Niveau : Admin Junior	Numéro de la Leçon	Dernière Modification
2/4	<progrecss 12/12 style=inline />	2020/01/30 03:28

Gestion du Réseau TCPv4

Modèles de Communication

Le modèle OSI

Le modèle OSI (Open System Interconnexion) qui a été proposé par l'ISO est devenu le standard en termes de modèle pour décrire l'échange de données entre ordinateurs. Cette norme se repose sur sept couches, de la une - la Couche Physique, à la sept - la Couche d'Application, appelés des services. La communication entre les différentes couches est synchronisée entre le poste émetteur et le poste récepteur grâce à ce que l'on appelle un protocole.

Dans ce modèle :

- **La Couche Physique** (Couche 1) est responsable :
  - du transfert de données binaires sur le câble physique ou virtuel
  - de la définition de tout aspect physique allant du connecteur jusqu'au câble en passant par la carte réseau, y compris l'organisation même du réseau
  - de la définition des tensions électriques sur le câble pour obtenir le 0 et le 1 binaires
- **La Couche de Liaison** (Couche 2) est responsable :
  - de la réception des données de la couche physique
  - de l'organisation des données en fragments, appelés des trames qui ont un format différent selon s'il s'agit d'un réseau basé sur la technologie Ethernet ou la technologie Token-Ring
  - de la préparation, émission et réception des trames
  - de la gestion de l'accès au réseau
  - de la communication nœud à nœud
  - de la gestion des erreurs
    - avant la transmission, le nœud émetteur calcule un code appelé un CRC et l'incorpore dans les données envoyées

        - le nœud récepteur recalcule un CRC en fonction du contenu de la trame reçue et le compare à celui incorporé avec l'envoi
        - en cas de deux CRC identique, le nœud récepteur envoie un accusé de réception au nœud émetteur
    - de la réception de l'accusé de réception
    - éventuellement de le ré-émission des données
    - En prenant ce modèle, l'IEEE (Institute of Electrical and Eletronics Engineers) l'a étendu avec le Modèle IEEE (802).
        - Dans ce modèle la Couche de Liaison est divisée en deux sous-couches importantes :
            - La **Sous-Couche LLC** (Logical Link Control) qui :
                - gère les accusés de réception
                - gère le flux de trames
            - La **Sous-Couche MAC** (Media Access Control) qui :
                - gère la méthode d'accès au réseau
                - le CSMA/CD dans un réseau basé sur la technologie Ethernet
                - l'accès au jeton dans un réseau basé sur la technologie Token-Ring
                - gère les erreurs

- **La Couche de Réseau** (Couche 3) est responsable de la gestion de la bonne distribution des différentes informations aux bonnes adresses en :
    - identifiant le chemin à emprunter d'un nœud donné à un autre
    - appliquant une conversion des adresses logiques (des noms) en adresses physiques
    - ajoutant des information adressage aux envois
    - détectant des paquets trop volumineux avant l'envoi et en les divisant en trames de données de tailles autorisées

- **La Couche de Transport** (Couche 4) est responsable de veiller à ce que les données soient envoyées correctement en :
    - constituant des paquets de données corrects
    - les envoyant dans le bon ordre
    - vérifiant que les données sont traités dans le même ordre que l'ordre d'émission
    - permettant à un processus sur un nœud de communiquer avec un autre nœud et d'échanger des messages avec lui

- **La Couche de Session** (Couche 5) est responsable :
    - de l'établissement, du maintien, et de la mise à fin de la communication entre deux noeuds distants, c'est-à-dire, de la session
    - de la conversation entre deux processus de vérification de la réception des messages envoyés en séquences, c'est-à-dire, le point de contrôle

- de la sécurité lors de l'ouverture de la session, c'est-à-dire, les droits d'utilisateurs etc.

- **La Couche de Présentation** (Couche 6) est responsable :
    - du formatage et de la mise en forme des données

    - des conversions de données telles le cryptage/décryptage

- **La Couche d'Application** (Couche 7) est responsable :
    - du dialogue homme/machine via des messages affichés
    - du partage des ressources
    - de la messagerie

Spécification NDIS et le Modèle ODI

<note tip> **Cliquez ici pour ouvrir le schéma Simplifié du Modèle OSI incluant la spécification NDIS** </note>

La spécification NDIS (Network Driver Interface Specification) a été introduite conjointement par les sociétés Microsoft et 3Com. Cette spécification ainsi que son homologue, le modèle ODI (Open Datalink Interface) introduit conjointement par les sociétés Novell et Apple à la même époque, définit des standards pour les pilotes de cartes réseau afin qu'ils puissent être indépendants des protocoles utilisées et les systèmes d'exploitation sur les machines. Des deux 'standards', la spécification NDIS est le plus répandu, intervenant a niveau de la sous-couche MAC et l a couche de liaison. Elle spécifie :

- l'interface pilote-matériel
- l'interface pilote-protocole
- l'interface pilote - système d'exploitation

Le modèle TCP/IP

<note tip> **Cliquez ici pour voir le modèle OSI incluant la suite des protocoles et services TCP/IP** </note>

La suite des protocoles TCP/IP (Transmission Control Protocol / Internet Protocol) est issu de la DOD (Dept. Américain de la Défense) et le travail de l'ARPA (Advanced Research Project Agency).

- La suite des protocoles TCP/IP
    - a été introduite en 1974
    - a été utilisée dans l'ARPAnet en 1975
    - permet la communication entre des réseaux à base de systèmes d'exploitation, architectures et technologies différents
    - est très proche du modèle OSI en termes d'architecture et se place au niveau de la couche d'Application jusqu'à la couche Réseau.

- est, en réalité, une suite de protocoles et de services :
  - **IP** (Internet Protocol)
    - le protocole IP s'intègre dans la couche Réseau du modèle OSI en assurant la communication entre les systèmes. Bien qu'il puisse découper des messages en fragments ou datagrammes et les reconstituer dans le bon ordre à l'arrivée, il ne garantit pas la réception.
  - **ICMP** (Internet Control Message Protocol)
    - le protocole ICMP produit des messages de contrôle aidant à synchroniser le réseau. Un exemple de ceci est la commande ping.
  - **TCP** (Transmission Control Protocol)
    - le protocole TCP se trouve au niveau de la couche de Transport du modèle OSI et s'occupe de la transmission des données entre noeuds.
  - **UDP** (User Datagram Protocol)
    - le protocole UDP n'est pas orienté connexion. Il est utilisé pour la transmission rapide de messages entre nœuds sans garantir leur acheminement.
  - **Telnet**
    - le protocole Telnet est utilisé pour établir une connexion de terminal à distance. Il se trouve dans la couche d'Application du modèle OSI.
  - **Ftp** (File Transfer Protocol)
    - le protocole ftp est utilisé pour le transfert de fichiers. Il se trouve dans la couche d'Application du modèle OSI.
  - **SMTP** (Simple Message Transfer Protocol)
    - le service SMTP est utilisé pour le transfert de courrier électronique. Il se trouve dans la couche d'Application du modèle OSI.
  - **DNS** (Domain Name Service)
    - le service DNS est utilisé pour le résolution de noms en adresses IP. Il se trouve dans la couche d'Application du modèle OSI.
  - **SNMP** (Simple Network Management Protocol)
    - le protocole SNMP est composé d'un agent et un gestionnaire. L'agent SNMP collecte des informations sur les périphériques, les configurations et les performances tandis que le gestionnaire SNMP reçois ses informations et réagit en conséquence.
  - **NFS** (Network File System)
    - le NFS a été mis au point par Sun Microsystems
    - le NFS génère un lien virtuel entre les lecteurs et les disques durs permettant de monter dans un disque virtuel local un disque distant
  - et aussi POP3, NNTP, IMAP etc ...

<note tip> **Cliquez ici pour voir les modèles TCP/IP et OSI** </note>

Le modèle TCP/IP est composé de 4 couches :

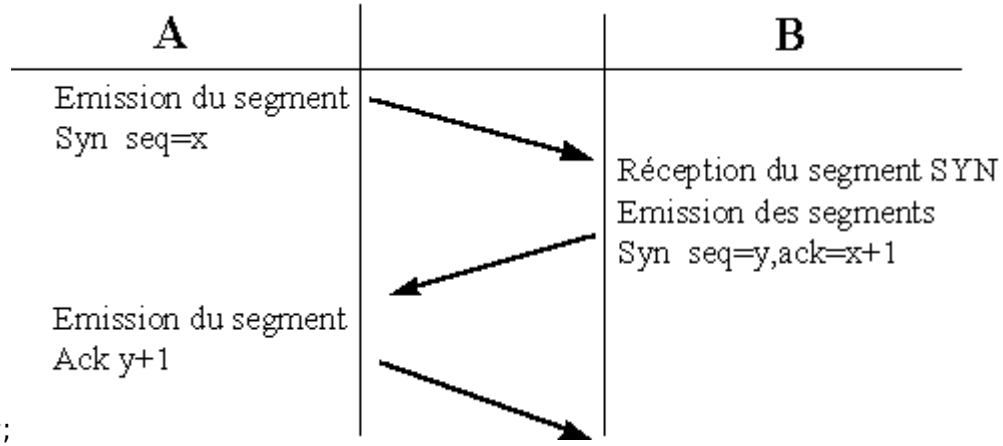
- La couche d'Accès Réseau
  - Cette couche spécifie la forme sous laquelle les données doivent être acheminées, quelque soit le type de réseau utilisé.
- La couche Internet
  - Cette couche est chargée de fournir le paquet de données.
- La couche de Transport
  - Cette couche assure l'acheminement des données et se charge des mécanismes permettant de connaître l'état de la transmission.
- La couche d'Application
  - Cette couche englobe les applications standards de réseau telles ftp, telnet, ssh, etc..

Message/Datagramme/Segment

Les noms des unités de données sont différents selon le protocole utilisé et la couche du modèle TCP/IP :

Couche	**TCP**	**UDP**
Application	Stream	Message
Transport	Segment	Packet
Internet	Datagram	Datagram
Réseau	Frame	Frame

Etablissement de la connexion TCP



;#; ;#;

L'établissement de la connexion TCP entre deux stations A et B se fait en trois temps.

1. A émet une demande de connexion avec un message TCP dont le bit SYN est positionné, et dans lequel est fourni son numéro de séquence initial (x).
2. B retourne un message avec les bits SYN et ACK, en acquittant le numéro de séquence de A (x+1) et en fournissant son numéro de séquence initial(y).
3. A retourne un acquittement du numéro de séquence de B (y+1).

En-tête TCP

L'en-tête TCP est codée sur 4 octets soit 32 bits :

1er octet	**2ème octet**	**3ème octet**	**4 ème octet**
Port source		Port destination	
Numéro de séquence			
Numéro d'acquittement			
Offset	Flags	Fenêtre	
Checksum		Pointeur Urgent	
Options			Padding
Données			

Vous noterez que les numéros de ports sont codés sur 16 bits. Cette information nous permet de calculer le nombres de ports maximum en IPv4, soit 2^{16} ports ou 65 535.

L'**Offset** contient la taille de l'en-tête.

Les **Flags** sont :

- URG - Si la valeur est 1 le pointeur urgent est utilisé. Le numéro de séquence et le pointeur urgent indique un octet spécifique.
- ACK - Si la valeur est 1, le paquet est un accusé de réception
- PSH - Si la valeur est 1, les données sont immédiatement présentées à l'application
- RST - Si la valeur est 1, la communication comporte un problème et la connexion est réinitialisée
- SYN - Si la valeur est 1, le paquet est un paquet de synchronisation
- FIN - Si la valeur est 1, le paquet indique la fin de la connexion

La **Fenêtre** est codée sur 16 bits. La Fenêtre est une donnée liée au fonctionnement d'expédition de données appelé le **sliding window** ou la **fenêtre glissante**. Puisque il serait impossible, pour des raisons de performance, d'attendre l'accusé de réception de chaque paquet envoyé, l'expéditeur envoie des paquets par groupe. La taille de cette groupe s'appelle la Fenêtre. Dans le cas d'un problème de réception d'une partie de la Fenêtre, toute la Fenêtre est ré-expédiée.

Le **Checksum** est une façon de calculer si le paquet est complet.

