Alias IP avec FreeBSD 2.x

Ben Black

Copyright © 1999 Groupe de Documentation de FreeBSD. Tous droits réservés.

<black@cypher.net>

La redistribution du code source (SGML), modifié ou non, et compilé (HTML, PostScript, etc.) est soumise aux conditions suivantes :

1. Le copyright ci-dessus, la présente liste de conditions et l'avertissement qui la suit doivent figurer dans le code source.
2. Le code source distribué sous forme compilée doit faire apparaître le copyright ci-dessus, la présente liste de conditions et l'avertissement qui la suit.

CE DOCUMENT EST FOURNI "TEL QU'EN L'ÉTAT" PAR LE PROJET DE DOCUMENTATION FRANÇAISE DE FreeBSD ET IL N'EST DONNÉ AUCUNE GARANTIE, IMPLICITE OU EXPLICITE, QUANT À SON UTILISATION COMMERCIALE, PROFESSIONNELLE OU AUTRE. LES COLLABORATEURS DU PROJET DE DOCUMENTATION FRANÇAISE DE FreeBSD NE PEUVENT EN AUCUN CAS ÊTRE TENUS POUR RESPONSABLES DE QUELQUE DOMMAGE OU PRÉJUDICE DIRECT, INDIRECT, SECONDAIRE OU ACCESSOIRE (Y COMPRIS LES PERTES FINANCIèRES DUES AU MANQUE À GAGNER, À L'INTERRUPTION D'ACTIVITÉS, OU LA PERTE D'INFORMATIONS ET AUTRES) DÉCOULANT DE L'UTILISATION DE LA DOCUMENTATION OU DE L'IMPOSSIBILITÉ D'UTILISER CELLE-CI, ET DONT L'UTILISATEUR ACCEPTE L'ENTIÈRE RESPONSABILITÉ.

Version française de Frédéric Haby <frederic.haby@mail.dotcom.fr>.

1. Introduction

Définir des *alias IP* consiste à affecter plus d'une adresse IP à une interface réseau. Les alias IP sont typiquement employés pour héberger des serveurs Web et FTP virtuels et pour réorganiser ses serveurs sans avoir à mettre à jour d'autres machines (ce dernier usage est particulièrement utile avec les serveurs de noms de domaines). Voyez un alias IP comme un petit enfant perché sur le dos de son interface parente (qui est l'adresse primaire de l'interface physique).

Les alias IP *n'ont rien à voir* avec la notion de *multi-hébergement* - "*multi-homing*" (lorsqu'une machine dispose de plus d'une interface réseau active), sauf au cas où une machine sur plusieurs réseaux fournit des services tels que ceux listés plus haut. La confusion entre les deux notions est une erreur propagée du monde Macintosh et vous seriez avisé de ne pas les suivre sur ce point.

Il y a deux commandes essentielles à la mise en place d'alias IP sur FreeBSD (et sur tous les autres systèmes Unix que j'aie jamais utilisés), ce sont `ifconfig` et `route`. L'utilisation correcte de ces commandes est décrite

ci-dessous. Pour connaître toutes les subtilités de leur emploi, reportez-vous aux pages de manuel correspondantes. Il y a deux autres commandes utiles pour vérifier que tout fonctionne comme il se doit et pour mettre les choses au point si ce n'est pas le cas. `netstat` vous permet de consulter la table de routage du noyau (entre autres choses utiles), et ping est un appréciable petit outil pour tester l'accès à votre nouvel alias (comme à toute autre adresse IP).

> **Important :** Vous *devez* être en session sous le compte super-utilisateur `root` pour pouvoir exécuter la plupart des commandes décrites dans ce qui suit.

2. Utiliser `ifconfig`

ifconfig sert à configurer les différentes caractéristiques d'une interface réseau (adresse, protocole, masque de réseau, etc.). ifconfig nous servira ici à ajouter (et parfois à supprimer) les adresses des alias des interfaces.

Dans l'exemple ci-dessous, nous supposerons que notre machine n'a qu'une seule interface Ethernet 3Com 3C509 (`ep0`) d'adresse `192.168.57.82` avec comme masque de sous-réseau `255.255.255.0`, en plus de l'interface "en boucle" standard `127.0.0.1`. Vous devrez bien sûr remplacer ces valeurs par l'adresse et le masque de sous-réseau de votre(vos) interface(s). L'adresse et le masque de sous-réseau de l'interface "en boucle" sont les mêmes sur toutes les machines.

