

L'Internet Rapide et Permanent

Le protocole DHCP

Par Christian Caleca

Date de publication : 6 mars 2010

DÉBUTANT

Vous disposez d'une connexion permanente et rapide... et maintenant, vous êtes perdu dans la technique...

Cette série « L'Internet Rapide et Permanent », que Christian Caleca nous a aimablement autorisés à reproduire, est là pour répondre à quelques-unes de ces questions. Cet article parlera du protocole DHCP.

N'hésitez pas à commenter cet article !

1 - Introduction.....	3
2 - Le protocole « DHCP ».....	3
2-1 - Position du problème.....	3
2-2 - Que disent les livres ?.....	4
2-2-1 - Détails sur le serveur DHCP.....	4
2-2-2 - Détails sur le bail.....	5
2-2-3 - Question subsidiaire.....	5
3 - Serveur DHCP.....	5
3-1 - Topologie du réseau.....	5
3-2 - Installation du serveur DHCP.....	6
3-3 - Configuration du serveur.....	6
3-3-1 - Le principe.....	6
3-3-2 - Une configuration basique.....	6
3-3-2-1 - Ce que nous voulons faire.....	6
3-3-2-2 - Note importante.....	7
3-3-2-3 - Ce que nous écrivons dans dhcpd.conf.....	7
4 - Les clients DHCP.....	8
4-1 - MS Windows™.....	8
4-1-1 - Configuration.....	8
4-1-2 - Contrôle.....	10
4-2 - GNU/Linux.....	11
4-2-1 - NetworkManager.....	11
4-2-2 - À l'ancienne.....	12
4-2-2-1 - ifconfig.....	13
4-2-2-2 - iproute.....	13
4-2-2-3 - Les logs.....	13
5 - Analyse du protocole.....	14
5-1 - Premier démarrage de la station.....	14
5-1-1 - Résumé de la capture.....	14
5-1-2 - Analyse détaillée.....	15
5-1-2-1 - Discover.....	15
5-1-2-2 - Ping.....	16
5-1-2-3 - Offer.....	17
5-1-2-4 - Request.....	18
5-1-2-5 - ACK.....	20
5-2 - Renouvellement.....	22
5-3 - Le grain de sable.....	25
5-3-1 - Renew.....	25
5-3-2 - Mort (et résurrection).....	29
6 - Conclusion.....	29
6-1 - Remerciements Developpez.....	30

1 - Introduction

Il s'agit d'un protocole qui permet dynamiquement d'attribuer une adresse IP à un système informatique qui se connecte au réseau.

Le serveur DHCP fournit également d'autres informations, comme la passerelle par défaut et le DNS (et éventuellement bien d'autres choses encore).

Ce système, bien pratique lorsqu'il fonctionne correctement, évite à l'utilisateur d'avoir à configurer manuellement sa pile IP. Il permet également à l'administrateur du réseau de modifier son architecture sans avoir à prévenir tous ses clients ni à se déplacer de poste en poste.

Toutes nos « machin-box » dès lors qu'elles sont utilisées en mode routeur fournissent cette fonctionnalité, de façon plus ou moins élaborée. Nous verrons cependant ce protocole à travers **le serveur DHCP d'ISC** installé, comme d'habitude, sur une Debian (Lenny).

La bonne compréhension de ce protocole implique un minimum de connaissances de TCP/IP. Si vous ne les avez pas, lisez d'abord le chapitre « **TCP/IP(v4)** ».

Nous allons dans un premier temps construire un serveur DHCP et le tester avec des clients sous GNU/Linux et même aussi sous Windows, ne soyons pas plus sectaires que de raison.

La dernière page du chapitre permettra d'analyser dans le détail ce qu'il se passe dans l'obscurité d'un réseau IP et qui fait que « ça marche » le plus souvent.

2 - Le protocole « DHCP »

2-1 - Position du problème

Lorsque vous connectez une machine à un réseau Ethernet TCP/IP, cette machine, pour fonctionner correctement, doit disposer :

- d'une adresse IP unique dans votre réseau et appartenant au même réseau logique que toutes les autres machines du réseau en question ;
- un masque de sous-réseau, le même pour tous les hôtes du réseau ;
- une adresse de DNS, pour pouvoir résoudre les noms des hôtes, surtout si votre réseau est connecté au Net ;
- l'adresse de la passerelle qui vous permet justement d'accéder au Net. (Nous supposerons que votre réseau domestique n'est pas suffisamment complexe pour contenir de multiples sous-réseaux.)

Si vous n'avez déjà rien compris à ce discours, alors il est nécessaire pour vous de lire d'abord les chapitres sur **TCP/IP(v4)** et **IP et le Routage**.

Pour configurer vos hôtes locaux, vous avez trois possibilités :

- vous utilisez « zeroconf » (**rfc3927**), la chose qui permet à chaque nœud d'un réseau de s'autoattribuer une adresse IP dans le bloc 169.254/16. Tout ce qu'il y a de plus basique, ce système remplit son office, mais ne permet aucune administration du réseau ;
- vous passez de machine en machine, avec un petit carnet et vous configurez à chaque fois tous les paramètres de la pile IP à la main, en n'oubliant pas de tout marquer dans votre carnet. Ce n'est pas le plus compliqué, ce qui est davantage gênant, c'est de ne jamais oublier de noter toutes les modifications que vous pourriez être amené à faire par la suite ;

- vous installez un serveur DHCP sur votre réseau et vous dites à vos clients d'aller chercher toute leur configuration IP sur ce serveur. En gros, il remplacera votre carnet, sera naturellement à jour et vous évitera des déplacements.

Comme vous le voyez, le luxe de la troisième solution est tout de même tentant, au point que nous allons le mettre en œuvre.

2-2 - Que disent les livres ?

Les choses se passent avec le peu de moyens dont vous disposez :

- votre adresse MAC que vous ne perdez jamais, puisqu'elle est écrite en dur dans votre interface Ethernet ;
- le « Broadcast » ou « Diffusion » qui permet d'envoyer des trames à toutes les machines du réseau physique.

Le dialogue est décrit de la manière suivante :

- lorsque le client DHCP démarre, il n'a aucune connaissance du réseau, du moins, en principe. Il envoie donc une trame « DHCPDISCOVER », destinée à trouver un serveur DHCP. Cette trame est un « broadcast », donc envoyé à l'adresse 255.255.255.255. N'ayant pas encore d'adresse IP, il adopte provisoirement l'adresse 0.0.0.0. Comme ce n'est pas avec cette adresse que le DHCP va l'identifier, il fournit aussi sa « MAC Address ». En réalité, 0.0.0.0 ne voulant rien dire, le système devra fonctionner uniquement avec les adresses MAC lors du premier dialogue. Autrement dit, le client présente son adresse MAC et effectue un broadcast ethernet sur ff:ff:ff:ff:ff:ff ;
- le, ou les serveurs DHCP du réseau qui vont recevoir cette trame vont se sentir concernés et répondre par un « DHCPOFFER ».
Cette trame contient une proposition de bail et la « MAC Address » du client, avec également l'adresse IP du serveur. Tous les DHCP répondent et le client normalement accepte la première réponse venue, sauf s'il a déjà quelques exigences.
Le « DHCPOFFER » sera un broadcast (Ethernet) ou non, suivant le serveur DHCP utilisé. Nous aurons l'occasion d'observer tout ceci dans la dernière page de ce chapitre ;
- le client répond alors par un DHCPREQUEST à tous les serveurs (donc toujours en « Broadcast ») pour indiquer quelle offre il accepte ;
- le serveur DHCP concerné répond définitivement par un DHCPACK qui constitue une confirmation du bail. L'adresse du client est alors marquée comme utilisée et ne sera plus proposée à un autre client pour toute la durée du bail.

2-2-1 - Détails sur le serveur DHCP

Un serveur DHCP dispose d'une plage d'adresses à distribuer à ses clients. Il tient à jour une base de données des adresses déjà utilisées et utilisées il y a peu (c'est ce qui explique que l'on récupère souvent la même adresse, le DHCP ayant horreur des changements).

Lorsqu'il attribue une adresse, il le fait par l'intermédiaire d'un bail. Ce bail a normalement une durée limitée dans le temps. Sur un réseau d'entreprise où l'on dispose largement d'assez d'adresses pour le nombre de postes et que ces derniers sont en service toute la journée, le bail peut être d'une semaine ou plus encore. Un bail long diminue le trafic réseau pour les renouvellements qui ont lieu moins souvent. En revanche, plus le bail est long, plus l'administrateur devra planifier longtemps à l'avance toute modification de l'architecture de son réseau.

Après expiration du bail, ou résiliation par le client, les informations concernant ce bail restent mémorisées dans la base de données du serveur pendant un certain temps. Bien que l'adresse IP soit disponible, elle ne sera pas attribuée en priorité à une autre machine. C'est ce qui explique que l'on retrouve souvent la même adresse d'une session à l'autre.