Le **Padding** est un champ pouvant être rempli de valeurs nulles de façon à ce que la taille de l'en-tête soit un multiple de 32

En-tête UDP

L'en-tête UDP est codée sur 4 octets soit 32 bits :

1er octet	**2ème octet**	**3ème octet**	**4 ème octet**
Port source		Port destination	
longueur		Checksum	
Données			

L'en-tête UDP a une longueur de 8 octets.

Fragmentation et Ré-encapsulation

La taille limite d'un paquet TCP, l'en-tête comprise, ne peut pas dépasser **65 535 octets**. Cependant chaque réseau est qualifié par son MTU (Maximum Tranfer Unit). Cette valeur est la taille maximum d'un paquet autorisée. L'unité est en **octets**. Pour un réseau Ethernet sa valeur est de 1 500. Quand un paquet doit être expédié sur un réseau ayant un MTU inférieur à sa propre taille, le paquet doit être **fractionné**. A la sortie du réseau, le paquet est reconstitué. Cette reconstitution s'appelle **ré-encapsulation**.

Adressage

L'adressage IP requière que chaque périphérique sur le réseau possède une adresse IP unique de 4 octets, soit 32 bits au format XXX.XXX.XXX.XXX De cette façon le nombre total d'adresses est de 2^{32} = 4.3 Milliards.

Les adresses IP sont divisées en 5 classes, de A à E. Les 4 octets des classes A à C sont divisés en deux, une partie qui s'appelle le **Net ID** qui identifie le réseau et une partie qui s'appelle le **Host ID** qui identifie le hôte :

	1er octet	**2ème octet**	**3ème octet**	**4 ème octet**
A	Net ID	Host ID		
B	Net ID		Host ID	
C	Net ID			Host ID
D	Multicast			
E	Réservé			

L'attribution d'une classe dépend du nombre de hôtes à connecter. Chaque classe est identifié par un **Class ID** composé de 1 à 3 bits :

Classe	**Bits ID Classe**	**Valeur ID Classe**	**Bits ID Réseau**	**Nb. de Réseaux**	**Bits ID hôtes**	**Nb. d'adresses**	**Octet de Départ**
A	1	0	7	2^7=128	24	2^{24}=16 777 216	1 - 126
B	2	10	14	2^{14}=16 834	16	2^{16}=65 535	128 - 191
C	3	110	21	2^{21}=2 097 152	8	2^8=256	192 - 223

Dans chaque classe, certaines adresses sont réservées pour un usage privé :

Classe	IP de Départ	IP de Fin
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

Il existe des adresses particulières ne pouvant pas être utilisées pour identifier un hôte :

Adresse Particulière	Description
169.254.0.0 à 169.254.255.255	Automatic Private IP Addressing de Microsoft
Hôte du réseau courant	Tous les bits du Net ID sont à 0
Adresse de réseau	Tous les bits du Host ID sont à 0
Adresse de diffusion	Tous les bits du Host ID sont à 1

L'adresse de réseau identifie le **segment** du réseau entier tandis que l'adresse de diffusion identifie tous les hôtes sur le segment de réseau.

Afin de mieux comprendre l'adresse de réseau et l'adresse de diffusion, prenons le cas de l'adresse 192.168.10.1 en classe C :

	1er octet	2ème octet	3ème octet	4 ème octet
	Net ID			Host ID
Adresse IP	192	168	10	1
Binaire	11000000	10101000	000001010	00000001
Calcul de l'adresse de réseau				
Binaire	11000000	10101000	000001010	**00000000**
Adresse réseau	192	168	10	0
Calcul de l'adresse de diffusion				
Binaire	11000000	10101000	000001010	**11111111**
Adresse de diffusion	192	168	10	255

Masques de sous-réseaux

Tout comme l'adresse IP, le masque de sous-réseau compte 4 octets ou 32 bits. Les masques de sous-réseaux permettent d'identifer le Net ID et le Host ID :

Classe	Masque	Notation CIDR
A	255.0.0.0	/8
B	255.255.0.0	/16
C	255.255.255.0	/24

Le terme **CIDR** veut dire **Classless InterDomain Routing**. Le terme Notation CIDR correspond au nombre de bits d'une valeur de 1 dans le masque de sous-réseau.

Quand un hôte souhaite émettre il procède d'abord à l'identification de sa propre adresse réseau par un calcul AND (ET) appliqué à sa propre adresse et son masque de sous-réseau qui stipule :

- 1 + 1 = 1
- 0 + 1 = 0
- 1 + 0 = 0
- 0 + 0 = 0

Prenons le cas de l'adresse IP 192.168.10.1 ayant un masque de 255.255.255.0 :

	1er octet	2ème octet	3ème octet	4 ème octet
Adresse IP	192	168	10	1
Binaire	11000000	10101000	00001010	00000001
Masque de sous-réseau				
Binaire	11111111	11111111	11111111	00000000
Calcul AND	11000000	10101000	00001010	00000000
Adresse réseau	192	168	10	0

Cet hôte essaie de communiquer avec un hôte ayant une adresse IP de 192.168.10.10. Il procède donc au même calcul en appliquant **son propre masque de sous-réseau** à l'adresse IP de l'hôte destinataire :

	1er octet	2ème octet	3ème octet	4 ème octet
Adresse IP	192	168	10	10
Binaire	11000000	10101000	00001010	00001010
Masque de sous-réseau				

	1er octet	2ème octet	3ème octet	4 ème octet
Binaire	11111111	11111111	11111111	00000000
Calcul AND	11000000	10101000	00001010	00000000
Adresse réseau	192	168	10	0

Puisque l'adresse réseau est identique dans les deux cas, l'hôte émetteur présume que l'hôte de destination se trouve sur son réseau et envoie les paquets directement sur le réseau sans s'adresser à sa passerelle par défaut.

L'hôte émetteur essaie maintenant de communiquer avec avec un hôte ayant une adresse IP de 192.168.2.1. Il procède donc au même calcul en appliquant **son propre masque de sous-réseau** à l'adresse IP de l'hôte destinataire :

	1er octet	2ème octet	3ème octet	4 ème octet
Adresse IP	192	168	2	1
Binaire	11000000	10101000	00000010	00000001
Masque de sous-réseau				
Binaire	11111111	11111111	11111111	00000000
Calcul AND	11000000	10101000	00000010	00000000
Adresse réseau	192	168	2	0

Dans ce cas, l'hôte émetteur constate que le réseau de destination 192.168.2.0 n'est pas identique à son propre réseau 192.168.10.0. Il adresse donc les paquets à la passerelle par défaut.

VLSM

Puisque le stock de réseaux disponibles sous IPv4 est presque épuisé, une solution a du être trouvée pour créer des sous-réseaux en attendant l'introduction de l'IPv6. Cette solution s'appelle le VLSM ou Variable Length Subnet Masks. Le VLSM exprime les masques de sous-réseaux au format CIDR.

Son principe est simple. Afin de créer des réseaux différents à partir d'une adresse réseau d'une classe donnée, il convient de réduire le nombre d'hôtes. De cette façon les bits 'libérés' du Host ID peuvent être utilisés pour identifier les sous-réseaux.

Pour illustrer ceci, prenons l'exemple d'un réseau 192.168.1.0. Sur ce réseau, nous pouvons mettre 2^8-2 soit 254 hôtes entre 192.168.1.1 au

192.168.1.254.

Supposons que nous souhaiterions diviser notre réseau en 2 sous-réseaux. Pour coder 2 sous-réseaux, il faut que l'on libère 2 bits du Host ID. Les deux bits libérés auront les valeurs binaires suivantes :

- 00
- 01
- 10
- 11

Les valeurs binaires du quatrième octet de nos adresses de sous-réseaux seront donc :

- 192.168.1.00XXXXXX
- 192.168.1.01XXXXXX
- 192.168.1.10XXXXXX
- 192.168.1.11XXXXXX

où les XXXXXX représentent les bits que nous réservons pour décrire les hôtes dans chacun des sous-réseaux.

Nous ne pouvons pas utiliser les deux sous-réseaux suivants :

- 192.168.1.00XXXXXX
- 192.168.1.11XXXXXX

car ceux-ci correspondent aux débuts de l'adresse réseau 192.168.1.0 et de l'adresse de diffusion 192.168.1.255.

Nous pouvons utiliser les deux sous-réseaux suivants :

- 192.168.1.01XXXXXX
- 192.168.1.10XXXXXX

Pour le premier sous-réseau l'adresse réseau et l'adresse de diffusion sont :

Sous-réseau #1	192	168	1	01XXXXXX
Calcul de l'adresse de réseau				
Binaire	11000000	10101000	00000001	01**000000**

Adresse réseau	192	168	1	**64**
Calcul de l'adresse de diffusion				
Binaire	11000000	10101000	00000001	01**111111**
Adresse de diffusion	192	168	1	**127**

- L'adresse CIDR du réseau est donc 192.168.1.64/26 car le Net ID est codé sur 24+2 bits.
- Le masque de sous-réseau est donc le 11111111.11111111.11111111.11000000 ou le 255.255.255.192
- Nous pouvons avoir 2^6-2 soit 62 hôtes.
- La plage valide d'adresses IP est de 192.168.1.65 à 192.168.1.126

Pour le deuxième sous-réseau l'adresse réseau et l'adresse de diffusion sont :

Sous-réseau #2	192	168	1	10XXXXXX
Calcul de l'adresse de réseau				
Binaire	11000000	10101000	00000001	10**000000**
Adresse réseau	192	168	1	**128**
Calcul de l'adresse de diffusion				
Binaire	11000000	10101000	00000001	10**111111**
Adresse de diffusion	192	168	1	**191**

- L'adresse CIDR du réseau est donc 192.168.1.128/26 car le Net ID est codé sur 24+2 bits.
- Le masque de sous-réseau est donc le 11111111.11111111.11111111.11000000 ou le 255.255.255.192
- Nous pouvons avoir 2^6-2 soit 62 hôtes.
- La plage valide d'adresses IP est de 192.168.1.129 à 192.168.1.190

La valeur qui sépare les sous-réseaux est 64. Cette valeur comporte le nom **incrément**.

Ports et sockets

Afin que les données arrivent aux applications que les attendent, TCP utilise des numéros de ports sur la couche transport. Le numéros de ports sont divisés en trois groupes :

- **Well Known Ports**
  - De 1 à 1023
- **Registered Ports**
  - De 1024 à 49151
- **Dynamic** et/ou **Private Ports**
  - De 49152 à 65535

Le couple **numéro IP:numéro de port** s'appelle un **socket**.

Configuration du Réseau

/etc/services

Les ports les plus utilisés sont détaillés dans le fichier **/etc/services** :

```
opensuse:~ # more /etc/services
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
#
# This list could be found on:
#            http://www.iana.org/assignments/port-numbers
#
# See also: services(5), http://www.sethwklein.net/projects/iana-etc/
#
# PORT NUMBERS
```

```
#
# (last updated 2008-01-25)
#
# The port numbers are divided into three ranges: the Well Known Ports,
# the Registered Ports, and the Dynamic and/or Private Ports.
#
# The Well Known Ports are those from 0 through 1023.
#
# DCCP Well Known ports SHOULD NOT be used without IANA registration.
# The registration procedure is defined in [RFC4340], Section 19.9.
#
# The Registered Ports are those from 1024 through 49151
#
# DCCP Registered ports SHOULD NOT be used without IANA registration.
# The registration procedure is defined in [RFC4340], Section 19.9.
#
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# A value of 0 in the port numbers registry below indicates that no port
# has been allocated.
#
--More--(0%)
```

Notez que les ports sont listés par deux :

- le port TCP
- le port UDP

La liste la plus complète peut être consultée sur le site Internet **www.iana.org**.

