

# *Interconnexion de réseaux*



HAMROUNI Pierlin GI04  
LOURD Rodolphe GI04  
ZEO Renan GI04  
ZERIANE Sofiane GI04

# SOMMAIRE

<b>1. Le routage .....</b>	<b>3</b>
1.1. Routeurs.....	3
1.2. Tables de routage .....	4
1.3. Protocoles de routage.....	5
<b>2. Routage statique et dynamique avec RIPv1 .....</b>	<b>7</b>
2.1. Objectif.....	7
2.2. Configuration IP des machines .....	7
2.3. Routage statique .....	9
2.4. Routage dynamique .....	10
<b>3. Routage dynamique avec RIPv2 et OSPF.....</b>	<b>15</b>
3.1. Objectif.....	15
3.2. Prise en main de Zebra.....	15
3.2. Optimisation RIP : utilisation de RIPv2.....	16
3.3. Routage dynamique avec OSPF et routeur CISCO.....	18
<b>4. Conclusion .....</b>	<b>19</b>

# 1. Le routage

## 1.1. Routeurs

Les routeurs sont les dispositifs permettant de "choisir" le chemin que les datagrammes vont emprunter pour arriver à destination. Il s'agit de machines ayant plusieurs cartes réseau dont chacune est reliée à un réseau différent. Ainsi, dans la configuration la plus simple, le routeur n'a qu'à "regarder" sur quel réseau se trouve un ordinateur pour lui faire parvenir les datagrammes en provenance de l'expéditeur.

Toutefois, sur Internet le schéma est beaucoup plus compliqué pour les raisons suivantes :

- *le nombre de réseau auxquels un routeur est connecté est généralement important*
- *les réseaux auxquels le routeur est relié peuvent être reliés à d'autres réseaux que le routeur ne connaît pas directement*

Ainsi, les routeurs fonctionnent grâce à des tables de routage et des protocoles de routage, selon le modèle suivant :

- *le routeur reçoit une trame provenant d'une machine connectée à un des réseaux auquel il est rattaché*
- *les datagrammes sont transmis à la couche IP*
- *le routeur regarde l'en-tête du datagramme*
- *si l'adresse IP de destination appartient à l'un des réseaux auxquels une des interfaces du routeur est rattaché, l'information doit être envoyée à la couche 4 après que l'en-tête IP ait été désencapsulée (enlevée)*
- *si l'adresse IP de destination fait partie d'un réseau différent, le routeur consulte sa table de routage, une table qui définit le chemin à emprunter pour une adresse donnée*
- *le routeur envoie le datagramme grâce à la carte réseau reliée au réseau sur lequel le routeur décide d'envoyer le paquet*

Ainsi, il y a deux scénarios, soit l'émetteur et le destinataire appartiennent au même réseau auquel cas on parle de *remise directe*, soit il y a au moins un routeur entre l'expéditeur et le destinataire, auquel cas on parle de *remise indirecte*.

Dans le cas de la remise indirecte, le rôle du routeur, notamment celui de la table de routage, est très important. Ainsi le fonctionnement d'un routeur est déterminé par la façon selon laquelle cette table de routage est créée :

- *si la table routage est entrée manuellement par l'administrateur, on parle de routage statique (viable pour de petits réseaux)*
- *si le routeur construit lui-même la table de routage en fonctions des informations qu'il reçoit (par l'intermédiaire de protocoles de routage), on parle de routage dynamique*

## 1.2. Tables de routage

La table de routage est une table de correspondance entre l'adresse de la machine visée et le nœud suivant auquel le routeur doit délivrer le message. En réalité il suffit que le message soit délivré sur le réseau qui contient la machine, il n'est donc pas nécessaire de stocker l'adresse IP complète de la machine : seul l'identificateur du réseau de l'adresse IP (c'est-à-dire l'ID réseau) a besoin d'être stocké.

La table de routage est donc un tableau contenant des paires d'adresses :

<i>adresse de destination</i>	<i>adresse du prochain routeur directement accessible</i>	<i>interface</i>
-------------------------------	---	------------------

Ainsi grâce à cette table, le routeur, connaissant l'adresse du destinataire encapsulée dans le message, va être capable de savoir sur quelle interface envoyer le message (cela revient à savoir quelle carte réseau utiliser), et à quel routeur, directement accessible sur le réseau auquel cette carte est connectée, remettre le datagramme. Ce mécanisme consistant à ne connaître que l'adresse du prochain maillon menant à la destination est appelé *routage par sauts successifs* (en anglais *next-hop routing*).

Cependant, il se peut que le destinataire appartienne à un réseau non référencé dans la table de routage. Dans ce cas, le routeur utilise un *routeur par défaut* (appelé aussi *passerelle par défaut*).

Voici, de façon simplifiée, ce à quoi pourrait ressembler une table de routage :

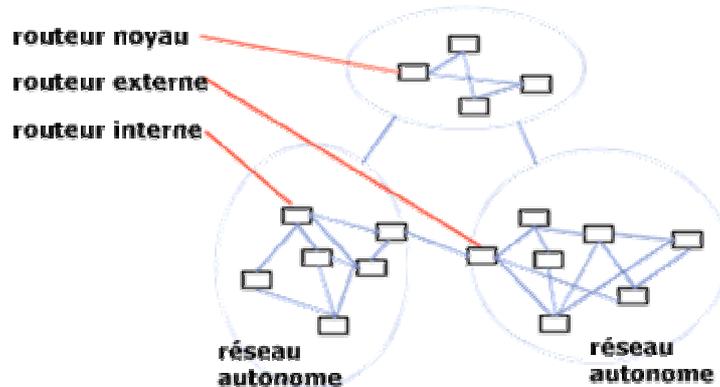
<i>adresse de destination</i>	<i>adresse du prochain routeur directement accessible</i>	<i>interface</i>
194.56.32.124	131.124.51.108	2
110.78.202.15	131.124.51.108	2
53.114.24.239	194.8.212.6	3
187.218.176.54	129.15.64.87	1

Le message est ainsi remis de routeur en routeur par sauts successifs, jusqu'à ce que le destinataire appartienne à un réseau directement connecté à un routeur. Celui-ci remet alors directement le message à la machine visée, etc.

Dans le cas du routage statique, c'est l'administrateur qui met à jour la table de routage. Dans le cas du routage dynamique, par contre, un protocole appelé *protocole de routage* permet la mise à jour automatique de la table afin qu'elle contienne à tout moment la route optimale.