2-2-2 - Détails sur le bail

Dans le bail, il y a non seulement une adresse IP pour le client, avec une durée de validité, mais également d'autres informations de configuration comme :

- l'adresse d'un ou de plusieurs DNS (résolution de noms) ;
- l'adresse de la passerelle par défaut (pour sortir du réseau IP où le DHCP vous a installé) ;
- l'adresse du serveur DHCP (nous allons voir pourquoi).

Cette liste est loin d'être complète, il existe en effet une grande quantité d'options qui peuvent être transmises.

Lorsque le bail arrive à environ la moitié de son temps de vie, le client va essayer de renouveler ce bail, cette fois-ci en s'adressant directement au serveur qui le lui a attribué. Il n'y aura alors qu'un DHCPREQUEST et un DHCPACK.

Si, au bout des 7/8e de la durée de vie du bail en cours, ce dernier n'a pu être renouvelé, le client essaiera d'obtenir un nouveau bail auprès d'un DHCP quelconque qui voudra bien lui répondre. Il pourra alors se faire que le client change d'adresse IP en cours de session. Normalement, cette situation ne devrait pas se produire, sauf en cas de panne du DHCP.

2-2-3 - Question subsidiaire

Il doit donc y avoir nécessairement un serveur DHCP par réseau physique et il doit disposer d'une adresse IP dans la même classe que celle qui constitue sa plage d'adresses ?

Non, pas nécessairement. Votre réseau physique peut être formé de plusieurs sous-réseaux logiques, avec des routeurs entre chaque sous-réseau et le tout peut fonctionner avec un seul serveur DHCP...

Mais alors, comment la négociation peut-elle se faire, puisque, normalement, un « broadcast » n'est pas retransmis par les routeurs ?

Les requêtes DHCP doivent pouvoir atteindre le serveur qui est situé sur un autre réseau logique, elles doivent donc passer les routeurs, ce qui n'est pas possible. Il est alors nécessaire d'installer sur un ou plusieurs routeurs un agent de relais qui va intercepter les requêtes en broadcast et les transmettre à un serveur DHCP connu de cet agent.

C'est l'agent de relais situé sur la passerelle qui va faire l'intermédiaire et le client réussira tout de même à obtenir un bail, donné par un DHCP situé sur un autre réseau et transmis par l'agent de relais.

Nous ne pousserons pas le luxe jusque-là, mais la solution existe. Le serveur DHCP sera même capable d'envoyer des paramètres différents, suivant le sous-réseau du client...

3 - Serveur DHCP

3-1 - Topologie du réseau

Nous allons prendre une configuration simple, avec une machine GNU/Linux qui va cumuler plusieurs fonctions :

- passerelle entre le réseau local et l'Internet (notre « box » ne sera ici qu'un simple modem ADSL) ;
- firewall ;
- serveur DHCP ;
- serveur DNS (pourquoi pas, puisqu'on est dans le luxe ?) ;
- les clients peuvent être de tout type : Windows, Mac OS, GNU/Linux...
- dans l'exemple, la passerelle est construite avec Debian Lenny.

Nous allons donc installer sur la passerelle un serveur DHCP. Le DNS est tout à fait optionnel, mais ce serait bien qu'il y soit, il peut même y être déjà, ça n'est absolument pas gênant. S'il n'y est pas encore, vous pourrez le rajouter par la suite.

Les fonctions de passerelle et de firewall ne sont pas non plus fondamentales, nous pourrions nous contenter d'un serveur GNU/Linux, non connecté au Net (mais qui peut le plus peut le moins).

Nous pourrions même ajouter un autre serveur au réseau local, chargé du DNS et du DHCP et ne laisser à la passerelle que les fonctions de routage et de firewall.

3-2 - Installation du serveur DHCP

Sur Debian, ceci se fait très simplement en installant les paquetages dhcp3-common et dhcp3-server. Dans la version Lenny de Debian, nous disposons de la version 3.1.1 du serveur. Il y a un seul fichier de configuration : /etc/dhcp3/dhcpd.conf, que nous pourrions configurer avec un éditeur de texte, où à travers l'interface Webmin. Ce que nous aurons à faire est suffisamment simple pour pouvoir le faire à la main.

3-3 - Configuration du serveur

3-3-1 - Le principe

Comme nous l'avons vu plus haut, un serveur DHCP, en plus de fournir la configuration IP « de base » (adresse et masque), peut aussi transmettre un nombre plus ou moins grand de paramètres supplémentaires. Nous aurons donc au moins deux choses à configurer :

- la réserve d'adresses dont le serveur pourra disposer pour les attribuer aux clients ;
- les paramètres optionnels à leur communiquer dans la foulée, comme l'adresse d'un DNS et de la passerelle par défaut. Dans le cas d'un réseau domestique, ce sera suffisant, mais il y a beaucoup d'autres options, plus ou moins spécifiques aux divers systèmes d'exploitation.

Nous avons vu qu'un seul serveur DHCP pouvait être utilisé pour plusieurs réseaux logiques interconnectés, pourvu que les interconnexions disposent d'un agent de relais DHCP. Dans un tel cas, le serveur DHCP devra disposer d'au moins une étendue d'adresses IP par réseau logique dont il aura la charge.

En ce qui concerne les options, nous disposons d'une architecture hiérarchique :

- nous pouvons définir des options globales, qui seront les mêmes pour tous les clients du DHCP, tous sous-réseaux confondus ;
- nous pouvons définir également des options propres à chaque sous-réseau, celles-ci écrasant les options globales, en cas de conflit ;
- si l'on veut aller encore plus loin, sachez que DHCP3-server peut créer des groupes distincts de machines dans un même sous-réseau et même gérer des clients de façon individuelle. C'est un **vrai** serveur DHCP...

3-3-2 - Une configuration basique

3-3-2-1 - Ce que nous voulons faire

Nous n'allons pas monter une usine à gaz, c'est inutile pour un petit réseau domestique et ça n'apporte rien de plus à la compréhension du protocole :

- nous avons un seul réseau, avec des IP choisies dans la classe C privée 192.168.0.0, donc avec un masque 255.255.255.0 ;

- nous avons donc une passerelle par défaut unique pour tous nos hôtes du réseau, dans l'exemple, ce sera 192.168.0.252 ;
- nous disposons enfin d'un DNS, également unique pour tous les hôtes du réseau, il est sur la même machine, à savoir 192.168.0.252. Le domaine que nous avons construit sur notre réseau local s'appelle maison.mrs. Il n'a aucune réalité sur le Net, mais ça n'a pas d'importance, puisque c'est un domaine qui ne doit pas être visible depuis le Net ;
- sur la totalité de la classe C disponible, nous allons réserver les adresses comprises entre 192.168.0.64 et 192.168.0.127 pour les clients du réseau. Cette plage constituera la réserve d'adresses que le DHCP pourra fournir aux clients. Bien entendu, nous aurions pu en mettre plus, mais il faut toujours se garder quelques adresses sous le coude, pour les machines configurées manuellement, comme les serveurs que l'on placerait sur le réseau ;
- un dernier point important, c'est la durée de vie du bail que le DHCP va attribuer aux clients. L'un des avantages de DHCP, c'est de pouvoir attribuer une configuration IP qui ne sera valide que dans un laps de temps donné, à charge pour le client de demander le renouvellement de ce bail avant chaque expiration. Ce temps de vie pourra aller de quelques minutes à l'infini, suivant les besoins. Sur un réseau qui évolue peu, le bail peut être sans problèmes de quelques jours, à quelques semaines, voire plusieurs mois. Sur un réseau où les hôtes vont et viennent, il sera plus sage de ne laisser vivre les configurations que quelques heures. Répétons-le, plus le bail est court, plus le trafic généré par DHCP devient important et plus la charge du serveur augmente. Ceci dit, ce n'est pas un argument très significatif sur un réseau ne dépassant pas une dizaine de clients. Dans l'exemple, nous utiliserons une heure (3600 secondes).

3-3-2-2 - Note importante

Le « daemon » dhcpd écoute par défaut sur toutes les interfaces réseau actives sur le serveur. Ce n'est pas forcément souhaitable, c'est même assez souvent ennuyeux.

Fort heureusement, ce comportement par défaut peut être modifié, mais pas dans le fichier de configuration. Il faut utiliser un paramètre dans la ligne de commande qui va démarrer dhcpd.

Dans le cas de Debian Lenny, il faut éditer le fichier /etc/default/dhcp3-server. Il est bien documenté et vous trouverez aisément la variable INTERFACES qu'il faut initialiser avec le nom de la ou des interfaces qui doivent être écoutées. Dans notre exemple, nous aurons :

```
INTERFACES="eth0"
```

3-3-2-3 - Ce que nous écrivons dans dhcpd.conf

```
# La ligne qui suit est nécessaire. Elle est en rapport avec
# la mise à jour dynamique du DNS, que nous n'utiliserons pas
# pour l'instant.
ddns-update-style none;

# Ce serveur fait autorité sur le réseau
authoritative;

# Les options globales
option time-servers 192.168.0.252;
option domain-name "maison.mrs";
max-lease-time 3600;
default-lease-time 3600;
option domain-name-servers 192.168.0.252;
option subnet-mask 255.255.255.0;
option routers 192.168.0.252;

# le réseau 192.168.0.0/24, avec la réserve d'adresses dynamiques
subnet 192.168.0.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.0.64 192.168.0.127;
}
```

Cette configuration simplissime va suffire à nos besoins.

