Dernière mise-à-jour : 2020/01/30 03:27

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

Cliquez ici pour ouvrir le schéma Simplifié du Modèle OSI incluant la spécification NDIS

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

Cliquez ici pour voir le modèle OSI incluant la suite des protocoles et services TCP/IP

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 ...

Cliquez ici pour voir les modèles TCP/IP et OSI

Le modèle TCP/IP est composé de 4 couches :

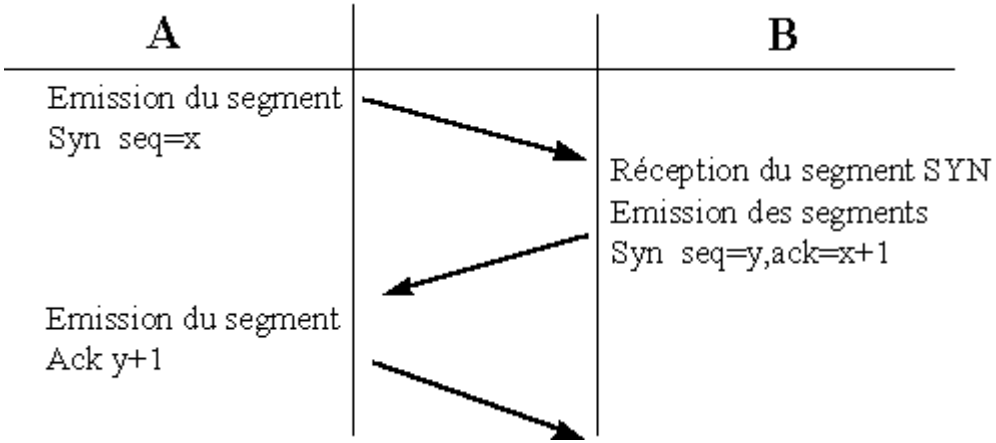
- La couche d'Accès Réseau
  - Cette couche spécifie la forme sous laquelle les données doivent être acheminées, quelque soit le type de réseau utilisé.
- La couche Internet
  - Cette couche est chargée de fournir le paquet de données.
- La couche de Transport
  - Cette couche assure l'acheminement des 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 :

<table>
<tr><th>1er octet</th><th>2ème octet</th><th>3ème octet</th><th>4 ème octet</th></tr>
<tr><td colspan="2">Port source</td><td colspan="2">Port destination</td></tr>
<tr><td colspan="4">Numéro de séquence</td></tr>
<tr><td colspan="4">Numéro d'acquittement</td></tr>
<tr><td>Offset</td><td>Flags</td><td colspan="2">Fenêtre</td></tr>
<tr><td colspan="2">Checksum</td><td colspan="2">Pointeur Urgent</td></tr>
<tr><td colspan="3">Options</td><td>Padding</td></tr>
<tr><td colspan="4">Données</td></tr>
</table>

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 x 1 = 1
- 0 x 1 = 0
- 1 x 0 = 0
- 0 x 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 Client Réseau

/etc/services

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

```
root@ubuntu:~# 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, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux      1/tcp               # TCP port service multiplexer
```

```
echo        7/tcp
echo        7/udp
discard     9/tcp        sink null
discard     9/udp        sink null
systat      11/tcp        users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
qotd        17/tcp        quote
msp     18/tcp               # message send protocol
msp     18/udp
chargen     19/tcp        ttytst source
chargen     19/udp        ttytst source
ftp-data    20/tcp
ftp     21/tcp
fsp     21/udp        fspd
ssh     22/tcp               # SSH Remote Login Protocol
ssh     22/udp
telnet      23/tcp
smtp        25/tcp        mail
time        37/tcp        timserver
--Plus--(5%)
```

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 :

```
root@ubuntu:~# netstat -an | more
Connexions Internet actives (serveurs et établies)
```

```
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp       28      0 10.0.2.15:54471         91.189.92.10:443        CLOSE_WAIT
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       1      0 ::1:54154               ::1:631                 CLOSE_WAIT
udp        0      0 0.0.0.0:23784           0.0.0.0:*
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp        0      0 0.0.0.0:53047           0.0.0.0:*
udp        0      0 127.0.1.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:68              0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp6       0      0 :::5353                 :::*
udp6       0      0 :::59655                :::*
udp6       0      0 :::34953                :::*
--Plus--
```

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

```
root@ubuntu:~# netstat -anp | more
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat       PID/Program name
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN      1076/dnsmasq
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      4826/cupsd
tcp       28      0 10.0.2.15:54471         91.189.92.10:443        CLOSE_WAIT  2201/unity-scope-ho
tcp6       0      0 ::1:631                 :::*                    LISTEN      4826/cupsd
tcp6       1      0 ::1:54154               ::1:631                 CLOSE_WAIT  921/cups-browsed
udp        0      0 0.0.0.0:23784           0.0.0.0:*                           4317/dhclient
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           691/avahi-daemon: r
udp        0      0 0.0.0.0:53047           0.0.0.0:*                           691/avahi-daemon: r
udp        0      0 127.0.1.1:53            0.0.0.0:*                           1076/dnsmasq
udp        0      0 0.0.0.0:68              0.0.0.0:*                           4317/dhclient
udp        0      0 0.0.0.0:631             0.0.0.0:*                           921/cups-browsed
udp6       0      0 :::5353                 :::*                                691/avahi-daemon: r
```

```
udp6       0      0 :::59655                :::*                                4317/dhclient
udp6       0      0 :::34953                :::*                                691/avahi-daemon: r
--Plus--
```

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 :

```
root@ubuntu:~# arp -a
? (10.0.2.2) à 52:54:00:12:35:02 [ether] sur eth0
? (10.0.2.3) à 52:54:00:12:35:03 [ether] sur eth0
```

