Configuration d'une interface réseau Ethernet

Philippe Latu
philippe.latu(at)inetdoc.net

http://www.inetdoc.net

Résumé

Ce support de travaux pratiques traite de la configuration d'une interface réseau Ethernet sur un système GNU/Linux. Les manipulations présentées suivent la modélisation réseau en remontant du niveau physique jusqu'à la validation des services de la couche application. Les questions cherchent à illustrer les relations entre les différents formats d'adressage utilisés à chaque niveau ainsi que les protocoles utilisés pour la «correspondance» entre ces formats.

Table des matières

1. Copyright et Licence

Méta-information

Cet article est écrit avec DocBook[1] XML sur un système Debian GNU/Linux[2]. Il est disponible en version imprimable au format PDF : `conf-intf-lan.pdf`[3].

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

[1] http://www.docbook.org
[2] http://www.debian.org
[3] http://www.inetdoc.net/pdf/conf-intf-lan.pdf

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. Identification d'une interface - 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 composant 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 Intel™ x86_64, un composant Ethernet est nécessairement 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
<snip/>
07:00.0 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
        Subsystem: Intel Corporation Gigabit ET Quad Port Server Adapter
        Flags: bus master, fast devsel, latency 0, IRQ 55
        Memory at ddf40000 (32-bit, non-prefetchable) [size=128K]
        Memory at de000000 (32-bit, non-prefetchable) [size=4M]
        I/O ports at dcc0 [size=32]
        Memory at ddf38000 (32-bit, non-prefetchable) [size=16K]
        Capabilities: <access denied>
        Kernel driver in use: igb
```

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

```
$ lsmod | grep igb
igb                   138336  0
i2c_algo_bit           12751  1 igb
i2c_core               24092  2 igb,i2c_algo_bit
dca                    13168  2 igb,ioatdma
ptp                    17460  1 igb
```

Contrôleur Ethernet sur bus USB

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

```
$ dmesg | grep -i Ethernet
[    3.114563] smsc95xx 1-1.1:1.0 eth0: register 'smsc95xx' at usb-bcm2708_usb-1.1, \
               smsc95xx USB 2.0 Ethernet, b8:27:eb:ea:29:72
```

Répertoire /sys/

Pour compléter les informations sur les interfaces réseau, il est possible de parcourir l'arborescence `/sys/`. Ainsi, dans le cas d'une architecture Raspberry Pi, on obtient les éléments suivants :

```
$ ls -go /sys/class/net/eth0/device/driver/❶
total 0
lrwxrwxrwx 1    0 janv. 27 09:18 1-1.1:1.0 -> \
            ../../../../devices/platform/bcm2708_usb/usb1/1-1/1-1.1/1-1.1:1.0❷
--w------- 1 4096 janv. 27 09:18 bind
lrwxrwxrwx 1    0 janv. 27 09:18 module -> \
            ../../../../module/smsc95xx❸
-rw-r--r-- 1 4096 janv. 27 09:18 new_id
-rw-r--r-- 1 4096 janv. 27 09:18 remove_id
--w------- 1 4096 janv. 27 09:18 uevent
--w------- 1 4096 janv. 27 09:18 unbind
```

❶ La dénomination `eth0` désigne la première interface Ethernet présente sur le système.
❷ Dans le répertoire relatif à l'interface, on repère la position du contrôleur Ethernet sur le bus USB.
❸ Dans le même répertoire, on identifie aussi le module de pilotage du contrôleur.

Pour résumer, les outils utiles pour l'identification des composants réseau et des modules logiciels asscié sont : **lspci**, **dmesg**, **lsmod** et le parcours de l'arborescence `/sys/`.

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 lien physique entre l'hôte et l'équipement réseau n'est pas actif. Sur les liaisons utilisant des 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 mêmes exemples que ci-dessus, on retrouve les informations suivantes :

Interface Intel™ Gbps

```
#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: Symmetric
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: Symmetric
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s ❷
        Duplex: Full
        Port: Twisted Pair ❸
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on ❹
        MDI-X: off (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 : 1000Mbps en mode Full-Duplex, 100Mbps en mode Full-Duplex, 100Mbps en mode Half-Duplex, 10Mbps en mode Full-Duplex et 10Mbps en mode Half-Duplex.
❷ Le lien entre l'interface `eth0` et l'équipement réseau est actif et le débit négocié est de 1000Mbps 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 connectée.

