Mini-HOWTO ARP-Proxy

Al Longyear, longyear@netcom.com 5 Décembre 1994

(Traduction française par Eric Dumas Eric.Dumas@freenix.fr (Août 1995)).

Table des matières

1 Note du traducteur

Ce document utilise souvent le terme technique *proxy*. Il est utile de savoir que ce terme anglais signifie un mandataire, une procuration. Un serveur proxy peut alors être comparé à un serveur servant de mandataire à toutes les machines s'y connectant. Ce serveur essaie de répondre aux requêtes, dans ce cas précis, ARP.

2 Introduction

Ce document a été conçu pour aider les personnes qui veulent utiliser le système ARP Proxy (Protocole de Résolution d'Adresses) avec **Linux** dans le cas de serveurs PPP et SLIP.

ARP Proxy est également appelé *l'ARP gracieux* dans certains ouvrages. Il y a eu pas mal de demandes au sujet de l'utilisation de l'ARP Proxy. Lorsqu'il ne peut être utilisé, certaines personnes considèrent que c'est dû à un défaut du programme et se demandent pourquoi cela ne fonctionne pas.

J'espère qu'avec le support de ce document, les gens en sauront un peu plus à propos d'ARP proxy, que cela soit utile ou non.

L'utilisation d'ARP proxy est utile lorsque vous possédez un serveur. Il va permettre la connexion dynamique des machines distantes sans avoir besoin de mettre à jour les tables de routages sur les autres machines, excepté le serveur associé.

Le terme de *serveur* est fort peu approprié. TCP/IP est un environnement réseau *Peer to Peer*. Il n'y a pas de client ayant une relation avec un serveur comme d'autres systèmes avec des données partagées sur des serveurs que les clients exploitent. Toutefois, il est pratique d'appeler serveur un système qui réponde au téléphone et client, un système qui appelle pour se connecter au serveur.

Le programme de gestion réseau de **Linux** gère directement ARP proxy. Il n'y a pas besoin d'un démon particulier comme `proxyarpd` utilisé sur certains systèmes.

De plus, le protocole PPP, `pppd` et au moins l'un des codes SLIP, dip-uri, gèrent ARP proxy. Le programme réseau, `arp` va gérer et afficher la table.

Pour comprendre comment fonctionne ARP proxy et lorsqu'il doit être utilisé, vous devez avoir une connaissance de base concernant le fonctionnement d'un réseau en général.

Les trois paragraphes suivants vont décrire brièvement la gestion réseau TCP/IP et le fonctionnement du routage.

3 Le côté matériel du réseau

Tout réseau utilisant Ethernet ou Token Ring fonctionne en utilisant une adresse *MAC* (*Medium Acces Control*). Il s'agit d'une adresse matérielle réseau associée à un contrôleur spécifique. Chaque adresse MAC est unique. Elles sont assignées par le constructeur du contrôleur. Toutefois elles peuvent être remplacées par logiciel, ce qui n'est pas une règle générale.

Les adresses IP sont converties en adresses MAC en utilisant une table particulière à l'intéreur du logiciel réseau appelé *ARP cache*. Lorsque le logiciel réseau souhaite envoyer une trame IP à une adresse spécifique, il consulte ce cache pour déterminer l'adresse MAC. Si l'entrée n'est pas trouvée dans le cache, une requête particulière est envoyée à toutes les machines du réseau pour convertir l'adresse IP en une adresse MAC. Ceci s'appelle une requête ARP.

La réponse à la requête ARP est une réponse avec l'adresse MAC. Cette adresse MAC est alors ajoutée au cache pour que les conversions puissent être réalisées sans l'aide d'ARP.

C'est cette requête ARP qui est utilisée par le système d'ARP proxy pour réaliser la gestion des connexions distantes.

Il y a des règles selon lesquelles des entrées sont supprimées du cache. Ces règles ne sont pas détaillées dans ce document et sont laissées à une documentation concernant la description technique du réseau IP.