Options de la commande

Les options de cette commande sont :

```
root@ubuntu:~# arp --help
Syntaxe:
  arp [-vn]  [<MAT>] [-i <if>] [-a] [<hôte>]             <-Affiche cache ARP
  arp [-v]          [-i <if>] -d  <host> [pub]               <-Delete ARP entry
  arp [-vnD] [<MAT>] [-i <if>] -f [<fichier>] <-Ajouter une entrée depuis un fichier
  arp [-v]   [<HW>] [-i <if>] -s  <host> <hwaddr> [temp]            <-Add entry
  arp [-v]   [<HW>] [-i <if>] -Ds <host> <if> [netmask <nm>] pub          <-''-

        -a                      affiche (tous) les hôtes en style BSD
        -s, --set               définit une nouvelle entrée ARP
        -d, --delete            supprime une entrée
```

```
       -v, --verbose            mode verbeux
       -n, --numeric ne résoud pas les noms
       -i, --device             spécifie l'interface réseau (p.ex. eth0)
       -D, --use-device         lit l'<adrmat> depuis le périphérique
       -A, -p, --protocol spécifie la famille de protocoles
       -f, --file lit les nouvelles entrées à partir d'un fichier ou de /etc/ethers

  <HW>=Utilisez '-H <hw>' pour spécifier le type d'adresse matériel. Défaut: ether
  Liste les types de matériels supportant ARP:
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
    dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) x25 (X.25 générique) eui64 (EUI-64 Générique)
```

Configuration de TCP/IP

La configuration TCP/IP se trouve dans le fichier **/etc/network/interfaces** :

/etc/network/interfaces

DHCP

```
root@ubuntu:~# cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
```

Dans ce fichier chaque déclaration est de la forme suivante :

```
interface   nom    type    mode
```

On peut constater donc dans notre exemple ci-dessus une déclaration pour l'interface **lo** de loopback.

IP Fixe

Dans le cas où l'interface eth0 était configuré en IP statique, la déclaration concernant eth0 prendrait la forme suivante :

```
auto eth0
iface eth0 inet static
    dns-nameservers 8.8.8.8 8.8.4.4
    dns-search fenestros.loc
        address 10.0.2.15
        netmask 255.255.255.0
        broadcast 10.0.2.255
        network 10.0.2.0
        gateway 10.0.2.2
```

Dans ce fichier vous pouvez constater les directives suivantes :

Directive	Description
dns-nameservers	Indique les adresses des serveurs DNS
dns-search	Indique le nom de notre domaine
address	Indique l'adresse IPv4 de l'interface
netmask	Indique le masque de sous-réseau IPv4
broadcast	Indique l'adresse de diffusion IPv4
network	Indique l'adresse réseau IPv4
gateway	Indique l'adresse IPv4 de la passerelle par défaut

Notez que VirtualBox fournit une passerelle par défaut (10.0.2.2).

Après avoir modifier le fichier **/etc/network/interfaces** vous devez arrêter le service **network-manager** utilisé pour la connexion DHCP puis le

désactiver. Commencez par arrêter le service network-manager :

```
root@ubuntu:~# service network-manager stop
network-manager stop/waiting
```

Editez ensuite le fichier **/etc/init/network-manager.conf** ainsi :

/etc/init/network-manager

```
# network-manager - network connection manager
#
# The Network Manager daemon manages the system's network connections,
# automatically switching between the best available.

description "network connection manager"

#start on (local-filesystems
#     and started dbus
#     and static-network-up)
start on runlevel [!0123456]
stop on stopping dbus

expect fork
respawn

script
    # set $LANG so that messages appearing on the GUI will be translated. See LP: 875017
    if [ -r /etc/default/locale ]; then
        . /etc/default/locale
        export LANG LANGUAGE LC_MESSAGES LC_ALL
    fi

    exec NetworkManager
```

```
end script
```

Redémarrez votre machine virtuelle puis vérifier si network-manager est bien arrêté :

```
root@ubuntu:~# initctl list | grep network
network-manager stop/waiting
network-interface (lo) start/running
network-interface (eth0) start/running
network-interface-security (network-interface/eth0) start/running
network-interface-security (network-interface/lo) start/running
network-interface-security (networking) start/running
networking start/running
network-interface-container stop/waiting
```

Vérifez ensuite la configuration IP :

```
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:79:00:d7
          inet adr:10.0.2.15  Bcast:10.0.2.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe79:d7/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:4 erreurs:0 :0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:393 (393.0 B) Octets transmis:8176 (8.1 KB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          Packets reçus:350 erreurs:0 :0 overruns:0 frame:0
          TX packets:350 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
```

```
          Octets reçus:27415 (27.4 KB) Octets transmis:27415 (27.4 KB)
```

Si le service networking réfuse de démarrer en produisant une erreur, le problème vient certainement du fait que votre interface réseau a été configurée par **udev** en **eth1**. La solution la plus simple est d'éditer le fichier **/etc/udev/rules.d/70-persistent-net.rules** en supprimant toutes les lignes qui ne commencent pas par le caractère **#** et de re-démarrer votre machine virtuelle.

/etc/networks

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

```
root@debian:~# cat /etc/networks
root@ubuntu:~# cat /etc/networks
# symbolic names for networks, see networks(5) for more information
link-local 169.254.0.0
```

/etc/resolv.conf

La configuration DNS est stockée dans le fichier **/etc/resolv.conf** :

```
root@ubuntu:~# cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 8.8.8.8
nameserver 8.8.4.4
search fenestros.loc
```

Notez que les DNS utilisés sont les serveurs DNS publics de Google.

/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 :

```
root@ubuntu:~# 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** :

```
root@ubuntu:~# cat /etc/hosts
127.0.0.1   localhost
127.0.1.1   ubuntu

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Testez votre Configuration DNS

Utilisez les commandes **nslookup** et **dig** :

```
root@ubuntu:~# nslookup www.linuxelearning.com
Server:     8.8.4.4
Address:    8.8.4.4#53

Non-authoritative answer:
www.linuxelearning.com  canonical name = linuxelearning.com.
Name:   linuxelearning.com
Address: 88.173.201.50

root@ubuntu:~# dig www.linuxelearning.com

; <<>> DiG 9.9.5-3-Ubuntu <<>> www.linuxelearning.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27057
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.linuxelearning.com.        IN    A

;; ANSWER SECTION:
www.linuxelearning.com. 21599    IN    CNAME    linuxelearning.com.
linuxelearning.com. 59    IN    A    88.173.201.50

;; Query time: 259 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Oct 31 15:39:21 CET 2014
;; MSG SIZE  rcvd: 81
```

