

ISET de Sfax  
Département Technologie Informatique

# Chapitre 2

## Configuration Réseau

---

**Azer ZAIRI**

**Courriels : Azer.ZAIRI@gnet.tn**

**Azer.ZAIRI@topnet.tn**

# Introduction

---

- La connexion d'un hôte à un réseau IP nécessite des opérations d'installation et de configuration.
- Les étapes de la mise en réseau sont :
  - **installation d'une interface réseau** : cette étape consiste à installer une (ou plusieurs) carte réseau et à paramétrer l'interface réseau correspondante.
    - Elle suit la procédure générale d'installation d'un périphérique.
    - Lors de l'installation du système Linux, les cartes réseaux sont automatiquement détectées et configurées ;
  - **configuration IP** : cette étape traite la configuration de niveau réseau ;
  - **test de la configuration** : cette étape consiste à tester la configuration précédemment effectuée et à utiliser les utilitaires de diagnostic en cas de problèmes réseaux.

# Interfaces réseau

- Les interfaces réseaux sont détectées au démarrage
- Le noyau Linux attribue des **noms d'interface** composés d'un **préfixe** précis en fonction du type d'interface :
  - **Ethernet** : eth0, eth1,... ou enp2s0, enp3s0,....
  - **Wifi** : wlan0, wlan1,... ou wlp3s0, wlp3s1,....
  - **Point à Point** : ppp0, ppp1,...
  - **Anneau à jeton** : tr0, tr1,...
  - **FDDI** : fddi0, fddi1,...
- L'adresse physique (@ MAC) des interfaces réseaux peut être détectée en exécutant la commande **ifconfig**
- Tous les pilotes pour les interfaces de réseau sont construits sous forme de modules
- **/etc/modules.conf** associe des noms logiques à des noms de module spécifiques
- **Exemple de contenu du fichier modules.conf**
  - alias eth0 interface\_reseau\_1
  - alias ppp0 modem1

en : EtherNet,  
p2 : Bus PCI Express numéro 2,  
s0 : index de fonction : Serial0

# Collecte d'information sur les interfaces

---

- La commande **lshw** permet de lister tous les périphériques détectés par Linux
  - Donc elle permet d'identifier les interfaces réseaux.
  - Elle affiche pour chaque interface l'information sur le bus, le détail du pilote et les fonctionnalités supportées.
- **Exemple:**
  - lshw -class network

# Paramétrage des interfaces Ethernet

---

- L'utilitaire `ethtool` affiche et modifie les paramètres des interfaces Ethernet tels que autonegotiation, port speed et duplex mode.

## EXEMPLE

- **`ethtool -s eth0 speed 100 duplex full autoneg`**
  - Cette ligne de commande change le mode de fonctionnement de l'interface « `eth0` » à « full duplex », fixe le débit à 100 Mb/s et active la négociation automatique.
- **`ethtool eth0`**
  - Cette ligne de commande affiche les informations sur le mode de fonctionnement de l'interface `eth0`.

# Paramétrage des interfaces sans fil

---

- ❑ Le paquetage wireless-tools contient les commandes de gestion des interfaces sans fil au niveau liaison de données qui sont iwconfig, iwlist, iwevent, iwgetid, iwpriv et iwspy.
- ❑ Les commandes **iwconfig** et **iwlist** sont les deux commandes les plus pertinentes de ce paquetage.
- ❑ La commande iwconfig est l'outil principal de paramétrage d'une interface sans fil.
- ❑ La commande iwconfig, exécutée sans paramètre, affiche les paramètres des interfaces sans fil ainsi que des statistiques sur la liaison.
- ❑ La syntaxe pour modifier un paramètre est la suivante :
  - iwconfig interface paramètre valeur

# La commande iwconfig (1)

---

- Les paramètres les plus utilisés, de la commande iwconfig, sont :
  - essid : spécifie l'identifiant du réseau ;
  - mode : spécifie le mode de fonctionnement de l'interface qui dépend de la topologie du réseau.
    - La valeur Managed désigne le mode infrastructure,
    - la valeur Ad-Hoc désigne le mode Ad-Hoc, etc. ;
  - ap : force la carte à s'associer à un point d'accès donné ;
  - key/enc : spécifie la clé de cryptage.

