Linux IPX-HOWTO

Kevin Thorpe, `kevin@pricetrak.com`
Version Française par Grégory Lajon `glajon@mail.dotcom.fr`

v2.3, 06 Mai 1998

Ce document tente de décrire comment obtenir, installer et configurer différents outils existant pour le système d'exploitation Linux utilisant le support du protocole IPX du noyau.

Contents

1 Introduction

Ceci est l'IPX-HOWTO. Il est conseillé de lire le NET-3-HOWTO de Linux en parallèle avec ce document.

1.1 Changements par rapport à la version précédente

```
Changement d'auteur:
Un grand merci à Terry Dawson pour avoir transmis ce document et félicitations pour être
devenu papa.
Ajouts :
Ajout d'une brève explication sur IPX en réponse à beaucoup de questions déroutantes sur
la liste de diffusion.
Corrections, mises à jour :
La nouvelle version de ncpfs supporte maintenant les logins NDS. C'est en beta test et
peut être interdit dans votre pays à cause de l'utilisation de technologies brevetées.
Ajout du support des droits d'accès dans mars_nwe.  Toujours en beta test.
```

1.2 Introduction

Comparé aux autres systèmes d'exploitation de la famille Un*x, le noyau de Linux possède une toute nouvelle implémentation de la partie réseau. Cette nouvelle approche de développement des logiciels réseau du noyau ont permis à Linux de pouvoir supporter des protocoles autres que TCP/IP. Le protocole IPX fait partie de ceux-là.

Le noyau de Linux supporte le protocole IPX seulement. Il ne prend pas encore en compte les protocoles tels que IPX/RIP, SAP ou NCP. Ces derniers sont gérés par d'autres logiciels comme nous le verrons plus loin dans ce document.

La gestion du protocole IPX a été initialement développée par Alan Cox <`alan@lxorguk.ukuu.org.uk`>, puis a été amélioré par Greg Page <`greg@caldera.com`>.

2 Responsabilité vis à vis de ce document

Je ne sais et ne peux connaître tout ce qu'il y a à savoir sur la partie logiciel réseau de Linux. Je vous prie donc d'accepter le fait qu'il puisse y avoir des erreurs dans ce document et d'être vigilant. Je vous conseille de lire tout fichier README inclus dans chacun des logiciels dont il sera question dans ce document pour de plus amples informations. Je tenterai de mon côté de conserver ce document à jour et sans erreurs autant que possible. Les versions des logiciels sont les versions disponibles au moment où ce document a été écrit.

En aucun cas les auteurs des logiciels cités dans ce document ou moi-même n'offrons de protection contre ce que vous faites. Si vous utilisez ce logiciel, même de la façon décrite dans ce document, et que cela cause des problèmes sur votre réseau, alors vous seul en endosserez la responsabilité. J'ai inclus cet avertissement parce que la mise en oeuvre et la configuration d'un réseau IPX n'est pas toujours aisée et que cela peut engendrer d'indésirables interactions avec les autres routeurs ou serveurs de fichiers si vous ne faites pas attention. J'ai également inclus cet avertissement parce qu'une personne suffisamment malchanceuse pour avoir appris cette leçon de la manière douloureuse me l'a demandé.

NdT : Le traducteur, qui n'est pas un spécialiste des réseaux Netware(tm), émet les mêmes réserves que l'auteur.

3 Documentation sur le même sujet

Ce document présume que vous savez compiler un noyau Linux avec les options réseau appropriées et que vous savez vous servir des outils de configuration de base des réseaux comme `ifconfig` et `route`. Si ce n'est pas le cas, je vous conseille de lire le

NET-3-HOWTO `NET-3-HOWTO.html`

lorsque le présent document y fait référence.

Les autres documents HOWTO de Linux qui pourraient vous être utiles sont :

L' *Ethernet-HOWTO* `Ethernet-HOWTO.html` qui décrit en détail la manière de configurer un port Ethernet pour Linux.

Le *PPP-HOWTO* `PPP-HOWTO.html` puisque la gestion d'IPX est implémentée à partir de la version 2.2.0d de PPP pour Linux.

3.1 Nouvelles versions de ce document

Si votre copie de ce document date de plus de deux mois, je vous recommande fortement d'obtenir une version plus récente. Les supports réseau pour Linux changent très rapidement avec de nouvelles fonctions et améliorations, en conséquence de quoi ce document change lui aussi fréquemment. La dernière version de ce document est disponible en version anglaise par le biais d'un ftp anonyme sur :

<`ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/IPX-HOWTO`> ou : <`ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/`

ou

sur le serveur du *Linux Documentation Project* `http://metalab.unc.edu/LDP/linux.html` à la page :

IPX-HOWTO `http://metalab.unc.edu/LDP/HOWTO/IPX-HOWTO.html`

ou sinon auprès de l'auteur, <`kevin@pricetrak.com`>. Il peut aussi être posté dans les forums de discussion `comp.os.linux.networking`, `comp.os.linux.answers` et `news.answers` de temps à autre.

NdT : La version française initialement traduite par Jean-Christophe Arnu est disponible de la même manière :

lip6 `ftp://ftp.lip6.fr/pub/Linux/french/docs/HOWTO/IPX-HOWTO.gz`

3.2 Votre avis

Envoyez-moi tout commentaire, améliorations ou suggestions : <`kevin@pricetrak.com`>. Si j'obtiens des retours d'informations rapidement je pourrai améliorer ou corriger ce document dans les plus brefs délais. Si vous rencontrez un quelconque problème avec ce document, veuillez m'en faire part directement par courrier électronique car je peux rater des informations postées dans les forums de discussion.

3.3 Liste de diffusion

Il existe une liste dans laquelle on discute des différents logiciels IPX pour Linux décrits dans ce document. Vous pouvez vous y abonner en envoyant un courrier à <`listserv@sh.cvut.cz`> avec `add linware` dans le corps du message. Vous envoyez vos contributions par courrier à <`linware@sh.cvut.cz`>. Je consulte régulièrement cette liste.

Cette liste est archivée sur le site *www.kin.vslib.cz* `http://www.kin.vslib.cz/hypermail/linware/`

4 Quelques termes utilisés dans ce document

Vous rencontrerez souvent au cours de ce document les termes `client` et `serveur`. Ce sont des termes spécifiques dont la définition est bien connue mais j'ai un peu généralisé leur définition :

client

C'est la machine ou le programme qui initie une action ou une connexion dans le but de pouvoir utiliser des services ou des données.

serveur

C'est la machine ou le programme qui accepte des connexions d'une ou plusieurs machines distantes et qui leur fournit un service ou des données.

Ces définitions ne sont pas très complètes, mais elles permettent de distinguer les participants des protocoles de bout en bout tels que *PPP* et *SLIP* qui n'ont pas réellement de client et de serveur.

Vous lirez aussi les termes suivants :

Bindery

La *bindery* est une base de données spécialisée qui stocke sur un serveur de fichiers Novell des informations concernant la configuration du réseau. Les clients Netware peuvent interroger la *bindery* pour obtenir de l'information sur les services disponibles, sur le routage et sur les utilisateurs.

Type de Trame (Frame type)

décrit le protocole utilisé pour transporter les paquets de données IPX (et IP) à travers vos segments Ethernet. On en rencontre couramment quatre :

Ethernet_II

C'est une version raffinée du standard Ethernet original DIX. Novell s'est vu attribuer un identifiant de protocole (protocol id) formel. Cela signifie que IPX et IP peuvent coexister joyeusement dans un environnement Ethernet_II. Ce type de trame est généralement utilisé dans les environnements Novell et c'est un bon choix.

802.3

C'est un protocole IEEE définissant un mécanisme à détection de porteuse et accès multiple avec détection des collisions (Carrier Sense Multiple Access with Collision Detection (CSMA/CD)). Il est basé sur le standard Ethernet original DIX, avec une importante modification. Le champ de type (protocol id) a été transformé en champ de longueur à la place. C'est pour cette raison que l'on ne devrait pas utiliser IPX avec ce type de trame. IEEE 802.3 a été conçu pour transporter des trames IEEE 802.2 **seulement**, mais il existe des implémentations qui l'utilisent pour transporter des trames IPX directement et cela fonctionne quand même. Evitez-le à moins de vouloir vous connecter à un réseau déjà configuré pour l'utiliser.

802.2

C'est un protocole IEEE qui définit un ensemble de procédures de contrôle de la couche liaison logique. Il fournit une façon simple de faire coexister plusieurs protocoles mais reste limité dans ce sens. Novell utilise un SAP (Service Address Point, équivalent d'un identifiant de protocole) non officiel. Mais comme tout le monde fait pareil, cela n'a encore jamais posé de problèmes.

SNAP

SNAP est le protocole d'accès au sous-réseau (Sub Network Access Protocol). Ce protocole est implémenté au-dessus de 802.3 et 802.2. Il étend les capacités multiprotocole de 802.2 et présente une certaine compatibilité avec les trames Ethernet et Ethernet_II existantes.

IPX

Internet Packet eXchange est un protocole utilisé par la société Novell fournissant un support d'interconnexion pour leur produit NetWare(tm). IPX a une fonction similaire au protocole IP utilisé par la communauté TCP/IP.

Adresse IPX d'un réseau (IPX network address)

Il s'agit d'un nombre qui identifie de manière unique un réseau IPX particulier. La notation courante de cette adresse est exprimée en hexadécimal. On peut donner comme exemple : 0x23a91002

Réseau IPX interne (IPX Internal network)

C'est un réseau IPX virtuel. Il est virtuel car il ne correspond à aucun réseau physique. Il est utilisé pour fournir un moyen unique d'identifier et d'adresser une machine IPX. Cette adresse est codée de la même manière qu'une adresse IPX de réseau. En général, ce n'est utile que pour les machines IPX qui sont physiquement connectées à plusieurs réseaux IPX comme les serveurs de fichiers.

RIP

Routing Information Protocol est un protocole utilisé pour informer automatiquement les différents ordinateurs présents sur un réseau IPX des différentes routes qui lui sont applicables. Il est fonctionnellement identique au RIP utilisé au sein de la communauté TCP/IP.

NCP

Netware Core Protocol est un protocole de système de fichiers réseau créé par la société Novell pour leur produit NetWare(tm). On peut comparer les fonctionnalités de NCP à NFS utilisé par la communauté TCP/IP.

SAP

Service Advertisement Protocol est un protocole créé par la société Novell. Il est utilisé pour avertir des services fournis sur le réseau dans un environnement NetWare(tm).

Adresse Physique (hardware address)

C'est un nombre qui identifie de manière unique une machine sur un réseau physique au niveau de la couche media (couche MAC). L'adresse Ethernet en est un exemple. Une adresse Ethernet est généralement codée par six valeurs hexadécimales séparées par deux points : ex. 00:60:8C:C3:3C:0F.

route

La *route* est le chemin emprunté par les paquets de données au travers d'un réseau pour atteindre leur destination.