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 Debian xinetd n'est pas installé par défaut. Installez-le grâce à apt-get :

```
root@ubuntu:~# apt-get install xinetd
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  xinetd
0 mis à jour, 1 nouvellement installés, 0 à enlever et 222 non mis à jour.
Il est nécessaire de prendre 102 ko dans les archives.
Après cette opération, 317 ko d'espace disque supplémentaires seront utilisés.
Réception de : 1 http://fr.archive.ubuntu.com/ubuntu/ trusty/main xinetd i386 1:2.3.15-3ubuntu1 [102 kB]
102 ko réceptionnés en 7s (13,8 ko/s)
Sélection du paquet xinetd précédemment désélectionné.
(Lecture de la base de données... 167686 fichiers et répertoires déjà installés.)
Préparation du décompactage de .../xinetd_1%3a2.3.15-3ubuntu1_i386.deb ...
Décompactage de xinetd (1:2.3.15-3ubuntu1) ...
Traitement déclenché pour  man-db (2.6.7.1-1) ...
Traitement déclenché pour  doc-base (0.10.5) ...
Traitement en cours 1 added doc-base file...
Traitement déclenché pour  ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on next reboot
Paramétrage de xinetd (1:2.3.15-3ubuntu1) ...
```

```
xinetd start/running, process 2706
Traitement déclenché pour  ureadahead (0.100.0-16) ...
```

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

```
root@ubuntu:~# cat /etc/xinetd.conf
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{

# Please note that you need a log_type line to be able to use log_on_success
# and log_on_failure. The default is the following :
# log_type = SYSLOG daemon info

}

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** :

```
root@ubuntu:~# ls -l /etc/xinetd.d
total 20
-rw-r--r-- 1 root root 640 oct.  26  2013 chargen
-rw-r--r-- 1 root root 502 oct.  26  2013 daytime
-rw-r--r-- 1 root root 391 oct.  26  2013 discard
-rw-r--r-- 1 root root 422 oct.  26  2013 echo
-rw-r--r-- 1 root root 569 oct.  26  2013 time
```

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

```
root@ubuntu:~# cat /etc/xinetd.d/chargen
# default: off
# description: An xinetd internal service which generate characters.  The
# xinetd internal service which continuously generates characters until the
# connection is dropped.  The characters look something like this:
# !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg
# This is the tcp version.
service chargen
{
    disable        = yes
    type        = INTERNAL
    id        = chargen-stream
    socket_type    = stream
    protocol    = tcp
    user        = root
    wait        = no
}

# This is the udp version.
service chargen
{
    disable        = yes
    type        = INTERNAL
    id        = chargen-dgram
    socket_type    = dgram
    protocol    = udp
    user        = root
    wait        = yes
}
```

Les directives principales de ce fichier sont :

Paramètre	Déscription
disable	**no** : Le service est actif. **yes** : Le service est désactivé

Paramètre	Déscription
type	Indique le type de service. Dans ce cas chargen est un service interne de xinetd
id	Indique nom de référence pour le service
socket_type	Nature du socket, soit **stream** pour TCP soit **dgram** pour UDP
protocol	Protocole utilisé soit **TCP** soit **UDP**
user	Indique le compte sous lequel le serveur est exécuté
wait	**no** : indique si xinetd active un serveur par client. **yes** : indique que xinetd active un seul serveur pour tous les client

Dans le cas d'une application-server telle proftpd, on trouve aussi les directives suivantes :

Paramètre	Déscription
port	Le numéro de port ou, à défaut, le numéro indiqué pour le service dans le fichier /etc/services
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 un service interne à xinetd ou 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

Le nom de la machine se trouve dans le fichier **/etc/hostname** :

```
root@ubuntu:~# cat /etc/hostname
ubuntu
```

Ce nom doit être un FQDN (*Fully Qualified Domain Name*). Modifiez donc ce fichier ainsi :

```
root@ubuntu:~# cat /etc/hostname
ubuntu.fenestros.loc
```

Afin d'informer le système immédiatement de la modification du FQDN, utilisez la commande **hostname** :

```
root@ubuntu:~# hostname
ubuntu
root@ubuntu:~# hostname ubuntu.fenestros.loc
```

```
root@ubuntu:~# hostname
ubuntu.fenestros.loc
```

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

```
root@ubuntu:~# uname -n
ubuntu.fenestros.loc
```

Options de la commande hostname

Les options de cette commande sont :

```
root@ubuntu:~# hostname --help
Usage: hostname [-b] {hostname|-F file}         set host name (from file)
       hostname [-a|-A|-d|-f|-i|-I|-s|-y]       display formatted name
       hostname                                 display host name

       {yp,nis,}domainname {nisdomain|-F file}  set NIS domain name (from file)
       {yp,nis,}domainname                      display NIS domain name

       dnsdomainname                            display dns domain name

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

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

Program options:
    -a, --alias            alias names
    -A, --all-fqdns        all long host names (FQDNs)
    -b, --boot             set default hostname if none available
```

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

Description:
   This command can get or set the host name or the NIS domain name. You can
   also get 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 :

```
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:79:00:d7
          inet adr:10.0.2.15  Bcast:10.0.2.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe79:d7/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:176 erreurs:0 :0 overruns:0 frame:0
          TX packets:157 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:130852 (130.8 KB) Octets transmis:16102 (16.1 KB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
```

```
          Packets reçus:145 erreurs:0 :0 overruns:0 frame:0
          TX packets:145 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:10223 (10.2 KB) Octets transmis:10223 (10.2 KB)
```

La commande ifconfig est également utilisée pour configurer une interface.

Créez maintenant une interface fictive ainsi :

```
root@ubuntu:~# ifconfig eth0:0 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
```

Constatez maintenant le résultat :

```
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:79:00:d7
          inet adr:10.0.2.15  Bcast:10.0.2.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe79:d7/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:176 erreurs:0 :0 overruns:0 frame:0
          TX packets:167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:130852 (130.8 KB) Octets transmis:17308 (17.3 KB)

eth0:0    Link encap:Ethernet  HWaddr 08:00:27:79:00:d7
          inet adr:192.168.1.2  Bcast:192.168.1.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          Packets reçus:145 erreurs:0 :0 overruns:0 frame:0
          TX packets:145 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
```

```
          Octets reçus:10223 (10.2 KB) Octets transmis:10223 (10.2 KB)
```

Options de la commande ifconfig

Les options de cette commande sont :