Dans ce fichier, il y a des directives, qui sont obligatoires :

- la directive `ddns-update-style` servira plus tard, ce sera la cerise sur le gâteau, pour ceux qui utilisent BIND 9 (le serveur DNS). Elle indique pour l'instant que nous ne ferons pas de mise à jour dynamique de DNS ;
- il existe une subtile différence entre les directives `max-lease-time` et `default-lease-time` (DHCP est plein de subtilités, pas toujours nécessaires dans un fonctionnement basique), la page `man dhcpd.conf` vous indiquera quelle est cette différence. Contentons-nous pour l'instant d'assigner la même valeur aux deux, ici 3600 secondes.

Et des options qui seront dans la pratique, des paramètres de configuration optionnels. Ici :

- `domain-name-servers`
qui attribuera aux hôtes une adresse de DNS. Dans l'exemple, notre DNS à nous. Si nous n'en avons pas, il faudra ici mettre l'IP du DNS de notre fournisseur d'accès. Éventuellement, nous pouvons spécifier plusieurs DNS ;
- `domain-name`
est vraiment optionnel, ça permettra aux clients de savoir dans quel domaine ils sont enregistrés ;
- `routers`
c'est la passerelle par défaut. Il pourrait y avoir plusieurs routeurs, mais tous les systèmes ne savent pas gérer de façon efficace une information contenant plusieurs passerelles.

Toutes les options qui figurent avant le paragraphe `subnet 192.168.0.0 netmask 255.255.255.0` sont des options globales, il n'y a ici aucune option d'étendue (de sous-réseau) de définie.

Cette configuration doit nous permettre d'être efficient dans notre contexte. Il nous suffit de lancer ou de relancer le serveur :

```
/etc/init.d/dhcpd restart
```

Et ça devrait fonctionner.

4 - Les clients DHCP

Tout système d'exploitation (y compris MS Windows™ depuis l'antique version 95) dispose d'un client DHCP dont le rôle est de rechercher sur le réseau un serveur DHCP et de négocier avec lui une configuration IP cohérente.

Si le poste de travail est correctement configuré, le client DHCP s'occupera de tout.

4-1 - MS Windows™

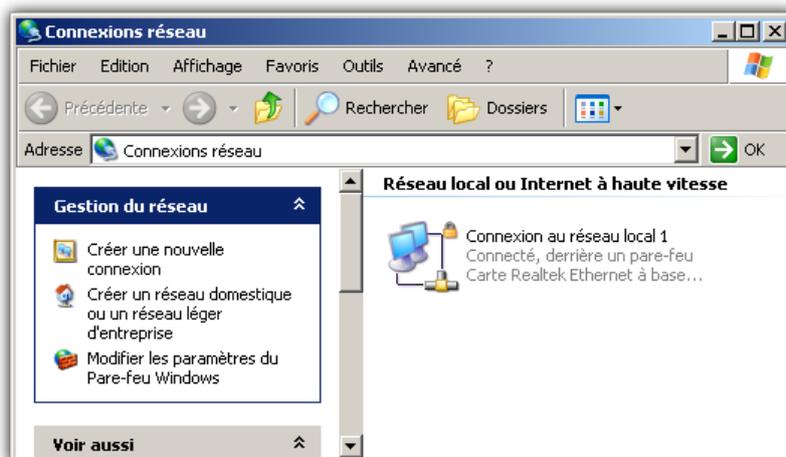
N'étant pas un fan de ces systèmes en général (ni de Vista™ en particulier), nous nous contenterons de la démonstration sur un Windows XP Pro (SP3).

4-1-1 - Configuration

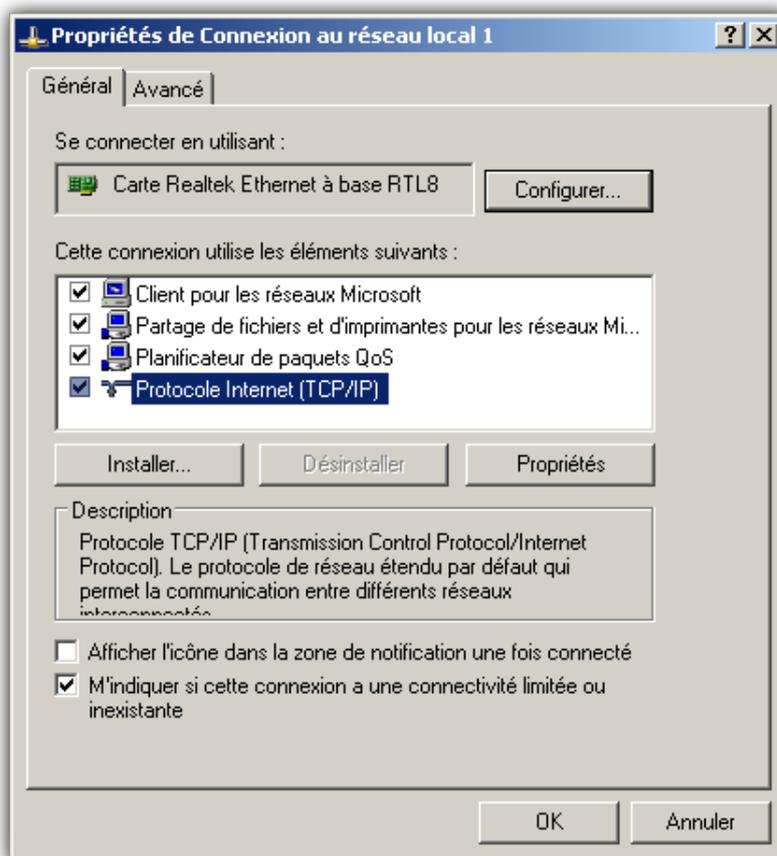
Voici un moyen (parmi d'autres) d'y arriver. Celle-ci s'appelle la « méthode en dix clics ».

Tu cliques à gauche sur Démarrer et, à condition d'avoir configuré son machin de cette façon, les favoris réseau apparaissent. Tu cliques à droite sur les favoris réseau et tu cliques à gauche sur propriétés.

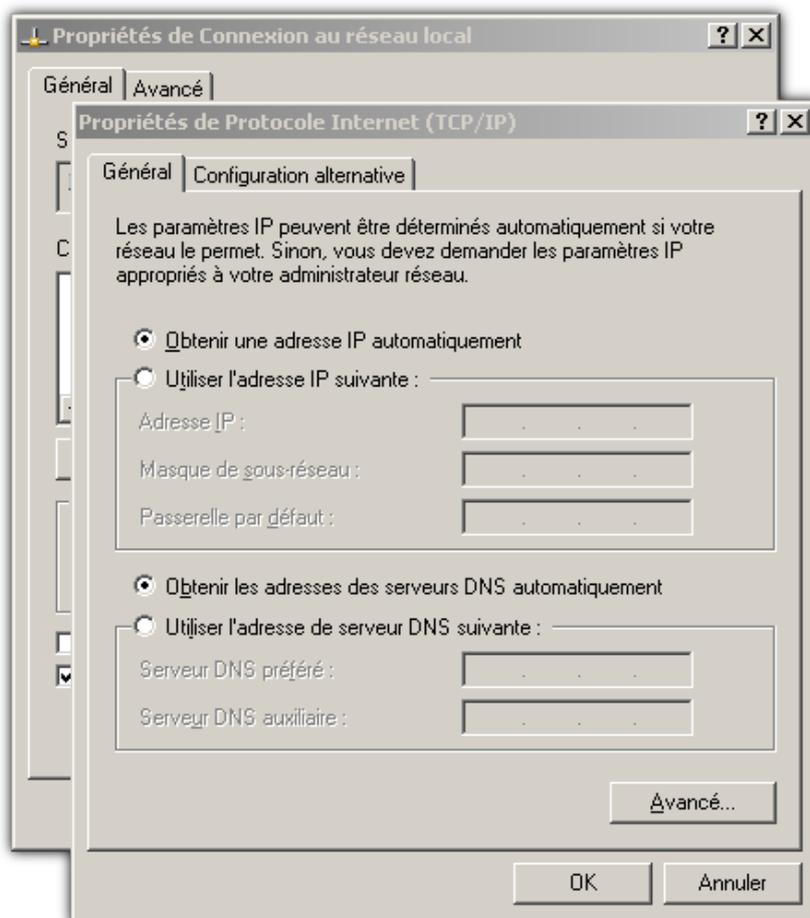
Tout ceci doit faire apparaître cette fenêtre :



Là, tu cliques à droite sur la Connexion au réseau local et tu cliques à gauche sur propriétés. Ceci t'amène à :



Tu cliques deux fois de suite à gauche sur Protocole Internet (TCP/IP). Tu es presque arrivé.



Ici, il suffit de cliquer à gauche successivement sur Obtenir une adresse IP automatiquement, sur Obtenir les adresses des serveurs DNS automatiquement et enfin sur OK. Et voilà le travail.