5 Une brève discussion sur la topologie des réseaux IPX

C'est une explication très simplifiée pour les personnes qui découvrent IPX. Les grands réseaux mettront probablement à mal beaucoup des règles expliquées ici. Dans les réseaux IPX complexes, l'administrateur devrait toujours être consulté.

Les réseaux IPX s'appuient sur un procédé de *réseaux* numérotés, à la différence de IP qui insiste plus sur les adresses d'*interface*. Un réseau est un ensemble d'équipements connectés à un même segment de réseau local (LAN) et *utilisant le même type de trame.* Des types de trame différents sur le même segment de réseau local sont traités comme des réseaux séparés.

Chaque réseau doit recevoir un numéro qui soit unique au travers de toute l'interconnexion de réseaux (internetwork). Ceci est habituellement réalisé par un serveur Netware(tm), mais peut facilement être réalisé par Linux. Les clients IPX recoivent ce numéro du serveur lors de leur démarrage, ils ont juste besoin de connaître le type de trame correct.

Le routage entre les réseaux est habituellement réalisé en mettant deux cartes réseaux dans un serveur. Ce serveur fait tourner le protocole RIP qui maintient une table de routage pour l'interconnexion de réseaux. Périodiquement, cette table de routage est échangée entre les serveurs. En peu de temps, chaque serveur 'découvre' la topologie de l'interconnexion de réseaux.

Si vous souhaitez uniquement utiliser les services d'un serveur NetWare existant, vous pouvez utiliser `ipx_configure` (section 7.1) pour définir automatiquement les interfaces en utilisant des requêtes de "broadcast" pour rechercher un serveur. Si cela échoue ou que vous voulez fournir des services IPX, vous devrez définir les interfaces manuellement en utilisant `ipx_interface` ou `mars_nwe`.

6 Les fichiers relatifs à IPX dans le répertoire `/proc`

Il existe un certain nombre de fichiers relatifs au support d'IPX par Linux dans le répertoire `/proc` :

/proc/net/ipx_interface

Ce fichier contient des informations au sujet des interfaces IPX configurées sur votre machine. Elles peuvent avoir été configurées manuellement par commande ou automatiquement détectées et configurées.

/proc/net/ipx_route

Ce fichier contient une liste de routes existantes dans la table de routage IPX. Ces routes peuvent avoir été ajoutées à la main par l'intermédiaire de commandes ou automatiquement par le démon de routage IPX.

/proc/net/ipx

Ce fichier contient la liste des sockets IPX qui sont ouverts sur la machine au moment de l'édition du fichier.

7 Les outils IPX de Greg Page

Greg Page <`greg@caldera.com`> de la société Caldera Incorporated a écrit un ensemble d'outils de configuration pour IPX et a amélioré le support IPX du noyau Linux.

Les améliorations du noyau permettent de configurer un système Linux comme pont ou routeur IPX parfaitement fonctionnel. Le support d'IPX amélioré a déjà été incorporé dans le corps du noyau de distribution. Vous en êtes donc certainement équipé.

Ces outils de configuration vous permettent de paramétrer vos périphériques réseau pour le support d'IPX et de mettre en place le routage pour ce protocole. D'autres fonctionnalités sont aussi incluses dans ce paquetage. Ces outils sont disponibles sur

metalab.unc.edu `ftp://metalab.unc.edu/pub/Linux/system/filesystems/ncpfs/ipx.tgz` .

NdT : les lecteurs connectés en France pourront utiliser un site miroir comme *ftp.lip6.fr* `ftp://ftp.lip6.fr/`

7.1 Les outils IPX en détails

ipx_interface

Cette commande est utilisée pour ajouter, enlever ou contrôler manuellement des propriétés d'IPX sur un périphérique réseau donné. Ce dernier est souvent un périphérique Ethernet comme `eth0`. Au moins une interface IPX doit être déclarée comme interface *primaire* par le paramètre `-p` suivi du nom de l'interface. Par exemple, l'attribution des capacités IPX au périphérique eth0, en tant que première interface utilisant le type de trame IEEE 802.2 et l'adresse réseau `39ab0222` se fera ainsi :

```
# ipx_interface add -p eth0 802.2 0x39ab0222
```

Si le type de trame diffère des serveurs NetWare(tm) sur ce réseau, ils vous ignoreront délibérement. Si le type de trame est correct, mais que le numéro de réseau diffère, ils vous ignoreront encore, mais afficheront des messages sur la console du serveur Netware(tm). Ce dernier cas vous promet d'avoir des problèmes avec votre administrateur NetWare et peut perturber les clients NetWare en cours d'utilisation.

Si vous obtenez une erreur en faisant tourner ce programme et que vous n'avez pas encore configuré TCP/IP, alors vous allez devoir démarrer manuellement l'interface `eth0` en utilisant la commande :

```
# ifconfig eth0 up
```

ipx_configure

Cette commande permet ou interdit la configuration automatique de l'interface et la mise en oeuvre de l'interface primaire.

`-auto_interface`

vous permet de décider si de nouveaux périphériques réseau doivent être automatiquement configurés comme des périphériques IPX ou pas.

`-auto_primary`

vous accorde le droit de décider si les logiciels IPX doivent sélectionner une interface primaire ou non. Des problèmes ont été reportés sur l'utilisation de ceci avec des clients Windows 95 sur le réseau.

Par exemple si l'on désire une configuration automatique des interfaces et une mise en oeuvre automatique de l'interface primaire on utilisera la ligne suivante

```
# ipx_configure --auto_interface=on --auto_primary=on
```

ipx_internal_net

Cette commande permet de mettre en place, de configurer ou retirer une adresse de réseau interne. Cette adresse de réseau interne est optionnelle, mais lorsqu'elle est configurée elle est toujours considérée comme interface primaire. Pour configurer une adresse réseau IPX ayant le numéro `ab000000` sur le noeud IPX `1` vous devez utiliser:

```
# ipx_internal_net add 0xab000000 1
```

ipx_route

Cette commande permet de modifier manuellement la table de routage IPX. On peut prendre comme exemple l'ajout d'une route vers le réseau IPX `39ab0222` via un routeur ayant le numéro de noeud `00608CC33C0F` sur le réseau IPX 39ab0108 :

```
# ipx_route add 0x39ab0222 0x39ab0108 0x00608CC33C0F
```

8 Configurer votre machine Linux comme routeur IPX

Si vous avez un certain nombre de segments IPX que vous souhaitez interconnecter, vous aurez besoin d'un routeur. Dans l'environnement Novell, la propagation d'au moins deux types d'informations est absolument primordiale sur le réseau : d'un côté, les informations de routage sur le réseau qui sont propagées en utilisant le RIP Novell, de l'autre côté, les informations d'avertissement du système qui sont véhiculées par le SAP Novell. Tout routeur doit pouvoir traiter et comprendre ces deux types de protocoles pour être utilisé dans toutes les situations.

Linux possède un support pour ces deux protocoles et peut donc être aisément configuré pour réaliser un routeur IPX complètement compatible Novell.

Le support IPX du noyau de Linux gère réellement le routage des paquets entre les interfaces, mais il le fait selon les règles données dans une table de routage. Linux nécessite un programme implémentant les protocoles RIP et SAP de Novell afin que la table de routage soit correctement construite et maintenue périodiquement, pour qu'elle reflète bien les changements d'état du réseau.

Volker Lendecke <`lendecke@namu01.gwdg.de`> a développé un démon de routage *ipxrid* qui fera tout ça pour vous. Le paquetage *mars_nwe* mentionné plus loin inclut un autre démon de routage.

Vous pouvez trouver *ipxrid* sur :

metalab.unc.edu `ftp://metalab.unc.edu/pub/Linux/system/filesystems/ncpfs/ipxripd-0.7.tgz`

ou directement chez Volker :

ftp.gwdg.de `ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/ipxripd-0.7.tgz`

Configurer une machine Linux en routeur IPX est une opération très linéaire. Procédez comme suit :

1. Construisez le noyau avec les supports d'IPX, d'Ethernet et le système de fichiers `/proc`
2. Procurez-vous, compilez et installez le démon *ipxd*.
3. Redémarrez la machine avec son nouveau noyau et assurez-vous que chacune des cartes Ethernet ont été correctement détectées et qu'il n'y a pas de conflits matériels.
4. Mettez en place le protocole IPX sur chacune des interfaces en utilisant la commande `ipx_interface` décrite ci-dessus.
5. Lancez le démon *ipxd*.

En considérant le simple réseau suivant :

```
 IPX Addr: 0x01000000  802.2
|--------------------------|
              |
              \_________________________
                                        \         Routeur Linux
 IPX Addr: 0x02000000  802.2             \
|--------------------------|              \   eth0/-----------\
              |                            \--====|           |
              \_________________________          | IPX route |
                                        \     eth1|  Table    |
 IPX Addr: 0x03000000  etherII           \----====|    ^      |
|--------------------------|                      |    |      |
              |                               eth2|  IPXd     |
```

```
                \______________________________/====|           |
                                                    |  SAPd     |
   IPX Addr: 0x04000000  etherII                eth3|           |
  |--------------------------|                 /====|           |
                |                              |    \___________/
                \______________________________/
```

La configuration pour le réseau ci-dessus ressemblera à :

```
# ipx_interface add eth0 802.2 0x0100000000
# ipx_interface add eth1 802.2 0x0200000000
# ipx_interface add eth2 etherii 0x0300000000
# ipx_interface add eth3 etherii 0x0400000000
# ipxd
```

Vous devrez attendre un petit moment pour vérifier le fichier `/proc/net/ipx_route` (`cat /proc/net/ipx_route`). Vous devriez retrouver les routes IPX inhérentes à votre configuration du système et toutes les configurations apprises par le contact avec les autres routeurs sur le réseau le cas échéant.

8.1 Dois-je utiliser un réseau interne ?

Novell a une fonctionnalité appelée réseau interne, qui est utilisée pour simplifier le routage lorsqu'une machine est connectée à plus d'un périphérique réseau. C'est utile dans le cas d'un serveur de fichiers connecté à plusieurs réseaux, en ce sens qu'une seule route doit être utilisée pour atteindre le serveur, quel que soit le réseau à partir duquel on opère.

Dans le cas d'une configuration où vous ne faîtes pas tourner un serveur de fichiers mais seulement un routeur IPX, il n'est pas simple de répondre. Des témoignages rapportent que l'utilisation de IPX/PPP marche "mieux" si vous utilisez également un réseau interne.

Dans tous les cas, c'est facile à faire, mais cela peut nécessiter de recompiler le noyau. Lorsque vous en êtes au `make config` du noyau, vous devez répondre Oui (`y`) à la question `Full internal IPX network` comme suit :

```
...
...
Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] y
...
...
```

Pour configurer le réseau interne, utilisez la commande *ipx_internal_net* déjà décrite dans la section outils IPX. La principale précaution à prendre est de s'assurer que l'adresse IPX réseau que vous choisissez est unique sur votre réseau et qu'aucune autre machine ou réseau ne l'utilise.

