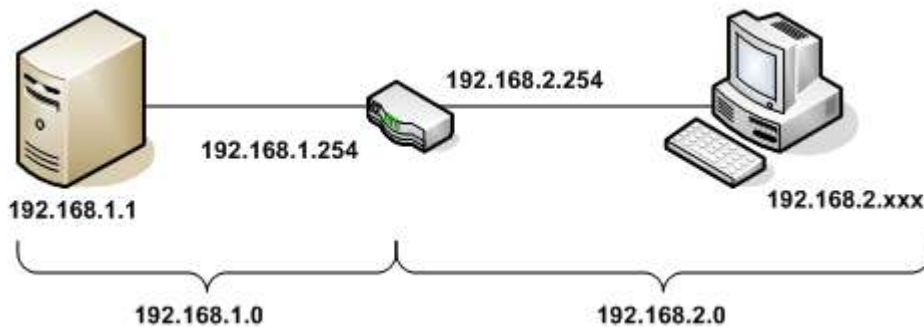


## Compte-rendu PTI #02

Cette deuxième PTI couvre le domaine du paramétrage réseau, de l'adressage dynamique via DHCP (sous GNU/Linux) et du routage matériel sur routeur Cisco.

### Principe

Le schéma suivi est relativement basique, à savoir :



### Légende :

- 192.168.1.1                    Serveur DHCP sous GNU/Linux
- 192.168.2.xxx                Client DHCP sous GNU/Linux (l'adresse IP est attribuée dynamiquement)
- 192.168.{1,2}.254            Passerelle entre les deux réseaux (routeur)

### Compétences

Cette PTI couvre les compétences suivantes :

- C21 - Installer et configurer un microordinateur
- C22 - Installer et configurer un réseau
- C26 - Installer un périphérique
- C31 - Assurer les fonctions de base de l'administration d'un réseau
- C32 - Assurer les fonctions de l'exploitation

### Outils utilisés

Au cours de la réalisation de la PTI, les outils suivants ont été utilisés :

#### Matériel :

- Micro-ordinateurs (un client, un serveur)
- Routeur Cisco 2621 series
- Câbles réseaux à paires torsadées croisées

#### Logiciels :

- Système d'exploitation                    GNU/Linux Slackware version 10
- Serveur DHCP                                ISC DHCP Server v3.0-pl2
- Client DHCP                                  DHCP Client Daemon v1.3.22-pl4
- Emulateur de terminal                      Minicom v2.1

### Configuration du serveur DHCP

Le serveur DHCP utilisé est celui écrit et recommandé par l'ISC (*Internet Software Consortium* - <http://www.isc.org/isc>) et le serveur DHCP par défaut sous Slackware Linux. Voici le fichier de configuration (`/etc/dhcpd.conf`) utilisé :

```
ddns-update-style interim;
ignore client-updates;

default-lease-time 86400;
max-lease-time 60480;

subnet 192.168.1.0 netmask 255.255.255.0
```

```
{
  range 192.168.1.2 192.168.1.10;
  option domain-name-servers 192.168.1.1;
  option broadcast-address 192.168.1.255;
  option routers 192.168.1.254;
}

subnet 192.168.2.0 netmask 255.255.255.0
{
  range 192.168.2.2 192.168.2.10;
  option domain-name-servers 192.168.1.1;
  option broadcast-address 192.168.2.255;
  option routers 192.168.2.254;
}

#host serveur
#{
#   hardware ethernet 00:0D:88:B4:F9:EC;
#   fixed-address 192.168.2.1;
#}
```

Le serveur est ensuite démarré par le *root* (l'administrateur) par la commande `dhcpd` ; ce serveur tourne en *daemon* (tâche de fond) et reste à l'écoute de connexions sur les cartes réseau affectées à chaque réseau.

### Configuration du routeur Cisco 2621

La configuration de ce routeur s'effectue par l'intermédiaire du port série, avec le câble console (RJ-45 <=> DB-9) fourni avec la machine. Pour pouvoir accéder au *shell* (interpréteur de commandes de l'OS du routeur) il faut passer par un émulateur de terminal (comme l'*hyperterminal* sous Windows, par exemple) : nous utiliserons pour notre part *Minicom*, un émulateur très répandu sous GNU/Linux. Avant de le lancer il convient de le configurer un minimum ; voici son fichier de configuration (`/etc/minirc.dfl`) une fois édité :

```
pr port /dev/ttyS0
pu baudrate 9600
pu bits 8
pu parity N
pu stopbits 1
pu rtscts No
```

Une fois le routeur relié au serveur par le câble console, on peut se logger à son OS avec Minicom en tapant la commande `minicom dfl` ; après chargement on arrive au *prompt* (invite de commande) de Cisco IOS, le système d'exploitation du routeur.

Une fois loggé, il faut acquérir les privilèges nécessaires pour configurer le système : il faut pour cela utiliser la commande `enable` (l'équivalent de `su` sous GNU/Linux) ; on remarque que le prompt change une fois la commande tapée.

```
Router> enable
Router#
```

Par définition, un routeur sert à faire transiter les paquets de données d'un réseau à un autre : il faut pour cela configurer ses interfaces (les cartes réseau) qui sont en contact avec ces réseaux. Cette tâche s'effectue après être entré dans la partie configuration grâce à la commande `conf t` :

```
Router# conf t
Router(config)#
```

Tant que nous sommes à ce stade de la configuration, nous pouvons activer le routage IP (chose étonnante qu'elle ne le soit pas par défaut sur un routeur...) et le relais DHCP avec les commandes suivantes :

```
Router(config)# ip routing
Router(config)# service dhcp
```

Puis nous passons au réglage de l'interface 0/0 (routeur <=> serveur DHCP) :

```
Router(config)# int f0/0
```

```
Router(config-int)# ip address 192.168.1.254 255.255.255.0
Router(config-int)# no shutdown
Router(config-int)# exit
```

Et enfin de l'interface 0/1 (routeur <=> réseau de clients) :

```
Router(config)# int f0/1
Router(config-int)# ip address 192.168.2.254 255.255.255.0
Router(config-int)# ip helper-address 192.168.1.1
Router(config-int)# no shutdown
Router(config-int)# exit
```

Les interfaces sont à présent convenablement configurées ; nous retournons à la racine de l'OS pour vérifier que le routeur a bien identifié et ajouté dynamiquement les routes vers les 2 réseaux attenants :

```
Router(config)# exit
Router# show ip route
```

### Tests de bon fonctionnement

Pour nous assurer que notre réseau fonctionne correctement et donc que notre routeur relaye bien les requêtes DHCP du client au serveur, nous exécutons le client DHCP (dhcpcd) sur la machine client :

```
root@b336-cl1:~# dhcpcd
root@b336-cl1:~#
```

Nous n'avons pas eu de message d'erreur à l'exécution de la commande (ce qui est plutôt bon signe) ; vérifions l'interface réseau pour confirmation :

```
root@b336-cl1:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:40:D0:58:18:77
          inet addr:192.168.2.10  Bcast:192.168.2.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2799719 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27411 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:186223763 (177.5 Mb)  TX bytes:4209522 (4.0 Mb)
          Interrupt:3 Base address:0xc000
```

Le client s'est bien vu attribuer une adresse IP par le serveur DHCP : le routeur laisse donc bien passer les broadcasts DHCP.