4-1-2 - Contrôle

Il existe une méthode, dite « méthode en sept clics », qui permet de consulter sa configuration IP. Nous allons en voir une autre dite « méthode de la ligne de commande ».

Tu ouvres une console DOS, aussi appelée invite de commandes et tu tapes :

```
ipconfig
```

Nous obtenons alors l'essentiel de la configuration IP.

Si nous souhaitons plus de détails, il faudra utiliser la commande :

```
ipconfig /all
```

Nous voyons ici toutes les informations concernant le bail dhcp obtenu. La « méthode en sept clics » n'apporte strictement aucune information supplémentaire, elle permet juste de faire travailler l'index et le majeur (éventuellement l'annulaire) de la main droite (habituellement), au détriment de tous les autres doigts.

4-2 - GNU/Linux

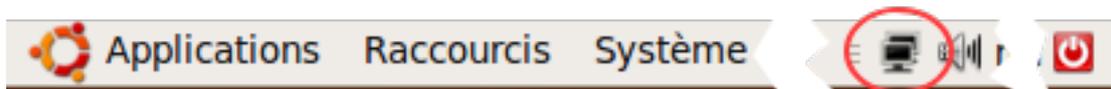
Ici, nous sommes dans le bazar et suivant les distributions, les choses peuvent changer de façon plus ou moins significative. Nous prendrons comme exemple une Debian Lenny et une Ubuntu Jaunty dont les architectures restent très proches puisque l'autre est dérivée de l'une.

GNU/Linux étant en perpétuelle évolution, les possibilités de configuration du réseau ont suivi le même chemin. Il existe en gros deux moyens de le faire :

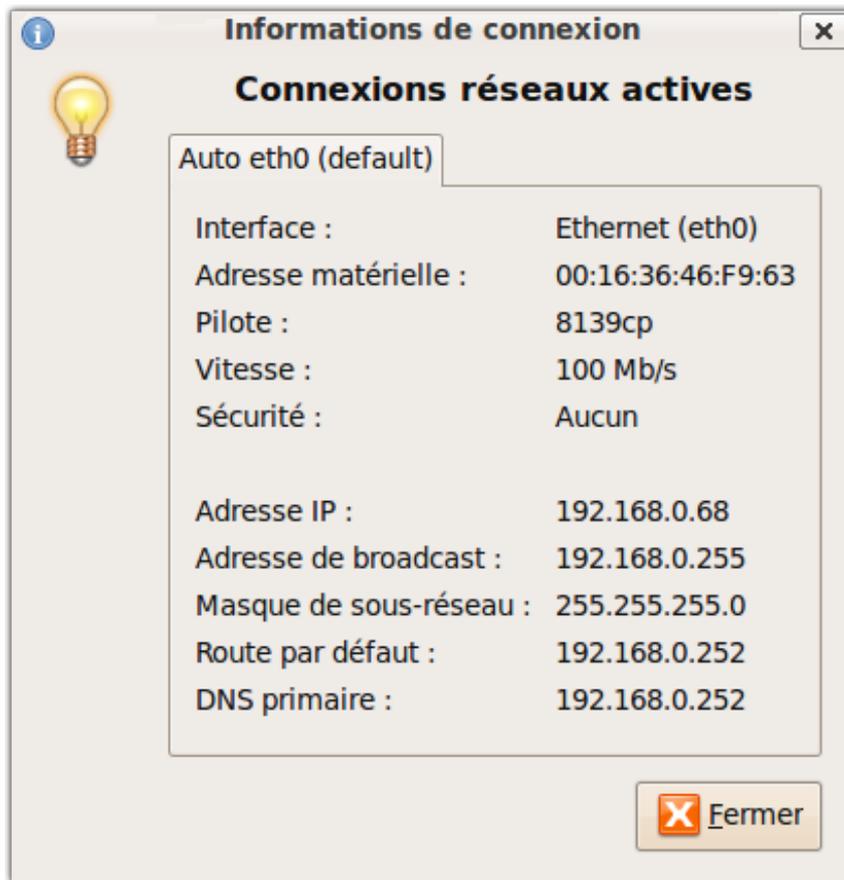
- des fichiers texte qui décrivent la configuration de chaque interface, c'est la bonne vieille méthode, encore tout à fait d'actualité dans le cas de machines sédentaires (poste de travail de bureau (desktop) et encore plus de serveurs ;
- un système dynamique, NetworkManager, qui s'appuie lui-même sur la couche HAL et sur DBUS en environnement graphique, qui est bien pratique pour les postes portables (laptop), surtout lorsqu'ils sont équipés d'une connexion WI-FI. NetworkManager dispose d'« applets » adaptés à la plupart des environnements graphiques (GNOME, KDE, XFCE...). Ce projet, initié par RedHat en 2004 manque peut-être encore un peu de maturité, mais commence à être tout à fait utilisable par une autre population que celle des « geeks ».

4-2-1 - NetworkManager

Commençons par le plus « simple », c'est-à-dire la méthode où ce sont les autres qui choisissent à votre place (courant très en vogue de nos jours). Un « desktop » avec Ubuntu 9.04 (Jaunty) est connecté au réseau où notre DHCP opère. Voici ce que ça donne :



Tu cliques à droite sur l'applet puis à gauche sur Informations de connexion (méthode en deux clics) et tu obtiens :



Bien entendu, tout ceci fonctionne bien parce que les concepteurs d'Ubuntu Jaunty ont correctement pensé l'architecture de la distribution, que tous les paquets nécessaires ont été installés et que les scripts de postinstallation ont correctement configuré tout ça. Les trois paquets essentiels, mais qui incluent beaucoup de dépendances, sont :

- network-manager pour la gestion automatisée des connexions actives ;
- network-manager-gnome pour que l'applet soit présent dans la zone de notification ;
- dhcp3-client qui est le client DHCP probablement le plus courant, bien qu'il en existe de nombreux autres. Car dans une architecture client/serveur, il faut un serveur mais aussi un client.

4-2-2 - À l'ancienne

Une méthode plus rustique, mais tout aussi efficace lorsque l'on est sédentaire consiste à configurer le réseau au moyen de fichiers texte. Dans l'architecture Debian et dérivées, tout se passe dans `/etc/network/interfaces`.

Nous sommes sur une Lenny sans interface graphique (pas de méthodes en X clics). Il nous faut bien sûr un client DHCP, pourquoi pas `dhcp3-client` qui est le choix par défaut sur Debian. En suite, nous pouvons créer ou modifier le fichier `/etc/network/interfaces` comme ceci :

```

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
    
```

La traduction est assez intuitive : nous voulons que `eth0` soit activée automatiquement et soit configurée par DHCP.

Après un démarrage de notre Lenny, voyons où nous en sommes. Il existe plusieurs façons de récolter les informations.

4-2-2-1 - ifconfig

Méthode traditionnelle :

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:16:36:51:5d:5a
          inet adr:192.168.0.67  Bcast:192.168.0.255  Masque:255.255.255.0
          adr inet6: 2a01:e35:2e52:9840:216:36ff:fe51:5d5a/64 Scope:Global
          adr inet6: fe80::216:36ff:fe51:5d5a/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:134 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:18211 (17.7 KiB)  TX bytes:9947 (9.7 KiB)
          Interruption:10 Adresse de base:0xc100
```

4-2-2-2 - iproute

Plus moderne :

```
# ip addr ls dev eth0
2: eth0:  mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:16:36:51:5d:5a brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.67/24 brd 192.168.0.255 scope global eth0
    inet6 2a01:e35:2e52:9840:216:36ff:fe51:5d5a/64 scope global dynamic
        valid_lft 86174sec preferred_lft 86174sec
    inet6 fe80::216:36ff:fe51:5d5a/64 scope link
        valid_lft forever preferred_lft forever
```

4-2-2-3 - Les logs

Si nous voulons vraiment savoir ce qu'il s'est passé au niveau DHCP, il suffit de consulter les logs du client dhcp3-client qui se trouvent sur Debian dans `/var/lib/dhcp3/dhclient.eth0.leases` :

```
# cat /var/lib/dhcp3/dhclient.eth0.leases
lease {
  interface "eth0";
  fixed-address 192.168.0.67;
  option subnet-mask 255.255.255.0;
  option routers 192.168.0.252;
  option dhcp-lease-time 3600;
  option dhcp-message-type 5;
  option domain-name-servers 192.168.0.252;
  option dhcp-server-identifiant 192.168.0.252;
  option netbios-name-servers 192.168.0.252;
  option domain-name "maison.mrs";
  renew 5 2009/05/08 08:38:00;
  rebind 5 2009/05/08 09:05:37;
  expire 5 2009/05/08 09:13:07;
}
```

Nous avons ici toutes les informations transmises. Notre système est « neuf », c'est la première fois que l'interface est configurée. Comme nous sommes curieux, nous avons mis en place un « sniffeur » (l'indispensable Wireshark), qui a capturé le dialogue, mais nous verrons ça plus tard.

Les heures sont indiquées en UTC.

La machine cliente a été démarrée à 8 h 12 UTC. Nous pouvons prendre une petite récréation. Rendez-vous vers 8 h 38 UTC.

