Configurer une interface réseau Ethernet

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Résumé

L'objectif de ces travaux pratiques est d'analyser et configurer une interface réseau Ethernet sur un système GNU/Linux. Les manipulations présentées suivent la modélisation réseau en remontant du niveau physique jusqu'à la couche application. Les questions illustrent les relations entre les différents formats d'adressage utilisés à chaque niveau ainsi que les protocoles utilisés pour les correspondances entre les différentes couches.

Table des matières

1. Copyright et Licence

```
Copyright (c) 2000,2020 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no
Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included
in the section entitled "GNU Free Documentation License".
```

```
Copyright (c) 2000,2020 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon
les termes de la Licence de Documentation Libre GNU (GNU Free Documentation
License), version 1.3 ou toute version ultérieure publiée par la Free Software
Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture,
et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est
incluse dans la section intitulée « Licence de Documentation Libre GNU ».
```

Méta-information

Cet article est écrit avec DocBook XML sur un système Debian GNU/Linux. Il est disponible en version imprimable au format PDF : `conf-intf-lan.pdf`.

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. C'est la distribution Debian GNU/Linux qui est utilisée pour les tests présentés. Voici une liste des principaux paquets contenant les commandes utilisées :

- ethtool - display or change Ethernet device settings
- iproute2 - networking and traffic control tools
- ifupdown - High level tools to configure network interfaces
- iputils-ping - Tools to test the reachability of network hosts
- procps - /proc file system utilities
- mtr-tiny - Full screen ncurses traceroute tool

Conventions typographiques

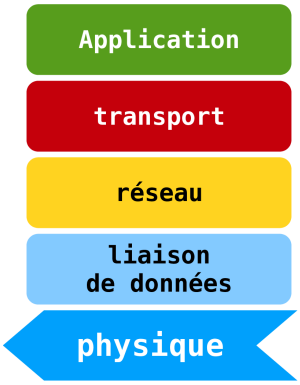
Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou prompt spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite `$` ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.
- Toute commande précédée de l'invite `#` nécessite les privilèges du super-utilisateur. Ces privilèges peuvent être délégués à l'aide de sudo.

2. Identifier les ressources matérielles

Couche physique

Avant de pouvoir configurer une interface, il faut que le pilote de périphérique correspondant ait été chargé en mémoire. Comme une interface réseau est un dispositif matériel, c'est au niveau du noyau Linux que l'opération doit s'effectuer. Soit le pilote d'interface a été inclus dans la partie monolithique du noyau soit il est chargé en mémoire sous forme de module. C'est cette dernière solution qui est le plus souvent retenue. Un module peut être chargé ou déchargé à volonté sans avoir à redémarrer la machine. De plus, les fonctions de reconnaissance automatique des composants périphériques permettent de ne charger que les modules correspondant aux composants effectivement présents sur le système.



Comment identifier un périphérique Ethernet ?

Il existe une grande variété de contrôleurs réseau Ethernet. À chaque famille de composants correspond un pilote logiciel spécifique. Qu'il s'agisse d'une carte additionnelle ou d'un composant intégré sur carte mère, le contrôleur peut être connecté via différents bus. Les bus PCI et USB sont les plus fréquemment utilisés. Voici deux exemples :

Contrôleur Ethernet sur bus PCI

Sur une architecture de type PC, un composant Ethernet est la plupart du temps relié au bus PCI. La commande lspci du paquet pciutils donne la liste des périphériques ainsi que les modules du noyau Linux associés à ces périphériques.

```
$ lspci -v | grep -A8 Ethernet
00:19.0 Ethernet controller: Intel Corporation Ethernet Connection (3) I218-V (rev 03)
        Subsystem: Intel Corporation Ethernet Connection (3) I218-V
        Flags: bus master, fast devsel, latency 0, IRQ 43
        Memory at f7100000 (32-bit, non-prefetchable) [size=128K]
        Memory at f713b000 (32-bit, non-prefetchable) [size=4K]
        I/O ports at f080 [size=32]
        Capabilities: <access denied>
  Kernel driver in use: e1000e
  Kernel modules: e1000e
```

Le module du noyau Linux nommé `e1000e` est chargé en mémoire automatiquement lors de l'initialisation du système. Il est présent dans la liste donnée par la commande lsmod.

```
$ lsmod | grep e1000e
e1000e                278528  0
ptp                    20480  1 e1000e
```

Contrôleur Ethernet sur bus USB

Sur une architecture Raspberry Pi, le composant Ethernet intégré est relié au bus USB et c'est la commande lsusb qui permet d'obtenir l'identification du composant.

```
$lsusb | grep Ethernet | fmt -t -w80
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514
   Fast Ethernet Adapter
```

Dans le cas d'un système Raspberry Pi, le logiciel de pilotage de l'interface Ethernet est placé dans la partie monolithique du noyau. Ce logiciel n'apparaît donc pas dans la liste des modules. Il faut consulter les messages système du noyau pour trouver la trace de l'initialisation de l'interface Ethernet. On peut utiliser la commande dmesg dans ce but.

```
$ dmesg | grep Ethernet | fmt -t -w80
[    2.016771] smsc95xx 1-1.1:1.0 eth0: register 'smsc95xx' at
   usb-3f980000.usb-1.1, smsc95xx USB 2.0 Ethernet, b8:27:eb:d7:21:80
[    9.733711] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
```

Pour résumer, les outils utiles pour l'identification des composants réseau et des modules logiciels asscié sont : lspci, lsusb, dmesg et lsmod.

Comment visualiser l'état du lien réseau ?

Même avec une configuration correcte de l'interface, il est possible que les communications soient bloquées si le raccordement physique entre l'hôte et l'équipement réseau n'est pas actif. Sur les câbles en paires torsadées cuivre, on peut visualiser l'état du lien à l'aide de la commande ethtool fournie avec le paquet du même nom. En reprenant les deux exemples de systèmes ci-dessus, on obtient les informations suivantes.

Contrôleur Ethernet sur bus PCI

```
$ sudo ethtool eth0
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full ❶
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s ❷
        Duplex: Full
        Port: Twisted Pair ❸
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on ❹
        MDI-X: on (auto)
        Supports Wake-on: pumbg
        Wake-on: g
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

❶ Cette liste correspond aux débits possibles sur cette interface.
❷ Le lien entre l'interface `eth0` et l'équipement réseau est actif et le débit négocié est le Gbps en mode Full-Duplex.
❸ Le câble connecté à cette interface est en paire cuivre torsadée ou twisted pair.
❹ Les résultats précédents ont été obtenus par auto négociation entre le contrôleur réseau et le commutateur auquel l'interface Ethernet est raccordée.

Contrôleur Ethernet sur bus USB

```
$ sudo ethtool eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Advertised pause frame use: Symmetric Receive-only
        Advertised auto-negotiation: Yes
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full
        Link partner advertised pause frame use: No
        Link partner advertised auto-negotiation: Yes
  Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: pumbag
        Wake-on: d
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

Relativement au contrôleur Ethernet sur bus PCI, le débit passe à 100Mb/s.

Pour aller plus loin dans l'étude des caractéristiques techniques des réseaux locaux Ethernet, il est conseillé de lire l'article Technologie Ethernet.

3. Lire et configurer l'état d'une interface

Couche liaison de données

Au niveau liaison de données de la modélisation, l'unité de donnée manipulée est la trame. Dans le cas de la technologie Ethernet, la trame contient les adresses MAC (media access control address) des hôtes source et destination du réseau de diffusion (LAN). À ce niveau, il est possible de configurer plusieurs fonctions. Vis-à-vis de la couche physique, on peut activer ou désactiver une interface. Vis-à-vis de la couche réseau, il existe de nombreux paramètres que l'on peut consulter et redéfinir.

Même si l'étude des paramètres définis au niveau liaison de données sort du cadre de ce document, on peut citer trois exemples significatifs.

- Il est possible de fixer la quantité de données provenant de la couche réseau à encapsuler dans une trame. Au delà de la valeur par défaut (1500 octets), on parle de Jumbo frame.
- On peut ajouter un jeu d'étiquettes aux trames en utilisant le standard IEEE 802.1Q de façon à définir des VLANs. Voir l'article Routage Inter-VLAN.
- On peut sélectionner et configurer les fonctions relatives à la classification et à la gestion de mise en file d'attente des paquets issus de la couche réseau. Voir HOWTO du routage avancé et du contrôle de trafic sous Linux.