# La commande iwconfig (2)

## EXEMPLE

- Cet exemple configure l'interface « wlan0 » afin de joindre le réseau sans fil en mode infrastructure (managed) d'identifiant « isetsf » avec la clé de cryptage WEP « 1234567890 ».
  - # iwconfig wlan0 mode Managed
  - # iwconfig wlan0 essid isetsf
  - # iwconfig wlan0 key 1234567890
- Pour que cette configuration soit permanente, ces paramètres doivent être spécifiés et enregistrés dans les fichiers de configuration correspondants.

**Distribution est de type *Debian* ou dérivées:**  
***fichier de configuration :***  
***/etc/network/interfaces***

```
auto wlan0
wireless-mode managed
wireless-essid isetsf
wireless-key 1234567890
```

**Distribution est de type *Red Hat* ou dérivées :**  
***fichier de configuration :***  
***/etc/sysconfig/networkscripts/***  
***ifcfg-wlan0***

```
TYPE=Wireless
MODE=Managed
ESSID=isetsf
KEY=1234567890
```



# La commande iwlist (1)

---

- ❑ La commande iwlist liste les canaux, les fréquences, les débits et d'autres informations disponibles pour une interface sans fil.
- ❑ La syntaxe de la commande iwlist est :
  - iwlist [interface] type
- ❑ Si l'interface n'est pas spécifiée alors les informations disponibles pour toutes les interfaces sans fil sont listées.
- ❑ Le paramètre « type » correspond au type d'information à lister.
- ❑ Les valeurs possibles sont :
  - scan ou scanning : affiche les paramètres des points d'accès et des cellules Ad-Hoc à la portée de l'interface ;
  - freq, frequency ou channel : liste les fréquences et le nombre des canaux disponibles au niveau de l'interface ;
  - rate, bit ou bitrate : liste les débits supportés par l'interface ;
  - keys, enc ou encryption : affiche les tailles des clés de cryptage supportées et liste toutes les clés installées dans l'interface ;

# La commande iwlist (2)

---

## EXEMPLE

- La commande iwlist suivante détecte un seul point d'accès à la portée de l'interface « wlan0 ».
- Elle liste toutes les informations de ce point d'accès telles que l'adresse MAC, l'identifiant réseau, le mode, etc.

- # iwlist wlan0 scan

- wlan0 Scan completed :

Cell 01 - **Address: 00:90:D0:E0:6B:5A**

Channel:1

Frequency:2.412 GHz (Channel 1)

Quality=43/70 Signal level=-67 dBm

Encryption key:on

**ESSID:« isetsf »**

Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s

24 Mb/s; 36 Mb/s; 54 Mb/s

Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s

**Mode:Master**

# Paramétrage des interfaces point-à-point (1)

---

- Le protocole PPP est utilisé pour établir des liens internet par une connexion série, (Connexion par RTC, connexion DSL, ...)
- Le démon `pppd` fonctionne avec le pilote `ppp` du noyau pour établir et maintenir une liaison `ppp` avec un hôte nommé « *peer* » et pour négocier les adresses IP pour chaque extrémité du lien.
- Il peut aussi gérer la procédure d'authentification avec l'hôte *peer*.
- Les principaux fichiers de configuration du démon `pppd` sont :
  - `/etc/ppp/options` : contient les paramètres généraux d'exécution de `pppd`,
  - `/etc/ppp/pap-secret` : contient les données d'authentification du protocole PAP (risque de sécurité)
  - `/etc/ppp/chap-secret` : contient les données d'authentification du protocole CHAP (plus sécurisé).