### 1.3. Protocoles de routage

Internet est un ensemble de réseaux connectés. Par conséquent tous les routeurs ne font pas le même travail selon le type de réseau sur lequel ils se trouvent. En effet, il y a différents niveaux de routeurs, ceux-ci fonctionnent donc avec des protocoles différents :



- les routeurs noyaux sont les routeurs principaux car ce sont eux qui relient les différents réseaux
- les routeurs externes permettent une liaison des réseaux autonomes entre eux. Ils fonctionnent avec un protocole appelé EGP (Exterior Gateway Protocol) qui évolue petit à petit en gardant la même appellation
- les routeurs internes permettent le routage des informations à l'intérieur d'un réseau autonome. Ils s'échangent des informations grâce à des protocoles appelés IGP (Interior Gateway Protocol), tels que RIP et OSPF

#### Protocole RIP

RIP signifie *Routing Information Protocol* (protocole d'information de routage). Il s'agit d'un protocole de type *Vector Distance* (Vecteur Distance), c'est-à-dire que chaque routeur communique aux autres routeurs la distance qui les sépare (le nombre de sauts qui les sépare). Ainsi, lorsqu'un routeur reçoit un de ces messages il incrémente cette distance de 1 et communique le message aux routeurs directement accessibles. Les routeurs peuvent donc conserver de cette façon la route optimale d'un message en stockant l'adresse du routeur suivant dans la table de routage de telle façon que le nombre de saut pour atteindre un réseau soit minimal. Toutefois ce protocole ne prend en compte que la distance entre deux machines en termes de sauts, mais il ne considère pas l'état de la liaison afin de choisir la meilleure bande passante possible.

#### Protocole OSPF

OSPF (*Open Shortest Path First*) est plus performant que RIP et commence petit à petit à le remplacer. Il s'agit d'un protocole de type *protocole route-link* (que l'on pourrait traduire par *Protocole d'état des liens*), cela signifie que, contrairement à RIP, ce protocole n'envoie pas aux routeurs adjacents le nombre de sauts qui les sépare, mais l'état de la liaison qui les

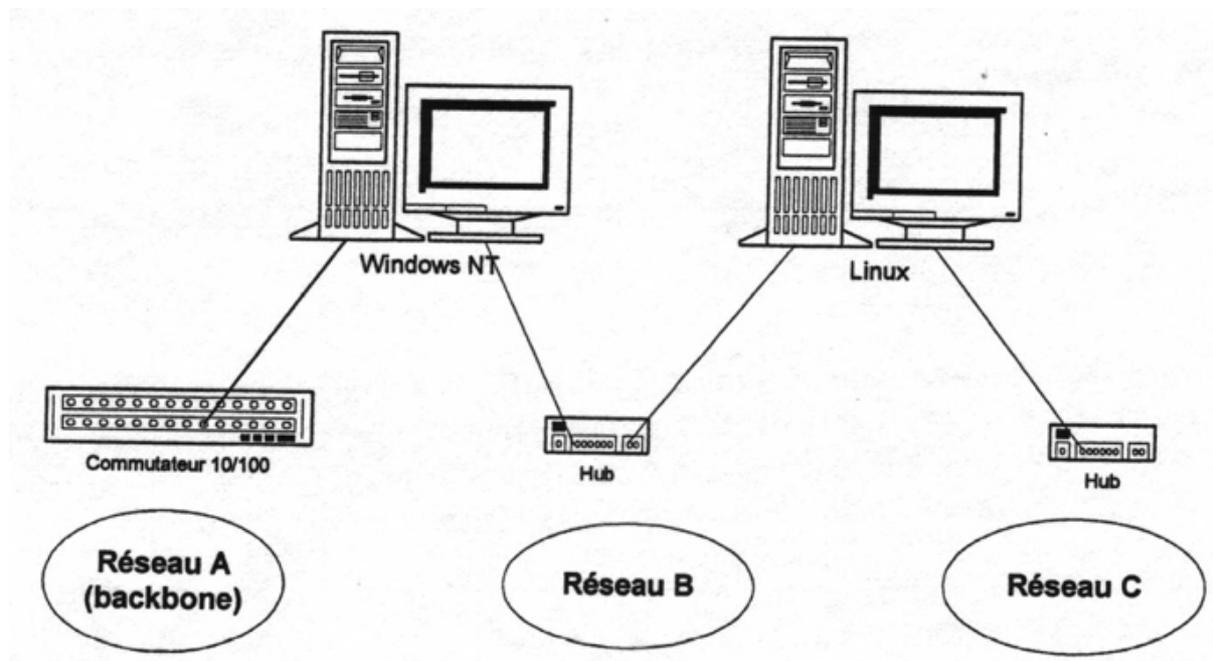
sépare. De cette façon, chaque routeur est capable de dresser une carte de l'état du réseau et peut par conséquent choisir à tout moment la route la plus appropriée pour un message donné. De plus, ce protocole évite aux routeurs intermédiaires d'avoir à incrémenter le nombre de sauts, ce qui se traduit par une information beaucoup moins abondante, ce qui permet d'avoir une meilleure bande passante utile qu'avec RIP.

## 2. Routage statique et dynamique avec RIPv1

### 2.1. Objectif

L'objectif est de configurer le routage statique puis dynamique à l'aide du protocole RIPv1 sur un réseau composé de machines fonctionnant sous Linux (distribution Debian) et Windows (version 2000) grâce notamment à l'utilisation de hubs/switches et de câbles réseau.