```
root@ubuntu:~# ifconfig --help
Utilisation :
  ifconfig [-a] [-v] [-s] <interface> [[<AF>] <addresse>]
  [add <adresse>[/<lg_prefixe>]]
  [del <adresse>[/<lg_prefixe>]]
  [[-]broadcast [<adresse>]]  [[-]pointopoint [<adresse>]]
  [Masque réseau <adresse>]  [Adresse distante <adresse>]  [tunnel <adresse>]
  [outfill <NN>] [keepalive <NN>]
  [hw <HW> <adresse>]  [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>=Type de matériel.
  Liste des types de matériels possibles:
    loop (Boucle locale) slip (IP ligne série) cslip (IP ligne série - VJ)
    slip6 (IP ligne série - 6 bits) cslip6 (IP ligne série - 6 bits VJ) adaptive (IP ligne série adaptative)
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) tunnel (IPIP Tunnel)
    ppp (Protocole Point-à-Point) hdlc ((Cisco)-HDLC) lapb (LAPB)
    arcnet (ARCnet) dlci (Frame Relay DLCI) frad (Périphériue d'accès Frame Relay)
    sit (IPv6-dans-IPv4) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) ec (Econet) x25 (X.25 générique)
    eui64 (EUI-64 Générique)
```

```
  <AF>=famille d'Adresses. Défaut: inet
  Liste des familles d'adresses possibles:
    unix (Domaine UNIX) inet (DARPA Internet) inet6 (IPv6)
    ax25 (AMPR AX.25) netrom (AMPR NET/ROM) rose (AMPR ROSE)
    ipx (Novell IPX) ddp (Appletalk DDP) ec (Econet)
    ash (Ash) x25 (CCITT X.25)
```

ping

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

```
root@ubuntu:~# 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_seq=1 ttl=63 time=0.319 ms
64 bytes from 10.0.2.2: icmp_seq=2 ttl=63 time=0.385 ms
64 bytes from 10.0.2.2: icmp_seq=3 ttl=63 time=0.379 ms
^C
--- 10.0.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.319/0.361/0.385/0.029 ms
```

Options de la commande ping

Les options de cette commande sont :

```
root@ubuntu:~# ping --help
ping: invalid option -- '-'
Usage: ping [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
            [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
            [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
            [-w deadline] [-W timeout] [hop1 ...] destination
```

netstat -i

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

```
root@ubuntu:~# netstat -i
Table d'interfaces noyau
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500 0       186      0      0 0           179      0      0      0 BMRU
eth0:0     1500 0       - pas de statistiques disponible -                BMRU
lo        65536 0       145      0      0 0           145      0      0      0 LRU
```

Options de la commande netstat

Les options de cette commande sont :

```
root@ubuntu:~# netstat --help
usage: netstat [-vWeenNcCF] [<Af>] -r         netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -i | [-cWnNe] -M | -s }

        -r, --route              affiche la table de routage
        -i, --interfaces         affiche la table d'interfaces
        -g, --groups             affiche les membres d'un groupe multicast
        -s, --statistics         affiche les statistiques réseau (comme SNMP)
        -M, --masquerade         affiche les connexions masquées

        -v, --verbose            mode verbeux
        -W, --wide               don't truncate IP addresses
        -n, --numeric ne résoud pas les noms
        --numeric-hosts ne résoud pas les noms d'hôte\n
        --numeric-ports ne résoud pas les noms de ports
        --numeric-users ne résoud pas les noms utilisateur
```

```
        -N, --symbolic           résoud les noms matériels
        -e, --extend affiche d'autres/plus d'informations
        -p, --programs           affiche le nom du programme/PID des sockets
        -c, --continuous         listing continu

        -l, --listening          affiche les sockets du serveur à l'écoute
        -a, --all, --listening   affiche toutes les prises (défaut: connectés)
        -o, --timers             affiche les timers
        -F, --fib                affiche la base d'information des redirection (Forwarding Information Base)
(défaut)
        -C, --cache              affiche le cache de routage au lieu de FIB

  <Socket>={-t|--tcp} {-u|--udp} {-w|--raw} {-x|--unix} --ax25 --ipx --netrom
  <AF>=Utilisez '-6|-4' ou '-A <af>' ou '--<af>'; défaut: inet
  Liste les familles d'adresses possibles (supportant le routage):
    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** :

```
root@ubuntu:~# route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         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     1000   0        0 eth0
```

```
192.168.1.0     *               255.255.255.0   U     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 :

```
root@ubuntu:~# route del -net 192.168.1.0 netmask 255.255.255.0
root@ubuntu:~# route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         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     1000   0        0 eth0
```

Pour ajouter la route vers le réseau 192.168.1.0 :

```
root@ubuntu:~# route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.2
root@ubuntu:~# route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         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     1000   0        0 eth0
```

```
192.168.1.0     192.168.1.2     255.255.255.0   UG    0      0        0 eth0
```

La commande utilisée pour ajouter une passerelle par défaut prend la forme suivante **route add default gw *numéro_ip interface***.

Options de la commande route

Les options cette commande sont :

```
root@ubuntu:~# route --help
Syntaxe: route [-nNvee] [-FC] [<AF>]           Liste les tables de routage noyau
       route [-v] [-FC] {add|del|flush} ...  Modifie la table de routage pour AF.

       route {-h|--help} [<AF>]              Utilisation détaillée pour l'AF spécifié.
       route {-V|--version}                  Affiche la version/auteur et termine.

        -v, --verbose            mode verbeux
        -n, --numeric ne résoud pas les noms
        -e, --extend affiche d'autres/plus d'informations
        -F, --fib                affiche la base d'information des redirection (Forwarding Information Base)
(défaut)
        -C, --cache              affiche le cache de routage au lieu de FIB

  <AF>=Utilisez '-A <af>' ou '--<af>'; défaut: inet
  Liste les familles d'adresses possibles (supportant le routage):
    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** :

```
root@ubuntu:~# netstat -nr
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic   MSS Fenêtre irtt Iface
0.0.0.0         10.0.2.2        0.0.0.0         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
192.168.1.0     192.168.1.2     255.255.255.0   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. Elle n'est pas installée par défaut sous Ubuntu :

```
root@ubuntu:~# apt-get install traceroute
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  traceroute
0 mis à jour, 1 nouvellement installés, 0 à enlever et 222 non mis à jour.
Il est nécessaire de prendre 44,0 ko dans les archives.
Après cette opération, 166 ko d'espace disque supplémentaires seront utilisés.
```