Dans cette section, le principal outil utilisé est la commande ip du paquet iproute2.

Comment visualiser l'état d'une interface réseau ?

Le simple fait de consulter l'état d'une interface fournit une grande quantité d'informations.

```
$ ip link ls dev eth0 | fmt -t -w80
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP>❶ mtu 1500❷ qdisc fq state UP
    mode DEFAULT group default qlen 1000 link/ether b8:ae:ed:73:06:de❸ brd
    ff:ff:ff:ff:ff:ff❹
```

❶ Les indicateurs d'état désignent les fonctions actives au niveau de l'interface.
❷ L'acronyme MTU signifie Maximum Transmission Unit. La valeur `1500` correspond à la quantité maximum d'octets transmis de la couche réseau à la couche liaison de données sans fragmentation.
❸ L'adresse MAC de l'interface joue un rôle essentiel. C'est cette adresse qui identifie l'hôte dans le réseau local (LAN). Cette adresse unique respecte un format bien particulier : EUI-48. Voir Types d'adresses MAC.
❹ L'adresse de diffusion utilisée par l'interface respecte les champs du format EUI-48 mais tous les bits des 6 octets sont à 1. Cette adresse est placée dans le champ adresse destination d'une trame d'annonce ou de requête vers tous les hôtes du réseau local (LAN).

Tableau 1. Indicateurs d'état d'une interface Ethernet

Indicateur	Description
`BROADCAST`	L'interface peut émettre du trafic à destination de tous les hôtes du réseau local.
`MULTICAST`	L'interface peut émettre et recevoir du trafic de type multidiffusion.
`UP`	L'interface est active et correctement configurée au niveau liaison de données.
`LOWER_UP`	L'interface est électriquement active au niveau physique. La LED du port est allumée.

Indicateur	Description
`PROMISC`	L'interface traite tout le trafic reçu et le transmet aux couches supérieures du sous-système réseau. Ce traitement inclut les trames dont l'adresse MAC destination est différente de celle de l'interface.
`ALLMULTI`	L'interface traite tout le trafic de multidiffusion reçu et le transmet aux couches supérieures. Ce mode est utile sur un système qui route le trafic de multidiffusion.

Comment visualiser les statistiques d'une interface réseau ?

En cas de problème de transmission, il est essentiel de connaître le nombre d'erreurs comptabilisé par le composant Ethernet ainsi que le nombre total de paquets émis ou reçus. Voici un exemple :

```
$ ip -s link ls dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq state UP
   mode DEFAULT group default qlen 1000
    link/ether b8:ae:ed:73:06:de brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    21622131   216995   0       0       0       263
    TX: bytes  packets  errors  dropped carrier collsns
    422328441  358477   0       0       0       0
```

Dans le cas d'une interface Ethernet filaire, les compteurs d'erreurs, de trames abandonnées et de collisions doivent impérativement rester à 0. En effet, une connexion Ethernet filaire en cuivre ou en fibre optique fonctionne normalement en full duplex ; c'est à dire que l'on dispose d'un premier canal de transmission réservé pour l'émission et d'un second canal réservé pour la réception.

Comment activer/désactiver une interface réseau ?

Ces opérations peuvent s'effectuer à différents niveaux bien distincts.

- Sur un système Debian GNU/linux ou apparenté, les scripts ifup et ifdown du paquet ifupdown utilisent les paramètres de configuration des interfaces donnés dans le fichier `/etc/network/interfaces` lors de l'activation ou la désactivation.
- Dans un contexte de mobilité avec un ordinateur portable, la quasi totalité des distributions Linux proposent d'utiliser NetworkManager. Cet outil permet à un utilisateur normal de définir ses propres configurations réseau en fonction du contexte sans obtenir les droits d'administration système.
- Les manipulations au niveau interface ne tiennent aucun compte du mode de configuration antérieur. L'exécution des outils de configuration dans l'espace utilisateur peut se poursuivre alors que l'interface associée est inactive. Une telle situation peut conduire à des problèmes de fonctionnement du système ! Il est donc important de recenser les paramètres associés à une interface avant de se lancer dans les manipulations directes.

> **Avertissement**
>
> La désactivation d'une interface entraîne la perte des routes vers les réseaux IP qui dépendent de ce lien.

Désactivation au niveau système

```
# ifdown eth0
```

Activation au niveau système

```
# ifup eth0
```

Désactivation d'une connection avec NetworkManager

On commence par une identification de la connection filaire active avant de la désactiver.

```
$ nmcli connection show --active
NAME             UUID                                  TYPE      DEVICE
Wired connection  d1cbf24c-f218-492e-a0ae-d99052b9fbb2  ethernet  enp0s31f6

$ nmcli device disconnect enp0s31f6
```

Activation d'une connection avec NetworkManager

```
$ nmcli device connect enp0s31f6
```

Désactivation au niveau interface

```
# ip link set dev eth0 down

$ ip link ls dev eth0 | fmt -t -w80
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
 group default qlen 1000 link/ether 00:26:18:27:07:54 brd ff:ff:ff:ff:ff:ff
```

Activation au niveau interface

```
# ip link set dev eth0 up

$ ip link ls dev eth0 | fmt -t -w80
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq state UP
    mode DEFAULT group default qlen 1000 link/ether 00:26:18:27:07:b9 brd
    ff:ff:ff:ff:ff:ff
```

Comment changer l'adresse MAC d'une interface réseau ?

Parmi les nombreuses manipulations possibles avec la commande ip link, il est possible de changer l'adresse MAC d'une interface. Voici un exemple.

```
# ip link set dev eth0 down

# ip link set address de:ad:be:ef:00:01 dev eth0

# ip link ls dev eth0 | fmt -t -w80
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
    group default qlen 1000 link/ether de:ad:be:ef:00:01 brd ff:ff:ff:ff:ff:ff

# ip link set dev eth0 up
```

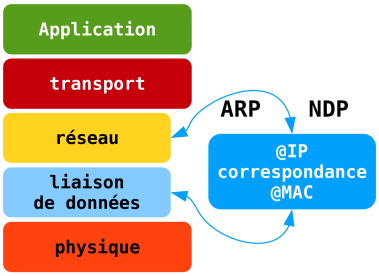
Pour aller plus loin dans les manipulations au niveau liaison de données, la consultation des pages de manuels est un excellent point de départ : `$ man ip-link`.

4. Reconnaître le voisinage réseau

Couches liaison de données et réseau

Dans un réseau IPv4, le protocole ARP ou Address Resolution Protocol a pour but de faire la correspondance entre une adresse MAC inconnue (celle de l'hôte destinataire) et une adresse IPv4 connue (encore celle de l'hôte destinataire).
Ce protocole fait le «lien» entre les mécanismes d'adressage de la couche réseau et de la couche liaison de données.

Dans un réseau IPv6, le protocole NDP ou Neighbor Discovery Protocol se substitue au protocole ARP pour faire la correspondance entre les adresses MAC et les adresses IPv6 de lien local appartenant au réseau `fe80/10`.



Si le routage assuré au niveau réseau permet d'acheminer le trafic utilisateur d'un réseau à l'autre, il ne permet pas de joindre directement un hôte dans un réseau local de diffusion comme Ethernet. Au niveau liaison de données les adresses MAC servent à repérer un hôte unique dans le réseau local de diffusion. Il faut donc établir une correspondance entre des adresses dont la portée ne dépasse pas le réseau local et d'autres adresses dont la portée recouvre de multiples réseaux.

Dans cette section, le principal outil utilisé est la commande ip du paquet iproute2.

Comment visualiser la table des voisins ?

Le sous-système réseau du noyau Linux maintient une table «table des hôtes voisins» contenant les correspondances avec les adresses IPv4 et IPv6.