Interface Raspberry Pi 100Mbps

```
$ 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: Symmetric
        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 à la copie d'écran ci-dessus, le débit passe à 100Mb/s. C'est le principal changement observé.

Pour aller plus loin dans l'étude des caractéristiques techniques des réseaux locaux, il est conseillé de lire l'article Technologie Ethernet[4].

[4] http://www.inetdoc.net/articles/ethernet/

3. Configuration d'une interface - couche liaison

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, on peut définir la quantité de données à encapsuler dans une trame. C'est aussi à ce niveau que l'on définit les champs du standard IEEE 802.1q. Cette dernière fonction sort du cadre de ce document (voir l'article Routage Inter-VLAN[5]). Les fonctions relatives à la gestion de mise en file d'attente des paquets issus de la couche réseau sortent aussi du cadre de ce document (voir HOWTO du routage avancé et du contrôle de trafic sous Linux[6]).

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
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP>❶ mtu 1500❷ qdisc pfifo_fast state UP \
                                                mode DEFAULT qlen 1000
    link/ether b8:27:eb:ea:29:72❸ brd ff:ff:ff:ff:ff:ff
```

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[7].

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.
`LOWER_UP`	Un «câble» est correctement connecté à l'interface.
`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.

[5] http://www.inetdoc.net/articles/inter-vlan-routing/
[6] http://www.inetdoc.net/guides/lartc/
[7] http://www.inetdoc.net/articles/inter-vlan-routing/inter-vlan-routing.lan.html#inter-vlan-routing.lan.mac-address

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 mq state UP \
                                                mode DEFAULT group default qlen 1000
    link/ether 00:1e:c9:f6:a2:cd brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    3535504061 4585598  0       0       0       43709
    TX: bytes  packets  errors  dropped carrier collsns
    3066244859 4103684  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 média de transmission réservé pour l'émission et d'un second média réservé pour la réception.

Comment activer/désactiver une interface réseau ?

Ces opérations peuvent s'effectuer à deux niveaux bien distincts : interface et/ou système.

Au niveau système, 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. Si une interface est paramétrée pour utiliser le client DHCP par exemple, les scripts se chargent du lancement et de l'arrêt du programme dhclient.

À l'inverse, 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 fonctions associées à une interface avant de se lancer dans les manipulations directes d'interfaces.

Désactivation au niveau interface

```
# ip link set dev eth0 down

# ip addr ls dev eth0
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN \
                                    group default qlen 1000
    link/ether ba:ad:00:ca:fe:b9 brd ff:ff:ff:ff:ff:ff
```


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 interface

```
# ip link set dev eth0 up

# ip addr ls dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP \
                                                group default qlen 1000
    link/ether ba:ad:00:ca:fe:b9 brd ff:ff:ff:ff:ff:ff
```