```
lease {
```

```

interface "eth0";
fixed-address 192.168.0.67;
option subnet-mask 255.255.255.0;
option routers 192.168.0.252;
option dhcp-lease-time 3600;
option dhcp-message-type 5;
option domain-name-servers 192.168.0.252;
option dhcp-server-identifiant 192.168.0.252;
option netbios-name-servers 192.168.0.252;
option domain-name "maison.mrs";
renew 5 2009/05/08 08:38:00;
rebind 5 2009/05/08 09:05:37;
expire 5 2009/05/08 09:13:07;
}
lease {
interface "eth0";
fixed-address 192.168.0.67;
option subnet-mask 255.255.255.0;
option routers 192.168.0.252;
option dhcp-lease-time 3600;
option dhcp-message-type 5;
option domain-name-servers 192.168.0.252;
option dhcp-server-identifiant 192.168.0.252;
option netbios-name-servers 192.168.0.252;
option domain-name "maison.mrs";
renew 5 2009/05/08 09:05:03;
rebind 5 2009/05/08 09:30:30;
expire 5 2009/05/08 09:38:00;
}

```

Notre client DHCP a demandé un renouvellement de son bail. Il le fera à 9 h 05 UTC, etc.

Notre Wireshark n'a rien perdu de tout ceci, nous pouvons aller y voir de plus près.

5 - Analyse du protocole

5-1 - Premier démarrage de la station

5-1-1 - Résumé de la capture

No.	Time	Source	Destination	Protocol	Info
1	66.901361	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xc0b5592f
2	66.901867	192.168.0.252	192.168.0.67	ICMP	Echo (ping) request
3	67.902846	192.168.0.252	192.168.0.67	DHCP	DHCP Offer - Transaction ID 0xc0b5592f
4	67.904780	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0xc0b5592f
5	67.930502	192.168.0.252	192.168.0.67	DHCP	DHCP ACK - Transaction ID 0xc0b5592f

- Le client effectue une découverte de serveur DHCP. Il n'a pas encore d'adresse IP et adopte donc l'adresse factice 0.0.0.0. En réalité c'est bien sûr au niveau Ethernet que les adresses seront significatives, l'analyse approfondie le montrera.
- Le serveur (192.168.0.252) effectue un ping sur l'adresse 192.168.0.67, parce qu'il a l'intention d'attribuer cette adresse au client. S'il recevait une réponse au ping, cela voudrait dire que cette adresse est déjà en service sur le réseau, à cause d'une anomalie quelconque. Il n'y a pas de réponse au ping, ce qui est a priori normal sur un réseau normalement géré.
- Le serveur offre une proposition au client.
- Le client fait une contre-proposition. L'analyse détaillée va montrer qu'en principe, elle est identique à la proposition du serveur.
- Le serveur accepte la contre-proposition. Le bail est donc validé par les deux protagonistes.

5-1-2 - Analyse détaillée

5-1-2-1 - Discover

```

Frame 1 (342 bytes on wire, 342 bytes captured)
  Arrival Time: May  8, 2009 10:13:07.933412000
  [Time delta from previous captured frame: 66.901361000 seconds]
  [Time delta from previous displayed frame: 66.901361000 seconds]
  [Time since reference or first frame: 66.901361000 seconds]
  Frame Number: 2
  Frame Length: 342 bytes
  Capture Length: 342 bytes
  [Frame is marked: False]
  [Protocols in frame: eth:ip:udp:bootp]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
Ethernet II, Src: QuantaCo_51:5d:5a (00:16:36:51:5d:5a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

En voilà un joli broadcast ethernet...
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Address: Broadcast (ff:ff:ff:ff:ff:ff)
  .... 1. .... = IG bit: Group address (multicast/broadcast)
  .... 1. .... = LG bit: Locally administered address (this is NOT the factory
default)
  Source: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
  Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
  .... 0. .... = IG bit: Individual address (unicast)
  .... 0. .... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
Broadcast qui se retrouve sur la couche IP
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00)
    0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
    .... 0. = ECN-Capable Transport (ECT): 0
    .... 0. = ECN-CE: 0
  Total Length: 328
  Identification: 0x0000 (0)
  Flags: 0x00
    0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..0. = More fragments: Not set
  Fragment offset: 0
  Time to live: 128
  Protocol: UDP (0x11)
  Header checksum: 0x3996 [correct]
    [Good: True]
    [Bad : False]
  Source: 0.0.0.0 (0.0.0.0)
  Destination: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
  Source port: bootpc (68)
  Destination port: bootps (67)
  Length: 308
  Checksum: 0x746b [correct]
    [Good Checksum: True]
    [Bad Checksum: False]
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xc0b5592f
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
    0... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)

```

```

Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Discover
  Option: (53) DHCP Message Type
  Length: 1
  Value: 01
Option: (t=55,l=12) Parameter Request List
  Option: (55) Parameter Request List
  Length: 12
  Value: 011C02030F06770C2C2F1A79
  1 = Subnet Mask
  28 = Broadcast Address
  2 = Time Offset
  3 = Router
  15 = Domain Name
  6 = Domain Name Server
  119 = Domain Search
  12 = Host Name
  44 = NetBIOS over TCP/IP Name Server
  47 = NetBIOS over TCP/IP Scope
  26 = Interface MTU
  121 = Classless Static Route
End Option
Padding

```

Nous avons dans cette requête la liste des paramètres que le client souhaite recevoir, en plus bien entendu de son adresse IP.

Cette capture est également l'occasion de constater que DHCP utilise UDP, sur le port 67 pour le client et le port 68 pour le serveur.

5-1-2-2 - Ping

```

Frame 2 (62 bytes on wire, 62 bytes captured)
  Arrival Time: May  8, 2009 10:13:07.933918000
  [Time delta from previous captured frame: 0.000506000 seconds]
  [Time delta from previous displayed frame: 0.000506000 seconds]
  [Time since reference or first frame: 66.901867000 seconds]
  Frame Number: 3
  Frame Length: 62 bytes
  Capture Length: 62 bytes
  [Frame is marked: False]
  [Protocols in frame: eth:ip:icmp:data]
  [Coloring Rule Name: ICMP]
  [Coloring Rule String: icmp]
Ethernet II, Src: D-Link_48:2b:84 (00:05:5d:48:2b:84), Dst: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
  Destination: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
    Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
      ....0 .... = IG bit: Individual address (unicast)
      ....0. .... = LG bit: Globally unique address (factory default)
  Source: D-Link_48:2b:84 (00:05:5d:48:2b:84)
    Address: D-Link_48:2b:84 (00:05:5d:48:2b:84)
      ....0 .... = IG bit: Individual address (unicast)
      ....0. .... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.252 (192.168.0.252), Dst: 192.168.0.67 (192.168.0.67)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
      ....0. = ECN-Capable Transport (ECT): 0
      ....0. = ECN-CE: 0
  Total Length: 48
  Identification: 0x0000 (0)
  Flags: 0x04 (Don't Fragment)

```

```

    0... = Reserved bit: Not set
    .1.. = Don't fragment: Set
    ..0. = More fragments: Not set
Fragment offset: 0
Time to live: 64
Protocol: ICMP (0x01)
Header checksum: 0xb83d [correct]
    [Good: True]
    [Bad : False]
Source: 192.168.0.252 (192.168.0.252)
Destination: 192.168.0.67 (192.168.0.67)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0 ()
Checksum: 0xd3c8 [correct]
Identifier: 0x2437
Sequence number: 0 (0x0000)
Data (20 bytes)

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0010 00 00 00 00 .....
      Data: 0000000000000000000000000000000000000000000000000000

```

Rien de bien particulier, un ping ICMP classique que le serveur fait sur l'adresse qu'il compte fournir à son client.

5-1-2-3 - Offer

```

Frame 3 (342 bytes on wire, 342 bytes captured)
Arrival Time: May  8, 2009 10:13:08.934897000
[Time delta from previous captured frame: 1.000979000 seconds]
[Time delta from previous displayed frame: 1.000979000 seconds]
[Time since reference or first frame: 67.902846000 seconds]
Frame Number: 4
Frame Length: 342 bytes
Capture Length: 342 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:bootp]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]
Ethernet II, Src: D-Link_48:2b:84 (00:05:5d:48:2b:84), Dst: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Ici, ce n'est plus du broadcast
Destination: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
.... 0... = IG bit: Individual address (unicast)
.... ..0. = LG bit: Globally unique address (factory default)
Source: D-Link_48:2b:84 (00:05:5d:48:2b:84)
Address: D-Link_48:2b:84 (00:05:5d:48:2b:84)
.... 0... = IG bit: Individual address (unicast)
.... ..0. = LG bit: Globally unique address (factory default)
Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.252 (192.168.0.252), Dst: 192.168.0.67 (192.168.0.67)
Le serveur répond au client sur sa potentielle future adresse IP.
Notez bien que le client ne la connait pas encore...
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00)
  0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
  .... ..0. = ECN-Capable Transport (ECT): 0
  .... ...0 = ECN-CE: 0
Total Length: 328
Identification: 0x0000 (0)
Flags: 0x00
  0... = Reserved bit: Not set
  .0.. = Don't fragment: Not set
  ..0. = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: UDP (0x11)
Header checksum: 0xb705 [correct]
  [Good: True]

```