```
Réception de : 1 http://fr.archive.ubuntu.com/ubuntu/ trusty-updates/universe traceroute i386 1:2.0.20-0ubuntu0.1
[44,0 kB]
44,0 ko réceptionnés en 3s (13,2 ko/s)
Sélection du paquet traceroute précédemment désélectionné.
(Lecture de la base de données... 167718 fichiers et répertoires déjà installés.)
Préparation du décompactage de .../traceroute_1%3a2.0.20-0ubuntu0.1_i386.deb ...
Décompactage de traceroute (1:2.0.20-0ubuntu0.1) ...
Traitement déclenché pour  man-db (2.6.7.1-1) ...
Paramétrage de traceroute (1:2.0.20-0ubuntu0.1) ...
update-alternatives: utilisation de « /usr/bin/traceroute.db » pour fournir « /usr/bin/traceroute » (traceroute)
en mode automatique
update-alternatives: utilisation de « /usr/bin/lft.db » pour fournir « /usr/bin/lft » (lft) en mode automatique
update-alternatives: utilisation de « /usr/bin/traceproto.db » pour fournir « /usr/bin/traceproto » (traceproto)
en mode automatique
update-alternatives: utilisation de « /usr/sbin/tcptraceroute.db » pour fournir « /usr/sbin/tcptraceroute »
(tcptraceroute) en mode automatique
```

Elle s'utilise de la façon suivante :

```
root@ubuntu:~# traceroute www.linuxelearning.com
traceroute to www.linuxelearning.com (88.173.201.50), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.320 ms  0.272 ms  0.265 ms
 2  172.16.100.1 (172.16.100.1)  28.722 ms  43.243 ms  43.761 ms
 3  192.168.1.1 (192.168.1.1)  43.605 ms  43.462 ms  43.310 ms
 4  41.138.85.113 (41.138.85.113)  45.292 ms  45.159 ms  45.021 ms
 5  41.215.248.125 (41.215.248.125)  44.882 ms  44.432 ms  44.274 ms
 6  197.155.94.59 (197.155.94.59)  214.112 ms  265.210 ms 197.155.94.29 (197.155.94.29)  264.972 ms
 7  197.155.94.62 (197.155.94.62)  60.963 ms  46.269 ms  49.292 ms
 8  197.155.94.11 (197.155.94.11)  63.579 ms  63.340 ms  63.807 ms
 9  197.155.94.14 (197.155.94.14)  63.629 ms  63.448 ms  63.113 ms
10  * * p11-crs16-1-be1001.intf.routers.proxad.net (78.254.249.5)  231.521 ms
11  p2-tho-teng0-0-0-1.liquidtelecom.net.10.11.5.in-addr.arpa (5.11.10.214)  217.557 ms  217.380 ms  218.367 ms
12  marseille-9k-1-be1002.intf.routers.proxad.net (78.254.249.162)  218.859 ms p1-tho-g0-0-0-1.liquidtelecom.net
(5.11.10.114)  218.599 ms marseille-9k-1-be1002.intf.routers.proxad.net (78.254.249.162)  218.694 ms
```

```
13  pe1-tho-TenG0-0-0-0.liquidtelecom.net (5.11.10.103)  218.406 ms  218.210 ms cor13-1-v900.intf.nra.proxad.net
(78.254.254.54)  250.444 ms
14  ge-8-22.car5.London1.Level3.net (195.50.118.25)  233.023 ms *  264.081 ms
15  ae-2-70.edge4.Paris1.Level3.net (4.69.168.71)  248.143 ms ae-4-90.edge4.Paris1.Level3.net (4.69.168.199)
249.653 ms  249.562 ms
16  au213-1-v902.intf.nra.proxad.net (78.254.254.66)  265.981 ms  265.852 ms 213.242.111.210 (213.242.111.210)
266.178 ms
17  cio13-1-v900.intf.nra.proxad.net (78.254.254.70)  265.541 ms p11-crs16-1-be1001.intf.routers.proxad.net
(78.254.249.5)  265.334 ms cio13-1-v900.intf.nra.proxad.net (78.254.254.70)  265.215 ms
18  marseille-crs8-1-be1000.intf.routers.proxad.net (78.254.249.90)  251.238 ms scy83-1-v902.intf.nra.proxad.net
(78.254.254.74)  233.100 ms marseille-crs8-1-be1000.intf.routers.proxad.net (78.254.249.90)  295.951 ms
19  marseille-9k-1-be1002.intf.routers.proxad.net (78.254.249.162)  250.564 ms  250.482 ms  250.429 ms
20  cor13-1-v900.intf.nra.proxad.net (78.254.254.54)  250.259 ms ban83-1-v902.intf.nra.proxad.net (78.254.254.82)
250.041 ms cor13-1-v900.intf.nra.proxad.net (78.254.254.54)  279.170 ms
21  peh13-1-v902.intf.nra.proxad.net (78.254.254.58)  295.414 ms sf283-1-v900.intf.nra.proxad.net (78.254.254.86)
249.464 ms  249.317 ms
22  lse83-1-v902.intf.nra.proxad.net (78.254.254.90)  249.103 ms au113-1-v900.intf.nra.proxad.net (78.254.254.62)
294.795 ms  294.738 ms
23  lse83-2.dslg.proxad.net (78.254.7.130)  250.741 ms  278.495 ms au213-1-v902.intf.nra.proxad.net
(78.254.254.66)  269.560 ms
24  cio13-1-v900.intf.nra.proxad.net (78.254.254.70)  277.889 ms * *
25  scy83-1-v902.intf.nra.proxad.net (78.254.254.74)  268.732 ms  268.592 ms  268.438 ms
26  * lbe83-1-v900.intf.nra.proxad.net (78.254.254.78)  249.634 ms  249.792 ms
27  * * ban83-1-v902.intf.nra.proxad.net (78.254.254.82)  249.234 ms
28  sf283-1-v900.intf.nra.proxad.net (78.254.254.86)  248.492 ms * *
29  lse83-1-v902.intf.nra.proxad.net (78.254.254.90)  273.368 ms *  265.049 ms
30  lse83-2.dslg.proxad.net (78.254.7.130)  265.140 ms * *
```

Options de la commande traceroute

Les options de cette commande sont :