# Paramétrage des interfaces point-à-point (2)

---

- Pour simplifier le paramétrage d'une interface ppp, des utilitaires de génération de fichiers de configuration sont utilisés :
  - utilitaires pppconfig et wvdialconf pour les connexions PPP
  - et pppoeconf pour les liaisons PPPoE (*PPP over Ethernet*).
- Les fichiers de configuration de ces utilitaires sont :
  - /etc/ppp/peers/<fournisseur-accès-Internet> : fichier de configuration spécifique à <fournisseur-accès-Internet> créé par pppconfig pour le démon pppd;
  - /etc/chatscripts/<fournisseur-accès-Internet> : fichier de configuration spécifique à <fournisseur-accès-Internet> créé par pppconfig pour le protocole chat ;
  - /etc/ppp/peers/wvdial : fichier de configuration spécifique à wvdial créé par wvdialconf pour le démon pppd ;
  - /etc/wvdial.conf : fichier de configuration créé par wvdialconf ;
  - /etc/ppp/peers/<fournisseur-accès-adsl> : fichier de configuration spécifique à PPPoE créé par pppoeconf pour le démon pppd

## 2. Configuration IP

---

- La configuration réseau consiste à :
  - attribuer, pour chaque interface réseau, une adresse IP et un masque de sous-réseau ;
  - définir la passerelle par défaut ;
  - configurer le service de résolution de noms d'hôtes.

## 2.1. Commandes de configuration IP

---

- ❑ Le **paquetage net-tools** inclut les utilitaires de contrôle du sous-système réseau : arp, ifconfig, route, netstat, etc.
- ❑ Les commandes de configuration réseau ifconfig et route de ce paquetage n'exploitent pas les nouvelles fonctionnalités avancées du noyau Linux.
- ❑ C'est dans ce cadre que le paquetage iproute (nommé aussi iproute2) a été développé.
- ❑ Les commandes ip et tc sont les commandes les plus importantes de ce paquetage.
- ❑ La plupart des distributions Linux installent, par défaut, ces deux paquetages.

## 2.1.1 Attribution d'adresse IP (1)

---

- Les commandes `ifconfig` et `ip` (avec le paramètre `addr`) sont utilisées pour la configuration des interfaces réseaux.

### SYNTAXES IPv4

- Les syntaxes usuelles des commandes `ifconfig` et `ip` pour l'attribution d'adresse IPv4 à une interface réseau sont :
  - **`ifconfig interface adresse-IPv4 [netmask masque] [broadcast adresse-diffusion]`**
  - **`ip addr action adresse_IPv4/masque [broadcast adresse-diffusion] dev interface`**

### SYNTAXES IPv6

- Les syntaxes usuelles pour l'attribution d'adresse IPv6 sont :
  - **`ifconfig interface inet6 action adresse-IPv6/masque`**
  - **`ip -6 addr action adresse_IPv6/masque dev interface`**

## 2.1.1 Attribution d'adresse IP (2)

---

- avec :
  - interface : désigne l'interface réseau à configurer ;
  - adresse-IPv4 : correspond à l'adresse IP version 4 à attribuer ;
  - adresse-IPv6 : correspond à l'adresse IP version 6 à attribuer ;
  - masque : représente le masque de sous réseau ;
  - adresse-diffusion : représente l'adresse de diffusion correspondante ;
  - action : représente l'action à effectuer telle que add ou del.



# Exemple 1

---

Les lignes de commandes suivantes sont toutes équivalentes et elles configurent l'interface « eth0 » avec l'adresse IP «192.168.1.1», le masque «255.255.255.0» (le masque par défaut des adresses IP de classe C) et l'adresse de diffusion «192.168.1.255».