```
[Bad : False]
Source: 192.168.0.252 (192.168.0.252)
Destination: 192.168.0.67 (192.168.0.67)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Source port: bootps (67)
Destination port: bootpc (68)
Length: 308
Checksum: 0x2a4d [correct]
  [Good Checksum: True]
  [Bad Checksum: False]
Bootstrap Protocol
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xc0b5592f
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
  0... .... .... .... = Broadcast flag: Unicast
  .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 192.168.0.67 (192.168.0.67)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Offer
  Option: (53) DHCP Message Type
  Length: 1
  Value: 02
Option: (t=54,l=4) Server Identifier = 192.168.0.252
  Option: (54) Server Identifier
  Length: 4
  Value: C0A800FC
Option: (t=51,l=4) IP Address Lease Time = 1 hour
  Option: (51) IP Address Lease Time
  Length: 4
  Value: 00000E10
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
  Option: (1) Subnet Mask
  Length: 4
  Value: FFFFFFF0
Option: (t=3,l=4) Router = 192.168.0.252
  Option: (3) Router
  Length: 4
  Value: C0A800FC
Option: (t=15,l=10) Domain Name = "maison.mrs"
  Option: (15) Domain Name
  Length: 10
  Value: 6D6169736F6E2E6D7273
Option: (t=6,l=4) Domain Name Server = 192.168.0.252
  Option: (6) Domain Name Server
  Length: 4
  Value: C0A800FC
Option: (t=44,l=4) NetBIOS over TCP/IP Name Server = 192.168.0.252
  Option: (44) NetBIOS over TCP/IP Name Server
  Length: 4
  Value: C0A800FC
End Option
Padding
```

Le serveur propose donc à notre client une configuration complète, avec tous les paramètres demandés que le serveur est en état de fournir.

5-1-2-4 - Request

Frame 4 (342 bytes on wire, 342 bytes captured)

```
Arrival Time: May 8, 2009 10:13:08.936831000
[Time delta from previous captured frame: 0.001934000 seconds]
[Time delta from previous displayed frame: 0.001934000 seconds]
[Time since reference or first frame: 67.904780000 seconds]
Frame Number: 5
Frame Length: 342 bytes
Capture Length: 342 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:bootp]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]
Ethernet II, Src: QuantaCo_51:5d:5a (00:16:36:51:5d:5a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... .1. .... = IG bit: Group address (multicast/broadcast)
      .... .1. .... = LG bit: Locally administered address (this is NOT the factory
default)
    Source: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
      Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
        .... .0. .... = IG bit: Individual address (unicast)
        .... .0. .... = LG bit: Globally unique address (factory default)
    Type: IP (0x0800)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00)
    0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
    .... .0. = ECN-Capable Transport (ECT): 0
    .... .0. = ECN-CE: 0
  Total Length: 328
  Identification: 0x0000 (0)
  Flags: 0x00
    0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..0. = More fragments: Not set
  Fragment offset: 0
  Time to live: 128
  Protocol: UDP (0x11)
  Header checksum: 0x3996 [correct]
    [Good: True]
    [Bad : False]
  Source: 0.0.0.0 (0.0.0.0)
  Destination: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
  Source port: bootpc (68)
  Destination port: bootps (67)
  Length: 308
  Checksum: 0xd980 [correct]
    [Good Checksum: True]
    [Bad Checksum: False]
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xc0b5592f
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
    0... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent IP address: 0.0.0.0 (0.0.0.0)
  Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
  Server host name not given
  Boot file name not given
  Magic cookie: (OK)
  Option: (t=53,l=1) DHCP Message Type = DHCP Request
    Option: (53) DHCP Message Type
    Length: 1
    Value: 03
```

```

Option: (t=54,l=4) Server Identifier = 192.168.0.252
  Option: (54) Server Identifier
  Length: 4
  Value: C0A800FC
Option: (t=50,l=4) Requested IP Address = 192.168.0.67
  Option: (50) Requested IP Address
  Length: 4
  Value: C0A80043
Option: (t=55,l=12) Parameter Request List
  Option: (55) Parameter Request List
  Length: 12
  Value: 011C02030F06770C2C2F1A79
  1 = Subnet Mask
  28 = Broadcast Address
  2 = Time Offset
  3 = Router
  15 = Domain Name
  6 = Domain Name Server
  119 = Domain Search
  12 = Host Name
  44 = NetBIOS over TCP/IP Name Server
  47 = NetBIOS over TCP/IP Scope
  26 = Interface MTU
  121 = Classless Static Route
End Option
Padding

```

Notre client effectue sa requête, toujours en broadcast. Il indique cependant :

- l'adresse IP du serveur DHCP auprès duquel il fait la demande, évitant ainsi, s'il y a d'autres serveurs, qu'ils poursuivent le dialogue ;
- l'adresse IP qu'il accepte.

Il n'a pas d'autres exigences.

5-1-2-5 - ACK

```

Frame 5 (342 bytes on wire, 342 bytes captured)
  Arrival Time: May 8, 2009 10:13:08.962553000
  [Time delta from previous captured frame: 0.025722000 seconds]
  [Time delta from previous displayed frame: 0.025722000 seconds]
  [Time since reference or first frame: 67.930502000 seconds]
  Frame Number: 6
  Frame Length: 342 bytes
  Capture Length: 342 bytes
  [Frame is marked: False]
  [Protocols in frame: eth:ip:udp:bootp]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
Ethernet II, Src: D-Link_48:2b:84 (00:05:5d:48:2b:84), Dst: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
  Destination: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
    Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
      ....0.... = IG bit: Individual address (unicast)
      ....0.... = LG bit: Globally unique address (factory default)
  Source: D-Link_48:2b:84 (00:05:5d:48:2b:84)
    Address: D-Link_48:2b:84 (00:05:5d:48:2b:84)
      ....0.... = IG bit: Individual address (unicast)
      ....0.... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.252 (192.168.0.252), Dst: 192.168.0.67 (192.168.0.67)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00)
    0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
    ....0. = ECN-Capable Transport (ECT): 0
    ....0. = ECN-CE: 0
  Total Length: 328
  Identification: 0x0000 (0)

```

```
Flags: 0x00
  0... = Reserved bit: Not set
  .0.. = Don't fragment: Not set
  ..0. = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: UDP (0x11)
Header checksum: 0xb705 [correct]
  [Good: True]
  [Bad : False]
Source: 192.168.0.252 (192.168.0.252)
Destination: 192.168.0.67 (192.168.0.67)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Source port: bootps (67)
Destination port: bootpc (68)
Length: 308
Checksum: 0x274d [correct]
  [Good Checksum: True]
  [Bad Checksum: False]
Bootstrap Protocol
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xc0b5592f
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
  0... .... .... .... = Broadcast flag: Unicast
  .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 192.168.0.67 (192.168.0.67)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
  Option: (53) DHCP Message Type
  Length: 1
  Value: 05
Option: (t=54,l=4) Server Identifier = 192.168.0.252
  Option: (54) Server Identifier
  Length: 4
  Value: C0A800FC
Option: (t=51,l=4) IP Address Lease Time = 1 hour
  Option: (51) IP Address Lease Time
  Length: 4
  Value: 00000E10
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
  Option: (1) Subnet Mask
  Length: 4
  Value: FFFFFFF0
Option: (t=3,l=4) Router = 192.168.0.252
  Option: (3) Router
  Length: 4
  Value: C0A800FC
Option: (t=15,l=10) Domain Name = "maison.mrs"
  Option: (15) Domain Name
  Length: 10
  Value: 6D6169736F6E2E6D7273
Option: (t=6,l=4) Domain Name Server = 192.168.0.252
  Option: (6) Domain Name Server
  Length: 4
  Value: C0A800FC
Option: (t=44,l=4) NetBIOS over TCP/IP Name Server = 192.168.0.252
  Option: (44) NetBIOS over TCP/IP Name Server
  Length: 4
  Value: C0A800FC
End Option
Padding
```

Le serveur donne donc son accord pour le bail avec ses paramètres définitifs. Il n'y a plus ici de broadcast, le serveur s'adresse en unicast à son client.

5-2 - Renouvellement

Lorsque nous sommes arrivé à l'heure renew, notre client va contacter le serveur DHCP :

```

No.      Time           Source           Destination      Protocol  Info
  1  0.000000    192.168.0.67    192.168.0.252    DHCP      DHCP Request - Transaction ID
    0xc0b5592f

Frame 1 (342 bytes on wire (342 bytes captured) on interface eth0)
  Arrival Time: May  8, 2009 10:38:01.936555000
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
  Frame Number: 1
  Frame Length: 342 bytes
  Capture Length: 342 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:udp:bootp]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]