```
root@ubuntu:~# traceroute --help
```

```
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ] [ -
p port ] [ -t tos ] [ -l flow_label ] [ -w waittime ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [ --
fwmark=num ] host [ packetlen ]
Options:
  -4                          Use IPv4
  -6                          Use IPv6
  -d  --debug                 Enable socket level debugging
  -F  --dont-fragment         Do not fragment packets
  -f first_ttl  --first=first_ttl
                              Start from the first_ttl hop (instead from 1)
  -g gate,...  --gateway=gate,...
                              Route packets through the specified gateway
                              (maximum 8 for IPv4 and 127 for IPv6)
  -I  --icmp                  Use ICMP ECHO for tracerouting
  -T  --tcp                   Use TCP SYN for tracerouting (default port is 80)
  -i device  --interface=device
                              Specify a network interface to operate with
  -m max_ttl  --max-hops=max_ttl
                              Set the max number of hops (max TTL to be
                              reached). Default is 30
  -N squeries  --sim-queries=squeries
                              Set the number of probes to be tried
                              simultaneously (default is 16)
  -n                          Do not resolve IP addresses to their domain names
  -p port  --port=port        Set the destination port to use. It is either
                              initial udp port value for "default" method
                              (incremented by each probe, default is 33434), or
                              initial seq for "icmp" (incremented as well,
                              default from 1), or some constant destination
                              port for other methods (with default of 80 for
                              "tcp", 53 for "udp", etc.)
  -t tos  --tos=tos           Set the TOS (IPv4 type of service) or TC (IPv6
                              traffic class) value for outgoing packets
```

```
  -l flow_label  --flowlabel=flow_label
                              Use specified flow_label for IPv6 packets
  -w waittime  --wait=waittime
                              Set the number of seconds to wait for response to
                              a probe (default is 5.0). Non-integer (float
                              point) values allowed too
  -q nqueries  --queries=nqueries
                              Set the number of probes per each hop. Default is
                              3
  -r                          Bypass the normal routing and send directly to a
                              host on an attached network
  -s src_addr  --source=src_addr
                              Use source src_addr for outgoing packets
  -z sendwait  --sendwait=sendwait
                              Minimal time interval between probes (default 0).
                              If the value is more than 10, then it specifies a
                              number in milliseconds, else it is a number of
                              seconds (float point values allowed too)
  -e  --extensions            Show ICMP extensions (if present), including MPLS
  -A  --as-path-lookups       Perform AS path lookups in routing registries and
                              print results directly after the corresponding
                              addresses
  -M name  --module=name      Use specified module (either builtin or external)
                              for traceroute operations. Most methods have
                              their shortcuts (`-I' means `-M icmp' etc.)
  -O OPTS,...  --options=OPTS,...
                              Use module-specific option OPTS for the
                              traceroute module. Several OPTS allowed,
                              separated by comma. If OPTS is "help", print info
                              about available options
  --sport=num                 Use source port num for outgoing packets. Implies
                              `-N 1'
  --fwmark=num                Set firewall mark for outgoing packets
  -U  --udp                   Use UDP to particular port for tracerouting
```

```
                              (instead of increasing the port per each probe),
                              default port is 53
  -UL                         Use UDPLITE for tracerouting (default dest port
                              is 53)
  -D  --dccp                  Use DCCP Request for tracerouting (default port
                              is 33434)
  -P prot  --protocol=prot    Use raw packet of protocol prot for tracerouting
  --mtu                       Discover MTU along the path being traced. Implies
                              `-F -N 1'
  --back                      Guess the number of hops in the backward path and
                              print if it differs
  -V  --version               Print version info and exit
  --help                      Read this help and exit

Arguments:
+     host          The host to traceroute to
      packetlen     The full packet length (default is the length of an IP
                    header plus 40). Can be ignored or increased to a minimal
                    allowed value
```

Activer/désactiver le routage sur le serveur

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

```
root@ubuntu:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@ubuntu:~# cat /proc/sys/net/ipv4/ip_forward
1
```

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

```
root@ubuntu:~# echo 0 > /proc/sys/net/ipv4/ip_forward
root@ubuntu:~# 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
```

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**.

Options de la commande telnet

Les options de cette commande sont :

```
root@ubuntu:~# telnet --help
telnet: invalid option -- '-'
Usage: telnet [-4] [-6] [-8] [-E] [-L] [-a] [-d] [-e char] [-l user]
    [-n tracefile] [ -b addr ] [-r] [host-name [port]]
```

ssh

Le serveur **openssh** n'est pas installé par défaut sous Ubuntu. Installez-le à l'aide de la commande **apt-get install openssh-server** en tant que root.

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

```
root@ubuntu:~# ssh -l trainee localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 0e:bf:26:8a:cb:e5:3d:19:9d:7a:08:7f:f2:43:94:53.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
trainee@localhost's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

trainee@ubuntu:~$ pwd
/home/trainee
trainee@ubuntu:~$ whoami
trainee
```

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

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

```
trainee@ubuntu:~$ exit
déconnexion
Connection to localhost closed.
```

Options de la commande ssh

Les options de cette commande sont :

```
root@ubuntu:~# ssh --help
unknown option -- -
usage: ssh [-1246AaCfgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-E log_file] [-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]
           [-Q cipher | cipher-auth | mac | kex | key]
           [-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 :

```
root@ubuntu:~# wget ftp://ftp2.fenestros.com/fenestros/files/fichier_test
--2014-10-31 15:59:47--  ftp://ftp2.fenestros.com/fenestros/files/fichier_test
```

```
          => «fichier_test»
Résolution de ftp2.fenestros.com (ftp2.fenestros.com)... 213.186.33.14
Connexion vers ftp2.fenestros.com (ftp2.fenestros.com)|213.186.33.14|:21... connecté.
Ouverture de session en anonymous... Session établie!
==> SYST ... complété.    ==> PWD ... complété.
==> TYPE I ... complété.  ==> CWD (1) /fenestros/files ... complété.
==> SIZE fichier_test ... 57
==> PASV ... complété.    ==> RETR fichier_test ... complété.
Taille: 57 (non certifiée)

100%[======================================>] 57          --.-K/s   ds 0,02s

2014-10-31 15:59:51 (3,59 KB/s) - «fichier_test» enregistré [57]
```