La commande `ip nei ls` fait apparaître toutes les adresses de voisins connus dans un affichage commun. Ces résultats sont obtenus via ARP pour les adresses IPv4 et NDP pour les adresses IPv6.

```
$ ip nei ls dev eth0
192.0.2.1❶ dev eth0❷ lladdr ba:f1:b6:e4:a0:bd❸ STALE❹
fe80::b8f1:b6ff:fee4:a0bd❺ dev eth0 lladdr ba:f1:b6:e4:a0:bd router REACHABLE
2001:db8:8083:c41e::1❻ dev eth0 lladdr ba:f1:b6:e4:a0:bd router STALE
```

❶ L'hôte avec l'adresse IPv4 `192.0.2.1` est un voisin appartenant au même domaine de diffusion.
❷ L'interface Ethernet `eth0` désigne le domaine de diffusion. Les voisins joignables via cette interface appartiennent tous au même domaine de diffusion.
❸ Cette adresse MAC a été obtenue grâce au protocole ARP. Dès qu'un paquet est émis à destination de l'hôte `192.0.2.1` ou à destination d'un autre réseau si ce même hôte est un routeur, la trame sera composée avec l'adresse MAC destination `ba:f1:b6:e4:a0:bd`.
❹ Les indicateurs d'état informent sur la validité de la correspondance entre les adresses de couche liaison de données et les adresses de couche réseau.
❺ La correspondance avec l'adresse IPv6 de lien local `fe80::b8f1:b6ff:fee4:a0bd` a été établie grâce au protocole NDP. les adresses IPv6 de lien local sont composées automatiquement à partir du préfixe `fe80::/10` et de l'adresse MAC au format EUI-64. Voir Types d'adresses MAC.
❻ Cette ligne correspond au même hôte voisin et donne son adresse IPv6 publique ou globale.

Tableau 2. Indicateurs d'état de la table des hôtes voisins

Indicateur	Description
`INCOMPLETE`	La résolution d'adresse de l'hôte voisin est en cours
`REACHABLE`	La correspondance entre les adresses IP et MAC a bien été établie et l'hôte voisin est apparemment joignable

Indicateur	Description
STALE	La correspondance entre les adresses IP et MAC a bien été établie mais l'hôte voisin n'est probablement plus joignable et une vérification sera lancée dès la première émission.
DELAY	Un paquet a été émis à destination d'un voisin dans l'état STALE et une confirmation de correspondance d'adresses est en attente
PROBE	La temporisation de l'état DELAY est expirée et la correspondance d'adresses n'a pas été confirmée ; une nouvelle résolution d'adresse a été initiée
FAILED	La résolution d'adresse a échoué
NOARP	Le voisin est validé ; aucune vérification ne doit être faite.
PERMANENT	Identique à NOARP ; seul le super utilisateur a la possibilité de supprimer l'entrée de la table

Les voisins IPv6 peuvent apparaître avec un indicateur `router` supplémentaire ; ce qui signifie que ce voisin se présente comme un routeur IPv6.

Comment effacer tout ou partie de la table des voisins ?

Pour tester le fonctionnement du mécanisme de résolution d'adresses, il peut être utile d'effacer une ou plusieurs entrées dans le but de provoquer une nouvelle résolution lors des prochains échanges réseau.

Suppression d'une entrée particulière

```
# ip nei del 192.168.1.2 dev eth0
```

Suppression de toutes les entrées relatives à l'interface eth0

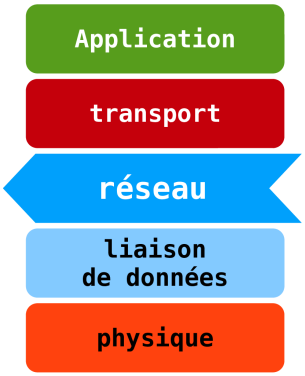
```
# ip neighbor flush dev eth0
```

Pour découvrir les autres manipulations possibles, il est conseillé de consulter les pages de manuels : `$ man ip-neighbour`.

5. Lire et configurer les adresses réseau d'une interface

Couche réseau

Au niveau réseau de la modélisation, l'unité de donnée manipulée est le paquet. Comme IPv4 et IPv6 sont des réseaux à commutation de paquets, chaque en-tête de paquet comprend les adresses source et destination. C'est sur la base de l'adresse IP destination et du masque réseau qu'un routeur prend ses décisions d'acheminement du trafic utilisateur.



Par définition, une adresse IP désigne à la fois un hôte et le réseau auquel il appartient. La distinction entre la partie réseau et la partie hôte d'une adresse se fait grâce au masque réseau. Il est donc logique que l'adresse et le masque soient les deux paramètres les plus importants dans la configuration IP d'une interface. Voir le document Adressage IPv4.

Dans cette section, le principal outil utilisé est la commande ip du paquet iproute2.

Comment visualiser la liste des adresses IP d'une interface ?

L'exemple d'exécution de la commande `$ ip addr ls dev eth0` donné ci-dessous caractérise bien le fait que l'on peut affecter plusieurs adresses réseau à une même interface. Chacune de ces adresses a une portée propre.

À la différence d'une adresse MAC qui n'est visible que dans son réseau local (LAN), une adresse IPv4 ou IPv6 peut être visible à plusieurs niveaux d'interconnexion de réseaux.

```
$ ip addr ls dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether❶ ba:ad:ca:fe:00:1e brd ff:ff:ff:ff:ff:ff
    inet❷ 192.0.2.30/26❸ brd 192.0.2.31❹ scope❺ global eth0
       valid_lft forever preferred_lft forever
    inet6 2001:db8:8083:c41e:b8ad:caff:fefe:1e/64❻ scope global dynamic mngtmpaddr
       valid_lft 86119sec preferred_lft 14119sec
    inet6 fe80::b8ad:caff:fefe:1e/64❼ scope link
       valid_lft forever preferred_lft forever
```

❶ L'affichage de la liste des adresses englobe les couches liaison de données et réseau.

❷ Le premier élément désigne la famille d'adresse. Les deux valeurs présentées ici son `inet` pour les adresses IPv4 et `inet6` pour les adresses IPv6.

❸ L'adresse IPv4 est donnée en notation CIDR. La valeur à droite du caractère `'/'` donne le nombre de bits à `1` du masque réseau. Voir le document Adressage IPv4.

❹ L'adresse de diffusion de niveau réseau est donnée après l'indicateur `brd`. Voir le document Adressage IPv4.

❺ Pour chaque adresse affichée, l'indicateur `scope` précise la portée de l'information. Ici les valeurs sont `global` lorsqu'une adresse est joignable depuis d'autres réseaux et `link` lorsque la portée se limite au seul domaine de diffusion.

❻ L'adresse IPv6 publique présentée ici est obtenue par configuration automatique sans état ou SLAAC. On reconnaît l'adresse MAC de l'interface dans les 4 hextets de droite avant le `'/'`. La partie hôte de cette adresse correspond aux 64 bits de poids faible. Voir IPv6.

❼ L'adresse IPv6 de lien local est composée automatiquement à partir du préfixe `fe80::/10` et de l'adresse MAC au format EUI-64. Cette adresse apparaît dès que l'interface est active sans aucune opération de configuration. Une adresse IPv6 de lien local est nécessaire au fonctionnement du protocole NDP. Voir Neighbor Discovery Protocol.

Pour découvrir la signification des autres champs possibles, il est vivement conseillé de consulter les pages de manuels : `$ man ip-address`.

Comment ajouter ou supprimer une adresse à une interface ?

Avertissement

La suppression de l'adresse IP d'une interface entraîne la perte des routes vers les réseaux qui dépendent de ce lien.

La syntaxe de suppression puis d'ajout d'une adresse à une interface Ethernet est donnée ci-dessous.

```
# ip addr del 192.168.1.1/24 dev eth0

# ip addr add 192.168.1.1/24 brd + dev eth0
```

Comment rendre la configuration permanente sur un système Debian GNU/Linux ?

Avec la distribution Debian GNU/Linux, ainsi que pour les distributions dérivées, les paramètres de configuration des interfaces réseau sont stockés dans le répertoire `/etc/network`. Le fichier `interfaces` de ce répertoire rassemble la configuration des interfaces réseau.

Voici l'exemple d'une interface Ethernet configurée à l'aide du protocole DHCP ou Dynamic Host Configuration Protocol.

