

# Protocoles Réseaux – TP4

---

**BONY Simon – MBOW Moustapha**

11 mai 2012

# Table des matières

---

<b>Introduction.....</b>	<b>3</b>
<b>I. Examen des connexions d'un routeur .....</b>	<b>4</b>
<b>II. Configuration IP des interfaces Ethernet.....</b>	<b>6</b>
<b>III. Configuration des interfaces série et vérification de la table de routage .....</b>	<b>9</b>
<b>IV. Vérification de la connectivité des périphériques directement connectés .</b>	<b>11</b>
<b>V. Suppression et configuration des routes statiques.....</b>	<b>13</b>
<b>VI. Configuration d'une route statique par défaut .....</b>	<b>16</b>
<b>VII. Câblage et configuration d'un routeur Cisco .....</b>	<b>19</b>
<b>VIII.Synthèse .....</b>	<b>21</b>
<b>Conclusion .....</b>	<b>24</b>

# Introduction

---

L'objectif de ce TP est d'étudier les systèmes de routage Cisco. Cisco est le leader mondial de la vente de matériels réseaux. De ce fait, leurs équipements sont très répandus et leur compréhension est un atout certain.

Afin de prendre connaissance de ces systèmes, nous effectuerons à la fois des manipulations directes de matériels Cisco et des simulations de réseaux à l'aide du logiciel Packet Tracer fournit par Cisco.

Nous commencerons donc par traiter la simulation de réseaux pour finir par la manipulation directe de matériels.

## I. Examen des connexions d'un routeur

L'objectif des manipulations que nous effectuerons au cours des 6 prochaines parties est de créer et rendre fonctionnel le réseau suivant :

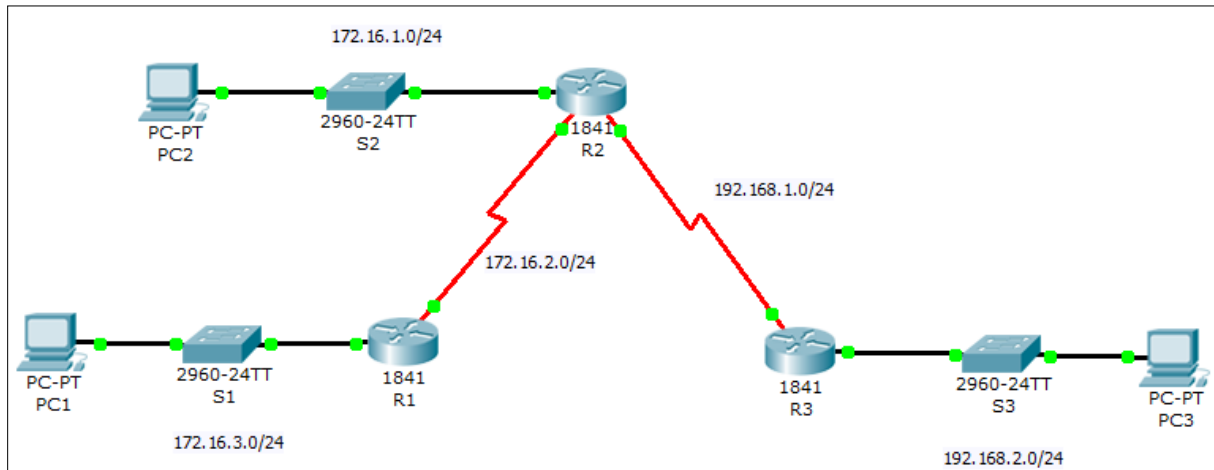


Fig 1: Réseau à mettre en place

La première étape va donc être de placer dans packet tracer les différentes machines et liaisons présentes. Pour cela, il nous suffit de sélectionner tour à tour les éléments désirés dans le catalogue proposé et de les glisser à leur emplacement respectif.

Une fois cette opération, nous devons ajouter aux routeurs un module **WIC-2T** pour qu'il soit compatible avec le travail exigé. Dès lors, nous pouvons adresser à chaque machine un nom afin qu'il corresponde à leur étiquette.

Nous lions alors les machines avec les câbles adéquats et nous obtenons alors la configuration suivante :

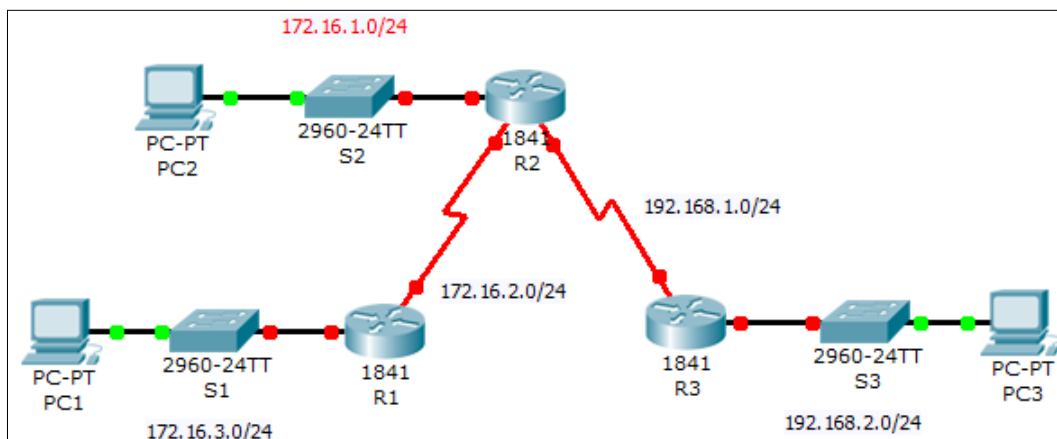


Fig 2: Installation des éléments

En faisant le point à ce niveau, nous pouvons expliquer certains choix concernant les éléments utilisés.