Un réseau doit être construit de manière à ce qu'il soit composé de deux machines fonctionnant en tant que routeur, par le biais d'une interface Ethernet :



### 2.2. Configuration IP des machines

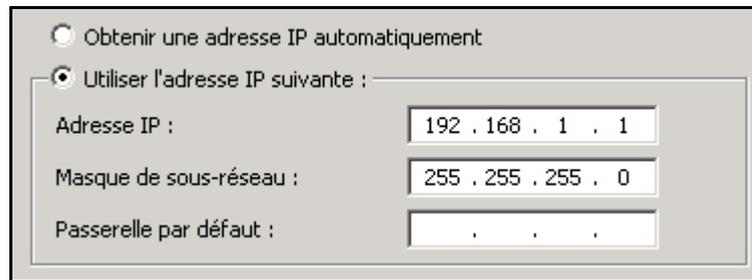
Celle-ci s'effectue suivant le plan d'adressage suivant :

- les adresses IP sont choisies dans la classe d'adresses privées C 192.168.x.y
- x et y sont choisis en fonction de l'interface réseau concernée (deux par machine) et de la disposition de la machine dans la salle pour éviter d'éventuels conflits d'adressage par la suite

La machine sous Linux Debian est configurée de la manière suivante :

- l'interface réseau (eth0) reliée au réseau B est paramétrée par la commande **ifconfig eth0 192.168.10.1 netmask 255.255.255.0 up** ; ceci monte l'interface réseau avec une adresse IP 192.168.10.1 et un masque de sous-réseau 255.255.255.0
- l'interface réseau (eth1) reliée au réseau C est paramétrée par la commande **ifconfig eth1 192.168.3.1 netmask 255.255.255.0 up**

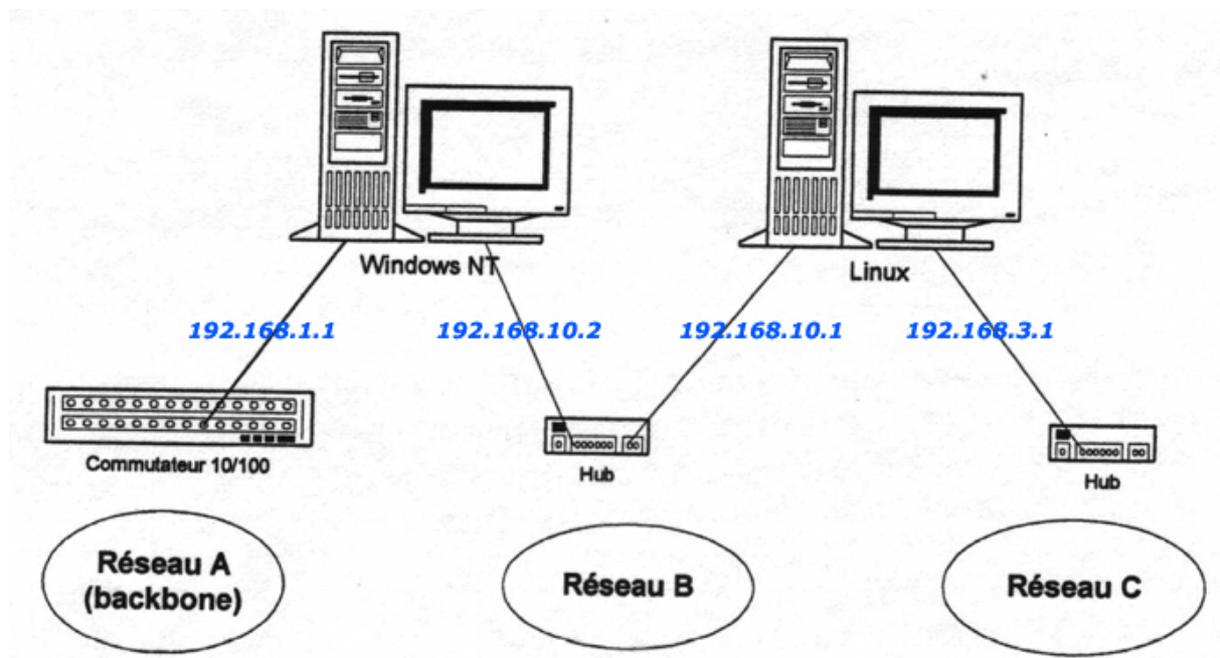
Sous Windows 2000 la configuration se fait par interface graphique (*panneau de configuration > connexions réseau*) :



- la première carte réseau (reliée au réseau A) est paramétrée avec une adresse IP 192.168.1.1 et un masque de sous-réseau 255.255.255.0 dans le protocole TCP/IP
- la seconde carte réseau (reliée au réseau B) est paramétrée avec une adresse IP 192.168.10.2 et un masque de sous-réseau 255.255.255.0

Ceci fait, on obtient la configuration suivante :

sous-réseau	netmask	@ sous-réseau	@ machine	OS machine
A	255.255.255.0	192.168.1.0	192.168.1.1	Windows
B	255.255.255.0	192.168.10.0	192.168.10.2	Windows
B	255.255.255.0	192.168.10.0	192.168.10.1	Linux
C	255.255.255.0	192.168.3.0	192.168.3.1	Linux



Nous avons réalisé différents tests de ping entre les deux machines, et leurs interfaces (cartes) réseau respectives par l'intermédiaire des quatre adresses IP à disposition.

Voici la liste récapitulative des adresses IP des machines et des sous-réseaux :

Depuis Linux on réalise un ping vers des adresses IP par la commande **ping adresse\_ip** :

- 192.168.10.2 (Windows – interface 2) => ça fonctionne
- 192.168.1.1 (Windows – interface 1) => ça ne fonctionne pas

De la même manière, on teste les adresses suivantes sous Windows 2000 :

- 192.168.10.1 (Linux – eth0) => ça fonctionne
- 192.168.3.1 (Linux – eth1) => ça ne fonctionne pas

L'explication est la suivante : lorsque l'on pingue une interface réseau reliée directement (physiquement) à l'interface source (par un hub/switch), l'interface destination répond correctement à la condition que les adresses IP soient de même classe et de même masque. En revanche, lorsque l'on pingue une interface qui ne se trouve pas sur le même chemin physique, celle-ci ne répond pas car la requête s'arrête avant, même si l'interface se trouve sur la même machine. Il faut donc effectuer un routage entre les deux cartes réseau afin que toutes les interfaces puissent communiquer correctement.

### 2.3. Routage statique

Le routage consiste à acheminer des informations à la bonne destination à travers un réseau. Selon le type du réseau, on envoie les données par paquets et on choisit leur chemin de différentes manières. Dans le cas du routage statique, il s'agit d'indiquer manuellement un chemin de manière arbitraire.