9 Configurer votre machine Linux en client NCP

Si vous êtes un utilisateur d'un réseau hétérogène où les protocoles IP et IPX sont utilisés, il est probable que vous ayez à un moment ou à un autre, besoin d'accéder à partir de votre machine Linux, à des données stockées sur un serveur de fichier Novell de ce réseau. Novell a longtemps proposé un ensemble logiciel NFS pour leurs serveurs de fichiers qui permettaient cet accès. Cependant si le site auquel vous êtes connecté est petit, le nombre de personnes nécessitant cet type d'accès ne justifie peut-être pas l'achat de ces outils logiciels commerciaux.

Volker Lendecke <`lendecke@namu01.gwdg.de`> a écrit un module du système de fichiers du noyau Linux supportant un sous-ensemble limité du protocole Novell NCP qui permet de monter une partition Novell dans votre système de fichiers Linux sans avoir recours à quelque produit additionnel sur votre serveur de fichiers.

Volker a appelé cet ensemble *ncpfs*. Ce dernier a été réalisé à partir des informations tirées principalement du livre "Netzwerkprogrammierung in C" de Manfred Hill et Ralf Zessin (de plus amples informations sont données dans le fichier README contenu dans l'ensemble *ncpfs*).

Ce logiciel permet à Linux d'émuler une station de travail Novell normale pour ce qui concerne la partie fichiers. Un petit utilitaire d'impression est également inclus pour vous permettre d'imprimer dans les queues d'impression Novell (ceci est documenté plus loin dans la section client d'impression). Le paquetage *ncpfs* fonctionne avec les serveurs de fichier Novell version 3.x et supérieurs, il ne fonctionne pas avec les serveurs de fichier Novell version 2.x . Le client *ncpfs* ne fonctionne qu'avec les produits réellement compatibles Novell, mais malheureusement, certains produits qui se disent compatibles ne le sont pas suffisamment. Pour utiliser *ncpfs* avec Novell 4.x, il est préférable d'utiliser le serveur Novell en mode d'emulation de *bindery*. Le support de la NDS est un ajout très récent de *ncpfs* et est toujours en version bêta. De plus son utilisation peut être interdite dans votre pays en raison de l'inclusion de technologie soumise à brevet.

9.1 Comment récupérer *ncpfs*

L'ensemble *ncpfs* a été développé afin d'être compilé avec les noyaux de Linux version `1.2.13` ou plus récents que `1.3.71` (ce qui inclut `2.x.x`). Si vous n'utilisez pas un de ces noyaux vous devrez vous en procurer et l'utiliser. Le *Kernel-HOWTO* `kernel-HOWTO.html` décrit comment faire cela en détail.

Vous pouvez vous procurer l'ensemble *ncpfs* par ftp anonyme sur le site de Volker, *ftp.gwdg.de* `ftp://ftp.gwdg.de/pub/ncpfs/` , sur *metalab.unc.edu* `ftp://metalab.unc.edu/pub/Linux/system/Filesystems/ncpfs`

ou sur un site miroir. La version à jour au moment où j'écris ces lignes est `ncpfs-2.0.11.tgz` ou `ncpfs-2.2.0.tgz` pour le support de la NDS.

9.2 Compiler *ncpfs* pour les noyaux 1.2.13

Compiler un noyau Linux avec les supports d'Ethernet et IPX

La première chose que vous ayez à faire est de vérifier que votre noyau a été compilé avec l'option IPX. Dans la version `1.2.13` du noyau, vous vous assurerez que la réponse à la question '`The IPX Protocol`' est `Y` (yes) comme illustré :

```
 ...
 ...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
 ...
 ...
```

Vous devez aussi vérifier que vous avez inclus le gestionnaire approprié pour votre carte Ethernet. Si vous ne savez pas comment procéder, je vous conseille de lire l'Ethernet-HOWTO

Ethernet-HOWTO `Ethernet-HOWTO.html` .

Vous pourrez alors procéder à la compilation. Je vous rappelle que le lancement de *lilo* pour installer le nouveau noyau est obligatoire une fois la compilation terminée.

Décompresser les programmes *ncpfs*

```
# cd /usr/src
# tar xvfz ncpfs-2.0.11.tgz
# cd ncpfs
```

Vérifier le Makefile

Si vous avez l'intention d'utiliser *kerneld* pour charger automatiquement le module noyau *ncpfs*, alors vous devez décommenter la ligne du `Makefile` qui fait référence à `KERNELD`. Si vous n'êtes pas certain de comprendre tout ce que cela signifie, vous devriez lire le

Kernel-HOWTO `Kernel-HOWTO.html` pour vous familiariser avec la configuration des modules noyau.

Compiler les outils *ncpfs*

Le logiciel doit se compiler proprement sans qu'une configuration ne soit nécessaire :

```
# make
```

Copier les outils IPX à un endroit utile de votre arborescence

Après cette compilation, vous trouverez tous les outils dont vous aurez besoin dans le répertoire `ncpfs/bin`. Vous pouvez utiliser

```
# make install
```

pour installer les outils dans le choix de répertoires de Volker. Si votre système est basé sur ELF, alors vous devrez taper `ldconfig -v` pour vous assurer que la bibliothèque partagée est accessible.

Copier le module *npfs.o* à un endroit approprié si nécessaire.

Après la compilation d'un noyau `1.2.x`, vous trouverez aussi dans le répertoire `ncpfs/bin` le fichier `ncpfs.o` Il s'agit du module *ncpfs* du noyau. Je vous conseille de copier ce fichier à un endroit utile. Par exemple sur ma distribution *Debian* de Linux, je l'ai copié dans le répertoire `/lib/modules/1.2.13/fs` et j'ai ajouté ncpfs dans le fichier `/etc/modules` de manière à ce qu'il soit automatiquement chargé lors du démarrage de Linux. Si vous utilisez d'autres distributions, vous devrez trouver où les modules sont généralement placés et copier le fichier `ncpfs.o` à cet endroit, ou sinon, copiez-le dans le répertoire `/etc`. Utilisez la commande suivante afin de charger le module en mémoire manuellement :

```
# insmod ncpfs.o
```

9.3 Compiler *ncpfs* pour des noyaux 1.3.71++/2.0.*

Pour la dernière version de *ncpfs*, vous devez utiliser un noyau 1.3.71 ou supérieur, ce qui inclut les noyaux 2.0.* .

Le code noyau de *ncpfs* a été inclus dans la distribution noyau standard, ainsi vous n'aurez qu'à répondre oui (`Y`) à la question suivante lors de la configuration de la compilation :

```
Networking options  --->
    ...
    ...
```

```
        <*> The IPX protocol
        ...
    Filesystems  --->
        ...
        ...
        <*> NCP filesystem support (to mount NetWare volumes)
        ...
```

Vous devez suivre les instructions pour la compilation des noyaux `1.2.*` afin de créer les outils IPX, mais il n'y aura pas de module à installer.

9.4 Configurer et utiliser *ncpfs*

Configurer les logiciels de réseau IPX

Deux moyens sont possibles pour configurer ceux-ci. Vous pouvez tout faire manuellement ou vous pouvez laisser le logiciel déterminer lui-même ses paramètres en utilisant

```
# ipx_configure --auto_interface=on --auto_primary=on
```

Cela devrait fonctionner dans la plupart des situations, mais si ce n'est pas votre cas, lisez la section "les outils IPX" située plus loin dans ce document pour configurer manuellement les logiciels. Des problèmes ont été rapportés concernant cette utilisation au sein de réseaux contenant des clients Windows95.

Tester la configuration

Maintenant que votre réseau IPX est configuré, vous devriez être capable d'exécuter la commande *slist* qui vous renverra la liste de tous les serveurs de fichiers Novell de votre réseau :

```
# slist
```

Si la commande slist affiche un message du type : `ncp_connect:  Invalid argument`, cela signifie que votre noyau ne supporte pas IPX. Vérifiez que vous avez démarré votre ordinateur avec le noyau approprié. Au démarrage du système, vous devriez voir des messages concernant "IPX" et "ncpfs". Si la commande *slist* ne renvoie pas la liste de tous les serveurs de fichiers, vous devrez utiliser la méthode de configuration manuelle.

Monter un volume ou un serveur Novell(tm)

Si votre logiciel réseau IPX fonctionne correctement, vous devriez pouvoir monter un volume du serveur de fichiers dans votre système de fichiers Linux. La commande *ncpmount* est utilisée à ces fins et nécessite que vous spécifiez au moins les informations suivantes :

1. le nom du serveur de fichiers ;
2. (éventuellement) le répertoire du serveur de fichier à monter ;
3. l'identifiant de login du serveur de fichiers. Dans le cas où il a un mot de passe, vous devrez aussi le donner à cette commande ;
4. le point de montage, c'est-à-dire l'endroit où vous voulez que le système de fichiers du serveur Novell soit monté dans votre système de fichiers. Ce doit être un répertoire déjà existant.

De la même manière, il existe une commande *ncpumount* pour démonter un système de fichiers NCP monté. Le système de fichiers NCP sera démonté proprement si vous éteignez votre machine normalement. Vous n'avez donc pas à vous tracasser pour *ncpumount*er manuellement vos systèmes de fichiers avant un halt ou un shutdown.

Par exemple si l'on veut monter le serveur de fichiers `ACCT_FS01` sous l'identifiant `guest` qui ne possède pas de mot de passe, dans le répertoire `/mnt/Accounts`, la commande devrait ressembler à :

```
# ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -n
```

Notez que l'utilisation de l'option `-n` indique qu'aucun mot de passe n'est nécessaire pour se loguer. Le même login avec le mot de passe `secret` ressemblera à :

```
# ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -P secret
```

Si vous ne spécifiez ni l'option `-n` ni l'option `-P`, le système vous demandera un mot de passe.

Vérifier le montage

Si le montage est réussi, vous pourrez accéder à tous les volumes auxquels l'utilisateur que vous avez spécifié a accès. Cet accès se fait de manière transparente comme si le volume se trouvait directement dans le répertoire spécifié lors du montage. Vous devez être capable de naviguer au travers des répertoires de ce volume et d'y trouver des fichiers. Vous pouvez également utiliser l'option `-V` pour monter un seul volume.

NCP ne fournit pas d'identificateur d'utilisateur (uid) ni d'identificateur de groupe (gid) pour les fichiers. Tous les fichiers auront les permissions données au répertoire de montage, restreints par les droits de permissions sur le serveur Novell. Gardez ce point à l'esprit lorsque vous partagerez des montages avec des utilisateurs Linux.