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# The first network card - this entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet dhcp
```

Voici un autre exemple avec une configuration statique de l'interface Ethernet. On affecte l'adresse IPv4, le masque réseau, la passerelle par défaut ainsi que le serveur DNS à contacter pour résoudre les noms de domaines.

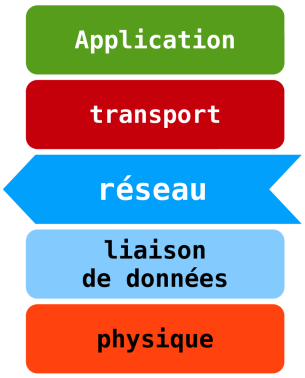
```
<snip/>
auto eth0
iface eth0 inet static
 address 192.168.1.10/24
 gateway 192.168.1.1
 dns-nameserver 8.8.8.8
```

La syntaxe de l'ensemble des options de configuration d'une interface réseau est décrite dans les pages de manuels : `$ man interfaces`.

6. Lire une table de routage simple et changer la passerelle par défaut

Couche réseau

Le routage est une fonction essentielle de la couche réseau. Les données du trafic utilisateur sont encapsulées en allant de la couche application jusqu'à la couche réseau dans des paquets IP. Ces paquets sont routés jusqu'à l'hôte correspondant à l'adresse IP destination. En fonctionnement normal, un routeur prend ses décisions d'acheminement en analysant l'adresse IP destination de chaque paquet. Ces prises de décisions se font à partir des informations présentes dans la table de routage.



Cela peut paraître surprenant, mais tout hôte disposant d'un sous-système réseau dans le noyau ou dans un composant équivalent, utilise une table de routage. Bien sûr, pour un système avec une interface Ethernet unique, le nombre d'entrées dans la table de routage est limité.

Dans cette section, le principal outil utilisé est la commande ip du paquet iproute2.

Comment visualiser la table de routage ?

Dans l'exemple ci-dessous, on visualise les tables de routage IPv4 et IPv6 d'un hôte dont l'interface Ethernet est nommée `eth0`.

```
$ ip route ls
default❶ via 192.0.2.1 dev eth0
192.0.2.0/26❷ dev eth0 proto❸ kernel scope link src 192.0.2.30

$ ip -6 route ls | fmt -t -w80
2001:db8:8083:c41e::/64❹ dev eth0 proto❺ kernel metric 256 expires 86049sec
    pref medium
fe80::/64❻ dev eth0 proto kernel metric 256 pref medium
default❼ via fe80::b8f1:b6ff:fee4:a0bd dev eth0 proto ra❽ metric 1024 expires
 1449sec hoplimit 64 pref medium
```

❶❼ Route par défaut.
L'entrée de table de routage qui débute par le mot clé `default` désigne la passerelle par défaut. Cette passerelle est un routeur voisin qui permet de joindre tous les réseaux inconnus de l'hôte qui émet le paquet. La plupart du temps il s'agit du routeur qui permet de joindre le reste de l'Internet.

Le mot clé `via` pointe vers l'adresse IPv4 ou IPv6 de ce routeur voisin.

Dans la table de routage IPv4, on trouve en fin de ligne le nom de l'interface à partir de laquelle les paquets sont émis vers la passerelle par défaut.

❷❹ Réseau local.
Le réseau local auquel est raccordé l'hôte est défini automatiquement lors de la configuration de l'interface Ethernet. Ces entrées de table de routage apparaissent automatiquement dès qu'une interface est configurée et active.

Dans le cas de la table de routage IPv4 uniquement, la portée de l'information est précisée avec le mot clé `scope`. Ici, le réseau `192.0.2.0/26` correspond au réseau local sur lequel l'interface `eth0` est raccordé. En fin de ligne on retrouve l'adresse IPv4 source utilisée lors de l'émission des paquets.

❸❺❽ Apprentissage des routes.
Le mot clé `proto` informe sur le mécanisme d'apprentissage de l'entrée de table de routage. Dans les deux exemples, toutes les informations proviennent du sous-système réseau du noyau Linux de l'hôte. C'est pourquoi on voit apparaître la réponse clé `kernel`.

Dans le cas de la table de routage IPv6, l'indicateur `ra` signifie Router Advertisement. Cette entrée de la table de routage a été obtenue via la configuration automatique sans état ou SLAAC. Voir IPv6.

❻ Route de lien local.
Avec le protocole IPv6, la notion de trafic de diffusion (broadcast) disparaît. C'est le trafic anycast associé au Neighbor Discovery Protocol qui permet de contacter les hôtes du voisinage réseau. Pour que les messages

ICMPv6 du protocole NDP puissent être échangés, il est nécessaire de disposer d'une route vers le réseau auquel une interface active est raccordé. Cette route utilise toujours le préfixe `fe80::/64`.

La syntaxe de l'ensemble des options de configuration d'une interface réseau est décrite dans les pages de manuels : `$ man ip-route`.

Comment changer de passerelle par défaut ?

En reprenant la table de routage affichée ci-dessus, imaginons que la passerelle par défaut IPv4 ne soit plus à l'adresse `192.0.2.1` mais à l'adresse `192.0.2.20`. Voici la syntaxe à utiliser pour réaliser ce changement.

Pour IPv6, la nouvelle passerelle par défaut est à l'adresse `fe80::20`.

```
# ip route del default

# ip route add default via 192.0.2.20

# ip -6 route del default

# ip -6 route add default via fe80::20 dev eth0
```

Comment ajouter ou supprimer une route statique ?

Imaginons que l'on veuille ajouter une entrée dans la table de routage présentée ci-dessus vers un nouveau réseau dont on connaît l'adresse. Voici la syntaxe à utiliser pour ajouter puis supprimer une entrée de table de routage avec les protocoles IPv4 et IPv6.

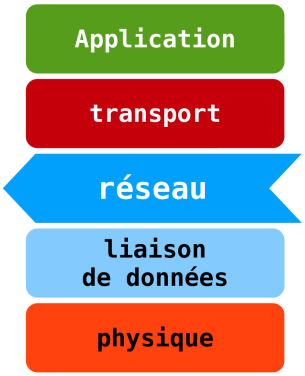
```
# ip route add 10.1.2.0/26 via 192.0.2.1

# ip route del 10.1.2.0/26

# ip -6 route add 2001:db8:2::/64 via fe80::b8f1:b6ff:fee4:a0bd dev eth0

# ip -6 route del 2001:db8:2::/64
```

7. Joindre un hôte réseau avec ICMP

Couches réseau & application

Le protocole Internet Control Message Protocol ou ICMP est décrit dans le document RFC792 Internet Control Message Protocol. C'est un protocole de la couche réseau. Comme le protocole IPv4 ne fournit aucun service de contrôle lors de la transmission des paquets sur le réseau, le rôle du protocole ICMP est d'informer l'émetteur sur les conditions de cette transmission.

Avec l'arrivée du protocole IPv6, 5 messages supplémentaires ont été ajoutés pour constituer le protocole ICMPv6. Ces 5 messages sont nécessaires à la reconnaissance du voisinage réseau IPv6 à l'aide du Neighbor Discovery Protocol.



La commande ping utilise principalement deux types de messages du protocole ICMP et fournit les informations suivantes.

- Le nombre de routeurs traversés pour joindre la destination
- Le temps de propagation aller retour (round-trip delay) lors de la communication avec l'hôte distant
- Le taux de pertes de paquets pendant la communication

Il existe 18 types de messages ICMP pour IPv4. Les deux types de messages employés par la commande ping sont les suivants.

- Le type 8 (`echo request`) est émis vers l'hôte distant.
- Le type 0 (`echo reply`) est émis par l'hôte distant en réponse.

Quelques autres types sont abordés dans la Section 10, « Lire et configurer les fonctions réseau du noyau Linux ».

Pour valider le bon fonctionnement des communications entre les adresses IP source et destination, on suit une séquence classique de tests :

1. adresse IP de l'interface de boucle locale : `lo`
2. adresse IP de la passerelle par défaut
3. adresse IP d'un hôte extérieur au réseau local

Comment savoir si un hôte distant est joignable ?

État de la pile TCP/IP

Le test suivant permet de valider les communications réseau IPv4 et IPv6 pour les processus appartenant au même système.