(Comme Token Ring est actuellement en dévelopement, et qu'il n'est disponible qu'en version de teste, le support de communication réseau courant sous **Linux** est Ethernet. J'emploierai le terme d'Ethernet à partir de maintenant. Des caractéristiques semblables sont disponibles pour Token Ring, indépendament du code source de routage.

4 Les raisons d'utiliser l'ARP Proxy ARP

Le principe d'ARP proxy est de permettre l'assignation de plus d'une adresse IP à une seule carte réseau.

Le fonctionnement consiste à la création d'une entrée dans le cache ARP de **Linux** en associant l'adresse IP supplémentaire à l'adresse matérielle du contrôleur Ethernet. Cela permet au système **Linux** de répondre à une requête ARP pour traduire une adresse IP en adresse matérielle.

5 Routage TCP/IP

Une courte préface est nécessaire. Elle décrit le principe de l'arbre de routage. Elle ne traite pas le routage source des trames IP. Le routage source réalisé par Token Ring n'est pas du routage IP source mais il s'agit d'une couche MAC qui réalise cette opération. L'utilisation du routage source IP est déconseillé. Le routage source MAC de Token Ring est nécessaire pour réaliser ce style de communication.

Pour comprendre un peu mieux ARP proxy, vous devez comprendre comment les trames IP sont routées à travers le réseau. Je ne vais pas trop détailler ce partie. Si vous voulez des informations supplémentaires, bon nombre de livres disponibles donnent des informations plus poussées. (Si vous ne voulez pas regarder les livres, vous pouvez alors lire les documents RFC.)

Les trames IP sont routées à chacune des étapes de leur passage à travers le réseau. Chaque machine, routeur ou passerelle décide de lui-même et avec sa propre copie de la table de routage ou la trame IP doit être envoyée.

Le routage est réalisé en utilisant un réseau IP (terme que j'utiliserai). A chaque interface réseau, on assigne un réseau IP unique. Chacun possède une adresse IP. Chacun possède un masque de réseau. Le réseau IP est simplement une opération binaire de l'adresse IP avec le masque réseau. Par exemple, l'adresse IP 10.124.35.40 et le masque de réseau 255.255.0.0 vont avoir un réseau IP de 10.124.0.0. J'utilise des masques réseau en octets mais la même logique peut s'appliquer à la limite à des masques réseaux qui ne le sont pas.

Linux associe le masque du réseau à l'entrée d'un chemin. Lorsque vous ajoutez un chemin dans le système, vous spécifiez une adresse IP et le périphérique de destination. Si vous ne spécifiez pas de masque réseau, celui-ci est choisi comme étant le masque de réseau par défaut du périphérique de destination. Il est positionné lorsque le périphérique est configuré par ifconfig.

Pour mieux comprendre le routage, regardez la configuration suivante :

```
Destination     Masque Reseau    Passerelle    Option   Peripherique

10.124.0.0      255.255.0.0      0.0.0.0       U        eth0
10.125.0.0      255.255.0.0      0.0.0.0       U        eth1
```

```
10.126.0.0      255.255.0.0      10.125.31.1   UG      eth1
10.124.12.5     255.255.255.255  0.0.0.0       UH      ppp0
0.0.0.0         0.0.0.0          10.124.25.1   U       eth0
```

Il s'agit d'un système possédant trois interfaces réseau. Il a deux contrôleurs Ethernet et un périphérique PPP. Les trames IP peuvent pénétrer dans ce système par n'importe laquelle des trois sources. En plus, des trames sont renvoyées à travers ce système à n'importe quel des trois périphériques de destination.

Le chemin par défaut est le périphérique de la passerelle 10.124.25.1 comme le montre la dernière entrée. Pour joindre la passerelle, la trame est envoyée à-travers le périphérique eth0.

Un a périphérique PPP connecté. Son adresse IP est 10.124.12.5.

Le périphérique eth0 est sur le réseau d'adresse IP 10.124.0.0 alors que le périphérique eth1 est sur celui d'adresse IP 10.125.0.0.