Activation au niveau système

```
# ifup eth0
```

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

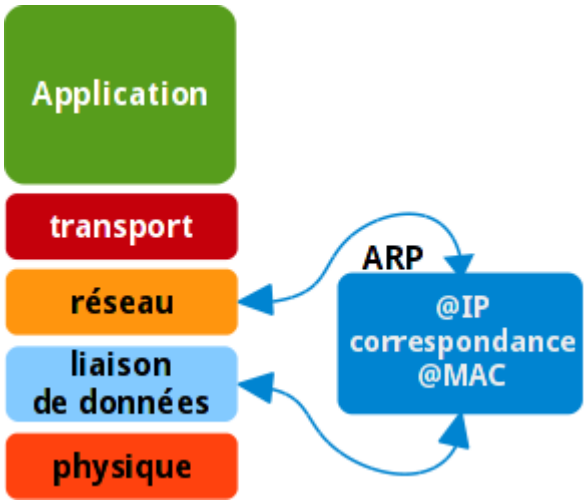
Parmi les nombreuses manipulations possibles avec la commande **ip link**, il est possible de changer l'adresse MAC d'une interface. Voici un exemple.

```
# ip link set dev eth0 down

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

# ip link ls dev eth0
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. Protocole ARP

Le protocole ARP ou Address Resolution Protocol a pour but de chercher à faire correspondre une adresse MAC inconnue (celle de l'hôte destinataire) avec une adresse IP connue (encore celle de l'hôte destinataire). Ce protocole assure la «liaison» entre les mécanismes d'adressage de la couche réseau (IP) et de la couche liaison de données (MAC).



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 ARP ?

Le sous-système réseau maintient une table de correspondance entre les adresses IP et les adresses MAC appelée cache ARP avec le protocole IPv4. Dans le contexte d'utilisation d'une double pile IPv4 et IPv6, on parle de la «table des hôtes voisins». Voici un exemple de consultation de cette table.

```
$ ip nei ls dev eth0
2001:db8:fe00:814f:226:18ff:fe27:754 dev eth0 lladdr 00:26:18:27:07:54 REACHABLE
fe80::ba27:ebff:feea:2972 dev eth0 lladdr b8:27:eb:ea:29:72 router STALE
fe80::226:18ff:fe27:754 dev eth0 lladdr 00:26:18:27:07:54 REACHABLE
2001:db8:fe00:814f::1 dev eth0 lladdr b8:27:eb:ea:29:72 router STALE
192.168.1.2 dev eth0 lladdr d4:8c:b5:9c:8e:60 REACHABLE
192.168.1.1 dev eth0 lladdr b8:27:eb:ea:29:72 STALE
192.168.1.4 dev eth0 lladdr 00:02:72:88:c5:9c REACHABLE
```

ARP vs NDP

Attention ! Le protocole ARP ne fonctionne qu'avec IPv4. Un nouveau protocole de correspondance plus sophistiqué a été introduit avec IPv6 : Network Discovery Protocol. Si la commande **`ip nei ls`** fait apparaître les résultats dans un affichage commun, ces résultats sont obtenus via deux protocoles distincts. La présentation du protocole NDP sort du cadre de ce document.

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
`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.

Indicateur	Description
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 ARP ?

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. Configuration 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[8]).

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

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

Dans l'exemple ci-dessous, l'adresse IPv4 de l'interface Ethernet correspond à l'hôte numéro `1` du réseau `192.168.1.0/24` dont le masque réseau développé est `255.255.255.0`.

```
$ ip addr ls
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether b8:27:eb:ea:29:72 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2001:db8:fe00:814f::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::ba27:ebff:feea:2972/64 scope link
       valid_lft forever preferred_lft forever
```

Cet exemple caractérise bien le fait qu'une même interface peut être configurée avec de multiples adresses. Chacune de ces adresses a une portée propre. Une adresse MAC est visible uniquement dans le réseau local (LAN) tandis qu'une adresse IP est visible à l'échelle d'une interconnexion de plusieurs réseaux.

Comment ajouter ou supprimer une adresse à une interface ?

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
```

Avertissement

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

Comment rendre la configuration permanente ?

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 :

[8] http://www.inetdoc.net/articles/adressage.ipv4/

```
# /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
```

Pour une configuration statique de l'interface, il faut utiliser les pages de manuels : **`$ man interfaces`**. Voici un exemple :

```
<snip/>
auto eth0
iface eth0 inet static
        address 192.168.1.1
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
```

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

6. Table de routage simple - couche réseau

Le routage est la 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 sont basées sur une table de routage.