Configurer le montage automatique des volumes ncp

Si vous avez besoin de monter de manière permanente vos partitions ncp, vous aurez à mettre les commandes de configuration ci-dessus dans vos fichiers *rc* de manière à ce que le montage soit opéré à chaque démarrage de Linux. Si votre distribution ne fournit pas déjà un moyen de configurer IPX comme le propose Debian, alors je vous recommande de les placer dans le fichier `/etc/rc.local` si vous en avez un. Vous devrez utiliser un script du type suivant :

```
#
# Démarrer le systéme de fichiers NCP
/sbin/insmod /lib/modules/1.2.13/fs/ncpfs.o

# Configurer le reseau IPX
ipx_configure --auto_interface=on --auto_primary=on

# Accéder au volume réservé à l'utilisateur guest sur le serveur de fichiers
ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -n
```

Il existe un autre moyen de configurer les montages NCP. Cela se fait au moyen du fichier `$HOME/.nwclient`. Ce fichier contient les informations spécifiques à l'utilisateur qui seront appliquées sur des montages NCP réguliers. Vous pourrez ainsi les monter sans avoir à spécifier tous les paramètres à chaque fois.

Le format de ce fichier est assez simple :

```
# Le premier paramètre est le 'serveur préféré'
# utilisé même si vous ne spécifiez pas de serveur lors du
# lancement du script.
#
# L'utilisateur TERRY se loguant sur le serveur de fichiers DOCS_FS01
# avec le mot de passe 'password'
DOCS_FS01/TERRY password
#
```

```
# Login Guest sur le serveur de fichiers ACCT_FS01 sans mot de passe.
ACCT_FS01/GUEST -
```

Vous pourrez, par la suite, utiliser :

```
$ ncpmount /home/terry/docs
```

afin de monter la partition `DOCS_FS1` avec le login `TERRY` sous le répertoire `/home/terry/docs`. Vous remarquerez que cette commande a été choisie car aucun serveur de fichiers n'a été spécifié dans la commande de montage. Si la commande suivante avait été utilisée :

```
$ ncpmount -S ACCT_FS01 /home/terry/docs
```

vous seriez entré sur le serveur de fichiers `ACCT_FS01` avec le compte `GUEST`.

Note : Pour que ce mécanisme fonctionne, il faut que la permission de `$HOME/.nwclient` soit de `0600`, vous devrez utiliser la commande suivante :

```
$ chmod 0600 $HOME/.nwclient
```

Si des utilisateurs non-root ont la permission d'utiliser ce mécanisme, la commande *ncpmount* doit utiliser un Set Userid Root, vous aurez donc à donner ces permissions :

```
# chmod 4755 ncpmount
```

Tester l'utilitaire nsend

nsend est un utilitaire pour envoyer des messages à des utilisateurs Novell qui est également inclus dans le paquetage. Il s'utilise de la manière suivante :

```
# nsend pierre coucou ici
```

qui va envoyer le message "coucou ici" à un utilisateur "pierre" déjà logué sur votre serveur de fichiers "primaire" (le premier qui apparaît dans votre fichier `.nwclient`). Vous pouvez spécifier un autre serveur de fichiers avec la même syntaxe que la commande *ncpumount*.

10 Configurer votre machine Linux en serveur NCP

Il existe deux paquetages permettant à Linux de fournir les fonctions d'un serveur de fichiers Novell. Tous deux permettent de partager des fichiers de la machine Linux entre les utilisateurs utilisant un client Novell Netware. Les utilisateurs peuvent attacher et utiliser des systèmes de fichiers Novell comme des volumes locaux à leurs machines juste comme s'ils provenaient d'un serveur de fichiers Novell. Je vous propose d'essayer les deux paquetages afin de voir celui qui correspond le mieux à vos besoins.

10.1 L'ensemble *mars_nwe*

Martin Stover <`mstover@freeway.de`> a développé *mars_nwe* pour permettre à Linux de fournir des services fichiers et impression pour des clients NetWare.

mars_nwe signifie "Martin Stover's Netware Emulator".

10.1.1 Possibilités offertes par *mars_nwe*

mars_nwe fournit un sous-ensemble de la spécification Novell NCP pour les services fichiers, pour les connexions aux disques et aussi pour les services d'impressions. Il est probable qu'il contient des bogues mais beaucoup de personnes l'utilisent et le nombre de bogues diminue avec les nouvelles versions.

10.1.2 Obtenir *mars_nwe*

Vous pouvez récupérer *mars_nwe* sur le serveur

ftp.gwdg.de `ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/` . ou sur

metalab.unc.edu `ftp://metalab.unc.edu/pub/linux/filesystems/ncpfs/` .
La version à jour lors de l'écriture de ce document est : `mars_nwe-0.99.pl10.tgz`.

10.1.3 Compiler l'ensemble *mars_nwe*

Compiler un noyau Linux supportant Ethernet et IPX.

Dans la version `1.2.13` du noyau, vous devrez juste vérifier que vous avez répondu oui (`Y`) à '`The IPX Protocol`' et non (`N`) à '`Full internal IPX network`' comme illustré :

```
 ...
 ...
The IPX protocol (CONFIG_IPX) [n] y
 ...
 ...
Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] n
 ...
 ...
```

Pour les noyaux plus récents, un procédé similaire est utilisé lors de la construction du noyau. La question posée peut légèrement changer.

Assurez-vous aussi d'un choix approprié pour votre carte Ethernet. Si vous ne savez pas comment procéder, vous devriez lire l'Ethernet-HOWTO,

Ethernet-HOWTO `Ethernet-HOWTO.html` .

Vous pouvez maintenant procéder à la compilation de votre noyau. Une fois cette étape finie, n'oubliez pas de lancer *lilo* afin d'installer le nouveau noyau.

Décompresser le paquetage *mars_nwe*.

```
# cd /usr/src
# tar xvfz mars_nwe-0.99.pl10.tgz
```

Compiler *mars_nwe*.

Cette opération est très simple. La première étape consiste simplement à lancer `make`, cela va vous créer un fichier `config.h`. Ensuite, vous devez regarder et modifier le fichier `config.h` si nécessaire. Cela vous permet de configurer des choses comme le répertoire d'installation ou le nombre maximum de sessions et de volumes que le serveur pourra supporter. Les lignes importantes à regarder sont :

`FILENAME_NW_INI`	l'emplacement des fichiers d'initialisation
`PATHNAME_PROGS`	l'emplacement du programme exécutable
`PATHNAME_BINDERY`	l'endroit où iront les fichiers de la 'bindery'
`PATHNAME_PIDFILES`	le répertoire où les fichiers PID seront écrits
`MAX_CONNECTIONS`	le nombre maxi de connexions simultanées autorisées
`MAX_NW_VOLS`	le nombre maxi de volumes supportés par mars_nwe
`MAX_FILE_HANDLES_CONN`	le nombre maxi de fichiers ouverts par connexion
`WITH_NAME_SPACE_CALLS`	si vous voulez supporter les clients ncpfs
`INTERNAL_RIP_SAP`	si vous voulez que mars_nwe fasse le routage rip/sap
`SHADOW_PWD`	selon que vous utilisez les 'shadow passwords' ou non

Les options par défaut devraient vous convenir mais vérifiez quand même.

Une fois ceci terminé tapez :

```
# make
# make install
```

Ces commandes compileront les différents serveurs et les installeront dans un répertoire approprié. Le script d'installation installe aussi le fichier de configuration `/etc/nwserv.conf`.

Configuration du serveur.