Tout d'abord, nous pouvons identifier **5 sous réseaux** (**172.16.1.0/24**, **172.16.2.0/24**, **172.16.3.0/24**, **192.168.1.0/24** et **192.168.2.0/24**).

Ensuite, nous remarquons la présence d'un commutateur entre les PC et les routeurs. Cela est dû au fait que le routeur ne possède qu'un nombre limité de ports physiques. Tous les PC d'un même sous réseau ne peuvent donc pas être directement reliés au routeur. Un commutateur va donc « centraliser » toutes ces connexions.

Les routeurs sont reliés entre eux par des câbles séries **DCE** afin de libérer des ports *Ethernet* souvent très demandés. De plus, la liaison série apportera une meilleure fiabilité qu'une connexion *Ethernet* qui est en *best-effort*.

Nous remarquons aussi la présence de câbles droits pour relier les routeurs aux commutateurs et les PC aux commutateurs. Cela est dû au niveau du matériel : un câble croisé n'est utilisé que pour relier des équipements de même niveau. Or, le PC est de niveau 1, le commutateur de niveau 2 et le routeur de niveau 3. Un câble droit est donc le plus adapté pour relier ces équipements.

## II. Configuration IP des interfaces Ethernet

Nous souhaitons maintenant attribuer les adresses IP de chaque machine.

Nous commençons avec les PC en configurant manuellement leur adresse via leur onglet **Config**. Nous leur attribuons les configurations suivantes :

PC1 : Adresse IP : **172.16.3.10**  
Masque : **255.255.255.0**  
Passerelle : **172.16.3.1**  
PC2 : Adresse IP : **172.16.1.10**  
Masque : **255.255.255.0**  
Passerelle : **172.16.1.1**  
PC3 : Adresse IP : **192.168.2.10**  
Masque : **255.255.255.0**  
Passerelle : **192.168.2.1**

Nous remarquons que pour le PC1 et le PC2, bien que leur adresse IP soit de classe B, nous utilisons un masque de sous réseau de classe C. Ce choix s'explique par le fait qu'avec un masque de classe B, PC1 et PC2 seraient dans le même sous réseau, **172.16.0.0/16**. Le masque de classe C nous permet ainsi de différencier les 2 sous réseaux.

Nous continuons avec les routeurs en utilisant le mode **CLI (Command Line Interface)** qui correspond à l'invite de commande des routeurs. Celui-ci va nous permettre de les configurer par lignes de commande. Nous devons donc, pour chaque routeur exécuter la suite de commandes suivante :

```
Router>enable
Router#config t
Router(config)#interface fa 0/0
Router(config-if)#ip address <IP address> 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#^Z
Router#copy run start
```

*Fig 3: Commandes de paramétrage IP des routeurs*

La commande **enable** nous permet de passer en mode root. Ce mode est reconnaissable par la modification du caractère '**>**' par '**#**'. La commande **config t** nous permet de passer en mode configuration, reconnaissable par la modification du nom *Router* par *Router(config)* et la commande **interface fa 0/0** nous permet de configurer l'interface *fastEthernet 0/0*. Ce dernier mode est reconnaissable à la modification du nom *Router(config)* par *Router(config-if)*.

La commande **ip adress <IP adress> 255.255.255.0** permet la configuration IP du routeur et **no shutdown** d'activer l'interface. Nous sortons alors du mode de configuration avec **CRTL+Z** puis nous enregistrons la configuration avec **copy run start**.

Nous utilisons donc ces commandes sur chaque routeur pour les configurer avec l'adresse IP **172.16.3.1** pour le routeur R1, **172.16.1.1** pour le routeur R2 et **192.168.2.1** pour le routeur R3.

Nous pouvons alors vérifier la mise en service de ces configurations à l'aide de la commande **show ip interface brief**

```
R1:
Router#show ip interface brief
Interface                IP-Address      OK? Method Status  Protocol
FastEthernet0/0          172.16.3.1      YES manual up      up
```

Fig 4: Vérification de la configuration du routeur R1

Le protocole *ARP* nous permet ainsi de vérifier le bon fonctionnement de l'adressage IP mis en place.