Pour connaitre la liste des sockets ouverts sur l'ordinateur, saisissez la commande suivante :

```
opensuse:~ # netstat -an | more
Active Internet connections (servers and established)
```

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
tcp        0      0 :::111                  :::*                    LISTEN
tcp        0      0 ::1:631                 :::*                    LISTEN
tcp        0      0 ::1:25                  :::*                    LISTEN
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp        0      0 0.0.0.0:48895           0.0.0.0:*
udp        0      0 0.0.0.0:989             0.0.0.0:*
udp        0      0 0.0.0.0:111             0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:9887            0.0.0.0:*
udp        0      0 :::29954                :::*
udp        0      0 :::989                  :::*
udp        0      0 :::546                  :::*
udp        0      0 :::111                  :::*
--More--
```

Pour connaitre la liste des applications ayant ouvert un port sur l'ordinateur, saisissez la commande suivante :

```
opensuse:~ # netstat -anp | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      3783/rpcbind
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      3830/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      3889/master
tcp        0      0 :::111                  :::*                    LISTEN      3783/rpcbind
tcp        0      0 ::1:631                 :::*                    LISTEN      3830/cupsd
tcp        0      0 ::1:25                  :::*                    LISTEN      3889/master
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           3448/avahi-daemon:
udp        0      0 0.0.0.0:48895           0.0.0.0:*                           3448/avahi-daemon:
udp        0      0 0.0.0.0:989             0.0.0.0:*                           3783/rpcbind
udp        0      0 0.0.0.0:111             0.0.0.0:*                           3783/rpcbind
```

```
udp        0      0 0.0.0.0:631             0.0.0.0:*                           3830/cupsd
udp        0      0 0.0.0.0:9887            0.0.0.0:*                           1697/dhclient6
udp        0      0 :::29954                :::*                                1697/dhclient6
udp        0      0 :::989                  :::*                                3783/rpcbind
udp        0      0 :::546                  :::*                                1697/dhclient6
udp        0      0 :::111                  :::*                                3783/rpcbind
--More--
```

Résolution d'adresses Ethernet

Chaque protocole peut être encapsulé dans une **trame** Ethernet. Lorsque la trame doit être transportée de l'expéditeur au destinataire, ce premier doit connaitre l'adresse Ethernet du dernier. L'adresse Ethernet est aussi appelée l'adresse **Physique** ou l'adresse **MAC**.

Pour connaître l'adresse Ethernet du destinataire, l'expéditeur fait appel au protocol **ARP**. Les informations reçues sont stockées dans une table. Pour visualiser ces informations, il convient d'utiliser la commande suivante :

```
opensuse:~ # arp -a
? (10.0.2.2) at 52:54:00:12:35:02 [ether] on eth0
```

Options de la commande

Les options de cette commande sont :

```
opensuse:~ # arp --help
Usage:
  arp [-vn]  [<HW>] [-i <if>] [-a] [<hostname>]             <-Display ARP cache
  arp [-v]          [-i <if>] -d  <hostname> [pub][nopub]    <-Delete ARP entry
  arp [-vnD] [<HW>] [-i <if>] -f  [<filename>]              <-Add entry from file
  arp [-v]   [<HW>] [-i <if>] -s  <hostname> <hwaddr> [temp][nopub] <-Add entry
  arp [-v]   [<HW>] [-i <if>] -s  <hostname> <hwaddr> [netmask <nm>] pub  <-''-
  arp [-v]   [<HW>] [-i <if>] -Ds <hostname> <if> [netmask <nm>] pub      <-''-
```

```
        -a                      display (all) hosts in alternative (BSD) style
        -s, --set               set a new ARP entry
        -d, --delete            delete a specified entry
        -v, --verbose           be verbose
        -n, --numeric           don't resolve names
        -i, --device            specify network interface (e.g. eth0)
        -D, --use-device        read <hwaddr> from given device
        -A, -p, --protocol      specify protocol family
        -f, --file              read new entries from file or from /etc/ethers

  <HW>=Use '-H <hw>' to specify hardware address type. Default: ether
  List of possible hardware types (which support ARP):
    strip (Metricom Starmode IP) ether (Ethernet) tr (16/4 Mbps Token Ring)
    tr (16/4 Mbps Token Ring (New)) ax25 (AMPR AX.25) netrom (AMPR NET/ROM)
    arcnet (ARCnet) dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface)
    hippi (HIPPI) irda (IrLAP) x25 (generic X.25)
    infiniband (InfiniBand)
```

Configuration de TCP/IP

La configuration TCP/IP se trouve dans le répertoire **/etc/sysconfig/network**. Les fichiers importants sont :

/etc/sysconfig/network/config

Ce fichier contient des directives applicables à toutes les interfaces réseau :

```
opensuse:~ # cat /etc/sysconfig/network/config
## Path:    Network/General
## Description: Set some general network configuration
## Type:    string("","-","+")
## Default: "+"
```

```
## ServiceRestart: network
#
# DEFAULT_BROADCAST is used when no individual BROADCAST is set. It can get one
# of the following values:
# ""  : don't set a broadcast address
# "-" : use IPADDR with all host bits deleted
# "+" : use IPADDR with all host bits set
DEFAULT_BROADCAST="+"

## Type:    yesno
## Default: yes
# sometimes we want some script to be executed after an interface has been
# brought up, or before an interface is taken down.
# default dir is /etc/sysconfig/network/if-up.d for POST_UP and
# /etc/sysconfig/network/if-down.d for PRE_DOWN
# Note: if you use NetworkManager then down scripts will be called after the
# interface is down and not before.
GLOBAL_POST_UP_EXEC="yes"
GLOBAL_PRE_DOWN_EXEC="yes"

## Type:        yesno
## Default:     no
# If ifup should check if an ip address is already in use, set this to yes.
# Make sure that packet sockets (CONFIG_PACKET) are supported in the kernel,
# since this feature uses arping, which depends on that.
# Also be aware that this takes one second per interface; consider that when
# setting up a lot of interfaces.
CHECK_DUPLICATE_IP="no"

## Type:        yesno
## Default:     no
# Switch on/off debug messages for all network configuration stuff. If set to no
# most scripts can enable it locally with "-o debug".
DEBUG="no"
```

```
## Type:        yesno
## Default:     yes
# All error and info messages from network and hardware configuration scripts go
# to stderr. Most tools that call sysconfig scripts (udev, rcnetwork, scpm,
# YaST) catch these messages and can log them. So some messages appear twice in
# syslog. If you don't like that, then set USE_SYSLOG=no.
USE_SYSLOG="yes"

# Handling of network connections
# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
# These features are designed for the convenience of the experienced
# user. If you encounter problems you don't understand then switch
# them off. That is the default.
# Please do not complain if you get troubles. But if you want help to
# make them smarter write to <http://www.suse.de/feedback>.

## Type:    yesno
## Default: no
#
# If you are interested in the connections and nfs mounts that use a
# network interface, you can set CONNECTION_SHOW_WHEN_IFSTATUS="yes".
# Then you will see them with 'ifstatus <interface>' (or 'ifstatus
# <config>')
# This one _should_ never harm ;)
#
CONNECTION_SHOW_WHEN_IFSTATUS="no"

## Type:    yesno
## Default: no
#
# If an interface should be set down only if there are no active
# connections, then use CONNECTION_CHECK_BEFORE_IFDOWN="yes"
#
CONNECTION_CHECK_BEFORE_IFDOWN="no"
```

```
## Type:    yesno
## Default: no
#
# If these connetions (without the nfs mounts) should be closed when
# shutting down an interface, set CONNECTION_CLOSE_BEFORE_IFDOWN="yes".
# WARNING: Be aware that this may terminate applications which need
# one of these connections!
#
CONNECTION_CLOSE_BEFORE_IFDOWN="no"

## Type:    yesno
## Default: no
#
# If you are a mobile laptop user and like even nfs mounts to be
# closed when you leave your current workplace, then set
# CONNECTION_UMOUNT_NFS_BEFORE_IFDOWN="yes". This does only work
# if CONNECTION_CLOSE_BEFORE_IFDOWN="yes", too.
# WARNING: Be aware that this may terminate applications which use
# these nfs mounts as working directory. Be very carefull if your home
# is mounted via nfs!!!
# WARNING: This may even lead to hanging ifdown processes if there are
# processes that could not be terminated. If you are using
# hotpluggable devices (pcmcia, usb, firewire), first shut them down
# before unplugging!
#
CONNECTION_UMOUNT_NFS_BEFORE_IFDOWN="no"

## Type:    yesno
## Default: no
#
# If terminating processes that use a connection or nfs mount is not
# enough, then they can be killed after an unsuccesfull termination.
# If you want that set CONNECTION_SEND_KILL_SIGNAL="yes"
#
```

```
CONNECTION_SEND_KILL_SIGNAL="no"

## Type:        string
## Default:     ""
#
# Here you may specify which interfaces have to be up and configured properly
# after 'rcnetwork start'. rcconfig will return 'failed' if any of these
# interfaces is not up. You may use interface names as well but better use
# hardware descriptions of the devices (eth-id-<macaddress> or eth-bus-...  See
# man ifup for 'hardware description'). The network start script will wait for
# these interfaces, but not longer as set in WAIT_FOR_INTERFACES.
# You need not to add dialup or tunnel interfaces here, only physical devices.
# The interface 'lo' is always considered to be mandatory and can be omitted.
#
# If this variable is empty, rcnetwork tries to derive the list of mandatory
# devices automatically from the list of existing configurations. Configurations
# with names bus-pcmcia or bus-usb or with STARTMODE=hotplug are skipped. (try
# '/etc/init.d/rc5.d/S*network start -o debug fake | grep MANDAT')
MANDATORY_DEVICES=""

## Type:    integer
## Default: 30
#
# Some interfaces need some time to come up or come asynchronously via hotplug.
# WAIT_FOR_INTERFACES is a global wait for all mandatory interfaces in
# seconds. If empty no wait occurs.
#
WAIT_FOR_INTERFACES="30"

## Type:    yesno
## Default: yes
#
# With this variable you can determine if the SuSEfirewall when enabled
# should get started when network interfaces are started.
```

```
FIREWALL="yes"

## Type:        string
## Default:     "eth*[0-9]|tr*[0-9]|wlan[0-9]|ath[0-9]"
#
# Automatically add a linklocal route to the matching interfaces.
# This string is used in a bash "case" statement, so it may contain
# '*', '[', ']'  and '|' meta-characters.
#
LINKLOCAL_INTERFACES="eth*[0-9]|tr*[0-9]|wlan[0-9]|ath[0-9]"

## Type:        string
## Default:     "-f -I"
#
# Set default options for ifplugd. You may also set them in an ifcfg-* file
# individually. Have a look at 'man ifplug' for details. We let ifplugd set the
# interface UP when starting, because there are many interfaces where link beat
# cannot be detected otherwise. If you want the interface to stay down then add
# the option '-a'. If you like ifplugd to beep on cable (un)plug, remove '-b'.
#
IFPLUGD_OPTIONS="-f -I -b"

## Type:        yesno
## Default:     no
#
# Instead of the usual network setup (now called 'NetControl') you may also
# use 'NetworkManager' to control your interfaces. This option is used by
# the /etc/init.d/network(-remotefs) script to control, which network stack
# has to be used: NetControl alias ifup or NetworkManager.
#
# NetControl is what you were used to in SUSE Linux up to now. It has a wide
# range of configurations means for setting up any number of different virtual
# and real interfaces. It should be used if you:
# - want a static network setup
```

```
# - have many interfaces
# - need VLAN, bonding, bridging, multiple IP addresses
# - must restrict network control to root
# It may also switch interfaces automatically, but does not provide a GUI.
# When you want a GUI, try out 'kinternet' using NetControl as backend.
#
# NetworkManager lets the user control interfaces and switches automatically if
# network interfaces lose/gain physical connection. It should be used if you:
# - move between networks frequently
# - want a GUI for network control
# Especially on mobile computers that use mainly one wired and one wireless
# interface NetworkManager may please you.
#
# If you are useing SCPM, then you might probably stay with NetControl. But at
# least try NetworkManager, because it can replace SCPM in some usage scenarios.
#
# Note: NetworkManager is not using any sysconfig settings but its own
#       configuration files only.
#
NETWORKMANAGER="no"

## Type:    int
## Default: 30
#
# When using NetworkManager you may define a timeout to wait for NetworkManager
# to connect in /etc/init.d/network(-remotefs) script.  Other network services
# may require the system to have a valid network setup in order to succeed.
#
# This variable has no effect if NETWORKMANAGER=no.
#
NM_ONLINE_TIMEOUT="30"

## Type:        string
## Default:     "dns-resolver dns-bind ntp-runtime nis"
```

```
#
# This variable defines the start order of netconfig modules installed
# in the /etc/netconfig.d/ directory.
#
# To disable the execution of a module, don't remove it from the list
# but prepend it with a minus sign, "-ntp-runtime".
#
NETCONFIG_MODULES_ORDER="dns-resolver dns-bind dns-dnsmasq nis ntp-runtime"