Celle-ci est simple. Il faut éditer le fichier`/etc/nwserv.conf`. Le format de ce fichier peut vous sembler assez énigmatique au premier abord, mais il est en fait bel et bien ordonné. Chaque ligne de ce fichier correspond à un point de configuration. Chacune de ces lignes est délimitée par le caractère espace et débute par un nombre qui indique le contenu de la ligne. Tous les caractères précédés du caractère '#' sont considérés comme étant un commentaire et de ce fait sont ignorés. Martin fournit un fichier d'exemple de configuration dans son paquetage, cependant, je vous donne ce que je considère comme étant une version simplifiée de ce fichier, et ce, afin que vous puissiez mieux le comprendre.

```
# VOLUMES (max. 5)
# Seul le volume SYS est obligatoire. Le répertoire contenant le volume
# SYS doit contenir les répertoires : LOGIN, PUBLIC, SYSTEM, MAIL.
# l'option 'i' ignore la casse (majuscules/minuscules)
# l'option 'k' convertit toutes les requêtes de nom de fichier en
# minuscules
# l'option 'm' indique que le volume est amovible (cdrom par ex.)
# l'option 'r' positionne le volume en lecture seule
# l'option 'o' indique le volume est un système de fichiers monté unique
# l'option 'P' permet aux commandes d'être utilisées comme des fichiers
# l'option 'O' permet l'utilisation de l'espace de noms (namespace) OS/2
# l'option 'N' permet l'utilisation de l'espace de noms (namespace) NFS
#
# Par défaut les noms de fichiers sont en majuscules.
# Syntaxe :
#    1 <Nom du volume> <Chemin du volume>   <Options>

1   SYS         /home/netware/SYS/                # SYS
1   DATA        /home/netware/DATA/   k           # DATA
1   CDROM       /cdrom                kmr         # CDROM

# NOM DU SERVEUR
# Si cette option n'est pas mise le nom du serveur Linux sera converti
# en majuscule et utilisé
# Syntaxe :
#    2 <Nom du serveur>

2   LINUX_FS01

# ADRESSE DE RESEAU INTERNE
# l'adresse de réseau interne est une fonctionnalité qui simplifie
# le routage IPX pour les machines reliées à plusieurs réseaux IPX.
# Syntaxe :
#    3 <adresse interne du réseau> [<Numéro de noeud>]
# ou
#    3 auto
```

```
# si vous utilisez 'auto' alors l'adresse IP de la machine sera
# utilisée. NOTE : cela peut être dangereux, assurez-vous de prendre
# un nombre unique sur votre réseau. Les adresses sont codées en
# hexadécimal sur 4 octets (le préfixe 0x est obligatoire).
3 0x49a01010 1

# PERIPHERIQUES RESEAU
# Cette option configure votre réseau IPX. Si votre réseau IPX est déjà
# configuré, vous n'en aurez pas besoin.
# Syntaxe
#    4 <Numéro de reseau IPX> <Nom de périphérique> <Type de trame> [<ticks>]
#                          Types de trames : ethernet_ii, 802.2, 802.3, SNAP

4  0x39a01010  eth0  802.3  1

# SAUVEGARDE DES ROUTES IPX APRES EXTINCTION DU SERVEUR.
# Syntaxe :
#    5 <drapeau>
#       0 = ne pas sauver les routes, 1 = sauver les routes.

5 0

# VERSION DE NETWARE
# Syntaxe :
#    6 <version>
#      0 = 2.15, 1 = 3.11

6 1

# GESTION DES MOTS DE PASSE
# les clients DOS de Novell supportent une fonctionnalité qui chiffre
# votre mot de passe (mdp) lorsque vous le changez. Vous pouvez
# configurer votre serveur mars pour supporter cette fonctionnalité.
# Syntaxe :
#    7 <drapeau>
# avec <drapeau> =
#    0 force chiffrage du mdp (les clients ne peuvent pas changer de mdp).
#    1 force chiffrage du mdp, permet les changements de mdp non chiffrés.
#    7 permet les mdp non chiffrés mais pas les mdp nuls.
#    8 permet les mdp non chiffrés ainsi que les mdp nuls.
#    9 seulement les mdp non chiffrés (ne marche pas avec OS/2).

7 1

# DROITS MINIMUM EN FONCTION DES GID ET UID
# Les permissions qui sont appliquées lors de connexions sans login
# Ces permissions seront utilisées pour les fichiers
# de votre serveur primaire.
# Syntaxe :
#    10 <gid>
#    11 <uid>
#    <gid> <uid> sont ceux donnés dans /etc/passwd et /etc/group

10  200
11  201
```

```
# MOT DE PASSE DU SUPERVISEUR
# Doit etre enlevé après le premier démarrage du serveur. Le serveur
# cryptera cette information dans le fichier de bindery après son
# lancement. Vous devriez éviter d'utiliser le compte 'root' et en choisir
# un autre pour administrer le serveur mars.
#
# Cette entrée est lue et chiffrée dans le fichier de bindery du serveur,
# elle n'a donc besoin d'être présente que la première fois que vous
# démarrez le serveur, pour s'assurer que le mot de passe ne soit pas volé
# Syntaxe :
#    12 <Login du superviseur> <Nom d'utilisateur UNIX> [<mot de passe>]

12  SUPERVISOR  terry  secret

# COMPTES DES UTILISATEURS
# Permet d'associer les comptes NETWARE aux comptes UNIX. Les mots de
# passe sont en option.
# Syntaxe :
#    13 <Login utilisateur> <Nom d'utilisateur UNIX> [<mot de passe>]

13  MARTIN martin
13  TERRY  terry

# ADMINISTRATION SYSTEME PARESSEUSE
# Si vous avez beaucoup d'utilisateurs et ne pouvez/voulez utiliser
# le champ 13 pour chaque utilisateur, vous pouvez faire correspondre
# automatiquement les logins mars_nwe aux logins des utilisateurs Linux
# MAIS il n'y a pour le moment aucun moyen d'utiliser les mots de passe
# des comptes Linux si bien que tous les utilisateurs configurés de cette
# façon devront utiliser le mot de passe fourni ici.
# Je vous déconseille de l'utiliser à moins que la sécurité ne fasse pas
# partie de vos préoccupations.
# Syntaxe :
#    15 <drapeau> <mot de passe commun>
# avec <drapeau> :
#    0  - ne configure aucun utilisateur automatiquement.
#    1  - configure automatiquement les utilisateurs non encore configurés.
#    99 - configure automatiquement tous les utilisateurs.

15 0 duzzenmatta

# VERIFICATION SIMPLE
# mars_nwe s'assurera automatiquement que certains répertoires existent
# si vous positionnez ce drapeau.
# Syntaxe :
#    16 <drapeau>
# avec <drapeau> à 0 pour non, ne fait rien et 1 pour oui, vérifie.

16 0

# QUEUES D'IMPRESSION
# Permet d'associer des imprimantes Netware à des imprimantes UNIX. Les
# répertoires de files d'impression doivent être crées manuellement
# avant de tenter toute impression.
```

```
# Syntaxe :
#    21 <Nom de queue> <répertoire de queue> <commande d'impression UNIX>

21  EPSON  SYS:/PRINT/EPSON lpr -h
21  LASER  SYS:/PRINT/LASER lpr -Plaser

# DRAPEAUX DE DEBOGAGE
# normalement, ils ne sont pas nécessaires, mais ils peuvent être utiles
# pour la résolution de problèmes.
# Syntaxe :
#    <élément de débogage> <drapeau de debogage>
#
#    100 = IPX KERNEL
#    101 = NWSERV
#    102 = NCPSERV
#    103 = NWCONN
#    104 = démarrage NWCLIENT
#    105 = NWBIND
#    106 = NWROUTED
#                 0 = supprime le débogage, 1 = active le débogage

100 0
101 0
102 0
103 0
104 0
105 0
106 0

# LANCER LE SERVEUR NETWARE EN TACHE DE FOND ET UTILISER UN FICHIER DE LOG
# Syntaxe :
#    200 <drapeau>
#        0 = Lance NWSERV en avant-plan et pas de fichier de log
#        1 = Lance NWSERV en tâche de fond et utiliser le fichier de log

200  1

# NOM DU FICHIER DE LOG
# Syntaxe :
#    201 <Fichier de log>

201  /tmp/nw.log

# AJOUT/ECRASEMENT DU FICHIER DE LOG
# Syntaxe :
#    202 <drapeau>
#        0 = ajouter dans le fichier de log existant
#        1 = écraser le fichier de log existant

202  1

# EXTINCTION DU SERVEUR
# Ce point modifie le délai entre l'affichage d'extinction du serveur et
# l'extinction réelle du serveur
# Syntaxe :
```

```
#    210 <temps>
#        en secondes (par défaut 10)

210  10

# INTERVALLE ENTRE LES DIFFUSIONS DE ROUTAGE
# la durée en secondes entre les diffusions (broadcast) de serveurs
# Syntaxe :
#    211 <temps>
#        en secondes (par défaut : 60)

211 60

# INTERVALLE ENTRE LES LOGS DE ROUTAGE
# configure le nombre de diffusions après lesquelles les informations
# de routage sont écrites dans les fichiers de log.
# Syntaxe :
#    300 <nombre>

300 5

# FICHIER DE LOG DE ROUTAGE
# configure le nom du fichier de log des informations de routage
# Syntaxe :
#    301 <nom de fichier>

301 /tmp/nw.route

# AJOUT/ECRASEMENT DU FICHIER DE LOG DE ROUTAGE
# configure si vous voulez ajouter les nouveaux logs à la fin du
# fichier existant ou bien le réecrire.
# Syntaxe :
#    302 <drapeau>
#    avec <drapeau> à 0 pour l'ajout et à 1 pour création/écrasement

302 1

# MINUTAGE DU CHIEN DE GARDE (watchdog)
# Configure les intervalles de temps entre les messages de garde pour
# s'assurer que le réseau est toujours vivant.
# Syntaxe :
#    310 <valeur>
# avec <valeur> =
#      0  - toujours envoyer des messages de garde.
#      <0 - ne jamais envoyer de message de garde.
#      >0 - envoyer des messages quand le trafic réseau chute en dessous
#           de 'n' ticks.

310 7

# FICHIER DE STATIONS
# configure le nom du fichier qui détermine les machines pour lesquelles
# ce serveur de fichiers sera le serveur primaire.
# La syntaxe de ce fichier est décrite dans le répertoire 'examples'
# du code source.
```

```
# Syntaxe :
#    400 <nom de fichier>

400 /etc/nwserv.stations

# OBTENTION DU SERVEUR DE FICHIERS LE PLUS PROCHE
# Configure la manière dont sont gerées les requêtes SAP 'obtention
# du serveur le plus proche' (Get Nearest Fileserver).
# Syntaxe :
#    401 <drapeau>
# avec <drapeau> =
#      0 : désactive cette requête
#      1 : le fichier de stations liste les stations à exclure
#      2 : le fichier de stations liste les station à inclure

401 2
```

Démarrer le serveur

Si vous avez configuré le serveur de telle sorte que d'autres programmes configurent le réseau et/ou fournissent les fonctions de routage, alors vous devez lancer ces programmes avant de démarrer le serveur. Si vous avez configuré le serveur pour qu'il configure vos interfaces et fournisse des services de routage, alors pour démarrer le serveur, il n'y a qu'à taper :

```
# nwserv
```

Tester le serveur

Pour tester le serveur vous devrez tout d'abord essayer de vous connecter et entrer par login grâce à un client Netware sur votre réseau. Vous pouvez alors faire un `CAPTURE` à partir de votre client et essayer d'imprimer. Si ces deux actions sont couronnées de succès, votre serveur fonctionne.

10.2 L'ensemble *lwared*

Ales Dryak <`A.Dryak@sh.cvut.cz`> a développé *lwared* pour permettre à Linux de fonctionner comme un serveur de fichiers basé sur NCP.

Lwared est l'abréviation de *LinWare Daemon*.

10.2.1 Possibilités offertes par *lwared*

Le serveur *lwared* est capable de fournir un sous-ensemble de services d'un serveur Novell NCP. Le système de messageries a été inclus mais il ne fournit pas de facilités d'impression. De plus il ne fonctionne pas bien avec des clients pour Windows 95 ou Windows NT. Ce serveur s'appuie sur des programmes extérieurs pour construire et maintenir le routage IPX et les tables SAP. Le mauvais comportement d'un client peut causer le "plantage" du serveur. Il est aussi important de savoir que la retranscription des noms de fichiers (minuscules/majuscules) n'a pas non plus été incluse.

Ce serveur fonctionne avec les shells Netware NETX et VLM.

10.2.2 Récupérer *lwared*

Le paquetage *lwared* peut être construit pour n'importe quel noyau supérieur à la version `1.2.0`, je vous recommande d'utiliser la version `1.2.13` pour laquelle aucun patch sur le noyau n'est nécessaire. Certaines fonctionnalités ont changé avec la version `1.3.*` du noyau, ce qui signifie que des patchs sont nécessaires pour ces versions afin de faire fonctionner ce serveur de manière correcte. Des patchs appropriés sont inclus pour les nouvelles versions de noyau, ainsi si vous devez absolument utiliser un de ces noyaux, vous serez toujours en mesure de faire fonctionner *lwared* correctement.

On trouve *lwared* sur

klokan.sh.cvut.cz `ftp://klokan.sh.cvut.cz/pub/linux/linware/` ,

Metalab `ftp://metalab.unc.edu/pub/Linux/system/Network/daemons` , ou sur un site miroir. La version à jour lors de l'écriture de ce document était : `lwared-0.95.tar.gz`

10.2.3 Compiler et installer *lwared*

Décompresser l'ensemble *lwared*

Ce qui devrait ressembler à :

```
# cd /usr/src
# tar xvpfz lwared-0.95.tar.gz
```

Compiler le noyau avec le support Ethernet et IPX

Si vous utilisez un noyau alpha `1.3.*` vous devriez essayer d'utiliser la version `1.3.17` ou une plus récente car les patchs fournis ont été faits pour celles-là. Les noyaux `1.3.*` plus anciens que `1.3.17` nécessiteront d'être modifiés à la main pour pouvoir installer lwared (*quelques indications sur la manière d'opérer ce genre de modifications sont données dans le fichier `INSTALL` du paquetage*). Pour installer les patchs pour la version `1.3.17` du noyau ou plus récent, vous devrez essayer la commande suivante :

```
        # make patch
```

La chose suivante que vous devrez faire après avoir appliqué le patch, le cas échéant, sera de vérifier que votre noyau a bien été compilé avec le support IPX actif. Dans la version `1.2.13`, vous n'aurez qu'à répondre oui (`Y`) à la question '`The IPX protocol`' de la manière suivante :

```
 ...
 ...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
 ...
 ...
```

Pour les noyaux plus récents, la question peut varier un peu, mais le procédé reste fondamentalement le même.