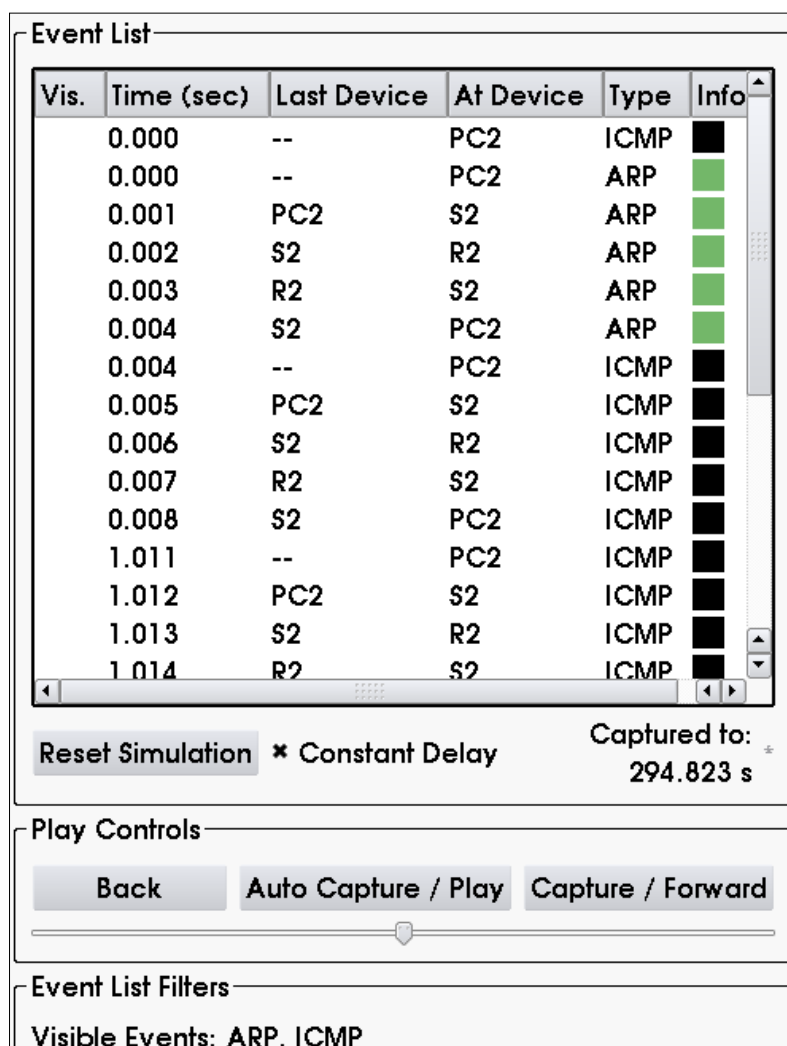
PC1, relié à R1 par un commutateur, ne possède aucune entrée dans sa table *ARP* (**arp -a**). C'est aussi le cas de R1 qui ne possède que sa propre interface dans la table *ARP* (**show arp**). Nous effectuons donc un ping de PC1 vers R1 pour remarquer dans les tables *ARP* de chacun la présence de l'autre, confirmant la bonne configuration IP.

```
PC>arp -a
Internet Address      Physical Address      Type
172.16.3.1            000c.cf38.9a01        dynamic

Router#show arp
Protocol  Address      Age(min)  Hardware Addr  Type Interface
Internet  172.16.3.1    -         000C.CF38.9A01  ARPA FastEthernet0/0
Internet  172.16.3.10  0         0001.C73B.E979  ARPA FastEthernet0/0
```

Fig 5: Vérification de la configuration par ARP

Nous pouvons capturer ces trames *ARP* afin de les étudier à l'aide de l'outil **Auto Capture/Play** dans le cas identique d'un ping entre PC2 et R2.



Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.000	--	PC2	ICMP	
	0.000	--	PC2	ARP	
	0.001	PC2	S2	ARP	
	0.002	S2	R2	ARP	
	0.003	R2	S2	ARP	
	0.004	S2	PC2	ARP	
	0.004	--	PC2	ICMP	
	0.005	PC2	S2	ICMP	
	0.006	S2	R2	ICMP	
	0.007	R2	S2	ICMP	
	0.008	S2	PC2	ICMP	
	1.011	--	PC2	ICMP	
	1.012	PC2	S2	ICMP	
	1.013	S2	R2	ICMP	
	1.014	R2	S2	ICMP	

Reset Simulation \* Constant Delay      Captured to: 294.823 s

Play Controls: Back    Auto Capture / Play    Capture / Forward

Event List Filters: Visible Events: ARP, ICMP

Fig 6: Capture des trames ARP et ICMP

Nous remarquons que les trames *ARP* sont émises avant les trames *ICMP*. Cela est dû au besoin de PC2 de connaître l'adresse MAC de R2 pour lui adresser les trames *ICMP*.



### III. Configuration des interfaces série et vérification de la table de routage

Maintenant que les liaisons internes des sous réseaux sont opérationnelles, il nous faut relier les sous réseaux eux mêmes.

Dans ce cas, une configuration des routeurs ainsi que leurs interfaces série devient nécessaire. Ceci nous amène à utiliser le même enchainement de commandes que pour la configuration IP à deux détails prêt :

```
Router>enable
Router#config t
Router(config)#interface <interface>
Router(config-if)#ip address <IP address> 255.255.255.0
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
Router(config-if)#^Z
Router#copy run start
```

Fig 7: Commandes de paramétrage des interfaces série

Nous devons préciser l'horloge de l'interface avec la commande **clock rate 64000**. Cela permet à deux routeurs connectés ensemble de savoir à quelle vitesse ils doivent émettre/réceptionner les données. Dans cette liaison, un des routeurs sera **DCE (Data Communication Equipement)**. Son rôle sera de synchroniser toutes les données à l'aide du *clock rate*. Tous les routeurs sont des équipements DCE tandis que les PC par exemple sont des **DTE (Data Terminal Equipement)**. Ils généreront les données passant sur la liaison. Dans le cas d'une liaison série, un des routeurs sera donc le DTE et l'autre le DCE.

Nous configurons donc les différentes interfaces de chaque routeur avec les adresses IP suivantes :

```
R1:  interface s/0/0/0: 172.16.2.1
R2:  interface s/0/0/0: 172.16.2.2
      interface s/0/0/1: 192.168.1.2
R3:  interface s/0/0/1: 192.168.1.1
```

Nous pouvons alors vérifier la prise en compte de la nouvelle configuration avec la commande **show ip interface brief** :

```
R2:
Router#show ip interface brief
Interface      IP-Address      OK? Method Status  Protocol
Serial0/0/0    172.16.2.2      YES manual up      up
Serial0/0/1    192.168.1.2     YES manual up      up
```

*Fig 8: Etat des interfaces série du routeur R2*

Etudions maintenant les tables de routage. Pour ce faire, nous utilisons la commande **show ip route**.

```
Router#show ip route
      172.16.0.0/24 is subnetted, 2 subnets
C       172.16.1.0 is directly connected, FastEthernet0/0
C       172.16.2.0 is directly connected, Serial0/0/0
C      192.168.1.0/24 is directly connected, Serial0/0/1
```

*Fig 9: routes du routeur R2*

Nous remarquons que les seules entrées sont les sous réseaux directement connectés au routeur.