- ❑ **# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255**
- ❑ **# ifconfig eth0 192.168.1.1**
- ❑ **# ip addr add 192.168.1.1/24 broadcast 192.168.1.255 dev eth0**
- ❑ **# ip addr add 192.168.1.1/24 dev eth0**

# Exemple 2

---

- Le deuxième exemple configure l'interface « eth1 » avec l'adresse « 192.168.2.1 » et un masque de sous-réseau égal à « 255.255.255.128 » (ou « /25 » en notation décimale).
- Toutes les commandes suivantes sont équivalentes.
  - **# ifconfig eth1 192.168.2.1 netmask 255.255.255.128 broadcast 192.168.2.127**
  - **# ifconfig eth1 192.168.2.1 netmask 255.255.255.128**
  - **# ip addr add 192.168.2.1/25 broadcast 192.168.2.127 dev eth1**

## 2.1.2. *Passerelle par défaut* (1)

---

- Les commandes `route` et `ip` (avec le paramètre `route`) permettent de gérer la table de routage d'un hôte et en particulier de fixer sa passerelle par défaut.

### SYNTAXE

- Les syntaxes usuelles des commandes `route` et `ip` pour la gestion de la table de routage d'un hôte sont :
- **`route action type destination [netmask masque] gw passerelle dev interface`**
- **`ip route action destination[/masque] via passerelle dev interface`**

## 2.1.2. *Passerelle par défaut* (2)

---

- avec :
  - action : représente l'action à effectuer telle que add ou del pour l'ajout ou la suppression d'une route ;
  - type : correspond à -net si la destination est un réseau et -host si la destination est un hôte ;
  - destination : correspond à l'adresse de la destination.
    - La valeur default correspond à la route par défaut ;
  - masque : correspond au masque de sous réseau de la destination ;
  - interface : correspond à l'interface de sortie des paquets envoyés vers la destination.

### Remarques

- L'exécution de la commande route sans paramètre permet d'afficher la table de routage
- l'option -n affiche les adresses numériques plutôt que des noms de domaines.
- Les routes vers les réseaux adjacents (réseaux auxquels sont connectés les interfaces de l'hôte) sont ajoutées automatiquement.

# Exemple

- Les lignes de commande suivantes sont équivalentes et permettent de fixer à « 172.16.0.254 » l'adresse de la passerelle par défaut, qui est joignable via l'interface « eth0 ».
  - **# route add -net default gw 172.16.0.254 dev eth0**
  - **# ip route add default via 172.16.0.254 dev eth0**
- La séquence suivante affiche la table de routage avec les deux commandes route et ip route.
- **# route -n**
  - Table de routage IP du noyau
  - | Destination | Passerelle   | Genmask     | Indic | Metric | Ref | Use | Iface |
|-------------|--------------|-------------|-------|--------|-----|-----|-------|
| 172.16.0.0  | 0.0.0.0      | 255.255.0.0 | U     | 1      | 0   | 0   | eth0  |
| 0.0.0.0     | 172.16.1.254 | 0.0.0.0     | UG    | 0      | 0   | 0   | eth0  |
- **# ip route**
  - 172.16.0.0/16 dev eth0 proto kernel scope link src 172.16.1.1 metric 1
  - default via 172.16.1.254 dev eth0

## 2.2. Fichiers de configuration réseau

---

- ❑ Les commandes `ifconfig`, `ip` et `route` manipulent les paramètres réseau du noyau.
- ❑ Ceci implique que la configuration réseau, fixée par ces commandes, n'est valable que pour la session système en cours.
- ❑ Pour avoir une configuration réseau permanente, les paramètres réseaux doivent être sauvegardés dans des fichiers de configuration dont les noms, les emplacements et les formats diffèrent selon les distributions.

## 2.2.1. Fichiers de configuration réseau pour les distributions compatibles Debian (1)

---

- Les distributions *Debian* et dérivées utilisent le fichier */etc/network/interfaces* pour la configuration permanente des interfaces réseaux.
- Ce fichier se compose de zéro ou plusieurs entrées iface, mapping, auto et allow- :
  - auto (ou allow-auto) : active une interface lors du démarrage du système ;
  - allow-hotplug : active une interface lorsque le noyau détecte un événement depuis celle ci ;
  - mapping : détermine le nom d'interface logique pour une interface physique ;
  - iface : permet de fixer les paramètres de configuration IP d'une interface.
- Le format de l'entrée iface est :
  - **iface interface famille méthode**
  - paramètre1 valeur1
  - paramètre2 valeur2
  - ...

