :

  Routeur1 (config-if)#

- On rentre ensuite la commande :

Routeur1(config-if)# **ip address 157.100.0.1 255.255.0.0**

La vitesse du lien en bits par seconde est en générale définie par le DCE, mais puisque nous avons désigné le routeur R1 comme DCE nous devons lui indiquer la vitesse du lien. Nous utilisons pour notre lien série un câble de type RS-232. La vitesse de 56000 bps est recommandée pour ce type de câble. Pour fournir cette valeur au routeur, on utilise la commande (sous le prompt de la configuration de l'interface s0) :

Routeur1(config-if)# clock rate 56000

On peut vérifier la configuration de cette interface en utilisant la commande **show interface s0** ou la commande **show controllers interface s0** qui permet de voir si l'interface est configurer en DTE ou DCE. La configuration de l'interface série est maintenant complète. Si on paramètre l'adresse IP de l'autre interface (DTE), alors la liaison est établie et on peut le vérifier en utilisant des commandes **ping** d'un routeur vers l'autre.

A l'aide de la commande **show interface** que nous avons effectué sur le routeur R1, on a pu voir la configuration du **clockrate** et du protocole d'encapsulation :

```
!
interface Serial0
 ip address 157.100.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 clockrate 56000
!
```

Le protocole utilisé par défaut sur les interfaces série de tout les routeurs Cisco, est le protocole HDLC. On peut éventuellement choisir un autre protocole, comme par exemple le protocole PPP. Pour configurer l'encapsulation PPP sur un routeur, on utilise la commande :

Routeur1(config-if)# **encapsulation ppp**

A l'aide du protocole CHAP ou PAP, on peut sécuriser (authentifier) les connexions entre nos routeurs. Pour cela il suffit d'activer l'authentification en tapant la commande :

Routeur1(config-if)# **ppp authentification chap**

L'utilisation du protocole CHAP nécessite un nom (celui du routeur distant) et un mot de passe sur chacun des deux routeurs. Pour que l'authentification réussisse, il faut que les mots de passe des routeurs communicants soient identiques. A l'établissement de la connexion le protocole CHAP utilise ce mot de passe pour l'authentification. On indique au routeur le nom (*hostname*) du routeur distant et le mot de passe en utilisant la commande (sous le prompt de la configuration globale) :