Nous souhaitons étudier la gestion des routes par le routeur. Nous désactivons ainsi les deux interfaces série à l'aide de la commande **shutdown** en mode configuration d'interfaces. Nous remarquons alors à l'aide de la commande **show ip route** que les deux routes vers les sous réseaux connectés sur les interfaces série ont été effacées. Nous utilisons alors la commande **debug ip routing** pour que le routeur recherche lui-même les routes disponibles. Nous réactivons les deux interfaces (**no shutdown**) et nous remarquons alors avec la commande **show ip route** la présence des deux routes séries.

## IV. Vérification de la connectivité des périphériques directement connectés

Nous cherchons maintenant à tester les machines accessibles depuis un point donné.

Nous commençons avec PC1 et R1. En effectuant un ping vers l'adresse **172.16.3.1** et **172.16.2.1**, nous confirmons la capacité de PC1 à atteindre les différentes interfaces du routeur R1. Cependant, en tentant un ping vers l'interface série de R2 reliant R1 et R2 (**172.16.2.2**), nous remarquons que le ping n'aboutit pas. Effectivement, le routeur R2 n'est pas accessible en liaison directe et PC1 ne possède aucune règle de routage pour accéder à cet autre sous réseau.

```
PC>ping 172.16.3.1
Pinging 172.16.3.1 with 32 bytes of data:
...
Ping statistics for 172.16.3.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

PC>ping 172.16.2.1
Pinging 172.16.2.1 with 32 bytes of data:
...
Ping statistics for 172.16.2.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

PC>ping 172.16.2.2
Pinging 172.16.2.2 with 32 bytes of data:
...
Ping statistics for 172.16.2.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

*Fig 10: pings depuis PC1*

Le tableau ci-dessous récapitule les interfaces accessibles depuis chaque PC:

X = OK	172.16.3.1	172.16.2.1	172.16.1.1	172.16.2.2	192.168.1.2	192.168.1.1	192.168.2.1
PC1	X	X					
PC2			X	X	X		
PC3						X	X

Nous remarquons donc que pour chaque PC, celui-ci ne peut atteindre que les interfaces de son routeur. De ce fait, PC3 ne peut pas pinger les interfaces série de R2.

De la même manière, nous effectuons plusieurs pings depuis chaque routeur vers les autres machines. Nous obtenons alors le tableau suivant :

X = OK	PC1	PC2	PC3
R1	X		
R2		X	
R3			X

Le tableau ci-dessous récapitule les interfaces accessibles depuis chaque routeur:

X = OK	172.16.3.1	172.16.2.1	172.16.1.1	172.16.2.2	192.168.1.2	192.168.1.1	192.168.2.1
R1	X	X		X			
R2		X	X	X	X	X	
R3					X	X	X

Nous en déduisons donc que dans le cas des routeurs, ceux-ci ne peuvent communiquer qu'avec les machines et interfaces directement connectées. R1 ne peut accéder à une interface de R2 qui n'est pas sous le même sous réseau.

## V. Suppression et configuration des routes statiques

Nous avons remarqué dans la partie précédente l'absence de route afin de permettre par exemple la communication entre les PC. C'est ce que nous allons maintenant configurer

Nous allons commencer par étudier le mode de configuration actuel du routage statique à l'aide de la commande **show running-config**.

```
R1#show running-config
...
ip classless
ip route 172.16.1.0 255.255.255.0 172.16.2.2
ip route 192.168.1.0 255.255.255.0 172.16.2.2
ip route 192.168.2.0 255.255.255.0 172.16.2.2
```

*Fig 11: Configuration actuelle du routage statique*

Les routeurs sont donc en mode **ip classless**, c'est-à-dire qu'ils recherchent les routes sans considération de la classe.

Nous pouvons ensuite étudier les routes présentes avec **show ip route** :

```
R1#show ip route
172.16.0.0/24 is subnetted, 3 subnets
S    172.16.1.0 [1/0] via 172.16.2.2
C    172.16.2.0 is directly connected, Serial0/0/0
C    172.16.3.0 is directly connected, FastEthernet0/0
S    192.168.1.0/24 [1/0] via 172.16.2.2
S    192.168.2.0/24 [1/0] via 172.16.2.2
```

*Fig 12: Routes du routeur R1*

Nous remarquons que **show ip route** nous indique la présence des 5 sous réseaux dans les routes là où **show running-config** ne nous donner que les routes indirectement connectées. Bien sûr, **show running-config** nous retourne aussi d'autres informations sur la configuration en cours.

Nous pouvons donc en théorie pinguer un PC vers un autre :

```
PC>ping 172.16.1.10

Pinging 172.16.1.10 with 32 bytes of data:

Request timed out.
Reply from 172.16.1.10: bytes=32 time=18ms TTL=126
Reply from 172.16.1.10: bytes=32 time=11ms TTL=126
Reply from 172.16.1.10: bytes=32 time=16ms TTL=126