Vous devrez également vous assurer du choix approprié du pilote de votre carte Ethernet. Si vous ne savez pas comment procéder, je vous conseille vivement de vous reporter à

Ethernet-HOWTO `Ethernet-HOWTO.html` .

Vous pouvez maintenant procéder à la compilation de votre noyau. N'oubliez pas de lancer *lilo* une fois la compilation finie, pour que le nouveau noyau soit correctement installé.

Compiler et installer *lwared*

Pour compiler *lwared* vous devez en premier lieu vérifier et éditer si nécessaire le fichier `server/config.h`. Ce fichier contient les éléments nécessaires pour piloter votre serveur et déterminer son comportement lorsqu'il fonctionnera. Le fichier par défaut est raisonnablement bien écrit mais vous pourrez tout de même vérifier si les répertoires de fichiers de log et le fichier de configuration conviennent à votre système.

```
# make depend
# make
# make install
```

Le '`make depend`' se plaint de ne pas trouver le fichier `float.h` sur mon système mais cela semble fonctionner tout de même. J'ai aussi découvert que lors des compilations avec gcc `2.6.3`, il fallait modifier la ligne :

```
#include <net/route.h>
```

en

```
#include <net/if_route.h>
```

dans le fichier `lib/ipxkern.c`. Ceci étant dû au fait que le fichier change parfois de nom.

Le '`make install`' tentera d'installer le serveur et les démons de routage dans le répertoire `/usr/sbin`, le programme *lwpasswd* dans le répertoire `/usr/bin`, les programmes utilitaires IPX seront copiés dans le répertoire `/sbin` et les pages de manuel dans la structure de répertoire `/usr/man`. Si un de ces endroits ne correspond pas à votre système, vous devrez éditer et modifier les fichiers `Makefile` correspondants.

10.2.4 Configurer et utiliser *lwared*

Voici la partie la plus amusante !

Configurer le réseau IPX

La première chose à faire est de configurer vos cartes Ethernet pour qu'elles puissent supporter les réseaux IPX connectés à votre serveur. Vous devez connaître les adresses des réseaux IPX pour chacun des segments du réseau, quel périphérique Ethernet (`eth0`, `eth1`, etc.) est en relation avec quel segment, quel type de trame est utilisé pour chacun des segments (`802.3`, `EtherII`, etc.) et quelle adresse de réseau interne votre serveur doit utiliser (ce qui est vraiment nécessaire si votre serveur doit fournir des services sur plusieurs segments). Par exemple, une configuration plausible d'un serveur étant sur deux segments IPX ayant pour adresse réseau `23a91300` et `23a91301` et ayant une adresse de réseau interne `bdefaced` ressemblera à :

```
# ipx_internal_net add BDEFACED 1
# ipx_interface add eth0 802.3 23a91300
# ipx_interface add eth1 etherii 23a91301
```

Démarrer les démons de routage

Le logiciel du noyau permet de faire du renvoi de paquet IPX comme il le fait pour les paquets IP, cependant le noyau requiert un programme additionnel pour gérer la mise à jour des tables de routage. Dans le cas d'IPX, deux démons sont nécessaires et tous deux sont fournis avec *lwared* : *ipxripd* gère l'information de routage IPX et *ipxsapd* gère les informations SAP. Pour faire démarrer les démons vous n'aurez qu'à spécifier l'endroit où vous voulez qu'ils écrivent les messages de log :

```
# ipxripd /var/adm/ipxrip
# ipxsapd /var/adm/ipxsap
```

Configurer le serveur *lwared*

Deux fichiers sont à configurer manuellement pour que les utilisateurs puissent se loguer sur votre serveur. Il s'agit de :

`/etc/lwpasswd`

Il s'agit du fichier dans lequel les informations sur les comptes utilisateurs de LinWare sont gardées. Le programme *lwpasswd* permet de le maintenir à jour. Dans sa forme la plus simple, le fichier `/etc/lwpasswd` ressemble à :

```
ales:
terryd:
guest:
```

Son format est une simple liste de noms de login suivis du caractère ':' puis de la version encryptée du mot de passe de login. Il faut cependant noter deux choses importantes : si les mots de passe ne sont pas chiffrés cela signifie qu'il n'y a pas de mot de passe. D'autre part, les utilisateurs de *LinWare* doivent avoir un compte Linux, c'est-à-dire que chaque utilisateur de *LinWare* présent dans `/etc/lwpasswd` doit aussi être présent dans `/etc/passwd`. Seul l'utilisateur `root` peut changer le mot de passe d'un autre utilisateur *LinWare*. Si vous êtes logué en tant `root`, vous pourrez par exemple taper la commande suivante pour changer le mot de passe d'un utilisateur :

```
# lwpasswd rodg
Changing password for RODG
Enter new password:
Re-type new password:
Password changed.
```

`/etc/lwvtab`

Il s'agit des tables des volumes LinWare où sont stockées les informations sur les répertoires à partager entre les utilisateurs de LinWare (ce fichier comporte des similitudes avec le fichier `/etc/exports` pour NFS). Un exemple simple de son format est :

```
SYS          /lwfs/sys
DATA         /lwfs/data
HOME         /home
```

Ce format est simple : nom du volume puis le répertoire Linux à exporter séparés par le caractère espace. Vous devez avoir, **au moins**, une entrée dans ce fichier pour le volume `SYS` afin que le serveur puisse démarrer. Si vous voulez que vos utilisateurs DOS puissent accéder à votre serveur LinWare comme serveur primaire, vous devrez installer une arborescence de volume `SYS` standard sous le répertoire que vous exportez comme volume `SYS`. Les fichiers devant être présents dans ces répertoires sont ceux de Novell et sont soumis à leurs copyrights. Vous devez avoir une licence pour les utiliser. Si vos utilisateurs utilisent un serveur de fichiers Novell comme serveur primaire, vous n'aurez pas besoin du volume `SYS`.

Démarrer le serveur *lwared*

```
# lwared
```

Plutôt simple n'est-ce pas ? Mais il reste une question. Quel est le nom du serveur de fichiers ? Si vous avez démarré le serveur LinWare de la façon exposée ci-dessus, son nom sera celui retourné par la commande *hostname* de Linux (le nom d'hôte). Si vous voulez un autre nom, il suffit de le spécifier dans la ligne de lancement de lwared comme ci-dessous par exemple :

```
# lwared -nlinux00
```

Ici le nom du serveur LinWare sera `linux00`.

Tester le serveur *lwared*

La première chose à faire est de vérifier que votre serveur LinWare apparaît bien dans une *slist* d'un client DOS sur votre réseau. Le programme *slist* est stocké dans le volume `SYS` d'un serveur de fichier Novell donc il vous faudra réaliser cette opération à partir d'une machine déjà loguée sur un serveur. Si ça ne marche pas, vérifiez que *ipxsapd* et *lwared* tournent en même temps sur la machine Linux. Si la slist est bonne vous pouvez essayer de connecter et de mapper un volume :

```
C:> attach linux00/ales
...
...
C:> map l:=linux00/data:
C:> l:
```

Vous devriez maintenant être en mesure d'utiliser le nouveau volume comme tout autre volume Novell. Les permissions des fichiers sont basées sur celles données par le serveur *Linux* en fonction de votre compte utilisateur.

11 Configurer votre machine Linux en client d'impression Novell

Le paquetage *ncpfs* inclut deux petits utilitaires qui vous permettent d'imprimer depuis votre machine Linux sur une imprimante reliée à un serveur d'impression Novell. La commande *nprint* vous permet d'imprimer un fichier vers une queue d'impression NetWare. La commande *pqlist* liste les queues d'impression disponibles sur un serveur NetWare.

Pour obtenir et installer ces commandes, vous n'avez qu'à suivre les instructions relatives au client NCP décrites précédemment.

Ces deux commandes nécessitent que vous fournissiez le nom d'utilisateur et le mot de passe. Vous devriez normalement penser à écrire des scripts pour simplifier les tâches d'impression.

Exemple :

```
# pqlist -S ACCT_FS01 -U guest -n
# nprint -S ACCT_FS01 -q LASER -U guest -n filename.txt
```

La syntaxe concernant le login est similaire à celle de la commande *ncpmount*. Les exemples ci-dessus font l'hypothèse qu'un serveur de fichiers `ACCT_FS01` a un compte `guest` sans mot de passe, qu'une queue d'impression appelée `LASER` existe et que `guest` a le droit d'imprimer dessus.

Sur mes machines Linux, j'ai un court script en shell pour chaque imprimante Novell. Il peut alors être utilisé comme un filtre d'impression afin de pouvoir imprimer en utilisant le spooler Linux standard.

12 Configurer votre machine Linux en serveur d'impression Novell

Un programme permettant à votre machine Linux d'agir comme serveur d'impression sur un réseau NetWare est inclus dans le paquetage *ncpfs*. Pour l'installer et le compiler, se reporter à la section 'client NetWare' au-dessus. Il y a également un autre support dans le paquetage *mars_ nwe*.

12.1 Prérequis

La configuration est relativement simple mais vous devez avoir une imprimante complètement configurée et fonctionnant sous Linux. Cette partie est couverte dans le

Printing-HOWTO `Printing-HOWTO.html` en détail.

12.2 Configuration

Une fois que vous avez une configuration imprimante qui fonctionne et que vous avez compilé et installé l'utilitaire *pserver*, vous devez ajouter des commandes dans vos fichiers `rc`.

La commande exacte à utiliser va dépendre de ce que vous voulez exactement qu'il fasse, mais dans sa façon la plus simple, quelque chose comme cela devrait fonctionner :

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET
```

Cet exemple demande à l'utilitaire *pserver* de se loguer sur le serveur `ACCT_01` avec le nom `LASER` et le mot de passe `secret` et de récupérer les travaux d'impression dans la queue `LASERJET`. Lorsqu'un nouveau travail d'impression est reçu, la commande d'impression par défaut de *lpr* sera utilisée pour fournir ce travail d'impression au démon d'impression Linux. La queue d'impression doit être déjà définie sur le serveur de fichiers et le nom d'utilisateur doit avoir les privilèges du serveur pour la queue.

Vous auriez pu, si vous aviez voulu, utiliser une autre commande Linux pour accepter et imprimer les travaux d'impressions. l'argument `-c` vous permet de spécifier la commande exacte à utiliser. Par exemple :

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET -c "lpr -Plaserjet"
```

fera exactement la même chose que l'exemple précédent sauf qu'il enverra le travail d'impression à la configuration *printcap* `laserjet` au lieu de celle par défaut.