```
$ ping -c 2 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.320 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.320 ms

--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.320/0.320/0.320/0.000 ms

$ ping -c 2 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.034 ms

--- ::1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1013ms
rtt min/avg/max/mdev = 0.028/0.031/0.034/0.003 ms
```

Test de la passerelle par défaut

On reprend le même test avec les adresses IPv4 et IPv6 de la passerelle par défaut.

```
$ ping -c 2 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
64 bytes from 192.0.2.1: icmp_seq=1 ttl=64 time=0.759 ms
64 bytes from 192.0.2.1: icmp_seq=2 ttl=64 time=0.218 ms

--- 192.0.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1030ms
rtt min/avg/max/mdev = 0.218/0.488/0.759/0.271 ms

$ ping -c 2 fe80::b8f1:b6ff:fee4:a0bd%eth0
PING fe80::b8f1:b6ff:fee4:a0bd%eth0(fe80::b8f1:b6ff:fee4:a0bd%eth0) 56 data bytes
64 bytes from fe80::b8f1:b6ff:fee4:a0bd%eth0: icmp_seq=1 ttl=64 time=0.171 ms
64 bytes from fe80::b8f1:b6ff:fee4:a0bd%eth0: icmp_seq=2 ttl=64 time=0.239 ms

--- fe80::b8f1:b6ff:fee4:a0bd%eth0 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.171/0.205/0.239/0.034 ms
```

Tests vers des adresses extérieures au réseau local

```
$ ping -c 2 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8:❶ icmp_seq=1❷ ttl=39❸ time=23.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=39 time=22.8 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 22.801/22.921/23.041/0.120 ms

$ ping -c 2 2001:4860:4860::8888
PING 2001:4860:4860::8888(2001:4860:4860::8888) 56 data bytes
64 bytes from 2001:4860:4860::8888:❹ icmp_seq=1❺ ttl=60❻ time=40.0 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=2 ttl=60 time=39.7 ms

--- 2001:4860:4860::8888 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 39.756/39.909/40.062/0.153 ms
```

❶❹ Adresse de réponse du message ICMP : destinataire du test
❷❺ Numéro de séquence du message
❸❻ La valeur du champ TTL d'un paquet IP correspond au nombre de routeurs traversés pour arriver à destination

Comment savoir si un hôte du réseau local est joignable avec IPv4 ?

La commande ping permet de qualifier les correspondances entre les adresses de la couche liaison de données (MAC) et les adresses de la couche réseau (IPv4 ou IPv6).

Prenons l'exemple d'une table du voisinage réseau avant et après l'émission de messages ICMP vers deux adresses IPv4.

1. Voisinage réseau avant émission des messages ICMP

```
$ ip neigh ls
192.0.2.1 dev eth0 lladdr ba:f1:b6:e4:a0:bd STALE
fe80::b8f1:b6ff:fee4:a0bd dev eth0 lladdr ba:f1:b6:e4:a0:bd router DELAY
```

2. Lancement des questions ICMP

```
$ ping -c 2 192.0.2.30
PING 192.0.2.30 (192.0.2.30) 56(84) bytes of data.
64 bytes from 192.0.2.30: icmp_seq=1 ttl=64 time=0.868 ms
64 bytes from 192.0.2.30: icmp_seq=2 ttl=64 time=0.296 ms

--- 192.0.2.30 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.296/0.582/0.868/0.286 ms

$ ping -c 2 192.0.2.20
PING 192.0.2.20 (192.0.2.20) 56(84) bytes of data.
From 192.0.2.29 icmp_seq=1 Destination Host Unreachable
From 192.0.2.29 icmp_seq=2 Destination Host Unreachable

--- 192.0.2.20 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1012ms
```

3. Voisinage réseau après émission des messages ICMP

```
$ ip neigh ls
192.0.2.1 dev eth0 lladdr ba:f1:b6:e4:a0:bd STALE
192.0.2.30 dev eth0 lladdr ba:ad:ca:fe:00:1e STALE
192.0.2.20 dev eth0  FAILED
fe80::b8f1:b6ff:fee4:a0bd dev eth0 lladdr ba:f1:b6:e4:a0:bd router DELAY
```

L'expérience caractérise indirectement l'utilisation du protocole ARP. Dans le cas de l'adresse destination `192.0.2.30`, la première réponse à la requête ICMP prend un temps beaucoup plus important que la seconde : `0.868 ms` contre `0.296 ms`. La différence de temps s'explique par le recours au protocole ARP pour établir la correspondance entre l'adresse IPv4 `192.0.2.30` et sont adresse MAC `ba:ad:ca:fe:00:1e`.

Avec l'adresse destination `192.0.2.20`, le protocole ARP n'est pas parvenu à établir une correspondance entre adresse IPv4 et adresse MAC. Les requêtes ICMP n'ont donc pas pu aboutir.

Comment obtenir la liste des voisins dans un réseau local IPv6 ?

On sait que dans un réseau IPv6, la notion de trafic de diffusion n'existe pas et que c'est le protocole Neighbor Discovery Protocol qui se charge des correspondances d'adresses. Il est possible d'émettre des messages ICMPv6 de multidiffusion pour solliciter les hôtes voisins dans un réseau local. Reprenons l'exemple de la table des voisins de la section précédente avant et après les messages de sollicitation ICMPv6.

1. Voisinage réseau avant émission des messages ICMPv6

```
$ ip neigh ls
192.0.2.1 dev eth0 lladdr ba:f1:b6:e4:a0:bd STALE
192.0.2.30 dev eth0 lladdr ba:ad:ca:fe:00:1e STALE
192.0.2.20 dev eth0  FAILED
fe80::b8f1:b6ff:fee4:a0bd dev eth0 lladdr ba:f1:b6:e4:a0:bd router DELAY
```

2. Lancement des sollicitations de noeuds ICMPv6

```
$ ping -c 2 ff02::1%eth0
PING ff02::1%eth0(ff02::1%eth0) 56 data bytes
64 bytes from fe80::21e:c9ff:fef6:a2cd%eth0: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from fe80::b8f1:b6ff:fee4:a0bd%eth0: icmp_seq=1 ttl=64 time=0.394 ms (DUP!)
64 bytes from fe80::b8ad:caff:fefe:1e%eth0: icmp_seq=1 ttl=64 time=0.521 ms (DUP!)
64 bytes from fe80::21e:c9ff:fef6:a2cd%eth0: icmp_seq=2 ttl=64 time=0.031 ms

--- ff02::1%eth0 ping statistics ---
2 packets transmitted, 2 received, +2 duplicates, 0% packet loss, time 1027ms
rtt min/avg/max/mdev = 0.031/0.248/0.521/0.214 ms
```

3. Voisinage après réception des réponses ICMPv6

```
$ ip neigh ls
192.0.2.1 dev eth0 lladdr ba:f1:b6:e4:a0:bd STALE
192.0.2.30 dev eth0 lladdr ba:ad:ca:fe:00:1e STALE
192.0.2.20 dev eth0  FAILED
fe80::b8ad:caff:fefe:1e dev eth0 lladdr ba:ad:ca:fe:00:1e STALE
fe80::b8f1:b6ff:fee4:a0bd dev eth0 lladdr ba:f1:b6:e4:a0:bd router REACHABLE
```

La commande `$ ping -c 2 ff02::1%eth0` lance un recensement de tous les hôtes actifs dans le réseau local et illsutre le fonctionnement du protocole NDP.

Comment savoir si un hôte est joignable en utilisant la résolution des noms de domaines ?

La commande ping est aussi utile pour savoir si la résolution des noms d'hôtes fonctionne correctement. Dans ce cas, on fait appel à un service Internet appelé Domain Name Service (DNS). Cet appel au service DNS suppose que la fonction resolver soit correctement configurée.