Ping statistics for 172.16.1.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 11ms, Maximum = 18ms, Average = 15ms
```

*Fig 13: ping de PC1 vers PC2*

Nous souhaitons maintenant supprimer ces routes afin de les reconfigurer manuellement en utilisant cette fois-ci l'argument de l'interface de sortie.

Commençons par supprimer les routes avec les commandes suivantes :

- R1(config)#**no ip route 172.16.1.0 255.255.255.0 172.16.2.2**
- R1(config)#**no ip route 192.168.1.0 255.255.255.0 172.16.2.2**
- R1(config)#**no ip route 192.168.2.0 255.255.255.0 172.16.2.2**
- R2(config)#**no ip route 172.16.3.0 255.255.255.0 172.16.2.1**
- R2(config)#**no ip route 192.168.2.0 255.255.255.0 192.168.1.1**
- R3(config)#**no ip route 172.16.1.0 255.255.255.0 192.168.1.2**
- R3(config)#**no ip route 172.16.2.0 255.255.255.0 192.168.1.2**
- R3(config)#**no ip route 172.16.3.0 255.255.255.0 192.168.1.2**

Puis nous adressons les mêmes routes en remplaçant les adresses IP de sorties par le nom des interfaces correspondantes.

- R1(config)#**ip route 172.16.1.0 255.255.255.0 s0/0/0**
- R1(config)#**ip route 192.168.1.0 255.255.255.0 s0/0/0**
- R1(config)#**ip route 192.168.2.0 255.255.255.0 s0/0/0**
- R2(config)#**ip route 172.16.3.0 255.255.255.0 s0/0/0**
- R2(config)#**ip route 192.168.2.0 255.255.255.0 s0/0/1**
- R3(config)#**ip route 172.16.1.0 255.255.255.0 s0/0/1**
- R3(config)#**ip route 172.16.2.0 255.255.255.0 s0/0/1**
- R3(config)#**ip route 172.16.3.0 255.255.255.0 s0/0/1**

Nous enregistrons les modifications avec **CTRL+Z** et **copy run start**. Nous pouvons ainsi vérifier leur prise en compte avec **show running-config** et/ou **show ip route**.

```
R1#show ip route
      172.16.0.0/24 is subnetted, 3 subnets
S       172.16.1.0 is directly connected, Serial0/0/0
C       172.16.2.0 is directly connected, Serial0/0/0
C       172.16.3.0 is directly connected, FastEthernet0/0
S      192.168.1.0/24 is directly connected, Serial0/0/0
S      192.168.2.0/24 is directly connected, Serial0/0/0
```

*Fig 14: Table de routage de R1*

Nous remarquons la bonne prise en compte des interfaces de sorties.

Nous pouvons alors vérifier le bon fonctionnement avec un ping entre les différents PC.

```
PC>ping 172.16.1.10

Pinging 172.16.1.10 with 32 bytes of data:

Reply from 172.16.1.10: bytes=32 time=18ms TTL=126
Reply from 172.16.1.10: bytes=32 time=22ms TTL=126
Reply from 172.16.1.10: bytes=32 time=20ms TTL=126
Reply from 172.16.1.10: bytes=32 time=7ms TTL=126

Ping statistics for 172.16.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 7ms, Maximum = 22ms, Average = 16ms
```

*Fig 15: ping de PC1 vers PC2*

Nous avons ici utilisé des routes statiques. Celles-ci ne seront jamais modifiées sans l'intervention de l'administrateur. En cas de suppression d'une machine, si le réseau est impacté, les changements seront à effectuer manuellement.

Sur des réseaux de plus grande ampleur, insérer une route statique pour chaque machine est bien trop exigeant. La recherche de route récursive est donc utilisée la plupart du temps. Elle consiste à ne connaître que la route vers le prochain routeur. Ainsi, sans connaître l'existence même d'un sous-réseau, un routeur peut envoyer un paquet en direction d'un routeur qu'il sait plus proche d'un tel sous-réseau. Celui-ci fera de même jusqu'à un routeur connaissant le sous-réseau.

## VI. Configuration d'une route statique par défaut

Dans la configuration précédente, tous les sous réseaux étaient présents dans la table de routage. Cependant, à l'aide de la recherche récursive, il est possible de réduire le nombre de sous-réseaux décrits. Effectivement, en utilisant une route par défaut, tous les paquets ne possédant pas de route plus précise pourront utiliser cette route-ci. Il est donc possible de « factoriser » plusieurs routes.

En partant de la configuration précédente, nous modifions la table de routage de R3 afin d'utiliser une route par défaut. Nous utilisons donc les commandes suivantes pour supprimer les anciennes routes :

```
no ip route 172.16.1.0 255.255.255.0 s0/0/1
```

```
no ip route 172.16.2.0 255.255.255.0 s0/0/1
```

```
no ip route 172.16.3.0 255.255.255.0 s0/0/1
```

Nous ajoutons alors la route par défaut **176.16.0.0** avec pour masque **255.255.252.0** sur l'interface **s0/0/1** :

```
ip route 172.16.0.0 255.255.252.0 s0/0/1
```

Ainsi, tous les paquets à destination d'un sous réseau allant de **176.16.0.0** vers **176.16.3.0** emprunteront cette route.

Nous pouvons alors enregistrer les modifications avec **CTRL+Z** et **copy run start** et les visionner avec **show running-config** par exemple :

```
R3#show running-config  
  
...  
  
ip classless  
ip route 172.16.0.0 255.255.252.0 Serial0/0/1
```

Fig 16: Visualisation de la table de routage de R3



Nous vérifions aussi que cette configuration est fonctionnelle avec un ping de PC3 vers PC1 par exemple :

```
PC>ping 172.16.3.10

Pinging 172.16.3.10 with 32 bytes of data:

Reply from 172.16.3.10: bytes=32 time=12ms TTL=125
Reply from 172.16.3.10: bytes=32 time=25ms TTL=125
Reply from 172.16.3.10: bytes=32 time=19ms TTL=125
Reply from 172.16.3.10: bytes=32 time=15ms TTL=125