## Type:        string
## Default:     "auto"
#
# Defines the DNS merge policy as documented in netconfig(8) manual page.
# Set to "" to disable DNS configuration.
#
NETCONFIG_DNS_POLICY="auto"

## Type:        string(resolver,bind,dnsmasq,)
## Default:     "resolver"
#
# Defines the name of the DNS forwarder that has to be configured.
# Currently implemented are "bind", "dnsmasq" and "resolver", that
# causes to write the name server IP addresses to /etc/resolv.conf
# only (no forwarder). Empty string defaults to "resolver".
#
NETCONFIG_DNS_FORWARDER="resolver"

## Type:        yesno
## Default:     yes
#
# When enabled (default) in forwarder mode ("bind", "dnsmasq"),
# netconfig writes an explicit localhost nameserver address to the
# /etc/resolv.conf, followed by the policy resolved name server list
# as fallback for the moments, when the local forwarder is stopped.
```

```
#
NETCONFIG_DNS_FORWARDER_FALLBACK="yes"

## Type:        string
## Default:     ""
#
# List of DNS domain names used for host-name lookup.
# It is written as search list into the /etc/resolv.conf file.
#
NETCONFIG_DNS_STATIC_SEARCHLIST="fenestros.loc"

## Type:        string
## Default:     ""
#
# List of DNS nameserver IP addresses to use for host-name lookup.
# When the NETCONFIG_DNS_FORWARDER variable is set to "resolver",
# the name servers are written directly to /etc/resolv.conf.
# Otherwise, the nameserver are written into a forwarder specific
# configuration file and the /etc/resolv.conf does not contain any
# nameservers causing the glibc to use the name server on the local
# machine (the forwarder). See also netconfig(8) manual page.
#
NETCONFIG_DNS_STATIC_SERVERS=""

## Type:        string
## Default:     "auto"
#
# Allows to specify a custom DNS service ranking list, that is which
# services provide preferred (e.g. vpn services), and which services
# fallback settings (e.g. avahi).
# Preferred service names have to be prepended with a "+", fallback
# service names with a "-" character. The special default value
# "auto" enables the current build-in service ranking list -- see the
# netconfig(8) manual page -- "none" or "" disables the ranking.
```

```
#
NETCONFIG_DNS_RANKING="auto"

## Type:        string
## Default:     "auto"
#
# Defines the NTP merge policy as documented in netconfig(8) manual page.
# Set to "" to disable NTP configuration.
#
NETCONFIG_NTP_POLICY="auto"

## Type:        string
## Default:     ""
#
# List of NTP servers.
#
NETCONFIG_NTP_STATIC_SERVERS=""

## Type:        string
## Default:     "auto"
#
# Defines the NIS merge policy as documented in netconfig(8) manual page.
# Set to "" to disable NIS configuration.
#
NETCONFIG_NIS_POLICY="auto"

## Type:        string(yes,no,)
## Default:     "yes"
#
# Defines whether to set the default NIS domain. When enabled and no domain
# is provided dynamically or in static settings, /etc/defaultdomain is used.
# Valid values are:
#  - "no" or ""         netconfig does not set the domainname
#  - "yes"              netconfig sets the domainname according to the
```

```
#                       NIS policy using settings provided by the first
#                       iterface and service that provided it.
#  - "<interface name>" as yes, but only using settings from interface.
#
NETCONFIG_NIS_SETDOMAINNAME="yes"

## Type:        string
## Default:     ""
#
# Defines a default NIS domain.
#
# Further domain can be specified by adding a "_<number>" suffix to
# the NETCONFIG_NIS_STATIC_DOMAIN and NETCONFIG_NIS_STATIC_SERVERS
# variables, e.g.: NETCONFIG_NIS_STATIC_DOMAIN_1="second".
#
NETCONFIG_NIS_STATIC_DOMAIN=""

## Type:        string
## Default:     ""
#
# Defines a list of NIS servers for the default NIS domain or the
# domain specified with same "_<number>" suffix.
#
NETCONFIG_NIS_STATIC_SERVERS=""

## Type:    string
## Default: ''
#
# Set this variable global variable to the ISO / IEC 3166 alpha2
# country code specifying the wireless regulatory domain to set.
# When not empty, ifup-wireless will be set in the wpa_supplicant
# config or via 'iw reg set' command.
#
# Note: This option requires a wpa driver supporting it, like
```

```
# the 'nl80211' driver used by default since openSUSE 11.3.
# When you notice problems with your hardware, please file a
# bug report and set e.g. WIRELESS_WPA_DRIVER='wext' (the old
# default driver) in the ifcfg file.
# See also "/usr/sbin/wpa_supplicant --help" for the list of
# available wpa drivers.
#
WIRELESS_REGULATORY_DOMAIN=''
```

Les directives activées de ce fichier sont :

```
DEFAULT_BROADCAST="+"
GLOBAL_POST_UP_EXEC="yes"
GLOBAL_PRE_DOWN_EXEC="yes"
CHECK_DUPLICATE_IP="no"
DEBUG="no"
USE_SYSLOG="yes"
CONNECTION_SHOW_WHEN_IFSTATUS="no"
CONNECTION_CHECK_BEFORE_IFDOWN="no"
CONNECTION_CLOSE_BEFORE_IFDOWN="no"
CONNECTION_UMOUNT_NFS_BEFORE_IFDOWN="no"
CONNECTION_SEND_KILL_SIGNAL="no"
WAIT_FOR_INTERFACES="30"
FIREWALL="yes"
LINKLOCAL_INTERFACES="eth*[0-9]|tr*[0-9]|wlan[0-9]|ath[0-9]"
IFPLUGD_OPTIONS="-f -I -b"
NETWORKMANAGER="no"
NM_ONLINE_TIMEOUT="30"
NETCONFIG_MODULES_ORDER="dns-resolver dns-bind dns-dnsmasq nis ntp-runtime"
NETCONFIG_DNS_POLICY="auto"
NETCONFIG_DNS_FORWARDER="resolver"
NETCONFIG_DNS_FORWARDER_FALLBACK="yes"
NETCONFIG_DNS_STATIC_SEARCHLIST="fenestros.loc"
NETCONFIG_DNS_RANKING="auto"
```

```
NETCONFIG_NTP_POLICY="auto"
NETCONFIG_NIS_POLICY="auto"
NETCONFIG_NIS_SETDOMAINNAME="yes"
```

<note important> Veuillez noter que chaque directive est détaillée dans le fichier lui-même. Notez aussi qu'openSUSE n'utilise pas NetworkManager par défaut mais le système de scripts **if***, à savoir **ifup** et **ifdown** pour contrôler l'interface réseau. </note>

/etc/sysconfig/network/dhcp

Ce fichier spécifie les valeurs des directives utilisées par **dhcpcd** quand l'interface réseau est configurée en mode **dhcp** :

```
opensuse:~ # cat /etc/sysconfig/network/dhcp
## Path:    Network/DHCP/DHCP client
## Description: DHCP configuration tweaking
#
# Note:
# To configure one or more interfaces for DHCP configuration, you have to
# change the BOOTPROTO variable in /etc/sysconfig/network/ifcfg-<interface>
# to 'dhcp' (and possibly set STARTMODE='onboot').
#
# Most of these options are used only by dhcpcd, not by the ISC dhclient
# (which uses a config file).
#
# Most of the options can be overridden by setting them in the ifcfg-* files,
# too.
#
# Note: The ISC dhclient started by the NetworkManager is not using any
# of these options -- NetworkManager is not using any sysconfig settings.
#

## Type:    string
## Default: ""
## ServiceRestart: network
```

```
#
# Which DHCPv4 client should be used?
# If empty, dhcpcd is tried, then dhclient
# Other possible values:
#   dhcpcd   (DHCP client daemon)
#   dhclient (ISC dhclient)
DHCLIENT_BIN=""

## Type:    string
## Default: ""
## ServiceRestart: network
#
# Which DHCPv6 client should be used?
# Currently only the dhcp6c client is supported.
#
DHCLIENT6_BIN=""

## Type:        string
## Default:     ""
## ServiceRestart: network
#
# Additional user start options to use when the 'dhcpcd' DHCPv4 client
# is enabled in the DHCLIENT_BIN variable (default).
#
DHCPCD_USER_OPTIONS=""

## Type:        string
## Default:     ""
## ServiceRestart: network
#
# Additional user start options to use when the 'dhclient' ISC DHCPv4
# client is enabled in the DHCLIENT_BIN variable.
#
DHCLIENT_USER_OPTIONS=""
```

```
## Type:        string
## Default:     ""
## ServiceRestart: network
#
# Additional user start options to use when the 'dhcp6c' DHCPv6 client
# is enabled in the DHCLIENT6_BIN variable (default).
#
DHCP6C_USER_OPTIONS=""

## Type:    yesno
## Default: no
#
# Start in debug mode? (yes|no)
# (debug info will be logged to /var/log/messages for dhcpcd, or to
# /var/log/dhclient-script for ISC dhclient)
#
DHCLIENT_DEBUG="no"

## Type: list("",yes,no,first)
## Default: ""
#
# Multiple DHCP clients:
#
# With two or more DHCP clients running, they would concurrently try to replace
# the default route or set the hostname. There are several ways of dealing with
# this conflict (and it is a conflict, because you can have only one default
# route even though routes are stackable and the dhcp clients would change it
# while every lease renew):
#
#  1) Allow both clients to do that stuff. This would work in many cases if
#     only one of the interfaces is used at a time. However, it would lead to
#     undefined behaviour such as changing default route e.g. on dhcp renew.
#
#  2) When both interfaces are connected to the same network, you may configure
```

```
#     a bonding interface in active-backup mode (or another, e.g. 802.3ad, when
#     supported and configured by the switch) and configure dhcp on the bonding
#     instead.
#
#  3) When only one of the interfaces is used at time, you may set STARTMODE to
#     ifplugd and specify the priority of the interfaces in IFPLUGD_PRIORITY.
#     This is a common scenario for notebooks to use the wired interface when
#     connected, wireless otherwise.
#
#  4) allow only one of the DHCP clients to do that stuff.
#     This implies that there would be a "primary" interface and a "secondary".
#     This is the assumption the default configuration is based on. But since
#     the system often can't guess which interface is "more important", we
#     simply choose one depending on related configuration or take the first
#     interface that is started with DHCP to be primary ("authoritative").
#     This can be configured by setting DHCLIENT_PRIMARY_DEVICE=yes in one of
#     the /etc/sysconfig/network/ifcfg-* files and DHCLIENT_PRIMARY_DEVICE=no
#     in /etc/sysconfig/network/dhcp (or in all other ifcfg files using DHCP).
#
# When DHCLIENT_PRIMARY_DEVICE is not explicitly configured to yes/no, the
# "primary" interface is choosed as follows:
#
# - On systems with iSCSI Boot Firmare Table, the iBFT primary interface
#   is used as the primary DHCP interface by default.
# - On systems booting via PXE, the interface specified by the BOOTIF kernel
#   parameter is used as primary DHCP interface. Set the global "ipappend 2"
#   parameter in pxelinux.cfg/* files, so the BOOTIF kernel parameter is set.
# - Otherwise, the DHCP client that is started first will be "primary" and
#   allowed the set the default route and hostname ("first up wins" mode,
#   the only one before openSUSE 11.4). To force this "first up wins" mode,
#   set DHCLIENT_PRIMARY_DEVICE="first" in /etc/sysconfig/network/dhcp.
#
# All other running dhcp clients will only configure the interface with an
# address and network routes, but not change the "global" default route or
```

```
# hostname.
# See also DHCLIENT_SET_DEFAULT_ROUTE and DHCLIENT_SET_HOSTNAME variables,
# that allow to modify the DHCLIENT_PRIMARY_DEVICE parameter behaviour once
# again.
#
# Thus, to specifically allow an interface's DHCP client to change "global"
# configuration, set the following variable to "yes". Or you can make an
# interface's DHCP client never change these settings if you set it to "no".
# If you leave it empty then ifup-dhcp will decide.
#
DHCLIENT_PRIMARY_DEVICE=""