Routeur2(config)# **username** *Routeur1* **password** *rst2002*

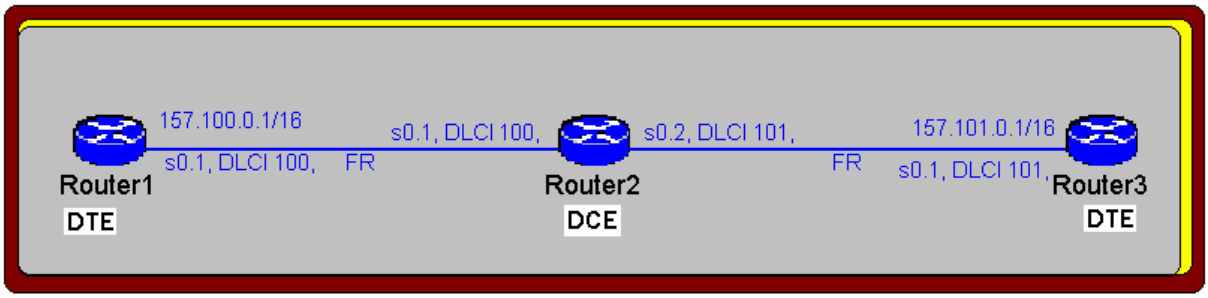
En revenant en mode administrateur **'exec privilégié'**, on peut vérifier la configuration à l'aide de la commande **show interface s0**. Le résultat de cette commande sur notre routeur R2 nous donne :

```
R2#sh int s0
Serial0 is up, line protocol is up
  Hardware is QUICC Serial
  Internet address is 157.100.0.2/16
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
```

```
Encapsulation PPP, loopback not set, keepalive set (10 sec)
LCP Open
Open: IPCP, CDPCP
Last input 00:00:06, output 00:00:06, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
   Conversations  0/1/256 (active/max active/max total)
   Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   553 packets input, 44713 bytes, 0 no buffer
   Received 227 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   554 packets output, 25226 bytes, 0 underruns
   0 output errors, 0 collisions, 89 interface resets
   0 output buffer failures, 0 output buffers swapped out
   0 carrier transitions
   DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Encapsulation Frame-Relay

Dans ce paragraphe nous considérons une configuration WAN à commutation Frame –Relay. Pour cela nous avons besoins d'un commutateur Frame-Relay ou d'un routeur qui jouera ce rôle donc ; un routeur avec au moins deux interfaces séries. Le dispositif que nous proposons est représenté par la figure suivante :



Le routeur 2 doit être configuré en commutateur Frame-Relay, ce qui explique l'absence des adresses IP de ses interfaces sur la figure. Lorsqu'il reçoit une trame sur l'interface S0/1 avec un DLCI égale à 100 elle l'a transmet sur l'interface S0/2 avec un DLCI égale à 101 et vice versa.

Les routeurs 1 et 3 doivent être configuré comme DTE et, le routeur 2 comme DCE. En générale c'est une interface d'un routeur qui est configurée comme DCE et non pas le routeur. On sera donc qu'il faut déclarer un *clockrate* pour chacune des deux interfaces du routeur 2.

Nous n'avons pas eu la chance de réaliser ce 'Lab' parce que nous ne disposions pas de routeur ayant la capacité de faire de la commutation Frame-Relay.

Filtrage du trafic

Le filtrage permet de faire une sélection des paquets au point d'entrée/sortie du réseau ; soit un routeur. Un filtre est un ensemble de règles que l'on applique à tout ou partie du trafic passant par une interface d'un routeur. Cet ensemble de règles (*access list*) constitue une table située dans le routeur. Le principe d'une liste de contrôle d'accès «access-list» est de définir successivement les types de trafic autorisé en entrée et en sortie. La syntaxe est différente selon que le niveau de filtrage mais elle prend toujours la forme:

autorisation protocole @Ipsource son_masque @IPdest son_masque opération port

qui, plus précisément, s'écrit :

(permit/deny) (tcp/ip…) @ et masques (eq/gt..) port

On peut confiner les règles de filtrage selon trois niveaux différents :

- Au niveau machine : on définit quelles sont les machines externes et internes qui peuvent communiquer entre elles.
- Au niveau application : on définit les serveurs internes qui sont accessibles à partir de l'extérieur.(Ex : serveur Web : HTTP sur le port 80).
- Au niveau machine - application : on définit les machines extérieures qui ont le droit de se connecter sur telle ou telle application interne.

Il existe deux politiques de sécurité :

- ✓ Tout ce qui n'est pas explicitement interdit est autorisé
- ✓ Tout ce qui n'est pas explicitement autorisé est interdit

La politique qui semble la plus raisonnable est la deuxième et c'est la politique la plus utilisée. Cette approche est la plus intéressante sachant qu'il est plus facile de décrire le trafic autorisé que de réaliser une liste de tout ce qui est permis.

La première règle, à appliquer sur l'interface d'entrée d'un routeur, est d'interdire le **source routing :** C'est à dire ne pas laisser passer les paquets contenant un itinéraire de la route à suivre. Cette configuration est indépendante du système de filtrage et s'obtient en introduisant la commande:

no ip source-route

Remarque : Les règles qui suivent et sauf mention du contraire sont configurées sur l'interface s0 d'entrée WAN sur notre routeur R1.

La mascarade d'adresse IP *(IP Spoofing)* est une technique d'attaque qui permet à une machine extérieure de se faire passer pour une machine du réseau. La protection contre la mascarade d'adresse se fait via une règle simple interdisant l'entrée à notre réseau de tout paquet dont l'adresse réseau source est égale à celle du notre. Cette règle qui très efficace s'écrit:

access-list 101 deny ip 157.160.0.0 0.0.255.255 0.0.0.0 255.255.255.255

Mise en place sur le routeur, cette règle veille à ce qu'aucune machine extérieure ne pourra avoir la même adresse IP qu'une machine intérieure. Si

c'est le cas, le flux provenant de cette machine est mise en quarantaine et est exploité selon la décision de l'administrateur. Cette règle fait partie du filtrage au niveau machine : on interdit ici les machines d'un réseau particulier. Un autre exemple de filtrage niveau machine est celui de la protection contre les attaques douteuses. Pour cela on interdit les adresses IP commençant par 127 et qui sont des adresses réservées.

access-list 101 deny ip 127.0.0.0 0.255.255.255 0.0.0.0 255.255.255.255

Un exemple concret et simple à mettre en œuvre est celui où on interdit l'accès aux machines d'un réseau particulier :

access-list 101 deny ip 157.161.0.0 0.0.255.255 0.0.0.0 255.255.255.255

On peut interdire l'accès à une application comme on peut ne pas l'autoriser que sur une machine. Par exemple, si on ne veut pas autoriser du telnet (port 23) vers notre réseau, la règle est (niveau application) :

access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 23

Si toute fois on possède une machine sécurisée d'adresse IP 157.160.1.23 et, sur laquelle on veut autoriser des sessions telnet, on ajoute tout simplement la ligne (règle niveau application+machine):

access-list 101 permit tcp 0.0.0.0 255.255.255.255 157.160.1.23 0.0.0.0 eq 23

On peut également ne pas autoriser à des sites douteux, que les applications dont les ports sont supérieurs à 1023 :

access-list 101 permit tcp 157.161.0.0 0.0.255.255 0.0.0.0 255.255.255.255 gt 1023

access-list 101 permit udp 157.161.0.0 0.0.255.255 0.0.0.0 255.255.255.255 gt 1023

Une fois constituées, ces *'access list'* peuvent être appliqué à n'importe quelle interface. On peut également appliquer plusieurs *'access list'* à une même interface. La syntaxe de la commande permettant d'associer une *'access list'* à une interface est de la forme suivante :

ip access-group numéro_access_list in/out

La plupart des routeurs sur le marché donnent plus de possibilités pour affiner le filtrage. On rappel également que chacune des interfaces d'un routeur peut avoir ses propres règles de filtrage. Les routeurs ne représentent qu'un moyen peut fiable pour la protection contre les attaques. Il existe d'autres équipements plus spécialisés en sécurité comme les *proxy* ou les *passerelles*.