13 Un aperçu des commandes utilisateur et d'administration de *ncpfs*

Les dernières versions du paquetage *ncpfs* de Volker incluent toute une série de commandes d'utilisation et d'administration que vous pourriez avoir envie d'utiliser. Ces outils sont compilés et installés dans le processus d'installation de *ncpfs*. Si vous ne l'avez pas encore fait, suivez les instructions fournies dans la section sur le client Novell au-dessus pour les compiler et les installer.

L'information détaillée est disponible dans les pages de *man* fournies, mais voici un bref résumé des commandes.

13.1 Commandes utilisateur

ncopy

Network Copy (copie réseau) - permet de copier efficacement des fichiers en utilisant une fonction NetWare plutôt que de copier à travers le réseau.

nprint

Network Print (impression réseau) - permet d'imprimer un fichier vers une queue d'impression Netware située sur un serveur Netware.

nsend

Network Send (envoi réseau) - permet d'envoyer des messages à d'autres utilisateurs connectés à un serveur NetWare.

nwbols

List Bindery Objects (liste les objets de la bindery) - permet de lister le contenu de la bindery d'un serveur NetWare.

nwboprops

List properties of a Bindery Object (liste les propriétés d'un objet de la bindery) - permet de voir les propriétés d'un objet de la bindery

nwbpset

Set Bindery Property (Configure des propriétés de bindery) - permet de configurer les propriétés d'un objet de la bindery NetWare.

nwbpvalues

Print Netware Bindery Objects Property Contents (affiche le contenu propriété des Objets de la Bindery) - permet d'afficher le contenu d'une propriété d'une bindery Netware.

nwfsinfo

Fileserver Information (information sur le serveur de fichiers) - affiche un résumé d'informations sur un serveur Netware.

nwpasswd

NetWare Password (mot de passe NetWare) - permet de changer le mot de passe d'un utilisateur NetWare.

nwrights

NetWare Rights (permissions NetWare) - affiche les droits associés à un fichier ou répertoire particulier.

nwuserlist

User List (liste d'utilisateurs) - liste les utilisateurs actuellement logués sur un serveur de fichiers NetWare.

pqlist

Print Queue List (Liste de queue d'impression) - affiche le contenu d'une queue d'impression NetWare.

slist

Server List (liste de serveurs) - affiche une liste de serveurs de fichiers NetWare connus.

13.2 Outils d'administration

nwbocreate

Create a Bindery Object (créer un objet de la bindery) - permet de créer un objet de la bindery.

nwborm

Remove Bindery Object (enlever un objet de la bindery) - permet d'effacer un objet de la bindery.

nwbpadd

Add Bindery Property (ajouter une propriété à la bindery) - permet de positionner la valeur d'une propriété existante d'un objet d'une bindery Netware.

nwbpcreate

Create Bindery Property (créer une propriété de bindery) - permet de créer une nouvelle propriété pour un objet déjà existant dans une bindery NetWare.

nwbprm

Remove Bindery Property (enlever une propriété de bindery) - permet d'enlever une propriété sur un objet d'une bindery NetWare.

nwgrant

Grant Trustee Rights (attribuer les droits d'accès) - permet d'assigner les droits d'accès à un répertoire sur un serveur de fichiers NetWare.

nwrevoke

Revoke Trustee Rights (retirer les droits d'accès) - permet de retirer les droits d'accès à un répertoire sur un serveur de fichiers NetWare.

14 Configurer PPP pour le support d'IPX

Les nouvelles versions du démon PPP *pppd* pour Linux permettent de transporter des paquets IPX à travers une liaison série PPP. Il faut au moins la version `ppp-2.2.0d` du démon. Voir le *PPP-HOWTO* `PPP-HOWTO.html` pour savoir où le trouver. Lorsque vous compilez *pppd*, assurez-vous d'activer le support IPX en ajoutant les deux lignes suivantes au fichier `/usr/src/linux/pppd-2.2.0f/pppd/Makefile.linux`

```
IPX_CHANGE = 1
USE_MS_DNS = 1
```

Le `IPX_CHANGE` est ce qui active le support d'IPX dans PPP. La définition de `USE_MS_DNS` permet aux machines Microsoft Windows 95 de faire de la recherche de nom (name lookup).

Le vrai truc pour arriver à le faire marcher, c'est de savoir le configurer.

Il y a plein de façons de faire mais je ne décrirai que les deux sur lesquelles j'ai reçu des informations. Je n'ai encore jamais essayé, donc considérez cette section comme expérimentale et si vous arrivez à faire marcher quelque chose, faites-le moi savoir s'il vous plait.

14.1 Configurer un serveur IPX/PPP

La première chose à faire est de configurer votre machine Linux en serveur IP/PPP. Pas de panique ! Ce n'est pas difficile. De nouveau, suivez les instructions dans le *PPP-HOWTO* `PPP-HOWTO.html` et ça devrait aller. Une fois que vous avez fait ceci, il y a quelques modifications pour faire fonctionner IPX avec la même configuration.

14.1.1 Première étape

Une des premières choses à faire est de configurer votre machine Linux en routeur IPX de la façon décrite plus haut dans ce document. Vous n'avez pas besoin d'utiliser la commande *ipx-route* pour l'interface `PPP` car *pppd* les configure pour vous comme il le fait pour IP. Lorsque le démon *ipxd* tourne, il détecte automatiquement toute nouvelle interface IPX et propage les routes pour elle. Ainsi, vos machines connectées en dialup seront vues par les autres machines automatiquement lors de la connexion.

14.1.2 Conception

Lorsque l'on tourne en tant que serveur, c'est normalement de sa responsabilité d'assigner des adresses réseau à chacune des liaisons PPP lorsqu'elles sont établies. Ceci est un point important, chaque liaison PPP sera un réseau IPX et aura une adresse réseau IPX unique. Cela signifie que vous devez décider comment vous allez allouer les adresses et quelles adresses utiliser. Une convention simple est d'allouer une adresse réseau IPX à chaque périphérique série qui supporte IPX/PPP. Vous pourriez allouer une adresse réseau IPX basée sur le login de l'utilisateur connecté, mais je ne vois pas de raison particulière de faire ainsi.

Je vais supposer pour la suite que vous avez choisi la première convention et qu'il y a deux périphériques réseau (modems) que nous utiliserons. Les adresses utilisées dans cet exemple sont :

```
Periph. Adresse reseau IPX
------- -------------------
ttyS0   0xABCDEF00
ttyS1   0xABCDEF01
```

14.1.3 Configurer *pppd*

Configurez votre fichier `/etc/ppp/options.ttyS0` comme suit :

```
ipx-network 0xABCDEF00
ipx-node 2:0
ipxcp-accept-remote
```

et votre fichier `/etc/ppp/options.ttyS1` :

```
ipx-network 0xABCDEF01
ipx-node 3:0
ipxcp-accept-remote
```

Cela va demander à *pppd* d'allouer l'adresse réseau IPX appropriée au lien lorsque la liaison est établie. Le numéro de noeud local sera positionné à `2` ou `3` et le noeud distant pourra l'écraser avec le numéro de noeud qu'il pense être le bon. Notez que toutes les adresses sont en hexadécimal et que `0x` est obligatoire au début de l'adresse réseau mais pas nécessaire au début de l'adresse de noeud.

Il y a d'autres endroits où l'on aurait pu stocker cette information. Si vous avez seulement un modem pour les appels entrants, alors une entrée aurait pu être ajoutée au fichier `/etc/ppp/options`. Cette information aurait également pu être passée sur la ligne de commande de *pppd*.

14.1.4 Tester la configuration du serveur

Pour tester la configuration, vous aurez besoin d'un client fonctionnant correctement. Quand l'utilisateur appelle, se logue et que *pppd* démarre, celui-ci va attribuer l'adresse réseau, avertir le client du numéro de noeud du serveur et négocier le numéro de noeud du client. Lorsque ceci est terminé et après que *ipxd* a détecté la nouvelle interface, le client doit pouvoir établir des connexions IPX avec des hôtes distants.

14.2 Configurer un client IPX/PPP

Dans une configuration client, le fait de configurer ou non votre machine Linux en routeur IPX dépend de votre volonté d'agir en routeur IPX sur le réseau local de la machine. Si vous avez une machine isolée qui se connecte à un serveur IPX/PPP alors vous n'avez pas besoin d'utiliser *ipxd*. En revanche, si vous avez un réseau local et souhaitez que toutes les machines du réseau puissent utiliser la route IPX/PPP alors vous devez configurer et faire tourner *ipxd* comme décrit. Cette configuration est plus simple car vous n'avez qu'un seul périphérique série à configurer.

14.2.1 Configurer *pppd*

La configuration la plus simple est celle qui permet au serveur de fournir toute l'information sur la configuration du réseau IPX. Cette configuration serait compatible avec la configuration du serveur décrite au-dessus.

Vous avez à nouveau besoin d'ajouter des options dans votre fichier `/etc/ppp/options` :

```
ipxcp-accept-network
ipxcp-accept-remote
ipxcp-accept-local
```

Les options indiquent à *pppd* d'agir de manière complètement passive et d'accepter tous les détails de configuration fournis par le serveur. Vous pouvez fournir ici des valeurs par défaut pour les serveurs qui ne fournissent pas ces détails en ajoutant des lignes `ipx-network` et `ipx-node` similaires à la configuration du serveur.

14.2.2 Tester le client IPX/PPP

Pour tester le client, vous avez besoin d'un serveur à appeler connu pour fonctionner correctement. Une fois que vous avez appelé et que *pppd* a démarré, vous devriez voir les détails de votre configuration IPX sur votre périphérique `ppp0` lorsque vous utilisez la commande *ifconfig*. Vous devriez également être capable d'utiliser *ncpmount*.

Je ne sais pas si vous devrez ajouter les routes IPX manuellement pour atteindre des serveurs distants ou non. Il semblerait que oui. Si quelqu'un qui a cette configuration pouvait me le dire, je lui en serais reconnaissant.

15 Tunnel IPX sur IP

Beaucoup d'entre vous doivent être dans la situation où deux réseaux locaux IPX sont séparés par une simple connexion IP. Comment pourra-t-on dans ce cas jouer un deathmatch à DOOM pour DOS dans cette situation, demanderiez-vous ? Andreas Godzinap <`ag@agsc.han.de`> a une réponse pour vous avec *ipxtunnel*.

ipxtunnel fournit une sorte de pont pour IPX en permettant à des paquets IPX d'être encapsulés dans des paquets de données TCP/IP de manière à ce qu'ils soient transportés par une connexion TCP/IP. Ce programme attend l'arrivée de paquets IPX et lors de leur arrivée, il les enferme dans un paquet de données TCP/IP et le route vers l'adresse IP distante que vous aurez spécifiée. Pour que cela fonctionne, la machine qui reçoit le paquet de données encapsulé doit, elle aussi, faire tourner une version identique d'*ipxtunnel*.