```
$ ping -c 2 www.nic.fr❶
PING www.nic.fr(lb01-1.nic.fr (2001:67c:2218:30::24))❷ 56 data bytes
64 bytes from lb01-1.nic.fr (2001:67c:2218:30::24): icmp_seq=1 ttl=59 time=12.7 ms
64 bytes from lb01-1.nic.fr (2001:67c:2218:30::24): icmp_seq=2 ttl=59 time=12.6 ms

--- www.nic.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 12.632/12.675/12.719/0.120 ms
```

```
$ ping -4 -c 2 www.nic.fr❶
PING lb01-1.nic.fr (192.134.5.24)❷ 56(84) bytes of data.
64 bytes from lb01-1.nic.fr (192.134.5.24): icmp_seq=1 ttl=58 time=32.8 ms
64 bytes from lb01-1.nic.fr (192.134.5.24): icmp_seq=2 ttl=58 time=31.1 ms

--- lb01-1.nic.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 31.188/32.036/32.884/0.848 ms
```

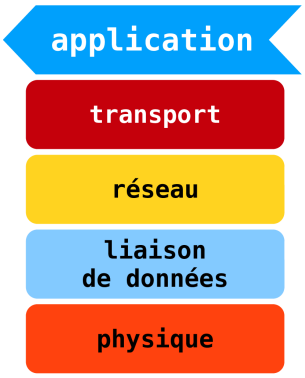
❶❶ Utilisation de la commande ping avec un nom d'hôte au lieu d'une adresse IPv4 ou IPv6. Par défaut, dès qu'une solution IPv6 est disponible, c'est ce protocole qui est utilisé.

❷❷ Affichage de la correspondance entre le nom de l'hôte et l'adresse IPv6 ou IPv4 suivant le contexte. L'utilisation de l'option `-4` avec la commande ping force la correspondance avec l'adresse IPv4.

En cas d'échec sur la résolution des noms, il faut contrôler la configuration de la partie cliente du service des noms de domaines. Cette partie est abordée dans la section suivante.

8. Lire et analyser une requête DNS

Couche application



Pour simplifier, on peut dire que le service Internet Domain Name System ou DNS fonctionne sur le même mode qu'un annuaire téléphonique dans lequel le numéro de téléphone est remplacé par l'adresse IP et le nom d'abonné est remplacé par le nom d'hôte.

DNS est un service de type client/serveur dont la fonction clé est la résolution entre des enregistrements et des adresses IP. Les enregistrements sont distribués entre les serveurs qui ont chacun autorité sur une partie de l'arborescence des noms de domaines.

Dans le contexte de ce document, on ne s'intéresse qu'à la partie cliente du service appelée resolver.

Comment visualiser la configuration du resolver ?

Généralement, la configuration du resolver d'un poste client est mise en place automatiquement grâce à des services tels que DHCP (Dynamic Host Configuration Protocol) ou RDNSS (Recursive DNS Server) et DNSSL (DNS Search List).

Sur les systèmes GNU/Linux, il existe des paquets tels que resolvconf qui gèrent dynamiquement la configuration du resolver en choisissant les paramètres en fonction des services d'autoconfiguration disponibles.

Voici une description des fichiers de configuration système qui jouent un rôle dans la résolution des noms de domaines. La liste des sources d'information puis l'ordre dans lequel on consulte ces sources sont les facteurs les plus importants.

`/etc/resolv.conf`

Le rôle de ce fichier est de désigner le serveur DNS qui doit prendre en charge les requêtes du système. Tout programme qui fait référence à un nom d'hôte sollicite cette ressource.

```
$ 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 192.0.2.1
```

Ici l'adresse IP du serveur DNS est 192.0.2.1. Dans un contexte domestique, on retrouve les mêmes informations via l'interface Web d'une «box ADSL».

`/etc/nsswitch.conf`

Le rôle du Name Service Switch dépasse le cadre de la simple résolution des noms d'hôtes. Tous les programmes font appel à la bibliothèque standard glibc. Lors des appels à cette bibliothèque, ce fichier est consulté pour connaître la liste des sources à utiliser.

```
$ grep ^hosts /etc/nsswitch.conf
hosts:          files mdns_minimal [NOTFOUND=return] dns mdns
```

Dans l'exemple ci-dessus, la scrutation des sources débute avec les fichiers locaux, la version minimale du service multicast DNS, le service DNS tel que configuré dans le fichier ci-dessus et enfin le service multicast DNS.

La syntaxe '`[NOTFOUND=return]`', implique que si l'un des deux services qui suivent dans la liste déclare que l'hôte est introuvable, la recherche s'arrête là.

`/etc/host.conf`

Ce dernier fichier est présent pour des raisons de compatibilité avec les anciennes versions de la bibliothèque standard.

```
$ cat /etc/host.conf
multi on
```

Comment analyser les résultats d'une requête DNS ?

Sur un système GNU/Linux, les deux commandes de référence sont dig et host. Elles servent à qualifier le bon fonctionnement du resolver sur le système en isolant le service DNS des autres traitements. Voici quelques exemples de requêtes.

Résolution complète d'un nom d'hôte

La question posée est : quelle sont les enregistrements DNS relatifs au nom de serveur Web www.nic.fr ?

```
$ host www.nic.fr
www.nic.fr is an alias for lb01-1.nic.fr.
lb01-1.nic.fr has address 192.134.5.24
lb01-1.nic.fr has IPv6 address 2001:67c:2218:30::24
```

Cet exemple illustre le fait que plusieurs questions ont été posées via la commande host puisque les adresses IPv4 et IPv6 ont été obtenues en retour.

Résolution simple d'un nom d'hôte

La question posée est : quelle est l'adresse IPv4 correspondant au nom de serveur Web www.nic.fr ?

```
$ dig +short www.nic.fr
lb01-1.nic.fr.
192.134.5.24
```

Résolution inverse d'une adresse IP

La question posée est : quel est le nom d'hôte correspondant à l'adresse IPv4 `192.134.5.24` ?

```
$ dig +short -x 192.134.5.24
lb01-1.nic.fr.
```

Requête DNS complète sur un nom d'hôte

```
$ dig www.iana.org

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> www.iana.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60063
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:❶
;www.iana.org.                  IN      A

;; ANSWER SECTION:❷
www.iana.org.           600     IN      CNAME   ianawww.vip.icann.org.
ianawww.vip.icann.org.  30      IN      A       192.0.32.8

;; AUTHORITY SECTION:❸
vip.icann.org.          3600    IN      NS      gtm1.dc.icann.org.
vip.icann.org.          3600    IN      NS      gtm1.lax.icann.org.

;; ADDITIONAL SECTION:❹
gtm1.dc.icann.org.      21600   IN      A       192.0.47.252
gtm1.dc.icann.org.      21600   IN      AAAA    2620:0:2830:296::252
gtm1.lax.icann.org.     21600   IN      A       192.0.32.252
gtm1.lax.icann.org.     21600   IN      AAAA    2620:0:2d0:296::252

;; Query time: 562 msec❺
;; SERVER: 192.0.2.1#53(192.0.2.1)❻
;; WHEN: Thu Jan 30 10:37:46 2014
;; MSG SIZE  rcvd: 211
```

❶ Le champ QUESTION reprend les termes de la requête DNS soumise au serveur.
❷ Le champ ANSWER liste les éléments de réponse à la requête. Ici, le nom d'hôte `www.iana.org` est en fait un alias de `ianawww.vip.icann.org`. Cet alias a pour adresse IP : `192.0.32.8`.
❸ Le champ AUTHORITY donne la liste des serveurs de noms qui ont autorité sur les enregistrements DNS. Ce sont les seuls serveurs aptes à fournir une réponse aux requêtes sur le domaine concerné.
❹ Le champ ADDITIONAL donne les adresses IP des serveurs DNS de référence du domaine.
❺ Le champ Query time donne le temps de traitement de la requête. La valeur obtenue permet de déduire si le serveur interrogé a déjà la réponse en mémoire cache ou non.
❻ Le champ SERVER identifie le serveur qui a pris la requête DNS en charge.

Pour aller plus loin dans l'étude du fonctionnement du service de noms de domaines, il est conseillé de lire le support Introduction au service DNS.

9. Tracer le chemin suivi par le trafic réseau

Couches réseau & application

Si la commande ping du protocole ICMP permet d'obtenir des informations l'état de l'hôte destination, elle ne permet pas de tracer le chemin suivi par les paquets IP. C'est justement l'objectif du service traceroute dont le principe est le suivant :

- La source émet un premier message avec la valeur 1 dans le champ TTL de l'en-tête IP.
- Le routeur qui reçoit ce message décrémente la valeur du champ TTL de l'en-tête IP et obtient 0. Il jette donc le message et émet un message ICMP à destination de l'émetteur indiquant qu'il est impossible d'atteindre la destination.
- La source émet un deuxième message avec la valeur 2 dans le champ TTL de l'en-tête IP.
- Cette fois-ci, c'est le deuxième routeur qui décrémente la valeur du champ TTL et obtient 0. C'est donc à lui d'émettre un message ICMP indiquant qu'il est impossible d'atteindre la destination.
- Ainsi de suite avec les valeurs du champ TTL de l'en-tête IP 3, 4, 5, etc.

Avertissement

Pour des raisons de sécurité, il peut être nécessaire de cacher le chemin suivi par le trafic utilisateur. C'est la raison pour laquelle les résultats obtenus varient énormément suivant les contextes d'interconnexion réseau. Il devient de plus en plus difficile d'obtenir une information correcte.

Pour illustrer le fonctionnement du service, on peut utiliser la commande mtr fournie par la paquet mtr-tiny. Cette commande possède de nombreuses options et fournit une présentation dynamique des résultats. Voici deux exemples qui illustrent la «dispersion» des résultats :

Exemple de rapport basé sur ICMP echo