En plus, un chemin vers le réseau IP 10.126.0.0 est disponible par la passerelle associée 10.125.31.1.

Pour comprendre comment le routage est réalisé, considérons une trame IP qui doit être envoyée à l'adresse 10.125.45.1.

Linux va consulter la table de routage et pour chacune des entrées, prendre le masque de réseau, puis effectuer une opération logique (et) sur le masque réseau. Enfin il va le comparer à l'entrée de l'adresse IP de destination. Si le résultat correspond, la trame est envoyée au périphérique indiqué.

Le résultat est que la dite trame pour l'adresse IP 10.125.45.1 sera envoyée au périphérique eth1.

De plus, une trame émise de l'adresse IP 10.124.12.5 sera envoyée au périphérique ppp0 alors qu'une trame provenant de l'adresse IP 10.124.12.6 sera envoyée au périphérique eth0 car le périphérique accepte pour seule et unique adresse, l'adresse IP 10.124.12.5.

Les trames pour l'adresse 10.126.31.4 sont différentes. Elles ont une *passerelle* associée. Le problème est résolu de la même manière. Toutefois, au lieu d'envoyer les trames simplemement au périphérique eth1, elles sont envoyées au système unique associé à l'adresse IP 10.125.31.1. Il s'agit de l'adresse IP qui est transformée en une adresse MAC, plutôt qu'en une adresse de destination 10.126.31.4.

Lorsqu'elle arrive sur le système d'adresse 10.126.31.1, celui-ci va renvoyer la trame sur la destination finale 10.126.31.4 en utilisant sa propre table de routage qui peut lui indiquer de rediriger la trame sur l'interface eth3.

De nombreuse conditions d'erreurs qui sont détectées par ce système de routage. Je ne veux pas toutes les détailler, toutefois, si par exemple 10.126.31.1 ne connaît pas le chemin pour atteindre l'adresse .4, alors il devrait renvoyer une trame ICMP (Protocole de Messages de Contrôles d'Internet) à l'envoyeur initial comme quoi il n'a pas de chemin pour la machine spécifiée.

6 Routage avec ARP Proxy

Enfin, nous atteignons le sujet de ce document maintenant que toutes les bases ont étés posées.

Il faut se rappeler que **Linux** va mettre une entrée dans le cache ARP pour l'adresse IP et l'adresse matérielle MAC associée lorsque l'on utilise ARP Proxy. Souvenez-vous que c'est ce cache qui est utilisé pour convertir une adresse IP en une adresse MAC.

Lorsqu'un site distant se connecte à l'adresse IP 10.124.12.5, le système **Linux** ajoutera l'adresse IP ainsi que l'adresse MAC associée au contrôleur eth0, dans le cache ARP.

Lorsqu'il reçoit une requête pour convertir l'adresse 10.124.12.5 en une adresse MAC, **Linux** va envoyer l'entrée de ses tables au demandeur. Le résultat est que la trame pour cette adresse IP sera envoyée au serveur et peut alors la transmettre au site distant.

C'est de cette manière qu'ARP proxy fonctionne. Le serveur est un proxy (un agent, un intermédiaire, etc.) pour l'adresse IP du site distant. C'est le terme employé pour un réseau qui peut accepter des trames de l'adresse IP distante et le servir en répondant aux requêtes ARP.

Donc, pour qu'ARP proxy fonctionne, l'adresse IP du site (10.124.12.5 dans mon exemple) doit être l'une des adresses IP pour l'une des cartes réseaux.

Il y a deux raisons pour justifier cette obligation :

1. L'adresse MAC d'un contrôleur est entrée dans le cache ARP pour y être associée à l'adresse IP. Une adresse MAC est nécessaire pour l'assignation ARP depuis que le cache ARP est une conversion d'adresse IP en une adresse MAC.

2. Tout système du réseau réalise son propre routage. Ces systèmes savent que pour envoyer une trame IP à un autre système distant ayant une adresse IP, il doit 'mettre la trame dans le même fil' qui est branché à la carte réseau.