Pour configurer la machine sous Linux, on effectue les opérations suivantes :

- on utilise la commande **route** pour visualiser la table de routage
- on ajoute une route statique pour permettre à la machine d'atteindre l'interface extérieure (réseau A) de la machine sous Windows en utilisant la commande **route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.10.2**
- ceci signifie que Linux va utiliser l'interface passerelle 192.168.10.2 de la machine Windows pour accéder indirectement au réseau 192.168.1.0
- on visualise la nouvelle table de routage pour vérifier que la route a bien été créée

#### Table de routage sous Linux

destination	gateway	netmask	flags	metric	ref	use	interface
192.168.3.0	*	255.255.255.0	UH	0	0	0	Eth1
192.168.10.0	*	255.255.255.0	UH	0	0	0	Eth0
192.168.1.0	192.168.10.2	255.255.255.0	U	0	0	0	Eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	Lo

Sous Windows 2000, le mode opératoire est similaire :

- on visualise la table de routage en utilisant la commande **route print**
- on ajoute une route permettant d'atteindre l'interface extérieure de la machine Linux à l'aide de la commande **route add 192.168.3.0 mask 255.255.255.0 192.168.10.1**

- la passerelle est ici l'interface eth0 de la machine Linux à l'adresse 192.168.10.1
- on visualise la nouvelle table de routage pour vérifier que le chemin a bien été créé

#### Table de routage sous Windows

destination	gateway	netmask	flags	metric	ref	use	interface
192.168.1.0	*	255.255.255.0	UH	0	0	0	Eth0
192.168.10.0	*	255.255.255.0	UH	0	0	0	Eth1
192.168.3.0	192.168.10.1	255.255.255.0	U	0	0	0	Eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	Lo

Désormais, les différents pings qui ne fonctionnaient pas auparavant sont corrects. On peut par exemple accéder à l'adresse 192.168.3.1 (réseau C) depuis Windows ou accéder à l'adresse 192.168.1.1 (réseau A) depuis Linux.

#### Remarques :

- sous les deux machines on peut visualiser les échanges des pings à l'aide de deux outils principaux : la commande **tcpdump** sous Linux et le « moniteur réseau » sous Windows ; ils sont très utiles pour repérer et tracer toute activité suspecte sur un réseau
- sous Windows NT et 2000 un outil graphique RRAS (Routing and Remote Access : Routage et Accès Distant) permet de créer des routes statiques sans avoir à passer par une ligne de commande
- pour supprimer une route, on utilise la même commande que lors d'un ajout, mais en remplaçant **add** par **del** ; ceci permet de corriger certaines erreurs

## 2.4. Routage dynamique

Lorsqu'un réseau atteint une taille assez importante, il est très fastidieux de devoir ajouter les entrées dans les tables de routage à la main. La solution est le routage dynamique. Cela permet de mettre à jour les entrées dans les différentes tables de routage de façon dynamique et ainsi d'automatiser le routage.

Le routage dynamique peut s'appuyer sur divers protocoles : RIPv1, RIPv2, EIGRP, OSPF... Dans notre cas le protocole RIPv1 sera utilisé.

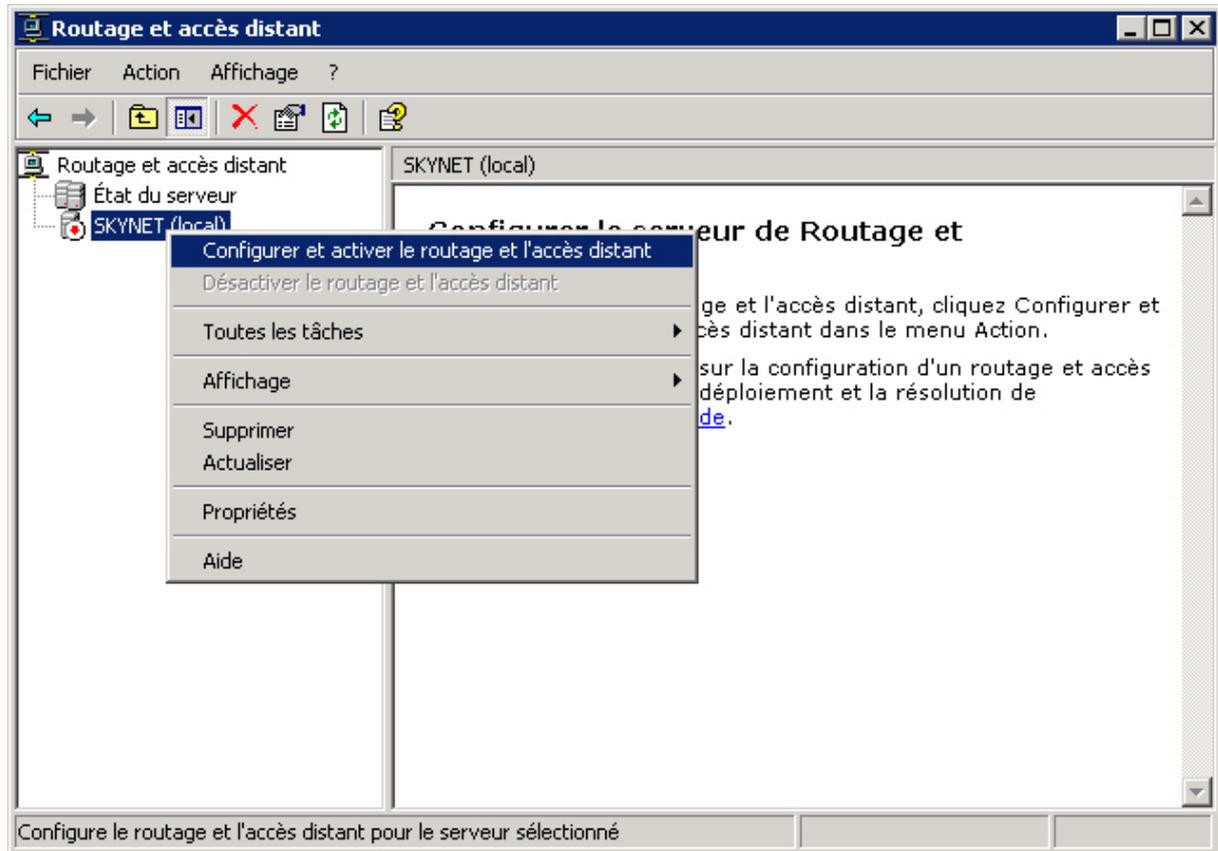
Tout d'abord, il est nécessaire de supprimer toutes les routes statiques créées précédemment sur les deux machines, afin d'éviter les conflits et surtout visualiser correctement les effets d'un tel routage. Ensuite, il faut activer le routage IP sur les deux machines.