Pour ajouter un alias à `ep0`, nous passons simplement une commande du style:

```
# ifconfig ep0 inet 192.168.57.10 netmask 255.255.255.255 alias
```

Ce qui dit à FreeBSD que nous définissons une adresse IP (`inet`) pour la carte 3Com (`ep0`) avec comme adresse `192.168.57.10`, comme masque de sous-réseau (`netmask`) `255.255.255.255` (i.e., tous les bits à 1) pour éviter que le système d'exploitation ne proteste qu'il y a déjà une adresse affectée à cette interface, et que c'est un alias pour cette interface (`alias`).

Vous pouvez aussi omettre le masque de sous-réseau (ou lui donner une autre valeur). Jusqu'à FreeBSD 2.1.0 compris, il y a néanmoins un bogue qui fait que les commandes `ifconfig alias` échouent à la première tentative, mais réussissent à la seconde, si vous n'indiquez pas `255.255.255.255` comme masque de sous-réseau. Si vous ne donnez pas ce masque de sous-réseau de `255.255.255.255`, vous devrez aussi ajouter une route manuellement, comme décrit ci-dessous.

Pour supprimer l'alias, nous exécutons une commande très similaire:

```
# ifconfig ep0 inet 192.168.57.10 -alias
```

Il n'y a pas besoin de paramètres supplémentaires, tels que le masque de sous-réseau, parce que l'alias est entièrement défini par l'adresse IP, de sorte que FreeBSD peut supprimer toutes les ressources associées en se basant sur cette adresse.

3. Utiliser `route`

route sert à modifier à la main la table de routage du noyau. Pour nos besoins, une fois que nous avons donné à FreeBSD l'alias pour l'interface, nous devons définir une route IP vers cet alias. Si vous avez précisé le masque de sous-réseau de `255.255.255.255` avec la commande précédente, vous pouvez vous dispenser de cette étape.

Pour ajouter une route vers notre nouvelle interface, nous exécutons:

```
# route add -host 192.168.57.10 127.0.0.1 0
```

Cela dit à FreeBSD que pour arriver à la machine `192.168.57.10`, les paquets doivent être envoyés sur l'interface "en boucle" (`127.0.0.1` aussi appelée `localhost` et que la distance (décompte de noeuds intermédiaires) est de `0`, ce qui signifie que c'est la machine locale.

Pour supprimer cette route, nous exécutons:

```
# route delete -host 192.168.57.10
```

De nouveau, comme la route est entièrement définie par les paramètres indiqués, nous pouvons laisser de côté les autres données telles que le masque de sous-réseau et la distance. Pour des informations complètes sur la commande route, voyez les pages de manuel.

4. Tester notre travail

Maintenant que nous avons (espérons-le) un alias défini et accessible, nous devons vérifier qu'il fonctionne correctement. Nous examinerons d'abord la table de routage du noyau, pour vérifier que la route vers l'alias est bien présente, puis nous utiliserons ping pour nous assurer que des paquets peuvent l'atteindre.

Pour voir la table de routage du noyau, sous forme numérique, tapons:

```
% netstat -nr
```

Dans notre exemple, voici à quoi ressemblerait le résultat:

```
Routing tables

Internet:
Destination        Gateway            Flags     Refs     Use     Netif Expire
default            192.168.57.1       UGSc         4     964       ep0
127.0.0.1          127.0.0.1          UH           1      39       lo0
192.168.57         link#2             UC           0       0
192.168.57.10/32   link#2             UC           0       0
192.168.57.82      127.0.0.1          UGHS         0       0       lo0
224/4              link#2             UCS          0       0
```

Nous y voyons la route par défaut, la route vers l'interface "en boucle", la route vers notre réseau local et une curieuse route vers notre alias. Le `/32` qui suit l'adresse de l'alias donne la longueur du masque de sous-réseau (le nombre de 1, en partant de la gauche). Comme nous avons utilisé un masque de sous-réseau de `255.255.255.255`, la longueur de celui-ci (le nombre de 1) est de 32.