## 2.2.1. *Fichiers de configuration réseau pour les distributions compatibles Debian* (2)

---

- avec :
  - interface : spécifie le nom de l'interface à configurer ;
  - famille : spécifie la famille d'adresses utilisée.
    - Les valeurs fréquentes sont inet pour la famille d'adresses IPv4 et inet6 pour la famille d'adresses IPv6 ;
  - méthode : spécifie la méthode utilisée pour l'attribution d'adresses.
    - Les méthodes les plus utilisées sont :
      - lo : c'est la méthode d'attribution d'adresses de boucle locale. Elle n'admet pas de paramètres ;
      - dhcp : c'est la méthode d'attribution d'adresses par le service DHCP. Elle admet des paramètres tels que hwaddress, leasetime, etc. En général, la méthode dhcp est utilisée sans paramètres ;



## *2.2.1. Fichiers de configuration réseau pour les distributions compatibles Debian*

*(3)*

---

- static : c'est la méthode pour attribuer des adresses manuellement. Les paramètres les plus fréquents sont :
  - address : spécifie l'adresse IP à affecter ;
  - netmask : spécifie le masque de sous réseau ;
  - broadcast : spécifie l'adresse de diffusion ;
  - gateway : spécifie l'adresse de la passerelle par défaut.

# Exemple

---

- Le fichier `/etc/network/interfaces` suivant illustre la configuration réseau d'un hôte ayant deux interfaces Ethernet : « `eth0` » configurée statiquement et « `eth1` » configurée dynamiquement par le service DHCP.
- Il illustre aussi les lignes correspondantes à l'interface de boucle locale (*loopback*) « `lo` ».

```
# cat /etc/network/interfaces
```

```
### The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

## Remarque

- À chaque modification du fichier de configuration des interfaces, il faut relancer le service réseau, avec une des méthodes suivantes :
  - **# service networking restart**
  - **# systemctl restart networking.service**
  - **# /etc/init.d/networking restart**
  - **# invoke-rc.d networking restart**

## 2.2.2. *Cas des distributions Red Hat et dérivées*

---

- ❑ Pour une configuration réseau permanente, Les distributions *Red Hat et dérivées* *utilisent* un fichier de configuration pour chaque interface.
- ❑ Ce fichier a pour nom ifcfg-interface et se situe sous le répertoire /etc/sysconfig/network-scripts/.
- ❑ Par exemple, le fichier /etc/sysconfig/network-scripts/ifcfg-eth0 est le fichier de configuration de l'interface « eth0 ».
- ❑ Chaque ligne des fichiers de configuration est sous la forme : « paramètre=valeur ».
- ❑ Il existe des paramètres communs à tout type d'interface (Ethernet, PPP, etc.) et d'autres spécifiques à un type donné.



---

Les paramètres fréquents dans le cas de la configuration d'une interface Ethernet sont :

- ❑ DEVICE : correspond au nom de l'interface réseau ;
- ❑ BOOTPROTO : correspond au protocole utilisé pour la configuration de l'interface. Les valeurs possibles sont : none, bootp ou dhcp
- ❑ ONBOOT : correspond à l'état de l'interface indiquant si celle-ci devrait être activée ou non lors du démarrage du système. Les valeurs possibles sont yes ou no ;
- ❑ IPADDR : correspond à l'adresse IP de l'interface ;
- ❑ NETMASK : correspond au masque du sous réseau ;
- ❑ BROADCAST : correspond à l'adresse de diffusion. Cette directive a été abandonnée car la valeur est calculée automatiquement avec ifcalc selon la valeur du NETMASK ;
- ❑ GATEWAY : correspond à l'adresse IP de la passerelle réseau ;
- ❑ USERCTL : correspond à la possibilité de contrôler l'interface par les utilisateurs autres que l'utilisateur root. Les valeurs possibles sont yes et no.

# Exemple 1

---

- Le premier exemple illustre une configuration statique de l'interface « eth0 ».

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE=eth0
```

```
BOOTPROTO=none
```

```
ONBOOT=yes
```

```
IPADDR=192.16.1.1
```

```
NETMASK=255.255.255.0
```

```
BROADCAST=192.168.1.255
```

```
USERCTL=no
```