Cela peut paraître surprenant, mais tout hôte disposant d'un sous-système réseau dans son 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 la table de routage d'un hôte dont l'interface Ethernet est nommée `eth0`.

```
$ ip route ls
default via 192.168.3.250 dev eth0  proto static
192.168.3.0/24 dev eth0  proto kernel  scope link  src 192.168.3.221
169.254.0.0/16 dev eth0  scope link  metric 1000
```

L'analyse de chacune des trois ligne donne les informations suivantes.

1ère ligne : route par défaut
: Le mot clé '`default`' désigne toutes les destinations non répertoriées dans les autres entrées de la table de routage ; autrement dit, tous les réseaux inconnus du système. Le mot clé '`via`' pointe vers l'adresse IP du routeur qui doit acheminer le trafic vers l'Internet. Ici, l'adresse `192.168.3.250` correspond à un routeur que l'on appelle communément la passerelle par défaut. Ainsi, tout le trafic émis par le système à destination de l'Internet doit être routé par cette passerelle. De plus, elle doit impérativement appartenir au réseau local sur lequel le système est raccordé.

2ème ligne : réseau local
: Cette entrée désigne le réseau local sur lequel le système est directement raccordé. Ce réseau a pour adresse `192.168.3.0/24` et c'est le noyau du système qui a inséré cette entrée dans la table de routage. La notation '`scope link`' indique que cette entrée n'est valide qu'à l'échelle de ce système. L'adresse IP source de l'interface Ethernet est `192.168.3.221`. Dans l'exemple étudié, les adresses IP de la passerelle par défaut et de l'interface Ethernet de l'hôte appartiennent bien toutes les deux au réseau `192.168.3.0/24`.

3ème ligne : réseau de lien local
: L'adresse réseau `169.254.0.0/16` est définie dans le standard RFC3927 Dynamic Configuration of IPv4 Link-Local Addresses[9]. En cas d'échec de configuration, il est possible d'attribuer automatiquement une adresse IP de ce préfixe à une interface. Le mot clé '`metric`' suivi de la valeur `1000` indique que cette entrée est une route de «dernier recours». Plus la métrique d'une route est élevée, moins elle est prioritaire.

Comment changer de passerelle par défaut ?

En reprenant la table de routage affichée ci-dessus, imaginons que la passerelle par défaut ne soit plus à l'adresse `192.168.3.250` mais à l'adresse `192.168.3.1`. Voici la syntaxe à utiliser pour réaliser ce changement.

[9] https://www.rfc-editor.org/rfc/rfc3927.txt

```
# ip route del default

# ip route add default via 192.168.3.1
```

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 ce réseau.

```
# ip route add 10.1.2.0/26 via 192.168.3.1

# ip route del 10.1.2.0/26
```

Pour aller plus loin dans les manipulations sur les routes au niveau réseau, la consultation des pages de manuels est un excellent point de départ : **`$ man ip-route`**.

7. Protocole ICMP - couche réseau

Le protocole Internet Control Message Protocol ou ICMP est décrit dans le document RFC792 Internet Control Message Protocol[10]. C'est un protocole de la couche réseau. Comme le protocole IP 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.

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. Les deux types de messages employés par la commande **ping** sont :

- 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, « 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 l'interface du poste de travail : `eth0`
3. adresse IP du destinataire de la passerelle par défaut
4. adresse IP extérieure au réseau local

Comment valider une interconnexion réseau ?

État de la pile TCP/IP
: Le test suivant permet de valider les communications réseau 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
```

Test de l'interface Ethernet
: On reprend le même test avec l'adresse IP de l'interface.