Ping statistics for 172.16.3.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 25ms, Average = 17ms
```

*Fig 17: ping de PC3 vers PC1*

Cette configuration assure dans le réseau actuel le bon routage des paquets. Mais il est possible d'assurer dans tous les réseaux possibles d'émettre les paquets sur la même liaison. Il suffit pour cela d'utiliser une route avec pour adresse IP **0.0.0.0** et en masque **0.0.0.0**. Essayons cette configuration avec le routeur R1.

Nous supprimons au préalable les 3 routes qu'il possède avec les commandes :

```
no ip route 172.16.1.0 255.255.255.0 s0/0/0
```

```
no ip route 192.168.1.0 255.255.255.0 s0/0/0
```

```
no ip route 192.168.2.0 255.255.255.0 s0/0/0
```

Puis nous rajoutons la route par défaut :

```
ip route 0.0.0.0 0.0.0.0 s0/0/0
```

Nous pouvons alors pinguer les autres PC depuis PC1 avec cette nouvelle configuration :

```
PC>ping 172.16.1.10
```

```
Pinging 172.16.1.10 with 32 bytes of data:
```

```
Reply from 172.16.1.10: bytes=32 time=21ms TTL=126
```

```
Reply from 172.16.1.10: bytes=32 time=18ms TTL=126
```

```
Reply from 172.16.1.10: bytes=32 time=10ms TTL=126
```

```
Reply from 172.16.1.10: bytes=32 time=11ms TTL=126
```

```
Ping statistics for 172.16.1.10:
```

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
Minimum = 10ms, Maximum = 21ms, Average = 15ms
```

*Fig 18: ping de PC1 vers PC2*

Le problème pour R1, c'est que sa route par défaut concerne tous les paquets sans exception. Une modification du réseau à son niveau entraînerait une modification de la règle. Cependant, il ne peut utiliser de route par défaut comme pour R3. Effectivement, les 3 sous-réseaux distants de R3 commencent tous par **172.16.x.x**. Déterminer une route relativement précise était possible. Pour R1, il doit atteindre un sous réseau en **172.16.1.x** et des sous-réseaux en **192.168.x.x**. La différence est trop importante ce qui nous oblige à utiliser cette route par défaut générale.

## VII. Câblage et configuration d'un routeur Cisco

Nous pouvons désormais effectuer des manipulations réelles de matériels Cisco. L'objectif est de créer 3 sous réseaux : 2 sous réseaux contenant 1 routeur et un PC et 1 sous réseau pour relier les 2 routeurs.

Nous commençons par effectuer en parallèle la création des deux sous-réseaux distants en reliant un routeur Cisco **C1700** à un PC via la liaison série. Nous lançons sur les PC le programme **minicom** afin de pouvoir configurer les routeurs. Nous commençons par vérifier que minicom est bien configuré en vérifiant que le port série 1 du PC est bien **ttyS0** et que la configuration par défaut est bien **9600N81**.

```
A - Serial Device      : /dev/ttyS0
B - Lockfile Location  : /var/lock
C - Callin Program     :
D - Callout Program    :
E - Bps/Par/Bits       : 9600 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No
```

Fig 19: Configuration de minicom

Nous allumons alors les routeurs, ce que *minicom* détecte en affichant le message de démarrage du routeur :

```
System Bootstrap, Version 12.0(3)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1999 by cisco Systems, Inc.
C1700 platform with 32768 Kbytes of main memory
```

Fig 20: Message de démarrage du routeur

Nous relierons alors le routeur avec le PC via un câble *Ethernet* et nous configurons les adresses IP des routeurs (voir **Fig. 3**) et des PC (**ifconfig eth0 <IP adress>**) :

PC1 : adresse IP : **10.1.0.1/16**

R1 : adresse IP : **10.1.0.2** - masque : **255.255.0.0**

PC2 : adresse IP : **10.2.0.1/16**

R2 : adresse IP : **10.2.0.2** - masque : **255.255.0.0**

Nous pouvons alors pinguer le routeur depuis le PC et inversement :

```
Router#ping 10.1.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4
ms
```

*Fig 21: ping de R1 vers PC1*

Nous relient alors les deux routeurs avec une liaison série puis nous configurons l'interface série de chaque routeur avec les adresses IP (voir **Fig. 7**) et la route (voir partie **VI**) de la manière suivante:

R1 : interface s1 : adresse IP : **10.3.0.1** - masque : **255.255.0.0**  
Route : **10.2.0.0** – masque : **255.255.0.0** - interface : s1  
R2 : interface s1 : adresse IP : **10.3.0.2** - masque : **255.255.0.0**  
Route : **10.1.0.0** – masque : **255.255.0.0** - interface : s1

Nous ajoutons aussi sur les PC une route par défaut vers le routeur à l'aide de la commande **route add default gw 10.1.0.2** pour PC1 et **route add default gw 10.2.0.2** pour PC2.

Le réseau est désormais opérationnel et nous pouvons le vérifier en effectuant un *traceroute* de PC1 vers PC2 :

```
root@pccop0b002b-03:~# traceroute -n 10.2.0.1
traceroute to 10.2.0.1 (10.2.0.1), 30 hops max, 60 byte packets
 1  10.1.0.2  2.428 ms  2.998 ms  3.611 ms
 2  * 10.3.0.2  26.607 ms *
 3  10.2.0.1  47.749 ms  59.907 ms  72.077 ms
```

*Fig 22: traceroute de PC1 vers PC2*

Nous passons effectivement par R1 (**10.1.0.2**) puis par R2 (**10.3.0.2**) pour arriver sur PC2 (**10.2.0.1**).