Maintenant que nous avons vérifié que la route est bien dans la table de routage du noyau, nous pouvons effectuer un contrôle rapide pour nous assurer que cette route fonctionne bien. Exécutons une commande ping comme ci-dessous:

```
% ping 192.168.57.10
```

Si tout va bien, voici ce que nous verrions:

```
PING 192.168.57.10 (192.168.57.10): 56 data bytes
64 bytes from 192.168.57.10: icmp_seq=0 ttl=255 time=0.313 ms
64 bytes from 192.168.57.10: icmp_seq=1 ttl=255 time=0.197 ms
64 bytes from 192.168.57.10: icmp_seq=2 ttl=255 time=0.190 ms
```

Appuyons simplement sur **Ctrl**-**C** pour arrêter ping si tout paraît satisfaisant. Il nous "crachera" alors quelques statistiques, que nous pouvons normalement ignorer. S'il n'y a pas de retour ou s'il y a des messages d'erreur, quelque chose ne va pas. Reprenez tout du début et lisez les pages de manuel (non, vraiment, *allez lire les pages de manuel*), et, finalement, posez vos questions sur la liste de diffusion pour les questions d'ordre général à propos de FreeBSD (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions).

5. Tout mettre en place

Maintenant que nous avons compris comment utiliser les commandes ifconfig et route pour ajouter et supprimer des alias IP, faisons en sorte que la machine exécute les commandes voulues au démarrage de façon à ce que l'alias soit toujours actif.

Vers la fin de notre fichier `/etc/rc.local`, nous ajouterons la ligne:

```
ifconfig ep0 inet 192.168.57.10 netmask 255.255.255.255 alias
```

Si nous utilisons un autre masque de sous-réseau que `255.255.255.255`, nous devrons aussi utiliser la commande route comme expliqué plus haut. Il est possible d'ajouter de cette façon autant d'alias que l'on veut (en supposant que nous disposons de suffisamment d'adresses allouées). Il suffit de mettre plusieurs commandes comme la précédente dans `/etc/rc.local`, avec les adresses appropriées. Il est même possible de définir comme alias des adresses d'un autre réseau IP, si ce réseau est aussi routé sur nous. C'est une situation différente de celle d'une machine avec plusieurs interfaces *physiques* qui doivent *toujours* avoir des adresses sur des réseaux IP différents.

Il y a un certains nombre de points à connaître lorsque l'on utilise un grand nombre d'alias (plus de ~10), que je citerai.

Tout d'abord, il y a un bogue dans toutes les distributions de **bind** (le serveur DNS **named** fait partie de **BIND**) qui empêche le démarrage correct de **named** lorsqu'il y a plus de 64 alias sur une interface. Il n'y a pas de palliatif particulièrement simple, je ne les mentionnerai donc pas ici.

En second lieu, pour éviter de truffer votre fichier `/etc/rc.local` de quantité d'ifconfig, je vous suggère d'utiliser un fichier à part dans `/etc` pour les y mettre. Créez par exemple un fichier `ifconfig.ep0` et mettez-y tous les `ifconfig alias`. Puis dans `/etc/rc.local`, ajoutez les lignes suivantes:

```
if [ -x /sbin/ifconfig ]; then
        echo -n 'Adding local ifconfigs '
        . /etc/ifconfig.ep0
        echo ' done.'
fi
```

Ce fragment de procédure provient d'un fichier `/etc/rc.local` fourni par Neil Fowler Wright `<neil@corpex.com>`. Le bricoleur trouvera moyen d'y mettre des boucles s'il y a plusieurs interfaces dont les alias sont définis dans différents fichiers `ifconfig.*`.

6. Où trouver plus d'aide

M'envoyer directement un courrier électronique n'est probablement pas le meilleur moyen d'obtenir de l'aide sur les problèmes d'alias IP. Je suis assez occupé la plupart du temps, et ai tendance à devenir assez rouspéteur. Néanmoins, les participants de la liste de diffusion pour les questions d'ordre général à propos de FreeBSD (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) sont amicaux et secourables et répondront à vos questions plus rapidement que je ne le ferais.

Cette page a grandement bénéficié des suggestions de Neil Fowler Wright, Paul DuBois, Gabe Schuyler, et de Brodie le chat.