```
$ ping -c 2 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_req=1 ttl=64 time=0.626 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=64 time=0.269 ms

--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.269/0.447/0.626/0.179 ms
```

[10] https://www.rfc-editor.org/rfc/rfc792.txt

Tests vers des hôtes du réseau local

Exemple d'échec :

```
$ ping -c 2 192.168.1.14
PING 192.168.1.14 (192.168.1.14) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Destination Host Unreachable
From 192.168.1.1 icmp_seq=2 Destination Host Unreachable

--- 192.168.1.14 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1007ms
```

Exemple de succès :

```
$ ping -c 2 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2:❶ icmp_req=1❷ ttl=255❸ time=1.61 ms
64 bytes from 192.168.1.2: icmp_req=2 ttl=255 time=1.16 ms

--- 192.168.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.161/1.388/1.616/0.230 ms
```

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

Comment valider la résolution des noms de domaines en plus de l'interconnexion réseau ?

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 web.nic.fr (192.134.4.20)❷ 56(84) bytes of data.
64 bytes from web.nic.fr (192.134.4.20): icmp_req=1 ttl=54 time=39.1 ms
64 bytes from web.nic.fr (192.134.4.20): icmp_req=2 ttl=54 time=34.7 ms

--- web.nic.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 34.754/36.960/39.167/2.214 ms
```

❶ Utilisation de la commande **ping** avec un nom d'hôte au lieu d'une adresse IP.
❷ Affichage de la correspondance entre le nom de l'hôte et son adresse IP.

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. Protocole 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). Il existe même un paquet appelé resolvconf qui améliore la gestion de la configuration du resolver en choisissant les paramètres en fonction des services d'autoconfiguration disponibles.

Voici une description succincte 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 la bibliothèque, celle-ci consulte ce fichier pour connaître les sources à consulter.

```
$ 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 visualiser 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 simple d'un nom d'hôte

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

```
$ dig +short www.nic.fr
web.nic.fr.
192.134.4.20
```

Résolution inverse d'une adresse IP

La question posée est : quel est le nom d'hôte correspondant à l'adresse IP `192.134.4.20` ?

```
$ dig +short -x 192.134.4.20
web.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[11].

[11] http://www.inetdoc.net/travaux_pratiques/index.html#sysadm-net.dns

9. Service traceroute

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'obejctif 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: Thu Jan 30 11:30:53 2014
HOST: vm0                          Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- 192.0.2.1                   0.0%    10    0.3   0.3   0.3   0.4   0.0
  2.|-- gw.iut-tlse3.fr            90.0%    10    0.7   0.7   0.7   0.7   0.0
  3.|-- rtr-toip1.cict.fr          90.0%    10    0.9   0.9   0.9   0.9   0.0
  4.|-- 194.167.94.2               90.0%    10    0.9   0.9   0.9   0.9   0.0
  5.|-- 194.199.11.2               90.0%    10    1.0   1.0   1.0   1.0   0.0
  6.|-- te1-3-toulouse-rtr-021.no 90.0%    10    1.3   1.3   1.3   1.3   0.0
  7.|-- te4-1-bordeaux-rtr-021.no 90.0%    10   18.7  18.7  18.7  18.7   0.0
  8.|-- te1-2-clermont-rtr-021.no 90.0%    10   19.3  19.3  19.3  19.3   0.0
  9.|-- 193.51.180.197             90.0%    10   17.1  17.1  17.1  17.1   0.0
 10.|-- 193.51.179.202             90.0%    10   12.3  12.3  12.3  12.3   0.0
 11.|-- 193.51.180.193             90.0%    10   22.1  22.1  22.1  22.1   0.0
 12.|-- ???                        100.0    10    0.0   0.0   0.0   0.0   0.0
 13.|-- ???                        100.0    10    0.0   0.0   0.0   0.0   0.0
 14.|-- ???                        100.0    10    0.0   0.0   0.0   0.0   0.0
 15.|-- ???                        100.0    10    0.0   0.0   0.0   0.0   0.0
 16.|-- web.nic.fr                  0.0%    10   20.0  19.9  19.8  20.1   0.0
```

Exemple de rapport basé sur UDP