Sous Linux, la configuration s'effectue de la manière suivante :

- on active le routage des paquets avec **# echo « 1 » > /proc/sys/net/ipv4/ip\_forward**
- on démarre le daemon de routage RIPv1 **routed** avec la prise en charge d'interfaces multiples

- on active la trace des échanges à l'aide de l'analyseur de protocoles **tcpdump**
- on visualise la table de routage à l'aide la commande **route**

Sous Windows, le routage se met en place par l'outil graphique RRAS :



- on cherche le nom de la machine
- dans la partie « routage IP », on lui ajoute le protocole RIP
- ensuite on ajoute une nouvelle interface avec le protocole RIPv1
- on répète l'opération pour les deux cartes (interfaces 192.168.1.1 et 192.168.10.2) afin que la machine sous Linux puisse accéder au backbone (réseau A)

Le routage silencieux permet de ne pas avoir à configurer manuellement des interfaces réseau chaque fois qu'un ordinateur du réseau interne veut accéder à un réseau externe. Cela permet d'automatiser le routage sans avoir à configurer chaque machine.

Une fois tout ceci mis en place, le routage fonctionne bien : la machine sous Windows accède correctement au réseau C et la machine sous Linux accède correctement au réseau A. Ces accès s'effectuent de manière dynamique ; si nous raccordions une machine au hub B avec une adresse IP de type 192.168.10.y, elle pourrait sans problème accéder aux réseaux A et C, et ce en ne configurant rien d'autre que l'adresse IP de la nouvelle machine.

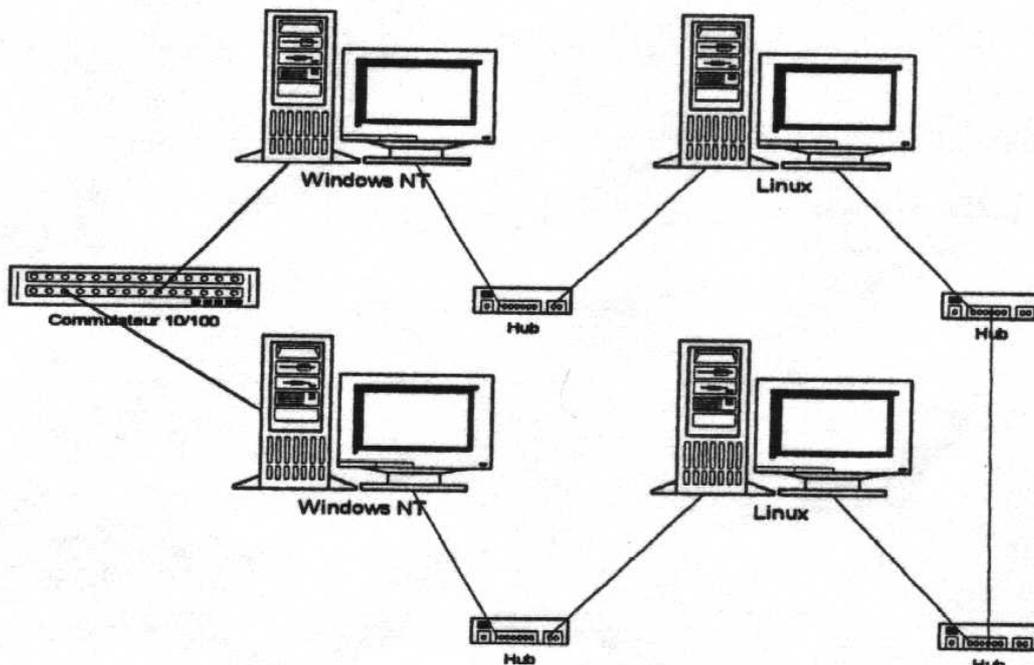
Les trames circulant sur le réseau peuvent être observées soit par la commande **tcpdump** (sous Linux), soit par le moniteur réseau (sous Windows). Lors d'un ping, les paquets

échangés entre les machines sont de deux types : ICMP et RIP. Les paquets ICMP correspondent aux pings (requêtes/réponses) respectifs des deux machines ; les paquets RIP mettent en évidence le protocole de routage utilisé, en l'occurrence RIPv1. A noter qu'il est possible de pinger continuellement sous Windows en utilisant la commande **ping -t @\_IP**.

Nous allons maintenant mettre en évidence deux lacunes propres au protocole RIP, à savoir la convergence des routes (un itinéraire n'est supprimé que lorsqu'il atteint une métrique de 16) et la possibilité de boucles.

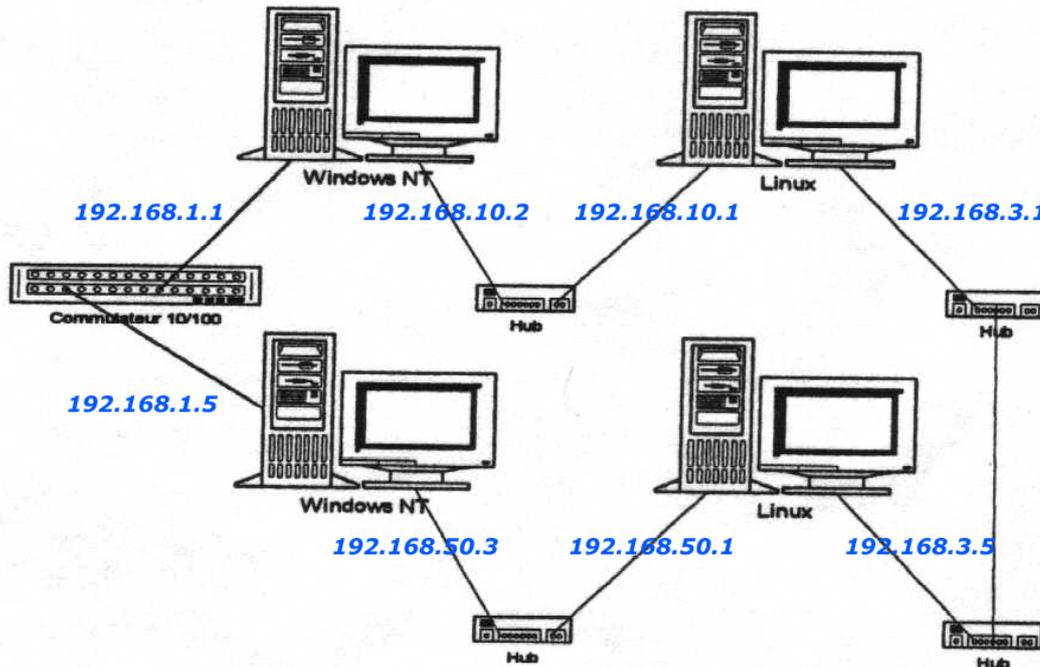
### 2.4.1. Convergence

Pour l'observer, nous allons mettre en réseau deux machines supplémentaires, organisées selon le schéma suivant :