```
$ mtr -4 -c 10 --report www.nic.fr
Start: 2018-01-08T14:55:31+0100
HOST: inetdoc-ttn                 Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- h7.tetaneutral.net         0.0%    10    0.3   0.3   0.3   0.3   0.0
  2.|-- te0-0-2-3.rcr11.tls01.atl  0.0%    10    1.1   1.4   1.0   3.6   0.8
  3.|-- te0-2-1-2.rcr21.bod01.atl  0.0%    10    4.1   4.2   4.1   4.4   0.1
  4.|-- be2840.rcr21.eas02.atlas.  0.0%    10    7.8   7.8   7.7   7.9   0.1
  5.|-- be2839.ccr52.bio02.atlas.  0.0%    10    8.8   8.8   8.7   8.8   0.1
  6.|-- be3358.ccr32.mad05.atlas.  0.0%    10   13.6  13.5  13.3  13.7   0.2
  7.|-- ae-20.r01.mdrdsp03.es.bb.  0.0%    10   13.6  13.6  13.4  14.3   0.2
  8.|-- ae-7.r24.londen12.uk.bb.g  0.0%    10   30.4  30.5  30.3  31.3   0.4
  9.|-- ae-1.r25.londen12.uk.bb.g  0.0%    10   30.4  30.4  30.3  30.9   0.2
 10.|-- ae-2.r04.parsfr01.fr.bb.g  0.0%    10   30.6  30.6  30.5  30.8   0.1
 11.|-- ae-5.r03.parsfr02.fr.bb.g  0.0%    10   30.7  30.7  30.6  30.9   0.1
 12.|-- ae-8.r02.parsfr02.fr.bb.g  0.0%    10   30.6  30.8  30.6  31.0   0.2
 13.|-- 82.112.96.178              0.0%    10   31.1  31.0  30.9  31.1   0.1
 14.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
 15.|-- isg-th3.interco.nic.fr     0.0%    10   30.3  30.3  30.3  30.4   0.1
 16.|-- lb01-1.nic.fr              0.0%    10   30.5  30.4  30.4  30.6   0.1
```

Exemple de rapport basé sur UDP

```
$ mtr -4 -u -c 10 --report www.nic.fr
Start: 2018-01-08T14:57:32+0100
HOST: inetdoc-ttn                 Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- h7.tetaneutral.net         0.0%    10    0.3   0.3   0.3   0.3   0.0
  2.|-- te0-0-2-3.rcr11.tls01.atl  0.0%    10    1.1   1.1   1.0   1.2   0.1
  3.|-- te0-2-1-2.rcr21.bod01.atl  0.0%    10    4.2   4.3   4.2   4.4   0.1
  4.|-- be2840.rcr21.eas02.atlas.  0.0%    10    7.8   7.8   7.7   8.0   0.1
  5.|-- be2838.ccr51.bio02.atlas.  0.0%    10    8.8   8.8   8.6   9.2   0.2
  6.|-- be3357.ccr31.mad05.atlas.  0.0%    10   13.6  13.5  13.2  13.7   0.1
  7.|-- ae-4.r01.mdrdsp03.es.bb.g  0.0%    10   14.3  13.6  13.4  14.3   0.3
  8.|-- ae-7.r24.londen12.uk.bb.g  0.0%    10   30.3  28.2  27.3  30.3   1.1
  9.|-- ae-1.r25.londen12.uk.bb.g  0.0%    10   27.3  28.8  27.3  31.5   1.4
 10.|-- ae-2.r04.parsfr01.fr.bb.g  0.0%    10   27.7  28.6  27.7  29.9   0.8
 11.|-- ae-5.r03.parsfr02.fr.bb.g  0.0%    10   28.5  28.5  27.1  31.2   1.5
 12.|-- ae-8.r02.parsfr02.fr.bb.g  0.0%    10   27.8  28.4  27.0  30.7   1.0
 13.|-- 82.112.96.178              0.0%    10   31.1  29.1  27.6  31.1   1.1
 14.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
```

La comparaison entre les deux rapports montre que le protocole ICMP subit un filtrage important relativement aux requêtes UDP. Les très nombreuses attaques de type «déni de service distribué» basées sur ICMP ont nécessité la mise en place de protections qui entraînent quelques désagréments dans les tests de fonctionnement des réseaux.

Pour aller plus loin dans les manipulations sur le tracé de route, il existe d'autres outils intéressants tels que tracepath fourni par le paquet iputils-tracepath.

10. Lire et configurer les fonctions réseau du noyau Linux

Un grand nombre de paramètres sont actifs par défaut sur les interfaces réseau. Sur un système GNU/Linux, ces paramètres sont placés dans les systèmes de fichiers virtuels `/proc` et `/sys`.

Comment visualiser les paramètres du noyau Linux pour une interface Ethernet ?

Tous les réglages possibles pour une interface pilotée par le sous-système réseau du noyau Linux sont accessibles depuis l'espace utilisateur via deux systèmes de fichiers virtuels appelés `procfs` et `sysfs`. L'outil qui permet de lire et configurer ces réglages s'appelle sysctl.

La description de tous les paramètres relatifs à l'interface Ethernet sort du cadre de ce document. Le but ici est de montrer que ces paramètres existent, qu'ils sont accessibles et que l'on sait où les trouver dans l'arborescence système.

Dans notre contexte, nous savons que le nom de l'interface Ethernet est `eth0`. On peut faire une recherche des paramètres relatifs à ce nom d'interface pour les protocoles IPv4 et IPv6.

```
$ sudo sysctl -a --pattern 'net.ipv(4|6).conf.eth0'
```

Comment changer la valeur d'un paramètre ?

Pour changer les valeurs attribuées par défaut lors de l'initialisation du système, on utilise à nouveau la commande sysctl. Le répertoire `/etc/sysctl.d` contient les fichiers de modification ou d'application de nouveaux paramètres. Historiquement, c'est le fichier `/etc/sysctl.conf` qui contenait la liste de ces modifications. Aujourd'hui, ce fichier est fourni par la distribution et il est préférable d'ajouter un fichier dédié dans le répertoire `/etc/sysctl.d`.

Pour ce qui est de la liste des paramètres d'une interface réseau, on peut prendre l'exemple du routage par l'adresse source comme axe de modification des valeurs par défaut. Le source routing est un mécanisme qui permet à un paquet IP d'indiquer au routeur le chemin que doit suivre le paquet retour. Il existe également une option pour enregistrer les sauts tout au long du chemin. La liste des sauts effectués ou l'enregistrement de route fournit à la destination un chemin de retour vers la source. Cela permet à la source (l'hôte émetteur) de spécifier la route, de manière lâche ou stricte, en ignorant les tables de routage de tout ou partie des routeurs. Il peut permettre à un utilisateur de rediriger le trafic réseau à des fins malveillantes. Par conséquent, le routage basé sur la source doit être désactivé.

Avant modification des paramètres, la situation est la suivante.