Options de la commande wget

Les options de cette commande sont :

```
root@ubuntu:~# wget --help
GNU Wget 1.15, un récupérateur réseau non interactif.
Usage: wget [OPTION]... [URL]...

Les arguments obligatoires pour les options de format long le sont
aussi pour les options de format court.

Démarrage:
  -V,  --version           afficher la version de Wget et quitter.
  -h,  --help              afficher l'aide-mémoire.
  -b,  --background        passer à l'arrière plan après le démarrage.
  -e,  --execute=COMMANDE  exécuter une commande `.wgetrc'-style

Journalisation et fichier d'entrée:
  -o,  --output-file=FICHIER journaliser les messages dans le FICHIER.
```

```
  -a,  --append-output=FICHIER accoler les messages au FICHIER.
  -d,  --debug                afficher beaucoup d'informations de débogage.
  -q,  --quiet                exécuter en mode silencieux (sans sortie d'affichage).
  -v,  --verbose              exécuter en mode bavard (mode par défaut).
  -nv, --no-verbose           éteindre le mode bavard, sans être silencieux.
       --report-speed=TYPE      afficher la bande passante en TYPE (bits par ex.)
  -i,  --input-file=FIC       télécharge les URLs trouvées dans FIChier local ou externe.
  -F,  --force-html           traiter le fichier d'entrée comme du HTML.
  -B,  --base=URL             résout les liens HTML du fichier en
                              entrée (-i -F) relativement à URL,
       --config=FICHIER         indiquer le FICHIER de configuration à utiliser.

Téléchargement :
  -t,  --tries=NOMBRE             fixer le NOMBRE de tentatives de reprises (0 : sans limite).
       --retry-connrefused        ré-essayer même si la connexion est refusée.
  -O,  --output-document=FICHIER écrire les documents dans le FICHIER.
  -nc, --no-clobber               sauter les téléchargements de fichiers
                                  déjà existants (qui auraient été écrasés).
  -c,  --continue                 poursuivre le téléchargement d'un fichier partiellement téléchargé.
       --progress=TYPE            sélectionner le type de jauge de progression de téléchargement.
  -N,  --timestamping             ne pas re-télécharger les fichiers à moins que
                                  qu'il y en ait de plus récents que les locaux.
  --no-use-server-timestamps      ne pas positionner la date locale du
                                  fichier avec celle du serveur.
  -S,  --server-response          afficher la réponse du serveur.
       --spider                   ne rien télécharger.
  -T,  --timeout=SECONDES         fixer toutes les valeurs de délai maximal d'attente à SECONDES.
       --dns-timeout=SECS         fixer le délai maximal d'attente de recherche DNS à SECS.
       --connect-timeout=SECS     fixer le délai maximal d'attente de connexion à SECS.
       --read-timeout=SECS        fixer le délai maximal d'attente de lecture à SECS.
  -w,  --wait=SECONDES            attendre SECONDES entre les essais.
       --waitretry=SECONDES       attendre 1..SECONDES entre les essais d'une récupération.
       --random-wait              attendre de 0.5 à 1.5 fois SECS s entre les tentatives.
       --no-proxy                 désactivier explicitement le proxy.
```

```
  -Q,  --quota=NOMBRE             fixer le quota de récupération à NOMBRE.
       --bind-address=ADRESSE     lier à l'ADRESSE (nom d'hôte ou adresse IP) sur l'hôte local.
       --limit-rate=TAUX          limiter le TAUX de téléchargement.
       --no-dns-cache             désactiver la mise en cache des résultats de recherche DNS.
       --restrict-file-names=OS   restreindre les caractères dans les noms de fichier à ceux permis par l'OS.
       --ignore-case              ignore la casse des caractères lors de l'examen des fichiers/répertoires.
  -4,  --inet4-only               connecter seulement sur des adresses IPv4.
  -6,  --inet6-only               connnecter seulement sur des adresses IPv6.
       --prefer-family=FAMILLE    connecter d'abord sur des adresses de la FAMILLE,
                                  soit IPv6, IPv4 ou none (pour aucun).
       --user=USAGER              fixer l'utilisateur à USAGER pour ftp et http.
       --password=MOT_DE_PASSE    fixer le MOT_DE_PASSE pour ftp et http.
       --ask-password             demander les mots de passe.
       --no-iri                   désactive le support des IRIs.
       --local-encoding=ENC       utiliser l'encodage local ENC pour les IRIs.
       --remote-encoding=ENC      utiliser l'encodage distant ENC par défaut.
       --unlink                   supprimer le fichier avant de l'écraser.

Répertoires :
  -nd, --no-directories           ne pas créer de répertoires.
  -x,  --force-directories        forcer la création de répertoires.
  -nH, --no-host-directories      ne pas créer de répertoires sur l'hôte.
       --protocol-directories     utiliser le nom du protocole dans les répertoires.
  -P,  --directory-prefix=PRÉFIXE enregistre les fichiers avec PRÉFIXE/...
       --cut-dirs=NOMBRE          ignorer le NOMBRE de composants des répertoires distants.

options HTTP :
       --http-user=USAGER      fixer l'USAGER http.
       --http-password=MDP    fixer le MDP (mot de passe) http.
       --no-cache              interdire les données mise en cache sur le serveur.
       --default-page=NOM      Change le nom de la page par défaut
                               (normalement "index.html").
  -E,  --adjust-extension      sauver les documents HTML avec l'extension adaptée.
       --ignore-length         ignorer le champ de l'en-tête `Content-Length'.
```

```
       --header=CHAÎNE          insérer la CHAÎNE parmi les en-têtes.
       --max-redirect           nbr maximum de redirections autorisées par page.
       --proxy-user=USAGER      fixer le nom d'USAGER proxy.
       --proxy-password=MDP     fixer le MDP (mot de passe) du proxy.
       --referer=URL            inclure l'en-tête `Referer: URL' dans la requête HTTP.
       --save-headers           enregistre les en-têtes HTTP dans le fichier.
  -U,  --user-agent=AGENT       s'identifier comme AGENT au lieu de Wget/VERSION.
       --no-http-keep-alive     désactiver l'option HTTP keep-alive (connexions persistentes).
       --no-cookies             ne pas utiliser les cookies.
       --load-cookies=FICHIER   charger les cookies à partir du FICHIER avant la session.
       --save-cookies=FICHIER   enregistre les cookies dans le FICHIER après la session.
       --keep-session-cookies   charge et enregistre les cookies de session non permanents.
       --post-data=CHAÎNE       utiliser une méthode POST; transmettre la CHAÎNE
                                    comme des données.
       --post-file=FICHIER      utiliser une méthode POST; transmettre le contenu du FICHIER.
       --method=MéthodeHTTP     utiliser la « MéthodeHTTP » dans l’en-tête.