7 Lorsque l'ARP Proxy ne fonctionne pas

Considérons ce qui pourrait se passer si l'adresse IP distante était 10.200.3.1 plutôt que 10.124.12.5.

1. Les systèmes distants peuvent ne pas savoir où envoyer cette adresse

   Ils savent tous que pour joindre le réseau IP 10.124.0.0, les trames doivent aller sur le câble connecté à eth0. Toutefois, il n'y a pas de réseau ayant l'adresse 10.200.0.0. Ils ne sauraient pas comment envoyer les trames au bon destinataire.

2. Le serveur risque ne pas savoir quelle carte utiliser pour l'adresse MAC appropriée lorsqu'il crée une entrée ARP.

C'est la raison la plus fréquente pour laquelle ARP proxy ne fonctionne pas chez certaines personnes. Elles ont un réseau IP différent associé à l'adresse IP du site distant plutôt qu'à l'une de leurs interfaces réseau.

8 Les problèmes avec ARP Proxy et qui doivent être évités

1. Ne pas avoir plus d'un système qui réponde à l'entrée ARP proxy d'une adresse IP particulière. Dans le cas de BSD, il faut vérifier qu'il n'y ait pas de conflit entre les adresses avec le proxy ARP. Pour un réseau basé sur un système de type BSD, vous devrez orienter le réseau tout entier sur un seul serveur.

Pour finir, les systèmes BSD n'aprécient gère de recevoir plus d'une réponse pour une requète ARP.

2. N'essayez pas de lancer ARP proxy pour une adresse déjà présente sur le réseau.

   C'est une petite variation du problème ci-dessus. Si vous essayez de lancer ARP proxy pour une adresse IP disponible sur le réseau, alors deux réponses vont être générées. Cela signifie que vous ne devriez pas prendre des adresses IP du réseau et les envoyer dans une connexion distante. Cela peut avoir pour concéquence que votre serveur réalise une opération ARP Proxy.

9 Que faire si vous ne pouvez utiliser ARP Proxy tout en voulant avoir les mêmes fonctionnalités ?

Il y a plusieurs choix possibles si vous êtes capables d'utiliser ARP proxy.

La méthode la plus simple est de créer un sous-réseau d'adresses IP pour que toutes les adresses externes aient leur propre adresse IP réseau. Puis, d'ajouter un chemin réseau dans chacun des routeurs (les périphériques indiqués par l'adresse passerelle dans chacun de vos fichiers *hosts*) pour que le réseau IP soit reconnu par le serveur à partir duquel les adresses IP se connectent.

Autrement, vous pouvez également utiliser des passerelles entre le serveur et les routeurs.

Une autre manière de faire est de mettre un chemin de machine si vous ne désirez pas employer un sous-réseau pour le réseau IP. Vous pouvez mettre les entrées dans chacun des routeurs pour toutes les adresses IP.

Vous ne devrez mettre à jour que les passerelles et les routeurs. Vous n'avez pas besoin de modifier toutes les machines de votre réseau. Le chemin par défaut que les machines utilisent pour envoyer des trames aux routeurs va provoquer ce qui s'appelle un *ICMP redirect* (une redirection ICMP) de trames vers la machine effectuant la requête. Cela va alors automatiquement ajouter un chemin vers le serveur approprié.

10 Conclusion

J'espère avoir un peu eclairci ARP proxy et son fonctionnement. Heureusement, si vous utilisez pppd ou dip-uri, vous n'avez pas besoin de connaître les différentes étapes du mécanisme. C'est réalisé automatiquement pour vous par ces programmes.

ARP Proxy n'est pas destiné à n'importe qui. C'est une solution qui fonctionne dans certains cas. Pouvoir déterminer ses besoins peux vous aider à résoudre vos problèmes de réseaux.

Des informations supplémentaires peuvent être trouvée dans le livre *TCP/IP Illustrated, volume 1 "The protocols"* par `W. Richard Stevens` et publié par Addison Wessley.

Merci !