Afin de garder une certaine cohérence dans le plan d'adressage, on répartit les adresses IP de la manière suivante :

sous-réseau	netmask	@ sous-réseau	@ machine	OS machine
A	255.255.255.0	192.168.1.0	192.168.1.1	Windows (1)
B	255.255.255.0	192.168.10.0	192.168.10.2	Windows (1)
B	255.255.255.0	192.168.10.0	192.168.10.1	Linux (2)
C	255.255.255.0	192.168.3.0	192.168.3.1	Linux (2)
C	255.255.255.0	192.168.3.0	192.168.3.5	Linux (3)
D	255.255.255.0	192.168.50.0	192.168.50.1	Linux (3)
D	255.255.255.0	192.168.50.0	192.168.50.3	Windows (4)
A	255.255.255.0	192.168.1.0	192.168.1.5	Windows (4)



Sur les deux machines ajoutées, on recommence les opérations de routage dynamique :

- sous Windows, on active RIP sur les interfaces réseau
- sous Linux, on démarre **routed**

Après vérification, toutes les machines sont atteignables entre elles ; le réseau fonctionne bien. On vérifie également le trajet d'un paquet sur le réseau lors d'un ping entre deux machines ; pour cela on utilise les commandes **tracert** sous Windows et **traceroute** sous Linux. Une fois les tables de routage complètes, on lance un ping continu depuis une machine vers une autre (sous Windows : **ping -t @\_IP** / sous Linux : **ping @\_IP**).

Ensuite on débranche un câble du réseau et on observe ce qu'il se passe grâce aux tables de routage mises à jour dynamiquement.

Lorsque l'on enlève un câble, les tables de routages ne sont pas modifiées immédiatement ; le protocole RIPv1 va effectuer 16 itérations de boucle avant de mettre à jour les tables de routage ; un itinéraire ne sera supprimé qu'au bout de ces 16 essais, soit le temps de convergence. Ceci représente un inconvénient majeur de RIPv1, là où RIPv2 ne réalise qu'une seule itération avant de supprimer la route.

#### 2.4.2. Mise en évidence de boucles dans RIP

Une boucle de routage apparaît lorsque deux passerelles routent les datagrammes destinés à une machine du réseau l'une vers l'autre. Lorsque plusieurs passerelles sont impliquées dans un boucle de routage, chacune d'elles route les datagrammes destinés à la machine vers la passerelle suivante dans la boucle. Lorsqu'un datagramme entre dans une boucle de routage il y tourne indéfiniment.

Sur les routeurs Windows, on effectue les opérations suivantes :

- on désactive pour le protocole RIP le découpage de l'horizon (split horizon), le traitement anti-poison (poison reverse) et les mises à jour déclenchées (triggered updates)
- on active les mises à jour toutes les 15 secondes sur l'un des routeurs et toutes les 90 secondes sur l'autre.

Lorsque l'on débranche un câble connecté à une des machines tournant sous Linux, on s'aperçoit que la table de routage n'est pas mise à jour avant la 16<sup>ième</sup> itération (métrique 16). La métrique de la route augmente donc logiquement jusqu'à 16, avant de retomber à 1 une fois la table correctement mise à jour, etc.

On observe donc logiquement un temps de réponse plus élevé lorsque l'on ping la machine Linux, sachant que le trajet va être modifié par l'apparition d'une boucle.

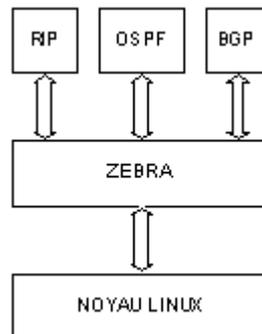
## 3. Routage dynamique avec RIPv2 et OSPF

### 3.1. Objectif

Le but de ce TP est d'étudier les protocoles de routage interne (RIPv2 et OSPF) ; de voir les mécanismes d'apprentissage, de découverte de la topologie et de la construction des tables de routage. Nous utiliserons l'application Zebra sous Linux pour la configuration RIP et OSPF.

### 3.2. Prise en main de Zebra

Zebra fonctionne sous Linux et BSD. C'est un routeur multi-protocole composé d'une suite de daemons, un par protocole de routage dynamique plus un démon central (zebra) utilisé pour le routage statique. De plus, lorsque le routage dynamique est mis en oeuvre, il est chargé de synthétiser dans une table de routage unique les informations rapportées par les autres daemons.



Zebra ne peut pas démarrer sans un fichier de configuration minimal que l'on doit saisir sur le routeur (machine Linux) :

- on édite le fichier `/etc/zebra/zebra.conf` à l'aide de la commande : `vi /etc/zebra/zebra.conf` ; le mot de passe est par défaut `zebra`
- on lance l'application zebra en mode daemon : `zebra -d -f /etc/zebra/zebra.conf`

Remarque : on obtient l'erreur suivante :

```
ZEBRA: can't create router advertisement socket: Address family not supported by protocol
```

Ceci est dû au fait que Zebra supporte le protocole IPv6, mais qu'il n'est pas activé sur le noyau, cela n'étant pas gênant pour la suite.