## VIII. Synthèse

L'objectif de cette partie est de configurer un nouveau réseau en utilisant tout ce que nous avons appris durant ce TP.

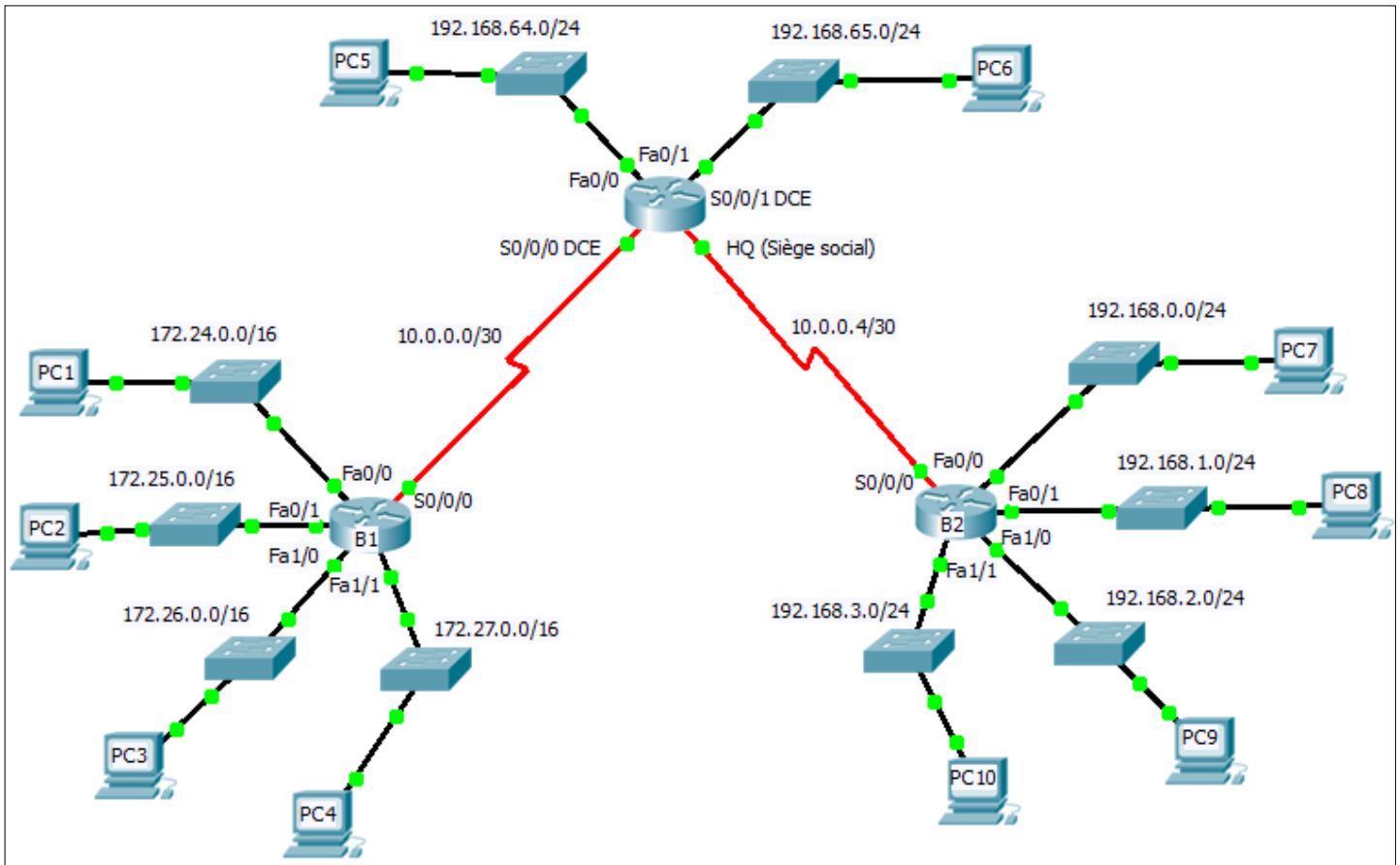


Fig 23: Réseau à créer

Nous commençons avec les machines déjà positionnées que nous devons relier avec les câbles adéquats. Nous utilisons donc des câbles *DCE* pour relier les 3 routeurs et des câbles directs pour relier les autres machines.

Nous obtenons alors la configuration suivante :