## Type:    yesno
## Default: no
#
# Should the DHCP client set the hostname? (yes|no)
#
# When it is likely that this would occur during a running X session,
# your DISPLAY variable could be screwed up and you won't be able to open
# new windows anymore, then this should be "no".
#
# If it happens during booting it won't be a problem and you can
# safely say "yes" here. For a roaming notebook with X kept running, "no"
# makes more sense.
#
DHCLIENT_SET_HOSTNAME="yes"

## Type:    yesno
## Default: yes
#
# Should the DHCP client set a default route (default Gateway) (yes|no)
#
# When multiple copies of dhcpcd run, it would make sense that only one
# of them does it.
```

```
#
DHCLIENT_SET_DEFAULT_ROUTE="yes"

## Type:    integer
## Default: ""
#
# Lease time to request ( -l option)
#
# Specifies (in seconds) the lease that is suggested to the server.
# The default is 1 hour, use -1 to request infinite lease time.
#
DHCLIENT_LEASE_TIME=""

## Type:        yesno
## Default:     yes
#
# dhcpcd -E/--lastlease option
#
# This setting controls whether dhcpcd should try to use DHCP settings
# provided in its last lease when the dhcp-server is not reachable and
# the lease hasn't expired yet.
# Set this variable to "no" to disable the fallback to the last lease.
#
DHCLIENT_USE_LAST_LEASE="yes"

## Type:    integer
## Default: "0"
#
# dhcpcd -t/--timeout option
#
# You can set the timeout - dhcpcd will terminate after this time when
# does not get a reply from the dhcp server. The dhcpcd default timeout
# is 20 seconds, we set it to 0 to and wait forever to get a lease.
#
```

```
# Note: In the past, this setting was set to a much higher value (999999)
# by default, because the dhcpcd < 3.2.3 didn't provided a infinite one.
#
DHCLIENT_TIMEOUT="0"

## Type:    string
## Default: AUTO
#
# specify a hostname to send ( -h option)
#
# specifies a string used for the hostname option field when dhcpcd sends DHCP
# messages. Some DHCP servers will update nameserver entries (dynamic DNS).
# Also, some DHCP servers, notably those used by @Home Networks, require the
# hostname option field containing a specific string in the DHCP messages from
# clients.
#
# By default the current hostname is sent ("AUTO"), if one is defined in
# /etc/HOSTNAME.
# Use this variable to override this with another hostname, or leave empty
# to not send a hostname.
#
DHCLIENT_HOSTNAME_OPTION="AUTO"

## Type:    string
## Default: ""
#
# specify a client ID ( -I option)
#
# Specifies a client identifier string. By default the hardware address of the
# network interface is sent as client identifier string, if none is specified
# here.
#
# Note that dhcpcd will prepend a zero to what it sends to the server. In the
# server configuration, you need to write the following to match on it:
```

```
#  option dhcp-client-identifier "\0foo";
#
DHCLIENT_CLIENT_ID=""

## Type:    string("dhcpcd dhclient")
## Default: ""
#
# specify a vendor class ID ( -i option)
#
# Specifies the vendor class identifier string. The default is dhcpcd-<version>.
#
DHCLIENT_VENDOR_CLASS_ID=""

## Type:    yesno
## Default: no
#
# Send a DHCPRELEASE to the server (sign off the address)? (yes|no)
# This may lead to getting a different address/hostname next time an address
# is requested. But some servers require it.
#
DHCLIENT_RELEASE_BEFORE_QUIT="no"

## Type:    yesno
## Default: no
#
# Send a DHCPv6 RELEASE to the server (sign off the address)? (yes|no)
# This may lead to getting a different address/hostname next time an address
# is requested. But some servers require it.
#
DHCLIENT6_RELEASE_BEFORE_QUIT="no"

## Type:    integer
## Default: 0
#
```

```
# Some interfaces need time to initialize. Add the latency time in seconds
# so these can be handled properly. Should probably set per interface rather than here.
#
DHCLIENT_SLEEP="0"

## Type:    integer
## Default: 15
#
# When the DHCP client is started at boot time, the boot process will stop
# until the interface is successfully configured, but at most for
# DHCLIENT_WAIT_AT_BOOT seconds.
#
# Note: RFC 2131 specifies, that the dhcp client should wait a random time
# between one and ten seconds to desynchronize the use of DHCP at startup.
#
DHCLIENT_WAIT_AT_BOOT="15"

## Type:        yesno
## Default:     yes
## ServiceRestart: yast2
#
# This option is read by YaST during network configuration.
#
# If set, then the hostname is added to /etc/hosts with IP address
# 127.0.0.2. This allows the hostname to be resolved (and thus, the
# host to be reached), if the real network is not reachable.
#
# If unset, YaST will not touch /etc/hosts.
WRITE_HOSTNAME_TO_HOSTS="yes"
## Path:    Network/DHCP/DHCP client
## Description: DHCP client configuration
## Type:    yesno
## Default: yes
#
```

```
# Should the DHCP client modify /etc/samba/dhcp.conf?
#
DHCLIENT_MODIFY_SMB_CONF="yes"
```

Les directives activées de ce fichier sont :

```
DHCLIENT_DEBUG="no"
DHCLIENT_SET_HOSTNAME="yes"
DHCLIENT_SET_DEFAULT_ROUTE="yes"
DHCLIENT_USE_LAST_LEASE="yes"
DHCLIENT_TIMEOUT="0"
DHCLIENT_HOSTNAME_OPTION="AUTO"
DHCLIENT_RELEASE_BEFORE_QUIT="no"
DHCLIENT6_RELEASE_BEFORE_QUIT="no"
DHCLIENT_SLEEP="0"
DHCLIENT_WAIT_AT_BOOT="15"
WRITE_HOSTNAME_TO_HOSTS="yes"
DHCLIENT_MODIFY_SMB_CONF="yes"
```

<note important> Les valeurs des directives ne sont ni utilisées par **NetworkManager**, ni utilisées par **dhclient**. </note>

/etc/sysconfig/network/ifcfg-ethX

Chaque interface réseau a son fichier de configuration propre. Dans le cas de l'interface **eth0**, le fichier est **/etc/sysconfig/networkifcfg-eth0** :

```
opensuse:~ # cat /etc/sysconfig/network/ifcfg-eth0
BOOTPROTO='dhcp'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR=''
MTU=''
NAME='82540EM Gigabit Ethernet Controller'
```

```
NETMASK=''
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='nfsroot'
USERCONTROL='no'
```

<note important> Toutes les variables se trouvant dans les fichiers **/etc/sysconfig/network/config** et **/etc/sysconfig/network/dhcp** peuvent être utilisées dans ce fichier au quel cas elles sont prioritaires par rapport à celles dans les deux fichiers de configuration générique. </note>

Dans le cas de l'exemple ci dessus, l'interface est en mode **dhcp**. Pour configurer l'interface en IP fixe, modifiez le fichier ainsi :

```
opensuse:~ # cat /etc/sysconfig/network/ifcfg-eth0
BOOTPROTO='static'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR='10.0.2.15/24'
MTU='1500'
NAME='82540EM Gigabit Ethernet Controller'
NETMASK='255.255.255.0'
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='auto'
USERCONTROL='yes'
```

<note warning> La configuration n'est pas encore complète car il faut configurer manuellement les serveurs DNS et la passerelle ! </note>

/etc/resolv.conf

Ce fichier contient une liste de serveurs DNS.

Modifiez donc le fichier ainsi :

```
opensuse:~ # cat /etc/resolv.conf
```

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

<note important> Notez que les DNS utilisés sont les serveurs DNS publics de Google. </note>

Afin de faire en sorte que vos modifications soient prises en charge par le système, il est nécessaire de modifier les fichiers **/etc/sysconfig/network/config** et **/etc/sysconfig/network/dhcp**.

Modifiez donc le fichier **/etc/sysconfig/network/config** ainsi :

```
...
# NETCONFIG_DNS_POLICY="auto"
...
# NETCONFIG_DNS_STATIC_SEARCHLIST="fenestros.loc"
NETCONFIG_DNS_STATIC_SERVERS=""
...
# NETCONFIG_DNS_STATIC_SEARCHLIST=""
NETCONFIG_DNS_STATIC_SERVERS="8.8.8.8 8.8.4.4"
...
```

Puis modifiez le fichier **/etc/sysconfig/network/dhcp** ainsi :

```
...
# DHCLIENT_SET_HOSTNAME="yes"
DHCLIENT_SET_HOSTNAME="no"
...
```

Redémarrez maintenant le service réseau afin que les modifications soient prises en compte :

```
opensuse:~ # service network restart
Shutting down network interfaces:
    eth0      device: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
    eth0                                                                                done
Shutting down service network  .  .  .  .  .  .  .  .  .                                done
```

```
Hint: you may set mandatory devices in /etc/sysconfig/network/config
Setting up network interfaces:
    eth0      device: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
    eth0      IP address: 10.0.2.15/24
    eth0                                                                                    done
Setting up service network  .  .  .  .  .  .  .  .  .  .                                   done
SuSEfirewall2: Setting up rules from /etc/sysconfig/SuSEfirewall2 ...
SuSEfirewall2: Firewall rules successfully set
```

Tester la Configuration d'IP Fixe

Utilisez maintenant la commande **ifstatus** pour vérifier la configuration d'eth0 :

```
opensuse:~ # ifstatus eth0
    eth0      device: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
eth0 is up
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:5a:43:78 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe5a:4378/64 scope link
       valid_lft forever preferred_lft forever
    eth0      IP address: 10.0.2.15/24
Configured IPv4 routes for interface eth0:
  169.254.0.0/16 - - eth0
Active IPv4 routes for interface eth0:
  169.254.0.0/16  scope link
1 of 1 configured IPv4 routes for interface eth0 up
```

<note important> Notez que dans la section “Configured IPv4 routes for interface eth0” il n'existe pas de passerelle par défaut. </note>

VirtualBox fournit une passerelle par défaut à l'adresse IP 10.0.2.2. Insérez donc cette information dans la table de routage du noyau avec la commande suivante :

```
opensuse:~ # route add default gw 10.0.2.2 eth0
```

<note important> La commande **route** est traitée dans le détail à la fin de cette unité. </note>

Utilisez de nouveau la commande **ifstatus** pour vérifier la configuration d'eth0 :

```
opensuse:~ # ifstatus eth0
    eth0      device: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
eth0 is up
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:5a:43:78 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe5a:4378/64 scope link
       valid_lft forever preferred_lft forever
    eth0      IP address: 10.0.2.15/24
Configured IPv4 routes for interface eth0:
  default 10.0.2.2 - eth0
  169.254.0.0/16 - - eth0
Active IPv4 routes for interface eth0:
  169.254.0.0/16  scope link
  default via 10.0.2.2    <--------------------------la passerelle par défaut
2 of 2 configured IPv4 routes for interface eth0 up
```

<note important> Notez que la passerelle par défaut a été renseignée. </note>

/etc/HOSTNAME

Ce fichier contient le nom d'hôte de la machine :

```
opensuse:~ # cat /etc/HOSTNAME
opensuse.fenestros.loc
```

/etc/networks

Ce fichier contient la correspondance entre des noms de réseaux et l'adresse IP du réseau :

```
opensuse:~ # cat /etc/networks
#
# networks  This file describes a number of netname-to-address
#       mappings for the TCP/IP subsystem.  It is mostly
#       used at boot time, when no name servers are running.
#

loopback    127.0.0.0
link-local  169.254.0.0

# End.
```

/etc/nsswitch.conf

L'ordre de recherche des services de noms est stocké dans le fichier **/etc/nsswitch.conf**. Pour connaître l'ordre, saisissez la commande suivante :

```
opensuse:~ # grep '^hosts:' /etc/nsswitch.conf
hosts:      files mdns4_minimal [NOTFOUND=return] dns
```

/etc/hosts

Le mot **files** dans la sortie de la commande précédente fait référence au fichier **/etc/hosts** :

```
opensuse:~ # cat /etc/hosts
#
# hosts         This file describes a number of hostname-to-address
```

```
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1       localhost

# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback

fe00::0         ipv6-localnet

ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts
127.0.0.2       opensuse.fenestros.loc opensuse
10.0.2.15       opensuse.fenestros.loc opensuse
```

Tester la Configuration de la Résolution des Noms

Pour tester la configuration de la résolution des noms, deux commandes sont possibles, **nslookup** et **dig** :