       --body-data=CHAÎNE       envoyer la CHAÎNE comme données.
                                --method doit être définie.
       --body-file=FICHIER      envoyer le contenu du FICHIER.
                                --method doit être définie.
       --content-disposition    tient compte de l'entête "Content-Disposition" pour
                                le choix des noms de fichiers locaux (EXPERIMENTAL).
       --content-on-error       afficher le contenu reçu après erreurs serveur.
       --auth-no-challenge      envoie l'information d'authentification basique HTTP
                                sans attendre d'abord le certificat du serveur.

options HTTPS (SSL/TLS):
       --secure-protocol=PR      choisir un protocole sécurisé PR parmi auto,
                                 SSLv2, SSLv3, TLSv1 et PFS.
       --https-only              ne suivre que les liens HTTPS sécurisé.
       --no-check-certificate    ne pas valider le certificat du serveur.
       --certificate=FICHIER     fichier du certificat client.
       --certificate-type=TYPE   type du certificat client, PEM ou DER.
       --private-key=FICHIER     fichier de la clé privée.
```

```
       --private-key-type=TYPE  type de clé privée, PEM ou DER.
       --ca-certificate=FICHIER fichier avec un lot de certificats autorités.
       --ca-directory=RÉP       répertoire où la liste de hash des certificats autorités est stockée.
       --random-file=FICHIER    fichier avec des données aléatoires pour le germe de SSL PRNG.
       --egd-file=FICHIER       dénomination de fichier du socket EGD avec données aléatoires.

options FTP:
       --ftp-user=USAGER       utiliser USAGER comme utilisateur pour le transfert ftp.
       --ftp-password=MDP      utiliser le MDP (mot de passe) pour les transfert ftp.
       --no-remove-listing     ne pas enlever les fichiers `.listing'.
       --no-glob               désactiver la mutilation des noms de fichiers par FTP.
       --no-passive-ftp        désactiver le mode de transfert passif.
       --preserve-permissions  préserver les permissions des fichiers distants.
       --retr-symlinks         lors de la récursion, prendre les fichiers attachés à des liens (pas les
répertoires).

options WARC :
       --warc-file=FICHER        sauver les données de requête et de réponse
                                 dans un fichier .warc.gz.
       --warc-header=CHAÎNE      insérer CHAÎNE dans l'enregistrement warcinfo.
       --warc-max-size=NOMBRE    définir la taille maximal de fichiers WARC.
       --warc-cdx                écrire les fichiers d'index CDX.
       --warc-dedup=FICHIER      ne pas garder enregistrements du fichier CDX.
       --no-warc-compression     ne pas compresser les fichiers WARC avec gzip.
       --no-warc-digests         ne pas calculer les hachages SHA1.
       --no-warc-keep-log        ne pas garder journal dans enregistrement WARC.
       --warc-tempdir=RÉPERTOIRE emplacement pour fichiers temporaires créés
                                 par l'écriture WARC.

Téléchargement récursif:
  -r,  --recursive          activer les téléchargements récursifs.
  -l,  --level=NOMBRE       profondeur maximale de récursion (inf ou 0 pour infini).
       --delete-after       détruire les fichiers localement après les avoir téléchargés.
  -k,  --convert-links      fait pointer les liens dans le HTML/CSS téléchargé vers des fichiers locaux.
```

```
  --backups=N          avant d'écrire le fichier X, en sauver un
                            exemplaire, et en garder au plus N.
  -K,  --backup-converted   avant de convertir le fichier X en faire l'archive sous X.orig.
  -m,  --mirror             option courte équivalente à -N -r -l inf --no-remove-listing.
  -p,  --page-requisites    obtenir toutes les images, etc. nécessaires à l'affichage de la page HTML.
       --strict-comments    activer le traitement strict (SGML) des commentaires HTML.

Acceptation/rejet récursif:
  -A,  --accept=LISTE              liste des extensions acceptées, séparées par des virgules.
  -R,  --reject=LISTE              liste des extensions rejetées, séparées par des virgules.
       --accept-regex=EXPRESSION_R expression rationnelle correspondant aux
                                   URL acceptées.
       --reject-regex=EXPRESSION_R expression rationnelle correspondant aux
                                   URL rejetées.
       --regex-type=TYPE           type d'expression rationnelle (posix).
  -D,  --domains=LISTE             liste des domaines acceptés, séparés par des virgules.
       --exclude-domains=LISTE     liste des domaines rejetés, séparés par des virgules.
       --follow-ftp                suivre les liens FTP à partir des documents HTML.
       --follow-tags=LISTE         liste des balises HTML à suivre,  séparées par des virgules.
       --ignore-tags=LISTE         liste des balises HTML ignorées, séparées par des virgules.
  -H,  --span-hosts                aller sur les hôtes externes en mode récursif.
  -L,  --relative                  suivre les liens relatifs seulement.
  -I,  --include-directories=LISTE liste des répertoires permis.
  --trust-server-names             utiliser le nom indiqué par le suffixe de
                                   l'URL de redirection.
  -X,  --exclude-directories=LISTE liste des répertoires exclus.
  -np, --no-parent                 ne pas remonter dans le répertoire parent.

Transmettre toutes anomalies ou suggestions à <bug-wget@gnu.org>.
```

ftp

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

```
root@ubuntu:~# ftp ftp2.fenestros.com
```

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:

!       debug        mdir        qc        send
$       dir        mget        sendport    site
account     disconnect    mkdir        put        size
append      exit        mls        pwd        status
ascii       form        mode        quit        struct
bell        get        modtime        quote        system
binary      glob        mput        recv        sunique
bye     hash        newer        reget        tenex
case        help        nmap        rstatus        tick
cd      idle        nlist        rhelp        trace
cdup        image        ntrans        rename        type
chmod       lcd        open        reset        user
close       ls        prompt        restart        umask
cr      macdef        passive        rmdir        verbose
delete      mdelete        proxy        runique        ?
```

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
root@ubuntu:~#
```

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 :

```
root@ubuntu:~# scp --help
usage: scp [-1246BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
           [-l limit] [-o ssh_option] [-P port] [-S program]
           [[user@]host1:]file1 ... [[user@]host2:]file2
```