```
$ mtr -4 -u -c 10 --report www.nic.fr
Start: Thu Jan 30 11:35:14 2014
HOST: vm0                          Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- 192.0.2.1                   0.0%    10    0.3   0.3   0.3   0.4   0.0
  2.|-- gw.iut-tlse3.fr             0.0%    10    0.6   0.6   0.6   0.7   0.0
  3.|-- rtr-toip1.cict.fr           0.0%    10    1.2   1.4   1.1   3.2   0.3
  4.|-- 194.167.94.2                0.0%    10    1.2   1.1   1.0   1.2   0.0
  5.|-- 194.199.11.2                0.0%    10    1.4   3.7   1.3  12.6   4.4
  6.|-- ???                        100.0    10    0.0   0.0   0.0   0.0   0.0
  7.|-- te4-1-bordeaux-rtr-021.no   0.0%    10   19.2  17.7  15.8  19.5   1.6
  8.|-- te1-2-clermont-rtr-021.no   0.0%    10   19.4  19.4  19.3  19.7   0.0
  9.|-- 193.51.180.197              0.0%    10   19.6  18.1  16.0  20.2   1.1
 10.|-- 193.51.179.202              0.0%    10   12.6  12.7  12.5  13.0   0.0
 11.|-- 193.51.180.193              0.0%    10   23.3  21.0  19.2  23.3   1.2
 12.|-- jaguar-network.sfinx.tm.f   0.0%    10   19.4  20.9  19.2  35.0   5.0
 13.|-- vl80.er01.par02.jaguar-ne   0.0%    10   19.6  19.4  19.2  19.6   0.0
 14.|-- cpe-et000652.cust.jaguar-   0.0%    10   20.2  20.3  20.0  21.8   0.5
 15.|-- ???                        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. Fonctions réseau du noyau Linux

Sur tous les systèmes, un certain nombre de paramètres sont actifs par défaut sur les interfaces réseau. Avec le noyau Linux, ces paramètres sont placés dans le système de fichiers virtuel `/proc`.

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

Sur les systèmes GNU/Linux, la granularité du paramétrage du sous-système réseau du noyau est très très fine. Il suffit de visualiser le résultat des commandes comme **`$ ls /proc/sys/net/ipv4/`** ou **`# sysctl -A | grep net`** pour le constater.

Sachant que le nom de l'interface Ethernet est '`eth0`', on peut commencer par faire une recherche des répertoires relatifs à ce nom d'interface.

```
$ find /proc/sys -type d -name '*eth0*'
/proc/sys/net/ipv4/conf/eth0
/proc/sys/net/ipv4/neigh/eth0
/proc/sys/net/ipv6/conf/eth0
/proc/sys/net/ipv6/neigh/eth0
```

Ensuite, on peut consulter le contenu d'un répertoire identifié dans la liste ci-dessus.

```
$ for param in `find /proc/sys/net/ipv4/conf/eth0 -type f`; do \
echo $param = `cat $param`; \
done
/proc/sys/net/ipv4/conf/eth0/accept_local = 0
/proc/sys/net/ipv4/conf/eth0/accept_redirects = 1
/proc/sys/net/ipv4/conf/eth0/accept_source_route = 1
/proc/sys/net/ipv4/conf/eth0/arp_accept = 0
/proc/sys/net/ipv4/conf/eth0/arp_announce = 0
/proc/sys/net/ipv4/conf/eth0/arp_filter = 0
/proc/sys/net/ipv4/conf/eth0/arp_ignore = 0
/proc/sys/net/ipv4/conf/eth0/arp_notify = 0
/proc/sys/net/ipv4/conf/eth0/bootp_relay = 0
/proc/sys/net/ipv4/conf/eth0/disable_policy = 0
/proc/sys/net/ipv4/conf/eth0/disable_xfrm = 0
/proc/sys/net/ipv4/conf/eth0/force_igmp_version = 0
/proc/sys/net/ipv4/conf/eth0/forwarding = 0
/proc/sys/net/ipv4/conf/eth0/igmpv2_unsolicited_report_interval = 10000
/proc/sys/net/ipv4/conf/eth0/igmpv3_unsolicited_report_interval = 1000
/proc/sys/net/ipv4/conf/eth0/log_martians = 0
/proc/sys/net/ipv4/conf/eth0/mc_forwarding = 0
/proc/sys/net/ipv4/conf/eth0/medium_id = 0
/proc/sys/net/ipv4/conf/eth0/promote_secondaries = 0
/proc/sys/net/ipv4/conf/eth0/proxy_arp = 0
/proc/sys/net/ipv4/conf/eth0/proxy_arp_pvlan = 0
/proc/sys/net/ipv4/conf/eth0/route_localnet = 0
/proc/sys/net/ipv4/conf/eth0/rp_filter = 0
/proc/sys/net/ipv4/conf/eth0/secure_redirects = 1
/proc/sys/net/ipv4/conf/eth0/send_redirects = 1
/proc/sys/net/ipv4/conf/eth0/shared_media = 1
/proc/sys/net/ipv4/conf/eth0/src_valid_mark = 0
/proc/sys/net/ipv4/conf/eth0/tag = 0
```