```
opensuse:~ # nslookup www.linuxelearning.com
Server:     8.8.8.8
Address:    8.8.8.8#53

Non-authoritative answer:
www.linuxelearning.com  canonical name = linuxelearning.com.
```

```
Name:   linuxelearning.com
Address: 212.198.31.61

opensuse:~ # dig www.linuxelearning.com

; <<>> DiG 9.7.3 <<>> www.linuxelearning.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19519
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linuxelearning.com.        IN    A

;; ANSWER SECTION:
www.linuxelearning.com. 41207    IN    CNAME    linuxelearning.com.
linuxelearning.com. 13    IN    A    212.198.31.61

;; Query time: 33 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue May 15 11:33:54 2012
;; MSG SIZE  rcvd: 70
```

Services réseaux

Quand un client émet une demande de connexion vers une application réseau sur un serveur, il utilise un socket attaché à un port local **supérieur à 1023**, alloué d'une manière dynamique. La requête contient le port de destination sur le serveur. Certaines applications serveurs se gèrent toutes seules, ce qui est la cas par exemple d'**httpd**. Par contre d'autres sont gérées par le service **xinetd**.

xinetd

Sous openSUSE xinetd est installé par défaut :

Le programme xinetd est configuré via le fichier **/etc/xinetd.conf** :

```
opensuse:~ # cat /etc/xinetd.conf
#
# xinetd.conf
#
# Copyright (c) 1998-2001 SuSE GmbH Nuernberg, Germany.
# Copyright (c) 2002 SuSE Linux AG, Nuernberg, Germany.
#

defaults
{
    log_type        = FILE /var/log/xinetd.log
        log_on_success  = HOST EXIT DURATION
    log_on_failure    = HOST ATTEMPT
#        only_from       = localhost
        instances       = 30
    cps        = 50 10

#
# The specification of an interface is interesting, if we are on a firewall.
# For example, if you only want to provide services from an internal
# network interface, you may specify your internal interfaces IP-Address.
#
#   interface    = 127.0.0.1

}

includedir /etc/xinetd.d
```

Ce fichier ne définit pas les applications serveurs directement. Il indique plutôt le répertoire qui contient les fichiers de définitions des applications serveurs qui est **/etc/xinetd.d** :

```
opensuse:~ # ls -l /etc/xinetd.d
total 76
-rw-r--r-- 1 root root  313 Feb 18  2011 chargen
-rw-r--r-- 1 root root  333 Feb 18  2011 chargen-udp
-rw-r--r-- 1 root root  256 Apr 29  2011 cups-lpd
-rw-r--r-- 1 root root  313 Feb 18  2011 daytime
-rw-r--r-- 1 root root  333 Feb 18  2011 daytime-udp
-rw-r--r-- 1 root root  313 Feb 18  2011 discard
-rw-r--r-- 1 root root  332 Feb 18  2011 discard-udp
-rw-r--r-- 1 root root  305 Feb 18  2011 echo
-rw-r--r-- 1 root root  324 Feb 18  2011 echo-udp
-rw-r--r-- 1 root root  492 Feb 18  2011 netstat
-rw-r--r-- 1 root root  207 Apr  4  2011 rsync
-rw-r--r-- 1 root root  337 Apr  6  2011 sane-port
-rw-r--r-- 1 root root  332 Feb 18  2011 servers
-rw-r--r-- 1 root root  334 Feb 18  2011 services
-rw-r--r-- 1 root root  277 Mar  1  2011 swat
-rw-r--r-- 1 root root  536 Feb 22  2011 systat
-rw-r--r-- 1 root root  339 Feb 18  2011 time
-rw-r--r-- 1 root root  333 Feb 18  2011 time-udp
-rw-r--r-- 1 root root 2317 May 17  2011 vnc
```

A l'examen de ce répertoire vous noterez que celui-ci contient des fichiers nominatifs par application-serveur, par exemple pour le serveur rsync :

```
opensuse:~ # cat /etc/xinetd.d/rsync
# default: off
# description: rsync file transfer daemon
service rsync
{
    socket_type    = stream
    protocol    = tcp
    wait        = no
    user        = root
    server        = /usr/sbin/rsyncd
```

```
    server_args    = --daemon
    disable        = yes
}
```

Les directives principales de ce fichier sont :

Paramètre	Déscription
disable	**no** : Le service est actif. **yes** : Le service est désactivé
port	Le numéro de port ou, à défaut, le numéro indiqué pour le service dans le fichier /etc/services
socket_type	Nature du socket, soit **stream** pour TCP soit **dgram** pour UDP
protocol	Protocole utilisé soit **TCP** soit **UDP**
wait	**no** : indique si xinetd active un serveur par client. **yes** : indique que xinetd active un seul serveur pour tous les client
user	Indique le compte sous lequel le serveur est exécuté
server	Indique le chemin d'accès de l'application serveur
env	Définit un environnement système
server_args	Donne les arguments transmis à l'application serveur

Afin d'activer une application serveur, il suffit de modifier le paramètre **disable** dans le fichier concerné et de relancer le service xinetd.

TCP Wrapper

TCP Wrapper contrôle l'accès à des services réseaux grâce à des **ACL**.

Quand une requête arrive pour un serveur, xinetd active le wrapper **tcpd** au lieu d'activer le serveur directement.

tcpd met à jour un journal et vérifie si le client a le droit d'utiliser le service concerné. Les ACL se trouvent dans deux fichiers:

- **/etc/hosts.allow**
- **/etc/hosts.deny**

Il faut noter que si ces fichiers n'existent pas ou sont vides, il n'y a pas de contrôle d'accès.

Le format d'une ligne dans un de ces deux fichiers est:

```
démon : liste_de_clients
```

Par exemple dans le cas d'un serveur **démon**, on verrait une ligne dans le fichier **/etc/hosts.allow** similaire à:

```
démon : LOCAL, .fenestros.loc
```

ce qui implique que les machines dont le nom ne comporte pas de point ainsi que les machines du domaine **fenestros.loc** sont autorisées à utiliser le service.

Le mot clef **ALL** peut être utilisé pour indiquer tout. Par exemple, **ALL:ALL** dans le fichier **/etc/host.deny** bloque effectivement toute tentative de connexion à un service xinetd sauf pour les ACL inclus dans le fichier /etc/host.allow.

Commandes de base

hostname

Lors du passage à une configuration en IPv4 fixe vous avez modifié la directive **HOSTNAME** du fichier **/etc/sysconfig/network** de **centos** à **centos.fenestros.loc**. Afin d'informer le système immédiatement de la modification du FQDN (*Fully Qualified Domain Name*), utilisez la commande **hostname** :

```
opensuse:~ # hostname
opensuse
opensuse:~ # hostname opensuse.fenestros.loc
opensuse:~ # hostname
opensuse.fenestros.loc
```

Pour afficher le FQDN du système vous pouvez également utiliser la commande suivante :

```
opensuse:~ # uname -n
opensuse.fenestros.loc
```

Options de la commande hostname

Les options de cette commande sont :

```
opensuse:~ # hostname --help
Usage: hostname [-v] {hostname|-F file}      set hostname (from file)
       domainname [-v] {nisdomain|-F file}   set NIS domainname (from file)
       hostname [-v] [-d|-f|-s|-a|-i|-y|-n]  display formatted name
       hostname [-v]                         display hostname

       hostname -V|--version|-h|--help       print info and exit

    dnsdomainname=hostname -d, {yp,nis,}domainname=hostname -y

    -s, --short           short host name
    -a, --alias           alias names
    -i, --ip-address      addresses for the hostname
    -f, --fqdn, --long    long host name (FQDN)
    -d, --domain          DNS domain name
    -y, --yp, --nis       NIS/YP domainname
    -F, --file            read hostname or NIS domainname from given file

   This command can read or set the hostname or the NIS domainname. You can
   also read the DNS domain or the FQDN (fully qualified domain name).
   Unless you are using bind or NIS for host lookups you can change the
   FQDN (Fully Qualified Domain Name) and the DNS domain name (which is
   part of the FQDN) in the /etc/hosts file.
```

ifconfig

Pour afficher la configuration IP de la machine il faut saisir la commande suivante :

```
opensuse:~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:5A:43:78
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5a:4378/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37329 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45185 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23538694 (22.4 Mb)  TX bytes:32937191 (31.4 Mb)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3801 (3.7 Kb)  TX bytes:3801 (3.7 Kb)
```

La commande ifconfig est également utilisée pour configurer une interface.

Créez maintenant une interface fictive ainsi :

```
opensuse:~ # ifconfig eth0:1 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
```

Constatez maintenant le résultat :

```
opensuse:~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:5A:43:78
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5a:4378/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37359 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45222 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:23546790 (22.4 Mb)  TX bytes:32941344 (31.4 Mb)

eth0:1    Link encap:Ethernet  HWaddr 08:00:27:5A:43:78
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3801 (3.7 Kb)  TX bytes:3801 (3.7 Kb)
```

Options de la commande ifconfig

Les options de cette commande sont :

```
opensuse:~ # ifconfig --help
Usage:
  ifconfig [-a] [-i] [-v] [-s] <interface> [[<AF>] <address>]
  [add <address>[/<prefixlen>]]
  [del <address>[/<prefixlen>]]
  [[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
  [netmask <address>]  [dstaddr <address>]  [tunnel <address>]
  [outfill <NN>] [keepalive <NN>]
  [hw <HW> <address>]  [metric <NN>]  [mtu <NN>]
  [[-]trailers]  [[-]arp]  [[-]allmulti]
  [multicast]  [[-]promisc]
  [mem_start <NN>]  [io_addr <NN>]  [irq <NN>]  [media <type>]
  [txqueuelen <NN>]
```

```
  [[-]dynamic]
  [up|down] ...

  <HW>=Hardware Type.
  List of possible hardware types:
    loop (Local Loopback) slip (Serial Line IP) cslip (VJ Serial Line IP)
    slip6 (6-bit Serial Line IP) cslip6 (VJ 6-bit Serial Line IP) adaptive (Adaptive Serial Line IP)
    strip (Metricom Starmode IP) ether (Ethernet) tr (16/4 Mbps Token Ring)
    tr (16/4 Mbps Token Ring (New)) ax25 (AMPR AX.25) netrom (AMPR NET/ROM)
    tunnel (IPIP Tunnel) ppp (Point-to-Point Protocol) arcnet (ARCnet)
    dlci (Frame Relay DLCI) frad (Frame Relay Access Device) sit (IPv6-in-IPv4)
    fddi (Fiber Distributed Data Interface) hippi (HIPPI) irda (IrLAP)
    x25 (generic X.25) infiniband (InfiniBand)
  <AF>=Address family. Default: inet
  List of possible address families:
    unix (UNIX Domain) inet (DARPA Internet) inet6 (IPv6)
    ax25 (AMPR AX.25) netrom (AMPR NET/ROM) ipx (Novell IPX)
    ddp (Appletalk DDP) x25 (CCITT X.25)
```

ifstatus

Rappelez-vous que cette commande fournit des informations concernant l'interface réseau passée en argument :

```
opensuse:~ # ifstatus eth0
    eth0      device: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
eth0 is up
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:5a:43:78 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
    inet 192.168.1.2/24 brd 192.168.1.255 scope global eth0:1
    inet6 fe80::a00:27ff:fe5a:4378/64 scope link
       valid_lft forever preferred_lft forever
    eth0      IP address: 10.0.2.15/24
```

```
    eth0:1    IP address: 192.168.1.2/24
Configured IPv4 routes for interface eth0:
  default 10.0.2.2 - eth0
  169.254.0.0/16 - - eth0
Active IPv4 routes for interface eth0:
  169.254.0.0/16  scope link
  default via 10.0.2.2
2 of 2 configured IPv4 routes for interface eth0 up
```

Options de la commande ifstatus

Les options de cette commande sont :

```
opensuse:~ # ifstatus --help

Usage: if{up,down,status} [<config>] <interface> [-o <options>]