Ethernet II, Src: QuantaCo_51:5d:5a (00:16:36:51:5d:5a), Dst: D-Link_48:2b:84 (00:05:5d:48:2b:84)
  Destination: D-Link_48:2b:84 (00:05:5d:48:2b:84)
    Address: D-Link_48:2b:84 (00:05:5d:48:2b:84)
      .... 0... = IG bit: Individual address (unicast)
      .... 0... = LG bit: Globally unique address (factory default)
  Source: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
    Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
      .... 0... = IG bit: Individual address (unicast)
      .... 0... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)

Internet Protocol, Src: 192.168.0.67 (192.168.0.67), Dst: 192.168.0.252 (192.168.0.252)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... 0.. = ECN-Capable Transport (ECT): 0
      .... 0.. = ECN-CE: 0
  Total Length: 328
  Identification: 0x0000 (0)
  Flags: 0x04 (Don't Fragment)
    0... = Reserved bit: Not set
    .1.. = Don't fragment: Set
    ..0. = More fragments: Not set
  Fragment offset: 0
  Time to live: 64
  Protocol: UDP (0x11)
  Header checksum: 0xb715 [correct]
    [Good: True]
    [Bad : False]
  Source: 192.168.0.67 (192.168.0.67)
  Destination: 192.168.0.252 (192.168.0.252)

User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
  Source port: bootpc (68)
  Destination port: bootps (67)
  Length: 308
  Checksum: 0x2eef [correct]
    [Good Checksum: True]
    [Bad Checksum: False]

Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xc0b5592f
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)

```

```

0... .. = Broadcast flag: Unicast
.000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 192.168.0.67 (192.168.0.67)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
  Option: (53) DHCP Message Type
  Length: 1
  Value: 03
Option: (t=55,l=12) Parameter Request List
  Option: (55) Parameter Request List
  Length: 12
  Value: 011C02030F06770C2C2F1A79
  1 = Subnet Mask
  28 = Broadcast Address
  2 = Time Offset
  3 = Router
  15 = Domain Name
  6 = Domain Name Server
  119 = Domain Search
  12 = Host Name
  44 = NetBIOS over TCP/IP Name Server
  47 = NetBIOS over TCP/IP Scope
  26 = Interface MTU
  121 = Classless Static Route
End Option
Padding

```

Et le serveur répond :

No.	Time	Source	Destination	Protocol	Info	
	2 0.027503	192.168.0.252	192.168.0.67	DHCP	DHCP ACK	- Transaction ID 0xc0b5592f

```

Frame 2 (342 bytes on wire, 342 bytes captured)
Arrival Time: May 8, 2009 10:38:01.964058000
[Time delta from previous captured frame: 0.027503000 seconds]
[Time delta from previous displayed frame: 0.027503000 seconds]
[Time since reference or first frame: 0.027503000 seconds]
Frame Number: 2
Frame Length: 342 bytes
Capture Length: 342 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:bootp]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]
Ethernet II, Src: D-Link_48:2b:84 (00:05:5d:48:2b:84), Dst: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Destination: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
.... ..0 .... = IG bit: Individual address (unicast)
.... ..0. .... = LG bit: Globally unique address (factory default)
Source: D-Link_48:2b:84 (00:05:5d:48:2b:84)
Address: D-Link_48:2b:84 (00:05:5d:48:2b:84)
.... ..0 .... = IG bit: Individual address (unicast)
.... ..0. .... = LG bit: Globally unique address (factory default)
Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.252 (192.168.0.252), Dst: 192.168.0.67 (192.168.0.67)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
.... ..0 = ECN-CE: 0
Total Length: 328
Identification: 0x0000 (0)
Flags: 0x04 (Don't Fragment)

```

```
0... = Reserved bit: Not set
.1.. = Don't fragment: Set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 64
Protocol: UDP (0x11)
Header checksum: 0xb715 [correct]
  [Good: True]
  [Bad : False]
Source: 192.168.0.252 (192.168.0.252)
Destination: 192.168.0.67 (192.168.0.67)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Source port: bootps (67)
Destination port: bootpc (68)
Length: 308
Checksum: 0x6661 [correct]
  [Good Checksum: True]
  [Bad Checksum: False]
Bootstrap Protocol
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xc0b5592f
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
  0... .... .... .... = Broadcast flag: Unicast
  .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 192.168.0.67 (192.168.0.67)
Your (client) IP address: 192.168.0.67 (192.168.0.67)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
  Option: (53) DHCP Message Type
  Length: 1
  Value: 05
Option: (t=54,l=4) Server Identifier = 192.168.0.252
  Option: (54) Server Identifier
  Length: 4
  Value: C0A800FC
Option: (t=51,l=4) IP Address Lease Time = 1 hour
  Option: (51) IP Address Lease Time
  Length: 4
  Value: 00000E10
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
  Option: (1) Subnet Mask
  Length: 4
  Value: FFFFFFF0
Option: (t=3,l=4) Router = 192.168.0.252
  Option: (3) Router
  Length: 4
  Value: C0A800FC
Option: (t=15,l=10) Domain Name = "maison.mrs"
  Option: (15) Domain Name
  Length: 10
  Value: 6D6169736F6E2E6D7273
Option: (t=6,l=4) Domain Name Server = 192.168.0.252
  Option: (6) Domain Name Server
  Length: 4
  Value: C0A800FC
Option: (t=44,l=4) NetBIOS over TCP/IP Name Server = 192.168.0.252
  Option: (44) NetBIOS over TCP/IP Name Server
  Length: 4
  Value: C0A800FC
End Option
Padding
```

Notez que dans le dialogue, le client annonce cette fois-ci son adresse IP et que le serveur la lui confirme. Toutes les autres options peuvent changer d'un bail à l'autre, ce qui permet, lorsque l'administrateur a planifié par exemple un changement d'adresse de passerelle ou de DNS, de s'arranger pour que la modification se passe en douceur dans un laps de temps que l'on peut estimer.

Ce renouvellement se fait entièrement en mode unicast.

5-3 - Le grain de sable

Tout ceci est parfait, mais imaginons que notre serveur DHCP tombe en panne. Que va-t-il se produire ? Faisons la manip. Nous jouons un sale tour à notre client en posant sur le serveur DHCP la règle IPTables :

```
iptables -A OUTPUT -d 192.168.0.67 -j DROP
```

Alors...

5-3-1 - Renew

À l'heure dite, le client va lancer un renew, mais le serveur ne répond pas...

Le client insiste :

No.	Time	Source	Destination	Protocol	Info
0xc0b5592f	1 0.000000	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	2 4.995823	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	3 14.995826	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	4 29.995825	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	5 44.995826	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	6 51.995826	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	7 62.995841	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	8 74.995824	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	9 95.995830	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	10 116.995824	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	11 129.995829	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	12 148.995836	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	13 163.995835	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	14 177.995833	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	15 193.995844	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	16 211.995839	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
0xc0b5592f	17 225.995830	192.168.0.67	192.168.0.252	DHCP	DHCP Request - Transaction ID
...					

Admirez la patience (obstination ?) de notre client qui va sans relâche insister jusqu'à :

No.	Time	Source	Destination	Protocol	Info
-----	------	--------	-------------	----------	------

1	0.000000	192.168.0.67	192.168.0.252	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
2	15.000000	192.168.0.67	192.168.0.252	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
3	23.999995	192.168.0.67	192.168.0.252	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
4	39.999999	192.168.0.67	192.168.0.252	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
5	58.000005	192.168.0.67	192.168.0.252	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
6	78.999998	192.168.0.67	192.168.0.252	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
7	89.999997	192.168.0.67	192.168.0.252	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
8	111.000002	192.168.0.67	255.255.255.255	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
9	131.999996	192.168.0.67	255.255.255.255	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
10	151.999988	192.168.0.67	255.255.255.255	DHCP	DHCP Request	- Transaction ID 0xc0b5592f
11	159.999990	192.168.0.67	255.255.255.255	DHCP	DHCP Request	- Transaction ID 0xc0b5592f

Notre client change de tactique. Il n'interroge plus 192.168.0.252. Il a fini par se rendre à l'évidence : ce serveur est hors service. Il va alors, tout en conservant son adresse IP actuelle, commencer une série de requêtes en broadcast, des fois qu'un bon administrateur aurait mis en place un autre DHCP, mais avec une autre adresse IP.

Voyons le détail des paquets 7 et 8 :