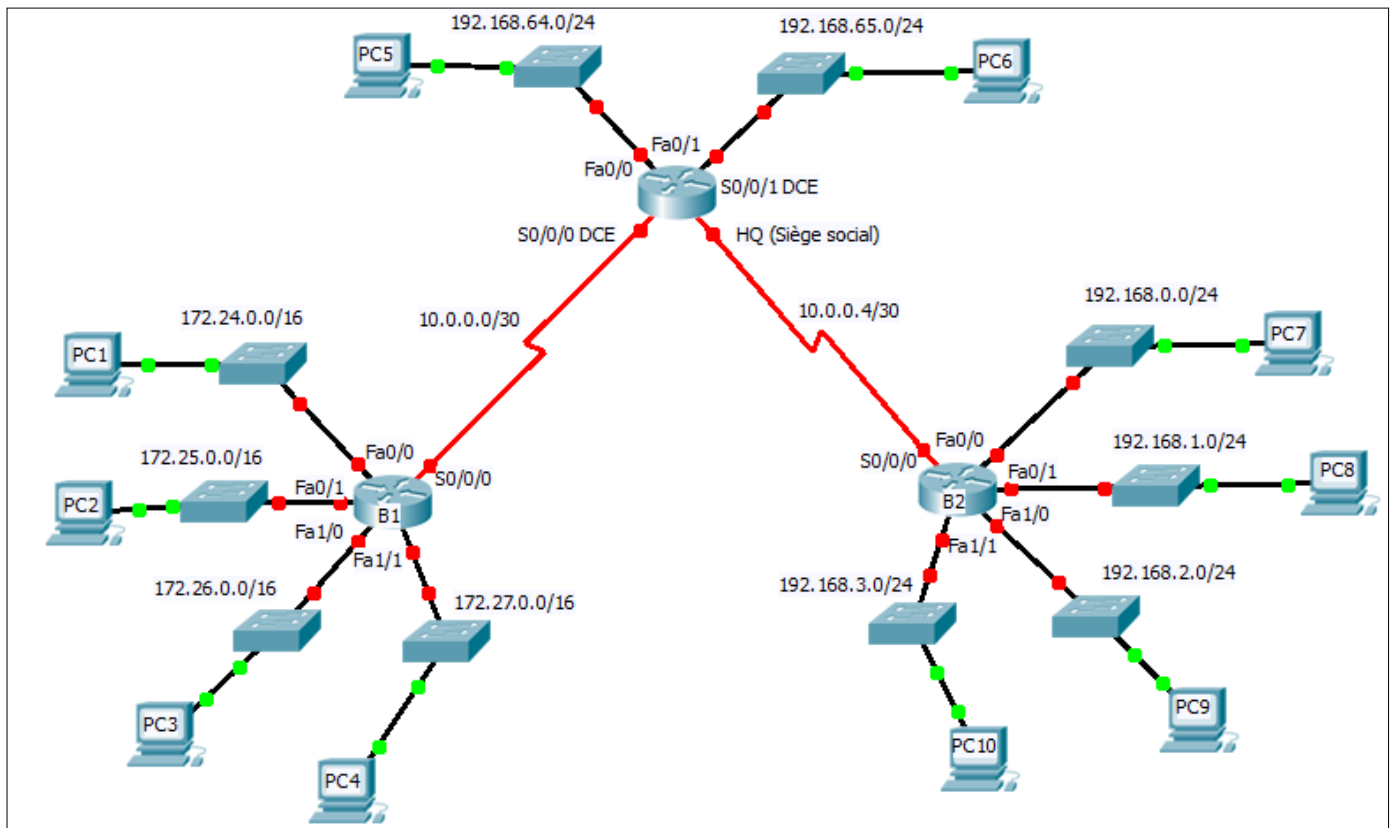


Fig 24: Connexion des machines

Nous attribuons maintenant manuellement à toutes les interfaces une adresse IP. Tous les PC auront pour adresse l'adresse de leur sous-réseau et le numéro de machine 10 (ex : *PC1* -> **172.14.0.10**). Les interfaces *fastEthernet* des routeurs auront pour adresse la première du sous réseau (ex : *B1 fa 0/0* -> **172.24.0.1**).

Nous n'oublierons pas de préciser l'interface de chaque PC dans leur configuration.

Pour les liaisons séries, le siège social prendra la première adresse du sous réseau (**10.0.0.1** pour *s0/0/0* et **10.0.0.5** pour *s0/0/1*) tandis que les autres routeurs auront la seconde adresse (**10.0.0.2** pour B1 et **10.0.0.6** pour B2).

Nous ajoutons aussi différents mots de passe :

- Mot de passe de ligne pour déverrouiller la console. Dans le mode *config*, se connecter sur le mode *line* avec la commande **line console 0**. Taper alors la commande **password cisco** puis **login** pour choisir *cisco* comme mot de passe
- Mot de passe d'enable. Dans le mode *config*, taper la commande **enable secret cisco** pour choisir *cisco* comme mot de passe.

Dès lors, toutes les machines sont connectées et configurées. Il s'agit maintenant de remplir statiquement les tables de routages de chaque routeur :

- B1 et B2 n'ont besoin que d'une route par défaut configurée comme suit :  
Adresse IP : **0.0.0.0** – masque : **0.0.0.0** – interface **s0/0/0**
- HQ a besoin de deux routes statiques configurées comme suit :  
Adresse IP : **172.24.0.0** – masque : **255.252.0.0** – interface **s0/0/0**  
Adresse IP : **192.168.0.0** – masque : **255.255.252.0** – interface **s0/0/1**

Le réseau est maintenant opérationnel. Toutes les machines peuvent se pinguer comme l'illustre la **Fig. 25** qui montre un ping entre la machine 1 et la 10 qui sont les machines les plus éloignées l'une de l'autre du réseau.

```
PC>ping 192.168.3.10

Pinging 192.168.3.10 with 32 bytes of data:

Reply from 192.168.3.10: bytes=32 time=31ms TTL=125
Reply from 192.168.3.10: bytes=32 time=29ms TTL=125
Reply from 192.168.3.10: bytes=32 time=25ms TTL=125
Reply from 192.168.3.10: bytes=32 time=24ms TTL=125

Ping statistics for 192.168.3.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 31ms, Average = 27ms
```

*Fig 25: ping entre PC1 et PC10*

# Conclusion

---

Nous avons commencé par simuler l'implantation et la configuration d'un réseau Cisco tout en étudiant les raisons de nos manipulations. Nous avons continué en pratiquant directement la création d'un réseau réel permettant la communication entre 2 PC distants. Nous avons terminé par finaliser ce TP avec une synthèse de toutes nos manipulations.

Ce TP nous a permis de découvrir et de prendre connaissance du matériel et des commandes Cisco et d'apprendre à utiliser celui-ci dans des cas pratiques.