Pour configurer l'application, on se connecte via `telnet` au daemon :

- dans une console, on tape `telnet localhost zebra`, puis le mot de passe (`zebra`)
- pour passer en mode super-utilisateur, on tape `enable`, le mot de passe étant `zebra`
- pour connaître les interfaces disponibles sur le routeur, on tape `show interface`
- pour passer en mode configuration, on tape `configure terminal`

- pour configurer les interfaces du routeur, on les distingue par leur nom (*eth0*, *eth1*) par la commande suivante : **interface eth0**
- on donne une adresse IP conformément au plan d'adressage du TP précédent : **ip address 192.168.3.3/24**
- on sort de l'interface *eth0* : **exit**
- on configure *eth1* : **interface eth1**
- on donne l'adresse IP correspondante : **ip address 192.168.30.3**
- on sort de l'interface *eth1* : **exit**
- on sort du mode configuration : **exit**
- on sauvegarde les modifications : **write file**

## 3.2. Optimisation RIP : utilisation de RIPv2

### 3.2.1. Configuration sous Linux

Afin de lancer le protocole de routage dynamique RIPv2, on active le daemon *ripd* qui nécessite l'utilisation d'un fichier de configuration **/etc/zebra/ripd.conf** :

- on vérifie que *routed* ne fonctionne plus (**ps -A** pour voir la liste de tous les processus actifs)
- on édite le fichier de configuration *ripd* : **vi /etc/zebra/ripd.conf**
- on lance le processus *ripd* en mode *daemon* : **ripd -d -f /etc/zebra/ripd.conf**

Pour configurer le processus *ripd*, on se connecte via *telnet* au daemon puis on passe en mode super-utilisateur. On lance le protocole de routage RIPv2 par la série de commandes suivante :

- **conf term**
- **router rip**
- **version 2**
- **end**
- **write file**

Pour que le processus de routage soit vraiment actif, on indique au routeur les interfaces sur lesquelles on va exécuter ce processus :

- **conf term**
- **router rip**
- **network 192.168.0.0/16**
- **end**
- **write file**

Le routeur n'a aucune route à annoncer, on lui précise donc quelles informations il doit transmettre. La commande **redistribute** permet de choisir les préfixes qui seront transmis sur le réseau :

- **conf term**
- **router rip**

- *redistribute connected*
- *end*
- *write file*

### 3.2.2. Configuration sous Windows

Sous Windows, la configuration de RIPv2 s'effectue par le biais de l'outil graphique RRAS :

- *on cherche le nom de la machine*
- *dans la partie « routage IP », on lui ajoute le protocole RIP*
- *on ajoute une nouvelle interface avec le protocole RIPv2 multi-diffusion*
- *on répète l'opération pour les deux interfaces (192.168.1.3 et 192.168.30.2)*

Une fois les routeurs configurés, on vérifie que toutes les machines se reconnaissent et puissent accéder aux réseaux externes A et C. La table de routage sous Windows doit être du type :

```

C:\WINNT\System32\cmd.exe
Table de routage
=====
Liste d'Interfaces
0x1 ..... MS TCP Loopback interface
0x2 ...00 50 04 35 e5 78 ..... 3Com EtherLink PCI
0x3 ...00 60 97 0d 19 7d ..... 3Com 3C90x Ethernet Adapter
=====

Itinéraires actifs :
Destination réseau      Masque réseau      Adr. passerelle      Adr. interface      Métrique
127.0.0.0                255.0.0.0          127.0.0.1            127.0.0.1            1
192.168.1.0              255.255.255.0     192.168.1.3         192.168.1.3            1
192.168.1.3              255.255.255.255   127.0.0.1            127.0.0.1            1
192.168.1.255           255.255.255.255   192.168.1.3         192.168.1.3            1
192.168.30.0            255.255.255.0     192.168.30.2        192.168.30.2            1
192.168.30.2            255.255.255.255   127.0.0.1            127.0.0.1            1
192.168.30.255         255.255.255.255   192.168.30.2        192.168.30.2            1
224.0.0.0                224.0.0.0          192.168.1.3         192.168.1.3            1
224.0.0.0                224.0.0.0          192.168.30.2        192.168.30.2            1
255.255.255.255         255.255.255.255   192.168.30.2        192.168.30.2            1
=====

Itinéraires persistants :
Aucun
C:\>

```

Un des défauts de RIPv1 est de ne pas gérer les adresses de sous-réseaux. Mais de telles entrées peuvent être annoncées via une interface appartenant à ce sous-réseau pour pouvoir bénéficier du masque qui y est attaché et être correctement interprétées. Enfin, RIP met un temps assez long (quelques minutes) pour se stabiliser après la défaillance d'une liaison ou d'un routeur ce qui peut occasionner des boucles de routage.

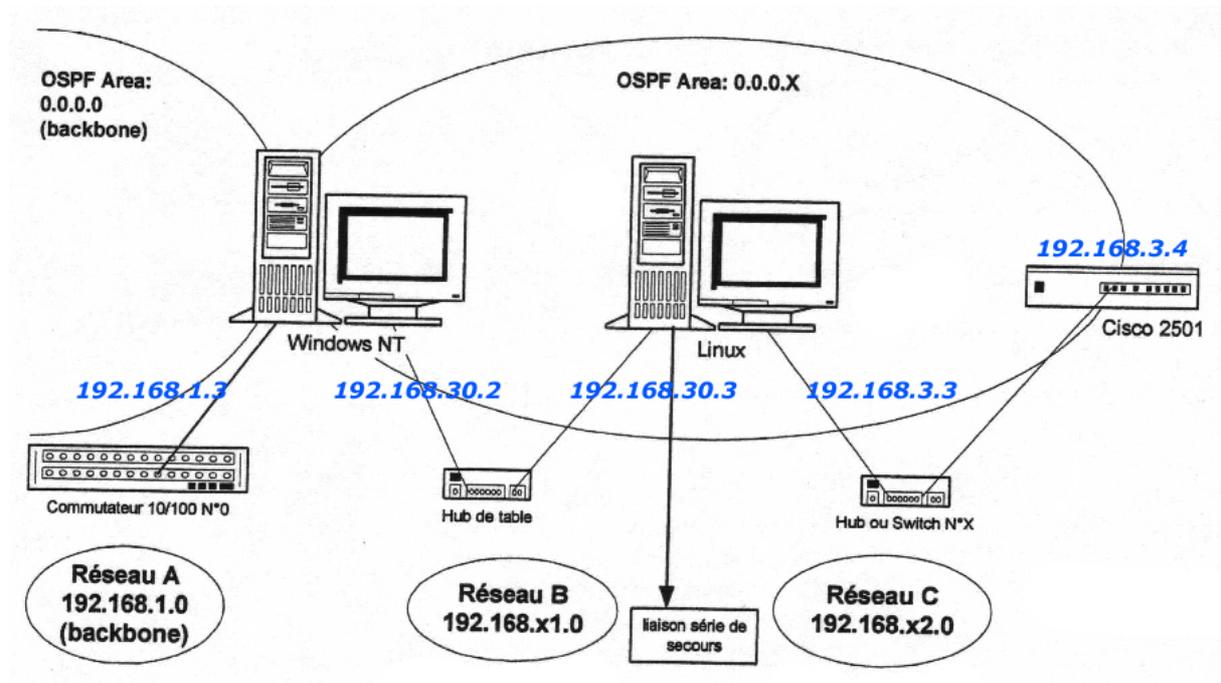
Les améliorations de RIPv2 sont les suivantes :