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.

Comment changer la valeur d'un paramètre ?

Pour changer les valeurs attribuées par défaut lors de l'initialisation du système, il existe plusieurs solutions. Pour faire simple, on se limite à l'utilisation de la commande **sysctl** fournie par le paquet procps. Le fichier de configuration de l'outil est : `/etc/sysctl.conf`. Si on édite ce fichier en modifiant ou en ajoutant un paramètre et que l'on applique ensuite la commande **`# sysctl -p`**, tous les changement seront effectifs immédiatement. Comme le fichier `/etc/sysctl.conf` est conservé, les paramètres seront à nouveau appliqués lors de l'initialisation du système. Voici un exemple très simple dans lequel on modifie les deux paramètres relatifs aux messages de redirection ICMP que l'on ne souhaite pas accepter pour des raisons de sécurité.

On décommente les deux lignes du fichier avec le mot clé '`accept_redirects`'. La partie modifiée du fichier est la suivante :

```
# Do not accept ICMP redirects (prevent MITM attacks)
net.ipv4.conf.all.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
```

Le résultat de l'appel à **sysctl** est :

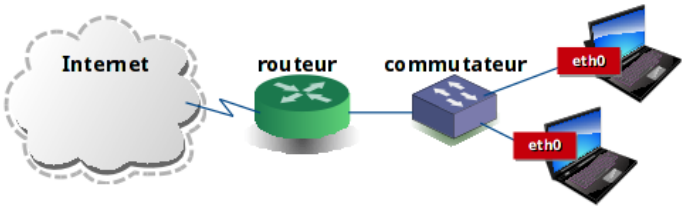
```
# sysctl -p
net.ipv4.conf.all.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
```

Les quelques paramètres des fonctions réseau du noyau Linux présentés ici ne constituent qu'une infime partie. Le document Ipsysctl tutorial[12] présente l'ensemble des paramètres utilisables pour ajuster le fonctionnement de la pile de protocoles TCP/IP.

[12] http://ipsysctl-tutorial.frozentux.net/ipsysctl-tutorial.html

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 :



Questions sur la configuration de l'interface Ethernet

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

Utiliser les commandes présentées dans la Section 2, « Identification d'une interface - couche physique » pour extraire la référence du composant.

Q2. Quelles sont les informations sur la connexion au réseau local Ethernet ?

Utiliser les commandes présentées dans la Section 2, « Identification d'une interface - couche physique » pour obtenir les informations sur le débit utile entre l'hôte et le commutateur, le mode de transmission full duplex ou half duplex et le type de média utilisé.

Q3. Quelles sont les différentes adresses de l'interface Ethernet aux niveaux liaison de données et réseau ? Quel est le protocole qui assure la correspondance entre ces adresses ?

Utiliser les commandes de visualisation présentée dans la Section 3, « Configuration d'une interface - couche liaison » et la Section 5, « Configuration d'une interface - couche réseau ».