# Exemple 2

---

- Le deuxième exemple illustre une configuration dynamique de l'interface « eth1 ».

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
DEVICE=eth1
```

```
BOOTPROTO=dhcp
```

```
ONBOOT=yes
```

# 3. Configuration multiréseau (1)

---

- ❑ Un hôte multiréseau (*multihomed host*) est un hôte équipé de plusieurs interfaces réseaux connectées chacune à un réseau différent.
- ❑ Cette configuration est utilisée dans le cas où un hôte Linux joue le rôle d'un routeur, d'une passerelle réseau ou d'un pare-feu.
- ❑ Par défaut, la fonction de routage du noyau Linux est désactivée.
- ❑ C'est à dire que les paquets IP reçus dont l'adresse IP de destination correspond à un autre hôte sont rejetés.
- ❑ Dans le cas d'un hôte multiréseau, la fonction de routage doit être activée.
- ❑ Ceci est assuré par l'affectation de la valeur 1 au contenu du fichier `/proc/sys/net/ipv4/ip_forward`
- ❑ On peut effectuer cette opération comme suit :
  - # echo "1" > **/proc/sys/net/ipv4/ip\_forward**
  - ou
  - # sysctl -w net.ipv4.ip\_forward=1

# 3. Configuration multiréseau (2)

---

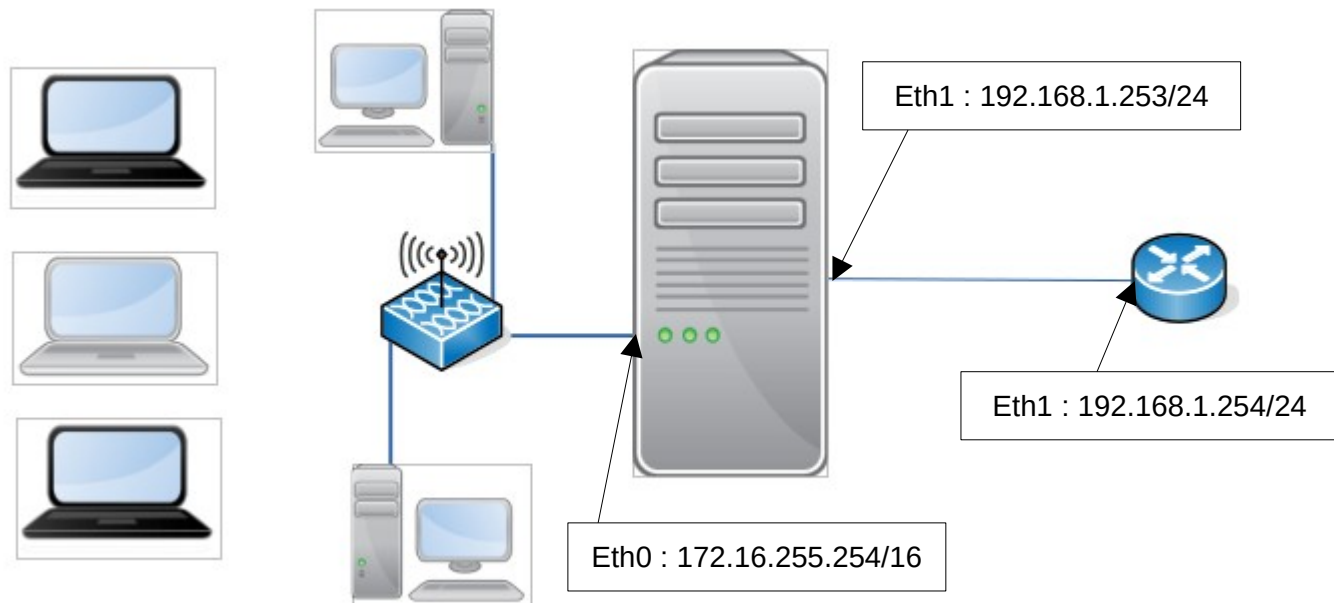
- ❑ Le fichier `/proc/sys/net/ipv4/ip_forward` correspond au paramètre « fonction de routage » de la session en cours du système.
- ❑ Les deux valeurs possibles sont :
  - 0 : fonction de routage désactivée et
  - 1 : fonction de routage activée.
- ❑ Pour rendre l'activation de la fonction de routage permanente (même en cas de redémarrage du système), le paramètre `net.ipv4.ip_forward` du fichier de configuration `/etc/sysctl.conf` doit être fixé manuellement à 1.
- ❑ La vérification de l'activation permanente de la fonction de routage peut être effectuée comme suit :
  - `# grep ip_forward /etc/sysctl.conf`
  - **`net.ipv4.ip_forward=1`**



# Exemple

(1)

- L'exemple suivant configure un hôte multiréseau connecté à un premier réseau local contenant les hôtes de type poste de travail et à un deuxième réseau contenant le routeur assurant la connexion vers Internet.
- Il est connecté au premier réseau d'adresse IP « 172.16.0.0/16 » à travers « eth0 » d'adresse « 172.16.255.254 » et au deuxième réseau d'adresse IP « 192.168.1.0/24 » à travers « eth1 » d'adresse « 192.168.1.253 ».
- Le routeur assurant la connexion vers Internet et jouant le rôle de passerelle par défaut de l'hôte multiréseau est d'adresse « 192.168.1.254 ».



# Exemple

(2)