15.1 Récupérer *ipxtunnel*

Vous pouvez obtenir *ipxtunnel* sur

Metalab `ftp://metalab.unc.edu/pub/Linux/system/network/daemons` ou sur un site miroir.

15.2 Compiler *ipxtunnel*

ipxtunnel se compile proprement chez moi en utilisant les commandes :

```
# cd /usr/src
# tar xvfz .../ipxtunnel.tgz
# cd ipxtunnel
# make
```

15.3 Configurer *ipxtunnel*

La configuration d'*ipxtunnel* est aisée. Admettons que la machine de vos amis soit `gau.somewhere.com` et que votre machine soit appelée `gim.sw.edu`. *ipxtunnel* utilise un fichier de configuration appelé `/etc/ipxtunnel.conf`. Ce fichier vous permet de spécifier le port UDP par défaut à utiliser pour la connexion TCP/IP par laquelle les données encapsulées seront envoyées ainsi que l'interface locale où ipxtunnel doit écouter et délivrer les paquets IPX.

Un simple fichier de configuration ressemblera à :

```
#
# /etc/ipxtunnel.conf pour gim.sw.edu
#
# Le port UDP à utiliser                (par défaut 7666)
port 7777
#
# Le nom de la machine distante où les paquets IPX doivent être envoyés
```

```
remote gau.somewhere.com
#
# Les interfaces IPX locales à écouter (par défaut eth0)
interface eth0
interface eth1
```

Évidemment l'autre machine aura un fichier de configuration similaire spécifiant votre machine comme hôte distant (`remote`).

15.4 Tester et utiliser *ipxtunnel*

ipxtunnel se comporte **comme** un pont IPX, ainsi les réseaux IPX à chaque extrémité de la liaison TCP/IP doivent probablement être les mêmes. Andreas n'a jamais testé *ipxtunnel* dans un environnement qui utilise réellement les serveurs de fichiers Novell, donc si vous l'essayez dans un tel environnement, donnez-en le résultat à Andreas pour qu'il sache si ça marche ou pas.

Si *ipxtunnel* fonctionne, vous devriez être en mesure de lancer DOOM sur les machines à chaque bout de la connexion utilisée en mode IPX et vous devriez vous voir l'un l'autre.

Andreas n'a testé ce logiciel que sur de bonnes lignes rapides et ne garantit pas sa performance si votre liaison est à faible vitesse. Une fois de plus, dites-lui ce qui marche et ne marche pas.

16 Support commercial IPX pour Linux

16.1 Caldera's Network Desktop

Caldera Inc. commercialise une distribution Linux possèdant de nombreuses fonctions et améliorations. Elle inclut un support totalement fonctionnel d'un client Novell Netware. La distribution de base est la respectée Red Hat Linux et Caldera y a ajouté son "Network Desktop". Le support Netware inclut un client Novell Netware possédant toutes les fonctionnalités requises, bâties sur une technologie licenciée par Novell Corporation. Ce client fournit un accès total à des serveurs de fichiers Novell 3.x et 4.5 et inclut des fonctionnalités comme le "Netware Directoy Service" (NDS) et l'encryptage RSA.

Vous pourrez obtenir plus d'informations et de détails sur le serveur web

Caldera Inc `http://www.caldera.com/` .

Si vous travaillez dans un environnement Netware 4.x et/ou NDS, alors le client Netware de Caldera est la seule solution disponible.

Si vous devez mettre en oeuvre une solution Novell pour Linux critique de manière professionnelle, vous devriez regarder le produit de Caldera de près.

17 Foire Aux Questions

Où puis-je trouver des logiciels IPX pour Linux fournis avec un support commercial ?

Caldera Corp. offre un client Netware 3.x et 4.x complètement supporté avec une licence complète. Pour de plus amples informations, consultez le site web de *Caldera Inc* `http://www.caldera.com/` .

Est-ce que les logiciels IPX fonctionnent avec Arcnet/Token Ring/etc. ?

Les logiciels IPX pour Linux fonctionnent avec les interfaces Arcnet et Token Ring. Je n'ai encore entendu personne essayer avec AX.25 . La configuration se passe de la même manière que pour l'Ethernet, sauf que vous devrez remplacer 'eth0' par le nom de périphérique approprié et mettre l'adresse matérielle appropriée lorsque nécessaire.

Comment est-ce que je configure plus d'une interface IPX ?

Si vous avez plus d'une interface dans votre machine, vous devez utiliser la commande *ipx_interface* pour configurer manuellement chacune d'entre elles. Vous ne devez pas utiliser la configuration "plug'n'play".

Comment est-ce que je choisis les adresses IPX ?

Les réseaux IPX sont similaires mais non identiques aux réseaux IP. Une différence majeure est la façon dont les adresses sont utilisées. IPX n'utilise pas le concept de sous-réseau. Le lien entre les adresses réseau et les réseaux est différent. Les règles sont relativement simples.

- Chaque adresse de réseau IPX doit être unique sur un réseau à grande échelle (WAN). Ceci inclut les adresses de réseau internes. Beaucoup d'organisations utilisant IPX sur des réseaux à grande échelle ont une politique d'adressage que vous devriez suivre.
- Chaque adresse d'hôte sur un réseau particulier doit être unique. Cela signifie que tous les hôtes sur chaque réseau IPX doivent avoir une adresse assignée unique. Dans le cas d'un réseau Ethernet, ce n'est pas difficile, car les cartes ont chacune une adresse unique. Dans le cas d'IPX/PPP, cela signifie que vous devez vous assurer d'allouer une adresse unique pour tous les hôtes du réseau, quel que soit le bout de la (des) liaison(s) auquel ils sont connectés. Les adresses des hôtes n'ont pas besoin d'être uniques à travers un WAN car l'adresse du réseau est combinée à l'adresse de l'hôte pour identifier l'hôte de manière unique.

Quels sont les types de trame que je devrais utiliser?

Il y a plusieurs types de trames au-dessus desquelles vous pouvez faire fonctionner IPX. Les plus utilisées sont décrites dans la section 'termes utilisés dans ce document' (sous l'entrée `'type de trame'`).

Si vous installez votre machine sur un réseau existant, alors vous devrez utiliser ce qui est déjà utilisé pour vous permettre de communiquer avec les autres machines du réseau. En revanche, si vous installez un réseau tout neuf, il y a toute une variété de protocoles pour transporter votre trafic IPX que vous pouvez utiliser. Si vous installez un réseau tout neuf et que vous envisagez d'y faire passer de l'IPX et de l'IP, je vous recommande des trames de type `Ethernet_II`.

Mes machines Windows95 brouillent la détection automatique du type de mes trames.

Apparement, ils y arrivent, ouais. Je pourrais faire des commentaires pas gentils, mais à la place, je vous suggérerais de configurer manuellement le type de trames plutôt que d'utiliser la configuration automatique. C'est probablement la meilleure façon en tout cas.

Pourquoi est-ce que j'obtiens le message 'invalid argument' quand je configure IPX ?

Vous n'avez probablement pas un noyau qui supporte IPX. Soit vous recompilez votre noyau pour qu'il le fasse, soit vous vérifiez que vous avez bien utilisé lilo pour installer et utiliser le nouveau noyau.

Pourquoi est-ce que j'obtiens le message 'package not installed' quand je configure IPX ?

Vous n'avez probablement pas un noyau qui supporte IPX. Soit vous recompilez votre noyau pour qu'il le fasse, soit vous vérifiez que vous avez bien utilisé lilo pour installer et utiliser le nouveau noyau.

Pourquoi est-ce que *pppd* m'envoie le message 'IPX support not in kernel' ?

Vous avez probablement compilé IPX comme un module et vous ne vous êtes pas assuré qu'il était chargé avant de démarrer *pppd*.

Comment exporter en NFS un système de fichiers NCP monté ?

Pour utiliser NFS pour exporter un système de fichiers NCP, vous devez le monter en utilisant l'option `-v` de *ncpmount*. Cette option permet de monter un seul volume d'un serveur de fichiers au lieu de les monter tous. Lorsque vous faîtes ainsi, le démon NFS vous permet d'exporter ce système de fichiers de la manière habituelle.

Pourquoi est-ce que slist ne marche pas quand j'ai un réseau interne avec *mars_ nwe* ?

Vous devez avoir la requête 'get nearest server' activée. L'entrée 401 dans `/etc/nwserv.conf` doit être 0, à moins que vous ayez une raison pour ne pas répondre à ces requêtes. Si vous voulez simplement que *slist* fonctionne et ne pas répondre à chaque requête 'get nearest server', ajoutez votre réseau interne et numéro de noeud dans `/etc/nwserv.stations` et positionnez l'entrée 401 de `/etc/nwserv.conf` à 2.

Est-ce que le paquetage ncpfs fonctionne avec mars_nwe ?

Les codes de Martin et Volkers commencent lentement à converger. Les versions récentes de *mars_ nwe* ont une option pour lui permettre de marcher avec *ncpfs*. Vous devez activer `WITH_NAME_SPACE_CALLS` dans le fichier `config.h` de *mars_ nwe*.

18 Copyright

19 Remerciements

Terry Dawson <`terry@perf.no.itg.telstra.com.au`> pour le document initial.

David E. Storey <`dave@tamos.gmu.edu`> et Volker Lendecke <`lendecke@namu01.gwdg.de`> qui m'ont assisté grandement en me fournissant des informations pour ce document. Gilbert Callaghan <`gilbert@pokey.inviso.com`>, David Higgins <`dave@infra.com`> et Chad Robinson <`chadr@brtgate.brttech.com`> pour leur contribution sur la configuration de IPX/PPP. Bennie Venter <`bjv@Gil-galad.paradigm-sa.com`> pour les informations utiles sur les types de trames. Christopher Wall <`vergil@idir.net`> pour ses suggestions utiles afin d'améliorer la lecture et l'agencement de ce document. Axel Boldt <`boldt@math.ucsb.edu`> pour ses suggestions utiles et le retour d'informations. Erik D. Olson <`eriko@wrq.com`> pour son retour d'informations utile et ses informations pour la configuration de PPP pour IPX. Brian King <`root@brian.library.dal.ca`> pour sa contribution à une question de la section FAQ

"NetWare" est une marque déposée de *Novell Corporation* `http://www.novell.com/` .

"Caldera" est une marque déposée de *Caldera Corporation* `http://www.caldera.com/` .

Kevin Thorpe

<kevin@pricetrak.com>