Q4. Quelles sont les informations données par la configuration sur l'état de l'interface ?

Reprendre les éléments du tableau des indicateurs d'états donné dans la Section 3, « Configuration d'une interface - couche liaison » et faire la correspondance avec la configuration visualisée.

Questions sur les mécanismes d'adressage

Q5. Quelle est l'adresse du réseau IP auquel l'interface Ethernet est raccordée ? Quelle est la plage des adresses IP utilisables ? Quelle est l'adresse de diffusion de ce réseau ?

Utiliser la commande proposée à la Section 5, « Configuration d'une interface - couche réseau » et le document Adressage IPv4[13] pour déterminer les bornes du réseau IP.

Q6. Est-il possible de déduire l'adresse IP de la passerelle par défaut à partir des informations fournies par la configuration de l'interface ?

Attention, même si des conventions font que l'on retrouve fréquemment les adresses IP des passerelles en première ou dernière position de l'espace d'adressage utile, il n'existe aucune règle définie.

Q7. Quels sont les espaces de validité respectifs des adresses MAC et IP ?

Retrouver les informations dans la Section 3, « Configuration d'une interface - couche liaison » et la Section 5, « Configuration d'une interface - couche réseau ».

[13] http://www.inetdoc.net/articles/adressage.ipv4/

Q8. Comment visualiser la table de correspondance entre les adresses MAC et IP connues de l'interface Ethernet ?

Utiliser la commande proposée dans la Section 4, « Protocole ARP ».

Q9. Comment forcer l'ajout d'une nouvelle entrée dans la table des voisins ? À quel réseau IP doivent appartenir les adresses à tester ?

Utiliser la commande proposée dans la Section 7, « Protocole ICMP - couche réseau » pour tester l'accessibilité d'une adresse IP.

Q10. Pourquoi des entrées apparaissent dans la table des voisins sans action particulière ?

Essayer de repérer les stations du réseau local qui ont contacté l'interface Ethernet.

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

Retrouver l'entrée de la table des voisins sollicitée par le trafic sortant vers l'Internet.

Q12. Pourquoi l'entrée avec l'adresse MAC de la passerelle par défaut est-elle «rafraîchie» après un test de la commande **ping** utilisant un nom d'hôte ?

Revoir les tests présentés à la Section 7, « Protocole ICMP - couche réseau » et les éléments de configuration donnés à la Section 8, « Protocole DNS - couche application ».

Questions sur le service DNS

Q13. Quelle est l'adresse IP du serveur DNS indiquée dans le fichier de configuration du client DNS ?

Utiliser les informations données à la Section 8, « Protocole DNS - couche application ».

Q14. Quelle est la commande à utiliser pour poser une requête DNS individuelle ? Donner un exemple ?

Utiliser les informations données à la Section 8, « Protocole DNS - couche application ».

Questions sur la table de routage

Q15. Combien y-a-t-il d'entrées dans la table de routage ? Quel est le rôle de chacune de ces entrées ?

Reprendre l'exemple donné à la Section 6, « Table de routage simple - couche réseau » et faire la correspondance avec les informations relevées.

Q16. Comment identifier la passerelle par défaut dans la table de routage ?

Relever l'identifiant et/ou la valeur numérique du réseau de destination pour cette entrée de la table de routage.

Q17. La passerelle par défaut peut-elle appartenir à un autre réseau que celui de la station ?

La fonction d'une passerelle par défaut est de fournir une voie de communication vers tous les autres réseaux. Compléter le raisonnement à partir du cas où cette voie de communication n'appartient pas au réseau local.

Questions sur traceroute

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

Relire la Section 9, « Service traceroute ».

Q19. À partir d'un exemple, donner pour chaque routeur traversé les valeurs du champ TTL de l'entête IP.

Reprendre l'exemple donné dans la Section 9, « Service traceroute » et faire la correspondance avec les informations relevées.

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

Relire l'avertissement donné dans la Section 9, « Service traceroute ».