- *diffusion des masques de sous-réseaux associés aux adresses réseaux (RIPv1 n'utilise que les masques réseau par défaut)*
- *utilisation d'adresses multicast pour diffuser les vecteurs de distance au lieu d'adresses de broadcast, ce qui réduit l'encombrement sur le réseau*
- *support de l'authentification en transportant un mot de passe crypté avec MD5*
- *interopérabilité entre protocoles de routage en diffusant des routes apprises à partir d'autres protocoles*

### 3.3. Routage dynamique avec OSPF et routeur CISCO

OSPF est un protocole à l'état de lien : il transmet sur le réseau uniquement les modifications de la table de routage et permet de canaliser le trafic en découplant le réseau en zones ou *area*.

On construit le réseau de la manière suivante :



#### 3.3.1. Mise en place sous Windows 2000

Pour cela on utilise l'outil graphique RRAS :

- on supprime le protocole RIP de toutes les interfaces afin d'éviter d'éventuels conflits
- on définit le numéro de zone et on ajoute les réseaux qu'elle gère
- on désactive les mots de passe en clair
- on ajoute l'interface connectée au réseau interne
- on vérifie que le routeur n'est pas défini comme « routeur frontière de système autonome »

#### 3.3.2. Mise en place sous Linux Debian

Pour lancer le protocole de routage dynamique OSPF, il faut désactiver le protocole RIPv2 et activer le daemon `ospfd` qui nécessite l'utilisation d'un fichier de configuration `/etc/zebra/ospfd.conf` :

- `conf term`
- `no router rip`
- `end`
- `ospfd -d -f /etc/zebra/ospfd.conf`

On effectue la séquence suivante pour OSPF :

- *router ospf*
- *no redistribute rip*
- *network 192.168.30.0/24 area 0.0.0.3*
- *network 192.168.3.0/24 area 0.0.0.3*
- *end*
- *write file*

### 3.3.3. Mise en place sur le routeur Cisco 2501

Pour configurer le routeur, on utilise *telnet* de la même manière que pour se connecter à Zebra. Malheureusement la commande **no router rip** ne fonctionne pas (plus précisément la commande **router**) dans notre cas, ce qui rend toute configuration OSPF impossible.

## 4. Conclusion

*Ces deux TPs nous ont permis de voir de manière pratique différentes facettes du routage. Ainsi nous avons pu découvrir plusieurs façons de réaliser un tel routage, avec différents protocoles et dans différentes situations. Nous avons réalisé cette approche à la fois sous environnement Windows et sous système Linux, et ce avec différents protocoles tels que RIP (RIPv1, RIPv2) et OSPF, que nous n'avons pas pu faire fonctionner correctement.*

*Comme expliqué dans la partie 1.3, le protocole OSPF est moins lourd que le protocole RIP en terme de trafic car il y a moins d'information qui circule entre les routeurs. De plus, le protocole OSPF ne nécessite pas l'incrémentation du nombre de sauts.*

*Néanmoins, l'algorithme qu'utilise le protocole OSPF est relativement lourd en terme de calculs ce qui se traduit par une charge importante en ressources système. Pour limiter cet inconvénient, il faudrait réduire le nombre de routeurs voisins pour diminuer l'utilisation des ressources.*

## **ANNEXES**

### **Plan d'adressage IP (TP2)**

Chaque table possède un numéro, celui-ci sera utilisé pour composer l'adresse IP des différentes machines.

#### **Réseau A**

Toutes les machines Windows sont connectées au backbone. Elles posséderont des adresses IP choisies dans la classe  $192.168.1.X - 255.255.255.0$  avec X correspondant au numéro de table.

#### **Réseau B**

On dispose d'une plage d'adresses de classe C  $192.168.Y.0 - 255.255.255.0$  avec  $Y = 10 * X$  où X correspond au numéro de table. On choisit ensuite l'adresse libre du début de la plage pour Linux et celle de fin de plage pour Windows.

#### **Réseau C**

Toutes les machines Linux possèdent des adresses IP choisies dans la classe  $192.168.3.X - 255.255.255.0$  avec X correspondant au numéro de table.

### **Présentation du routeur Cisco 2501 (TP3)**

Ce routeur possède trois interfaces : ethernet 10BaseT et deux liens série synchrone haut débit (jusqu'à 2 Mbits/s), un port de console pour y connecter un terminal, un port auxiliaire pour les backups. La version de l'IOS est 11.3 ; l'accès console nécessite 9600,8,N,1 avec contrôle de flux matériel ; le mot de passe pour l'accès à distance est cisco. Le mot de passe du mode privilégié (ou mode enable) qui permet la configuration complète du routeur est *enable*.

Pour configurer le routeur il existe plusieurs possibilités :

- un utilitaire graphique sous Windows (*configmaker*) pour la configuration de base (adresse IP, interface) mais pas pour le routage OSPF
- connexion avec un terminal sur la liaison série
- un *telnet* sur le routeur
- utilisation d'un navigateur http

## Aperçu des commandes IOS

- visualisation de la configuration actuelle : **show running-config** en mode privilégié (**enable**)
- configuration du routeur : **configure** en mode privilégié
- aide sur la syntaxe des commandes : **commande ?**
- suppression d'une configuration : **no configure** en mode privilégié
- sortie du mode configure : **exit**

## Exemple d'un fichier de configuration Cisco

```
hostname Cisco2501
no snmp-server location
no snmp-server contact
!
enable password enable
!
!
ip subnet-zero
no ip domain-lookup
ip routing
!
interface Ethernet 0
no shutdown
description connected to EthernetLAN
ip address 192.168.99.99
255.255.255.0
keepalive 10
!
interface Serial 0
shutdown
no description
no ip address
!
interface Serial 1
shutdown
no description
no ip address
!
!
router rip
version 2
network 192.168.99.0
no auto-summary
!
ip http server
no ip name-server
snmp-server community public ro
!
line console 0
password cisco
login
exec-timeout 0 0
!
line vty 0 4
password cisco
login
!
end
```