```

Frame 7 (342 bytes on wire, 342 bytes captured)
  Arrival Time: May  8, 2009 16:25:50.932358000
  [Time delta from previous captured frame: 10.999999000 seconds]
  [Time delta from previous displayed frame: 10.999999000 seconds]
  [Time since reference or first frame: 89.999997000 seconds]
  Frame Number: 7
  Frame Length: 342 bytes
  Capture Length: 342 bytes
  [Frame is marked: False]
  [Protocols in frame: eth:ip:udp:bootp]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
Ethernet II, Src: QuantaCo_51:5d:5a (00:16:36:51:5d:5a), Dst: D-Link_48:2b:84 (00:05:5d:48:2b:84)
  Destination: D-Link_48:2b:84 (00:05:5d:48:2b:84)
    Address: D-Link_48:2b:84 (00:05:5d:48:2b:84)
      ....0... = IG bit: Individual address (unicast)
      ...0... = LG bit: Globally unique address (factory default)
  Source: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
    Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
      ....0... = IG bit: Individual address (unicast)
      ...0... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.67 (192.168.0.67), Dst: 192.168.0.252 (192.168.0.252)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    ....0. = ECN-Capable Transport (ECT): 0
    ...0.0 = ECN-CE: 0
  Total Length: 328
  Identification: 0x0000 (0)
  Flags: 0x04 (Don't Fragment)
    0... = Reserved bit: Not set
    .1.. = Don't fragment: Set
    ..0. = More fragments: Not set
  Fragment offset: 0
  Time to live: 64
  Protocol: UDP (0x11)
  Header checksum: 0xb715 [correct]
    [Good: True]
    [Bad : False]

```

```
Source: 192.168.0.67 (192.168.0.67)
Destination: 192.168.0.252 (192.168.0.252)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Source port: bootpc (68)
Destination port: bootps (67)
Length: 308
Checksum: 0x27f7 [correct]
  [Good Checksum: True]
  [Bad Checksum: False]
Bootstrap Protocol
Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xc0b5592f
Seconds elapsed: 1784
Bootp flags: 0x0000 (Unicast)
  0... .... .... .... = Broadcast flag: Unicast
  .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 192.168.0.67 (192.168.0.67)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
  Option: (53) DHCP Message Type
  Length: 1
  Value: 03
Option: (t=55,l=12) Parameter Request List
  Option: (55) Parameter Request List
  Length: 12
  Value: 011C02030F06770C2C2F1A79
  1 = Subnet Mask
  28 = Broadcast Address
  2 = Time Offset
  3 = Router
  15 = Domain Name
  6 = Domain Name Server
  119 = Domain Search
  12 = Host Name
  44 = NetBIOS over TCP/IP Name Server
  47 = NetBIOS over TCP/IP Scope
  26 = Interface MTU
  121 = Classless Static Route
End Option
Padding
```

Paquet 7, la requête est bien encore unicast. Dans la suivante :

```
Frame 8 (342 bytes on wire, 342 bytes captured)
Arrival Time: May 8, 2009 16:26:11.932363000
[Time delta from previous captured frame: 21.000005000 seconds]
[Time delta from previous displayed frame: 21.000005000 seconds]
[Time since reference or first frame: 111.000002000 seconds]
Frame Number: 8
Frame Length: 342 bytes
Capture Length: 342 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:bootp]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]
Ethernet II, Src: QuantaCo_51:5d:5a (00:16:36:51:5d:5a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Address: Broadcast (ff:ff:ff:ff:ff:ff)
  .... .1. .... .... .... = IG bit: Group address (multicast/broadcast)
  .... .1. .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
Source: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
```

```
Address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
.... ..0 .... = IG bit: Individual address (unicast)
.... ..0 .... = LG bit: Globally unique address (factory default)
Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.67 (192.168.0.67), Dst: 255.255.255.255 (255.255.255.255)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00)
 0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
.... ..0. = ECN-Capable Transport (ECT): 0
.... ..0 = ECN-CE: 0
Total Length: 328
Identification: 0x0000 (0)
Flags: 0x00
 0... = Reserved bit: Not set
 .0.. = Don't fragment: Not set
 ..0. = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: UDP (0x11)
Header checksum: 0x78aa [correct]
 [Good: True]
 [Bad : False]
Source: 192.168.0.67 (192.168.0.67)
Destination: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Source port: bootpc (68)
Destination port: bootps (67)
Length: 308
Checksum: 0xe986 [correct]
 [Good Checksum: True]
 [Bad Checksum: False]
Bootstrap Protocol
Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xc0b5592f
Seconds elapsed: 1805
Bootp flags: 0x0000 (Unicast)
 0... ..0 = Broadcast flag: Unicast
 .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 192.168.0.67 (192.168.0.67)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: QuantaCo_51:5d:5a (00:16:36:51:5d:5a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
Option: (53) DHCP Message Type
Length: 1
Value: 03
Option: (t=55,l=12) Parameter Request List
Option: (55) Parameter Request List
Length: 12
Value: 011C02030F06770C2C2F1A79
 1 = Subnet Mask
 28 = Broadcast Address
 2 = Time Offset
 3 = Router
 15 = Domain Name
 6 = Domain Name Server
 119 = Domain Search
 12 = Host Name
 44 = NetBIOS over TCP/IP Name Server
 47 = NetBIOS over TCP/IP Scope
 26 = Interface MTU
 121 = Classless Static Route
End Option
Padding
```

Nous avons bien ici du broadcast, mais le reste de la requête reste inchangé. Le client conserve l'espoir de trouver un autre serveur DHCP qui lui renouvellera son bail actuel. Notre client a changé de tactique à l'heure rebind indiquée dans le bail précédent.

Cependant notre règle IPTables est encore plus obstinée que notre client, il n'y a pas de nouveau serveur DHCP sur le réseau et finalement, le bail expire à l'heure expire :

5-3-2 - Mort (et résurrection)

No.	Time	Source	Destination	Protocol	Info
0xc0b5592f	1 0.000000	192.168.0.67	255.255.255.255	DHCP	DHCP Request - Transaction ID
0xc0b5592f	2 14.999984	192.168.0.67	255.255.255.255	DHCP	DHCP Request - Transaction ID
0xc0b5592f	3 24.999987	192.168.0.67	255.255.255.255	DHCP	DHCP Request - Transaction ID
0xc0b5592f	4 38.999972	192.168.0.67	255.255.255.255	DHCP	DHCP Request - Transaction ID
0xc0b5592f	5 47.999964	192.168.0.67	255.255.255.255	DHCP	DHCP Request - Transaction ID
0xc0b5592f	6 64.999972	192.168.0.67	255.255.255.255	DHCP	DHCP Request - Transaction ID
0xf4b2bf16	7 71.045835	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID
0xf4b2bf16	8 71.046413	192.168.0.252	192.168.0.67	DHCP	DHCP Offer - Transaction ID
0xf4b2bf16	9 71.046704	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID
0xf4b2bf16	10 71.115898	192.168.0.252	192.168.0.67	DHCP	DHCP ACK - Transaction ID

La trame 7 montre que notre client a perdu son adresse IP, mais il ne renonce toujours pas. Il fait maintenant une recherche de DHCP (Discover) et là, le miracle se produit, il redécouvre notre DHCP qui lui attribue alors un bail tout neuf. Contre toute attente, l'histoire finit bien.

Mais est-ce vraiment un miracle ?

En réalité, l'obstination de notre client a eu raison de celle de la règle IPTables. En effet, notre Netfilter ne laisse rien sortir vers 192.168.0.67, mais notre client a repris l'adresse factice 0.0.0.0 et Netfilter, leurré, laisse tomber son fromage. Notre client ne manque alors pas de s'en saisir.

S'il s'était agi d'une vraie panne de DHCP, il n'y aurait pas eu de fromage et notre client serait resté le bec dans l'eau.

6 - Conclusion

Nous avons pu constater ici que DHCP est un protocole extrêmement opiniâtre et prudent. Le client se laisse de la marge en cas d'accident et commence à demander un renouvellement bien avant l'heure d'expiration, en cas d'accident, il essaye de retrouver un autre serveur qui lui renouvellerait son bail, et même mort, il essaye encore.

Il n'aura pas échappé au lecteur attentif que, bien que le bail ait expiré, le nouveau bail récupéré par tromperie de Netfilter propose la même adresse IP que la précédente. Est-ce un hasard ?

Pas du tout. En réalité le serveur garde en mémoire toutes les informations concernant les baux qu'il distribue et dans toute la mesure du possible, cherchera à attribuer la même adresse IP à une adresse MAC donnée. C'est généralement le cas, sauf lorsqu'il y a pénurie d'adresses IP. Le serveur est alors obligé de donner des adresses déjà attribuées, mais libérées, à de nouveaux clients.

Nous n'avons pas vu en détail toutes les possibilités de DHCP, mais ce chapitre a pu montrer le principe de base. Un client peut avoir quelques exigences sur divers paramètres, qu'il va alors annoncer dans sa requête. Le serveur

pourra ou non satisfaire à ces exigences une négociation plus ou moins serrée pourra s'en suivre terminée par une entente cordiale ou non. Ce genre de situation reste cependant assez rare.

Pour tout savoir sur DHCP, le mieux est de poursuivre par la lecture des **RFC 2131 : « Dynamic Host Configuration Protocol »** et aussi pourquoi pas, des **RFC 2132 : « DHCP Options and BOOTP Vendor Extensions »** .

6-1 - Remerciements Developpez

Vous pouvez retrouver l'article original ici : **L'Internet Rapide et Permanent**. Christian Caleca a aimablement autorisé l'équipe « **Réseaux** » de **Developpez.com** à reprendre son article. Retrouvez tous les articles de Christian Caleca **sur cette page**.

Nos remerciements à **ClaudeLELOUP** pour la relecture orthographique de cet article.

N'hésitez pas à commenter cet article !