Options are:
    [on]boot : we are currently booting (or shutting down)
    auto     : alias for boot
    hotplug  : we are handling a hotplug event
    manual   : we do it manually (default, just needed with 'rc'
    rc       : we are called by a rc script (implies auto)
    dhcp     : we are called from dhcp client
    prov=<n> : use provider <n> (for dial up interface)
    nodeps   : don't shut down interfaces depending on this
    debug    : be verbose
    syslog   : write to syslog if USE_SYSLOG=yes
    check    : return R_BUSY (=10) if there are
               active connections on this interface
If options are contradictionary, last option wins. Unknown options
are simply ignored, so be careful.
```

ping

Pour tester l'accessibilité d'une machine, vous devez utiliser la commande **ping** :

```
opensuse:~ # ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56(84) bytes of data.
64 bytes from 10.0.2.2: icmp_req=1 ttl=63 time=0.233 ms
64 bytes from 10.0.2.2: icmp_req=2 ttl=63 time=0.350 ms
64 bytes from 10.0.2.2: icmp_req=3 ttl=63 time=0.348 ms
64 bytes from 10.0.2.2: icmp_req=4 ttl=63 time=0.463 ms
64 bytes from 10.0.2.2: icmp_req=5 ttl=63 time=0.367 ms
^C
--- 10.0.2.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.233/0.352/0.463/0.074 ms
```

Options de la commande ping

Les options de cette commande sont :

```
opensuse:~ # ping --help
ping: invalid option -- '-'
Usage: ping [-LRUbdfnqrvVaAD] [-c count] [-i interval] [-w deadline]
            [-p pattern] [-s packetsize] [-t ttl] [-I interface]
            [-M pmtudisc-hint] [-m mark] [-S sndbuf]
            [-T tstamp-options] [-Q tos] [hop1 ...] destination
```

netstat -i

Pour visualiser les statistiques réseaux, vous disposez de la commande **netstat** :

```
opensuse:~ # netstat -i
Kernel Interface table
Iface   MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0    1500   0    38605      0      0      0    47019      0      0      0 BMRU
eth0:1  1500   0     - no statistics available -                        BMRU
lo     16436   0       30      0      0      0       30      0      0      0 LRU
```

Options de la commande netstat

Les options de cette commande sont :

```
opensuse:~ # netstat --help
usage: netstat [-veenNcCF] [<Af>] -r         netstat {-V|--version|-h|--help}
       netstat [-vnNcaeol] [<Socket> ...]
       netstat { [-veenNac] -i | [-cnNe] -M | -s }

        -r, --route              display routing table
        -i, --interfaces         display interface table
        -g, --groups             display multicast group memberships
        -s, --statistics         display networking statistics (like SNMP)
        -M, --masquerade         display masqueraded connections

        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        --numeric-hosts          don't resolve host names
        --numeric-ports          don't resolve port names
        --numeric-users          don't resolve user names
        -N, --symbolic           resolve hardware names
        -e, --extend             display other/more information
        -p, --programs           display PID/Program name for sockets
        -c, --continuous         continuous listing
```

```
       -l, --listening          display listening server sockets
       -a, --all, --listening   display all sockets (default: connected)
       -o, --timers             display timers
       -F, --fib                display Forwarding Information Base (default)
       -C, --cache              display routing cache instead of FIB

       -T, --notrim             dont't trim address information
  <Socket>={-t|--tcp} {-u|--udp} {-w|--raw} {-x|--unix} --ax25 --ipx --netrom --sctp
  <AF>=Use '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
```

Routage Statique

La commande route

Pour afficher la table de routage de la machine vous pouvez utiliser la commande **route** :

```
opensuse:~ # route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
10.0.2.0        *               255.255.255.0   U     0      0        0 eth0
link-local      *               255.255.0.0     U     0      0        0 eth0
loopback        *               255.0.0.0       U     0      0        0 lo
default         10.0.2.2        0.0.0.0         UG    0      0        0 eth0
```

La table issue de la commande **route** indique les informations suivantes:

- La destination qui peut être un hôte ou un réseau et est identifiée par les champs **Destination** et **Genmask**

- La route à prendre identifiée par les champs **Gateway** et **Iface**. Dans le cas d'une valeur de 0.0.0.0 ceci spécifie une route directe. La valeur d'Iface spécifie la carte à utiliser,
- Le champ **Indic** qui peux prendre un ou plusieurs de svaleurs suivantes:
  - U - **Up** - la route est active
  - H - **Host** - la route conduit à un hôte
  - G - **Gateways** - la route passe par une passerelle
- Le champ **Metric** indique le nombre de sauts (passerelles) pour atteindre la destination,
- Le champ **Ref** indique le nombre de références à cette route. Ce champs est usilisé par le Noyau de Linux,
- Le champ **Use** indique le nombre de recherches associés à cette route.

La commande **route** permet aussi de paramétrer le routage indirect. Par exemple pour supprimer la route vers le réseau 192.168.1.0 :

```
opensuse:~ # route del -net 192.168.1.0 netmask 255.255.255.0
opensuse:~ # route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.2.0        *               255.255.255.0   U     0      0        0 eth0
link-local      *               255.255.0.0     U     0      0        0 eth0
loopback        *               255.0.0.0       U     0      0        0 lo
default         10.0.2.2        0.0.0.0         UG    0      0        0 eth0
```

Pour ajouter la route vers le réseau 192.168.1.0 :

```
opensuse:~ # route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.2
opensuse:~ # route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     192.168.1.2     255.255.255.0   UG    0      0        0 eth0
10.0.2.0        *               255.255.255.0   U     0      0        0 eth0
link-local      *               255.255.0.0     U     0      0        0 eth0
loopback        *               255.0.0.0       U     0      0        0 lo
default         10.0.2.2        0.0.0.0         UG    0      0        0 eth0
```

<note important> La commande utilisée pour ajouter une passerelle par défaut prend la forme suivante **route add default gw *numéro_ip***

interface. </note>

Options de la commande route

Les options cette commande sont :

```
opensuse:~ # route --help
Usage: route [-nNvee] [-FC] [<AF>]           List kernel routing tables
       route [-v] [-FC] {add|del|flush} ...  Modify routing table for AF.

       route {-h|--help} [<AF>]              Detailed usage syntax for specified AF.
       route {-V|--version}                  Display version/author and exit.

        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        -e, --extend             display other/more information
        -F, --fib                display Forwarding Information Base (default)
        -C, --cache              display routing cache instead of FIB

  <AF>=Use '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
```

La commande netstat

Pour afficher la table de routage de la machine vous pouvez aussi utiliser la commande **netstat** avec les options **-nr** :

```
opensuse:~ # netstat -nr
Kernel IP routing table
```

```
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.1.0     192.168.1.2     255.255.255.0   UG        0 0          0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         10.0.2.2        0.0.0.0         UG        0 0          0 eth0
```

La table issue de la commande **netstat -nr** indique les informations suivantes:

- La champ **MSS** indique la taille maximale des segments TCP sur la route,
- Le champ **Window** indique la taille de la fenêtre sur cette route,
- Le champ **irrt** indique le paramètre IRRT pour la route.

La commande traceroute

La commande ping est à la base de la commande **traceroute**. Cette commande sert à découvrir la route empruntée pour accéder à un site donné :

```
opensuse:~ # traceroute www.linuxelearning.com
traceroute to www.linuxelearning.com (212.198.31.61), 30 hops max, 40 byte packets using UDP
 1  * * *
 2  172.18.199.129 (172.18.199.129)  1.833 ms  1.808 ms  1.777 ms
 3  80.10.46.241 (80.10.46.241)  29.394 ms  29.234 ms  29.498 ms
 4  10.163.103.199 (10.163.103.199)  30.499 ms  30.702 ms  31.119 ms
 5  212-198-31-61.rev.numericable.fr (212.198.31.61)  39.672 ms * *
```

Options de la commande traceroute

Les options de cette commande sont :

```
opensuse:~ # traceroute --help
traceroute: invalid option -- '-'
usage: traceroute [-nFV] [-f first_ttl] [-m max_hops] [-p port]
```

```
           [-S source_addr] [-I interface] [-g gateway]
           [-t tos] [-w timeout] [-q nqueries] host [packetlen]
```

Activer/désactiver le routage sur le serveur

Pour activer le routage sur le serveur, il convient d'activer la retransmission des paquets:

```
opensuse:~ # echo 1 > /proc/sys/net/ipv4/ip_forward
opensuse:~ # cat /proc/sys/net/ipv4/ip_forward
1
```

Pour désactiver le routage sur le serveur, il convient de désactiver la retransmission des paquets:

```
opensuse:~ # echo 0 > /proc/sys/net/ipv4/ip_forward
opensuse:~ # cat /proc/sys/net/ipv4/ip_forward
0
```

Connexions à Distance

Telnet

La commande **telnet** est utilisée pour établir une connexion à distance avec un serveur telnet :

```
  # telnet numero_ip
```

<note important> Le service telnet revient à une redirection des canaux standards d'entrée et de sortie. Notez que la connexion n'est **pas** sécurisée. Pour fermer la connexion, il faut saisir la commande **exit**. La commande telnet n'offre pas de services de transfert de fichiers. Pour cela, il convient d'utiliser la command **ftp**. </note>

Options de la commande telnet

Les options de cette commande sont :

```
opensuse:~ # telnet --help
telnet: invalid option -- '-'
Usage: telnet [-8] [-E] [-L] [-S tos] [-a] [-c] [-d] [-e char] [-l user]
    [-n tracefile] [-b hostalias ] [-r]
    [host-name [port]]
```

ssh

Sous openSUSE, le serveur **sshd** n'est pas démarré par défaut :

```
opensuse:~ # service sshd status
Checking for service sshd
unused
```

Démarrez donc ce dernier :

```
opensuse:~ # service sshd start
Generating /etc/ssh/ssh_host_key.
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
80:19:02:26:7e:7e:fe:91:5f:7e:bd:1f:fe:8d:33:9b root@opensuse.fenestros.loc
The key's randomart image is:
+--[RSA1 1024]----+
|oo. .            |
|+  . +           |
| . .o .          |
```

```
|  o    .         |
|   . .  S        |
|    o   .        |
|     . o   .  .. |
|      . o o  .++o|
|       . . .. E*B|
+-----------------+
Generating /etc/ssh/ssh_host_dsa_key.
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
aa:bd:c8:9f:58:e9:5b:e1:2c:66:34:2e:6d:c1:2a:c9 root@opensuse.fenestros.loc
The key's randomart image is:
+--[ DSA 1024]----+
|                 |
|                 |
|                 |
|     .           |
|      = S        |
| . . = B .       |
|  E o @ +        |
|   o % +         |
|    =.Bo         |
+-----------------+
Generating /etc/ssh/ssh_host_rsa_key.
Generating public/private rsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
53:bd:7f:80:79:df:74:97:d4:0a:04:b2:1c:6e:bf:c7 root@opensuse.fenestros.loc
The key's randomart image is:
+--[ RSA 1024]----+
|        o ...    |
```

```
|       o + o    .|
|        = . o  ..|
|       . o   =...|
|        S . + +.+|
|         . o o ++|
|          . E . +|
|           .   . |
|                 |
+-----------------+
Generating /etc/ssh/ssh_host_ecdsa_key.
Generating public/private ecdsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_ecdsa_key.
Your public key has been saved in /etc/ssh/ssh_host_ecdsa_key.pub.
The key fingerprint is:
22:db:d5:60:b4:3a:55:df:74:c4:24:0b:a8:49:21:53 root@opensuse.fenestros.loc
The key's randomart image is:
+--[ECDSA  256]---+
|      o.E.... o++|
|       +.o.. + +.|
|       .=o  . o  |
|       +oo       |
|    . + S .      |
|     + +         |
|    . .          |
|                 |
|                 |
+-----------------+
Starting SSH daemon
```

La commande **ssh** permet d'établir des connexions sécurisées avec une machine distante :

```
opensuse:~ # ssh -l trainee localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is 22:db:d5:60:b4:3a:55:df:74:c4:24:0b:a8:49:21:53.
```

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Password:
Last login: Mon Apr  9 15:46:08 2012 from console
Have a lot of fun...
trainee@opensuse:~> pwd
/home/trainee
trainee@opensuse:~> whoami
trainee
```

<note important> Notez que dans cet exemple vous vous connectez au serveur ssh sur votre propre machine virtuelle en tant que l'utilisateur **trainee**. </note>