```
# ifconfig eth0 172.16.255.254 netmask 255.255.0.0
# ifconfig eth1 192.168.1.253 netmask 255.255.255.0
# route add -net default gw 192.168.1.254 dev eth1
# route -n
```

Table de routage IP du noyau

Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
172.16.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth0
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
0.0.0.0	192.168.1.254	0.0.0.0	UG	0	0	0	eth1

```
# sysctl -w net.ipv4.ip_forward=1
```

# 4. Diagnostic réseau

(1)

- 
- On est souvent confronté à des problèmes réseaux tels que le ralentissement du trafic, la déconnexion d'un hôte ou d'un réseau, la non disponibilité d'un service réseau, etc.
  - Pour identifier ces problèmes, les opérations de diagnostic suivantes peuvent être effectuées :
    - l'utilisation des commandes de paramétrage des interfaces et de configuration réseau précédemment traitées ;
    - la vérification des fichiers de configuration ;
    - le suivi et l'analyse des journaux (les fichiers log) ;
    - l'utilisation d'outils dédiés aux tests et aux diagnostics réseau dont les plus utilisés seront traités dans cette section.

## 4. Diagnostic réseau

(2)

- Les principales commandes et outils de diagnostic réseau sont :
  - **ping** : pour le test de connectivité réseau
  - **nslookup** et **dig** : pour le test de la résolution du nom de hôte
  - **traceroute** : pour le test du chemin réseau
  - **telnet** et **nc** : pour le test de la connectivité applicative
  - **netstat** : pour le diagnostic de l'état du réseau
  - **lsof** : pour le diagnostic de l'état des fichiers ouverts par les daemons réseaux
  - **nmap** : pour le scan des ports
  - **wireshark** et **tcpdump** : pour l'analyse du trafic réseau