```
$ sudo sysctl -a --pattern 'net.ipv(4|6).conf.eth0.*source'
net.ipv4.conf.eth0.accept_source_route = 1
net.ipv6.conf.eth0.accept_source_route = 0
```

1. Dans un premier temps, il est possible de modifier le paramètre voulu individuellement.

   ```
   $ sudo sysctl -w net.ipv4.conf.eth0.accept_source_route=0
   net.ipv4.conf.eth0.accept_source_route = 0
   ```

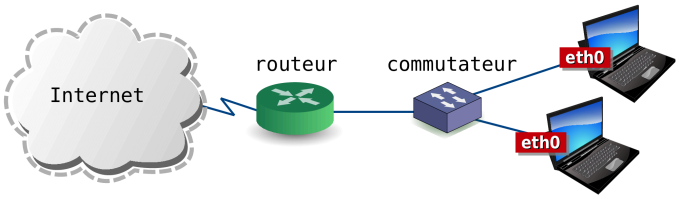
2. Dans un second temps, il est possible de rendre cette modification permanente au niveau système en éditant le fichier historique `/etc/sysctl.conf`. Il faut décommenter la ligne relative au paramètre sur le routage basé sur la source pour IPv4.

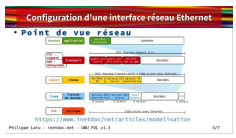
   ```
   $ sudo sed -ie '/#net.ipv4.conf.all.accept_source_route = 0/ s/^#//' /etc/sysctl.conf
   $ sudo sysctl --system
   * Applying /etc/sysctl.d/99-sysctl.conf ...
   net.ipv4.conf.all.accept_source_route = 0
   * Applying /etc/sysctl.d/protect-links.conf ...
   fs.protected_hardlinks = 1
   fs.protected_symlinks = 1
   * Applying /etc/sysctl.conf ...
   net.ipv4.conf.all.accept_source_route = 0
   ```

Comme le fichier `/etc/sysctl.conf` est conservé, les paramètres seront à nouveau appliqués lors de l'initialisation du système.

11. Travaux pratiques

Pour traiter les questions de cette section, on suppose que le poste client dispose d'une interface Ethernet déjà configurée avec un accès à un réseau local puis à l'Internet via une passerelle par défaut. La topologie type est la suivante :





Capture vidéo : Approche par l'exemple - 05:40

Questions sur l'identification de l'interface Ethernet

Q1. Quelles sont les informations disponibles sur le composant contrôleur Ethernet et son pilote logiciel sur votre système ?

Utiliser les outils présentés dans la Section 2, « Identifier les ressources matérielles » pour obtenir la référence et l'adresse du composant contrôleur Ethernet ainsi que le nom du module utilisé pour son pilotage.

Attention ! Le logiciel de pilotage du contrôleur Ethernet peut avoir été intégré à la partie monolithique du noyau Linux. Dans ce cas, aucun module ne correspond au pilotage de l'interface réseau et il faut consulter les messages système pour retrouver la trace de cette interface.

Q2. Quelles sont les informations sur le média de raccordement physique au réseau local Ethernet ?

Utiliser les outils présentés dans la Section 2, « Identifier les ressources matérielles » pour obtenir les informations sur le type de média utilisé et le débit binaire entre l'hôte et le commutateur. Préciser le mode de transmission full-duplex ou half-duplex.

Q3. Quelles sont les informations sur l'état de l'interface Ethernet ?

Utiliser les informations sur les indicateurs d'état présentés dans la Section 3, « Lire et configurer l'état d'une interface ».

Capture vidéo : Identifier les ressources matérielles et les indicateurs d'état d'une interface - 09:56

Questions sur l'adressage de l'interface Ethernet

Q4. Quelles sont les différentes adresses affectées à l'interface Ethernet de votre système ?

Utiliser les commandes de visualisation présentées dans la Section 3, « Lire et configurer l'état d'une interface » et la Section 5, « Lire et configurer les adresses réseau d'une interface ».

Q5. Quel est le rôle de chacune de ces adresses ?

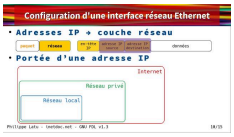
Le rôle d'une adresse est essentiellement lié à sa portée vis à vis des autres réseaux.

Q6. Quelles sont les adresses des réseaux IPv4 et IPv6 associées à cette interface ?

À l'aide des commandes présentées dans la Section 5, « Lire et configurer les adresses réseau d'une interface », retrouver les limites de l'espace d'adressage de chaque réseau.

On peut utiliser le document Adressage IPv4 pour déterminer l'adresse du réseau IPv4.

Capture vidéo : Repérer les différentes adresses et leur portée - 11:40



Questions sur le voisinage réseau de l'interface Ethernet

Q7. Quelles sont les correspondances d'adresses données par la table du voisinage pour les réseaux IPv4 et IPv6 ? Quels sont les deux protocoles utilisés pour obtenir ces correspondances entre adresses MAC et adresses réseau ?

Reprendre les informations proposées à la Section 4, « Reconnaître le voisinage réseau ».

Attention ! Suivant l'activité entre les hôtes du réseau local ou la passerelle par défaut, le nombre d'entrée de la table du voisinage réseau peut beaucoup varier.

Q8. Comment provoquer l'ajout d'une ou plusieurs entrées IPv4 et IPv6 dans la table des voisins ?

Utiliser les instructions proposées dans la Section 7, « Joindre un hôte réseau avec ICMP » pour tester l'accessibilité de différentes adresses IP.

Q9. Est-ce que l'adresse IPv4 ou IPv6 d'un voisin peut appartenir à un réseau extérieur au réseau local ?

Reprendre les informations proposées à la Section 4, « Reconnaître le voisinage réseau ».

Q10. Pourquoi des entrées apparaissent dans la table des voisins sans trafic initié depuis votre interface Ethernet ?

Repérer les différences entre les hôtes contactés depuis votre système et les hôtes qui ont contacté votre système.

Q11. Est-il possible de déduire l'adresse d'une passerelle par défaut à partir des informations fournies par la table des voisins ?

Dans le cas du protocole IPv6 la réponse se lit directement si une passerelle est présente dans la table des voisins. Pour IPv4, il est possible de déduire l'adresse d'une passerelle à partir de la fréquence de sollicitation d'une entrée par rapport aux autres.

Questions sur la lecture d'une table de routage simple

Q12. Combien y-a-t-il d'entrées dans la table de routage ? Quel est le rôle de chacune de ces entrées ?

Faire la correspondance entre la table de routage de votre système et l'exemple donné à la Section 6, « Lire une table de routage simple et changer la passerelle par défaut ».

Q13. Quel est le rôle de la passerelle par défaut dans l'acheminement du trafic de votre système ?

Identifier les destinations desservies par cette passerelle par défaut relativement aux autres entrées des tables de routage IPv4 et IPv6.

Q14. La passerelle par défaut peut-elle appartenir à un réseau extérieur au réseau local ?

Et si c'était le cas ? Comment acheminer les paquets émis par votre système ?

Questions sur la résolution des noms de domaine

Q15. Quelles sont les adresses IPv4 ou IPv6 des serveurs DNS données dans le fichier de configuration de votre système ?

Retrouver le fichier de configuration du client DNS dans la Section 8, « Lire et analyser une requête DNS » et consulter son contenu.

Q16. Comment poser une requête DNS individuelle ? Donner un exemple ?

Utiliser les exemples donnés dans la Section 8, « Lire et analyser une requête DNS ».

Q17. Comment interpréter le temps de réponse d'une requête DNS émise avec la commande dig ?

Rechercher les informations dans l'exemple d'exécution de la commande à la Section 8, « Lire et analyser une requête DNS ».

Questions sur le chemin suivi par le trafic réseau

Q18. Quel est le rôle du service traceroute relativement au protocole ICMP ?

(Re)Lire le début de la Section 9, « Tracer le chemin suivi par le trafic réseau ».

Q19. À partir d'un exemple, dessiner pour chaque routeur traversé les valeurs du champ TTL de l'en-tête IP.

Reprendre l'exemple donné dans la Section 9, « Tracer le chemin suivi par le trafic réseau » et faire la correspondance avec les informations relevées.

Q20. Dans quelles conditions les informations renvoyées par les routeurs sont incomplètes ?

(Re)Lire l'avertissement donné dans la Section 9, « Tracer le chemin suivi par le trafic réseau ».