Pour fermer la connexion, utilisez la commande **exit** :

```
trainee@opensuse:~> exit
logout
Connection to localhost closed.
opensuse:~ #
```

Options de la commande ssh

Les options de cette commande sont :

```
opensuse:~ # ssh --help
usage: ssh [-1246AaCfgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-e escape_char] [-F configfile]
           [-I pkcs11] [-i identity_file]
           [-L [bind_address:]port:host:hostport]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-R [bind_address:]port:host:hostport] [-S ctl_path]
           [-W host:port] [-w local_tun[:remote_tun]]
```

```
            [user@]hostname [command]
```

wget

La commande **wget** est utilisée pour récupérer un fichier via http ou ftp :

```
opensuse:~ # wget ftp://ftp2.fenestros.com/fenestros/files/fichier_test
asking libproxy about url 'ftp://ftp2.fenestros.com/fenestros/files/fichier_test'
libproxy suggest to use 'direct://'
--2012-05-16 08:57:00--  ftp://ftp2.fenestros.com/fenestros/files/fichier_test
           => `fichier_test'
Resolving ftp2.fenestros.com... 213.186.33.14
Connecting to ftp2.fenestros.com|213.186.33.14|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /fenestros/files ... done.
==> SIZE fichier_test ... 57
==> PASV ... done.    ==> RETR fichier_test ... done.
Length: 57 (unauthoritative)

100%[======================================================================================>] 57          --.-K/s   in 0s

2012-05-16 08:57:01 (1.20 MB/s) - `fichier_test' saved [57]
```

Options de la commande wget

Les options de cette commande sont :

```
opensuse:~ # wget --help
GNU Wget 1.12, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...
```

```
Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version           display the version of Wget and exit.
  -h,  --help              print this help.
  -b,  --background        go to background after startup.
  -e,  --execute=COMMAND   execute a `.wgetrc'-style command.

Logging and input file:
  -o,  --output-file=FILE    log messages to FILE.
  -a,  --append-output=FILE  append messages to FILE.
  -d,  --debug               print lots of debugging information.
  -q,  --quiet               quiet (no output).
  -v,  --verbose             be verbose (this is the default).
  -nv, --no-verbose          turn off verboseness, without being quiet.
  -i,  --input-file=FILE     download URLs found in local or external FILE.
  -F,  --force-html          treat input file as HTML.
  -B,  --base=URL            resolves HTML input-file links (-i -F)
                             relative to URL.

Download:
  -t,  --tries=NUMBER            set number of retries to NUMBER (0 unlimits).
       --retry-connrefused       retry even if connection is refused.
  -O,  --output-document=FILE    write documents to FILE.
  -nc, --no-clobber              skip downloads that would download to
                                 existing files.
  -c,  --continue                resume getting a partially-downloaded file.
       --progress=TYPE           select progress gauge type.
  -N,  --timestamping            don't re-retrieve files unless newer than
                                 local.
  -S,  --server-response         print server response.
       --spider                  don't download anything.
  -T,  --timeout=SECONDS         set all timeout values to SECONDS.
       --dns-timeout=SECS        set the DNS lookup timeout to SECS.
```

```
       --connect-timeout=SECS    set the connect timeout to SECS.
       --read-timeout=SECS       set the read timeout to SECS.
  -w,  --wait=SECONDS            wait SECONDS between retrievals.
       --waitretry=SECONDS       wait 1..SECONDS between retries of a retrieval.
       --random-wait             wait from 0...2*WAIT secs between retrievals.
       --no-proxy                explicitly turn off proxy.
  -Q,  --quota=NUMBER            set retrieval quota to NUMBER.
       --bind-address=ADDRESS    bind to ADDRESS (hostname or IP) on local host.
       --limit-rate=RATE         limit download rate to RATE.
       --no-dns-cache            disable caching DNS lookups.
       --restrict-file-names=OS  restrict chars in file names to ones OS allows.
       --ignore-case             ignore case when matching files/directories.
  -4,  --inet4-only              connect only to IPv4 addresses.
  -6,  --inet6-only              connect only to IPv6 addresses.
       --prefer-family=FAMILY    connect first to addresses of specified family,
                                 one of IPv6, IPv4, or none.
       --user=USER               set both ftp and http user to USER.
       --password=PASS           set both ftp and http password to PASS.
       --ask-password            prompt for passwords.
       --no-iri                  turn off IRI support.
       --local-encoding=ENC      use ENC as the local encoding for IRIs.
       --remote-encoding=ENC     use ENC as the default remote encoding.

Directories:
  -nd, --no-directories           don't create directories.
  -x,  --force-directories        force creation of directories.
  -nH, --no-host-directories      don't create host directories.
       --protocol-directories     use protocol name in directories.
  -P,  --directory-prefix=PREFIX  save files to PREFIX/...
       --cut-dirs=NUMBER          ignore NUMBER remote directory components.

HTTP options:
       --http-user=USER        set http user to USER.
       --http-password=PASS    set http password to PASS.
```

```
       --no-cache              disallow server-cached data.
       --default-page=NAME     Change the default page name (normally
                               this is `index.html'.).
  -E,  --adjust-extension      save HTML/CSS documents with proper extensions.
       --ignore-length         ignore `Content-Length' header field.
       --header=STRING         insert STRING among the headers.
       --max-redirect          maximum redirections allowed per page.
       --proxy-user=USER       set USER as proxy username.
       --proxy-password=PASS   set PASS as proxy password.
       --referer=URL           include `Referer: URL' header in HTTP request.
       --save-headers          save the HTTP headers to file.
  -U,  --user-agent=AGENT      identify as AGENT instead of Wget/VERSION.
       --no-http-keep-alive    disable HTTP keep-alive (persistent connections).
       --no-cookies            don't use cookies.
       --load-cookies=FILE     load cookies from FILE before session.
       --save-cookies=FILE     save cookies to FILE after session.
       --keep-session-cookies  load and save session (non-permanent) cookies.
       --post-data=STRING      use the POST method; send STRING as the data.
       --post-file=FILE        use the POST method; send contents of FILE.
       --content-disposition   honor the Content-Disposition header when
                               choosing local file names (EXPERIMENTAL).
       --auth-no-challenge     send Basic HTTP authentication information
                               without first waiting for the server's
                               challenge.

HTTPS (SSL/TLS) options:
       --secure-protocol=PR     choose secure protocol, one of auto, SSLv2,
                                SSLv3, and TLSv1.
       --no-check-certificate   don't validate the server's certificate.
       --certificate=FILE       client certificate file.
       --certificate-type=TYPE  client certificate type, PEM or DER.
       --private-key=FILE       private key file.
       --private-key-type=TYPE  private key type, PEM or DER.
       --ca-certificate=FILE    file with the bundle of CA's.
```

```
       --ca-directory=DIR       directory where hash list of CA's is stored.
       --random-file=FILE       file with random data for seeding the SSL PRNG.
       --egd-file=FILE          file naming the EGD socket with random data.

FTP options:
       --ftp-user=USER         set ftp user to USER.
       --ftp-password=PASS     set ftp password to PASS.
       --no-remove-listing     don't remove `.listing' files.
       --no-glob               turn off FTP file name globbing.
       --no-passive-ftp        disable the "passive" transfer mode.
       --retr-symlinks         when recursing, get linked-to files (not dir).

Recursive download:
  -r,  --recursive          specify recursive download.
  -l,  --level=NUMBER       maximum recursion depth (inf or 0 for infinite).
       --delete-after       delete files locally after downloading them.
  -k,  --convert-links      make links in downloaded HTML or CSS point to
                            local files.
  -K,  --backup-converted   before converting file X, back up as X.orig.
  -m,  --mirror             shortcut for -N -r -l inf --no-remove-listing.
  -p,  --page-requisites    get all images, etc. needed to display HTML page.
       --strict-comments    turn on strict (SGML) handling of HTML comments.

Recursive accept/reject:
  -A,  --accept=LIST               comma-separated list of accepted extensions.
  -R,  --reject=LIST               comma-separated list of rejected extensions.
  -D,  --domains=LIST              comma-separated list of accepted domains.
       --exclude-domains=LIST      comma-separated list of rejected domains.
       --follow-ftp                follow FTP links from HTML documents.
       --follow-tags=LIST          comma-separated list of followed HTML tags.
       --ignore-tags=LIST          comma-separated list of ignored HTML tags.
  -H,  --span-hosts                go to foreign hosts when recursive.
  -L,  --relative                  follow relative links only.
  -I,  --include-directories=LIST  list of allowed directories.
```

```
  -X,  --exclude-directories=LIST  list of excluded directories.
  -np, --no-parent                 don't ascend to the parent directory.

Mail bug reports and suggestions to <bug-wget@gnu.org>.
```

ftp

La commande **ftp** est utilisée pour le transfert de fichiers:

```
opensuse:~ # ftp ftp2.fenestros.com
Connected to anonymous.ftp.ovh.net.
220 anonymous.ftp.ovh.net NcFTPd Server (licensed copy) ready.
Name (ftp2.fenestros.com:trainee): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230 Logged in anonymously.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Une fois connecté, il convient d'utiliser la commande **help** pour afficher la liste des commandes disponibles :

```
ftp> help
Commands may be abbreviated.  Commands are:

!       delete        idle        mode        preserve    restart        tenex
$       dir         image        modtime        progress    rhelp        throttle
account     disconnect    lcd         more         prompt         rmdir         trace
append      edit        less        mput         proxy         rstatus         type
ascii       epsv4        lpage         msend         put         runique         umask
bell        exit        lpwd        newer        pwd         send        unset
binary      features    ls        nlist        quit        sendport    usage
bye     fget        macdef        nmap        quote        set        user
```

```
case        form        mdelete        ntrans        rate        site        verbose
cd      ftp        mdir        open        rcvbuf        size        xferbuf
cdup        gate        mget        page        recv        sndbuf        ?
chmod       get        mkdir        passive        reget        status
close       glob        mls        pdir        remopts        struct
cr      hash        mlsd        pls        rename        sunique
debug       help        mlst        pmlsd        reset        system
```

Le caractère **!** permet d'exécuter une commande sur la machine cliente

```
ftp> !pwd
/root
```

Pour transférer un fichier vers le serveur, il convient d'utiliser la commande **put** :

```
ftp> put nom_fichier_local nom_fichier_distant
```

Vous pouvez également transférer plusieurs fichiers à la fois grâce à la commande **mput**. Dans ce cas précis, il convient de saisir la commande suivante:

```
ftp> mput nom*.*
```

Pour transférer un fichier du serveur, il convient d'utiliser la commande **get** :

```
ftp> get nom_fichier
```

Vous pouvez également transférer plusieurs fichiers à la fois grâce à la commande **mget** (voir la commande **mput** ci-dessus).

Pour supprimer un fichier sur le serveur, il convient d'utiliser la commande **del** :

```
ftp> del nom_fichier
```

Pour fermer la session, il convient d'utiliser la commande **quit** :

```
ftp> quit
221 Goodbye.
opensuse:~ #
```

Options de la commande ftp

Les options de cette commande sont :

```
opensuse:~ # ftp --options
ftp: invalid option -- '-'
usage: ftp [-AadefginpRtvV] [-o outfile] [-P port] [-r retry]
           [-T dir,max[,inc][[user@]host [port]]] [host:path[/]]
           [file:///file] [ftp://[user[:pass]@]host[:port]/path[/]]
           [http://[user[:pass]@]host[:port]/path] [...]
       ftp -u url file [...]
```

scp

La commande **scp** est le successeur et la remplaçante de la commande **rcp** de la famille des commandes **remote**. Il permet de faire des transferts sécurisés à partir d'une machine distante :

```
# scp compte@numero_ip(nom_de_machine):/chemin_distant/fichier_distant /chemin_local/fichier_local
```

ou vers une machine distante :

```
# scp /chemin_local/fichier_local compte@numero_ip(nom_de_machine):/chemin_distant/fichier_distant
```

Options de la commande scp

Les options de cette commande sont :

```
opensuse:~ # scp --help
usage: scp [-12346BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
           [-l limit] [-o ssh_option] [-P port] [-S program]
           [[user@]host1:]file1 ... [[user@]host2:]file2
```

~~DISCUSSION:off~~

Donner votre Avis

{(rater>id=opensuse_11_l118|name=cette page|type=rate|trace=user|tracedetails=1)}